# Quality-Based Prioritization: An Approach for Prioritizing Software Requirements

Emmanuel O.C Mkpojiogu and Nor Laily Hashim
*Software Technology Research Group, School of Computing,*
*Universiti Utara Malaysia, 06100 Sintok, Kedah, Malaysia.*
*laily@uum.edu.my*

***Abstract*—There are several requirements prioritization techniques, but none is considered the best. Most of the techniques have not been widely adopted. Research reveals that most of the techniques suffer from usability and scalability problems. Also, most of the techniques are not quality-based and are not geared at capturing the importance value of quality attributes and incorporating the satisfaction of stakeholders into the process of prioritization of requirements. In this study, a new approach for prioritizing software requirements that is quality-based and that utilizes Kano theory of quality attributes is proposed. The approach models customer satisfaction using requirements prioritization aspects as contributing variables. Three indexes were used in this study, with Satisfaction Index (SI), Dissatisfaction Index (DI) and Average Satisfaction Coefficient (ASC) (all representing customer satisfaction that results from whether or not requirements are met), as dependent variables and requirements prioritization aspect(s) as independent variable(s). The estimates from the derived models were used to prioritize software requirements. The result showed that the estimates from the three models produced consistent requirements prioritization, implying that any of the models can be used for quality based requirements prioritization.**

***Index Terms*—Requirements Prioritization; Quality Based Prioritization; Kano Method.**

## I. INTRODUCTION

Requirements prioritization can be defined as the process of attaching priority to elicited requirements based on some criteria and ordering them on the basis of importance to assist in decision making and in planning for product design and releases [1]. The prioritization of requirements is taken as a significant and integral part of software requirements engineering. The process involves ranking elicited requirements based on some aspects, so as to build quality software products within the constraints of available resources. The constraints of budget and time along with the need for a clue for decision making, make prioritization of requirements inevitable [3]. Requirements are given weights and ranked accordingly based on their importance and are put in a priority list to be implemented in subsequent releases [1]. The decision on which requirements are to be implemented in terms of time and order of implementation is a crucial process in the engineering of software products [2]. The quality of software and the timeliness of its release are hinged on how well the requirements prioritization process is managed [1]. Several techniques on requirements prioritization have been proposed, while some are already in use, most of the proposed approaches have not been adopted widely [1],[3]. According to [3], the reason for their not being widely adopted or the slowness in their adoption is because they are difficult to implement, too complex, time consuming and/ or inconsistent. In other words, these approaches have poor usability [3]. As indicated by [1], prioritization techniques should be easy to both use and learn and should win the confidence and interest of the user. Another challenge is the scalability of some of these techniques as some of them are not applicable to the prioritization of large number of requirements [1], among such techniques with scalability problem are: Numerical assignment, Simple Ranking, One Hundred Dollar etc. [7] [8].

The 'best' prioritization approaches are situation dependent and study centric and not universal [10]. Most of the techniques have not been widely adopted as most of them have usability and/or scalability limitations [1] [3]. In addition, these techniques do not model the perceived satisfaction or dissatisfaction of stakeholders if the requirements are met or unmet. More so, these approaches do not take into account the customer or users' perception of the product quality on the ground that the requirements are either fulfilled or not fulfilled. It is on this note that this study proposes a new approach for prioritizing software requirements that will take into account the modeling of stakeholders satisfaction or dissatisfaction if requirements are met or unmet respectively. It will also take into account the perception of the quality of the intended product based on whether the requirements are fulfilled or not, from the point of view of the customers. This new approach: Quality-based prioritization (QBP) is based on Kano theory of quality attributes (Kano et al, [5] [11].

The remaining part of this paper is in five sections, namely: section 2-background, section 3-related works, section 4- the proposed approach, section 5-implementation, section 6-discusion, and section 7-conclusion and future works.

