Issues in Informing Science and Information Technology **Volume 12, 2015**

Cite as: Davey, B., & Parker, K. (2015). Requirements elicitation problems: A literature analysis. *Issues in Informing Science and Information Technology, 12,* 71-82. Retrieved from http://iisit.org/Vol12/IISITv12p071-082Davey1929.pdf

Requirements Elicitation Problems: A Literature Analysis

Bill Davey
RMIT University
Melbourne, Australia

Kevin R. Parker
Idaho State University
Pocatello, Idaho, USA

Bill.Davey@RMIT.edu.au **parkerkr@isu.edu**

Abstract

Requirements elicitation is the process through which analysts determine the software requirements of stakeholders. Requirements elicitation is seldom well done, and an inaccurate or incomplete understanding of user requirements has led to the downfall of many software projects. This paper proposes a classification of problem types that occur in requirements elicitation. The classification has been derived from a literature analysis. Papers reporting on techniques for improving requirements elicitation practice were examined for the problem the technique was designed to address. In each classification the most recent or prominent techniques for ameliorating the problems are presented. The classification allows the requirements engineer to be sensitive to problems as they arise and the educator to structure delivery of requirements elicitation training.

Keywords: requirements elicitation, requirements gathering, software development, systems development, software failure.

Introduction

Although often overlooked or assumed to be a straightforward task (Kitapci & Boehm, 2007), requirements gathering for software development projects is the most critical phase of any software development methodology. Determining software requirements is the cornerstone to any successful project. Requirements are critical for defining, estimating, and managing any project (Raisinghani, Riggen, & Ryan, 2014). Quality requirements are essential to the success of any software development project, regardless of whether it is based on agile or traditional methodologies. The requirements gathering process followed in a traditional frameworks and agile approaches may differ, but the criteria for quality requirements remain constant (Nee, 2009).

An important part of requirements gathering is obtaining requirements from people: requirements elicitation (RE). Without an accurate understanding of what the stakeholders really want and need, projects cannot develop what the stakeholders desire. Thus, requirements elicitation is a crucial first step in the software development process (Kitapci & Boehm, 2007). Failure of information systems is common, and effective requirements elicitation is an important factor in avoiding system failure. The most efficient and well-engineered system must be useful to end users and that

Editor: Eli Cohen

is contingent on the right specifications being obtained in the first instance. The purpose of this paper is to present a view of what can go wrong in requirements elicitation based on a study of published research designed to improve RE. First some justification will be made for the contentions that systems failure is common and that good RE is important. Then the results of the literature analysis will be explained.

Failure and Abandonment Rates

The industry would like to be able to use the term ‘engineering’ when talking about professional activities. While civil engineering projects are abandoned or sometimes fail, collapsing bridges or buildings tend to make the international news. This is also the case with aeronautical engineering problems. Using these other engineering areas as a benchmark, information systems figures for failure are abysmal.

The Standish Group has conducted large-scale surveys of project managers from 1994 until 2008 and has reported the findings in the ‘Chaos reports’. These surveys classified project outcomes as ‘success’, ‘challenged’ or ‘failed’. A summary by Eveleens and Verhoef (2010) of the first five Standish surveys is shown in Table 1 and indicates that outright success of information systems projects is rare.

Table 1: Success rates of IS projects taken from Eveleens and Verhoef (2010, p. 2)

YEAR	SUCCESS	CHALLENGED	FAILED
1994	16%	53%	31%
1996	27%	33%	40%
1998	26%	46%	28%
2000	28%	49%	23%
2004	29%	53%	18%

The Standish surveys have been replicated by others in the United States with similar outcomes (Emam & Koru, 2008). These empirical results are supported by authors of studies of information systems failure (Anbari, 2003; Basili & Boehm, 2001; Briggs & Gruenbacher, 2002; Dalcher & Drevin, 2003; Eberlein & Leite, 2002). Studies have found very similar problems in other countries. For example, Sauer looked at the UK. In 565 projects they found 5% abandoned and 55% over budget and 20% of projects delivered with less than 80% of specifications (Sauer & Cuthbertson, 2002). Also in the UK a study in 2000 found that “Out of 1,027 projects covered only 130, or 12.7%, were successful” (Taylor, 2000, p3). These examples are consistent with several investigations of failed projects in Australia (Brouwer, 2011; Pearson, 2012), the Netherlands (Tut, 2009) and China (Xiangnan, Hong, & Weijie, 2010).

