# Using the Analytical Hierarchy Process as a Ranking Tool for User Story Prioritization Techniques

Sultan Alshehri and Luigi Benedicenti
Software Systems Engineering
University of Regina, Regina
Regina, SK, Canada
Email: aljumais@uregina.ca, luigi.benedicenti@uregina.ca

*Abstract*— **The Analytic Hierarchy Process (AHP) has been applied in many fields and especially to complex engineering problems and applications. AHP is capable of structuring decision problems and finding mathematically determined judgments built on knowledge and experience. This suggests that AHP should prove useful in agile software development, where complex decisions occur routinely. This paper provides a ranking approach to help stakeholders select the best prioritization technique for prioritizing the user stories. A case study demonstrated the effectiveness of this approach.**

***Keywords-Extreme Programming; User Stories; Analytic Hierarchy Process.***

## I. INTRODUCTION

The quality of Extreme Programming (XP) development results from taking 12 core practices to their logical extremes [1]. One such practice is the planning game, in which customers and developers cooperate to develop requirements that produce the highest value for customers as rapidly as possible. This is accomplished as follows. Customers write system requirements as user stories. User stories are defined as “short descriptions of functionality told from the perspective of a user that are valuable to either a user of the software or the customer of the software” [2]. Developers review the stories to ensure domain-specific information is sufficient for their implementation. Developers evaluate user stories using story points to identify the complexity and cost of their implementation. Then, user stories are broken down into small tasks. Finally, customers and developers collaborate in prioritizing user stories based on their value and other relevant factors.

To reconcile conflicting opinions among them, customers and developers often adopt a prioritization technique [3,4,5]; but, this adoption process is usually not formalized. In this paper, the Analytical Hierarchy Process (AHP) is utilized as a well-structured multi-criteria decision making tool to help XP software development teams rank six prioritization techniques: 100-Dollar Test (Cumulative Voting), MoSCow, Top-Ten Requirements, Kano Model, Theme Screening, Relative Weighting.

This paper is organized as follows: Sections 2 to 6 describe the AHP method; the six prioritization techniques are presented in Section 7; four criteria for ranking the prioritization techniques are proposed in Section 8; a case study, its results and its findings are presented in Section 9 and 10, and Section 11 concludes the paper.

## II. RELATED WORK

There is no consensus in the literature on the most important factors determining the priority of system requirements. However, almost all the factors taken into consideration aim to maximize the value delivered to the customer. Bakalova et al. proposed to use project context, effort estimation, dependencies, input from the developers, learning experiences and external change [6]. Hoff et al. relied on four factors: cost-benefit to the organization, impact of maintenance, complexity and performance effects [7]. They also considered fixed errors, requirement dependencies, complexity, and delivery data/schedule as ancillary factors. Somerville and Sawyer prioritize requirements based on the viewpoint approach that represents information about the system requirements from different perspectives representing different types of stakeholder [8]. Davis used Triage as an evaluation process considering time, available resources, and requirements interdependencies [9]. Lutowski prioritized the requirements based on the importance or immediacy of need [10]. Bhoem considered the cost of implementing the requirement as the most important factor for prioritization [11]. In Bhoem’s work, cost is related to the technical environment, complexity, quality, timeframe, documentation, availability reusable software, participant competencies, and stability of requirements. Berander and Andrews surveyed the literature and found common aspects in prioritizing requirements such as penalty, cost, time, risk and volatility [12]. The authors added that other aspects like financial benefits, competitors, release theme, strategic benefit, competence/resource, and ability to sell should also be considered.

In the agile methodology domain, Patel and Ramachandran prioritized user stories based on business functionality, customer priority, core value, market values, implementation cost, and business risk [13]. Many well-established prioritization technique available are applicable

ISBN: 978-1-61208-304-9

to requirements prioritization: Ping Pong Ball, Pair-Wise Analysis, Weighted Criteria Analysis, Dot Voting, Binary Search Tree, Ranking, Numeral Assignment Technique, Requirements Triage, Wieger's Matrix Approach, Quality Function Deployment, Bucket technique, Cumulative Voting, Round-the-Group Prioritization, Theory-W, and Theme Scoring [14,15].

