

Available online at www.sciencedirect.com**ScienceDirect**

Procedia Technology 11 (2013) 733 – 739

Procedia
Technology

The 4th International Conference on Electrical Engineering and Informatics (ICEEI 2013)

Prioritizing Requirements in Agile Development: A Conceptual Framework

Rami Hasan AL-Ta'ani*, Rozilawati Razali

*Center of Software Technology and Management, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia,
43600 UKM Bangi, Selangor DarulEhsan*

Abstract

The advance of technology and business needs have triggered user requirements to evolve almost every day. It is therefore impossible to fulfill the requirements at once. Agile methods have been introduced to mitigate such issues by implementing user requirements incrementally and iteratively. Selecting a group of prominent requirements to be implemented in iterations is considered as essential in agile development. This can be achieved by performing continuous requirements prioritisation. Practitioners however are confronted with difficulties in making decisions about which requirements should be considered for implementation at the inter-iteration period. This paper proposes a conceptual framework that outlines the contributing factors and how these factors affect the process and the final product of requirements prioritisation process in agile methods. The factors were identified by reviewing the related work. The data were analysed by using content analysis. The proposed conceptual framework aims to guide practice and future research on effective requirements prioritisation process that can produce high quality requirements which meet the needs of the intended users.

© 2013 The Authors. Published by Elsevier Ltd. Open access under [CC BY-NC-ND license](http://creativecommons.org/licenses/by-nc-nd/4.0/).

Selection and peer-review under responsibility of the Faculty of Information Science & Technology, Universiti Kebangsaan Malaysia.

Keywords: Requirements Prioritization, Agile Methods

1. Introduction

Business is presently operated in parallel with global and rapidly changing environment. Organizations have to respond to new opportunities and markets, dynamic economic conditions and the emergence of competing products and services. Nowadays, software systems have become the backbone of most business operations. They influence

* Corresponding author. Tel.: +6-018-974-9116;
E-mail address: ramitaani@yahoo.com

the structure and operation of organizations more profoundly than any other technology ever has. To fulfill the current business trends, new and improved software systems are expected to be available instantly in order to allow organizations to optimize the arriving opportunities and to respond to competitive pressure. This implies that the traditional plan-driven software development approach, which requires comprehensive process flow and milestones, may become no longer appropriate. Rather, a development methodology is highly needed for delivering software systems quickly without jeopardizing their quality. In general, a software system is considered as having high quality if it satisfies the needs of its users.

Agile methods have been proposed in 1990s with an aim to minimise process bureaucracy by avoiding unnecessary milestones due to the extensive documentation [31]. The methods are intended to deliver a software system quickly to users, who can then propose and change new business requirements into the system. The philosophy behind agile methods is reflected in the agile manifesto [31], which values individuals and interactions, working software, customer collaboration and responding to change. To fulfill those principles, the methods address small, co-located, dedicated and highly collaborative teams [9,17,33]. Some examples of agile methods are Feature Driven Development [47], Scrum [14], Extreme Programming [6], crystal [13], Dynamic System Development Method [46] and Adaptive Software Development [24].

Agile methods are categorised as an iterative and incremental development approach that promotes constant delivery of product features [42]. The methods aim to deliver features iteratively in the form of several releases or iterations. The features to be implemented and delivered as releases are prioritised based on customer requirements and changes [31]. The continuous requirements prioritisation is thus the core activity of agile methods [36]. Despite its importance, it is unclear how the requirements prioritisation process in agile methods is conducted. The process is not considered as similar to the ones conducted in conventional approaches due to the unique principles that the methods employ. Since the process determines the quality of the requirements produced, it is essential to explore the contributing factors of an effective requirements prioritisation process in agile development.

The aim of this paper is to identify the factors that influence the selection of requirements during prioritisation process in agile development. The paper is organised as follows: Section 2 investigates the related work regarding requirements prioritisation process in agile development. Section 3 explains the proposed conceptual framework and discusses how it can be applied. Finally, Section 4 concludes the paper with a summary of the main findings and future work.