These gloomy figures for information systems project failure, although consistent, should be read with some note of warning. The most common definition of failure in this context is when a project goes over time, over budget or does not meet its objectives in any way. There is some evidence (Glass, 2002, 2005; Savolainen, Ahonen, & Richardson, 2012) that ‘failed’ projects are sometimes delivered and go on to meet company needs. While this definition of project failure is seen by some authors to be inappropriate there seems to be no commonly accepted definition that distinguishes between outright failure and poor estimation of cost or time. Some authors make a distinction between the success of the project management and the success of the project, for example, Savolainen, Ahonen, and Richardson (2012). Of course this does not apply to the case of abandonment of the project where nothing is delivered.

Cost of Failure

The cost of this failure is enormous. Estimates of the cost of failure of systems in the USA have been between $59 billion and $81 billion (Dalcher & Drevin, 2003). More recent studies showed this had risen to between $84 billion (Reichental, 2006) and $100 billion (Miller & Luse, 2004). Miller and Luse (2004) report that a study by Clark (2002) of 134 companies in the United States, United Kingdom, Africa, and Australia, found that 56 percent of the companies had cancelled at least one IS project during the last year at an average cost of US $13.6 million.

A very recent case in which a thorough study has been conducted by an independent governmental authority is that of an education system in the Australian State of Victoria. The system, called 'the Ultranet', was investigated by the State Auditor General, D. D. R. Pearson, and so the findings were the result of an unfettered investigation. It was found that the system, originally to have cost $60.5 million Australian dollars but was now expected to cost about $180 million by June 2013. It was also found that the system was used by only 10% of students and that only 27% of teachers had logged in from July 2011 to May 2012 (Pearson, 2012).

In the same jurisdiction a study by the Ombudsman, G. E. Brouwer, of ten ICT enabled projects found that $1.44 billion Australian dollars had been spent in excess of budget and that several had failed or been abandoned with no outcome for the expenditure (Brouwer, 2011). The detailed study of ten projects included found that none of the projects had met expectations and that the cost of their failure was significant:

> On average, projects will have more than doubled in cost by the time they are finished. Two of the projects will have more than tripled their original budgets in order to reach completion: CRIS, originally budgeted at $22 million, has cost $70 million; and Link, originally budgeted at $59 million, would cost $187 million if it were to be completed. Together, the two largest projects will require almost $600 million more than originally planned: myki, originally costed at $999 million, will require at least an additional $350 million to complete and HealthSMART, originally budgeted at $323 million, will require an additional $243 million to complete. (Brouwer, 2011, p. 4)

Clearly the failure of information systems projects is a global problem and of significant size.

Persistence of Failure

Failure of IS projects is not only at high levels, but has been persistent over many years. In 1997 Sauer summarised ten years of research by concluding:

> IS failure remains an important, unsolved problem. Failures have been (1) persistent, insofar as failure rates have not declined over the last 10 years, (2) pervasive, insofar as failures occur even in companies with a track record of successful IS, such as American Airlines (Oz, 1994) and the Commonwealth Bank of Australia (Maiden & Rugg, 1996), and (3) pernicious, insofar as direct and consequential costs can be severely damaging. (Sauer, Southon, & Dampney, 1997, p. 349)

Since that time commentary on information systems project failure has continued to indicate that little progress is being made in this costly problem.

The very large survey conducted by the KPMG company in 2005 found that "IT project success has improved marginally since the last survey" (KPMG, 2005, p. 1).

Investigations into failure in specific jurisdictions continue to find significant failures like that of the Dutch IT industry:

> Even in business, health care and education, it occasionally goes wrong. Computable put 22 remarkable Dutch ICT failures in 2009 among them (Tut, 2009) (translated from the Dutch original).

Other studies showing the persistence of failure in the UK, Australia, China and the United States include those by Brouwer (2011), Pearson (2012), Shepherd, Patzelt, and Wolfe (2011), Taylor (2000), and Xiangnan, Hong, and Weijie (2010).

