I.J. Education and Management Engineering, **2015, 3, 21-29**
Published Online July 2015 in MECS (http://www.mecs-press.net)
DOI: 10.5815/ijeme.2015.03.03

Available online at http://www.mecs-press.net/ijeme

Architecture Design Selection Scheme for Usability Quality Attribute

Rupesh Nagendra[a]

[a] *Assistant Professor, Information Technology Department*
Shri Vaishnav Institute of Technology and Science Indore,M.P.,India

Abstract

Usability quality attribute is one of the important quality attributes because it is a basic need for end-user stakeholder. Usability provides the ease of use and learnability to end user. Usability is very useful quality attribute of software architecture and architect should remember about usability aspect. Usability aspect means such type of software architecture which provides the learnability, memorability and performance. In this paper we discuss software architecture design based on usability aspect. Usability is necessary for end-users and business stakeholders. We measure and evaluate the usability through one of the mathematical equations. For the selection of usability aspect, or to evaluate the highest usability score in different architecture designs, we have taken the questionnaire from technical persons on the basis of nonfunctional requirements or sub-characteristics of usability quality attributes such as learnability, memorability and performance. Then finally we calculate the usability score.

Index Terms: Architecture design, Learnability, Quality Attributes, Software Architecture and Usability.

1. Introduction

Usability aspect is the basic need of end-users as they all need the various usability aspects like user interface, look and feel, learnability, memorability and quick response (like performance). Learnability, the proposed system in one way or the other viz., end users, developers, maintenance engineers, architects, business people etc. Their expectations and requirements are converted into functional and non- non-functional requirements of the proposed system. However, sometimes architect gets confused with the requirements. Functional attributes anyhow can be determined easily, but determination of quality attributes requirement is difficult. Architect thinks in different scenarios. In this paper a hypothetical mathematical equivalent method [7] is proposed, in which the appropriate result through architecture design score is determined.

* Corresponding author.
E-mail address: rupeshnagendra@gmail.com

2. Related Work

4+1 is a view model designed by Philippe Kruchten for describing the architecture of software-intensive systems, based on the use of multiple, concurrent views. The views are used to describe the system from the viewpoint of different stakeholders, such as end-users, developers and project managers. The four views of the model are logical, development, process and physical view. In addition selected use cases or scenarios are utilized to the architecture serving as the 'plus one' view. This model is useful for functional requirements and non-functional requirements. We can say, the industry uses this model for testing of non-functional requirements with functional requirements.

In all the four views of the model logical, development, process and physical view, the ability of particular quality attributes is checked. Tradeoff Analysis Method (ATAM) [3], Software architecture Analysis Method (SAAM) fail to quantitatively analyze software architecture structures based on quality attributes. The proposed method creates a support framework using Analytical Hierarchy Process for comparison of different software architectural structures for a specific software quality attribute. Moreover, given a prioritization of quality attributes for the software system, or a part thereof, the most suitable software architecture structure can be indicated using the created framework.

3. Proposed Quality requirement of usability

Quality requirement of usability specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. This should be contrasted with functional requirements that define specific behavior or functions. The plan for implementing functional requirements is described in the system design. The quality requirements that we have considered are as follows:

1) *Learnability*: To understand any system by an end-user, system must be learnable, otherwise usability will get decreased.
2) *Memorability*: System should be developed in such a manner that it can be memorized easily so that end-user is comfortable with it.
3) *Performance*: Whenever any end-user sends a request, quick response is needed. Response can be in the form of output, error message or waiting.

4. Scenarios

A scenario is a description of a person’s interaction with a system. Scenarios help to focus on design efforts on the user’s requirements, which are distinct from technical or business requirements. Scenarios may be related to ‘use cases’, which describe interactions at a technical level. Unlike use cases, however, scenarios can be understood by people who do not have any technical background. They are therefore suitable for use during participatory design activities.

In the proposed approach scenarios as software design are taken. Scenarios are widely used by organizations of all types to understand different ways that future events might unfold. Scenario planning or scenario analysis is a complex business process related to future studies.

5. Proposed work and result

The series of steps involved in the selection process are as follows:

(i). Identify stakeholder‘s quality requirement.
(ii). Normalization.

