

12-15-2012

IS Product Enhancements: An exploration of two competing techniques for evaluation of user feature requests

Adarsh Kumar Kakar

Randy Smith

David Hale

Joanne Hale

Follow this and additional works at: <http://aisel.aisnet.org/irwitpm2012>

Recommended Citation

Kakar, Adarsh Kumar; Smith, Randy; Hale, David; and Hale, Joanne, "IS Product Enhancements: An exploration of two competing techniques for evaluation of user feature requests" (2012). *International Research Workshop on IT Project Management 2012*. 6. <http://aisel.aisnet.org/irwitpm2012/6>

This material is brought to you by the International Research Workshop on IT Project Management (IRWITPM) at AIS Electronic Library (AISeL). It has been accepted for inclusion in International Research Workshop on IT Project Management 2012 by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

IS Product Enhancements: An Exploration of Two Competing Techniques for Evaluation of User feature Requests

Adarsh Kumar Kakar
University of Alabama
askakar@crimson.ua.edu

Randy Smith
University of Alabama
RSmith@cs.ua.edu

David Hale
University of Alabama
dhale@cba.ua.edu

Joanne Hale
University of Alabama
jhale@cba.ua.edu

ABSTRACT

Time-to-market is critical for product success. Software product managers must quickly determine which features should be included into an IS (Information System) product to satisfy evolving user needs. However, the diverse needs of a large number of anonymous users pose special challenges for quickly analyzing feature requests and selecting a critical subset for product upgrade. This study therefore first investigated the relevance of commonly used prioritization techniques and found the Priority groups method and the Kano survey method appropriate in the context of IS product enhancements. Thereafter exploratory experiments were conducted to compare the efficacy of these two methods. The results of the experiments showed that the Kano survey method was effective in identifying critical features to be implemented into an IS product.

Keywords

IS development, Requirements analysis, IS project management

INTRODUCTION

While developing IS (Information Systems) products for the markets the initial set of features are not elicited from the users but invented by the developers (Potts, 1995). Developers use domain knowledge, creativity, product vision and strategic business goals to design and build the initial product. However, soon after the first release, there is a steady stream of new requirements, improvements suggestions, complaints and bug reports from existing and potential users of the product (Karlsson, Dahlstedt, Regnell, Dag and Persson, 2007). Developers use these inputs to continually enhance the features of the IS (Information System) products to make them more attractive and valuable for the customers and to retain or gain market share.

But unlike bespoke development where a change in the system is funded by the customer, introduction of new features into a product needs resource commitment from the developer. It is therefore important to identify and pursue only those enhancements which give maximum improvement in the perceived value of the product. For this user involvement is fundamental. But unlike in bespoke development, where developers are in close contact with the customer and the scope of the system is guided by a mutual contract, product development for the market involves dealing with a large number of nameless and faceless customers. The lack of day to day interactions, negotiations and conflict resolution with customers makes the task of capturing and selecting the critical requirements to be built into the product even more daunting.

Developers of IS products have therefore evolved various mechanisms to capture information from the users. Of these the use of websites for gathering and prioritizing customer requirements is becoming increasingly prevalent (Laurent and Cleland-Huang, 2009). Web-based methods have been found to be particularly useful for engaging a large number of existing and potential customers in a two-way communication and capturing innovative ideas. The websites include both forums and collaborative tools, and are designed to allow large numbers of stakeholders to participate in the requirements gathering process. The forums' postings or discussions are often displayed in a

threaded format which allows everyone to see the discussion unfold enabling the project team, product managers and users to closely communicate and actively collaborate.

By actively engaging the users, more feature requests are often elicited than are needed to build into the system resulting in congestion in the requirements engineering process (Karlsson et al, 2007; Regnell and Eklundh, 1998). While on the one hand excluding a high value feature may mean losing customers 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. The challenge for the developers is therefore to be able to distinguish between which requirements really add value for the maximum number of users and which requirements do not. But how can developers capture all these ideas without being overwhelmed by them? How can they make the best use of user inputs? How should an organization assimilate and select enhancement requests? To address these questions this study first reviewed IS and non-IS literature to discover prioritization techniques suitable in the context of special challenges posed by product enhancements. Exploratory experiments were then conducted using feature requests of a popular Task Manager application to determine which of the two methods was more effective in separating the vital few requirements from the trivial many.

