An Evaluation of Software Requirement Prioritization Techniques

Mohammad Shabbir Hasan[1, a], Abdullah Al Mahmood[2, a], Md. Jahangir Alam[3, b], Sk. Md. Nahid Hasan[4, a], Farin Rahman[5, a]

E-mail: [1]*shabbir_cse03@yahoo.com*, [2]*sajibcseofkuet@gmail.com*,[3]*palash_14@yahoo.com*, [4]*auvi_2k5@hotmail.com*,[5]*farinrahman.cse@gmail.com*,

a. Panacea Research Lab, Dhaka, Bangladesh

b. Dhaka International University, Dhaka, Bangladesh

***Abstract*— Requirements prioritization plays an important role in the requirement engineering process, particularly, with respect to critical tasks like requirements negotiation and software release planning. Selecting the right set of requirements for a product release largely depends on how successfully the requirement candidates are prioritized. There are different requirement prioritization techniques available which are some more elaborated than others. This paper takes a closer look at nine different techniques of requirement prioritization namely Analytical Hierarchy Process (AHP), Hierarchy AHP, Minimal Spanning Tree, Bubble Sort, Binary Search Tree (BST), Priority Group, Planning Game (PG), 100 points method and Planning Game combined with AHP (PGcAHP) and then put them into a controlled experiment, in order to find out the best one. The evaluation was done on the basis of some criteria like: ease of use, certainty, accuracy of result, method's ability to scale up to many more requirements, required number of comparisons, and required time to make decision. Analysis of the data from the experiment indicates that the analytic hierarchy process to be a promising candidate, although it may be problematic to scale-up. However, the result clearly indicates that the Planning Game (PG) yields accurate result, is able to scale up, requires least amount of time, the easiest method to use and so on. For these reasons, finding of the experiment is, the Planning Game (PG) method is supposed to be the best method for prioritizing requirements.**

Keywords- Requirement Engineering, Requirement Prioritization, Requirement Negotiation, Software Product Management, Software Release Planning.

I. INTRODUCTION

In market-driven software development, products which are intended for an open market are developed in several consecutive releases. Market-driven development does not have easily identifiable customers and the requirements often need to be invented based on the needs of several potential users [1]. When only one stakeholder is involved in the project, it is relatively easy to make decisions since only one stakeholder's opinion needs to be considered. But when more than one stakeholder is involved in the project, decisions can be harder to make, since different stakeholders have different perspectives and in this case, more requirements are yield than can be implemented at once. Again, not all the requirements contain equal user satisfaction. For example, project developers look for the requirements which can be implemented fast, financial managers look for the requirements with low cost, market managers look for the requirements with high market value, and end users look for the requirements which are easy to use. One requirement may be of low cost, with short implementation time, but also have low market value and be hard to use. Conversely, another requirement may have a high cost, but short time to be implemented, high market value and be easy to use. It can be a challenge for the software development team to decide which requirements need to be implemented first. Requirements prioritization is a technique that can uncover the most important requirements to maximize the stakeholders' satisfaction.

In commercial software development, decision makers need to make many different decisions regarding the release plan according to some issues like available resources, milestones, conflicting stakeholder views. Available market opportunity, risks, product strategies, and costs need to be taken into consideration when planning future releases. Nowadays, unfortunately, projects are suffering low success rates. According to an annual report named 'CHAOS Summary 2009' prepared by Standish Group [27], only 32% of all projects were considered as successful which are delivered on time, on budget, with required features and functions. Among the rest, 44% were challenged which are late, over budget, and/or with less than the required features and functions and 24% failed which are cancelled prior to completion or delivered and never used. Ten main factors causing challenged or failed projects are unveiled. Four of them are lack of user involvement, lack of resources, unrealistic expectations, and changing requirements and specifications. Requirements prioritization increases user involvement by letting the stakeholders decide which requirements the project should contain. It helps stakeholders to be realistic by letting them understand the current constraints on resources and accepting the trade-off

http://sites.google.com/site/ijcsis/
ISSN 1947-5500

decisions on conflicting perspectives. Karlsson et al think it helps stakeholders to allocate resources based on the priorities of the requirements [2], detect requirements defects, such as misunderstanding or ambiguous requirements [3] and reduce the number of changes to requirements and specifications in the later stage of projects. Hatton [4] says requirements prioritization has become an essential step in the software development process in order to reduce software failure. Ngo-The and Ruhe [5] note requirements prioritization has been recognized as one of the most important decision making processes in the software development process.

