Association for Information Systems
AIS Electronic Library (AISeL)

International Research Workshop on IT Project Management 2013

International Research Workshop on IT Project Management (IRWITPM)

12-14-2013

Lean Software Development: Evaluating Techniques for Parsimonious Feature Selection of Evolving Information Systems Products

Adarsh Kumar Kakar
Alabama State University, akakar@alasu.edu

Joanne E. Hale
Alabama State University, jhale@cba.ua.edu

David P. Hale
Alabama State University, dhale@cba.ua.edu

Lean Software Development: Evaluating Techniques for Parsimonious Feature Selection of Evolving Information Systems Products

Adarsh Kumar Kakar
Alabama State University
akakar@alasu.edu

Joanne E. Hale
Alabama State University
jhale@cba.ua.edu

David P. Hale
Alabama State University
dhale@cba.ua.edu

ABSTRACT

Lean software development is a product development paradigm with focus on creating value for the customer and eliminating waste from all phases of the development life cycle. Applying lean principles, empirical studies were conducted focusing on identifying and assessing methods that parsimoniously select features from a given set of user feature requests. The results of the studies show that the Kano survey method has potential. It demonstrated efficacy in not only identifying the feature subset, from a given set of feature requests, that maximizes value to the users but also in eliminating waste by identifying the subset of features which does not provide significant value to the users when implemented into the software product. The design and results of one study is elaborated in this article. The findings obtained in the study have useful implications for practice and opens up new avenues of research for evolving market-driven software products.

Keywords

Lean software development, feature selection, user satisfaction

INTRODUCTION

Although the lean concept has its origins in manufacturing, specifically the Toyota production system, the principles of Lean manufacturing can be used as a framework, or a guideline to address issues of software development (Kumar, 2009). Applying lean principles to selecting which features should be implemented in a software product can result in tremendous downstream benefits such as reduced cost and complexity of the product.

Of the seven wastes (Table 1) identified in Lean manufacturing one of the most important is avoiding waste of overproduction in the system which in the context of software development translates to adding extra (wasteful) features into the product (Kumar, 2009; Jailia et al., 2011). According to data reported at the 2000 International Software Engineering Research Network (ISERN) Workshop, individuals working alone used only 12 to 16 percent of Microsoft Word and PowerPoint features, whereas a 10-person group used 26 to 29 percent of these features (Basili and Boehm, 2001). Yet, software products must be incessantly adapted (evolved) to match any changes in the real world by adding new features because most software in regular use in businesses and organizations cannot be completely specified (Lehman and Ramil, 2002). Although these adaptations are an economic necessity they must still conform to the lean principle of not adding anything that does not create value for the customer.

Serial Number	Seven Wastes of Manufacturing	Seven Wastes of Lean Software Development

1	Overproduction	Extra Features
2	Inventory	Inventory
3	Extra Processing Steps	Extra Steps
4	Motion	Finding Information
5	Defects	Bugs not caught by tests
6	Waiting	Waiting for decisions including from customers
7	Transportation	Handoffs

Table 1.The Seven Wastes (adapted from Kumar (2009))

Adding features that do not add value to the product has adverse implications for both the user and producer of the software product as shown in Table 2.

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 (Basili and Boehm, 2001)	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 (Karlson et al., 2007)
Overloading the product with features causes "feature fatigue" i.e. the more features a product boasts, the harder it is to use (Thompson, Hamilton and Rust, 2005)	Building new features makes the product complex and more difficult to maintain (Mens et al', 2005)
May degrade quality and make products unreliable (Mens et al., 2005)	Increases time-to-market as even providing features that do not add value to the user requires additional time to implement

Table 2. Adding Non-Valued Product Features

However, selection of only those features which add value pose special challenges for market-driven software products compared with in-house development or software developed for single customer. Producers of market-driven software products have to deal with anonymous users, requirements overload due to large number of feature requests, time-to-market pressures and lack of day to day interaction and negotiation with the user base making the traditional requirement engineering techniques impractical to use (Karlsson et al., 2007). Keeping this context in view, this study reviews and identifies promising methods for feature selection from non-software product domains, such as product development and quality literatures, as well as suitable methods from requirement engineering literature for prioritization of software requirements. It then assesses which of these methods demonstrate greater efficacy in identifying features which add value to the software product and features which do not.

LITERATURE REVIEW

Lean Approach to Software Feature Selection

The first lean principle states "Specify value from the standpoint of the end customer" (Jailia et al., 2011). Software products are developed with the goal of satisfying user needs (Alves, 2003) and user satisfaction is a measure of value provided by the software product (Calisir and Calisir, 2004). It is one of the most prevalent measures of software success and use (Delone and McLean, 1992; Zviran and Erlich, 2003). We therefore chose user satisfaction as dependent variable in the study.

Further, as "the set of requirement selected for implementation is a primary determinant of customer satisfaction" (Karlsson and Ryan, 1996), the feature subsets selected by different methods from a given set of user feature requests was used as an independent variable. The efficacy of feature selection methods in identifying features that add value to the user will thus be determined by which method/s provides a feature subset that delivers maximum satisfaction to the user. The efficacy of the methods in identifying waste will be determined by which among the complementary sets (set of user feature requests – features subset that add value to the user) delivers minimum user satisfaction.

Exploration of Feature Selection Methods

This section continues by providing reviews of feature selection methods from requirements engineering literature. Further, this section reviews non-software product development and product quality literatures to identify promising methods of selection of software product features for comparison with the traditional requirements engineering techniques.

