Navigating Complex Systems Design with the PEArL Framework

Dr Donna Champion
School of Business and Economics
Loughborough University
Loughborough
LE11 3TB
United Kingdom

Email: d.champion@lboro.ac.uk

Abstract

The problems and difficulties in complex systems design are more subjective and ambiguous than commonly acknowledged and soft systems tools and frameworks can help to address gaps in knowledge and support judgement throughout the design process. This paper describes the application of the PEArL framework, an intellectual device based on systemic principles, to support a team in a complex product environment manage the 'soft' challenges in complex systems design. The PEArL framework facilitated the creation and management of inter-disciplinary relationships in a pressurised business environment and provided a structure for validation of the way in which the design process had been undertaken.

Key words:

Complex Systems Design Systems Soft Systems

PEArL

1. INTRODUCTION

Traditionally the design of systems was undertaken by a relatively close-knit team of experts and stakeholders working to address a problem or take advantage of an opportunity. Today, the design of complex systems, such as aircraft, automobiles, trains, public services, manufacturing environments etc., require the collaboration of many hundreds of experts with different domain expertise and varied professional training backgrounds, often across a global supply chain. The systems being developed are created around multiple embedded software systems, and are designed in complex, dynamic organizational environments and this complexity creates a step change in the level of difficulty in understanding the emerging properties of a system and the potential interactions between systems at physical, software and human levels (Bonjour and Micaelli, 2010; Ellims et al, 2006). This increased systems complexity and the pressure to deliver products to market in shorter time scales increases the risk of undetected errors and emergent properties developing once a system or product is in operation (Sikora et al, 2012). The process of complex systems design is still managed within the traditional technology-focused, systematic methods that have previously been applied to standard systems design and found wanting. ‘Soft systems’ frameworks for systems design (e.g. Checkland and Holwell, 1998) offer a more holistic approach that can support the designers in thinking through the more subjective aspects of the process. This paper describes the application of the PEArL framework, (Champion, 2007, 2014; Champion and Stowell, 2001, 2003; Champion and Wilson, 2010), to support a team in a complex product environment within the Automotive sector manage the ‘soft’ challenges in complex systems design.

The paper first sets out a brief review of the literature and summarises the problems and challenges associated with the current approaches to designing and managing complex systems. The paper then argues for a systemic approach to inquiry in complex systems design environments and the PEArL framework is explained. The fourth section of the paper describes an application of the PEArL framework within a complex product design environment in the Automotive sector and the paper concludes by setting out some of the strategic advantages of working systemically, rather than systematically, when designing complex systems.

2. THE COMPLEX SYSTEMS DESIGN LITERATURE

2.1. The Limitations of Traditional Approaches to the Design of Complex Systems

The growing trend for complex products to incorporate semi-autonomous operation or self-diagnosis within the product (making them potentially much safer and easier to use) is creating levels of complexity for the manufacturer that had not been envisaged even five years ago (Broy et al, 2010). For example, self-parking systems or collision avoidance systems are appearing in automobiles as must-have features, but these raise important questions of legality, safety and societal trust issues as well as an assumed role and skill set of the operator. Consequently, this level of software and technology integration demands detailed functional safety to be an integral part of the whole design and manufacturing processes. As a result, the scale of information that needs to be managed, communicated and shared to design today’s complex products involves many disciplines where the product design team spans different parts of an enterprise including research, engineering, software development, manufacturing, service, dealerships and external to the enterprise, often across an extended supply chain of Small to Medium sized Enterprises (SMEs).