Changes to requirements in a plan-based environment are difficult and costly. Thus, a change the user considers simple may translate into a painful process for the developers. By definition, this is not the case for requirements in agile methods. This fundamental difference may have an impact on the optimal choice of prioritization technique.

Mead conducted a case study to determine the most suitable requirements prioritization methods to be used in software development [5]. This study compared three common methods: Numeral Assignment Technique, Theory-W, and AHP. The prioritization method comparison was based on five aspects: clear-cut steps, quantitative measurement, high maturity, low labor-intensity, and shallow learning curve. The results indicated that the AHP ranked the highest score of 16, while the Numeral Assignment Technique scored a 12, and Theory-W scored an 8.

## III. Methodology

The primary objective of this study is to investigate how the AHP can be used to rank the user stories prioritization techniques. The methodology used in this study is the case study methodology described in [16].

The following research questions provided a focus for our case study investigation:

(1) How does the AHP help select a prioritization technique for user stories?

(2) How do the AHP results affect the relationships among developers relation and their performance?

The units of analysis for this study derive from these research questions. The main focus is to rank several tools that can be used to prioritize user stories. Accordingly, ranking and the evaluation process are two the units of analysis for this study. Also, we consider the developers view of how the AHP benefits each XP practice. As result, our study is designed as multiple cases (embedded) with two units of analysis.

## IV. Data Collection and Sources

In the beginning of the study, we found the criteria affecting the ranking process and helping to examine the AHP tool ability and benefits. This data was collected from literature review and previous studies. To increase the validity of this study, data triangulation was employed. The data sources in this study were:

1. Archival records such as study plans from the graduate students.
2. Questionnaire given to the participants when developing the XP project.
3. Open-ended interviews with the participants.
4. Feedback from the customer.

## V. Case Study

The case study was conducted in the Advanced Software Design course offered to graduate students in Fall 2012 at the University of Regina. The participants were 12 Master's students and a client from a local company in Regina. Participants have various levels of programming experience and a good familiarity with XP and its practices. The students background related to the case study included several programming languages such as Java, C, C#, and ASP.net. All participants had previous project development experience. The study was carried out throughout 15 weeks; the students were divided into two teams. Both teams were assigned to build a project called "Issue Tracking System" brought in by the client along with a set of requirements compatible with current industry needs. The project evolved through 5 main iterations and by the end of the semester, all software requirements were implemented. The students were requested to try all requirements in each prioritization technique before applying AHP to rank them. Participants were given detailed lectures and supporting study materials on Extreme Programming practices that focused on planning game activities which included writing user stories, prioritizing the stories, estimating process parameters, and demonstrating developers commitments. The students were not new to the concept of XP, but they gained more knowledge and foundation specifically in the iteration plan, release planning and prioritizing the user stories. In addition, the students were exposed to the AHP methodology and learned the processes necessary to conduct the pairwise comparisons and to do the calculations. Several papers and different materials about AHP and user stories were given to the students to train them and increase their skills in implementing the methodology. Finally, a survey was distributed among students to get further information about their personal experiences and knowledge.

## VI. The Analytical Hierarchy Process

AHP is a systematic approach for decision-making that involves the consideration of multiple criteria by structuring them in a hierarchical model. AHP reflects human thinking by grouping the elements of a problem requiring complex and multi-aspect decisions [17]. The approach was developed by Thomas Saaty as a means of finding an effective and powerful methodology that can deal with complex decision-making problems [8]. AHP comprises the following steps: 1) Structure the hierarchy model for the problem by breaking it down into a hierarchy of interrelated decision elements. 2) Define the criteria or factors and construct a pairwise comparison matrix for them; each criterion on the same level of the decision hierarchy is compared with other criteria in respect of their importance to the main goal. 3) Construct a pairwise comparison matrix for alternatives with respect to each objective in separate matrices. 4) Check the consistency of the judgment errors by calculating the consistency ratio. 5) Calculate the weighted average rating for each decision alternative and choose the one with the highest score. More details on the method, including a step-by-step example calculation, are found in [17].

Saaty developed a numerical scale for assigning the weight for criteria or alternative by giving a value between 1 (equal importance) and 9 (extreme importance) [18]; see Table 1 for details.