This persistence is remarkable as significant work has been done in addressing failure and it would be expected that this work would impact professional practice and hence performance. Stoica and Brouse capture this apparent anomaly as an issue in saying:

> In the field of Information Technology (IT) there is an observable trend toward project failure. Although multiple actions have attempted to address this failure trend, they have not impacted the extent of the trend (Stoica & Brouse, 2013, p728).

It seems clear from these studies over a period of time, and including very recent work, that information systems project failure continues to be a significant problem.

The Role of RE in Information Systems Project Failure

The next proposition to be examined is that RE is an important aspect of failure. There is no suggestion here that RE is the only or the critical factor in failure, but that, as an individual factor it is significant when compared with others.

The contention can be broken into two propositions:

- The greatest contribution to system failure comes from poor RE
- The cost of fixing RE problems is higher than other sources of error.

There is a general agreement that poor RE is an important and potentially damaging part of building a system.

The Standish reports are used to justify a claim that more than half of overruns and failures occur due to poor RE (Briggs & Gruenbacher, 2002; Eberlein & Leite, 2002; Lamsweerde, 2000). This finding is replicated in other studies in Europe (Lamsweerde, 2000) and in studies by Boehm (2000) and Moløkken-Østvold and Jørgensen (2003). Over a similar period various attempts have been made to measure the effect of poor requirements elicitation. Variously these studies have resulted in figures ranging from 12 per cent to 71 per cent of failure being attributed to poor requirements elicitation:

> …accurately capturing system requirements is the major factor in the failure of 90% of large software projects" (Davis et al., 2006), "Poor requirements management can be attributed to 71 percent of software projects that fail; greater than bad technology, missed deadlines, and change management issues. (Lindquist, 2005, p. 54)

There is also a general agreement amongst commentators that fixing the results of poor RE is more expensive than for other mistakes.

Significant research into information systems project failure has been conducted using the general framework of project management. In this domain the single most common work is the project management core body of knowledge (PMBoK) (IEEE Computer Society, 2008). This document points to the importance of RE as it happens at the start of the project and the cost of rectifying changes is most affected by this stage.

It seems clear from this general agreement that RE is critical to project success. In the next section evidence is examined as to the relevance of the interview of clients to good RE.

Method

A search for literature was conducted using the databases: ACM: Association for Computing Machinery Digital Library, Expanded Academic ASAP (Gale), IEEE Xplore, ProQuest, CiteSeerX, Science Citation Index Expanded, and ScienceDirect (Elsevier). The search terms [requirements, elicitation, communication, systems investigation, systems analysis, business analysis, specification, requirements definition, negotiation, system failure, and cognitive difficulty] were initially used and additional searches generated from citations within the found articles. Articles before 2000 were omitted unless cited by several more recent papers.

Papers (1680) were filtered by removing those with no mention of improvements or identification of problems in RE resulting in 420 final papers. Papers reporting empirical research and meta analyses were given more weight. Papers were then assigned a number of potential key concepts in terms of the problem solved by the treatment proposed or problems identified. These key concepts were then aggregated into the problem types. For each problem type a study was selected that the authors considered to be a useful approach to amelioration of the problem type.

A Classification of Problems in RE

Analysis of the literature generated a very long list of problems identified as leading to poor requirements elicitation. This list comes from both those who have summarised the literature of RE and those writers who use their own judgement as to some underlying causes of poor RE and proceed to investigate some solution to those causes. It is clear from the wide variety of sources and range of reported problems that RE is a complex and difficult task. It is unlikely that a simple "solution" exists to such a complex problem.

Meta-analysis of the literature of RE has been conducted by many authors including Hansen, Berente, and Lyytinen (2009), Dieste and Juristo (2011), Appan and Browne (2012), Zave (1997), Harris (2006), Davis, Dieste, Hickey, Juristo, and Moreno (2006), Hickey and Davis (2004), Reichental (2006), Marakas and Elam (1998), Pan (2005), Finkelstein (1994), Davidson (1996), Majchrzak, Beath, Lim, and Chin (2005), Zowghi and Coulin (2005), Pitts and Browne (2007) and in a less formal way by Jain, Vitharana, and Zahedi (2003).