2. Related Work

Customers' willingness of buying a software system depends on the extent of which the product meets their needs. Establishing priorities offer opportunities for getting good results and customer satisfaction [20,50]. Requirements prioritisation is defined as an action during which the significant system requirements are identified and ordered based on their importance [5,26]. The requirements are then developed iteratively as releases or iterations. The idea is that the highest priority requirement has to be implemented first before the others [34,44].

During the requirements prioritisation process, a number of stakeholders determine which requirement should be implemented as releases [37,38, 48]. Being skillful people, the developers are considered as the most influential stakeholders in requirements prioritisation process [4]. However, the requirements priorities are best determined by customers who have the authority towards the system [48]. Both developers and customers have to decide which requirements should be prioritised [32]. This process is complicated particularly when it involves multiple stakeholders who have to reach consensus under hectic and often disoriented situations [44].

Choosing the right stakeholder to be involved in requirements prioritisation process is important in agile development [36]. The stakeholders should possess significant knowledge in agile development [22] as well as customers' needs and interests [4]. The contemplating process in prioritisation relies on joined learning experience of the stakeholders with respect to systematic (explicit) and anecdotal (tacit) knowledge [4, 35]. The stakeholders also must trust each other [11], which can be established through direct communication [43].

The requirements selection is accomplished by employing a specific requirements prioritisation technique, which has to be firstly identified. The outcome is a set of prioritised project backlog [38], which is a group of requirements that are significant for the project [4]. The requirements that are at the top of the prioritised project backlog list will be considered for implementation in the first iteration. These requirements are called sprint backlog. There are

occasions where some requirements in the sprint backlog could not be implemented in an iteration. They therefore are sent to the initial project backlog and subjected to reprioritisation before the new iteration starts. During reprioritisation, stakeholders identify the next portion of requirements that will go to the next sprint backlog [38].

Requirements are prioritised based on project constraints. Requirements with the least costs are more likely to be given higher priority [10]. Through cost-benefit analysis, the resources spent on implementing an iteration are compared with the benefit received [51, 52]. The analysis is important as the accurate blend of requirements for an iteration would exploit values for the business [18,45]. On the other hand, the process also depends on the personnel available to implement the iteration and the project schedule. It is well known that an iteration could not be implemented unless the right personnel are available at the specified time [51]. In agile development, fixed delivery dates are rigidly enforced as time to market is essential. Postponing requirements to a later iteration is an alternative of going over time [15,21,39]. Moreover, the possibility of the requirements to cause project failure is also considered during the prioritisation process [18,23].

Besides project, the nature of requirements also influences the prioritisation process. Requirement dependencies increase the complexity of requirements and thus complicate the selection for a certain iteration [4]. Dependencies can be chronological or architectural. A requirement that is judged to be complex and risky may not be implemented [51]. The importance and urgency of the requirements also play a role, which depend highly on the perspective of the stakeholders [51]. Important requirements are functions that are required early and bring strategic business values to the organisation [28,29]. In addition, requirements stability or volatility is also taken into consideration, as unstable requirements affect the cost and schedule of a project [27].

The above reviews indicate that previous studies have attempted to identify the contributing factors for requirements prioritisation process in agile development. However, these factors are isolated. None of the studies have clearly depicted the interrelation among those factors. This paper therefore aims to theoretically integrate the factors as a conceptual framework. The framework could then be used to guide future work on the subject matter.

3. The Conceptual Framework

This paper addresses the following questions:

What are the contributing factors that influence requirements prioritisation process in agile development? How are the factors interrelated?

