Association for Information Systems
AIS Electronic Library (AISeL)

AMCIS 2008 Proceedings

Americas Conference on Information Systems (AMCIS)

2008

Requirement Prioritization Decision Factors for Agile Development Environments

Greg Hoff
University of Nebraska at Omaha, ghoff@unomaha.edu

Ann Fruhling
University of Nebraska at Omaha, afruhling@unomaha.edu

Kerry Ward
University of Nebraska at Omaha, kward@unomaha.edu

Requirement Prioritization Decision Factors for Agile Development Environments

Greg Hoff
University of Nebraska at Omaha
ghoff@unomaha.edu

Ann Fruhling
University of Nebraska at Omaha
afruhling@unomaha.edu

Kerry Ward
University of Nebraska at Omaha
kwward@unomaha.edu

ABSTRACT

In an agile development environment, project planners continuously prioritize work tasks so requirements that provide the most value are delivered first. This strategy is based on Value Based Software Engineering principles that different requirements deliver different levels of value and diverse stakeholders view the importance of the value of various requirements differently and thus, will prioritize them differently. However, we found that there are several core values that stakeholders have more agreement in terms of relevancy and importance than others. By knowing these core values, project planners have increased insights as to which requirements should be prioritized higher, hence, hopefully increasing overall stakeholder satisfaction and reducing project risk.

Keywords

Value Based Software Engineering, Agile Development, Requirements Engineering.

INTRODUCTION

The prioritization of system requirements provides a roadmap for agile software development projects. In an agile development environment, stakeholders are required to prioritize requirements so the most important requirements are worked first. Often there are competing goals among stakeholders. Executives and shareholders want systems that make the organization more effective and efficient, thus leading to increased profitability. System users want systems that make their jobs easier to accomplish. System developers are concerned with responding to production problems. And, system architects may favor implementing new and novel technology.

Adding to the complexity of the problem is that requirements and priorities are constantly changing. These changes may be due to: 1) changing users 2) changing business problem 3) changing technology and 4) changing markets. The challenge then is how does the project manager and business owner rank requirements and account for all the competing factors? What criteria govern the selection of which requirements to be included in the next software build?

There are several effective approaches to dealing with these competing goals such as expectation management, visualization and trade-off analysis techniques, prioritization, groupware, and business case analysis [3]. In this research we further examine the prioritization approach.

In our study we solicited the perceived relevance and importance requirement criteria from information technology professionals and other stakeholders from medium to large corporations. Gaining a better understanding of the stakeholders' motives for requirements selection and prioritization has several implications. First, understanding stakeholders' motives should allow practitioners to address these motives directly, increasing stakeholder satisfaction. Second, this understanding should lead to more effective management decisions as to whether requirements should be included and their priority in a

software build. Third, this understanding of motives and expected impact on requirements determination should reduce the risk of project failure.

The remainder of this paper is structured as follows. Section 2 discusses system requirements importance, value-based software engineering principles, agile development approaches, and system risk management. Section 3 describes the various decision factors for the inclusion and prioritization of system requirements. The research method is discussed in Section 4. Section 5 presents our findings. Lastly, Section 6 discusses the implications of our research, limitations and future direction.

LITERATURE REVIEW

Importance of Requirements

The main purpose of identifying and defining requirements is to describe the functions for an information system that needs to be developed and to guide the system design architecture. Requirements are important to systems analysts for several reasons. First, requirements identify what a system should and should not do. Second, requirements define what can be changed, what cannot be changed and what should be changed. Third, when system analysts do not have control of all the systems under development, requirements can be a starting point to help coordinate among many parties. Finally, it is important to have clear, well-defined requirements that can guide system design so that the problems being addressed are solved as efficiently as possible.

Requirements are also important in managing the risk associated with project failure [16]. Several aspects of requirements such as poorly defined or inadequate requirements increases the likelihood that they system will fail to meet the needs of the users and the organization. Requirements can also indirectly impact risk management by triggering other risk factors such as cost overruns. Improving the requirements analysis process and understanding the requirement selection criteria (referred to as decision factors) can further improve the success of system development projects.