Many other authors merely state a type of problem as an assumption underlying their study of a 'cure.'

The identified problems with RE can be summarised in nine categories:

- There are human aspects of RE that preclude simple communication between consultant and client
- The language of humans is not always suitable for technological solution
- Requirements change as the project proceeds
- Clients will sometimes ask for requirements that the organization does not need
- The client cannot say what the business needs
- Some clients do not want to help you with the project
- RE failed because it was not done properly
- Symptoms that are not problems are often reported
- RE is not deterministic

Each of these will be discussed in detail in the following sections.

There are Human Aspects of RE that Preclude Simple Communication between Consultant and Client

This problem type has several manifestations. There are promising results from studies that may address each manifestation. The problems arising from human limitations in communication can be summarised as follows:

- Humans have cognitive limitations that preclude complete communication (Browne & Ramesh, 2002; Jain et al., 2003; Majchrzak et al., 2005; Pitts & Browne, 2007).
- A suggestion for approaching this is the use of procedural prompts when framing questions. These encourage deeper thought before answers and have been tested several times with measurable success.
- People have different cultures (business or social) and backgrounds and so a common language does not always exist (Alvarez, 2002; Boehm, Grunbacher, & Briggs, 2001; Davidson, 1996; Harris, 2006; Jain et al., 2003; Saiedian & Dale, 2000; Urquhart, 1999; Zave, 1997; Zowghi & Coulin, 2005). For instance, technical people do not understand business concepts and business people do not understand IT concepts (Davidson, 1996; Boehm et al., 2001; Jain et al., 2003; Saiedian & Dale, 2000; Tsumaki & Tamai, 2006; Zowghi & Coulin, 2005).
- Some success has been obtained in bringing together the frames of reference of consultant and client through discourse analysis. This method tries to understand the frames of reference through understanding the language used by both parties and normally involves an “observer.”
- Some plain language statements can mean two different things, e.g., words can be synonyms or homonyms (Tsumaki & Tamai, 2006). The way people express problems can be misleading (Arthur & Groner, 2005).
- The amount of information presented can be too large to analyse (Tsumaki & Tamai, 2006).
- People who must be consulted disagree on what the requirements should be (Alvarez, 2002; Briggs & Gruenbacher, 2002; Davidson, 1996; Tsumaki & Tamai, 2006; Zave, 1997).

The Language of Humans Is Not Always Suitable for Technological Solution

To convert from English into a technically robust specification entails a range of problems. These include:

- Many terms used in the real world, e.g., ‘user friendliness’ and ‘reliability’, do not have exact meanings in a technical sense (Zave, 1997).
- Some statements of the problem cannot be used to create a solution because of their form or language (Zowghi & Coulin, 2005).
- Not everything that can be done has equal importance. In a technical solution everything included has the same priority (Zave, 1997).
- Some methodologies have gaps (Harris, 2006; Jain et al., 2003).
- The problem is interpreted as being bigger than the originally intended problem (Pitts & Browne, 2004; Tsumaki & Tamai, 2006).
- Real world problems are very complex. They are also wicked. A wicked problem is one where the definition of the problem is difficult (Briggs & Gruenbacher, 2002; Browne & Ramesh, 2002; Davidson, 1996; Nguyen, Armarego, and Swatman, 2002; Saiedian & Dale, 2000).

- Some consultants find the process difficult and rely upon their knowledge of previous solutions (Zowghi & Coulin, 2005).

Requirements Change as the Project Proceeds

A distinct type of reported problem arises independent of the accuracy of an initial specification. This can be seen as one of three types of change:

- Clients learn what is possible during the project (Briggs & Gruenbacher, 2002; Jain et al., 2003; Majchrzak et al., 2005; Robertson, 2001; Saiedian & Dale, 2000; Zowghi & Coulin, 2005).
- Business is essentially dynamic and so requirements change during the lifetime of the project (Boehm & Turner, 2004; Davis, Nurmuliani, Park & Zowghi, 2008; Larman, 2004; Pitts & Browne, 2007).
- People change their mind about what they want (Tsumaki & Tamai, 2006).