In order to answer the questions, this study employed content analysis to categorise the factors. Content analysis is a scientific method that summarises and analyses textual messages. Content analysis compresses words in a text in a replicable and systematic way to obtain fewer content categories depending on coding explicit rules [49]. The coding process is conducted by assigning a label for each text segment which in turn may be ranged from a few words to a full paragraph. The purpose of such coding is to rearrange and integrate the interrelated words, sentences or paragraphs to obtain a meaningful description about the data [49]. In this study, the results of content analysis are categories of factors that influence requirement prioritisation process. The categories are conceptualised as a framework, as shown in Fig. 1.

The conceptual framework consists of three parts; Environment, Process and Product. The first part encompasses Environment aspect, which is categorised into three main factors, namely Stakeholders Characteristics, Project Constraints and Requirements Nature. The second part is Process, which comprises the steps involved in requirement prioritisation process. The Product is the third part that outlines the outcomes of the process. The following paragraphs describe the parts and factors involved.

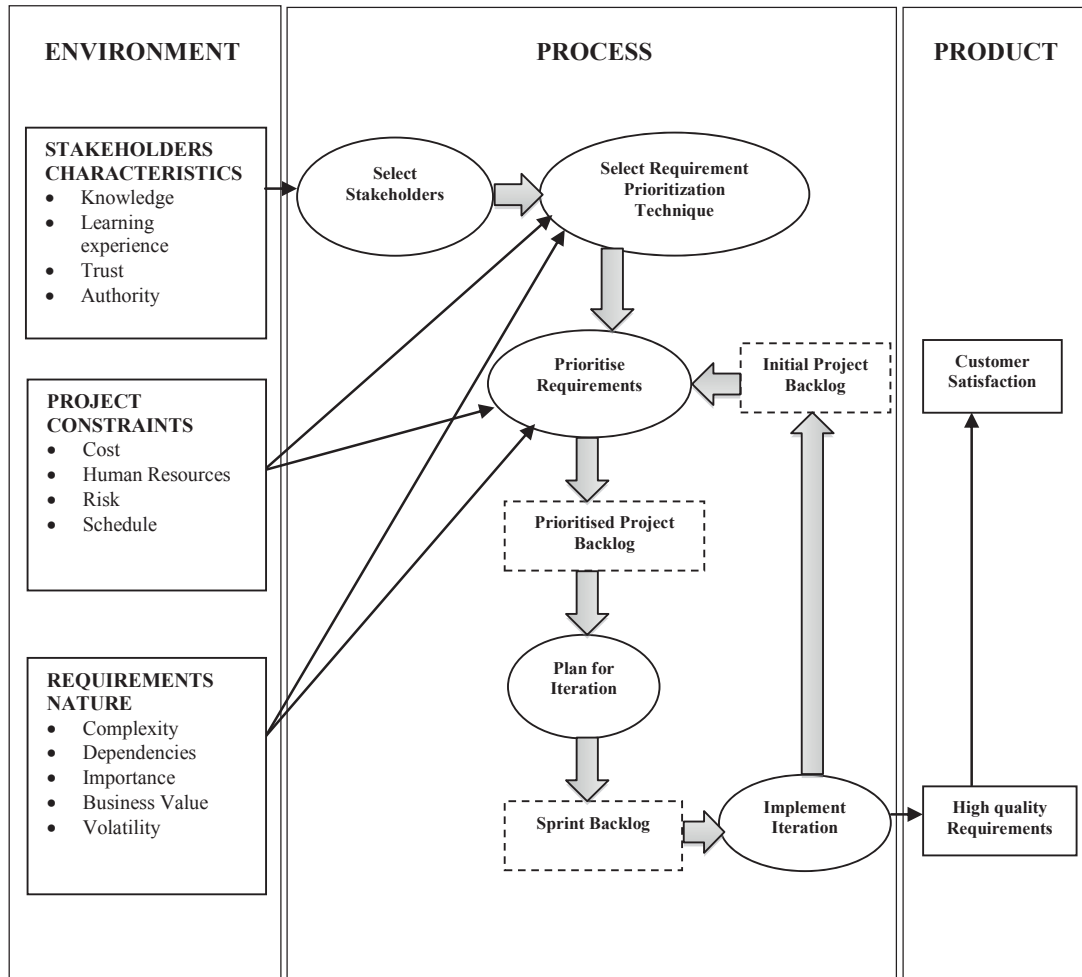


Figure.1: The Conceptual Framework of Requirements Prioritisation Process in Agile Development

