A SYSTEMIC FRAMEWORK FOR IMPROVING CLIENTS' UNDERSTANDING OF SOFTWARE REQUIREMENTS

Research in Progress

Wing, Jeanette, Durban University of Technology, Durban, South Africa, jwing@dut.ac.za

Andrew, Theo, Durban University of Technology, Durban, South Africa, tandrew@dut.ac.za

Petkov, Doncho, Eastern Connecticut State University, Willimantic, CT, USA, petkovd@easternct.edu

Abstract

This research proposes a systemic framework for the understanding of client requirements in Information System development that is particularly relevant for project contexts characterized by diversity of stakeholder values and significant complexity. In spite of the strong research tradition associated with Soft Systems Methodology and the growing interest in the Work System Method, the level of use of those by practitioners is not high as such situations require harnessing the strengths of more than one methodology. The paper explores the selection of techniques from three systems methodologies and mixing them to be applied in the process of requirements understanding by clients. The mixing of methods from various methodologies is justified through the principles of Critical Systems Practice and the process of their use is guided by Action Design Research. The contribution of the paper for the field of Information Systems is in the proposal of a more powerful framework for promoting organizational learning about software requirements understanding by clients.

Keywords: software requirements, soft systems thinking, critical systems practice, action design research

1 Introduction

IT project success rates show a need for improvement (Eveleens and Verhoef, 2010; Standish Group, 2013). According to Whitaker (2010), problems of project failure are attributed to requirements being incomplete, inaccurate, inconsistent, or missing. User participation in the information system development (ISD) process has often been hypothesized as being a solution to the above issues, and a contributor towards systems success, and yet similar problems appear to be ongoing. Hirschheim and Klein (2012) point that user participation in the ISD process was the topic of application of the original socio-technical systems ideas pioneers and they list socio–technical systems (STS) as one of the key schools of thought in the field of Information Systems. They conclude also that while STS research on the topic had mixed empirical results, it was clear that user involvement and participation are important in the systems development process. The problem of user involvement has been investigated by many researchers over the years but was brought again to the attention of the IS community by a thought provoking paper by Markus and Mao (2004). They call for rich participation by clients in software projects stating "that it is not the mere fact or quantity of participation that matters, but also the quality of participation. In particular, we describe participation activities in terms of participants' behavioral

experiences (the types and richness of participation activities) and in terms of the design choices made by change agents (the method or techniques and conditions of participation)" (Markus and Mao, 2004: 536).They further point that rich participation is encouraged by using analysis techniques that are appropriate for users with non-specialist IT knowledge, by choosing analysis techniques that capture socio-technical requirements in addition to functional requirements, and in using a "facilitation" approach rather than a "technical expert" approach to participation. Participation therefore needs to be such that the influence of the users' participation is reflected in the resulting ISD project and its implementation.

Further extension of the ideas of Markus and Mao (2004) is provided by Alter (2009) who calls for "project collaboration" rather than "user participation". Alter uses the term "project collaboration" as a way to clarify the extent to which users should be involved, and the need for a comprehensive consideration of all aspects of the project rather than just the technical issues. His focus is on extending the call by Markus and Mao (2004) for a more detailed investigation of participation to project collaboration which "explicitly focuses on work system projects rather than projects whose main goal is the development and implementation of software/hardware configurations" (Alter, 2009: 12). The Work System Method (WSM) (see Alter, 2006, 2008) is a more recent socio-technical approach developed to reduce the gap between the understanding of software requirements by clients and developers. It focuses as a starting point of analysis on the work system, on the organization and not on the technical aspects of the information system to be developed. The challenge to apply WSM ideas for studying client participation in IS projects and finding ways to make it more meaningful was one of the motivations for this research.