Clients will Sometimes Ask for Requirements that the Organization Does Not Need

Requirements can be specified directly from client conversations. This is not always sensible, for instance:

- The client asks for something that is not really needed (Davis et al., 2008; Tsumaki & Tamai, 2006).
- The client asks for something they are not committed to (Tsumaki & Tamai, 2006).

The Client Cannot Say What the Business Needs

Problems can arise when the assumption the clients understand their business needs turns out to be unfounded:

- Some requirements are tacit. That is, understood by the client, but not stated by them as it forms part of their tacit knowledge (Hudlicka, 1999; Robertson, 2001; Tsumaki & Tamai, 2006; Zowghi & Coulin, 2005).
- Some clients only know about a single section of the business that needs to be fixed (Arthur & Groner, 2005).

Some Clients Do Not Want To Help You with the Project

Requirements can be made difficult to elicit if the client representatives are not committed to the project. This can take several forms:

- A client representative has interests that conflict with others in the project or with the aims of the project (Alvarez, 2002; Briggs & Gruenbacher, 2002; Davidson, 1996; Giorgini, Massacci, Mylopoulos & Zannone, 2006; Zowghi & Coulin, 2005).
- Some people will use resistance tactics to avoid a conclusion to RE (Saiedian & Dale, 2000; Tsumaki & Tamai, 2006; Zowghi & Coulin, 2005).
- Clients see the new system as a part of power struggles in the organisation (Alvarez, 2002; Davidson, 1996; Zowghi & Coulin, 2005).

RE Failed Because It Was Not Done Properly (e.g., User Input Not Allowed)

Another distinct type of problem is lack of professional training or behaviour in the consultant or analyst:

- No user input was obtained (Hansen et al., 2009).
- Theory of RE was not used in practice (Hansen et al., 2009; Harris, 2006).

Symptoms That Are Not Problems Are Often Reported

Statements are often made using the words "sources of problem" when the description is of a symptom, e.g., "the three leading sources of project difficulty – i.e., lack of user input, incomplete requirements, and changing specifications" (Hansen et al., 2009, p.46).

The most common of these symptoms reported is the outcome of RE being an incomplete or incorrect set of requirements. This can take a few forms:

- Incomplete requirements (Arthur & Groner, 2005; Browne & Ramesh, 2002; Browne & Rogich, 2001;Harris, 2006; Lee & Rine, 2004; Marakas & Elam, 1998; Tsumaki & Tamai, 2006; Zave, 1997; Zowghi & Coulin, 2005).
- Changing specifications (Hansen et al., 2009).
- Finished system does not include all requirements asked for (Harris, 2006; Marakas & Elam, 1998).
- Incorrect requirements (Marakas & Elam, 1998).

RE Is Not Deterministic

In his PhD thesis, Harris (2006) found an "elliptical misalignment" between the theoretical and empirical worlds of RE. He maintains that this indicates that there are no causal connections between RE activities and eventual requirements. This type of thinking has been reflected in a number of post-modern approaches to studying RE. Examples include the case study of failure by Mitev (2000), the HSO process proposed by Andreou (2003), a framing model developed from social cognitive theory in a PhD by Davidson (1996) and the comparison of RE techniques by Coughlan and Macredie (2002) using socially oriented methodologies.

Conclusion

All of the papers studied that identified problems with RE were examined and nine types of problem were identified that incorporate all those sources. The categorization is particularly useful as each problem category is the result of work done to find methods of addressing the problem type. A software engineer finding a problem type occurring can use the category to identify how to approach the job. An educator can use the categorization to structure lessons. This is important as there is some evidence that requirements elicitation is not emphasised in CS and IS courses. In particular there is evidence that graduates are especially weak at requirements gathering.

References

Alvarez, R. (2002). Confessions of an information worker: A critical analysis of information requirements discourse. *Information and Organization, 12*(2), 85-107.

Anbari, F. T. (2003). Earned value project management methods and extensions. *Project Management Journal, 34*(4), 24-33.

Andreou, A. S. (2003). Promoting software quality through a human, social and organisational requirements elicitation process. *Requirements Engineering, 8,* 85-101.

Appan, R., & Browne, G. J. (2012). The impact of analyst-induced misinformation on the requirements elicitation process. *MIS Q. 36*(1), 85-106.