LITERATURE REVIEW

There are broadly two types of traditional methods for prioritization of requirements, the grouping methods and the non-grouping methods (Wieggers, 1999). Grouping methods, such as the Priority group method, group requirements into categories which can be assigned priority based on the group to which they belong. On the other hand, the non-grouping methods generate a ranked list of requirements in the order of their priority either by using pair wise comparisons (such as Binary tree or AHP) or by suitably assigning weights to each requirement (Planning game, 100 points, Wieggers' method). A brief description of the most commonly used methods is given in Table 1.

The non-grouping methods have an advantage over grouping methods as they providing a ranked priority list at individual requirement level. However, the non-grouping methods are time consuming and difficult to implement in a market-driven product enhancement context where the developer has to deal with a large number of anonymous users. Even in bespoke development where there are a limited number of users, deploying the non-grouping methods of prioritization often becomes a problem. In their study, "Suitability of Requirements Prioritization Methods for Market Driven Software Product Development", Lehtola and Kauppinen (2006) found that pair-wise comparisons with over 20 requirements were difficult in practice. After working for half an hour with the same 20 requirements in the pair-wise comparison case, their first user became so irritated that she was not able to really concentrate on comparing the requirements. All she wanted was to finish the task quickly. The researchers had to abandon their experiment with the other three subjects.

In a IS product enhancement context users continuously post feature requests on user community forums. At any point in time there are typically hundreds of active feature requests. For users to take a snap shot of pending feature requests and rank them by either using pair-wise comparisons or by weighting each feature request would be tedious and time consuming. The time to do pair wise comparisons and assigning weights to each requirement rises exponentially with each increase in number of feature requests. This will make it extremely difficult to find volunteers among users who will be willing to spend that much time. In addition, the large number of anonymous users will make it difficult for product developers to train actual and potential users for applying these techniques correctly and uniformly.

Technique	Description
Priority Groups (IEEE (1998))	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.
Wiegiers' Method (Wiegiers, 1999)	Using a scale of 1 to 9, a requirements value is assessed by determining both its importance to the customer and the consequences that requirement were not implemented.
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.
Binary tree (Karlsson et al, 1998)	Uses the popular binary search algorithm to sort and search information

Table 1: Common Requirements Prioritization Techniques

It is not surprising therefore that 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 (e.g. Lehtola and Kauppinen, 2006; Berander and Andrews, 2005). In this study, we therefore compared the efficacy of the relatively less time consuming and simpler single pass grouping methods for evaluating users' features requests. The goal was to determine which of these two grouping methods, the Priority groups method and the Kano survey method, is more effective in prioritizing requirements by separating the critical features from the non-critical ones.

The Priority groups method, is an IEEE recommended method (Sillitti and Succi, 2006) and among the most traditional and best known (Lehtola and Kauppinen, 2006). 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, 2000). The description for these groups is as follows (Wiegiers, 1999):

High priority requirements are mission critical requirements; required for next release

Medium priority requirements support necessary system operations; required eventually but could wait until a later release

Low priority requirements are a function or quality enhancement; would be nice to have someday if resources permit

Using the Priority groups method users assign a priority to a feature requests he is interested in based on these definitions. The individual responses are then aggregated to determine the category to which the requirement belongs. Based on this categorization IS developers may determine the minimum critical set. They may choose to incorporate in the development plan only those requirements categorized in the High (H) category or only High + Medium (H+M) categories or all requirements in the High + Medium + Low (H+M+L) categories depending on the available resources, technical constraints and customer satisfaction goals.

