Deriving Business Value from Asymmetric Penalty-Reward Perspectives of IS users

Adarsh Kumar Kakar
University of Alabama , Tuscaloosa, AL, United States., askakar@crimson.ua.edu

David Hale
Management Information System, University of Alabama, Tuscaloosa, AL, United States., dhale@cba.ua.edu

Joanne Hale
Management Information System, University of Alabama, Tuscaloosa, AL, United States., jhale@cba.ua.edu

Deriving Business Value from Asymmetric Penalty-Reward Perspectives of IS users

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

ABSTRACT

When features are added to an existing IS product in response to market demands it is important to assess their business value before implementing them into the product. But how does one estimate the true value of a new feature? Is it sufficient to consider only the customer reward for including a feature into the product or is it also useful to evaluate the customer penalty for not building the feature into the product? The current methods for feature selection capture only the customer inputs for building a feature into the product. The implications of not building the feature are not captured. This study investigates the adequacy of such an approach and discovers that additional business information can be derived by considering both, the reward as well as the penalty perspective of the customer. This information can be utilized by IS developers in adopting the right strategy for product enhancement.

Keywords

Penalty-reward perspective, three factor theory, kano survey method

INTRODUCTION

Soon after the first release of an IS product, there is a steady stream of new requirements, improvements suggestions, complaints and bug reports from existing and potential users/ customers 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 in market-driven development 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. Yet 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 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.

However, by actively engaging the users, more feature requests are often elicited than are needed to build into the system (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.

One of the common methods for capturing and prioritizing user requirements is the Voting method. The idea is very simple. Using forums and collaborative tools on developer websites, users submit their feature requests, and express their priorities as part of their comments. Other users go through these feature requests and/or the comments and Vote for them either by providing a supportive comment and/ or by clicking the Voting button. The developers monitor the feature requests and based on the popularity of the feature requests, as determined by the number of Votes, and possibly the comments and discussions in the user forum, decide to implement them into the system.

Voting provides customers' opinion on whether the feature should be implemented into the product. It does not provide the customers' perspective of the consequences if the feature is not implemented into the product. If the user response to the reward and penalty perspective is symmetric, that is the customer penalty for not implementing a feature into the product is the inverse of customer reward for implementing a feature into the product, then there is no problem. One can estimate the users' penalty response based on the reward response: For example if the user would like a requirement to be implemented into the system then he would dislike if the requirement is not implemented into the system. If the user is indifferent if the requirement is not implemented into the system, then he would be indifferent if the requirement is implemented into the system. If the user would dislike a requirement to be implemented into the system then he would like if the requirement is not implemented into the system

However if the reward and penalty responses are asymmetric, that is, if the user gives a seemingly irrational response by saying that he would like to have a particular feature implemented into a product but is indifferent if the feature is not implemented into the system then there is a potential issue. Would not the developers then be wasting their scarce resources in implementing a feature that does not significantly add value to the customer? To find out whether the IS user perspectives are symmetric or asymmetric, we first surveyed existing literature on the subject and then conducted an experiment with actual users of a mobile task manager application.

LITTERATURE REVIEW

The three factor theory, popular in quality literature as the "theory of attractive quality" (Kano et al, 1984), is based on Herzberg's two factor theory or Motivation-Hygiene theory (Herzberg, 1966). The Motivation-Hygiene (Herzberg, 1966) theory was developed by Frederick Herzberg as an alternative to Maslow's theory (1970) for studying job satisfaction. According to the Motivation-Hygiene theory (Herzberg, 1966) job satisfaction and dissatisfaction are determined by two different sets of factors. Factors found to affect job satisfaction (recognition, achievement, work itself, advancement, and responsibility) are called "motivation factors." Factors found to affect job dissatisfaction (salary, company policies, interpersonal relations and working conditions), are called "hygiene factors" (Brenner, Cormack and Weinstein, 1971).

By implication, customer requirements can be classified into two categories, those that cause customer dissatisfaction if not fulfilled but no significant satisfaction if fulfilled and those that cause customer satisfaction if fulfilled but no dissatisfaction if not fulfilled. "Hygeine" factors are also called "Dissatisfiers" and "Motivation" factors are called "Satisfiers" (Zhang, VonDran, Small and Barcellos, 2000).

