Lean Product Development process structure and its effect on the performance of NPD projects

Saeed Taheri

A thesis submitted in partial fulfilment of the requirements of Nottingham Trent University for the degree of Doctor of Philosophy

August 2017

ساحل افتاده گفت گرچه بسی زیستم
هیچ نه معلوم شد آه که من کیستم
موج ز خود رفته‌ای، تیز خرامید و گفت
هستم اگر می‌روم گر نروم نیستم

Abstract

New product development (NPD) has a pivotal role in the industrial competition, and makes a basis for long-term prosperity of companies. To survive in today's fast-changing market environment, companies are always trying to improve the performance of their NPD projects, by implementing new approaches, such as Lean Product Development (LPD). Nevertheless, applying such approaches is not straightforward, mainly due to the high level of interdependency between development activities and the role of dynamic effects in the project performance. Understanding the combined effects of dynamic features, including feedback loops, time delays and nonlinear causal relationships, is the main step for achieving higher project performance.

In this thesis, the dynamics of LPD process structure is investigated to find the ways it could affect the time, cost and quality performance of a development project. As there is no consensus about the definition of LPD among researchers in this filed, first through a comprehensive literature review different approaches to LPD are studied. Two major approaches for LPD are introduced based on the adaptation of lean manufacturing tools and techniques for optimizing NPD processes, or extracting LPD specific tools and techniques from Toyota Product Development System (TPDS). The second approach is proved to be more applicable, mainly due to fundamental differences between manufacturing and NPD environments, and the LPD process design based on TPDS is selected as the focal point for this research.

The combination of Set-Based approach to design and Concurrent Engineering in the form of SBCE is identified as the unique feature of LPD process structure which have been the topic of several researches in this field during past decade. Set-based design approach calls for the higher number of iteration cycles at the front end of the projects, and is responsible for higher project effectiveness while increases the time and effort invested. On the other hand, concurrent engineering targets the project duration, and is an efficiency factor, but if not structured properly it could have an opposite effect through increasing the number of rework cycles. Although the performance of TPDS which is the best benchmark for LPD shows the positive effect of SBCE on the projects performance, the reasons behind it and the way through which two approaches could be structured to achieve the favourable results is not clear yet. In addition, while different types of new product development projects, based on

their levels of complexity and innovation, are defined and executed in companies, it is not clear if SBCE approach has the same impact on all project types.

To investigate the reasons behind the superiority of SBCE and its effects in different types of development projects the systems thinking approach is selected as the main research methodology to provide a holistic view on the development projects through looking on interdependencies between performance measures and process structure. System dynamics modelling is used as the research method, due to its capacity in modelling feedback loops and iteration and rework cycles, as underlying factors which determine the time, cost and quality performance in projects. The model is built based on verified structures for rework cycle and resource allocation as the platform for the model, and becomes more specific for the purpose of this research by adding structures related to the iteration cycles, number of initial concepts, and effect of project type. After passing the standard system dynamics validation tests, the model is calibrated using the historical project data from different projects in a major car manufacturing company. The calibrated and verified model then used for the policy analysis by defining different scenarios based on the number of iteration cycles during the conceptual design phase, number of initial concepts and the type of project. All types of projects show the improved performance metrics when moving towards the SBCE approach by increasing the number of iteration cycle. However, the degree of improvement for projects with higher levels of complexity is more profound. In addition, it is concluded for projects with the high level of complexity that increasing the number of initial concepts has the positive effect on all project performance measures.

This research results have a methodological contribution by providing a method for rigorous representation of the impact of LPD process structure on projects performance through simulation. From the practical point of view, the developed model could be used by project managers as a guide for making informed decisions which guarantee the long-term success of development projects.

Acknowledgements

When I finished my Master's degree in Iran, I told myself that this would be my last official educational degree. This thesis manifests the different turn of events which wouldn't have been possible without several individuals to whom I am most grateful.

First and foremost, I would like to express my deep gratitude to my director of study, Professor Baback Yazdani. His guidance was invaluable, which enabled me to develop a strong understanding of the subject and supported me to complete this thesis. I owe him lots of gratitude for all he has done.

Secondly, I would like to express my heartfelt thanks to my second supervisor, Dr Michael Zhang, for his true support and inspiration to accomplish this work. I could have not completed my PhD without his help and support. Besides, I would like to thank my internal examiners, Professor Robert Ackrill, Dr Piers Thompson, and Dr Roy Stratton, and my external examiner, Dr Moray Kidd, for their valuable comments during the monitoring sessions, and my Viva.

Great appreciation goes to Nottingham Business School (NBS) for providing me with a fully-funded scholarship, and NTU Doctoral School for providing all facilities and support. I am particularly grateful to Ms Kim Keirnan, Ms Imogen Edwards, Ms Dawn James, and Ms Yasmin Malik for their very kind administrative support.

Furthermore, I would like to thank Dr Ehsan Sabet for all his moral support to start my PhD and all of my other friends who made my student life easier in Nottingham.

I owe special thanks to my parents, and my lovely sisters, Sima and Sara, to whom I will remain indebted for the rest of my life, for their love, support, guidance, and prayers.

Finally, I would like to say that without the love, care, tears, smiles, adventure, fun, distraction, needs, hope, trust, respect, encouragement, patience, and support of my wife, Farinaz, I really do not know how I would have completed this thesis. Thank you.

Table of Contents

1 Introduction 1

1.1 Research background 3

1.2 Research aims and objectives 5

1.3 Thesis structure 7

2 Literature Review 9

2.1 Introduction 11

2.2 Introduction to Lean Thinking 11

2.3 New product development 13

2.3.1 NPD performance 17

2.3.2 Value in NPD 18

2.3.3 Waste in NPD 21

2.3.4 Iteration cycles in NPD 26

2.4 The Foundation of Lean Product Development 31

2.5 Levels of lean implementation 35

2.6 Positioning the research among different classes of LPD research 38

2.6.1 A process-view of LPD 42

2.7 Designing NPD processes 43

2.8 Approaches to designing NPD processes 44

2.8.1 Point-based approach 44

2.8.2 Concurrent (simultaneous) engineering 46

2.8.3 Limitations of concurrent engineering 47

2.8.4 Research trends in concurrent engineering 49

2.8.5 Set-based approach 50

2.9 Research gap 55

2.10 Summary 56

3 Research Methodology 57

3.1 Introduction 59

3.2 Research philosophies and paradigms 59

3.2.1 Ontological position 59

3.2.2 Epistemological position 59

3.3 Research methods and tools 61

3.3.1 Principles of Systems Thinking 62

3.3.2 Modelling as a research method 65

3.3.3 The purpose of modelling 69

3.3.4 Simulation modelling 70

3.3.5 Types of simulation models 72

3.3.6 Simplicity vs. reality in simulation models 75

3.4 System Dynamics Modelling 75

3.4.1 Basic concept and assumptions of system dynamics 76

3.4.2 Philosophical standpoint of system dynamics 80

3.4.3 Modelling process 82

3.4.4 Application of system dynamics modelling in management research 87

3.5 Research design 90

3.5.1 Phase 1: Exploration 90

3.5.2 Phase 2: Development 90

3.5.3 Phase 3: Implementation 91

3.6 Methods employed for Data Collection 92

3.6.1 Literature review 92

3.6.2 Performance data related to NPD projects 93

4 Model construction 95

4.1 Introduction 97

4.2 Common tools for modelling of dynamic systems 97

4.2.1 Mental models 97

4.2.2 Causal Loop diagrams 99

4.2.3 Stock-And-Flow diagrams 100

4.3 Elements of system dynamic models 100

4.3.1 Stock variables 102

4.3.2 Flow variables 102

4.3.3 Auxiliary variables 102

4.3.4 Causal links and polarities 103

4.3.5 Delay 103

4.3.6 Nonlinear relationships 104

4.4 Software selection 104

4.5 The qualitative modelling stage 106

4.5.1 Defining the modelling purpose and the model boundary 106

4.5.2 Defining terms used in the model 108

4.5.3 Developing a causal loop diagram 113

4.6 The quantitative modelling stage 121

4.6.1 Introduction 121

4.6.2 Information-based view of NPD 121

4.6.3 An overview of the model 122

4.7 The system dynamics model 145

4.8 Summary 151

5 Model Validation 153

5.1 Introduction 155

5.2 The purpose of model validation 155

5.3 Tests for validating the model 157

5.4 Structural validity 158

5.4.1 Boundary Adequacy Test 158

5.4.2 Dimensional Consistency Tests 158

5.4.3 The structure assessment of the model 162

5.4.4 Extreme Conditions Analysis 173

5.5 Behaviour Reproduction Test 174
5.5.1 Data used for calibration 175
5.5.2 Optimization process 176
5.6 Summary 185
6 Analysis and Discussion 187
6.1 Introduction 189
6.2 Scenario analysis 189
6.2.1 The effect of the conceptual phase duration on the project performance 190
6.2.2 The effect of the number of concepts on project performance 202
6.3 Summary 230
7 Conclusion 233
7.1 Introduction 235
7.2 Summary of research 235
7.3 Revisiting research questions 239
7.4 Research Contributions 241
7.5 Research Limitations and suggestions for future research 243
References 245
Appendix 261

List of Figures

Figure 1: the generic NPD process (Ulrich and Eppinger, 2016) 15

Figure 2: definition of the NPD system in the context of the corporate enterprise system (Hoppmann, 2009) 16

Figure 3: Time share of several types of activities in NPD (Oehmen and Rebentisch, 2010)........ 21

Figure 4: interdependency between sources of waste 25

Figure 5: two levels of lean implementation........ 36

Figure 6: the sustainable Lean iceberg model (Hines et al., 2011) 37

Figure 7: Framework for classification of various aspects of lean implementation........ 38

Figure 8: different impacts of the NPD process on product cost (Holston, 2011) 44

Figure 9: Four modes of upstream-downstream communication (Wheelwright and Clark, 1992) 46

Figure 10: the effect of concurrency between phases on process time 47

Figure 11: Total design method (Pugh, 1991) 51

Figure 12: principles of SBCE (Nahm and Ishikawa, 2006) 52

Figure 13: The effect of SBCE on the process lead time compared to PBCE........ 53

Figure 14: Modelling vs real-world experiment (Borshchev and Filippov, 2004)........ 66

Figure 15: Model classification based on the level of abstraction (Bradley et al., 1977)........ 68

Figure 16: simulation modelling taxonomy........ 74

Figure 17: the position of system dynamics approach in relation to the hard and soft systems approaches........ 76

Figure 18: Exponential growth behaviour and its general structure (Sterman, 2000)........ 79

Figure 19: Goal seeking behaviour and its general structure (Sterman, 2000)........ 79

Figure 20: Shift from exponential growth to exponential decay (S-shaped behaviour) (Sterman, 2000) 79

Figure 21: three domains in critical realism 81

Figure 22: System dynamics modelling process by Richardson and Pugh (1981) 83

Figure 23: System dynamics steps by Forrester (1994b) 84

Figure 24: System dynamics modelling process by Sterman (2000) 85

Figure 25: Difference between planned and actual resource allocation to NPD projects (Lyneis and Ford, 2007) 88

Figure 26: Comparing the planned and actual project progress (Lyneis and Ford, 2007) 89

Figure 27: The role of causal loop diagrams and stock-and flow diagrams in building a system dynamics model 99

Figure 28: elements required for the representation of dynamic systems (Grösser and Schaffernicht, 2012) 101

Figure 29: a simple stock-and-flow diagram 103

Figure 30: the representation of positive and negative causal links, and feedback loops 103

Figure 31: the representation of a delay 104

Figure 32: Vensim software package 105

Figure 33: the model boundary 108

Figure 34: narrowly (A) and broadly (B) defined concepts in the design space for three product attributes X, Y, Z 112

Figure 35: terms used in the model and their relationships 112

Figure 36: SBCE loop 114

Figure 37: the goal seeking behaviour of the SBCE loop 114

Figure 38: the effect of the project complexity on the concept confidence 115

Figure 39: the schedule pressure loop 116

Figure 40: the rework loop 117

Figure 41: the concurrency loop 118

Figure 42: the corruption loop 119

Figure 43: the summay of the causal loop diagram 121

Figure 44: generic process structure of NPD 122

Figure 45: Simple stock-and-flow diagram for a single phase of the NPD project 124

Figure 46: variables controlling the rates of performing activities 125

Figure 47: iteration cycle 126

Figure 48: Uncertainty model 127

Figure 49: the relationship between confidence increase rate and the number of concepts 128

Figure 50: the relationship between confidence increase rate and the number of iterations 128

Figure 51: the relationship between confidence increase rate and the project complexity 129

Figure 52: the stock-and-flow model of design confidence 130

Figure 53: the relationship between the design confidence and the rework probability at each phase of the project........131

Figure 54: working in parallel on defined concepts in the conceptual design phase........132

Figure 55: the stock-and-flow model of the conceptual design phase........134

Figure 56: the stock-and-flow model of other phases of the project........136

Figure 57: modelling the schedule pressure........137

Figure 58: modelling the effect of the schedule pressure on other variables in the model........138

Figure 59: the model structure for desired resources........139

Figure 60: the model structure for the allocated resources........140

Figure 61: the model structure for the capacity for different activities........141

Figure 62: the model for calculating the duration of individual phases and the overall project........142

Figure 63: the model for calculating the cost of individual phases and the overall project........142

Figure 64: the model for calculating the percentage of rework for individual phases and the overall project........143

Figure 65: the behaviour of stocks in the workflow sector of conceptual design phase........164

Figure 66: changes in the level of confidence in the conceptual design phase for the simulation with one concept........164

Figure 67: changes in the level of uncertainty in the conceptual design phase for the simulation with one concept........165

Figure 68: iteration cycles during the detail design phase........165

Figure 69: change in the level of design confidence in the detail design phase after 50 months........166

Figure 70: the stocks behaviour in the conceptual design phase........167

Figure 71: the stocks behaviour in two initial iteration cycles in scenario 2........167

Figure 72: changes in the confidence and uncertainty level in conceptual design phase........167

Figure 73: the behaviour of stocks in the detail design phase........168

Figure 74: the behaviour of stocks in the tooling phase........168

Figure 75: the effect of concurrency on the behaviour of stocks........169

Figure 76: the effect of concurrency on the corruption rate........169

Figure 77: the rate of decrease in the level of uncertainty for two different project complexities........170

Figure 78: total number of iterations for two different project complexities........171

Figure 79: the effect of the schedule pressure on the target confidence and the reduction ratio....171

Figure 80: changes in the level of uncertainty and the number of iterations in the conceptual design phase by adopting the PBCE approach....172

Figure 81: drop in the level of schedule pressure by adopting the PBCE approach....172

Figure 82: the effort devoted to different phases of the project- Scenario 2....173

Figure 83: comparison between the total efforts devoted to projects with different levels of complexity- Scenario 2....173

Figure 84: results of extreme condition test....174

Figure 85: the NPD process design....175

Figure 86: Data related to the fraction of tasks reworked in four projects....176

Figure 87: calibration payoff function project time term....178

Figure 88: calibration payoff function phase time term....178

Figure 89: calibration payoff function cost term....179

Figure 90: calibration payoff function rework term....179

Figure 91: the combined model....180

Figure 92: Best fit results for project duration....182

Figure 93: Best fit results for project cost....183

Figure 94: Best fit results for rework....183

Figure 95: Best fit results for the finish time of conceptual design phase....184

Figure 96: Best fit results for the finish time of detail design phase....184

Figure 97: Best fit results for the finish time of tooling phase....185

Figure 98: Error values in calibrated data....185

Figure 99: the concept confidence and schedule pressure loops....191

Figure 100: effect of change in iterations number on duration of conceptual design phase in project A....193

Figure 101: effect of change in iterations number on the cost of conceptual design phase in project A....193

Figure 102: effect of change in iterations number on duration of conceptual design phase in all projects....194

Figure 103: effect of change in iterations number on cost of conceptual design phase in all projects....194

Figure 104: the effect of change in number of iterations on the duration of project A 195

Figure 105: the effect of change in number of iterations on the cost of project A 196

Figure 106: committed cost against time throughout the project (INCOSE, 2006) 197

Figure 107: relationship between rework percentage and concept confidence for different scenarios in project A 198

Figure 108: the effect of change in number of iterations on the duration of all projects 199

Figure 109: the effect of change in number of iterations on the cost of all projects 199

Figure 110: the effect of change in number of iterations on rework percentage in all projects 199

Figure 111: the total design confidence in project A for different number of iterations 200

Figure 112: the total design confidence and rework percentage in project B for different number of iterations 201

Figure 113: the total design confidence and rework percentage in project C for different number of iterations 201

Figure 114: the total design confidence and rework percentage in project D for different number of iterations 201

Figure 115: the effect of increasing the number of concepts on the duration of project A 204

Figure 116: the effect of increasing the number of concepts on the cost of project A 204

Figure 117: the effects of the number of concepts on the duration of conceptual design phase: Type 1 206

Figure 118: the pattern of uncertainty reduction during the conceptual design phase in three different scenarios 206

Figure 119: the effects of the number of concepts on the duration of conceptual design phase: Type 2 207

Figure 120: the effects of the number of concepts on the duration of conceptual design phase: Type 3 207

Figure 121: changes in the stock of 'Tasks Pending Decision' for three types of duration reduction in the conceptual design phase... 209

Figure 122: effect of the number of concepts on the conceptual design phase duration for project A 210

Figure 123: adding the expected behaviour graph to conceptual design phase duration for project A 210

Figure 124: number of iteration cycles in conceptual design phase for project A............................. 212

Figure 125: the effect of the number of concepts on the conceptual design phase cost for project A 214

Figure 126: the effect of increasing the number of concepts on corruption in downstream phases of project A 216

Figure 127: indirect effect of concepts number on the percentage of reworked tasks in project A .217

Figure 128: changes in duration and cost of detail design phase of project A due to concepts increasing 217

Figure 129: the effect of the number of concepts on the conceptual design phase duration for project B........219

Figure 130: the effect of the number of concepts on the conceptual design phase cost for project B 220

Figure 131: Effect of the number of concepts on the conceptual design phase duration for project C 220

Figure 132: the effect of the number of concepts on the conceptual design phase cost for project C 221

Figure 133: effect of the number of concepts on the conceptual design phase duration for project D 222

Figure 134: effect of the number of concepts on the conceptual design phase cost for project D ...222

Figure 135: changes in duration and cost of detail design phase of project B due to concepts increasing 223

Figure 136: changes in duration and cost of detail design phase of project C due to concepts increasing 224

Figure 137: changes in duration and cost of detail design phase of project D due to concepts increasing 224

Figure 138: changes in the duration and cost of project A in different scenarios........225

Figure 139: the percentage of increase in the cost of conceptual design phase between scenarios 2 and 10 for project A........225

Figure 140: changes in the duration and cost of project B in different scenarios........226

Figure 141: increase in cost of conceptual design phase between scenarios 2, 7 and 10 for project B 226

Figure 142: changes in the duration and cost of project C in different scenarios........227

Figure 143: increase in cost of conceptual design phase between scenarios 2, 6 and 10 for project C 227

Figure 144: changes in the duration and cost of project D in different scenarios 228

Figure 145: the percentage of increase in the cost of conceptual design phase between scenarios 2, 4 and 10 for project D........228

Figure 146: comparing the changes in projects duration in different scenarios........229

Figure 147: comparing the changes in projects cost in different scenarios......................................229

Figure 148: comparing the changes in projects rework in different scenarios.................................230

List of Tables

Table 1: Research questions aligned with objectives 6

Table 2: Thesis organization 7

Table 3: Process categories in an organization 13

Table 4: generic phases of NPD 15

Table 5: Several NPD Value Definitions in the Literature 19

Table 6: Different classifications of waste in NPD 24

Table 7: high-level practices for application of lean enterprise principles 36

Table 8: elements related to general lean thinking principles 39

Table 9: grouping LPD-specific elements and constructs 40

Table 10: classification of LPD elements and constructs 41

Table 11: operational level and process-related LPD elements extracted from Table 10 42

Table 12: effects of concurrent engineering on lead time 47

Table 13: SBCE steps and enablers (Kerga et al., 2014) 54

Table 14: Hard versus Soft systems thinking (Pidd, 1996; Checkland and Scholes, 1999) 64

Table 15: Types of models in management science (Ragsdale, 2008) 70

Table 16: research design 91

Table 17: Strengths and weaknesses of different types of publications (Khan, 2012) 93

Table 18: different frameworks proposed for the process of system dynamics modelling 99

Table 19: the presence of system dynamics elements in causal loop and stock-and-flow diagrams ((Grösser and Schaffernicht, 2012) 101

Table 20: System dynamics modelling software packages 105

Table 21: types of tests used for the validation of the model 158

Table 22: list of stock and flow variables in the model and their dimensions 159

Table 23: list of auxiliary variables in the model and their dimensions 160

Table 24: list of constants in the model and their dimensions 161

Table 25: dummy data used for the structural assessment test of the model 162

Table 26: Time and cost project data used for calibration of the model 175

Table 27: Variables used in the model..176

Table 28: classifying the model constants based on their dependency on the type of project 180

Table 29: different combinations of weight values used for calibration...181

Table 30: model constants and their estimated values...182

Table 31: scenarios for investigation the effect of the conceptual design phase duration................192

Table 32: the summary of the effect of SBCE approach on performance of projects........................202

Table 33: the slope of the phase duration graph 210

List of abbreviations

DMM	Domain Mapping Matrix
DSM	Design Structure Matrix
GERT	Graphical Evaluation and Review Technique
KPI	Key Performance Indicator
LPD	Lean Product Development
NPD	New Product Development
PBCE	Point-Based Concurrent Engineering
SBCE	Set-Based Concurrent Engineering
TPDS	Toyota Product Development System
TPS	Toyota Production System

1 Introduction

1.1 Research background

The pressure of stiff global competition, as well as the socio-technical advancements, have forced companies to address the requirements of ever-changing marketplace by acquiring necessary resources and capabilities to sustain their businesses. Because of increased global demand for faster introduction of cheaper and better products, and reduced products' lifespan, the long term prosperity of companies depends on their ability in new product introduction. From the start of the industrial revolution, when companies for the first time faced the need of developing new products for a particular market, traditional product development processes have been in use, which included the selection of a single design solution during the concept phase, and working on it throughout the entire process. Ever since their introduction, these NPD processes have been widely adopted and understood by manufacturing companies worldwide. Although by increasing the sophistication of products due to the increase in customers' demands, the complexity of these processes also evolved, the main principles of traditional product development processes remained unchanged.

By the end of the 1980s the engineering paradigm in manufacturing companies shifted from '*anything to enhance capability*' to '*better, faster and cheaper*' (Khan, 2012). This paradigm shift resulted in more focus on the incremental innovations through projects with shorter lead times and higher cost effectiveness. Despite all progresses made, there is still a long way to achieve the '*better, faster, and cheaper*' goal. Less investment in research and development, changes in customers' requirements due to technical advancements, adherence to rigorous standards and regulations, and economic crisis are among the main issues in recent years, which reduced innovation, impeded fast project lead times, and obstructed cost effectiveness.

New product development (NPD) could be defined as the transformation process of a market opportunity into a product which satisfies customers' needs, using a system of interrelated activities (Krishnan and Ulrich, 2001; Martínez León and Farris, 2011). To be able to survive in a fast-changing market environment companies are always seeking for new ways to increase the performance of their NPD processes in order to increase their profitability, market share and long term competitive advantage (Browning et al., 2002). However, changes in the competitive environment, including globalization, more sophisticated and demanding customers and rapid technological changes, have made it much more difficult. As the traditional NPD processes are mainly developed for stable markets with long products' life cycles, they are not capable for making companies competitive by fast and inexpensive introduction of high quality new products (Clark and Fujimoto, 1991). Consequently, firms have responded to these challenges by improving and re-engineering their new product development processes. New approaches such as Lean Product Development (LPD) have replaced traditional, functional-based systems to increase the performance of development projects. Lean principles first introduced by Womack et al. (1990) based on the comprehensive study on Japanese car manufacturing companies. Although historically related to the advanced management systems of Toyota in automotive industry, still there is not a common definition of LPD as some researchers even expanded it to include other improvement techniques which could result in faster development of a better product with less effort (Karlsson and Åhlström, 1996; Martínez León and Farris, 2011).

Lean thinking has been the subject of research for more than two decades, where the main focus has been on improving manufacturing processes, as well as administration, management and the supply chain. Nevertheless, there has been comparatively less research done to apply lean to new product development projects, despite the fact that NPD has the greatest influence on the profitability of products. One possible explanation for this could be unstructured and iterative approaches implemented in traditional new product development. As the result, research undertaken to improve NPD using lean principles is instrumental in the progress of engineering projects.

Implementing such new approaches in NPD processes, however, are not as simple as it first appears. The first reason is that LPD is still a new field of research and there is not a possible to find a single accepted definition for it in academic literature. It is due to the fact that lean concept is first introduced in the manufacturing shop floor, and some researchers solely attempt to adopt lean manufacturing tools for the NPD context, in spite of the fundamental differences between two environments. While there are other groups of researchers and practitioners who are trying to define LPD based on best-practices in NPD, with the focus on Japanese car manufacturing companies, and especially Toyota. In addition to this fact, in general restructuring the NPD processes have some unintended side-effects (Lyneis and Ford, 2007; Parvan et al., 2015). These side-effects are the result of high interdependencies between development activities which lead to the higher complexity of NPD projects and increase the role of dynamic effects on project performance (Ford and Sterman, 1998; Lyneis and Ford, 2007). The combined effects of dynamic features, including feedback loops, time delays and nonlinear causal relationships between project components result in project systems behaving in complex ways which are difficult to be clearly understood and managed by project managers (Ford and Sterman, 1998).

To provide a better understanding about the concept of LOD, and to provide a tool to show the practical effects of lean implementation in the context of new product development projects this research is defined to investigate the dynamics of Lean product development process structure and the way it impacts the project performance. Failure to understanding the impact of dynamic features on project performance results in failure to manage projects effectively, so it is the main step for improving the managers' mantel models and decision making process to achieve higher project performance. In order to achieve this goal, the focus of the research is first to have a definition of LPD as there is no single agreed-upon definition in literature, and then to develop and validate a simulation model and to use it to investigate different policies for having higher performance in NPD projects.

1.2 Research aims and objectives

The motivation for this research is three-folded: first, traditional approaches to NPD could no longer provide companies with enough power to survive in the highly competitive market environment. Second, using Lean principles as a new approach for organizing NPD processes has recently attracted the attention of researchers and practitioners, but there is no single-accepted definition for it in literature. And finally, there is no study to show the effect of implementation lean principles on the development projects' outcomes.

To satisfy these needs this research is defined with the aim to investigate the impact of the implementation of Lean in NPD on the performance of development projects. This aim is in response to the overall research question: '*how Lean product development process structure affects the performance of development projects?'* System dynamics modelling is used to make a model of LPD processes at the project level. System dynamics is considered an appropriate tool for this research due to its capability in modelling rework and iteration cycles as the underlying reasons for schedule slippage, cost overrun and quality problems in development projects (Lyneis and Ford, 2007).

The objectives of this research are defined based on the research questions as shown in Table 1. Four research objectives are as followed:

- Reviewing lean product development approaches and examining the current state of literature on the subject of LPD
- Extracting process-specific elements and components of LPD from literature

- Developing a process model through which the impact of LPD process structure of the performance of development projects can be studied.
- Validate and test the model using industrial data.

This research contributes to knowledge by giving a definition for LPD from the process viewpoint and showing how the combination of the process-specific features of LPD could contribute to the success of NPD projects. It also contributes to practice by helping companies in designing their NPD processes to achieve higher performance. To the best of the author's knowledge there is no model-based research which consider the combined impact of LPD process-specific features on the performance of NPD projects.

Table 1: Research questions aligned with objectives

General research idea: Implementation of Lean in New Product Development projects	
General focus research question: How Lean Product Development process structure affect the performance of development projects?	
Research question 1 What is Lean Product Development?	**Research objectives 1 and 2** Reviewing lean product development approaches and examining the current state of literature on the subject of LPD Extracting process-specific elements and components of LPD from literature
Research question 2 What is the impact of Set-based approach to design and concurrent Engineering as elements of LPD process structure on the performance of development projects? **Supplementary questions:** - How the duration of conceptual design phase affect the time, cost and percentae of reworks in different types of NPD projects? - How the number of initial concepts affect the time, cost and percentage of reworks in different types of NPD projects?	**Research objective 3 and 4** Developing a process model through which the impact of LPD process structure of the performance of development projects can be studied. Validate and test the model using industrial data.

1.3 Thesis structure

This thesis comprises of seven chapters which are structured according to the progression of the research conducted. An overview of the contents of these chapters is provided in Table 2. Each chapter starts with an introduction intended to help the reader understand the rationale behind the Chapter organisation. A summary is provided at the end of each chapters to help recap Chapter contents and sum up any salient points.

Table 2: Thesis organization

Chapter 1	Introduction	• Research background • Research aims and objectives • Thesis structure**Error! Reference source not found.**
Chapter 2	Literature Review	• Introduction to Lean Thinking • New product development • The Foundation of Lean Product Development • Levels of lean implementation • Positioning the research among different classes of LPD research • Designing NPD processes • Approaches to designing NPD processes • Research gap
Chapter 3	Research Methodology	• Research philosophies and paradigms • Research methods and tools • System Dynamics Modelling • Research design • Methods employed for data collection
Chapter 4	Model construction	• Common tools for modelling of dynamic systems • Elements of system dynamic models • Software selection • The qualitative modelling stage • The quantitative modelling stage • The system dynamics model
Chapter 5	Model Validation	• Tests for validating the model • Structural validity • Behaviour Reproduction
Chapter 6	Analysis and Discussion	• Scenario analysis
Chapter 7	Conclusion	• Summary of research • Revisiting research questions • Research Contributions • Research Limitations and suggestions for future research
Chapter 8	Appendix	• Model variables

2 Literature Review

2.1 Introduction

This chapter presented a review of literature related to LPD. After the introduction, this chapter starts with providing a review about the concept of lean, its origin, and principles in Section 2.2. As the research is in the context of NPD, Section2.3 gives a summery about NPD, its definition and performance measures, including two concepts of efficiency and effectiveness. Then, the value generation and the waste elimination, two important principles of lean thinking, are discussed in relation to the NPD processes, and following a review of literature about iteration cycles, a comprehensive classification of them in NPD is presented. The discussion about LPD is the topic of Section 2.4. Different approaches to LPD in literature are presented in this Section, and after reviewing the levels of lean implementation in Section 2.5, the research position among different approaches to LPD is justified in Section 2.6. In Section 2.7 contemporary approaches to the NPD process design by focusing on Point-based Concurrent Engineering, concurrent engineering, and Set-based Concurrent Engineering are discussed, in addition to the review of current research trends in these approaches. The chapter is finished by presenting the research gap in Section 2.9 and a summary of the chapter in Section 2.10.

2.2 Introduction to Lean Thinking

The term *Lean* was first introduced in an article published Krafcik (1988) in *MIT Sloan Management Review* to describe the *Toyota Production System (TPS)* as opposed to *Ford's* mass production system. During early 1990s, Womack et al. in their seminal book '*The machine that changed the world'* introduced the concept of *Lean Manufacturing* as the combination of principles and ideas based on the success of Japanese car manufacturing companies, especially Toyota to minimize the waste in processes while increasing the customers' value and continuously improving the processes (Womack et al., 1990). This shared the hidden secret behind TPS for the first time with other companies outside Japan. Satisfactory results of implementing lean manufacturing in western companies was a driving force for attempts to adapt this concept in other disciplines, even outside manufacturing industries, especially after the publication of "Lean thinking" by Womack and Jones (1996) in which they proposed the lean principles applicable to the whole enterprise. Five principles of lean thinking are briefly explained below:

- The first principle of lean is to specify value from the point of view of the final customer. Customers' value could be defined as how the customer perceives the product or service offered by the company. Any feature of a product or service which is not requested by the customer is waste, so to eliminate wastes it is essential to accurately identify value.
- The second principle of lean is to identify the value stream of a product or service and eliminate all wasteful steps which do not create any value for the customer. Value stream could be defined as the set of all specific process steps and actions necessary for bringing a specific product or service to the customer.
- The third principle is to let all value-adding activities to flow without any interruption. This enables companies to discover and solve problems which interrupt the flow by taking quick corrective actions. Continuous flow results in lower process lead-time, and cost, thus increasing the efficiency in the system.
- The forth principle is to let the customer pull the value by accurately responding to the demand of the customer. It means that upstream processes should produce a product or service just when is asked by the customer at downstream (Womack and Jones, 1996).
- Finally, the fifth principle is to continuously improve the system by restarting this process again to endlessly identify innovative ways to increase the value provision, reduce the costs of non-value adding activities and eliminate wastes.

Lean manufacturing has found a great acceptance in the production environment (Liker, 2004) as a new way to provide skills and a shared means of thinking to design in a systematic process, and have been successfully applied to managing manufacturing companies, considering any activities that create value and eliminate waste (Haque and Moore, 2004). The results have been providing opportunities to improve the production time and cost, and thus, the production efficiency (van der Krogt et al., 2010). In addition, Lean implementation has allowed companies to work more smartly with continuous improvements and has driven growth of the manufacturing companies with marginal benefits (Oosterwal, 2010). However, to be more effective for manufacturing companies, lean implementation should not stop at the shop-floor, otherwise companies almost never form a true learning culture in their processes. Assuming lean manufacturing as the first step in lean management, the implementation of lean principles in NPD processes would be a true advancement for it. LPD could be described as an incremental progression in the journey of lean thinking. Ward, et al. was the first team of researchers who realised fundamental flaws in conventional NPD processes and introduced the first approach to LPD through in-depth study of *Toyota Product Development System (TPDS)* and what they coined the second Toyota paradox (Sobek et al., 1999; Ward, 2007). LPD is now considered to be the new area for the journey of lean thinking, and researchers and practitioners are making significant efforts to develop models, tools, and methodologies for implementing it. However, LPD is a new research area and still in its infancy, and more endeavour is needed for developing a holistic best practice in LPD.

2.3 New product development

Before going through the concept of LPD it is first necessary to have a clear definition of the NPD system. According to Hines et al. (2011) to ensure that the organization remains competitive it is needed to design and manage key business processes efficiently and effectively. Key processes as defined by Hines and Taylor are “patterns of interconnected value-adding relationships designed to meet business goals and objectives” (Hines and Taylor, 2000: P15). Finding key processes to the core business and the way they could be designed and optimized to deliver higher value are fundamental in discussion about processes. Murman et al. (2002) and Hines et al. (2011) divided all processes in an organization into three groups. Table 3 represents these classifications, their definitions, and examples of key processes in each category.

Table 3: Process categories in an organization

Classification by Hines et al. (2011)	*Classification by Murman et al. (2002)*	*Definition*	*Example of key processes*
Strategic processes	*Leadership processes*	*Processes which focus on the overall direction of the organization and have a critical role in the transformation of an enterprise to lean*	-Strategy and policy deployment -Strategic planning -Organization structure and integration
Core Processes	*Lifecycle processes*	*Processes which are directly involved in delivering the target results and creation of products and services to customers, and directly contribute to revenue generation*	*-Production* *-New product development* *-Supply chain management* *-Distribution and support*

Support processes	*Support processes*	*Processes which are directly involved in delivering the target results by supporting core processes*	*-Human resource management* *-Finance* *- IT*

NPD as one of the core processes, especially in manufacturing companies, as shown in Table 3, is the main source of competitive advantage for companies, as successful firms have the capability of introducing valuable products with a faster rate and cheaper than their competitors (Machado et al., 2014). Ulrich and Eppinger defined NPD as "*a set of activities beginning with the perception of a market opportunity and ending in production, sale, and delivery of a product*" (Ulrich and Eppinger, 2016, p. 2). The goal of manufacturing companies is to quickly provide customers with products at low cost and based on their identified needs. Achieving these goals is the result of cross-functional cooperation between several functions inside a company, from marketing to engineering and manufacturing. Success of a manufacturing company, defined by their ability to produce new products and sell them profitably, is the direct consequence of their high-performance NPD processes and could be assessed using criteria such as product quality, product cost, development time, development cost and development capabilities (Ulrich and Eppinger, 2016). As an interdisciplinary activity, NPD calls for the contribution of almost all functional groups in a company, yet *Marketing*, *Engineering,* and *Manufacturing* have the central roles. Identification of a gap in the market which could potentially be filled by introducing a new product and fulfilling the customers' needs is the responsibility of the marketing department. The engineering department is accountable for defining the physical form of the new product based of what customers define and finally, the manufacturing department has the responsibility of designing a production system and operating it to produce the new product.

The NPD as a process consists of a set of sequential or sometimes parallel activities, mostly intellectual rather than physical, for picturing, designing, and commercializing a product (Ulrich and Eppinger, 2016). A well-defined NPD process enables companies to be assured about the quality of the final product by accurately timing of development phases and positioning the milestones. In addition, as a 'Master-Plan' it defines team members' contributions to the project and the roles they should play in the team, while also lets managers to identify the improvement areas by comparing the actual progress of the project against it. Developing new product is also a project job, done by a project team which consists of a team leader, a small core team and an extended team where in case of complex products, such as an aircraft, could involve thousands of members. Based on the complexity of the project, it could take from one to ten years to develop a new product. Number of people involving in NPD projects and the duration of the project have a direct impact on the cost of project.

The NPD process could be viewed in several ways. It could be an information-processing system which receives inputs in different forms of information, processes them with a number of activities, and finally, provides outputs which contain all information required for supporting the production and sale of the new product. NPD process could also be a risk management system, starts with the identification of uncertainties and risks, continues while eliminating uncertainties and reducing risks and completes when the project team has a defined level of confidence of the product and its performance in the market. The generic NPD process as Ulrich and Eppinger (2016) conceptualized consists of six phases and milestones as shown in Figure 1 and described briefly in Table 4.

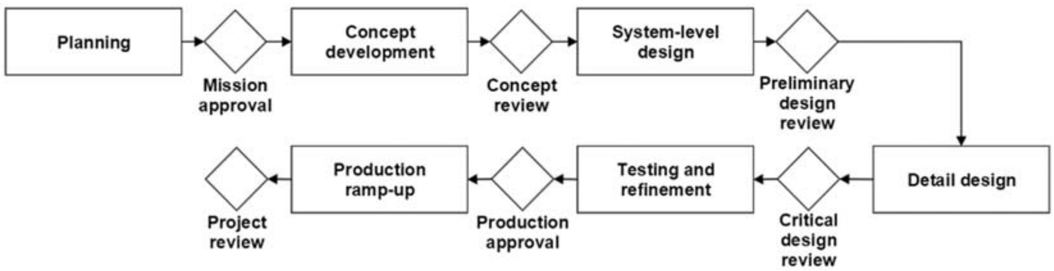


Figure 1: the generic NPD process (Ulrich and Eppinger, 2016)

Table 4: generic phases of NPD

Phase	*Main activities*
Planning	*Opportunity identification guided by company strategy*
Concept development	*Generating and evaluating alternative product concepts*
System-level design	*Defining the product architecture, decomposing the product into subsystems and preliminary design of key components*
Detail design	*Preparing the complete specifications of geometry, materials, and tolerances for all the product components, designing tools*
Testing and refinement	*Evaluating different versions of product*
Production ramp-up	*Making the product using the intended production system, training the workforce and finding remaining problems*

Several value streams are distinguishable in a company based on what they want to deliver to the customers. For example, in the production, the goal of value stream is to create physical value for the customers, whereas in NPD the focus is on generating and exchanging information. According to Hoppmann (2009) four value streams are identifiable in companies which develop and manufacture new products. As shown in Figure 2, two vertically oriented value streams are related to the delivery of products to the customers, including generating an order by marketing department which triggers the production, and product transformation flow parallel to that to make sure that the right product is delivered to the right customer at the right time. Although having these value streams makes companies able to satisfy the customers' demands, it is not sufficient in a competitive environment, in which technological and market changes dictate the need for continuous updating of products and processes. Horizontally oriented value streams, which are defined for this purpose, include product and process definition and production ramp-up. They act with different rates compared to the vertical value streams and define the flow of information about the products to customers, and the flow of materials, including all requirements for the testing and prototyping, to achieve a seamless innovation of products and processes (Hoppmann, 2009). The aggregation of these two horizontal value streams is defined as the NPD system, and includes all necessary activities for generating and documenting the needed information for the successful production of physical products (Wheelwright and Clark, 1992; Ulrich and Eppinger, 2016).

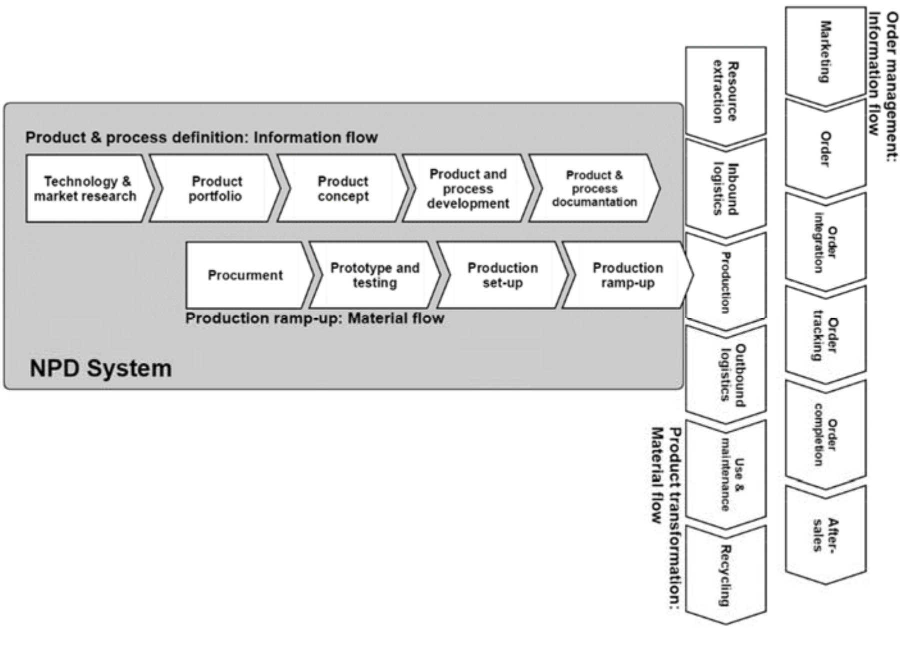


Figure 2: definition of the NPD system in the context of the corporate enterprise system (Hoppmann, 2009)

2.3.1 NPD performance

By viewing NPD as a transformation process which uses allocated resources to produce a definite output, its performance could be translated as the efficiency and effectiveness of the purposeful action (Neely et al., 2005). As Tangen (2005) mentioned most researchers agree that the efficiency is input-oriented and related to the internal performance of a process while the effectiveness is about the results, thus output-oriented, and is linked to the external performance. Efficiency as defined by Neely et al. (2005) is the measure of economically utilization of a firm's resources to provide a certain level of customer satisfaction. On the other side, effectiveness of NPD processes is about the extent to which customer requirements are met and thus, very difficult to be quantified in most cases (Neely et al., 2005). Unlike efficiency, effectiveness is about the outcomes, results, and the ability of a firm to reach a desired objective or the degree to which desired results are achieved (Tangen, 2005). It is possible for an efficient system to be ineffective, and for an effective system to be inefficient, thus it is the combination of high values of efficiency and effectiveness in a transformation process, such as NPD, which leads to higher achievements.

Performance measures should be dynamic to be industrially applicable (Yazdani, 2000). Clarks and Fujimoto (1991) identified lead time, productivity, and total product quality as performance dimensions based on the long-term competitiveness of NPD processes. Lead time is a key component in a time-based strategy, and has become increasingly important for managing NPD processes in a fast-changing business environment (Chen et al., 2010). Productivity is the level of resource consumption, including engineering hours worked, and the cost of equipment, services, and materials used mainly for prototyping and testing, which are required to take the project from the concept to the commercial product. Total product quality is the degree of match between objective characteristics of the product such as the design quality, and customers' expectations (Clark and Fujimoto, 1991). It is also under the influence of subjective evaluations such as the degree of match between the final design and the defined concept from the point of view of aesthetics, style, and experience (perceived quality). Lead time and productivity are measures of the efficiency in the NPD processes, while total product quality is a measure of the effectiveness. Smith and Reinertsen (1998) stated that there are four objectives, namely development speed, product cost, product performance and overall development expenditure, which should be measured for managing NPD processes. There are trade-offs between these objectives, which should be modelled based on specific company conditions and the economic balance, to allow managers to make dynamic decisions at the project and the company level.

2.3.2 Value in NPD

As discussed in Section 2.2, identifying value is the most important principle of lean thinking. Therefore, this section provides an explanation of value in the context of NPD.

Value is a multidimensional concept with quantitative and qualitative aspects that may be difficult to attribute to certain features of a product, system, or item (Gudem et al., 2011). Some scholars emphasised on having a precise definition of value for a successful NPD and the waste reduction (Womack et al., 1990; Haque and Moore, 2004; Baines et al., 2006). However, the definition of value in NPD processes is different compared to the manufacturing operations, because in NPD the value streams are in the form of information and knowledge which tracking them is much harder than material flow in the manufacturing (McManus and Millard, 2002). In addition, according to Siyam et al. (2015) key differences between value in NPD and manufacturing contexts are:

- the distinction between value and waste is ambiguous in NPD processes;
- the mix of necessary and unnecessary wastes is unclear. In other words, waste in NPD is not necessarily associated with performing unnecessary activities, but could be the result of activities with wrong inputs, where inputs might change later as the result of emerging new knowledge about the design. It is a common case in concurrent engineering and is discussed with more detail in this chapter;
- establishing unambiguous measurements of value in NPD is difficult, because of unavailability of firm thresholds for the satisfaction of stakeholders (Siyam et al., 2015).

While the emphasis in defining of the value is often on the customer value, customers in NPD are not just end users, but include a range of stakeholders along the product transformation process, from suppliers, to operations divisions, maintenance, and recycling (Hoppmann, 2009). These stakeholders sometimes have complex and conflicting expectations. For instance, manufacturing asks for the ease of production, service and maintenance asks for the ease of disassembly and reuse, and the end user asks for the high functionality of the product. These factors make difficult to have a definition for value in NPD.

There are many definitions and perspectives on value as shown in Table 5. But, as each definition has its own purposes and there are multiple perspectives on NPD, value definitions appear inconsistent. According to Zeithaml (1988), value is a cognitive trade-off between benefits and sacrifices, and customers perceive it based on their perception of what is gained and what is given. In the context of NPD, Browning et al. (2002) described the concept of customer value from the viewpoint of performance risk. According to them, the certainty and confidence of companies about their products is of direct value to customers, because costs of uncertainty are directly passed along to customers. Browning (2003) defined value as the ratio of the performance of the system to costs of the system. Similarly, Olaru et al. (2008) defined the customer value as a function of trade-off between benefits achieved, including product quality, services received and relationships developed, and sacrifices made, which are mostly financial. This definition stated the relationship between value and waste in NPD processes, where reducing wastes results in increasing the value, through lowering costs. In other words, the value in NPD processes is what would remain after wastes are eliminated, which is according to the activities classification, by Womack and Jones (1996) into three categories of value-added, necessary value-added, and unnecessary value-added. This is a reasonable definition, as the identification and quantification of wastes in a process in much more straightforward than the value.

Table 5: Several NPD Value Definitions in the Literature

Terminology	Author	Value definition
Value of product or service	Gale, 1994	Market perceived quality adjusted for the relative price of the product/service, where market perceived value is customer's opinion of product/service relative to the competitive market
Earned value	PMI, 1996	Value is defined as the sum of financial values of intermediate project deliverables. Used as a measure of project performance, comparing the value of work planned with the value delivered.
Value according to value engineering	Park, 1998	Value is provided by product, service or system functions that incur a given cost to provide. Cost may be reduced through a structured process considering alternative methods to provide those functions.
Value to employee	Donovan et al., 1998	A function of both compensation the employee receives from the company as well as job quality

Value to shareholder	Slack, 1999	The potential for future sales and profits of the realized product
Value in PD	Walton, 1999	The right information/product delivered at the right time to downstream process/customers where it is quantified by form, fit, function, and timeliness
Value to organizational stakeholder	Mills et al.2006	Value is associated with the trade-offs between what each stakeholder gets and what they have to give up, in terms of benefits obtained and sacrifices made, and the resources consumed
Project value	Zhai, 2009	The explicit and implicit functions of the project, which can satisfy all needs of stakeholders, such as time, cost, quality, commercial interests, social benefits, and technological development.
Value in lean system engineering	Oppenheim et al., 2011	Complex system value is associated with satisfying all stakeholders, which implies a flawless product, delivered with minimum cost, and the shortest possible schedule.
PD value	Siam et al. 2015	The degree to which a capability satisfies all relevant stakeholders, is delivered to them according to product or service quality, cost, and timeliness requirements, and is developed by performing effective and efficient processes that design and produce the satisfying capability within their budget and time constraints.

Several lean practices have been developed by researchers to assist value creation in NPD, many of them are not new, but have been adopted because they are thought to help achieving lean principles. In addition, frameworks which bring together these practices and fit them into a holistic system have also been discussed in literature. Siyam et al. (2015) provided an overview of these frameworks, among them they mentioned set-based framework, proposed for the reduction of reworks and iterations in NPD processes through the early identification of knowledge gaps in the process and not processing until they are all closed. Further in this chapter, LPD frameworks are discussed briefly, while emphasising on this lean value creation method in NPD which is the focus of this research. Before that, s value creation and waste elimination are two inseparable principles of lean thinking, it would be helpful to consider wastes in the context of NPD processes.

2.3.3 Waste in NPD

Based on principles of lean thinking (Womack and Jones, 1996), understanding the sources of wastes in processes is one of the main steps for implementing lean. Haque and Moore (2004) stated that to have a lean system it is necessary to eliminate wastes and unnecessary actions and to link value creating steps in a continuous sequence. Gudem, et al. (2011) in the same way defined lean as an operational philosophy whose goal is to maximize customer value while minimizing wastes. The starting point of a lean system, according to Gudem, et al. (2011), is separating value-added activities from wastes by focusing on customers. In all these definitions, the emphasis is equally on enhancing value while finding wastes and eliminating or minimizing them to provide the flow of value-added activities in the system.

In NPD, the overall value-added time is less than 30 percent in most companies (Welo and Ringen, 2016). Oppenheim (2004) reported that about 60-90 percent of time in NPD processes is estimated to be wasted. Similarly, Oehmen and Rebentisch (2010) indicated that about two third of the time of an NPD project is wasted because activities are idle, while, on average, only 12 percent of the time is spent on value adding activities (Figure 3). This data, although only a rough estimation, shows the large room for improvement in NPD processes only by attacking the wastes.

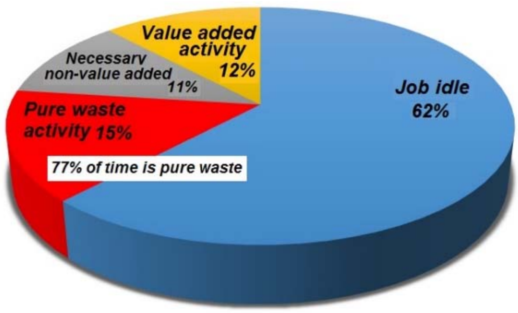


Figure 3: Time share of several types of activities in NPD (Oehmen and Rebentisch, 2010)

However, in a different approach, Browning and Sanders (2012) emphasised that lean in NPD is about maximizing value of the entire system, not just minimizing wastes. Reinertsen (2007) indicated that focusing on flow is the only way to release the potential of lean in NPD processes to simultaneously improve cycle time, quality, and efficiency. By increasing the pace of flow, development teams receive the information and feedback much quicker. It also increases the efficiency by detecting defects more rapidly which not only eliminates their recurring, but also reduces wasting resources by working on defective information.

Reinertsen (2007) also claimed increasing innovativeness by emphasising on fast feedback loops as the result of improved flow. However, reaching to a fast flow without identification and elimination the sources of wastes in NPD processes would be pointless. In fact, flow and waste are two sides of the same coin, and improving the flow is not possible without systematic attacking the sources of waste in the process.

Ohno, the father of TPS, categorized seven types of wastes in manufacturing activities, and based on his taxonomy, different researchers tried to define waste types in NPD processes. Oehmen and Rebentisch (2010) prepared a comprehensive summary of studies done on waste in NPD as shown in Table 6 and explained below:

- Overproduction is producing more than what the next stage in the process is needed, or earlier than is expected. The main reason for this type of waste could be unsynchronized processes (Morgan and Liker, 2006), lack of communication and coordination, and insufficient standardization of processes (Oehmen and Rebentisch, 2010). Unsynchronized information delivery results in building the inventory of information as the next stage in the process is not able to immediately start using it because of the lack of capacity or missed complementary information needed for starting the work. Anything that disrupt the cadence of the process, whether rework because of making defects or planning deficiencies, could result in overproduction.
- Waiting for receiving information or allocation of resources to start an activity, are the most common types of wastes in NPD processes (Morgan and Liker, 2006). Waiting could be due to the lack of discipline in scheduling, unexpected changes in design requirements, low process performance, tight and inflexible schedules, or the result of other types of wastes which interrupt the process flow (Oehmen and Rebentisch, 2010). Reintersen (2009) stated that the main reason for the long project time is not slow activities, but is waiting that makes queues and inventory in the process and destroys the quality and efficiency.
- Miscommunication is the result of unnecessary handoffs because of changes in the ownership of information, and leads to the loss of momentum and accountability in the process.
- Information Inventory could be the direct result of overproduction. In a balanced process with uninterrupted flow the amount of inventory would be zero. According to Oehmen and Rebentisch (2010) the size of information inventory could be an indicator of the quality of the process as it shows the higher amount of

synchronization in the process. High variability in the process as the result of lack of standardization, or elevated level of capacity utilization leads to the variability in processing time and unsynchronized stages of the process which cause information inventory. Although the information inventory in the NPD processes is invisible, it could be detected through its effects on the cycle time or delayed feedback on defects (Reinertsen, 2009).

- Unnecessary movement of people to find their required information, or to attend in unnecessary meetings.
- Over-processing is principally about generating information beyond the required specifications.
- Defect generation is the result of the lack of quality, outdated information in the inventory, or, as Oehmen and Rebentisch (2010) mentioned, the result of the defective information characteristics. Defect generation leads to waiting and rework as other types of waste. Time pressure because of approaching deadlines could be the main cause of defects as people take shortcuts instead of established processes, and quick-fixing instead of root-cause analysis of defects. The probability of using defective input or obsolete information in the presence of time pressure also increases (Oehmen and Rebentisch, 2010).
- Rework improves defective information to meet the requirements and quality standards.

Table 6: Different classifications of waste in NPD

(Ohno, 1998) (in Oehmen and Rebentisch 2010)	Millard (2001) (in Oppenheim, 2004)	(Bauch 2004)	(Kato, 2005) (in Oehmen and Rebentisch 2010)	(McManus 2005)
Over-production	Over-production: Creating Unnecessary information	Over-production Unsynchronized processes	Over-production of information	Over-production
Processing itself	Over-processing: Working more than	Over-processing Re-invention	Over-processing Re-invention	Over-processing
Transportation	Transportation: Inefficient transmittal of information	Transport Handoff	Transportation of information Hand-offs	Transportation
Stock on hand	Inventory: Keeping more information than needed.	Inventory	Inventory of Information	Inventory
Making defective products	Defects: Insufficient quality of information	Defects	Defective information	Defective product
			Rework	
Time on hand	Waiting: For information, data, etc	Waiting	Waiting of people	Waiting
Movement	Unnecessary movement: People having to move to gain access to information	Movement	Motion of people	Unnecessary movement

(Morgan and Liker 2006)	(Pessôa, 2008) (in Oehmen and Rebentisch 2010)	(Anand and Kodali 2008)	(Oehmen and Rebentisch 2010)
Over-production	Over-production	Over-production or early production	Over-production of information
Processing	Over-processing	poor process design	Over-processing of information
Conveyance	Transportation	Transportation or movement	Miscommunication of information
Inventory	Inventory	Unnecessary inventory	Stockpiling of information
	Defects	Defects	Generating defective information
Correction	Correction		Correcting information
Waiting	Waiting	Waiting or delays	Waiting of people
Motion	Motion	Unnecessary motion or inefficient performance of design	Unnecessary movement of people

Morgan and Liker (2006) stated that these types of wastes in NPD go in the category of *Muda* (non-value-added), while there are also *Muri* (overutilization of resources) and *Mura* (unevenness in workflow) as two other categories of wastes which although interrelated to Muda, almost always are neglected by companies. The interdependency between these three types of wastes, as shown in Figure 4, creates a loop in which the higher amount of unevenness in the process creates Muda, eliminating Muda is accompanied by increasing the level of capacity utilization, thus overburdening resources, and creating more unevenness. As Morgan and Liker (2006) truly indicated the challenge is continuously managing an overburden and uneven system to create a continuous unimpeded flow in the NPD processes from the concept to the market.

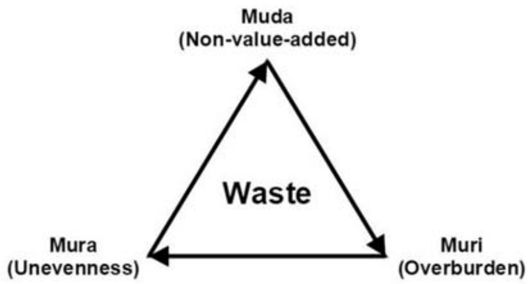


Figure 4: interdependency between sources of waste

To find the impact of value creation methods and wastes in NPD processes Siyam, et al. (2012) used a Domain Mapping Matrix (DMM) method based on a literature survey. Among value methods they considered the integration of suppliers and customers, Set-based Concurrent Engineering (SBCE), cross-functional teams, pull and flow. According to the analysis of the dependencies between value methods and waste types, they argued that waiting, iteration, defects and over-processing of information are the most critical types of wastes, and most value methods are trying to eliminate them. Among value methods they mentioned flow as the most challenging one to be implemented in NPD processes, because it could be disrupted by various sources of wastes, such as iteration, over-production, over-processing, and defects.

All innovative processes, including NPD and design, are inherently iterative (Lévárdy and Browning, 2009; Meier et al., 2015; Wynn and Eckert, 2016). However, as Ballard (2000) stated, a part of iteration cycles in NPD could be eliminated without losing any value. As the result and based on the definition by Womack and Jones (1996), this portion of iteration cycles could be assumed a type of waste. Ballard reported the result of informal surveys of NPD teams which estimated up to 50 percent of the time of the project is wasted by these iteration cycles. This trend shows a high opportunity for improvement in the NPD projects provided that wasteful iterations could be identified and eliminated. This is very critical because the value is also created in NPD mainly by iterations. In the rest of this Section, a comprehensive review of literature about iteration is provided which leads to having a classification of iteration cycles and the ability to identify value-added and wasteful iterations.

2.3.4 Iteration cycles in NPD

The importance of iteration cycles is well-acknowledged in NPD and design research. Understanding the source and nature of iterations in the projects enables managers to find solutions which increase the speed of the project, whether by adding more resources to execute iterations faster, or by restructuring the processes to reduce the number of iterations, thus pivotal in the managing and improving NPD projects (Smith and Eppinger, 1997a). According to Yassine and Braha (2003) iteration is unavoidable due to inability of design teams to consider all problems confronting a design decision in advance, impossibility of computing a design directly from a set of requirements, and unpredictability of the design performance. Iterations could happen in the processes because of different internal and external causes (Le et al., 2010). Technological and market uncertainties because of introducing modern technologies and changes in customer's requirements are among external causes of iteration cycles. Causes internal to the company could be product complexity, interdependency between tasks because of interdependency between product components (Smith and Eppinger, 1997b), arrival or discovery of new information (Smith and Eppinger, 1997b), rapid learning and experimentation (Kennedy et al., 2014), information transfer and converging on a satisfactory solution (Meier et al., 2015), overlapping sequential phases of the NPD in concurrent engineering (Krishnan et al., 1997; Terwiesch and Loch, 1999), and finally, finding errors and defects (Ford and Sterman, 1998). These causes are discussed with mode detail later in this section.

In their descriptions of iteration cycles, researchers considered NPD processes from different viewpoints and focused more on issues relevant to their objectives. They also referred to iteration using different terminologies, including rework, feedback loop, churn, and loopback (Wynn and Eckert, 2016). In this section, first a review on literature focused on iteration in NPD project is conducted to see different insights developed about this concept by researchers. Afterwards, a more comprehensive classification for types of iteration, developed by Wynn and Eckert (2016) by integrating ideas extracted from literature on iteration, is presented. This classification helps to clarify the possible ways to explain the iteration, and help to recognize its important characteristics.

Iteration cycles are necessary for integrating information which emerges during the NPD process. This approach provides flexibility and allow innovative new products to be developed in a market that changes rapidly (Wynn and Eckert, 2016). In this regard, Iansiti and MacCormack (1997) highlighted iteratively overlapping the conceptual design and detail design phases, emphasising on the necessity of avoiding committing early to a solution. In a similar way, according to MacCormack et al. (2001), to avoid costly reworks late in the project, it is required to iteratively search around the main sources of uncertainty in NPD projects, while Terwiesch and Loch (1999) found that if design specifications are frozen early in the project iteration cycles would have higher negative impact on the project lead time.

Empirical research about iteration in NPD projects could not find a universal relationship between iteration and project performance, implying that the impact of iteration on a project possibly depends on situation-specific factors (Wynn and Eckert, 2016). For example, while, according to Eisenhardt and Tabrizi (1995), a process might be accelerated by increasing the number of iteration cycles, Terwiesch and Loch (1999) reported a positive correlation between the number of iterations and the project duration. Also Love (2002) concluded from his study on construction projects that the amount of rework does not explain schedule overruns in projects, although more than 50 percent of budget overruns are related to rework.

Not only is the iteration influenced by managerial approaches, but also characteristics of the design and the complexity associated with interconnected design problems could impact iteration, and thus, the project performance. In this regard, Yassine et al. (2003) showed the increasing effect of having higher density of pairwise task coupling in a design on the amount of iteration which ultimately causes instability in the NPD process. Similarly, Loch et al. (2003) indicated that the convergence of an iterative process is less likely and more time-consuming when the number of coupled tasks increases. Eckert et al. (2004) highlighted the increase in rework costs as the result of higher levels of integration between the components of the product. MacCormack et al. (2001) claimed that the flexibility at the architecting stage of a project results in lower rework costs, later. In their case study on TPDS, Liker and Morgan (2006) found that product standardization at Toyota, through using shared architecture, and modularity, results in lower variability across designs, and reduces iterations. However, Engel and Reich (2015) stated that higher interface complexity due to excessive modularity leads to more iteration cycles.

In all mentioned studies about iteration in NPD projects, the focus of research was on some factors, while in the real-world, a network of causal relationships influences the iteration behaviour (Wynn and Eckert, 2016). Following a Systems Thinking approach, Love and Edwards (2004) concluded that a complicate and complex network of factors contributes to the rework behaviour and to decrease their impact, the entire system of interactions among factors should be considered. Le et al. (2012), similarly indicated the presence of the network of causal relationships which influences the iteration effects.

Concurrent engineering and SBCE, two approaches related to LPD, have also been present in studies about iteration. In the context of concurrent engineering, researchers focused on the impact of overlapping dependent activities on the iterative behaviour in NPD projects (Wynn and Eckert, 2016). In the context of SBCE, the focus of researchers has been on describing the impact of this approach, as observed in Toyota, on reducing the negative effects of iteration. These topics are explained separately later in this chapter.

There are several taxonomies for types of iterations, proposed by researchers. Although used different terminologies, all these taxonomies are based on dividing iterations between two groups of value-adding and wasteful iterations. Terms such as creative/dysfunctional, intentional/unintentional, positive/negative, planned/unplanned, progressive/corrective, and creative/superfluous have been used by researchers for the classification of types of iteration cycles in NPD projects (Ballard, 2000; Yassine et al., 2003; Le et al., 2012; Browning and Eppinger, 2002). For example, Browning and Eppinger (2002) used terms *planned* and *unplanned* iterations. Planned iterations are executed purposefully to create value in form of useful information for the new product by going through cycles of design exploration and convergence, whereas unplanned iterations are caused by revisiting prior decisions due to new learnings in development teams which invalidate previous assumptions, making critical decisions early and before having needed knowledge, making decisions which are technically infeasible or difficult to be achieved (Kennedy et al., 2014), and errors (Lévárdy and Browning, 2009). Using several views on iterations in NPD projects, presented by researchers, Wynn and Eckert (2016) provided a classification to organize different types of iterations, based on what is achieved or is intended by iterating tasks, activities, or aspects of a development project. Their classification is based on the distinction between positive and negative iterations. They also added a third class to include ideas about the combined positive and negative effects of iteration. Following, these three classes are discussed briefly.

2.3.4.1 Progressive iteration (planned iteration):

This class of iteration cycles involves revisiting issues in an NPD project considering the information and knowledge created during design and problem-solving activities. Many researchers have a consensus that this class of iterations could not be entirely avoided during an NPD project (Wynn and Eckert, 2016), so they could be anticipated in an NPD project design and are allocated with the required time and resources. The uncertainty in the problem and solution definition, due to the lack of knowledge, and bounded rationality of decision makers are among factors which contribute to the inevitability of progressive iterations.

Different objectives have been mentioned in researches for this class of iteration cycles. The first objective is to explore the design space and to iteratively discover, structure and address emerging issues, because of initially ill-defined goals and activities in a creative problem-solving process, such as NPD. Adding more details to the design as the ambiguity about the design reduces progressively due to creasing more information about it (Costa and Sobek, 2003), is the second objective of progressive iterations. The third objective is to optimize and adjust details and parameter values at a certain level of design definition, when the results of tests or manufacturing constraints become clear. In forth position is excessive refinement of the design or enhancing secondary characteristics of the design, while primary objectives are met, in cases when additional time is available. Finally, the last objective of progressive iterations is repeating a task to incrementally arrive at a desired goal. The main difference between the last objective and other four is that here, a task or a process is reiterated on various aspects of design, instead of revisiting the same aspect with different levels of details or using newly arrived information (Wynn and Eckert, 2016)

2.3.4.2 Corrective iteration (Unplanned iteration)

The second class of iteration cycles happens when additional information reveals problems in design. Although corrective iterations could still have positive effects, especially when the changes add value to the design, they are generally undesirable. The objectives of this class of iteration cycles, as Wynn and Eckert (2016) indicated are three-folded. The first objective is to address issues which were not initially recognized and need to meet requirements in a different way. Rework is the second and the most cited objective of corrective iterations. Using Kennedy et al. (2014: P279) definition, rework "occurs when a prior decision that was assumed to be final for that project is changed because it was later found to be defective". Arising additional information about design problems, and arriving wrong information at the wrong time, due to poor process architecture (Browning et al., 2002), are among the causes of rework. As rework cycles are mostly unpredictable (Browning and Eppinger, 2002) they consume a significant share of allocated time and money to NPD projects (Le et al., 2010), and often cause delay in project completion and budget overrun (Le et al., 2012). It is reported that between 30-80 percent of development capacity in a project are consumed by reworks (Reichelt and Lyneis, 1999; Terwiesch et al., 2002; Chucholowski et al., 2012; Kennedy et al., 2014). The final objective of corrective iterations is revisiting a problem repeatedly, because each attempt to solve the problem creates another set of problems. The reason for this is the complexity of design, exogenous changes, imperfect testing, and oscillatory allocation of resources due to firefighting (Yassine et al., 2003)

2.3.4.3 Coordinative iteration

This type of iteration cycles is associated with SBCE, and process concurrency, and happens to make the process more efficient and effective by reducing the amount or the risk of more costly iteration elsewhere in the process. The main objectives of coordinative iterations are to repeat parts of upstream work to communicate updates, or revisiting parts of the downstream works for accommodate successive changes in concurrent engineering, or providing enough information to decide between several alternatives in SBCE. In addition, coordinative iterations are done to enable feedback to make sure the project is completed on time, budget, and quality, and to let each team involved in an NPD project to understand other participants' goals and constraints.

For the sake of simplicity, in the rest of this research the '*Iteration'* and '*Rework*' terms are used as representations for value-adding and wasteful types of iteration cycles. In practice, because iterations are so common in NPD projects, most managers consider all types of them unavoidable and put their efforts mainly on firefighting and treating the symptoms instead of addressing the root causes to eliminate wasteful iterations, while by having a different approach to project management and development organization most of them could be avoided. However, some researches have been conducted about the mitigation of negative effects of iterations on the performance of NPD projects. For instance, Krishnan et al. (1997) proposed a way for determining the optimum amount of concurrency between interdependent process stages, based on the characteristics of exchanged information between them, to reduce reworks. Similarly, Terwiesch and Loch (1999) discussed how the design uncertainty and the level of dependency between stages of the process affect the time-saving properties of concurrency. In another work, Roemer et al. (2000) talked about the trade-off between the amount of time saved as the result of concurrent engineering and the cost incurred on the system. Ward, et al. (1995b) and Sobek et al. (1999) suggested the application of SBCE to reduce reworks arising from issues in the large-scale engineering design. The combined effect of concurrent engineering and SBCE on the NPD performance through their impact on the amount of iterations and reworks in development projects is the focus of this research and is discussed in more detail in the rest of this chapter.

2.4 The Foundation of Lean Product Development

The lean concept recently has gone beyond the manufacturing, where it has been traditionally developed and applied, to other contexts such as service industries (Morgan and Liker, 2006). One of the critical disciplines in companies, which plays a vital role in their success in today's turbulent and competitive world, is NPD. However, NPD processes cannot be managed exactly as manufacturing processes since they are certainly different subjects from the viewpoint of tasks, people involved, and processes. It is necessary to know how and to what extent lean principles can be applied to help improve NPD efforts in companies without any negative impact on their ability of innovation and creativity. This challenge has been the focus of an emerging field of research and practice about LPD (Letens et al., 2011). Despite of nearly two decades of research about LPD, it still suffers from lack of a single accepted definition (Martínez León and Farris, 2011). Little has been written and found to provide an answer to the issue of whether an adaptation of the Lean approach to the management of knowledge-intensive and innovative processes such as NPD is feasible and desirable. As Letens et al. (2011) stated the overall agreement on LPD definition is that it uses a system of engineering and work organization principles mostly benchmarked from Toyota, with the goal of attaining shorter lead-times, reduced cost, and higher quality than traditional NPD approaches.

The number of studies about LPD so far is quite limited (Martínez León and Farris, 2011). In a systematic literature review, Martínez León and Farris (2011) found that the underlying motive for research about LPD is to increase the competitive advantage of companies by improving their NPD processes using Lean philosophy. Although having this common aim, due to the lack of a single definition for LPD, researchers have had different approached to this concept. To be able to review the current state of research it is necessary to classify these approaches based on the contribution made by researchers. As the term Lean first used for the process improvement approaches used by Toyota in its manufacturing operations (Womack et al. 1990), one approach assumed the transferability of lean principles from manufacturing to NPD processes (Cooper and Edgett, 2005; Anand and Kodali, 2008; Gautam and Singh, 2008; Reinertsen, 2009; Beauregard et al., 2011; Nepal et al., 2011; Wang et al., 2012). In this approach, the focus has been on adapting various constituents of lean manufacturing which make sense in NPD, especially, the identification and elimination of sources of wastes, based on the Ohno's definition of waste in manufacturing (Oehmen and Rebentisch, 2010), to increase the efficiency and effectiveness of NPD processes. In some cases, lean manufacturing has also been mixed with other theories and approaches to ensure the relevance of proposed LPD approach (Khan et al., 2013).

However, there is not any consensus on the sources and types of wastes in NPD. While Haque and Moore (2002) identified three categories of wastes, namely strategic, organizational, and operational, Oppenheim (2004) emphasised on poor leadership and planning, and bureaucracy, and Ward (2007) specifically focused on knowledge wastes. On the other hand, Browning (2003) argued that LPD is more about maximising value rather than eliminating waste. Other researchers in this area also tried to adopt some lean manufacturing tools, such as value stream mapping (McManus and Millard, 2002; McManus, 2005), pull and just-in-time (Smith and Reinertsen, 1998; Reinertsen, 2009).

In the second approach, the argument is that applying lean concepts to engineering operations, such as NPD, where work is less repetitive than the manufacturing and the product is less tangible, is not straightforward (Liker and Morgan, 2006). In this approach, instead of just adopting tools and techniques from the manufacturing, the identification of a more comprehensive set of principles and mechanisms, directly related to NPD and leverage the benefits of Lean principles, from TPDS, has been at the centre of attention (Ward et al., 1995a; Sobek et al., 1999; Morgan and Liker, 2006; Ward, 2007; Hoppmann et al., 2011; Liker and Morgan, 2011; Khan et al., 2013).

In addition to the mentioned classification, using the research themes and approaches which have been used so far in literature it is possible to categorize research about LPD into five different areas; namely *Knowledge-Based Network*, *Strategy*, *Decision Making*, *Performance*

Measurement, and *Process Modelling*. Studies in Knowledge-based network area has more focused on the organizational learning, and assumed it as the main goal in implementing LPD. According to Nonaka and Takeuchi (1996) one of the factors that helps Toyota to outperform its western counterparts in developing new products is the creation of organizational knowledge. In addition, Ward (2007) argued that one of the important principles in LPD is the ability of companies to transform data into the usable knowledge. In order to achieve these goals, different approaches have been proposed, such as job rotation between functional groups and projects (Nonaka and Takeuchi, 1996), implementing set-based concurrent engineering (Sobek et al. 1999), and sub-networks within larger learning networks (Dyer and Nobeoka, 2000).

In strategy area, the focus of research has been mainly on how companies could best execute a set of projects over time, instead of concentrating of a single project. Product-platform strategy, proposed by Robertson and Ulrich (1998), is about sharing the components between different product platforms to increase the speed of development and manufacturing while offering a range of products to the market. Another strategic approach is multi-project management introduced by Cusumano and Nobeoka (1998). They classified projects according to the number of shared components and technologies among them, and based on this classification, proposed a strategy to coordinate functional groups across development projects. Both mentioned strategies are emphasising on standardization and reutilization, which as Martínez León and Farris (2011) stated are associated with set-based approach as one of the principles of LPD. In addition, these strategies emphasise on the importance of coordination mechanisms between functional groups across different projects to synchronize them and make cadence which result in uninterrupted flow in development processes.

Identifying the main categories of decisions in NPD and defining the way decisions should be made based on LPD are the focus of research in decision-making area. The type and content of decisions and the selection of optimum decision methods based on their contents are fundamental factors influencing the overall effectiveness of decision making processes. Decisions could be divided based on whether they are made in a single project (operational decisions), or between different projects (strategic decisions) (Krishnan and Ulrich, 2001). Operational decisions could be related to various stages of a single project, from concept development to production ramp-up, while strategic decisions are mainly about the product strategy, planning, the organization of product development, and managing projects

(Krishnan and Ulrich, 2001). However, these decisions are interrelated and need a coordinated approach across functional groups. In another study, Ward (1995a) emphasised on the importance of delaying decisions at the outset of projects to have a better view on alternative designs before selecting the best one, because these early decisions as Clark and Fujimoto (1991) mentioned have the highest impact on the efficiency and effectiveness of the project. The role of chief engineers as decision-makers in Toyota (Sobek et al., 1998), and the way that conflicts between chief engineers and functional managers are solved through standard methods, tools and coordination mechanisms have also been the subject of research in this area (Martínez León and Farris, 2011).

Research in performance measurement area has been about the metrics which should be measured in LPD and defining input factors which result in higher performance in LPD processes, instead of relying on outcome metrics, such as lead time, cost, and quality. Among these input factors are heavyweight project managers, concurrency of activities, strong supplier relationship (Cusumano and Nobeoka, 1992; Ward et al., 1995a; Sobek et al., 1998), upfront assessment of customer requirements (Ward et al., 1995a; Cooper and Kleinschmidt, 2007), having multifunctional teams, and having a formal NPD process (Griffin, 1997). On the other hand, researchers such as Oppenheim (2004), Browning and Ramasesh (2007), and Ward (2007) emphasised on the importance of the quality of information, as the interim deliverables which create value in LPD processes. To assess the performance of lean application in product development, Al-Ashaab, et al. (2013) designed a qualitative tool based on Balanced Scorecards. They used Knowledge, continuous improvement, value focus and set-based approach as fundamental enablers for LPD, and using these tools tried to compare the current and desired situation and identify the areas of improvement.

Finally, the main goal behind studying LPD from the process perspective has been to optimize the process structure, with the assumption that the NPD process structure could affect its performance. This goal could be achieved by the improvement in the information flow as discussed by Morgan and Liker (2006) in the process subsystem of LPD, and by Reinertsen's (2009) argument about making product development flow.

To position this research among different classifications of LPD mentioned in this Section, it is first needed to have a closer look at the levels of lean implementation in an organization, which is presented in the following Section.

2.5 Levels of lean implementation

Lean has mainly been viewed by researchers and practitioners as a set of tools and methods for improving the performance of business activities, because it originated and flourished in the manufacturing industries. However, a point raised by lean critics is that the long-term sustainability of lean implementation needs special attention to the human dimensions of motivation, empowerment, and respect for people (Hines et al., 2004). In addition, the lack of sustainability in many lean transformation programs could be attributed to putting more emphasis on the application of tools and techniques, instead of having lean program at strategic level.

Lean journey is evolutionary by its nature, which seeks improvement through a gentle but continuous stream of changes (Ellis, 2016). The first step in each study related to lean should be defining the specific area in which lean is needed to be applied, and based on that, finding the appropriate approaches and practices. The main reason for misunderstandings in lean transformation programs as Hines et al. (2004) mentioned is the inability of distinction between levels of lean implementation. According to Figure 5 they introduced two levels for lean transformation programs; the *strategic level* of lean, including the customer-centred strategic thinking, is universal and applicable everywhere, whereas the *operational level*, including tools and techniques and process improvement, is more contingent to the environment.

Strategic level refers to the implementation of lean at the enterprise level to transfer the organization to “an integrated entity that efficiently creates value for its multiple stakeholders by employing lean principles and practices” (Murman et al., 2002: P144). Application and embodiment of lean principles throughout the enterprise needs supporting practices at the operational level. These practices, although interdependent in nature, could be classified under two dimensions of human-related and process-related practices (Murman et al., 2002) as represented in Table 7.

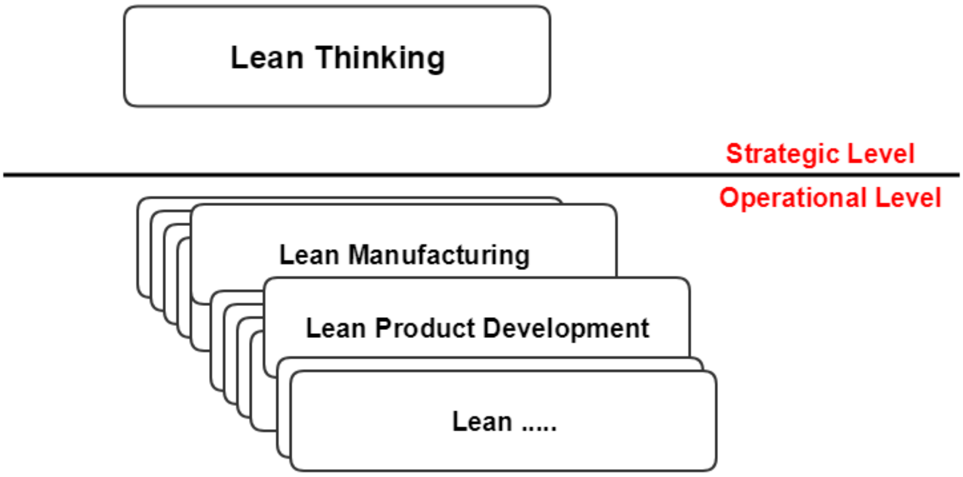


Figure 5: two levels of lean implementation

Table 7: high-level practices for application of lean enterprise principles

Human-related practices	Process related practices
Promote lean leadership at all levels	Assure seamless information flow
Relationship based on mutual trust and commitment	Implement integrated product and process development
Make decisions at lowest appropriate level	Ensure process capability and maturation
Optimize capability and utilization of people	Maintain challenges to existing processes
Continuous focus on customers	Identify and optimize enterprise flow
Nurture a learning environment	Maintain stability in changing environment

In another attempt to clarify the lean concept, Hines et al.(2011) visualised lean elements as an iceberg including a small visible and a big invisible part (Figure 6). The first invisible factor in the lean iceberg is strategy and alignment. For example, '*Hoshin Kanri*' or '*Policy Deployment*' in TPS acts as a strategic management system for setting the direction of the organization and sharing the goals and visions of the business with all employees in the organization. According to Hines et al. (2011) there should be a clear alignment between *Key Performance Indicators (KPI)* in a company, the strategy, and the lean improvement projects, otherwise, KPIs become wasteful and only consume valuable resources in the company. Leadership is the second invisible factor in the lean iceberg. Poor sustainability of lean transformation programs results from poor leadership, as they are leaders in an organization who create the ground for fostering changes, establishing direction, developing the visions of the future, setting strategies for achieving these visions, and inspiring others to be willing to change. Finally, behaviour and engagement, as the third invisible factor in the lean iceberg, focuses on people and the way they could be motivated to adopt lean behaviours and become engaged in the transformation. The crucial factor for sustaining the lean change is engaging all employees from the very start of lean transformation program. Employees need to be involved in setting KPIs to feel the ownership and understand the importance of meeting the key targets. Trust, honesty, openness, consistency, respect, reflection, observation, objectivity, and listening are among the examples of lean behaviour, and to change the employees' behaviour towards lean, changing the organization's culture is a necessity.

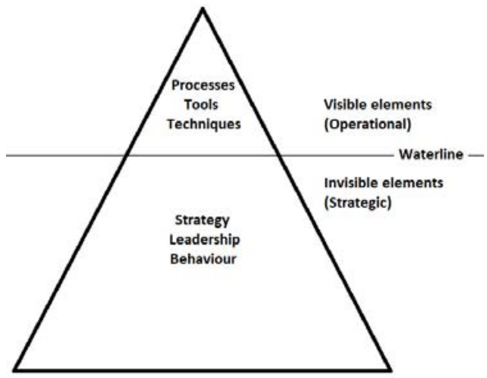


Figure 6: the sustainable Lean iceberg model (Hines et al., 2011)

To conclude, despite years of research and practice about the implementation of lean, there still is a misunderstanding about the concept and its application in different areas. The problem gets even worse in innovative approaches to lean such as LPD. These shortcomings are mainly attributable to the inability in understanding the position of research against different approaches to implementation of lean as discussed before and summarized as followed:

- the dichotomy of the strategic level versus the operational level of lean implementation;
- the dichotomy of visible elements versus invisible elements of lean according to the lean iceberg model;
- the dichotomy of human-related practices versus process-related practices at the operational level to support the implementation of lean at the strategic level;

Figure 7 provides a framework for classification of different approaches to lean implementation based on their levels of abstraction and the focus of practices.