Some of the more flexible approaches to systems design such as Rapid Application Development (Boehm, 1991) or Agile Development techniques such as ‘Scrum’ (Sutherland, 2004) are unsuited to managing the development of very complex products as these methods do not facilitate the full traceability and detailed documentation required for safety critical systems. A more structured design approach is needed. The NASA (2011) Vee model is the basis for the International Standard for Systems Engineering: ISO 15288 and hence is widely applied in engineering firms as a broad framework for creating and building complex systems and products. The Vee model facilitates a component-based view of the systems under development which has, to date, made such projects manageable (Boehm and Basili, 2001; Valverde et al, 2011). But the component based approach to systems design also creates significant practical challenges when building complex systems, as unforeseen and unexpected relationships between systems often only become apparent when a design enters the build phase. Issues identified at a late stage of the development process are notoriously difficult and expensive to fix. For example, unexpected integration issues between software controllers embedded in ‘black box’ components sourced globally can lead to expensive changes and reworking of designs further down the work stream (Holmstrom and Sawyer, 2011; Valverde et al, 2011).

To address some of these issues, many Original Equipment Manufacturers (OEMs) are introducing Model-Based approaches to managing the design and development process for complex systems (Diaz et al, 2011). When applied in a practical engineering environment such approaches are often referred to as Product Line Engineering (PLE) (Lee and Kotonya, 2010). These approaches focus on achieving re-usability and commonality across large-scale complex programme lines where products are categorised according to specific customer segments and managed as a coherent group (Diaz et al, 2011). Model based/PLE approaches help to reduce costs and keep programmes to schedule through the reuse of design and testing information for particular groups of products where planned changes can be confidently predicted and planned (Dehlinger and Lutz, 2011; Diaz et al, 2011). But such frameworks are still difficult to implement on a large scale (Ghanam et al, 2010) and the focus is on developing stable statically configured products. More advanced research is focusing on developing approaches to dynamically configure complex products using ideas such as 'self-healing systems' (Garlan and Schmerl, 2002); ubiquitous computing environments (Sousa and Garlan, 2002) or more recently dynamic integration of physically identifiable features (Lee and Kotonya, 2010). All of these approaches are focused on addressing the problem within the product creation process, but for modern OEMs this framing of the problem is insufficient.

One of the main challenges to overcome in developing dynamic product configurations is how to manage and integrate the different sources and diverse forms of information not just during product creation but through a product's life cycle and across the supply and service chain. Such integration needs to encompass information from design and development software packages and large scale business packages such as project management; Enterprise Resource Planning (ERP) software and Customer Relationship Management systems. What is needed is a holistic approach to managing the organisational and business challenges across the various teams and disciplines who are engaged in creating technologies, platforms and programmes and managing these complex products out in the field. These communication, relationship building and maintaining activities are essential to successful design, but to date, there has been very little research into how to support and manage this aspect of complex systems design.

2.2 Organizational and Business Challenges in Creating Complex Products

In addition to managing the information associated with product creation for complex systems, it is important to manage the associated organizational and resource issues. PRojects In Controlled Environments (PRINCE, 2011) is a widely used approach to managing projects that follow traditional approaches to systems design. Dynamic Systems Development Methodology (DSDM, 2011) has also been developed to support agile development processes, though it is only relatively recently that agile practices have been implemented in large-scale environments such as manufacturing organizations. The challenges associated with scaling up agile approaches have become better understood recently (Beuche et al 2007), but there are still frictions that occur in practice. For example, some sub-systems of complex products need to be developed with a formal methods approach in order to comply with safety requirements, but integrating the design and development cycles for these sub-systems designed according to formal and traditional approaches with other systems developed through an agile approach is difficult (Gil and Tether, 2011). Creating an accompanying document trail is even more challenging. Finding ways of facilitating different approaches to design and testing within company processes would remove the temptation to do 'off-process', ad hoc design and better documentation trails would also provide relevant information to use in order to improve practice (Champion and Rivett, 2014).