Value Based Software Engineering

Value Based Software Engineering (VBSE) theory states stakeholders beyond users are critical to the success of the software development project and must be considered. VBSE asserts that traditional software engineering practice and research is value-neutral where each deliverable is given equal importance across the board i.e. no prioritization is done and no cut-offs are set when unit costs exceed derived unit benefit [2]. Additionally, VBSE makes the assumption that traditionally software engineers confine themselves to software development and do not consider business impacts or concerns. The goal of VBSE is thus to change the frame-of-mind of software engineers in which decisions are made within a software development project so that factors that more truly determine success and cost/benefit analysis are taken into account. In other words, VBSE represents a paradigm shift from value-neutral to value-based thinking.

At the core of Value Based Software Engineering is Theory W. Theory W is defined by Boehm [2] as determining what is important to each of the success-critical stakeholders (SCS) and defining how success is assured for all SCSs. The desired end state for requirements then is a negotiated win-win state in which the system stakeholders agree to an option from which all can derive benefit. The win-win scenario usually takes greater effort to achieve and requires a change in mindset away from constant competition. In the case of requirement negotiation a win-win outcome is optimal. Because of the focus on multiple stakeholder "wins", VBSE has been viewed as a valid risk reduction strategy for managing systems development risk. In particular, theory W has been cited as a way to mitigate project risk associated with lack of management and user project commitment [9].

Agile Development Approach

The recent introduction of the agile software development approach was motivated partially by the notion that traditional plan-driven software teams are hampered by an ever-increasing amount of processes and documentation. The aim of the agile approach is to streamline the software development process. Whereas traditional plan-driven development is perceived to be rigid and cumbersome, agile approaches acknowledge volatile and emerging requirements management. Agile methods allow users to alter their requirements once they see and test the system. This often occurs in iterations. Each new iteration includes reprioritizing requirements as one of the first steps. In an iterative environment not only is it important to prioritize the requirements quickly, it is also important to consider the needs of the various stakeholders. Following an iterative and reprioritization process may assist in minimizing some of the risks of project failure because of the increased user involvement.

Risk Management

Managing risks associated with software development project failure has become a substantial concern for organizations [16]. Risk management is the active process of controlling risk and includes the identification of potential threats (risk assessment) and taking actions to eliminate or minimize the threats (risk mitigation). Given the well publicized catastrophic failure of a number of high profile software development projects - for example the FAA's $2.6 billion air traffic control system failure - it is easy to understand the importance of managing the risk of software development projects (see also [4] for additional examples of high profile software project failures).

One specific source of software project risk is requirements determination and prioritization. In general requirements risk refers to the uncertainty related to requirements [16]. Risks associated with requirements development can be of three types: 1) requirements identity – the ability to correctly identify requirements, 2) requirements volatility – how likely the requirements are to change, and 3) requirements complexity – how clearly the requirements can be established and understood [11]. What makes the risks associated with requirements determination of particular importance is that while a lot of software project risks lie outside the direct control of the project manager, risks associated with requirements are risks the project manager can control [9]. It has been suggested that project managers can mitigate the risk associated with requirements by managing the change and ambiguity, for example by using evolutionary (agile) approaches to systems development and making sure the users are driving the requirements [9]

CONCEPTUAL FOUNDATION – REQUIREMENT DECISION FACTORS

Requirements can be categorized according to their intended purpose or the effect they will have on the proposed system. In the context of this paper, the intended purpose or effect is referred to as a decision factor. Examples of requirement decision factors are: cost-benefit to the organization, complexity, impact of maintenance, increased performance, etc. There are different opinions on which decision factors are the most important when determining the highest priority requirements. A summary of the various requirement decision factors is shown in Table 1. A discussion of the literature associated with each decision factor follows.