The issue of client understanding of requirements becomes especially complicated for projects characterised with multiple stakeholders with diverse interests and larger project complexity. Typically the outcomes of such IT projects involve changes in the business process and work practices of organizations (see Alter and Browne, 2005). In such projects the clients often have a vague idea about their perceptions of the requirements and as a result those need to be defined through a careful interaction between the developers and the clients. The need to carefully address the social construction of software requirements is demonstrated by Holstrom and Sawyer (2011: 44): "The RE community appears to have continued to focus on the artifact, and thus failed to exploit the potential value of social constructivist approaches, which are highlighted by our findings. There seems to be a clear need for more of a social process perspective, and awareness of the need for multiple perspectives of IS requirements". Such situations involve necessary organizational learning and the application of methods that support such learning. This was another motivation for the development of the framework proposed in this research in progress paper.

Organizational learning has been promoted among other methods also through systems thinking. According to Mora *et al.* (2007: 1) the IS discipline "has been driven by a dual research perspective: technical (design engineering oriented) or social (behavioural focused). This duality of man-made non-living (hardware, software, data, and procedures) and living systems (human beings, teams, organizations, and societies), the multiple interrelationships among these elements, and the socio-cultural-economic-politic and physical-natural environment, make IS a complex field of enquiry" and hence they motivate that the systems approach is most likely to complement the technical analysis of a problem. Such a conclusion was based on the apparent lack of systems thinking in IS research at the start of the new millennium. While there has been considerable progress towards the use of Soft Systems Methodology (SSM) as a sense making tool in complex IT problems (following Rose, 1997, Rose and Haynes, 1999), little success has been registered in linking SSM to more formal software requirements elicitation

techniques. In spite of the strong IS research tradition in SSM and the growing interest in other systems methodologies like the Work System Method (WSM), the level of their practical use by practitioners are not high, probably because more complex project situations require harnessing the strengths of more than one methodology. The paper considers the potential selection of techniques from several systems methodologies and mixing them in an intervention aiming at the understanding of software requirements by clients in a process of organizational learning.

The possibilities for combining WSM with SSM for more complicated project situations were explored in Petkov *et al.* (2012). While such a combination allows for compensation by SSM for the fact that the WSM provides insufficient means of analysis of the political and cultural aspects of an IT problem there are situations in which the need to reveal sources of motivation, knowledge and legitimacy in the analysis of requirements require the use of the boundary questions of Critical Systems Heuristics (CSH) (see Ulrich, 2005).

The *purpose of the paper* is to provide a justification for the use of several techniques from three systems methodologies, SSM, WSM and CSH in a framework to bring about client understanding of requirements in complex IT projects. The use of several techniques in the proposed framework is based on the principles of Critical Systems Practice (CSP) (see Jackson, 2003), a meta-methodology enabling mixing of methods in a systemic intervention. Since the result of the application of the framework is both a product of design and action research, the process of applying the the proposed mix of techniques will be guided by Action Design Research (ADR) (Sein *et al.*, 2011). Action Design Research is a recently proposed meta-approach to design science that integrates it with Action Research (see also Papas, O'Keefe and Seltsikas, 2012). The refined process of ADR resolves the problems in design research resulting from the separation of design from implementation as is pointed in the introduction of Sein *et al.* (2011).

The theoretical contribution of the paper for the field of Information Systems is in the proposal of a more powerful framework for promoting organizational learning about understanding of software requirements in a project situation characterised by diversity of stakeholder values and significant complexity in comparison to the use of only one systems methodology. From a practical point of view the chosen few methods are simpler to use than the whole methodologies to which they belong. The paper proceeds with an overview and justification of the techniques from the three systems methodologies to be mixed for better understanding of software requirements by clients, followed by a summary of the framework and a conclusion.