Feature Selection Methods from Requirements Engineering Literature. Several techniques have been used for prioritization of requirements. Table 3 (adapted from Berander and Andrews, 2007) 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.

Technique	Description
Priority Groups (Wiegers, 1999)	Requirements are classified into a small number (often 3) of priority categories, such as High (critical), Medium (regular), and Low (nice to have). Individual results may be aggregated by majority, plurality, or consensus.
Planning Game (Beck, 2001)	The development team sorts the requirements by value, risk, and effort. Based on the relative assessments, the scope of the next release is set.
100 Points (Leffingwell and Widrig, 2003)	Each stakeholder is given a total of 100 points that can be allocated (or "spent") on the requirements. Requirement priority is then determined by sorting the requirements by total points spent by all participants.
Theory W (Boehm and Ross, 1989)	To ensure that every stakeholder wins, each ranks the requirements and notes which are most important and which they would be willing to remove. The stakeholder groups then negotiates the prioritized list.
AHP (Saaty, 1980)	Built to address multi-criteria decision-making situations, AHP conducts a comprehensive comparison of the value and cost of each requirement pair.
B-Tree Prioritize (Heger, 2004)	Uses an algorithm for arriving at the priority list of requirements from a given candidate set of requirements by economizing on the number of comparisons.
Value-Oriented Prioritization (Azar, Smith and Cordes, 2007)	In this method after identifying the core business value categories, company executives rank each value on a relative scale. Thereafter all requirements are identified a weight in each value category and a ranked list of requirements is generated.

Table 3: Methods from Requirements Engineering Literature

Other Feature Selection Methods. Table 4 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 (Kano, 1984) with the following definition for the three factors:

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

Performance factors: are product features that the customer deliberately seeks to fulfill. They are uppermost in her consciousness. Fulfilling these requirements leads to customer satisfaction and not fulfilling them leads to dissatisfaction.

Excitement factors: are those product features that the customer did not expect. They surprise the consumer by adding unexpected value to the product thereby delighting her. Not fulfilling excitement requirements do not lead to consumer dissatisfaction.

In addition to the techniques based on the three factor theory we included a determinant-attribute approach (Myers and Alpert, 1968) which has been used for feature selection in industries as diverse as construction materials (Sinclair and Stalling, 1990) to health care systems (Lim and Zallocco, 1988). As opposed to in-house developed and software products developed for single customers, which often have restricted product choices, software product evolution exists in an open environment, in which marketplace alternatives exist. As shown in Table 3, Myers and Alpert (1968) present a means to incorporate marketplace issues such as alternative products into the Dual Questioning Technique.

Method	**Description**
Penalty-Reward Contrast Analysis (Brandt, 1987)	Classifies each product feature requirement into Basic, Performance and Excitement categories by analyzing the impact of high and low feature level satisfaction on overall product satisfaction using regression analysis with two set of binary dummy variables for each product feature.
Importance Grid (Vavra, 1997)	Classifies each product feature requirement in Basic, Performance and Excitement categories Users explicitly express preferences using 5 point Likert-like scale) and implicitly (using partial beta coefficients)
Direct Classification Method (Emery and Tian, 2002)	Classifies each product feature requirement directly into Basic, Performance and Excitement after the theory underlying this categorization is explained to the respondent
Kano Survey Method (Kano, 1984)	Classifies each product feature requirement into Basic, Performance, and Excitement categories based on two questions 1. the functional question "How do you feel if this feature is present?" and 2. the dysfunctional question "How do you feel if this feature is NOT present?". Users' response to these questions on a five point Likert-like scale
Dual Questioning Method (Myers and Alpert, 1968)	Classifies each product feature based on users explicit expression of Importance to Not Important on a 5 point scale with 5 being Extremely Important and 1 being Not Important) and Difference among compared products (4 – Very Different;, 1- Very Similar)

Table 4. Methods from Quality and Product Development Literature

Selection of Feature Selection Method

Five feature selection methods were selected for evaluation. The rationale behind the selection of these five methods for evaluation from among the methods described in the previous sections is described in the following paragraphs.

Binary Search Tree. Racheva, Daneva and Buglione (2008) reviewed a number requirements prioritization techniques and classified them into two main categories: techniques used to prioritize small number of requirements (small-scale) and techniques that scale up very well (medium-scale or large-scale). Bebensee, van de Weird and Brinkkemper (2010) observed that as software products are developed for the market rather than a single customer, one can expect a larger number of feature requests from users. Hence techniques that scale up well are most appropriate for software products. They found that the Binary Search tree method scales up well for software products with medium-scale requirements. Another study by Ahl (2005) investigating the five ranking techniques of requirements prioritization - AHP, Binary Search Tree, Planning Game, 100 Points Method and PGcAHP (Planning Game combined with AHP) - found that Binary search tree was superior to all other methods on many counts including accuracy of results and scalability. Binary search tree was therefore chosen from among the non-grouping (ranking) techniques as the first technique to be assessed in the study.

Priority Groups Method. Medium-scale or large-scale prioritization techniques might be based on relatively complex algorithms or at least due to the large amount of requirements need tool support (Rachdeva, Daneva and Buglione, 2008). However, sophisticated prioritization techniques are found to have limited ability to support requirements prioritization in market-driven product development with professionals in industry preferring simple tools instead (Lehtola and Kauppinen, 2006; Berander and Andrews, 2006). The Priority groups method is one such simple classification technique which ranks requirements into three priority categories, High, Medium and Low (Wiegers, 1999). It is an IEEE recommended method (Sillitti and Succi, 2006) and among the most traditional and best known (Lehtola and Kauppinen, 2006). Priority Groups technique was therefore chosen as the second technique for comparison.