Wohlin and Aurum [17, 18] find that the stakeholder priority of requirement, delivery dates, requirement's issuer and development cost-benefit are the most important decision factors when prioritizing requirements. Stakeholder priority takes into account how customers and external market forces view the importance of a requirement. Delivery dates are common schedule pressures associated when a requirement has been promised. Requirement issuer focuses on who requested the functionality. Development cost-benefit compares the resources expended to implement functionality versus the gain the business will receive for having implemented the functionality.

Sommerville and Sawyer [14] prioritize requirements based on the "importance to stakeholders" and contribution to the "overall success of the system". These prioritization attributes are reflected as "Stakeholder priority of requirement" and "Cost-Benefit to the Organization".

Davis [5] advocates evaluating each requirement's priority according to the available resources to implement the requirement which then can be used to calculate the requirement's probability of success. Requirements with greater success probabilities are delivered first.

Lutowski [10] prioritizes requirements based on "importance or immediacy of need", referred to as "Impact to the Organization". Functionality that is needed the soonest and the most in the organization is identified.

Royce [12] discusses assessing requirements based on consistency of requirements to organization vision. Requirements are derived from business cases. The compliance to organizational vision from the business case is interpreted as "Stakeholder priority of requirement". In this case the stakeholders are more likely to be executives who are responsible for developing the vision instead of system users. The second contribution Royce [12] makes falls under the decision factor of "requirement volatility" i.e. taking into account the impact of changing requirements to the system.

A requirement prioritization strategy could be based on cost reduction. Requirements that minimize organizational costs are given higher priority. Factors that contribute to software costs are related to the technical environment, participant competencies, quality goals, timeframe, documentation required, amount of available reusable software, complexity, and requirement stability [1].

Almost all requirements for a system are negotiable and even more so when requirements are changing. Favaro [6] advocates looking for the right mix of requirements that will maximize value for the organization through cost-benefit analysis. Further,

given the importance of managing the risk to the project, it is intuitive that factoring the risk related to project failure should be considered in the final prioritization of requirements.

	Competitors / market forces	Requirement's issuer	Stakeholder priority of requirement	Requirements Volatility	Support for Education / Training	Cost-benefit to the organization	Resources / competencies	Delivery date / schedule	System Impact	Complexity	Requirement dependencies	Evolution	Impact of maintenance	Fixes error	Impact to the organization	Probability of success	Testability
Boehm [1]				●		●	●	●	●	●	●		●				
Davis [5]																●	
Favaro [6]						●											
Jung [9]						●											
Karlsson [8]						●											
Lutowski [10]															●		
Royce [12]			●	●		●		●	●		●						
Ruhe & Saliu [13]			●			●	●				●						
Sommerville [14]			●			●											
Thayer [15]			●	●													●
Wohlin & Aurum [17]	●	●	●	●	●	●	●	●	●	●	●	●	●				
Young [19]		●		●		●											
Pilot Study														●			

Table 1. Summary of requirement decision factors

METHODOLOGY

The survey research methodology is employed in this study. It is one of the most commonly used research methodologies in empirical MIS research [7]. Survey research has three important characteristics for theory development. First, the information collected is from asking people for information in some structured format. Second, it has a quantitative orientation and lastly, it samples the population with the intention of being able to generalize findings [7]. Survey research can be classified into two categories: exploratory research and explanatory research. Exploratory research or descriptive research is used to understand more about the topic; whereas, on the other hand, explanatory research analyzes the causal relationships among variables. This research focuses on the exploratory emphasis of survey research.