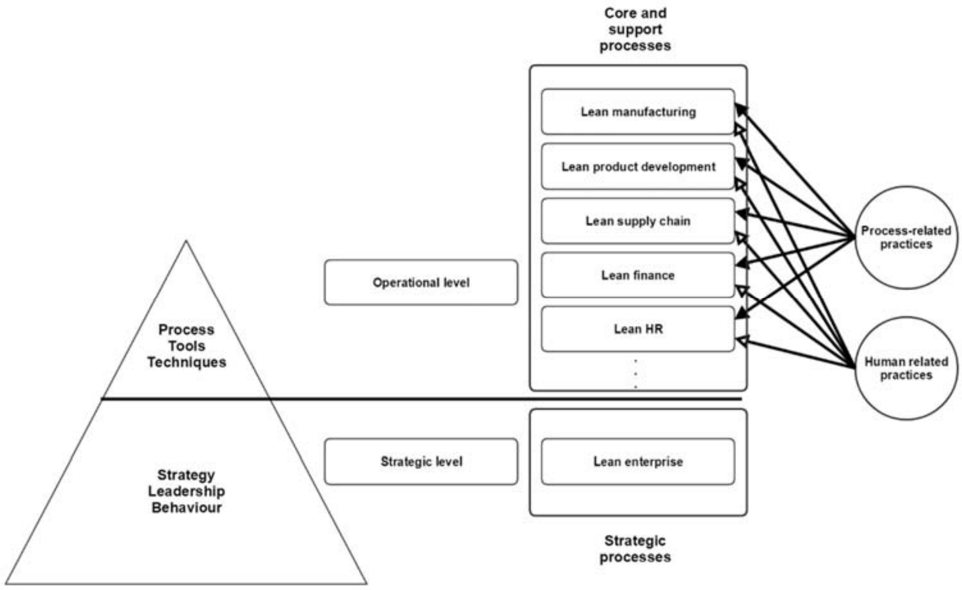


Figure 7: Framework for classification of various aspects of lean implementation

2.6 Positioning the research among different classes of LPD research

Increasing the competitive advantage of companies by improving their NPD processes using Lean concept has been the subject of research in recent years (Martínez León and Farris, 2011). Going back to the framework in Figure 7, implementation of lean at the operational level in manufacturing and NPD requires different approaches to the processes, tools and techniques, and the same approach could not be used for the implementation of lean in these two different areas. In addition, applying lean manufacturing principles to NPD processes, although may result in some short-term benefits (Khan, 2012), in the long-term produces inconsistencies due to fundamental differences between operations in manufacturing and NPD. For instance, as Khan (2012) mentioned waste elimination in NPD processes does not result in higher product quality and value stream mapping would not be able to identify all value-added steps in NPD as they are hidden in processes. Consequently, in this research LPD is viewed based on TPDS, not an extension of lean manufacturing in NPD processes.

On the other hand, constructs, elements, and tools proposed by authors who used TPDS as their benchmark for LPD still shows a combined focus on the operational vs. strategic level, and process-related vs. human-related practices. For example, strategic management of NPD projects as proposed by Karlsson and Åhlström (1996) as an element of LPD is at the strategic level, whereas they also put concurrent (simultaneous) engineering in their framework which is at the operational level. Similarly, strong project manager as a component of LPD proposed by Hoppmann et al. (2011) is human-related, while standardization in their framework is process-related. This mixture of various levels and practices of LPD makes them unclear and complex. To shed some light on this issue, publications which developed frameworks for LPD with the concentration on TPDS are reviewed and 94 constructs, elements and tools suggested by them are extracted and classified based on the framework in Figure 7. Although all selected publications take TPDS as the basis for LPD, still 15 elements in their frameworks are identified which either address general principles of lean thinking proposed by Womack and Jones (1996), or are among tools and techniques of lean manufacturing; they are shown in Table 8 and excluded from this classification. Other 79 elements and constructs are compared to find those address the same concept but with different terminologies. Finally, they all are grouped under 10 headings as shown in Table 9.

Table 8: elements related to general lean thinking principles

Elements	**Source**
Continuous improvement	(Nepal, et al., 2011)
Continuous improvement culture	(Khan et al., 2011)
Build in learning and continuous improvement	(Morgan and Liker, 2006)
Build a culture to support excellence and relentless improvement	(Morgan and Liker, 2006)
Establish customer-defined value to separate value-added from waste	(Morgan and Liker, 2006)
Value focus	(Ward, 2007)
Simple and specified pathways for information flow	(Nepal, et al., 2011)
Cadence, pull and flow	(Ward, 2007)
Effective flow of requirements down the project structure	(Haque and James-Moore, 2004)
Effective flow of information and technology into the projects	(Haque and James-Moore, 2004)
Create a level product development process flow	(Morgan and Liker, 2006)
Kaizen events	(Letens, et al., 2011)
Value and risk assessment from market and resource perspective	(Letens, et al., 2011)
Workload levelling	(Hoppmann et al., 2011)
Value stream mapping	(Letens, et al., 2011)

Table 9: grouping LPD-specific elements and constructs

Group	**Number of elements**
Chief engineer system	9
Cross functional team	13
Knowledge management	4
Responsibility-based planning	6
Process management	2
Set-based concurrent engineering	12
Standardization	13
Strategic management	3
Supplier involvement	8
Tools and technology	9

Table 10 represents all elements and constructs of LPD extracted from publications based on TPDS and classified according to framework in Figure 7. Elements which address general concepts and are applicable everywhere in an organization are labelled strategic, whereas elements applicable only to the NPD environment are labelled operational. For example, '*coordination and rich communication*' which proposed by Nepal et al. (2011) is labelled strategic-level element, and '*heavyweight team structure*' proposed by Karlsson and Åhlström (1996) is labelled operational-level element. In addition, using the classification of practices in Table 7, elements and constructs are grouped under human-related and process-related practices. As the result, practices such as '*coordination and rich communication*' and '*heavyweight team structure*' which promote lean leadership and the relationship based on the mutual trust and commitment are labelled human-related practice, whereas '*concurrent engineering'* which assures seamless information flow, process capability and maturation in NPD is labelled process-related practice. Analysing 11 classes of elements and constructs in LPD shows that *Chief Engineer System*, *SBCE*, *Standardization*, *Process Management*, and *Tools and Technology* are fully operational, whereas *Knowledge Management*, *Responsibility-Based Planning*, *Strategic Management*, and *Supplier Involvement* are fully strategic. On the other hand, *SBCE*, *Process Management*, *Supplier Involvement*, *Tools and Technology*, and *Standardization* mainly include process-related practices, while *Chief Engineer System*, *Cross Functional Team*, *Knowledge Management*, and *Responsibility-Based Planning* mainly include human-related practices.

Table 10: classification of LPD elements and constructs

	Group	Constructs / Elements / Tools	Level	Practice	Source
1	**Chief engineer system**	Chief engineer or leadership	Operational	Human-related	(Sobek, et al., 1998)
		Direct supervision	Operational	Human-related	(Sobek, et al., 1998)
		Technical or System designer entrepreneurial leadership	Operational	Human-related	(Kennedy, 2003)
		Heavyweight project manager	Operational	Human-related	(Haque and James-Moore, 2004)
		Develop a chief engineer system to integrate development from	Operational	Human-related	(Morgan and Liker, 2006)
		Entrepreneurial system designer or Chief engineer	Operational	Human-related	(Ward, 2007)
		Strong project manager	Operational	Human-related	(Hoppmann et al., 2011)
		Chief engineer technical leadership	Operational	Human-related	(Khan et al., 2011)
		chief engineering system	Operational	Human-related	(Rossi, et al., 2012)
2	**Cross functional team**	Coordination and rich communication	Strategic	Human-related	(Nepal, et al., 2011)
		Workforce expertise or Expert engineering workforce	Operational	Human-related	(Kennedy, 2003)
		Balance functional expertise and cross-functional integration	Operational	Human-related	(Morgan and Liker, 2006)
		Cross-functional teams	Operational	Human-related	(Karlsson and Åhlström, 1996)
		Integrated rather than coordinated functional aspects	Operational	Human-related	(Karlsson and Åhlström, 1996)
		Heavyweight team structure	Operational	Human-related	(Karlsson and Åhlström, 1996)
		cross functional teams involving customers and suppliers	Operational	Human-related	(Haque and James-Moore, 2004)
		Teams of responsible experts	Operational	Human-related	(Ward, 2007)
		Creating integrated product development team	Operational	Human-related	(Letens, et al., 2011)
		Creating flexible functional resources	Operational	Human-related	(Letens, et al., 2011)
		Develop towering competence in all engineers	Strategic	Human-related	(Morgan and Liker, 2006)
		Specialist career path	Strategic	Human-related	(Hoppmann et al., 2011)
		Teaching and learning	Strategic	Human-related	(Nepal, et al., 2011)
3	**Knowledge management**	knowledge acquisition, transfer, application and reuse	Strategic	Human-related	(Haque and James-Moore, 2004)
		Cross-project knowledge transfer	Strategic	Human-related	(Hoppmann et al., 2011)
		Knowledge-based environment	Strategic	Human-related	(Khan et al., 2011)
		Cross functional knowledge capture, use and reuse	Strategic	Human-related	(Rossi, et al., 2012)
4	**Responsibility-based planning**	Responsibility-based planning & control	Strategic	Human-related	(Kennedy, 2003)
		Responsibility-based planning and control	Strategic	Human-related	(Hoppmann et al., 2011)
		Value-focused planning and development	Strategic	Human-related	(Khan et al., 2011)
		Black box engineering	Strategic	Process-related	(Karlsson and Åhlström, 1996)
		formal requirement capture, management and concretization	Strategic	Human-related	(Haque and James-Moore, 2004)
		Decentralized decision making	Strategic	Human-related	(Letens, et al., 2011)
5	**Process management**	Process management and performance measurement	Operational	Process-related	(Haque and James-Moore, 2004)
		processes for managing NPD including change control, configuration, management, and appropriate reviews	Operational	Process-related	(Haque and James-Moore, 2004)
6	**Concurrent engineering**	Concurrent engineering	Operational	Process-related	(Karlsson and Åhlström, 1996)
		Concurrent engineering	Operational	Process-related	(Haque and James-Moore, 2004)
		Concurrent engineering	Operational	Process-related	(Letens, et al., 2011)
		Concurrent engineering	Operational	Process-related	(Rossi, et al., 2012)
		Concurrent engineering	Operational	Process-related	(Letens, et al., 2011)
		Simultaneous engineering	Operational	Process-related	(Hoppmann et al., 2011)
7	**Set-based design**	Set-based concurrent engineering (SBCE)	Operational	Process-related	(Kennedy, 2003)
		Set-based concurrent engineering (SBCE)	Operational	Process-related	(Ward, 2007)
		Set-based concurrent engineering (SBCE)	Operational	Process-related	(Khan et al., 2011)
		Set-based concurrent engineering (SBCE)	Operational	Process-related	(Rossi, et al., 2012)
		Set-based design	Operational	Process-related	(Hoppmann et al., 2011)
		Front–end loaded	Operational	Process-related	(Letens, et al., 2011)
		Front–end loaded	Operational	Process-related	(Morgan and Liker, 2006)
8	**standardization**	Standardization of processes	Operational	Process-related	(Sobek, et al., 1998)
		Standard skills	Operational	Human-related	(Sobek, et al., 1998)
		Standardization of processes	Operational	Process-related	(Nepal, et al., 2011)
		Design standards	Operational	Process-related	(Sobek, et al., 1998)
		Common parts/specifications/design re-use	Operational	Process-related	(McManus, et al., 2005)
		Variability reduction/dimensional management	Operational	Process-related	(McManus, et al., 2005)
		Utilize rigorous standardization to reduce variation, and create	Operational	Process-related	(Morgan and Liker, 2006)
		Use powerful tools for standardization and organizational learning	Operational	Process-related	(Morgan and Liker, 2006)
		Product variety management	Operational	Process-related	(Hoppmann et al., 2011)
		Process standardization	Operational	Process-related	(Hoppmann et al., 2011)
		Standardized critical components	Operational	Process-related	(Letens, et al., 2011)
		Grouping project fimilies	Operational	Process-related	(Letens, et al., 2011)
		standardization (product variety management)	Operational	Process-related	(Rossi, et al., 2012)
9	**strategic management**	Strategic management of development project	Strategic	Human-related	(Karlsson and Åhlström, 1996)
		Long term development	Strategic	Human-related	(Sobek, et al., 1998)
		product development driven by vision and strategy	Strategic	Human-related	(Haque and James-Moore, 2004)
10	**supplier involvement**	Supplier relationships	Strategic	Process-related	(Nepal, et al., 2011)
		Supplier and customer involvement in design	Strategic	Process-related	(Karlsson and Åhlström, 1996)
		Communication with partners	Strategic	Process-related	(Sobek, et al., 1998)
		cross functional teams involving customers and suppliers	Strategic	Process-related	(Haque and James-Moore, 2004)
		Fully integrate suppliers into the product development system	Strategic	Process-related	(Morgan and Liker, 2006)
		Supplier integration	Strategic	Process-related	(Hoppmann et al., 2011)
		Supplier integration	Strategic	Process-related	(Rossi, et al., 2012)
		Developing long term relationship with strategic partners	Strategic	Process-related	(Letens, et al., 2011)
11	**Tools and technology**	organizational infrastructure which supports visual assessment and performance monitoring mechanisms	Operational	Process-related	(Haque and James-Moore, 2004)
		Design for X	Operational	Process-related	(McManus, et al., 2005)
		Solid-model-based design	Operational	Process-related	(McManus, et al., 2005)
		Production simulation	Operational	Process-related	(McManus, et al., 2005)
		Adapt technologies to fit your people and process	Operational	Process-related	(Morgan and Liker, 2006)
		Align your organization through simple visual communication	Operational	Process-related	(Morgan and Liker, 2006)
		Rapid prototyping, simulation and testing	Operational	Process-related	(Hoppmann et al., 2011)
		Visual management	Operational	Process-related	(Letens, et al., 2011)
		Design trade offs	Operational	Process-related	(Letens, et al., 2011)

2.6.1 A process-view of LPD

The focus of this research is on the process design and structure in LPD and its effects on the performance of the NPD system. It is obvious that only a well-managed process could result a good financial performance (Hines et al., 2011), so the main motive for studying LPD from the process perspective is to explore alternative process structures, and to find a linkage between the process structure and the system behaviour which affects the overall performance of the system. Assuming that LPD process structure could affect the system performance, combined with the idea that the value in NPD processes streams in the form of information (Clark and Fujimoto, 1991; Reinertsen, 2005), it could be concluded that improving information flow through optimizing the process architecture leads to the higher performance. Studying the process aspects of LPD to find its relationship with the performance measures at the level of a single project provides the ability to define the most appropriate process architectures based on the project characteristics, which leads to the faster development of products with lower cost and higher quality.

On the other side, *System Thinking* principles emphasis on the role of process structure in the overall behaviour of the system by articulating that different people produce the comparable results if placed in the same system (Senge, 1992). This idea dictates this research to focus on process-related practices of LPD. In addition, NPD could be defined at various levels of the organization's hierarchy (Haque and Moore, 2004). While the high-level process should be communicated across all business units, targeting NPD as a low-level process in an organization (compare to processes directly related to the strategy and leadership aspects) results in concentrating more on operational level, instead of strategic-level elements of lean implementation. Extracting from Table 10, elements and constructs which are in operational level and include process-related practices are represented in Table 11.

Table 11: operational level and process-related LPD elements extracted from Table 10

Group	Number of elements/ constructs	Operational elements/ constructs	Strategic elements/ constructs	Human-related elements/ constructs	Process-related elements/ constructs
Process management	2	2	0	0	2
Standardization	13	13	0	1	12
Set-based concurrent engineering	13	13	0	0	13

Among groups of LPD elements and constructs identified in Table 8, the combination of concurrent engineering and set-based approach, as the unique concept of *SBCE is* the focus of this research. *Process Management* or *Value Stream Management* is the process of "*defining, analysing, documenting, controlling, and improving the business processes or value streams to make them effective, efficient, and adaptable so that customers (external and internal) expectations are exceeded and waste within the processes is eliminated*" (Haque and Moore, 2004: P13). It is only proposed by Haque and Moore as an element of LPD. Standardization in NPD as stated by Morgan and Liker (2006) is about the reduction of variations while keeping the necessary creativity for the innovative process. Three broad categories of standardization in TPDS are *Product Standardization* by using common architectures and modular, reusable, and shared components, *Skill Standardization* which provides flexibility in staffing and project planning, and *Process Standardization* by identifying repetitive tasks and activities across different projects and making them standard (Hoppmann et al., 2011). Between categories of standardization in NPD, only the process standardization belongs to the process-related elements of LPD. Following SBCE approach is not possible without having a standardized process in NPD, so it is presumed that studying SBCE approach covers the subject of process standardization.

To explain the SBCE approach as a unique approach in LPD, it is helpful first to have a general overview on different approaches to the NPD process design.

2.7 Designing NPD processes

NPD projects, as mentioned by Cross (2008), have several characteristics. The design problem defined at the start of the project is unstable and temporary, because while more information become available the definition of the design problem is continuously changed. These changes could be due to emerging new design requirements as the result of design solutions evaluation, or the changes in customers' requirement. Especially, for complex products, such as automobiles and aircrafts, where the design process takes years to be completed, the probability of changes in the customers' requirements during this prolonged period is high. In addition, there is no standard rule for obtaining a solution for engineering design problems. It is difficult to formulate the design problem without referring to a solution method, because the way that the design problem is formulated has an impact on the method to obtain the solution. However, most important among all characteristics, there are always several practical solutions for a design problem and design teams must find the most promising solution among them. Finding this best solution, yet, is only possible while existing solutions are iteratively refined and evolved over design phases.

To cope with these problems, a methodical sequence of steps in the NPD processes to be followed by design teams to solve the design problems, could be employed (Riaz, 2015). Although there are several designs for NPD processes, the good process is the one which enables designers to systemically follow its steps and provides them with a framework for managing the design duration while clarifying the design problem, fostering innovation, managing the complexity of the product, and finally, enabling collaboration between multiple design teams towards a single goal (Holston, 2011).

Dieter (2000) articulated the importance of a good NPD process design by showing the effect of decisions made during the design process on the overall product cost. As shown in Figure 8, while the NPD project cost makes about 5 percent of the cost of the final product, decisions made during the NPD process have an impact on about 80 percent of the total product cost. However, at the upfront of the project there is limited amount of knowledge available for making these decisions. It highlights the importance of employing an NPD process design which enables design teams to make informed decisions by providing them with the opportunity to gather more knowledge about the design problem.

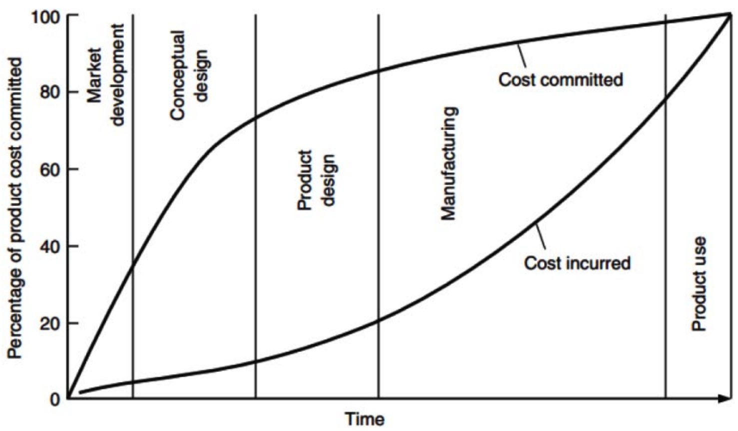


Figure 8: different impacts of the NPD process on product cost (Holston, 2011)

2.8 Approaches to designing NPD processes

2.8.1 Point-based approach

The NPD process design has been the subject of numerous researches throughout the years, most of them suggested a model to show how it proceeds. The most-common NPD process design follows a '*synthesise, analysis, modify'* approach which through cycles of iteration an initial design is proposed, tested, and after finding a problem goes back to the first step to be changed and modified, and follow the cycle again. The NPD process starts by early decisions about selecting a single concept using the previous projects experience. Through iteration cycles the concept is analysed and modified to finally meet the design requirements. The design of NPD process proposed by Pahl and Beitz (2007) is a good example of this iterative stage-based. Their process design involves four phases; the main activities in the first phase are analysing the design problem to define the product functions and design constraints in the form of a design specification. The conceptual design is the second phase which is responsible for generating and evaluating solutions for defined design functions. The third phase is embodiment or system design through which the chosen design concept from the conceptual design phase is elaborated by defining the layout and interfaces of the product components. Finally, there is the detail design phase which specifies detailed dimensions, the shape, materials for components and prepare an instruction to produce the product.

This iterative stage-based approach also called point-based design, because at each point in the design process, design teams only work on a single design solution (a single point in the design space). According to Liker et al. (1996) a point-based approach has five basic steps:

- understanding customers' needs and defining the problem in the form of product requirements based on them;
- generating alternative design concepts by design teams;
- conducting preliminary analysis and selecting a single concept for further development;
- analysis and modification of selected concept to meet the product requirements;
- repeat the process several times if the selected concept fails to meet the goals.

One of the main limitation of the point-based approach is costly design changes during the process. The main cause is little or no communication between functional groups involve in the NPD process and '*throw it over the wall'* approach (Bernstein, 1998). As the result, the viewpoints of other functional groups, such as manufacturing, sales and marketing and service do not consider into design. This highly iterative approach is very inefficient especially in the design and development of complex products which need groups of engineers and designers to be brought together to collectively work on the design problem (Krishnan et al., 1997). According to Wheelwright and Clark (1992) the interaction between upstream and downstream phases could be defined in four modes, as shown in Figure 9. The highest intensity of communication between phases occurs in mode 4 (Integrated Problem Solving), much earlier than other modes, and facilitates the incorporation of upstream and downstream knowledge into the design of the product.

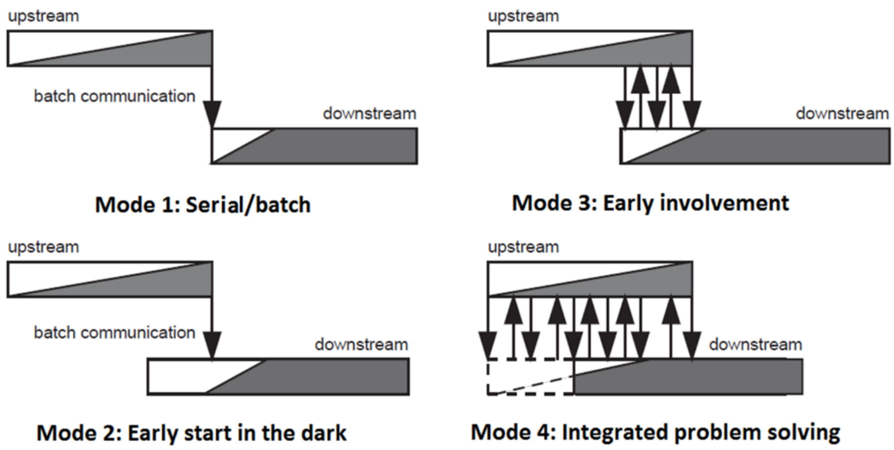


Figure 9: Four modes of upstream-downstream communication (Wheelwright and Clark, 1992)

2.8.2 Concurrent (simultaneous) engineering

To overcome the communication problem in the point-based approach and to facilitate better interaction between design teams, based on integrated problem-solving mode in Figure 9, concurrent engineering method is developed as one of the common themes in LPD, highly cited as an approach to reduce the lead time of projects (Clark and Fujimoto, 1991; Wheelwright and Clark, 1992; Karlsson and Åhlström, 1996; Haque and Moore, 2004; Ward, 2007; Hoppmann et al., 2011). Concurrent engineering involves overlapping of dependent product development phases and activities. It emphasises on earlier sharing information that needed by downstream activities, and simultaneous development of products and their related manufacturing and support processes. The combination of the point-based design and concurrent engineering is called *Point-Based Concurrent Engineering (PBCE)*. The immediate benefit of this approach is receiving quick and extensive feedback from downstream by upstream and a decrease in the number of late design changes. As the result, especially for the activities on the critical path of the project, concurrent engineering could be a successful approach for the reduction of the overall lead time of the project (Zhang and Bhuiyan, 2015), while could also lead to lower cost, and improved productivity (Prasad, 1996).

Figure 10 shows the effect of concurrency between two interdependent phases on lead time in more detail. In a sequential process, the upstream phase starts and finishes before starting the downstream phase, so all required information is available at the outset of downstream phase. But, in concurrent engineering, the downstream phase starts its work earlier using preliminary information from upstream phase. Concurrent engineering has three types of effects on the process lead time (Ford and Sterman, 1998) which are listed in Table 12.

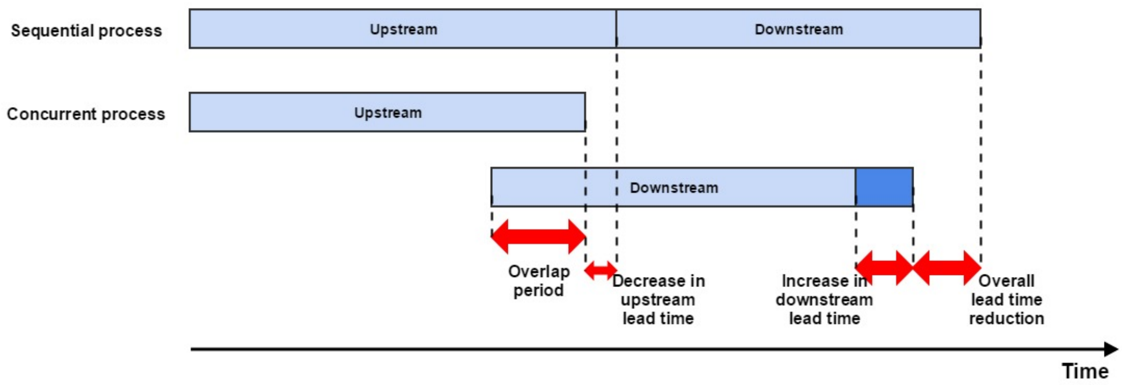


Figure 10: the effect of concurrency between phases on process time

Table 12: effects of concurrent engineering on lead time

Effects	**Reasons**
Decreasing the lead time of upstream phase	The feedback on quality of work in upstream phase receiving from downstream results in quicker detection of errors
Increasing the lead time of downstream phase	Changes in preliminary information received from upstream phase which could corrupt the correctly done work in downstream
Reducing the overall process time	Overlapping process phases

2.8.3 Limitations of concurrent engineering

While empirical works showed the relationship between concurrency and having higher development speeds, some researchers argued that concurrency could increase the development costs, and risks in product quality (Krishnan et al., 1997; Smith and Eppinger, 1997b; Calantone and Di Benedetto, 2000). According to Terwiesch et al. (2002) and Lin et al. (2010), if not designed and managed properly, concurrent engineering could be a potential source of rework cycles. The time advantage of concurrent engineering highly depends on the quality of assumptions, the stability of inputs, the magnitude of the impact of any change in inputs on the quality of work in recipient activity, and the speed of reworks in comparison with the primary work. In other words, significant changes in upstream information and high dependency of downstream to outputs from upstream make overlapping even disadvantageous for the performance of the project (Lin et al., 2009). Yet, the complexity in the dependent behaviour of overlapping phases make it difficult to predict the resulting performance (Zhang and Bhuiyan, 2015).

Overlapping of activities and phases increases coordination costs and risk because the work in downstream stages starts using assumptions or preliminary data from upstream (Browning and Eppinger, 2002). Meier, et al. (2015) stated that in NPD, where the input needed by an activity is in the form of information, often a percentage or even the whole activity could be done based on assumptions instead of actual information. Based on the quality of assumptions and the speed of reworks, concurrency could reduce the overall time and cost of the process, or, in contrary, results in unnecessary reworks which consume resources. Meier, et al. (2015) considered a time-cost trade-off created by the process concurrency and Swink et al. (2006) stated that additional costs are more than what repaid by benefits in product quality and downstream launch efficiencies.

The source of concurrent engineering limitations is starting the work in downstream phase using preliminary upstream information (Krishnan et al., 1997). As this information is not finalized yet probable changes could increase the probability of invalidating (corrupting) decisions made in downstream phase based on upstream preliminary information and result in reworks. Corruption probability is an almost ignored negative effect of concurrent engineering (Lin et al., 2008). Due to corruption, reworks in downstream phase become necessary, whether the work has been done correctly or not, as it is based on faulty information from upstream phase (Lin et al., 2008). So, the time reduction because of concurrency is not always a linear function of the length of overlapped period. To reduce the corruption probability and to simplify the interactions between design teams and functional groups one way is to finalize and freeze upstream information and products requirements quickly. However, this could result in the quality loss in upstream information due to sacrificing the chance of future information update. Increasing the degree of concurrency between interdependent phases of a project results in higher corruption probability, which consumes more resources due to reworks and unplanned iterations, thus increasing the lead time and the cost of the project while decreasing the productivity of the process. In addition, the proposed changes in preliminary information could produce conflicts between design teams as their needs and limitations are not necessarily clear for each other (Bernstein, 1998). These changes also could propagate through the system if the product components are tightly coupled (Liker et al., 1996).

2.8.4 Research trends in concurrent engineering

The focus of researchers in the field of concurrent engineering has been on developing models to study the effect of different process variables related to concurrent engineering on the projects performance. In general, main approaches for modelling concurrent engineering in NPD projects, as adopted by researchers, are as followed:

- Using mathematical models and optimization techniques by formulating the overlapping problem as an objective function and defining related constraints (Roemer et al., 2000; Roemer and Ahmadi, 2004; Lin et al., 2009; Lin et al., 2010; Zhang and Bhuiyan, 2015)
- Modelling using *Design Structure Matrix (DSM)* to map existing dependencies between design stages and sequence them optimally to reduce the probability of rework in concurrent engineering (Browning and Eppinger, 2002; Meier et al., 2007; Yan et al., 2010; Meier et al., 2015)
- Modelling using *Graphical Evaluation and Review Technique (GERT)* (Wu et al., 2010; Nelson et al., 2016)
- System dynamics modelling (Ford and Sterman, 1998; Ford and Sterman, 2003; Lin et al., 2008)

Models of concurrent engineering have evolved from taking simple assumptions to establishing complex functions (Zhang and Bhuiyan, 2015). In an early work, using an analytical model, AitSahlia et al. (1995) studied the execution of tasks with various levels of concurrency and found that depends on the probability of creating iteration there is a tipping point for the level of concurrency at which more concurrency would have negative impact on the project time. Krishnan et al. (1997), in their model of overlapping between two dependent activities, defined upstream evolution and downstream sensitivity concepts which their combination could be used as a measure for determining the overlapping strategy according to the dependency between activities. Upstream evolution is defined as the speed of modification in preliminary information, and downstream sensitivity represents the magnitude of the impact of these modifications on the downstream work (Lin et al., 2009). The combination of low upstream evolution and high downstream sensitivity results in high dependency between activities, so most of the work done using preliminary information in downstream must be reworked which makes concurrency disadvantageous. Loch and Terwiesch (1998) introduced the concept of impact function in downstream which was correspondent with the time delay in downstream progress resulted from receiving upstream modifications. Probability of rework as a function of the duration of overlap between two phases was introduced by Roemer et al. (2000) who studied the time-cost trade-off for different concurrency strategies to find the optimum one. Using a system dynamics model Ford and Sterman (2003) made a relationship between the increase of unplanned iterations in NPD projects, and the delay in discovering rework in concurrent processes. Bhuiyan et al. (2004) incorporated two types of iterations into their model; one due to the transfer of preliminary information between phases of the project, and another for integrating engineering functions within those phases and using this model indicated the positive effect of the increase in functional interaction on cross-phase iterations which resulted in the reduction of overall project time and effort. Roemer and Ahmadi (2004) addressed the concurrency and crashing as two common tools for reducing the lead time of NPD projects through a formal model to find the optimal policies. While in most of mentioned researches the cost of communication between overlapped stages is neglected, Lin et al. (2009; 2010) assumed time and cost for setting meetings and information exchanges and built their analytical model based on the effect of functional interactions on the performance of concurrent engineering. They concluded that the impact of upstream modifications is a function of downstream progress, which is the amount of work completed in downstream stage unaffected by received information and depends on the type of the project. Meier et al. (2015), using DSM, built a model which considered architecture, iteration, crashing, and overlapping in NPD projects and showed the effect of work policy decisions on project outcomes. Finally, Nelson et al. (2016) developed a time-computing *GERT* model to estimate the lead time of concurrent NPD projects by incorporating factors such as the degree of information dependence, the experience of engineers, and the complexity of tasks.

2.8.5 Set-based approach

Although adding concurrent engineering to the point-based approach aims to reduce the time and the effort wasted due to design specifications which do not meet the downstream requirements, PBCE does not change the nature of interactions between upstream and downstream phases (Bernstein, 1998). The problem is while the earliest decisions made during the design process have the highest impact on the final cost and quality of product, these decisions in PBCE are made with the least information and knowledge about various aspects of the final product (Ward et al., 1995a). As Reinertsen (2009) stated, the cost of changes in product design increases exponentially during the NPD process. In addition, late in the process the ability to influence these costs is low as the alternatives available for design changes are very limited due to the constraints made by earlier decisions. The conclusion is a dilemma which design teams are facing; when they have the maximum power to influence the final product, their knowledge is limited, and when they gain enough knowledge, their ability to affect the final product is minimized.

All mentioned problems make having a paradigm shift in NPD process design unavoidable. The new approach starts with multiple design solutions and eliminated their inferiors through the progress of the design process when more knowledge about design problem is accessible. The total design method proposed by Pugh (1991) is an example of this approach. In a two-step process design teams first synthesise many design solutions and then evaluate them against customers' requirements, and discard weaker solutions. Then the number of concepts increases again through generating additional solutions by modification of existing ones or proposing new ones. This cycle of expansion and contraction is repeated several times while each iteration results in narrower design space until finally one design solution remains (Figure 11).

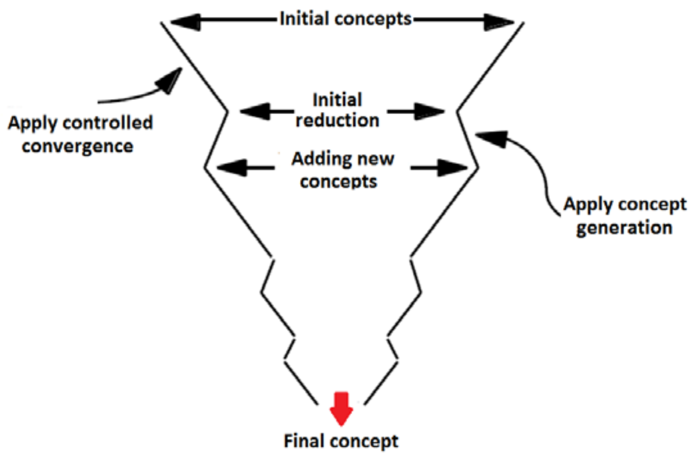


Figure 11: Total design method (Pugh, 1991)

The design-build-test cycle proposed by Wheelwright and Clark (1992) is another example which starts with several alternative concepts based on understood requirements of customers. Physical or virtual models are built and tested to find design changes which close the gap between the product performance and requirements. The design-build-test cycle is repeated several times, until all the requirements are satisfied. The effectiveness of each cycle, the way the results of each cycle combines into the coherent solution and the number of cycles which are completed determine the effectiveness of this method (Wheelwright and Clark, 1992).

Both total design and design-built-test methods emphasise on considering multiple concepts for an extended period during the project to understand the impact of different design parameters on the ability of the concept to fulfil customers' requirements. While still these approaches are very time-consuming and costly, another approach is introduced by Ward, et al. (1995a), called *SBCE* based on their study on TPDS and LPD. Unlike PBCE approach in which a single design solution is modified iteratively to fit the specifications (Morgan and Liker, 2006), SBCE considers alternative design solutions which are systematically converged towards a single solution by eliminating inferior ones (Malak et al., 2009). In other words, in the SBCE approach selecting the final concept is delayed while a set of alternatives concepts are gradually narrowed down based on their performance, reliability, cost, manufacturability, and systems integration (Sobek et al., 1999; Morgan and Liker, 2006; Hoppmann et al., 2011; Khan et al., 2013). As shown in Figure 12, the NPD process starts with considering a wide range of design solutions which are narrowed down gradually using explicitly communication and reasoning in design teams about the existing design solutions and eliminating the infeasible ones until a final design solution emerges.

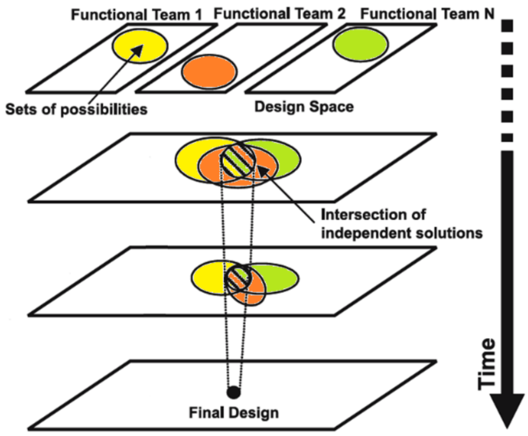


Figure 12: principles of SBCE (Nahm and Ishikawa, 2006)

Figure 13 schematically shows the comparison between the SBCE and PBCE, and their impact on the project lead time. At the top, the PBCE approach is shown which includes selecting a unique concept as early as possible during the development process. The sketch at the bottom shows SBCE which although has a longer conceptual design phase due to the delay in the concept freeze, decreases the lead time of the project because of decreasing the probability of late design changes. Without SBCE, there is a high probability of reworks and late changes in downstream phase, which could result in increasing the lead time of this phase to a level that counteract the positive effect of the concurrency on overall project lead time. This could also have a negative effect on the project cost and quality.

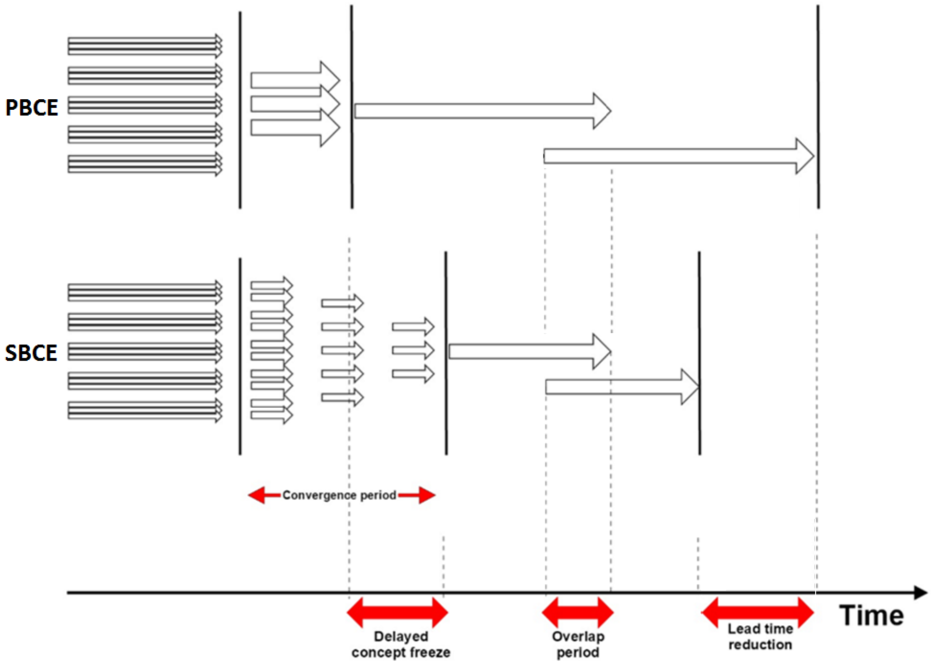


Figure 13: The effect of SBCE on the process lead time compared to PBCE

The superiority of SBCE process design lies in enabling design teams to make informed decisions by postponing critical decisions which decreases the late design changes and foster innovation (Riaz, 2015). Khan, et al. (2013) indicated that SBCE addresses some typical challenges in NPD projects, including the amount of rework, sub-optimal designs, lack of innovation, and high product cost. Each design team independently explore the design space and develop their own sets of solutions, based on their different engineering specialties. The emphasis is then, on inter and intra-teams' communications and interactions to compare sets of design solutions which are developed based on each team's perspective, and to find regions of overlap in design alternatives to narrow the alternatives down. Sobek et al. (1999) characterized the principles of SBCE into three steps, in addition to the enablers for each step indicated by Kerga et al. (2014), as shown in Table *13* and briefly discussed below:

- map the design space through defining feasible regions, exploring trade-offs by designing and communicating sets of alternative solutions;

- integrate by intersection which involves finding intersections of feasible sets, imposing minimum constraint, and seeking conceptual robustness;
- establishing feasibility before commitment through gradually narrowing down sets while increasing details, staying within sets once committed, and controlling by managing uncertainty at process gates.

Table 13: SBCE steps and enablers (Kerga et al., 2014)

Steps of SBCE	SBCE enablers
Exploration of alternative sub-systems	Quality Function Deployment Tradeoff curves
Looking for intersection among alternative subsystems	Checklists to find intersection sets
Eliminate technically infeasible alternatives, narrow down the design space until the single compatible and feasible consept remains	Limit curves from testing Time and cost analysis to select the optimal alternative

In the first step design solutions should be modelled as sets, instead of point values. It could be done using several ways, such as parametrization, modularity, or creating a range of discrete solutions which satisfy various parts of the requirements range (Levandowski, 2014). SBCE approach in the conceptual design phase emphasises more on defining what is possible in the form of regions in the design space, rather than trying to find a solution which is dominated as in PBCE approach. The result is a relatively large initial set with a number of possible design solutions (Bernstein, 1998).

Eliminating a design solution within the set is done based on information gathered about its inferiority to other solutions, incompatibility with other solutions, or immaturity of its technology. It is important to build a solid knowledge basis to store and reuse the information related to solutions which have been eliminated, in the next development projects. While sets are narrowing down the level of details in design is also increased. Before making commitment to keep a set at each level of details, it must prove to be feasible and meet the product's requirement (feasibility before commitment as (Sobek et al. (1999) stated). The level of details in tests should be appropriate to the level of details in sets. In other words, the detail of the tests is in direct relationship with the size of the sets. For instance, at initial stages, when the set is still large but with low level of details, tests are simple and quick to only determine the probability of failure in meeting the requirements instead of proving success.

A critical point to notice while narrowing down sets, as Sobek et al. (1999) stated, is to stay within the committed set. This keeps the early communications valid, while are still enriched by more precise information, and allows the reliance of other design teams on the communicated information. Integrating the solutions in SBCE approach results in optimization of the system (Bernstein, 1998). Finding an intersection between sets increases the possibility for the solution contained in the intersection to be globally optimal and acceptable by all design teams. Inability to find such an intersection forces design teams to expand the sets.

SBCE facilitates concurrent engineering in the design phase (Levandowski, 2014) by increasing the likelihood of communicating proper information between design teams, and improving the use of early design information. Communication about sets of design concepts provides each design team with the information about the limits they face, and results in more-informed decisions which consider the needs of other teams (Bernstein, 1998). In addition, the likelihood of changing information when communicating about a single concept in PBCE is much higher than sets of concepts where the level of their data certainty is precisely indicated. The result would be an effective use of preliminary information from upstream teams, thus improving the ability to implement concurrent engineering.

2.9 Research gap

LPD is an emerging area of research in the field in NPD projects. The extensive literature review conducted about this topic revealed that although different researchers identified the importance of SBCE and concurrent engineering as key elements of LPD (Morgan and Liker, 2006; Ward, 2007; Hoppmann et al., 2011; Khan et al., 2013), there is no single study which shows their combined impact on the performance of development projects. It is important to use concurrent engineering when it results in time and cost saving in values which justify the additional time and cost imposed by reworks due to the corruption. It is not possible without having better understanding of the dependent behaviour of processes using an appropriate model of NPD projects which underlies activities and phases of the project, and dependency between them as the result from information flow paths. Iteration cycles have been the subject of different researches, however, the relationship between the implementation of SBCE and concurrent engineering approaches, especially on the elimination of reworks is not well documented in literature. In addition, in SBCE, considering the right number of concepts which add value without increasing the process cost needs more attention. It is also not clear in the literature that how the complexity of NPD projects could affect the benefits of implementation of SBCE approach. These are gaps in the literature which this research tries to fill. Current researches on simulation modelling have focused more on providing a general understanding about NPD process structure, and its behaviour over time. Some researchers in this area tried to build a LPD-specific model, but they only focused on a part of the processes. So, there is a need for a single simulation model which covers all aspects of LPD process structure while have practical applications for the companies as a tool for designing their NPD processes.

2.10 Summary

In this chapter, a literature review of the topic of LPD is presented. The term lean was introduced as the result of extensive studies on Japanese manufacturing companies, especially Toyota, to find the causes of the performance gap between them and their western counterparts. Although first introduced as a philosophy for manufacturing operations, the positive impact of lean manufacturing on the performance measures results in extending it to other functions inside the company. NPD is one of the key disciplines in companies with the significant impact on the profitability of companies, as high possibility for improvement in processes. Two major approaches for LPD are introduced based on the adaptation of lean manufacturing tools and techniques for optimizing NPD processes, or extracting LPD specific tools and techniques from TPDS. The second approach shows more reasonable based on evidences provided, thus being adopted for this research. However, the concept of value and waste from lean manufacturing is also considered by the identification of value-added and wasteful iteration cycles in the NPD projects. Following a process-view of LPD, SBCE and concurrent engineering as the process-specific approaches of LPD are identified and selected for the further study in this research. Finally, by identification of the gaps in the literature the aim of this research is defined to investigate the combined effect of SBCE and concurrent engineering on the amount of iteration and rework, and consequently, on the performance of several types of development projects.

3 Research Methodology

3.1 Introduction

This chapter explains the way that this research is carried out including research ontological and epistemological positions, research design, and methods and tools used. It starts with providing details about the research methods and tools employed, continues with outlining the philosophical stance considered in this research and in the end, explains the research design used to answer the research questions, and the method for data gathering.

3.2 Research philosophies and paradigms

3.2.1 Ontological position

Ontology, epistemology, and methodology are constituents of a research paradigm (Blanche et al., 2007). Ontology is defined as the science of being and involves a set of assumptions about the nature of reality. Viewing the reality from an objective or subjective viewpoint determines the answer to questions such as whether the reality occurs naturally or it is the construct of social interactions between individuals. An objective perspective believes in the existence of social entities independent of social actors, while subjectivism tries to find the meaning that social actors attach to social phenomena, as it believes that social phenomena are formed from the perceptions of social actors, and their actions based on those perceptions (Saunders et al. 2009). As the author believes that the phenomena under investigation exist independent of his perceptions and interpretations, an objective ontological assumption is selected for this research. As the result, the organization of NPD processes is viewed as a tangible object which obey rules, regulations, and hierarchy, and uses standard procedures for performing the works.

3.2.2 Epistemological position

Epistemology determines the way through which the knowledge about the reality is obtained (Saunders et al. 2009). There are three epistemological approaches in social science; namely *Positivist*, *Interpretivist*, and *Realist*. *The* positivist Approach primarily uses quantitative methods, while the interpretivist approach is based mainly on qualitative methods and the subjective reality. The epistemological approach implemented in this research is realist which is an intermediate approach between the positivist and interpretivist approaches. In the following, three epistemological positions are briefly explained. Later in this chapter the relationship between the realist approach and the method implemented in this research is more discussed.

3.2.2.1 Positivist approach

According to this school of thought only verifiable assertions, which are based on the observation and experience, could be considered genuine knowledge (Saunders et al. 2009). As the result, a positivist studies social sciences in the same way as a natural scientist studies scientific domains. This epistemological position rules out knowledge claims about unobserved entities, views scientific laws as statements of general and repetitive patterns of experience, and asks for conducting scientific studies in a value-free way (Benton and Craib, 2010). To establish the social science as a discipline for studying social and human life, positivists believe that scientific enquiry methods need to be employed in these studies.

3.2.2.2 Interpretivist approach

This philosophical position argues that the complexity of the social world is a barrier for theorising it in the same way as natural sciences. It is critical, hence, for researchers to enter the social world of research subjects, and see the world from their viewpoint (Saunders et al. 2009).

3.2.2.3 Realist approach

This school of thought is an alternative for positivism and interpretivism as two predominant epistemological positions in social science (Benton and Craib, 2010). Similar to the positivist approach, the realist approach is also related to the scientific inquiry as it assumes a scientific approach to the development of knowledge (Saunders et al.2009), and believes in the existence of an outside world which is independent of the knowledge of the observer. However, unlike positivism, it assumes that the knowledge of reality is the result of social conditioning, and interpretations make the world meaningful (Thomas, 2004). There is a three-step process for building the realist approach. It starts with collecting evidence about the patterns of observable phenomena, continues with identification and explanation of the underlying mechanisms, and finishes with conducting more experiments and observations, while assume that these mechanisms really do exist (Benton and Craib, 2010). Realists are against reductionism which assumes the complex system is the sum of its parts. They see the mechanisms inside and between parts of a system which make the system results to be more than the sum of the results on its individual parts (Spiegler, 2013).

3.3 Research methods and tools

LPD has attracted a reasonable amount of attention in recent years. Characteristics of LPD as defined by Liker and Morgan (2011) make research in this field challenging. LPD as a system of interwoven components needs an integration between people, processes, and tools to be effective (Morgan and Liker, 2006), and looking at its features in isolation breaks the integrity of the system and results in misleading conclusions. As a result, conventional, cross-sectional surveys that look at collections of best practices as independent variables which predict outcomes is not a suitable research methodology in this area. Industrial processes are truly complex and change over the time, so reducing them to one simple factor that is kept stable during the measurement often gives an incomplete picture (Ottosson, 2004). In addition, the reliability and validity of research on industrial processes could not be achieved simply since processes cannot be repeated with the same result. While conducting real-world experimentation is one way to overcome these problems, this approach could be very expensive, time-consuming, or even risky if the decision is proved wrong.

Several researchers have conducted case study researches to empirically study the effect of implementing SBCE. Bernstein (1998) through multiple case studies in the aerospace industry reported 50 percent reduction in rework costs due to having SBCE approach. Raudberget (2010) conducted several case studies in Swedish companies and found significant reduction in design and warranty cost, and time-to-market. Kerga et al. (2014) reported 30 percent cost reduction in materials and manufacturing and enhancement in innovation using SBCE in a case company. In another work Kerga et al.(2016) studied the improvement in the efficiency of multiple derivative NPD projects using SBCE approach and reported more than 20 percent improvement in the cost and speed of the projects while also having higher ROI.

Analytical and simulation modelling have also been methods used by researchers for studying LPD. Using a system dynamics model Ford and Sobek (2005) simulated an NPD process where four car concepts were concurrently developed. Combining the system dynamics model with *Real Option Theory* they concluded that keeping alternatives alive and delaying managerial decision have a potential to significantly increase the value in the NPD project. Using the similar method, Belay et al. (2014) investigated frontloading of NPD projects using SBCE, and stated more than 50 percent cost reduction and 20 percent time reduction. Malak et al. (2009) combined *Multi-Attribute Utility Theory* with SBCE approach which allows designers to make rational decisions in the conceptual design phase without ignoring the inherent uncertainty. They concluded that following this approach results in a better choice of concepts even if designers must make an arbitrary choice. Analytical modelling has been used by Nahm and Ishikawa (2005; 2006), Telerman et al. (2006), Avigad and Moshaiov (2010), Qureshi et al. (2010), and Inoue et al.(2013) to study SBCE, from the point of view of decisions under uncertainty, design optimisation and incorporating designer preferences.

As the phenomena under investigation in this research involves questions of the causality between different elements and mechanisms in the system of interest (NPD), it could be more related to the *Systems Thinking* approach. This approach tries to improve the system by adopting a holistic understanding of it through its parts, links, goals, and feedback mechanisms. Quantitative methods, mainly simulation, are the dominant methods used in this approach.

3.3.1 Principles of Systems Thinking

The development of systems thinking principles in its modern form could be traced back to the earlier decades of twenties century (Mingers and White, 2010). Later, von Bertalanffy recognized the application of systems thinking concepts in what he called *General Systems Theory*. *Cybernetics*, *System Dynamics*, *Systems Engineering*, and *Complexity Theory* are among the examples of the application of systems thinking approach in the field of operations research and management science (Mingers and White, 2010).

Several structural concepts of systems thinking could be extracted from its central idea, which is the dependency of the behaviour of the system to the structure of its components, not to the properties of individual parts. One concept is the *Emergent Properties* of the system, defined as the properties own by the system as a whole, and emerge from the properties of the components, and the structure of relationships between components which make the system. It could be stated that a system at any level is a component of a wider system, or on the other hand, components of any system are, themselves, systems at another level, consisting of components and relationships. Systems at any level have their emergent properties, and are in interaction with each other through the higher-level system structure they all involved in. Results of these interactions are new levels of systems with their emergent properties (Mingers, 2014). As the emergent properties come from the properties of components in a system and their structures, the system needs to be distinguishable from its environment by its boundaries, although defining the boundary is not a trivial task when dealing with complex systems, such as social systems.

The evolution of systems thinking could be viewed in two phases; the first phase consists of the development of the positivist *Hard Systems Thinking*, and the second phase is the emergence of the interpretivist *Soft Systems Thinking*. Systems thinking in the form of a hard systems approach is based on this pivotal idea that the behaviour of a system could not be explained merely by looking at its components and parts. Parts and components are interrelated in a system in a way that the behaviour of the system is irreducible to the properties of its components, or as Aristotle phrased "*whole is more than the sum of its parts*" (Mingers, 2014 P:71). Recognising the importance of wholes over parts in systems thinking is against the reductionist approach to the science as the dominant research approach over past several hundred years, which is based on splitting up entities until reaching to their ultimate components. This paradigm shift started from the biology to explain the complex behaviour of cells as a whole, and then extended to psychology, arguing the occurrence of perceptions and thoughts as wholes in themselves, atomic physics by recognising the subatomic particles as webs of interacting forces, and finally the management science.

The hard systems approach is characterized by the idea that a system could be engineered to meet a particular objective. It is about closing the gap between the present state of the system and its desired future by using quantitative approaches. In this approach, the problems under study need to be well-defined, structured, and quantifiable. For ill-defined and fuzzy problems, characterized by unpredictability and uncertainty, the soft systems approach is more applicable which is characterized by structuring the problem situation, rather than by problem solving. In the soft systems approach the mathematical modelling part of the hard systems approach is abandoned, and the focus shifts towards defining the situation, resolving conflicting view points, and reaching to an agreement about future actions (Forrester, 1994b). The characteristics of hard and soft systems thinking approaches are summarized in Table 14.

Table 14: Hard versus Soft systems thinking (Pidd, 1996; Checkland and Scholes, 1999)

Hard system thinking	Soft system thinking
Purpose	
Goal seeking oriented Taken as a given at the start	Learning oriented Remains problematical at the start
Philosophy	
Positivist approach	Interpretivist approach
System perspective	
Assumes the world consists of systems which could be engineered The complex, dynamic behavior lies in the world	Assumes the world is problematic but could be explored using system models The complex, dynamic behavior lies in the process of inquiry into the world
Modelling	
Assumes system models as models of the world (ontology-based) Shared representation of the real world Models as means for understanding or changing the world	Assumes system models as intellectual constructs (epistemology-based) Representation of concepts relevant to the real world Models as means to support learning
Problem definition	
Talks of "problems" and "solutions" Clear and single dimensional (single objective)	Talks of "issues" and "accommodations" Ambiguous and multidimensional (multiple objectives)
Advantages	
Allows the use of powerful techniques	Available to both problem owners and professional practitioners Keeps in touch with the human content of problem situations
Disadvantages	
May need professional practitioners May lose touch with aspects beyond the logic of the problem situation	Does not produce final answers Accepts that inquiry is never-ending

3.3.2 Modelling as a research method

A model is defined as "*an external and explicit representation of parts of reality as seen by the people who wish to use that model to understand, to change, to manage and to control that part of reality*" (Pidd, 1996: p15). In the management science, models are used for understanding, changing, managing, and controlling reality. Pidd emphasised on simplification in modelling, due the fact that it is not possible or feasible to model an operational system in its full details. Models, as simplified versions of reality, are used extensively in decision making due to their lower costs, timely delivery of needed information, ability in examining subjects which would be experimentally impossible, and providing insights and understandings about the decision problems of interest (Ragsdale, 2008). Figure 14 represents the general process of problem solving using a model as opposed to the experiment in the real world. As shown in this Figure, modelling starts with mapping the problem from the real world into the virtual world (abstraction), analysis and optimization of the model, and mapping back the solution into the real world. Human cognition is usually erroneous and results in inaccurate judgments due to factors such as irrationality in decision making (Ragsdale, 2008). Consequently, the main reason of using models is improving the process of problem solving and decision making by discovering and eliminating misunderstood elements of a problem during building the model, or gaining insights needed to make a decision through careful analysis of a completed model. Problem solving starts with the identification of the problem as its key step. Based on the nature of the problem, the simplest type of models which fits the problem and accurately reflects its characteristics is created and formulated. The selected model then is used to analyse the problem by generating and evaluating alternative scenarios which could lead to a solution. The feasibility and quality of each potential solution needs to be tested, before implementation.

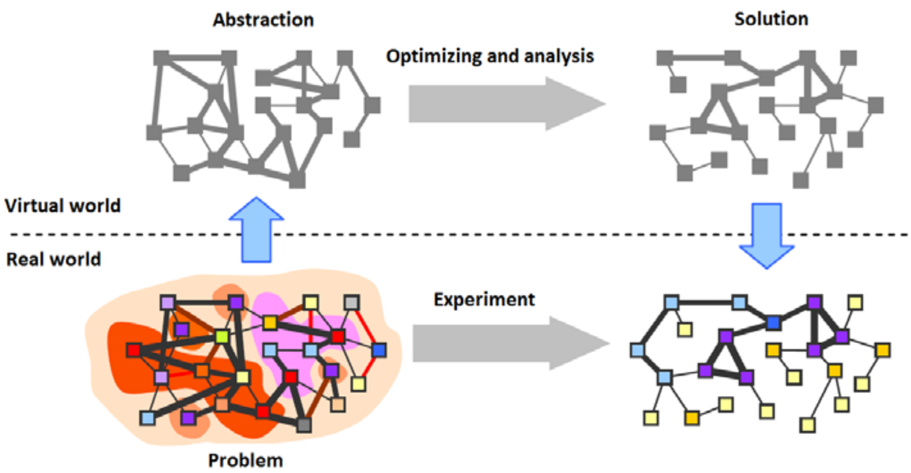


Figure 14: Modelling vs real-world experiment (Borshchev and Filippov, 2004)

Compared to the real-world experiments, modelling has lower cost, and is faster, safer, and more legally compatible while it is also possible to replicate the same conditions in order to repeat the simulation with any combination of decisions. In addition, it is rarely feasible or even possible to conduct a real-world experiment, so it is inevitable to use a model. Based on their level of abstraction models could be classified into four categories as shown in Figure 15. In general, more accurate results are achieved when the level of abstraction in the model is low, while the modelling cost and difficulty increase with decreasing the level of abstraction. The *Operation Exercise*, as the first category with lowest level of abstraction, directly operates with the real environment in which the problem under study exists. External abstractions and oversimplifications are narrowly introduced in this approach, so it has the highest level of reality among other modelling approaches. It involves designing and conducting a set of experiments in the real environment, and analysing their results. Experiments should be designed carefully, consider errors resulting from measurement inaccuracies while evaluating results, and make inferences based upon the performed observations. Similar to the empirical research in the natural sciences, the operation exercise has an inductive approach in essence, allowing generalization of results drawn from observations of a given phenomenon. The drawbacks of operation exercise approach, similar to the real-world experiment, are its high implementation cost, and difficulty in thoroughly analysing available alternatives, which lead to suboptimal conclusions. Due to these problems, although this approach has a significant contribution in improving the managerial decision making, still its usage is limited (Bradley et al., 1977).

The second approach to modelling, according to Figure 15, is *Gaming*, which provides decision makers with a responsive mechanism to check the performance of available alternatives. In some cases, games are identified as a type of simulation modelling. It is a useful tool for helping managers to cope with the inherent complexities in a decision-making process. Human interactions affecting the decision environment are active participants in gaming. Compared with the operation exercise approach, some degree of realism is lost in gaming, due to operating in an abstract environment, although a part of human interactions in the real environment are maintained. However, as the result of the higher level of abstraction, the cost of processing different alternatives would be lower, and the performance of these alternatives would be measured with higher speeds, compared with the previous approach (Bradley et al., 1977).

While in mentioned modelling approaches, human interactions are parts of the modelling process, in two other approaches human is external to the modelling process. As the third approach to modelling, *Simulation* provides a tool for evaluating the performance of existing alternatives. The main difference of the simulation with the gaming is removing human interactions from the modelling process. Similar to previous approaches, simulation is inductive in essence, and does not generate new alternatives or an optimum answer for the problem under study, but just evaluate previously identified alternatives. In simulation, exclusive definition of the problem using analytic terms is not necessary, giving this approach the flexibility in model formulation which is useful in the presence of uncertainties in decisions (Bradley et al., 1977).

The last category of modelling approaches with the highest level of abstraction is *Analytical Models*. In this approach, using an objective function, the problem is completely represented in mathematical terms. Decision conditions are portrayed as a set of mathematical constraints, and the objective function is sought to be maximized or minimized in a way to satisfy all constraints to reach to an optimum solution for the model. This type of modelling approach has the lowest cost and is the easiest model to develop, while having the least accuracy in the results due to the high degree of assumptions simplification (Bradley et al., 1977).

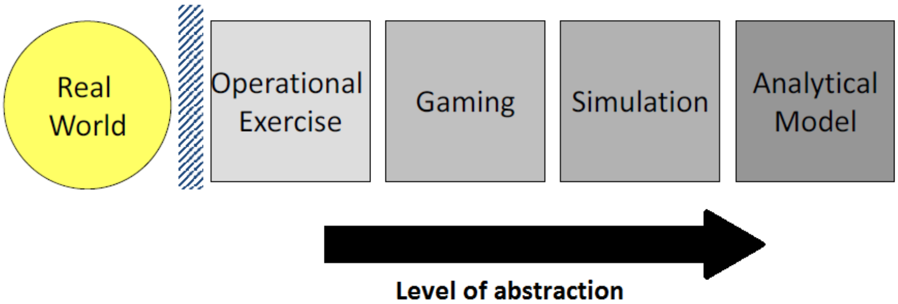


Figure 15: Model classification based on the level of abstraction (Bradley et al., 1977)

In a different approach to the research, Harrison, et al. (2007) stated that theoretical and empirical analyses are two methods which scientific progress has historically relied on. The former is based on the formulation of a set of assumptions as mathematical relationships and deducing the consequences of those assumptions through mathematical proves. According to Meredith et al. (1989), in the theoretical method a complex phenomenon is simplified and primarily addressed with mathematical models. The researcher looks at the problem through mathematical models and tries to find solutions within the defined model and to make sure that these solutions provide insights to the structure of the problems as defined in the model. Empirical research, as the second type, is driven mainly by empirical results and measurements (Burton and Obel, 1995). It starts with the observation of variables in real life and then analysing them to find the patterns of relationships, and is more concerned with the fit between the model and the observation in real-life situations (Bertrand and Fransoo, 2002). Both approaches have some shortcomings, for example, in the theoretical analysis approach, especially in social science, due to the complexity and stochastic nature of social phenomena, analytically determination of results of assumptions using mathematical techniques is very difficult. This difficulty leads to choosing the assumptions mostly based on their usefulness for driving the desired consequences, not their correspondence to reality. In empirical approach, the main problem is the unavailability of data and difficulty to measure them, in addition to the need for comparable measures across a sample or an extended time period (Harrison et al., 2007).

Harrison et al. (2007) considered the introduction of simulation models as the third method of research which allows researchers to handle complex mathematical relationships using computer-based numerical methods. This makes them able to use more realistic assumptions instead of compromising them with analytically convenient assumptions. In addition, using simulation, researchers could produce their own virtual data, thus overcoming the data availability problem in empirical approach, to some extent.

3.3.3 The purpose of modelling

Models are good instruments for explaining a certain type of behaviour in the system (Mark, 2002), examination of the proposed theoretical explanations for phenomena (Denrell, 2004), criticizing the pre-existing explanations and finding simpler ones, prescribing a better method of organizing, and finding an expected covariation between some variables which could be used as a hypothesis in systematic empirical research. Although the idea of using models and simulation in the problem solving and decision analysis is not new, just in past two decades, the power of simulation and analytical modelling as the most effective way for analysing and evaluating decision alternatives in business and management environment has been articulated (Ragsdale, 2008). Different techniques are proposed for the development and implementation of simulation and analytical models. Models could be categorized based on their purposes. The form of relationship between the independent variables (input to the model) and dependent variables (output from the model), and the values of independent variables could be used to determine the purpose of models. If the functional relationship between independent and dependent variables is clear and well-defined, and the decision maker has the control over values of independent variables, the model's purpose is just to determine the values for inputs in a way that result in best possible output values. As these models tell the decision maker what type of action to take, they are categorized under *Prescriptive Models*. If the values of one or more independent variables are uncertain, while the relationship between them and dependent variables is clear, the model is built with the goal to describe the outcomes of an operation or a system. These models are called *Descriptive Models*. Finally, in cases when the functional relationship between independent and dependent variables is unclear, the goal of the model is to estimate the value of dependent variables for specific values of independent variables. This type of models is classified under *Predictive Models* (Ragsdale, 2008). In other words, finding a relationship among variables using the analysis of the model's outputs could be viewed as the predictive power of models. The confirmation of this prediction using empirical testing could provide indirect support of the theory embodied in the model (Harrison et al., 2007). Yet, it is possible for a simulation model to serve more than one purpose or there could be an overlap in the purposes of specific simulation studies.

Table 15: Types of models in management science (Ragsdale, 2008)

	Model characteristics		**Modelling techniques**
	Input-Output relationship	**Input**	
Prescriptive model	Known	Known or under the control of descision maker	Linear Programming, Network Podels, Integer and Mixed Integer Programming, Nonlinear Programming
Descriptive model	Known	Unknown	Simulation, Queuing Theory, PERT, Inventory Models
Predictive model	Unknown	Known or under the control of decision maker	Regression Analysis, Time Series Analysis

3.3.4 Simulation modelling

Complexity in the real-world systems usually does not allow to evaluate these systems realistically using analytical and mathematical models. The ability of traditional theory development approaches in analysing complex systems with multiple interdependent processes which operate concurrently is limited (Harrison et al., 2007). This is because of complicated and unforeseen ways of interaction in these processes. The result of these interactions is generally a nonlinear system behaviour with the feedback, so empirical analysis which uses linear models offer limited value. In these cases, simulation helps to evaluate a model and to estimate its desired characteristics (Law and Kelton, 1991). According to Kotzab et al. (2006) simulation offers a middle ground between pure analytical modelling and empirical observation and experiments. Even without analytically solvable mathematical equations, simulation models could proceed using numerical approximation methods (Spiegler, 2013). The simulation modelling as a more systematic method for developing and analysing theory (Harrison et al., 2007) offers higher value when is used for studying complex organizational behaviour. Although the simulation modelling has gradually become accepted as a research method in social science its potential has not yet been completely used for contributing to management theory. Harrison, et al. (2007) measured the proportion of articles in leading management journals[1] which mainly used simulation modelling as their method, in the period of 1994-2003, and concluded that compared to other social science branches, such as psychology, economics, and political science, using simulation methods in management studies has less an impact.

A simulation model is an abstract representation of the system, and contains structural, logical, or mathematical relationships which describe the state of the system. Harrison et al. defined a simulation model as "*a computational model of system behaviour coupled with an experimental design*" (Harrison et al., 2007 P: 1234). In *Operations Management* "*simulation models are based on a set of variables that vary over a specific domain, while quantitative and causal relationships have been defined between these variables*" (Bertrand and Fransoo, 2002 P: 242). The first step in using simulation analysis in management studies is to understand the way they work and the benefits they offer, as well as their probable shortcomings. Simulations are formal models including a precise formulation of the variables relationships and the way their values change over time (Harrison et al., 2007). As the result, building a simulation model starts with recognizing these underlying relationships which have key effects on the behaviour of the system under study, and then continues with formalizing them as the mathematical equations or a set of transformation rules. Because the results of the complex interactions between the components of a simulation model is not clear, normally hypotheses could not be set in this type of research, thus the entire process of simulation, which starts with assumptions and ends with findings as the prediction of the theory, makes a methodology for developing theory.

The development of several software packages in recent years has eased the simulation modelling of large-scale and complex systems. The real-world system of interest which is intended to be study using simulation is defined as a collection of entities which act and interact together over time to accomplish one or a set of logical goals (Law and Kelton, 1991). The state of the system is a collection of variables which are required to describe that system in a particular time and relative to the purposes of study. The elements of a simulation model are variables, and rules and equations for changing these variables. These equations and rules determine the state of the system at time ($t+\Delta t$) based on the state of the system at

[1] Academy of management journal, Administrative science quarterly, Management science, Organization science, Strategic management journal

time t. Variables, in general, are dependent or independent. The value of the independent variable (for instance, x) determines the value of the dependent variable (y) through a causal relationship, so y is defined as a quantifiable function of x (y=f(x)), and any specific change in x explicitly results in changes in y. Although in other types of research, such as survey research, relationships are also defined between variables, these relationships are not always causal and quantifiable. Being quantifiable and causal helps simulation modelling research to be also used for forecasting the future state of the system under study, instead of merely explaining the current state.

In order to carry out the computation, some constant parameters should be set by the researcher in mathematical equations. During the validation of the model using the empirical data, the value of these parameters could be verified or changed. Initial conditions, or the state of the system at time zero, should also be determined in advance, in addition to the time structure, the outcome determination, and variations. The length of each time period of simulation and the number of time periods in a simulation run are determined as the time structure. The model outcomes are calculated from the variables in each time period, as a function of the behaviour of the system. Repeating the entire simulation process may be done using different initial conditions and parameter values to examine the effect of these changes on the overall behaviour of the system. The analysis of the sensitivity of the simulation model to variations in parameters and initial conditions is another reason for repeating the simulation to see if the behaviour of the system is robust or needs further investigations. The results of simulation runs are in the form of the values of variables and outcomes for each time period, in addition to parameter settings and initial conditions which could be analysed, to some extent, in the same way as empirical data (Harrison et al., 2007).

3.3.5 Types of simulation models

Simulation models could be classified along three different dimensions as shown in Figure 16 and briefly discussed below (Law and Kelton, 1991).

- *Static* vs. *Dynamic* simulation models: if the simulation model represents the system at a specific time, or if time does not play a role in the model, that model is static, such as *Monte Carlo Simulation*. On the other side, in a dynamic model, the system is represented as it evolves during a time period.
- *Deterministic* vs. *Stochastic* simulation models: deterministic models do not contain any random variables, so their output is determined by knowing the input values to

the model and the relationships between inputs and outputs. In stochastic models, there is at least one random variable among independent variables, so the output of this type of models is random itself and should be treated as an estimation for the true characteristics of the model. In this type of models, the simulation should be repeated several times to produce output distributions which shows the system's behaviour, while in deterministic models, ceteris paribus, each run produces the same results.

- *Discrete-event* vs. *Continuous* simulation models: if the system is modelled as it evolves over time and changes in the system state happen at some separate points in time, it is called a discrete-event simulation model. If the state of the system changes continuously over time, it is called a continuous simulation model. Discrete-event simulation models are characterized by blocks of time during which nothing happens, punctuated by events which change the state of the system (McHaney, 2009). Discrete-event simulation is more appropriate for systems which need to be modelled in detail, especially when individual items should be tracked through the system (Robinson, 2014). Warburton and Disney (2007) argued that management insights gained from discrete and continuous time modelling approaches are very similar as their qualitative nature is essentially equivalent. However, according to Tako and Robinson (2012), it is widely accepted that for modelling problems at a strategic level, system dynamics (continuous time modelling) is more implemented, while discrete-event modelling is more used at an operational and tactical level.

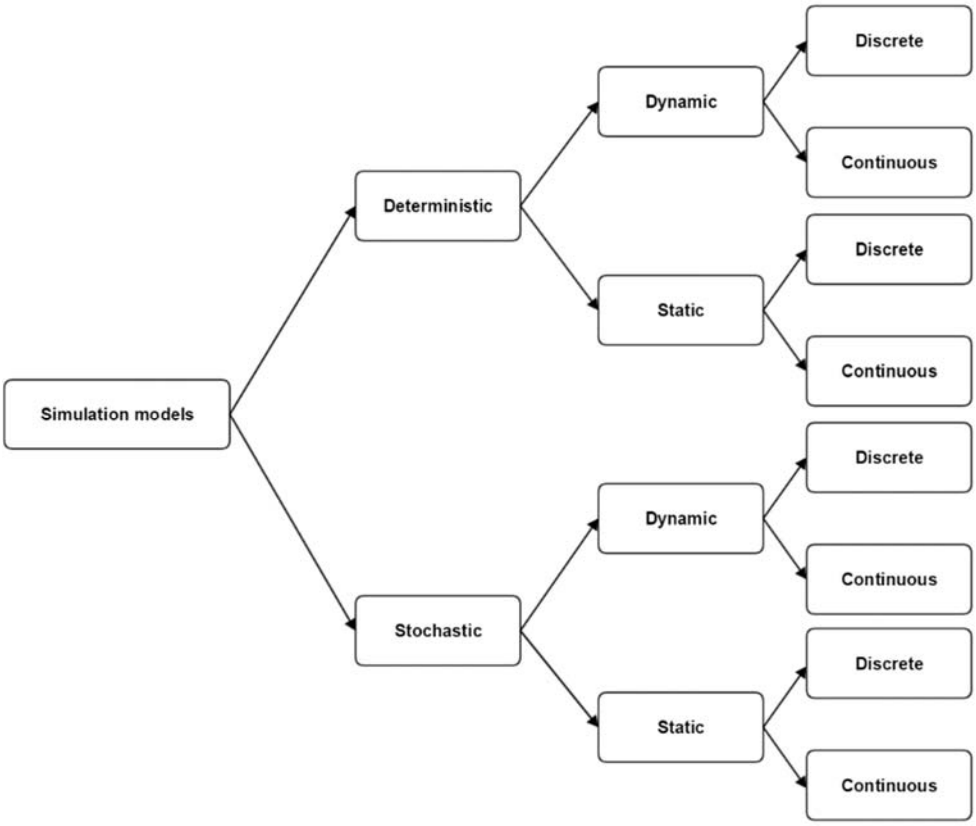


Figure 16: simulation modelling taxonomy

In addition to mentioned typologies, simulation models could also be divided between two types of *Agent-Based models* and *System Dynamics Models* (Harrison et al., 2007). In agent-based models the focus of modelling is on the behaviour of agents (actors) who make a social system and interact with each other, and the behaviour of the system is the result of these interactions. Agents could be individuals, a group of people interacting in an organization, or organizations interacting in an industry. The outcome of interest in agent-based models is the effect of interactive actors' behaviour on the entire social system.

While in agent-based models the system's behaviour is modelled indirectly through the interaction of its actors, in system dynamics modelling the focus is on the modelling of the target system as a whole, by simulating the processes which lead to changes in the system through the time, instead of focusing on the behaviour of actors within the system (Forrester, 1961). While agent-based models are specified using equations, rules, or a combination on them, formalization in system dynamics modelling is done mainly using the differential equations, but generally, the decision to formalization using equations, rules or both does not have any intrinsic reason and depends on the nature of the system being modelled, as well as the preference of the researcher (Harrison et al., 2007).

3.3.6 Simplicity vs. reality in simulation models

In addition to several advantages of simulation models, there are also some issues related to the level of complexity of models which need to be addressed. Pidd (1996) defined a model as an external and explicit representation of a part of reality, and Box (In Browning et al., 2006 P: 105) stated "*all models are wrong, but some are useful*". This is mainly due to the complexity of the operational processes which makes them difficult to be modelled. Not all aspects of reality could be included in the model, so a model is always an abstraction of the reality (Burton and Obel, 1995; Bertrand and Fransoo, 2002; Briggs, 2007; Harrison et al., 2007). There is no doubt that adding more variables and relationships to a model makes it more realistic, but also increases its complexity and results in difficulty in interpretation of model outcomes (Harrison et al., 2007). Simplification of problems means only those elements are included in the model which are assumed to be relevant from the viewpoint of the purpose of the model. This simplification further extends the abstraction of the model from reality to make it more manageable for the analysis. Ignoring unrelated aspects of the problem in simulation modelling is based on the assumption that these aspects have no influence on the effectiveness of solutions, and the key elements of the relevant processes are retained. Nelson and Winter (1982) stated that artful simplification of models makes them more understandable and transparent, and is the symbol of skilful modelling.

This simplification, however, has some disadvantages, because a part of key elements might be accidently excluded from the model, thus limiting the usefulness of the simulation model as an aid for understanding the system's behaviour. Although determining the level of simplicity suitable for the modelling of a complex system before starting the modelling is difficult, intuition and objectives of the researcher and the nature of the system under study could be used as guides for making decision about the relevancy of different system elements (Nelson and Winter, 1982). Overall, there should be a balance between simplifying the model and including enough elements to help the understanding of the behaviour of the system.

3.4 System Dynamics Modelling

System dynamics modelling as the main method employed for this research is discussed in detail in this Section. System dynamics belongs to the bigger category of Systems Thinking approaches which are discussed in detail in previous section. In its initial form, as introduced by Forrester (1961), System dynamics was a hard systems thinking approach. However, later

Forrester (1994a) stated that the soft systems approach covers the conceptualization phase of the system dynamics modelling. Similarly, Senge (1992) and Sterman (2000) made a distinction between the behaviour of the system and its underlying unobservable pattern of causal relationships which builds the system structure and generates its behaviour. This shows the combination of hard and soft approaches in the systems dynamics and moves system dynamics from the fully hard concept towards a softer paradigm. Figure 17 represents the position of system dynamics in relation to the hard and soft systems approaches.

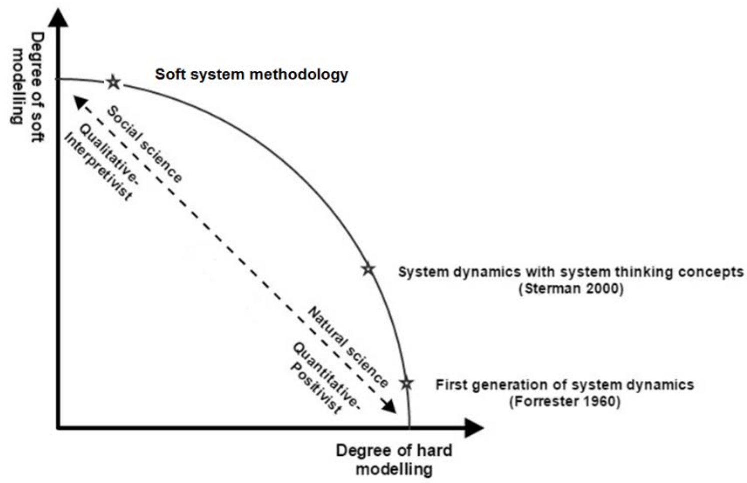


Figure 17: the position of system dynamics approach in relation to the hard and soft systems approaches

In the following section basic concepts and assumptions of system dynamics modelling and its philosophical standpoint are presented and approaches to the process of building a model are reviewed.

3.4.1 Basic concept and assumptions of system dynamics

System Dynamics was introduced by Forrester in 1958, under the name of *Industrial Dynamics* which reflects its origin as a modelling tool for solving problems in industrial supply chain. The first idea was to use the concept of control engineering in solving management issues. Later, Meadows expanded the application of this method beyond the industrial topics (Dangerfield, 2014). As the flows in the system are assumed to be continuous, system dynamics could be classified as a continuous simulation modelling method. In system dynamics issues are addressed in the form of aggregate entities, unlike discrete-event simulation models which uses individual entities, or agent-based modelling which uses individual agents. Although the initial stimulus for the dynamics of the system in this approach could be exogenous to the system, in general, system dynamics reflects that the dynamics is the result of information feedback effects and component interactions, inside the system. From the ontological point of view, system dynamics assumes the complex interconnection between parts of a system. It views the world as a system made up of stocks, flows and feedback loops, in which delays and nonlinearities play a crucial role, and the behaviour arises out from the structure (Meadows, 1989). As the result, the foundation of system dynamics method is around recognizing the structure of the system, and then, trying to explain the long-term behaviour of the system based on its internal structure. In most applications, the focus of system dynamics modelling in on the analysis of dynamic systems using the methods and tools it offers (Größler et al., 2008). Quantifying the variables and linking them together create a system of differential equations which could be simulated using numerical algorithms (Sterman, 2000). The validity of models could be supported using a variety of existing tests which are always relative to the purpose of the model (Sterman, 2000; Barlas, 1996; Qudrat-Ullah and Seong, 2010).

The epistemological position of system dynamics is shaped around the mental models as cognitive structures (Forrester, 1971). Olaya (2009) defined mental models as images of the selected relationships and concepts in the real world which are used for the explanation of particular systems' behaviour. Although in some cases these mental models could provide highly reliable information about the structure and relationships in a dynamic system, their accuracy has been criticized because of the arguments related to the concept of '*Bounded Rationality*' (Todd, 2001), according to it the human mind could not correctly sense the complex consequences of decisions (Forrester, 1971). It is the limited capability for processing information which results in simplification of causal relationships in complex systems. In addition, the limitation in memory and cognitive skills for viewing the dynamics in systems with feedback loops, results in the inability to see the consequences of initial assumptions. To overcome these cognitive limitations in understanding complex systems and their behaviours, system dynamics models would be helpful.