Kano Survey Method. A review of the advantages and disadvantages of techniques for feature selection based on the three factor theory such as the Direct Classification method, Importance Grid method, Penalty-Reward contrast analysis method and Kano survey method by Mikulic and Prebez (2011) suggests that the Kano method was the most suitable. It was found to be both a valid and a reliable method for categorizing feature requests according to the three factor theory. Another study by Witell and Lofgren (2007) comparing Direct Classification method, Importance grid method, Kano survey method and a variant of the Kano survey method which used a 3 level questionnaire rather than the 5 level questionnaire of the Kano survey method (see Table 4) came to the same conclusion and recommended that practitioners continue to use the Kano survey method. For this study the Kano survey method was therefore chosen from among the various techniques based on the three factor theory as the third technique for evaluation.

Dual Questioning Method. One of the limitations of the techniques listed above is that they do not take in consideration market factors such as the availability of the features being assessed in competitive products. As this study is exploring a suitable technique for market-driven software products, it will also investigate the potential of the determinant attribute approach (Myers and Alpert, 1968) using the dual questioning technique as the fourth technique for evaluation.

Hybrid Method. In addition a fifth technique which is a combination of Dual questioning method and the Kano survey method is suggested for comparing its efficacy in feature selection. Although the three factor theory allows producers to make a strategic choice through classifying product feature requests into the three categories, it does not rank features within a category. In addition, it does not take in consideration market factors such as the availability of these features in competitive products. In the hybrid method, detailed in the experimental treatments section, the Dual questioning approach is expected to complement the Kano technique by providing a method for ranking the features within each category, keeping competition in view, after they have been categorized using the Kano method. This we expect will be relevant for producers of market-driven software products. It will provide them with additional information to select a lean set of features that give maximum user impact for the resources invested while simultaneously keeping the strategic options open for the management.

METHOD

An Experimental method was used in the study. Experimental research offers a methodical way of comparing differences in the effect of treatments (features selected using various feature selection techniques) on the dependent variable (user satisfaction).

Experimental Setting

Gmail is an exemplar of an evolving software product from Google. Since it was first introduced in April 2004, Gmail has today evolved to become a leading web based email platform by introducing innovative features. Ten randomly selected user feature requests were chosen in late October 2012 as the test instrument for this study. The pilot study used fifteen feature requests in the test instrument. But because subjects, during the debriefing session of the pilot study, gave feedback of cognitive overload, it was decided to include only 10 feature requests in the actual study. A sample feature in the set included "Allow sending emails/ replies to emails at a later time or date. Presently if the user has to send an email or a reply to email at a later date she can only save the draft and remember to send it when the date arrives." For the full set of 10 feature sets included in the study please see APPENDIX.

Subjects

A young adult (ages 19-24) cohort was used as subjects because users in this age group are recognized as early adopters of the latest technologies and responsive to innovations (new features in our experimental context) (Ehrenberg et al., 2008). Two groups of subjects participated in the experiment. In the first round 139 students participated out which 122 valid responses were obtained. The valid responses from 69 females outnumbered the valid responses from 53 male subjects. In the second round we collected data from another group of 62 subjects, of which 59 responses were found valid from 30 female and 29 male subjects. The average age of the subjects was 21.3 years with the female subjects averaging 21.3 years and the male subjects averaging 21.2 years.

Actual users of Gmail were involved in the experiment because features should be important from the users' perspective not the developer's (Fellows and Hooks, 1998). The subjects were recruited from a state university as all subjects were required to use the university Gmail. All subjects were trained on the methods of feature selection used in the experiments and their consent taken before conducting the study. The 10 Gmail feature requests were read out aloud and subject response taken on whether they have understood the user requirements.

The sample size for the experiment was determined based on the effect size found during the pilot study. The pilot study was conducted with 49 subjects who were users of Astrid Task Manager a mobile app. Assuming a power of 0.8, alpha=0.05 (one tail) and a medium effect size obtained in the pilot, a look up of Cohen's power primer (Cohen, 1992) gave the sample size. To account for mortality rate, as two rounds of experiments with a gap of one week between them were conducted in the study, and the possibility of invalid responses from the subjects, the sample size obtained from Cohen's table was inflated by 20 % to get the required sample size of 54 subjects. As an exercise of abundant caution data was obtained from 139 subjects in the first round but the analysis was restricted to 122 valid responses. The data in the second round was collected from 59 subjects and analysis restricted to 52 valid responses.

Experimental Treatments

The requirement prioritization methods used by the subjects in arriving at the critical feature subset from the set of 10 Gmail user feature requests are described below.

Binary search tree method treatment. The Binary Search Tree Method has been used previously for software product feature prioritization. It provides a ranked list of requirements according to user preference. Prioritizing software requirements using this technique involves subjects constructing a binary search tree consisting of nodes equal to the number of candidate requirements. First a single node holding one requirement is created. Then the next requirement is compared to this node. If it is of lower priority than this node then it is assigned to the left of this node else it is assigned to the right of this node. This process continues until all requirements have been inserted into the binary search tree. The node at the extreme left of the binary search tree is of the lowest priority while the node at the extreme right is of the highest priority. If the nodes in a binary search tree are traversed in *in order*, then the requirements are listed in a ranked order of priority. Thus using the binary search tree approach involved subjects selecting the requirements one at a time and creating a binary search tree and then traversing the binary search tree in order to generate a ranked list.