## II. BACKGROUND

With the increasing complexity associated with software and with managers of software projects being compelled to make trade-offs and concessions in order to complete projects within schedule, requirements prioritization has thus become an increasing part of making sure that projects are successful. The limitation in available resources implies that not all elicited requirements can be implemented at least in the first release. This makes requirements prioritization a compelling exercise [3]. There are a number of aspects of requirements prioritization as is shown in the next subsection.

### *A. Aspects of Requirements Prioritization*

Quality software development is dependent on selecting

the right candidate requirements that are prioritized based on some priority aspects [1]. Aspects are criteria, factors, properties or attributes that can be utilized in prioritizing software requirements [7]. Some of the identified prioritization aspects are [7][8]: Importance: To find out the requirements that are most important, stakeholders can prioritize the requirements on the basis of how important they are to them.

Time: Time expended in implementing the candidate requirement successfully can be captured as an aspect for requirements prioritization.

Cost: Cost can be the money expended in implementing the candidate requirement successfully. This cost can be influenced by staff hours and extra resources needed for requirements implementation among others.

Penalty: How much is to be paid if a given requirements is not fulfilled is the penalty, that is, the cost of an unmet requirement.

Risk: These are the uncertainties associated with projects like unrealistic schedules, limited budgets, requirements changes, gold plating, developing wrong function etc.

Others include: volatility, strategic benefit, market value and available resources [7] [8]. Requirements can be prioritized based on single or several aspects. According to Ma [7], requirements prioritization based on single aspect is easier than those based on multiple aspects; however, it is important to utilize several aspects so as to increase the extent of the success of the final product. But what aspect should be considered depends on the situation at the time.

### *B. Techniques of Requirements Prioritization*

There are several techniques on software requirements prioritization available in the literature [14]. These techniques can be classified based on the kind of scale they support. A few of the popular requirements are categorized and explained as follows [7, 8, 14]: Nominal scale: 1) Numerical Assignment etc. With Numerical assignment, requirements can be grouped into different priority groups, say, “critical”, “standard” and “optional”. All the requirements in a priority group have equal priority [7].

Ordinal scale: 1) Simple Ranking, 2) Bubble Sort etc. Simple ranking is the ranking of requirements from 1…n, where the least important among the requirements is ranked n and the most important is ranked 1. With bubble sort, two requirements are compared at a time and swap if they are in the wrong order, this continues until there is no more swap needed. The bubble sort results in a list of ranked requirements [9, 15].

Ratio Scale: 1) Hundred Dollar Method (100 Point), 2) Analytic Hierarchy Process (AHP), Hierarchy AHP, Minimal Spanning Tree etc. The Hundred dollar method is also known as Cumulative Voting. In this technique, each stakeholder is to distribute an imaginary $100 to the requirements. AHP is designed for decision making of a complex type. AHP compares all pairs of hierarchical requirements to get the priority. With AHP, the attributes and alternatives are first identified for each requirement and then used to build a hierarchy. An n*(n-1)/2 pair-wise comparisons are required for the AHP method. Hierarchical AHP uses AHP technique to prioritize requirements that are in the same level of hierarchy. This method reduces the number of decisions when compared to AHP. The number of redundant comparisons is reduced since not all requirements are compared. Minimal Spanning Tree is a direct graph that is minimally connected. It constructs unique pairs of requirements and reduces the number of pair-wise comparisons. Its ability to identify judgments that are inconsistent is however low [15].

## III. Related Works

Otero et al [17] proposed a quality-based prioritization (QBP) approach that uses desirability function. Using this approach, once requirements are elicited, a set of quality attributes are identified as criteria for evaluation. The attributes are defined in terms of several features with each feature determined to be present or absent. After identifying all features, each requirement is evaluated against each feature employing a binary scale. The requirements that meet the highest number of features exposes higher quality level (or priority) for the given quality attribute. Desirability functions are used to fuse all measurements into a unified value that represents the overall quality of the requirement, once all the requirements are evaluated and all the measurements are computed. Using desirability function, each system response, yi, is converted to a function, di, that ranges from 0 to 1 (1 is when a requirement is met and 0 is when it is not). The binary numbering, underscores the attached priorities. The approach in this study does not use desirability function to prioritize, but estimates from a model that is built using Kano model’s Customer Satisfaction (CS) coefficient as dependent variables and requirements aspect(s) as independent variable(s) [17].