2 On elements from three systems methodologies considered in the framework for software requirements understanding

The systems approach describes the general properties of a system as: "wholeness, purposefulness, emergence, organization, hierarchical order, interconnectedness, competence, information based controllability, progressive mechanization, and centralization."(Mora *et al.*, 2007: 3). Midgley (2011: 5) states two assumptions that are fundamental to most (if not all) systems approaches: "everything in the universe is directly or indirectly connected to everything else, and, we cannot have a 'God's eye view' of this interconnectedness: our understandings in any situation are inevitably limited." To address the limitation of our understanding of a project situation we propose a multiple perspective framework that combines the strengths of methods from the systems methodologies discussed next. According to Jackson (2003) these methods all belong to soft systems thinking as a strand in the systems approach. Soft systems

thinking as applied to information systems promotes organizational learning and recognizes the complexity of the social, cultural, and power dynamics within which an information system is required to function (see Jackson, 2003). Recognizing and working within this social context is key to soft systems thinking.

When information systems support simple transaction processing and functions close to the daily processing tasks in a business it may be possible to use a hard systems approach such as methods from Operations Research when eliciting requirements. As stated by Jackson (2003: 62), hard systems approaches are applicable "when world views converge and the problem becomes one of finding the most efficient means of arriving at agreed-on objectives". When information systems contexts involve too many elements and are characterized with greater diversity in the stakeholder interests, then soft systems approaches are more appropriate as they recognize the world as complex, and also perceived differently by each observer (Stowell, 2009). The emphasis of soft systems approaches in that case is based on an understanding of the problem developed through a learning process rather than seeking a single optimal solution.

Among the most widely known soft systems approach is Soft Systems Methodology (SSM). Given the misinterpretation and incorrect implementation of SSM highlighted by Holwell (2000), it is important to understand and use SSM within the context in which it is framed. SSM facilitates the emergence of the problem as a result of debate. Tools used to facilitate the debate among stakeholders are rich pictures, CATWOE analysis and root definitions. Initially Checkland formulated the process of SSM as a seven stage model that was applied iteratively in order for the problem to emerge (Mode 1 SSM according to Checkland (1999)). The debate in this process should involve all stakeholders and consider issues of power and politics that will exist within the social structure which resulted from the experience gained in using SSM into SSM Mode 2 (see Checkland and Scholes, 1990). It separated the technical stream of inquiry into a problem from the stream of social, cultural and political inquiry. More details on ways in which SSM can be applied are provided in Checkland and Winter (2006). A four stage analysis process is summarized including a rich picture depicting the problem, analysis of the social characteristics of the situation, analysis of the power and politics associated with the problem and an analysis for the intervention itself. The first three stages relate to the content and context regarding the situation of concern and the last one relates to the process that will be followed.

The above four stages require considerable skills from the facilitator of the intervention and the participating stakeholders. In order to craft a more user friendly framework, only the techniques of rich pictures, CATWOE analysis and root definitions are considered from SSM. While SSM assumes participation of multiple stakeholders, it is not explicit in the process of determining stakeholders and the essential elements of a problem situation. A suitable technique for identifying the essential elements is the Work System Snapshot from the Work System Method (WSM) which is proposed for inclusion in the framework.

The Work System Method is a pragmatic approach designed to enable business professionals to describe a work system as a problem situation (Alter, 2006, 2008). WSM and its more recent extension into Work System Theory (WST) were proposed according to Alter (2013) to address the differences between the espoused theory in IS as a systems discipline and its "theory in use" as a non-systems discipline. The work system method includes a static representation of a problem situation (called the Work System Framework), dealing with the organizational external environment, its strategic objectives, infrastructure and the elements of the snapshot: customers, products, work practices, participants (or stakeholders), information and technology. The evolution of a work system in time is studied through the work system

life cycle model (see Alter, 2006; Alter, 2008). The Work System Method includes three stages of analysis, dealing with the statement of a problem, analysis of possibilities and provision of recommendations for improvement and their justification. The work system theory (WST) proposed in Alter (2013) is an extension of the above as an integrated body of theory following the typology of Gregor (2006) that includes a Type 1 analytical theory (the work system framework) and a Type 2 explanatory theory (the work system life cycle model), which in combination give the basis of a Type 5 design theory (WSM).