Priority groups method treatment. The Priority Groups Method has been used previously for software product feature prioritization. It is based on grouping requirements into different (highest to lowest) priority groups, with clear and consistent definitions of each group. Although the number of priority groups may vary the use of three groups (High, Medium and Low) is the most common (Leffingwell and Widrig, 2003). The description for these groups is as follows (Wiegers, 1999):

> **Definition: High priority requirements** are mission critical requirements; required for next release
> **Definition: Medium priority requirements** support necessary system operations; required eventually but could wait until a later release
> **Definition: Low priority requirements** are a function or quality enhancement; would be nice to have someday if resources permit

Subjects used this description to categorize each Gmail feature request into one of the three groups.

Kano survey method treatment. The Kano Survey Method involved subjects responding to two questions for the every product feature request: the functional question "How do you feel if this feature is present?" and dysfunctional question "How do you feel if this feature is NOT present?" The first question concerns the reaction of the user if the product includes that feature, the second concerns his reaction if the product does not include that feature. The user has to choose one of the five possible options for the answers for both the functional and dysfunctional question:

1. I like it this way
2. I expect it this way
3. I am neutral
4. I can live with it this way
5. I dislike it this way

Asking both functional and dysfunctional questions helps product managers assess user priorities. If the user expects some feature to be present, but can live without the feature, it is not a mandatory feature. Based on the user responses to the questions in both functional and dysfunctional form for each of the user's requirements, the quickest way to assess the questionnaires is to map each response in Table 5 and determine the requirement category to which it belongs.

Dual questioning technique treatment. In the Dual Questioning Technique consumers are:

1. asked which features they consider important and then
2. asked how they perceive this feature as differing among the competitor products

Features ranked high in rated importance (5- Extremely Important 1 – Not Important) but not thought to differ much (4 – Very Different, 1- Very Similar) among the various products may not be the most determinant factor. The product of attribute importance and difference among products determines the ranking of feature requests. Attributes that are ranked high in importance and difference ratings among products in the same product category are considered more determinant than attributes that are ranked low in importance and difference ratings among products.

Experimental Design and Procedure

Round 1. Two groups of subjects took part in the experiment. In round 1 each subject in the first group of 139 subjects provided their requirement prioritization of the 10 feature requests by users of Gmail through a paper-based instrument that included questions related to the Binary Search Tree Method, Priority Grouping Method, Kano Survey Method and Dual Questioning Technique using the methods detailed in the previous section. The data obtained from the subjects in Round 1 was used to select a subset of features that added value to the software product and a subset of features that do not. The details of how the two subsets were created for each of the five methods for comparison of their efficacy are detailed in the method of analyses section.

Functional Question	Dysfunctional Question				
	Like	Expect	Neutral	Live with	Dislike
Like	Q	E	E	E	P
Expect	R	I	I	I	B
Neutral	R	I	I	I	B
Live with	R	I	I	I	B
Dislike	R	R	R	R	Q

B-Must have requirements or Basic Features
P-Performance requirements or Expected Features
E-Excitement requirements or Augmented Features
R-Reverse, i.e. wrong features, that would make the consumer experience worse
Q-Questionable, i.e. the consumer answers is inconsistent
I-Indifferent, i.e. the consumer does not care about this feature

Table 5. Categorization of Subject Responses

Round 2. In Round 2 the data on user satisfaction with the current version of Gmail and after implementing the two feature subsets obtained from round 1 was captured from the second group of 68 subjects. Perceived user satisfaction was used as a dependent variable because the producer would want to know the impact of the feature subsets before rather than after implementing the features. Subjects rated their satisfaction for each of these experimental conditions using a single item 7 point scale (Andrews and Withey, 1976) with a neutral midpoint of 4, terrible at one end of the scale (1) and delighted at the other end of the scale (7): 1 - Terrible 2 – Unhappy 3 – Mostly Dissatisfied 4 – Neither Satisfied nor Dissatisfied 5 – Mostly Satisfied 6 – Pleased 7 – Delighted. Single-item measures offer advantages of being short, flexible and easy to administer (Pomeroy, Clark and Philip, 2001). They are also less time consuming and not monotonous to complete (Gardner et al., 1998), thus reducing response biases (Drolet and Morrison, 2001a). Hence they are appropriate for use in large scale studies (Robins, Hendin, and Trzesniewski, 2001).

Control Procedures. Control procedures were used to mitigate effects due to extraneous variables. The extraneous variable i.e. “user segment” of Gmail users was controlled through the use of a homogeneous sample of student subjects. The “sequence effect” of manipulating different treatments a counterbalancing design using Latin squares (Sheehe and Bross, 1961) was used to get subject responses for different methods of feature selection. Every fifth subject got the same sequence (see Table 6).

The “individual differences” among subjects in the sample was controlled by using the repeated measure design. The measurement of dependent variable (user satisfaction) was repeated as subjects rated their responses on “user satisfaction” for each of the five methods of prioritization i.e. Priority Group, Kano survey method, Binary Search tree, Dual method and Hybrid method. All the feature requests in the survey instrument were randomly selected from actual pending feature requests of users of Gmail. They were re-worded in a simple and standard style to avoid bias. Shifts in structure, content and format may introduce unwanted sources of variability that may confound subject response.