When considering the academic literature, one gap that becomes apparent is the lack of guidance for managing communication issues and working practices during complex systems design and development. The only communication issues that are specifically identified in the literature have been those associated with prioritization issues and also release activities and most authors in the Engineering literature recommend a ‘single-capture process’ to deal with this issue (see Boehm, 1991; Charette, 1989; Egbert and Neve, 2001; Keil et al, 1999). In practice, in large manufacturing environments with simultaneous design and development of several product lines, a ‘single-capture process’ approach to requirements is infeasible. One of the impractical assumptions that is associated with adopting a single-capture requirements process is that it is considered a relatively easy task to assign ‘content ownership’ of any requirement (Gil and Tether, 2011). Identifying content owners for each requirement is increasingly difficult where a physical component might be controlled through software located in a completely different section of the architecture. A better approach is to identify those responsible for communication activities; cross-stream collaboration and feedback mechanisms. This holistic consideration is essential in order to gain understanding and insight into the otherwise unexpected relationships and emergent properties that can occur later in the development cycle. In engineering environments this approach still seems to be the exception rather than the norm, performance management frameworks tend to focus on product deliverables, rather than rewarding the less tangible but increasingly important soft skills.

In the literature a number of frameworks and design and development approaches have been suggested to address some of the ‘soft’ problems associated with the technology focused systems design approaches discussed above. (See Avison and Woodharper, 1990; Checkland, 1981; Checkland and Holwell, 1998; Champion et al, 2005; Petkov et al, 2013a). However, these approaches have been developed to address systems design for relatively small-scale development projects and can be criticised for not offering sufficient support for managing the organisational issues associated with the more technical design activities. Petkov et al (2013b) raise the important issue of *context* and discuss the variety of different organisational settings in which software projects and complex systems and products can be designed within and the need for flexible, multi-dimensional approaches. Recent research in the automotive sector suggests that current ‘soft-systems’ approaches do not currently offer comprehensive and practical support for an engineer in addressing the scale of information that design for such complex systems entails on a daily basis (Champion, 2012). There is still a need to incorporate the softer issues associated with roles, responsibilities, expertise and cross-functional relationship management into the more traditional structured design approaches.

The challenge of managing design and development work for complex systems across an organization is significant (Wastell, 2010). The challenge applies to small start-up companies developing new technologies, such as new motors or batteries, who need to establish they are ‘customer-ready’ so they can supply large OEMs. The challenge also applies to the OEMS themselves. Large OEMs need to manage the development of potentially many concurrent product lines, each with differing development timescales, with many possible variants and configurations. It is also important to integrate financial and project management information and teams to facilitate joint development work and cross-stream communication with both internal company stakeholders and external participants such as suppliers and component manufacturers. Managing the scale of communication and information exchange associated with such tasks is an area that remains largely ignored in the research literature. There is a need to focus on understanding the relationships, judgements and subjective decisions needed to successfully progress through the design process.

2.3 Challenges in Integrating Information across Functional Teams and Disciplines

The integration of new complex technologies, applications or systems into an already highly complex 'system of systems' is primarily managed through software. But within complex products and systems, some software applications will manage functionality in tightly coupled systems, whereas other software applications will control functionality across many disparate systems. This means that the implications of software change, and updates, can be difficult to predict and, in the pressure to deliver products on time, can often be underestimated even by relatively experienced project managers and engineers (Lancaster, 2012). Adding to this ambiguity, deciding the level of further testing and analysis that should be undertaken is an inexact judgement (Champion and Rivett, 2014). Such decisions are often taken across multi-disciplinary teams each with different delivery targets and performance criteria to meet. Most systems design frameworks focus on managing and measuring progress towards the deliverables, rather than considering the subjective and ambiguous activities inherent in the design process. The aim of this research was to consider if the PEArL framework (Champion and Stowell, 2001; 2003; Champion and Wilson, 2010) could support and facilitate the cross-disciplinary communication, and relationship building that was needed to successfully design complex products in a real-world fast moving product creation environment.

