3-22-2019

Insights into the Importance-Performance Paradox of Software Product Attributes

Ashish Kakar
Johns Hopkins University, akakar1@jhu.edu

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

INSIGHTS INTO THE IMPORTANCE-PERFORMANCE PARADOX OF SOFTWARE PRODUCT ATTRIBUTES

Ashish Kakar
John Hopkins University
akakar1@jhu.edu

Adarsh kumar Kakar
Alabama State University
akakar@alasu.edu

ABSTRACT

The importance–performance analysis (IPA) is widely used for identifying which quality attributes should be improved for maximizing user satisfaction. The two dimensional grid of IPA are based on user-perceived attribute importance and performance. If the user-perceived attribute importance is high but its performance low then enhancing the performance of this attribute is likely to result in higher user satisfaction. But some studies have found that user importance may depend on attribute performance. This confounds the IPA analysis. Yet there is no study which has investigated whether this phenomenon is applicable for IS (Information Systems) products. This study conducted with an ERP system users show that user importance of an attribute is indeed dependent on its performance. For some attributes users overestimate their importance when the performance is low and underestimate them when the performance is high while for others the reverse is the case. Implications of this phenomenon for practice are discussed.

Keywords

Importance-Performance Analysis, Binary Search Tree, Penalty-Reward Contrast Analysis

INTRODUCTION

Quality and customer satisfaction are key drivers of product performance. It is argued that satisfaction leads to increased loyalty, reduced price elasticity, increased cross-buying, and positive word of mouth. Numerous empirical studies confirm a positive relationship between customer satisfaction and profitability (e.g., Anderson, Fornell, & Lehmann, 1994; Eklo¨f, Hackl, & Westlund, 1999; Ittner & Larcker, 1998)). For industrial products, the importance of assessing and managing customer/ user satisfaction is widely recognized (e.g., Tikkanen, Alajoutsija¨rvi, and Ta¨htinen, 2000). End-user satisfaction is also an important area of IS (Information Systems) research because it is considered a significant factor in measuring IS success and use (Ives, Olson, and Baroudi, 1983; Torkzadeh and Doll, 1991; Delone and Mc Lean, 1992; Seddon, 1997). It is therefore crucial to identify the critical factors that determine user satisfaction.

As software projects are often constrained by limited resources, project managers must decide how to deploy the scarce resources parsimoniously to achieve the highest level of user-satisfaction. An effective method to set priorities for industrial products is importance–performance analysis (IPA). It analyses quality attributes on two dimensions: their performance level (satisfaction) and their importance to the customer. Evaluations of attributes on these two dimensions are then combined into a matrix that allows a firm to identify key drivers of satisfaction, to formulate improvement priorities, and to find areas of possible overkill and areas of ‘‘acceptable’’ disadvantages (Matzler, Bailom, Hinterhuber, Renzl and Pichler, 2004) (see Figure 1).

The importance-performance grid is constructed by plotting an attribute based on their current performance (user satisfaction) and their importance to the user. The importance-performance axes intersect at mean importance and mean performance. Attributes plotted in each of these quadrants are interpreted as follows:

A. Concentrate here. Users feel that the attribute is very important but indicate low satisfaction with performance.

B. Keep up with the good work. Users value this attribute and are pleased with its performance

C. Low priority. The product is rated low on this attribute but users do not perceive it to be important.

D. Possible overkill. The product is perceived to be high in performance with respect to this attribute but the user attaches only a small importance to it.

In practice, IPA is considered a simple but effective tool (e.g., Hansen & Bush, 1999). It is very helpful in deciding how to best allocate scarce resources in order to maximize satisfaction. However, in the IS domain, requirements engineering techniques have examined only one aspect of IPA - attribute importance - rather than both attribute importance and performance. Therefore we first examine whether IPA can be applied to IS products. Secondly, there is an implicit assumption underlying the IPA that attribute performance and attribute importance are two independent variables. But what if attribute importance is dependent on attribute performance? Currently there is a gap in IS literature. No study has addressed these issues.