3.1 Environment

Environment can be categorised into three factors: Stakeholders Characteristics, Project Constraints and Requirements Nature. Stakeholders can be classified into four subfactors: Trust, Knowledge, Learning Experience and Authority. The identification and consideration of these aspects are essential for selecting the appropriate stakeholders to be involved in requirements prioritisation process. The project constraints such as Cost, Human Resources, Risk and Schedule need to be considered in prioritising requirements in agile. Similarly, the requirements' Complexity, Dependencies, Importance, Business Value and Volatility also influence the selection.

3.2 Process

Prior to requirements prioritising process, the right group of stakeholders should be determined and selected [37,38,48]. Stakeholders should possess the required characteristics as mentioned earlier. Later, the

appropriate requirements prioritisation technique is selected by the stakeholders based on project constraints and requirements nature. Some examples of requirements prioritisation techniques that can be used include analytical hierarchy process, numerical assignment, top-ten requirements or combination of techniques [8]. By using the technique, a prioritised project backlog is generated. Project backlog area set of requirements that are substantial for the project. In agile setting, the requirements with high priority will be firstly implemented in the first iteration. The selected requirements to be implemented in an iteration are called sprint backlog. The requirements which are not implemented in the iteration due to dropping or adding requirements are brought into the initial project backlog so that reprioritisation process can be executed. During reprioritisation process, the previous unimplemented requirements are considered together with other requirements in the list. Similar to prioritisation, the reprioritisation is based on project constraints and requirements nature. The outcome of the process is a sprint backlog for the next iteration.

3.3 Product

If the requirements prioritisation process is executed effectively, high quality requirements could be generated. Effectiveness is defined as the extent to which planned activities are realised and planned results are achieved [12]. High quality requirements normally contribute towards customers' satisfaction. In agile development, customer satisfaction is the ultimate goal to achieve.

The conceptual framework offers a theoretical understanding on how the above mentioned factors are interconnected. Practitioners can use the conceptual framework as a guide to perform effective requirements prioritisation process in agile development. The framework indicates that practitioners should pay attention to not only the prioritisation itself but also the stakeholders and technique selection process. To researchers, the proposed conceptual framework highlights some research opportunities in agile requirements prioritisation. It is necessary to devise mechanisms for selecting stakeholders and requirements prioritisation technique in agile development. The framework also seems to indicate that the requirements prioritisation and reprioritisation process in agile development is complex. Therefore, a systematic process needs to be formulated.

4. Conclusions and Future Work

This paper has discussed the factors that have to be taken into consideration while performing requirements prioritisation in agile development. This is crucial as erroneous requirements prioritisation may increase the cost of development and lead to project and system failures. In this study, the contributing factors concerning effective requirements prioritisation in literature were collated through content analysis as a conceptual framework. The findings indicate that there are three aspects involved in requirements prioritisation process in agile development: Environment, Process and Product. The Environment consists of three factors, namely Stakeholders Characteristics, Project Constraints and Requirement Nature. The Process outlines the activities involved in the requirements prioritisation process, which are executed based on the identified factors. The Product shows the ultimate outcome of the Process. The proposed conceptual framework emphasises the significant elements that are worthwhile for further exploration. For example, future research may investigate the methods or techniques that can help in selecting stakeholders and prioritisation techniques as well as conducting systematic requirements prioritising and reprioritising process in agile development.