Earlier empirical studies (Swan and Combs,1976; Maddox 1981; Cadotte and Turgeon, 1988; Johnston and Selvestro, 1990) of customer requirements found support for Herzberg's (Herzberg, 1966) two factors classification. However, later studies (Brandt, 1987; Brandt and Reffet, 1989; Stauss and Hentschel,1992; Johnston,1995; Anderson and Mittal,2000) found empirical support for the three-factor theory, the third factor leading to dissatisfaction as well as satisfaction. Today the three factor theory is widely accepted. According to the three factor theory or theory of attractive quality, requirements can be classified into three categories or factors:

Basic factors: They are prerequisites and must be satisfied first, at least at threshold levels, for the product to be accepted. The fulfillment of basic requirements is a necessary but not a sufficient condition for satisfaction. The customer takes Basic requirements for granted, and therefore does not explicitly ask for them. They are similar to Herzberg's "Hygiene factors" or "Dissatisfiers". The other names used for Basic factors are Minimum Requirements (Brandt, 1988), Must-be requirements (Kano et al, 1993), Implied requirements (ISO/IEC 9126-1, 2001).

Performance factors: These are requirements 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.

The other names for Performance factors are One-dimensional requirements (Kano et al, 1984), Stated requirements (ISO/IEC 9126-1,2001).

Excitement factors: Excitement requirements are those that the customer did not expect. They surprise the user by adding unexpected value to the product thereby delighting her. The Excitement factors are similar to Herzberg's "Motivation factors" or "Satisfiers". Not fulfilling excitement requirements do not lead to user dissatisfaction. The other names for Excitement requirements are Attractive requirements (Kano et al, 1984), Value enhancing requirements (Brandt, 1988).

Thus a review of literature indicates that except for Performance factors the customer reward for fulfilling a requirement and penalty for not fulfilling a requirement are not symmetric. The Kano (Kano, Seraku, Takahashi and Tsuji,1984) survey method, developed by Dr. Noriaki Kano of Tokyo Riko University, is a widely accepted method for categorization of requirements by capturing the asymmetry in customer penalty-reward perspectives. 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

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 1 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 1: 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

To investigate whether the asymmetry in customer response exists for IS products we conducted an experiment.

METHOD

An experimental method was used in this investigation. The experiment was 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. As we wanted to have a homogeneous group, all subjects selected for the experiment were from a senior undergraduate class of MIS students of a large public university. All subjects, 16 male and 7 female students, were users of the Astrid Task Manager. The test instrument was a list of 15 randomly chosen feature requests from amongst 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 2 as shifts in structure, content and format may introduce unwanted sources of variability that may confound subject response.

No	Feature 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 2: Sample of feature description in the test instrument

Each subject responded to two questions of the Kano survey. 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 details regarding the questionnaire.

ANALYSIS OF RESULTS

The results of the experiment are summarized in Table 3. On analyzing the results, the asymmetric customer reward-penalty perspective becomes apparent. Out of the total of 15 requirements, 3 requirements were classified in the Basic category, 3 in Performance category, 1 in Excitement category and 8 were classified as Indifferent. If the customer reward-penalty was symmetric all requirements would have been classified either as Indifferent, Reverse or in the Performance category (refer Table 1). Of the 8 features which were classified as Indifferent by the Kano survey method, developers may have built 4 features into the product if only the functional survey would have been conducted as the users’ had indicated that they expect these 4 features to be built into the product. But since in the dysfunctional survey users’ had mentioned that they were either “neutral” or can “live with it’ if those 4 features were not included into the product, they got categorized in the Indifferent category. Hence taking both reward and penalty perspective helped in identifying a lean list of only those 7 features from a set of 15 features which have an impact on user outcomes.

The features classified in the Basic category are prerequisites. If they are not fulfilled they lead to extreme dissatisfaction. Hence they are absolutely necessary for the product to survive in the market. But as they are taken for granted by the customer fulfilling them does not increase customer satisfaction. The performance features generate symmetric user response. If they are fulfilled they increase customer satisfaction but if they are not fulfilled they lead to customer dissatisfaction. Developers should therefore be competitive with respect to fulfilling Performance requirements of the user. The excitement factors lead to customer delight and are differentiators in the market place. Developers should therefore ensure that they at least have some features in the product that excite their customers.

Feature Categories	No. of features
Basic (B)	3
Performance (P)	3
Excitement (E)	1
Indifferent (I)	8
Total	15

Table 3: Categories derived from Kano survey

A qualitative analysis of the experiment results show that the users classified the following three of the fifteen requirements of the Mobile app as Basic requirements:
1. Choose date from a calendar – currently users enter the date manually
2. Purging completed tasks - currently tasks have to be purged one by one
3. Create tasks that repeat yearly - presently system allows creation of daily, weekly and monthly tasks only

Intuitively it makes sense to classify these three requirements in the Basic category since the features are so basic that they may be entirely taken for granted by the users that the developer will provide for them in the application. For example 'choosing date from a calendar' is now a commonly available feature across applications and users do not expect to enter date manually. But not providing a calendar feature will cause extreme dissatisfaction as the developer has failed to provide an essential product feature.