Round 1: Feature Selection Method				
Subject 1	Priority groups	Kano	Binary Tree	Dual
Subject 2	Kano	Dual	Priority groups	Binary tree
Subject 3	Dual	Binary Tree	Kano	Priority Groups
Subject 4	Binary Tree	Priority Groups	Dual	Kano

Table 6. Sequencing of Methods

Method of Analyses. Comparison of feature selection techniques that are structurally different from each other requires that care be taken during analysis. For example, the Kano survey method and the Priority groups method both group requirement into three different categories. But each of these methods has different categories and has a different basis for categorization. To compound the problem the binary search tree method and the dual questioning method do not group requirements into categories but produce a ranked order list of requirements with no clear direction on the cut-off point for either for selection of requirements or for the exclusion of requirements to be built into the product.

The Kano survey method identifies both a subset of features that add value to the product (Basic + Performance + Excitement features) and those that do not add value to the product (Indifferent features). Kano survey method was therefore used as the baseline. For instance, if the Kano survey method identifies ‘n’ features that are likely to add value to the product, then in the complementary subset there are ’10-n’ features that are not likely to add value to the product. The top ranked ‘n’ requirements identified by the Priority groups method, Binary Search tree method and Dual method were then chosen for comparison of efficacy in identifying features that provide value and bottom ’10-n’ features were chosen for comparison of efficacy in identifying waste. For the ranking methods such as Binary Search tree and Dual questioning method the set of ‘n’ value-add and ’10-n’ non-value added feature sets could be easily derived from the rank order. For Priority groups method the ranking order was determined by High > Medium > Low and within each category by the descending order of the number of users who selected the features in that category. The number of features that add value for the Hybrid method was determined by considering only the common features identified by Kano survey and Dual questioning methods.

Aligned with the lean principles of focusing on value creation for the customer and elimination of waste, the top ‘n’ ranked features from the 5 methods would be compared for determining the efficacy of the methods in identifying the feature subset from a give set of feature requests that creates value for the users. The bottom 10-n’ features would be used to compare the efficacy of the methods in identifying waste. While one can deduce that the method/s which demonstrate greater efficacy in identifying a subset of features that create value for the user would also be the method/s that demonstrate greater efficacy in identifying the complementary subset of features that are wasteful, this may not be always the case. This is because IS product features exhibit complex inter-dependencies among themselves. The overall satisfaction with features in the feature subset may not be additive (Ruhe, 2005). Although a feature selection method in comparison with other feature selection methods may be able to identify a feature subset that provides higher perceived user satisfaction, its complementary feature subset may not necessarily generate lower perceived user satisfaction. Thus only methods which demonstrate efficacy in identifying both value added and non-valued feature subsets are likely to show promise for lean software development.

Repeated measure ANOVA was used to test the difference in user satisfaction as the same subjects take part in all conditions of the experiment in the second round. Feature selection technique was the independent variable and user satisfaction was the dependent variable. The measurement of dependent variable (user satisfaction) was repeated as each subject rated her responses on “user satisfaction” for all the five methods of prioritization i.e. Priority Group, Kano survey method, Binary Search tree, Dual method and Hybrid method. Using a standard ANOVA in this case is not appropriate because it fails to model the correlation between the repeated measures as the data violate the ANOVA assumption of independence. IBM© SPSS© Statistics Version 19 was used to run repeated measures ANOVA. ANOVA is robust against violations of normality but requires that the variances for each set of different scores and their covariances are equal. Violations of this assumption of sphericity can invalidate the analysis. The Mauchly’s (1940) sphericity test was therefore conducted to evaluate sphericity.

RESULTS AND ANALYSES

Feature Number	Basic	Performance	Excitement	Indifferent	Questionable + Reverse
1	**56**	23	31	10	2+0
2	13	**68**	8	22	7+4
3	12	41	25	**44**	0+0
4	**81**	12	7	19	2+1
5	21	31	9	**59**	1+1
6	33	20	15	**52**	0+2
7	37	18	**44**	35	0+2
8	11	24	32	**54**	0+0
9	18	**71**	14	9	0+0
10	29	17	**61**	15	0+0

Table 7. Consolidation of subject responses from Kano survey

	Binary Search Tree Method		Dual Questioning Method		Priority Groups Method		
Feature Number	**Ranks 1-6**	**Ranks 7-10**	**Ranks 1-6**	**Ranks 7-10**	**High**	**Medium**	**Low**
1	36	**86**	**75**	47	**56**	35	31
2	**82**	40	**69**	53	48	8	**68**
3	**71**	51	**81**	41	**81**	34	7
4	43	**79**	**90**	32	32	29	**61**
5	**77**	45	29	**93**	29	**76**	17
6	31	**91**	**66**	56	44	24	**54**
7	**83**	39	37	**85**	**60**	18	44
8	55	**67**	21	**101**	21	31	**70**
9	**88**	34	51	**71**	18	**71**	14
10	**63**	59	**73**	49	20	**87**	15

Table 8. Consolidation of subject responses from Other methods

Analyzing the subject responses obtained from round 1, we find that the Kano survey method (Table 7) identified (in bold) 6 features (Basic + Performance + Excitement features) that provided value to the user and 4 features which the users were indifferent to. The value added and non-value added features for other methods were determined by further analyzing the data obtained (Table 8) using the process detailed in the Method of Analyses subsection. The results of round 1 of the study are summarized in Table 9. As can be seen from Table 9 each method identifies a unique subset of features that are likely to add value to the product (features in black) and a unique subset of features that are not likely to add value to the product (in white). For a full description of each feature see APPENDIX.

Feature Number	Kano	Priority Group	Dual	Binary Search Tree	Hybrid
1	■	■	■		■
2	■		■	■	■
3		■	■	■	
4	■		■		■
5		■		■	
6			■		
7	■	■		■	
8					
9	■	■		■	
10	■	■	■	■	■

Table 9. Summary of Features selected

The descriptive statistics of the mean user satisfaction (V=value added features, NV=non-value adding features)) under different experimental treatments is shown in Table 10.