A common criticism of SSM and WSM is the lack of techniques in them for promotion of stakeholder participation and for their motivation. Critical Systems Heuristics CSH) is suggested for inclusion in the proposed framework as a possible way to address those shortcomings of SSM and WSM.

CSH was proposed in 1983 by Ulrich as a methodology for critical systems thinking as a means for planning. An important notion in Ulrich's theory is boundary judgements (see Ulrich, 2005). These are the facts and values to be considered that define the boundaries of the planning effort. According to Ulrich boundary judgments need to be concerned with the sources of motivation, sources of power, sources of knowledge and sources of legitimacy. Ulrich (2005) proposes for each of the four groups above three boundary questions. The first question in each group is related to a social role e.g. who ought to be the plan's client? The second question relates to role's specific concerns e.g. what ought to be the plan's purpose? The third to the problem of dealing with the conflicting concerns that are part of social reality e.g. what ought to be the underlying measure of improvement? Each question can be answered from different viewpoints, not only from the standpoint of those involved, but also from the perspective of those concerned and potentially affected.

3 A systemic framework for facilitating software requirements understanding in the early stages of a complex project

A framework is designed with a specific purpose, and is often depicted in a picture that shows implications of related concepts. The framework is a meta-model for assisting methodology users (Jayaratna, 1994). In this case the framework aims to bring individuals to an understanding of software requirements. To face the complexity of a problem situation that involves multiple stakeholders with diverse interests and a high number of elements it is necessary to draw on a mix of methods from different systems methodologies as is shown in Figure 1. The mixing of methods within the framework is justified using critical systems practice (CSP), developed in the field of Critical Systems thinking (Jackson, 2003). Critical systems practice is considered as a meta-methodology as it involves in its process a deliberate attempt to reflect theoretically about what methods are appropriate for use, and when to use them.

The three commitments of CSP: critical awareness, improvement and pluralism (Jackson, 2003) are the *guiding principles in the justification of the framework.* In this case 'multi-paradigm multimethodology is used which means that parts of different methodologies can be used together in the same problem intervention. The process of applying CSP involves four phases which are completed keeping in mind the three commitments mentioned above. These phases are: a creativity phase, a choice phase, an implementation phase and reflection. While the process of CSP is useful for a general framing of the inquiry in a systemic intervention it does not include two issues of importance: the need for evaluation of the validity of a framework like the one proposed here and also the issue of generalization of the findings.

The above shortcomings of the process of CSP are addressed in another meta-methodology proposed recently in the field of Information Systems. It was specifically developed for integrating action research with design science and is called action design research (ADR) (see Sein *et al.*, 2011). A quick comparison between the two shows some similarities. The first stage of ADR is labelled Problem formulation and it includes activities that are slightly similar to what is performed in the creativity and choice phases of CSP but are less articulated than them. Its second phase integrates design and evaluation work and corresponds to the implementation phase in CSP. One may argue that though evaluation was not explicitly included in the CSP process it has always been considered implicitly as part of a methodology in Operational Research and Systems Thinking. The next stages in both meta-methodologies are both dedicated to reflection while fourth stage in ADR is dedicated to the formalization of learning through generalization of the outcomes along the guidelines for design research in Hevner *et al.* (2004).