TABLE 1. AHP NUMERICAL SCALE DEVELOPED BY SAATY..

| Scale | Numerical Rating | Reciprocal |
|---|---|---|
| Equal importance | 1 | 1 |
| Moderate importance of one over other | 3 | 1/3 |
| Very strong or demonstrated importance | 7 | 1/7 |
| Extreme importance | 9 | 1/9 |
| Intermediate values | 2,4,6,8 | 1/2, 1/4, 1/6, 1/8 |

## VII. PRIORITIZATION TECHNIQUES

There are several methods for prioritizing the system requirements; the six most commonly used can be summarized as follows:

### 1) *The 100-Dollar Test (Cumulative Voting)*

This is a straightforward technique described by Leffingwell and Widrig where each stakeholder gets 100 imaginary units (money, hours, etc) to distribute among the given requirements [19]. If the requirements are too many, it is recommended to use more units of value for more freedom in the prioritization [20]. After distributing the units on the requirements, stakeholders calculate the total for each requirement and rank the requirements accordingly.

### 2) *MoSCoW*

This is one of the methods for prioritization originating from the Dynamic Software Development Method (DSDM) [21]. The requirements are classified into four groups depending on the importance of the functional requirements [22]:

- M: MUST have this. It is the highest priority and without it the project considered a failure.
- S: SHOULD have this requirement if possible. Customer satisfaction depends on this requirement. But we cannot say its absence causes a project to fail.
- C: COULD have this requirement if it doesn't affect anything else.
- W: WON'T have the requirement this time but WOULD like to in the future.

This technique helps understand customer needs. The problem with this method is the difficulty of distinguishing the terms "Must" and "Should" as they both express a customer preference or desire..

### 3) *Top-Ten Requirements*

In this approach, the stakeholders select their top ten requirements without giving them a specific priority [23]. This is to avoid the conflict between stakeholders that may arise from the desire to support specific requirements. However, if stakeholder alignment is low, it is possible that none of the choices for some stakeholders will appear in the aggregated top priority requirement list.

### 4) *Kano Model*

This method was established for product development by Noriako Kano in 1987 to classify the requirements into five categories based on the answers to two questions about every requirement: 1) functional question: "How do you feel if this feature is present?"; 2) dysfunctional question: "How do you feel if this feature is NOT present?" [24].

The customer has to choose one of the five possible options for the answers [25]:

1. I like it.
2. I expect it.
3. I'm neutral.
4. I can tolerate it.
5. I dislike it.

### 5) *Themes Screening*

This is a technique employed when stakeholders have many relevant user stories that need to be grouped together. While writing the stories, stakeholders eliminate similar stories or ones that have already been covered by others. Then they follow the steps below [26]:

1. Identify 5-9 (approximately) selection criteria that are important in prioritizing the themes.
2. Identify a baseline that is approved and understood by all the team members.
3. Compare each theme to the baseline theme for each criterion. Use "+" for themes that rank "better than" the baseline theme, "-" for themes that rank "worse than" the baseline theme and "0" for themes that rank "equal" to the baseline theme.
4. Calculate the "Net Score" by summing up all the plusses and minuses. Rank as number one the theme that received the highest Net Score.

### 6) *Relative Weighting*

This technique involves the evaluation of each requirement based on the effect of its presence and its absence. A scale from 0 to 9 is identified for each requirement, 0 being a low effect and 9 being a high effect. Stakeholders will give every feature a value for its presence as well as a penalty for its absence and estimate its implementation cost. The priority is calculated by dividing the total value by the total cost to generate a prioritization indicator [26].

## VIII. PROPOSED CRITERIA FOR RANKING

To rank each technique, it is necessary to determine the most important criteria that affect the participants when choosing a prioritization process. The resulting criteria will be compared among each other. Finally, the prioritization techniques will be compared against each of the criteria [27]. In this paper, we propose four prioritization criteria that emerged during the course of the case study we conducted, but the method described in this paper can be applied to any set of criteria. The criteria shown below are simply illustrative of the prioritization method.