Several approaches have been proposed [6-10] which adopts a common model for the requirements prioritization process. This paper provides an investigation of nine candidate methods for prioritizing requirements: Analytic Hierarchy Process (AHP), Hierarchy AHP, Minimal Spanning Tree, Bubble Sort, Binary Search Tree (BST), Planning Game (PG), Priority Group, 100 points method and Planning Game Combined with AHP (PGcAHP). To study these methods, we systematically applied all methods to prioritize 14 well-defined quality requirements of a mobile set. We then categorized the methods from a user's perspective according to a number of criteria such as accuracy, certainty, method's ability to scale up to many more requirements, required time to make decision, total number of decisions and ease of use.

This paper is organized as follows. Section 2 motivates this work, and the paper continues in Section 3 by outlining the nine different prioritizing methods. Section 4 describes the evaluation framework and Section 5 presents the way to find out the best one among the techniques under consideration followed by a discussion of the result in Section 6. We finish by drawing some broad and necessarily speculative and personal conclusions in Section 7.

II. MOTIVATION

Industrial software development has a growing acknowledgement that requirements are of varying importance. Yet there has been little progress to date, either theoretical or practical, on the mechanisms for prioritizing software requirements [11]. In a review of the state of the practice in requirements engineering, Lubars et al. [12] found that many organizations believe that it is important to assign priorities to requirements and to make decisions about them according to rational, quantitative data. Still it appeared that no company really knew how to assign priorities or how to communicate these priorities effectively to project members [3].

A sound basis for prioritizing software requirements is the approach provided by the analytic hierarchy process, AHP [13] where decision makers compare the requirements pair-wise to determine which of the two is more important, and to what extent. In industrial projects, this approach has been experienced as being effective, accurate and also to yield informative and trustworthy results [7]. Probably even more important, after using the approach in several commercial projects, practitioners are found to be very attracted by the approach, and continue to use it in other projects [3]. AHP has only been used in few applications in the software industry. Finnie et al. [14], for example, used AHP to prioritize software development factors. Other applications of AHP include a telecommunications quality study performed by Douligeris and Pereira [15], and software requirements prioritizing in a commercial development project by Karlsson [16]. Despite some positive experience, AHP has a fundamental drawback which impedes its industrial institutionalization. Since all unique pairs of requirements are to be compared, the required effort can be substantial. In small-scale development projects this growth rate may be acceptable, but in large-scale development projects the required effort is most likely to be overwhelming [3].

Since AHP may be problematic for large-scale projects, Karlsson et al [3] identified five complementary approaches to challenge AHP. All of these methods involve pair-wise comparisons, since previous studies indicate that making relative judgments tends to be faster and still yield more reliable results than making absolute judgments [7]. They focused on methods which may reduce the required effort, but still able to produce high-quality results, considered trustworthy by its users. Again, Paetsch et al [17] claims that agile software development has become popular during the last few years and in this field, one of the most popular methods is the extreme programming, which has a prioritization technique called Planning Game (PG). In this paper, we investigated PG with all the requirement prioritization techniques used in the experiment carried out by Karlsson et al [3]. We also investigated a rather easy and quick method (at least according to the theory), and that is the 100 points method. Next section gives a brief description of each method, both in theory and then how it works practically.

III. PRIORITIZATION METHODS

Prioritizing methods guide decision makers to analyze requirements to assign numbers or symbols that reflect their importance. According to Karlsson et al [3], a prioritizing session may consist of three consecutive stages:

(1) The *preparation* stage where a person structures the requirements according to the principle of the prioritizing methods to be used. A team and a team leader for the session is selected and provided all necessary information.
(2) The *execution* stage where the decision makers do the actual prioritizing of the requirements using the information they were provided with in the previous stage. The evaluation criteria must be agreed upon by the team before the execution stage is initiated.

http://sites.google.com/site/ijcsis/
ISSN 1947-5500