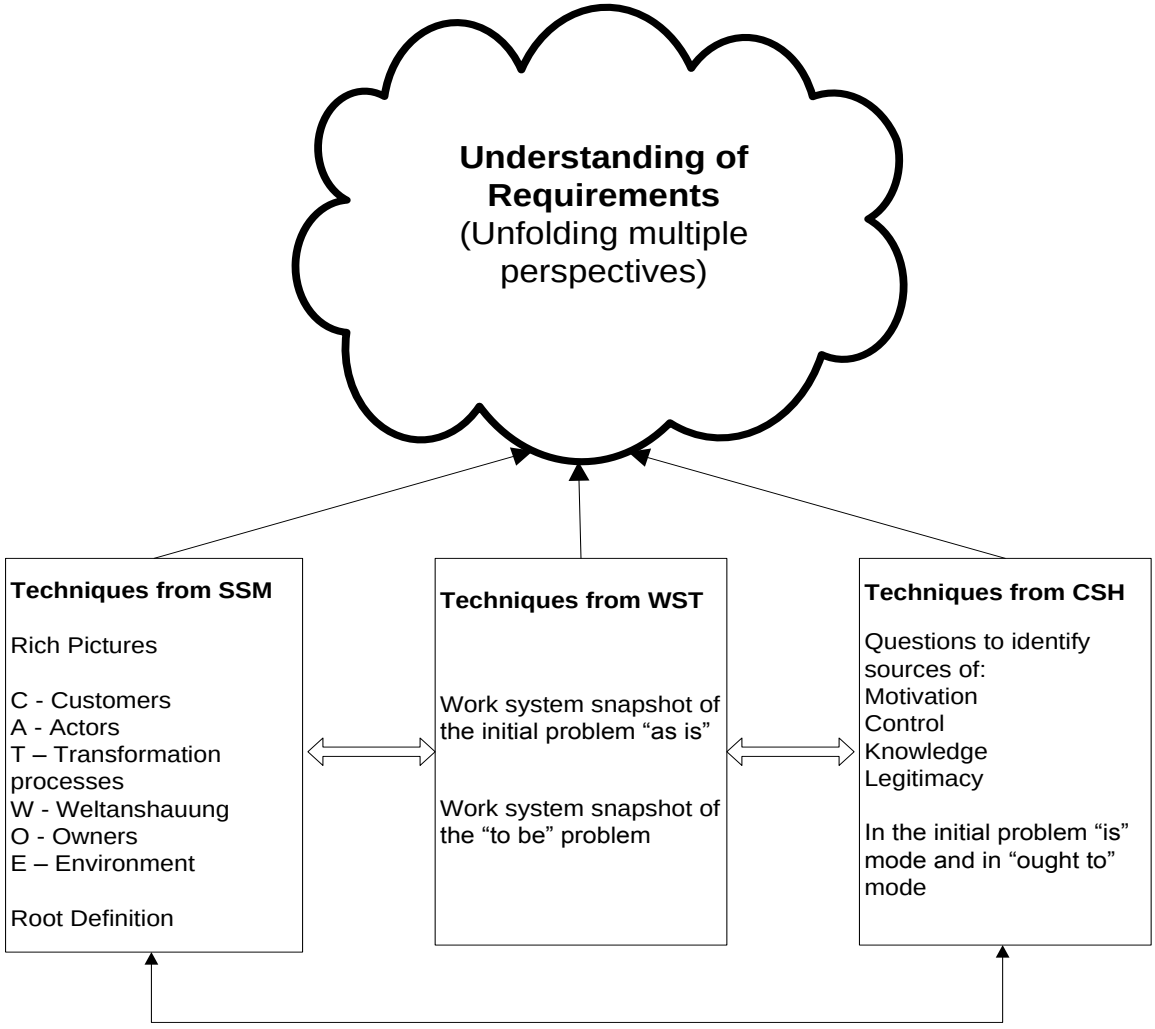

Figure 1. Framework to facilitate client understanding of requirements

Another similarity between CSP and ADR is the significant role of Action Research in both of them as the application of CSP is entirely guided by Action Research (Jackson, 2003) while ADR integrates the phases of Design with Action Research in creating the resulting IT artefact. Both CSP and ADR are seen as meta-theories influencing this research, but they are each used in a different way. CSP is used to justify the methods used in the framework, and the mixing of methods in drawing together the framework. ADR is used to explain the process of applying the methods of the framework in a specific project intervention.