The claim of system dynamics modelling is building the model of social systems (Forrester, 1961 in Lane, 2001a) which means the subject of modelling is situations involving human agents as active parts who make decisions according to the existing policies, and using available information. It involves a set of powerful concepts and tools for modelling the complex system's behaviour. The patterns of behaviour which feedback loops create could be found in several types of system, which result in the applicability of system dynamics in different fields, from healthcare and environmental studies to supply chain, production, and development projects (Mingers and White, 2010).

Starting point and finishing point of system dynamics models are the mental models of individuals involved in these modelling projects (Größler et al., 2008). They start using mental models as the most important source of information, especially about the social systems, and finish by generating new insights about the relationship between the structure and behaviour of the system, thus improving mental models. These generated insights could also be used for changing policies and structures in real-life systems to improve the performance in organizational contexts (Größler et al., 2008).

Following, feedback loops, and accumulation and delays as the main characteristics of a system dynamics model are presented briefly.

3.4.1.1 Feedback loops

Feedback loops are the main building block of systems in system dynamics, through which information about the current state of the system is used to adjust controls. When information related to some actions circulates around the system and finally returns to its point of origin makes a feedback loop which could influence the behaviour of the system in future (Richardson, 2009) in two distinctive ways; it could be a *Positive Feedback Loop* and reinforce the behaviour of the system, or it could be a *Negative Feedback Loop* and balance the system's behaviour. Reinforcing (positive) feedback loops represent a change which propagates through a system and produces more change in the same direction, which is the characteristic behaviour of the exponential growth (Forrester, 1994a; Sterman, 2000) (Figure 18). Negative feedback loops seek balance, equilibrium, and stability (Sterman, 2000). They act to bring the state of the system in line with a goal or desired state by comparing the desired and actual conditions and taking corrective actions (Sterman, 2000). When the gap between actual and desired conditions is linearly related to the corrective action, the rate of adjustment is exactly proportional to the size of the gap and the resulting goal-seeking behaviour becomes exponential decay (Figure 19) (Sterman, 2000). The shift from an exponential growth to a goal-seeking behaviour which produces an *S-shaped* behaviour does not depend on the number of reinforcing and balancing feedback loops in the system, but the shift in the dominance of loops determines this special type of behaviour. Shifting from the reinforcing to balancing feedback also shifts the behaviour from the exponential growth to the goal seeking, while it is the strength of these loops which determines the characteristics of the S-shaped behaviour (Figure 20) (Forrester, 1994a; Sterman, 2000).

A system dynamic model consists of several feedback loops which are linked together, representing a close system with endogenous and exogenous variables. The presence of system elements, including stocks and flows, which create delays in feedback loops affect the timing of the system's behaviour. Using these system elements, the researcher would be able to interpret the system's behaviour and find causal hypotheses about these behaviours.

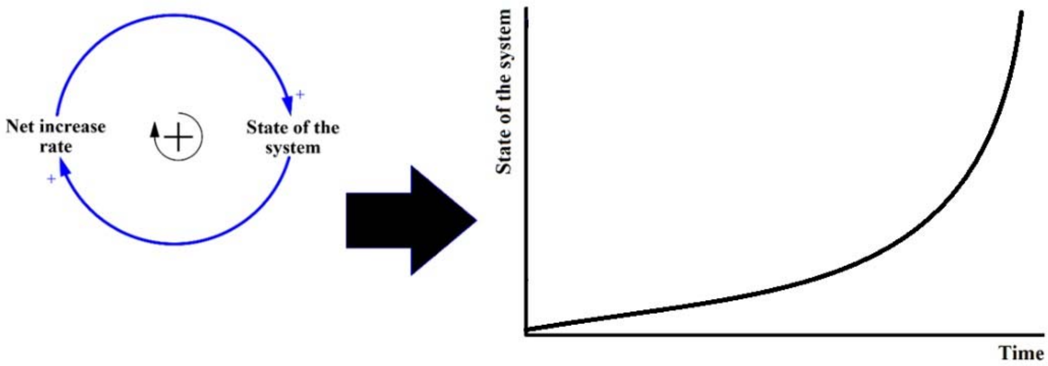


Figure 18: Exponential growth behaviour and its general structure (Sterman, 2000)

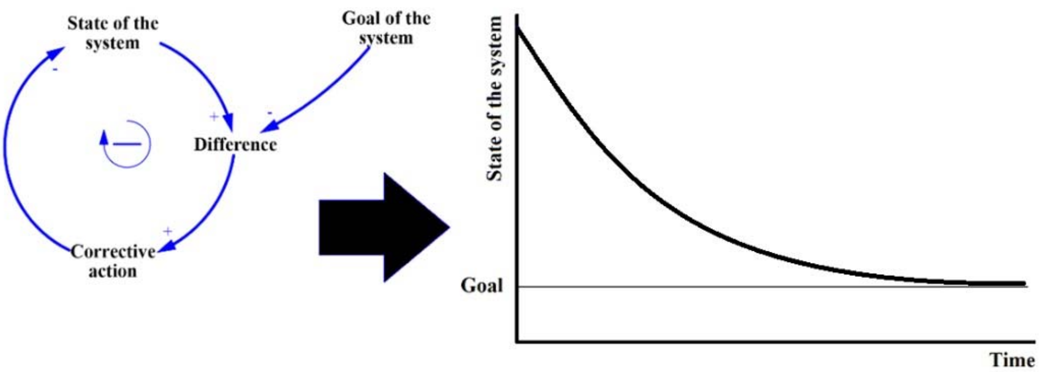


Figure 19: Goal seeking behaviour and its general structure (Sterman, 2000)

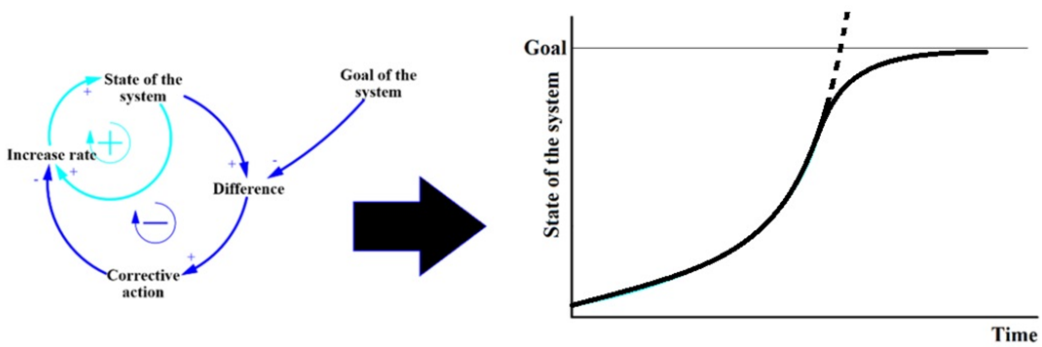


Figure 20: Shift from exponential growth to exponential decay (S-shaped behaviour) (Sterman, 2000)

3.4.1.2 Accumulation and delays

The concept of accumulation has been argued by some researchers to be even more important than feedback loops in the system dynamics (Größler et al., 2008). *Stocks* (*Levels*) and *Flows* (*Rates*) are two main types of variables in system dynamics. There is at least one stock variable in each feedback loop which represents the accumulation and conserves the state of the system. Stocks characterise the path of the system through time by incorporating the history and determining the future of the system. Flows contain mechanisms for changing the state of the system. The idea of delay is closely connected to concepts of feedback loops and accumulation in system dynamics. Delays are the direct effects of making decisions, and always present in business processes (Größler et al., 2008). System dynamics models help to analyse the effects of delays by compressing them within an acceptable time frame. The combination of several feedback loops, accumulations, and delays often results in nonlinear behaviour of the system, which is hardly understandable by the human cognition. While analytical solutions are not usually available for them, simulation is the best way to explore and analyse these systems.

3.4.2 Philosophical standpoint of system dynamics

As system dynamics is rooted in engineering (Forrester 1960; Richardson 1991), the language that system dynamics researchers have used have nothing in common with social scientists (Lane, 2001b). Because of that, it is very seldom that system dynamics literature mention the social theory basis for their research. However, several researchers tried to find the fit between system dynamics and social science paradigms. For instance, Lane (2001a) stated that system dynamics could not be placed in any one of traditional paradigms, such as positivism and interpretivism. He found strong similarities between system dynamics and social theories which try to integrate agency and structure (Lane, 2001b; Mingers and White, 2010), and suggested exploring contemporary social science theories to find more appropriate ones for the field of system dynamics.

Mingers (2000) argued that *Critical Realism*, as a philosophical perspective which involves the integration of agency and structure, is matched with system dynamics. Critical realism was developed in response to the critics against realism (Mingers, 2000). It started with the argument that unlike the positivist approach, the scientific reality is not just about the coexistence of observable events, but there are also unobservable entities which create these observable events. So, a domain of events independent of the perception of social actors should exist which only a small part of it could be perceived and experienced empirically. The bigger part remains unknown, which does not mean that it does not exist (Mingers, 2000). It could be argued that causal laws and mechanisms are independent of the events they generate. While positivists and empiricists argue that anything that could not be experienced does not exist, and only things exist which could be perceived, according to the critical realist approach the ontological domain of existence should not be reduced to the epistemological domain of knowledge. Accordingly, this approach emphasises on the dominance of ontology over epistemology by arguing that "*world would exist whether or not humans did*" (Mingers, 2000 p: 1261); in other words, having a causal effect on world means existence.

In the critical realist approach, reality exists independent of social actors, in the form of structures and causal mechanisms (*Real Domain*), events resulting from these structures and mechanisms (*Actual Domain*), and experiments as a small part of these events which could be observed and experienced (*Empirical Domain*) (Mingers, 2000; Gorski, 2013) . In addition, structures, and causal mechanisms (*Generative Mechanisms*) interact dynamically, thus resulting in the emergence of other mechanisms and structures. Figure 21 illustrates these domains in relation to each other, and generative mechanisms and observations.

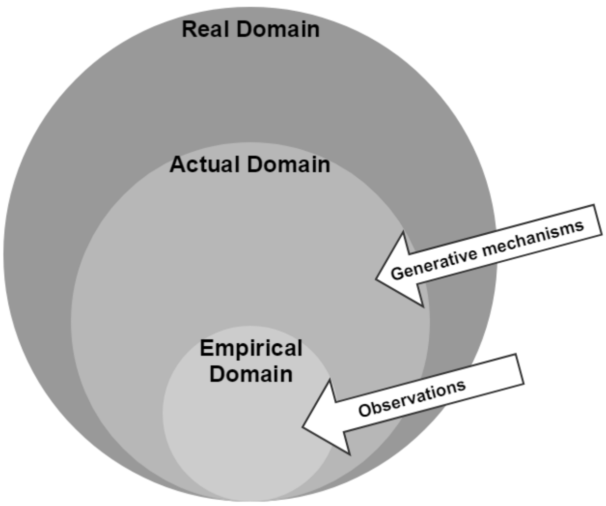


Figure 21: three domains in critical realism

According to Mingers (2000) system dynamics seems to embody some of the important principles of critical realism. The correspondence between system dynamics and critical realism could be shown as followed:

- System dynamics has its roots in systems thinking and systems approach, which believe in having a holistic view to the system, instead of reducing it to a set of diverse elements. It puts higher emphasis on the interactions between elements as drivers for the system behaviour than on elements themselves (Mingers and White, 2010). This view is in correspondence with the description of the real domain in critical realism.
- In system dynamics, changes experienced over a period of time are the result of feedback structures and causal relationships, so the structure of the system is the driver of its dynamic behaviour. This argument corresponds with the division between the structure and the mechanisms, as the real domain, and events, as the actual domain in the critical realist approach.
- In system dynamics, the balance between feedback loops and their relative strengths determine the actual behaviour of the system. It is similar to the idea of generative mechanisms in the critical realist approach.
- In system dynamics, particular structures could be found in systems (systems archetypes as Senge (1992) called them), which generate particular patterns of behaviour. As feedback loops could be combined in a limited number of ways, system archetypes are distinguishable based on the position and the relationships between different types of feedback loops. Understanding system archetypes leads to the holistic understanding of the system, by viewing beyond its apparent behaviour (Mingers and White, 2010). Generative mechanisms in the critical realist approach, as potential properties of the system which are caused by a particular structural configuration, are matched with the system archetypes concept in system dynamics.

3.4.3 Modelling process

As mentioned before, formal simulation models are tools which used to make reliable inferences about the behaviour of complex systems. There are different approaches to the process of building a system dynamics model, implemented by researchers. The approach described by Richardson and Pugh (1981), as shown in Figure 22, the process starts with a descriptive model to define the problem. This leads to developing a conceptual model as a causal loop diagram. The conceptual model, then, is converted to a stock-and-flow diagram as a formal quantitative model using mathematical equations. This formal model then is simulated and validated to test the related hypotheses and policies.

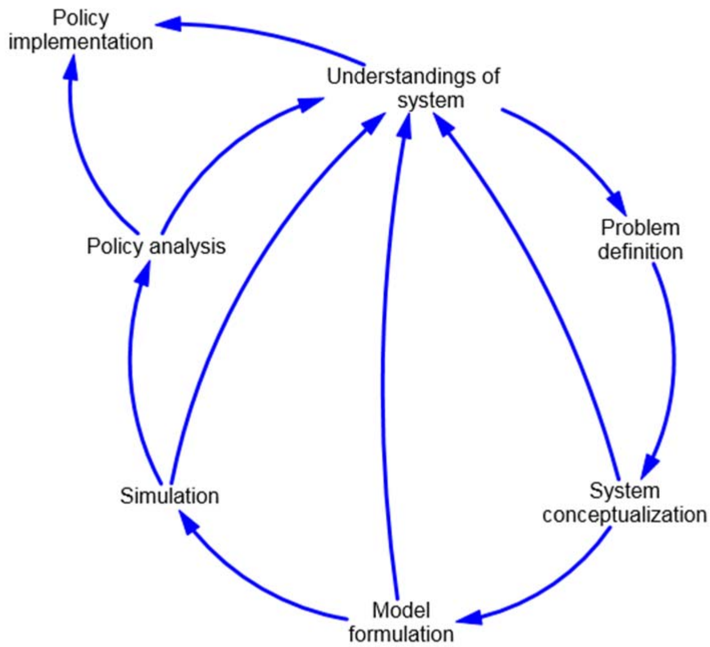


Figure 22: System dynamics modelling process by Richardson and Pugh (1981)

Another approach addressed by Forrester (1994b) contains six interconnected steps starting from the system description and ended with policy changes in a highly iterative manner (Figure 23). He mentioned the conceptualization phase as the most critical and most difficult step in building a model. The process starts with the creation of a hypothesis which relates the behaviour of the system to its structure. This hypothesis in the next step is formulated using system dynamics concepts. The model, then, is simulated to identify the problematic policies which cause problematic behaviour in the system. Substitution of these policies with alternatives requires intense education and debates to overcome resistances against changing the traditional beliefs and actions in organizations (Forrester, 1994b). If the model is relevant, and education is sufficient, implementation of new policies as the last step of the process would not be so difficult.

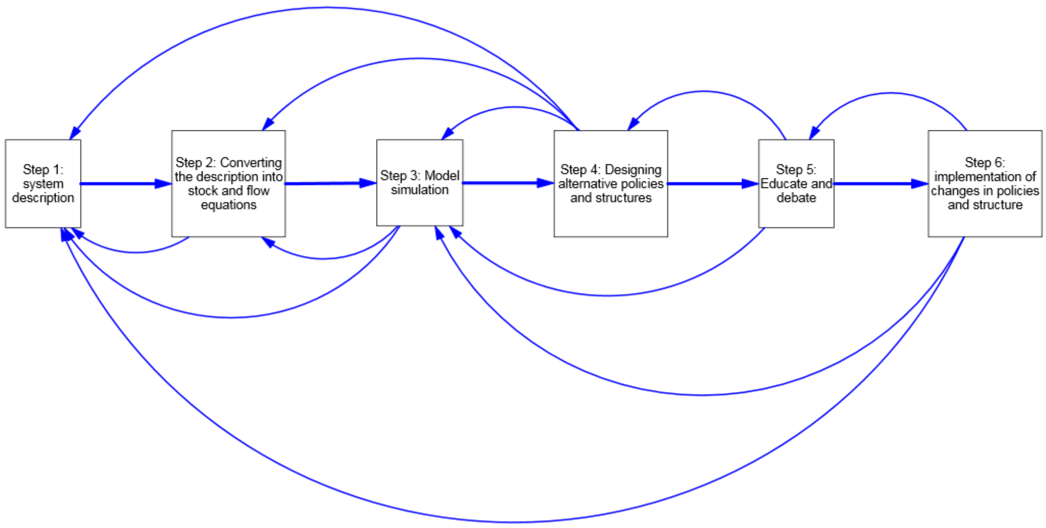


Figure 23: System dynamics steps by Forrester (1994b)

The final approach to the process of system dynamics modelling is introduced by Sterman (2000). Real world in the outer layer of the model in Figure 24, which represents a learning and action cycle, is the place where strategy, structure and decision rules are built, based on mental models. These mental models are adjusted through a trial and error approach by comparing the achievements with intentions (Morecroft, 2015). The process of system dynamics modelling embedded in this learning and action cycle is helpful in clarification and improvement of mental models. The information from real world and mental models, which informs policies, strategies, and decision rules, could be virtually tested using the simulation model and results could be fed back to the real world to change the mental models and design new policies. Implementation of these new policies in the real world and their effects provide feedback which lead to further improving the mental and simulation models. In Sterman's approach system dynamics modelling is an ongoing and highly iterative process of feedback cycles between the real and virtual world. The process (inner cycle in Figure 24) includes five steps, as followed:

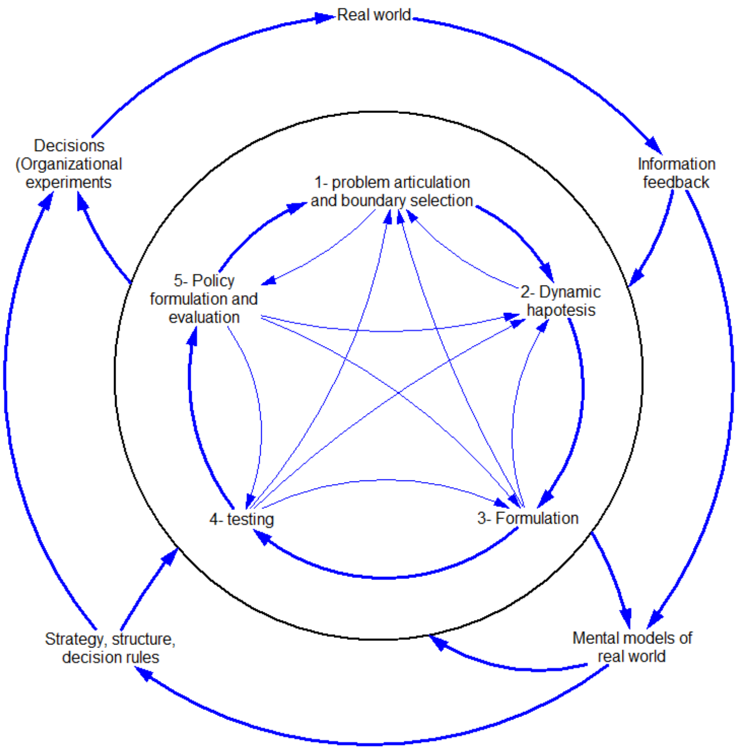


Figure 24: System dynamics modelling process by Sterman (2000)

1- Problem articulation

Problem articulation as the key step in modelling determines the purpose of the model and the problem which the model is trying to address. A clear purpose always helps revealing the usefulness of the model in addressing the problem. To be useful, the model should simply address a specific problem rather that mirroring the entire system in detail. The usefulness of the models lies in the fact that they simplify reality, by creating a comprehensive representation of it. The purpose of the model is the criterion for making decisions on the boundary of the model. Without the purpose, the model would have a very wide boundary and include an overwhelming array of variables. Base on the model boundary, key variables, and anticipations for the key factors within the boundary of the study are also determined.

2- Formulating a dynamic hypothesis

Next step after identifying the problem, is developing a *Dynamic Hypothesis*. The dynamic hypothesis is a theory about the effects of the structure and decision policies on generating the system's behaviour. It is a hypothesis because it is always provisional, subject to revision or abandonment. The rest of the modelling process is mainly about testing this dynamic hypothesis, both with the simulation model and by experiments and data collection in the real world. A variety of tools, such as causal loop diagram and stock-and-flow diagrams, help representing the boundary of the model and its causal structure. Causal loop diagrams are flexible tools for diagramming the feedback structure of a system and the causal links among variables in any domain, and stock-and-flow diagrams track accumulation of material or information as they move through the system. Stocks characterize the state of the system and generate the information upon which decisions are based. The decisions change the flows, altering the stocks' value, and close the feedback loops in the system.

3- Formulating a simulation model

Testing the dynamic hypothesis in a virtual world needs a fully specified formal model with equations, parameters, and initial conditions. Formulation helps in recognizing unclear concepts and resolving unnoticed contradictions during the conceptual phase. Sometimes finding inconsistencies in the mapping of the concerned subject matter may force the researcher to return to the previous steps. Several computer software packages [2] are available to run system dynamic simulations.

4- Testing

System dynamics practice includes a large variety of tests for making sure that the model works correctly, identifying flaws in proposed formulations and improving the understanding of the system. Testing includes comparing the simulated behaviour of the model to the actual behaviour of the system. Each variable must correspond to a meaningful concept in the real world, each equation must be checked for dimension consistency, and the sensitivity of the model behaviour must be assessed considering the uncertainty in assumptions, both parametric and structural. The result of testing may force returning to the first three steps to fix problems. These iterations continue until the model gains the required level of confidence and become adequate for its purpose (Forrester, 1994b).

[2] Available software packages for modelling system dynamics with quite similar performances are DYNAMO by Pugh-Roberts Associates; STELLA or iThink from High Performance systems Inc; Vensim from Ventana Systems Inc; Microworld Creator by Microworlds Inc, and Powersim from Powersim Corporation.

5- Policy design and evaluation

After developing the confidence in the structure and behaviour of the model, it could be used to design and evaluate policies for improvement. Policy design includes the creation of entirely new strategies, structures, and decision rules. Since the feedback structure of a system determines its dynamic, most of the time, high leverage policies involve changing the dominant feedback loops by redesigning the stock-and-flow structure, eliminating time delays, changing the flow and the quality of information available at key decision points, or fundamentally, reinventing the decision processes of the actors in the system.

3.4.4 Application of system dynamics modelling in management research

System dynamics modelling has been used in a variety of researches in the field of operations management. Größler et al. (2008) identified five major areas of application of system dynamics as a method in operations management research, as followed:

- research about issues in the production flow and supply chain management;
- research on improvement programs in operations;
- researches addressing issues in the project management;
- research in the field of new product development, innovation, and diffusion;
- research about the effects of different production technologies.

NPD project management has been an example of success for the application of system dynamics modelling. Lyneis and Ford (2007) categorized the structures used in system dynamics models of projects in four groups:

- project features, including development processes, resource allocation, managerial mental models and decision making;
- rework cycle as the well-recognized and most important feature in the system dynamics models of development projects;
- project control efforts through which managers try to close the gap between project performance and targets and meet the project deadlines;
- ripple and knock-on effects as the primary and secondary side effects of project control efforts and the concept of policy resistance in projects.

Due to similarities in the basic structures of development projects across industries, mentioned structures become foundations for studying the dynamic of development projects. Using these project structures researchers mainly tried to find explanations and suggest improvements for the failure of NPD projects to meet performance targets, as the most cited behaviour of NPD projects in literature. The failure in meeting the project targets is shown in Figure 25 and Figure 26 as the difference between the planned and the actual resource allocation, and the progress over time, respectively. Compared to the planned trajectory, the actual resource allocation in Figure 25 starts with a delay, overshoots the planned pick and remains longer at the peak, mainly due the overestimation of the productivity of resources and the underestimation of the project scope by the managers, unavailability of resources, and the impact of ripple and knock-on effects on the productivity and the quality of work (Lyneis and Ford, 2007). In Figure 26, a period of slow progress close to the end of the project could be due to finding the quality issues later in the project (Ford and Sterman, 2003).

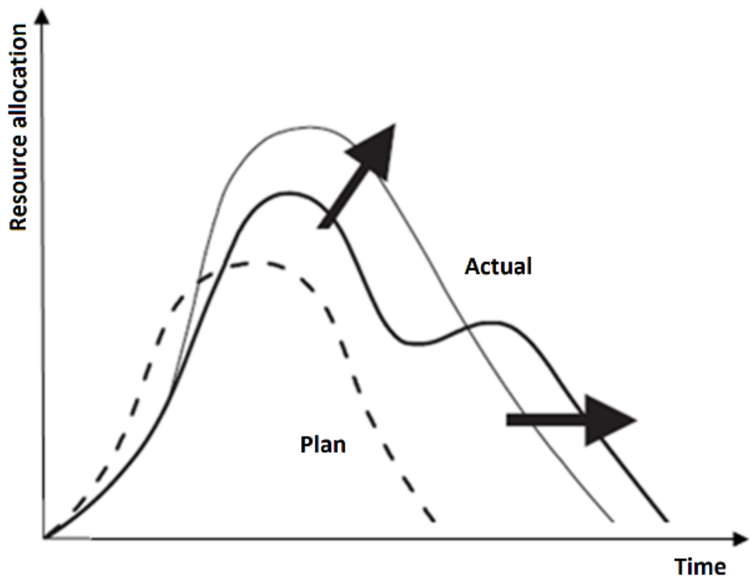


Figure 25: Difference between planned and actual resource allocation to NPD projects (Lyneis and Ford, 2007)

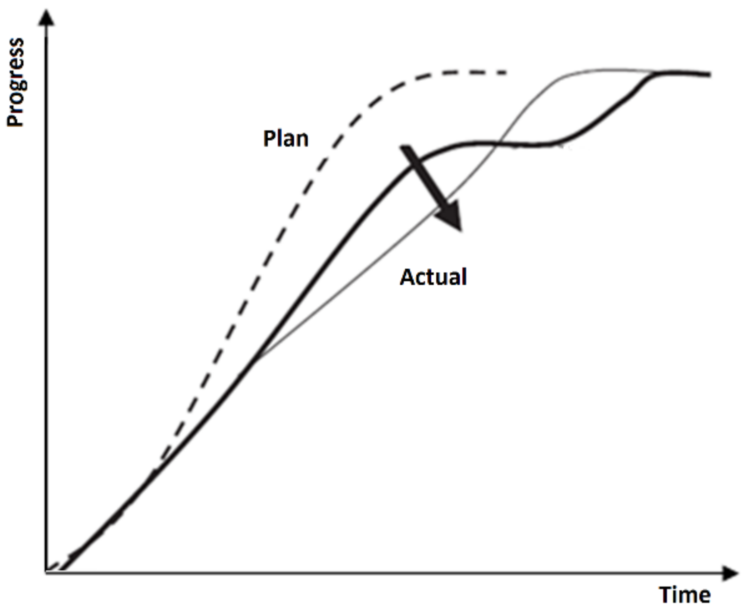


Figure 26: Comparing the planned and actual project progress (Lyneis and Ford, 2007)

The research and applications of system dynamics modelling address diverse aspects related to the project management. One important application of system dynamics modelling is to represent the direct impact of any externally and internally-caused changes on the duration of the project from the original plan. Models are used to identify the contribution of these changes to project performance issues by removing the impact of each change as an input to the model. This allows to find changes with the high impact on the project and addressing their risks in future projects. Works by Abdel-Hamid and Madnick (1989), Abdel-Hamid (1989), and Abdel-Hamid (1996) are among examples related to this area of research.

The way management should respond to these changes is another area of system dynamics modelling application in the project management. Change management involves mitigating the effect of changes on the project performance by for example, schedule extension (Williams, 1999), or compressing the project's schedule (Howick and Eden, 2001). In addition to changes, underestimating the project scope which results in the under-budgeting is among issues deteriorating the project performance. This results in another application of system dynamics modelling in improving project estimating and risk assessment (Lyneis and Ford, 2007). System dynamics modelling has provided key project control lessons related to managing the rework cycle, by improving the quality and reducing errors, recognizing undiscovered rework, and avoiding its consequences (Park and Peña-Mora, 2004), and implementing improved policies. Finally, in case of existence of an infeasible initial plan, minimizing ripple and knock-on effects by easing performance targets through slipping the deadlines, increasing the budget, and reducing the scope, or by adding resources, and using staff more efficiently, are among examples of suggestions provided by system dynamics models (Lyneis and Ford, 2007).

3.5 Research design

This research could be divided into three chronological phases as shown in Table 16. The first phase is the exploratory phase wherein the initial research objective is addressed. This is followed by the development phase, in which a system dynamics model is developed thus satisfying objective 2. The third and final phase is the implementation phase through which the final objective is addressed.

3.5.1 Phase 1: Exploration

The goal of this phase of the research is to perform an extensive analysis of literature related to the research topic to identify the trends and gaps. Typical system dynamics models start with empirical observations, often from one or a few case studies. Using this approach researchers could stay close to the empirical phenomena and built their formulations based on tangible examples. In contrast, in this research, the goal is to build a generic model from theoretical constructs in the literature. To overcome the issue with the absence of direct empirical guidance for modelling, as Rahmandad (2015) stated, it is important to concentrate only on capturing the key mechanism of interest from the literature, which in this research is the effect of LPD on efficiency and effectiveness of development projects. In this regard, adding any theoretical construct which does not directly contribute to this mechanism must be avoided. In addition, extensive sensitivity analysis and calibration is required to increase the confidence in the robustness of the results, including testing results under various alternative assumptions.

3.5.2 Phase 2: Development

During the second phase of research a system dynamics model is developed to support the implementation of LPD. This involved further analysis of literature and the identification of principles, methods and tools related to this modelling approach. Building Formal simulation models, as Wolstenholme (1993) stated, includes two parts; in the qualitative part, the causal relationships between the elements inside the system boundary are demonstrated using *Causal Loop Diagrams*. In quantitative part, these causal loop diagrams are transformed to *Stock-and-Flow Diagrams,* and quantitatively modelled using mathematical equations, and then calibrated. Größler et al. (2008) called these two phases, conceptualization, and simulation. Following this approach, the development phase is divided into two subsections as shown in Table 16. In the first subsection, the conceptual model in the form of a causal loop diagram is built using the theoretical constructs extracted from the literature on LPD. This is a purely qualitative work, which as mentioned before in Section 3.3.1, represents the soft systems approach. Second subsection is purely quantitative, and includes building a stock-and-flow diagram based on the developed causal loop diagram and mathematically formulating the relationships between different types of variables in the model. At this stage, the model is ready for simulation.

3.5.3 Phase 3: Implementation

This phase, first includes the validation of model using the actual data, to make its outputs close to the real-world NPD projects. The model developed then is simulated based on several defined '*What-If*' scenarios and different combinations of parameters values to analyse managerial policies related to the LPD process design, and their effects on the performance metrics.

Table 16: research design

<table>
<tr><td colspan="2">Phase 1: Exploration
Research objective 1: Reviewing lean product development approaches and examining the current state of literature on the subject of LPD
Research objective 2: Extracting process-specific elements and components of LPD from literature</td></tr>
<tr><td colspan="2">Phase 2: Development
Research objective 3: Developing a process model through which the impact of LPD process structure of the performance of development projects can be studied.</td></tr>
<tr><td>2-1: Developing causal-loop diagrams</td><td>2-2: Quantifying the model</td></tr>
<tr><td colspan="2">Phase 3: Implementation
Research objective 4: Validate and test the model using industrial data.</td></tr>
</table>

3.6 Methods employed for Data Collection

In this research, data is used to build, and calibrate the system dynamics model. This section provides an overview of the methods employed for data collection.

3.6.1 Literature review

As mentioned in Section 3.5, the model is built based on theoretical constructs related to LPD, extracted from literature. So, literature review is the first data collection method in this research. The literature includes what is already known and written down relevant to the research project (Robson, 2011). The literature review is one of the key methods for data collection (Mays et al., 2001). The main reasons for conducting the literature review are to develop the knowledge and understanding about the research topic, to identify general patterns of research and key findings through the analysis of publications in the same area, to identify various definitions used by researchers for key concepts related to the topic, to identify appropriate research methodologies, and finally, to find the gap in existing literature (Hart, 1998; Robson, 2011).

As the first step, the scope of literature review was limited by setting the search timeframe from 1990, the year of publishing "The machine that Changed the World" by Womack et al., to 2017. However, if during the reading of a publication a relevant title outside this timeframe was found in its bibliography, that title was also considered for the review. Using keywords is regarded as an effective method to find relevant publications as well as assessing the response to a related topic. The selection of keyword combinations, including *Lean Product Development*, *Lean Design*, *Lean Product Engineering*, *Lean Innovation*, *Toyota Product Development*, was based on Martínez León and Farris (2011). The keyword search was performed in bibliographic databases, such as *Google Scholar*, *Scopus*, *Emerald*, *Science Direct*, and *EBSCO*. The goal was to identify peer-reviewed publications, in addition to relevant books, theses, and conference papers which contained a clear link to LPD. Khan (2012) evaluated strengths and weaknesses of several types of publications, as shown in Table 17. In most cases making decision about the relevance of a publication to the research topic was possible by just reading its title; however, in cases where the information provided by the title was not sufficient for decision, first the abstract and then the full article was read.

Table 17: Strengths and weaknesses of different types of publications (Khan, 2012)

Type of publication	Strengths	Weaknesses
Book	Unconstrained length allows room for elaboration	Peer-review process is often unstructured and may not be robust
Journal article	Academic rigour and originality are the main criteria	Readability may be compromised
Conference article	Succinct communication of original research	Academic rigour varies from one conference to another
Thesis	Comprehensive representation of research	Quality varies depending on the researcher and research institution
Research project report	Flexible structure and representation	Stakeholder influence may affect the research results

3.6.2 Performance data related to NPD projects

Calibration of the model needs quantitative data related to the performance metrics of NPD projects, including project lead time, project cost and quality. Gathering this type of data related to current NPD projects in the manufacturing industry is cumbersome, time consuming and costly. In addition, since the information related to NPD projects is considered as business sensitive assets in the manufacturing industry, companies are not willing to share any project information which includes the monetary values, even when the names and specifications of the projects are fully concealed. This is evident from reviewing publications on LPD, as the number of quantitative studies using empirical data is very limited. To cope with the sensitivity problem, because there is no difference between using data related to current or past projects from the point of view of calibrating the model, it was decided to use the historical project data. Fortunately, the access to this data related to several types of projects from a major car manufacturing company in the UK become possible. In addition to calibrating the model, having the historical data related to the performance of different types of projects made it possible to have a cross-project study on the effect of LPD process structure which is explained further in the Discussion Chapter.

4 Model construction

4.1 Introduction

This chapter presents the construction of the system dynamics model of LPD. The model presented in this chapter addresses the second research objectives: ‘To develop a system dynamics model to investigate the impact of LPD on development projects’ performance’. This chapter is organized as followed:

In Section 4.2, a review of common tools in system dynamics modelling is presented. Elements of a system dynamics model, including stock variables, flow variables, auxiliary variables, causal links, polarities, delays, and nonlinear relationship are reviewed in Section 4.3. Then, in Section 4.4 the qualitative stage of modelling is explained. This Section covers defining the model purpose and boundary, defining some terms used in the model, such as design confidence, uncertainty, iteration, rework, project complexity, and number of concepts, and finally, identification of positive, and negative feedback loops in the model. Section 4.6 is about the quantification of the model introduced in Section 4.4, in which three sectors of the stock-and-flow model, variables, and their mathematical formulations are explained. The chapter ends with a summary Section.

4.2 Common tools for modelling of dynamic systems

Mental models, causal loop diagrams, and stock-and-flow diagrams are three concepts which repeatedly appear in the literature about system dynamics modelling. Before explaining the modelling process, it needs to have a review on these three concepts.

4.2.1 Mental models

Mental models are concepts and relationships which everybody uses to represent the real world. Forrester defined a mental model as” *the mental image of the world around us that we carry in our heads*” (Doyle and Ford, 1998: P6). As defined by Maani and Cavana, "*mental models reflect the beliefs, values, and assumptions that we personally hold, and they underlie our reasons for doing things the way we do*" (Maani and Cavana, 2007: P15). Senge (1992) described them as internal images about how the world works. While humans use their mental models as the first tool in the process of decision making, these models have some shortcomings, especially in capturing complex phenomena. Sterman (2000) stated that mental models of a system are very simplified, compared with the complexity of the systems, mainly due to ignoring feedback processes, time delays between actions and responses, and nonlinearities. Grösser and Schaffernicht (2012) indicated that the dynamic complexity, including the dynamic behaviour of accumulation processes and causal feedback relationships with time delays, are among characteristics of a system which are difficult for humans to mentally comprehend. This fact that mental models are unreliable in anticipating the dynamic behaviour of information-feedback systems leads to using system dynamics modelling to explain dynamically complex systems to bring a practical understanding to the real-world's complex phenomena and to support dynamic decision making. As the result, a considerable portion of researches in the field of system dynamics have been devoted to developing techniques and tools for mapping and representing mental models to improve the quality of dynamic decisions (Doyle and Ford, 1998).

Causal Loop Diagrams and *Stock-And-Flow Diagrams* are two well-known tools for the conceptual representation of mental models. To see the position of these tools, it is helpful to review again the stages of building a system dynamics model, mentioned in the previous Chapter. It is well established that system dynamics modelling is a process (Mashayekhi and Ghili, 2012), and there are several frameworks, designed by researchers, to describe approaches to this process. All frameworks follow a general pathway, including describing the real-world problem, mathematically formulating this description, and simulating it. Table *18* represents some examples of frameworks for the process of system dynamics modelling. Although adapting different terminologies for stages of system dynamics modelling, all classifications could be summarized into the two-phase process of Wolstenholme (1993), which consists of a qualitative stage or *Model Conceptualization*, and a quantitative stage or *Model Exposition*. As almost all researchers agree (e.g. Lane, 2008), main tools for depicting the model in the quantitative stage are causal loop diagrams, and in the quantitative stage are stock-and-flow diagrams. Figure 27 schematically demonstrates the role of mental models, causal loop diagrams and stock-and-flow diagrams in building a system dynamics model.

Table 18: different frameworks proposed for the process of system dynamics modelling

<table>
<tr><th>Richardson and Pugh (1981)</th><th>Wolstenholme (1993)</th><th>Forrester (1994)</th><th>Sterman (2000)</th><th>Maani and Cavana, 2007</th></tr>
<tr><td>Understanding of the system</td><td rowspan="3">Qualitative phase</td><td rowspan="3">Describing the system</td><td rowspan="2">Problem articulation</td><td rowspan="2">Problem structuring</td></tr>
<tr><td>Problem definition</td></tr>
<tr><td>System conceptualization</td><td>Dynamic hypothesis</td><td>Causal loop modelling</td></tr>
<tr><td>Model formulation</td><td rowspan="5">Quantitative phase</td><td>Converting the description to Stock-And-Flow equations</td><td>Formulation</td><td>Dynamic modelling</td></tr>
<tr><td>Simulation</td><td>Model simulation</td><td>Testing</td><td>Scenario planning and modelling</td></tr>
<tr><td rowspan="3">Policy analysis</td><td>Designing alternative policies</td><td rowspan="3">Policy formulation</td><td rowspan="3">Implementation and organizational learning</td></tr>
<tr><td>Educate and debate</td></tr>
<tr><td>Implemantation of policy changes</td></tr>
</table>

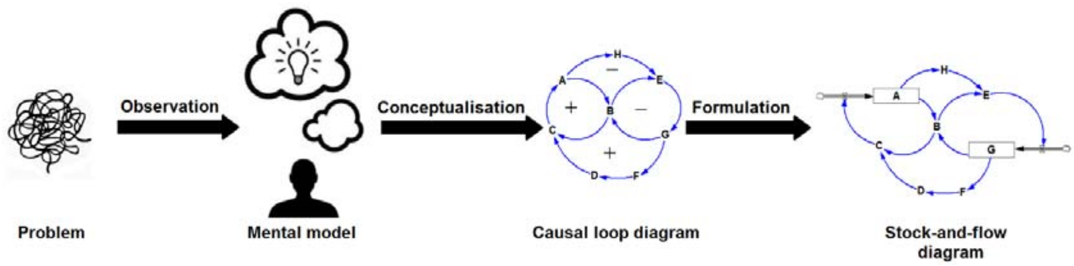


Figure 27: The role of causal loop diagrams and stock-and flow diagrams in building a system dynamics model

4.2.2 Causal Loop diagrams

Causal loop diagrams are important tools of system dynamics modelling for capturing the structure of the system. As Senge stated, causal loop diagrams provide "*a framework for seeing interrelationships rather than things, for seeing patterns of change rather than static snapshots*" (Senge, 1992: P68). They are visually representing the interactions and feedback loops, and mapping the cause-and-effect relationships between different variables within the system. To become prepared for a quantitative formulation, causal loop diagrams are focused on shaping a qualitative discussion about the effects of feedbacks (Lane, 2008). The direction and type of causality among different variables, the direct or indirect (through intermediate variables) impact of each variable on the model outcomes, and the effects of variables on each other could be clearly depicted using causal loop diagrams. The emphasis in a causal loop model is on the representation of feedback loops and delays (Sterman, 2000; Lane, 2008), while there is no differentiation between different types of variables. To have a simpler model, some variables may be excluded from these diagrams, however, they have a high concentration of feedback loop structures (Lane, 2008). More importantly, the dynamic hypothesis is represented by the causal relationships among system elements which expressed in causal loop diagrams. The dynamic hypothesis, as a qualitative representation of the relationships in the system, makes a basis for quantitatively modelling the system using the stock-and-flow diagrams. Developing a causal loop diagram for this research is explained in detail in Section 4.4.

4.2.3 Stock-And-Flow diagrams

Stock-and-flow diagrams represent the structure of a system with more level of details, and contain all the necessary information to determine the dynamic behaviour of a system. Several types of variables are defined in these diagrams, and mathematical equations address the relationships between them. Stock-and-flow diagrams are the quantitative system dynamics model which could be simulated using software packages. More detail about the stock-and-flow diagram and the steps needed to be followed to develop it for this research could be found in Section 4.6.

Causal loop diagrams and stock-and-flows diagrams contain several elements specific for conceptually representing the dynamic models. Following section represented these elements in brief.

4.3 Elements of system dynamic models

Grösser and Schaffernicht (2012) classified elements required for the conceptual representation of a dynamic system, in a hierarchical order, as shown in Figure 28. At the highest level is the feedback loop, as a concept with the highest necessity for the endogenous representation of dynamic systems. Variables and causal links, as the building blocks of feedback loops, are at the second level. In addition to them, the polarities of feedback loops need to be explicitly shown in a conceptual structure. Variables should be differentiated between stocks, flows and auxiliary variables. The polarity of causal links, and types of relationship they represent (linear or non-linear) could be used to make a distinction between them. Causal loops with time delays also need to be differentiated in the representation. As mentioned earlier and emphasised by Lane (2008), the interactions of these elements in a system are beyond the ability of human mind to infer, thus highlighting the importance of using other modelling tools, such as causal loop diagrams and stock-and-flow diagrams, for deducing the consequences of these interactions which often form a counter-intuitive behaviour in the system. Table 19 differentiates between the elements used in each of these two types of diagrams. In the following sections, except the concept of feedback loops which is explained in detail in Section 3.4.1.1, other elements are described briefly.

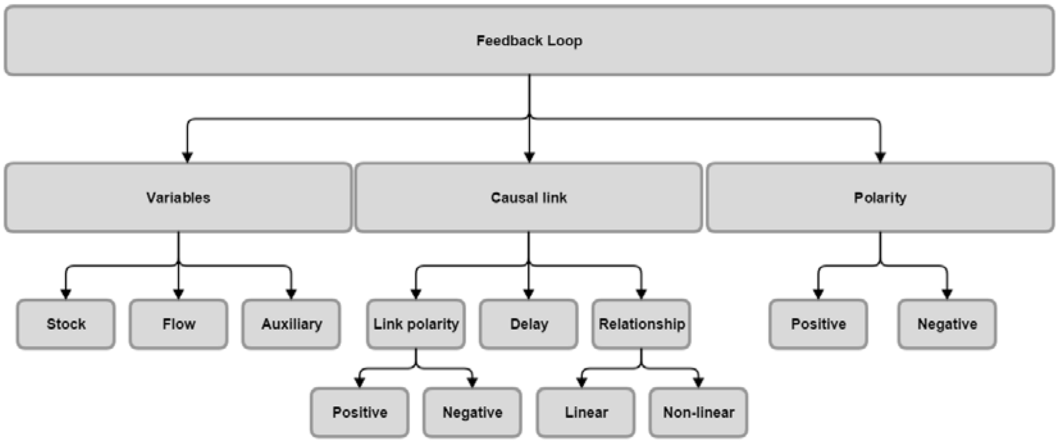


Figure 28: elements required for the representation of dynamic systems (Grösser and Schaffernicht, 2012)

Table 19*: the presence of system dynamics elements in causal loop and stock-and-flow diagrams ((Grösser and Schaffernicht, 2012)*

	Elements	**Causal loop diagram**	**Stock-and-flow diagram**
Feedback loop	Feedback loop	Always present	Rarely present
	Feedback loop polarity	Always present	Rarely present
Variable	Stock /Level/State variable	Not present	Always present
	Flow/Rate variable	Not present	Always present
	Auxiliary/Intermediate variable	Always present	Always present
Causal link	Causal link polarity	Always present	Always present
	Type of relationship (linear or non-linear)	Not present	Not present*
	Delay	Always present	Rarely present
*: non-linearity is embedded in equations			

4.3.1 Stock variables

A *Stock*, or *Level,* or *State Variable* describes the current condition of a dynamic System. At least one stock must be included in each system dynamics model to represent the system state. The only way that state variables could change is through making accumulation into them, or draining out of them (Lane, 2008), using flow variables.

4.3.2 Flow variables

A *Flow variable*, or *Rate* describes the way that the state of a system changes over time. Flows are quantities of accumulations or drainages which integrate up into the stock variables (Lane, 2008). The values of flow variables depend on a combination of endogenous and exogenous variables. In general, at least one stock is required for the formulation of a flow (Grösser and Schaffernicht, 2012).

4.3.3 Auxiliary variables

An *Auxiliary* or *Intermediate Variable* provides clarity in the conceptualization process and formulation of causal relationships. Mathematical descriptions of a system require only stocks and flows, yet, for clarity, it is helpful to define some intermediate or auxiliary variables whose values are a function of stocks, and other endogenous or exogenous inputs (Sterman, 2000). They help to explain the causal structure and could be used to improve the transparency and comprehension in the model. Auxiliary variables are used in the formulation of flows, while flows, in turn, influence stocks and change the state of the system.

Figure 29 shows a representation of stock, flow, and auxiliary variables as a simple stock-and-flow diagram. Stocks are represented by rectangles, inflows and outflows are represented by pipes, and valves are controllers of the flows. Clouds represent stocks outside the boundary of the model which have infinite capacities, and never constrain the flows they support (Sterman, 2000). According to Table *19*, in causal loop diagrams there is no differentiation between different type of variables, and all are represented as auxiliary variables (Figure 30), while in stock-and-flow diagrams (Figure 29) they are graphically differentiated.

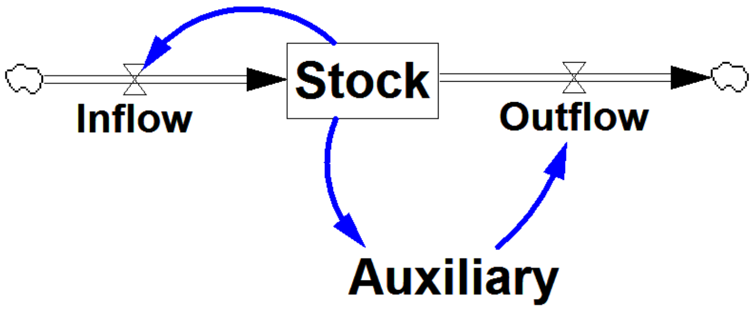


Figure 29: a simple stock-and-flow diagram

4.3.4 Causal links and polarities

Causal links connect variables, while their polarities are shown using arithmetic operators. Causal links could connect a flow to a stock (*Conserved Flow Links*), or connect a stock to auxiliary variables, two auxiliary variables, and an auxiliary variable to a flow variable (*Information Links*). The polarity of a causal link is the direction of the effect which an influencing variable has on an influenced variable (Lane, 2008). Ceteris paribus, if an increase in the variable A increases the variable B above what it would have been, the polarity is positive, whereas if an increase in the variable A decreases the variable B below what it would have been, the polarity is negative (Figure 30). As seen also in Figure 30, a solid arrow represents the causal links. The polarity of feedback loops could be established by multiplying the signs of the link polarities in a loop and finding the net sign.

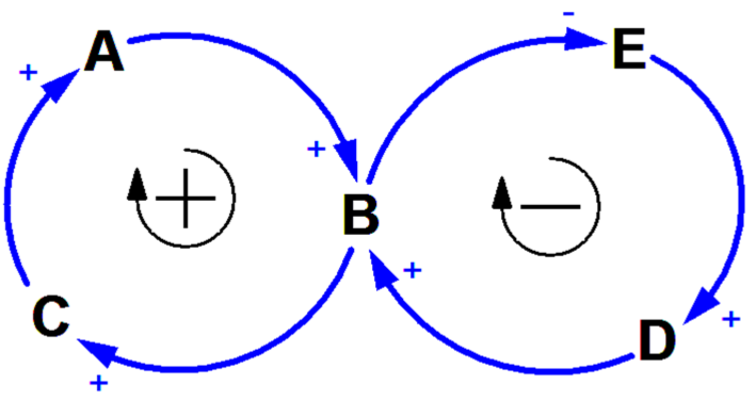


Figure 30: the representation of positive and negative causal links, and feedback loops

4.3.5 Delay

In system dynamics, because the integration processes take place with respect to stock variables, there is a time delay between causes and effects. While in causal loop diagrams, delays are explicitly represented using a double line orthogonal to a causal link (Figure 31), in stock-and-flow diagrams they are represented implicitly by equations.

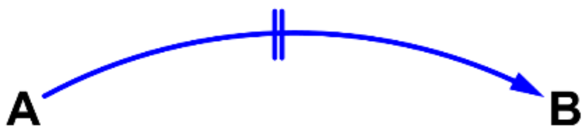


Figure 31: the representation of a delay

4.3.6 Nonlinear relationships

Nonlinearity, as an inherent aspect of the behaviour of a social or physical system, is an indicator of the presence of interaction effects between variables (Grösser and Schaffernicht, 2012). If changes in the output of a system are not proportional to changes in the input, that system exhibits nonlinear relationships. In other words, in a nonlinear relationship the dependent variables cannot be written as a linear combination of independent variables. System dynamics is well known for its capacity to describe and formulate nonlinear relationships. However, in a representation of a dynamic system, either in the form of a causal loop diagram, or a stock-and-flow diagram, nonlinearities are not graphically represented. They are normally embodied in multiplicative and divisional operations in auxiliary variables.

After describing the elements of a system dynamics models in Section 4.3, the following Sections explain the process of building the model, in the form of two distinctive qualitative and quantitative stages.

4.4 Software selection

For the purpose of modelling in this research Vensim DSS software package, version 7 is used (Figure 32). Vensim is industrial-strength simulation software for improving the performance of real systems. Vensim's rich feature set emphasizes model quality, connections to data, flexible distribution, and advanced algorithms. Vensim is used for developing, analysing, and packaging dynamic feedback models, and provides high quality, with dimensional consistency and reality check, connections to data and sophisticated calibration methods, instant output with continuous simulation, flexible model publication, and model analysis, including optimization and Monte Carlo simulation.

For model building, Vensim and other system dynamics languages have a great deal in common. The available functions and default graphical presentations are similar. Vensim is more flexible than most in the appearance of the model diagram, allowing you to easily mix stock and flow and causal loop elements. On the analysis side, Vensim is unique. It contains a set of analysis tools that use the structure of the model to present information to quickly find problems and investigate sources of behavior. It allows users to instantly see the behavior of a variable and the variables that connect to it.

Vensim is also very strong in terms of capacity, performance and functionality. Simulation speed is fast, the optimization capabilities are powerful. The sensitivity analysis is both fast and powerful, there are no practical limits on model size, and it is easy to extend the base capabilities using external functions of the Vensim. The list of common software packages for system dynamics modelling is available in Table 20.

Figure 32: Vensim software package

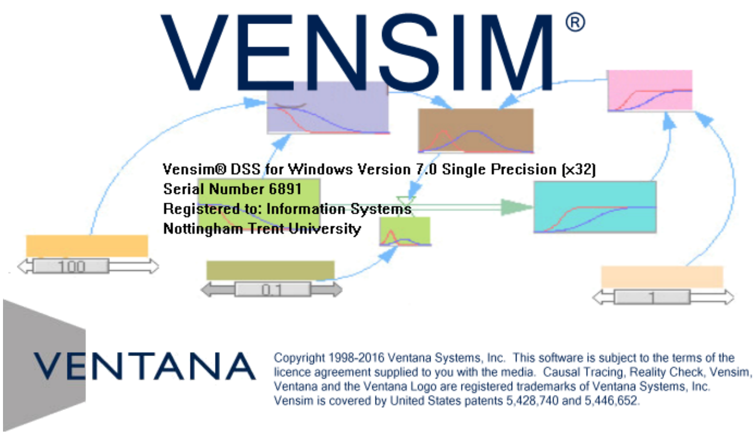


Table 20: System dynamics modelling software packages

Package name	language	More info
AnyLogic	Java	Supports system dynamics, agent based and discrete event modeling, allows making hybrid models.
DYNAMO	Pascal	Historic DYNAMO models are available at the MIT system dynamics website. DYNAMO software for microcomputers may be available via eBay or other resale sites.
Powersim	C++	Supports system dynamics; building graphical diagrams using stocks and flow, including delays and feedback for non-linear models. Supports units, multi-dimensions running scenario simulations and Monte Carlo simulations.
Vensim	C	Continuous simulation with stocks and flows, some discrete delay and discrete event functionality. Flexible array syntax with mapping among dimensions. Extensive support for time series data import and export, with calibration optimization and Markov chain Monte Carlo for estimation. Monte Carlo and other sensitivity simulation methods. Graphical model construction and interfaces. External functions and compiled simulation.

4.5 The qualitative modelling stage

Refer to the stages of system dynamics modelling process outlined in Table *18*, the qualitative stage includes defining the purpose of the model, determining the model boundaries, and building the causal loop diagrams. These parts are covered in following Sections.

4.5.1 Defining the modelling purpose and the model boundary

First part of the qualitative stage includes the identification of the purpose of the model and the determination of the model boundaries (Mashayekhi and Ghili, 2012). Sterman (2000) described the purpose of the model as a criterion for choosing essential features necessary to fulfil that purpose, without it the model ends up including a vast array of variables. The purpose of the model in this research, as is indicated before, is to study the LPD process structure. The system dynamics model which is intended to be developed seeks to provide a linkage between the process-specific features of LPD and the performance measures in NPD projects, by which to explain the reasons of superiority of an LPD process design over traditional approaches.

In literature review Chapter, concurrent engineering and SBCE are recognized as process-specific approaches in LPD. Concurrent engineering involves overlapping phases of an NPD project which has the potential to significantly reduce the duration and cost of projects, while requires a higher communication effort to guarantee an efficient coordination of previously sequential phases. In theory, the time-saving effect of concurrent engineering could theoretically be at least equal to the time length of the overlapping period between two phases. However, in reality, the overlapping could even worsen the situation and results in delays and budget-overrun in projects. The reason, as described previously, is changes in the primarily information received from the upstream phase, based on that the work in the downstream phase starts, which result in rework cycles.

While concurrent engineering is concerned with the simultaneous execution of formerly sequential phases of the project and early integration of functional groups, the focus of SBCE is on how a particular solution for a component or module could be chosen during the conceptual design phase of the project. It is argued that the success of the concurrent engineering approach is highly related to the implementation of SBCE approach, which enables the early consideration of constraints in the project, and avoid time-consuming and costly rework cycles at later points, when decisions are already locked in.

Defining the model boundary includes identifying key variables and necessary elements for generating the behaviour of interest as set by the model purpose (Forrester, 1994a). It is explained as the level of aggregation by Sterman (2000). While having fewer elements in a high-level of aggregation reduces the model complexity, it should be considered that the nature of the problem is not changed because of the aggregation. The model boundary is reflecting the scope and the focus of the model by classifying key variables into *Endogenous*, *Exogenous*, and *Excluded* groups. Endogenous elements are dynamic variables which are directly involved in the feedback loops of the system, and their values change dynamically during a simulation run. Exogenous elements are variables which are possible to be assumed as constants over the time horizon of interest in the model. Their values, although affect the model outcomes, remain unchanged throughout a simulation run. Finally, excluded elements are variables which are assumed to neither have an effect, nor be affected during a simulation run. In other words, they are variables which are chosen not to be modelled explicitly and are, therefore, outside of the model boundary (Sterman, 2000).

The level of aggregation for this research is the internal structure of a single NPD project. Consequently, the interaction among multiple projects is excluded from the model. The reason for choosing this level of aggregation is that understanding the structure and behaviour at this level makes a foundation for the investigation in higher levels, such as portfolio, program, and company level. The model simulates multiple phases within an NPD project which are defined around similar development activities. Market and organization environment are assumed stable during the execution of the project, so their probable changes do not have any effect on the project performance. Details of the model boundary are depicted in Figure 33.

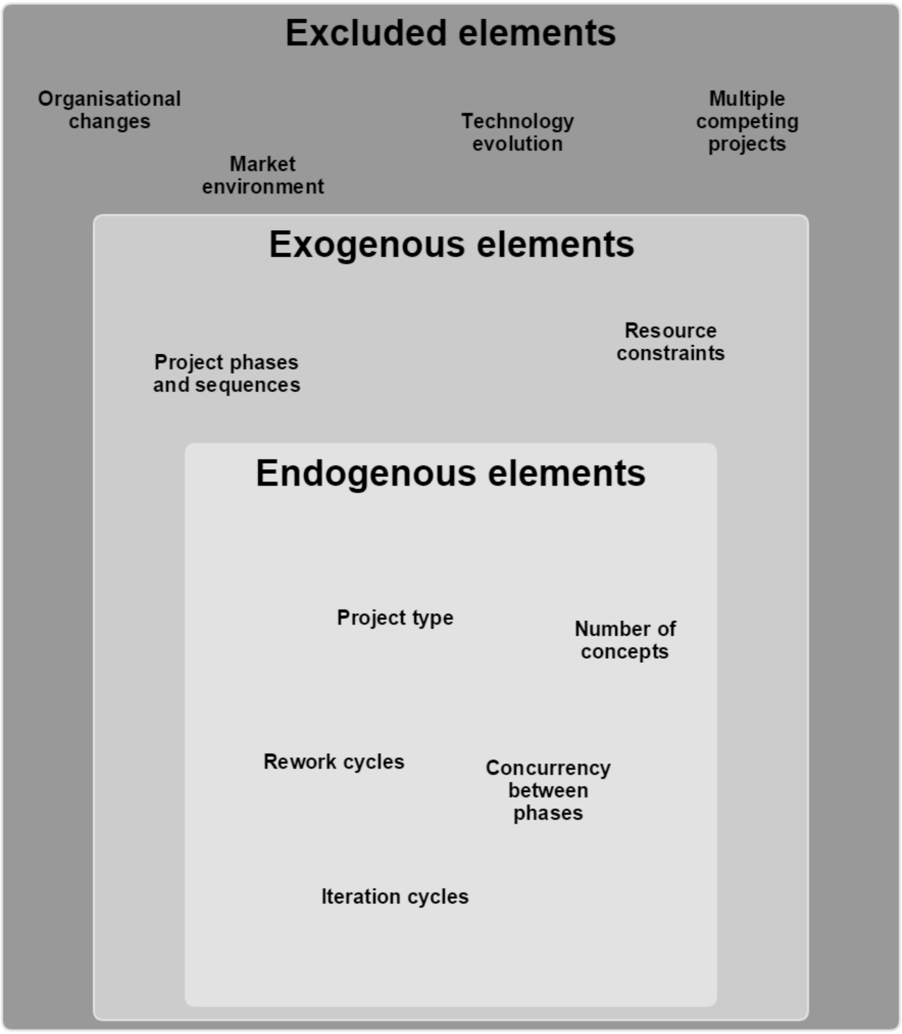


Figure 33: the model boundary

4.5.2 Defining terms used in the model

Before the start of the modelling process some concepts used in the model need to be clarified:

4.5.2.1 Design confidence

Design confidence (or design maturity), according to Ahmadi and Wang (1999), is defined as a measure for the design conformance quality and the perceived engineering reliability of the product under design. Having high design confidence means that the values of design parameters are detailed, accurate, robust, well understood, and physically realistic. In each phase of a project there is always a gap between '*as is condition*' or the current level of the design confidence, and '*as to be condition*' or the target design confidence, and a phase is considered complete once the sufficient level of confidence is achieved. In other words, design is a progressive process (Maier et al., 2014), and engineering efforts continue during each phase of the project until the confidence gap is closed. According to Wynn (2007) and Le et al. (2012) there is a direct relationship between the design confidence and the work quality, where the work quality tends to increase as the design progresses, and the design confidence increases.

In the model, the design confidence is defined separately for each phase of the project, where in the conceptual design phase, it is called "*Concept Confidence*". The rate of increasing the design confidence in each phase determines the phase duration, and consequently, the overall project time. It is also assumed that the level of design confidence never reduces during the project, in other words, the rate of confidence increase during each phase could be reduced to zero, but never becomes negative.

4.5.2.2 Uncertainty

In the model, the confidence gap in each phase of the NPD project is attributed to the uncertainty in design, which is a more common term. Uncertainty in design could be divided into the *Random* (*Aleatory*) and *Epistemic* types (Malak et al., 2009; Kerga et al., 2016). Epistemic uncertainty is due the lack of knowledge and information. Especially in the conceptual design phase, which a considerable proportion of uncertainty is epistemic (Malak et al., 2009), the source of this type of uncertainty is the lack of knowledge about the final design when decisions must be made about the design concepts. On the other side, the random uncertainty causes by naturally random behaviour and variations in processes which is beyond the control of designing teams, such as changes in customers' requirements, errors, and defects (Malak et al., 2009). The sources of random uncertainty are irreducible, however, gaining more knowledge could allow their influence to be mitigated (Suss and Thomson, 2012).

4.5.2.3 Iteration and rework

Two terms of '*Iteration'* and '*Rework*', as discussed in literature review Chapter, are used as representatives for value-adding and wasteful types of iteration cycles in NPD projects. Due to the iterative nature of NPD projects engineering efforts for reducing the epistemic uncertainty and closing the confidence gap in each phase of the project could be modelled through a series of iteration cycles. As Wynn and Eckert (2016) stated, iteration cycles improve the design confidence, because development teams could feel less likely to have overlooked something important related to the design. In the model, rework cycles originate from two different sources; one is the random uncertainty in the processes, which results a fixed probability for rework in the project, and another one is corruption due to the concurrency between phases. Increasing iteration cycles at the outset of the project as the result of implementation of SBCE approach, could reduce the number of rework cycles later during the project.

4.5.2.4 Project complexity

The type of NPD project is accounted as a variable in this model, because according to Le et al. (2012) when the project complexity is high, the confidence in the design is built up slowly. It highlights the importance of classification of projects in the model, based on their level of complexity. Several ways are proposed by researchers to classify NPD projects. Shenhar (2001) proposed a project classification by separating the levels of complexity from the level of uncertainty in technical and engineering-based projects. Based on the level of technological uncertainty at the outset of the project, he defined four classes of projects as followed.

- Low technological uncertainty projects, which involve implementation of well-known and well-established technologies. There is no difficulty and uncertainty in obtaining and execution of this type of technology. Construction projects are good examples of this category.
- Medium technological uncertainty projects, which although are built mainly on existing mature technologies, incorporates some new technological feature to the product. This category includes building a new product in a well-established industry, developing a derivative of the previous design for achieving better performance, or higher reliability and operational life. Projects in the automotive industry could be classified under this type.
- High technological uncertainty projects within them more than 50% of employed technology is new. The basis of these projects is on technical feasibility rather than market needs. Most projects in aerospace, electronics, and computer industries could be put in this category.
- Super high technological uncertainty projects which require developing innovative technologies simultaneous with the execution of the project. Because of unavailability of the technology these types of projects are very risky, while their successful execution makes good opportunities for the company. This category mainly includes defence projects.

Shenhar (2001) also used different levels of design and managerial implications, including the extent of planning, control, and coordination in the projects, for classification of projects:

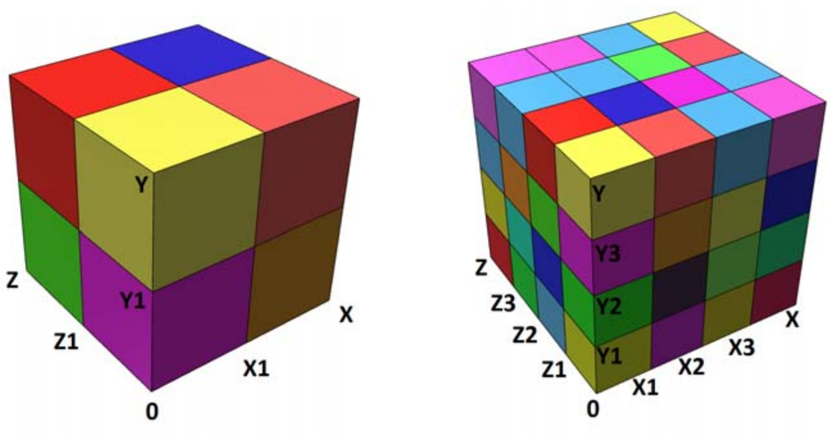
- a subsystem project which deals with a subsystem with a defined function in a larger system;
- a system project which deals with a single product with a combination of interactive elements and subsystems capable of performing a wide range of functions;
- a super-system project which is a conglomeration of systems functioning together to achieve a common goal.

For the model in this research, a combination of two classification approaches proposed by Shenhar (2001) is used. It is similar to a four-class typology proposed by Morgan and Liker (2006), based on their study on TPDS. The scale for numbering the project types is based on a combination of innovation level, technology readiness level, and complexity.

- Level 1: incremental product improvement projects which have the lowest degree of novelty and because of fast technology change cycles and better-informed customers are less feasible for companies.
- Level 2: derivative products which are built on existing product platforms and are very common in industries;
- Level 3: product platform-development projects which have fundamentally new systems and components and utilize improved versions of existing products;
- Level 4: radically new projects which represents breakthroughs in products or technology, and although less common in established companies, still need to be put into account;

4.5.2.5 Number of concepts

The SBCE approach has its main effects in the conceptual design phase, so the focus of the model is more on this phase, where in addition to the project complexity, the rate of confidence increase depends highly on the number of the initial concepts. The responsibility of the conceptual design phase is to develop the principle solution structures for the detail design phase, from a list of objectives and requirements as inputs to this phase (Malak et al., 2009). It is done through the identification of functions which the design must perform, and comparing all possible working principles for each function. The result is having different combinations of working principles for multiple functions, where each combination could be a potential concept for the final design. Partitioning the design space through narrowly or broadly defined concepts could affect the uncertainty in the conceptual design phase. Figure 34 schematically illustrates the narrow and broad partitioning for a design space which consists of three uncertain attributes of a product as ranges between 0 and X, Y, and Z. Having higher number of concepts by narrow partitioning the design space reduces the level of uncertainty and facilitates comparing concepts, eliminating inferior concepts, and finding the most feasible one (Malak et al., 2009). As the result, the number of concepts at the start of the conceptual design phase affects the rate of confidence increase during this phase. However, it should be noticed that working on each concept consumes engineering resources, so having many concepts results in excessive costs. Thus, there should a balance between having high rates of confidence increase by defining more concepts and the cost of the project.



(A) *(B)*

Figure 34: narrowly (A) and broadly (B) defined concepts in the design space for three product attributes X, Y, Z

Figure 35 demonstrates all terms used in the model, their relationship, and their effect on the project performance measures, including the project duration, cost, and quality.

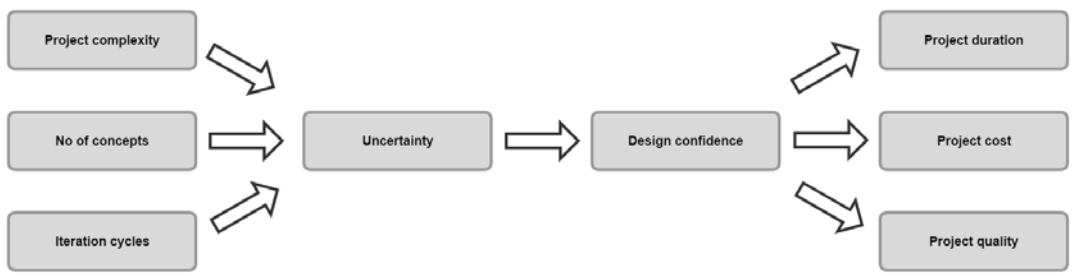


Figure 35: terms used in the model and their relationships

4.5.3 Developing a causal loop diagram

In a dynamic system such as the NPD projects the role of managerial decisions is changing the strength of several causal loops which are active in the system, and create or delete feedback mechanisms to shift their dominance (Ford, 1995). Through these dominance shifts managers could affect the behaviour of the system. However, to achieve this, project managers need to understand the dynamic nature of NPD projects to adjust their policies to the evolution of project behaviours. Poor understanding of the complexity and dynamic features of projects prevents making informed decisions, and consequently, degrades the project performance. Following, different causal loops which are identified for this research in an NPD project are presented.

4.5.3.1 SBCE Loop

Using a causal loop diagram, Figure 36 represents what is happening when the SBCE approach is followed during the conceptual design phase of an NPD project. First the design space for the product is partitioned into several concepts. In development teams, concepts go through different cycles of iterations, in each cycle the level of concept confidence increases, accordingly. Uncertainty is defined as the difference between the level of concept confidence and the target confidence. Reducing the uncertainty to a specified level, development teams start to eliminate less feasible concepts and converge towards the final concept. This is done in the design review sessions after the completion of an iteration cycle. Based on the gap between the current level of confidence in concepts and the target confidence, and consequently the uncertainty level, the decision could be to start another iteration cycle. However, if it is possible, using the current level of confidence less feasible concepts are eliminated, so for the new iteration cycle, less engineering resources are needed. This process continues until a point where the confidence gap closes, and the most promising concept becomes apparent for the development teams.

The described process during the conceptual design phase forms a negative (balancing) feedback loop, which is called *SBCE Loop* in Figure 36. The behaviour of the SBCE loop, as shown in Figure 37, is goal seeking (Sterman, 2000), in which the concept confidence increases to reach to the predetermined level of target confidence (the goal), and the uncertainly decreases to zero which indicate the readiness of the final concept to be sent for the detail design. The duration of the conceptual design phase is under the influence of the level of target confidence. Having a high target confidence increases the duration and the cost of the conceptual design phase, while has the positive effect on the quality of the final product.

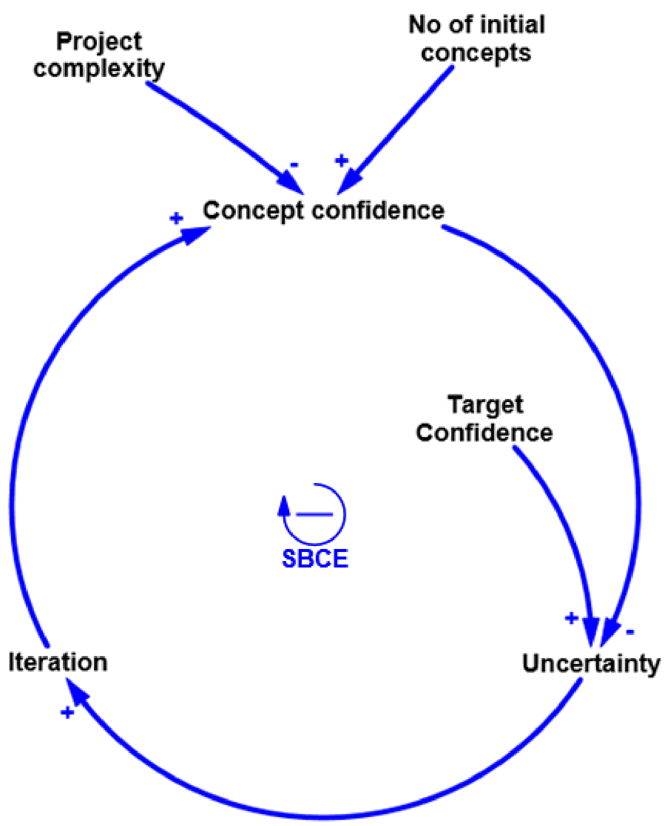


Figure 36: SBCE loop

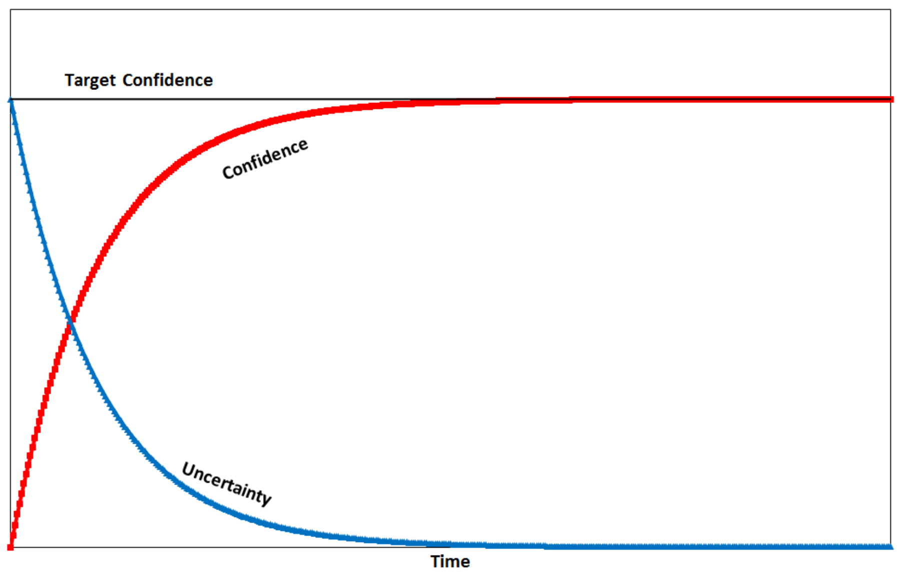


Figure 37: the goal seeking behaviour of the SBCE loop

As shown in Figure 36, the number of iteration cycles and the number of initial concepts have a positive causal link with the concept confidence, while the project complexity has a negative causal link. The project complexity and the number of initial concepts, are defined as exogenous variables in the model, while the number of iteration cycles is an endogenous variable and its value is changing throughout the process. Having a high level of project complexity reduces the confidence increase rate, while defining more concepts increases the confidence increase rate. As shown in Figure 38, the confidence level increases slower for a project with the higher level of complexity.

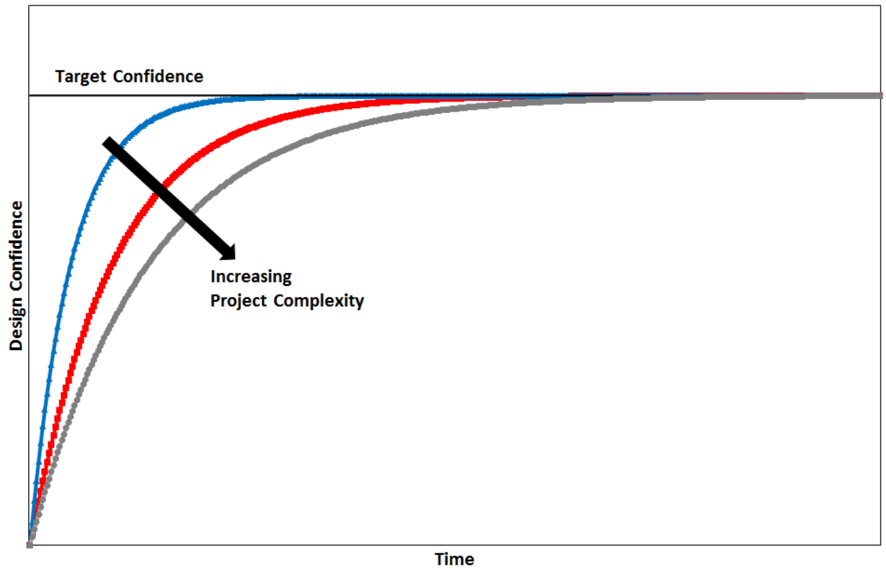


Figure 38: the effect of the project complexity on the concept confidence

4.5.3.2 Schedule Pressure Loop

The first perceived effect of increasing the level of target confidence is the increase in the duration and cost of the conceptual design phase, whereas its positive effect on the quality of the final product which results in the higher probability of success in the market is not immediate. The increasing level of the schedule pressure as the deadline of the conceptual design phase is approached, in addition to the fear of the delay in the market introduction of the product, which depends on the product type could result in a profit loss, forces decision makers to intervene to reduce the schedule pressure. The most common firefighting action by managers, as articulated in literature, is increasing the capacity by engaging more engineering resources in the project (Lyneis and Ford, 2007). Although this type of reaction has its own shortcoming, which has been the subject of many researches, the focus of this research is on shifting from a SBCE approach to a PBCE approach to compensate for the delays in the conceptual design phase. It is done by activating the second feedback loop, the *Schedule Pressure Loop*, as shown in Figure 39. Based on this feedback loop, increasing the number of iteration cycles, and consequently, the schedule pressure in the conceptual design phase, forces managers to reduce the level of target confidence, to reduce the probability of the project delay. As the result, the confidence gap is closed with the less number of iteration cycles, and converging towards the final concept is speeded up. The schedule pressure loop is again a negative feedback loop and shows a goal seeking behaviour to prevent the project delay by closing the gap between the project deadline and the project duration. In addition, as this approach reduces the resources allocated to the conceptual design phase, the cost of the project is also reduced.

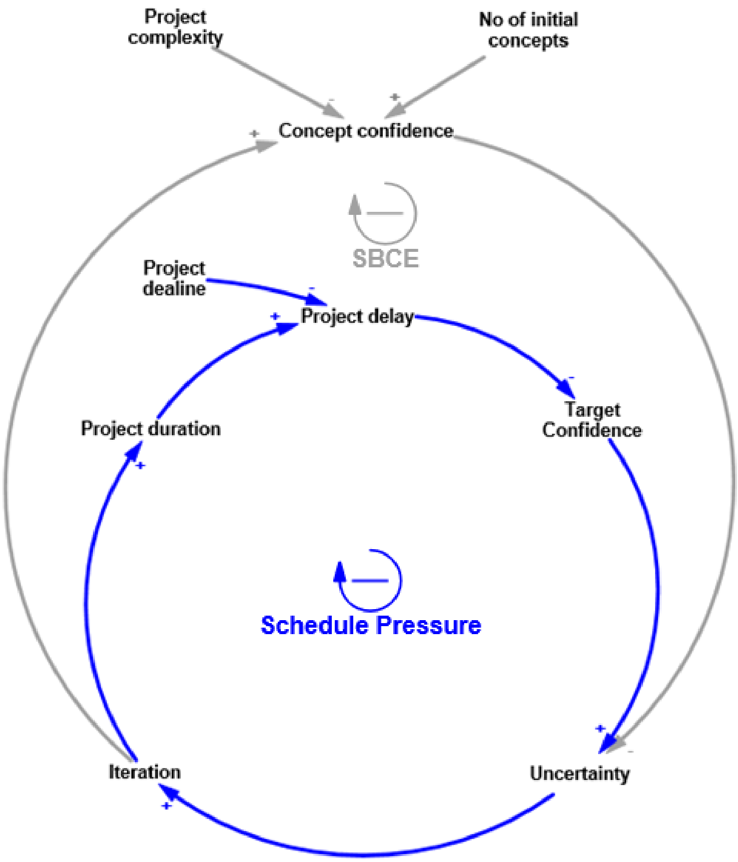


Figure 39: the schedule pressure loop

4.5.3.3 Rework Loop

As mentioned in the previous section, the perceived result of the schedule pressure loop is reducing the project duration; however, its long-term effect is often neglected. This effect is shown as the *Rework Loop* in Figure 40. Reducing the level of the target confidence in the conceptual design phase leads to the higher probability of errors and design changes during the later phases of the project, because of the epistemic uncertainty remains from the conceptual design phase. This increases the probability of rework cycles later in the project, and as each rework cycle takes time and consumes resources, could remove the perceived time and cost-saving effects of the schedule pressure loop.

The rework loop is a positive (reinforcing) feedback loop which shows an exponential growth behaviour (Sterman, 2000). Depends on the level of the target confidence in the conceptual design phase, the rework loop could exponentially increase the project duration and even halt the overall progress of the project. Repenning et al. (2001), and Taylor and Ford (2008; 2006) used the concept of *Tipping Point* to explain this effect which resulted in the failure in many construction projects. According to Taylor and Ford (2006) tipping points are feedback structures in NPD projects which could result rework overwhelms the progress of the project. However, due to the delay in the effect of reducing the target confidence on the number of rework cycles, and its short-term positive effects, it is common for managers to activate the rework loop and put the project into the vicious cycle of increasing late changes and rework.