3. MANAGING AMBIGUITY IN COMPLEX SYSTEMS DESIGN

3.1 The Contribution of a Systems Approach

C. West Churchman (1968) asked the question “How can we design improvement in large systems without understanding the whole system, and if the answer is that we cannot, how is it possible to understand the whole system?” (p. 2). For Churchman, there were two aspects to answering this question: the application of science to understand what ‘improvement’ might look like, and also the application of ethical judgement (Ulrich, 2004). In messy, real-world social situations, subjective judgements are made throughout the design process and the idea underpinning the PEArL framework is to focus attention on these subjective aspects of a collaborative design process in order to develop and maintain the necessary communication and relationship network. The PEArL framework (P- Participants, E – Engagement, A- Authority, r –relationships[1] and L- Learning) is an intellectual device underpinned by Systems theory (see Champion and Stowell, 2001 for a full exposition of the underpinning theory) that offers practical support for managing a social inquiry process, such as complex systems design. The diagram in Figure 1 depicts the phases of inquiry from a Systems perspective, e.g. appreciating the problem, structuring and assessing ideas for action that may bring improvement (the design process) and then taking action (the build phase). In a real world situation the elements of PEArL will be in constant flux.

Figure 1: The PEArL Framework (From Champion, 2007).

The framework acts as a guide for planning, managing and then for reflecting back upon on the changing character of the design process as it progresses (Champion, 2007). Within

[1] The small ‘r’ for relationships has been used deliberately to emphasise this element as being the most important element of inquiry within human situations.

different organisational environments, the elements of PEArL (and how they change over time), gives insight into the way that the soft, subjective elements of design are managed and this can guide change in the management of teams, and drive improvements to communication and network building activities. The PEArL Framework has been applied in many different contexts from information systems design (Champion and Stowell, 2001), to supporting key workers in homeless hostels (Champion, 2007) and also in supporting practitioners adapt to legislative changes (Champion, 2014). Table 1 sets out the elements of the PEArL framework in more detail:

Table 1: The Elements of PEArL

The fluidity of collaborative inquiry is often overlooked in published research. It is insufficient to think of the elements of PEArL being set out in the planning phase of an intervention and then remaining constant. As the design process unfolds, participants change and modes of interaction and engagement can change. A key emphasis in the PEArL framework is the support offered to participants to create a shared appreciation of the problem situation at each stage of the design process. Traditional engineering approaches focus on constructing a description of the current system and then aim to abstract the requirements for a new system from their models. (For an example of this approach see Sommerville and Sawyer, 1997). In order to make sure the social and subjective elements of the design process are effectively managed, PEArL focuses attention on managing the changing relationships and thinking how to engage different people in the process. People can leave and join a project at any time in practical environments, e.g. due to changing jobs, parental leave or promotion. New group members need introducing to the problems and context all the time. The aim underpinning PEArL is that the framework facilitates design teams to *navigate* through the design process, accepting that the environment will be continually changing. The elements of PEArL help to maintain coherence in the face of a fluid practical reality.

SECTION 4: PEARL INTO PRACTICE

4.1 Cross-Functional Working Relationships

Managing the level of complexity in the design of products, such as a car, is challenging. Information and knowledge input is needed from across multiple teams from different locations and disciplines, and also from different concurrent development lifecycles ongoing in a large OEM (Original Equipment Manufacturer). Integrating new technology onto new, or existing platforms, requires alignment of not just technical design information, but also with project management information, finance systems, diagnostics and service systems too. Most OEMs designing automobiles apply variations of the Systems Engineering Vee model to organise their work streams (NASA, 2011). Figure 2 illustrates how the various different development processes relate to each other as a series of repeating processes of development.

Figure 2: Integrating technologies across platforms (V&V: Verification and Validation).

The degree of uncertainty at different stages of integration activity is often difficult to assess and in order to integrate information and knowledge assets across these various development streams, a great deal of cross-functional, cross-process and cross-discipline exchange and problem awareness is required. This is not just a matter of knowing who has certain roles and responsibilities in an enterprise, but knowing where the intellectual authority and practical experience for various issues can be found and building the relationships needed to 'manage the gaps' in understanding (Champion and Rivett, 2014).