(iii). Identify the quality attributes score from technical stakeholders.
(iv). Calculating the score of software architecture design with hypothetical equivalence approach.
(v). Selection of the Architecture design through mathematical approach.

(i). Identify stakeholder‘s quality requirement-

For developing any type of software application, there is a need of some quality attributes, also a lot of stakeholders are involved and every stakeholder’s requirements addressed the quality attributes. In table 1, we illustrate the quality requirements needed by the various stakeholders. Manager thinks about the cost, team size (man power) and time, Technical person requires the quality attributes such as Maintainability, Reliability and Reusability and End - user requires Learnability, Memorability and Performance.

Table 1. Quality requirement of stakeholders

Stakeholder	**Quality Attributes**		
Manager	Cost	Team Size	Time
Technical Person	Maintainability	Reliability	Reusability
End-User	Learnability	Memorability	Performance

(ii). Normalization

When aggregating quality attributes that have different units of measurement, normalization is a common way to eliminate the problem. During normalization, the lowest range value is usually taken as 0 and the highest range value as 100.

(iii). Identify the quality attributes score from technical stakeholders

The end-user (client) stakeholder firstly meets with the manager and technical person stakeholder and has told the requirements of that project or problem. Architect gathers the requirements from end users. On the basis of understanding of the problem, Technical stakeholders who are the architecture designers, collectively decide that which quality attribute should be present in what percentage. The technical person is expert on his field, so that the organization can put questionnaire from them. But view of all experts might have some conflict; in that case best architecture design is determined.

For that purpose the four Architecture designers for usability illustrated in table 2. The table 2 is created between architecture designers and quality attributes. In this table technical person keeps his view in terms of quality attributes. The technical persons have created the architecture design according to quality attribute score. The quality attributes have normalized in 0 to 100, and total score should not be greater or less than 100.

Table 2. Quality Attribute score from Stakeholder

Architecture Designer **Quality Attribute**	**Technical person1 (v1)**	**Technical person2 (v2)**	**Technical person3 (v3)**	**Technical person4 (v4)**
Learnability (w1)	35	40	65	60
Memorability (w2)	35	20	25	35
Performance (w3)	30	40	10	5
Total score	35w1+35w2+30w3	40w1+20w2+40w3	65w1+25w2+10w3	60w1+35w2+5w3

(iv). Calculating the score of software architecture design with hypothetical equivalence approach

The hypothetical equivalence approach determines the attribute weights using a set of preferences rather than selecting weights arbitrarily based on intuition or experience. The approach is based on developing a set of hypothetical alternatives that the decision maker is indifferent between. In other words, it is based on identifying hypothetical alternatives that have equal value to the decision maker. The approach is best illustrated through the use of an example: Consider the group of technical persons and their three quality attributes of interest viz., Performance, Learnability and Memorability. However, the technical person's group can also consider other pertinent metrics for design quality viz., cohesion and coupling. At times, the quality attributes chosen by the group may be conflicting and this would be reflected in the architecture's support for quality. Four hypothetical architecture Design (Technical person1, Technical person2, Technical person3, and Technical person4), values are prepared and presented to the group as in Table 2.

The group of architecture designers feel that technical person1 is equivalent to technical person2, and technical person3 is equivalent to technical person4. Total score as given in table 2 result in the following equations:

$$35w1+35w2+30w3w3 = 40w1+ 20w2 +40w3 \quad (1)$$

$$65w1+ 25w2 + 10w3 = 60w1 + 35w2 + 5w3 \quad (2)$$

The normalization equation is

$$w1 + w2 + w3 = 1 \quad (3)$$

By solving the three equations, we got the weights which are fixed as:

$$w1 = 0.33,\ w2 = 0.33,\ \text{and}\ w3 = 0.34$$

Hence, the weight of preference is 0.33 for learnability, 0.33 for memorability and 0.34 for maintainability. Similarly weight of preference for the quality attributes of the roles viz., Manager and end-user are computed.