1. Simplicity: What is the simplest prioritization technique in terms of ease of understanding and application?
2. Time: Which one of these techniques will save the most time when the team applies it to the user stories?
3. Accuracy: Which one of these techniques will give the most accurate results?
4. Collaboration: Which one of these techniques will achieve the highest degree of collaboration among the stakeholders and the XP team in general?

## IX. AHP IN PRACTICE

The first step in the Analytic Hierarchy Process is to structure the problem as a hierarchy. In this paper, such a hierarchy includes three levels. The top level is the main objective: ranking the prioritization techniques. The second level is the prioritization criteria: simplicity, time, accuracy, and collaboration. The third level is the alternatives: 100-Dollar, Top-Ten, Kano Model, Theme Screening, Relative Weighting, and MoSCow. Fig. 1 illustrates the AHP hierarchy we chose for this paper.

Then, the hierarchy is used to generate appropriate AHP tables. All team members receive these tables, which shortens the time to fill them and facilitates the comparison process. A cover page dedicated to collecting general information of each team member including experience, type, and level of programming skills is also handed out. A matrix is then used to compare the four prioritization criteria.

Accordingly, we required all students to use the prioritization techniques throughout the project to experience their advantages and disadvantages. Then, we asked the students to evaluate these techniques based on the prioritization criteria. To accomplish this, we provided them with the AHP tables and cover page described above.

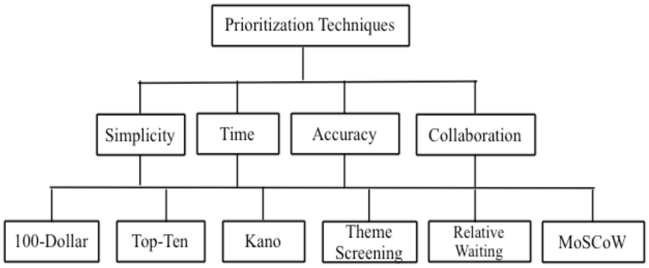

Figure 1. AHP Structure for Ranking the Prioritization Techniques

The students first compared the criteria among each other using the Saaty scale, ranging from 1 to 9. The students used a checklist with the following questions:

- Which is more important: simplicity or time and by how much?
- Which is more important: simplicity or accuracy and by how much?
- Which is more important: simplicity or collaboration and by how much?
- Which is more important: time or accuracy and by how much?
- Which is more important: time or collaboration and by how much?
- Which is more important: accuracy or collaboration and by how much?

After finishing the criteria comparisons, the students had to evaluate all the prioritization techniques against each other based on each criterion every time. An example follows:

- In term of simplicity, which is simplest: 100-Dollar or Top-Ten and by how much?

The same questions and comparisons were repeated for all prioritization techniques and criteria.

## X. FINDINGS AND RESULTS

Each student individually evaluated the prioritization techniques based on the criteria mentioned earlier. The Expert Choice software [28] was used to calculate the aggregation results for the entire two teams.

The results for Team 1 show that the highest rank was given to the relative weighting technique, followed by MoScoW, Theme Screening, Kano, Top-Ten and 100-Dollar. Table 2 provides the relative scores of each ranking as percentages.

The software also allows us to examine the importance of each criterion as perceived by Team 1 (Fig. 2). It appears that accuracy was the most relevant criterion for the team, followed by simplicity, collaboration and time.

TABLE 2. PRIORITIZATION TECHNIQUE RANKING FOR TEAM 1

| Technique | Scores |
|---|---|
| Relative Weighting | 24.39% |
| MoScoW | 20.38% |
| Them Screening | 17.70% |
| Kano | 15.81% |
| Top-Ten | 12.75% |
| 100-Dollar | 8.97% |

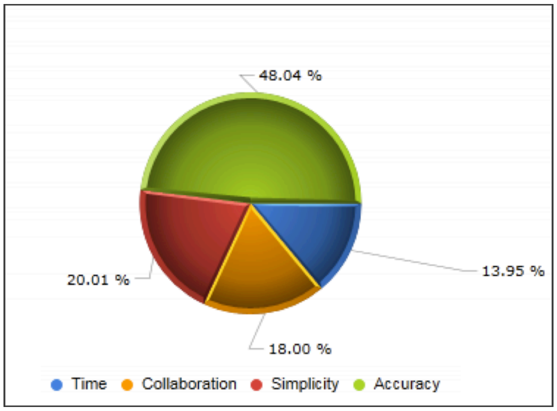

Fig.2 The Importance of the Criteria by Team 1

TABLE 3. PRIORITIZATION TECHNIQUE RANKING FOR TEAM 2

| Technique | Scores |
|---|---|
| Relative Weighting | 32.67 % |
| Top-Ten | 26.12 % |
| MoScoW | 15.44 % |
| Theme Screening | 15.35 % |
| 100-Dollar | 7.15 % |
| Kano | 3.27 % |