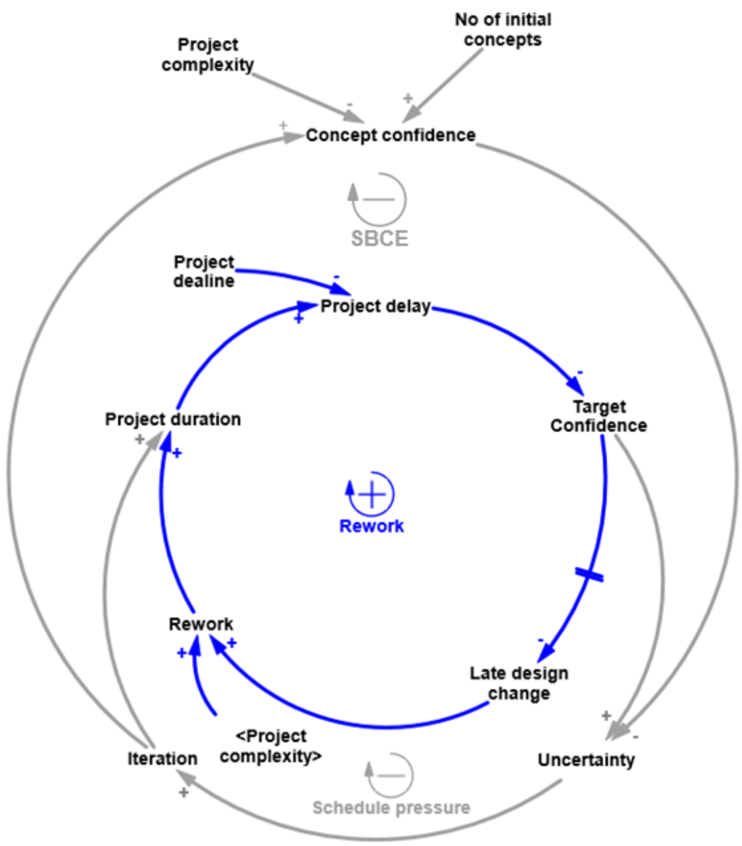


Figure 40: the rework loop

4.5.3.4 Concurrency Loop

Concurrent engineering, as mentioned before, is an approach for reducing the project duration by overlapping sequential phases of the NPD project. This could be explained, as shown in Figure 41, using a negative feedback loop, which is called the *Concurrency Loop*. In this simple loop, increase in the perception of the delay in the project results in starting the downstream phase before the upstream phase reaches its target confidence level. The perceived effect of this policy, similar to previously identified negative feedback loops, would be a goal seeking behaviour which closes the gap between the project deadline and the project duration, thus eliminating the delay in the project.

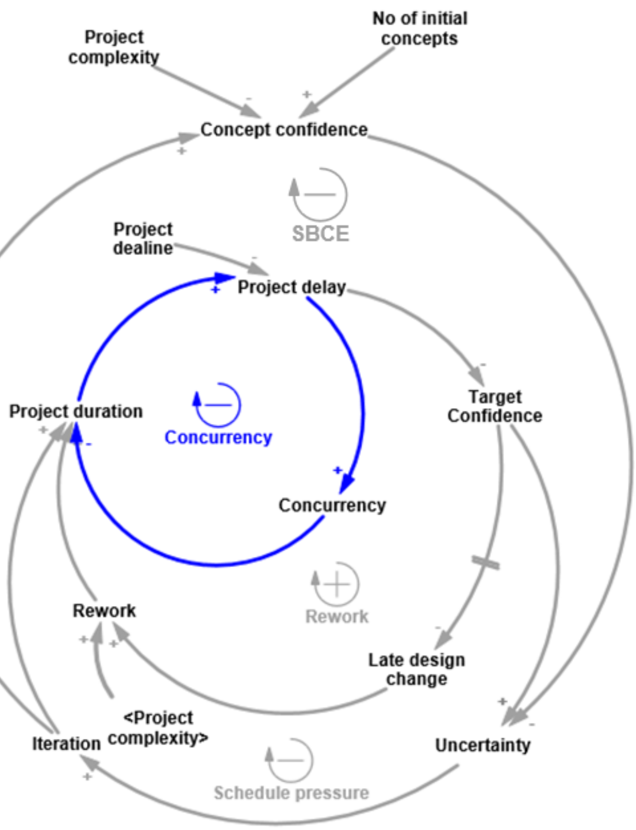


Figure 41: the concurrency loop

4.5.3.5 Corruption Loop

The concurrency between phases of the project has also a long-term effect which makes a positive feedback loop. It is represented in Figure 42 as the *Corruption Loop*. As in the concurrency loop, the downstream phase starts before the confidence gap in the upstream phase is closed, the work simultaneously progresses in both phases. However, if this confidence gap in the upstream phase is wide, the information could highly change, while the work in the downstream phase have already started using the preliminary information. This results in the second type of rework in the project, due to the corruption of the work correctly completed in the downstream phase. The epistemic uncertainty remains from the conceptual design phase due to abandoning the SBCE approach is the main cause of this confidence gap. This shows the combined effect of the SBCE loop and the concurrency loop on the probability of activation of the corruption loop.

As shown in Figure 42, the Corruption loop, similar to the rework cycle, is also a positive feedback loop which directly affects the project duration and could result in delays due to the increase in the number of rework cycles. In addition, as mentioned before, the project cost also increases due to the resources allocated to reworks. The corruption loop has also a delayed link between increasing the concurrency and increasing the rework cycles, which makes difficult for managers to anticipate its effects.

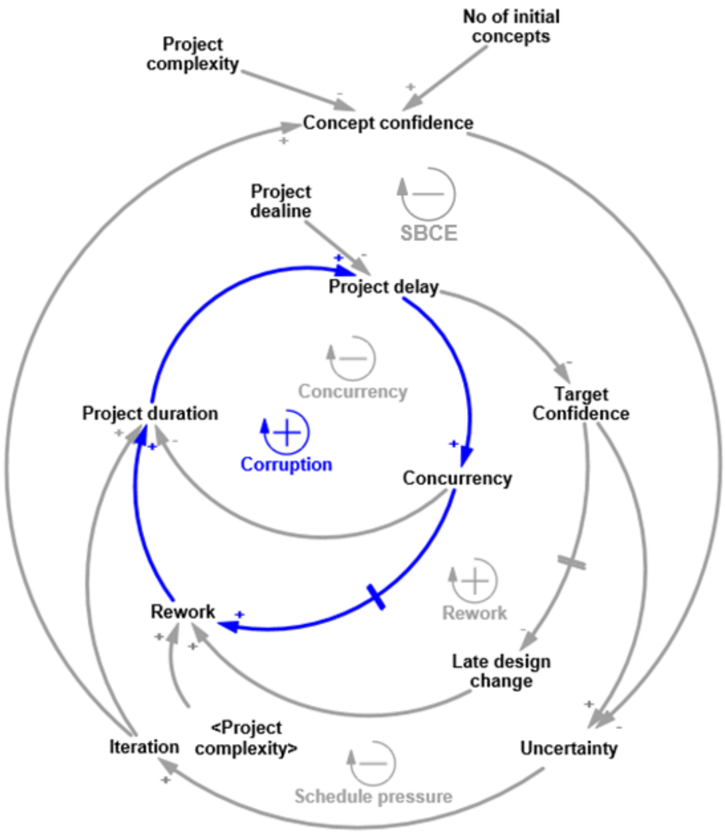


Figure 42: the corruption loop

In addition to described feedback loops and delayed causal links in the developed causal loop diagram, the relationship between the target confidence and the late design changes in the rework loop (Figure 40), and between the concurrency and the rework in the corruption loop (Figure 42) are nonlinear. These nonlinearities are added to the dynamic characteristics of the system and make it difficult for managers to capture the complex behaviour of NPD projects. However, as mentioned at Section4.3.6 , representing nonlinearities is not possible in causal loop diagrams. The only way to capture these nonlinear links is through mathematical equations developed for the stock-and-flow diagram, which is described in detail in Section 4.6.

In contrast to mentioned nonlinear links in the model, the relationship between variables in negative feedback loops are linear. For instance, in the concurrency loop (Figure 41), the overlap between sequential phases of the project could theoretically reduce the project duration to the extent equal to the time length of the overlap period. In addition, in the schedule pressure loop (Figure 39) the project duration could be reduced to the extent equal to the time length of the iteration cycles which are eliminated during the conceptual design phase. This is the reason that managers are more intended to activate these loops through their decisions, for controlling the performance of the NPD projects, while unintentionally neglecting their nonlinear and time-delayed consequences.

4.5.3.6 Summary

Figure 43 shows a summary of the qualitative model built for this research using causal loop diagrams. The schedule pressure loop is representing the traditional PBCE approach in managing the NPD projects. While the schedule pressure loop has a perceived negative link with the project duration (shown by a grey line in Figure 43), its positive links with the rework and corruption loops results in increasing the project duration. The SBCE loop, on the other hand, is representing the SBCE approach in managing NPD projects, and is appreciated in LPD. It has a completely different effect, shown by the negative links between it and the rework and corruption loops. To activate the corruption loop, in addition to the schedule pressure loop, it is needed to have the concurrent engineering approach in the project, represented by the concurrency loop in Figure. While the concurrency loop has a negative link with the project duration, its combined effect with the schedule pressure loop activates the corruption loop, and increases the project duration. The same argument is in place for the effect of these feedback loops on the project cost. Corruption and rework loops, which both are positive feedback loops and tend to increase the project duration, have time delays and nonlinear links, making it more difficult for managers to fully capture the behaviour of the system which is dominated by these feedback loops.

The next Section deals with developing a stock-and-flow diagram based on the described feedback loops, and formulating it to make a simulation model.

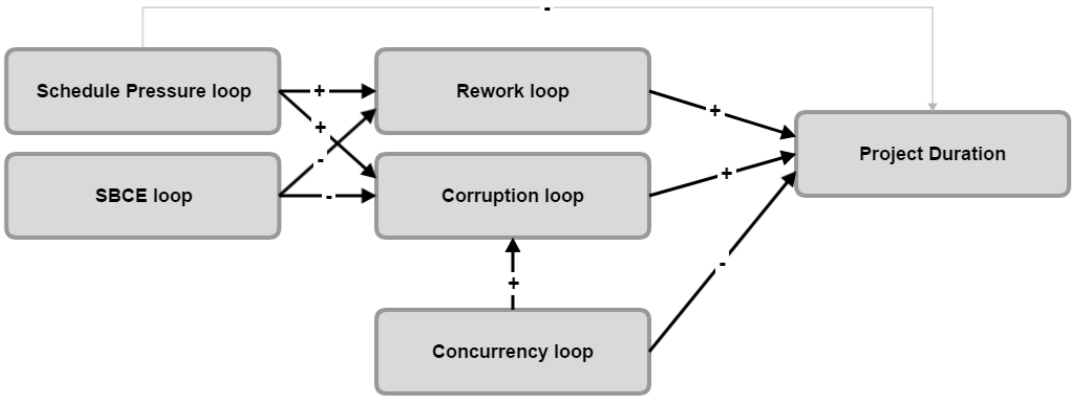


Figure 43: the summay of the causal loop diagram

4.6 The quantitative modelling stage

4.6.1 Introduction

The causal loop diagram described in previous sections is a qualitative model for explaining the structure of the NPD projects and the causal links which affect the performance of a project. However, this diagram could not capture the quantitative features of the system. Due to this, in following sections the formulation of a stock-and-flow diagram is explained.

4.6.2 Information-based view of NPD

The output of NPD projects is not physical objects, but rather information, as the goal of NPD is to create a recipe for producing a product (Reinertsen, 1999). The information that flows from task to task, and makes the fundamental dependency between tasks in an NPD project could be in the form of design information which can be specified directly, e.g., materials and geometry, performance information which is a consequence of design information, e.g., fatigue life and weight, or requirements which may constrain design or performance information, such as the requirements that the geometry of a component has to fulfil to meet the overall performance of the whole product.

As the result, for building the model, according to Bhuiyan et al. (2004) and Lin, et al. (2008), the information-based view of NPD (Clark and Fujimoto, 1991) is followed. According to this view, in NPD, each activity requires and produces information, and it is the information which flows throughout the processes and between activities (Browning et al., 2006). Looking at the process of NPD as an information system and focusing on the ways that information is created, distributed, and used, and identifying the problem from the viewpoint of information processing helps clarifying the role of NPD in the context of competitive environment. By adopting this view, NPD is defined as the process of transformation of data about the market opportunities and technical possibilities into sets of information which are created, combined, and transferred between different media and in the final stage articulated as detailed product and process designs which will be utilized in the production processes. As a result, NPD performance could be translated as the ability of a product design to create a satisfactory product experience. In addition, each development activity is defined as an information-processing unit (Lin et al., 2008), or task, which receives information from its upstream activity, transforms it to the new information, and then transmits it to the downstream activity.

4.6.3 An overview of the model

The model has a generic process structure adopted from Ulrich and Eppinger (2016) as shown in Figure 44. It includes three major phases, conceptual design, detail design, and tooling. Each phase has three main types of activity, namely *Initial Completion*, *Test,* and *Rework*, performing them consumes a part of time and resources allocated to the project. The NPD phases are interdependent, so while performing in parallel (based on the concurrent engineering approach) there are feedback and feed-forward information exchanges between.

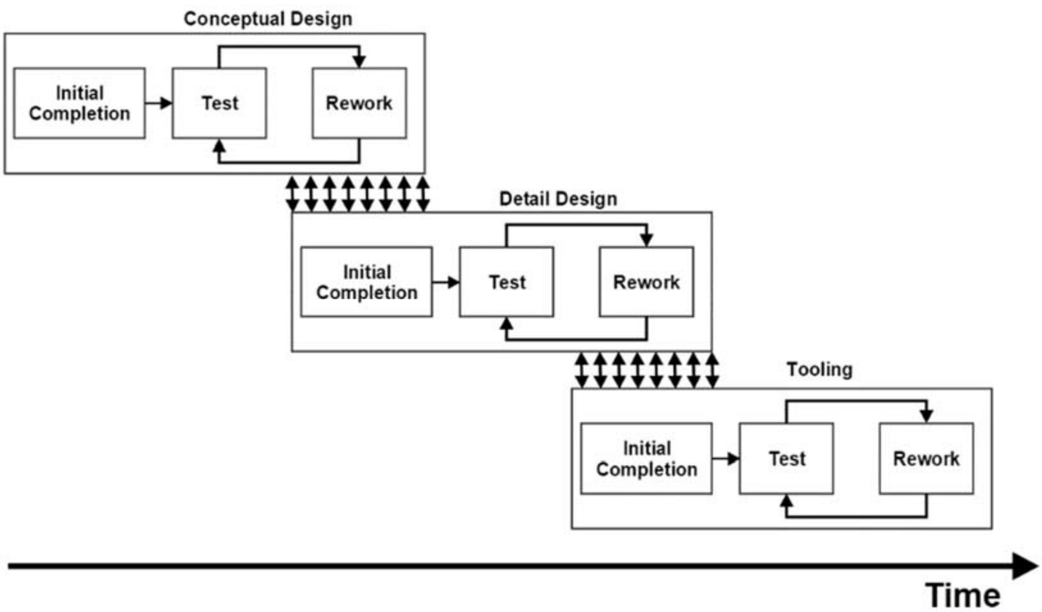


Figure 44: generic process structure of NPD

The model consists of four sectors: the workflow sector, the resource management sector, the schedule pressure sector, and a performance measurement sector.

4.6.3.1 workflow sector

The workflow sector of the model has a general part which is identical for all three phases of the project and consists of the rework cycle and iteration cycle and their related stocks and flows. In addition to this general part, there are some additional parts to capture the effect of implementing the SBCE approach in the conceptual design phase, and the concurrent engineering approach in other phases. These parts are explained separately in following Sections.

4.6.3.1.1 The general part of the workflow sector

Each NPD project could be conceptualized as a collection of tasks, disaggregated into detailed phases, which flow through the processes (Lyneis and Ford, 2007). Tasks are atomic units of development works, assumed to be uniform in size and be interchangeable. From the quality point of view, tasks are assumed small enough to be either entirely defective, or not. In other words, it is assumed that multiple defects could not occur in one task.

As shown in Figure 45, in the simple stock-and-flow diagram of a single phase of the NPD project, there are three types of development activities, represented as flows, while each task could be in one of four different states, represented by stocks. The scope of each phase defines the number of tasks which are needed to be completed during that phase. All tasks first start in the stock of '*Tasks not completed'*. Performing the completion activity, through the flow of '*Completion Rate',* change the state of tasks from '*Not Completed*' to '*Pending Test*', and completed tasks accumulate in the stock of 'Tasks *Pending Test'.* During the testing activity, if no defect is detected in tasks, they pass through the flow of '*Approve Rate*' and reside in the stock of '*Tasks Pending Decision*'. If the testing activity detects any defect in tasks, they are send to the stock of '*Tasks Pending Rework*' through the flow of '*Rework Detection Rate*'. Here the rework, as the third type of activities in the model, send tasks again to the stock of 'Tasks *Pending Test'*, though to the flow of '*Rework Rate*'. The rework, itself, could also be flawed and needs to be retested.

This makes a loop, known as '*Rework Cycle*', as introduced by Cooper (in Lyneis and Ford, 2007) and further developed by Ford and Sterman (1998; 2003) and Rahmandad and Hu (2010). The rework cycle is the central feature of the system dynamics models of NPD projects (Schmidt et al., 2015; Lyneis and Ford, 2007), and is the quantified representation of the rework loop explained in Section 4.5.3.3. Understanding the effect of the rework cycle is the core for understanding the causes of time and cost overrun and quality issues in development projects (Lyneis and Ford, 2007). Depends of the quality of completion and rework activities, a task circulates in the rework cycle between the stocks of rework and testing until the test shows that the defect is removed, and then pass the task through the flow of '*Approve Rate*' to the stock of '*Tasks Pending Decision*'. The mathematical relationship between stocks and flows in the system dynamics modelling is always in the form of differential equations. A stock could be formulated as the integration of inflows to the stock and outflows from the stock, as shown in Equations 1 to 3 for stocks in Figure 45, except the stock of '*Tasks Pending Decision*' which is described later. The initial value of the stock of '*Tasks Not Completed*' is equal to the scope of phase and is added to its equation, while for other stocks the initial value is equal to zero.

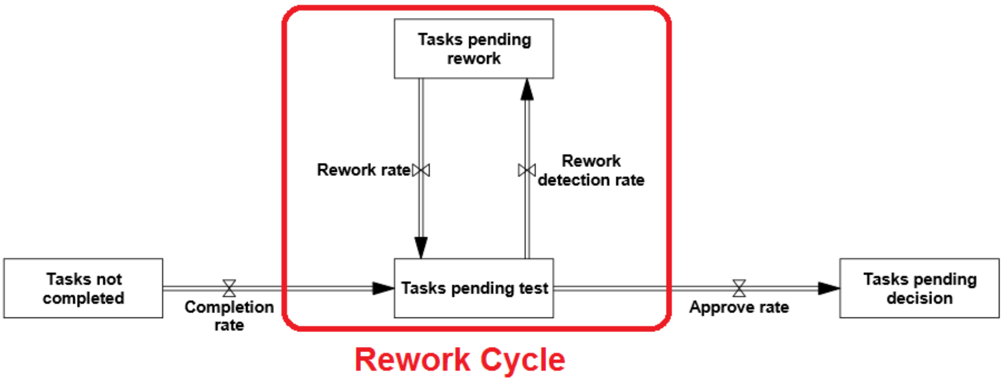


Figure 45: Simple stock-and-flow diagram for a single phase of the NPD project

$$Tasks\ not\ completed = Phase\ Scope + \int_0^t (Corruption\ rate + Iteration\ rate - Completion\ rate)dt \quad 1$$

$$Tasks\ pending\ test = \int_0^t (Completion\ rate + Rework\ rate - Approve\ rate - Rework\ detection\ rate)dt \quad 2$$

$$Tasks\ pending\ rework = \int_0^t (Rework\ detection\ rate - Rework\ rate)dt \quad 3$$

Flows are controlled by three types of constraints; the number of tasks ready for performing the activity, the minimum time that it takes to perform that activity which is the time required to complete a task if all prerequisites including information and resources are available, and the project capacity allocated to the activity (Ford and Sterman, 1998). These constraints are shown in Figure 46 as variables connected to the flows using arrows. If the capacity allocated to an activity is less than what is required, defined by the ratio of tasks ready for that activity and the duration of activity, then it is the allocated capacity which controls the rate of performing that activity. This is formulated using a MIN function as shown in Equations 4 to 7. The variable called '*Rework probability*' which represents the amount of random uncertainty in the process determines the portion of tasks which need to be reworked after the testing activity. The effects of '*Rework probability*' on the flows of '*Rework Detection Rate*' and '*Approve Rate*' are formulated in Equations 6 and 7.

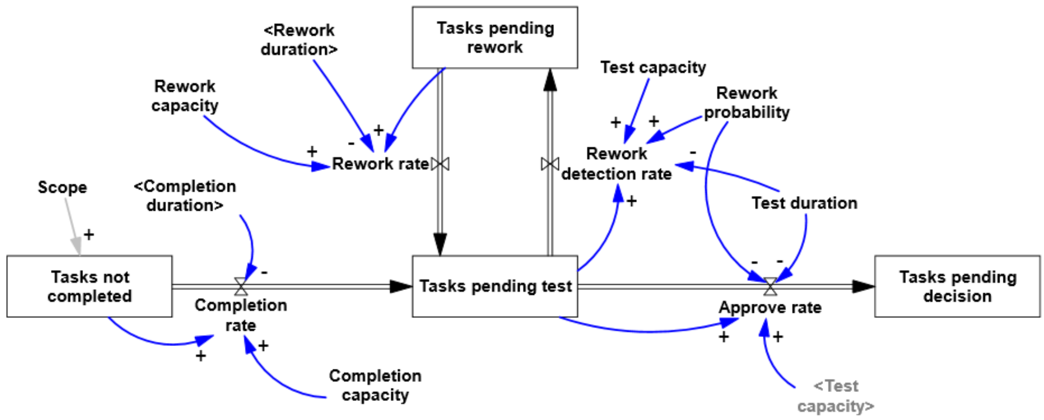


Figure 46: variables controlling the rates of performing activities

$$Completion\ rate = MIN(Tasks\ not\ completed/Completion\ duration, Completion\ capacity) \quad 4$$

$$Rework\ rate = MIN(Tasks\ pending\ rework/Rework\ duration, Rework\ capacity) \quad 5$$

$$Rework\ detection\ rate = MIN\left(Tasks\ pending\ test * \frac{(Rework\ probability)}{(Test\ duration)}, Test\ capacity\right) \quad 6$$

$$Approve\ rate = MIN\left(Tasks\ pending\ test * \frac{1 - Rework\ probability}{(Test\ duration)}, Test\ capacity\right) \quad 7$$

As mentioned earlier, this model makes a distinction between the rework and iteration cycles. While rework is unplanned, and mainly due to the random uncertainty in the projects which shows itself in the form of defects, the source of iteration cycles is the epistemic uncertainly and lack of knowledge and information about the design. Figure 47 represents the iteration cycle. Tasks, as shown previously in Figure 45, after passing the rework cycle, reside in the stock of '*Tasks Pending Decision*'. When the number of tasks in this stock becomes equal to the scope of the phase, one iteration cycle is completed. Completion of each single iteration cycle is followed by a design review session to make the decision about the rest of the project. This decision is based on the level of design confidence gained after completing the whole iteration cycle, and its difference with the level of target design confidence, which is corresponding to the level of epistemic uncertainty. If the gap still exists, tasks are sent back again through the flow of '*Iteration Rate*' to the stock of '*Tasks Not Completed*' to start another cycle of iteration, otherwise tasks are released to the downstream phase through the flow of '*Release Rate*'. This process is repeated until the confidence gap becomes closed, and the epistemic uncertainty becomes zero. Based on new defined flows in the model, the stock of '*Tasks Pending Decision*' could be formulated as shown in Equation 8. Flows of '*Iteration Rate*' and '*Release Rate*' are formulated using the conditional equations based on the value of the uncertainty at the end of each iteration cycle as shown in Equations 9 and 10. These two flows are not defined as activities in the model, but they just show the transition in the state of the tasks in the stock of '*Tasks Pending Decision*'. Because of that, these flows do not occupy the project capacity, and their duration is equal to the 'Time Step' defined for the simulation of the model.

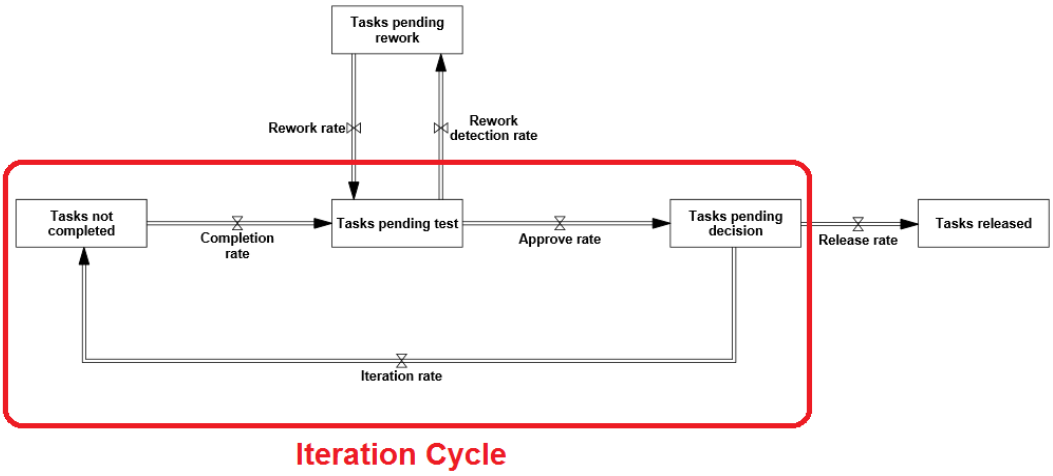


Figure 47: iteration cycle

$$Tasks\ pending\ decision = \int_0^t (Approve\ rate - Iteration\ rate - Release\ rate)dt \qquad 8$$

$$Release\ rate = \begin{cases} \frac{Tasks\ pending\ decision}{Time\ step}, & Uncertainty \leq 0 \\ 0, & Uncertainty > 0 \end{cases} \qquad 9$$

$$Iteration\ rate = \begin{cases} \frac{Tasks\ pending\ decision}{Time\ step}, & Uncertainty > 0 \\ 0, & Uncertainty \leq 0 \end{cases} \qquad 10$$

The epistemic uncertainty in each phase, as mentioned in Equation 11, is equal to the difference between the level of design confidence after each cycle of iteration and the level of target confidence. As shown in Figure 48, the design confidence is modelled as a first-order stock-and-flow diagram, which its values could be changed using the flow of '*Confidence Increase Rate*' (Equation 12). The formulation of 'confidence increase rate' in the conceptual design phase is different compared to other phases of the project.

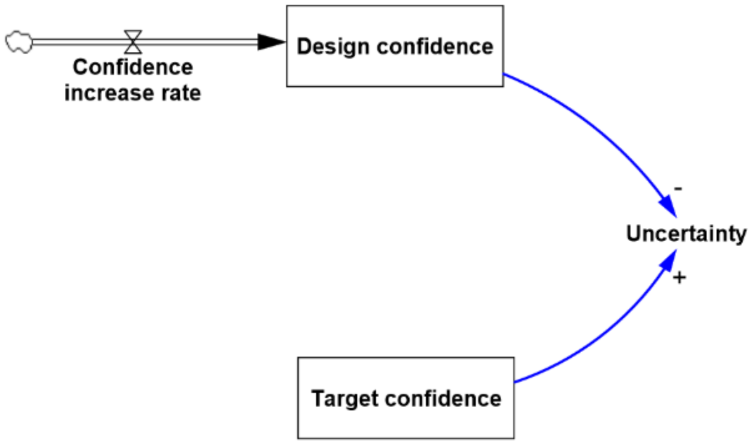


Figure 48: Uncertainty model

$$Uncertainty = Target\ confidence - Design\ confidence \quad 11$$

$$Design\ (concept) confidence = \int_0^t (Confidence\ increase\ rate) dt \quad 12$$

As discussed in Section 4.5.2, the number of concepts, the project complexity and the number of iteration cycles affect the confidence increase rate in the conceptual design phase. As mentioned before, the number of concepts at the start of the conceptual design phase affects the level of uncertainty, and consequently, the rate of confidence increase. This relationship is shown using a power function in Equation 13, where '*PC'* is defined as the strength of the effect of the number of concepts on the rate of confidence increase, and is always positive and less than one. Based on this formulation, '*confidence increase rate*' increases with narrow partitioning of the design confidence, but the graph slope decreases with increasing the number of concepts (Figure 49). This is in accordance with Malak et al. (2009) who argued that the level of uncertainty in the conceptual design phase could not be decreased infinitely by increasing the level of details in concepts. in addition, the cost of the conceptual design phase increases with increasing the number of concepts, which demands an optimum point for the number of concepts.

$$Confidence\ increase\ rate \propto (No\ of\ concepts)^{PC} \quad 13$$

$0 < PC < 1$

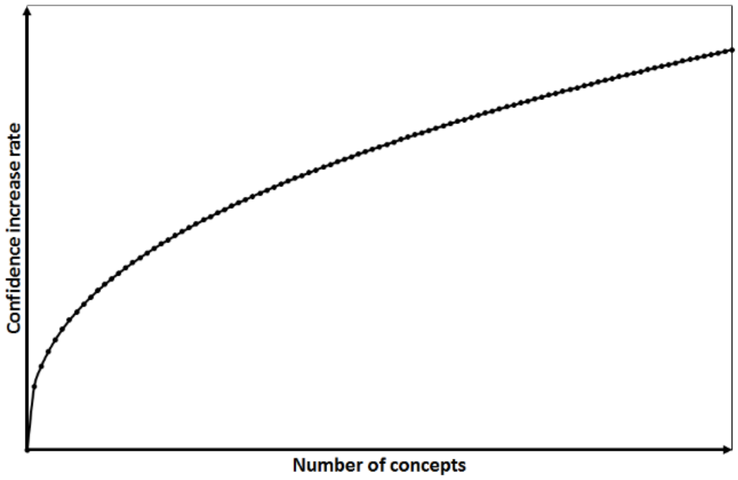


Figure 49: the relationship between confidence increase rate and the number of concepts

The relationships between the number of iteration and the project complexity with the rate of confidence increase are also formulated using power equations. In Equation 14, '*PI*', as the strength of the effect of the number of iterations on the rate of confidence increase, is always greater than one, which shows an always increasing effect of iteration cycles on the confidence increase rate. In contrast, in Equation 15, '*PS*', as the strength of the effect of the project complexity on the rate of confidence increase, is always negative, which implies 'confidence increase rate' reduces exponentially for projects with higher complexity. Figure 50 and Figure 51 are graphical representations of these two relationships.

$$Confidence\ increase\ rate \propto (No\ of\ iterations)^{PI} \qquad 14$$

$$PI > 1$$

$$Confidence\ increase\ rate \propto (Project\ complexity)^{PS} \qquad 15$$

$$PS < 0$$

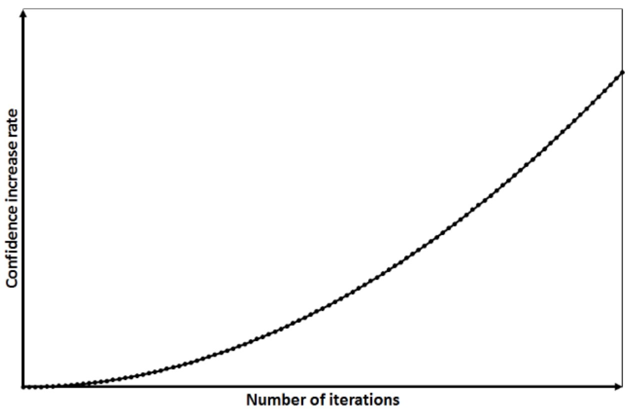


Figure 50: the relationship between confidence increase rate and the number of iterations

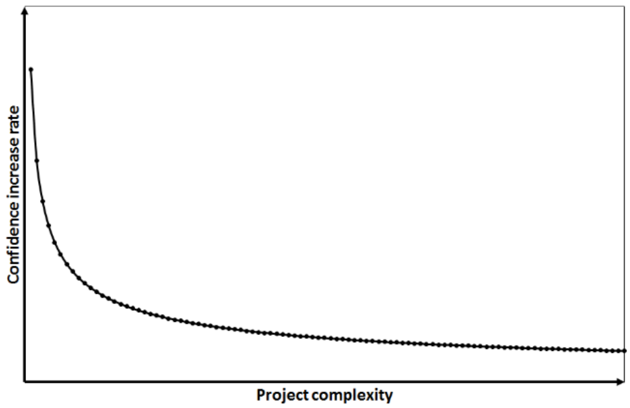


Figure 51: the relationship between confidence increase rate and the project complexity

For other phases of the NPD project, instead of the relationship between the number of concepts and 'confidence increase rate' (Equation 13) which is only applicable for the concept confidence phase, there is a relationship between the level of confidence of the final concept which leaves the conceptual design phase, and the rate of increase in the level of design confidence in other phases. This relationship is shown in Equation 16. The relationship is still formulated using a power function in which '*PD'*, as the strength of the effect of the concept confidence on the rate of confidence increase in downstream phases, is always greater than one, which means increasing the concept confidence always exponentially increases the rate of increase in confidence in other phases of the project. Relationships shown in Equation 14 and Equation 15 are applicable for formulating 'concept increase rate' in other phases. Based on Equations 13, 14, 15, and 16, the flow of 'confidence increase rate' in all phases of the project could be formulated as shown in Equations 17 and 18. In these equations, '*FC'* is a positive coefficient added to convert relationships into a single mathematical equation.

$$Confidence\ increase\ rate \propto (Concept\ confidence)^{PD} \quad 16$$

$PD > 1$

Conceptual design phase:

$$Confidence\ increase\ rate = FC * (No\ of\ concepts)^{PC} * (No\ of\ iterations)^{PI} * (Project\ complexity)^{PS} \quad 17$$

$FC > 0$

Other phases:

$$Confidence\ increase\ rate = FC * (Concept\ confidence)^{PD} * (No\ of\ iterations)^{PI} * (Project\ complexity)^{PS} \quad 18$$

$FC > 0$

Figure 52 shows the stock-and-flow model developed for the design confidence, and all variables affecting it. In this Figure, 'Confidence Increase Switch' is an auxiliary conditional variable defined to allow the confidence to increase just once per iteration cycle (Equation 19). To count the number of iteration cycles a simple stock-and-flow diagram is used as shown in Figure 52 and Equations 20, and 21. Based on the model the formulation of the flow of 'confidence increase rate' could be updated as shown in Equations 22, and 23.

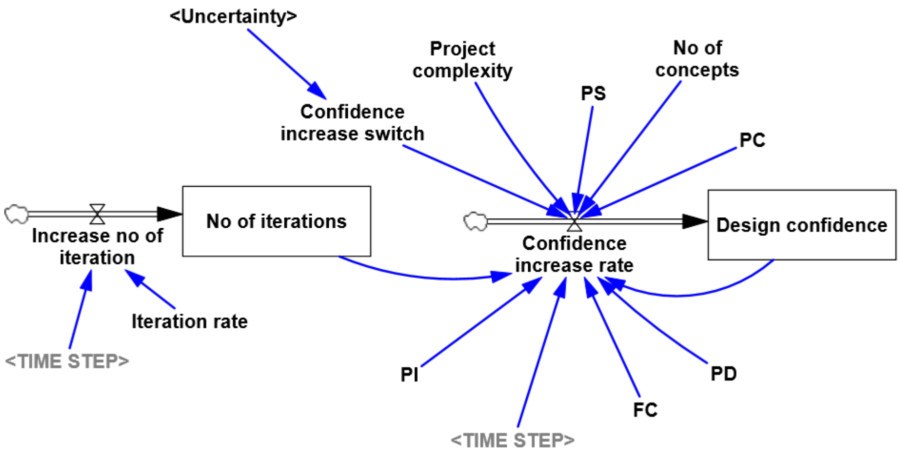


Figure 52: the stock-and-flow model of design confidence

$$Confidence\ increase\ switch = \begin{cases} 1, & Iteration\ rate > 0\ and\ Uncertainty > 0 \\ 0, & Iteration\ rate = 0\ or\ Uncertainty \leq 0 \end{cases} \qquad 19$$

$$No\ of\ iterations = \int_0^t (Increase\ no\ of\ iteration)dt \qquad 20$$

$$Increase\ no\ of\ iteration = \begin{cases} \frac{1}{Time\ Step}, & Iteration\ rate > 0 \\ 0, & Iteration\ rate = 0 \end{cases} \qquad 21$$

Conceptual design phase:

$$Confidence\ increase\ rate = \begin{cases} \frac{(FC*(No\ of\ concepts)^{PC}*(No\ of\ iterations)^{PI}*(Project\ complexity)^{PS})}{Time\ Step}, & Confidence\ increase\ switch = 1 \\ 0, & Confidence\ increase\ switch = 0 \end{cases} \qquad 22$$

Other phases:

$$Confidence\ increase\ rate = \begin{cases} \frac{(FC*(Concept\ confidence)^{PD}*(No\ of\ iterations)^{PI}*(Project\ complexity)^{PS})}{Time\ Step}, & Confidence\ increase\ switch = 1 \\ 0, & Confidence\ increase\ switch = 0 \end{cases} \qquad 23$$

The last point in this Section is the effect of the design confidence in each phase of the project on the '*Rework probability*. As mentioned in Section 4.6.3.1.1, after the stock of 'Tasks *Pending Test'* tasks flow through '*Rework Detection Rate*' or '*Approve Rate*' based on the value of '*Rework probability*'. '*Rework probability*' is equal to the amount of random uncertainty in the process, which although is irreducible, its effects could be mitigated (Suss and Thomson, 2012). To show this mitigation, at each point during the design process, '*Rework probability*' is calculated as a function of '*Initial Rework probability*', which is defined as a constant for each phase of the project to represent the irreducible nature of the random uncertainty, and 'design confidence' at that point. The relationship between the design confidence and the rework probability is captured using a power function. Figure 53 shows that increasing the design confidence during each phase of the project reduces the rework probability during that phase, so a bigger percentage of tasks are approved after the testing process. Based on this relationship, the rework probability could be formulated as shown in Equation 24, in which '*PG', as* the strength of the effect of design confidence on rework probability, is a positive number greater than one.

$$Rework\ probability = Initial\ rework\ probability * (1 - Design\ confidence)^{PG} \qquad 24$$

$$PG > 1$$

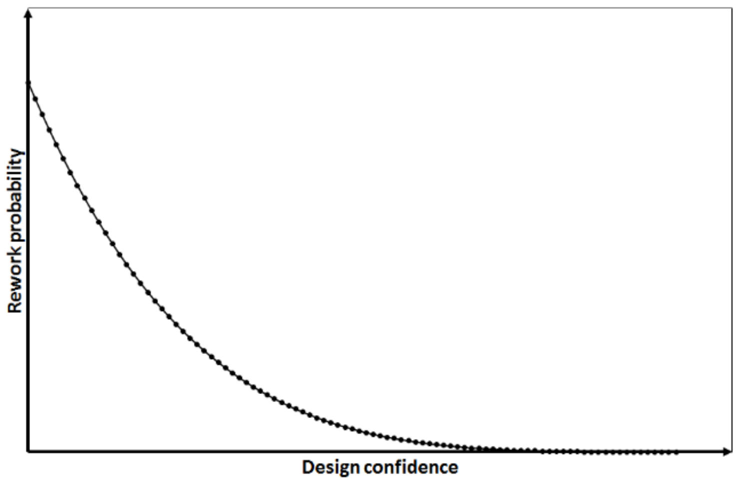


Figure 53: the relationship between the design confidence and the rework probability at each phase of the project

4.6.3.1.2 The work flow sector for the conceptual design phase

In addition to the general part of the workflow sector, the conceptual design phase includes an additional part to model the SBCE approach, and to quantify the SBCE loop described in Section 4.5.3.1. As mentioned earlier, in the SBCE approach the project starts with partitioning of the design space into a set of concepts. Working on each concept could be modelled using a separate workflow sector as shown in Figure 54. Similar to the description in the previous section, the level of confidence for each concept increases simultaneously after each cycle of iteration until reaching to a point when the convergence is starting. The level of concept confidence at this point, which is represented in the model by a variable called '*F1'*, is a percentage of the level of target confidence in the final concept.

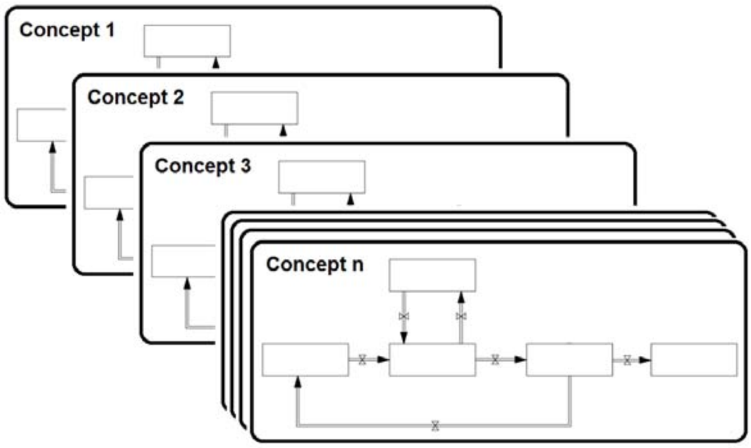


Figure 54: working in parallel on defined concepts in the conceptual design phase

To model the convergence process through which less feasible concepts are eliminated, a new outflow, called '*Convergence Rate*', from the stock of '*Tasks Pending Decision*' to the new stock of '*Set Reduced*' is added to the workflow sector. Adding a new outflow changes the formulation of the stock of '*Tasks Pending Decision*' (Equation 25). To formulate the flow of '*Convergence Rate*' an auxiliary variable, called '*Iteration Switch*' is added to the model. '*Iteration switch*' is a conditional variable, which its value depends on the amount of uncertainty and a variable called '*Tasks Remained'*. '*Tasks remained*' calculates the number of tasks still in the process (Equation 27), and is an indicator of the completion of an iteration cycle. If an iteration cycle is not completed yet, the value of the tasks remained is greater than zero, so the iteration switch stays off, because only after a complete iteration cycle the design review session is held to make the decision about starting another cycle of iteration. Reducing '*tasks remained*' to zero means that all tasks reside in the stock of '*Tasks Pending Decision*'. In this situation, as mentioned previously, the value of uncertainty determines if there is a need for another cycle of iteration or not. If uncertainty is still less than '*F1'*, 'iteration switch' takes the value of one, and according to Equation 28, all tasks in the stock of '*Tasks Pending Decision*' flow through '*Iteration Rate*' back to the stock of 'tasks not complete' to start another cycle of iteration. If the uncertainty exceeds the value of '*F1'*, '*iteration switch*' becomes two, which lets the convergence to start. According to Equation 30, a fraction of tasks resides in the stock of '*Tasks Pending Decision*', which is determined by the value of '*Reduction Ratio'*, flow through '*Convergence Rate*' to the stock of 'sets reduced'. The rest of tasks still flow back to the stock of '*Tasks Not Completed*' to start a new iteration cycle, as the uncertainty is still greater than zero. This process continues until the value of uncertainty falls to zero, and the number of tasks in the process become equal to the number of tasks in a single concept (calculated using Equation 30). This situation indicates the final concept is emerged out of the iteration-and-convergence process, and is ready to be released to the detail design phase. For activating the '*Release Rate*' another conditional variable, called '*Release Switch*' is added to the model, and formulated according to Equation 31. The updated formulation for the '*Release Rate*' is shown in Equation 32. '*F1'*, '*Reduction Ratio*', and the number of concepts are SBCE-specific variables, defined for this model. Having higher '*F1'* means that concepts spend more time in iteration cycles before converging starts. In addition, lower reduction ratio means that the convergence period takes longer time during which iteration cycles take place. In both situation, the level of the final concept confidence increases.

$$Tasks\ pending\ decision = \int_0^t (Approve\ rate - Iteration\ rate - Release\ rate - Convergence\ rate)dt \quad 25$$

$$Sets\ reduced = \int_0^t (Convergence\ rate)dt \quad 26$$

$$Tasks\ remained = (Tasks\ not\ completed + Tasks\ pending\ rework + Tasks\ pending\ test) * No\ of\ concepts \quad 27$$

$$Iteration\ switch = \begin{cases} 0, & Tasks\ remeined > 0 \\ 1, & Tasks\ remeined = 0\ and\ Uncertainty \leq F1 \\ 2, & Tasks\ remeined = 0\ and\ Uncertainty > F1 \end{cases} \quad 28$$

$$Iteration\ rate = \begin{cases} 0, & Iteration\ switch = 0 \\ \frac{Tasks\ pending\ decision}{Time\ Step}, & Iteration\ switch = 1 \\ \frac{(1-Reduction\ ratio)*Tasks\ pending\ decision}{Time\ Step}, & Iteration\ switch = 2 \end{cases} \quad 29$$

$$Convergence\ rate = \begin{cases} 0, & Iteration\ switch < 2 \\ \frac{Reduction\ ratio*Tasks\ pending\ decision}{Time\ Step}, & Iteration\ switch = 2 \end{cases} \quad 30$$

$$Release\ switch = \begin{cases} 0, & Sets\ reduced < \frac{Scope*(No\ of\ concepts-1)}{No\ of\ concepts}\ and\ Uncertainty \neq 0 \\ 1, & Sets\ reduced \geq \frac{Scope*(No\ of\ concepts-1)}{No\ of\ concepts}\ or\ Uncertainty = 0 \end{cases} \quad 31$$

$$Release\ rate = \begin{cases} \frac{Tasks\ pending\ decision}{Time\ Step}, & Release\ switch = 1 \\ 0, & Release\ switch \neq 1 \end{cases} \quad 32$$

As the described process in the general part of the workflow sector is performed in parallel for all defined concepts, increasing the number of concepts does not add to the duration of the conceptual design phase. Rather, as mentioned in Equations 22, and 23, having higher number of concepts, through the narrow partitioning of the design space, increases the 'concept confidence rate', thus reducing the conceptual design phase duration. Due to this and as all concepts are assumed to be identical in size in this model, a single model for the workflow sector would be enough for simulating the conceptual design phase. However, it should be considered that increasing the number of concepts increases the number of engineering resources allocated to the conceptual design phase, thus raising the cost of this phase. This effect is captured in the resource allocation sector of the model and is explained later. The complete stock-and-flow model of the conceptual design phase is illustrated in Figure 55.

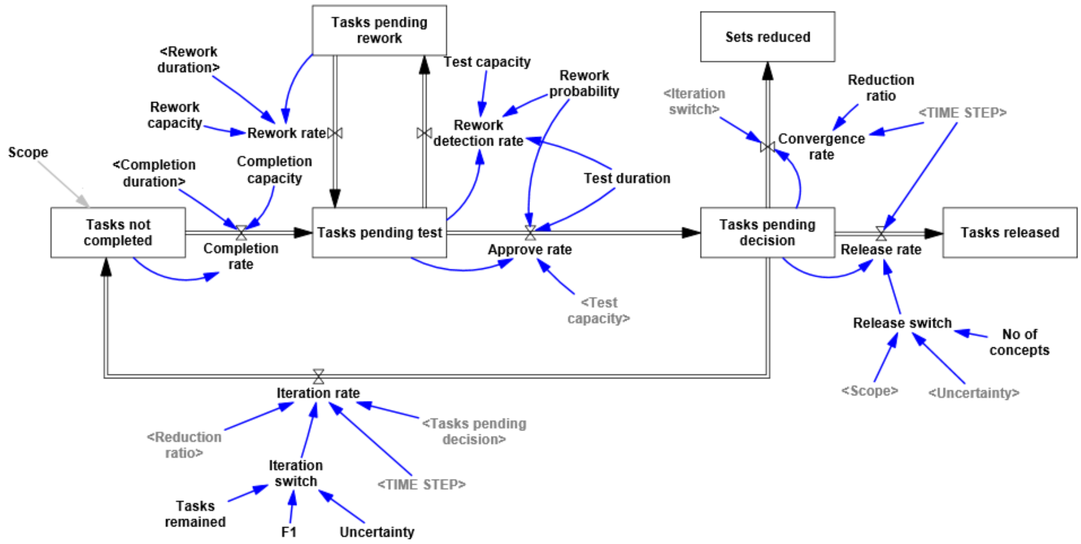


Figure 55: the stock-and-flow model of the conceptual design phase

4.6.3.1.3 The work flow sector for other phases

Other phases of the project have the general part of the workflow sector, explained in Section 4.6.3.1.1, in common with the conceptual design phase. However, to capture the effect of concurrent engineering on these phases, and to quantify the concurrency and corruption loops explained in Sections 4.5.3.4, and 4.5.3.5, they also have specific parts which are explained in this Section. As mentioned in Section 4.5.3.5 the long-term effect of the concurrency between phases of the project could be represented using the 'corruption loop'. Similar to the model developed by Taylor and Ford (2006), the corruption effect is added to the workflow sector in the form of a new inflow, called '*Corruption Rate*' to the stock of '*Tasks Not Completed*'. Similar to '*Iteration Rate*', this new flow just shows the transition in the state of the tasks and its duration is equal to the '*Time Step'* defined for the simulation of the model. As shown in Equation 33, '*corruption rate*' is equal to a fraction of tasks in the stock of '*Tasks Pending Decision*' which flow into the stock of '*Tasks Not Completed*' in the unit of time. This fraction is defined by '*Corruption Probability*', as an auxiliary variable whose value depends on the concurrency between upstream and downstream phases, the progress of upstream phase, and the rework probability in the upstream phase which results in updating the information has previously been sent to the downstream phase. In this model, 'corruption probability' is formulated as a conditional variable, according to Equation 34. Tasks remain in upstream phase represented the progress of work in this phase, so if it is equal to zero, or there is no concurrency between upstream and downstream phases, the corruption probability would be equal to zero. On the other hand, if the progress of work in the upstream phase is not completed yet, and some tasks are still in the process, and there is a level of concurrency between phases, the corruption probability is calculated as a fraction of rework probability in the upstream phase. '*F3*' is a new constant in the model which defines the strength of corruption. Concurrency is defined as a switch in the model. For formulating this variable, first the total number of tasks in each phase of the project should be calculated using Equation 35. In the conceptual design phase, '*Total Phase tasks*' is equal to the phase scope, but in other phases it is different, due to the inflow of '*Corruption rate*' which is added to the initial number of tasks defined by the phase scope. Next step is to compare the total tasks with the number of tasks which reside in the stock of '*tasks released*' plus the stock of 'set reduced' for the conceptual design phase. According to Equation 36, if the start time of the downstream phase is passed while still all tasks in the upstream phase are not released, there is concurrency between phases, so the concurrency switch is on. The value of '*Time'* in the Equation 36 is calculated by the built-in counter in the simulation software, while the value of '*Phase Start Time*' should be entered to the model from the real project data. Finally, to have phases performed concurrently, Equation 4 for the flow of '*Completion Rate*' as the first activity in the phase, which represents the starting point of each phase, should be updated for downstream phases, according to the conditional Equation 37. The complete stock-and flow model for phases of the project, except the conceptual design phase, is shown in Figure 56.

$$Corruption\ rate = \frac{Corruption\ probability * Tasks\ pending\ decision}{Time\ Step} \qquad 33$$

$$Corruption\ probability_{Down} = \begin{cases} F3\ * Rework\ probability_{Up}, & Tasks\ remained_{Up} > 0\ and\ Concurrency_{Up=Down} > 0 \\ 0, & Tasks\ remained_{Up} = 0\ or\ Concurrency_{Up=Down} = 0 \end{cases} \qquad 34$$

$$Total\ phase\ tasks = Tasks\ remained + Tasks\ released + Tasks\ pending\ decision + Sets\ reduced \qquad 35$$

$$Concurrency = \begin{cases} 1, & Tasks\ released_{Up} < Total\ phase\ tasks_{Up} and\ Time\ \geq Phase\ start\ time_{Down} \\ 0, & Tasks\ released_{Up} \geq Total\ phase\ tasks_{Up} or\ Time < Phase\ start\ time_{Down} \end{cases} \qquad 36$$

$$Completion\ rate = \begin{cases} MIN(Tasks\ not\ completed / Completion\ duration, Completion\ capacity), & Time\ \geq Phase\ start\ time \\ 0, & Time < Phase\ start\ time \end{cases} \quad (37)$$

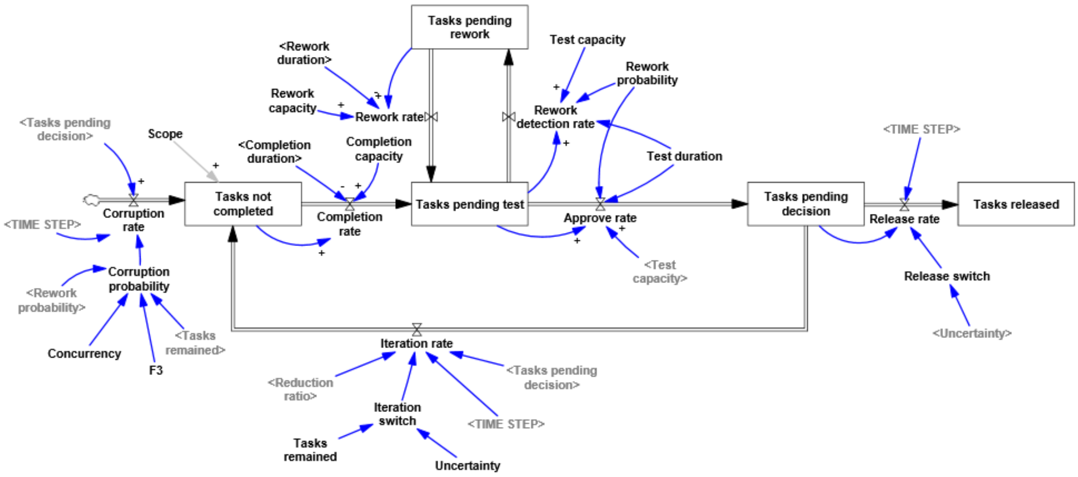


Figure 56: the stock-and-flow model of other phases of the project

4.6.3.2 The schedule pressure sector

Using the workflow sector of the model, the SBCE, rewok, concurreny, and corruption loops identified in the qualitative model are quantified. The schedule pressure loop, as explained in Section 4.5.3.2, could be quantified using the structure and equations involve in the schedule pressure sector of the model. As each phase of the project has a tight deadline, approaching it increases the amount of the schedule pressure on the managerial team. *'Schedule pressure'*, as shown in Figure 57 and Equation 38, is the difference between the time required to finish a phase of the project and the time available to finish that phase. '*Time required*' is calculated, according to Equation 39, by deviding the number of tasks remined to be completed in each phase (Equation 27), to the total capacity allocated to that phase, which is explained in Section 4.6.3.3. Equation 40 shows the formulation of '*Time available*' as the difference between the phase deadline and the time at the instant of calculation. As mentioned earlier, '*Time'* is calculated by the built-in counter in the simulation software, and the value of '*Phase deadline*' should be entered as an external constant to the model from the real project data, separately for each phase.

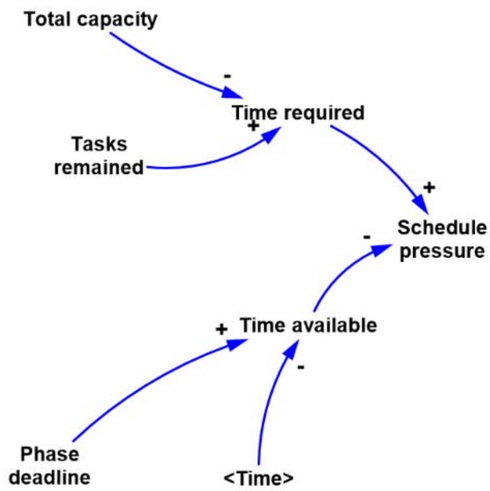


Figure 57: modelling the schedule pressure

$$Schedule\ pressure = \frac{Time\ required}{Time\ available} \quad 38$$

$$Time\ Required = \frac{Tasks\ remained * no\ of\ concepts}{Total\ capacity} \quad 39$$

$$Time\ available = Phase\ deadline - Time \quad 40$$

The schedule pressure could result in different intervention by managers, as the on-time delivery of products is always the first priority in companies. For this model, as shown in Figure 58, two types of managerial interventions which affect the project structure and result in moving from a SBCE approach to the PBCE approach are discussed. The schedule pressure could also result in increasing the overlap between consecutive phases of the project, but for this research the concurrency between phases is modelled as a constant which is externally determined using data from the real projects. In addition, slipping deadlines of the project, increasing the work intensity and overtime, and hiring new resources for the project could be other consequences of the schedule pressure, which are extensively discussed by Lyneis and Ford (2007), but are omitted in this model.

'*Target confidence*' as a variable in the model is previously used in Equation 11 to calculate the level of uncertainty in the conceptual design phase. According to Equation 11 the level of uncertainty could be decreased by either increasing the level of design confidence through iteration cycles, or by decreasing the level of target confidence. Changing the type of 'Target confidence' from an auxiliary variable to a first-order stock-and-flow diagram, as shown in Figure 58 and Equation 41, allows it to be decreased under the influence of schedule pressure. The rate of decreasing the target confidence due to the schedule pressure could be formulated using Equation 43, and a conditional variable. '*F2*' is a constant added to the model for calculating the increments of target change. By reducing the target confidence, the gap between it and the current level of design confidence decreases more rapidly, and with less number of iteration cycles, so the duration of the conceptual design phase decreases.

'*Reduction ratio*' is a SBCE-specific variable defined in this model which regulates the duration of the convergence period. As mentioned in Section 4.6.3.1.2, '*Reduction ratio*' affects the duration of the convergence period during the conceptual design phase, as having low reduction ratio means performing more iteration cycles to reduce the uncertainty. Having a long convergence period means spending more time for comparing concepts and eliminating inferior ones, and as Ford and Sobek (2005) stated, is one of distinctive features of the SBCE approach, compared to the PBCE approach. The effect of the schedule pressure on increasing the reduction ratio is modelled and formulated similar to the target confidence, which is discussed above, and is shown in Equations 42, and 44.

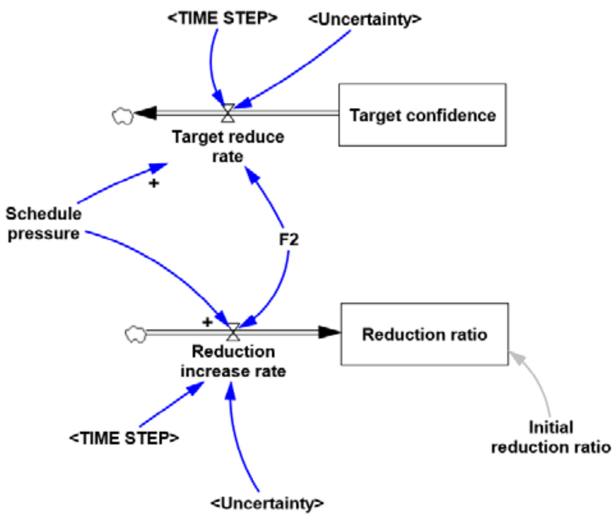


Figure 58: modelling the effect of the schedule pressure on other variables in the model

$$Target\ confidence = \int_0^t (-Target\ reduce\ rate)dt \qquad 41$$

$$Reduction\ ratio = \int_0^t (Reduction\ increase\ rate)dt \qquad 42$$

$$Target\ reduce\ rate = \begin{cases} \frac{F2}{Time\ Step}, & Schedule\ pressure \geq 1\ and\ Uncertainty > 0 \\ 0, & Schedule\ pressure < 1\ or\ Uncertainty = 0 \end{cases} \qquad 43$$

$$Reduction\ increase\ rate = \begin{cases} \frac{F2}{Time\ Step}, & Schedule\ pressure \geq 1\ and\ Uncertainty > 0 \\ 0, & Schedule\ pressure < 1\ or\ Uncertainty = 0 \end{cases} \qquad 44$$

4.6.3.3 The resource management sector

As mentioned earlier, completion, testing and rework are three types of activities which are performed in each phase of the project. They are modelled as flow variables and formulated according to Equations 4 to 7. In these equations the project capacity for each activity, which depends on the number of engineering resources allocated to that activity, is a variable which constrains the rate of performing that activity. In addition, the number of resources is a variable used directly towards the calculation of the project cost, which is explained in performance measurement sector of the model in Section 4.6.3.4.

As the focus of this research is on the process structure of NPD projects, rather than the resource management, allocating resources is modelled based on a simple approach, called the direct proportional policy by Joglekar and Ford (2005). This policy is based on the current stocks of work to be serviced by the different development activities. Allocating resources to each activity is in the same proportion that the activity's stock contributes to the total amount of work waiting to be done. In other words, each type of activity, represented by a flow in the model, receives resources equal to the size of its corresponding stock. The resource allocation is a dynamic process and changes constantly as the sizes of stocks differ. In addition to the size of stocks the '*Resource Productivity*' is defined in the model as a constant which affects the number of allocated resources. The resource productivity is equal to the amount of work each resource could perform in the unit of time. Figure 59 shows the structure used for modelling the desired number of resources for each activity. The number of tasks reside in the stock related to each activity, the time duration of performing each activity, and the productivity of resources influence the desired resources for each activity. In addition, in the conceptual design phase, as mentioned in Section 4.6.3.1.2, the number of concepts needs to be considered while calculating the resources. Equations 45, 46, and 47 are used to calculate the number of resources required for completion, rework, and testing activities.

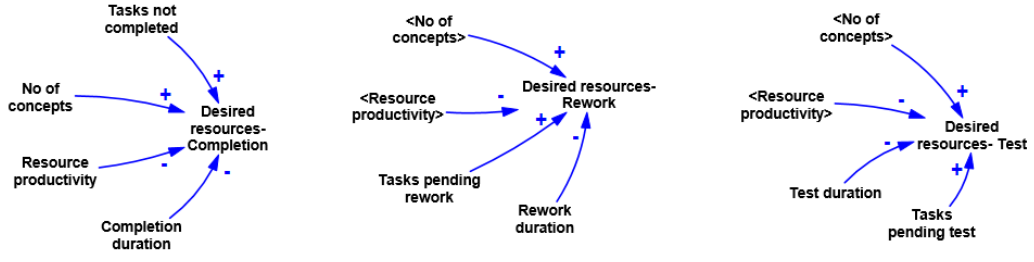


Figure 59: the model structure for desired resources

$$Desired\ resources_{Completion} = No\ of\ concepts * (Tasks\ not\ completed)/(\text{Completion duration})/(\text{Resource productivity}) \quad 45$$

$$Desired\ resources_{Rework} = No\ of\ concepts * (Tasks\ pending\ rework)/(\text{Rework duration})/(\text{Resource productivity}) \quad 46$$

$$Desired\ resources_{Test} = No\ of\ concepts * (Tasks\ pending\ test)/(\text{test duration})/(\text{Resource productivity}) \quad 47$$

Using the amount of desired resources for each activity, and the total number of resources available for this NPD project, it is possible now to calculate the number of resources allocated to each type of activity. The structure used in the model is shown in Figure 60. First it is needed to calculate the total number of desired resources for the project according to Equation 48, and then based on the ratio of total available resources and total desired resources, calculate the number of resources which is possible to allocate for each type of activity. Equations 49, 50, and 51 are used for the calculation of resources allocated to completion, rework, and testing activities.

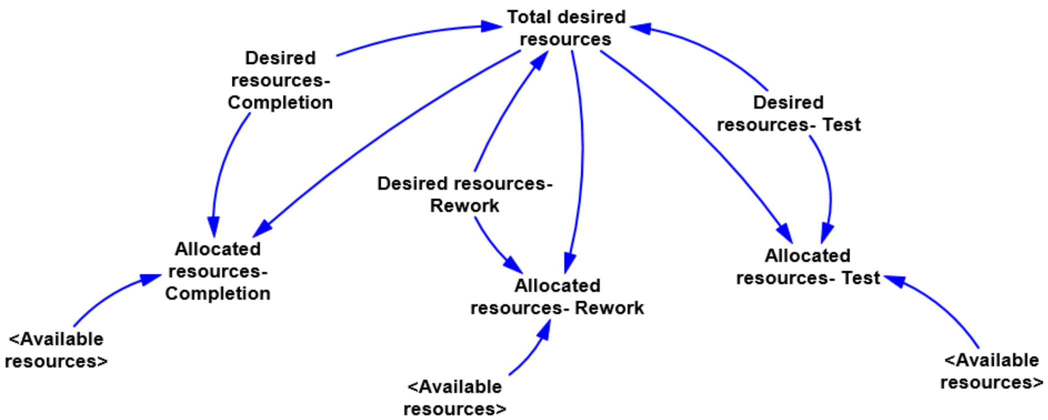


Figure 60: the model structure for the allocated resources

$$Total\ desired\ resources = Desired\ resources_{Completion} + Desired\ resources_{Rework} + Desired\ resources_{Test} \quad 48$$

$$Allocated\ resources_{Completion} = Desired\ resources_{Completion} * Available\ resources/Total\ desired\ resources \quad 49$$

$$Allocated\ resources_{Rework} = Desired\ resources_{Rework} * Available\ resources/Total\ desired\ resources \quad 50$$

$$Allocated\ resources_{Test} = Desired\ resources_{Test} * Available\ resources/Total\ desired\ resources \quad 51$$

Finally, using the number of allocated resources to different types of activities, and the resource productivity it is possible to calculate the capacity of those activity. The model structure is shown in Figure 61, and Equations 52, 53, and 54 represent the mathematical formulations for the completion, rework, and test capacity. The total capacity of the project, which is used in calculation of '*Time Required*' in the schedule pressure sector of the model is calculated using Equation 55.

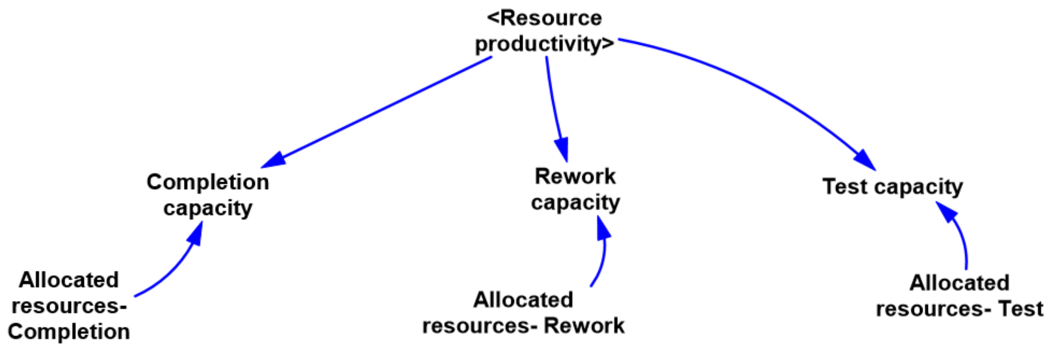


Figure 61: the model structure for the capacity for different activities

$$Completion\ capacity = Allocated\ resources_{Completion} * Resource\ productivity \tag{52}$$

$$Rework\ capacity = Allocated\ resources_{Rework} * Resource\ productivity \tag{53}$$

$$Test\ capacity = Allocated\ resources_{Rework} * Resource\ productivity \tag{54}$$

$$Total\ capacity = Total\ allocated\ resources * Resource\ productivity \tag{55}$$

$$Total\ allocated\ resources = Allocated\ resources_{Completion} + Allocated\ resources_{Rework} + Allocated\ resources_{Test} \tag{56}$$

4.6.3.4 The performance measurement sector

The performance measurement sector models the outputs of the model in the form of several performance indicators of the NPD project. Time, cost and quality are paremeters which are measured in the model to indicate the performance of the project, as discussed in following sections.

4.6.3.4.1 Duration

The duration is measured separately for each phase, and for the whole project. Figure 62 shows the model structure employed for the calculation of the duration. '*Phase Start Time*' is assumed constant in the model and, as mentioned earlier, should be entered externally from the real project data. But '*Phase Finish Time'* is modelled as a stock variable and is calculated using Equation 57. The flow of '*Change in Phase Time*' is a conditional variable and is formulated as Equation 58. The difference between '*Phase Finish Time' and* '*Phase start Time'* is used for calculating the duration of each phase. '*Project duration'* is also modelled as a stock variable, which its value is changed based on the value of '*Phase Finish Time*' as shown in Figure 62 and Equation 59. According to Equation 60, as long as the sum of stocks of ‘*Phase Finish Time’* for all phases is greater than zero, the flow of ‘*Change in Project Time*’ increases the value of the stock of ‘*Project duration’.*

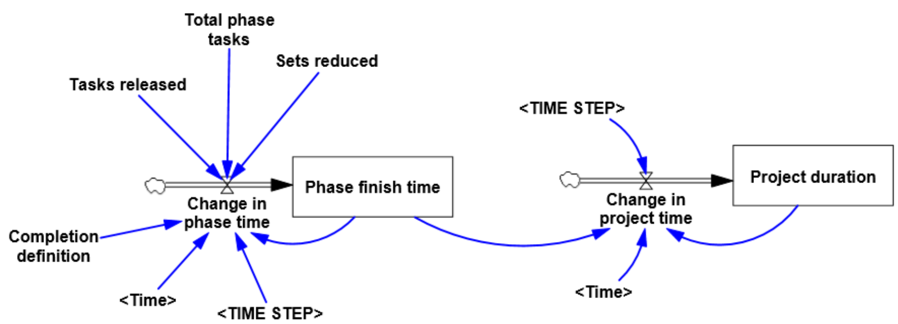


Figure 62: the model for calculating the duration of individual phases and the overall project

$$Phase\ finish\ time = \int_0^t (Change\ in\ phase\ time)dt \tag{57}$$

$$Change\ in\ phase\ time = \begin{cases} \frac{Time}{Time\ Step}, & Tasks\ released + Sets\ reduced \geq Total\ phase\ tasks \\ 0, & Tasks\ released + Sets\ reduced < Total\ phase\ tasks \end{cases} \tag{58}$$

$$Project\ duration\ = \int_0^t (Change\ in\ project\ time)dt \tag{59}$$

$$Change\ in\ project\ time = \begin{cases} \frac{Time}{Time\ Step}, & SUM(Phase\ finish\ time) \neq 0 \\ 0, & SUM(Phase\ finish\ time) = 0 \end{cases} \tag{60}$$

4.6.3.4.2 Cost

Similar to the duration, the cost of phases and the overall project cost are calculated separately in the model (Figure 63). The stock of ‘*Person Month*’ is defined as an auxiliary variable for calculating the cost, which its value is changed, according to Equation 61, by integrating the total resources allocated to each phase of the project. The formulation of ‘*Total Allocated Resources*’ is shown as Equation 56 in Section 4.6.3.3. ‘*Phase cost*’ is the product of ‘*Person Month*’ and ‘*Unit Cost*’ (Equation 62). ‘*Unit Cost*’ is a constant which its value is calculated during the model calibration. Finally, the overall cost of the project is calculated by adding the costs of individual phases according to Equation 63.

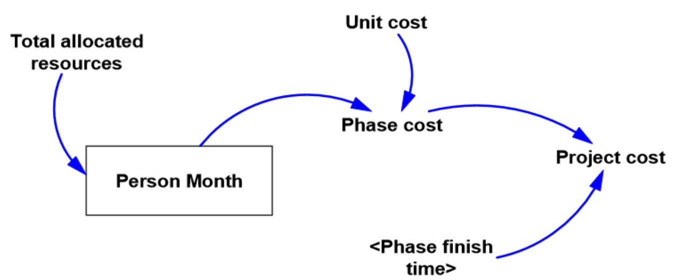


Figure 63: the model for calculating the cost of individual phases and the overall project

$$Person\ Month\ = \int_0^t (Total\ allocated\ resources)dt \quad 61$$

$$Phase\ cost\ = Person\ Month * Unit\ cost \quad 62$$

$$Project\ cost = \begin{cases} SUM(Phase\ cost), & SUM(Phase\ finish\ time) \neq 0 \\ 0, & SUM(Phase\ finish\ time) = 0 \end{cases} \quad 63$$

4.6.3.4.3 Quality

The percentage of tasks which are reworked throughout the project, and the final value of '*Design Confidence*' are used as measures for the quality in this model. '*Rework per phase*' is modelled as a stock variable whose value depends on '*Rework Rate*' at each phase of the project (Figure 64 and Equation 64). '*Reworks per Project*' is calculated by adding the value of '*Rework per phase'* for individual phases of the project as shown in Equation 65. The percentage of reworks throughout the project is formulated as the ratio of '*Reworks per Project*' and '*Total project activities*' (Equation 66). The value of '*Total Project Activities*' is calculated based on the number of activities in each phases of the project using Equations 67 and 68.

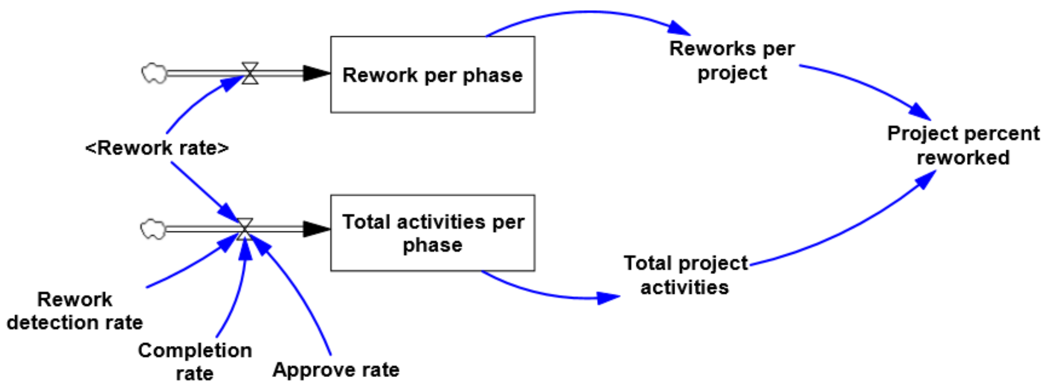


Figure 64: the model for calculating the percentage of rework for individual phases and the overall project

$$Rework\ per\ phase\ = \int_0^t (Rework\ rate)dt \quad 64$$

$$Reworks\ per\ project = SUM(Rework\ per\ phase) \quad 65$$

$$Project\ percent\ reworked = \frac{Reworks\ per\ project}{Total\ project\ activities} * 100 \quad 66$$

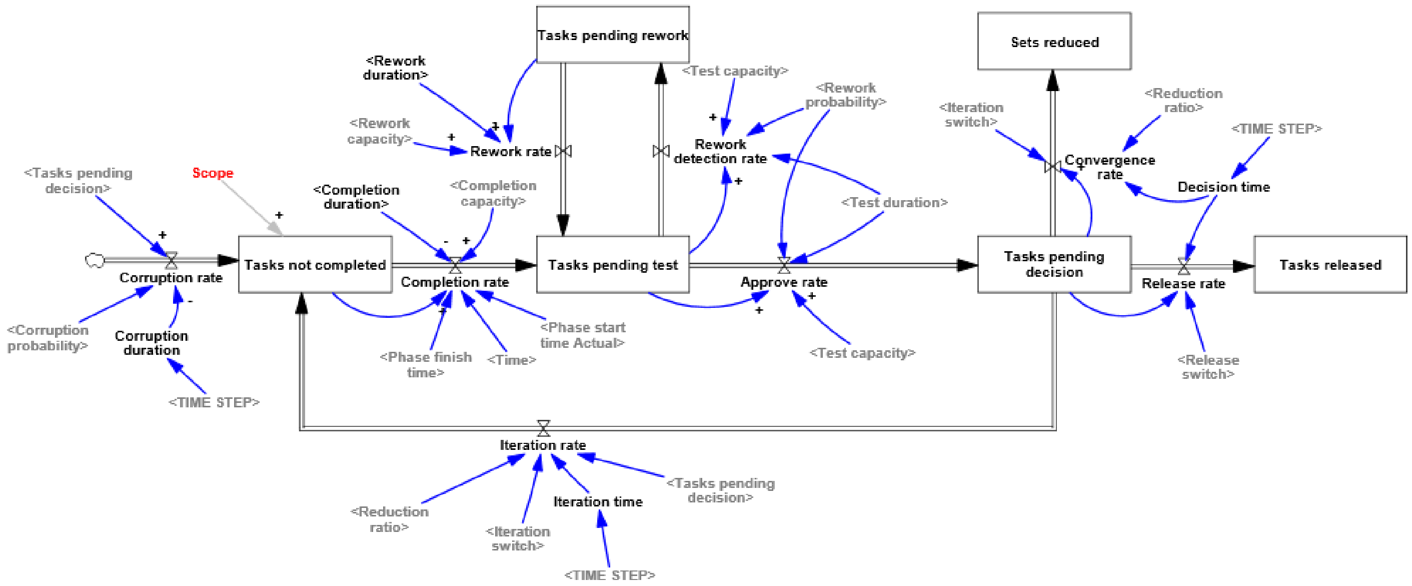
$$Total\ activities\ per\ phase\ = \int_0^t (Approve\ rate + Completion\ rate + Rework\ detection\ rate + Rework\ rate)dt \quad 67$$

$$Total\ project\ activities = SUM(Total\ activities\ per\ phase) \quad 68$$

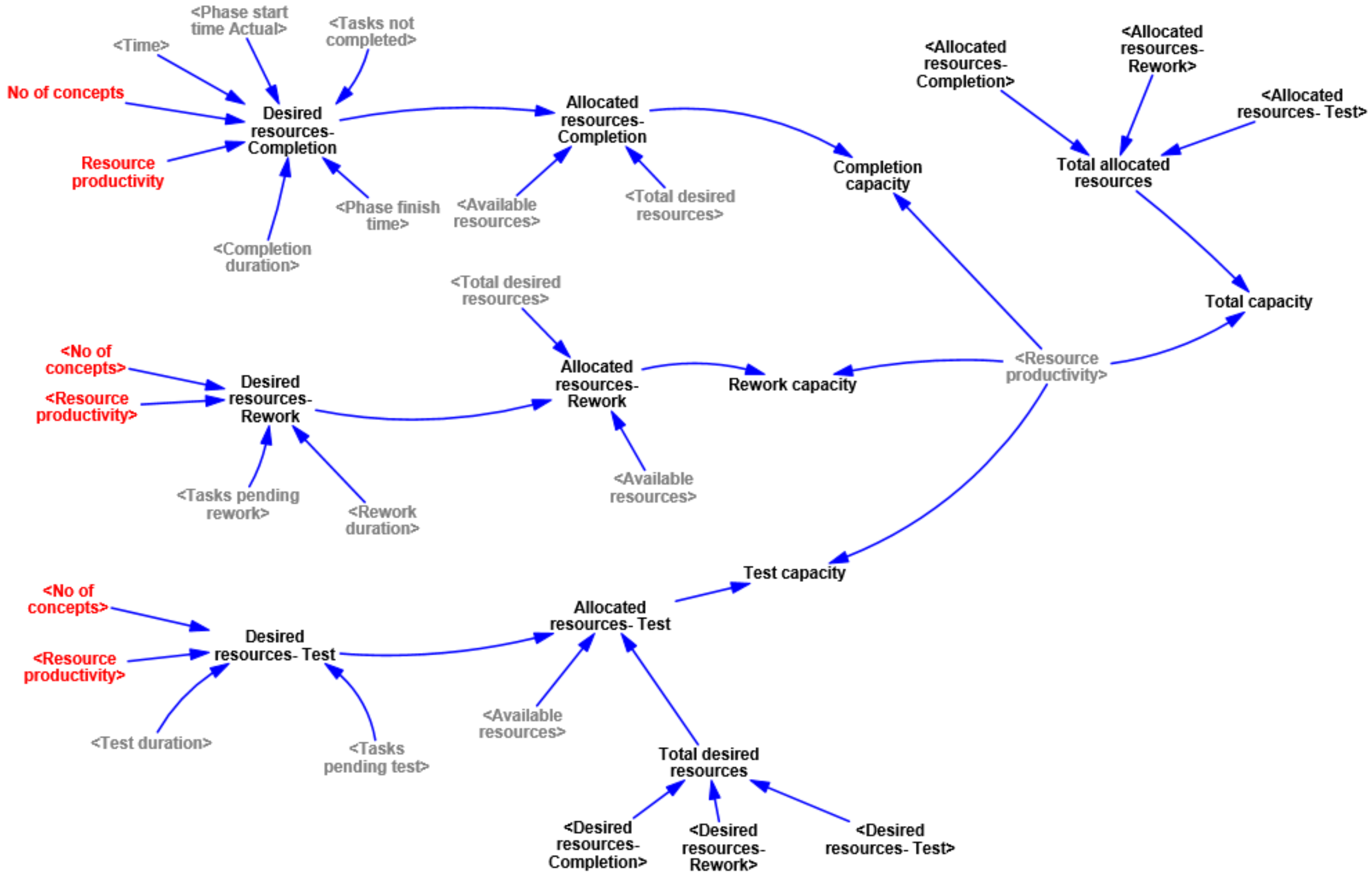
4.7 The system dynamics model

The stock-and-flow model developed for this research with all sectors described in the previous section is presented in this section.