(v). Selection of the Architecture design through mathematical approach

For all candidate architectures, the quality attribute values of each group are normalized using the corresponding strength of preference graphs. As an example, quality attributes of the group of technical persons are normalized and tabulated in Table 3. The total score of each of the architectures is computed by aggregation of the row values. Before aggregation, the row values have to be multiplied by their corresponding weights. Similarly, Value Tables for the other groups viz., User and Manager are constructed.

The value table of each of the groups is examined and the total score of each of the architectures is aggregated to obtain the score of architectures design:

Total score of

$$\text{Architecture Design Score} = \sum_{i=1}^{j} (\mathrm{W_i} * \mathrm{V_i})$$

Here,
i= initialized by l
Where
w_i = Weight of the particular attribute in
v_i = Value of the attribute

j = Number of attributes in the table

From table 3, it is assumed that different architecture designers propose different architecture designs. Technical person 1 proposes architecture design 1, technical person 2 proposes architecture design 2 and so on. Architecture Design Score is illustrated in table 4. Architecture designer select the architecture design. From the table 4, it is evident that architecture design 4 gives the maximum satisfaction to the stakeholders. From hypothetical equivalence approach [7] architecture design 4 provides maximum satisfaction for usability.

Table 3. Architecture Design Score

Architecture Designer **Quality Attribute**	**Technical person1 (v1)**	**Technical person2 (v2)**	**Technical person3 (v3)**	**Technical person4 (v4)**
Learnability (w1 = 0.33)	35	40	65	60
Memorability (w2 = 0.33)	35	20	25	35
Performance (w3 = 0.34)	30	40	10	10
Total score	35w1+35w2+30w3= 33.3	40w1+20w2+40w3= 33.4	65w1+25w2+10w3 = 33.1	60w1+35w2+10w3= 34.75

Table 4 contains the results scores between different types of architecture design in which data is given by technical person. Then Architecture design1 has got 33.3, architecture design 2 got 33.4, architecture design 3 got 33.1, architecture design 4 got 34.75. Figure 2 is same as Figure 1 but in this figure red color point denotes maximum score of architecture design. It also shows architecture design 4 has got maximum score 34.75. Now architecture design 4 for usability which is prepared by technical person 4 has got maximum score, it means that architecture design 4 has highest usability score. Table 4 illustrates architecture design scores.

Table 4. Final Architecture Design Score for Usability

Architecture Design	**Score**
Architecture Design1	33.3
Architecture Design2	33.4
Architecture Design3	33.1
Architecture Design4	34.75

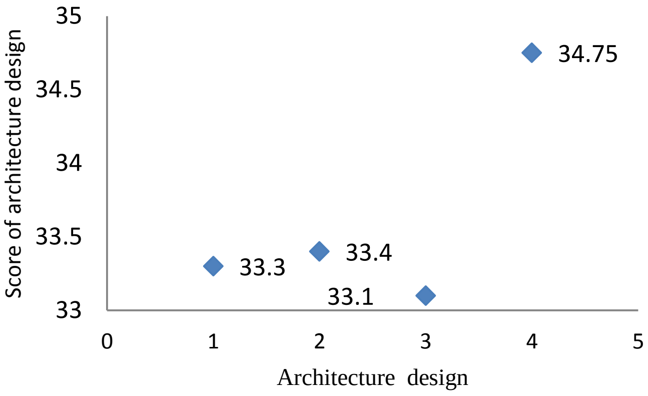


Fig.1. Architecture design score

In figure 1, it has shown the score of all type architecture design, architecture design 1, architecture design2, architecture design 3 and architecture design 4.

In figure 2, it is illustrated the maximum score in all four architecture design, finally architecture design 4 has got maximum score. It means that for particular problem, usability is best in architecture design 4.

In figure 3, It is shown that how any software product fails if it does not support the usability aspect. Figure 3 also shows that software product does not provide good user interface, learnability, performance, look and feel. From above reason end-user is not interested to use a particular software product,. This will cause a big loss to business stakeholers.

