Separating the Wheat from the Chaff: Extracting Business Value from Feature Requests Posted in User Forums

Adarsh Kumar Kakar, Alabama State University, Montgomery, AL, USA

ABSTRACT

When making a decision to add features to an existing software product in response to feature requests posted in user forums, it is important to pursue only those changes that deliver value to both the user and the producer. But selecting critical user requirements expressed as features requests is a challenging task. While excluding a high value requirement may mean losing customers to a competing product, including a requirement that is unneeded increases time to market and introduces unnecessary costs and complexity in the product. Keeping these issues in focus, promising methods of feature selection were first identified from a review of requirements engineering, product development and quality literatures. An empirical study was then conducted to investigate the efficacy of methods in separating the vital few user feature requests from the trivial many posted in user forums without adversely impacting user satisfaction. The result of the empirical study demonstrates that the Kano survey method has potential in separating the wheat from the chaff. The reasons for this finding is empirically investigated and discussed.

KEYWORDS

Business Value, Feature Requests, Feature Selection, Kano Survey Method, User Forums

INTRODUCTION

Involving users as active contributors in the product development process rather than as passive participants has been expressed in literature for a long time (Gardiner and Rothwell, 1985; Leonard-Barton, 1995; Rothwell, 1976; von Hippel, 1988; Witell, Lofgren and Gustaffsson, 2011). Collectively users constitute a source of massive amounts of product innovation (Vonn Hippel, Ogawa and de Jong, 2012). When users are viewed merely as recipients of innovation, the firm does not have access to user knowledge and experience developed through product use (Sawhney, Verona and Prandelli, 2005). Users are co-creator of value (Vargo and Lusch, 2006) and useful partners in the innovation process (Kristensson et al; 2004). Magnusson et al. (2003) found that ordinary users were the best source for new ideas in terms or originality. Through their unique knowledge derived from use of the product ordinary users were better equipped to generate ideas that are relevant to customers than R and D (Research and Development) employees and engineers (Kristensson et al, 2004; Magnusson, 2003).

It has been suggested that the product evolution should be innovative in the users’ frame of mind not the developers’ (Fellows and Hooks, 1998). New product features that do not resonate with the users create wasted development effort, delay in time-to-market, increased complexity and operational costs of the product. Therefore to actively engage users organizations have evolved various mechanisms. Of these the use of websites for capturing and prioritizing user requirements is becoming increasingly prevalent. The websites include both forums and collaborative tools, and are
designed to allow large numbers of users to participate in the requirements gathering and analysis process. The success of websites and collaborative tools that have been used to gather inputs from users, demonstrates that, given the opportunity, users too are willing to take the time to contribute feedback and ideas (Laurent and Cleland-Huang, 2009).

However, by actively engaging the users, more new feature requests are often elicited than are needed to build into the system. It is therefore as critical for organizations to identify non-value adding (or marginal value adding) feature requests as it is for identifying feature requests that add value to the users. While on the one hand excluding a high value feature may mean losing users to a competing product, on the other hand including a requirement that is unneeded creates wasted development effort, delays in time-to-market, and increased complexity, maintenance and operational costs of the product. Implementing non-value adding features has negative impacts for both producers and users of software products (Table 1). So how do producers of software products accurately identify from a given set of user feature requests only those features that add significant value to the users of the software product?

Keeping this context in view, this study identifies 5 promising methods for feature selection from a review of requirement engineering, product development and quality literatures – Binary search tree, Priority groups, Kano survey method, Dual questioning method and Hybrid method. The objective is to assess which of these methods demonstrate greater efficacy in engaging users for identifying features which add value to the users of information systems (IS) product as well as features which do not.

LITERATURE REVIEW

Feature Selection Methods

This section reviews feature selection methods from software development as well as non-software product development literatures. Compared to software developed for single customer or for internal use, feature selection for market-driven software products involves addressing special challenges such as anonymous users, lack of day to day negotiations and interactions with the user base, requirements overload due to large number of feature requests and time-to-market pressures (Karlsson et al., 2007). We therefore include in our evaluation promising feature selection methods from quality and product management literatures in addition to those from software requirements engineering.

Methods from Requirements Engineering Literature

The commonly cited feature selection methods from requirement engineering literature are listed in Table 2 (adapted from Berander and Andrews, 2006).

<table>
<thead>
<tr>
<th>User</th>
<th>Producer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requires investment in additional computing resources such as memory and computing power for non-value adding software features (Basili and Boehm, 2001)</td>
<td>Requires investing in features that have no positive impact on business outcomes as upgrades of market-driven products are not funded by customers (Karlsson et al., 2007)</td>
</tr>
<tr>
<td>Causes “Feature fatigue” (Thompson, Hamilton and Rust, 2005)</td>
<td>Makes the software more difficult to maintain (Mens et al., 2005)</td>
</tr>
<tr>
<td>Increases software complexity with deleterious effects on product quality and reliability (Mens et al., 2005)</td>
<td>Increases time to market for software releases</td>
</tr>
</tbody>
</table>
Related Content

**UDOO App Inventor: Introducing Novices to the Internet of Things**
[www.igi-global.com/article/udoo-app-inventor/160365?camid=4v1a](www.igi-global.com/article/udoo-app-inventor/160365?camid=4v1a)

**Adaptive Presentation and Scheduling of Media Streams on Parallel Storage Servers**
[www.igi-global.com/chapter/adaptive-presentation-scheduling-media-streams/24478?camid=4v1a](www.igi-global.com/chapter/adaptive-presentation-scheduling-media-streams/24478?camid=4v1a)
The Human Side of Information Systems Development: A Case of an Intervention at a British Visitor Attraction
Brian Lehaney, Steve Clarke, Sarah Spencer-Matthews and Vikki Kimberlee (2002).
*Advanced Topics in End User Computing, Volume 1* (pp. 116-127).
www.igi-global.com/chapter/human-side-information-systems-development/4428?camid=4v1a

An Experimental Investigation of Comparative Operating Efficiency of a 4GL (Focus) and COBOL in a Workstation Environment
www.igi-global.com/article/experimental-investigation-comparative-operating-efficiency/55658?camid=4v1a