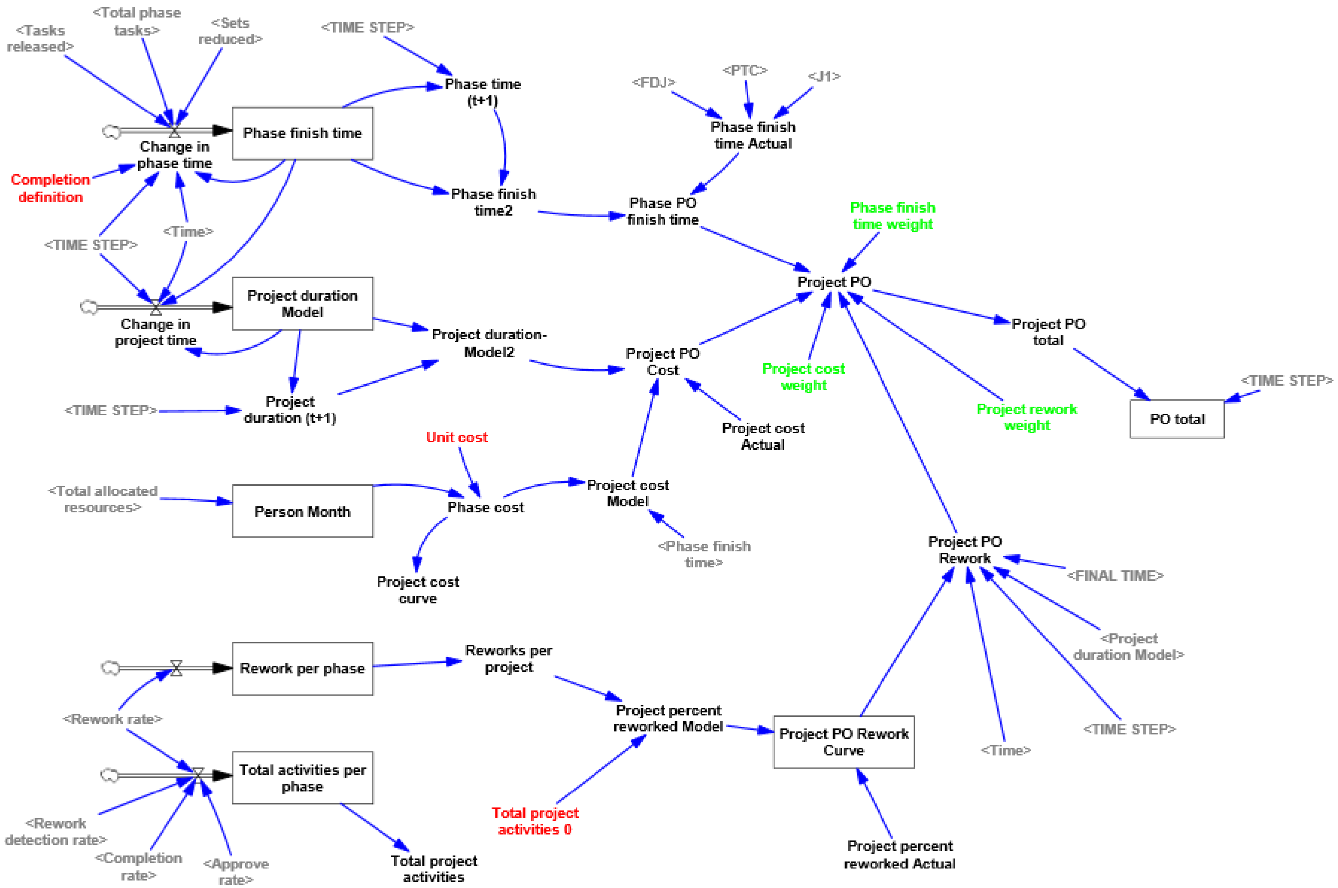
The Workflow Sector



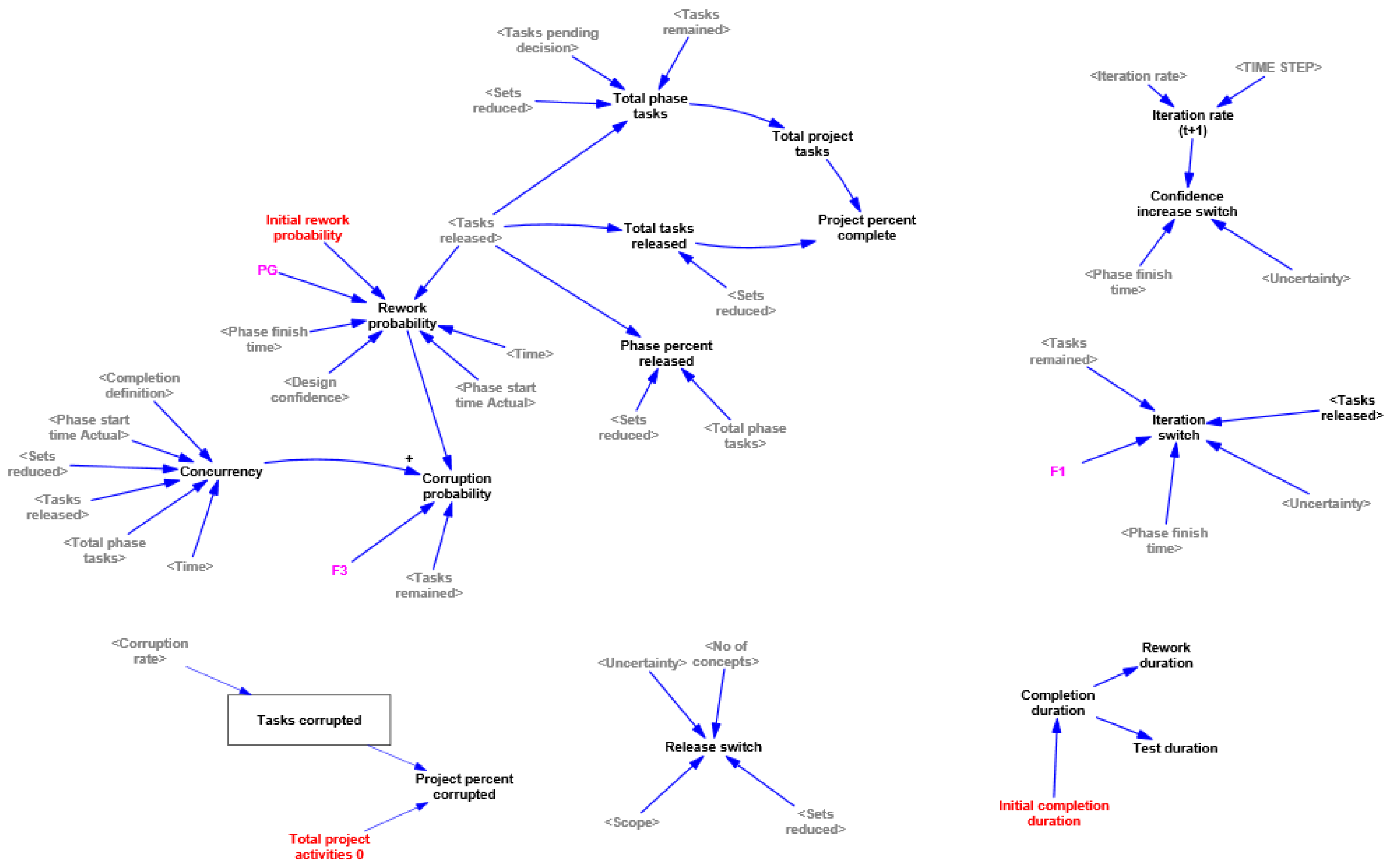
The Resource Management Sector

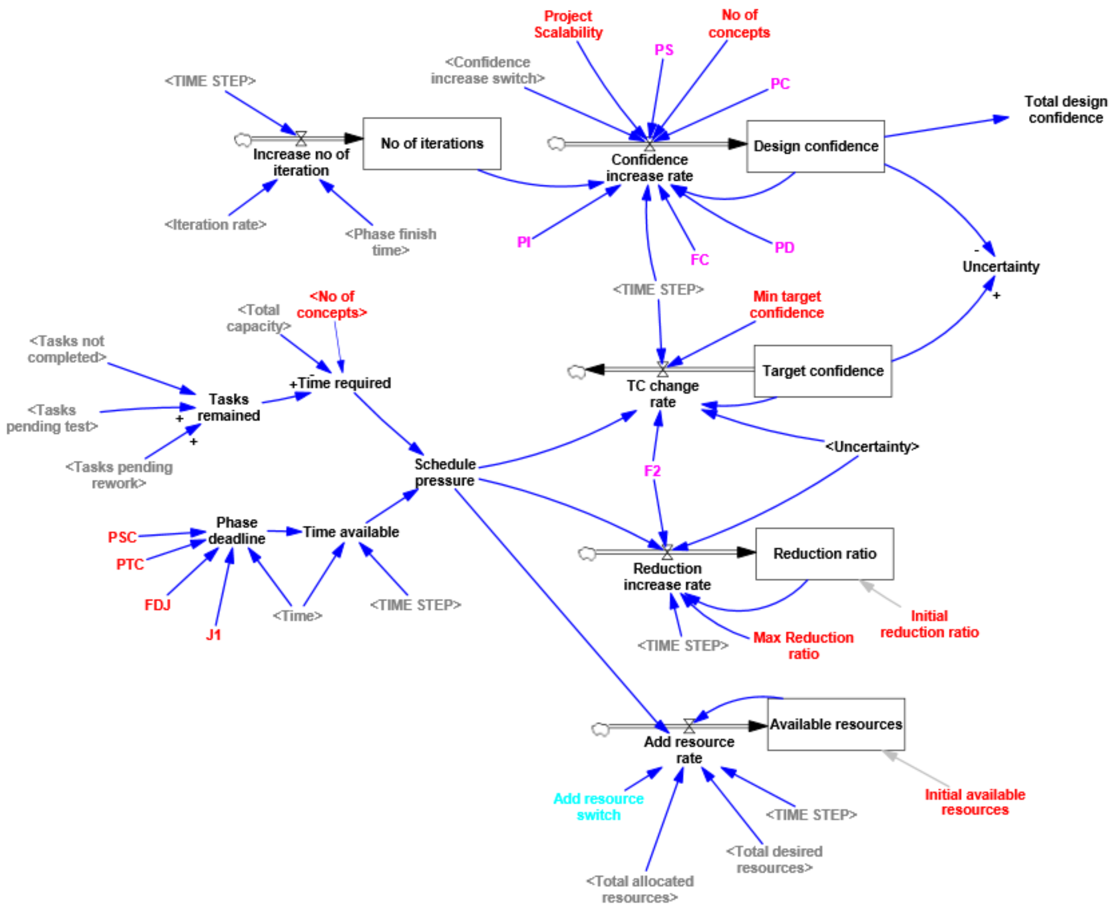


The Performance Measurement Sector



Auxiliaries





Project Scalability
No of concepts
PS
PC
<Confidence increase switch>
<TIME STEP>
Total design confidence
Increase no of iteration
No of iterations
Confidence increase rate
Design confidence
<Iteration rate>
<Phase finish time>
PI
FC
PD
Uncertainty
<TIME STEP>
Min target confidence
<Total capacity>
<No of concepts>
<Tasks not completed>
Time required
Tasks remained
<Tasks pending test>
<Tasks pending rework>
TC change rate
Target confidence
<Uncertainty>
Schedule pressure
F2
Phase deadline
Time available
PSC
PTC
FDJ
J1
<Time>
<TIME STEP>
Reduction increase rate
Reduction ratio
Initial reduction ratio
<TIME STEP>
Max Reduction ratio
Add resource rate
Available resources
Add resource switch
<TIME STEP>
Initial available resources
<Total desired resources>
<Total allocated resources>

4.8 Summary

In this Chapter, the process of building a system dynamics model for studying the effect of LPD process structure on the performance of a development project is discussed in detail. First, the causal relationships between the elements inside the system boundary are demonstrated using several causal loop diagrams in the qualitative stage. Then based on these feedback loops, a stock-and-flow diagram is developed and mathematically formulated. The development of the model covers the second research objective: '*To develop a system dynamics model to investigate the impact of LPD on development project performance*'. All variables in the model and their mathematical formulations are shown in Appendix. In the next chapter, testing the model, validating it using the real project data, and running it to compare different scenarios developed based on the research objectives are discussed in detail.

5 Model Validation

5.1 Introduction

In the previous chapter the system dynamics model is built qualitatively using the causal loop diagrams, and then formulated mathematically in the form of stock-and-flow diagrams. The model is now ready to address the purpose of this research and be used for the study of the LPD process structure to find an explanation for the superiority of an LPD process design over traditional approaches. However, before starting the policy analysis using the model as the third objective of this research, it is needed to validate the model. In section 3.5.2, model validation is mentioned as a part of the development phase of the research to address the second research objective.

This Chapter starts with introducing the validation tests selected for this research in Section 5.3. The result of the boundary adequacy test is explained in Section 5.4. Section 5.4.2 includes the results of the dimensional consistency test. The structure of the model is assessed in Section 5.4.3, which is divided into assessing the workflow sector of the model and the effect of the number of concepts, concurrency, and the project complexity, the schedule pressure sector, and the resource management sector. In Section 5.4.4 the extreme condition test is explained, and finally, in Section 5.5 the behaviour reproduction test through which the model is calibrated using the real data is explained. The Chapter ends with a summary Section.

5.2 The purpose of model validation

One of the fundamental questions when using simulation models to address issues in organizations is their validity. In other words, making a relationship between artificial simulation experiments and the real-world behaviour is an issue in simulation models. Harrison, et al. (2007) proposed several ways for empirically grounding the model. If the model is built based on an empirical work there would not be any problem for validation. However, even if empirical information is not available, still it is possible to use the sensitivity analysis as a tool for testing the robustness of the model outcomes. The simulation results could be validated using a comparison with empirical works, or they could make a foundation for future empirical works. Making decision about the types of validation is based on the purpose of the model; for example, a predictive model uses the empirical testing of the result, while in a prescriptive model, to increase the usefulness of the results, the processes and relationships should also be validated. Even models which could not be empirically assessed, as Harrison, et al. (2007) stated, are valuable research tools to explore the results of theoretically derived processes.

Although the validity of a simulation model is assumed to be related to its degree of closeness to the reality, it is more encompassing to focus on the relevancy of the model to its purpose than on its level of abstraction, as relevancy could be obtained even with models which are not necessarily close to reality (Burton and Obel, 1995). For example, in system dynamics, Forrester (1994b) stated that there is not a way for validation of a theory which claims representing the behaviour of a real system. He more emphasised on '*adequacy*' instead of '*validity*' and claimed that only a certain level of confidence is achievable in a system dynamics model. This level of confidence is a trade-off between the adequacy of the model for its purpose, and the time and effort for further improving the model.

Although the fundamental concern of validity in social science and simulation models are the same, validity constructs in social science are not directly applicable to the modelling research. Emphasising on the realism in simulation models may result in losing the purpose of the model in the quest for realism, but on the other hand, without some degree of realism, the model would be just a logical or numerical exercise. Burton and Obel (1995) stated that the validity is a multi-criteria issue and simulation models could not perfectly address all these criteria. Therefore, there is a need for a trade-off between them. Among the concepts of validity, for instance, the content validity asks for the model to capture important aspects of the phenomena under study. The construct validity needs the model to contain variables and relationships which result in outcomes comparable to the real world, and criterion-related validity asks about the relationships between the purpose of the model and the real-life intent and how the results of the model are used (Burton and Obel, 1995).

Burton and Obel (1995) argued that for the validity of simulation models there should be a balance between the purpose of the model, the computational model, and the data analysis. A simple model which addresses its purpose is preferable to a complex model which lead to the poor answers. Without this balance, the outcome would be difficult to analyse, the purpose would not be met and the model would be too complex. After defining the purpose of the model as a statement of variables, parameters and relations could be built to meet this purpose. It is the purpose of the model that determine which variables should be included in the model. The last step is analysing and comparisons for different parameters values to test a hypothesis. The risk of having false implications from simple models is less than complex ones. So, the level of simplicity in the model needs to be in balance with its complexity.

5.3 Tests for validating the model

A system dynamics model of LPD is developed in the previous Chapter to describe the superiority of lean implementation in NPD projects over traditional approaches. Before conducting experiments, the developed model needs to be validated in terms of the correspondence of the model structure and the robustness of the model's behaviour. After validation and establishing confidence in the behaviour of the model, using a series of scenarios the variations in system behaviours are explained, and the consequences in terms of their effects on the performance of the NPD project are examined.

The aim of a system dynamics model is to link observable patterns of behaviour of a system to micro-level structure and decision-making processes (Qudrat-Ullah and Seong, 2010). Model validation is to make sure that the model supports the objective truth (Sterman, 2000), predictions made by the model reflect reality within the intended domain (Law and Kelton, 1991), and the model mimics the real world well enough for its stated purpose (Qudrat-Ullah and Seong, 2010). The goal of validating the model in this research is to confirm that the quantified model in the previous Chapter is correct and covers the elements of the qualitative model, which is built based on the knowledge acquired from the literature about LPD and to make sure that the model's behaviour is as intended.

Validation of the structure and the behaviour of the model could be assessed using several tests, which are extensively applied by researchers. It involves the identification of the appropriate structure which adequately captures the real system (structural validity), and assessing the closeness of the behaviour generated by the model to the observed behaviour in the real system (behavioural validity). The combination of the structural and behavioural validity makes the overall validity and confidence of the model (Qudrat-Ullah and Seong, 2010). Table *21* represents tests applied in this research for the structural and behavioural validity of the model, based on the suggestions by Barlas (1996), Sterman (2000) and Qudrat-Ullah and Seong (2010). In following sections, the steps taken to validate the system dynamics model are explained.

Table 21: types of tests used for the validation of the model

	Type of test	**Purpose of test**
Tests for structural validity	Boundary adequacy test	To determine if the important concepts and structures for addressing the problem are endogenous to the model
	Structural assessment test	To determine if the model structure is consistent with relevant descriptive knowledge of the system
	Dimensional consistency test	To determine if each equation in the model is corresponding dimensionally to the real system
	Parameter assessment test	To determine the consistency of model parameters with relevant descriptive and numerical knowledge of the real system
	Extreme condition analysis test	To determine the plausible responsiveness of the model when extreme values are assigned to parameters
Tests for behavioural validity	Behavioural reproduction test	To determine the ability of the model to reproduce behaviour pattern displayed by the real system
	Sensitivity analysis	To determine the plausibility in the model behaviour when the parameters value are plausibly shifted

5.4 Structural validity

5.4.1 Boundary Adequacy Test

The goal of applying boundary adequacy test is to assess if all relevant elements which are needed to satisfy the purpose of the model are included in the structure of the model. For this research, in the qualitative stage of the model development process, causal loop diagrams are created using the knowledge from the literature review. Figure 33 in previous Chapter summarizes the model boundary and the major endogenous and exogenous variables in the model. Comparing the model boundary and the causal loop diagrams indicates that, consistent with the purpose of the model, all variables for addressing the policy issues in NPD projects are generated endogenously.

5.4.2 Dimensional Consistency Tests

Dimensional consistency tests are conducted to ensure that, the measurement units of all variables and constants involved in each mathematical equation in the model are dimensionally consistent. In other words, phrases at both sides of the equal sign in mathematical equations should have homogeneous dimensions. This test is done using the dimensional analysis function embedded in *Vensim®* software, which automatically checks the units of variables in the model. The dimensional analysis shows that there is no unit error and all the variables and equations are consistent. Table 22 to

Table 24 shows the lists of all variables and constants in the model and their dimensions.

Table 22: list of stock and flow variables in the model and their dimensions

Type	**Variable**	**Dimension**
Stock	Tasks not completed	Task
	Tasks pending test	Task
	Tasks pending rework	Task
	Tasks pending decision	Task
	Sets reduced	Task
	Rework per phase	Task
	Tasks corrupted	Task
	Total activities per phase	Task
	Design confidence	Dimensionless
	No of iterations	Dimensionless
	Reduction ratio	Dimensionless
	Target confidence	Dimensionless
	Phase finish time	Month
	Project duration	Month
	Person Month	Month*Person
Flow	Increase no of iteration	1/Month
	Reduction increase rate	1/Month
	Target change rate	1/Month
	Confidence increase rate	1/Month
	Change in phase time	Dimensionless
	Change in project time	Dimensionless
	Completion rate	Task/Month
	Rework detection rate	Task/Month
	Rework rate	Task/Month
	Approve rate	Task/Month
	Release rate	Task/Month

	Iteration rate	Task/Month
	Convergence rate	Task/Month
	Corruption rate	Task/Month

Table 23: list of auxiliary variables in the model and their dimensions

Type	Variable	Dimension
Auxiliary	Phase cost	$
	Confidence increase switch	Dimensionless
	Iteration switch	Dimensionless
	Phase percent released	Dimensionless
	Project percent complete	Dimensionless
	Project percent corrupted	Dimensionless
	Project percent reworked	Dimensionless
	Schedule pressure	Dimensionless
	Total design confidence	Dimensionless
	Total No of iteration	Dimensionless
	Concurrency	Dimensionless
	Corruption probability	Dimensionless
	Release switch	Dimensionless
	Rework probability	Dimensionless
	Uncertainty	Dimensionless
	Time available	Month
	Time required	Month
	Desired resources- Completion	Person
	Desired resources- Rework	Person
	Desired resources- Test	Person
	Allocated resources- Completion	Person
	Allocated resources- Rework	Person
	Allocated resources- Test	Person
	Total allocated resources	Person
	Total desired resources	Person
	Reworks per project	Task
	Total phase tasks	Task
	Total project activities	Task
	Total project tasks	Task
	Total tasks released	Task
	Tasks remained	Task
	Completion capacity	Task/Month
	Rework capacity	Task/Month
	Test capacity	Task/Month
	Total capacity	Task/Month

Table 24: list of constants in the model and their dimensions

Type	Variable	Dimension
Constant	Scope- Conceptual design	Task
	Scope- Detail design	Task
	Scope- Tooling	Task
	Initial rework probability- Conceptual	Dimensionless
	Initial rework probability- Detail design	Dimensionless
	Initial rework probability- Tooling	Dimensionless
	Completion duration- Conceptual design	Month
	Completion duration- Detail design	Month
	Completion duration- Tooling	Month
	Rework duration- Conceptual design	Month
	Rework duration- Detail design	Month
	Rework duration- Tooling	Month
	Test duration- Conceptual design	Month
	Test duration- Detail design	Month
	Test duration- Tooling	Month
	Phase deadline- Conceptual design	Month
	Phase deadline- Detail design	Month
	Phase deadline- Tooling	Month
	Available resources- Conceptual design	Person
	Available resources- Detail design	Person
	Available resources- Tooling	Person
	F1	Dimensionless
	F2	Dimensionless
	F3	Dimensionless
	FC	Dimensionless
	PC	Dimensionless
	PD	Dimensionless
	PG	Dimensionless
	PI	Dimensionless
	PS	Dimensionless
	Unit cost	$/(Person*Month)
	Resource productivity	Task/(Person*Month)

5.4.3 The structure assessment of the model

In system dynamics models, in addition to the ability of output behaviour to replicate the real-world behaviour, the internal structure which shapes the model behaviour is also important to be validated (Barlas, 1996; Qudrat-Ullah and Seong, 2010). The system dynamics model built for this research comprises of a series of sectors. Some parts of the model, such as the rework cycle in the workflow sector, or the structure used in the performance measurement sector are structurally valid because they are built based on previously tested models, especially models developed by Ford and Sterman (1998), Ford and Sterman (2003), Ford and Sobek (2005), and Jalili and Ford (2016), and literature about LPD, concurrent engineering and set-based design. Validating of that structures of other parts of the model could be done by tests which are run based on simple scenarios and using dummy data for the model constants. The values of dummy data used for the structural validation of the model are provided in Table 25, and tests are explained in following Sections.

Table 25: dummy data used for the structural assessment test of the model

Type	Variable	Value	Dimension
Constant	Scope- Conceptual design	100	Task
	Scope- Detail design	100	Task
	Scope- Tooling	100	Task
	Initial rework probability- Conceptual	0.4	Dimensionless
	Initial rework probability- Detail design	0.4	Dimensionless
	Initial rework probability- Tooling	0.4	Dimensionless
	Completion duration- Conceptual design	0.2	Month
	Completion duration- Detail design	0.2	Month
	Completion duration- Tooling	0.2	Month
	Rework duration- Conceptual design	0.1	Month
	Rework duration- Detail design	0.1	Month
	Rework duration- Tooling	0.1	Month
	Test duration- Conceptual design	0.2	Month
	Test duration- Detail design	0.2	Month
	Test duration- Tooling	0.2	Month
	Reduction ratio	0.2	Dimensionless
	Phase start time- Conceptual design	0	
	Phase start time- Detail design	40	Month
	Phase start time - Tooling	80	Month
	Available resources- Conceptual design	40	Person
	Available resources- Detail design	80	Person
	Available resources- Tooling	120	Person
	F1	0.8	Dimensionless
	F2	0	Dimensionless
	F3	0.1	Dimensionless
	FC	1	Dimensionless

	PC	0.5	Dimensionless
	PD	2	Dimensionless
	PG	2	Dimensionless
	PI	2	Dimensionless
	PS	-1	Dimensionless
	Target confidence	1	Dimensionless

5.4.3.1 The Workflow Sector

As discussed in Section 4.6.3.1, the *Workflow Sector* is the main building block of the model which captures the effects of some fundamental elements of LPD, including the number of concepts, the duration of the conceptual design phase and the convergence period, as elements related to the SBCE approach, the concurrency between phases, and the effect of schedule pressure on the deviation of the NPD project from the SBCE approach towards the PBCE approach. These elements are validated separately in following Sections.

5.4.3.1.1 The effect of the number of concepts

To examine the structural soundness of the model for capturing the effect of the number of concepts, in the first scenario, the model is simulated assuming that the conceptual design phase starts with only one concept. As the goal is to examine the effect of the number of concepts, the starting time and finish time of phases are determined in a way that there is no concurrency between consecutive phases in the model, and each phase starts after its upstream phase finishes its activities. The behaviour of stocks in the workflow sector of the conceptual design phase of the model, as shown in Figure 65, is similar to what have been described in the previous system dynamics model of NPD projects (e.g. Ford and Sterman, 1998; Ford and Sterman, 2003). Having only one concept reduces the duration of the conceptual design phase, so the time period in Figure 65 is five months to capture all the details in curves. All tasks first reside in the stock of '*Tasks Not Completed*' and during the simulation flow to the stock of '*Tasks released*', while temporarily residing in stocks of 'Tasks Pending Test', 'Tasks Pending Rework', and 'Tasks Pending Decision'. As in Figure 65, it takes around 2 months for all tasks to complete the process and reside in stock of '*Tasks released*'.

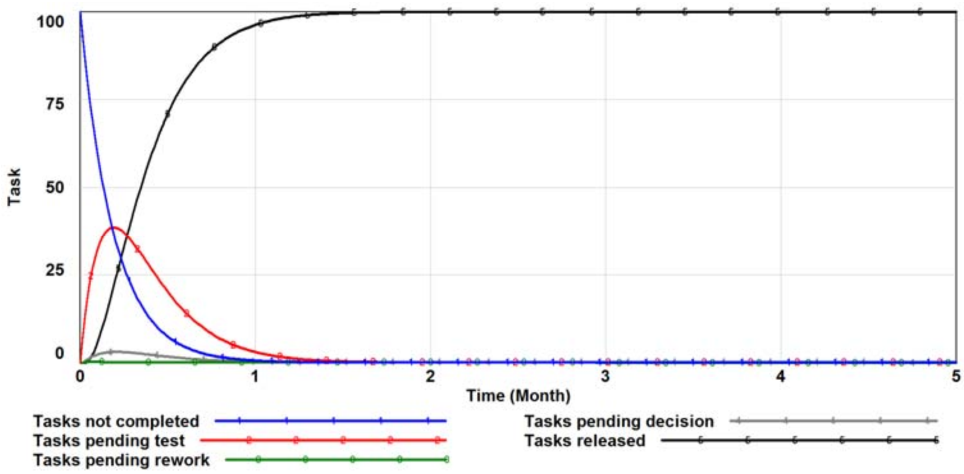


Figure 65: the behaviour of stocks in the workflow sector of conceptual design phase

Because there is just one concept in the process, the '*Iteration Switch'* remains off, and no task flows back to the stock of '*Tasks Not Completed*', and the conceptual design phase terminates with one iteration cycle. As mentioned before, each iteration cycle increases the confidence and reduces the level of uncertainty in available concepts, according to Equation 17, to close the confidence gap. However, as there is just one iteration cycle, the level of confidence in the concept does not reach to the level of target confidence which is set to be one for this model, so the final concept has a high level of uncertainty when is passed to the next phase. Changes in the level of concept confidence and the uncertainty level during the conceptual design phase is shown in Figure 66 and Figure 67.

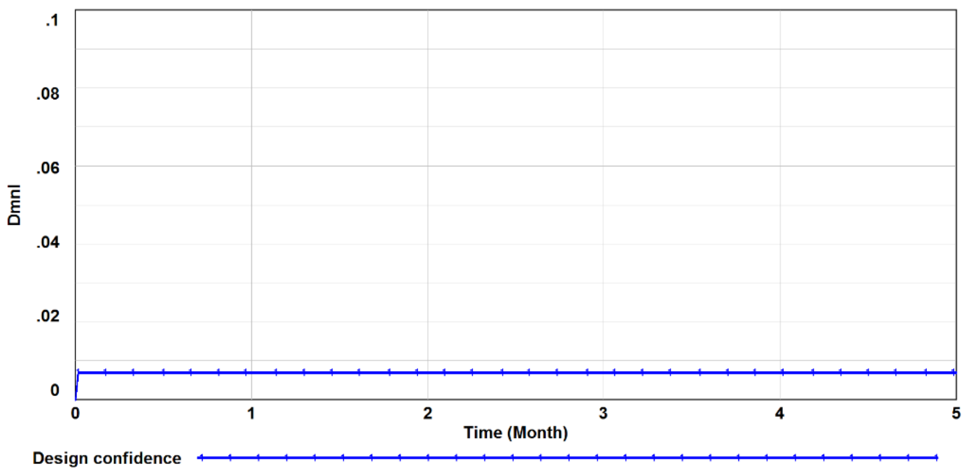


Figure 66: changes in the level of confidence in the conceptual design phase for the simulation with one concept

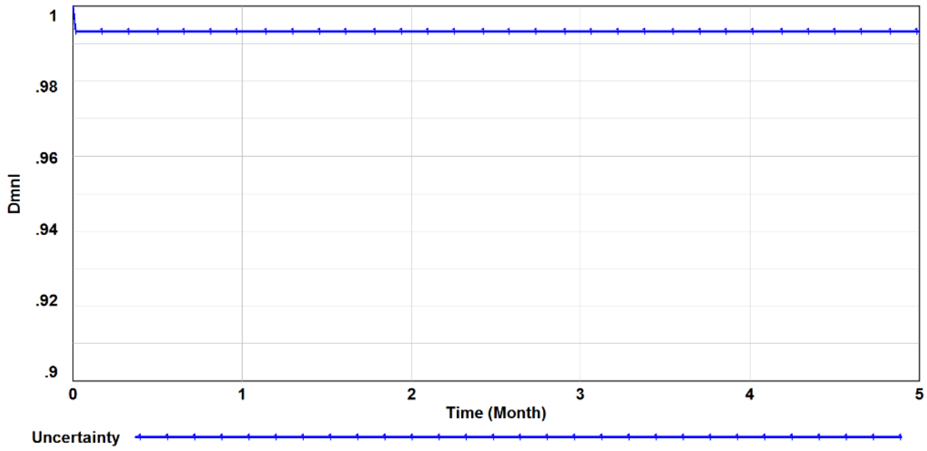


Figure 67: changes in the level of uncertainty in the conceptual design phase for the simulation with one concept

The detail design phase starts after the final concept is released from the conceptual design phase. Figure 68 shows the behaviour of stocks in the workflow sector during the detail design phase for a time period of 50 months. In this Figure, only '*Tasks Not Completed*' and '*Tasks Released*' stocks are shown and all intermediate stocks are removed for more clarity. Each pulse in Figure 68 represents an iteration cycle during the detail design phase which reduces the level of uncertainty in design according to Equations 17 and 18. The final design is released from this phase only after the confidence gap is closed and the '*Release switch*' is '*On'*. As the level of confidence in the concept arrived at this phase is low and the rate of increase in the design confidence in the detail design phase is directly related to the concept confidence (Equation 18), the confidence gap is not closed even after passing 50 months, so the detail design phase never finishes. Figure 69 shows the level of design confidence after running the simulation model for 50 months, and its distance to reach to one as the set value for the level of the target confidence.

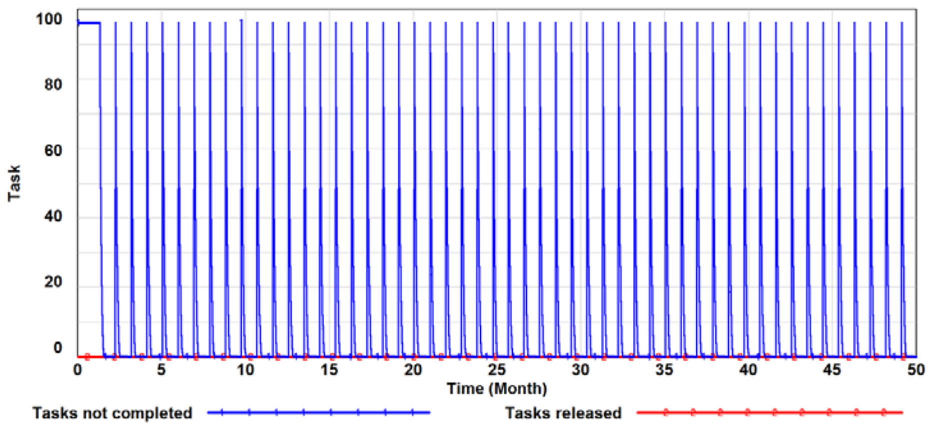


Figure 68: iteration cycles during the detail design phase

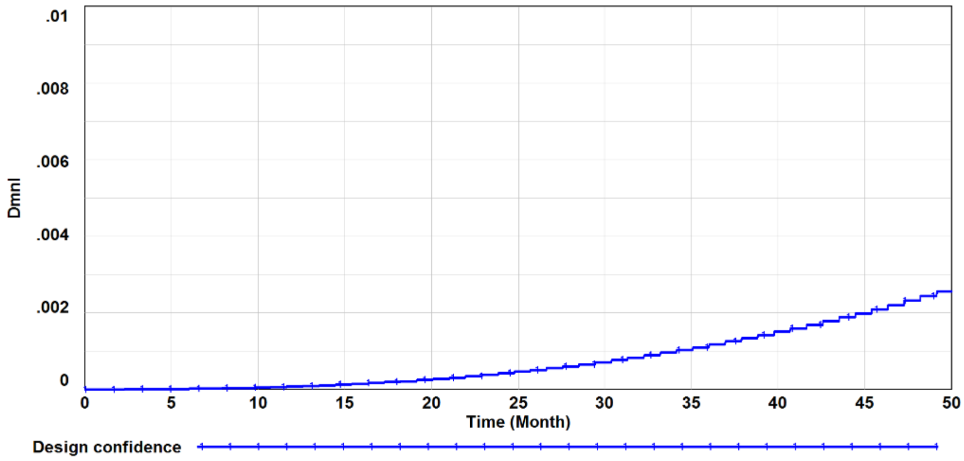
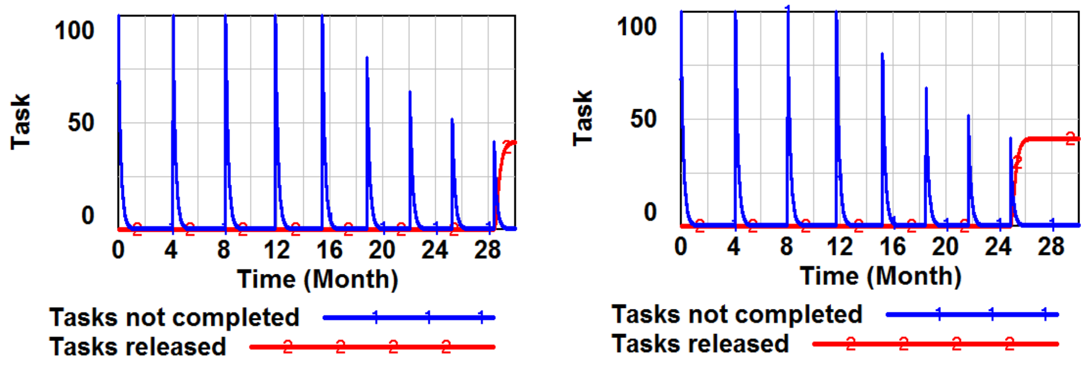


Figure 69: change in the level of design confidence in the detail design phase after 50 months

In the second and third scenarios, the number of initial concepts are increased to two and three, respectively, while other parameters remain unchanged and the simulation model is run for another 50 months. Figure 70 shows the workflow stocks in the conceptual design phase. Having two and three concepts in the conceptual design phase activates the *'Iteration Switch*', and results the process to continue until the confidence gap is closed. As shown in Figure 70, before tasks flow to the stock of '*Tasks Released*' at around month 28 for scenario 2, and month 25 for scenario 3, nine and eight iteration cycles occur, respectively. Figure 71 shows the stocks behaviours in two initial iteration cycles in scenario two where each iteration shows a similar behaviour to Figure 65. The rate of increase in the concept confidence in this scenario is higher compared to the scenario 1 due to having more concepts. In addition, shorter duration of the conceptual design phase in the scenario 3 compared to the scenario 2 explains the argument by Malak et al. (2009) about the effect of narrow partitioning of the design space on the duration of the conceptual design phase. In Figure 72, each step in the confidence graphs shows completing an iteration cycle in the process. Due to these iteration cycles, the uncertainty level reduces to zero before the final concept is passed to the detail design phase.



(a) scenario 2 *(b) scenario 3*

Figure 70: the stocks behaviour in the conceptual design phase

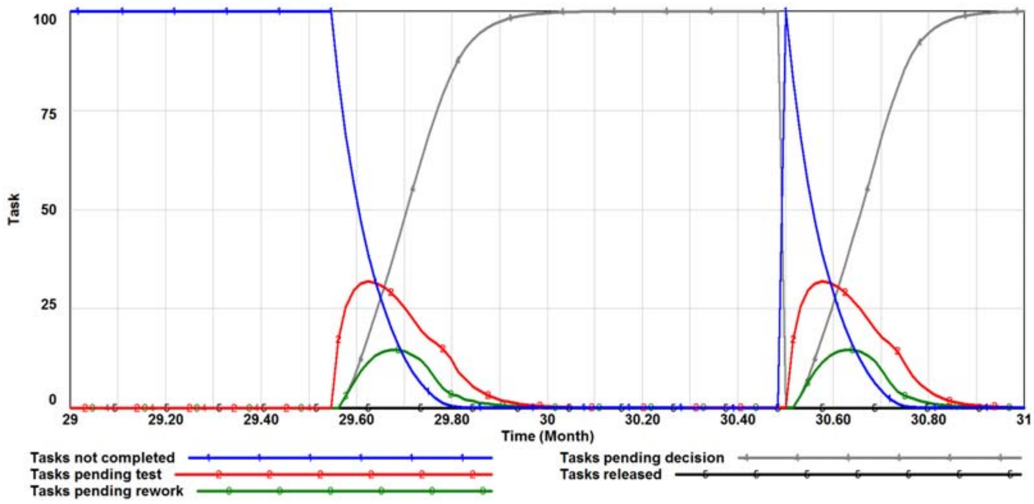
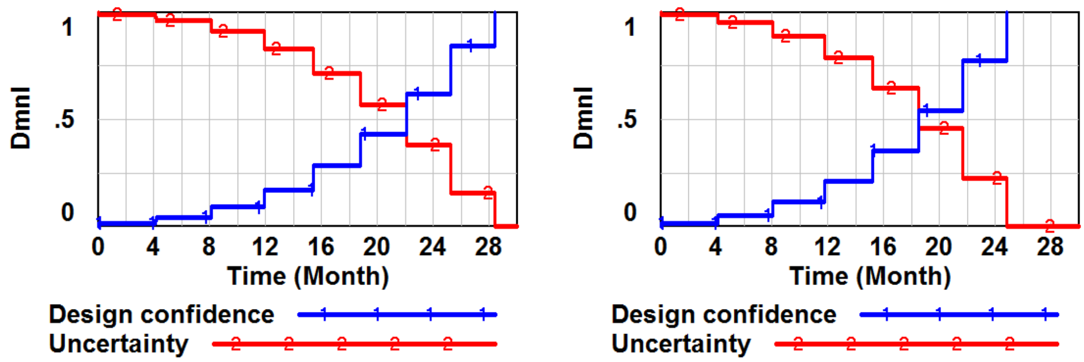


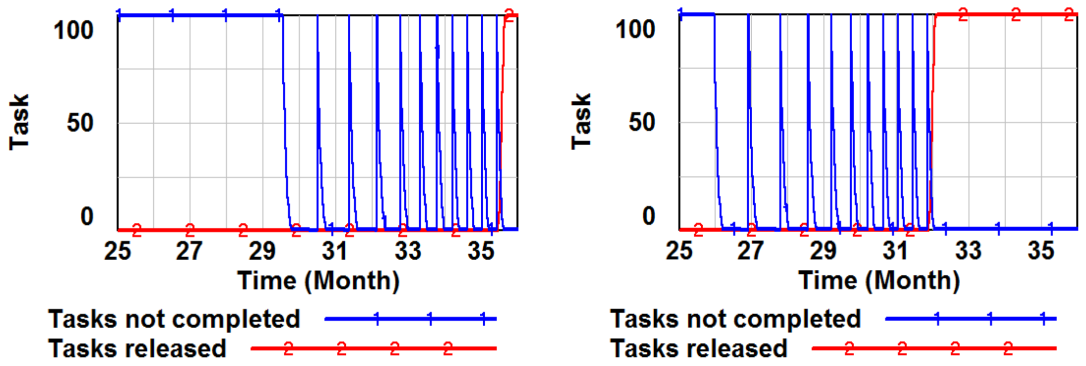
Figure 71: the stocks behaviour in two initial iteration cycles in scenario 2



(a) scenario 2 *(b) scenario 3*

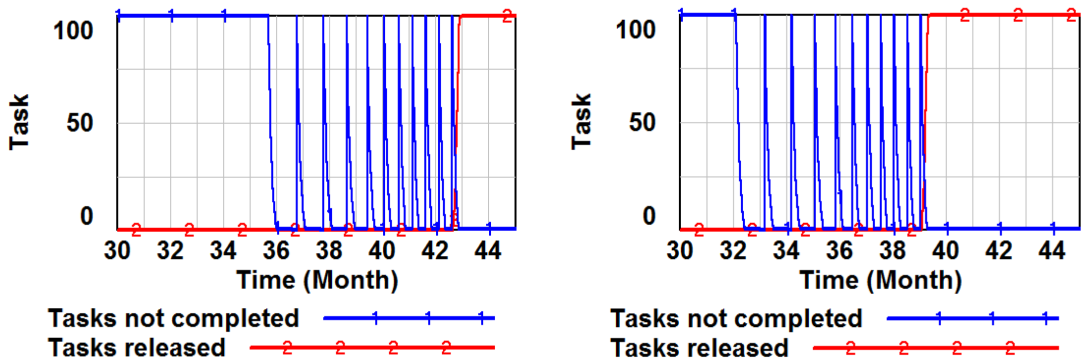
Figure 72: changes in the confidence and uncertainty level in conceptual design phase

As mentioned before, there is no overlap between phases, so the detail design phase starts after all activities in the conceptual design phase are completed. Compared with the scenario 1 in which the process never finishes due to the high level of uncertainty in the concept, the detail design phase in scenarios 2 and 3 terminate after several iteration cycles. As in both scenarios the level of concept confidence arrived at the detail design phase, and other variables affecting the rate of confidence increase in this phase according to Equation 18 are similar, the duration of the detail design phase in both scenarios are the same. However, in scenario 2, the detail design phase starts and ends sooner, compared to the scenario 3, due to the difference between durations of the conceptual design phase. After finishing the detail design phase, the tooling phase starts as shown in Figure 74. The behaviour of stocks in this phase for both scenarios is similar to the detail design phase, because all parameters are kept unchanged during the process. The NPD project finishes at the end of the tooling phase. As shown in Figure 74, in the scenario 2 the project duration is about 43 months, while in the scenario 3 where the number of initial concepts are higher, the project duration is about 39 months.



(a) scenario 2 *(b) scenario 3*

Figure 73: the behaviour of stocks in the detail design phase

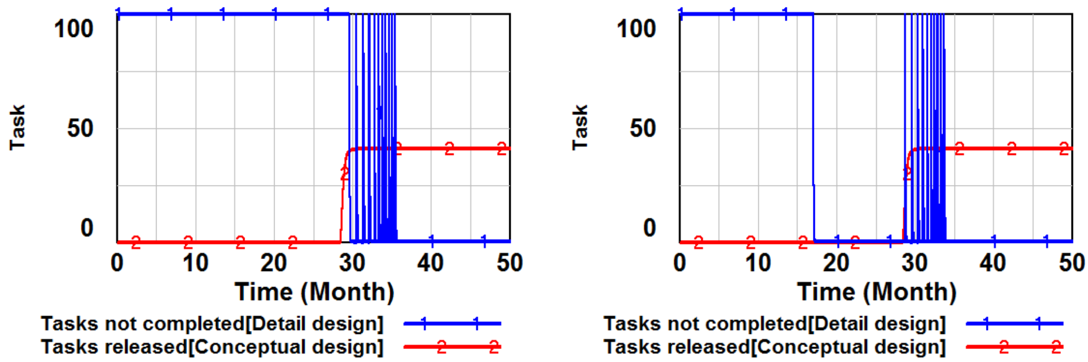


(a) scenario 2 *(b) scenario 3*

Figure 74: the behaviour of stocks in the tooling phase

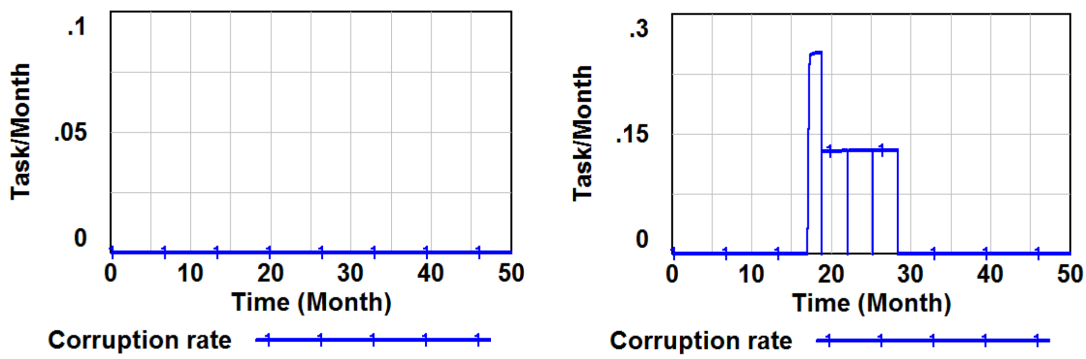
5.4.3.1.2 The effect of concurrency

The workflow sector, as described in Section 4.6.3.1.3 should also capture the effect of concurrency in the model. As mentioned in the previous Section, the effect of the number of concepts is controlled by running the model without any overlap between phases. To check the ability of the model to show the concurrency effect, the model is run again using the scenario 2, after the level of overlap between the conceptual design and the detail design phase increases by changing the '*Phase Start Time*' for the detail design phase, while other parameters are kept unchanged. Figure 75 shows the behaviors in the stock of '*Tasks released*' in the conceptual design phase, and the stock of '*Tasks not completed*' in the detail design phase, for sequential and concurrent runs. The differences between two runs are clear where in the concurrent run, before releasing tasks in the conceptual design phase, activities in the detail design phase start. Figure 76 shows the effect of concurrency on the corruption rate in the concurrent run of the scenario 2, while in the sequential run the corruption rate is zero. Due to the effect of corruption, in the sequential run, the duration of the detail design phase is about six months (according to Figure 75-a it starts at month 30 and finishes at month 36), while in the concurrent run, the duration of detail design phase is about 17 months (according to Figure 75-b it starts at month 17 and finishes at month 34). Nevertheless, compared to the sequential run, the detail design phase in the concurrent run starts sooner due to concurrency and finishes sooner, although its overall duration is longer due to the corruption effect.



(a) scenario 2- sequential *(b) scenario 3- concurrent*

Figure 75: the effect of concurrency on the behaviour of stocks

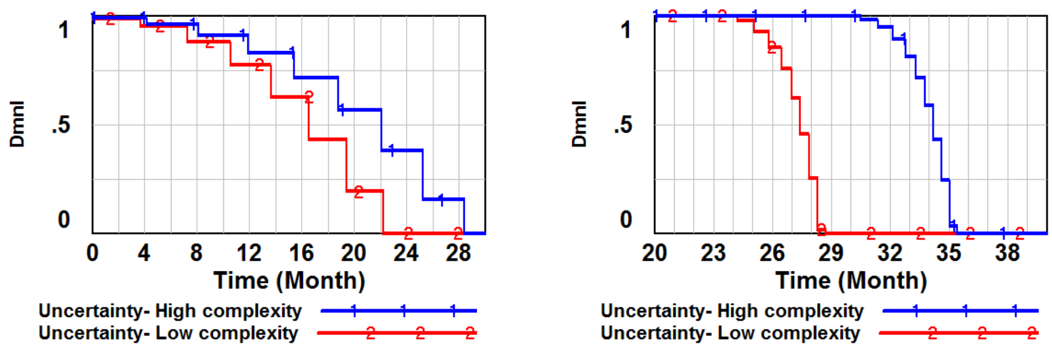


(a) scenario 2- sequential *(b) scenario 2- concurrent*

Figure 76: the effect of concurrency on the corruption rate

5.4.3.1.3 The effect of project complexity

As shown in Equations 17 and 18, the project complexity shows its effect on the rate of increase in the design confidence in all phases of the project. Using the explanations in Section 4.5.2.4, four levels of the project complexity are defined in the model. The highest level of project complexity is assigned with the value four, while other levels are three, two, and one, respectively. In all previous simulation runs the project complexity is set on one (highest level). To check the effect of project complexity in the model, the scenario 2 is simulated again while the project complexity is reduced to two, and other parameters are kept unchanged. Figure 77 shows the effect of the project complexity on the changes in the level of uncertainty during the conceptual design and the detail design phases of the model. In both phases the level of uncertainty drops sooner when the project complexity is low. Figure 78 shows the total number of iterations throughout the project for two different levels of project complexity. Because the level of uncertainty drops sooner when the project complexity is low, it reduces the number of iterations needed to close the confidence gap in each phase, thus decreasing the total number of iterations.



(a) conceptual design phase *(b) detail design phase*

Figure 77: the rate of decrease in the level of uncertainty for two different project complexities

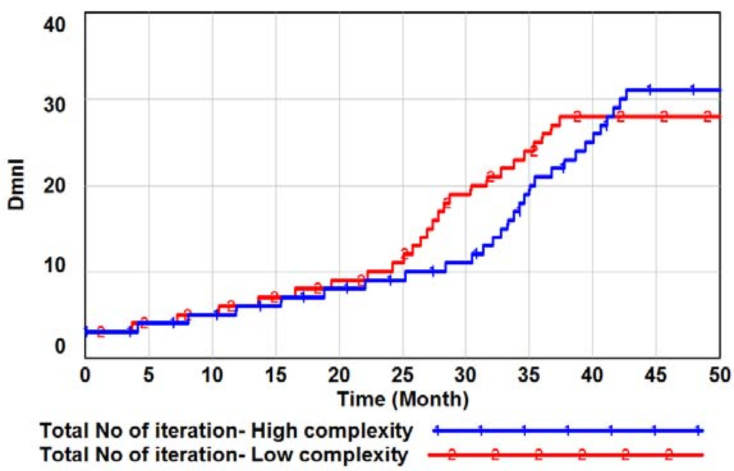


Figure 78: total number of iterations for two different project complexities

5.4.3.2 The Schedule Pressure Sector

As described in Section 4.6.3.2, increasing the level of schedule pressure during the conceptual design phase results in managerial decisions which lead the project towards a PBCE approach, through reducing the level of target confidence, and increasing the reduction ratio in the conceptual design phase (Equations 43 and 44). In the previous Section for checking the workflow sector of the model, the phase deadlines are selected in a way that '*Time remained*' in the conceptual design phase is always greater than '*Time required*', so according to Equation 38, the schedule pressure is always less than one. In this situation, according to Figure 70-a, the duration of the conceptual design phase for scenario 2 is about 29 months. To check the effect of the schedule pressure, the phase deadline for the scenario 2 is set to 20 months and '*F2'* is set to one, while other parameters are kept unchanged, and the simulation model is run. '*F2'* plays as a switch for the effect of schedule pressure on the flows of '*Target Change Rate*' and '*Reduction Increase Rate*'. While the value of '*F2'* is equal to zero the schedule pressure does not have any effect on mentioned flows, so the project follows a SBCE approach, but changing the value of 'F2' to one results in change in mentioned flows under the effect of the schedule pressure, so the PBCE approach becomes dominant. Figure 80 shows the drop in the target confidence and the rise in the reduction ratio as the result of increasing the schedule pressure during the conceptual design phase. As the result, according to Figure 80, the level of uncertainty in the conceptual design phase reaches sooner to zero, and the phase finishes with lower number of iterations. Finally, Figure 81 shows that by adopting the PBCE approach the level of schedule pressure drops sooner, compared to the SBCE approach.

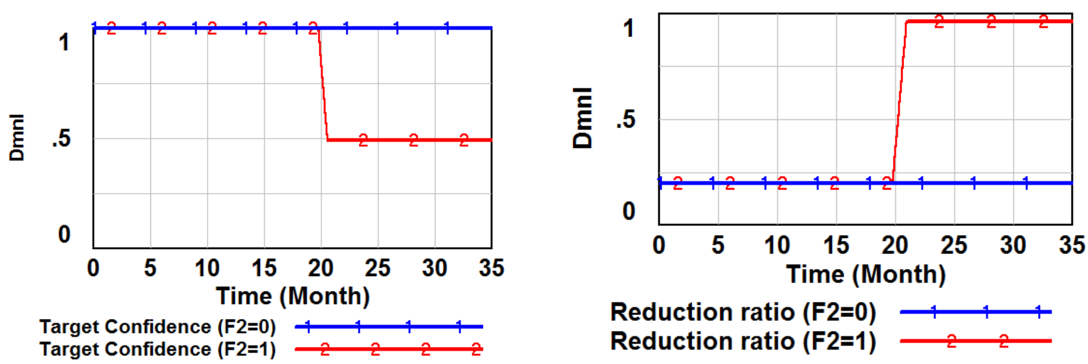


Figure 79: the effect of the schedule pressure on the target confidence and the reduction ratio

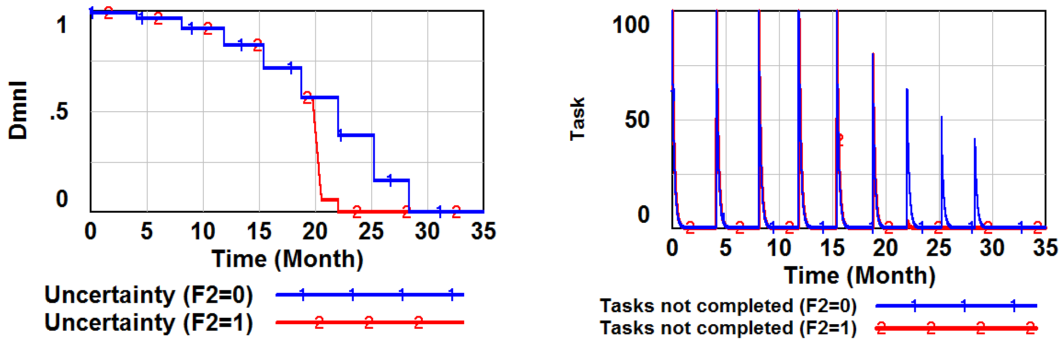


Figure 80: changes in the level of uncertainty and the number of iterations in the conceptual design phase by adopting the PBCE approach

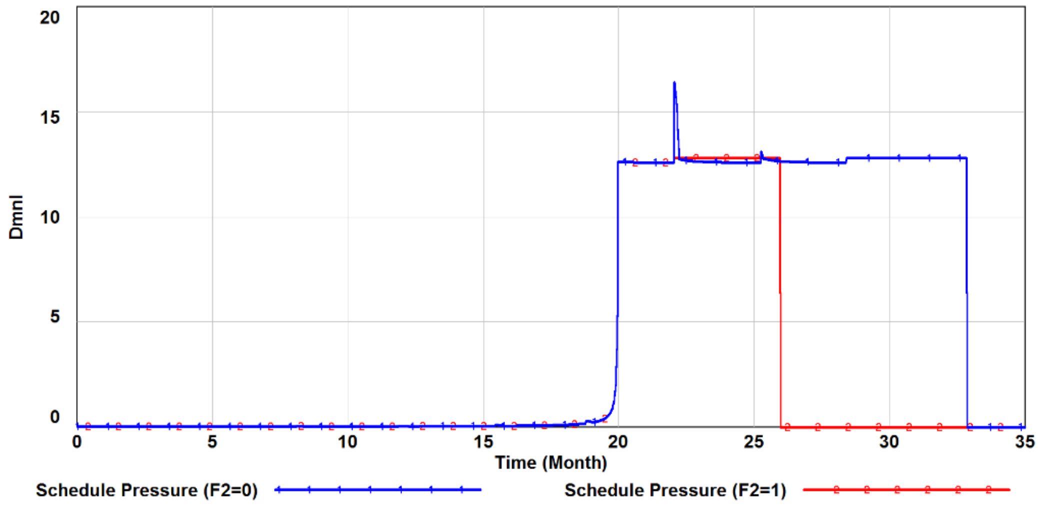


Figure 81: drop in the level of schedule pressure by adopting the PBCE approach

5.4.3.3 The Resource Management Sector

As mentioned in the previous Chapter, in the model developed for this research, the resource allocation is not considered as a factor affecting the performance of the NPD project. It is assumed that there are enough number of resources to be allocated to activities in the project and their productivity is constant throughout the project. The goal for including the resource management sector in the model is to calculate the project cost as one of the performance metrics, and to evaluate the effect of different policies on the overall cost. The model calculates the amount of effort devoted to different phases of the project as the product of the number of allocated resources and the duration of each phase. Figure 82 shows the result for the scenario 2. Using this information, the total effort devoted to a project could be calculated as shown in Figure 83. According to this figure, the total amount of effort in a project with low level of complexity is lower than the project with high level of complexity, which is an expected result

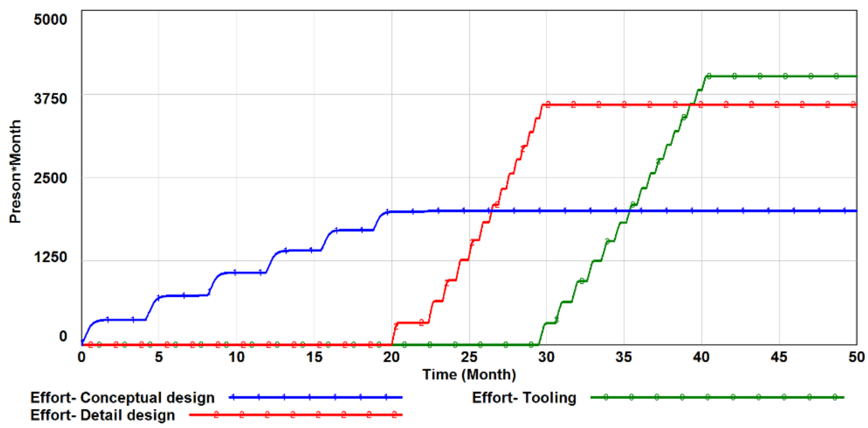


Figure 82: the effort devoted to different phases of the project- Scenario 2

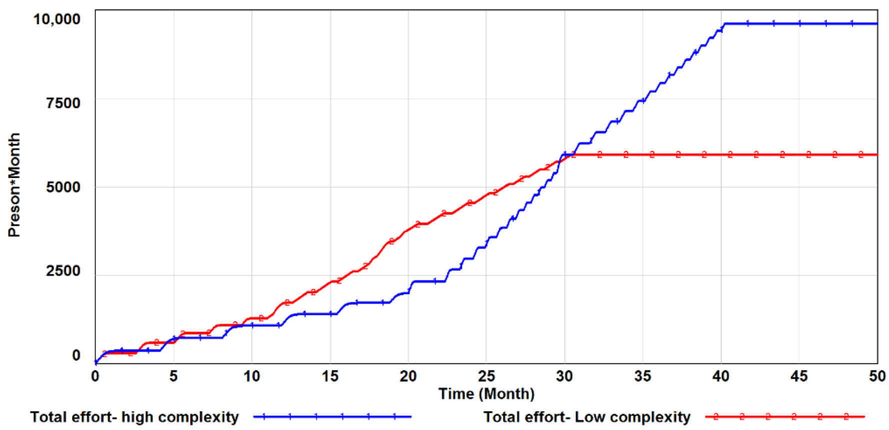


Figure 83: comparison between the total efforts devoted to projects with different levels of complexity- Scenario 2

5.4.4 Extreme Conditions Analysis

This test evaluates the model robustness under extreme conditions by assigning extreme values to selected constants in the model and comparing the model behaviour with the anticipated behaviour. For this test, the model run under scenario 2, as described in Section 5.4.3.1.1 is selected as the base case. From the list of constants mentioned in Table *25*, scope, initial rework probability, and completion duration in all phases are selected for this test. The values of scope and completion duration are increased 10 times, and the values of initial rework probability are increased to 0.99, and in each case, the model is run, separately, while all other parameters are kept unchanged. The duration of the project is selected as the model output to be compared. As expected, and shown in Figure 84, except the base case, in other cases the project could not be finished in a 100-month time period.

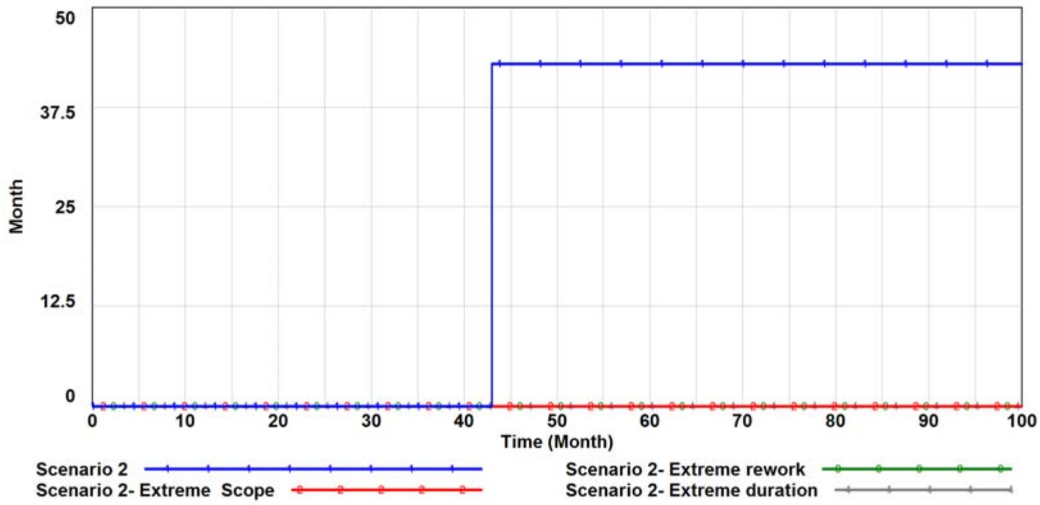


Figure 84: results of extreme condition test

5.5 Behaviour Reproduction Test

The goal of this test is to show if the model could reproduce the behaviour pattern displayed by the real system, to reveal flaws in the structure and parameters, and to assess if the model is fit for its purpose. Historical data could be used to determine the value of constants and to validate the model by comparing the model outcomes with real-world data. However, as Radzicki (2009) stated to raise the confidence in the model outcomes, it is not necessary to tightly fit models to time series data. The reason is that a particular time path that a project follows through its progress, is just one of an infinite number of paths that it could take. So, it is more important for a model to mimic the basic character of the data, rather than fit it point-by-point. In addition, the system dynamics model typically provides a set of policies which improve system performance and increase system robustness. These policies are usually feedback-based rules that do not require the accurate point predictions of system variables (Radzicki, 2009).

To pass the behaviour reproduction test, it is needed to calibrate the model using time series project data. The goal of calibration is to estimate the values of model constants in a way that allows the model to generate behaviours which fit the real-world data. Calibration of the model is done through a numerical optimization process to minimize the distance between the model outcomes and the actual data by searching for the best model parameters (Parvan et al., 2015). The optimization incudes exploring a vector of parameters to find a particular combination of parameters which offers the best fit between the chosen variables in the model and a past time series dataset of this variable.

5.5.1 Data used for calibration

For calibration of the model, secondary historical data related to four types of projects from a major car manufacturing company are used. Figure 85 shows a simple version of the NPD process design in the company including the conceptual design phase, detail design phase, and tooling phase, the milestones of these phases of the project, and the flow of the design in the project. Table *26* represents sets of data related to the projects and Figure 86 represents the rework curves in these projects. Each project is selected from one level of the project complexity, as described below and also in section 5.4.3.1.3.

- Project A: incremental product improvement projects which have the lowest degree of novelty and because of fast technology change cycles and better-informed customers are less feasible for companies.
- Project B: derivative products which are built on existing product platforms and are very common in industries;
- Project C: product platform-development projects which have fundamentally new systems and components and utilize improved versions of existing products;
- Project D: radically new projects which represents breakthroughs in products or technology.

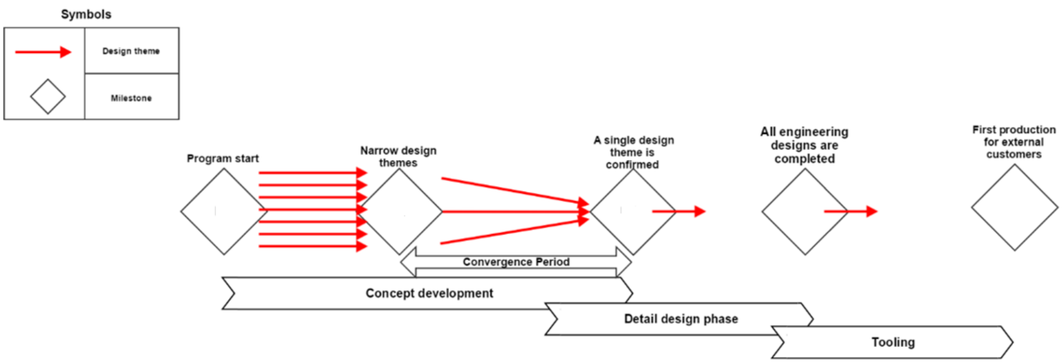


Figure 85: the NPD process design

Table 26: Time and cost project data used for calibration of the model

Project	**Project complexity**	**Conceptual design phase**		**Detail design phase**		**Tooling phase**		**Project finish time (Month)**	**Project cost (M$)**
		Start	Finish	Start	Finish	Start	Finish		
A	4	0	17	7	26.5	26.5	44	44	956
B	3	0	14.5	5	24	24	40	40	595

C	2	0	11.5	4	20	20	35	35	460
D	1	0	8	2	14.5	14.5	29.5	29.5	381
Project start and finish time are reported as the number of month passed from the start of the project. Month 0 is the month project started.									

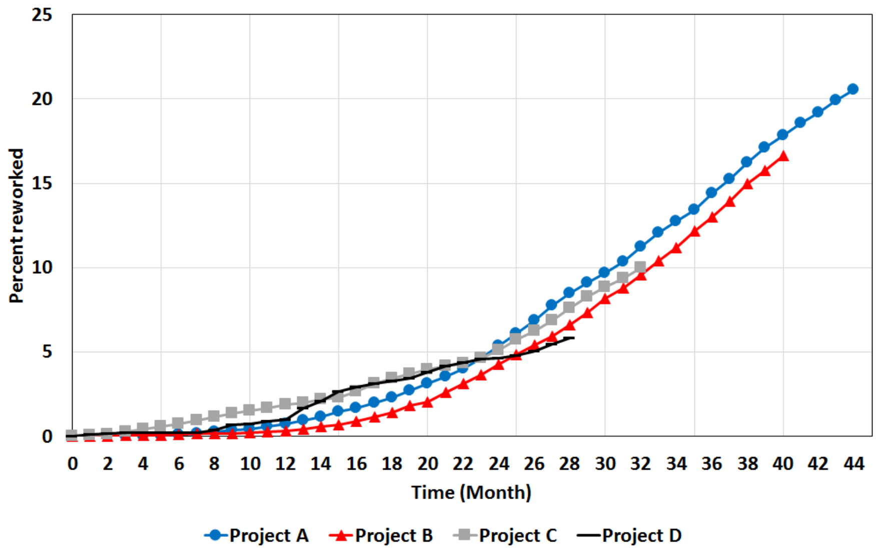


Figure 86: Data related to the fraction of tasks reworked in four projects

5.5.2 Optimization process

The developed system dynamics model could be defined as the function shown in Equation 69, in which *Y* is the model output vector, *X* is the model input vector, and *a* is the parameter vector. The input vector includes constants which their values are known, while the model parameter vector includes all unknown constants whose values should be estimated in the calibration process. The output vector includes the value of all outputs which the model produces after each run of simulation. All the model variables and actual project variables used for calibration are listed in Table 27. As shown in Table 27, the model is calibrated upon the actual data, including the actual finish time of the project and phases, actual project cost, actual rework curve.

Table 27: Variables used in the model

Output Vector	Symbol	Actual data	Symbol
Project Duration- Model	T	Project Duration- Actual	t
Project Cost- Model	CO	Project Cost- Actual	co
Phase Duration- Model	TP	Phase Duration- Actual	tp
The rework curve- Model	R	The rework curve- Actual	r

Input Vector	Symbol	Parameter Vector	Symbol
Phase deadline- Conceptual design	DC	Scope	SC
Phase deadline- Detail design	DD	Initial rework probability	PR
Phase deadline- Tooling	DT	Completion duration	CD
Project complexity	COM	F1	-
Target Confidence	TC	F2	-
Concurrency	CON	F3	-
Available resources	RES	FC	-
Resource productivity	PROD	PC	-
		PD	-
		PG	-
		PI	-
		PS	-
		Unit cost	UC

$$Y = f(X, a) \quad 69$$

$$X = [DC, DD, DT, COM, TC, CON, RES, PROD]$$

$$a = [SC, PR, CD, F1, F2, F3, FC, PC, PD, PG, PI, PS, UC]$$

$$Y = [T, CO, TCD, TDD, TT, R]$$

Calibration as an optimization problem framework is shown in Equation 70 where C is the set of projects selected for calibrating the model. *Vensim®* has an embedded optimization module which is used to calibrate the model. It requires to define a payoff function which summarizes the closeness between results of the simulation and the actual data. In addition to defining the payoff function, the model constants over which the model is optimized should be chosen. The optimization process tries to find the best values for chosen constants to make the payoff function as small as possible. The payoff function defines the distance between the model outputs and the actual project data as the sum of the squared percentage errors (*SSPE*) according to the concept of the *Least Square Method* (Parvan et al. 2015). The error is the subtraction of the model outputs from the actual data. For the developed model in this research, the payoff function is the weighted sum of four error terms including project and phases durations, project cost and cumulative rework curve according to Equation 71 and Figure 87 to Figure 90.

$$\textit{Objective function: } Minimize\ \{payoff\ function\} \quad 70$$

$$\textit{Decision function: } a_i \qquad \forall i \in C$$

Constraint:

$$F1, F2, F3, PH, DUR, UC, RD > 0$$

$0 < PC < 1$

$PI > 1$

$PS < 0$

$PD > 1$

$FC > 0$

$PG > 1$

$$Payoff\ function = \sum_i W_t(\frac{T_i - t_i}{t_i})^2 + \sum_{ij} W_{tp}(\frac{TP_{ij} - tp_{ij}}{t_{ij}})^2 + \sum_i W_{co}(\frac{CO_i - co_i}{co_i})^2 + \sum_i W_r \int_0^{ti} (\frac{R(t)_i - r(t)_i}{r(t)_i})^2 dt \qquad 71$$

The equation linearly add the weighted error terms. W_t, W_{tp}, W_{CO} and W_r are the values of error weights which show the relative importance of different terms in the payoff function. Other variables in the payoff function are introduced in Table *27*.

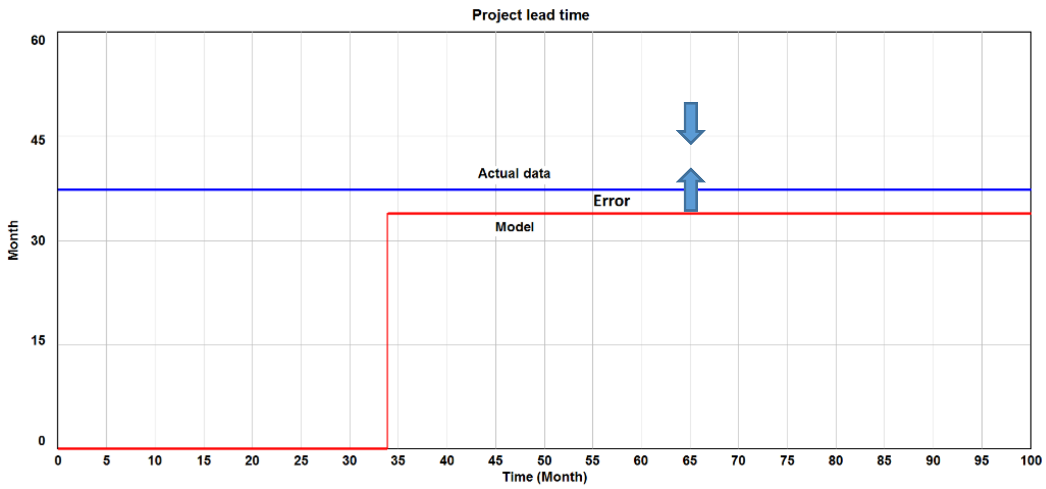


Figure 87: calibration payoff function project time term

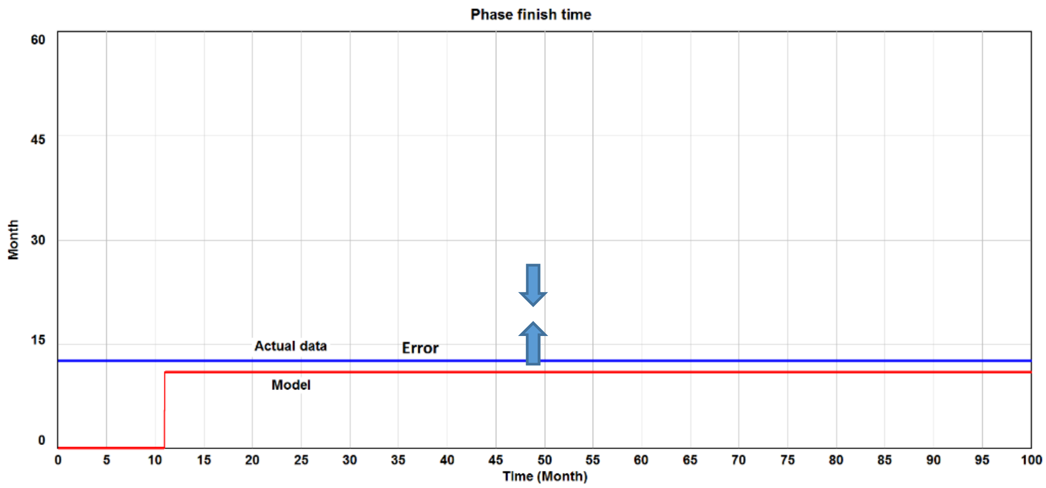


Figure 88: calibration payoff function phase time term

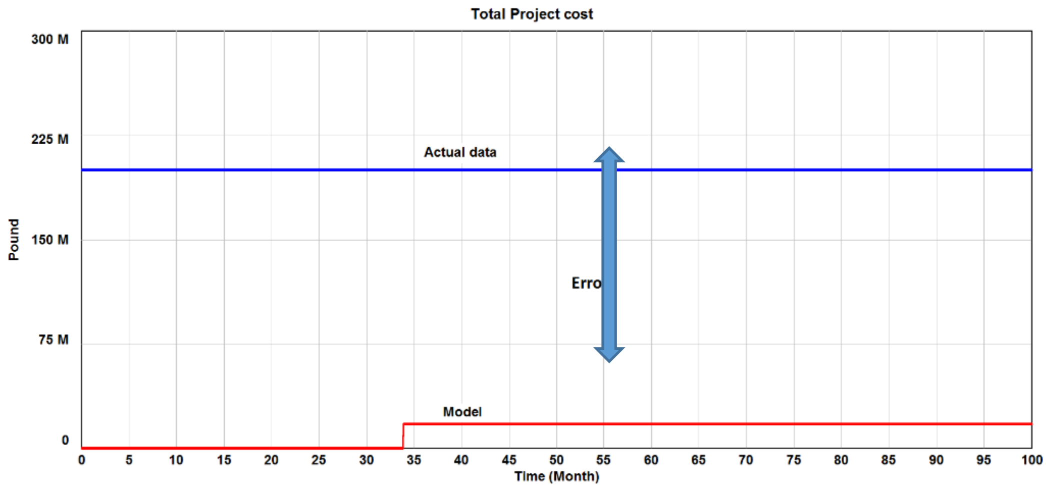


Figure 89: calibration payoff function cost term

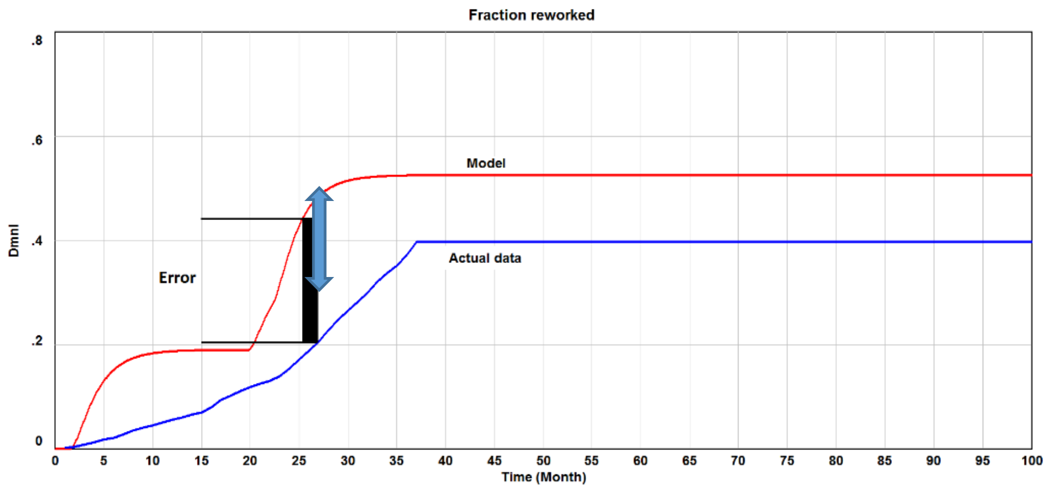


Figure 90: calibration payoff function rework term

The first term in the payoff function is the sum of the squared percentage error of the project duration (*SSEP[T]*) defined in Equation 72, where *T* is the model finish time, *t* is the actual finish time, and *i* is the number of projects used for calibration. The second term is the sum of the squared percentage error of the phases finish time (*SSEP[T_P]*), as in Equation 73, where *TPij* is the model finish time for the phase *j* in project *i* and *tpij* is the actual finish time of phase *j* in project *i*. The third term is the sum of the squared percentage error of project cost (*SSEP[CO]*) defined in Equation 74, where *CO* is the model cost and *co* is the actual cost. The last term is the sum of the square error of the cumulative fraction of reworks (*SSEP[R]*) defined in Equation 75, where *R* is the model fraction reworked, and *r* is the actual fraction reworked.

$$SSEP[T] = \sum_i (\frac{T_i - t_i}{t_i})^2 \quad 72$$

$$SSEP[T_P] = \sum_{ij}(\frac{TP_{ij}-tp_{ij}}{t_{ij}})^2 \tag{73}$$

$$SSEP[CO] = \sum_{i}(\frac{CO_i-co_i}{co_i})^2 \tag{74}$$

$$SSEP[R] = \sum_{i}\int_0^{T_i}(\frac{R(t)_i-r(t)_i}{r(t)_i})^2 dt \tag{75}$$

As all selected projects are independent from each other, the model could be calibrated individually for each project, to provide an estimation for the unknown constants. However, unknown constants could be divided into two groups of project-dependent parameters which change from one project to other, and project-independent parameters which are common across all different types of projects (Table *28*). To maximize the statistical power in estimating the project-independent constants, projects are linked together through these constants to simultaneously estimate them for all projects, instead of one-by-one estimation. To be able to do this, a combined model of four projects is built, as shown in Figure 91. Layers of the model have the project-independent constants shared among them.

Table 28: classifying the model constants based on their dependency on the type of project

Project-dependent constants	**Project-independent constants**
Scope	F1
Initial rework probability	F2
Completion duration	F3
	FC
	PC
	PD
	PG
	PI
	PS
	Unit cost

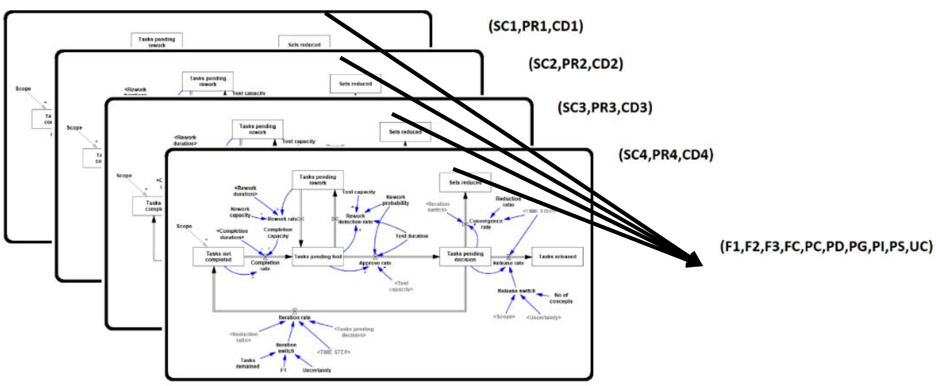


Figure 91: the combined model

According to Parvan et al. (2015), the calibration of the model is conducted in three steps. In the first step, a neighbourhood for constants is found using built-in optimization algorithm of Vensim®, by implementing a course Time-Step and optimization settings. In the second step, first, using the values from the previous step, the project-independent constants are fixed and the model is optimized individually for each project to find the values of project-dependent constants. Then, the project-dependent constants are fixed and the combined model is optimized in the search of values of project-independent constants. This step is repeated several times using several combinations of error weight values to reach to the least payoff value. The error weight values (W_t, W_{tp}, W_{CO} and W_r) are assigned subjectively to reach to the lowest possible value for the payoff function, as shown in Table 29. Finally, in the third step, all constants in a single optimization process are fine-tuned with higher-resolution time-step and optimization settings.