The Kano (Kano, Seraku, Takahashi and Tsuji, 1984) survey method, developed by Dr. Noriaki Kano of Tokyo Riko University, is another widely accepted method for categorization of requirements. Although it is popular in the domain of non-IS products but its applicability has also been tested in the IS domain (e.g. Tan, Xie, and Chia, 1998; Zhang and Von Dran, 2002). The Kano survey includes two questions for the every product feature: a functional question "How do you feel if this feature is present?" and a dysfunctional question "How do you feel if this feature is NOT present?" The first question reflects the user reward for including the feature into the product and the second question reflects his penalty for not including the feature into the product. The customer 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 customer priorities. If the user expects some feature to be present, but can live without the feature, it is not really a mandatory feature. Based on the user responses to the questions in both functional and dysfunctional form for each of his requirements, the quickest way to assess the questionnaires is to map each response in Table 2 and determine the category. Aggregating this response across users will then determine the category to which a particular requirement belongs according to the majority of users.

		Dysfunctional question				
		Like	Expect	Neutral	Live with	Dislike
Functional question	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

Table 2: Matrix for assessing Kano categories

B-Must have or Basic requirements

P-Linear or Performance requirements

E-Excitement requirements

R-Reverse, i.e. wrong features, that would make the user experience worse

Q-Questionable, i.e. the user answers is inconsistent

I-Indifferent, i.e. the user does not care about this feature

METHOD USED

Exploratory experiments were conducted for an actual and widely used software product – the Astrid Task Manager. Astrid is a popular open source task tracking application developed for the Android operating system. To examine the effectiveness of the prioritization techniques subjects were provided an overview of the Astrid application, its

existing features, and a list of 15 randomly chosen feature requests from amongst the list of all pending feature requests. The feature requests are posted on the Astrid user community forum (http://getsatisfaction.com/todoroo/products/todoroo_astrid). The feature requests taken from the company web site were re-worded in a simple and standard style, a sample set is shown in Table 3 as shifts in structure, content and format may introduce unwanted sources of variability that may confound subject response.

No	Requirement description
1	Choose from a calendar Allow dates to be chosen from a calendar. Currently the user has to manually enter the date
2	Auto Color Task As the user browses through the pending task the color of the task should visually indicate to him how far it is from the due date.
3	Creating tasks that repeat yearly Allow creation of yearly recurring tasks to remind users about important events such as birthdays, anniversaries etc. Currently the application allows daily, weekly and monthly recurring tasks only
4	Geolocation reminders Provide a feature to remind users that they are passing through an important geolocation. For example if the user is passing a favorite supermarket, then remind her that she is doing so and ask whether she needs to purchase anything.
5	Grocery shopping list Provide a feature to enable users to create and update a regular grocery list. This will enable the users to tick off the items purchased from the stores, so that they do not miss anything
6	Make Quiet Hours completely quiet Have a new option - "Super Quiet Hours" - during which all reminders should be disabled. Currently during "Quiet Hours" the vibrator is enabled

Table 3

As we wanted to have a homogeneous group, all subjects selected for the experiment were from a senior undergraduate class of MIS students.

EXPERIMENT 1: CATEGORIZING REQUIREMENTS INTO SUBSETS

Two groups of randomly assigned subjects were formed. Of these, one group of 26 subjects was randomly assigned to categorize feature requests using the Priority group technique. Each of the three Priority group categories was clearly defined and subjects chose the one category out of High/ Medium/ Low which they thought was most relevant for that requirement. The second group of 23 subjects was randomly assigned to categorize feature requests using the Kano survey method. They responded to two questions for each proposed requirement. First, they were asked to give their opinion if the proposed requirement "IS" included in the next release of Astrid. Second, they were asked to give their opinion if the requirement "IS NOT" included in the next release. The reader is directed to Figure 1 for more detail regarding the questionnaires.

The 26 participants using the Priority group technique took an average of 10 minutes to complete the questionnaire. The 23 participants using the Kano technique took an average of 16 minutes to complete the two questionnaires.