The case described below was created through an action research project undertaken within a large international car manufacturer and OEM. The research discussed here was undertaken within an action research framework in order to produce theoretically rigorous knowledge outcomes and also useful practical outcomes for the collaborators. The academic researcher was actively engaged in the practical situation and designed the project collaboratively with participants from the OEM. The theoretical basis for the action research framework was the FMA model (Checkland and Holwell, 1998), where F is a Framework of ideas; M the Methodology applied and A the Area of concern. Checkland and Holwell (1998) set out a notion of *recoverability* where they argue to achieve theoretical rigour in action research it is essential to declare F, M and A before the research begins. Here the framework ideas applied is the PEArL framework, the underpinning theory has been previously published through a series of papers (Champion, 2007; Champion and Stowell, 2001, 2003; Champion and Wilson, 2010). The methodology is that of an interpretivist, where it is assumed that within social situations, meaning is continually being created and recreated, and interpretations are subjective. The area of focus was the automotive sector with a specific focus on the issues around project gateway evaluations. The case is set out within the PEArL framework below.

Participants: The project focused on the cross-stream relationships required in order to improve the decisions taken at project gateways. This was one of the areas highlighted as a problem area by the engineers in the OEM. The research team consisted of a small core focused on the project but wider engagement involved a network of over 50 personnel associated with the gateway decisions for a product in this OEM. Some participants did not work directly for the OEM, but belonged to supplier organisations and consultancies, all working with the OEM on product creation and delivery projects. As set out above, the PEArL framework was applied to gain an appreciation of how the five elements of the mnemonic applied to project gateways. The participants in the project are summarised in Figure 3.

Figure 3: An overview of the participants in the project

Engagement: In order to understand the soft, organisational issues each of the participants had experienced during gateway evaluations a series of workshops was held. At the workshops the focus was on how decisions at gateway events were taken, who was invited, who controlled the budgets and which teams ran the best (and worst) gateway meetings and why people judges them as good or bad. Nothing was recorded audibly and quotations were not attributed to individuals, people were encouraged to speak freely.

Authority: For the purpose of this example, we have focused on the various aspects of financial authority and intellectual authority that were discussed through the workshops. In the context of this OEM, any financial authority was usually held by a senior manager within the enterprise. If part of a design was contracted out to a supplier company, a senior manager from the OEM was responsible for ensuring the supplier met the conditions in the contract. This manager was often not the person who had written the original contract, but financial authority was limited to a relatively select set of senior managers as a means of control of costs. Additionally, the managers who held the financial authority for programme delivery were not usually perceived as holding the intellectual authority within the context of the decision being made. What became apparent was that at gateway evaluations in this OEM, the manager with the financial authority was present but those with intellectual authority were often not invited, as they were not deemed to be sufficiently senior enough to attend. Consequently the manager with financial authority could push for the project to proceed through the gateway and so adherence to budget was maintained. By excluding those with intellectual authority problems and issues were often not discussed, and difficult decisions were taken without a full consideration of the implications for downstream integration and build. This problem was further exasperated through the way in which individual and team performance was managed in the OEM.

Relationships: In this problem situation there were two inter-related aspects of the relationships that created problems: the issues associated with cross-stream tensions and also the approach to performance evaluations in the OEM. In any large project it is essential to manage cross-team relationships and communication. A number of individuals across the design work streams were recognised experts in their field, but performance measurement in the company (including the evaluation of contract success) was all centred on delivery of product –to time and to budget. In practice this meant there were disincentives across the OEM for people to be honest about problems and to try and fix the problems early on in the design process. Any delay in a project passing through a gateway could impact on an individual’s score for performance. The approach to performance evaluation was imposed on the OEM by its parent company. However the performance criteria had been originally devised for a manufacturing environment where products were much more mechanical in nature and was ill-suited to managing design work for complex products with embedded software. The OEM was experiencing a significant number of integration issues late in the design and build processes that had delayed programme launch on a number of occasions.

What was needed was a new approach to managing the difficult communication exchanges when a project was not progressing smoothly.