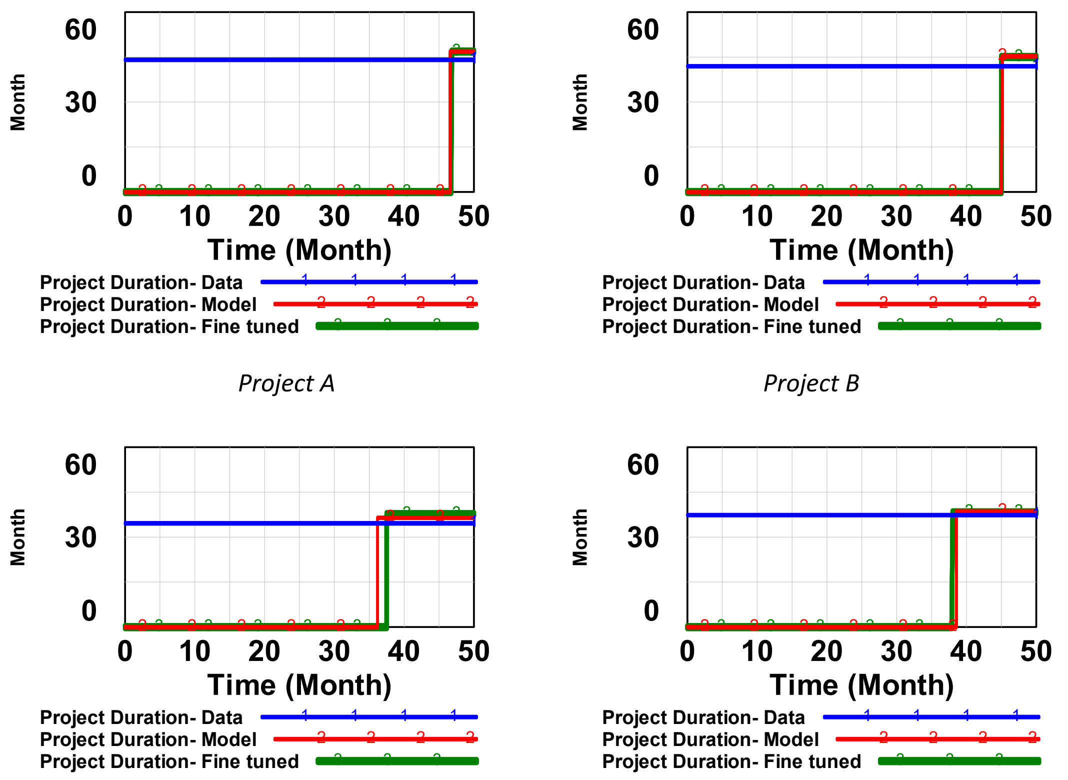
Table 29: different combinations of weight values used for calibration

	Calibration -1	**Calibration -2**	**Calibration -3**	**Calibration -4**	**Calibration -5**	**Fine tune**
Project cost weight (W_{CO})	0.5	1	0.5	0.4	0.5	0.5
Project rework weight (W_r)	2.5	1	1	0.6	0.25	0.25
Project lead time weight (W_t)	5	1	0.5	0.4	0.5	0.5
Project finish time weight (W_{tp})	2.5	1	0.25	0.2	0.25	0.25
No of simulations	6841	12565	12106	13033	9678	16449
Payoff	0.080279	0.031132	0.026508	0.016227	0.013137	0.010047

Two groups of model constants and their estimated values, as the results of the model calibration process, are shown in Table 30. Figure 92 to Figure 97 show the fit between the calibrated and the actual curves for the performance measures in all projects. The payoff values for different calibration scenarios are shown in Figure 98.

Table 30: model constants and their estimated values

	Parameters	Phases	Project A	Project B	Project C	Project D
Project-dependent parameters	SC	Conceptual design	100	50	50	50
		Detain design	96.0959	83.27	83.26	81.2057
		Tooling	149.999	107.9	106	99.766
	PR	Conceptual design	0.432113	0.43	0.2997	0.289241
		Detain design	0.25255	0.25	0.1	0.099964
		Tooling	0.610349	0.55	0.3995	0.389761
	CD	Conceptual design	0.2	0.1794	0.1656	0.127973
		Detain design	0.0247927	0.02479	0.02478	0.0247
		Tooling	0.026279	0.02989	0.02989	0.0297353
Project-independent parameters	FC	16.1302				
	PC	0.556888995				
	PD	2				
	PG	4.89721				
	PI	1.5				
	PS	-3.12565				
	F1	0.001				
	F2	0.802811				
	F3	3.5				
	Unit cost	53572.6				



Project A

Project B

Project C

Project D

Figure 92: Best fit results for project duration

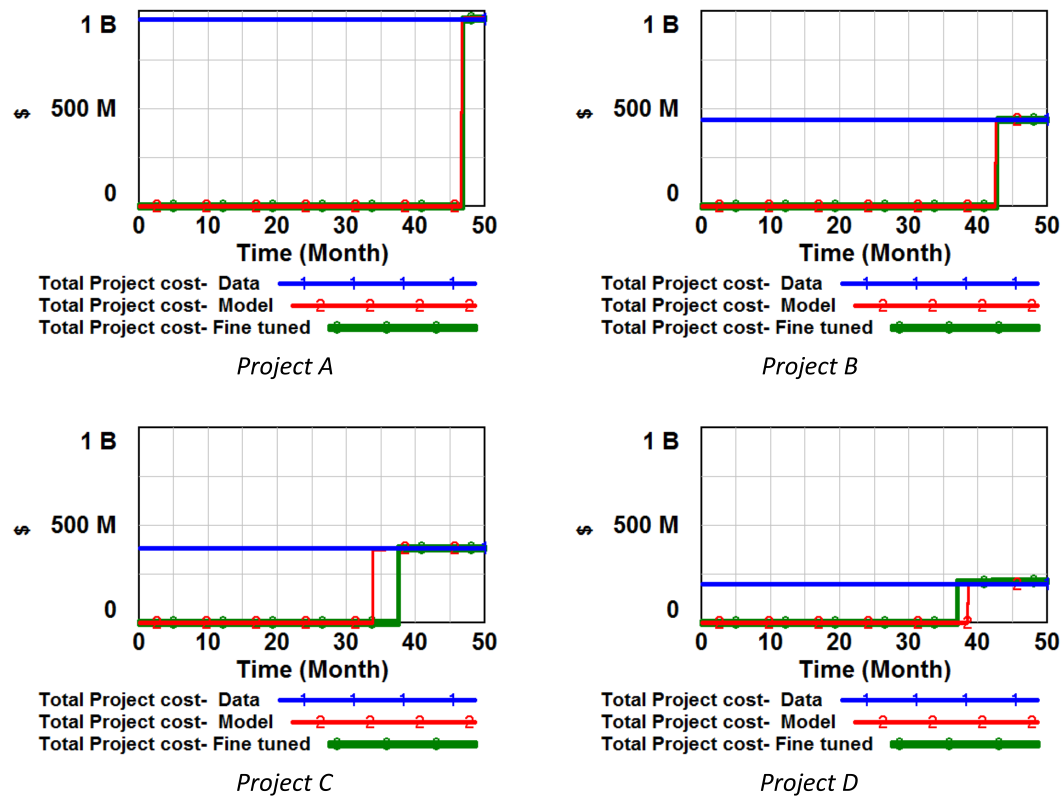


Figure 93: Best fit results for project cost

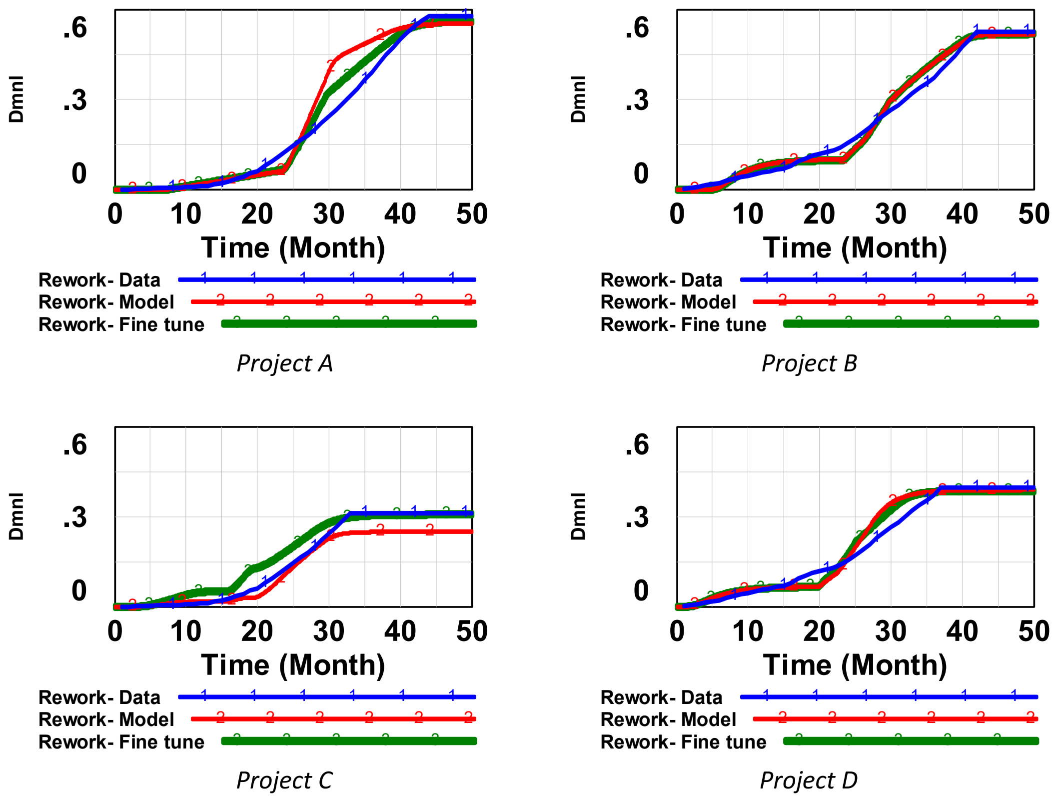
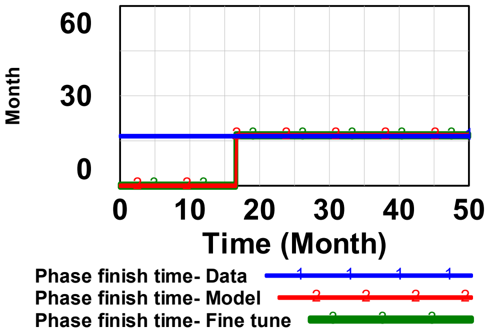
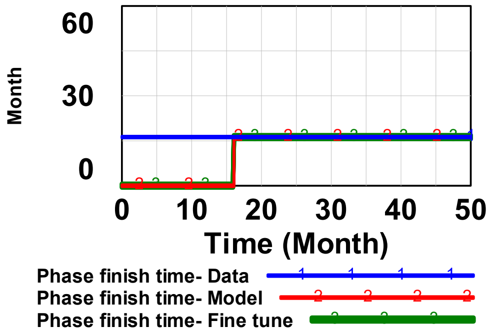


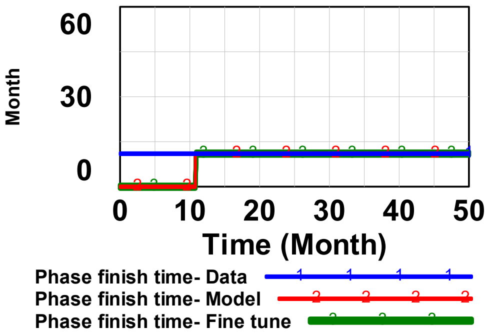
Figure 94: Best fit results for rework



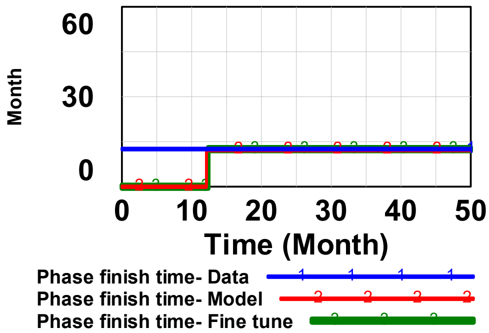
Project A



Project B

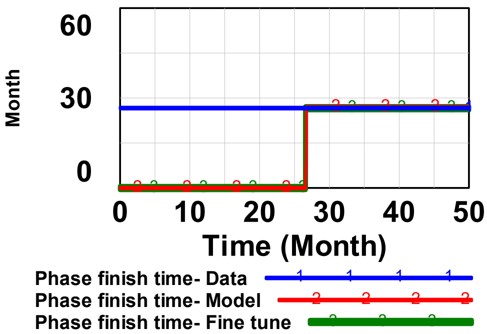


Project C

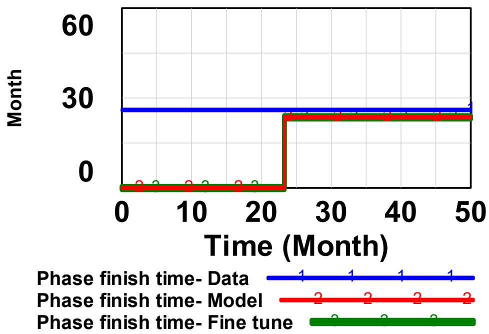


Project D

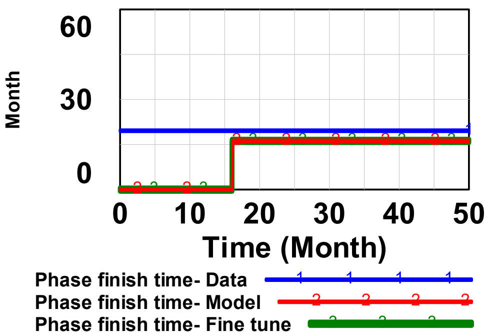
Figure 95: Best fit results for the finish time of conceptual design phase



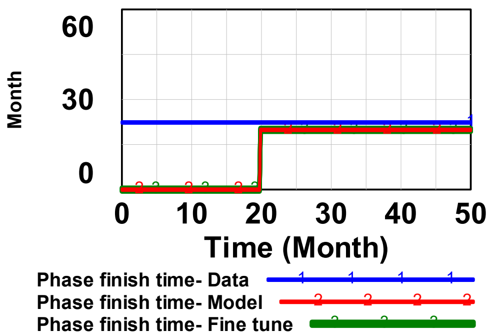
Project A



Project B

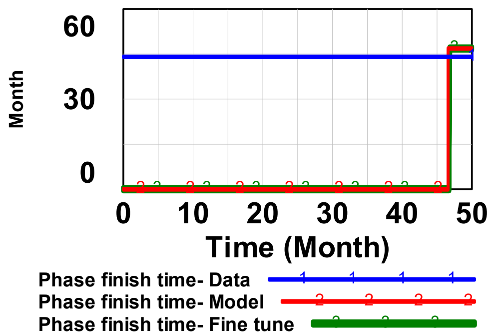


Project C

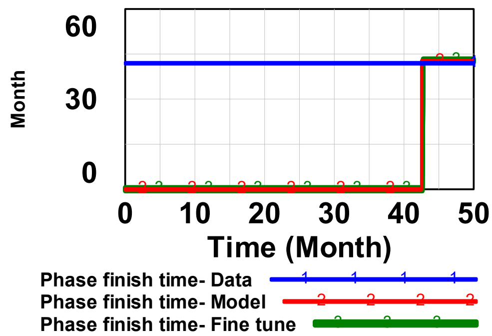


Project D

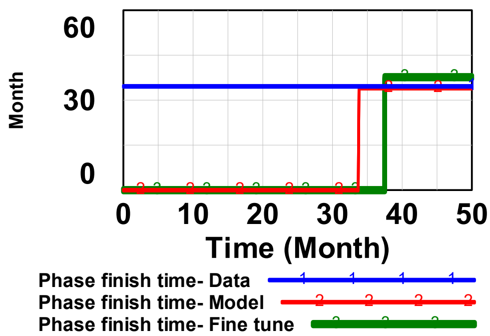
Figure 96: Best fit results for the finish time of detail design phase



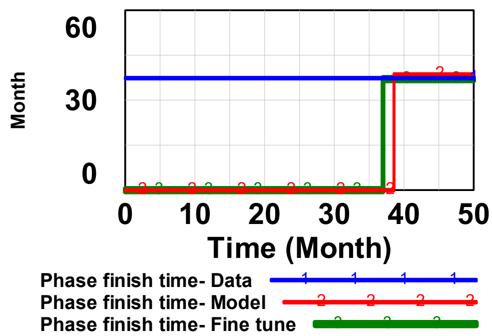
Project A



Project B



Project C



Project D

Figure 97: Best fit results for the finish time of tooling phase

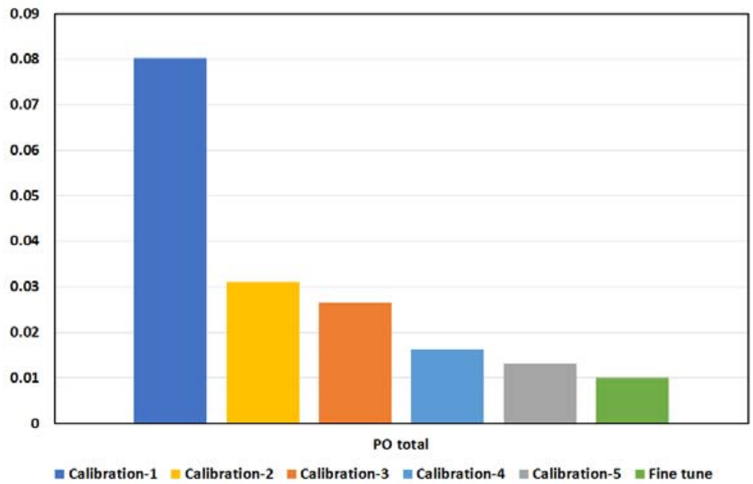


Figure 98: Error values in calibrated data

5.6 Summary

This chapter includes the process of model validation to make it ready for the policy analysis. Different tests are designed by researchers for validating the system dynamics models, dividing between two groups of structural and behavioral validity tests. For the structural validity of the model, the boundary adequacy test, the dimensional consistency test and the structure assessment test are used in this Chapter. The test for the behaviour validity needs to show the similarity of the model behaviour with the real-world situation. For this reason, the model is calibrated using historical data from a car-manufacturing company to estimate the values of unknown constants in the model. After defining an optimization equation, including payoff and decision functions and constraints, using the built-in module in Vensim® the model is optimized for different types of projects. The optimization process first is done individually for each type of projects to find a neighborhood for model constants. Then, repeatedly by fixing the project-independent and project-independent constants, in turn, values of other constants are estimated for each project. Finally, using a combined model, including all project types, and in one step, the values of all constants are fine-tuned. The result of running the model using the estimated values for constants shows a good fit between the model outcomes and the real project performance metrics. Performing the model validation completes the development phase of the research. Following the described process in this Chapter, the developed system dynamics model is validated and its behaviour matches the expected results which ensures its reliability and applicability. In the next chapter, the implementation of the model for the policy analysis using scenario analysis approach is described to address the last research objective.

6 Analysis and Discussion

6.1 Introduction

After establishing enough confidence in the ability of the model to replicate the historical behaviour of the real system, it could be used to analyse different policies and scenarios. The model includes several variables which through manipulating different combinations of them, numerous scenarios could be built and analysed. In this chapter, a number of different combinations of likely scenarios which reflect the purpose of the model are used for further analysis.

In this Chapter, after an introduction about the scenario analysis in Section 6.2, two series of scenarios, based on the objectives of the research are created using changing some variables in the models, and their effects on the model outputs are investigated. In Section 6.2.1, the first groups of scenarios based on changing the duration of the conceptual design phase, as one of important variables related to the SBCE and PBCE approaches are analysed. Then, in Section 6.2.2, the second series of scenarios based on changing the number of initial concepts as another variable related to the SBCE and PBCE approaches are analysed. The Chapter finishes with a summary of the results of scenario analysis.

6.2 Scenario analysis

Scenario analysis is a tool to assist in understanding and analysing the impacts of probable future events on a specific outcome or a set of outcomes. Scenarios typically include a set of explicit 'if−then' rules that explore the consequences of a range of influenced assumptions to reveal the impact of each potential combination of values.

Implementation of the PBCE or SBCE approaches during the conceptual design phase of an NPD project and their effects on the performance metrics of the project are central points of this study. According to LPD literature (e.g. Sobek et al., 1999; Ford and Sobek, 2005; Malak et al., 2009; Khan et al., 2011), the main difference between these two approaches is based on two variables related to the conceptual design phase. The first variable is the duration of the conceptual design phase, consisting of the time period which design teams concurrently work on all concepts, and the convergence period when design teams narrow down the design space to reach to the final concept. The second variable is the number of concepts which the uncertain design space is partitioned to, at the outset of the project. Changing these two variables separately, creates different scenarios, based on them to run the simulation model to investigate their effects.

Despite other system dynamics models of NPD project, in the developed model for this research the changes in the duration of phases does not happen continuously. To model the iterative nature of NPD projects, activities in each phase of the project are conducted through several cycles of iteration, where adding or eliminating each cycle affects the duration and cost of the phase. This is used to develop different scenarios based on the duration of the conceptual design phase. In each scenario, the time length of the conceptual design phase varies by changing the number of iteration cycles while other variables are kept unchanged. The effect of the number of concepts is studied similarly by defining several scenarios, in each one the number of concepts in the conceptual design phase is different. Changing the number of concepts in defined scenarios results in different duration and cost results for the conceptual design phase. Other variables stay unchanged to provide a situation through which it is possible to investigate the effect of changing in the number of concepts on project performance metrics. In following Sections, the results of running the model based on these scenarios are presented and analysed separately for each type of projects.

6.2.1 The effect of the conceptual phase duration on the project performance

The duration of the conceptual design phase in the model is determined by the number of iteration cycles. As shown in the concept confidence loop (Figure 99), the level of uncertainty in the concepts, which is equal to the difference between the instantaneous level of concept confidence and the level of target confidence, determines the number of required iterations. Each iteration cycle reduces the level of uncertainty in concepts in a way that, by the end of this phase, the most feasible single concept could be identified. Design review meetings at the end of each iteration cycle is a managerial control tool with the goal of minimizing development risks (Ahmadi and Wang, 1999). The decision made during each meeting is whether to repeat another cycle of iteration, or to terminate them because the level of confidence built up in the concepts is enough. According to the Equation 17 defined in Section 4.6.3.1.1, the number of concepts and the project complexity affect the rate of confidence increase in the conceptual design phase.

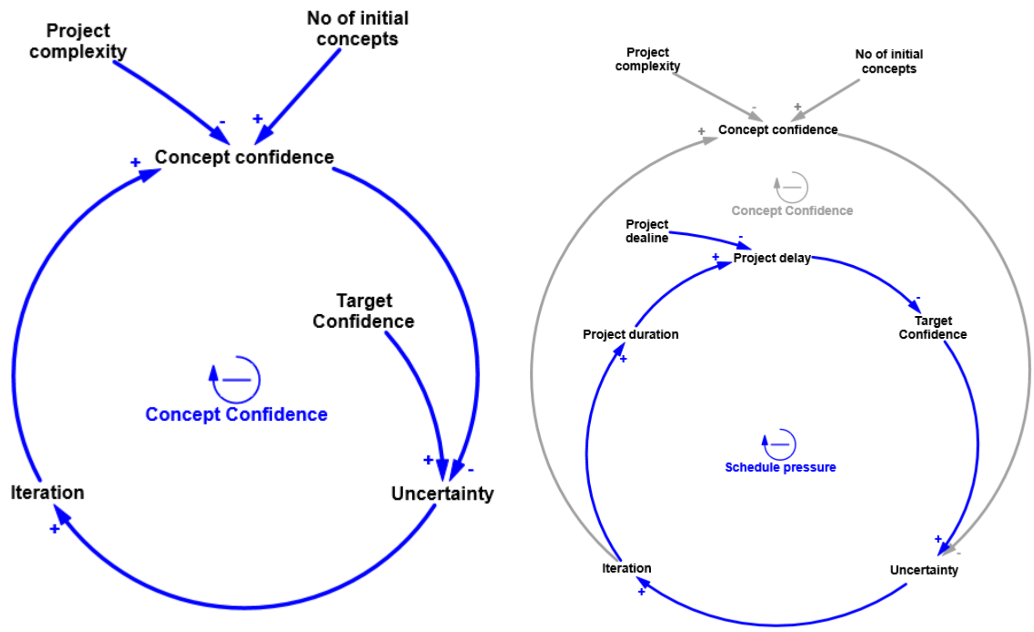


Figure 99: the concept confidence and schedule pressure loops

In the SBCE approach, instead of selecting one concept as soon as possible, design teams gradually eliminate the inferior concepts and allow the best concept to be emerged at the end of the process (Raudberget, 2010; Khan et al., 2011; Levandowski, 2014; Kerga et al., 2014). The conceptual design phase in the SBCE approach could be divided between two sections; in the first section, design teams work simultaneously on all concepts to reduce the epistemic uncertainty by gathering new information and increasing the level of confidence in each individual concept, individually. In the second section, which is called "*Convergence Period*" (Ford and Sobek, 2005) design teams start to identify and eliminate inferior concepts. The duration of the conceptual design phase in the sum to the duration of these two sections. The duration of the first section is determined by the level of target confidence, while the duration of the convergence period depends on the concept elimination rate. Increasing the concept elimination rate and reducing the target confidence results in terminating the conceptual design phase with lower number of iteration cycles, thus reducing the duration and cost of this phase.

The schedule pressure loop shown in Figure 99 represents the effect of target confidence and iteration cycles on the project duration. To simulate these effects in the model, both the level of target confidence and the concept elimination rate are gradually changed as a function of the schedule pressure, while the number of concepts are kept unchanged. In the schedule pressure sector of the model, as discussed in Section 4.6.3.2, the target confidence could be modelled using a first-order stock-and-flow diagram where the stock value changes under the schedule pressure using the flow of '*Target reduce Rate*'. The value of variable '*F2'* defines the strength of the effect of the schedule pressure on the level of target confidence. The value of '*Concept Elimination Rate*', which determines the duration of the convergence period, could also be changed using the stock-and-flow diagram which is used to model the '*Reduction Ratio*' in Section 4.6.3.2. The value of the reduction ratio increases under the effect of the schedule pressure using the flow of '*Reduction Increase Rate*'. Similarly, the strength of the effect of the schedule pressure rate on the reduction ratio is determined by the value of '*F2'*. So, changing the value of variable 'F2' in the model creates different scenarios needed to investigate the effect of the duration of the concept design phase. If the value of '*F2'* is equal to zero, the schedule pressure does not have any effect on the values of the target confidence and the reduction ratio. This shapes the base case scenario in all projects which represents the SBCE approach. Except the value of '*F2'* which is zero, the values of other constants, as determined using the model calibration and represented in Table 10 in previous Chapter, are used to run the base case scenario. Moving towards the PBCE approach is achieved by increasing the value of 'F2' which activates the schedule pressure loop and decreases the number of iteration cycles, stepwise. Table 31 represents all scenarios created for different types projects. As shown in the table, projects A and B, which have the highest level of project complexity in the model, have seven scenarios, because in the SBCE approach, they need seven iteration cycles in the conceptual design phase to close the confidence gap. For projects C, and D the number of iteration cycles in their conceptual design phase reduces to six, and five, respectively, due to their lower level of complexity. In total, 25 scenarios are defined to investigate the effect of the duration of the conceptual design phase on the performance of different types of NPD projects. The model is simulated for each scenario and the model outputs are captured separately for the further analysis.

Table 31: scenarios for investigation the effect of the conceptual design phase duration

	Project A	**Project B**	**Project C**	**Project D**
	No of iteration cycles in the conceptual design phase			
Scenario 1	1	1	1	1
Scenario 2	2	2	2	2
Scenario 3	3	3	3	3
Scenario 4	4	4	4	4
Scenario 5	5	5	5	5 *
Scenario 6	6	6	6 *	
Scenario 7	7 *	7 *		
* Base case scenario				

Figure 100 shows the duration of the conceptual design phase for defined scenarios in project A. The left side of the graph shows the SBCE approach and the right side shows the PBCE approach. Moving from the SBCE towards the PBCE approach happens by reducing the number of iteration cycles. As the result, the duration of conceptual design phase decreases linearly towards the PBCE approach, as expected. As mentioned before, adding each cycle of iteration needs more engineering resources to work on available concepts during that iteration. Consequently, eliminating each iteration cycle means that engineering resources are released earlier from their duties in this phase, and has a direct effect on the cost of conceptual design phase. This cost reduction effect is also linear as shown in Figure 101.

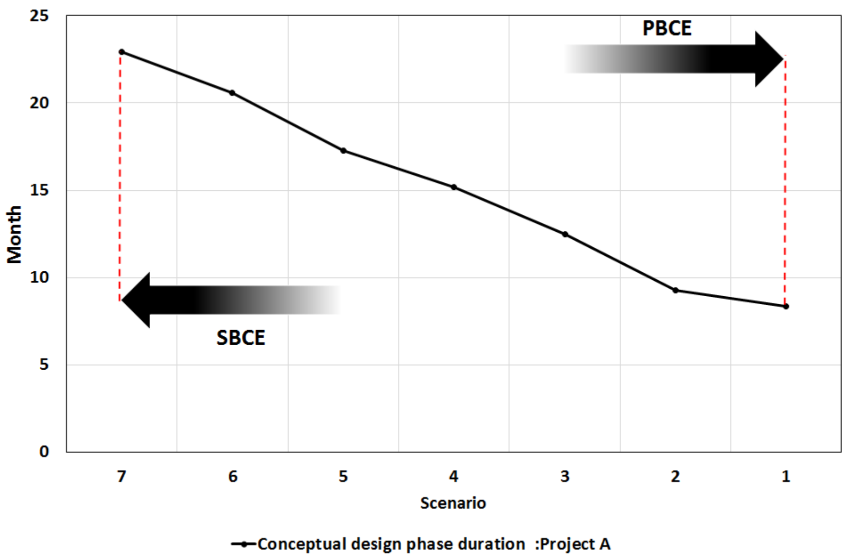


Figure 100: effect of change in iterations number on duration of conceptual design phase in project A

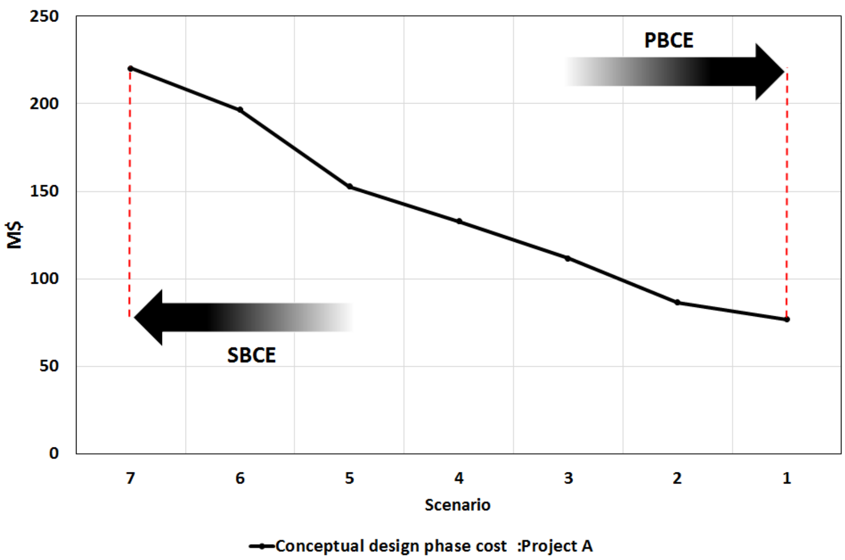


Figure 101: effect of change in iterations number on the cost of conceptual design phase in project A

Figure 102 and Figure 103 compare the duration and the cost of the conceptual design phase for projects A, B, C, and D. Similar to project A, moving towards the PBCE approach by reducing the number of iteration cycles results in a near linear reduction in the duration and the cost of the conceptual design phase for these projects. However, due to the different levels of project complexity, the rate of changes in the duration and cost of this phase are different among compared projects. In addition, and as mentioned before, while in SBCE approach projects A, and B need seven iteration cycles during their conceptual design phase, it is six for project C and five for project D, due to their lower complexity.

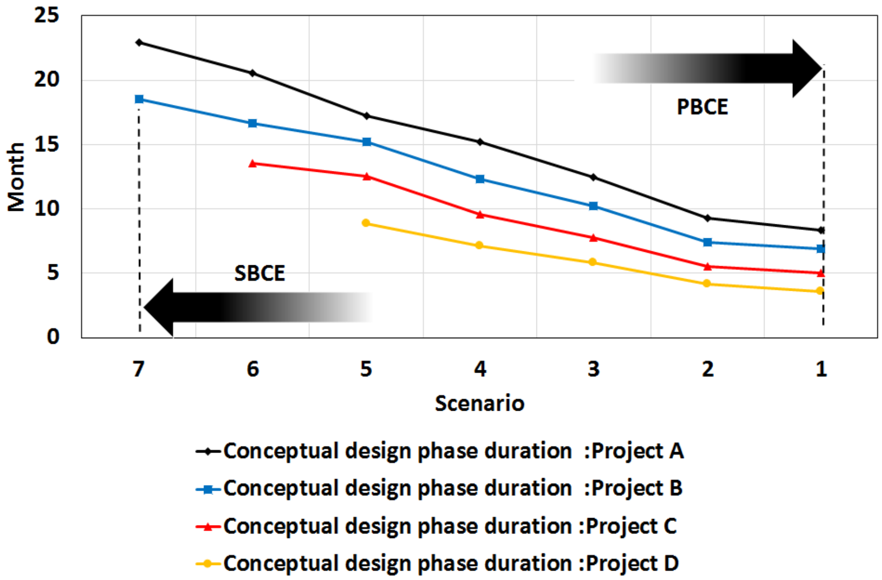


Figure 102: effect of change in iterations number on duration of conceptual design phase in all projects

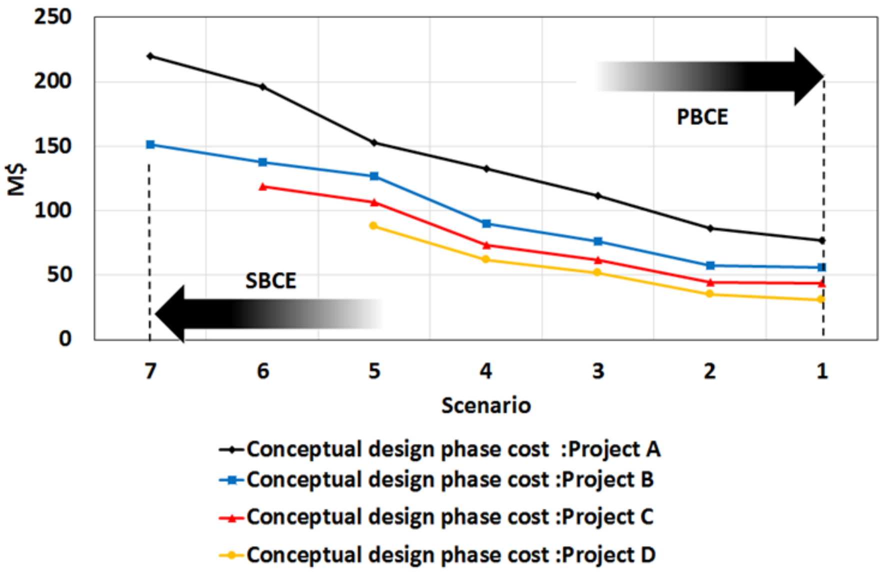


Figure 103: effect of change in iterations number on cost of conceptual design phase in all projects

The goal of decreasing the number of iteration cycles in the conceptual design phase is to reduce the level of schedule pressure, bringing the project back on schedule, and consequently, reducing the project duration, according to the schedule pressure loop. However, the results of running the model based on defined scenarios show unintended results. Figure 104 and Figure 105 represent nonlinear increasing trends in the duration and cost in defined scenarios of project A, as the result of reducing the number of iterations in the conceptual design phase. While elimination of one iteration cycle from scenario 7 to 6 results in more than 9% increase in the project duration and about 8% increase in the project cost, the exponential behaviour of graphs results in a huge gap between the performance metrics when getting closer to the PBCE approach. As shown in Figures, by eliminating six iteration cycles from scenario 7 to 1, there are more than 10 times increase in both project duration and project cost.

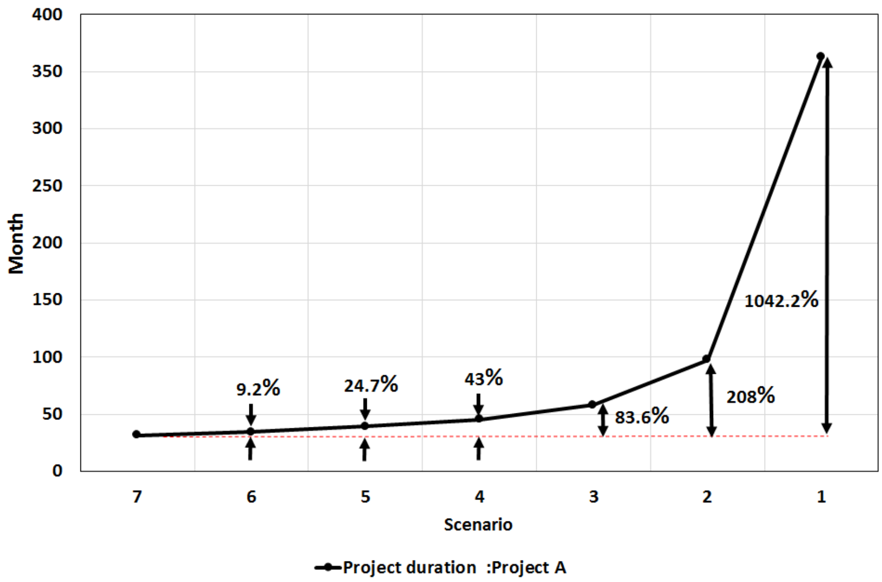


Figure 104: the effect of change in number of iterations on the duration of project A

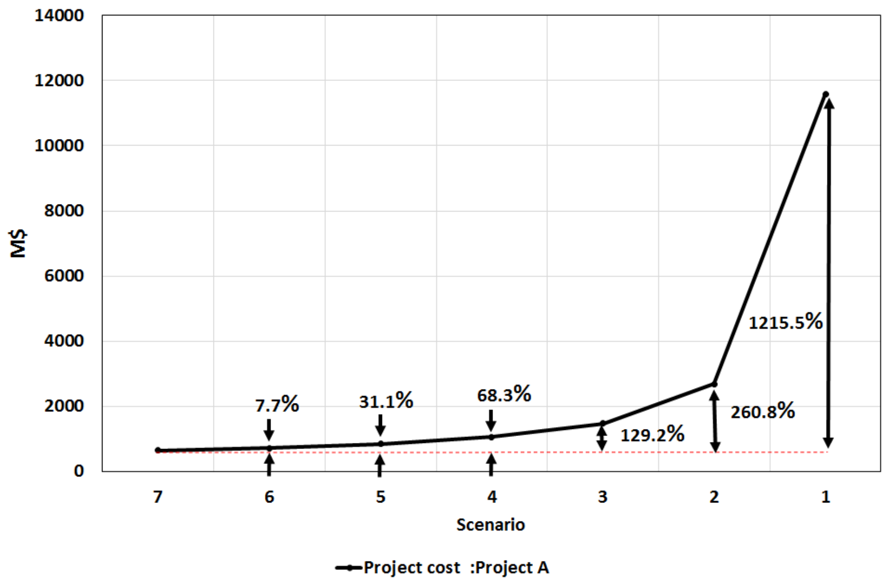


Figure 105: the effect of change in number of iterations on the cost of project A

To explain the reason of these unexpected behaviours in Figure 104 and Figure 105, it is needed to explore again the effect of iteration cycles in NPD projects. As mentioned before, iteration cycles in the conceptual design phase increase the level of confidence in the concepts, step by step, while decreasing the epistemic uncertainty to a level which make it possible to make informed decisions about concepts. These decisions involve the identification of inferior concepts, narrowing down the design space, and finally, making an agreement among all design teams about the final concept for starting the costly activities in the detail design and tooling phases. The immediate consequence of having higher confidence in concepts, as Kennedy et al. (2014) mentioned, is reducing the rework probability in the following phases of the NPD project. Rework cycles mainly occur due to the invalidation of prior decisions through emerging critical knowledge about the design, late in the project. This is captured in Equation 24, where the rework probability depends on the level of confidence in the final concept. Moving towards the PBCE approach by reducing the number of iteration cycles results in making critical decisions early and before emerging the required knowledge (Kennedy et al., 2014), thus decreasing the concept confidence.

Increase in the project cost as shown in Figure 105 could be explained using the graph in Figure 106, which represents the increase in the cost of rework when defects are detected late in the project. In other words, errors are less costly to remove early in the project lifecycle. The reason is the high amount of cost committed by the decisions taken early in the project, during the conceptual design phase, which reduce the flexibility of design to change. Shortening the conceptual design phase by eliminating iteration cycles does not allow the sources of rework to be identified and fixed in this early stage of the project, thus increasing the project cost by postponing the reworks. As the result, adopting a PBCE approach not only increases the project duration, but also has a negative effect on the overall cost of the project, although initially it shows a positive effect on both performance metrics (Figure 100and Figure 101).

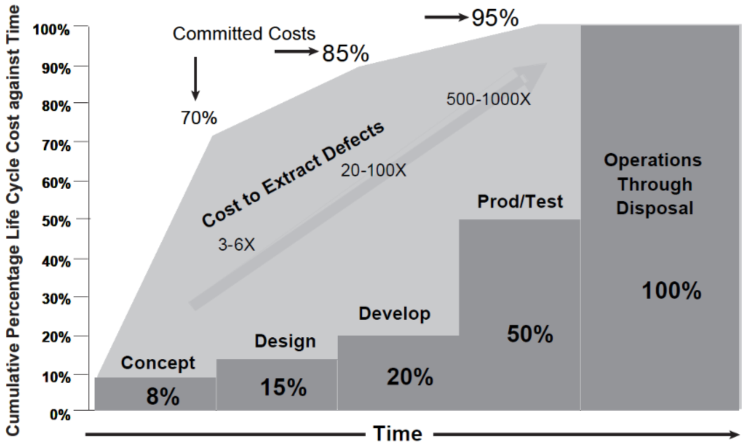


Figure 106: committed cost against time throughout the project (INCOSE, 2006)

Figure 107 compares the percentage of tasks which go through the rework cycle in different scenarios of project A, and clarifies the relationship between the level of design confidence in the conceptual design phase and the rework percentage in the project. In the SBCE approach, having higher number of iteration cycles closes the gap between the level of concept confidence and the target confidence, thus the concept does not contain any epistemic uncertainty while receiving by downstream phases. Consequently, in project A less than 10 percent of tasks are reworked, due to the random uncertainty, throughout the project. In contrary, in the PBCE approach, the confidence level in the final concept when sent to the downstream phases is still low, because the schedule pressure forces design teams to make their decision about the final concept when there is still a high level of uncertainty in the conceptual design phase. Consequently, the concept contains the high level of epistemic uncertainty which results the rework value to be reached to about 90 percent, showing more than eleven-time increase compared to the SBCE approach.

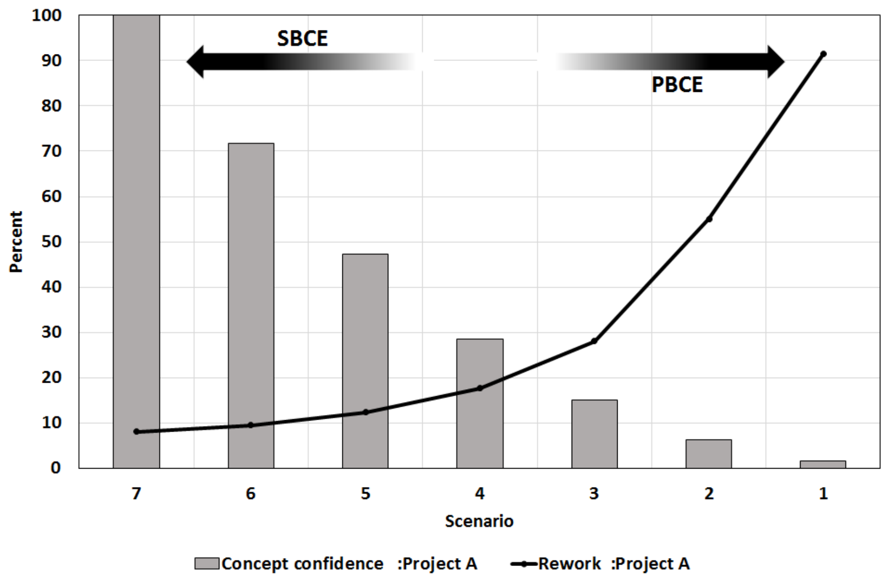


Figure 107: relationship between rework percentage and concept confidence for different scenarios in project A

Reducing the number of iteration cycles in the conceptual design phase has similar effects, as on project A, on the duration and cost of projects B, C, and D. According to Figure 108 and Figure 109, the project duration and cost for all projects increase nonlinearly, when the number of iteration cycles is reduced. Figure 110 shows the changes in the percentages of rework in projects A, B, C, and D for different scenarios. As discussed for project A, the rework rate in all projects is under the effect of the level of concept confidence. While reducing the number of iteration cycles reduces the duration of the conceptual design phase, it does not allow the level of confidence in the final concept to reach to the target level. Passing the final concept with high epistemic uncertainty to downstream phases increases the probability of rework, and make delays and cost overrun in projects. Comparing the behaviour of projects performance graphs in Figure 108, Figure 109, and Figure 110 shows that although reducing the duration of the conceptual design phase has negative effects on all performance measures in all types of projects, these effects are more distinctive in projects with higher levels of complexity.

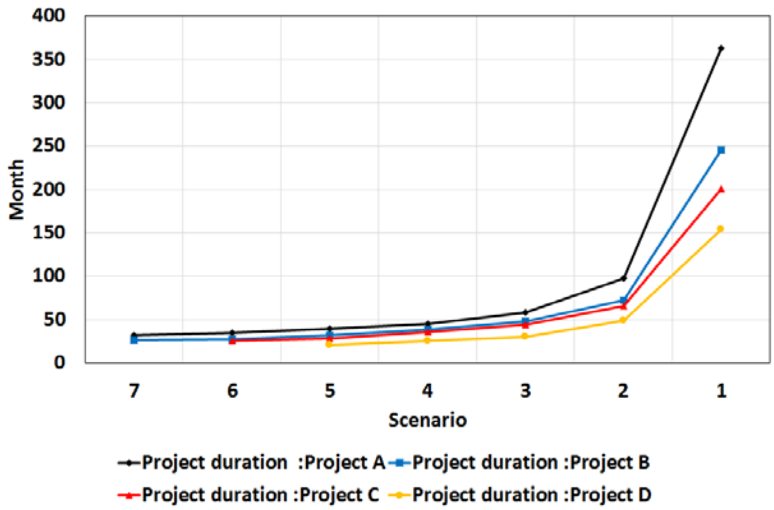


Figure 108: the effect of change in number of iterations on the duration of all projects

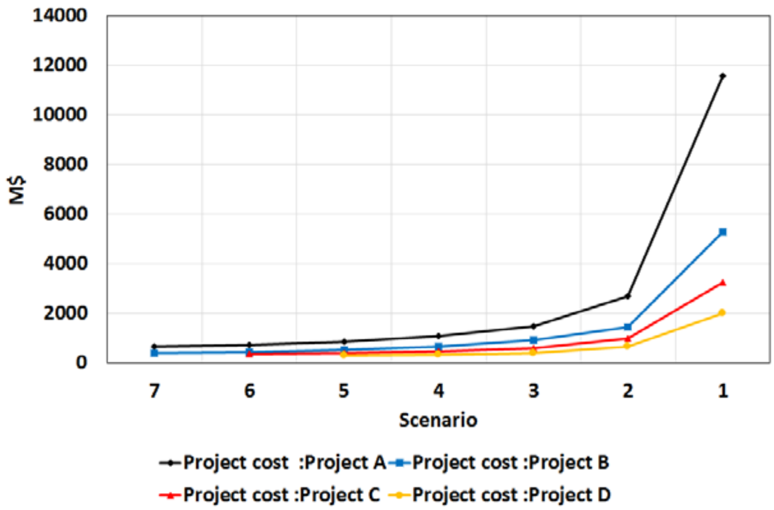


Figure 109: the effect of change in number of iterations on the cost of all projects

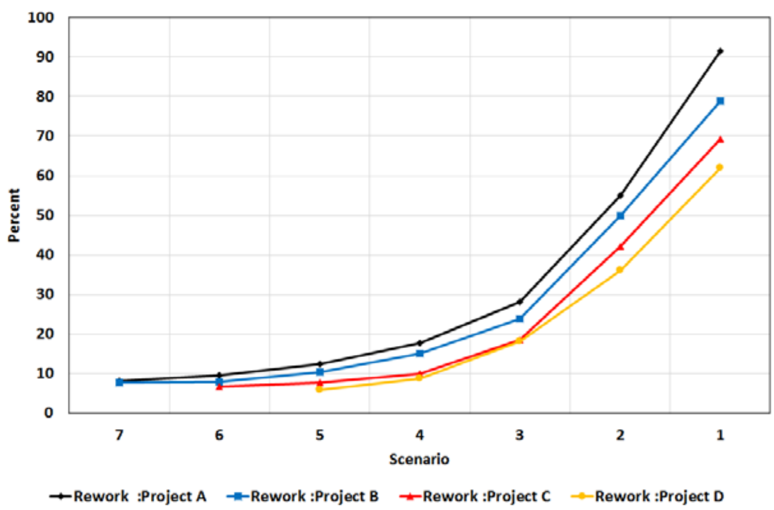


Figure 110: the effect of change in number of iterations on rework percentage in all projects

The last implication from the analysis of scenarios is related to the quality measure. It is traditionally believed that increasing expenditure and the project duration is the only way to achieve higher quality in NPD projects (Harter et al., 2000; Suss and Thomson, 2012). In other words, it is considered that there is a trade-off between time, cost, and quality measures in an NPD project. However, Toyota using LPD has shown that increasing the quality results in less project duration and cost. To falsify the conventional belief, the total percentage of rework and the total confidence of the design are assumed as indicators of the quality in the model. As shown before in Figure 107 and Figure 110, reducing the number of iteration cycles results in an increasing trend in the percentage of the rework, in all projects. The total design confidence is calculated for different scenarios as the average of the level of confidence measured in three phases of the project. The result, as shown in Figure 111, indicates a reducing trend by moving from the SBCE approach, towards the PBCE approach, where the total design confidence for scenario 1 is less than a third of scenario 7. These two measures imply that the quality deteriorates by reducing the number of iterations in the conceptual design phase while the project duration and cost increase. So, it could be concluded that unlike the conventional belief, project performance metrics, including the duration, cost, and quality, are complementary, and implementing LPD reduces the project time and project cost, while improves quality. Figure 112 to Figure 114 shows the changes in the total design confidence and the percentage of rework for projects B, C, and D. the SBCE approach represented on the left-hand side of these graphs represents the highest level of quality, compared with all other scenarios developed for these projects.

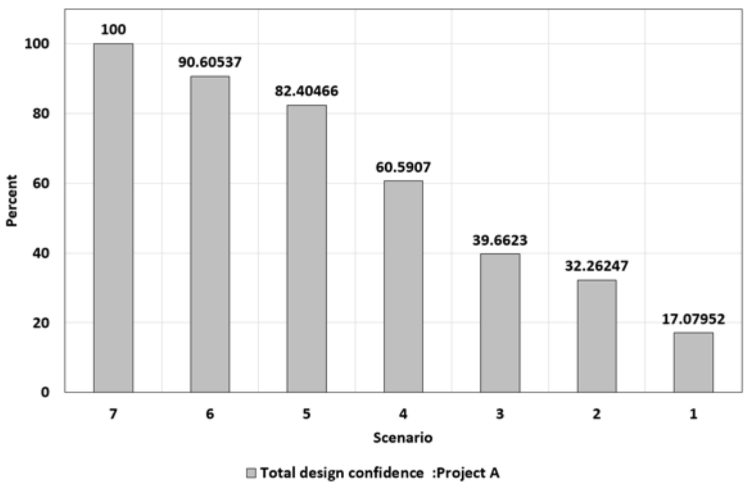


Figure 111: the total design confidence in project A for different number of iterations

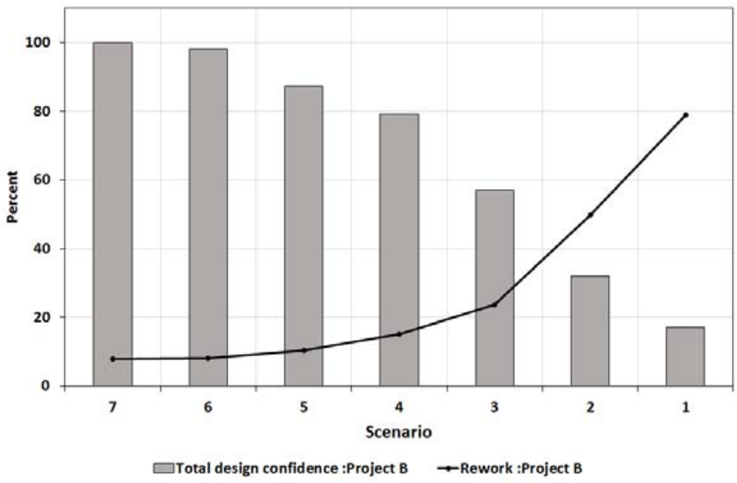


Figure 112: the total design confidence and rework percentage in project B for different number of iterations

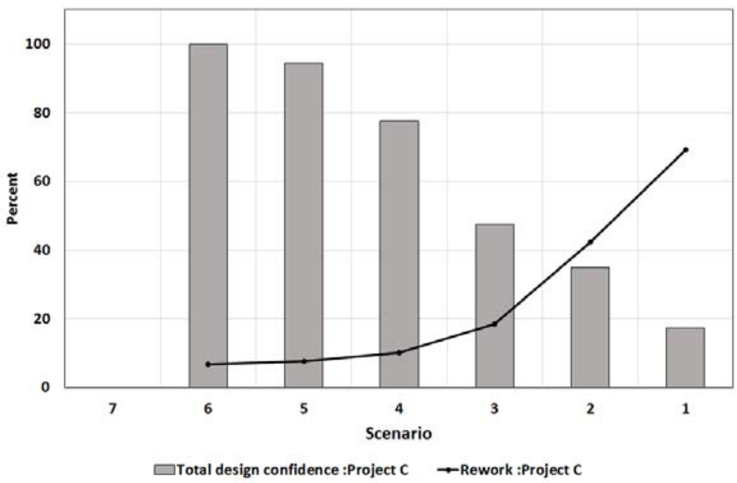


Figure 113: the total design confidence and rework percentage in project C for different number of iterations

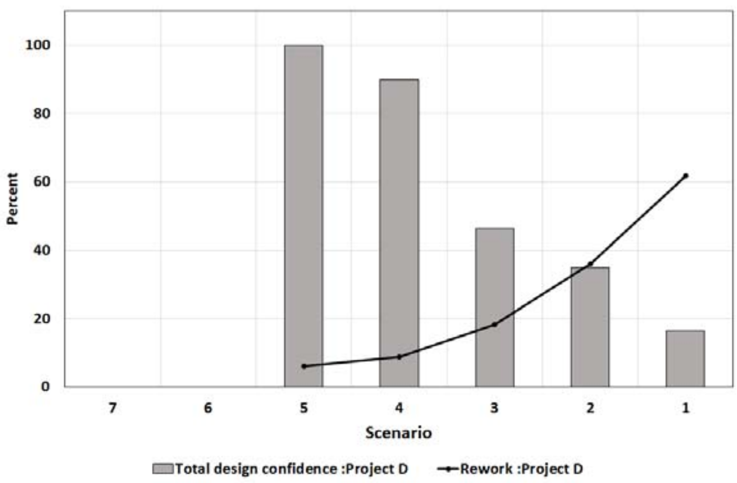


Figure 114: the total design confidence and rework percentage in project D for different number of iterations

This section explains the effect of increase in the duration of the conceptual design phase, on the performance metrics of NPD projects, including the total project duration, cost, and quality, by defining different scenarios and running the model based on them. The duration of the conceptual design phase depends on the number of iteration cycles in this phase, and the base case scenario for different projects represents the situation when all iteration cycles needed to close the confidence gap in the concept and eliminate the epistemic uncertainty are performed. For other scenarios, the number of iteration cycles decreases, stepwise. Table 32 summarizes the differences made in the project performance measures between the base case scenario (representing the SBCE approach), and the scenario 1(representing PBCE approach). While in all projects the performance metrics decline due to the implementation of the PBCE approach, the extents for projects with lower complexity is less than higher complexity projects.

Table 32: the summary of the effect of SBCE approach on performance of projects

		Project A	**Project B**	**Project C**	**Project D**
Moving from SBCE towards PBCE	**Change in project duration (%)**	207.9	176.5	167.8	139.9
	Change in project cost (%)	308.4	225.8	179.8	122.8
	Change in rework (%)	-576.7	-536.2	-518.4	-504.5

6.2.2 The effect of the number of concepts on project performance

6.2.2.1 Analysis of project A

One of the distinctive features of the SBCE approach compared to the PBCE approach is starting the project with higher number of concepts. It is clear from the studies on Toyota product development system (Ward et al., 1995a; Sobek et al., 1999; Morgan and Liker, 2006) and literatures on SBCE (Malak et al., 2009; Kerga et al., 2016) that increasing the number of concepts at the front-end of the project would result in higher project performance. However, the way concept increasing affects the project metrics is unclear. Especially, as increasing the number of concepts is equal to allocating more resources at the upfront of the project which is expected to increase the total cost of the project. To clarify this process, the model is run for all four types of projects, while keeping all variables unchanged and just increasing the number of concepts. The minimum number of concepts is set to be two, because with only one concept having the conceptual design phase would be pointless. The number of concepts increases to a point where a pattern in the project performance behaviour emerges and makes it possible to confidently analyse the emerged behaviour patterns. Other assumptions for these simulation runs are as below:

- Iteration in each phase continues until the level of epistemic uncertainty in that phase falls to zero. As the uncertainty level in the model is directly related to the level of confidence in design, and design confidence is the main indicator for the quality of final product, it is assumed that the final quality would not be compromised for the project time and cost.
- The starting time of each phase is fixed for each type of projects. This means the downstream phase starts its work weather the upstream phase is finished or has delay. This provides the possibility to investigate different degrees of concurrency between phases and their effect on the project performance.
- There is no resource constraint for the project, so the project performance metrics are just under the influence of the process structure.

The simulation results for the effect of increasing the number of concepts on the duration and cost of project A are shown in Figure 115 and Figure 116. The duration graph shows a downward trend, while the cost graph is slightly upward. In Figure 115, increasing the number of concepts from 2 to 5 shows a drop in the project duration (more than 16%), while increasing concepts from 5 to 9 results in less than 3% decrease in the project duration. On the other side, the cost of the project, as shown in Figure 116, is almost descending when the number of concepts increases from 2 to 5 (with an exception when increase concepts from 3 to 4), and results in more than 5% reduction in the project cost. The graph becomes ascending when the number of concepts increases from 5 to 9, even though from 9 to 10, there is again a big decline in the project cost, almost equal to the total cost reduction between 2-5 concepts. To understand the reason for such an unanticipated behaviour in the project duration and cost it is necessary to go deeper and investigate the project phases separately.

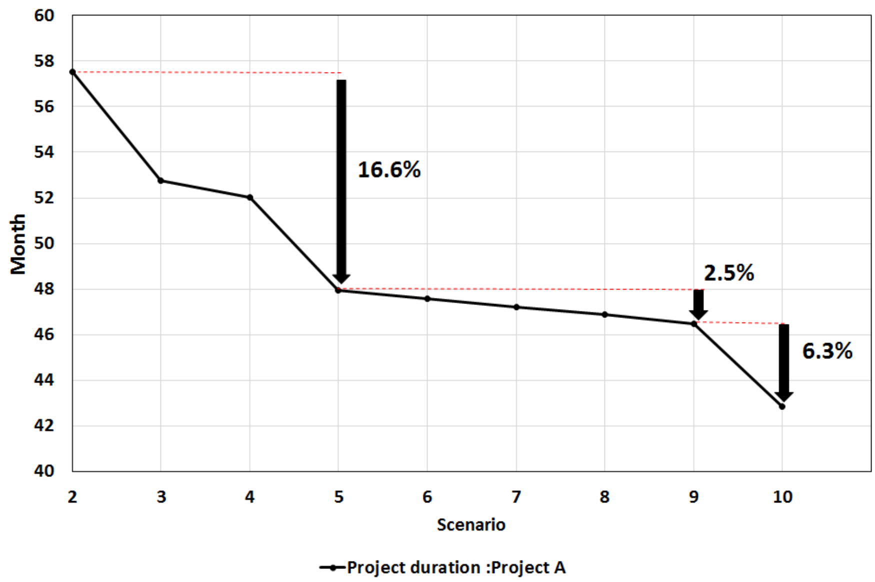


Figure 115: the effect of increasing the number of concepts on the duration of project A

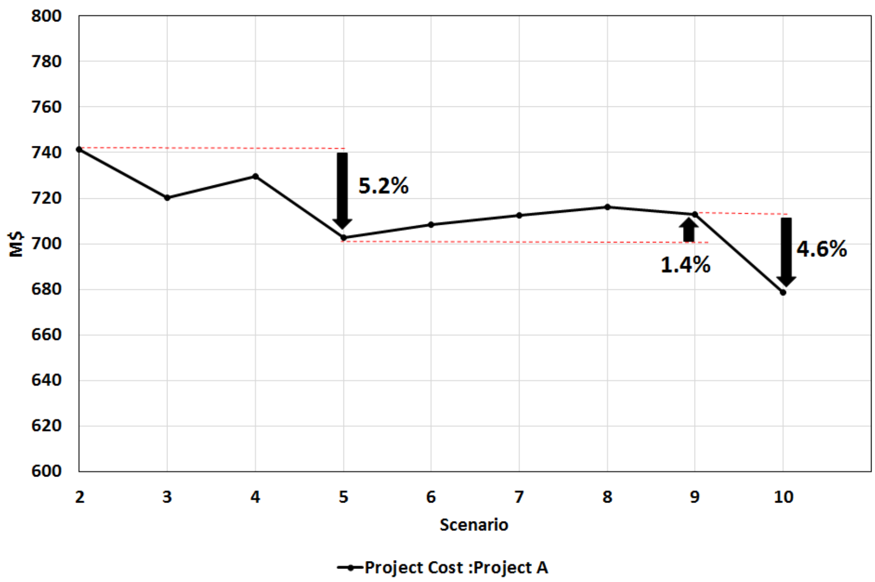


Figure 116: the effect of increasing the number of concepts on the cost of project A

6.2.2.1.1 The effect of the concept number on conceptual design phase

As mentioned previously, the first task of design teams during the conceptual design phase is the identification of the functions which the product must perform and for each function, breaking down the existing design space into a number of concepts. Design concepts are abstractions of a large set of possible design solutions for the final product (Malak et al., 2009). Multiple functions could be combined in diverse ways to form the final concept for detail design. However, during the conceptual design phase when the specific design details are still uncertain, making decision about the set of alternative concepts is very challenging.

Dividing the existing design space into more concepts reduces the level of uncertainty in each concept (Malak et al., 2009). Uncertainty level is modelled as the rate of change in the level of confidence in design towards the target level. Having the higher number of concepts, rises the rate of increase in design confidence during the conceptual design phase, according to Equation 17. The immediate effect of the higher '*Confidence Increase Rate*' on the duration of the conceptual design phase could be classified into three distinct types:

- Type 1: as schematically shown in Figure 117, due to increasing the number of concepts in case (b) compared to case (a), the convergence period starts after fewer number of iteration cycles. The reason is the dependency of the rate of uncertainty reduction in the conceptual design phase on the number of concepts. Figure 118 compares the rate of reduction the uncertainly level in three different scenarios. Based on the level of complexity in the project, the duration of each cycle of iteration differs. Meanwhile, each iteration cycle decreases the epistemic uncertainty in design by gathering more information and increasing the knowledge about it (Kerga et al., 2016). As the process continues, each succeeding iteration cycle becomes shorter than its preceding cycle, due to the learning effect, and this pattern remains throughout the phase. Due to this pattern, iteration cycles before the start of the convergence period have longer duration as the level of uncertainty in this period is still high and design teams should spend more time in each iteration. After starting the convergence period and along with eliminating the inferior concepts, the duration of iteration cycles also reduces. So, elimination of an iteration cycle before the start of the convergence period, as represented in Figure 117, have a greater effect on the overall phase duration, provided that the total number of iteration cycles in conceptual design phase also reduces. Case (c) in Figure 117 is a special case where the total number of the iterations in the conceptual design phase does not change while the convergence period starts after fewer number of iteration. In this case the number of concepts is large enough to let the convergence period starts sooner. However, this case still reduces the duration of the conceptual design phase, as the duration of the eliminated iteration cycle is longer than the iteration cycle added.

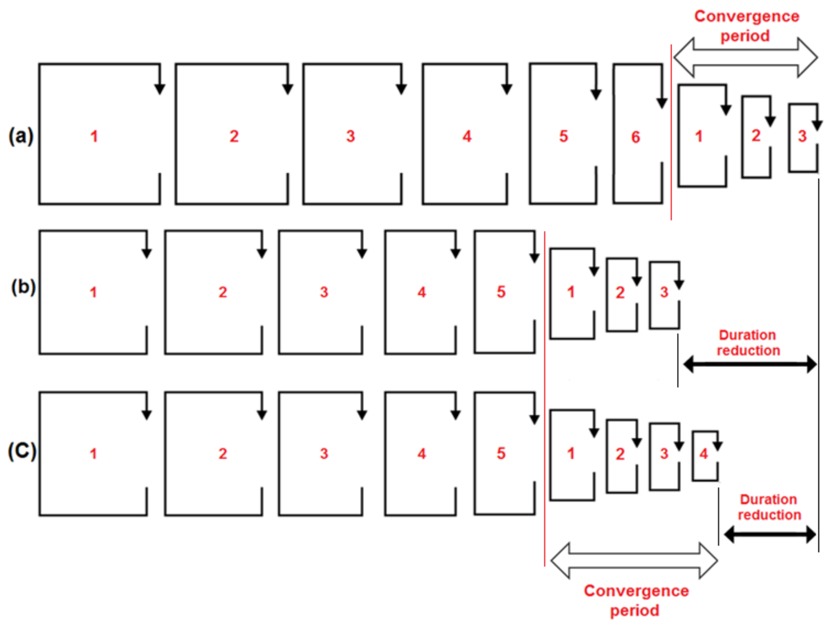


Figure 117: the effects of the number of concepts on the duration of conceptual design phase: Type 1

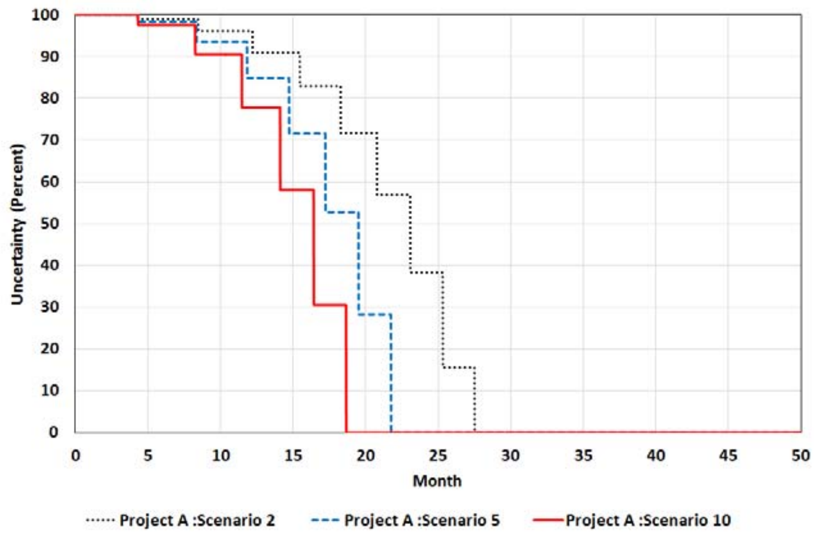


Figure 118: the pattern of uncertainty reduction during the conceptual design phase in three different scenarios

- Type 2: as shown in Figure 119, in this type the number of iteration cycles before the start of the convergence period is equal in both cases (a) and (b), because the effect of increasing the number of concepts on the rate of uncertainty reduction is not large enough to expedite the convergence period. However, it could terminate the convergence period with less number of iterations, as the information about inferior concepts emerges sooner, and narrowing down towards a single concept happens with the higher rate. Although compared with the previous type, type 2 has lower impact on the phase duration, as it still leads to a reduction in the total number of iteration cycles throughout the conceptual design phase, its effect is considerable.

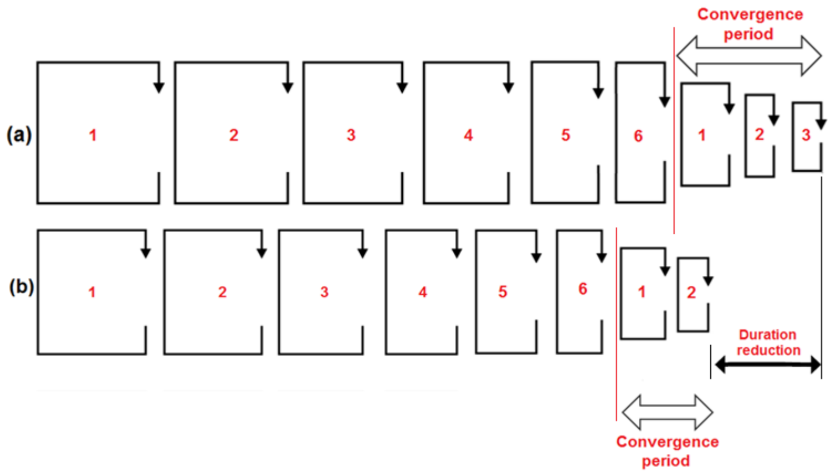


Figure 119: the effects of the number of concepts on the duration of conceptual design phase: Type 2

- Type 3: As shown in Figure 120, in both cases (a) and (b) the conceptual design phase finishes with equal number of iteration cycles. It means that the effect of the concepts number on the rate of uncertainty reduction is not large enough to eliminate iteration cycles. But, having higher number of concepts in case (b) still shortens the duration of each iteration cycle. As the result, even if the number of iteration cycles does not change, higher number of concepts leads to shorter conceptual design phase, however, its impact is lower compared with two other points.

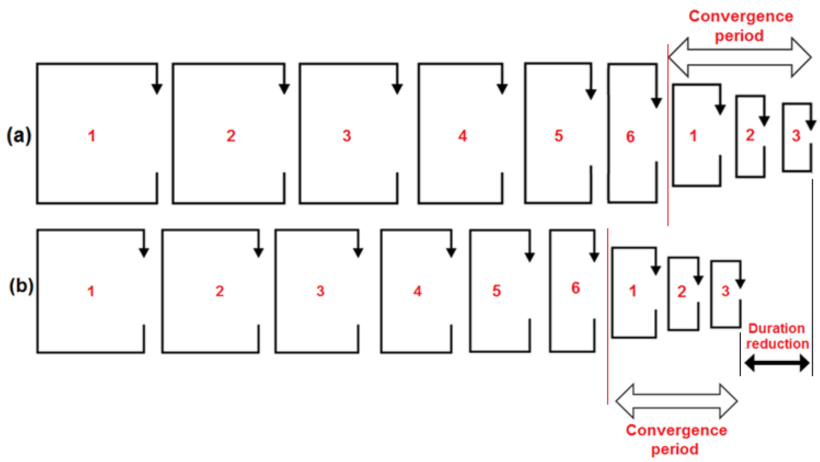
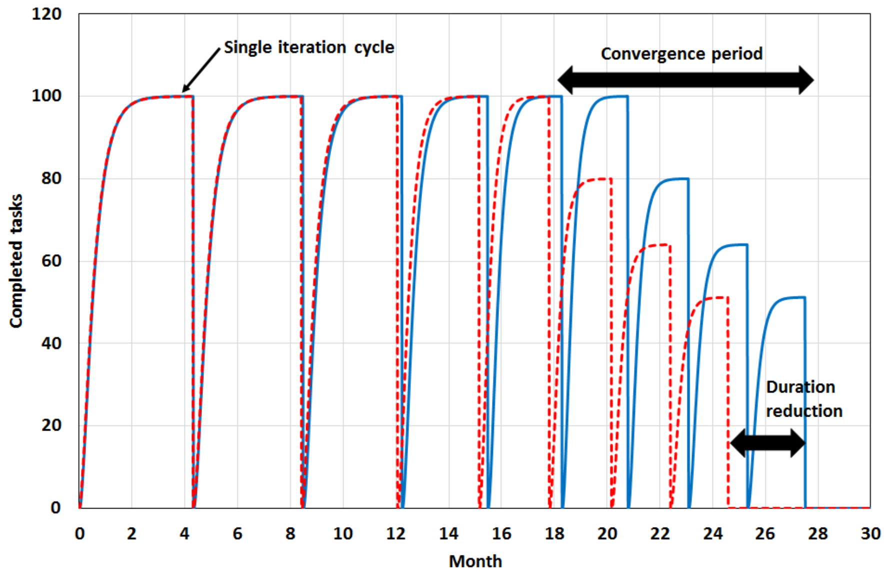


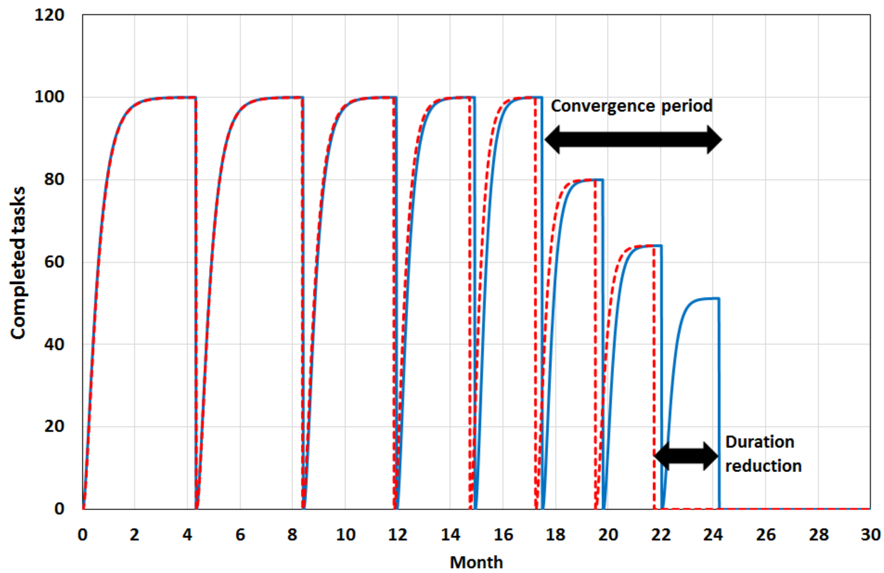
Figure 120: the effects of the number of concepts on the duration of conceptual design phase: Type 3

Figure 121 shows the situations where these three types of duration reduction happen while running the model. The figure shows the changes in the value of stock of '*Tasks Pending*

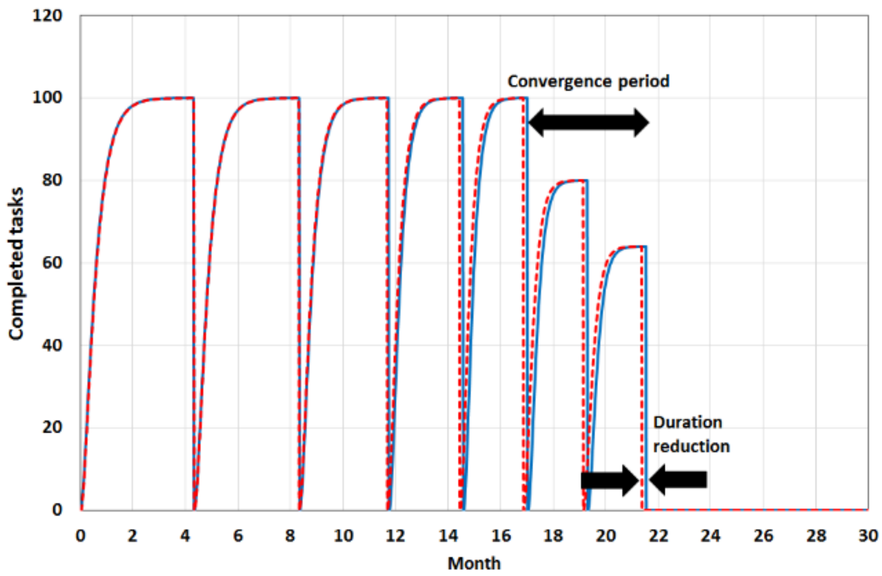
Decision' in the conceptual design phase for two scenarios with different number of initial concepts. The dashed line represents the scenario with higher number of concepts. While each single iteration in this scenario is shorter than its corresponding iteration cycle in the scenario with less concepts, in type one, the exploration of the design space reaches sooner to a point when teams could start narrowing down the number of concepts. As the result, in type 1, one cycle of iteration is eliminated before the convergence period, in type 2 one cycle of iteration is eliminated during the convergence period, and in type 3, both scenarios have the same number of iterations, but with different durations per iteration cycle.



Type 1



Type 2



Type 3

Figure 121: changes in the stock of 'Tasks Pending Decision' for three types of duration reduction in the conceptual design phase

Figure 122 shows the effect of increasing the number of concepts on the duration of conceptual design phase, for different scenarios in project A, which has the highest level of complexity among studied projects. Table 33 represents the slopes of different parts of the graph, based on them, between scenarios 3 to 4 and 5 to 8 the graph follows the same decreasing nonlinear trend, which is interrupted at some points by stepwise drops in the phase duration as the result of increasing the number of concepts. The expected behaviour graph is added to the model based on the trendline of nonlinear parts which is extrapolated throughout the graph, as shown in Figure 123. It reveals that two mentioned parts of the conceptual design phase duration graph of project A follow the expected behaviour. According to Figure 123, the graph deviates from the expected behaviour in four parts of it. The amount of deviation is at its maximum between scenarios 2 and 3. Between scenarios 8 and 9 there is a small deviation from the exponential trend, which is shown in the larger scale, in this Figure. For analysing the behaviour of the conceptual design phase duration of project A, three questions need to be answered:

- Why the graph follows a nonlinear trend?
- Why in four parts of the graph, there are stepwise deviations from the expected behaviour?
- Why in these four parts the values of deviations are different?

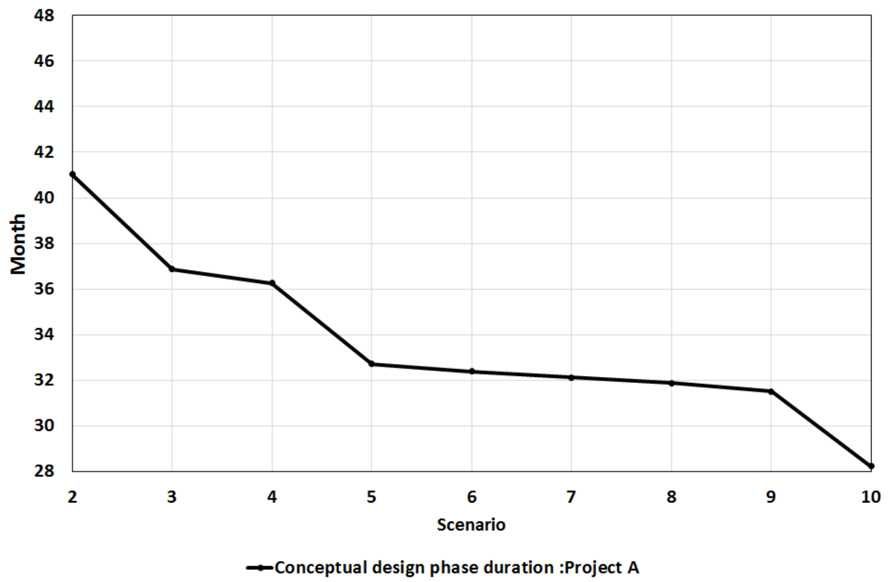


Figure 122: effect of the number of concepts on the conceptual design phase duration for project A

Table 33: the slope of the phase duration graph

Scenario	2to3	3to4	4to5	5to6	6to7	7to8	8to9	9to10
Slope	4.16	0.61	3.55	0.33	0.27	0.23	0.38	3.28

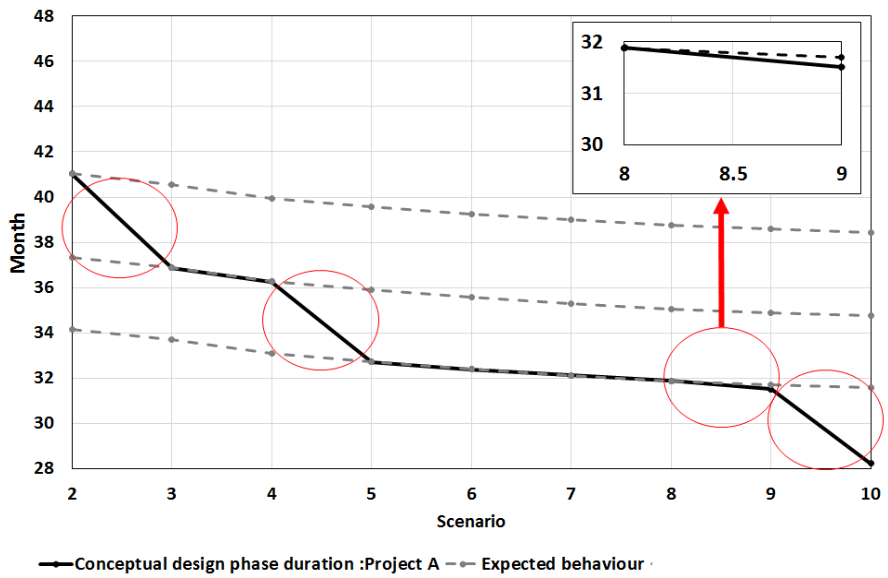


Figure 123: adding the expected behaviour graph to conceptual design phase duration for project A

The nonlinear behaviour of the duration graph results from the fact that reducing uncertainty by increasing the number of concepts shows a goal-seeking behaviour. Malak et al. (2009) mentioned two ways to reduce the level of uncertainty in design, including gathering more information through iteration cycles, and increasing the number of concepts. The level of uncertainty in the conceptual design phase could be reduced by narrowly partitioning the existing design space into more concepts which also reduces the indeterminacy in comparing alternative concepts. As the result, the duration of each single iteration cycle is decreased, which their aggregated effects result in the shorter phase duration (Figure 120). However, it should be noted that each design concept is still an abstraction of the large set of possible design implementations (Malak et al., 2009), and contains a level of uncertainty which is reducible to some extent by narrowing down the partitioning. The rest of the epistemic uncertainty is reducible through cycles of iterations by gathering more information and increasing the knowledge about the design concepts. Moving from scenario 3 to 4, or from scenario 5 to 6 happen by increasing one concept, however, in the first case where the number of concepts is low and the level of uncertainty is high, adding one concept decreases the level of uncertainty with the higher rate, compared to the second case where the level of uncertainty in lower due to higher number of concepts. This difference in the rate of uncertainty reduction results in having a nonlinear trend in the phase duration graph, as shown in Figure 123 for project A.