The Requirements Decision Factors survey is an extension of the work done by Wohlin and Aurum [17, 18].In our study, we expanded the survey to include a more comprehensive list of requirement decision factors based on the literature. The respondent pool was broadened by including a more diverse set of companies. Our literature review confirmed the use of most of Wohlin and Aurum reported attributes, although, we also identified four additional decision factors which are:

1. Impact to the organization - Is this requirement something that will cause significant change or disruption? The greater the change or potential disruption to the organization, the greater risk the potential project failure is to the organization. Furthermore, we may not want to include a high impact requirement until additional process changes, training, support structures, etc. are in place.
2. Probability of success - Are there more factors contributing to a satisfactory system completion state versus unsatisfactory? Requirements that have a low probability of success or have elements working against succeeding may not want to be attempted.
3. Fixes error - In a previous delivery, a requirement was incorrectly implemented and is now being rectified. Therefore, a new requirement is added to the queue to fix an unforeseen problem and needs to be included in the prioritization.
4. Testability – A test exists for the requirement so it can be unambiguously passed or failed. In other words, how easy or difficult is it to validate the requirement. Requirements requiring subjective evaluation are more difficult to reach stakeholder consensus in determining completion than those that have objective evaluation.

The decision factors are grouped into three categories. Each category represents a different perspective for approaching requirement selection and prioritization. The external market / customer category contains factors from outside the organization. The company management categories are business considerations internal to the organization. And, the development / maintenance category are internal technical considerations. In Table 2 we list the decision factors associate with each category.

Category	Decision Factors	Comments
External market / customer	Competitors	The status of the competitors with respect to the requirement. In other words, it is taken into account whether a competitor has the implied functionality implemented or not.
	Requirement's issuer	The actual issuer of the requirement is taken into account, i.e. which stakeholder (internal or external) generated the requirement.
	Stakeholder priority of requirement	The importance of the requirement from stakeholders' perspectives. For example, this could include usability for system users, or strategic initiatives from business owners, or regulatory requirements from society.
	Requirements volatility	This criterion is related to whether the requirement is likely to change or not. Requirements volatility is a known requirements determination risk [11].
Company management	Support for Education/Training	The ability and possibility to provide technical support, education, and/or training to customers, markets and so forth with respect to the requirement.
	Development cost-benefit	The benefit derived by implementing the requirement is greater than the cost of development and implementation, i.e. the value-added to the organization.
	Resources/competencies	The availability of resources with the right competencies to implement the requirement.
	Delivery date / Calendar time	The ability to meet the project deadline.
	* Impact to the organization	Will this requirement cause significant change or disruption? We may not want to include a high impact requirement until additional process changes, training, support structures, etc. are in place.
	* Probability of success	Are there more factors contributing to satisfactory completion state versus unsatisfactory. Requirements that involve high risk or have elements working against succeeding may not want to be attempted
Development / maintenance personnel	System impact	The impact of the requirement on the existing system.

Category	Decision Factors	Comments
	Complexity	The estimated complexity of the requirement and the associated challenges in implementing it. Requirements complexity is a known requirements determination risk [11]
	Requirements dependencies	The dependencies between this specific requirement and other requirements, either already implemented or other posed requirements.
	Evolution	The impact on the future evolution of the system. Evolution is associated with requirements volatility, a requirements determination risk[11].
	Maintenance	The impact, positive or negative, on the maintenance of the current system.
	* Fixes error	In a previous delivery, a requirement was incorrectly implemented and is now being rectified. We may want to include a fix with an importance higher than the requirement's previous importance due to the stigma of delivering 'buggy' software.
	* Testability	A test exists for the requirement so it can be unambiguously passed or failed. Requirements requiring subjective evaluation are more difficult to reach stakeholder consensus in determining completion.

Table 2. Decision Factor Categorization [17, 18]

* New Requirement Decision Factor

SURVEY RESULTS AND ANALYSIS

Analysis of the Stakeholders

The field survey occurred at a large information technology conference for IT professionals in the Midwest. Twenty-nine surveys were collected. Seven were discarded due to incompleteness. The demographic information collected indicated a range of experience and roles listed in Table 3 and 4, respectively. Over half of the participants had more than ten years of information technology experience. The roles of the participant in the organization were fairly distributed. Also, the diversity of the industries is shown in Table 5.