Arthur, J. D., & Groner, M. K. (2005). An operational model for structuring the requirements generation process. *Requirements Engineering, 10,* 45-62.

Basili, V. R., & Boehm, B. (2001). COTS-based systems top 10 list, *Computer, 34*(5), 91-95.

Boehm, B. (2000). The art of expectations management. *Computer, 33*(1), 122-124.

Boehm, B., Grunbacher, P., & Briggs, R. O. (2001). Developing groupware for requirements negotiation: Lessons learned. (organizational change). *IEEE Software, 46*(10).

Boehm, B., & Turner, R. (2004). *Balancing agility and discipline.* Boston, MA, USA: Addison-Wesley.

Briggs, R. O., & Gruenbacher, P. (2002). *EasyWinWin: Managing complexity in requirements negotiation with GSS*. 35th Hawaii International Conference on System Sciences, Hawaii, IEEE.

Brouwer, G E. (2011). *Report of an own motion investigation into ICT-enabled projects*. Victorian Government Ombudsman

Browne, G. J., & Ramesh, V. (2002). Improving information requirements determination: A cognitive perspective. *Information and Management, 39,* 625.

Browne, G. J., & Rogich, M. B. (2001). An empirical investigation of user requirements elicitation: Comparing the effectiveness of prompting techniques. *Journal of Management Information Systems, 17*(4), 223.

Coughlan, J., & Macredie, R. D. (2002). Effective communication in requirements elicitation: A comparison of methodologies. *Requirements Engineering Journal, 7,* 47-60.

Dalcher, D., & Drevin, L. (2003). Learning from information systems failures by using narrative and antenarrative methods. *Proceedings of SAICSIT 2003*.

Davidson, E. J. (1996). *Framing information systems requirements: An investigation of social cognitive processes in information systems delivery*. PhD, MIT.

Davis, A. M., Dieste, O., Hickey, A. M., Juristo, N., & Moreno, A. M. (2006). *Effectiveness of requirements elicitation techniques: Empirical results derived from a systematic review*. 14th IEEE International Requirements Engineering Conference (RE'06).

Davis, A. M., Nurmuliani, N., Park, S. & Zowghi, D. (2008). *Requirements change: What's the alternative?* Computer Software and Applications, 2008. COMPSAC '08, IEEE.

Dieste, O., & Juristo, N. (2011). Systematic review and aggregation of empirical studies on elicitation techniques. *Software Engineering, IEEE Transactions on, 37*(2), 283-304.

Eberlein, A., & Leite, J. (2002). *Agile requirements definition: A view from requirements engineering*. Paper presented at the International Workshop on Time-Constrained Requirements Engineering (TCRE'02), Essen, Germany.

Emam, K. E., & Koru, A. G. (2008). *A replicated survey of IT software project failures*. IEEE SOFTWARE.

Eveleens, J. L., & Verhoef, C. (2010). The rise and fall of the Chaos report figures. *IEEE Software, 27*(1), 30-36. Retrieved from http://www.cs.vu.nl/~x/the_rise_and_fall_of_the_chaos_report_figures.pdf

Finkelstein, A. (1994). *Requirements engineering: A review and research agenda*. Software Engineering Conference, 1994.

Giorgini, P., Massacci, F., Mylopoulos, J., & Zannone, N. (2006). *Detecting conflicts of interest.* 14th IEEE International Requirements Engineering Conference (RE'06).

Glass, R. L. (2002). Failure is looking more like success these days. *IEEE SOFTWARE, 19*(1), 103-104.

Glass, R. L. (2005). IT failure rates--70% or 10-15%? *IEEE SOFTWARE, 22*(3), 111-112.

Hansen, S., Berente, N., & Lyytinen, K. (2009). Requirements in the 21st Century: Current Practice and Emerging Trends. In K. Lyytinen (Ed.), *Design requirements engineering: A ten-year perspective* (44-87). Berlin, Heidelberg: Springer.

Harris, H. J. (2006). *Requirements dilemma.* PhD, Brunel Univeristy.

Hickey, A. M., & Davis, A. M. (2004). A unified model of requirements elicitation. *Journal of Management Information Systems, 20*(4), 65-84.