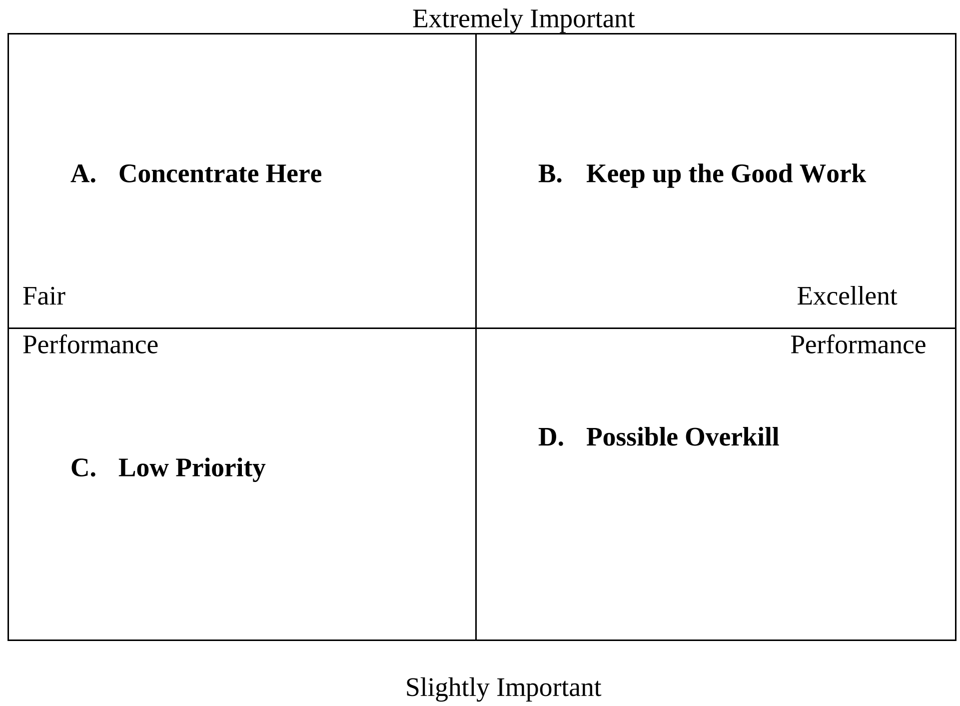

Figure 1. Importance-Performance Analysis

However they are important to investigate. As stakes become higher in the market place, software product managers should know which approach is best for managing user satisfaction. Keeping this context in view, this study first investigates whether attribute importance for IS products is dependent on attribute performance. Further, based on the findings a method is suggested to resolve this paradox. The findings and suggestions made in this study have useful implications for practitioners and future research.

METHOD

Study Setting

149 users of ERP system in HR, Finance and Administration departments at a large public university answered a pencil and paper survey. The university has successfully implemented the following ERP modules since 2006: Financial Accounting, Controlling, Asset Management, Materials Management and Human resources. Data was collected on users' importance and satisfaction rating for the following 6 product attributes – Functionality, Reliability, Portability, Efficiency, Usability and Maintainability - using relevant items from the survey instrument designed for ERP products (Alrawashdeh, Muhairat and Althunibat,2013) based on ISO 9126 (2001) model..

Method of Analyses

To establish reliability and validity of the measures used in the study factor analysis was performed and internal reliabilities and correlation matrix of the measures were examined. After establishing the reliability and validity of the measures and based on the attribute importance and performance rating of the users an IPA was conducted. To construct the importance–performance matrix, the mean of the user satisfaction ratings was calculated. Attribute importance was measured using a multiple regression analysis with overall satisfaction as the dependent and attribute performance as the independent variables. The means of attribute importance and performance (satisfaction) were used to split the axes. Users rated their importance and satisfaction ratings for each item in the survey instrument.

Next we used PRCA (Penalty Reward Contrast Analysis) (e.g., Brandt, 1987; Anderson & Mittal, 2000; Brandt, 1988; Matzler & Sauerwein, 2002; Mittal, Ross, & Baldasare, 1998l Kakar, 2015a; Kakar, 2015b;Kakar, 2017a; Kakar, 2017b; Kakar, 2017c; Kakar, 2018) technique to determine if importance was a function of performance. It involved the use of regression analysis with dummy variables. One set of dummy variables was created and used to quantify high performing attributes, and another set was created to quantify low performing attributes. In order to conduct the analysis, ‘‘low performance’’ is coded (0,1), ‘‘high performance’’ (1,0), and ‘‘average performance’’ (0,0). Based on this coding scheme, a multiple regression analysis is conducted for each variable. Two regression coefficients are obtained—one to measure the impact when performance is low, the other one when performance is high—in order to estimate the impact of attribute-level performance on overall performance (Matzler, Bailom, Hinterhuber, Renzl and Pichler, 2004).

For comparison with IPA, we used the Binary Search Tree Method a well-known requirements engineering technique to get user importance rating for the attributes. Bebensee, van de Weird and Brinkkemper (2010) found that the Binary Search tree method scales up well for software products. 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.