	Valid responses	
Under one year	2	9%
One to five years	4	18%
Five to ten years	4	18%
Over ten years	12	55%

Table 3. Years of experience with software development and/or acquisition

	Valid responses	
Executive	2	9%
Manager / team lead	5	23%
Business analyst	0	0%
Programmer	8	36%
Engineer/technical	5	23%
Testing	1	4.5%
Non-professional	1	4.5%
No response	0	0%

Table 4. Participant's role in their organization

	Valid responses	
Services	2	9%
Products/manufacturing	7	32%
Financial	6	27%
Government	2	9%
Military	3	14%
Healthcare	1	4.5%
Other	0	0%
No response	1	4.5%

Table 5. Industry of participant's organization

Analysis of the Data

We analyzed the requirement decision factors from two perspectives: relevancy to the stakeholder and the importance in comparison to other decision factors. We begin our discussion examining the stakeholder consensus on requirement decision factor relevancy.

Decision Factor Relevancy

The response on whether each decision factor was relevant to the decision maker is shown in Table 6. The two decision factors that ranked the highest in relevancy were Fixes Errors and Cost-Benefit to the Organization. In fact, 91% of the respondents rated these as high relevancy. Therefore, when system managers are prioritizing requirements these two decision factors should be considered first when trying to build consensus and deciding which requirements to include in the next iteration. Because of the high rating one could expect that most stakeholders would be in agreement (win-win) with requirements associated with these intended values.

The decision factor Fixes Errors was suggested during the pilot test as a primary motivation for prioritizing a requirement high. Strictly speaking using plan-based development, if a requirement has not been previously implemented properly, the fix would not create a new requirement; however, when following an agile approach and in an iterative environment Fixes Errors becomes more relevant. Also, since we surveyed businesses where they are measured on financials it makes sense that Cost-Benefit was one of the most agreed upon decision factors.

The next set of factors that received fairly high consensus was Requirement Dependencies, Delivery Date/Schedule and Complexity. Requirement Dependencies and Delivery Date/Schedule seem logical because of their importance to project planning. It is interesting that Complexity has so much agreement on relevancy. We discuss Complexity more below.

The two decision factors that had the lowest relevancy were Evolution and Support for Education/Training. Given the industries and the demographics of the respondents it is not a complete surprise that Support for Education/Training was low. Education and training is not usually in the forefront of the individuals surveyed. With regards to Evolution, there were a few comments that this term was vague and thus, not understood. Overall, 9 out of 17 decision factors had greater than 81% consensus that they are relevant.

The Development/Maintenance and Company Management Categories decision factors had more consensus than External Market Factors.

From a risk management perspective, the high agreement on the relevancy of Cost-Benefit to the Organization indicates that stakeholders are concerned about risk that limits loss to the organization from project failure in their decisions. Also, Complexity another decision factor related to risk had high consensus on its relevance. However, Requirements Volatility and Evolution which may also influence project risk only had 73% and 55% agreement, respectively. These mixed results may indicate that stakeholders have a more diverse view on requirement decision factors associated with risk and that those surveyed do not directly associate requirements determination factors with the risks posed to the organization. This finding suggests further research is warranted.

Decision Factor	**Yes**	**No**	**Percent Yes**	**Category**
Fixes error	20	2	91%	Development / maintenance
Cost-benefit to the organization	20	2	91%	Company management
Complexity	19	3	86%	Development / maintenance
Requirement dependencies	19	3	86%	Development / maintenance
Delivery date / schedule	19	3	86%	Company management
Impact of maintenance	18	4	82%	Development / maintenance
Impact to the organization	18	4	82%	Company management
Resources / competencies	18	4	82%	Company management
System Impact	18	4	82%	Development / maintenance
Probability of success	17	5	77%	Company management
Stakeholder priority of requirement	16	6	73%	External market / customer
Requirements Volatility	16	6	73%	External market / customer
Testability	15	7	68%	Development / maintenance
Requirement's issuer	15	7	68%	External market / customer
Competitors / market forces	14	8	64%	External market / customer
Evolution	12	10	55%	Development / maintenance
Support for Education / Training	11	11	50%	Company management