The results for Team 2 paint a somewhat different picture: the Relative Weighting technique is still on top, but it is followed by Top-Ten, MoScoW, Theme Screening, 100-Dollar and finally Kano. Table 3 provides the relative scores of each ranking as percentages.

As for the importance of each criterion as perceived by Team 1 (Fig. 3), it appears that accuracy was still the most relevant prioritization criterion, followed by time, collaboration and simplicity.

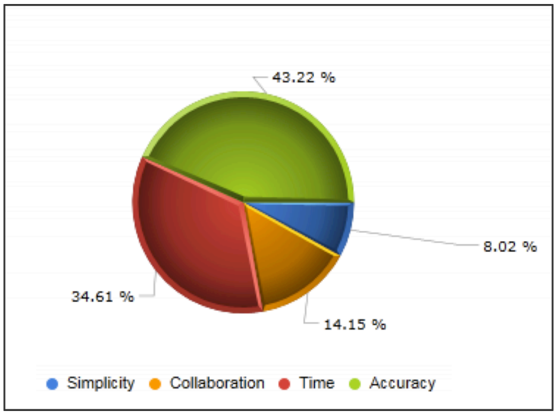

Fig. 3 The Importance of the Criteria by Team 2

## XI. OBSERVATIONS

### a) AHP Ranking Result

- When all the criteria were considered together, the Relative Weighting technique was ranked the highest by both teams. The MoScoW technique was ranked in the second position by Team 1 and third position by Team 2. The 100-Dollar technique was ranked in the last position by Team 1 and in the second to last position by Team 2.
- Both teams considered accuracy as the most important criteria. Simplicity in Team 1 and time in Team 2 respectively were considered to be the second highest important criterion.
- When the prioritization techniques were ranked considering each criterion individually, we found that for Team1 the MoScoW technique was ranked the highest in terms of simplicity and time criteria. Relative weighting was ranked the highest in terms of accuracy and collaboration criteria. Results related to Team2 are slightly different: the Top-Ten technique ranked the highest in terms of simplicity and time criteria. Relative weighting ranked the highest in terms of accuracy and collaboration criteria.
- These results are indicative of different choices made in each team. Although the ranking was achieved through individual comparisons, the group behavior was consistent as reflected in the consistency scores, which allowed the software to aggregate results from team members.

### b) Interview Results

The interview was conducted after showing the participants the results of the AHP evaluation for all the XP practices. Some of the results were surprising and others were expected. The interview included open questions to obtain the students' general opinions about AHP, the advantages and disadvantage of the using AHP, and the best experience of AHP among all the XP practices. As noted previously, the data was collected in the form of handwritten notes during the interviews. These notes were organized in a folder for the sake of easy access and analysis.

From the interviews, we found very positive feedback from the participants regarding AHP. It was felt that AHP resolved any conflicting opinions and brought each team member's voice to the decision in a practical way. AHP also emphasized the courage of the team by letting every opinion be heard. The time and the number of comparisons were the main concerns of the participants. All of them recommended using AHP in the future with XP. There were a few additional recommendations as well, such as developing an automated tool to reduce the time required for the AHP calculation, adding the mobility features, performing cost and risk analysis, and trying AHP in other XP areas and studying the outcomes.

### *c) Questionnaires*

Questionnaires were also given to the participants in order to obtain their perceptions of and experiences with AHP. The questionnaires were divided into two main parts. The first part contained questions about AHP as a decision and ranking tool. The second part contained questions regarding the direct benefits of the XP practice and investigated the participants' satisfaction. We used a seven-point Likert scale to reflect the level of acceptability of the AHP tool as follows:

1. Totally unacceptable
2. Unacceptable.
3. Slightly unacceptable.
4. Neutral.
5. Slightly acceptable.
6. Acceptable.
7. Perfectly Acceptable.