ANALYSIS OF RESULTS OF EXPERIMENT 1

		High(H)	Medium(M)	Low(L)
Basic (B)	3	2	1	-
Performance (P)	3	1	2	-
Excitement (E)	1	1		-
Indifferent (I)	8	-	4	4
Total	15	4	7	4

Table 4

The results of the Stage 1 experiment are summarized in Table 4. On analyzing the results it becomes apparent that out of the total of 15 requirements, the 7 Basic, Performance and Excitement requirements, categorized by the Kano survey method, were categorized as either High or Medium priority requirements by the Priority groups method. 4 Indifferent requirements classified by the Kano survey method was classified as Low Priority by the Priority groups method. However, another 4 requirements identified in the Indifferent category were classified as Medium priority by the Priority groups method. One would expect the equivalent classification of requirements identified in the Indifferent category by the Kano method to be Low Priority by the Priority groups method in line with its description as a “requirement that would be nice to have someday if resources permit”. Thus the question arose about which grouping method is accurate. Has the Kano method identified these 4 requirements as Indifferent wrongly or has the Priority groups method identified them wrongly as belonging to the Medium priority category?

EXPERIMENT 2: EVALUATING USER SATISFACTION FOR THESE SUBSETS

We therefore conducted another experiment a week later to measure the user satisfaction with the resulting requirement sets. The same subjects who participated in Experiment 1 were randomly assigned to two groups, one focusing on the Kano technique results and the second focusing on the Priority group technique results. One group of 24 subjects rated their satisfaction if the 7 requirements (B+P+E) out of 15 were implemented into the IS products. The rating was done on a 9 point scale that had a low of “extremely dissatisfied” to a high of “delighted”. The other group of 28 subjects rated their satisfaction with 11 requirements (H+M) out of 15 on the same 9 point scale. The objective was to see whether there was any significant difference in the rating between the two sets of requirements. If a significant difference is found between the mean satisfaction ratings of the two sets then the users are not really Indifferent to the 4 Indifferent requirements which were classified as Medium priority by the Priority groups method as $(H+M)=(B+P+E+4I)$ - see Table 4. If no significant difference is found then it would demonstrate that adding those 4 requirements classified as Indifferent by the Kano survey method did not add to user satisfaction. This implies that the users are unconcerned about whether these 4 Indifferent requirements are included into the product or excluded from it.

ANALYSIS OF RESULTS OF EXPERIMENT 2

Mean satisfaction level – H+M set (4 High + 7 Medium)	7.10
Mean satisfaction level – B+P+E set (3 Basic + 3 Performance + 1 Excitement)	6.80
H ₀ : Mean satisfaction (H+M) = Mean satisfaction (B+P+E)	p = 0.449
Testing Method: Two-sample independent t-test	

Table 5: Satisfaction Level Comparison

A two sample independent t test (Table 5) showed that the mean satisfaction levels for requirement set of 7 requirements (3B+3P+1E) was not significantly different from the mean satisfaction levels for requirement set of 11 requirements (4H+7M) by Priority groups method. As (4H+7M) = (3B+3P+1E+4I) from Table 4, the results show that there was no significant difference between the mean satisfaction ratings of (3B+3P+1E) and (3B+3P+1E+4I), indicating that the 4 Indifferent requirements did not add to the mean satisfaction level of the users. Thus the Kano method had accurately identified the 4 requirements in the Indifferent category. On the other hand The Priority groups method should have classified these 4 Indifferent requirements in the Low priority category rather than in the Medium priority category.

DISCUSSION

The Priority groups method elicits a one-sided user response from the user reflecting the users' importance to have a feature request included in the application but not their reaction to the opposite situation of the same feature not being included in the system. On the other hand the Kano survey method captures both the reward for fulfillment and the penalty for non-fulfillment by using a functional question and a dysfunctional question for each feature request. Past research has shown that the user penalty for fulfillment and the reward for non-fulfillment are not symmetric and that there are three distinct categories of requirements each with their own characteristics (e.g. Brandt 1987; Brandt and Reffet 1989; Stauss and Hentschel 1992; Johnston 1995). For hygiene type (Basic) requirements there is a huge penalty for non-fulfillment but no reward for fulfillment. For motivators (Excitement) there is a huge reward for fulfillment and no penalty for non-fulfillment. For hybrids (Performance) there is both, a reward for fulfillment, and a penalty for non fulfillment.