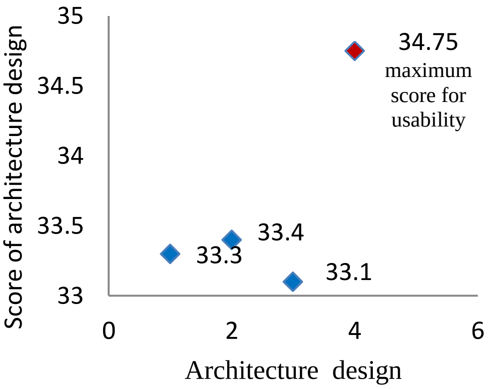


Fig.2. Maximum score for usability

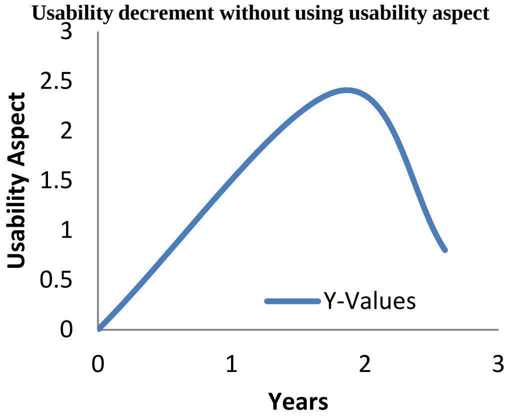


Fig.3. decrement of usability without using usability aspects

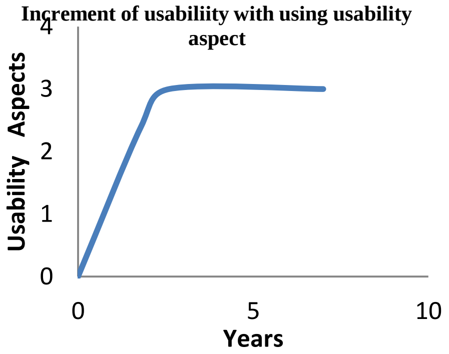


Fig.4. Increment of usability with using usability aspects

Figure 4 shows the increment life of software product. It also illustrates that the usability aspects are supported in a particular software product. End-users are interested to use such a software product. So bussiness stakeholers will get high profit.

6. Implementation Approach

We have developed a form in which technical persons can fill their views in terms of quality attributes. After that it submits on database. This form is illustrated in figure 5. Then we calculate the program and result displays the architecture score. This is illustrated in Figure 6.

To get the result as illustrated in figure 6, we have developed a class to solve the equations but they can be solved manually too. It means that this program is specific. Finally we get result; architecture design 4 has got maximum score in usability. In usability design for particular problem, the quality attribute score should be learnability (60%), memorability (35%) and performance (10%).

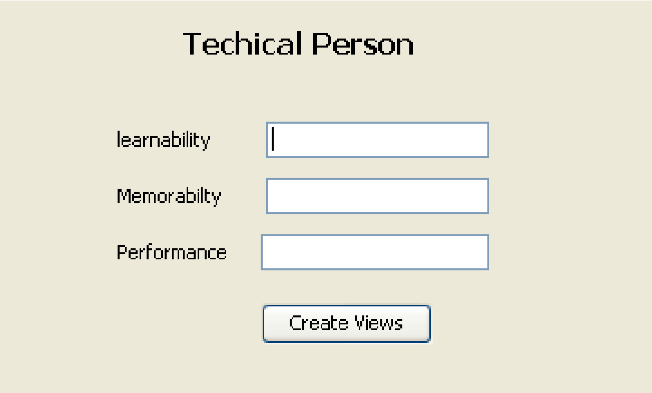
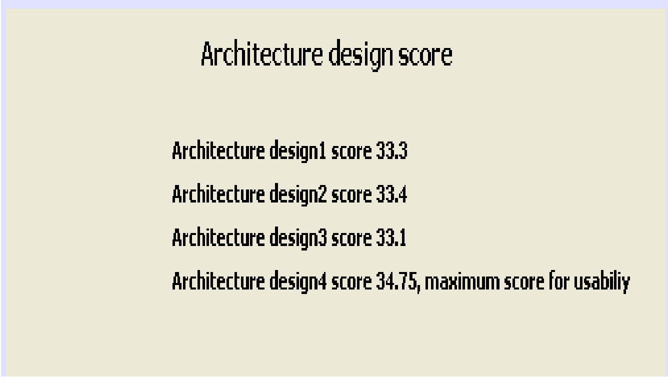