EXPERIMENTAL CONDITIONS	Mean User Satisfaction (NV)	Mean User Satisfaction (V)
Current version (1)	4.541	4.541
Kano Survey Method (2)	4.655	4.984
Priority Groups Method (3)	4.721	4.679
Dual method (4)	4.789	4.936
Binary Search tree method (5)	4.749	4.656
Hybrid method (6)	4.771	4.682

Table 10. Descriptive Statistics

After analysis of data obtained from the first group of subjects in round 1, we collected data from the second group of subjects in round 2. Each subject rated their satisfaction with the 6 features, obtained using the 5 different methods, that added value to the user and 4 features that did not add value to the user.

To determine if there is a significant difference between satisfaction with the various feature subsets that add value a repeated measure ANOVA was performed. Mauchly's (1940) sphericity test was conducted to evaluate sphericity. The data violated the assumption of sphericity as the probability of Mauchly's test statistic (p=0.000) was less than 0.05. Applying the Greenhouse-Geisser (1959) correction it was found that the difference in mean scores of user satisfaction with Gmail under the experimental conditions (treatments) for subsets that add value to the Gmail are statistically significant (p=0.001). However, although an overall significant difference in means was observed one does not know where those differences occurred. The Bonferroni post-hoc test results summarized in Table11 were therefore examined (row - column) to discover which specific means differed significantly.

	1	**2**	**3**	**4**	**5**	**6**
1	0					
2	0.443**	0				
3	0.138	-0.306**	0			
4	0.395**	-0.048	0.257*	0		
5	0.115	-0.328**	-0.022	-0.280*	0	
6	0.141	-0.302*	0.003	-0.254*	0.026	0

p < .05 ** P < .01 ***p<.001

Table11. Difference in User Satisfaction (V)

To determine if there is a significant difference between satisfaction with the various feature subsets that did not add value to the software product a repeated measure ANOVA was performed again.

	1	**2**	**3**	**4**	**5**	**6**
1	0					
2	0.114	0				
3	0.180	0.066	0			
4	0.248*	0.134	0.068	0		
5	0.208*	0.094	0.028	-0.040	0	
6	0.230*	0.116	0.050	-0.018	0.022	0

*** p < .05 ** P < .01 ***p<.001**

Table 12. Difference in User Satisfaction (NV)

Mauchly's (1940) sphericity test was conducted to evaluate sphericity. The data did not violate the assumption of sphericity as the probability of Mauchly's test statistic (p=0.45) was greater than 0.05. Assuming sphericity it was found that the difference in mean scores of user satisfaction with Gmail under the different experimental conditions (treatments) for subsets that add value to Gmail are statistically significant (p=0.000). However, although an overall significant difference in means was observed one does not know where those differences occurred. The Bonferroni post-hoc test results summarized in Table 12 were therefore examined (row - column) to discover which specific means differed significantly.

DISCUSSION AND INTERPRETATION OF RESULTS

Looking at the column values representing Current version and titled '1' in both Table 11 we see that the Kano survey method (row 2) demonstrated superior efficacy (significantly higher perceived user satisfaction) in identifying a feature subset that add value to the users of the software product compared to all other methods expect the Dual questioning method. Also among the five methods only the feature subsets identified by Kano survey method and Priority groups method demonstrated efficacy in identifying non-value added feature subsets that did not impact user satisfaction significantly (see column 1, Table 12)..

Thus the results show that overall the Kano survey method demonstrates promise for lean software development. While the Dual method performed better statistically than the Binary Search tree method, Priority groups method and the Hybrid method in identifying the features that add value to Gmail, it did not demonstrate efficacy in identifying features that do not add value to Gmail. Also, while the Priority groups method performed better statistically than the Dual method, Binary Search tree method and the Hybrid method in identifying the features that do not add value to Gmail, it did not demonstrate efficacy in identifying features that add value to Gmail. Only the Kano method demonstrated efficacy in identifying features that add value to the IS product as well as in identifying features that did not add value to the IS product.

The proposed Hybrid technique using the common feature sets derived from Kano survey method and Dual method showed mixed results. On the negative side, in spite of having to ask four questions to the user for each feature, the hybrid method demonstrated lower efficacy in identifying a feature set that provides maximum value to the user compared to the Kano survey method. On the positive side the subjects using the Hybrid method on an average identified a significantly ($p=0.000$) lower number of features (3.41) that add value to the software product compared to all the other methods (4.35) in Round 1. On the aggregate level the comparison for the mean user satisfaction by subjects in Round 2 was made for 4 value added features for the Hybrid method compared with 6 value added features for all the other methods (i.e. 22 .22 % lesser features) and yet there was no significant difference in the mean satisfaction feature set identified by the Hybrid method compared to the feature sets identified by the Dual the Priority groups and the Binary search tree methods.

CONTRIBUTION

Feature selection is a critical process that impacts development expenses and software products' market potential, yet it is a vexing issue for a software development organization. Choosing the appropriate set of next-release requirements from a larger set of candidate requirements has consequences for the customer and the development organization. For the customer of a market-driven software product, the appropriate set of selected requirements must deliver the expected functionality of the application domain and differentiate the product meaningfully from its competitors. For the development organization, the appropriate set of requirements must meet the customer's desired functional expectations and minimize the resource outlay.