Once the participants completed the questionnaire, we aggregated the responses and presented the total percentage of the acceptability for each statement.

The total percentage of the acceptability was calculated as follows:

*d) The total percentage of acceptability (TPA)*

= The average of the score for each team * 100 / 7.

*e) The average of the score for each team =*

= The sum of the scores given by the team members / number of the team.

The following percentages show the acceptability level for the AHP as a ranking tool:

- Improving team communication: Team 1 scored 83% and Team 2 scored 86%.
- Creating a healthy discussion and learning opportunities: Team 1 scored 74% and Team 2 scored 93%.
- Clarifying the ranking problem: Team 1 scored 86% and Team 2 scored 93%.
- Resolving conflicting opinions among members: Team 1 scored 78% and Team 2 scored 93%.
- Increasing team performance: Team 1 scored 74% and Team 2 scored 88%.

## XII. VALIDITY

Construct validity, Internal Validity, External Validity and Reliability describe common threats to the validity of the study [29]. "Empirical studies in general and case studies in particular are prone to biases and validity threats that make it difficult to control the quality of the study to generalize its results" [30]. In this section, relevant validity threats are described. A number of possible threats to the validity of this work can be identified.

### *a)* Construct validity

Construct validity deals with the correct operational measures for the concept being studied and researched. The major threat to this study is the small number of participants in each case study.

This threat was mitigated by using several techniques in order to ensure the validity of the findings.

- Data triangulation: A major strength of case studies is the possibility of using many different sources of evidence [29]. This issue has been taken into account through the use of surveys and interviews with different types of participants from different environments with various levels of skills and experiences, and through the use of several observations as well as feedback from those involved in the study. By establishing a chain of evidence, we were able to reach a valid conclusion.
- Methodological triangulation: The research methods employed were a combination of a project conducted to serve this purpose, interviews, surveys, AHP results comparisons, and researchers' notes and observations.
- Member checking: Presenting the results to the people involved in the study is always recommended, especially for qualitative research. This is has been done by showing the final results to all participants to ensure the accuracy of what was stated and to guard against researcher bias.

### *b)* Internal validity

Internal validity is only a concern for an explanatory case study [29], and it focused on establishing a causal relationship between Students and educational restraints.

This issue can be addressed by relating the research questions to the study's propositions and other data sources providing information regarding the questions.

### *c)* External validity

External validity is related to the domain of the study and the possibilities of generalizing the results. To provide external validity to this study, we will need to conduct an additional case study in the industry involving experts and developers and then observe the similarities and the differences in the findings of both studies. Thus, future work will contribute to accrue external validity.

### *d)* Reliability

Reliability deals with the data collection procedure and results. Other researchers should arrive at the same case study findings and conclusions if they follow the same procedure. We address this by making the research questions, case study set up, data collection and analysis procedure plan available for use by other researchers.

## XIII. Conclusions

After using AHP to rank the common requirement prioritization techniques used in XP development to prioritize the user stories, AHP was found to be a relevant and useful tool that affords very good vision to stakeholders when they want to decide on which prioritization technique is the most suitable. Considering simplicity, time, accuracy and collaboration when selecting a prioritization technique could bring many advantages to the XP team, including the stakeholders. The relative weighting technique was the most preferred method for both teams in our case study, but the procedure we followed is general and thus the ranking can change depending on the team. More importantly, though, AHP helped students evaluate each prioritization technique from different viewpoints. In addition, they could mathematically reconcile the conflict of opinions among them. AHP introduces a cooperative decision making environment, which accelerates the XP development process and maximizes the effectiveness of the software developed.

## References

[1] K.Beck, “Extreme Programming Explained: Embrace Change,” 2nd edition, Addison Wesley, 2000.

[2] M.Cohn. “Advantage of User Stories for Requirements, Information Network,” (October 2004)

[3] K.Wiegers, “Software Requirements,” Microsoft Press, Redmond, In Engineering and Managing Software Requirements,2003.