References

- [1] Ahmed, W. B., & Yannou, B. Polysemy of values or conflict of interests: A multi-disciplinary analysis. *International Journal of Value-Based Management* 2003; 16 (2), p. 153-179.
- [2] Ambler, S. *Agile modeling: Effective practices*: John Wiley and Sons; 2002.
- [3] Aurum, A., & Wohlin, C. *Engineering and managing software requirements*: Springer-Verlag New York Inc; 2005.
- [4] Bakalova, Z., Daneva, M., Herrmann, A., & Wieringa, R. Agile requirements prioritization: what happens in practice and what is described in literature. *Requirements Engineering: Foundation for Software Quality*; 2011. p. 181-195.

- [5] Bebensee, T., van de Weerd, I., & Brinkkemper, S. Binary Priority List for Prioritizing Software Requirements. *Requirements Engineering: Foundation for Software Quality*; 2010.p.67-78.
- [6] Beck, K. *Extreme programming explained: embrace change*: Addison-Wesley; 2001.
- [7] Beck, K., & Fowler, M. *Planning extreme programming*: Addison-Wesley Professional; 2001.
- [8] Berander, P. and A. Andrews, *Requirements Prioritization; Engineering and managing software requirements*: Springer2005.
- [9] Boehm, B. Get ready for agile methods, with care. *Computer* 2002; 35(1), 64-69.
- [10] Boehm, B. W. The high cost of software. *Practical Strategies for Developing Large Software Systems*; 1975. p. 3-15.
- [11] Cao, L., & Ramesh, B. Agile requirements engineering practices: An empirical study. *Software, IEEE* 2008; 25(1), 60-67.
- [12] Carmichael, R.M., Measures of efficiency and effectiveness as indicators of quality: a systems approach. *Journal of institutional research South East Asia* 2002; 1(1): 3-14.
- [13] Cockburn, A. *Crystal clear: a human-powered methodology for small teams*: Addison-Wesley Professional; 2004.
- [14] Cohn, M. *Succeeding with agile: software development using Scrum*: Addison-Wesley Professional; 2009.
- [15] Dahlstedt, A., Karlsson, L., Persson, A., NattochDag, J., & Regnell, B. *Market-Driven Requirements Engineering Processes for Software Products—a Report on Current Practices*; 2003.
- [16] Davis, A. M. The art of requirements triage. *Computer*2003; 36(3), 42-49.
- [17] Dyba, T., & Dingsoyr, T. Empirical studies of agile software development: A systematic review. *Information and Software Technology* 2008; 50(9-10), 833-859.
- [18] Favare, J. Managing requirements for business value. *Software, IEEE* 2002; 19(2), 15-17.
- [19] Firesmith, D. Prioritizing requirements. *Journal of Object Technology* 2004; 3(8), 35-47.
- [20] Girase, S. Comparison of various Elicitation Techniques and Requirement Prioritisation Techniques. *International Journal of Engineering* 2012; 1(3).
- [21] Gorschek, T., Gomes, A., Pettersson, A., & Torkar, R. Introduction of a process maturity model for market-driven product management and requirements engineering. *Journal of Software: Evolution and Process* 2012; 24(1), 83-113.
- [22] Harris, R., & Cohn, M. Incorporating Learning and Expected Cost of Change in Prioritizing Features on Agile Projects. *Extreme Programming and Agile Processes in Software Engineering*; 2006.p.175-180.
- [23] Herrmann, A., & Paech, B. Practical challenges of requirements prioritization based on risk estimation. *Empirical Software Engineering* 2009; 14(6), 644-684.
- [24] Highsmith, J. A., & Orr, K. *Adaptive software development: a collaborative approach to managing complex systems*: Dorset House Pub ; 2000.
- [25] Hoff, G., Fruhling, A., & Ward, K. *Requirement Prioritization Decision Factors for Agile Development Environments* ; 2008.
- [26] Karlsson, J., Wohlin, C., & Regnell, B. An evaluation of methods for prioritizing software requirements. *Information and software technology*1998; 39(14), 939-947.
