A Collaborative Approach for Improvisation and Refinement of Requirement Prioritization Process

Ankita Gupta, Jaypee Institute of Information Technology, Noida, India
Chetna Gupta, Jaypee Institute of Information Technology, Noida, India

ABSTRACT

This article presents a systematic approach to prioritize requirements and estimate risk associated with each requirement. It first aims at providing short training to both developers and stakeholders to bridge the gap of understanding and comprehend requirements so that a refined priority value for each requirement can be obtained. Secondly, it presents a requirement risk and re-prioritization estimation model to make sure that a right decision has been taken by stakeholder and developers. The entire process has been supported with an example case study and by a survey that is conducted at IT companies. The response for the applicability of the proposed approach from the industry is appreciable and is promising as it will help in minimizing the disagreements between stakeholders and developers, thereby resulting in better collaboration. It will also help in minimizing the overall risk associated with development as only the most important functionality will be delivered in right sequence to the client thereby reducing the overall development time.

KEYWORDS
Collaboration, Requirement Engineering, Requirement Prioritization, Risk Estimation, Software Engineering, Stakeholders

INTRODUCTION

Requirement engineering (RE) is the first and most important phase of software development life cycle as requirements define what a software system should do. Studies have shown that RE does not receive more than 2-4% of the total effort applied in the entire project (Firesmith, 2004) and therefore this phase is time-bound and weak requirement analysis may lead to software failure (Hofmann & Lehner, 2001). Requirements impose constraints on both the product (the software to be developed) and the process which is used to develop the software. Hence, it is a complex task and requires a lot of effort in making decisions on various points like redundancy, non-redundancy of the requirements, preferences of the requirements, selection of right and valuable requirements. One solution to cope with these complex decisions is requirement prioritization.

According to Hoffman and Lehner (Hofmann & Lehner, 2001), requirement prioritization is one of the most crucial part of the requirement engineering phase. The most important aspect in requirement prioritization is to decide which requirement is more desirable than the other. And the challenging task in prioritizing requirements is the phrase “prioritizing requirements” itself as different stakeholders have different meanings of the word “priority” and same is true from the prospective of developers. For any software project there are hundreds of individual requirements and it is likely that stakeholder’s
desires to include all to the software project. But the reality is, that all features or requirements are not really essential and many stakeholders will not agree to which requirements are important and which are not. For the success of software system both stakeholders and developers must collaborate on requirements prioritization. Thus, the decision of selecting the requirement for inclusion in the project is cumbersome and crucial which is governed by various aspects related to requirement. One such aspect is importance. This aspect is extremely multifaceted since it depends on the personal perspectives of both stakeholder(s) and developers. Prioritizing requirements based on importance could be based on urgency, usefulness, importance to business value, importance from the point of view of product architecture, cost of implementation, time, relationships and dependencies among requirements, importance due to platform on which it is to be implemented, because of team members skills, risk associated with implementation, etc. Hence it is likely that there will be different ranking for same requirement by both stakeholders and developers, as both have different point of view on same requirement. Another aspect is limitations in project resources, schedules and overall requirement analysis which are generally handled by project manager while estimating risk. In addition to the above factors budget, time, staff, large number of requirements, requirement evolution, requirement incapability and most importantly stakeholder and developer collaboration pose constraints while prioritization. Hence, it becomes a responsibility of developers and stakeholders to consider all these factors into consideration while selecting and prioritization requirements.

The success of any project is highly dependent upon how well the requirements have been prioritized (Lauesen, 2002). While prioritizing if a single aspect is to be taken into consideration then it becomes easier to take decision but if more than one aspect is to be considered then the decision becomes difficult because a wrong decision can result in extra cost to the organization and the impact of same can be manifold (Ruhe, Eberlein & Pfahl, 2003) as it may impact other requirements, their relationship values and dependencies (Perini, Susi & Avesani, 2013). Therefore, it is very essential while prioritization to estimate the impact of selecting requirement depending upon specific constraints and conditions suitable for project environment. If we look at these aspects and constraints closely it can be concluded that large number of aspects are directly depending upon the understanding of the word “priority” by different stakeholders and developers from their own perspective. If we could map this difference in assigning priority of individual requirements by both stakeholders and developers we will have to face fewer challenges during the course of development. Therefore, requirements prioritization is a collaborative task wherein both stakeholders and developers in a software project work together with the goal to achieve a common opinion regarding the prioritization of a given set of requirements (Ruhe, Eberlein & Pfahl, 2003). Developers do not always know which requirements are most important to the customers and customers cannot judge the cost and technical difficulty associated with specific requirements. Hence, there lies a gap between stakeholder and developers which should be bridged.

MOTIVATION

Communication among stakeholders is not supported by most of the prioritization techniques. This has been reported as one of the limitations of their requirement prioritization research technique by (Perini, Susi & Avesani, 2013). Many researchers have reported lack of user involvement as one the major cause of project failure (Attarzadeh, & Ow, 2008), (Gold, Gray, Jones& Miller, 2010), (Savolainen, 2011), (The Standish Group, 2009). A project is considered fail if it does not meet users’ requirements, even if it is delivered on time, within budget. Nearly more than two decades ago a survey report of Standish group (Clancy, 2008) revealed alarming figures related to project failures. According to this report the success rate was only 16.2% (developed within budget and on time), 52.7% were categorized as challenged projects due to budget overruns, missed deadlines, feature sets that did not meet user requirements, and 31.1% contributed towards cancelled or ended in failure project list. This survey also provides a top ten list of critical factors responsible for project success
Ensuring Correctness, Completeness, and Freshness for Outsourced Tree-Indexed Data

Mobile Learning and an Experience with Blended Mobile Learning
Michelle Pieri and Davide Diamantini (2009). Encyclopedia of Information Communication Technology (pp. 548-553). www.igi-global.com/chapter/mobile-learning-experience-blended-mobile/13404?camid=4v1a