[4] Lawson. “Software Requirements-Styles and Techniques,” Pearson Education, Essex, 2002.

[5] R. Mead, “Requirements Prioritization Introduction,” Software Engineering Institute, 2006-2008 Carnegie Mellon University.

[6] Z. Bakalova, M. Daneva, A. Herrmann, and R. Wieringa, “Agile Requirements Prioritization: What Happens in Practice and What Is Described in Literature,” In D. Berry & X. Franch (Eds.), Requirements engineering: Foundation for software quality, LNCS, vol. 6606, 2011, pp. 181-195. Heidelberg, Germany: Springer Berlin Heidelberg.

[7] G. Hoff, A. Fruhling, and K.Ward, “Requirements Prioritization Decision Factors for Agile Development Environments,” University of Nebraska at Omaha, 2008.

[8] I. Sommerville and P. Sawyer, “Requirements Engineering: A Good Practice Guide,” John Wiley & Sons Ltd, Chichester, England, 1997.

[9] A. Davis, “The Art of Requirements Triage,” IEEE Computer, Vol. 36, No. 3, March 2003, pp. 42- 49.

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

[11] B. Boehm, “The High Cost of Software,” Practical Strategies for Developing Large Software Systems, Addison- Wesley, Reading MA, 1975.

[12] P. Berander and A. Andrews, "Requirement Prioritization," in Engineering and Managing Software Requirements, Berlin, Deutschland, 2005.

[13] C. Patel and M. Ramachandran, “Story Card Based Agile Software Development,” in International Journal of Hybrid Information Technology, vol. 2, no. 2, April.2009.

[14] Z. Racheva, M. Daneva, and L. Buglione, “Supporting the Dynamic Reprioritization of Requirements in Agile Development of Software Products,” Second International Workshop on Software Product Management, 2008.

[15] Q. Ma, “The Effectiveness of Requirements Prioritization Techniques for a Medium to Large Number of Requirements: A Systematic Literature Review,” thesis for a degree of master of Computer and Information Sciences, Auckland University of Technology, 2009.

[16] K. Yin, “Case Study Research: Design and Methods,” Second Edition, SAGE Publications, 1994.

[17] N. Tiwari. “Using the Analytic Hierarchy Process (AHP) to Identify Performance Scenarios for Enterprise Application” (2006)

[18] T. Saaty, “The Analytic Hierarchy Process,” McGraw-Hill, New York, 1980.

[19] D. Leffingwell and D. Widrig, “Managing Software Requirements: A Use Case Approach,” 2nd ed. Addison-Wesley, Boston (2003).

[20] P. Berander and C. Wohlin, “Different in Views between Development Roles in Software Process Improvement – A Quantitative Comparison,” In: Proceedings of the 8th International Conference on Empirical Assessment in Software Engineering (EASE 2004). IEE, Stevenage, 2004, pp. 57-66.

[21] K. Waters, “Prioritization Using MoSCoW,” Agile Planning, (12 January 2009)

[22] The MoSCoW Prioritization Technique, LMR Technologies, Agile Practices: Scrum, XP, Lean, Kanban: www.lmrtechnologies.com [retrieved: October, 2013].

[23] K.Wiegers, First Things First: Prioritizing Requirements, Software Development, vol. 7, no. 9, September 1999.

[24] E. Zultner, H. Mazur. The Kano Model: Recent Developments, Richard QFD Institute, Austin, Texas,2006.

[25] A. Hand, “Applying the Kano Model to User Experience Design,” UPA Boston Mini-Conference,May 2004.

[26] M. Cohn, “User Stories Applied for Agile Software Development,” Addison-Wesley Professional; 1 edition (March 11, 2004)

[27] T. Saaty, “How to Make a Decision: the Analytic Hierarchy Process,” Interfaces, vol. 24, no. 6, 1994, pp.19-43.

[28] Expertchoice for Collaborative Decision Making: http://www.expertchoice.com [retrieved: October, 2013].

[29] R.K. Yin, Case Study Research – Design and Methods, 3rd edition, Sage Publications, London, 2003.

[30].R. Lincke, “How do PhD Students Plan and Follow-up their Work? – A Case Study,” School of Mathematics and Systems Engineering, University Sweden.