- [27] Lauesen, S. *Software requirements: styles and techniques*: Addison-Wesley Professional ; 2002.
- [28] Lehtola, L., Kauppinen, M., & Kujala, S. Requirements prioritization challenges in practice. *Product focused software process improvement*; 2004.p.497-508.
- [29] Lutowski, R. *Software requirements: encapsulation, quality, and reuse*: CRC Press ; 2005.
- [30] Maciaszek, L. *Requirements Analysis and System Design—Development Informatik Systems with UML*: Addison Wesley Publ ; 2001.
- [31] Manifesto, A. *Manifesto for Agile Software Development*. from <http://agilemanifesto.org/>; 2001.
- [32] McDaniels, T., & Small, M. J. *Risk analysis and society: an interdisciplinary characterization of the field*: Cambridge Univ Pr ;2004.
- [33] Nerur, S., Mahapatra, R. K., & Mangalaraj, G. Challenges of migrating to agile methodologies. *Communications of the ACM*. 2005;48(5), 72-78.
- [34] Paetsch, F., Eberlein, A., & Maurer, F. *Requirements engineering and agile software development* ; 2003.
- [35] Petersen, K., & Wohlin, C. A comparison of issues and advantages in agile and incremental development between state of the art and an industrial case. *Journal of Systems and Software*. 2009; 82(9) 1479-1490.
- [36] Pinheiro, F.A., *Requirements honesty*. *Requirements Engineering*. 2003; 8(3):183-192.
- [37] Racheva, Z., Daneva, M., Herrmann, A., & Wieringa, R. J. A conceptual model and process for client-driven agile requirements prioritization; 2010.
- [38] Razali, R., & Anwar, F. Selecting the Right Stakeholders for Requirements Elicitation: A Systematic Approach. *Journal of Theoretical and Applied Information Technology*2011; 33(2).
- [39] Regnell, B., Beremark, P., & Eklundh, O. A market-driven requirements engineering process: results from an industrial process improvement programme. *Requirements Engineering* 1998; 3(2), 121-129.
- [40] Royce, W. *Software Project Management: A Unified Framework*, 1998: Addison Wesley, ISBN:-0-201-30958-0.
- [41] Ruhe, G., Eberlein, A., & Pfahl, D. Trade-off analysis for requirements selection. *International Journal of Software Engineering and Knowledge Engineering* 2003; 13(4), 345-366.
- [42] Shammii, M., et al., *Agile Process for Integrated Service Delivery*. Delft University of Technology; 2011.
- [43] Sillitti, A., Ceschi, M., Russo, B., & Succi, G. Managing uncertainty in requirements: a survey in documentation-driven and agile companies ; 2005.

- [44] Sommerville. *Software Engineering* 5th edition ; 1996.
- [45] Sommerville, I., & Sawyer, P. *Requirements engineering: a good practice guide* John Wiley & Sons. Chichester, England; 1997.
- [46] Stapleton, J. *DSDM: Business focused development*: Addison-Wesley Professional; 2003.
- [47] Stephen R, P., & John M, F. *A Practical Guide to Feature Driven Development. A Practical Guide to Feature Driven Development*; 2002.
- [48] Svensson, R. B., Gorschek, T., Regnell, B., Torkar, R., Shahrokni, A., Feldt, R., et al. *Prioritization of quality requirements: State of practice in eleven companies*; 2011.
- [49] Weber, R. P. *Basic content analysis (Vol. 49)*: Sage Publications, Incorporated; 1990.
- [50] Wieggers, K. *First things first: prioritizing requirements. Software Development* 1999; 7(9), 48-53.
- [51] Wohlin, C., & Aurum, A. *What is important when deciding to include a software requirement in a project or release?*; 2005.
- [52] Wohlin, C., & Aurum, A. *Criteria for selecting software requirements to create product value: An industrial empirical study. Value-Based Software Engineering*; 2006. p. 179-200.