While the nonlinear behaviour of the phase duration graph is attributable to the increase in the number of concepts according to the type 3 of duration reduction, types 1 and 2 of the duration reduction could explain the stepwise drops in the phase duration in some points of the graph. Although all three duration reduction types result in the reduction of the phase duration, the effect of eliminating one iteration cycle is much more profound, compared to the effect of shortening the duration of a single iteration cycle (Figure 121). Meanwhile, the elimination of iteration cycles is more a discrete event, not happening continuously in all cases when the design space is partitioned more narrowly and the number of concepts increases. In other words, just when the rate of uncertainty reduction (or the rate of confidence increase) due to increasing the number of concepts reaches to a certain level and when the target confidence becomes achievable with fewer iteration cycles, a single cycle would be eliminated. These discrete events show their effects on the phase duration as steps, resulting in the deviation from the expected behaviour.

For project A, as shown in Figure 124, nine iteration cycles are needed in scenario 2 when the conceptual design phase starts with the minimum number of concepts. Adding one concept in scenario 3 reduces the number of iterations to eight, by eliminating a single iteration before the start of the convergence period (Figure 117). Adding the forth concept in scenario 4 does not reduce the number of iteration cycles, thus decreasing the duration only by shortening the time length of each single iteration cycle (Figure 120). As the result, between scenarios 3 and 4 the conceptual design phase duration graph in Figure 123 follows the expected behaviour. The elimination of another iteration cycle happens once again between scenarios 9 and 10, although this time the iteration cycle is eliminated during the convergence period (Figure 119). In addition, between scenarios 8 and 9, although the total number of iteration cycles remains unchanged (Figure 124), the duration graph in Figure 123 shows a small deviation from the expected behaviour, because the convergence period starts sooner due to elimination of a single iteration cycle. However, this iteration cycle just moves to the convergence period (Figure 117 (c)), so the total number of iteration cycles remains the same and the duration reduction effect is not profound.

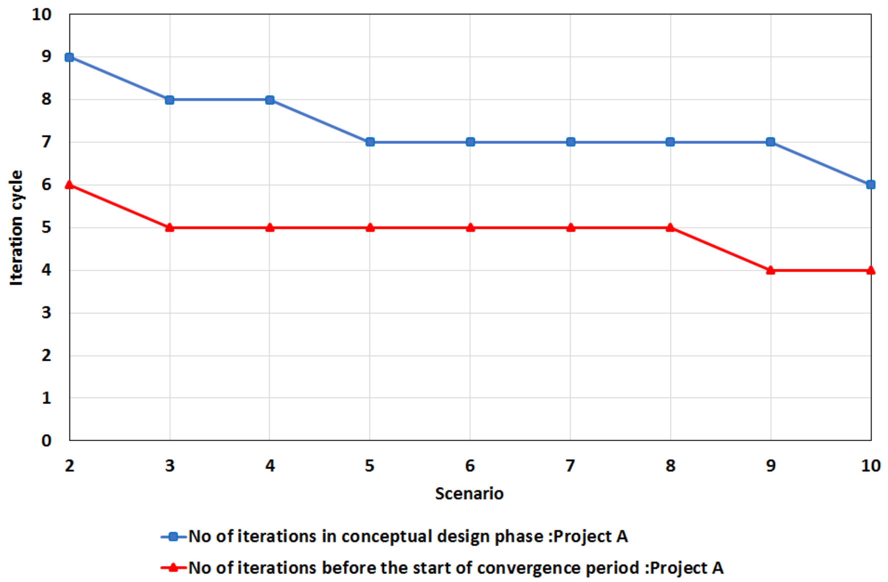


Figure 124: number of iteration cycles in conceptual design phase for project A

Although increasing the number of concepts reduces the duration of conceptual design phase, the trade-off between reducing the level of uncertainty by increasing the number of concepts and the cost of project should also be considered (Malak et al., 2009). Increasing the number of concepts means that more resources need to be allocated to the upfront of the project to form the design teams and concurrently work on concepts. This increases the cost of the conceptual design phase and is expected to be linear, proportional to the number of concepts in each scenario. However, as discussed earlier, increasing the number of concepts also results in the reduction of the duration of the conceptual design phase according to three duration reduction types (Figure 121). Having a shorter phase means consuming less resources which in turn decreases the phase cost. As the result, increasing the number of concepts during the conceptual design phase has two opposite effects on the cost of this phase, and the combination of these opposite effects results in the nonlinear behaviour of the cost of the conceptual design phase.

Figure 125 shows the changes in the cost of conceptual design phase of project A, for different scenarios. Similar to the phase duration graph in Figure 123, the expected behaviour graph, using the slopes of nonlinear parts of the graph, is also added to this Figure, to make the deviations from it clearer. Without any change in the number of iteration cycles, the cost graph would follow the expected behaviour. A stepwise deviation from the expected behaviour occurs in the cost graph when types 1 and 2 of the duration reduction occur, as shown in Figure 118 and Figure 119. The biggest deviation from the expected behaviour happens between scenarios 2 and 3, where a time-consuming iteration cycle before the start of the convergence period is eliminated and results in a drop in the phase cost. As the increasing effect of the number of concepts on the phase cost is still dominant, the graph remains ascending, but with lower slope compared to the expected behaviour. There are also deviations from the expected behaviour between scenarios 4 and 5, and 9 and 10, where one iteration cycle is eliminated during the convergence period (type 2 duration reduction) and reduces the phase cost. Although the values of these deviations are less than the deviation between scenarios 2 and 3, however, due to the lower slope of expected behaviour, the decreasing effect of the number of concepts on the phase cost is dominant, so the graph becomes descending in both cases. The minimum deviation from the expected behaviour occurs between scenarios 8 to 9, where the total number of iteration cycles does not change, while the start of convergence period accelerates. In this case, the combined effect of the number of concepts on the project cost is almost zero, consequently the phase cost remains constant.

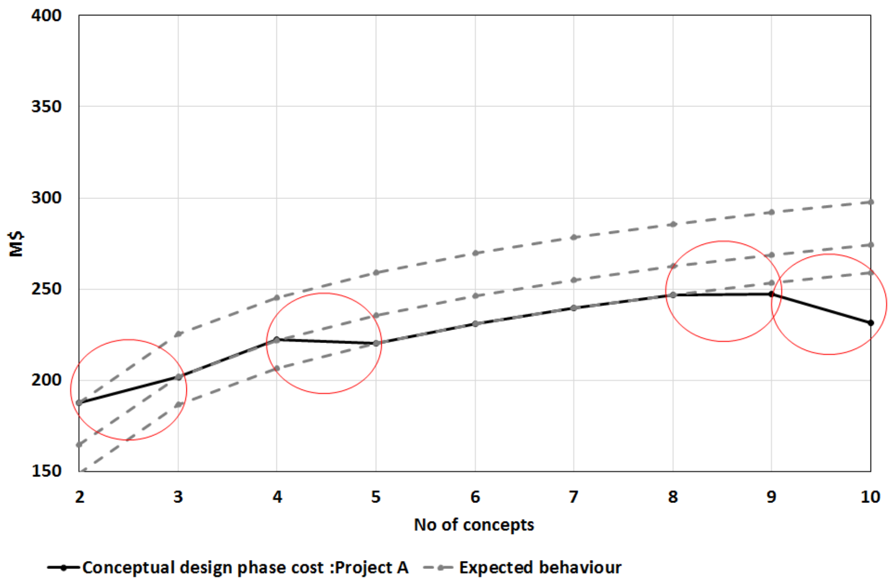


Figure 125: the effect of the number of concepts on the conceptual design phase cost for project A

The main implication of analysing the performance of conceptual design phase of project A under different scenarios is that although in general having higher number of concepts increases the phase cost, an unexpected behaviour in the form of discrete steps could reduce the phase cost. The reducing effect of having more concepts on the phase duration, in some points, even overtakes its increasing effect and leads to lower phase cost with higher number of concepts. The duration of conceptual design phase is directly proportional to the number and duration of iteration cycles which are performed to reduce the level of epistemic uncertainty in concepts and to find the most feasible one for sending to downstream phases. Increasing the number of concepts leads to higher rates of uncertainty reduction, thus reducing the number and duration of iteration cycles. If increasing the number of concepts only reduces the duration of iteration cycles the phase duration would be reduced nonlinearly (type 3 of duration reduction), however, if one or more iteration cycles could be eliminated (Types 1 and 2) the effect on the phase duration would be greater, resulting in a stepwise decrease in the phase duration.

6.2.2.1.2 The effect of the concept number on other phases

The duration and cost of two other phases of the NPD project are affected by the conceptual design phase performance. As mentioned before in the assumption of simulation runs, the starting times of downstream phases are assumed fixed in the model. In other words, the advancement or delay in the conceptual design phase does not directly affect the start and finish time of detail design and tooling phases. However, the key point is the indirect effects by which the performance of the conceptual design phase could affect the rest of project.

Three potential sources of delay and cost overrun in downstream phases are described as below:

- Similar to the conceptual design phase, the number of iteration cycles in the detail design and tooling phases of the project determines the duration and cost of these two phases. Both downstream phases have their own defined target confidence, and increasing the level of design confidence in each cycle of iteration finally results in closing the confidence gap and terminating the phase. The rate of confidence increasing during each iteration cycle, according to Equation 23, depends on the concept confidence. However, in all scenarios defined based on the number of concepts, iteration cycles during the conceptual design phase continue until the concept confidence reaches to the target confidence. This means that the concept confidence at the end of the conceptual design phase is equal in all scenarios. As the result, the rate of confidence increasing in detail design and tooling phases do not change in different scenarios and consequently, the number of iteration cycles to complete the detail design and tooling phases remain constant. This is unlike the scenarios defined based on the duration of the conceptual design phase, where the level of concept confidence in different scenarios are different, thus affecting the rate of design confidence increase in downstream phases.
- Corruption of tasks in concurrently executed phases is the most important source of delay and cost overrun in downstream phases. Corruption in directly affected by the degree of concurrency between phases. It is discussed in more detail previously in Section 4.5.3.5 in Model Construction Chapter. As mentioned before, the start time of phases are fixed, therefore the delay in the conceptual design phase results in an increase in the concurrency between this phase and detail design phase, thus increasing the level of corruption. Similar to the work of Taylor and Ford (2006) corruption is modelled using a flow of tasks to the '*Tasks Not Completed*' stock. These tasks are added to the current scope of the phase and increase the phase duration and cost, proportionately. The same argument could be used for the concurrency and corruption between detail design and tooling phases.
- While the rework probability in downstream phases is not affected directly by the increase in the number of concepts and remains unchanged in all simulation runs, increasing the number of tasks in downstream phases due to corruption increases

the percentage of tasks which goes through the rework cycle, thus increasing the duration and cost of downstream phases.

Figure 126 shows the changes in the percentage of tasks which are corrupted in downstream phases on project A. As mentioned before, increasing the number of concepts during the conceptual design phase of the project A results in the reduction of the duration of this phase. Therefore, the degree of concurrency between the conceptual design and detail design phases decreases, thus reducing the corruption level, because the corruption only happens when there is a level of overlap between two interrelated sequential phases. In addition to the corruption level, reducing the rework as an indirect effect of increasing the number of concepts also result in a decrease in the duration and cost of the detail design phase. Figure 127 shows this effect for project A which as expected, follows the same trajectory as the corruption percentage in Figure 126. Finally, the combination of the effects of iteration cycles, corruption and rework shapes the performance behaviour of detail design and tooling phases. While the number of iteration cycles is unchanged, increasing the number of concepts in different scenarios which decrease the corruption and rework percentages, results in a decline in the duration and cost of downstream phases. For project A, the overall duration and cost of the detail design phase reduces more than 34% and 17%, respectively, between scenarios 2 and 10, as shown in Figure 128.

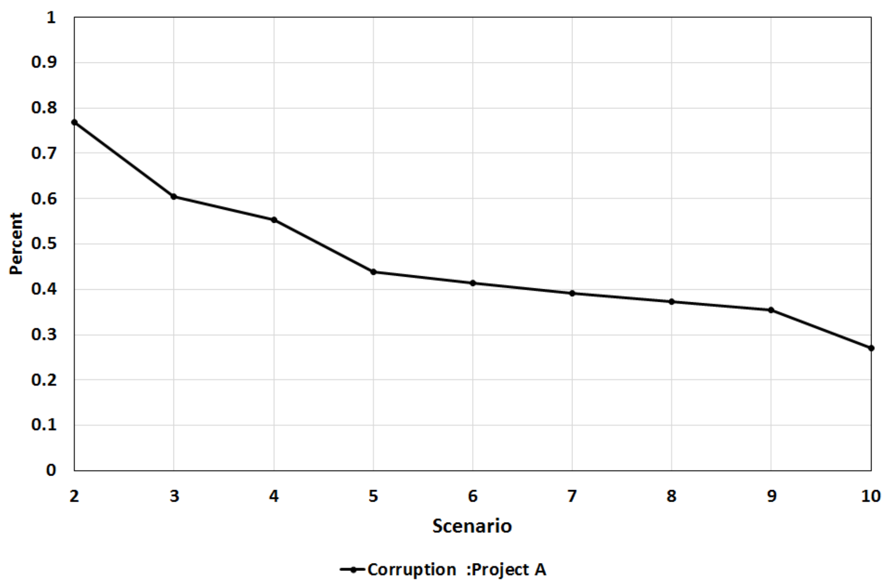


Figure 126: the effect of increasing the number of concepts on corruption in downstream phases of project A

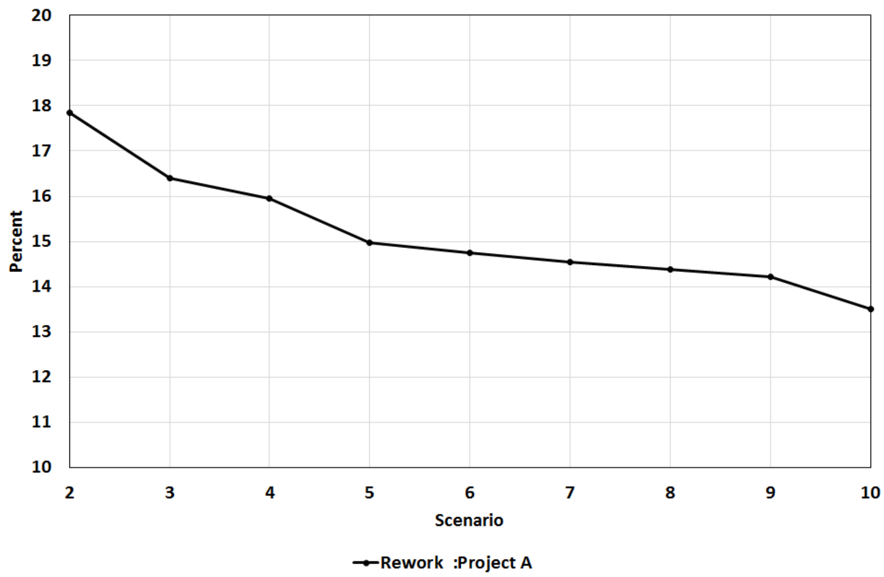


Figure 127: indirect effect of concepts number on the percentage of reworked tasks in project A

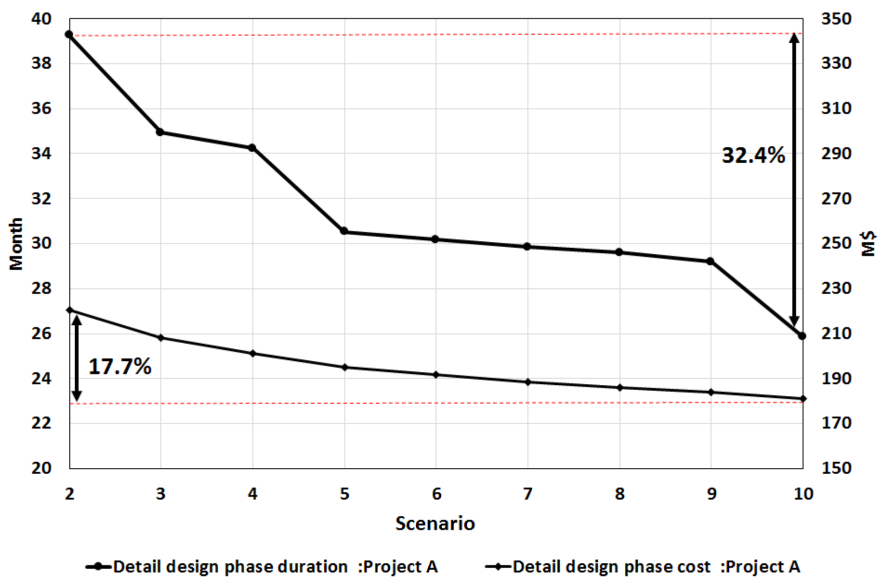


Figure 128: changes in duration and cost of detail design phase of project A due to concepts increasing

Analysing the behaviour of the duration and cost graphs of different phases of project A under the effect of increasing the number of concepts in the conceptual design phase, clarifies the causes behind the changes in the overall project duration and project cost as shown in Figure 115 and Figure 116. Total changes in the project duration as the result of increasing the number of concepts is calculated by adding the duration of individual phases, and subtracting the concurrency periods between them, while changes in the project cost is simply calculated by adding the costs in three project phases. As increasing the number of concepts results in lower duration in all phases, the project duration is also descending.

Although the concurrency between phases reduces the overall project duration, and the concurrency is higher for projects with lower number of concepts due to fixing the starting time of phases in the model, however, they still face higher project durations compared with the projects with more concepts. The reason, as discussed before, lies in the fact that in these projects the duration reduction due to concurrency is less than the duration increasing as the result of corruption and rework increase. Following the discussion about three types of duration reduction in conceptual design phase and their effects on the time performance in all phases, the overall project duration also follows the same trajectory. More than 65% of the total duration reduction happens because of the elimination of iteration cycles between scenarios 2 and 5. As the project cost also reduces, it is logical to increase the number of concepts up to 5 in project A. It would be still room for more reduction in the project duration by increasing the number of concepts over 5, however, adding more concepts increases the cost of project, as well. This puts a trade-off between the time and cost performance metrics of the project. To find if it is worth to decrease the project duration and consequently, time-to market of the new product while compromising the cost of the project, as Reinertsen (2009) clearly suggested having an economic view of the project is mandatory. The economic view helps to have an exact estimate for the cost of delay in the introduction of the new product to the market. Comparing this cost of delay with the incurred cost of increasing the number of concepts helps managers to make more informed decisions which results in the higher profitability as the single goal of most businesses. Yet, although Chen et al (2010). highlighted “a shift in management focus from a more-traditional cost orientation to a time orientation suitable for a fast-changing business environment” (Chen et al., 2010: P17), due to the contingency of the cost of delay to the type of product a company produces and its market environment, finding a single answer for this trade-off problem is challenging.

6.2.2.2 Analysis of project B, C, D

The analysis of the effect of increasing the number of concepts on the performance measures in project A, as presented in the previous Section, could be used for explaining the behaviour of duration and cost graphs in other projects. As previously mentioned, the level of complexity is decreased from project A towards project D. The reason for analysing different types of projects is to find if there is any difference between their behaviour under the same assumptions. Similar to the previous Section, first the behaviour of the conceptual design phase for different scenarios is explained. Figure 129 represents the changes in the duration of the conceptual design phase of project B, for different defined scenarios. Some parts of the graph follow the expected nonlinear descending trend, while in three parts, stepwise deviation from the expected behaviour is obvious. Between scenarios 2 and 3, and 6 and 7, the duration reduction follows the type 1, as explained before, due to the elimination of one iteration cycle, thus creating a drop in the phase duration. between scenarios 4 and 5, only the convergence period starts sooner, without any change in the total number of iterations, so, as shown in Figure 117-C, although still a drop in the phase duration is created, its value is small. The value of the decrease in the duration of the conceptual design phase after scenario 7 is very small.

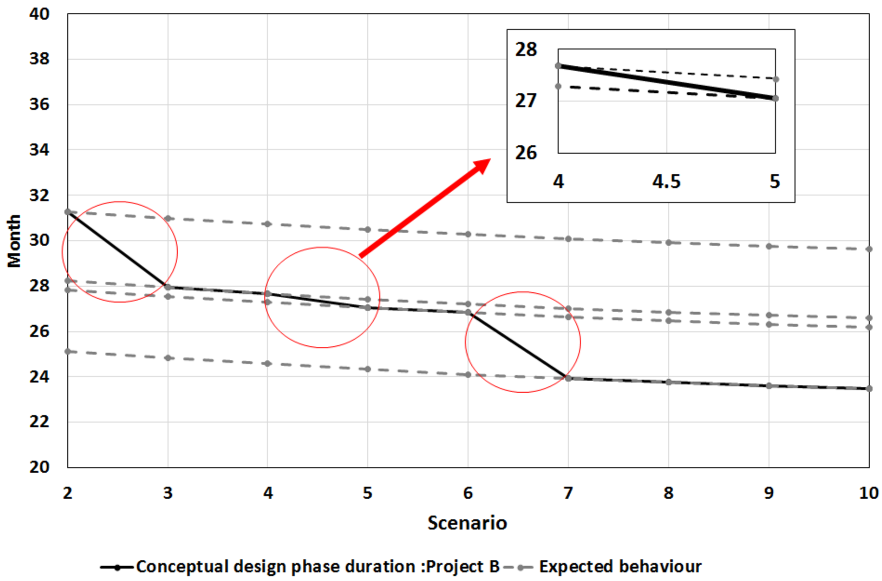


Figure 129: the effect of the number of concepts on the conceptual design phase duration for project B

Figure 130 represents the changes in the cost of the conceptual design phase for project B, under different scenarios. Similar to project A, the cost graph shows the combination of the cost reducing effect of concept number, due to shortening the phase duration, and cost increasing effect, due to allocating more resources. Although between scenarios 2 and 3, and 4 and 5 there are deviations from the expected behaviour, still the cost graph shows an ascending behaviour, because the cost increasing effect is dominant. But between scenarios 6 and 7, the cost reduction effect becomes dominant, as the result, the cost of the conceptual design phase decreases by increasing the number of concepts from 6 to 7.

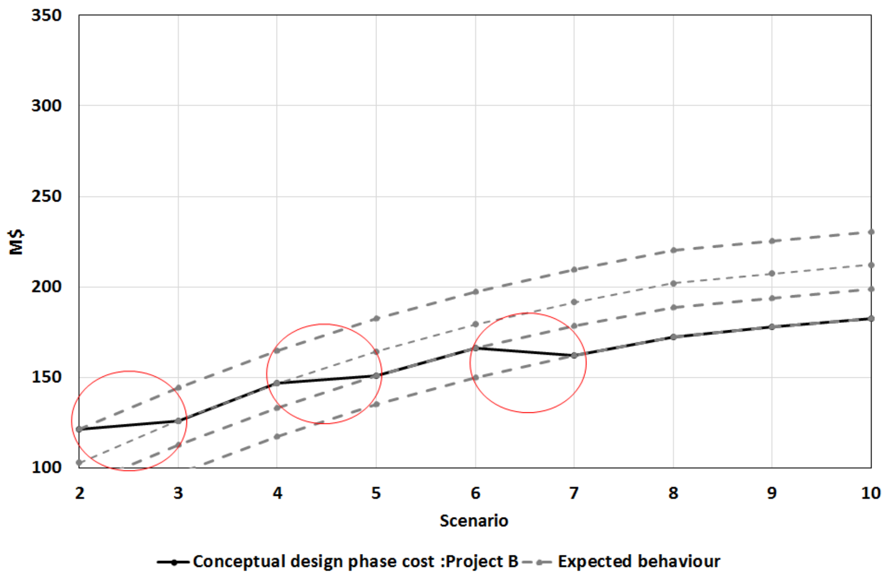


Figure 130: the effect of the number of concepts on the conceptual design phase cost for project B

For project C, as shown in Figure 131, the duration of the conceptual design phase shows deviations from the expected behaviour in three points. Between scenarios 2 and 3, the number of iteration cycles remains unchanged, but the convergence period starts sooner, so a small drop is created. Between scenarios 3 and 4, and 5 and 6, an iteration cycle is eliminated due to increasing the number of concepts, so the drop in the phase duration is more profound, compared to the previous one. After scenario 6, increasing the number of concepts does not show any changes in the duration of the phase.

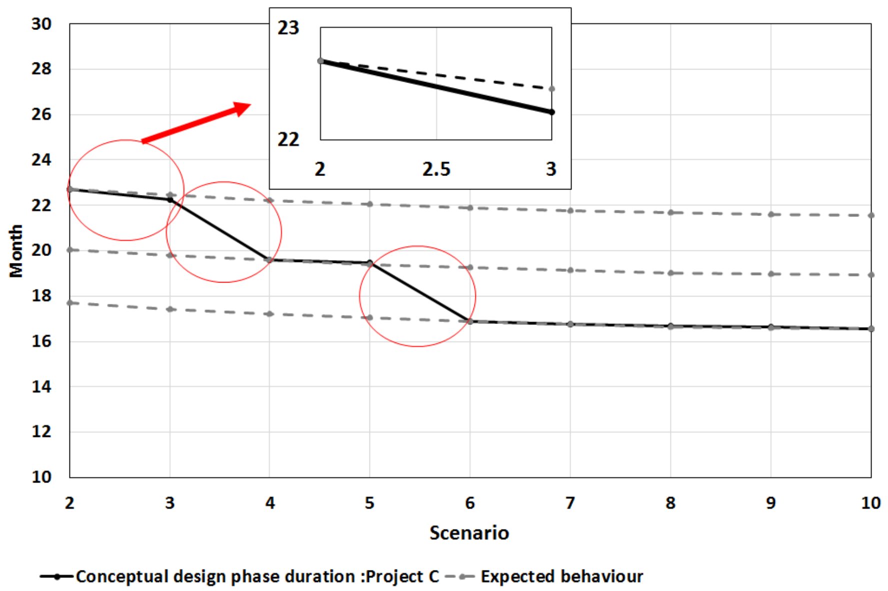


Figure 131: Effect of the number of concepts on the conceptual design phase duration for project C

Similar to the cost graph for project B, in Figure 132, the cost of the conceptual design phase of project C increases by increasing the number of concepts. While there are three deviations from the expected behaviour in the graph, as previously explained, between scenarios 5 and 6, the cost of the conceptual design phase reduces, due to the dominance of the reducing effect of the number of concepts on the phase cost. After scenario 6, the cost still increases and follows the expected behaviour.

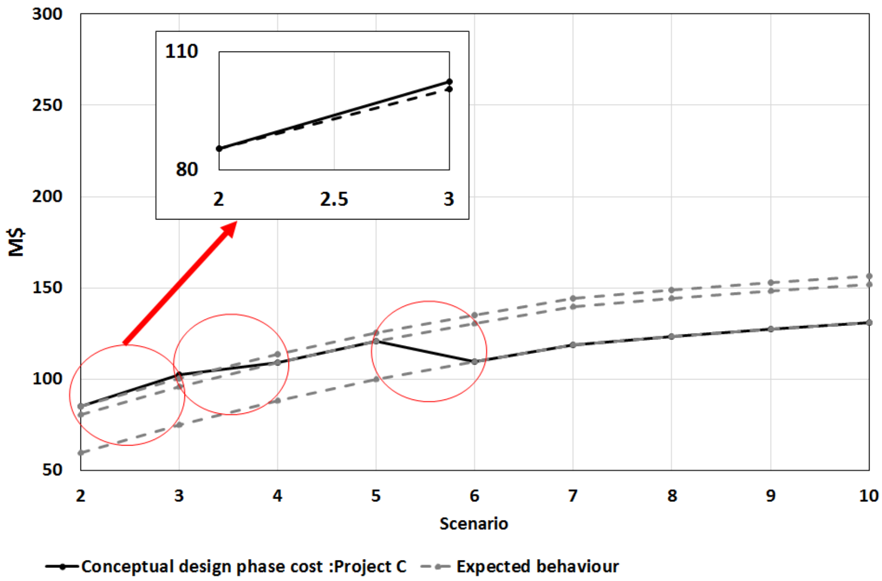


Figure 132: the effect of the number of concepts on the conceptual design phase cost for project C

For project D, there is only one iteration cycle eliminated while increasing the number of concepts. Because of this type 1 duration reduction, a stepwise deviation from the expected behaviour is created in the conceptual design phase duration graph, as shown in Figure 133. Increasing the number of concepts more than four, does not have any effect on the duration of this phase in project D. the cost of conceptual design phase, as shown in Figure 134, has a small drop between scenarios 3 and 4, while in other scenarios the cost increases based on the expected behaviour.

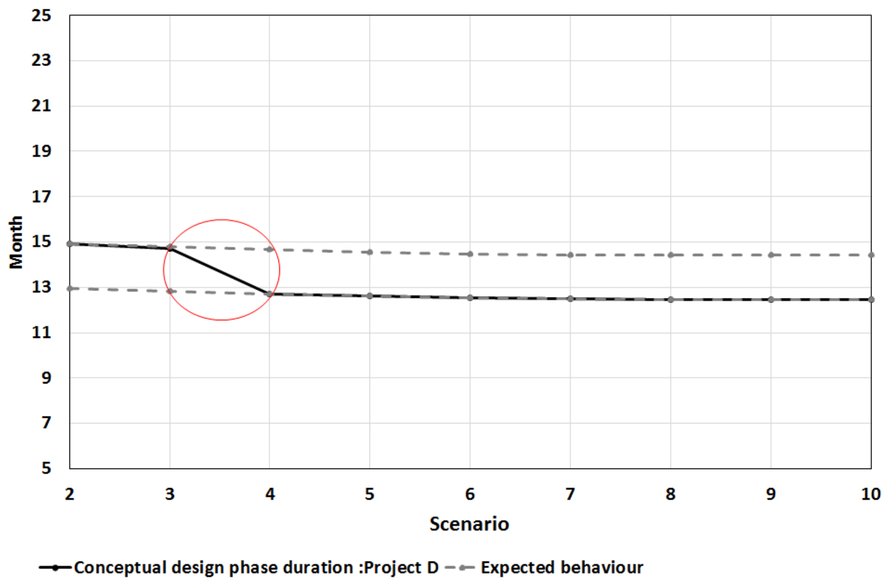


Figure 133: effect of the number of concepts on the conceptual design phase duration for project D

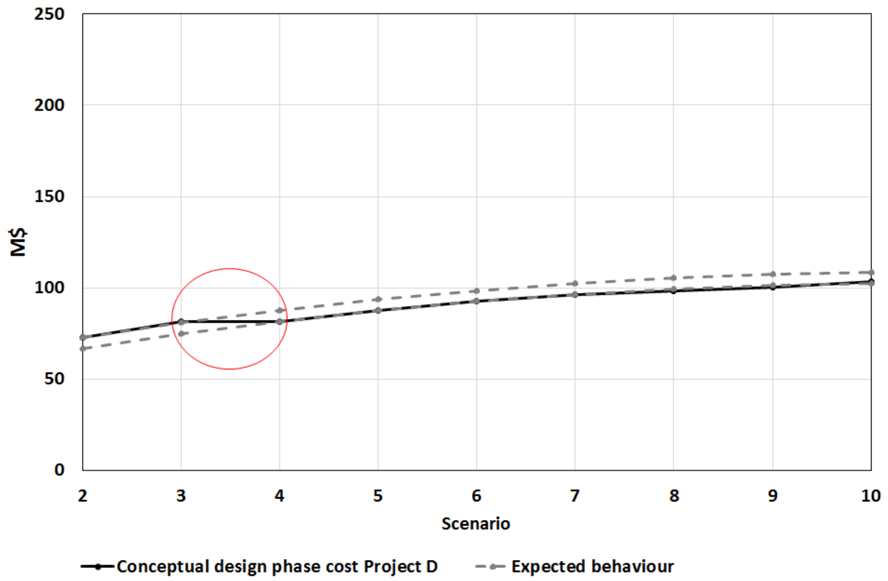


Figure 134: effect of the number of concepts on the conceptual design phase cost for project D

The effect of the performance in the conceptual design phase of projects on the behaviour of other downstream phases are shown in Figure 135 to Figure 137 for projects B, C and D. In all three projects, similar to project A (Figure 128), the cost and duration of the detail design phase reduce, as the result of increasing the number of concepts. As mentioned earlier for project A, the effect of the conceptual design phase on downstream phases of the project could be captured from the viewpoint of the percentage of corruption and rework. The delay in the completion of the conceptual design phase, when the number of concepts is small, increases the concurrency between this phase and the detail design phase, because in the model it is assumed that the starting point of all phases is fixed. Increasing the concurrency results in higher percentage of corruption in the detail design phase, which is added as new tasks to the stock of '*Tasks not Completed*'. This also increases the number of tasks which go through the rework cycle, so increasing the rework percentage. Both corruption and rework percentages result in increasing the duration of the detail design phase, while also increase the phase cost due to the need for allocation of more resources.

The deviations from the expected behaviour in all graphs follow the deviations in their corresponding conceptual design phases. From project B to D, it is clear from Figures that the increase in the number of concepts shows less effect on the performance of the detail design phase. In project B, between scenarios 2 and 10, the duration of the detail design phase decreases by about 27%. For projects C and D, the reduction is 26.5% and 15%, respectively. From the cost point of view, the reduction for project B between scenarios 2 and 10 is 13.5%, for project C is 6.7%, and for project D is 4.3%. Comparing with the results in Figure 128, it could be noticed that while for project A the reductions are happening until the last scenario, for project B, the bigger part of duration and cost reduction happen until scenario 7, and for projects C and D it happens before scenario 6 and 4. The same argument could also be used for the performance of the tooling phase of the projects under study.

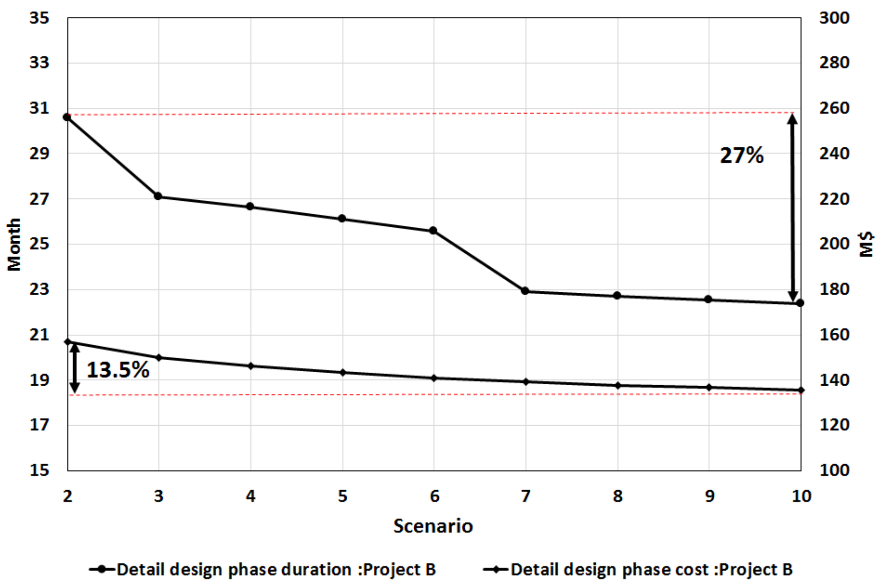


Figure 135: changes in duration and cost of detail design phase of project B due to concepts increasing

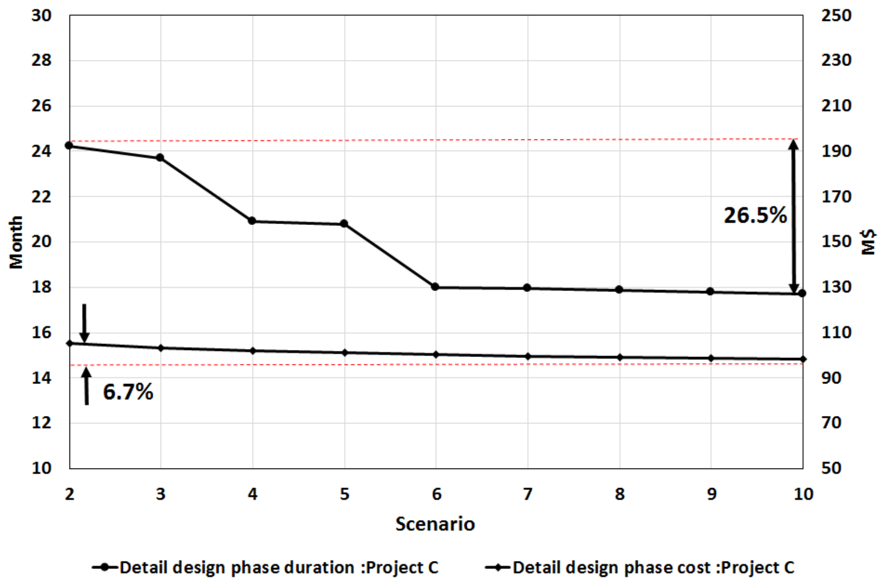


Figure 136: changes in duration and cost of detail design phase of project C due to concepts increasing

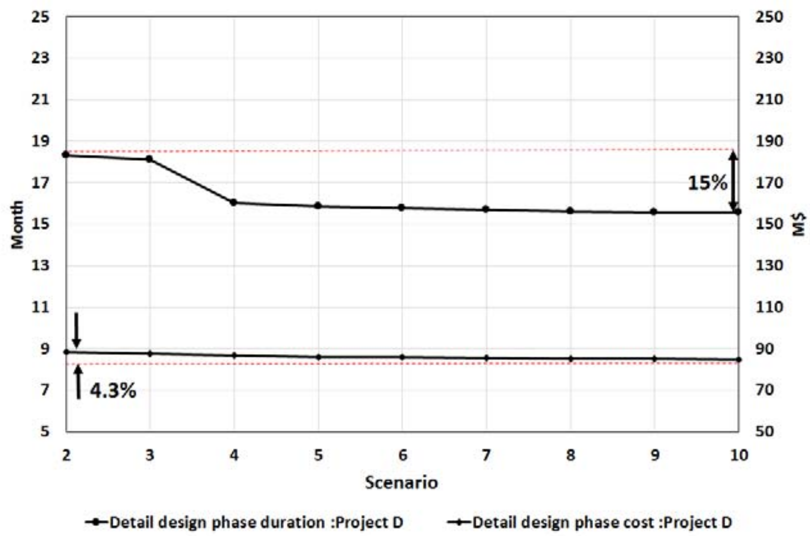


Figure 137: changes in duration and cost of detail design phase of project D due to concepts increasing

Finally, to find the optimum number of concepts to start the NPD project with in different types of projects, the total project duration and cost graphs need to be compared. As explained before, the project duration is the sum of the duration of individual phases, minus the overlapping periods between them, and the project cost is the sum of the cost of each individual phase. As shown in Figure 138, scenario 10 represents the best time and cost performance. In other words, in project A, increasing the number of initial concepts to 10 results in about 25.5% reduction in the total project duration and about 8.5% reduction in the total cost of the project. However, as explained for Figure 125, and also shown in Figure 139, achieving this needs more than 23% increase in the investment upfront of the project.

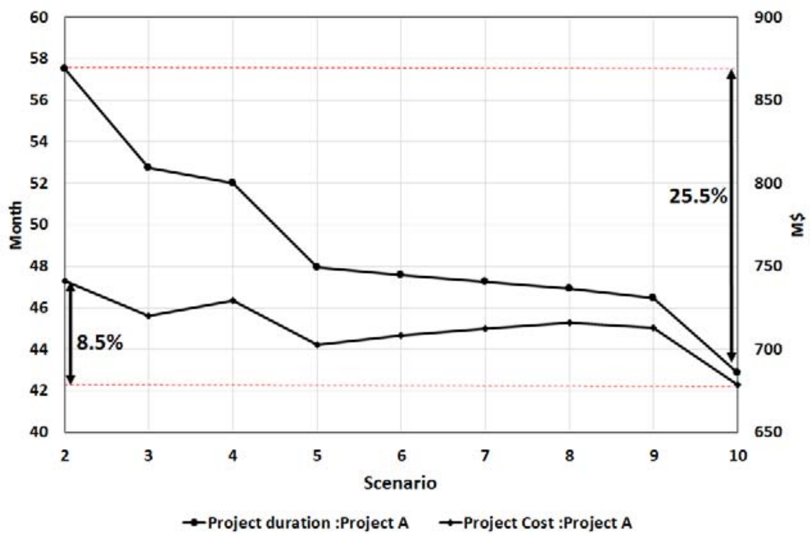


Figure 138: changes in the duration and cost of project A in different scenarios

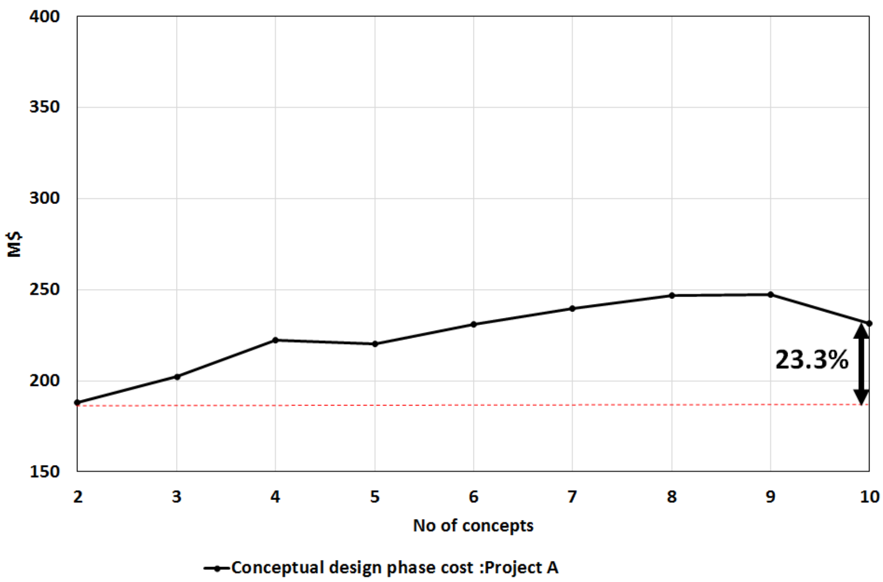


Figure 139: the percentage of increase in the cost of conceptual design phase between scenarios 2 and 10 for project A

For project B, as shown in Figure 140, the optimum point is scenario 7, by it the total project duration reduces about 18%, while the project cost increases only 3%. For 1.3% more reduction in the project duration by increasing the number of concepts to 10, the project cost increases 2.7% more, so the trade-off between the cost and time of the project determines the final choice in this case. it should be noticed in Figure 141, that increasing the number of concepts to seven needs more than 30% increase in the upfront cost, while to increase the number of concepts to 10, the required increase in upfront cost increase to about 50%.

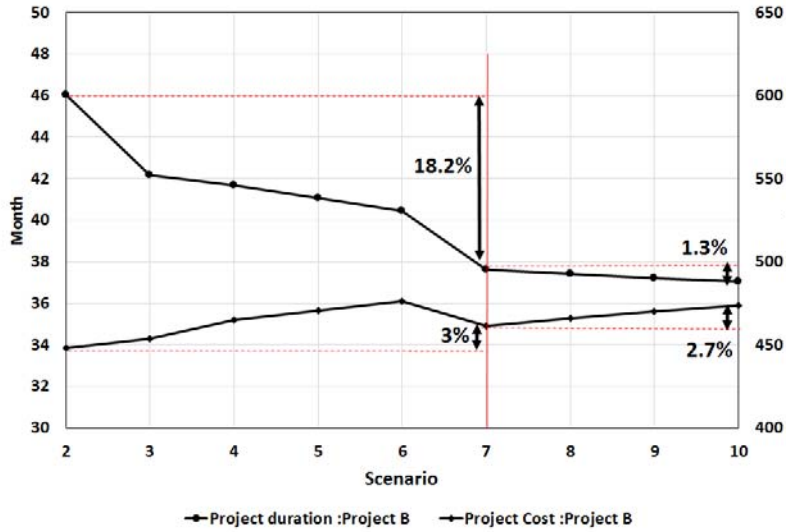


Figure 140: changes in the duration and cost of project B in different scenarios

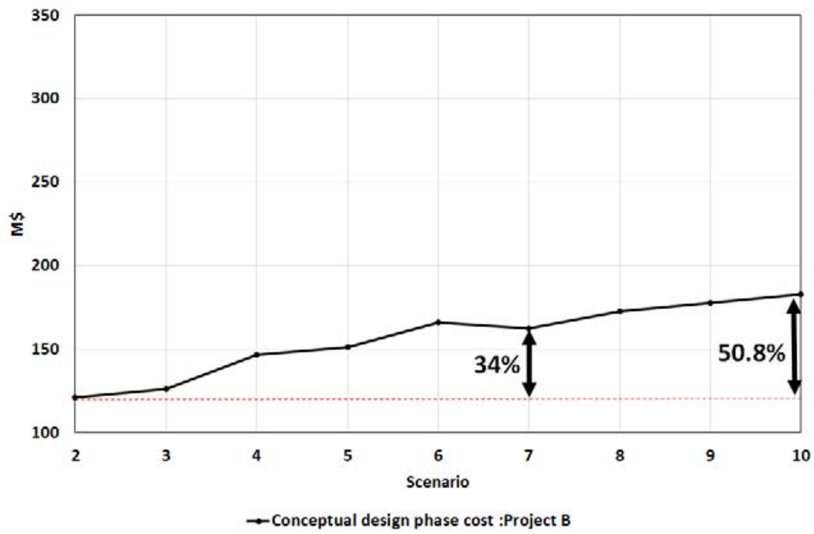


Figure 141: increase in cost of conceptual design phase between scenarios 2, 7 and 10 for project B

For project C, the optimum point is scenario 6. While having six initial concepts reduces the total project duration for more than 16%, and makes 6% increase in the project cost, having more concepts does not have any effect on the project duration, as shown in Figure 142. The point is, although the project duration does not change, due to allocating more resources into the conceptual design phase, the total project cost increases by 4.6%, between scenarios 6 and 10. According to Figure 143, the increase in the number of concepts to six, calls for more than 28% increasing in the upfront investment. This Figure also shows that more than

53% increase in upfront investment is needed if scenario 10 is targeted, while as mentioned, implementing this scenario does not improve the time performance of project C.

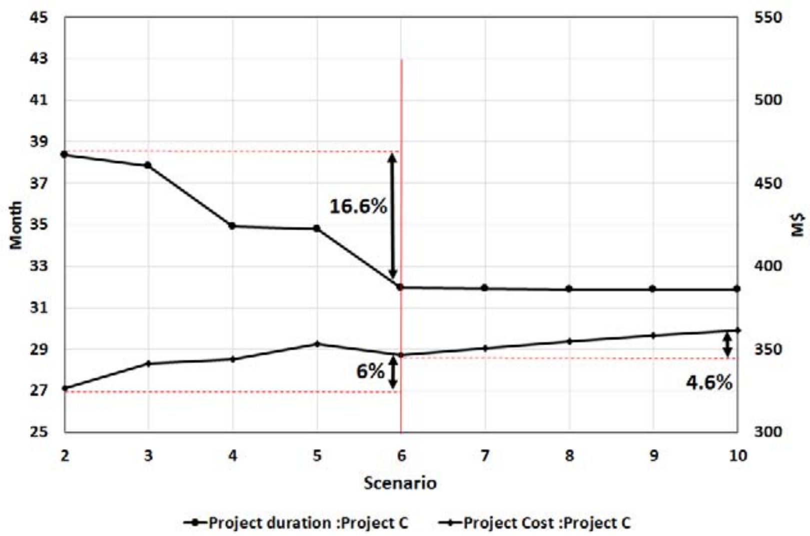


Figure 142: changes in the duration and cost of project C in different scenarios

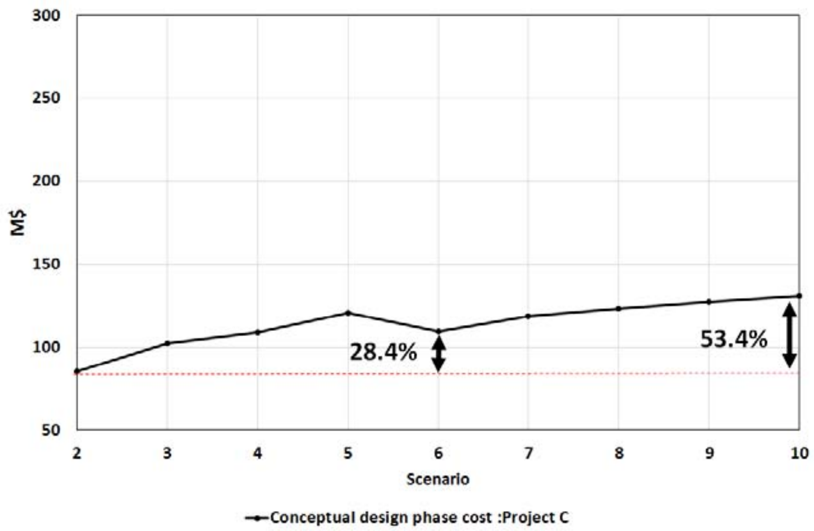


Figure 143: increase in cost of conceptual design phase between scenarios 2, 6 and 10 for project C

The final case to analyse in project D. according to Figure 144, scenario 4 is the optimum point for this project. Increasing the number of initial concepts to 4 reduces the total project duration by more than 8%, while needs 4.5% more investment in the project. Based on the market condition, the decision could be made to introduce the product sooner to the market by more investment. After scenario 4, as is clear in Figure 144, the project duration does not show any considerable change by increasing the number of initial concept, while increasing the number of concepts to 10 increases the project cost more than 9%. Increasing the number of concepts to four in project D, needs about 12% increase in the upfront project investment, while for increasing the number of initial concepts to 10, the project upfront investment should be increased by 42%, as represented in Figure 145.

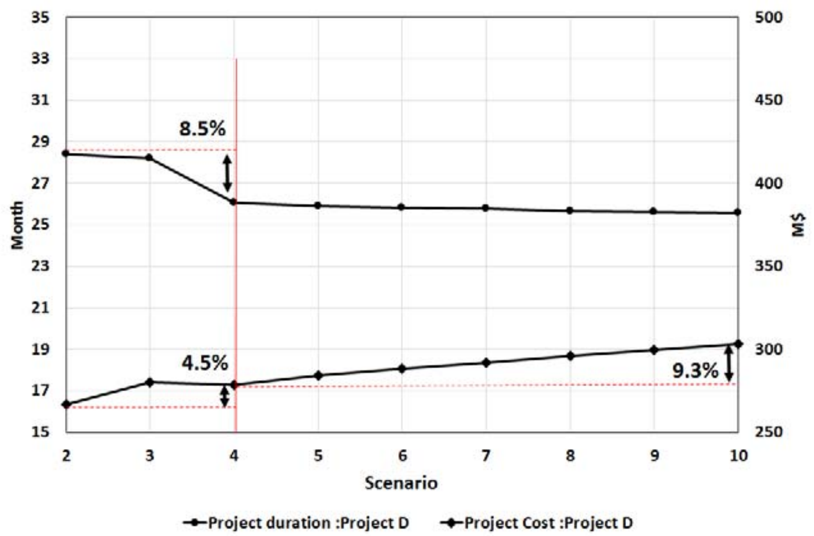


Figure 144: changes in the duration and cost of project D in different scenarios

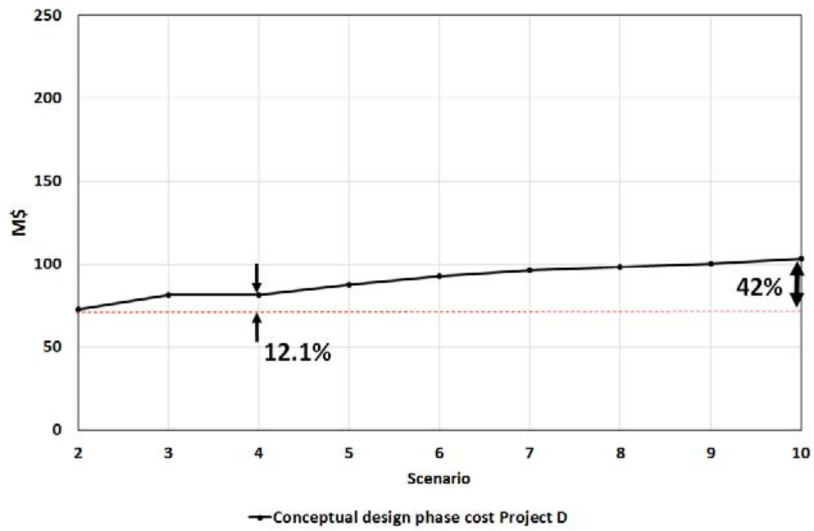


Figure 145: the percentage of increase in the cost of conceptual design phase between scenarios 2, 4 and 10 for project D

To have a conclusion about the effect of increasing the number of concepts on the performance of projects and a cross-project analysis, Figure 146 to Figure 148 represent all performance measures of different types of projects in single graphs. Figure 146 compares the effect of increasing the number of initial concepts on the total duration of different types of projects. As explained before and shown in Figure, project A shows the biggest reduction in the duration by increasing the number of concepts. This duration reduction in project A is accompanied by a reduction in the project cost, as shown in Figure 147. The main reason for the cost reduction as the result of increasing the number of concepts in project A in a big drop in the percentage of rework in downstream phases of this project. As the result, while more concepts mean allocating more resources to the upfront of the project, its effect of late reworks in project A is dominant and decreases the total project cost. In other projects, total cost increases as the result of increasing the number of concepts, which shows that in these projects the rework decrease could not cover the costs of increasing the number of concepts. In addition, these projects have an optimum number of concepts where exceeding them does not have any meaningful effect on the duration of these projects, while still increases the cost of projects. As explained before and shown also in Figure 146 and Figure 147, for projects B, C and D the optimum number of concepts in the conceptual design phase are seven, six, and four, respectively. In addition, Figure 148 shows that exceeding these optimum point does not have any effect on the quality of these projects.

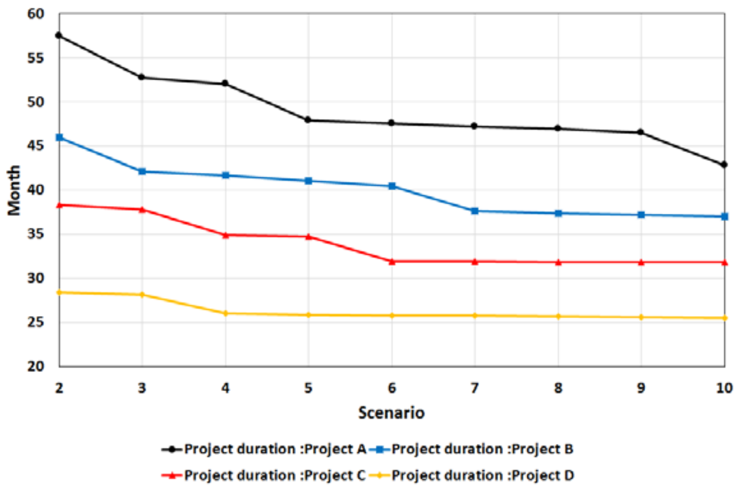


Figure 146: comparing the changes in projects duration in different scenarios

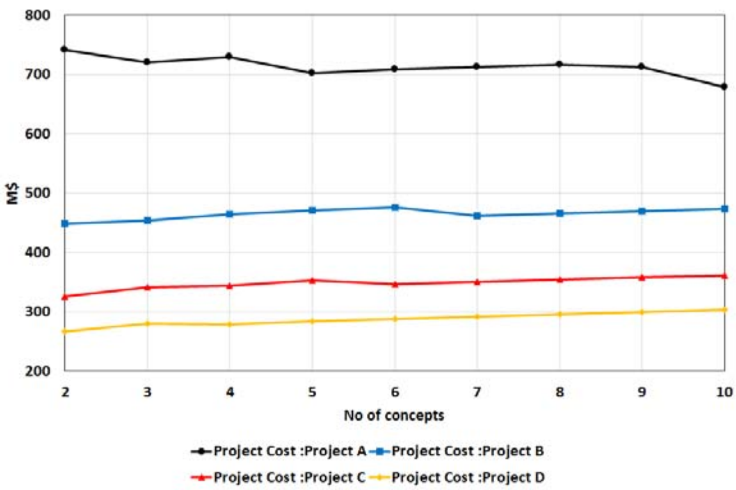


Figure 147: comparing the changes in projects cost in different scenarios

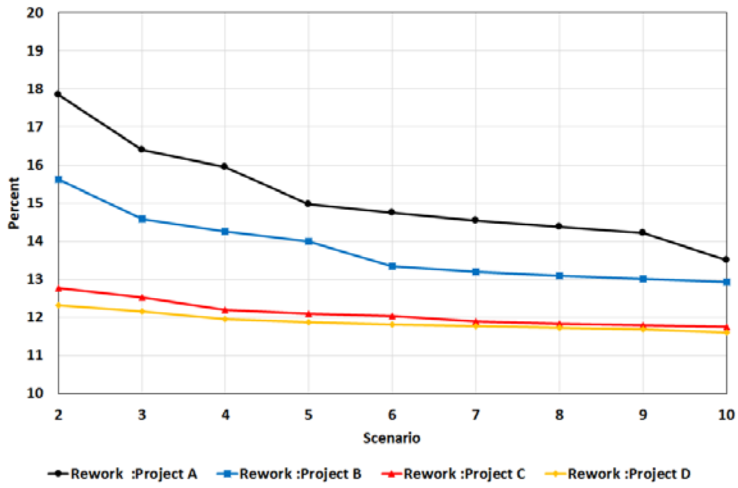


Figure 148: comparing the changes in projects rework in different scenarios

6.3 Summary

After building the system dynamics model of LPD in Chapter 4 and testing its validity in Chapter 5, in this Chapter the model is used for analysing different scenarios built based on the third objective of this research. The SBCE approach, as one of the unique features of LPD, includes frontloading the development project and delaying decisions, while working concurrently on sets of concepts and gradually eliminating the less feasible concepts, to allow the best concept to emerge at the end of the conceptual design phase. One of the variables which make the SBCE approach distinguishable from the PBCE approach is the duration of the conceptual design phase, including the period which all concepts are alive and design teams try to reduce the level of epistemic uncertainty about them by gathering more information and built up their knowledge through several iteration cycles, and the convergence period where teams are confident enough to identify and eliminate less feasible concepts. Another distinctive variable in the SBCE approach is the number of concepts which the design space at the start of the conceptual design phase is partitioned into. Two sets of scenarios are defined in this Chapter based on these variables. The model is run for the first set including seven scenarios with different number of iteration cycles in their conceptual design phase, and the model outputs, including the duration and cost of different phases, total project time and cost, as well as the total design confidence, and the percentage of rework during the project are compared. Based on the analysis, all types of projects show the improved performance metrics when moving from the PBCE approach towards the SBCE approach by increasing the number of iteration cycle. However, the degree of improvement for projects with higher levels of complexity is more profound.

The model is run again for the second set of scenarios, including 10 scenarios for each type of project with different number of initial concepts. The result of comparing the performance measures for different scenarios is the identification of the optimum number of concepts based on the trade-off between the time and cost measures. It is concluded for project A with the highest level of complexity that increasing the number of initial concepts to 10 has the positive effects on all project performance measures, while when the project complexity is lower, the optimum number of concepts is less. Nevertheless, based on the market situation and type of the product, still there are managers who could make the decision to compromise the project cost, or even quality to introduce the product sooner to the market.

7 Conclusion

7.1 Introduction

The main aim of conducting this research is to study the effect of implementation of Lean on the performance of NPD projects using a system dynamics model. The research is organized in three phases. In the first phase of the research, the theory of LPD is explored through an in-depth literature review. In the second phase, the system dynamics model of LPD is constructed based on principles found in literature in the first phase. The model is tested and validated using the historical data from a car manufacturing company. In the third phase of the research, the model is implemented for scenario analysis to compare the performance of NPD projects under different settings.

This chapter provides a summary about the key research findings. This shows the ability of the developed system dynamics model to offer an insight to decision makers about the superiority of LPD over traditional approaches to development projects. The limitations of the research are explained and the opportunities arising from the developed model are suggested for the future improvements and applications.

The chapter is divided into 4 sections: after this introduction, a summary of the research is provided in the next section. Section 7.4 provides a brief about the key findings of this research. In Section 7.4 the contribution of the research to the knowledge and practice is discussed. The Chapter finishes with Section 7.5 which mentions the research limitations and provides some suggestions for the future research.

7.2 Summary of research

This research is based on the idea that traditional approaches to NPD project management could no longer provide companies with enough ability to survive in the highly competitive market environment. Innovative approaches, such as LPD, has recently attracted the attention of researchers and practitioners. The idea of Lean thinking and Lean Management first introduced in the manufacturing environment, after years of extensive study on Japanese car manufacturing companies, especially Toyota, to find the reasons of their outperformance. Their findings which were published as books and journal papers in late 1980s and early 1990s (e.g. Ohno, 1988; e.g. Krafcik, 1988; Clark et al., 1987; Clark and Fujimoto, 1989a; Clark and Fujimoto, 1989b; Womack et al., 1990; Clark and Fujimoto, 1991; Wheelwright and Clark, 1992), revealed the huge performance gap between Japanese car manufacturers and their western counterparts, and introduced what today is called '*Lean Manufacturing*' as opposed to '*Mass production'*, or '*Fordism'*.

Lean Management has not been limited to the shop floor and soon, after showing its potentials, has been spread to other disciplines, inside and outside the manufacturing industry. Defining five principles of Lean Thinking by Womack and Jones (1996) was the starting point of this movement. However, to be adoptable in other disciplines, Lean management needs a set of applicable tools and techniques, in addition to its philosophical underpinnings. Because of the availability of these tools and techniques for lean manufacturing, extracted from Toyota Production System, many researchers and practitioners have tried to adopt them in other disciplines.

Similar to the manufacturing activities, the role of New Product Development in the outperformance of Japanese companies has also been emphasised by researchers such as Clark, Fujimoto, and Wheelwright, but Lean Product Development (LPD) is much younger than Lean manufacturing. Over almost two decades from the first appearance of Lean product development in research titles ('The difficult path to lean product development' by Karlsson and Åhlström, 1996), scholars have taken two parallel approaches to this topic. The first approach, as mentioned before, tries to extract tools and techniques from lean manufacturing, and apply them in NPD environment. The focus has been more on increasing the efficiency of NPD activities, by the identification and the elimination of several types of wastes in processes. The second approach emphasises more on the fundamental differences between manufacturing and NPD environments, and inadaptability of lean manufacturing toolset in NPD. The focus in this approach has been more on increasing the effectiveness of NPD processes by focusing on value generation, instead of waste elimination. Researchers have tried to identify tools and techniques of LPD, directly from Toyota product development System.

This dichotomy in approaches to LPD is the motivation for this research. The first research objective is defined as '*finding the components of lean applicable in NPD which result in improving the efficiency and effectiveness of development projects*'. To achieve this objective, in the first phase of the project, an in-depth literature review is conducted to have a better understanding about two approaches to LPD, and explain the concepts of efficiency and effectiveness in NPD projects. Based on the literature review results the ambiguity in the definition of LPD is more concerned with the lack of distinction between levels of lean implementation. Lean management is a philosophy which is applicable through its five principles as defined by Womack and Jones at the strategic level of organizations. However, moving to the operational level needs defining unique tools and techniques for each different environment. In NPD processes, although the emphasis is more on value generation, achieving a continuous value stream is not possible without attacking the sources of wasteful activities. Among the lean components extracted from TPDS, and identified by researchers, the SBCE approach and Concurrent Engineering are unique LPD process-specific components which are selected for the further study in this research. To make a better understanding about the waste and value in NPD, distinguishing between iteration cycles which are the main sources of creating value in development projects, especially at the earlier stages of the project, and rework cycles which are wasteful and reduce the process efficiency, is necessary. Iteration cycles in the front-end of NPD projects are attributable to the SBCE approach. This approach not only increases the effectiveness of projects through frontloading, but also increases the efficiency by reducing the late rework cycles. On the other hand, the success in applying concurrent engineering, with the goal of increasing the efficiency of projects, is not achievable in the presence of these rework cycles. So, there is an interdependency between the SBCE and concurrent engineering approaches, which has not been studied so far, so investigating the combined effect of SBCE and concurrent engineering on the amount of iteration and rework cycles, and consequently, on the performance NPD projects is identified as a gap in the literature to be investigated further in this research.

In the Research Methodology Chapter, it is discussed that comparing the SBCE approach with traditional PBCE approach, and finding the underlying structure which governs the relationship between the SBCE approach, concurrent engineering, and the performance of NPD projects is not possible with methods such as case study. In addition, the relevancy between the research objectives and the Systems Thinking approach, which emphasises on the distinction between events, patterns of events, structures, and mental models, is discussed. The modelling as a research method, in general, and specifically, the system dynamics modelling, as a quantifiable subcategory of systems thinking approach, are explained as the method adopted for this research. The qualitative and quantitative stages of system dynamics modelling, and activities in these stages, including problem articulation and boundary selection, defining the dynamic hypothesis, formulation and testing of the model, and policy formulation and evaluation are discussed. Finally, the research design, incorporating three phases of exploration, development, and implementation, in line with the objectives of the research, and the methods for data gathering, including the literature review and historical project data are explained.

As mentioned before, it is not clear in the literature why implementing LPD could result in higher performance in new product development projects. In other words, although most researches in this field have justified the advantage of LPD using the Toyota performance, there are a few attempts to find its underlying reasons, especially regarding the design of the processes in NPD projects using the SBCE and concurrent engineering approaches. So, the second objective of the research is defined as '*developing a system dynamics model to investigate the impact of LPD on the performance of development projects'*. To achieve this objective in the Model Construction Chapter, first through a qualitative approach, the model boundary is defined and a causal loop diagram is built which describes the dynamic hypothesis of the model. In the causal loop diagram, the schedule pressure loop is representing the traditional PBCE approach in managing the NPD projects, while the SBCE loop is representing the SBCE approach. In addition, the concurrency loop and the rework loop are defined which have positive and negative effects on the project performance, respectively. The last loop is the corruption loop, which is activated as the result of parallel effect of the schedule pressure and the concurrency loops, and deteriorates the performance of the project.

The quantification of the model in the form of a stock-and-flow model is discussed in the quantitative stage of the modelling. Three sectors of the model, namely the workflow sector, the resource management sector, and the performance measurement sector are explained with their structure, and the mathematical equations governing the relationships between the variables in the model. The relationship between the number of iteration cycles in the conceptual design phase of the project and the rate of confidence increase and the rework probability in the downstream phase are formulated and connected to the project duration, project cost and the rework percentage as the performance metrics of the project.

In the Model Validation Chapter, the standard system dynamics tests for checking the validity of the model are applied. For the structural validity of the model, the boundary adequacy test, the dimensional consistency test and the structure assessment test are used. For the behaviour validity, the model is calibrated using historical data from a car-manufacturing company. The reason is to make the fit between the model outcomes and the real-world data, by estimating the values of unknown constants in the model. It is done using an optimization process, by formulating an objective function, and optimization parameters, and using the built-in optimization capability of Vensim® software. The final results show a good fit between the model outcomes and the real project performance metrics, which completes the development phase of the research.

To analyse the impact of LPD on the performance of development projects, it is needed to use the validated model for analysing different policies in the management of NPD projects. The implementation phase of the research is defined in response to the third research objective, which is '*Using the model for policy analysis*'. To achieve this objective, in the Analysis and Discussion Chapter, two sets of scenarios are defined which cover the number of initial concepts, and the duration of the conceptual design phase, as two main variables related to the SBCE approach. The model is run based of these different scenarios and across different types of projects, and several sets of data are gathered in the form of the project performance measures for each scenario. The key research findings, resulted from comparing these data sets, are summarized in the next section.

7.3 Revisiting research questions

RQ1: What is Lean Product Development?

The research provides a unique classification of constructs, tools and techniques of LPD, found in literature, which provides a better understanding of this concept. Lean journey is evolutionary by its nature, which seeks improvement through a gentle but continuous stream of changes. The first step in each study related to lean should be defining the specific area in which lean is needed to be applied, and finding the appropriate approaches and practices based on that understanding. The main reason for misunderstandings in lean transformation programs is the failure to distinction between levels of lean implementation. Two levels for lean transformation programs are introduced in literature review chapter; the *strategic level* including the customer-centred strategic thinking, which is universal and applicable everywhere, and the *operational level* including tools and techniques and process improvement, which is more contingent to the environment. Strategic level refers to the implementation of lean at the enterprise level to transfer the organization to an integrated entity that efficiently creates value for its multiple stakeholders by employing lean principles and practices. Application and embodiment of lean principles throughout the enterprise needs supporting practices at the operational level.

Two major approaches to LPD are identified in literature review based on the adaptation of lean manufacturing tools and techniques for optimizing NPD processes, or extracting LPD specific tools and techniques from TPDS. It is discussed that the implementation of lean at the operational level in manufacturing and NPD requires different approaches to the processes, tools and techniques, and the same approach could not be used for the implementation of lean in these two different areas. In addition, applying lean manufacturing principles to NPD processes, although may result in some short-term benefit, in the long-term produces inconsistencies due to fundamental differences between operations in manufacturing and NPD. Consequently, this research defined LPD based on TPDS, not an extension of lean manufacturing in NPD processes.

In addition, constructs, elements, and tools proposed by authors who used TPDS as their benchmark for LPD shows a combined focus on the operational and strategic levels of lean implementation. This mixture of various levels and practices of LPD makes them unclear and complex. Analysing 11 classes of elements and constructs in LPD shows that *Chief Engineer System*, *SBCE*, *Standardization*, and *Process Management* are fully operational, whereas *Knowledge Management*, *Responsibility-Based Planning*, *Strategic Management*, and *Supplier Involvement* are fully strategic. *Process Management* is only proposed by one researcher as an element of LPD. In addition, as the implementation of SBCE approach is not possible without having a standardized process in NPD, so it is presumed that studying SBCE approach covers the subject of process standardization. As the result the combination of Concurrent Engineering and Set-based approach to design is assumed to be the unique process-specific approach in LPD.

RQ2: What is the impact of Set-based approach to design and concurrent Engineering as elements of LPD process structure on the performance of development projects?

Building and validating a system dynamics model with some unique elements and formulations which makes a relationship between iteration and rework cycles, design confidence and the level of uncertainty, and the scenario analysis using this model provides some key findings related to the impact of LPD process design on the performance of NPD projects:

RQ2-1: How the duration of conceptual design phase affect the time, cost and percentage of reworks in different types of NPD projects?

Moving towards the SBCE approach by increasing the duration of the conceptual design phase, and adding to the number of iteration cycles in this phase shows that, independent of the type of the project, whether it is an incremental product improvement projects with the low degree of novelty, or projects involving breakthroughs in products or technology, implementation of the SBCE approach improves the project performance, across time, cost, and quality measures. However, the performance improvement as the result of implementation of the SBCE approach in more innovative projects, with the higher degree of the project complexity is more distinctive.

RQ2:2: How the number of initial concepts affect the time, cost and percentage of reworks in different types of NPD projects?

Changing the number of initial concepts in the SBCE approach shows that there is an optimum number of initial concepts for each type of projects which results in achieving the highest level of project performance. In addition, the optimum number of concepts increases with increasing the level of project complexity.

7.4 Research Contributions

The research presented in this thesis contributes to the knowledge in many ways. It is believed based on the author's awareness of literature related to LPD and the in-depth review carried out, that four key contributions are made.

- So far researchers have implemented different approaches to LPD. Different frameworks, constructs and tools have been introduced by researchers in the field of LPD which due to the ambiguity in the levels of lean implementation presented, especially between strategic and operational levels, do not give a common understanding about LPD. Using a comprehensive literature review, this research provides a unique framework to classify the elements, constructs, tools and techniques proposed for LPD. The framework helps researchers and practitioners to select appropriate tools and techniques for the implementation of Lean in new product development projects, based on their approach. In addition, the literature review mainly emphasizes that at the operational level of implementation, instead of trying to adopt tools and techniques from lean manufacturing, Toyota product development system should be the benchmark for LPD.

- From the methodology point of view, this research adds new features to the system dynamics model of development projects, by distinguishing between the iteration cycles as the value-adding activities and rework-cycles as wasteful activities in NPD. This new feature shows the increase in the number of iteration cycles during the conceptual design phase and its effect on reducing rework cycles during later phases of the project as the result of the SBCE approach, thus clarifying the positive impact of LPD on project performance measures.
- Another methodological contribution of the research is adding a first-order stock-and-flow structure to the model in order to control the progress of the project based on the number of iteration cycles and to relate it to the level of confidence which is built in the design. This structure especially helps to investigate the effect of higher concept confidence, as the result of SBCE and frontloading the projects, on the duration, cost and number of rework cycles in development projects.
- A unique structure is introduced in the model to show the process of narrowing down the design space and the elimination of less feasible concepts during the conceptual design phase of projects, when SCBE approach is implemented.
- This research provides a state-of-the-art analysis of the effect of the duration of the conceptual design phase, and the number of initial concepts, as the main variables of the SBCE approach, on the project performance. The model provides the possibility to change the number of initial concepts, and the duration of the conceptual design phase, and to compare the outcomes in the form of changes in the performance metrics of the project.
- Different levels of innovativeness and complexity could be defined in the model in order to provide a better understanding about the effect of LPD on different types of projects.
- Using the scenario analysis the model shows the positive effect of increasing the duration of conceptual design phase on the performance measures in all different types of projects, while increasing the number of concepts have an optimum point which exceeding it increases the cost of projects without improving the time performance.

- Finally, form the practice point of view, the model could be used as a tool by practitioners to make more informed decisions in designing NPD projects, based on the type of projects, and their levels of complexity.

7.5 Research Limitations and suggestions for future research

There are a number of limitations associated with this research, which could be addressed in the future to improve the work presented in this thesis:

- All information used to develop the causal-loop diagrams for the model in this research are extracted from the literature review. Another approach which is proposed for the future research is to conduct a series of interviews with project managers in order to gather up-to-date data about the contemporary issues in the management of the projects and to define more accurate dynamic hypotheses for the research.
- The level of aggregation defined for the modelling purpose in this research is a single NPD project, while in the real world situations, development teams in companies simultaneously work on different types of projects. In the future research, the model could be developed in a way which incorporate a number of projects with different levels of complexity.
- For validating the model, historical data related to four different projects in the car-manufacturing industry are used. The limited number of projects do not allow the model to be generalizable to other industries. It is recommended for the future research to have a combination of data related to different types of projects in different companies in order to increase the robustness of the model and the results of the scenario analysis.
- Engineering resources are scarce in companies, and different projects are competing to attract more resources. However, in the model, the availability of resources does not constrain the progress of the project. This assumption is in line with the purpose of the research which is focused on the effect of LPD process structure. However, for the future research it is recommended to investigate the combined effects of resource scarcity and process structure on the performance of NPD projects.
- The productivity of engineering resources is assumed constant in the model. However, it is evident that different factors such as the work pressure, working overtime, and approaching towards the project deadline could affect the

productivity of resources. For the future research it is recommended to add these factors into the model and to formulate the productivity as a dependent variable.

- It is recommended that the LPD model is tested through further implementation in industrial cases and defining action research. The model is expected to produce various results and such research will help to refine the model further.
- The degrees of concurrency between different phases of the project are defined as a constant in the model. It is recommended for the future research to investigate the effect of different degrees of concurrency between phases of an NPD project, in combination with different durations of the conceptual design phase for several types of projects to able to define a standard process structure for each class of projects.
- The variations in the corruption rate as the result of concurrency between phases of an NPD projects, and the impact of SBCE approach on this rate is investigated briefly in the model. Gathering more data could help to have a more detailed formulation for corruption rate and a better understanding about its effect of the project performance measures.
- The design confidence and target confidence are two variables introduced for the first time in this model to control the progress of the NPD project based on the perceived quality of the design. For the future research the idea could be developed more and detailed formulations for these variables could be developed, especially by gathering experts' opinions about these concepts and by interviewing project managers in the manufacturing industry.

References

INCOSE Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities. 2006. 3rd ed. San Diego, CA: International Council on Systems Engineering.

Abdel-Hamid, T.K., 1996. The slippery path to productivity improvement. *IEEE Transactions on Software Engineering,* 13 (4), 43-52.

Abdel-Hamid, T.K., 1989. The dynamics of software project staffing: a system dynamics based simulation approach. *IEEE Transactions on Software Engineering,* 15 (2), 109-119.

Abdel-Hamid, T.K., and Madnick, S.E., 1989. Lessons learned from modeling the dynamics of software development. *Communications of the ACM,* 32 (12), 1426-1438.

Ahmadi, R., and Wang, R.H., 1999. Managing development risk in product design processes. *Operations Research,* 47 (2), 235-246.

AitSahlia, F., Johnson, E. and Will, P., 1995. Is concurrent engineering always a sensible proposition? *IEEE Transactions on Engineering Management,* 42 (2), 166-170.

Al-Ashaab, A., Petritsch, C., Gourdin, M., Aliende Urrutia, U., Andino, A., Varro, B., Rigatti, C., Golob, M., Summers, M. and El-Nounu, A., 2013. Lean Product Development Performance Measurement Tool. *In: Proceedings of the 11th International Conference on Manufacturing Research (ICMR2013), 2013.*

Anand, G., and Kodali, R., 2008. Development of a Conceptual Framework for Lean New Product Development Process. *International Journal of Product Development,* 6 (2), 190-224.

Avigad, G., and Moshaiov, A., 2010. Set-based concept selection in multi-objective problems involving delayed decisions. *Journal of Engineering Design,* 21 (6), 619-646.

Baines, T., Lightfoot, H., Williams, G.M. and Greenough, R., 2006. State-of-the-art in lean design engineering: A literature review on white collar lean. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture,* 220 (9), 1538-1547.

Ballard, G., 2000. Positive vs negative iteration in design. *In: Eighth Annual Conference of the International Group for Lean Construction, IGLC-8, Brighton, UK,* pp. 17-25.

Barlas, Y., 1996. Formal aspects of model validity and validation in system dynamics. *System Dynamics Review,* 12 (3), 183-210.

Beauregard, Y., Bhuiyan, N. and Thomson, V., 2011. Post-Certification Engineering Taxonomy and Task Value Optimization in the Aerospace Industry. *EMJ - Engineering Management Journal,* 23 (1).

Belay, A.M., Welo, T. and Helo, P., 2014. Approaching lean product development using system dynamics: investigating front-load effects. *Advances in Manufacturing,* 2 (2), 130-140.

Benton, T., and Craib, I., 2010. *Philosophy of social science: The philosophical foundations of social thought.* 2nd ed. New York: Palgrave Macmillan.

Bernstein, J.I., 1998. *Design Methods in the Aerospace Industry: Looking for Evidence of Set-based Practices.* Masters., Massachusetts Institute of Technology.

Bertrand, J.W.M., and Fransoo, J.C., 2002. Operations management research methodologies using quantitative modeling. *International Journal of Operations & Production Management,* 22 (2), 241-264.

Bhuiyan, N., Gerwin, D. and Thomson, V., 2004. Simulation of the new product development process for performance improvement. *Management Science,* 50 (12), 1690-1703.

Blanche, M.T., Durrheim, K. and Painter, D., 2007. *Research in practice: Applied methods for the social sciences.* 2nd ed. Cape Town, South Africa: University of Cape Town Press.

Borshchev, A., and Filippov, A., 2004. From system dynamics and discrete event to practical agent based modeling: reasons, techniques, tools. *In: Proceedings of the 22nd international conference of the system dynamics society, Oxford, UK,* .

Bradley, S., Hax, A. and Magnanti, T., 1977. *Applied mathematical programming.* 1st ed. Addison Wesley.

Briggs, A.R., 2007. The use of modelling for theory building in qualitative analysis. *British Educational Research Journal,* 33 (4), 589-603.

Browning, T.R., 2003. On customer value and improvement in product development processes. *Systems Engineering,* 6 (1), 49-61.

Browning, T.R., Deyst, J.J., Eppinger, S.D. and Whitney, D.E., 2002. Adding value in product development by creating information and reducing risk. *IEEE Transactions on Engineering Management,* 49 (4), 443-458.

Browning, T.R., and Eppinger, S.D., 2002. Modeling impacts of process architecture on cost and schedule risk in product development. *IEEE Transactions on Engineering Management,* 49 (4), 428-442.

Browning, T.R., Fricke, E. and Negele, H., 2006. Key concepts in modeling product development processes. *Systems Engineering,* 9 (2), 104-128.

Browning, T.R., and Ramasesh, R.V., 2007. A Survey of Activity Network-Based Process Models for Managing Product Development Projects. *Production and Operations Management,* 16 (2), 217-240.

Browning, T.R., and Sanders, N.R., 2012. Can innovation be Lean? *California Management Review,* 54 (4), 5-19.

Burton, R.M., and Obel, B., 1995. The validity of computational models in organization science: From model realism to purpose of the model. *Computational & Mathematical Organization Theory,* 1 (1), 57-71.

Calantone, R.J., and Di Benedetto, C.A., 2000. Performance and time to market: accelerating cycle time with overlapping stages. *IEEE Transactions on Engineering Management,* 47 (2), 232-244.

Checkland, P., and Scholes, J., 1999. *Soft systems methodology: a 30-year retrospective.* 1st ed. Chichester: John Wiley.

Chen, J., Damanpour, F. and Reilly, R.R., 2010. Understanding antecedents of new product development speed: A meta-analysis. *Journal of Operations Management,* 28 (1), 17-33.

Chucholowski, N., Langer, S., Ferreira, M.G.G., Forcellini, F.A. and Maier, A., 2012. *Engineering change management report 2012: Survey results on causes and effects, current practice, problems, and strategies in Brazil.* Universidade Federal de Santa Catarina.

Clark, K.B., Chew, W.B. and Fujimoto, T., 1987. Product development in the world auto industry. *Brookings Papers on Economic Activity,* , 729-781.

Clark, K.B., and Fujimoto, T., 1991. *Product development performance: Strategy, organization, and management in the world auto industry.* First ed. Boston: Harvard Business School Press.

Clark, K.B., and Fujimoto, T., 1989a. Lead time in automobile product development explaining the Japanese advantage. *Journal of Engineering and Technology Management,* 6 (1), 25-58.

Clark, K.B., and Fujimoto, T., 1989b. Reducing the time to market: The case of the world auto industry. *Design Management Journal (Former Series),* 1 (1), 49-57.

Cooper, R.G., and Edgett, S.J., 2005. *Lean, rapid, and profitable new product development.* 1st ed. Canada: Product development institute.

Cooper, R.G., and Kleinschmidt, E.J., 2007. Winning businesses in product development: the critical success factors. *Research Technology Management,* 50 (3).