The understanding of the problem situation and of client requirements emerges as a result of applying the framework. The stages and principles of ADR facilitate the learning that needs to take place in the implementation of the framework. The first iteration through the learning cycle is expected to bring to light the divergent views, as well as identify any stakeholders that may not necessarily have been considered. It is these different views that need to be reconciled in order to have a common understanding of the problem emerge. It is envisaged that different steps in the learning cycle will receive more attention according to the actual circumstances of a project to which the framework is being applied. The guiding learning cycle is expected to support the unfolding of multiple perspectives and promote mutual understanding of the problem. The outcome from applying the proposed ideas as a design artefact is the knowledge about software requirements generated through an intervention guided by the framework for understanding of client requirements which in itself will be shaped in its final form through the interaction between the client and the developers.

Using parts of SSM, WST and CSH in combination to work towards a better understanding of the requirements for software development provides benefits as each individual method brings something to the framework. Cases of overlap in ideas raised from some methods can be used for validation of the importance of a point raised. Triangulation will be used to cross reference points raised through various methods, identifying their importance, and also the different views on similar issues. An example would be to compare the CATWOE analysis from SSM with the work system snapshot from WST. Triangulation of data is often discussed in the qualitative, quantitative debate (Venkatesh, Brown and Bala, 2013). Triangulation is therefore an important feature supported by the framework leading to potential improvement of the reliability of results.

4 Conclusion

The proposed systemic framework for understanding of client requirements in complex IS projects is an interventionist approach justified by the principles of Critical Systems Practice and guided by the process of Action Design Research. Its benefits are in the integration of several simple methods originating from different systems and exploiting the results from each method as an input to other methods or techniques. This research builds upon past work in this area, extending it by showing how CSP can be used for justification of the methods to be included in a framework for improved understanding of user requirements. ADR is used to justify the process of applying these methods. For space reasons the justification for why other methods were not considered in the framework has not been included in this research in progress paper but that topic was addressed in our project.

The proposed framework benefits from combining the rigour of CSP in selecting and justifying the methods that are included and from the flexible process for implementation of ADR. This integrated use of two meta-methodologies is one of its theoretical contributions. The proposed mix of methods from the

three methodologies, SSM, WST and CSH was not applied before to problems in IT to the best of our knowledge and that is part of the practical contribution of this work.

The purpose of the framework is to facilitate better client understanding of ISD requirements, and thereby gain meaningful and valuable ideas from stakeholders by supporting a process of organizational learning. The framework provides clear guidance on what methods are applied. Their sequencing depends on the needs of the intervention guiding the judgment of the facilitator. The framework is intended to promote accommodation of different stakeholder needs and interests and the emergence of understanding of client requirements. The framework considers the ensemble view of the IT artefact (see Sein *et al.*, 2011) embedded within a complex social environment and which encompasses the interactions between the technology and the people using the technology, as well as the social and cultural perspectives of the people concerned. Further research work is needed on the validation of the framework and its testing in field conditions. The results will provide possible insights on whether the framework enables meaningful client participation in the understanding of software requirements. Another line of investigation would be whether the choice of following the process of ADR suggested here provides practical and theoretical benefits compared to adhering to the stages in CSP which is based on action research only.

Acknowledgements

The authors are grateful to the anonymous reviewers for their valuable suggestions that helped improve the paper.

References

Alter, S. 2006. The Work System Method: Connecting People, Processes, and IT for Business Results. Larkspur, CA: Work System Press.

Alter, S. 2008. Defining information systems as work systems: implications for the IS field. European Journal of Information Systems, 17 (5): 448-469.

Alter, S. 2009. Project Collaboration, not just User Participation. In: Proceedings of Americas Conference on Information Systems. San Francisco, California, 6th-9th August 2009. 1-12.

Alter, S. 2013. Work system theory: overview of core concepts, extensions, and challenges for the future. Journal of the Association for Information Systems: 72.

Alter, S. and Browne, J. 2005. A Broad View of Systems Analysis and Design: Implications for Research. Communications for the Association for Information Systems, 15 (2005): 981-999.

Checkland, P. 1999. Systems Thinking, Systems Practice: Includes a 30-year retrospective. 1st ed. Chichester, West Sussex, England: John Wiley & Sons Ltd.