On analyzing the raw data pertaining to Kano surveys it was found that although many users mentioned in the functional survey that they expect a particular feature to be present in the product, in the dysfunctional survey they mentioned that they could live without the same feature being present in the product or that they are neutral if the feature is not there in the product. As a result more than half (8 out of 15) requirements got classified in the Indifferent category resulting in their being excluded from the prioritized subset by the Kano method. But the Priority groups method, lacking the penalty perspective, classified half (4 out of 8) of these Indifferent requirements in the Medium priority category. Thus while the priority group method does provide a useful but one dimensional perspective of the importance assigned by the user for including a feature into the product, the Kano technique generates from a superset of potential requirements a wealth of customer preference understanding that can drive software requirements prioritization. It provides to the development team distinct sets of candidate requirements, each to be considered differently:

- A set of **Basic** requirements that meet customer core needs and must serve as the kernel of development effort. Not meeting these requirement causes a high level of dissatisfaction although meeting these requirements do not result in increase in customer satisfaction. For example in this experiment users classified "Choosing dates from a calendar" as a Basic requirement.
- A set of **Performance** requirements that should be strongly considered for implementation and further assessed using value-oriented techniques. Performance requirements may be staged over several releases to

deliver immediate value and level resource demands. In this study Users classified “Auto color tasks to indicate how far they are from due date” as a Performance requirement.

- A set of **Excitement** requirements from which the development team should select a small number to delight customers who have competitive procurement alternatives. As suggested for Performance requirements, final selection of Excitement requirements should be based on value oriented prioritization techniques. Users chose “Provide a grocery shopping list” as an Excitement requirement.
- A set of **Indifferent** requirements that can be removed from consideration, significantly reducing wasted requirements and their associated issues. “Make quiet hours completely quiet” was rated in this category by users.

Using these categories developers can systematically reduce the number of requirements delivered while still maintaining customer satisfaction. In our study the Kano method it seems was right. By classifying the 4 requirements as Indifferent which subjects using the Priority group method had categorized in the Medium category will result in over 50 % savings in the number of requirements to be built into the system for the developer without significantly impacting customer satisfaction. The Priority groups technique has an advantage that it is simpler to implement. It requires customers to answer only one survey while the Kano method requires customers to answer two surveys, the functional and the dysfunctional surveys. But it was observed in this experiment that it did not put much strain on the subjects who took on an average only 6 minutes more to respond to the survey compared to the Priority group survey.

CONCLUSION

When customer expectations are high, timelines short, and resources limited, the most essential functionality of the product should be delivered as early as possible (Wieggers 1999). Requirements selection is therefore a critical process. For the development organization, the appropriate set of requirements must meet the customer’s desired functional expectations, minimize the resource outlay and differentiate the product meaningfully from its competitors. For the customer or user of a software product, the appropriate set of selected requirements must deliver the expected functionality of the application domain quickly.

However, the special challenges posed in the context of market-driven product development render many of the common used prioritization techniques impractical to use (Karlsson, Dahlstedt, Natt Och Dag, Regnell and Persson, 2007). This study therefore identified two grouping methods which may be suitable to use in the context of IS product enhancement. Both these methods are less time consuming and do not overly burden the cognitive load of the user.

The results of this study show that the Kano survey method provides a mechanism to satisfy both the customer and the developer of IS products. The Kano survey method, although it took slightly longer than the Priority group method, could identify 8 out of 15 user feature requests that will not significantly impact user satisfaction if not included into the product. Identifying and reducing the waste from requirements is critical to avoiding the creation of waste later in the development process (Womack and Jones, 1998). Priority groups, the other prioritization method, demonstrated limited efficacy in identifying feature requests that are not critical for the users. By using the Kano survey method, the IS organization on the one hand is freed from pursuing “maximum requirements coverage” to being empowered with information allowing it to meet customer expectations while at the same optimally utilizing its resources. On the other hand the customers have the satisfaction of seeing their critical product upgrade requests quickly implemented into the product.