Prioritizing software attributes using this technique involves subjects constructing a binary search tree consisting of nodes equal to the number of attributes. First a single node holding one attribute is created. Then the next attribute 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 attributes 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 thus providing a ranked list of attributes.

Experimental Procedure and Controls

The experiment was conducted in two rounds to minimize compounding and order effects. If the respondent is asked in one question about the importance of an attribute and in the next question about his satisfaction with that attribute, his answer to the first may influence his response to the second. By grouping all of the importance measures in one section and all of the performance measures in a later section, there a distinct separation between his ratings for each attribute. Further, previous research demonstrates that the temporal separation between measures reduces potential effects due to Common Method Variance (Sharma et al., 2009).

A sample question for performance of an item in the Usability scale included: "How easy is it to use the ERP system?" anchored at 9 = Extremely and 1 = Not at all. A sample question for importance of an item in the usability scale was: "How important is it to be able to easily use the ERP system?" anchored at 9 = Critical and 1 = Not at all. The overall satisfaction with ERP was measured by asking the question "How satisfied are you with ERP" anchored at 9=Extremely and 1= Not at all in the first round of the study.

RESULTS AND ANALYSIS

1. The factor analysis procedure was done using IBM SPSS Statistics Version 19. Dimension reduction was performed on the data pertaining to all the 6 measurement scales. The results of Varimax rotation show that the 6 factors extracted represented each of the 6 scales. All items of a scale loaded on the respective factors with no significant (> .40) cross loading between factors. We then measured the internal reliabilities of the scales used in the study. As can be seen from the Table 1, the alpha reliabilities are all greater than .70.

Name of the scale	Cronbach's Alpha	Number of Items
Functionality	.956	5
Reliability	.941	4
Portability	.889	4
Usability	.862	5
Maintainability	.884	4
Efficiency	.785	3

Table 1. Internal Reliability of Scales

2. Functionality and Reliability were at the highest level of user priority (see table 4) according to Binary Search tree (ranks 1 and 2) and therefore it recommends investing in improving their performance to increase user satisfaction. IPA recommends a status quo (see Figure 2) on functionality and reliability as even though the attributes are high in user importance investing in further enhancing functionality and reliability would probably give diminishing returns as they are already performing at a high level. PRCA highly recommends investing in enhancing functionality as it would increase user satisfaction irrespective of the current level of performance (see table 3). However it recommends investing in Reliability only at low performance to prevent dissatisfaction.

Name of the scale	Importance	Performance
Functionality	7.12	6.98
Reliability	7.32	6.73

Portability	3.17	3.88
Usability	8.62	4.11
Maintainability	8.84	4.21
Efficiency	3.78	7.11

Table 2. Users' Importance-Performance Ratings

3. Efficiency ranked 5 (out of 6 ranks) in importance rating by Binary Search tree (Table 4) and therefore this attribute would be low in its priority for further improvements. IPA suggests concentrating on efficiency is an overkill (figure 2), PRCA recommends that investing in improving efficiency at low performance can increase user satisfaction but not at high performance.

Name of the scale	High Performance	Low Performance
Functionality	.312 **	-.319**
Reliability	.042	-.276**
Portability	.046	.097
Usability	.006	-.391***
Maintainability	.062	-.276**
Efficiency	.056	-.156*

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

Table 3. Results of PRCA

4. Binary Search tree accords medium priority (Ranks 3 and 4) for improvements in usability and maintainability (table 4) while IPA recommends highest priority be given to maintainability and usability attributes (figure 2), PRCA recommends that at low performance this focus is important to invest in enhancing maintainability and usability but at high performance it is not remunerative (table 3).

5. Portability is the only attribute where there seems to be a consensus among all methods – Binary search tree IPA and PRCA – that investing in its improvement would be unremunerative.