Hudlicka, E. (1999). Knowledge elicitation in complex military environments. In *Engineering of Computer-Based Systems, 1999. Proceedings. ECBS'99. IEEE Conference and Workshop on* (pp. 328-334). IEEE.

IEEE Computer Society. (2008). *IEEE guide adoption of the Project Management Institute* (PMI®) Standard. New York: IEEE.

Jain, H., Vitharana, P., & Zahedi, F. M. (2003). An assessment model for requirements identification in component-based software development. *ACM SIGMIS Database, 34*(4).

Kitapci, H., & Boehm, B.W. (2007). *Formalizing informal stakeholder decisions--A hybrid method approach*. HICSS 2007. 40th Annual Hawaii International Conference on System Sciences

KPMG. (2005). *Global IT project management survey*.

Larman, C. (2004). *Agile and iterative development*. Boston, MA, USA: Addison-Wesley.

Lamsweerde, A. V. (2000). *Requirements engineering in the year 00: A research perspective*. Paper presented at the 22nd International Conference on Software Engineering, Limerick, Ireland.

Lee, S. W., & Rine, D. C. (2004). *Missing requirements and relationship discovery through proxy viewpoints model.* 2004 ACM Symposium on Applied Computing.

Lindquist, C. (2005). Required: Fixing the requirements mess; The requirements process, literally, deciding what should be included in software, is destroying projects in ways that aren't evident until its too late. Some CIOs are stepping in to rewrite the rules. *CIO, 19*, 53-60.

Maiden, N. A. M., & Rugg, G. (1996). ACRE: Selecting methods for requirements acquisition. *Software Engineering Journal, 11*, 183-192.

Majchrzak, A., Beath, C. M., Lim, R. A., & Chin, W. W. (2005). Managing client dialogues during information systems design to facilitate client learning. *MIS Quarterly, 29*(4), 653.

Marakas, G., & Elam, J. (1998). Semantic structuring in analyst acquisition and representation of facts in requirements analysis. *Information Systems Research, 9*, 37-63.

Miller, R. A., & Luse, D. W. (2004). Advancing the IS curricula: The identification of important communication skills needed by IS staff during systems development. *Journal of Information Technology Education, 3*, 117-131. Retrieved from http://www.jite.org/documents/Vol3/v3p117-131-107.pdf

Mitev, N. (2000). Toward social constructivist understandings of IS success and failure: Introducing a new computerized reservation system. *Proceedings of the Twenty First International Conference on Information Systems*, Brisbane, Queensland, Australia ACM.

Moløkken-Østvold, K., & Jørgensen, M. (2003). *A review of software surveys on software effort estimation*. Paper presented at the 2003 International Symposium on Empirical Software Engineering, 2003. ISESE 2003.

Nee, N.Y. (2009). *Ensuring requirements gathering success in an agile environment*. ESI Horizons, July 2009. Retrieved from http://www.esi-intl.co.uk/horizons/publication/2009/200907_agile.asp

Nguyen, L., Armarego, J., & Swatman, P. (2002). Understanding requirements engineering: A challenge for practice and education. *School Working Papers Series 2002*. Faculty of Business and Law, Deakin University.

Pan, G. S. C. (2005). Information system project abandonment: A stakeholder analysis. *International Journal of Information Management, 25,* 174.

Pearson, D. D. R. (2012). *Learning technologies in government schools*. Victorian Government Auditor General.

Pitts, M. G., & Browne, G. J. (2004). Stopping behavior of systems analysts during information requirements elicitation. *Journal of Management Information Systems, 21*, 213.

Pitts, M. G., & Browne, G. J. (2007). Improving requirements elicitation: An empirical investigation of procedural prompts. *Information Systems Journal, 17*, 89-110.

Raisinghani, R., Riggen, R., & Ryan, K. (2014). *Adopting an agile methodology requirements-gathering and delivery*. PricewaterhouseCoopers. Retrieved from https://www.pwc.com/en_US/us/insurance/publications/assets/pwc-adopting-agile-methodology.pdf

Reichental, J. (2006). *An evaluation of the effectiveness of interview techniques in the elicitation of tacit knowledge for requirements engineering in small software projects*. Nova Southeastern University.