The participants chosen for the empirical study were college students between 18-24 years of age. The rationale behind this approach was to get as homogenous a group of sample as possible. This would enable the investigation to focus on validity by limiting the alternative explanations of the results obtained. However, this confines the applicability of the findings to undergraduate college students. For greater generalizability, future research may consider testing the Kano survey method for other user segments and software product categories. If the results hold it would guide practice on ways and means of developing lean products that provide value to both the users and producers by minimizing waste.

REFERENCES

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

- Berander, P. and Andrews, A. (2005) Requirements prioritization, in Engineering and Managing Software Requirements, eds. A. Aurum and C. Wohlin (Springer Verlag, Berlin, 2005), pp. 69-94.
- Brandt DR (1987) A procedure for identifying value-enhancing service components using customer satisfaction survey data. *Add Value to Your Service*, American Marketing Association 61-5.
- Brandt DR, Reffet, KL (1989) Focusing on customer problems to improve service quality. *The Journal of Services Marketing* 3(4) 5-14.
- IEEE Std 830-1998. IEEE recommended practice for software requirements specifications. IEEE standard.
- Johnston R (1995) The determinants of service quality: satisfiers and dissatisfiers. *International Journal of Service Industry Management* 6(5) 53-71.
- Kano, N., Seraku, N., Takahashi, F. and Tsuji, S. (1984) Attractive quality and must-be quality (in Japanese), *Journal of the Japanese Society for Quality Control*, 14 (2), 39-48.
- Karlsson, J., Wohlin, C. and Regnell, B. (1998) "An evaluation of methods for prioritizing software requirements," *Information and Software Technology*, 39, 14-15.
- Karlsson, L., Dahlstedt Å. G., Natt Och Dag, J., Regnell, B., and Persson, A. (2007) Requirements engineering challenges in market-driven software development An interview study with practitioners. *Information and Software Technology*, 49, 588-604.
- Laurent, P. and Cleland-Huang, J. (2009) Lessons learned from open source projects for facilitating online requirements processes, 15th Intl. Working Conf. on Requirements Engineering: Foundations for Software Quality, Amsterdam, Netherlands.
- Lehtola L. and Kauppinen, M. (2004) Empirical Evaluation of Two Requirements Prioritization Methods in Product Development Projects, *Proceedings of the European Software Process Improvement Conference*, Springer-Verlag, Berlin Heidelberg, 161-170.
- Leffingwell, D. and Widrig, D. (2000) *Managing software requirements – A unified approach*, Addison-Wesley, 2000.
- Leffingwell D. and Widrig D. (2003) *Managing Software Requirements: A Use Case Approach*, Addison-Wesley, USA, 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.
- Poppendieck, T. and Poppendieck, M. (2003) *Lean software development: An agile toolkit for software development managers*, Addison-Wesley, London UK.
- Potts, C. (1995) Invented Requirements and Imagined Customers: Requirements Engineering for Off-The-Shelf Products, *Proc. 2nd International Symposium Req. Eng.*, IEEE Comp. Soc. Press, 128-130.
- Regnell, B. P. and Eklundh, O. (1998) A market-driven requirements engineering process – results from an industrial process improvement, *Requirements Engineering Journal*, 3, 121-129.
- Saaty, T.L. (1980) *The Analytic Hierarchy Process*, McGraw-Hill, Inc.
- Sillitti, A. and Succi, G. (2006) Requirements Engineering for Agile Methods, in *Engineering and Managing Software Requirements*, Springer, 2006, 309-326.
- Stauss B, Hentschel B (1992) Attribute-based versus incident-based measurement of service quality: results of an empirical study in the German car service industry, *Quality Management in Services*. 59-78.
- Tan, K.C., Xie, M. and Chia, E. (1998) Quality function deployment and its use in designing information technology systems, *International Journal of Quality & Reliability Management*, 15 (6), 634-45.
- Wieggers, K.E. (1999) Automating requirements management, *Software Development*, 7 (7), 51-56.
- Womack, J.P. and Jones, D.T. (1998) *Lean thinking: Banish waste and create wealth in your corporation*, Simon & Schuster.
- Zhang, P., and G.M. Von Dran, G. M. (2002) User expectations and rankings of quality factors in different Web site domains, *International Journal of Electronic Commerce* 6 (2), 9-33.