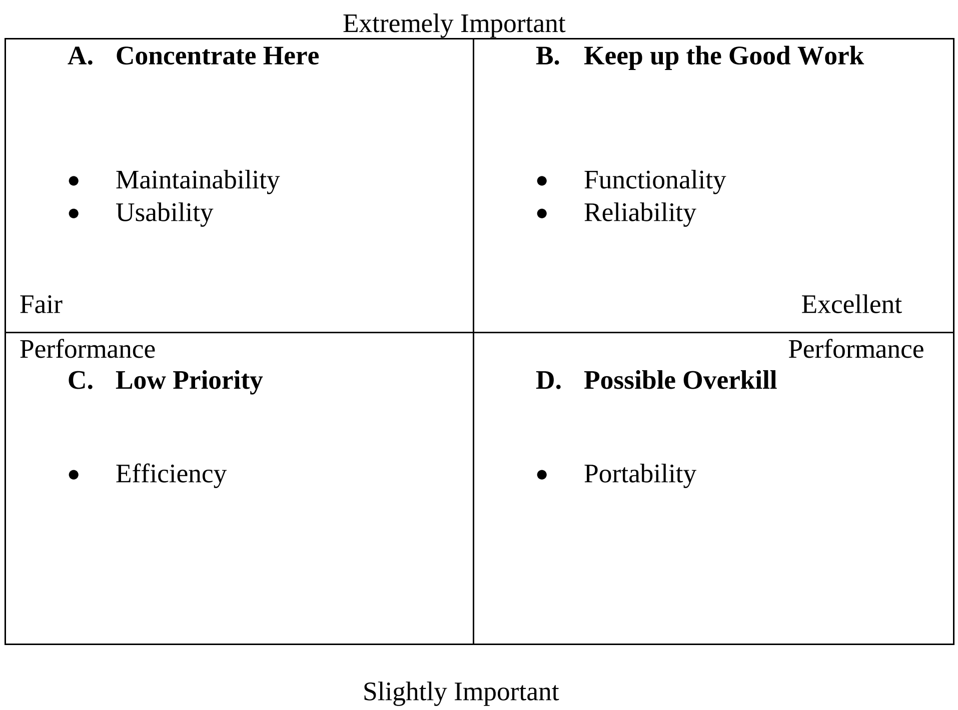

Figure 2. IPA for ERP attributes

Name of the scale	Ranks
Functionality	1
Reliability	2
Portability	6
Usability	3
Maintainability	4
Efficiency	5

Table 4. Results of Binary Search Tree

DISCUSSION

Analysis of the results of the study show that except for Portability the recommendations of Binary Search Tree, IPA and PRCA for selecting attributes for improvement vary. This is because their underlying assumptions are different. Requirements engineering techniques such as Binary Search tree consider only user Importance for prioritization of attribute improvements. It is assumed that higher the user importance for an attribute the more satisfaction he will experience when the performance of that attribute is enhanced. However user Importance does not provide a complete picture. Techniques such as IPA therefore also consider the current performance of the attribute along with its importance to user. The underlying assumption is that if the current performance is high then investing in improvement of this attribute may not be remunerative to the producer of the software for two reasons. First, more resources are needed to improve an attribute already performing at a high level than to improve an attribute performing at a low level. Secondly the impact on the user by improving this attribute may not be as high as when a low performing but important attribute is improved.

However, the IPA technique assumes that there is no interaction effect between attribute Important and Performance. But the results of the study show that attribute Importance is dependent on Performance. Thus the IPA technique is not a valid techniue for requirements prioritization. If this relationship is linear and the Importance of the attribute increases with the decrease in Performance then using the existing requirements engineering techniques such as Binary Search tree would be appropriate even though it considers only user importance for prioritization. But as the results of the study show this does not seem to be the case as the Importance of some attributes such as functionality was high irrespective of the current level of performance. For other attributes such as Portability the Importance was low (insignificant) irrespective of Performance. The importance of other attributes such as Reliability, Usability, Maintainability and Efficiency was high at low performance levels but insignificant at high performance levels. Thus techniques such as Binary Search Tree based only on Importance are also not valid.

Under these conditions PRCA appears to be an appropriate technique to be used. It considers both Importance and Performance but does not need to assume any relationship between the two. Further, the impact of these attributes at high as well as low levels of performance on overall user satisfaction with the software product can be predicted by the regression model. The absolute values of the standardized coefficients (Table 3) obtained by PRCA represent the magnitude of attribute impact on overall user satisfaction.
The attributes which have a –ve standardized coefficient at low attribute performance but +ve standardized coefficient at high performance are dissatisfiers. Improving such attributes can prevent disatisfaction but cannot increase overall satisfaction with the product. From Table 3 we can see that Reliability, Usability, Maintainability and Efficiency are dissatisfiers. Improving attributes such as Functionality (Table 3) can increase overall user satisfaction but at low performance can cause dissatisfaction. Some attributes such as portability do not have significant impact on satisfaction at both high and low performance indicating user indifference with these attributes. PRCA thus provides product/ project managers with much more useful information for making trade-off decisions that both IPA and requirements engineering techniques such as Binary Search Tree.