Learning Outcomes: In the current extremely competitive economic environment, one urgent issue for managers was the need to meet customer requirements in shorter timescales. There is only one opportunity to launch a new product, such as a car, and so only one opportunity to make a first impression; it is essential to get it right first time. In the automotive sector in particular, the speed of response needs to be very fast to meet customer expectations. There is often pressure on engineers to undertake accelerated development and to introduce new features onto a programme late in the programme. This means that there is all the more reason for there to be an environment and reward structure that actively values knowledge exchange across disciplines, specialisms and professions. The approach to performance evaluation made it very difficult for an individual to point out problems and issues in the OEM. To overcome this, a new cross-functional team of technical specialists was set up. This group were given the authority to over-ride gateway decisions if significant design issues arose that could cause costly mistakes during build or life of the product. The decision was taken by the group, not by a single individual. The technical specialist team included an independent member from outside the company and this independent person also attended the Steering Group meeting of senior managers who oversaw progress towards targets.

This new approach to managing difficult discussions was implemented and has been operational for over eighteen months. The process has evolved over time and senior managers now often attend the technical specialist group meetings to improve their understanding of the situation. The decision is always regarded as being a group one and so individuals are not impacted by a delay in a programme passing through a gateway. Several significant problems have been identified early by this group and the approach is beginning to establish its value to programme managers. The PEArL framework is applied to monitor and record who was involved through a process and how the participants were engaged. This approach has built up a better appreciation of how the enterprise approach decisions and led to evaluations having more of a workshop structure, than a formal meeting structure. It is reported by participants in the original study that this has improved the standard of debate.

4 CONCLUSIONS

The PEArL framework has been established as offering support to manage the ambiguities in complex systems design and also to support those making subjective judgements in fast-moving, market-driven environments. The impacts of the project set out here have been sustained with one large Original Equipment Manufacturer implementing PEArL to design new governance structures, cross-functional relationships and training initiatives to give better oversight of product creation. These new relationships have proved their value in acknowledged better co-ordination across engineering and back-office teams. One manager stated that PEArL had enabled the product creation teams to identify and drive business transformation towards a fully co-ordinated systemic design pipeline (See the Impact Case for the Research Evaluation Framework: Champion, 2014). The ability to understand the context of information and knowledge through leveraging cross-functional relationships facilitates the safe manipulation of information in the logical architecture throughout the life of a complex product. This ability to re-conceptualise and manipulate performance of a product in life and not just during the design phase is the ultimate in flexibility, offering manufacturers a means of upgrading and potentially gaining new value from products throughout their lifetime. The information held within traditional models and documents used in complex systems design currently do not give much insight into the context of decisions, or as to who was involved in making decisions throughout the design process. PEArL, based on systemic principles, can support ongoing reflection and the maintenance of essential relationships so the subjective reasons for judgements are understood, and potentially revisited in future design work. The effective development of cross-stream relationships is essential for high quality knowledge exchange, and in designing complex systems and products, a systems approach to support these more ambiguous and subjective activities has been established as being capable of adding value.

The next stage for this research is to develop tools to support reflection on each of the elements of the PEArL framework. Currently, although the mnemonic focuses attention on aspects of the organisational context required for problem structuring and solving in complex inquiries, the framework could be extended to give advice and support in considering how to manage each of the five elements through each stage of the inquiry, design and implementation process. Work is progressing in this area with applications in complex software development and infrastructure projects across the automotive sector. The 'Connected Car' of the future is dependent on complex software components being an integral part of the vehicle and this is raising new and challenging issues that will need to be addressed from social and policy perspectives as well as the technical challenges inherent in designing such a dynamically changing complex product. This research has been focused on extending current development approaches for complex software systems. The Systems community have substantial expertise in addressing such problems that, with more international collaboration and the diversity of knowledge and experience inherent in our community, could significantly contribute to making the step change in approach required for successful complex product design and management.

REFERENCES

Avison, D. E. and Wood-Harper, A. T. (1990). Multiview: An Exploration in Information Systems Development. New York: McGraw-Hill.

Beuche, D., Birk, A., Dreiger, H., Fleischmann, A., Galle, H., Heller, G., Janzen, D. John, I., Kolagari, R.T., von der Massen, T., and Wofram, A. (2007). "Using Requirements Management Tools in Software Product Line Engineering: The State of the Practice" in Software Product Line Conference, 2007. Pp. 84-96.