In addition, Chen and Chang [16] proposed a framework that applies Kano model and CS Coefficient to prioritize the action items of the 5S practice: Seiri, Seiton, Seiso, Seiketsu, and Shitsuke [16], to assist managers in allocating limited resources to places most valued by customers. The 5S are five Japanese words meaning Organization, Neatness, Cleaning, Standardization and Discipline respectively. These are baseline Total Quality Management (TQM) practice [16]. In the framework, they proposed the use of Kano model for the classification of the 5S action items into the respective Kano quality attribute categories. And then utilize the CS-Coefficient to prioritize the items to enable managers to be able to allocate resources that are available appropriately so as to meet customers’ needs and expectation. Their method employed the following five (5) steps: 1. Prepare a 5S checklist by 5S committee; 2. Construct Kano questionnaire; 3. Administer customer interviews; 4. Categorize action items per the result of Kano questionnaire, and 5. Prioritize 58 check points using CS-Coefficients [16].

Essentially, Chen and Chang [16] approach uses the CS-Coefficients computed from the Kano analysis in prioritizing requirements. The aim is to produce prioritized requirements that satisfy customers’ expectations since it emanates from the Kano process that ensured that customer satisfying requirements were elicited from a Kano survey. The process, though not directly applied to software requirements can be applied to software requirements prioritization. However, its application is limited. It only accounts for customer satisfaction based on whether requirements are met or unmet. It does not account for other aspects of requirements prioritization such as importance, cost, risk, time etc.

The approach proposed in this study, uses CS-Coefficients as dependent variables and requirements aspects as independent variables to build a model that predicts customers’ satisfaction if requirements are met or unmet. This overcomes the weakness of the Chen and Chang’s

framework. Our approach allows for multiple aspects to contribute to the computed CS-Coefficients, thus allowing one or more aspects to account for or contribute to customers' satisfaction.

## IV. THE PROPOSED METHOD

The QBP approach as was in Chen and Chang [16], is based on the Kano's theory of quality attributes. Kano as cited in [6] developed the 'M-H property of quality. He adapted the work of Herzberg et al as cited in [6], that is, the 'Motivation-Hygiene Theory'. In addition, Kano et al [12] proposed a two dimensional model on quality, based on customers' experience and perception. The Kano model grouped requirements and product features into three main categories as follows [6]:

- One-dimensional attributes: this set of requirements result in customer satisfaction when fulfilled/met and dissatisfaction when not fulfilled or unmet. The better the attributes the more the customer likes them. This kind of requirements/features has a linear characteristic.
- Attractive attributes: This set of requirements do not cause dissatisfaction when absent or when not fulfilled/ met, because they are not being expected by customers, who are unaware of what they are missing, but if these requirements are fulfilled, the customer will be delighted and satisfied.
- Must-be attribute: This set of requirements are taken for granted by the customer when fulfilled, but if the requirement is not met, the customer is very dissatisfied.
- Other categories include: Indifferent attribute where the customer does not care whether or not the requirement is met. Reverse attribute, where the customer has a reverse or an inverse expectation. If the requirement is fulfilled, the customer is rather dissatisfied.

Kano model is used as it allows for user satisfying requirements to be elicited and categorized based on quality attributes. It also captures the perceived quality of yet-to-be designed/ developed products from the point of view of customers/stakeholders.