Fig.5. Form put the entry for quality score by Technical person


Architecture design score

Architecture design1 score 33.3

Architecture design2 score 33.4

Architecture design3 score 33.1

Architecture design4 score 34.75, maximum score for usabiliy

Fig.6. Architecture design score for usability

7. Conclusion

Choosing an architecture that satisfies the quality attribute requirements is vital to the success of any software architecture. Nonetheless, the requirements that will shape the architecture design are the quality attribute requirements. We identify stakeholder‘s quality requirements and also identity the quality attributes score from technical stakeholders. Here we have proposed the Architecture design selection approach for usability, it has very simple process and it is easy to find the architecture design which is suitable for usability quality attribute. This method gives response faster than 4+1 view architecture. We get the architecture design rank with the mathematical hypothetical equivalence approach. According to usability score we get architecture design rank, and with the help of rank, we get the best architecture design.

In future, we should research for software architecture design selection with cost benefit analysis methods. The usability rank is obtained through the non-functional requirement. In future we should work with functional requirements also.

References

[1] M. Svahnberg et al, "A Method for Understanding Quality Attributes in Software Architecture structures", ACM transactions (2002).

[2] Liliana Dobrica et al, "A Survey on Software Architecture Analysis Methods", IEEE transaction on Software Engineering volume 28 (July 2002).

[3] Kazman et al., "Architecture Tradeoff Analysis Method", Technical report, Carnegie Melon University, Software Engineering Institute (1998).

[4] Kazman et al, "Quantifying the Costs and benefits of Architectural Decisions", IEEE 2001.

[5] Eelko K.R.E. Huizingh and. Hans C.J. Vrolijk, "Decision Support for Information Systems Management: Applying Analytic Hierarchy Process", Research school systems Organization and management (SOM) research report number 95B26, University of Groningen (1995).

[6] Axel Volker et al, "Requirements Interdependencies and Stakeholder's preferences", Proceedings of IEEE joint international conference on requirement Engineering (2002).

[7] Tung-King See et al, "Multi attribute Decision Making Using Hypothetical Equivalents", Proceedings of Design Engineering Technical Conferences and Computers and Information in Engineering Conference ASME (2002).

[8] Chen, Wand Wassenaar H.J., "An approach to decision-Based design", ASME design Technical conference Pittsburgh, Pennsylvania (2001).

[9] Alison R. Callaghan, "A 2-Phase Aspiration-Level and Utility Theory Approach to Large Scale Design" Proceedings of DETC '00, ASME 2000 Design Engineering Technical Conferences and Computers and Information in Engineering Conference Baltimore, Maryland, September10-13, (2000).

[10] Van Lamsweerde R et al, "Goal-Oriented Requirements Engineering: A Guided Tour", Proc RE'0l-International Joint Conference on Requirement Engineering, Toronto, IEEE, PP249-263 (2001).

[11] Barry Boehm et al, "Win Win Requirements Negotiation Process: A Multi Project Analysis", Proceedings of the 5th international conference on software process (1998).

[12] Farcisca losavio et al, "Quality Characteristics of Software Architecture", Journal of object Technology Volume 2 (2003).

[13] H. Reza, and E. Grant. Quality Oriented Software Architecture. The IEEE International Conference on Information Technology Coding and Computing (ITCC’05), Las Vegas, USA, April 2005.

[14] Quality Assurance Institute, “The top ten challenges of software project management,” www.qaiusa.com.

[15] www.solutionmatrix.com; “Return on investment: What is ROI analysis?”

Author(s) Profile

Rupesh Nagendra is an Assistant Professor in Information Technology Department at Shri Vaishnav Institute of Technology and Science Indore. He studied M.E. in Computer Engineering Department at, Shri G.S. Institute of Technology and Science, Indore and B.E. in I.T. from Madhav Institute of Technology and Science, Gwalior, India. Email: rupeshnagendra@gmail.com

How to cite this paper: Rupesh Nagendra,"Architecture Design Selection Scheme for Usability Quality Attribute", IJEME, vol.5, no.3, pp.21-29, 2015.DOI: 10.5815/ijeme.2015.03.03