Boehm, B. (1991). "Software Risk Management: Principles and Practices" in IEEE Softw., Vol. 8, No. 1, Pp. 32-41.

Boehm, B. and Basili, V. (2001). "Software Defect Reduction: Top 10 list" in IEEE Computer, Vol. 21, No. 5, Pp. 61-72.

Bonjour, E. and Micaelli, J. (2010). "Design Core Competence Diagnosis: A Case from the Automotive Industry" in IEEE Trans. Eng Man Vol. 57, No. 2, Pp. 323-337.

Broy, M., Feilkas, M., Herrmannsdoerfer, M., Merenda,S. and Ratiu, D. (2010). "Seamless Model-based Development: From Isolated Tools to Integrated Model Engineering Environments", in Proceedings of the IEEE, Vol. 98, No. 4, Pp. 526-545.

Champion D. (2007). "Managing Action Research: The PEArL Framework". Systemic Practice and Action Research, Vol. 20, No. 6, Pp 455-465. DOI: 10.1007/s11213-007-9070-8

Champion, D. (2012). "A Framework for Managing Information during the Design and Development of Complex Systems" Conference of the UKAIS, March 26th -28th, New College, Oxford. Paper Available at: http://aisel.aisnet.org/ukais2012/ (Accessed: 12th June 2015).

Champion, D. (2014). "The PEArL Framework: Facilitating Change in Complex Social and Commercial Settings", Available at: http://results.ref.ac.uk/Submissions/Impact/343

Champion, D. and Stowell, F.A. (2001). "PEArL: A Systems Approach to Demonstrating Authenticity in Information Systems Design", Journal of Information Technology, Vol. 16, No. 1, Pp 3-12, ISSN 02683962.

Champion, D. and Stowell, F.A. (2003). "Validating Action Research Field Studies: PEArL", Systemic Practice and Action Research. Vol. 16, No. 1, Pp. 21-36.

Champion, D., Stowell F.A. and Callaghan, A. (2005). “Client-Led Information System Creation (CLIC): Navigating the Gap”, Info Sys Journal, Vol 15, Pp. 213-231.

Champion, D. and Wilson, J.M. (2010). “The Impact of Contingency Factors on Validation of Problem Structuring Methods”, Journal of the Operational Research Society, Vol. 61, No. 1, pp. 1420-1431.

Champion, D. and Rivett, R. (2014). “The Strategic Implications of Functional Safety Standard ISO 26262” 22nd Safety-critical Systems Symposium, Brighton, 4th – 6th February, 2014.

Charette, R.N. (1989). Software Risk Analysis and Management. New York, McGraw-Hill.

Checkland, P.B. (1981). Systems Thinking, Systems Practice. Wiley, Chichester.

Checkland, P.B. and Holwell, S. (1998). Information, Systems and Information System. Wiley.

Churchman, C.W. (1968). Challenge to Reason. New York: McGraw-Hill.

Dehlinger, J. and Lutz, R.R. (2011). “Gaia-PL: A Product Line Engineering Approach for Efficiently Designing Multiagent Systems” in ACM Transactions on Software Engineering and Methodology, Vol. 20, No. 4, Article 17, Pp. 17.1-17.27.

Diaz, J., Perez, J., Alarcon, P.P. and Garbajosa, J. (2011). “Agile Product Line Engineering – a Systematic Literature Review” in Software: Practice and Experience, Vol. 41, Pp. 921-941.
DSDM, (2011). Dynamic Systems Development Methodology Available at: www.dsdm.org Accessed on 20th March 2012.

Egbert, C. and Neve, P.D. (2001). “Surviving global software development” IEE Softw Vol 18, No. 2, Pp. 62-69.

Ellims, M., Bridges, J. and Ince, D.C. (2006). “The Economics of Unit testing” in Empirical Software Engineering, Vol. 11, Pp 5-31.