Robertson, S. (2001). Requirements trawling: Techniques for discovering requirements. *International Journal of Human-Computer Studies, 55*, 405-421.

Saiedian, H., & Dale, R. (2000). Requirements engineering: Making the connection between the software developer and customer. *Information and Software Technology, 42*(6), 419-428.

Sauer, C., Southon, G., & Dampney, C. N. G. (1997). *Fit, failure, and the house of horrors: Toward a configurational theory of IS project failure*. Paper presented at the Eighteenth International Conference on Information Systems, Atlanta Georgia.

Sauer, C., & Cuthbertson, C. (2002). UK project management is healthier than supposed, CW360.com survey suggests. *Computerworld*.

Savolainen, P., Ahonen, J. J., & Richardson, I. (2012). Software development project success and failure from the supplier's perspective: A systematic literature review. *International Journal of Project Management, 30*(4), 458-469. doi: http://dx.doi.org/10.1016/j.ijproman.2011.07.002

Shepherd, D. A., Patzelt, H., & Wolfe, M. (2011). Moving forward from project failure: Negative emotions, affective commitment, and learning from the experience. *Academy of Management Journal, 54*(6), 1229-1259. doi: 10.5465/amj.2010.0102

Stoica, R., & Brouse, P. (2013). IT project failure: A proposed four-phased adaptive multi-method approach. *Procedia Computer Science, 16*, 728-736. doi: http://dx.doi.org/10.1016/j.procs.2013.01.076

Taylor, A. (2000). Professional practice: IT projects: Sink or swim? *The Computer Bulletin*. Retrieved from http://archive.bcs.org/bulletin/jan00/article1.htm

Tsumaki, T., & T. Tamai (2006). Framework for matching requirements elicitation techniques to project characteristics. *Software Process Improvement and Practice, 11*, 505-519.

Tut, D. (2009). *2009: 22 high-profile IT failures*. Retrieved 20 March, 2013, from http://www.computable.nl/artikel/nieuws/ictbranche/3203753/2379258/2009-22-spraakmakende-ictfiascos.html

Urquhart, C. (1999). Themes in early requirements gathering: The case of the analyst, the client and the student assistance scheme. *Information Technology and People, 12*(1), 44.

Xiangnan, L., Hong, L., & Weijie, Y. (2010). *Analysis failure factors for small and medium software projects based on PLS method*. Paper presented at the Information Management and Engineering (ICIME), 2010 The 2nd IEEE International Conference on 16-18 April 2010.

Zave, P. (1997). Classficiation of research efforts in requirements engineering. *ACM Computing Surveys, 4*(29), 315-321.

Zowghi, D., & Coulin, C. (2005). Requirements elicitation: A survey of techniques, approaches, and tools. In A. Aurum & C. Wohlin (Eds.), *Engineering and Managing Software Requirements* (pp. 19-46). Berlin: Springer.

Biographies

Dr. **Bill Davey** is a Senior Lecturer in the School of Business Information Technology at RMIT University, Melbourne, Australia. His research interests include methodologies for systems analysis and systems development, information systems curriculum, eGovernment and the elderly, and information technology in educational management. Bill and Kevin have worked together co-operatively on many occasions. They have co-operated on several joint research projects and coauthored several papers relating to management information systems, programming, computers in management, eGovernment and the elderly, and IS curriculum.

Dr. **Kevin R. Parker** is Chair and Professor of the Department of Informatics and Computer Science at Idaho State University. Parker's research interests include eGovernment and the elderly, business intelligence, and the impact of developments in informatics on curriculum. He has published in such journals as *Education and Information Technologies*, *Informing Science*, *Journal of Information Systems Education*, and *Communications of the AIS*. Dr. Parker's teaching interests include web development technologies, programming languages, data structures, and database management systems. Dr. Parker holds a B.A. in Computer Science from the University of Texas at Austin (1982), an M.S. in Computer Science from Texas Tech University (1991), and a Ph.D. in Management Information Systems from Texas Tech University (1995). He chairs a hybrid academic department that spans the College of Business and the College of Science and Engineering, and offers degrees in business informatics, health informatics, and computer science.