

A Collaborative Effort-Benefit-Value Analysis Model to Support Requirements Reuse for Software Requirements Prioritization

Ankita Gupta, Jaypee Institute of Information Technology, India

Chetna Gupta, Jaypee Institute of Information Technology, India

ABSTRACT

This paper proposes a collaborative method to prioritize the set of requirements backing the concept of requirement re usability through crowd sourcing platform. Both stakeholder and developer perspectives are captured and effort-benefit-value analysis model is presented. Stakeholder priority is captured using Fuzzy TOPSIS model, and developer priority is captured by analyzing inter-relationships among requirements in the form of dependency. Further, a provision is provided to resolve conflicts between requirement pairs in an interactive manner to produce final ranking accepted to both with minimum conflict. An experimental proof of concept is provided by comparing results of proposed approach with state of art, fuzzy analytic hierarchy process (FAHP), interactive genetic algorithm (IGA) techniques, and on a dataset. Results of experimentation conclude that presented approach is capable of producing accurate and comparable results in addition to handling issue of dependency, multi- criteria preferences, reliability, and robustness to errors.

KEYWORDS

Collaboration, Requirement Dependency, Requirement Prioritization

1. INTRODUCTION

Requirement engineering is a crucial stage in the development of any software project and a very intricate part of this phase is Requirement Prioritization (RP). It is indispensable to complete a project without any failure if the requirements are not prioritized correctly. For success of any project it is important to confine, examine and prioritize requirements, which largely depends upon acceptance of stakeholder's requirements, concerns and criteria's (Davis, 2003; Mead, 2006; Perini et al., 2007). Most software projects have more candidate prerequisites that can be acknowledged within available budget, time constraints, stakeholder expectations and technical constraints. For effective requirement prioritization it is imperative to consolidate distinctive aspects by both stakeholders and developers to help in decision-making process. In past, a large of requirement prioritization techniques have

DOI: 10.4018/IJSI.2021010104

been proposed AHP (Saaty, 2004), Cumulative Voting (Chatzipetrou et al., 2010), Weiger's method (Wieggers, 1999), \$100-Test (Leffingwell & Widrig, 2000), Numerical Assignment (Sommerville & Sawyer, 1997), Cost Value approach (Karlsson & Ryan, 1997) etc., most of which takes the assumption that the requirements are independent of each other which may lead to erroneous priority list of the requirements.

The major factors which can influence this decision of requirement prioritization may include inter-relationships among requirements, inability to deal with complex situations, unaccepted or biased final ranking. Inter-relationship among requirements is a crucial factor which must be considered before assigning priority. There are only a few algorithms that consider the dependency amongst the requirements e.g. IGA (Tonella et al., 2013), EVOLVE (Greer & Ruhe, 2004), DRANK (Shao et al., 2017) etc. To ponder upon, there are only few algorithms that consider that the opinions of both stakeholders and developers are important for prioritizing the set of requirements, which is again a bad league to follow. The stakeholders define the real face value of the project while the developers most importantly know about the hidden interiorities of the requirements in terms of technicality, complexity and resource requirements and their availability. So, building up the right project requires the involvement of both the stakeholders and developers. Years of literature (Achimugu et al., 2014; Perini et al., 2013; Ruhe et al., 2003; Savolainen, 2011) recommend that a large portion of the prioritization approaches experiences following issues: unfit to acknowledge imperative of dependency, are time consuming, are sensitive – a solitary mistake may influence entire prioritization order, can't resolve logical inconsistency, unfit to deal with complex situations, questionable final ordering and biased ranking. Due to this, software development is still a greatest challenge for project team to produce successful software fulfilling stakeholder's scope and objective within the predicted time and cost (Kerzner, 2018). Requirement prioritization is a cost effective mechanism that plays an important role in estimating cost and time of a project. To further save cost, decrease time for product development efforts must be made in the direction of reusing potential requirements. Requirements reusability can be defined as the capability to use in project requirements that have already been used before in other projects. It helps software organization optimize resources and reduce errors. The latest requirement management tools consider requirement reusability as a part of their important functions for e.g.: DOORS, JAMA, P&PM. (Re-Magazine, n.d.). *"In software engineering particularly in requirements engineering (RE), research has shown that crowd-sourcing has the potential to aid requirements elicitation by facilitating the discovery and involvement of a wider range of crowd contributors and users"* (Hosseini et al., 2014). To efficiently capture requirements an emerging business model of crowd-sourcing is used in this study.

This research focuses on providing a collaborative decision support to select right ordering of requirements out of a given superset of candidate requirements by considering effort, benefit and value of a particular requirement. This paper first aims to provide support of requirement reusability through crowd-sourcing platform where the cloud sourcer's will define, broadcast, analyze and make selection for reusable requirements. Selection of reusable requirements will depend upon closeness towards required objective and granularity of tasks that are accomplished by individual contributors. Lastly, we define a scalable Effort-Benefit-Value Analysis Model which analyses and visualizes data and computes overall effort required to implement a particular requirement on various parameters. Finally ranks are generated by resolving conflicts of stakeholder's and developer's initial ranking with effort in an interactive manner to produce final priority values. To capture stakeholder's priority a support for MCMD is provided using Fuzzy TOPSIS (Wang et al., 2009) model and to capture developers priority inter-relationships among requirements in the form of dependency are analysed. With this aim, this research tries to overcome the limitations in existing approaches such as considering effective user knowledge though collaboration, analysis of requirement dependencies, handling issue of multi- criteria preferences thereby assuring correct, effective and comparable final ranking to support reliability and robustness to errors. The whole process is tested for its accuracy on different requirement sets as a part of experimentation. For validation we have compared our results

13 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage: www.igi-global.com/article/a-collaborative-effort-benefit-value-analysis-model-to-support-requirements-reuse-for-software-requirements-prioritization/266281

Related Content

I-Way: A Cloud-Based Recommendation System for Software Requirement Reusability

Chetna Gupta, Surbhi Singhal and Astha Kumari (2020). *Crowdsourcing and Probabilistic Decision-Making in Software Engineering: Emerging Research and Opportunities* (pp. 23-34).

www.irma-international.org/chapter/i-way/235760

Trends in Information Security

Partha Chakraborty and Krishnamurthy Raghuraman (2013). *Software Development Techniques for Constructive Information Systems Design* (pp. 354-376).

www.irma-international.org/chapter/trends-information-security/75757

Semi-Automatic Annotation of Natural Language Vulnerability Reports

Yan Wu, Robin Gandhi and Harvey Siy (2013). *International Journal of Secure Software Engineering* (pp. 18-41).

www.irma-international.org/article/semi-automatic-annotation-of-natural-language-vulnerability-reports/83633

Autonomic Business-Driven Dynamic Adaptation of Service-Oriented Systems and the WSPolicy4MASC Support for Such Adaptation

Vladimir Tasic (2010). *International Journal of Systems and Service-Oriented Engineering* (pp. 79-95).

www.irma-international.org/article/autonomic-business-driven-dynamic-adaptation/39100

Fault-Tolerant Software: Basic Concepts and Terminology

Vincenzo De Florio (2009). *Application-Layer Fault-Tolerance Protocols* (pp. 21-52).

www.irma-international.org/chapter/fault-tolerant-software/5122