Garlan, D. and Schmerl, B. (2002). “Model-Based Adaptation for Self-Healing Systems,” Proc. 1st Workshop Self-Healing Systems (WOSS 02), ACM Press, Pp. 27–32.

Ghanam, Y., Andreychuk, D. and Maurer, F. (2010). “Reactive Variability Management in Agile Software Development” in Agile’10, Proceedings of the International Conference on Agile Methods in Software Development. IEEE Computer Society, Silver Spring, MD, Pp. 27-34.

Gil, N. A. and Tether, B.S. (2011). "Project Management and Design Flexibility: Analysing a Case and Conditions of Complementarity" Research Policy, Vol. 40, Pp 415-428.

Holmstrom, J. And Sawyer, S. (2011). "Requirements engineering blinders: exploring information systems developers' black-boxing of the emergent character of requirements" in European Journal of Information Systems, Vol. 20, No. 1, Pp.34-47.

Keil, M., Cule, P.E., Lyytinen K. and Scmidt, R.C. (1999). "A Framework for Identifying Software Project Management Risks" in Commun ACM, Vol. 41, No 11, Pp. 76-83.

Lancaster, G. (2012). Senior Manager Electronics and Software, Jaguar Land Rover, Personal Communication.

Lee, J. and Kotonya, G. (2010). "Combining Service Orientation with Product Line Engineering" available at: http://comp.eprints.lancs.ac.uk/2375/1/IEEE-SW-sopl.pdf Accessed January, 2013.

NASA (2011). NASA Systems Engineering Handbook. Available at: http://education.ksc.nasa.gov/esmdspacegrant/Documents/NASA%20SP-2007-6105%20Rev%201%20Final%2031Dec2007.pdf (Accessed, 4th January, 2012).

Petkov, D., Alter, S., Petkova, O. and Andrew, T. (2013a). "On the Suitability of Soft Systems Methodology and the Work Systems Method in Some Software Project Contexts", in International Journal of Information Technology and Systems Approach, Vol. 6, No. 2, Pp. 22-34.

Petkov, D., Petkova, O. and Andrew, T. (2013b). "On Some Lessons from Modelling Contexts in Complex Problem Solving in Information Technology", in International Journal of Information Technology and Systems Approach, Vol. 6, No. 4, Pp. 55-70.

PRINCE, (2011). PRojects in Controlled Environments. Information available at: http://www.prince2.com/ Accessed on 20th March 2012.

Sikora, E., Tenbergen, B. And Pohl, K. (2012). "Industry Needs and Research Directions in Requirements Engineering for Embedded Systems" in Requirements Engineering Vol 17, Pp. 57-78.

Sommerville, I. and Sawyer, P. (1997). Requirements Engineering: A Good Practice Guide. Chichester, John Wiley and Sons.

Sousa, J.P. and Garlan, D. (2002). "Aura: An Architectural Framework for User Mobility in Ubiquitous Computing Environments," in Proc. 3rd IEEE/IFIP Conf. Software Architecture: System Design, Development and Maintenance, Kluwer Academic Publishers, Pp. 29–43.

Sutherland, J. (2004). “Agile Development: Lessons Learned from the First Scrum” Available at: http://www.scrumalliance.org/resources/35 (Accessed June, 2013).

Ulrich, W. (2004). “In memory of C. West Churchman (1913–2004): Reminiscences, retrospectives, and Reflections” in Journal of Organisational Transformation and Social Change, Vol. 1, Nos. 2–3, Pp. 199-219.

Valverde, R. Toleman, M. and A. Cater-Steel, A. (2011). “A method for comparing traditional and component-based models in information systems re-engineering” in Info Sys E-Bus Man Vol. 9, Pp. 89-107.

Vickers, G. (1983). Human Systems are Different, London, Harper and Row.
Wastell, D. (2010). “Managing as designing: ‘opportunity knocks for the IS field’” *European Journal of Information Systems,* Vol. 19, No. 4, Pp. 422-435.