CONTRIBUTION

In response to user demands Project/ Product Manager have to often decide between conflicting user requirements and priorities for improvement in performance of product attributes. To address this problem various techniques have been developed for managing the trade-offs involved in making this decision. This study compares three popular techniques for their efficacy in identifying the priority of attributes, the Binary Search Tree and IPA. Based on the results obtained the study calls into question the validity of two of the three assessed techniques – Binary Search Tree, IPA and PRCA. We discuss why the use of IPA and Binary Search Tree techniques may not be able to accurately identify product attributes for improvements. We also discuss the managerial recommendations for investing resources in selected attributes based on the PRCA technique.

REFERENCES

1. Ahl, V. 2005. An Experimental Comparison of Five Prioritization Methods, *Master's Thesis, School of Engineering*, Blekinge Institute of Technology, Suecia.
2. Alrawashdeh, T. A., Muhairat, M., & Althunibat, A. (2013) Evaluating the Quality of Software in ERP Systems Using the ISO 9126 Model. *International Journal*.
3. Anderson, E. W., Fornell, C., & Lehmann, D. R. (1994) Customer satisfaction, market share and profitability: Findings from Sweden. Journal of Marketing, 58, 2, 112– 122.
4. Anderson, E.W., ad Mittal, V. (2000) Strengthening the satisfaction-profit chain, *Journal of Service Research*, 3,2, 107-20.
5. 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.
6. 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, 61-5.
7. Brandt, D. R. (1988) How service marketers can identify value-enhancing service elements, *Journal of Services Marketing*, 2, 35–41.
8. DeLone, W. H., and McLean, E. R. (1992) Information Systems Success: The Quest for the Dependent Variable, *Information Systems Research*, 3, 1, 60-95.
9. Eklo¨ f, J. A., Hackl, P., & Westlund, A. (1999) On measuring interactions between customer satisfaction and financial results. *Total Quality Management*, 10, 4-5, 514– 522.
10. Hansen, E., and Bush, R. J. (1999) Understanding customer quality requirements: Model and application. *Industrial Marketing Management*, 28,2, 119-130.
11. ISO/IEC 9126-1 (2001) Software Engineering Product Quality-Part 1: Quality Model, International Organization for Standardization.
12. Ittner, C. D., & Larcker, D. F. (1998) Are non-financial measures leading indicators of financial performance? An analysis of customer satisfaction. Journal of Accounting Research, 36, 3, 1 – 35.
13. Ives, B., Olson, M. H., and Baroudi, J. J. (1983) The Measurement of User Information Satisfaction, *Communications of ACM* , 26, 10, 785-793.
14. Kakar, A. K. (2015). Why do users speak more positively about Mac OS X but are more loyal to Windows 7?. *Computers in Human Behavior*, 44, 166-173.
15. Kakar, A. K. (2015) Investigating the penalty reward calculus of software users and its impact on requirements prioritization. *Information and Software Technology*, 65, 56-68.
16. Kakar, A. K. (2017) Do Reflexive Software Development Teams Perform Better?. *Business & Information Systems Engineering*, *59*, 5, 347-359.
17. Kakar, A. K. (2017) Investigating the Prevalence and Performance Correlates of Vertical Versus Shared Leadership in Emergent Software Development Teams, *Information Systems Management*, 34, 2, 172-184.
18. Kakar, A. K. S. (2017) Why do users prefer the hedonic but choose the Utilitarian? Investigating user dilemma of hedonic-utilitarian choice. *International Journal of Human-Computer Studies*, 108, 50-61.
19. Kakar, A. K. S. (2018) Engendering cohesive software development teams: Should we focus on interdependence or autonomy?. *International Journal of Human-Computer Studies*.
20. Matzler, K., Bailom, F., Hinterhuber, H. H., Renzl, B., and Pichler, J. (2004) The asymmetric relationship between attribute-level performance and overall customer satisfaction: a reconsideration of the importance–performance analysis, *Industrial Marketing Management* 33,4, 271-277.
21. Mittal, V., Ross, W. and Baldasare, P. (1998) The Asymmetric Impact of Negative and Positive Attribute-Level Performance on Overall Satisfaction and Repurchase Intentions, *Journal of Marketing*, 62, 33-47.
22. Seddon, P. B. 1997. A Respecification and Extension of the DeLone and McLean Model of IS Success, *Information Systems Research* , 8, 3, 240-253.
23. Tikkanen, H., Alajoutsijarvi, K., & Tahtinen, J. (2000) The concept of satisfaction in industrial markets: A contextual perspective and a case study from the software industry. Industrial Marketing Management, 29, 4, 373–386.
24. Torkzadeh, G., and Doll, W.J. (1991) Test-Retest Reliability of the End-User Computing Satisfaction Instrument, *Decision Sciences*, 22, 1, 26-37.