Checkland, P. and Scholes, J. 1990. Soft Systems Methodology in Action. Chichester: John Wiley & Sons Ltd.

Checkland, P. and Winter, M. 2006. Process and content: Two ways of using SSM. Journal of the Operational Research Society, 57 (12): 1435-1441.

Eveleens, J. L. and Verhoef, C. 2010. The rise and fall of the chaos report figures. IEEE software, 27 (1): 30-36.

Gregor, S. 2006. The nature of theory in information systems. MIS Quarterly: 611-642.

Hevner, A. R., March, S. T., Park, J. and Ram, S. 2004. Design Science in Information Systems Research. *MIS Quarterly*, 28 (1): 75-105.

Hirschheim, R. and Klein, H. K. 2012. A Glorious and not-so-short history of the information systems field. *Journal of the Association for Information Systems*, 13 (4): 188-235.

Holstrom, J. and Sawyer, S. 2011. Requirements engineering blinders: exploring information systems developers' black-boxing of the emergent character of requirements. European Journal of Information Systems, 20: 34-47.

Holwell, S. 2000. Soft Systems Methodology: Other Voices. Systemic Practice and Action Research, 13 (6): 25.

Jackson, M. C. 2003. Systems Thinking: Creative Holism for Managers. 1st ed. Chichester: John Wiley & Sons Ltd.

Jayaratna, N. 1994. Understanding and evaluating methodologies: NIMSAD, a systematic framework. McGraw-Hill, Inc.

Markus, M. L. and Mao, J.-Y. 2004. Parrticipation in Development and Implementation - Updating an Old, Tired Concept for Today's IS Contexts. Journal of the Association for Information Systems (JAIS), 5 (11-12): 514-544.

Midgley, G. 2011. Theoretical pluralism in systemic action research. Systemic Practice and Action Research, 24 (1): 1-15.

Mora, M., Gelman, O., Forgionne, G., Petkov, D. and Cano, J. 2007. Integrating the fragmented pieces of IS research paradigms and frameworks: A systems approach. Information Resources Management Journal (IRMJ), 20 (2): 1-22.

Papas, N., O'Keefe, R. M. and Seltsikas, P. 2012. The action research vs design science debate: reflections from an intervention in eGovernment. European Journal of Information Systems, 21 (2): 147-159.

Petkov, D., Alter, S., Wing, J., Singh, A., Petkova, O., Andrew, T. and Sewchurran, K. 2012. Project Contexts and the Possibilities for Mixing Software Development and Systems Approaches. Research Methodologies in Systems/Software Engineering and Information Systems: Philosophies, Methods and Innovations, IGI Global: 360-375.

Rose, J. 1997. Soft systems methodology as a social science research tool. Systems Research and Behavioral Science, 14 (4): 249-258.

Rose, J. and Haynes, M. 1999. A soft systems approach to the evaluation of complex interventions in the public sector. Journal of Applied Management Studies, 8 (2): 199-216.

Sein, M. K., Henfridsson, O., Purao, S., Rossi, M. and Lindgren, R. 2011. Action Design Research. MIS Quarterly, 35 (1)

Standish Group. 2013. The CHAOS Manifesto–Think Big, Act Small. Available: http://www.versionone.com/assets/img/files/CHAOSManifesto2013.pdf (Accessed 27 June 2014).

Stowell, F. 2009. Soft Systems and research. Kybernetes, 38 (6): 879-896.

Ulrich, W. 2005. A brief introduction to critical systems heuristics (CSH). Available: http://projects.kmi.open.ac.uk/ecosensus/about/csh.html (Accessed 26 November 2011).

Venkatesh, V., Brown, S. A. and Bala, H. 2013. Bridging the qualitative-quantitative divide: Guidelines for conducting mixed methods research in information systems. MIS Quarterly, 37 (1): 21-54.

Whitaker, K. 2010. Principles of Software Development leadership. Applying Project Management Principles to Agile Software Development. Boston, USA: Cengage Learning (course technology).