### *A. Coefficient of Customer Satisfaction*

The Kano model was designed to provide qualitative categorization of requirements and attributes of intended products and thus was limited for quantitative evaluation. In the light of this, Berger et al [11] improved on the model by providing the coefficient of customer satisfaction (CS) as shown in Equations 1 and 2:

$$SI = \frac{A + O}{A + O + M + l} \tag{1}$$

$$DI = \frac{O + M}{A + O + M + l} \tag{2}$$

From the above equations, SI (Satisfaction Index), is the degree or extent of satisfaction when the requirements are fulfilled, DI (Dissatisfaction Index), is the degree or extent of dissatisfaction when the requirement is not fulfilled. SI and DI are also a measure of the customers' perception of the quality of the would-be product and the influence of the fulfillment of the requirements on such product. They capture the importance value of quality attributes. Products that satisfy customers are perceived by them as being of quality; the reverse is the case for products that do not delight them. A, is Attractive attribute or requirement, O, is One-dimensional attribute or requirement, M, is Must-be attribute or requirement and I, is Indifferent attribute or requirement. The minus sign placed in the DI equation emphasizes the negative influence of the attribute on customers' satisfaction. The CS-coefficient ranges from 0 to 1 or -1. A positive CS-Coefficient runs from 0 to 1 while a negative CS-Coefficient ranges from 0 to -1. Zero (0) implies no influence on satisfaction if the requirement is met (as in SI) or on dissatisfaction if the requirement is not met (as in DI). The closer the value is to 1, the greater the impact of meeting the requirement is on user or customer satisfaction (that is, for SI) and the closer the value is to -1, the greater the influence of not meeting the requirement is on user or customer dissatisfaction (that is, for DI).The closer the value is to zero, the lesser the influence [6], implying that the particular requirement or feature has lesser impact on user or customer satisfaction and on the perceived software product quality.

Furthermore, Park et al [5] proposed the Average Satisfaction Coefficient (ASC) as shown in Equation 3 to determine the importance value of quality attributes. They showed that SI and DI can be averaged to obtain an Average Satisfaction Coefficient, another measure of the extent of satisfaction customers derive from met features or fulfilled requirements. The measure also captures the perceived quality of would-be products and the influence of requirements on such products.

$$ASC = \frac{(SI + DI)}{2} \tag{3}$$

### *B. Proposed QBP Methodology*

The QBP Methodology consists of five (5) steps: 1) Obtain Kano model coefficient of customer satisfaction (CS), 2) Regress a linear combination of prioritization aspects on CS, 3) Obtain the estimates of CS (SI, DI or ASC). 4) Rank the estimates of CS (SI, DI or ASC), 5) Compute the 1st, 2nd, and 3rd quartiles (Qs) for the estimates of CS, and 6) Assign priority: <Q1 (Low); >=Q1&<Q2 (Medium); >=Q2&<Q3 (High); >=Q3 (Very High). This is shown in the Figure 1.

Step 1: Obtain Kano model coefficient of customer satisfaction: This is obtained from a Kano survey and from Kano classification table (See Matzer et al [13] for clue on how to conduct a Kano survey and capture the Kano attribute categories). From the Kano classification table, compute the coefficient of CS using equations (1), (2) and/ or (3) above.

Step 2: Regress a linear combination of prioritization aspects on CS: The CS consisting of SI, DI and ASC are dependent variables. These variables represent the perceived quality of the software product due to the influence of the requirements or features been met or unmet. The prioritization aspects are the independent variables. Depending on the situation, one or more aspects can be combined and regressed on CS. Below are the models of CS regressed with requirements prioritization aspects:

$$SI = \alpha + \beta 1(Aspect1) + \cdots + \beta n(Aspectn) \tag{4}$$

$$DI = \alpha + \beta 1(Aspect1) + \cdots + \beta n(Aspectn) \tag{5}$$

$$ASC = \alpha + \beta 1(Aspect1) + \cdots + \beta n(Aspectn) \tag{6}$$

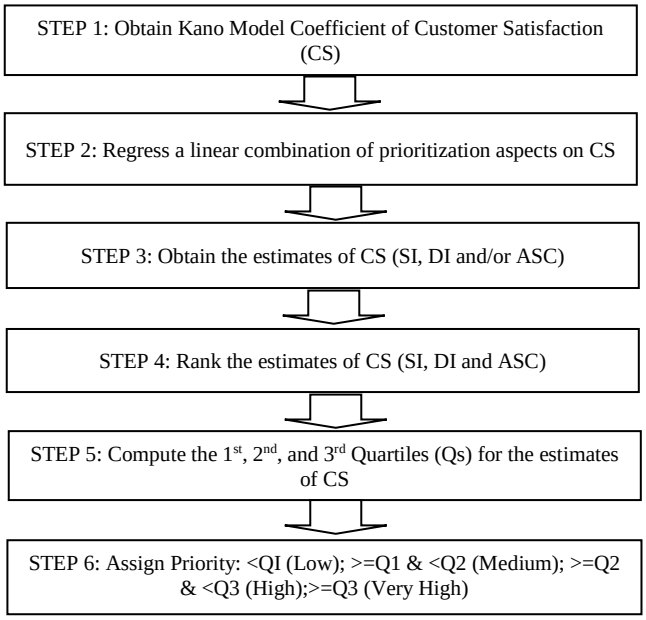


Figure 1: QBP Methodology

Step 3: Obtain the estimates of CS (SI, DI or ASC): From the regression in step 2, obtain the estimates of SI, DI and/or ASC. These estimates are a measure of the combined impact of the linear combination of the prioritization aspects on the customers or users' satisfaction and on their perceived quality of the would-be product that will have these features.

Step 4: Rank the estimates of CS (SI, DI or ASC): Perform a simple ranking of the obtained CS estimated from 1…n, where 1 has the highest priority and n has the least priority. The estimates represent individual requirements corresponding to them.

Step 5: Compute the 1st, 2nd, and 3rd Quartiles (Qs) for the estimates of CS: Using the formula for Quartile computation, compute the 1st (Q1), 2nd (Q2), and 3rd (Q3) Quartiles for the estimates of CS. The Q1 is the 25th percentile; Q2 is the 50th percentile while Q3 is the 75th percentile of the CS estimates. Below are the equations for the computation of Quartile:

$$Q1 = \frac{(n+1)th}{4} term \quad (7)$$

$$Q2 = \frac{(n+1)th}{2} term \quad (8)$$

$$Q3 = 3\frac{(n+1)th}{4} term \quad (9)$$

Step 6: Assign priority: <Q1 (Low); >=Q1&<Q2 (Medium); >=Q2&<Q3 (High); >=Q3 (Very High): Assign priority to set of requirements within the range above. Requirements within the range >=Q3 (Very High) have the highest priority while those in <Q1 (Low) has the least priority. Those in >=Q2&<Q3 have high priority (High) while those in >=Q1&<Q2 (Medium) have a medium priority.

## V. IMPLEMENTATION

To illustrate the QBP approach, CS scores from a recent Kano survey that captured requirements for the design of an e-health awareness system was used [4]. From the same survey [4] users' self-stated importance rating were elicited using a 7-point Likert scale ranging from (1) totally unimportant to (7) very important. The self-stated importance rating will be used as a sole aspect to regress on SI, DI and ASC to capture how important the requirements are to the potential users of the proposed system and to assess their perception of the quality of the intended product and then based on this, capture their priorities for the requirements.

Table 1
Requirements Categories from Kano Survey [4]

| Req | M | O | A | I | Category |
|---|---|---|---|---|---|
| Req1 | 11 | 14 | 16 | 06 | *A* |
| Req2 | 05 | 15 | 13 | 14 | *O* |
| Req3 | 07 | 22 | 11 | 07 | *O* |
| Req4 | 07 | 22 | 12 | 06 | *O* |
| Req5 | 04 | 18 | 06 | 18 | *O* |

Table 1 was derived from [4] and it represents the categorization of requirements based on the Kano theory/model. The SI, DI, and ASC in Table 2 were obtained from this table.