Table 6 – Decision factor relevancy

Decision Factor Importance

Next, we asked the participants to rank the decision factors in importance compared to one another. Participants were asked to consider the importance from two perspectives, how important the decision factor is today in their organization (see Table 7) and how important they perceive the decision factor should be in a "perfect world" environment (See Table 8). In other words, we wanted to compare "reality" to "ideal". As shown below, the weighted importance of criteria Impact to the Organization, Fixes Errors, Delivery Date/Schedule were relatively high and Requirement's issuer, Volatility and Support for Education/Training were ranked low.

Requirement attribute	**Importance Percentage ***	**Category**
Impact to the organization	10.3%	Company management
Fixes error	8.7%	Development / maintenance
Delivery date / schedule	8.1%	Company management
Cost-benefit to the organization	7.4%	Company management
System Impact	7.0%	Development / maintenance
Stakeholder priority of requirement	6.4%	External market / customer
Resources / competencies	5.9%	Company management
Probability of success	5.8%	Company management
Requirement dependencies	5.6%	Development / maintenance
Competitors / market forces	5.2%	External market / customer
Complexity	5.1%	Development / maintenance
Impact of maintenance	5.1%	Development / maintenance
Testability	4.7%	Development / maintenance
Evolution	4.1%	Development / maintenance
Requirement's issuer	3.9%	External market / customer
Requirements Volatility	3.6%	External market / customer
Support for Education / Training	3.2%	Company management

Table 7 - Importance of decision factors – Reality

* Percentage of points assigned by survey respondents asked to consider the importance of the attribute when prioritizing requirements in a software release or project.

Of the four new criteria introduced into this study, Fixes Errors and Impact to the Organization were shown to be of relative high importance with Probability of Success and Testability not rating as high in reality.

A statistical analysis using t-tests for each variable was done to determine if there was a significant difference. The results did not show any statistical significant difference between the reality decision factors and the ideal decision factors at a 95% confidence level. We included the Bonferroni correction as part of our statistical analysis. So, one may conclude that the stakeholders surveyed are consistent and not compromising their decision criteria.

Requirement Attribute	* Importance Percentage	Category
Impact to the organization	9.5%	Company management
Cost-benefit to the organization	9.1%	Company management
Probability of success	8.2%	Company management
Fixes error	8.1%	Development / maintenance
Delivery date / schedule	6.7%	Company management
System Impact	6.7%	Development / maintenance
Stakeholder priority of requirement	5.6%	External market / customer
Competitors / market forces	5.5%	External market / customer
Resources / competencies	5.4%	Company management
Complexity	5.4%	Development / maintenance
Requirement dependencies	5.4%	Development / maintenance
Impact of maintenance	5.2%	Development / maintenance
Testability	4.7%	Development / maintenance
Evolution	4.0%	Development / maintenance
Requirements Volatility	3.7%	External market / customer
Support for Education / Training	3.5%	Company management
Requirement's issuer	3.2%	External market / customer

Table 8 - Importance of decision factors - Ideal

* Percentage of points assigned by survey respondents asked to consider the importance of the attribute when prioritizing requirements in a software release or project.

CONCLUSION

We found that there are several core decision factors that stakeholders have more agreement in terms of relevancy and importance than others. By knowing these core decision factors project planners have additional insights as to which requirements should be prioritized higher according to their value, hence, this may increase the overall stakeholder satisfaction and reduce project risk. The insights into stakeholders' motives for requirements selection and prioritization may also increase win-win solutions that could directly impact stakeholder satisfaction. Better requirement management and increase stakeholder commitment all decrease the likelihood of project failure.

