Information and Software Technology 65 (2015) 56–68

Contents lists available at ScienceDirect

Information and Software Technology

journal homepage: www.elsevier.com/locate/infsof

Investigating the penalty reward calculus of software users and its impact on requirements prioritization

Adarsh Kumar Kakar *

Alabama State University, Montgomery, AL 36101, United States

ARTICLE INFO

Article history:
Received 4 December 2014
Received in revised form 26 March 2015
Accepted 11 April 2015
Available online 22 April 2015

Keywords:
User satisfaction
Requirements prioritization
Multiplicative effect
Asymmetric effect
Penalty reward contrast analysis

ABSTRACT

Context: The current requirements engineering techniques for prioritization of software requirements implicitly assume that each user requirement will have an independent and symmetric impact on user satisfaction. For example, it is assumed that implementing a high priority user requirement will positively impact his satisfaction and not implementing a high priority user requirement will negatively impact his satisfaction. Further, the impacts of implementing multiple user requirements on his satisfaction are expected to be additive. But is this always the case?
Objective: This paper empirically examines whether the assumptions of symmetric and multiplicative impacts of user requirements on his satisfaction are valid. Further, the study assesses the relative efficacy of 5 methods of requirements prioritization in managing these effects as reflected by the user satisfaction with the prioritized requirement sets.
Method: To test for existence and mitigation of asymmetric effects an adaptation of the widely accepted PRCA (Penalty Reward Contrast Analysis) method was used for 5 requirements prioritization techniques. To test for existence and mitigation of multiplicative effects MHMR (Moderated Hierarchical Multiple Regression) a well-accepted technique for testing interaction effects was used.
Results: Both asymmetric and multiplicative effects of software requirements on user satisfaction were observed for requirements prioritized using all 5 requirements prioritization methods raising questions about the efficacy of present day requirements prioritization techniques. Further, the results of the experiment led to proposing a new method for requirements prioritization for managing these effects.
Conclusion: The study empirically demonstrates the complexities of prioritizing software requirements and calls for a new generation of methods to address them. Understanding and resolving these complexities will enable software providers to conserve resources by enabling them to parsimoniously selecting only those requirements for implementation in the software product that have maximum incremental impact on user satisfaction.

1. Introduction

During software product evolution, one of the main objectives is to continually enhance the features of the product to make it more attractive and valuable for the user. It is therefore important for the producers to be able to distinguish between those user requirements, expressed in the form of feature requests, which add value for the maximum number of users and those user requirements that do not. Introducing non-value adding requirements has detrimental impact for both users and producers of software products (Table 1).

If user requirements had no interdependencies then the present requirements prioritization methods which focus on individual user requirements and prioritize them based on their relative ranking or grouping on specified criterion or criteria may suffice. Requirements prioritization from user standpoint would involve selecting the top 'n' user requirements from a given set of user requirements which users perceive are most important to them. But user requirements are not stand alone artifacts [22]. They may exhibit complex interdependencies among each other and therefore cannot be treated independently [57,16]. Ruhe [59] noted that overall satisfaction with requirements in a requirements subset may not be additive. They may exhibit both synergistic and antagonistic impacts on user satisfaction.

Requirements prioritization must therefore include approaches for managing requirements interdependencies and multiplicative effects to fully support providers of software products [40]. To complicate matters further non-software product development literature [38,3,14,37,46] has found that user requirements are

* Tel.: +1 334 229 6804.
E-mail address: akakar@alasu.edu

http://dx.doi.org/10.1016/j.infsof.2015.04.004
0950-5849/

Table 1
Impacts of adding non-valued product features.

User	Producer
Users have to expend resources in terms of memory and computing power for running additional features that add no value to their work [7]	Producers have to utilize their scarce resources in building features that have no positive business outcomes as customers do not fund upgrades of market-driven products [39]
Overloading the product with features causes "feature fatigue" i.e. the more features a product boasts, the harder it is to use [66]	Building new features makes the product complex and more difficult to maintain [49]
May degrade quality and make products unreliable [49]	Increases time-to-market as even providing features that do not add value to the user requires additional time to implement

known to demonstrate asymmetric effects on user satisfaction. For example, a user requirement may impact user satisfaction significantly and negatively when not implemented in a product but may not impact user satisfaction significantly and positively when implemented into the product and vice versa.