In response to calls in literature this study explored techniques not common in the field of requirements engineering and compared them with some of the commonly used techniques for software products. Developers of market-driven products face special challenges and unlike in-house development, the customer of market-driven software products does not fund the product upgrade. Thus lean development principles are appropriate as their focus is to reduce waste during software development (Jailia et al., 2011). In line with lean principles, this pioneering study assessed the efficacy of various methods for reducing waste in software development due to adding extra features which do not add value to the users of the software product. Although a number of research studies have compared the efficacy of methods of requirements prioritization based on various criteria, none of them have investigated their efficacy in reducing waste from the view point of the users.

The results of this empirical study corroborate observations made in research literature that traditional requirements engineering techniques may not be best suited for market-driven products such as the software products. The Kano survey method from product quality literature demonstrated potential in accurately identifying those features that users value as well as those that users do not care about. This has implications for producers of software products.

Producers can evaluate the incremental impact of adding new features on user satisfaction. In addition, the software organization is freed from pursuing "maximum requirements coverage" to being empowered with information allowing it to meet the customer expectations while at the same time conserving organizational resources.

LIMITATIONS AND FUTURE RESEARCH

The subjects chosen for the empirical study were youth between 19-24 years of age. The rationale was to get as homogenous a group of sample as possible as the objective of the study was to control extraneous variables such as segmental difference in user preferences and mitigate alternative explanations for the results obtained. These design choices may limit the generalizability of the findings of this study. To assess the generalizability of results obtained in this study, the study may be replicated for other user segments and other software products. In addition, this study considered a mature and established software product such as Gmail. Future studies may replicate this study for software products in other stages of their life cycle, such as Introductory, Growth and Decline stages. Future researchers may also investigate the potential of the Hybrid technique in identifying critical user requirements under severe resource constraint conditions.

REFERENCES

Ahl, V. (2005) An Experimental Comparison of Five Prioritization Methods, *Master's Thesis, School of Engineering*, Blekinge Institute of Technology, Suecia.

Alves, C. (2003) COTS-based requirements engineering, in: Component-based Software Quality, *LNCS*, Springer, Berlin, Heidelberg, 2693, 21-39.

Andrews, F. M. and Withey, S. B. (1976) Social indicators of well-being, New York: Plenum Press.

Azar, J., Smith, R. and Cordes, R. (2007) Value-Oriented Requirements Prioritization in a Small Development organization, *IEEE Software*.

Basili, V. and Boehm, B. (2001) COTS-Based Systems Top 10 List, *IEEE Computer,* 34, 5, 91-93.

Beck K. (2001) Extreme programming: explained, Addison-Wesley, USA, December 2001, 7th edition, 417-432.

Bebensee, T., van de Weerd, I. and Brinkkemper, S. (2010) Binary priority list for prioritizing software requirements, *Requirements Engineering: Foundation for Software Quality,* 67-78.

Berander, P. and Andrews, A. (2006) Requirements Prioritization, in *Engineering and Managing Software Requirements*, A. Aurum and C. Wohlin, eds., Springer, 69–94.

Boehm, B. and Ross, R. (1989) Theory W software project management: principles and examples, *IEEE Transactions on Software Engineering*, 15, 902–916.

Brandt, D. R. (1987) A procedure for identifying value-enhancing service components using customer satisfaction survey data, Add Value to Your Service, *American Marketing Association*, 61-65.

Calisir, F., and Calisir, F. (2004) The relation of interface usability characteristics, perceived usefulness, and perceived ease of use to end user satisfaction with enterprise resource planning (ERP) systems, *Computers in Human Behavior,* 20, 4, 505-515.

Cohen, J. (1992) A power primer, *Psychological Bulletin*, 112, 155–159.

DeLone, W. H. and McLean, E. R. (1992) Information Systems Success: The Quest for the Dependent Variable, *Information Systems Research*, 3, 1, 60-95.

Drolet, A. L., Morrison, D. G. (2001a) A practitioner's comment on Aimee L. Drolet and Donald G. Morrison's "Do we really need multiple-item measures in service research?", *Journal of Service Research*, 3, 196–204.

Ehrenberg, A., Juckes, S., White, K. M. and Walsh, S. P. (2008) Personality and self-esteem as predictors of young people's technology use, *CyberPsychology & Behavior*, 11, 6, 739-741.

Emery, C. R. and Tian, R. G. (2002) Schoolwork as products, professors as customers: A practical teaching approach in business education. *Journal of Education for Business*, 78, 2, 97-102.

Fellows, L. and Hooks, I. (1998) A Case for Priority Classifying Requirements, in *Third International Conference on Requirements Engineering (ICRE'98)*, 62-65.

Gardner, D. G. Cummings, L. L. and Dunham, R. B. and Pierce, J. L. (1998) Single-item versus multiple-item measurement sales: An empirical comparison, *Educational and Psychological Measurement*, 6, 898–915.

Greenhouse, S.W. and Geisser, S. (1959) On methods in the analysis of profile data, *Psychometrika*, 24, 95-112.

Heger, D. A. (2004) A Disquisition on the Performance Behavior of Binary Search Tree Data Structures, *European Journal for the Informatics Professional*, 5, 5.

Jailia, K., E. K. J., Sujata, M. S., Jailia, M. M. J., and Agarwal, M. M. A. (2011) LEAN SOFTWARE DEVELOPMENT ("As a Survival Tool in Recession"), *International Journal of Software Engineering and Its Applications*, 5, 3, 63-70.

Kano, N. (1984) Attractive quality and must be quality, *Hinshitsu (Quality)*, 14, 2, 147– 156 (in Japanese).