The results from this study found that the decision factors Fixes Errors, Cost-Benefit to the Organization, Complexity, Requirement Dependencies, and Delivery Date/Schedule have the most relevancy consensus among stakeholders when evaluating the value of a system requirement. The data suggests that Impact to the Organization, Fixes Errors, Cost-benefit to the Organization are the most important decision factors when prioritizing requirements for implementation. It appears that Evolution was the least intuitive and perhaps may not really be a decision factor. Likewise, Support for Training and Education maybe should also be eliminated.

Further research may help determine whether additional demographics such as company size, development staff attributes and context have an effect on decision factors' importance. Secondly, future research needs to understand more how risk management affects requirement prioritization in an agile development project. The main limitations of this research are the small sample size and the inability to generalize the results.

REFERENCES

[1] Boehm, B.W., "The High Cost of Software" *Practical Strategies for Developing Large Software Systems* ed. E. Horowitz, Addison-Wesley, Reading MA, 1975.

[2] Boehm, B.W., and Ross, R. "Theory-W Software Project Management: Principles and Examples" *IEEE Transactions on Software Engineering*, Vol. 15, No. 7, July 1989, pp. 902 - 916.

[3] Boehm, B.W., "Value-based software engineering"

ACM SIGSOFT Software Engineering Notes, Vol. 28 No. 2, March 2003, pp 1 - 12.

[4] Charette, R.N., "Why Software Fails" IEEE Spectrum, Vol. 42 No. 9 pp 42- 49

[5] Davis, A., "The Art of Requirements Triage", *IEEE Computer*, Vol. 36, No. 3, March 2003, pp. 42-49.

[6] Favaro, J. "Managing requirements for business value", *IEEE Software,* Vol. 19, No. 2, Mar/Apr 2002, pp. 15-17.

[7] Grover, V., "A tutorial on survey research: from constructs to theory," http://dmsweb.badm.sc.edu/grover/survey/MIS-SUVY.html, 1998.

[8] Jung, Ho-Won "Optimizing value and cost in requirements analysis" *IEEE Software* Vol. 15, No, 4, July-Aug. 1998, pp 74 - 78.

[9] Keil, M., Cule, P.E., Lyytinen, K. and Schmidt, R.C. A Framework for Identifying Software Project Risks, Communications of the ACM, Vol. 41, No. 11, November 1998, pp 76-83.

[10] Lutowski, R., Software Requirements, Auerbach Publications, Boca Raton, 2005.

[11] Mathiassen, L., Tuunanen, T., Saarinen, T, and Rossi, M. "A Contigency Model for Requirements Development" *Journal of the Association for Information Systems*, Vol. 8, No. 2 pp 569-597.

[12] Royce, W. *Software Project Management. A Unified Framework*, Addison-Wesley, Reading, MA, 1998.

[13] Ruhe, G., and M.O. Saliu, "The Art and Science of Software Release Planning" *IEEE Software*, Vol. 22, No. 6, Nov.-Dec. 2005, pp 47 - 53.

[14] Sommerville, I. and P. Sawyer, *Requirements Engineering: A Good Practice Guide*, John Wiley & Sons, Chichester, England, 1997.

[15] Thayer, R. and M. Dorfman *Software Requirements Engineering,* 2nd ed., IEEE Computer Society, Los Alamitos, CA, 2000.

[16] Wallace, L., Keil, M. and Rai, A. How Project Software Risk Affects Project Performance: An Investigation into the Dimension of Risk and an Exploratory Model, Decision Sciences, Vol. 35, No. 2, Spring 2004, pp 289-321.

[17] Wohlin, C. and A. Aurum, "Criteria for Selecting Software Requirements to Create Product Value: An Industrial Empirical Study", *Value-Based Software Engineering*, eds. S. Biffl, A. Aurum, B. Boehm, H. Erdogmus, P. Grünbacher, Springer, Berlin, 2006.

[18] Wohlin, C.; Aurum, A., "What is important when deciding to include a software requirement in a project or release?" *International Symposium on Empirical Software Engineering*, Nov. 2005, pp 246-255.

[19] Young, R.R., *The Requirements Engineering Handbook*, Artech House, Inc., Boston, 2004.