Keeping this context in view we first empirically investigate using 5 requirements prioritization techniques whether user requirements exhibit interactive and asymmetric impact on user satisfaction occurs for software products. Currently there is a gap in software development literature. To the best of our knowledge, an in-depth investigation into these phenomena has never been conducted. This investigation is important. If validated these phenomena may call into question the efficacy of present day software requirements prioritization techniques in accurately identifying user requirements to meet user satisfaction goals. The present day requirements prioritization techniques treat each user requirement as an independent artifact and assume symmetric impacts of implementation and non-implementation of a requirement on user satisfaction. Secondly, we assess the relative efficacy of 5 methods of requirements prioritization in managing these effects as reflected by the user satisfaction with the prioritized requirement sets. Further, based on the analyses of the results of the experiment, a new method for requirements prioritization is proposed.

2. Literature review

2.1. User satisfaction

User Satisfaction is one of the most prevalent and enduring measures of software success and use [36,67,23,61,75]. The concept of IS user satisfaction can be traced to the work of Cyert and March [21] who proposed that an information system which met the needs of its users would reinforce satisfaction with the system. In the early 1970s, Powers and Dickson [55] studied factors affecting IS success, and identified user satisfaction as one of the key factors affecting it. They assumed that if users are satisfied with an IS, they use it. Therefore, satisfaction is a good measure of IS success. If the users do not perceive a system as satisfactory, they are unlikely to use it. Thus, in order to improve a system, it is important to know how its users perceive it, and where its weak points lie.

The reason for the popularity of user satisfaction as a measure of software success is the difficulty of operationalizing economics-based constructs, thus accelerating the search for constructs for which variables could be identified and more easily measured (e.g., [55,53,68,27]). Gelderman [33] found that user satisfaction was significantly related to system performance factors "providing empirical evidence for the popular assumption that user satisfaction is the most appropriate measure for IS success available".

As "the set of requirements selected for implementation is a primary determinant of customer satisfaction" [41], user satisfaction was chosen as a dependent variable in this study. We define user satisfaction as the user evaluation of the degree to which his requirements are met by the software. Further, given the premise that software is developed with the goal of satisfying user needs [2], the challenge for the product managers is to distinguish between which software requirements add value to the user of an evolving software product and which requirements do not [41]. Customer satisfaction is strongly related to the concept of value (Woodruff, 1997) and is considered a measure of value provided by the software product [15].

Therefore, in line with literature which considers customer value as an antecedent to, and a primary and direct link of, customer satisfaction [17,31,12,29,5] we conceptualize Value adding (VA) requirements as those that affect user satisfaction significantly and positively. Non-Value adding (NVA) requirements are those that do not significantly impact user satisfaction or impact user satisfaction negatively when implemented into the software product. The efficacy of requirements prioritization methods therefore lies in identifying a VA requirements set which maximizes user satisfaction and a NVA set of requirements that have non-significant (or negative) impact on user satisfaction.

2.2. Exploration of requirements prioritization methods

This section continues by providing reviews of requirements prioritization methods from requirements engineering literature. Further, requirements prioritization for market-driven software products face special challenges compared software developed for single customer or for internal use. Developers of market-driven software products have to deal with anonymous users, requirements overload due to large number of new feature requests, time-to-market pressures and lack of day to day interaction and negotiation with the user base making it imperative to evaluate requirements prioritization methods beyond those assessed in requirement engineering literature [39]. Therefore this section also reviews non-software product development and product quality literatures to identify promising methods of prioritization of software product requirements. The objective is to determine if the phenomenon of multiplicative and asymmetric effects are observed for user requirements across methods of requirements prioritization.

2.2.1. Requirements prioritization methods from requirements engineering literature

Several techniques have been used for prioritization of requirements. Table 2 adapted from [10] provides often cited examples of requirement engineering techniques including the Grouping methods such as Priority Groups method and the non-grouping (ranking) methods such Planning Game method, 100 points method, Priority Groups method, Theory W method, AHP method, Binary Search Tree method and Value-Oriented Prioritization method.

2.2.2. Other requirements prioritization methods

Table 3 provides a list of techniques from Quality and Product Development Literature. The basis for many of these techniques can be traced to the three factor model [38] with the following definition for the three factors:

Basic factors: These requirements are prerequisites and must be satisfied first, at least at threshold levels, for the product to be accepted. The customer takes Basic requirements for granted, and therefore does not explicitly ask for them. Basic requirements are critical when they are not met, but users remain Indifferent if they are provided for in the product.