The following sets of three features out of fifteen were classified by respondents under the Performance requirements category:
4. Auto color Tasks – to indicate to the users how far it is from due date
5. Shortcut to create tasks – currently it requires 3 clicks to go to the task creation option
6. Color tasks based on priority – to enable users to visually see task priority

This classification also made sense because these are user specific requirements that are not basic to this product (a task tracking system) or even this class of products. Satisfying them will enhance user satisfaction while not satisfying them will result in user disappointment.

One requirement was classified in the Excitement (Unstated) category:
7. Grocery shopping list – to enable users to create and update a regular grocery shopping list that will enable them to check and tick off the items purchased from the store.

This is an innovative feature which a typical user would not normally expect, but would be thrilled to have if provided.

Thus the asymmetric customer response to product features has helped extract additional business information by classifying features into three categories with their own distinctive characteristics. Developers can utilize this information to implement

product those features that target their specific business goals. In addition taking both the reward-penalty perspectives helps identify waste by identifying features which customers are Indifferent to.

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 (Wiegers 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.

The results of this study show that the customers' reward-penalty perspectives are not symmetric. Hence to fully understand the business information contained in a feature request, the developer should capture information using the functional and the dysfunctional survey as exemplified in the Kano survey method. Using the functional and the dysfunctional survey provides a mechanism to satisfy both the customer and the developer of IS products. On the one hand the IS organization 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.

LIMITATIONS AND FUTURE RESEARCH

This study was conducted with a homogeneous group of senior undergraduate MIS students for a Mobile task manager application. The homogeneity of the subject group was necessary for control and internal validity of the results. The downside of this homogeneous sample is that the results of the study may be applicable to only the population from which the sample was taken. As the implications of research in the area could be of considerable value to practitioners, for greater validity and generalization of results future research could replicate this study with different IS products and non-student user segments.

Figure 1: Requirements Prioritization Questionnaire

Part A: All questionnaires begin with a textual and graphical overview of the Astrid app

Part B: Participants were presented with 15 candidate new Astrid features, and then asked for their response. For example, one proposed feature was presented as follows:

Purging completed tasks: Provide a feature to purge all completed tasks. Currently the tasks have to be deleted one by one.

Participants in the Kano survey were asked 2 Multiple Choice Questions:	
How would you feel if the product *did* have this feature? • I like it this way • I expect it this way • I am neutral • I can live with it this way • I dislike it this way	How would you feel if the product *did not* have this feature? • I like it this way • I expect it this way • I am neutral • I can live with it this way • I dislike it this way

REFERENCES

1. Anderson, E.W., Mittal, V. (2000) Strengthening the satisfaction-profit chain, *Journal of Service Research*, 3(2), 107-20.
2. Beck, K. (2001) Extreme programming: explained, Addison-Wesley, USA, 7th edition, 417-432.
3. 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.
4. Brandt DR, Reffet, KL (1989) Focusing on customer problems to improve service quality. *The Journal of Services Marketing* 3(4) 5-14.
5. Brenner, V., Cormack, C., and Weinstein, M (1971) An empirical test of the motivation-hygiene theory, *Journal of Accounting Research,* 9(2), 359-366.
6. Cadotte, E.R., Turgeon, N. (1988) Dissatisfiers and Satisfiers: suggestions from consumer complaints and compliments, *Journal of Consumer Satisfaction, Dissatisfaction and Complaining Behavior*, 1, 74-9.
7. Herzberg, F. (1966). Work and Nature of Man, World Publishing, Cleveland, OH.
8. ISO/IEC 9126-1 (2001) Software Engineering Product Quality-Part 1: Quality Model. International Organization for Standardization.
9. Johnston, R., Silvestro, R. (1990) The determinants of service quality – a customer-based approach, *The Proceedings of the Decision Science Institute Conference*.
10. Johnston R (1995) The determinants of service quality: satisfiers and dissatisfiers. *International Journal of Service Industry Management* 6(5) 53-71.
11. 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.
12. 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.
13. 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.
14. Maddox R (1981) Two-factor theory and consumer satisfaction: Replication and extension. *Journal of Consumer Research,* 8(1): 97-102.
15. Maslow, A. H. (1970) Motivation and Personality, 2nd ed. New York: Harper & Row Publishers, Inc.
16. 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.
17. 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, Q*uality Management in Services,* 59-78.
18. Swan, J.E., Combs, L.J., (1976) Product performance and consumer satisfaction: a new concept, *Journal of Marketing*, 40, 25-33.
19. Wiegers, K.E. (1999) Automating requirements management, *Software Development*, 7 (7), S1–S6.
20. Womack, J.P. and Jones, D.T. (1998) Lean thinking: Banish waste and create wealth in your corporation, Simon & Schuster.

21. Zhang, P. Von Dran, G. Small, R. and Barcellos, S. (2000) A two-factor theory for Web site design, *In Proceedings of the 33d Hawaii International Conference on Systems Science (HICSS 33*). Maui, HI: IEEE Computer Society.