Table 2
Coefficients of customer satisfaction & users' self-stated requirements importance (IMP)

| Req | SI | DI | ASC | IMP |
|---|---|---|---|---|
| Req1 | .64 | .53 | .59 | 5.32 |
| Req2 | .60 | .43 | .52 | 5.28 |
| Req3 | .70 | .62 | .66 | 5.74 |
| Req4 | .72 | .62 | .67 | 5.52 |
| Req5 | .52 | .48 | .50 | 5.04 |

Using data [4] in Table 2, the following are the resulting models:

$$SI = -.864 + 279IMP \quad (10)$$

$$DI = -.855 + 259\text{IMP} \quad (11)$$

$$ASC = -.859 + 269IMP \quad (12)$$

SI, DI, and ASC were regressed with self-stated (requirements) importance (IMP). IMP was used as the only aspect to determine its influence on the importance value of quality attributes (that is, the influence of perceived importance of requirements on the satisfaction or dissatisfaction of users or customers, (if the requirements are met or unmet); and hence, by implication, its impact on the perceived quality of the would-be product). The three models (Equations 10, 11, and 12), produced similar results, resulting also in similar requirements ranking and prioritization. This result shows that any or all the models can be utilized in prioritizing software requirements. The results of the estimates, ranking and prioritization for SI, DI and ASC, are shown in Tables 3, 4, and 5 below.

Table 3
SI estimates with corresponding ranks & priority

| Req | SI | Rank | Priority |
|---|---|---|---|
| Req1 | .62 | 3rd | Medium |
| Req2 | .61 | 4th | Medium |
| Req3 | .74 | 1st | V. High |
| Req4 | .68 | 2nd | High |
| Req5 | .54 | 5th | Low |

Table 4
DI estimates with corresponding ranks & priority

| Req | DI | Rank | Priority |
|---|---|---|---|
| Req1 | .52 | 3rd | Medium |
| Req2 | .51 | 4th | Medium |
| Req3 | .63 | 1st | V. High |
| Req4 | .57 | 2nd | High |
| Req5 | .45 | 5th | Low |

Table 5
ASC estimates with corresponding ranks & priority

| Req | ASC | Rank | Priority |
|---|---|---|---|
| Req1 | .57 | 3rd | Medium |
| Req2 | .56 | 4th | Medium |
| Req3 | .69 | 1st | V. High |
| Req4 | .63 | 2nd | High |
| Req5 | .50 | 5th | Low |

## VI. DISCUSSION

Using Equation (10), the estimates of SI were ranked and prioritized. It follows that requirement 3 (Req3) has the highest rank with a very high priority. The least ranked requirement is requirement 5 (Req5) with a low priority. The second ranked requirement (Req4) has a high priority while the third and fourth ranked requirements (Req1 and Req2) are of medium priority. As with Table 3, Table 4 reveals similar results. Equation 11 was used in this case. The ranking and prioritization pattern is the same as in Table 3. The estimate of ASC was produced using Equation 12 and like the results in Tables 3 and 4, the ranking and prioritization of requirements were the same. The outcome from the three models produced consistent results. As can be seen from Tables 3 to 5. QBP approach is a novel method of prioritizing software requirements. It can handle multiple prioritization aspects. One aspect was used in this study for the purpose of illustration. This approach is effective, efficient, simple, easy to learn and use, and scalable where it has the capacity to prioritize several requirements.

## VII. CONCLUSION AND FUTURE WORK

This new approach to prioritizing software requirements stems from the need to consider prioritizing requirements from the view point of quality as perceived by users, customers and stakeholders. It is evidenced from this study that this approach is necessary to be co-opted as a prioritization approach. Future work will apply multiple aspects into the model and prioritize several requirements. Also, the approach will be automated to further enhance efficiency.

## REFERENCES


[1] Sher, F., Jawawi, D.N.A., Mohamad, R., Baber, .M.I. 2014. Requirements Prioritization Techniques and Different Aspects for Prioritization: A Systematic Literature Review Protocol. *2014 8th Malaysian Software Engineering Conference (MySEC)*: 31-36.