Karlsson J. and Ryan K. (1996) Supporting the Selection of Software Requirements, in *Proceedings of the 8th International Workshop on Software Specification and Design (IWSSD '96)*.

Karlsson, L., Dahlstedt, Å.G., Regnell, B., Natt och Dag, J. and Persson, A. (2007) Requirements engineering challenges in market-driven software development – An interview study with practitioners, *Information and Software Technology*, 49, 588-604.

Kumar, D. R. (2009) Lean Software Development, *The Project Perfect White Paper Collection*.

Lehman, M. M. and Ramil, J. F. (2002). Software evolution and software evolution processes. *Annals of Software Engineering*, *14*(1-4), 275-309.

Leffingwell, D. and Widrig, D. (2003) Managing Software Requirements: A Use Case Approach, Addison-Wesley, USA, 2nd edition, 124-125.

Lehtola, L. and Kauppinen, M. (2006) Suitability of Requirements Prioritization Methods for Market Driven Software Product Development, *Software Process Improvement and Practice*, 11, 1, 7-19.

Lim, J. S. and Zallocco, R. (1988) Determinant attributes in formulation of attitudes toward four health care systems, *J Health Care Mark*, 8. 2, 25-30.

Witell, L. and Löfgren, M. Classification of quality attributes, *Managing Service Quality,* 17, 1, 54-73.,

Mauchly, J. W. (1940) Significance Test for Sphericity of a Normal n-Variate Distribution, *The Annals of Mathematical Statistics*, 11, 204-209.

Mens, T., Wermelinger, M., Ducasse, S., Demeyer S. and Hirschfeld R. (2005) Challenges in software evolution, In *8th International Workshop on Principles of Software Evolution*. IEEE Press.

Mikulic, J, and Prebez, D. (2011) A critical review of techniques for classifying quality attributes in the Kano model, *Managing Service Quality*, 21, 1.

Myers, J. H. and Alpert, M. I. (1968) Determining Attributes: Meaning and Measurement, *Journal of Marketing*, 32, 4, 13-20.

Pomeroy, I. M., Clark, C. R. and Philip, I. (2001) The effectiveness of very short scales for depressions screening in elderly medical patients, *International Journal of GeriatricPsychiatry*, 3, 321–326.

Racheva, Z., Daneva, M. and Buglione, L. (2008) Supporting the Dynamic Reprioritization of Requirements in Agile Development of Software Products, In: *Proceedings of the Second International Workshop on Software Product Management*, 49-58.

Robins, R. W., Hendin, H. M. and Trzesniewski, K. H. (2001) Measuring global self-esteem: Construct validation of a single-item measure and the Rosenberg Self-Esteem Scale, *Personality and Social Psychology Bulletin*, 27, 2, 151-161.

Ruhe, G. (2005) Software release planning, *Handbook of software engineering and knowledge engineering,* 3, 365-394.

Saaty T. (1980) The Analytic Hierarchy Process: Planning, Priority Setting, Resource, Allocation, McGraw-Hill, New York.

Sharma, R., Yetton, P., and Crawford, J. 2009 Estimating the Effect of Common Method Variance: The Method–Method Pair Technique with an Illustration from TAM Research, *MIS Quarterly,* 33, 3, 473-490.

Sheehe, P. R. and Bross, I. D. J. (1961) Latin squares to balance immediate residual and other effects, *Biometrics*, 17, 405–414.

Sillitti, A. and Succi, G. (2006) Requirements Engineering for Agile Methods, in Engineering and Managing Software Requirements, A. Aurum, and C. Wohlin, eds., Springer, 309-326.

Sinclair, S.A. and Stalling, E.C. (1990) How to identify differences between market segments with attribute analysis, *Industrial Marketing Management*, 19, 2, 31-40.

Thompson, D. V., Hamilton, R. W. and Rust, R. T. (2005) Feature fatigue: When product capabilities become too much of a good thing, *Journal of Marketing Research*, 431-442.

Vavra, T. G. (1997) Improving Your Measurement of Customer Satisfaction: A Guide to Creating, Conducting, Analyzing and Reporting Customer Satisfaction Measurement Program, ASQC Quality Press, Milwaukee, WI.

Wiegers, K.E. (1999) Automating requirements management, *Software Development*, 7, 7, S1–S6.

Zviran, M. and Erlich, Z. (2003) Measuring IS user satisfaction: review and implications, *Communications of the AIS*, 12, 5, 81–104.

APPENDIX. The 10 User Feature Requests Used in the Study

No	Feature description
1	Allow sending emails/ replies to emails at a later time or date. Presently if the user has to send an email or a reply to email at a later date she can only save the draft and remember to send it when the date arrives.
2	Allow user to have another view of their inbox below the message they are composing. This will allow users to reference information from one or more emails, if required, while composing
3	Provide preview of media stored on other sites within an incoming Gmail message when the sender includes only a link. Users get tired of clicking on links to get to the videos and photos of friends
4	Allow sub-string, partial word and wildcard search to provide a powerful mechanism for searching relevant emails
5	Threaded conversations should be made optional to users. Presently it is a mandatory feature
6	Gmail should allow users to report spams to the appropriate authority automatically. This will discourage spammers from spamming in future
7	Send an email to the user's inbox when she marks Bcc (Blind carbon copy) to herself. Currently there is no email sent to the user if she Bcc's herself
8	Allow change in account name without losing contents. Currently the user password can be changed but not the account name
9	Open more than one Gmail account at the same time. Presently the user can only open one Gmail account at a time
10	Allow use of specific colors for emails received from sources specified by the user. This will allow the user to quickly focus clearly on those emails that are important to her