Costa, R., and Sobek, D.K., 2003. Iteration in engineering design: inherent and unavoidable or product of choices made? *In: Proceedings of DETC03 ASME 2003 design engineering* *technical conferences and computers and information in engineering conference,* American Society of Mechanical Engineers, pp. 669-674.

Cross, N., 2008. *Engineering design methods: strategies for product design.* 4th ed. Wiley & sons.

Cusumano, M.A., and Nobeoka, K., 1998. *Thinking beyond lean: how multi-project management is transforming product development at Toyota and other companies.* Simon and Schuster.

Cusumano, M.A., and Nobeoka, K., 1992. Strategy, structure and performance in product development. Observations from the auto industry. *Research Policy,* 21 (3), 265-265.

Dangerfield, B., 2014. Systems thinking and system dynamics: a primer. *In:* B. Dangerfield, S. Brailsford and L. Churilov, eds., *Discrete-event simulation and system dynamics for management decision making.* 1st ed. New York City: Wiley, 2014, pp. 26-51.

Denrell, J., 2004. Random walks and sustained competitive advantage. *Management Science,* 50 (7), 922-934.

Dieter, G.E., 2000. *Engineering Design: A Materials and Processing Approach.* New York, NY: McGraw Hill, Inc.

Doyle, J.K., and Ford, D.N., 1998. Mental models concepts for system dynamics research. *System Dynamics Review,* 14 (1), 3-29.

Dyer, J.H., and Nobeoka, K., 2000. Creating and managing a high-performance knowledge-sharing network: the Toyota case. *Strategic Management Journal,* 21 (3), 345-367.

Eckert, C.M., Clarkson, P.J. and Zanker, W., 2004. Change and customisation in complex engineering domains. *Research in Engineering Design,* 15 (1), 1-21.

Eisenhardt, K.M., and Tabrizi, B.N., 1995. Accelerating adaptive processes: Product innovation in the global computer industry. *Administrative Science Quarterly,* 40 (1), 84-110.

Ellis, G., 2016. Lean Product Development. *In:* G. Ellis, ed., *Project Management in Product Development.* Boston: Butterworth-Heinemann, 2016, pp. 177-222.

Engel, A., and Reich, Y., 2015. Advancing Architecture Options Theory: Six Industrial Case Studies. *Systems Engineering,* 4 (18), 396-414.

Ford, D.N., 1995. *The dynamics of project management: an investigation of the impacts of project process and coordination on performance.* PhD thesis., Massachusetts Institute of Technology.

Ford, D.N., and Sobek, D.K., 2005. Adapting real options to new product development by modeling the Second Toyota Paradox. *IEEE Transactions on Engineering Management,* 52 (2), 175-185.

Ford, D.N., and Sterman, J.D., 2003. Overcoming the 90% syndrome: Iteration management in concurrent development projects. *Concurrent Engineering,* 11 (3), 177-186.

Ford, D.N., and Sterman, J.D., 1998. Dynamic modeling of product development processes. *System Dynamics Review,* 14 (1), 31-68.

Forrester, J.W., 1994a. *A guide to learning system dynamics.* Cambridge, MA: MIT.

Forrester, J.W., 1994b. System dynamics, systems thinking, and soft OR. *System Dynamics Review,* 10 (2-3), 245-256.

Forrester, J.W., 1971. Counterintuitive behavior of social systems. *Theory and Decision,* 2 (2), 109-140.

Forrester, J.W., 1961. *Industrial Dynamics.* Cambridge, MA: MIT Press.

Gautam, N., and Singh, N., 2008. Lean product development: Maximizing the customer perceived value through design change (redesign). *International Journal of Production Economics,* 114 (1), 313-332.

Gorski, P.S., 2013. What is critical realism? And why should you care? *Contemporary Sociology: A Journal of Reviews,* 42 (5), 658-670.

Griffin, A., 1997. The effect of project and process characteristics on product development cycle time. *Journal of Marketing Research,* 34 (1), 24-35.

Grösser, S.N., and Schaffernicht, M., 2012. Mental models of dynamic systems: taking stock and looking ahead. *System Dynamics Review,* 28 (1), 46-68.

Größler, A., Thun, J. and Milling, P.M., 2008. System Dynamics as a Structural Theory in Operations Management. *Production & Operations Management,* 17 (3), 373-384.

Gudem, M., Steinert, M., Welo, T. and Leifer, L.J., 2011. Customer Value is Not a Number– Investigating the Value Concept in Lean Product Development. *In: Proceedings of the 18th International Conference on Engineering Design (ICED 11), Denmark, August 2011.*

Haque, B., and Moore, M.J., 2004. Applying lean thinking to new product introduction. *Journal of Engineering Design,* 15 (1), 1-31.

Haque, B., and Moore, M.J., 2002. Characteristics of lean product introduction. *International Journal of Automotive Technology and Management,* 2 (3), 378-401.

Harrison, J.R., Lin, Z., Carroll, G.R. and Carley, K.M., 2007. Simulation modeling in organizational and management research. *Academy of Management Review,* 32 (4), 1229-1245.

Hart, C., 1998. *Doing a literature review: Releasing the social science research imagination.* 1st ed. London, UK: Sage.

Harter, D.E., Krishnan, M.S. and Slaughter, S.A., 2000. Effects of Process Maturity on Quality, Cycle Time, and Effort in Software Product Development. *Management Science,* 46 (4), 451.

Hines, P., Found, P., Griffiths, G. and Harrison, J.R., 2011. *Staying Lean: thriving, not just surviving.* CRC Press.

Hines, P., Holweg, M. and Rich, N., 2004. Learning to evolve: A review of contemporary lean thinking. *International Journal of Operations & Production Management,* 24 (10), 994-1011.

Hines, P., and Taylor, D., 2000. *Going lean: a guide to implementation.* Lean Enterprise Research Centre Cardiff.

Holston, D., 2011. *The Strategic Designer: Tools & techniques for managing the design process.* 1st ed. Simon and Schuster.

Hoppmann, J., 2009. *The Lean Innovation Roadmap-A Systematic Approach to Introducing Lean in Product Development Processes and Establishing a Learning Organization.* Diploma Thesis., Supervisors Technical University of Braunschweig.

Hoppmann, J., Rebentisch, E., Dombrowski, U. and Zahn, T., 2011. A framework for organizing lean product development. *EMJ - Engineering Management Journal,* 23 (1), 3-15.

Howick, S., and Eden, C., 2001. The impact of disruption and delay when compressing large projects: going for incentives? *Journal of the Operational Research Society,* 52 (1), 26-34.

Iansiti, M., and MacCormack, A., 1997. Developing products on Internet time. *Harvard Business Review,* 75 (5), 108-117.

Inoue, M., Nahm, Y., Tanaka, K. and Ishikawa, H., 2013. Collaborative engineering among designers with different preferences: Application of the preference set-based design to the design problem of an automotive front-side frame. *Concurrent Engineering Research and Applications,* 21 (4), 252-267.

Jalili, Y., and Ford, D.N., 2016. Quantifying the impacts of rework, schedule pressure, and ripple effect loops on project schedule performance. *System Dynamics Review,* 32 (1), 82-96.

Joglekar, N.R., and Ford, D.N., 2005. Product development resource allocation with foresight. *European Journal of Operational Research,* 160 (1), 72-87.

Karlsson, C., and Åhlström, P., 1996. The difficult path to lean product development. *Journal of Product Innovation Management,* 13 (4), 283-295.

Kennedy, B.M., Sobek, D.K. and Kennedy, M.N., 2014. Reducing Rework by Applying Set-Based Practices Early in the Systems Engineering Process. *Systems Engineering,* 17 (3), 278-296.

Kerga, E., Rossi, M., Taisch, M. and Terzi, S., 2014. A serious game for introducing set-based concurrent engineering in industrial practices. *Concurrent Engineering,* 22 (4), 333-346.

Kerga, E., Schmid, R., Rebentisch, E. and Terzi, S., 2016. Modeling the benefits of frontloading and knowledge reuse in lean product development. *In: Proceedings of PICMET '16: Technology Management for Social Innovation, Portland, US, 2016.* pp. 2532-2542.

Khan, M.S., 2012. *The construction of a model for lean product development.* PhD., Cranfield University.

Khan, M.S., Al-Ashaab, A., Doultsinou, A., Shehab, E., Ewers, P. and Sulowski, R., 2011. Set-based concurrent engineering process within the LeanPPD environment. *In: 18th ISPE International Conference on Concurrent Engineering, CE 2011, Boston, MA, 4 July 2011 through 8 July 2011.* pp. 433-440.

Khan, M.S., Al-Ashaab, A., Shehab, E., Haque, B., Ewers, P., Sorli, M. and Sopelana, A., 2013. Towards lean product and process development. *International Journal of Computer Integrated Manufacturing,* 26 (12), 1105-1116.

Kotzab, H., Seuring, S., Müller, M. and Reiner, G., 2006. *Research methodologies in supply chain management.* 1st ed. Springer Science & Business Media.

Krafcik, J.F., 1988. Triumph of the lean production system. *MIT Sloan Management Review,* 30 (1), 41.

Krishnan, V., Eppinger, S.D. and Whitney, D.E., 1997. A model-based framework to overlap product development activities. *Management Science,* 43 (4), 437-451.

Krishnan, V., and Ulrich, K.T., 2001. Product development decisions: A review of the literature. *Management Science,* 47 (1), 1-21.

Lane, D.C., 2008. The emergence and use of diagramming in system dynamics: a critical account. *Systems Research and Behavioral Science,* 25 (1), 3-23.

Lane, D.C., 2001a. Rerum cognoscere causas: Part I—How do the ideas of system dynamics relate to traditional social theories and the voluntarism/determinism debate? *System Dynamics Review,* 17 (2), 97-118.

Lane, D.C., 2001b. Rerum cognoscere causas: Part II—Opportunities generated by the agency/structure debate and suggestions for clarifying the social theoretic position of system dynamics. *System Dynamics Review,* 17 (4), 293-309.

Law, A.M., and Kelton, W.D., 1991. *Simulation modeling and analysis.* 1st ed. McGraw-Hill New York.

Le, H.N., Wynn, D.C. and Clarkson, P.J., 2012. Impacts of concurrency, iteration, design review, and problem complexity on design project lead time and error generation. *Concurrent Engineering,* 20 (1), 55-67.

Le, H.N., Wynn, D.C. and Clarkson, P.J., 2010. Evaluating the positive and negative impact of iteration in engineering processes. *In:* Evaluating the positive and negative impact of iteration in engineering processes. *Modelling and Management of Engineering Processes.* First ed. London: Springer, 2010, pp. 89-100.

Letens, G., Farris, J.A. and van Aken, E.M., 2011. A Multilevel Framework for Lean Product Development System Design. *EMJ - Engineering Management Journal,* 23 (1), 69-85.

Levandowski, C., 2014. *Platform Lifecycle Support using Set-Based Concurrent Engineering.* PhD., Chalmers University of Technology.

Lévárdy, V., and Browning, T.R., 2009. An adaptive process model to support product development project management. *IEEE Transactions on Engineering Management,* 56 (4), 600-620.

Liker, J.K., 2004. *The Toyota Way: 14 Management Principles from the World's Greatest Manufacturer.* New York: McGrow-Hill.

Liker, J.K., and Morgan, J.M., 2011. *Lean Product Development as a System: A Case Study of Body and Stamping Development at Ford.* American Society for Engineering Management.

Liker, J.K., and Morgan, J.M., 2006. The Toyota Way in Services: The Case of Lean Product Development. *Academy of Management Perspectives,* 20 (2), 5-20.

Liker, J.K., Sobek, D.K., Ward, A.C. and Cristiano, J.J., 1996. Involving suppliers in product development in the United States and Japan: Evidence for set-based concurrent engineering. *IEEE Transactions on Engineering Management,* 43 (2), 165-178.

Lin, J., Chai, K.H., Brombacher, A.C. and San Wong, Y., 2009. Optimal overlapping and functional interaction in product development. *European Journal of Operational Research,* 196 (3), 1158-1169.

Lin, J., Chai, K.H., San Wong, Y. and Brombacher, A.C., 2008. A dynamic model for managing overlapped iterative product development. *European Journal of Operational Research,* 185 (1), 378-392.

Lin, J., Qian, Y., Cui, W. and Miao, Z., 2010. Overlapping and communication policies in product development. *European Journal of Operational Research,* 201 (3), 737-750.

Loch, C.H., Mihm, J. and Huchzermeier, A., 2003. Concurrent engineering and design oscillations in complex engineering projects. *Concurrent Engineering,* 11 (3), 187-199.

Loch, C.H., and Terwiesch, C., 1998. Communication and uncertainty in concurrent engineering. *Management Science,* 44 (8), 1032-1048.

Love, P.E., 2002. Influence of project type and procurement method on rework costs in building construction projects. *Journal of Construction Engineering and Management,* 128 (1), 18-29.

Love, P.E., and Edwards, D.J., 2004. Forensic project management: The underlying causes of rework in construction projects. *Civil Engineering and Environmental Systems,* 21 (3), 207-228.

Lyneis, J.M., and Ford, D.N., 2007. System dynamics applied to project management: a survey, assessment, and directions for future research. *System Dynamics Review,* 23 (2-3), 157-189.

Maani, K., and Cavana, R.Y., 2007. *Systems thinking, system dynamics: Managing change and complexity.* 2nd ed. Auckland, NZ: Prentice Hall.

MacCormack, A., Verganti, R. and Iansiti, M., 2001. Developing products on "Internet time": The anatomy of a flexible development process. *Management Science,* 47 (1), 133-150.

Machado, M.A., Ericsson, E. and Verghese, G., 2014. New Product Development effectiveness: A pathway to sustainable competitive advantage. *In: PICMET 2014 - Portland International Center for Management of Engineering and Technology, Proceedings: Infrastructure and Service Integration,* pp. 2111-2117.

Maier, J.F., Wynn, D.C., Biedermann, W., Lindemann, U. and Clarkson, P.J., 2014. Simulating progressive iteration, rework and change propagation to prioritise design tasks. *Research in Engineering Design,* 25 (4), 283-307.

Malak, R.J., Aughenbaugh, J.M. and Paredis, C.J., 2009. Multi-attribute utility analysis in set-based conceptual design. *Computer-Aided Design,* 41 (3), 214-227.

Mark, N.P., 2002. Cultural transmission, disproportionate prior exposure, and the evolution of cooperation. *American Sociological Review,* , 323-344.

Martínez León, H.C., and Farris, J.A., 2011. Lean Product Development Research: Current State and Future Directions. *EMJ - Engineering Management Journal,* 23 (1), 29-51.

Mashayekhi, A.N., and Ghili, S., 2012. System dynamics problem definition as an evolutionary process using the concept of ambiguity. *System Dynamics Review,* 28 (2), 182-198.

Mays, N., Roberts, E. and Popay, J., 2001. Synthesising research evidence. *Studying the Organisation and Delivery of Health Services: Research Methods,* , 188-220.

McHaney, R.W., 2009. *Understanding computer simulation.* Roger McHaney & Ventus Publishing.

McManus, H.L., 2005. *Product Development Value Stream Mapping (PDVSM) Manual Release 1.0.* Cambridge, MA: MIT, Lean Aerospace Initiative.

McManus, H.L., and Millard, R., 2002. Value stream analysis and mapping for product development. *In: Proceedings of the International Council of the Aeronautical Sciences, 23rd ICAS Congress, Toronto Canada,* pp. 6103.1-6103.10.

Meadows, D.H., 1989. System dynamics meets the press. *System Dynamics Review,* 5 (1), 69-80.

Meier, C., Browning, T.R., Yassine, A.A. and Walter, U., 2015. The Cost of Speed: Work Policies for Crashing and Overlapping in Product Development Projects. *IEEE Transactions on Engineering Management,* 62 (2), 237-255.

Meier, C., Yassine, A.A. and Browning, T.R., 2007. Design process sequencing with competent genetic algorithms. *Journal of Mechanical Design,* 129 (6), 566-585.

Meredith, J.R., Raturi, A., Amoako-Gyampah, K. and Kaplan, B., 1989. Alternative research paradigms in operations. *Journal of Operations Management,* 8 (4), 297-326.

Mingers, J., 2014. *Systems thinking, critical realism and philosophy: A confluence of ideas.* 1st ed. Routledge.

Mingers, J., 2000. The Contribution of Critical Realism as an Underpinning Philosophy for OR/MS and Systems. *The Journal of the Operational Research Society,* 51 (11), 1256-1270.

Mingers, J., and White, L., 2010. A review of the recent contribution of systems thinking to operational research and management science. *European Journal of Operational Research,* 207 (3), 1147-1161.

Morecroft, J.D., 2015. *Strategic modelling and business dynamics: A feedback systems approach.* John Wiley & Sons.

Morgan, J.M., and Liker, J.K., 2006. *The Toyota product development system: integrating people, process, and technology.* New York: Productivity Press.

Murman, E.M., Allen, T., Bozdogan, K., Cutcher-Gershenfeld, J., McManus, H.L., Nightingale, D., Rebentisch, E., Shields, T., Stahl, F. and Walton, M., 2002. *Lean enterprise value : insights from MIT's Lean Aerospace Initiative.* New York: Palgrave.

Nahm, Y., and Ishikawa, H., 2006. Novel space-based design methodology for preliminary engineering design. *The International Journal of Advanced Manufacturing Technology,* 28 (11-12), 1056-1070.

Nahm, Y., and Ishikawa, H., 2005. Representing and aggregating engineering quantities with preference structure for set-based concurrent engineering. *Concurrent Engineering,* 13 (2), 123-133.

Neely, A., Gregory, M. and Platts, K., 2005. Performance measurement system design: A literature review and research agenda. *International Journal of Operations & Production Management,* 25 (12), 1228-1263.

Nelson, R.R., and Winter, S.G., 1982. *An evolutionary theory of economic change.* 1st ed. Cambridge: Harvard University Press.

Nelson, R.G., Azaron, A. and Aref, S., 2016. The use of a GERT based method to model concurrent product development processes. *European Journal of Operational Research,* 250 (2), 566-578.

Nepal, B.P., Yadav, O.P. and Solanki, R., 2011. *Improving the NPD Process by Applying Lean Principles: A Case Study.* American Society for Engineering Management.

Nonaka, I., and Takeuchi, H., 1996. The Knowledge-creating Company: how Japanese companies create the Dynamics of Innovation. *Long Range Planning,* 29 (4), 592.

Oehmen, J., and Rebentisch, E., 2010. *Waste in Lean Product Development.* Cambridge: Massachusetts Institute of Technology.

Ohno, T., 1988. *Toyota production system: beyond large-scale production.* Productivity press.

Olaru, D., Purchase, S. and Peterson, N., 2008. From customer value to repurchase intentions and recommendations. *Journal of Business & Industrial Marketing,* 23 (8), 554-565.

Olaya, C., 2009. System dynamics philosophical background and underpinnings. *In:* System dynamics philosophical background and underpinnings. *Complex Systems in Finance and Econometrics.* Springer, 2009, pp. 812-832.

Oosterwal, D.P., 2010. *The lean machine: how Harley-Davidson drove top-line growth and profitability with revolutionary lean product development.* 1st ed. NewYork, USA: AMACOM.

Oppenheim, B.W., 2004. Lean product development flow. *Systems Engineering,* 7 (4), 352-376.

Ottosson, S., 2004. Dealing with innovation push and market need. *Technovation,* 24 (4), 279-285.

Pahl, G., and Beitz, W., 2007. *Engineering design: a systematic approach.* 3rd ed. Springer.

Park, M., and Peña-Mora, F., 2004. Reliability buffering for construction projects. *Journal of Construction Engineering and Management,* 130 (5), 626-637.

Parvan, K., Rahmandad, H. and Haghani, A., 2015. Inter-phase feedbacks in construction projects. *Journal of Operations Management,* 39, 48-62.

Pidd, M., 1996. *Tools for thinking: Modelling in management science.* 1st ed. Chichester: John Wiley & Sons.

Prasad, B., 1996. *Concurrent engineering fundamentals.* 1st ed. NJ: Prentice Hall Englewood Cliffs.

Pugh, S., 1991. *Total design: integrated methods for successful product engineering.* 1st ed. Addison-Wesley.

Qudrat-Ullah, H., and Seong, B.S., 2010. How to do structural validity of a system dynamics type simulation model: the case of an energy policy model. *Energy Policy,* 38 (5), 2216-2224.

Qureshi, A.J., Dantan, J., Bruyere, J. and Bigot, R., 2010. Set based robust design of mechanical systems using the quantifier constraint satisfaction algorithm. *Engineering Applications of Artificial Intelligence,* 23 (7), 1173-1186.

Radzicki, M.J., 2009. System dynamics and its contribution to economics and economic modeling. *In:* System dynamics and its contribution to economics and economic modeling. *Complex Systems in Finance and Econometrics.* Springer, 2009, pp. 727-737.

Ragsdale, C.T., 2008. *Spreadsheet modeling and decision analysis.* 5th ed. USA: Thomson South-Western,.

Rahmandad, H., and Hu, K., 2010. Modeling the rework cycle: capturing multiple defects per task. *System Dynamics Review,* 26 (4), 291-315.

Rahmandad, H., 2015. Connecting strategy and system dynamics: an example and lessons learned. *System Dynamics Review,* 31 (3), 149-172.

Raudberget, D., 2010. Practical applications of set-based concurrent engineering in industry. *Journal of Mechanical Engineering,* 56 (11), 685-695.

Reichelt, K., and Lyneis, J.M., 1999. The dynamics of project performance: benchmarking the drivers of cost and schedule overrun. *European Management Journal,* 17 (2), 135-150.

Reinertsen, D.G., 2009. *The principles of product development flow: second generation lean product development.* Celeritas Redondo Beach,, Canada.

Reinertsen, D.G., 2007. Rethinking lean NPD: A distorted view of lean product development. *Strategic Direction,* 23 (10), 32-34.

Reinertsen, D.G., 2005. Let it flow. *Industrial Engineer,* 37 (6), 40-45.

Reinertsen, D.G., 1999. Lean thinking isn’t so simple. *Electronic Design,* 47 (10), 48.

Repenning, N.P., Gonçalves, P. and Black, L.J., 2001. Past the tipping point: the persistence of firefighting in product development. *California Management Review,* 43 (4), 44-63.

Riaz, A., 2015. *A Set-Based Approach to Passenger Aircraft Family Design.* PhD., Cranfield University.

Richardson, G.P., 2009. System dynamics, the basic elements of. *In:* System dynamics, the basic elements of. *Complex Systems in Finance and Econometrics.* Springer, 2009, pp. 856-862.

Richardson, G.P., and Pugh III, A.I., 1981. *Introduction to system dynamics modeling with DYNAMO.* Productivity Press Inc.

Robertson, D., and Ulrich, K.T., 1998. Planning for product platforms. *MIT Sloan Management Review,* 39 (4).

Robinson, S., 2014. *Simulation: the practice of model development and use.* 1st ed. Palgrave Macmillan.

Robson, C., 2011. *Real World Research: a resource for users of social research methods in applied settings.* 1st ed. Chichester: Wiley.

Roemer, T.A., and Ahmadi, R., 2004. Concurrent crashing and overlapping in product development. *Operations Research,* 52 (4), 606-622.

Roemer, T.A., Ahmadi, R. and Wang, R.H., 2000. Time-cost trade-offs in overlapped product development. *Operations Research,* 48 (6), 858-865.

Saunders, M., Lewis, P. and Thornhill, A., 2009. *Research methods for business students.* Fifth ed. Harlow: FT Prentice Hall, Pearson Education.

Schmidt, E., Kasperek, D. and Maurer, M., 2015. A Guideline for Adapted System Dynamics Modeling of Rework Cycles in Engineering Design Processes. *In: Proceeding of the 22nd ISPE inc. International Conference On Concurrent Engineering, Delft, Netherlands, July 20-23.* pp. 357-366.

Senge, P.M., 1992. *The fifth discipline: The art and practice of the learning organization.* 1st ed. London: Century Business.

Shenhar, A.J., 2001. One size does not fit all projects: Exploring classical contingency domains. *Management Science,* 47 (3), 394-414.

Siyam, G.I., Kirner, K.G., Wynn, D.C., Lindemann, U. and Clarkson, P.J., 2012. Value and waste dependencies and guidelines. *In: DSM 2012 Proceedings of the 14th International DSM Conference, Kyoto, Japan, September 2012.* pp. 65-78.

Siyam, G.I., Wynn, D.C. and Clarkson, P.J., 2015. Review of Value and Lean in Complex Product Development. *Systems Engineering,* 18 (2), 192-207.

Smith, P.G., and Reinertsen, D.G., 1998. *Developing products in half the time: new rules, new tools.* 2nd ed. New York: John Wiley & Sons.

Smith, R.P., and Eppinger, S.D., 1997a. Identifying controlling features of engineering design iteration. *Management Science,* 43 (3), 276-293.

Smith, R.P., and Eppinger, S.D., 1997b. A predictive model of sequential iteration in engineering design. *Management Science,* 43 (8), 1104-1120.

Sobek, D.K., Liker, J.K. and Ward, A.C., 1998. Another look at how Toyota integrates product development. *IEEE Engineering Management Review,* 26 (4), 69-77.

Sobek, D.K., Ward, A.C. and Liker, J.K., 1999. Toyota's principles of set-based concurrent engineering. *MIT Sloan Management Review,* 40 (2), 67-84.

Spiegler, V.L., 2013. *Designing supply chains resilient to nonlinear system dynamics.* PhD., Cardiff University.

Sterman, J.D., 2000. *Business Dynamics: Systems Thinking and Modeling for a Complex World.* 1st ed. Boston: McGrow Hill Higher Education.

Suss, S., and Thomson, V., 2012. Optimal design processes under uncertainty and reciprocal dependency. *Journal of Engineering Design,* 23 (10-11), 829-851.

Swink, M.L., Talluri, S. and Pandejpong, T., 2006. Faster, better, cheaper: a study of NPD project efficiency and performance tradeoffs. *Journal of Operations Management,* 24 (5), 542-562.

Tako, A.A., and Robinson, S., 2012. The application of discrete event simulation and system dynamics in the logistics and supply chain context. *Decision Support Systems,* 52 (4), 802-815.

Tangen, S., 2005. Demystifying productivity and performance. *International Journal of Productivity and Performance Management,* 54 (1), 34-46.

Taylor, T.R., and Ford, D.N., 2008. Managing tipping point dynamics in complex construction projects. *Journal of Construction Engineering and Management,* 134 (6), 421-431.

Taylor, T.R., and Ford, D.N., 2006. Tipping point failure and robustness in single development projects. *System Dynamics Review,* 22 (1), 51-71.

Telerman, V., Preis, S., Snytnikov, N. and Ushakov, D., 2006. Interval/set based collaborative engineering design. *International Journal of Product Lifecycle Management,* 1 (2), 150-163.

Terwiesch, C., and Loch, C.H., 1999. Measuring the effectiveness of overlapping development activities. *Management Science,* 45 (4), 455-465.

Terwiesch, C., Loch, C.H. and Meyer, A.D., 2002. Exchanging preliminary information in concurrent engineering: Alternative coordination strategies. *Organization Science,* 13 (4), 402-419.

Thomas, A.B., 2004. *Research skills for management studies.* 1st ed. London: Psychology Press.

Todd, P.M., 2001. Fast and frugal heuristics for environmentally bounded minds. *In: Bounded rationality: The adaptive toolbox: Dahlem Workshop Report,* MIT Press, pp. 51-70.

Ulrich, K.T., and Eppinger, S.D., 2016. *Product design and development.* 6th ed. New York: McGraw-Hill.

van der Krogt, R., Geraghty, J., Salman, M.R. and Little, J., 2010. On supporting Lean methodologies using constraint-based scheduling. *Journal of Scheduling,* 13 (4), 301-314.

Wang, L., Ming, X.G., Kong, F.B., Li, D. and Wang, P.P., 2012. Focus on implementation: a framework for lean product development. *Journal of Manufacturing Technology Management,* 23 (1), 4-24.

Warburton, R.D., and Disney, S.M., 2007. Order and inventory variance amplification: The equivalence of discrete and continuous time analyses. *International Journal of Production Economics,* 110 (1), 128-137.

Ward, A.C., 2007. *Lean product and process development.* Cambridge: Lean Enterprise Institute.

Ward, A.C., Liker, J.K., Cristiano, J.J. and Sobek, D.K., 1995a. The Second Toyota Paradox: How Delaying Decisions Can Make Better Cars Faster. *MIT Sloan Management Review,* 36 (3), 43-61.

Ward, A.C., Sobek, D.K., Christiano, J.J. and Liker, J.K., 1995b. Toyota, concurrent engineering, and set-based design. *In:* Toyota, concurrent engineering, and set-based design. *Engineered in Japan: Japanese technology- management practice.* First ed. Oxford University Press, 1995b, pp. 192-216.

Welo, T., and Ringen, G., 2016. Beyond waste elimination: Assessing lean practices in product development. *In: 26th CIRP Design Conference,* Elsevier, pp. 179-185.

Wheelwright, S.C., and Clark, K.B., 1992. *Revolutionizing product development: quantum leaps in speed, efficiency, and quality.* 1st ed. New York: The free press.

Williams, T., 1999. Seeking optimum project duration extensions. *Journal of the Operational Research Society,* 50 (5), 460-467.

Wolstenholme, E.F., 1993. A case study in community care using systems thinking. *Journal of the Operational Research Society,* 44 (9), 925-934.

Womack, J.P., and Jones, D.T., 1996. *Lean Thinking: Banish Waste And Create Wealth In Your Corporation.* New York: New York : Simon & Schuster.

Womack, J.P., Jones, D.T. and Roos, D., 1990. *The machine that Changed the World.* 1st ed. New York: Simon and Schuster.

Wu, D.D., Kefan, X., Gang, C. and Ping, G., 2010. A Risk Analysis Model in Concurrent Engineering Product Development. *Risk Analysis,* 30 (9), 1440-1453.

Wynn, D.C., 2007. *Model-based approaches to support process improvement in complex product development.* PhD., University of Cambridge.

Wynn, D.C., and Eckert, C.M., 2016. Perspectives on iteration in design and development. *Research in Engineering Design,* , 1-32.

Yan, H.S., Wang, B., Xu, D. and Wang, Z., 2010. Computing Completion Time and Optimal Scheduling of Design Activities in Concurrent Product Development Process. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans,* 40 (1), 76-89.

Yassine, A.A., and Braha, D., 2003. Complex concurrent engineering and the design structure matrix method. *Concurrent Engineering,* 11 (3), 165-176.

Yassine, A.A., Joglekar, N.R., Braha, D., Eppinger, S.D. and Whitney, D.E., 2003. Information hiding in product development: the design churn effect. *Research in Engineering Design,* 14 (3), 145-161.

Yazdani, B., 2000. *A comparative study of design definition models and product development performance in the automobile industry.* PhD thesis., University of Warwick.

Zeithaml, V.A., 1988. Consumer perceptions of price, quality, and value: a means-end model and synthesis of evidence. *Journal of Marketing,* 52 (3), 2-22.

Zhang, D., and Bhuiyan, N., 2015. A Study of the Evolution of Uncertainty in Product Development as a Basis for Overlapping. *IEEE Transactions on Engineering Management,* 62 (1), 39-50.

Appendix

Model variables

Abbreviation	L	F	DE	LI	C	A	D	SUB
Meaning	Level (stock)	Flow	Delay	Level initial	Constant	Auxiliary	Data	Subscript

Type	Variable Name and Description
#1 F,A	**Add resource rate** (Person/Month) Add resource rate [Phase,Project] = Add resource switch*IF THEN ELSE(Total desired resources[Phase,Project]>Total allocated resources[Phase,Project]:AND:Schedule pressure[Phase,Project]>1, Available resources[Phase,Project]/TIME STEP, 0) **Present in 1 view:** • Auxiliaries **Used by:** • Available resources
#2 C	**Add resource switch** (Dmnl [0,1,1]) = 0 **Description:** *If zero, schedule pressure does not have any effect of the number of resources available to be allocated to different phases of the project* **Present in 2 views:** • Parameters • Auxiliaries **Used by:** • Add resource rate
#3 A	**Allocated resources- Completion** (Person) Allocated resources- Completion [Phase,Project] = "Desired resources-Completion"[Phase,Project]*MIN(1,ZIDZ(Available resources[Phase,Project],Total desired resources[Phase,Project])) **Present in 1 view:** • Resource management **Used by:** • Completion capacity • Total allocated resources
#4 A	**Allocated resources- Rework** (Person) Allocated resources- Rework [Phase,Project] = "Desired resources-Rework"[Phase,Project]*MIN(1,ZIDZ(Available resources[Phase,Project],Total desired resources[Phase,Project])) **Present in 1 view:** • Resource management **Used by:** • Rework capacity

	• Total allocated resources
#5 A	**Allocated resources- Test** (Person) Allocated resources- Test [Phase,Project] = "Desired resources-Test"[Phase,Project]*MIN(1,ZIDZ(Available resources[Phase,Project],Total desired resources[Phase,Project])) **Present in 1 view:** • Resource management **Used by:** • Test capacity • Total allocated resources
#6 LI,F,A	**Approve rate** (Task/Month) Approve rate [Phase,Project] = MIN((1-Rework probability[Phase,Project])*Tasks pending test[Phase,Project]/Test duration[Phase,Project],Test capacity[Phase,Project]) **Present in 2 views:** • Workflow • Performance measures **Used by:** • Tasks pending decision • Tasks pending test • Total activities per phase
#7 L	**Available resources** (Person) Available resources [Phase,Project] = $\int$ Add resource rate[Phase,Project] *dt* + [Initial available resources[Phase,Project]] **Description:** *The level of concurrency between phases.Note: Concurrency[Phase] is between the phase and its donstream phase, for example, Concurrency [Conceptual design] is between conceptual design and detail design phases.* **Present in 2 views:** • Resource management • Auxiliaries **Used by:** • Add resource rate • Allocated resources- Completion • Allocated resources- Rework • Allocated resources- Test
#8 F,A	**Change in phase time** (Dmnl) Change in phase time [Phase,Project] = IF THEN ELSE(Tasks released[Phase,Project]+Sets reduced[Phase,Project]>=Total phase tasks[Phase,Project]*Completion definition:AND:Phase finish time[Phase,Project]=0,Time/TIME STEP,0) **Present in 2 views:** • Parameters • Performance measures **Used by:** • Phase finish time

#9 F,A	**Change in project time** (Dmnl) Change in project time [Project] = IF THEN ELSE(Phase finish time[Conceptual design,Project]<>0:AND:Phase finish time[Detail design,Project]<>0:AND:Phase finish time[Tooling,Project]<>0:AND:Project duration Model[Project]=0, Time/TIME STEP, 0) **Present in 2 views:** • Parameters • Performance measures **Used by:** • Project duration Model
#10 A	**Completion capacity** (Task/Month) Completion capacity [Phase,Project] = "Allocated resources-Completion"[Phase,Project]*Resource productivity[Phase] **Present in 2 views:** • Workflow • Resource management **Used by:** • Completion rate
#11 C	**Completion definition** (Dmnl) = 0.99 **Present in 2 views:** • Performance measures • Auxiliaries **Used by:** • Change in phase time • Concurrency
#12 A	**Completion duration** (Month) Completion duration [Phase,Project] = Initial completion duration[Phase,Project] **Description:** *Changes in completion time due to the learning effect* **Present in 3 views:** • Workflow • Resource management • Auxiliaries **Used by:** • Completion rate • Desired resources- Completion • Rework duration • Test duration
#13 LI,F,A	**Completion rate** (Task/Month) Completion rate [Conceptual design,Project] = IF THEN ELSE(Time>=Phase start time Actual[Conceptual design,Project], MIN(Tasks not completed[Conceptual design,Project]/Completion duration[Conceptual design,Project],Completion capacity[Conceptual design,Project]), 0) Completion rate [Detail design,Project] = IF THEN ELSE(Time>=Phase start time Actual[Detail design,Project]:OR:Phase finish time[Conceptual design,Project]<>0, MIN(Tasks not completed[Detail

	design,Project]/Completion duration[Detail design,Project],Completion capacity[Detail design,Project]), 0) Completion rate [Tooling,Project] = IF THEN ELSE(Time>=Phase start time Actual[Tooling,Project]:OR:Phase finish time[Detail design,Project]<>0,MIN(Tasks not completed[Tooling,Project]/Completion duration[Tooling,Project],Completion capacity[Tooling,Project]), 0) **Present in 2 views:** • Workflow • Performance measures **Used by:** • Tasks not completed • Tasks pending test • Total activities per phase
#14 A	**Concurrency** (Dmnl [0,1]) Concurrency [Conceptual design,Project] = IF THEN ELSE(Tasks released[Conceptual design,Project]+Sets reduced[Conceptual design,Project]<=Total phase tasks[Conceptual design,Project]*Completion definition:AND:Time>=Phase start time Actual[Detail design,Project], 1, 0) Concurrency [Detail design,Project] = IF THEN ELSE(Tasks released[Detail design,Project]<=Total phase tasks[Detail design,Project]*Completion definition:AND:Time>=Phase start time Actual[Tooling,Project], 1, 0) **Description:** *The level of overlap between phases. Note: Concurrency[Phase] is between the phase and its downstream phase, exp. Concurrency [Conceptual design] is between conceptual design and detail design phases* **Present in 1 view:** • Auxiliaries **Used by:** • Corruption probability
#15 F,A	**Confidence increase rate** (1/Month) Confidence increase rate [Conceptual design,Project] = IF THEN ELSE(Confidence increase switch[Conceptual design,Project]=1,(FC*No of iterations[Conceptual design,Project]^PI*No of concepts[Conceptual design,Project]^PC *Project Complexity[Project]^PS/ TIME STEP), 0) Confidence increase rate [Detail design,Project] = IF THEN ELSE(Confidence increase switch[Detail design,Project]=1,(FC*No of iterations[Detail design,Project]^PI*MIN(1,Design confidence[Conceptual design,Project])^(PD) *Project Complexity[Project]^PS/ TIME STEP) , 0) Confidence increase rate [Tooling,Project] = IF THEN ELSE(Confidence increase switch[Tooling,Project]=1,(FC*No of iterations[Tooling,Project]^PI*MIN(1,Design confidence[Conceptual design,Project])^(PD) *Project Complexity[Project]^PS/ TIME STEP) , 0) **Present in 2 views:** • Parameters • Auxiliaries **Used by:** • Design confidence

#16 A	**Confidence increase switch** (Dmnl) Confidence increase switch [Phase,Project] = IF THEN ELSE("Iteration rate (t+1)"[Phase,Project]>0:AND:Phase finish time[Phase,Project]=0:AND:Uncertainty[Phase,Project]>0, 1,0) **Present in 2 views:** • Auxiliaries • Auxiliaries **Used by:** • Confidence increase rate
#17 LI,C,F,A	**Convergence rate** (Task/Month) Convergence rate [Conceptual design,Project] = IF THEN ELSE(Iteration switch[Conceptual design,Project]=0, 0,IF THEN ELSE(Iteration switch[Conceptual design,Project]=2, Reduction ratio[Project]*Tasks pending decision[Conceptual design,Project]/Decision time , 0)) Convergence rate [Detail design,Project] = 0 Convergence rate [Tooling,Project] = 0 **Present in 1 view:** • Workflow **Used by:** • Sets reduced • Tasks pending decision
#18 A	**Corruption duration** (Month) Corruption duration [Phase] = TIME STEP **Present in 1 view:** • Workflow **Used by:** • Corruption rate
#19 C,A	**Corruption probability** (Dmnl) Corruption probability [Conceptual design,Project] = 0 Corruption probability [Detail design,Project] = IF THEN ELSE(Tasks remained[Conceptual design,Project]>=0 :AND: Concurrency[Conceptual design,Project]>0,F3*Rework probability[Conceptual design,Project], 0) Corruption probability [Tooling,Project] = IF THEN ELSE(Tasks remained[Detail design,Project]>=0:AND:Concurrency[Detail design,Project]>0,F3*Rework probability[Detail design,Project],0) **Description:** *Probability of corrupting the work in upstream phase due to rework in downstream depends on the level of concurrency between phases, rework probability in upstream and interdependency between phases represented by F3(Maier, et al. 2014) used the term "change propagation" to address the corruption probability in design process* **Present in 2 views:** • Workflow • Auxiliaries **Used by:** • Corruption rate
#20 F,A	**Corruption rate** (Task/Month) Corruption rate [Phase,Project] = Corruption probability[Phase,Project]*Tasks

	pending decision[Phase,Project]/Corruption duration[Phase] **Present in 2 views:** • Workflow • Auxiliaries **Used by:** • Tasks corrupted • Tasks not completed
#21 A	**Decision time** (Month) = TIME STEP **Present in 1 view:** • Workflow **Used by:** • Convergence rate Release rate
#22 L	**Design confidence** (Dmnl) Design confidence [Phase,Project] = $\int$ Confidence increase rate[Phase,Project] *dt* + [0] **Description:** *the level of confidence in design built during the project* **Present in 3 views:** • Parameters • Auxiliaries • Auxiliaries **Used by:** • Confidence increase rate • Rework probability • Uncertainty
#23 A	**Desired resources- Completion** (Person) Desired resources- Completion [Conceptual design,Project] = IF THEN ELSE(Time>=Phase start time Actual[Conceptual design,Project],MAX(0,Tasks not completed[Conceptual design,Project]/Completion duration[Conceptual design,Project]/Resource productivity[Conceptual design])*No of concepts[Conceptual design,Project], 0) Desired resources- Completion [Detail design,Project] = IF THEN ELSE(Time>=Phase start time Actual[Detail design,Project]:OR:Phase finish time[Conceptual design,Project]<>0,MAX(0,Tasks not completed[Detail design,Project]/Completion duration[Detail design,Project]/Resource productivity[Detail design])*No of concepts[Detail design,Project], 0) Desired resources- Completion [Tooling,Project] = IF THEN ELSE(Time>=Phase start time Actual[Tooling,Project]:OR:Phase finish time[Detail design,Project]<>0,MAX(0,Tasks not completed[Tooling,Project]/Completion duration[Tooling,Project]/Resource productivity[Tooling])*No of concepts[Tooling,Project], 0) **Present in 1 view:** • Resource management **Used by:** • Allocated resources- Completion • Total desired resources

#24 A	**Desired resources- Rework** (Person) Desired resources- Rework [Phase,Project] = MAX(0,Tasks pending rework[Phase,Project]/Rework duration[Phase,Project]/Resource productivity[Phase])*No of concepts[Phase,Project] **Present in 1 view:** • Resource management **Used by:** • Allocated resources- Rework • Total desired resources
#25 A	**Desired resources- Test** (Person) Desired resources- Test [Phase,Project] = MAX(0,Tasks pending test[Phase,Project]/Test duration[Phase,Project]/Resource productivity[Phase])*No of concepts[Phase,Project] **Present in 1 view:** • Resource management **Used by:** • Allocated resources- Test • Total desired resources
#26 D	**F1** (Dmnl) = GET XLS CONSTANTS('variables.xlsx', 'data', 'B2') **Description:** *Determines the level of confidence in concepts when narrowing down starts during conceptual design phase* **Present in 2 views:** • Parameters • Auxiliaries **Used by:** • Iteration switch
#27 D	**F2** (Dmnl [0,?,0.1]) = GET XLS CONSTANTS('variables.xlsx', 'data', 'B3') **Description:** *Control the rate of concurrency change, TC change, and reduction rate change* **Present in 2 views:** • Parameters • Auxiliaries **Used by:** • Reduction increase rate • TC change rate
#28 D	**F3** (Dmnl) = GET XLS CONSTANTS('variables.xlsx', 'data', 'B4') **Description:** *Affects the value of corruption and is affected by the interdependency between upstream and downstream phases. Equivalent to downstream sensitivity to upstream changes acc. to (1997 Krishnan, Eppinger)* **Present in 2 views:** • Parameters • Auxiliaries **Used by:** • Corruption probability

#29 D	**FC** (Dmnl) = GET XLS CONSTANTS('variables.xlsx', 'data', 'B5') **Description:** *the constant coefficient in "confidence increase rate" equation* **Present in 2 views:** • Parameters • Auxiliaries **Used by:** • Confidence increase rate
#30 D	**FDJ** (Month) FDJ [Project] = GET XLS CONSTANTS('Variables.xlsx', 'Time & Cost data', 'B7') **Description:** *Final data judgement milestone. Finish of detail design phase.* **Present in 3 views:** • Parameters • Performance measures • Auxiliaries **Used by:** • Phase deadline • Phase finish time Actual
#31 C	**FINAL TIME** (Month) = 500 **Description:** *The final time for the simulation.* **Present in 1 view:** • Performance measures **Used by:** • Project PO Rework
#32 F,A	**Increase no of iteration** (1/Month) Increase no of iteration [Phase,Project] = IF THEN ELSE(Iteration rate[Phase,Project]>0:AND:Phase finish time[Phase,Project]=0,1/TIME STEP, 0) **Present in 3 views:** • Parameters • Performance measures • Auxiliaries **Used by:** • No of iterations
#33 D	**Initial available resources** (Person) Initial available resources [Conceptual design,Project] = GET XLS CONSTANTS('Variables.xlsx', 'Data', 'c19') Initial available resources [Detail design,Project] = GET XLS CONSTANTS('Variables.xlsx', 'Data', 'c20') Initial available resources [Tooling,Project] = GET XLS CONSTANTS('Variables.xlsx', 'Data', 'c21') **Description:** *Number of engineering resources available for the project* **Present in 2 views:** • Parameters • Auxiliaries **Used by:** • Available resources

#34 D	**Initial completion duration** (Month) Initial completion duration [Conceptual design,Project] = GET XLS CONSTANTS('Variables.xlsx', 'Data', 'c26') Initial completion duration [Detail design,Project] = GET XLS CONSTANTS('Variables.xlsx', 'Data', 'c27') Initial completion duration [Tooling,Project] = GET XLS CONSTANTS('Variables.xlsx', 'Data', 'c28') **Description:** *Average time required to initially complete a task (work package)* **Present in 2 views:** • Parameters • Auxiliaries **Used by:** • Completion duration
#35 LI,C	**Initial reduction ratio** (Dmnl [0,100,0.1]) = 0.2 **Description:** *Normal reduction ratio for narrowing down concepts during conceptual design phase* **Present in 2 views:** • Parameters • Auxiliaries **Used by:** • Reduction ratio
#36 D	**Initial rework probability** (Dmnl) Initial rework probability [Conceptual design,Project] = GET XLS CONSTANTS('Variables.xlsx', 'Data', 'c23') Initial rework probability [Detail design,Project] = GET XLS CONSTANTS('Variables.xlsx', 'Data', 'c24') Initial rework probability [Tooling,Project] = GET XLS CONSTANTS('Variables.xlsx', 'Data', 'c25') **Present in 2 views:** • Parameters • Auxiliaries **Used by:** • Rework probability
#37 C	**INITIAL TIME** (Month) = 0 **Description:** *The initial time for the simulation.* **Not Present In Any View** **Used by:** • Time
#38 LI,A	**Iteration rate** (Task/Month) Iteration rate [Conceptual design,Project] = IF THEN ELSE(Iteration switch[Conceptual design,Project]=0, 0,IF THEN ELSE(Iteration switch[Conceptual design,Project]=2, (1-Reduction ratio[Project])*Tasks pending decision[Conceptual design, Project]/Iteration time,Tasks pending decision[Conceptual design, Project]/Iteration time)) Iteration rate [Detail design,Project] = IF THEN ELSE(Iteration switch[Detail

	design,Project]=1,Tasks pending decision[Detail design,Project]/Iteration time, 0) Iteration rate [Tooling,Project] = IF THEN ELSE(Iteration switch[Tooling,Project]=1,Tasks pending decision[Tooling,Project]/Iteration time, 0) **Present in 3 views:** • Workflow • Auxiliaries • Auxiliaries **Used by:** • Increase no of iteration • Iteration rate (t+1) • Tasks not completed • Tasks pending decision
#39 DE	**Iteration rate (t+1)** (Task/Month) Iteration rate (t+1) [Phase,Project] = DELAY FIXED (Iteration rate[Phase,Project], TIME STEP, Iteration rate[Phase,Project]) **Present in 1 view:** • Auxiliaries **Used by:** • Confidence increase switch
#40 A	**Iteration switch** (Dmnl) Iteration switch [Conceptual design,Project] = (IF THEN ELSE(Tasks remained[Conceptual design, Project]>=0 :OR: Phase finish time[Conceptual design,Project]<>0,0, IF THEN ELSE(Uncertainty[Conceptual design,Project]<=F1:AND:Tasks released[Conceptual design, Project]=0,2,1))) Iteration switch [Detail design,Project] = IF THEN ELSE(Uncertainty[Detail design,Project]>0:AND:Tasks remained[Detail design,Project]<0:AND:Tasks released[Detail design,Project]=0,1,0) Iteration switch [Tooling,Project] = IF THEN ELSE(Uncertainty[Tooling,Project]>0:AND:Tasks remained[Tooling,Project]<0:AND:Tasks released[Tooling,Project]=0,1,0) **Present in 2 views:** • Workflow • Auxiliaries **Used by:** • Convergence rate • Iteration rate
#41 A	**Iteration time** (Month) = TIME STEP **Present in 1 view:** • Workflow **Used by:** • Iteration rate
#42 D	**J1** (Month) J1 [Project] = GET XLS CONSTANTS('Variables.xlsx', 'Time & Cost data', 'B8')

	Description: *Job No 1. Start of mass production* **Present in 3 views:** • Parameters • Performance measures • Auxiliaries **Used by:** • Phase deadline • Phase finish time Actual
#43 C	**Max Reduction ratio** (Dmnl) = 0.95 **Present in 1 view:** • Auxiliaries **Used by:** • Reduction increase rate
#44 C	**Min target confidence** (Dmnl) = 0.5 **Present in 2 views:** • Parameters • Auxiliaries **Used by:** • TC change rate
#45 C,D	**No of concepts** (Dmnl [1,?,1]) No of concepts [Conceptual design,Project] = GET XLS CONSTANTS('Variables.xlsx', 'Data', 'c22') No of concepts [Detail design,Project] = 1 No of concepts [Tooling,Project] = 1 **Description:** *No of concepts working on simultaneously during conceptual design phase. for all other phases it should be equal to 1 showing just 1 concept is sent for detail design phase* **Present in 4 views:** • Parameters • Resource management • Auxiliaries • Auxiliaries **Used by:** • Confidence increase rate • Desired resources- Completion • Desired resources- Rework • Desired resources- Test • Release switch • Time required
#46 L	**No of iterations** (Dmnl) No of iterations [Phase,Project] = $\int$ Increase no of iteration[Phase,Project] *dt* + [0] **Description:** *Number of iteration during phases of the project* **Present in 2 views:**

	• Parameters • Auxiliaries **Used by:** • Confidence increase rate • Total No of iteration
#47 D	**PC** (Dmnl [1,?]) = GET XLS CONSTANTS('variables.xlsx', 'data', 'B6') **Description:** *The strength of number of concepts effect: P(Concept)* **Present in 2 views:** • Parameters • Auxiliaries **Used by:** • Confidence increase rate
#48 D	**PD** (Dmnl) = GET XLS CONSTANTS('variables.xlsx', 'data', 'B7') **Description:** *The strength of the effect of the final design confidence in upstream phase: P(design confidence)* **Present in 2 views:** • Parameters • Auxiliaries **Used by:** • Confidence increase rate
#49 L	**Person Month** (Month*Person) Person Month [Conceptual design,Project] = $\int$ Total allocated resources[Conceptual design,Project] *dt* + [0] Person Month [Detail design,Project] = $\int$ Total allocated resources[Detail design,Project] *dt* + [0] Person Month [Tooling,Project] = $\int$ Total allocated resources[Tooling,Project] *dt* + [0] **Present in 2 views:** • Parameters • Performance measures **Used by:** • Phase cost
#50 D	**PG** (Dmnl) = GET XLS CONSTANTS('variables.xlsx', 'data', 'B8') **Description:** *The strength of design confidence effect in rework probability equation: P(Design Confidence)* **Present in 2 views:** • Parameters • Auxiliaries **Used by:** • Rework probability

#51 Sub	**Phase** : Conceptual design, Detail design, Tooling **Present in 6 views:** • Parameters • Workflow • Resource management • Performance measures • Auxiliaries • Auxiliaries **Used by:** • Add resource rate • Allocated resources- Completion • Allocated resources- Rework • Allocated resources- Test • Approve rate • Available resources • Change in phase time • Change in project time • Completion capacity • Completion duration • Completion rate • Concurrency • Confidence increase rate • Confidence increase switch • Convergence rate • Corruption probability • Corruption rate • Design confidence • Desired resources- Completion • Desired resources- Rework • Desired resources- Test • Increase no of iteration • Iteration rate • Iteration rate (t+1) • Iteration switch • No of iterations • Person Month • Phase cost • Phase finish time • Phase finish time2 • Phase percent released • Phase PO finish time • Phase time (t+1) • Project cost curve • Project cost Model • Project PO

	• Reduction increase rate • Release rate • Release switch • Rework capacity • Rework detection rate • Rework duration • Rework per phase • Rework probability • Rework rate • Reworks per project • Schedule pressure • Sets reduced • Target confidence • Tasks corrupted • Tasks not completed • Tasks pending decision • Tasks pending rework • Tasks pending test • Tasks released • Tasks remained • TC change rate • Test capacity • Test duration • Time available • Time required • Total activities per phase • Total allocated resources • Total capacity • Total design confidence • Total desired resources • Total No of iteration • Total phase tasks • Total project activities • Total project tasks • Total tasks released • Uncertainty
#52 A	**Phase cost** ($) Phase cost [Phase,Project] = Person Month[Phase,Project]*Unit cost **Present in 2 views:** • Parameters • Performance measures **Used by:** • Project cost curve • Project cost Model

#53 A	**Phase deadline** (Month) Phase deadline [Conceptual design,Project] = IF THEN ELSE(Time<PSC[Project], PSC[Project], PTC[Project]) Phase deadline [Detail design,Project] = FDJ[Project] Phase deadline [Tooling,Project] = J1[Project] **Description:** *projects milestones* **Present in 1 view:** • Auxiliaries **Used by:** • Time available
#54 L	**Phase finish time** (Month) Phase finish time [Phase,Project] = $\int$ Change in phase time[Phase,Project] *dt* + [0] **Present in 6 views:** • Parameters • Workflow • Resource management • Performance measures • Auxiliaries • Auxiliaries **Used by:** • Change in phase time • Change in project time • Completion rate • Confidence increase switch • Desired resources- Completion • Increase no of iteration • Iteration switch • Phase finish time2 • Phase time (t+1) • Project cost Model • Rework probability
#55 A	**Phase finish time Actual** (Month) Phase finish time Actual [Conceptual design,Project] = PTC[Project] Phase finish time Actual [Detail design,Project] = FDJ[Project] Phase finish time Actual [Tooling,Project] = J1[Project] **Present in 1 view:** • Performance measures **Used by:** • Phase PO finish time
#56 D	**Phase finish time weight** (Dmnl) = GET XLS CONSTANTS('Variables.xlsx', 'Weights', 'B3') **Present in 2 views:** • Parameters • Performance measures **Used by:**

	• Project PO
#57 A	**Phase finish time2** (Month) Phase finish time2 [Phase,Project] = Phase finish time[Phase,Project]-"Phase time (t+1)"[Phase,Project] **Present in 1 view:** • Performance measures **Used by:** • Phase PO finish time
#58 A	**Phase percent released** (Dmnl) Phase percent released [Phase,Project] = ZIDZ((Sets reduced[Phase,Project]+Tasks released[Phase,Project]),Total phase tasks[Phase,Project]) **Present in 1 view:** • Auxiliaries **Used by:** • **This is a supplementary variable.**
#59 A	**Phase PO finish time** (Dmnl) Phase PO finish time [Phase,Project] = IF THEN ELSE(Phase finish time2[Phase,Project]=0, 0, ZIDZ((Phase finish time2[Phase,Project]-Phase finish time Actual[Phase,Project])^2,Phase finish time2[Phase,Project]^2 +Phase finish time Actual[Phase,Project]^2)) **Present in 2 views:** • Parameters • Performance measures **Used by:** • Project PO
#60 D	**Phase start time Actual** (Month) Phase start time Actual [Phase,A] = GET XLS CONSTANTS('Variables.xlsx', 'Time & Cost data', 'B17') Phase start time Actual [Phase,B] = GET XLS CONSTANTS('Variables.xlsx', 'Time & Cost data', 'B18') Phase start time Actual [Phase,C] = GET XLS CONSTANTS('Variables.xlsx', 'Time & Cost data', 'B19') Phase start time Actual [Phase,D] = GET XLS CONSTANTS('Variables.xlsx', 'Time & Cost data', 'B20') **Present in 4 views:** • Parameters • Workflow • Resource management • Auxiliaries **Used by:** • Completion rate • Concurrency • Desired resources- Completion • Rework probability
#61 DE	**Phase time (t+1)** (Month) Phase time (t+1) [Phase,Project] = DELAY FIXED (Phase finish

	time[Phase,Project],TIME STEP,Phase finish time[Phase,Project]) **Present in 1 view:** • Performance measures **Used by:** • Phase finish time2
#62 D	**PI** (Dmnl [1,?,0.1]) = GET XLS CONSTANTS('variables.xlsx', 'data', 'B9') **Description:** *The strength of iteration effect: P(Iteration)* **Present in 2 views:** • Parameters • Auxiliaries **Used by:** • Confidence increase rate
#63 L	**PO total** (Dmnl) $= \int$ Project PO total/TIME STEP *dt* + [0] **Present in 1 view:** • Performance measures **Used by:** • **This is a supplementary variable.**
#64 Sub	**Project** : A, B, C, D **Description:** *A: project complexity=4; B: project complexity=3; C: project complexity=2; D: project complexity=1* **Present in 6 views:** • Parameters • Workflow • Resource management • Performance measures • Auxiliaries • Auxiliaries **Used by:** • Add resource rate • Allocated resources- Completion • Allocated resources- Rework • Allocated resources- Test • Approve rate • Available resources • Change in phase time • Change in project time • Completion capacity • Completion duration • Completion rate • Concurrency • Confidence increase rate • Confidence increase switch

	• Convergence rate • Corruption probability • Corruption rate • Design confidence • Desired resources- Completion • Desired resources- Rework • Desired resources- Test • Increase no of iteration • Iteration rate • Iteration rate (t+1) • Iteration switch • No of iterations • Person Month • Phase cost • Phase deadline • Phase finish time • Phase finish time Actual • Phase finish time2 • Phase percent released • Phase PO finish time • Phase time (t+1) • Project cost curve • Project cost Model • Project duration (t+1) • Project duration Model • Project duration- Model2 • Project percent complete • Project percent corrupted • Project percent reworked Model • Project PO • Project PO Cost • Project PO Rework • Project PO Rework Curve • Project PO total • Reduction increase rate • Reduction ratio • Release rate • Release switch • Rework capacity • Rework detection rate • Rework duration • Rework per phase • Rework probability • Rework rate • Reworks per project • Schedule pressure

	• Sets reduced • Target confidence • Tasks corrupted • Tasks not completed • Tasks pending decision • Tasks pending rework • Tasks pending test • Tasks released • Tasks remained • TC change rate • Test capacity • Test duration • Time available • Time required • Total activities per phase • Total allocated resources • Total capacity • Total design confidence • Total desired resources • Total No of iteration • Total phase tasks • Total project activities • Total project tasks • Total tasks released • Uncertainty
#65 D	**Project cost Actual** ($) Project cost Actual [Project] = GET XLS CONSTANTS('Variables.xlsx', 'Time & Cost data', 'B9') **Present in 2 views:** • Parameters • Performance measures **Used by:** • Project PO Cost
#66 A	**Project cost curve** ($) Project cost curve [Project] = $\sum$(Phase cost[Phase!,Project]) **Present in 2 views:** • Parameters • Performance measures **Used by:** • **This is a supplementary variable.**
#67 A	**Project cost Model** ($) Project cost Model [Project] = IF THEN ELSE(Phase finish time[Conceptual design,Project]<>0:AND:Phase finish time[Detail design,Project]<>0:AND:Phase finish time[Tooling,Project]<>0, $\sum$(Phase

	cost[Phase!,Project]), 0) **Present in 2 views:** • Parameters • Performance measures **Used by:** • Project PO Cost
#68 D	**Project cost weight** (Dmnl) = GET XLS CONSTANTS('Variables.xlsx', 'Weights', 'B1') **Present in 2 views:** • Parameters • Performance measures **Used by:** • Project PO
#69 DE	**Project duration (t+1)** (Month) Project duration (t+1) [Project] = DELAY FIXED (Project duration Model[Project],TIME STEP,Project duration Model[Project]) **Present in 1 view:** • Performance measures **Used by:** • Project duration- Model2
#70 L	**Project duration Model** (Month) Project duration Model [Project] = $\int$ Change in project time[Project] *dt* + [0] **Present in 2 views:** • Parameters • Performance measures **Used by:** • Change in project time • Project duration (t+1) • Project duration- Model2 • Project PO Rework
#71 A	**Project duration- Model2** (Month) Project duration- Model2 [Project] = Project duration Model[Project]-"Project duration (t+1)"[Project] **Present in 1 view:** • Performance measures **Used by:** • Project PO Cost
#72 A	**Project percent complete** (Dmnl) Project percent complete [Project] = Total tasks released[Project]/Total project tasks[Project]*100 **Present in 2 views:** • Parameters • Auxiliaries **Used by:** • **This is a supplementary variable.**

#73 A	**Project percent corrupted** (Dmnl) Project percent corrupted [Project] = Tasks corrupted[Project]/Total project activities 0[Project]*100 **Present in 2 views:** • Parameters • Auxiliaries **Used by:** • **This is a supplementary variable.**
#74 LI,D,F,A	**Project percent reworked Actual** (Dmnl) Project percent reworked Actual [A] = GET XLS DATA('Variables.xlsx', 'Rework data', 'A', 'B3') Project percent reworked Actual [B] = GET XLS DATA('Variables.xlsx', 'Rework data', 'A', 'C3') Project percent reworked Actual [C] = GET XLS DATA('Variables.xlsx', 'Rework data', 'A', 'D3') Project percent reworked Actual [D] = GET XLS DATA('Variables.xlsx', 'Rework data', 'A', 'E3') **Present in 2 views:** • Parameters • Performance measures **Used by:** • Project PO Rework Curve
#75 LI,F,A	**Project percent reworked Model** (Dmnl) Project percent reworked Model [Project] = ZIDZ(Reworks per project[Project],Total project activities 0[Project])*100 **Present in 2 views:** • Parameters • Performance measures **Used by:** • Project PO Rework Curve
#76 A	**Project PO** (Dmnl) Project PO [Project] = $\sum$(Phase PO finish time[Phase!,Project])*Phase finish time weight+Project PO Cost[Project]*Project cost weight+Project PO Rework[Project]*Project rework weight **Present in 1 view:** • Performance measures **Used by:** • Project PO total
#77 A	**Project PO Cost** (Dmnl) Project PO Cost [Project] = IF THEN ELSE("Project duration-Model2"[Project]=0, 0, ZIDZ((Project cost Model[Project]-Project cost Actual[Project])^2, Project cost Model[Project]^2+Project cost Actual[Project]^2)) **Present in 2 views:** • Parameters • Performance measures **Used by:**

	• Project PO
#78 A	**Project PO Rework** (Dmnl) Project PO Rework [Project] = IF THEN ELSE(Time=FINAL TIME-2*TIME STEP,Project PO Rework Curve[Project]/IF THEN ELSE(Project duration Model[Project]=0, FINAL TIME, Project duration Model[Project]), 0) **Description:** *Normalized squared error of project percent reworked* **Present in 2 views:** • Parameters • Performance measures **Used by:** • Project PO
#79 L	**Project PO Rework Curve** (Month) Project PO Rework Curve [Project] = $\int$ ZIDZ((Project percent reworked Model[Project]-Project percent reworked Actual[Project])^2, Project percent reworked Model[Project]^2+Project percent reworked Actual[Project]^2) *dt* + [0] **Present in 1 view:** • Performance measures **Used by:** • Project PO Rework
#80 F,A	**Project PO total** (Dmnl) = $\sum$(Project PO[Project!]) **Present in 1 view:** • Performance measures **Used by:** • PO total
#81 D	**Project rework weight** (Dmnl) = GET XLS CONSTANTS('Variables.xlsx', 'Weights', 'B2') **Present in 2 views:** • Parameters • Performance measures **Used by:** • Project PO
#82 D	**Project Complexity** (Dmnl) Project Complexity [Project] = GET XLS CONSTANTS('Variables.xlsx', 'Time & Cost data', 'B3') **Description:** *The level of innovativeness of the project* **Present in 2 views:** • Parameters • Auxiliaries **Used by:** • Confidence increase rate
#83 D	**PS** (Dmnl) = GET XLS CONSTANTS('variables.xlsx', 'data', 'B10')

	Description: *The strength of project complexity effect: P(Complexity)* **Present in 2 views:** • Parameters • Auxiliaries **Used by:** • Confidence increase rate
#84 D	**PSC** (Month) PSC [Project] = GET XLS CONSTANTS('Variables.xlsx', 'Time & Cost data', 'B5') **Description:** *Program strategy confirmed milestone. Start of narrowing down concepts during conceptual design phase* **Present in 2 views:** • Parameters • Auxiliaries **Used by:** • Phase deadline
#85 D	**PTC** (Month) PTC [Project] = GET XLS CONSTANTS('Variables.xlsx', 'Time & Cost data', 'B6') **Description:** *Program target compatibility milestone. Finish of conceptual design phase. A single concept in confirmed.* **Present in 3 views:** • Parameters • Performance measures • Auxiliaries **Used by:** • Phase deadline Phase finish time Actual
#86 F,A	**Reduction increase rate** (1/Month) Reduction increase rate [Project] = IF THEN ELSE(Schedule pressure[Conceptual design,Project]>1:AND:Uncertainty[Conceptual design,Project]>0:AND:Reduction ratio[Project]<Max Reduction ratio, F2/TIME STEP, 0) **Present in 1 view:** • Auxiliaries **Used by:** • Reduction ratio
#87 L	**Reduction ratio** (Dmnl) Reduction ratio [Project] = $\int$ Reduction increase rate[Project] *dt* + [Initial reduction ratio] **Description:** *The rate of narrowing down sets of concepts during conceptual design phase.* **Present in 2 views:** • Workflow • Auxiliaries **Used by:** • Convergence rate • Iteration rate • Reduction increase rate

#88 LI,F,A	**Release rate** (Task/Month) Release rate [Phase,Project] = (IF THEN ELSE(Release switch[Phase,Project]=1,Tasks pending decision[Phase,Project]/Decision time, 0)) **Present in 1 view:** • Workflow **Used by:** • Tasks pending decision • Tasks released
#89 A	**Release switch** (Dmnl) Release switch [Conceptual design,Project] = IF THEN ELSE(Sets reduced[Conceptual design,Project]>=Scope[Conceptual design,Project]*(No of concepts[Conceptual design,Project]-1)/No of concepts[Conceptual design,Project]:OR:Uncertainty[Conceptual design,Project]=0, 1, 0) Release switch [Detail design,Project] = (IF THEN ELSE(Uncertainty[Detail design,Project]<=0, 1, 0)) Release switch [Tooling,Project] = (IF THEN ELSE(Uncertainty[Tooling,Project]<=0, 1, 0)) **Description:** *it is defined for conceptual design phase based on the idea that narrowing down concepts continue to a point that only one concept remains as the final selected concept to send for detail design.* **Present in 2 views:** • Workflow • Auxiliaries **Used by:** • Release rate
#90 C	**Resource productivity** (Task/(Person*Month)) Resource productivity [Phase] = 1 **Description:** *The productivity, although constant in this model, depends on the work pressure and Task complexity effects.* **Present in 2 views:** • Parameters • Resource management **Used by:** • Completion capacity • Desired resources- Completion • Desired resources- Rework • Desired resources- Test • Rework capacity • Test capacity • Total capacity
#91 A	**Rework capacity** (Task/Month) Rework capacity [Phase,Project] = "Allocated resources-Rework"[Phase,Project]*Resource productivity[Phase] **Present in 2 views:** • Workflow • Resource management

	Used by: • Rework rate
#92 LI,F,A	**Rework detection rate** (Task/Month) Rework detection rate [Phase,Project] = MIN(Tasks pending test[Phase,Project]*Rework probability[Phase,Project]/Test duration[Phase,Project],Test capacity[Phase,Project]) **Present in 2 views:** • Workflow • Performance measures **Used by:** • Tasks pending rework • Tasks pending test • Total activities per phase
#93 A	**Rework duration** (Month) Rework duration [Phase,Project] = 0.5*Completion duration[Phase,Project] **Description:** *Average time required to rework a task (work package) if it is found faulty during test. Acc. to Browning and Eppinger (2002), often it takes less effort to rework an activity than to do it the first time, so here it is assumed to be the half of completion duration.* **Present in 3 views:** • Workflow • Resource management • Auxiliaries **Used by:** • Desired resources- Rework • Rework rate
#94 L	**Rework per phase** (Task) Rework per phase [Conceptual design,Project] = $\int$ Rework rate[Conceptual design,Project] *dt* + [0] Rework per phase [Detail design,Project] = $\int$ Rework rate[Detail design,Project] *dt* + [0] Rework per phase [Tooling,Project] = $\int$ Rework rate[Tooling,Project] *dt* + [0] **Present in 2 views:** • Parameters • Performance measures **Used by:** • Reworks per project
#95 A	**Rework probability** (Dmnl) Rework probability [Conceptual design,Project] = IF THEN ELSE(Time>=Phase start time Actual[Conceptual design,Project]:AND:Tasks released[Conceptual design,Project]=0,Initial rework probability[Conceptual design,Project]*(1-Design confidence[Conceptual design,Project])^PG, 0) Rework probability [Detail design,Project] = IF THEN ELSE((Time>=Phase start

	time Actual[Detail design,Project]:OR:Phase finish time[Conceptual design,Project]<>0):AND:Tasks released[Detail design,Project]=0,Initial rework probability[Detail design,Project]*(1-Design confidence[Detail design,Project])^PG, 0) Rework probability [Tooling,Project] = IF THEN ELSE((Time>=Phase start time Actual[Tooling,Project]:OR:Phase finish time[Detail design,Project]<>0):AND:Tasks released[Tooling,Project]=0,Initial rework probability[Tooling,Project]*(1-Design confidence[Tooling,Project])^PG, 0) **Description:** *The probability of rework during each phase of the project. the amount of rework probability reduces during the project due to the learning effect and the reduction of uncertainty.* **Present in 2 views:** • Workflow • Auxiliaries **Used by:** • Approve rate • Corruption probability • Rework detection rate
#96 LI,F,A	**Rework rate** (Task/Month) Rework rate [Phase,Project] = MIN(Tasks pending rework[Phase,Project]/Rework duration[Phase,Project],Rework capacity[Phase,Project]) **Present in 2 views:** • Workflow • Performance measures **Used by:** • Rework per phase • Tasks pending rework • Tasks pending test Total activities per phase
#97 A	**Reworks per project** (Task) Reworks per project [Project] = $\sum$(Rework per phase[Phase!,Project]) **Present in 2 views:** • Parameters • Performance measures **Used by:** • Project percent reworked Model
#98 A	**SAVEPER** (Month [0,?]) = TIME STEP **Description:** *The frequency with which output is stored.* **Not Present In Any View**
#99 A	**Schedule pressure** (Dmnl) Schedule pressure [Phase,Project] = Time required[Phase,Project]/Time available[Phase,Project] **Present in 1 view:** • Auxiliaries **Used by:** • Add resource rate

	• Reduction increase rate • TC change rate
#100 D	**Scope** (Task [0,?,1]) Scope [Conceptual design,Project] = GET XLS CONSTANTS('Variables.xlsx', 'Data', 'c16') Scope [Detail design,Project] = GET XLS CONSTANTS('Variables.xlsx', 'Data', 'c17') Scope [Tooling,Project] = GET XLS CONSTANTS('Variables.xlsx', 'Data', 'c18') **Description:** *Number of tasks (work packages) per phase (In Conceptual design phase, no of tasks (work packages) per concept)* **Present in 3 views:** • Parameters • Workflow • Auxiliaries **Used by:** • Release switch • Tasks not completed
#101 L	**Sets reduced** (Task) Sets reduced [Phase,Project] = $\int$ Convergence rate[Phase,Project] *dt* + [0] **Description:** *Tasks (work packages) related to concepts reduced during conceptual design phase* **Present in 4 views:** • Parameters • Workflow • Performance measures • Auxiliaries **Used by:** • Change in phase time • Concurrency • Phase percent released • Release switch • Total phase tasks • Total tasks released
#102 L	**Target confidence** (Dmnl) Target confidence [Phase,Project] = $\int$ -TC change rate[Phase,Project] *dt* + [1] **Description:** *Desired level of confidence in design. The initial value of target confidence is assumed to be 100% for each phase which could be reduced as the result of schedule pressures.* **Present in 1 view:** • Auxiliaries **Used by:** • TC change rate • Uncertainty

#103 L	**Tasks corrupted** (Task) Tasks corrupted [Project] = $\int\sum$(Corruption rate[Phase!,Project]) *dt* + [0] **Present in 2 views:** • Parameters • Auxiliaries **Used by:** • Project percent corrupted
#104 L	**Tasks not completed** (Task) Tasks not completed [Phase,Project] = $\int$ Corruption rate[Phase,Project]+Iteration rate[Phase,Project]-Completion rate[Phase,Project] *dt* + [Scope[Phase,Project]] **Present in 3 views:** • Workflow • Resource management • Auxiliaries **Used by:** • Completion rate • Desired resources- Completion • Tasks remained
#105 L	**Tasks pending decision** (Task) Tasks pending decision [Phase,Project] = $\int$ Approve rate[Phase,Project]-Convergence rate[Phase,Project]-Iteration rate[Phase,Project]-Release rate[Phase,Project] *dt* + [0] **Description:** *Tasks approved waiting to be released* **Present in 2 views:** • Workflow • Auxiliaries **Used by:** • Convergence rate • Corruption rate • Iteration rate • Release rate • Total phase tasks
#106 L	**Tasks pending rework** (Task) Tasks pending rework [Phase,Project] = $\int$ Rework detection rate[Phase,Project]-Rework rate[Phase,Project] *dt* + [0] **Description:** *Tasks (work packages) faulty waiting for rework* **Present in 3 views:** • Workflow • Resource management • Auxiliaries **Used by:**

	• Desired resources- Rework • Rework rate • Tasks remained
#107 L	**Tasks pending test** (Task) Tasks pending test [Phase,Project] = $\int$ Completion rate[Phase,Project]+Rework rate[Phase,Project]-Approve rate[Phase,Project]-Rework detection rate[Phase,Project] *dt* + [0] **Description:** *Tasks (work packages) completed waiting for test* **Present in 3 views:** • Workflow • Resource management • Auxiliaries **Used by:** • Approve rate • Desired resources- Test • Rework detection rate • Tasks remained
#108 L	**Tasks released** (Task) Tasks released [Phase,Project] = $\int$ Release rate[Phase,Project] *dt* + [0] **Description:** *Tasks (work packages) released to the next phase* **Present in 4 views:** • Parameters • Workflow • Performance measures • Auxiliaries **Used by:** • Change in phase time • Concurrency • Iteration switch • Phase percent released • Rework probability • Total phase tasks • Total tasks released
#109 A	**Tasks remained** (Task) Tasks remained [Phase,Project] = Tasks not completed[Phase,Project]+Tasks pending rework[Phase,Project]+Tasks pending test[Phase,Project] **Description:** *Total number of tasks (work packages) per phase waiting for action* **Present in 2 views:** • Auxiliaries • Auxiliaries **Used by:** • Corruption probability • Iteration switch

	• Time required • Total phase tasks
#110 F,A	**TC change rate** (1/Month) TC change rate [Phase,Project] = IF THEN ELSE(Schedule pressure[Phase,Project]>1:AND:Uncertainty[Phase,Project]>0:AND:Target confidence[Phase,Project]>Min target confidence, F2/TIME STEP, 0) **Present in 2 views:** • Parameters • Auxiliaries **Used by:** • Target confidence
#111 A	**Test capacity** (Task/Month) Test capacity [Phase,Project] = "Allocated resources-Test"[Phase,Project]*Resource productivity[Phase] **Present in 2 views:** • Workflow • Resource management **Used by:** • Approve rate • Rework detection rate
#112 A	**Test duration** (Month) Test duration [Phase,Project] = Completion duration[Phase,Project] **Description:** *Average time required to test a task (work package) after completion. here it is assumed to be equal to completion duration acc. to (Ford, Sterman, 1998).* **Present in 3 views:** • Workflow • Resource management • Auxiliaries **Used by:** • Approve rate • Desired resources- Test • Rework detection rate
#113 A	**Time available** (Month) Time available [Phase,Project] = MAX(Phase deadline[Phase,Project]-Time,TIME STEP) **Present in 1 view:** • Auxiliaries **Used by:** • Schedule pressure
#114 A	**Time required** (Month) Time required [Phase,Project] = ZIDZ(Tasks remained[Phase,Project]*No of concepts[Phase,Project],Total capacity[Phase,Project]) **Description:** *Time required to complete each phase* **Present in 1 view:** • Auxiliaries **Used by:**

	• Schedule pressure
#115 C,F	**TIME STEP** (Month [0,?]) = 0.015625 **Description:** *The time step for the simulation.* **Present in 5 views:** • Parameters • Workflow • Performance measures • Auxiliaries • Auxiliaries **Used by:** • Add resource rate • Change in phase time • Change in project time • Confidence increase rate • Corruption duration • Decision time • Increase no of iteration • Iteration rate (t+1) • Iteration time • Phase time (t+1) • PO total • Project duration (t+1) • Project PO Rework • Reduction increase rate • SAVEPER • TC change rate • Time available
#116 L	**Total activities per phase** (Task) Total activities per phase [Conceptual design,Project] = $\int$ Approve rate[Conceptual design,Project]+Completion rate[Conceptual design,Project]+Rework detection rate[Conceptual design,Project]+Rework rate[Conceptual design,Project] *dt* + [0] Total activities per phase [Detail design,Project] = $\int$ Approve rate[Detail design,Project]+Completion rate[Detail design,Project]+Rework detection rate[Detail design,Project]+Rework rate[Detail design,Project] *dt* + [0] Total activities per phase [Tooling,Project] = $\int$ Approve rate[Tooling,Project]+Completion rate[Tooling,Project]+Rework detection rate[Tooling,Project]+Rework rate[Tooling,Project] *dt* + [0] **Present in 2 views:** • Parameters • Performance measures **Used by:**

	• Total project activities
#117 F,A	**Total allocated resources** (Person) Total allocated resources [Phase,Project] = "Allocated resources-Completion"[Phase,Project]+"Allocated resources-Rework"[Phase,Project]+"Allocated resources- Test"[Phase,Project] **Present in 3 views:** • Resource management • Performance measures • Auxiliaries **Used by:** • Add resource rate • Person Month • Total capacity
#118 A	**Total capacity** (Task/Month) Total capacity [Phase,Project] = Total allocated resources[Phase,Project]*Resource productivity[Phase] **Present in 2 views:** • Resource management • Auxiliaries **Used by:** • Time required
#119 A	**Total design confidence** (Dmnl) Total design confidence [Project] = (MIN(1,Design confidence[Conceptual design,Project])+MIN(1,Design confidence[Detail design,Project])+MIN(1,Design confidence[Tooling,Project]))/3*100 **Present in 2 views:** • Parameters • Auxiliaries **Used by:** • **This is a supplementary variable.**
#120 A	**Total desired resources** (Person) Total desired resources [Phase,Project] = "Desired resources-Completion"[Phase,Project]+"Desired resources-Test"[Phase,Project]+"Desired resources- Rework"[Phase,Project] **Present in 2 views:** • Resource management • Auxiliaries **Used by:** • Add resource rate • Allocated resources- Completion • Allocated resources- Rework • Allocated resources- Test
#121 A	**Total No of iteration** (Dmnl) Total No of iteration [Project] = $\sum$(No of iterations[Phase!,Project]) **Present in 1 view:** • Parameters

	Used by: • **This is a supplementary variable.**
#122 A	**Total phase tasks** (Task) Total phase tasks [Phase,Project] = Tasks remained[Phase,Project]+Tasks released[Phase,Project]+Tasks pending decision[Phase,Project]+Sets reduced[Phase,Project] **Present in 2 views:** • Performance measures • Auxiliaries **Used by:** • Change in phase time • Concurrency • Phase percent released • Total project tasks
#123 A	**Total project activities** (Task) Total project activities [Project] = $\sum$(Total activities per phase[Phase!,Project]) **Present in 2 views:** • Parameters • Performance measures **Used by:** • **This is a supplementary variable.**
#124 D	**Total project activities 0** (Task) Total project activities 0 [Project] = GET XLS CONSTANTS('Variables.xlsx', 'Data', 'c14') **Description:** *Final number of activities in a project extracted from Total Project activity curve.* **Present in 3 views:** • Parameters • Performance measures • Auxiliaries **Used by:** • Project percent corrupted • Project percent reworked Model
#125 A	**Total project tasks** (Task) Total project tasks [Project] = Total phase tasks[Conceptual design,Project]+Total phase tasks[Detail design,Project]+Total phase tasks[Tooling,Project] **Present in 2 views:** • Parameters • Auxiliaries **Used by:** • Project percent complete
#126 A	**Total tasks released** (Task) Total tasks released [Project] = Tasks released[Conceptual design,Project]+Tasks released[Detail design,Project]+Tasks

	released[Tooling,Project]+Sets reduced[Conceptual design,Project] **Present in 2 views:** • Parameters • Auxiliaries **Used by:** • Project percent complete
#127 A	**Uncertainty** (Dmnl) Uncertainty [Phase,Project] = MAX(0,(Target confidence[Phase,Project]-Design confidence[Phase,Project])) **Description:** *Equals to the gap between the design confidence and the targeted confidence for each phase of the project. the initial value of target confidence is assumed to be 100% for each phase which could be reduced due to schedule pressures.* **Present in 2 views:** • Auxiliaries • Auxiliaries **Used by:** • Confidence increase switch • Iteration switch • Reduction increase rate • Release switch • TC change rate
#128 D	**Unit cost** ($/(Person*Month)) = GET XLS CONSTANTS('Variables.xlsx', 'Data', 'c15') **Description:** *Unit cost depends on the phase and project* **Present in 2 views:** • Parameters • Performance measures **Used by:** • Phase cost