[2] Perini, A., Susi, A., Avesani, P. 2013. A Machine Learning Approach to Software Requirements Prioritization, *IEEE Trans. on Software Engineering*, 39(4), April: 445-461.

[3] Ejnioui, A., Otero, C.E., Otero, L.D. 2012. A Simulation Based Fuzzy Multi-Attribute Decision Making for Prioritizing Software Requirements. RIIT'12: *Proceedings of the 1st Annual Conference on Research in Information Technology*, ACM, Oct., New York: 37-42.

[4] Hussain A., Mkpojiogu, E.O.C., Kamal, F.M. 2015. Eliciting User Satisfying Requirements for an e-Health Awareness System Using Kano Model. *Proceedings of the 14th WSEAS International Conference on Applied Computer and Applied Computational Science*, 23-25 April, Kuala Lumpur, Malaysia.

[5] Park, Y.T., Jang, H., Song, H. 2012. Determining the Importance Value of Quality Attributes Using ASC. *Journal of Korean Society of Quality Management*, 40(4).

[6] Zhu, D., Lin, C., Tsai, C., Wu, J. 2010. A Study on the Evaluation of Customers' Satisfaction-The Perspective of Quality. *4th International Quality Conference*, Center for Quality, University of Kragujevac, 19 May: 309-324.

[7] Ma Q. 2009. The Effectiveness of Requirements Prioritization Techniques for a Medium to Large Number of Requirements: A Systematic Literature Review. *Masters Dissertation, Auckland University of Technology*, November.

[8] Khan, K.A. 2006. A Systematic Review of Software Requirements Prioritization. *Masters Thesis, School of Engineering, Blekinge Institute of Technology, Ronneby, Sweden*, Oct.

[9] Karlsson, L., Wohlin, C., Regnell, B. 1998. An Evaluation of Methods for Prioritizing Software Requirements. *Info. & Soft. Technology*, 39(14) & 15: 939-947.

[10] Elsood, M.A.A., Hefny, H.A. 2014. A Goal-based Technique for Prioritizing Requirements. *INFOS 2014*, 15-17 Dec, Cairo, IEEE: SW-18-SW-24.

[11] Berger, C., Blauth, R., Boger, D., Bolster, C., Burchill, G., DuMouchel, W., Pouliot, F., Richter, R., Rubinoff, A., Shen, D., Timko M., and Walden, D. 1993. Kano's Methods for Understanding Customer-defined Quality, *CQMJ*, 4, Fall:3-36.

[12] Kano, N., Seraku, N., Takhashi, F., Tsuji, S. 1984. Attractive Quality and Must-be Quality. Hinshitsu: *The Journal of the Japanese Society for Quality Control*, April: 39-48.

[13] Matzler, K., Hinterhuber, H.H., Bailom, F., Sauerwein, E. 1998. How to Delight your Customers. *Journal of Product and Brand Management*, 5 (2):6-18.

[14] Mustafa, B.A., Zainuddin, A. 2014. An Experimental Design to Compare Software Requirements Prioritization Techniques. *2014 IEEE International Conference on Computational Science and Technology*, Kota Kinabalu, 27-28 Aug:1-5

[15] Hatton, S. 2007. Early Prioritization of Goals. *Advances in Conceptual Modeling-Foundations and Applications*, 4802(235-244):517-526.

[16] Chen, C., Chang, Y. 2011. Prioritizing 5S Activities by Kano Model for a Semiconductor Water Fabrication. *Proceedings of the 4th WSEAS Int'l Conf. on Manufacturing Engineering, Quality and Production Systems*, Barcelona, Spain, 15-17 Sept:52-56.

[17] Otero, C.E., Dell, E., Qureshi, A., Otero, L.D. 2010. A Quality-based Requirement Prioritization Framework Using Binary Inputs. *IEEE 4th Int'l Conf. on Mathematical/Analytical Modeling and Computer Simulation*. Kota Kinabalu, Malaysia, 26-28, May: 187-192.