Achieving the Benefits of Agility in Software Architecture-XP

Zafar Karimi[1], Sajjad Behzady, Ali Broumandnia[3]

[1] Graduate Student of Software engineering, Islamic Azad University at Qazvin, Iran
zk_karimi2006@yahoo.com,
[2]Software engineer Graduate, Islamic Azad University at Zanjan, Iran
Sajjad_behzady@yahoo.com
[3]Assistant Professor of Computer Engineering Faculty,
Islamic Azad University at Qazvin, Iran
broumandnia@gmail.com

ABSTRACT

One challenge of XP agile method in software development is this method's underestimation over software quality attribution; these attributes are of main indices of software architecture. In this study, a method is represented for responding this challenge on the bases of probability theory. In this method, firstly, the rating matrix is structured on the bases of quality attributions an architecture solutions. Each element of this matrix shows a rating for every solution and the ratings will be initialized through analytic hierarchy process (AHP). Via quality weight implement on above matrix the rating vector is created whose each element represents the rate of each solution in reaching those quality attributes in rating matrix. Because rating vector follows normal probable distribution, its elements' rating probable density is mostly gathered around the mean.

In this study, the probable density of architecture will be defined, the appropriateness of a solution in comparison to other solutions will be evaluatedor and the respond to XP challenge will be done easily in order to reach architecture solutions in the frame of rating vector. A complex of data are gathered and the results are compared in an experimental method in order to investigate the represented method of this paper.

KEY WORDS

Quality Attributes, Architecture Solutions, Normal Distribution and Analytic Hierarchy Process AHP

1. INTRODUCTION

XP software development process in several publications and will be repeated and at the end of each iteration, a set of system requirements will be implemented. In each iteration, two-person teams or pairs, programmers are working to implement new functionality so that the customer is in the form of a user. Each pair of programmers, receive system functionality as an input. Then paired to create unit tests for functional called the action. The development is done in the existing codes may be used by programmers. When implementing the functionality ends, will be integrated with existing code. Since this process in parallel by different couples, and quality

DOI : 10.5121/ijcsit.2012.4509

control is done, get a great strengthen the architecture of weak structure that influence the quality attributes of software, the software architecture can be extremely difficult. Gradual increase over these weaknesses reforms and makes unmanageable the system architecture. Thus, this need is felt that the work done from the perspective of software architecture in XP[1]. Further efforts are to be achieved through tasks and activities of the XP process: 1) Classification of quality attributes and architectural approaches to calculate their value and functionality for the system matrix: 2) A mathematical format - reforming and restructuring of the architecture. In this regard, an architect with knowledge and personal experience, the qualitative features of the software at any stage of the process, and will to achieve them in the system, with the help of other professionals, it offers a better approach.

More of this paper is structured. In part 2 basic concepts and related works are discussed. Part 3 examines the characteristics of quality, architectural approaches and how they relate to each other. In Part 4 of a concern with the proposed method has been checked and in Part 5 and Part 6 study of a concern to continue the discussion and conclusions and recommendations.

2. BASIC EXPRESSIONS AND RELATED ACTIVITIES

The benefit of software architecture in achieving agility in the process is a valuable activity. In some articles, how to find appropriate and effective quality attributes in software systems has been discussed. And in some other types of methods and techniques that agile methods has been investigated as a way to empower the perspective of software architecture. in [25] all required quality of software has been discussed and the mechanisms of how they are selected by the Traffic Study is a qualitative traits. The approaches to agile methods in order to provide a way to empower the perspective of software architecture, two methods are developed as follows:

- Software architecture approach to integration method with XP: article [10] is trying to obtain software architecture benefits by adding existing architecture methods to XP method such that these benefits are achieving quality characteristics, developing a conceptual model from software and proper relation between software Stakeholders. In this approach QAW[2], ADD[3], and ATAM/CBAM methods are added to XP[1] method as individual and default modules. QAW is workshop in order to achieve a better understanding of the qualitative characteristics. ADD is proposing to design and architecture in ATAM along with CBAM to evaluate the architecture. These procedures are performed at certain times of the project.

- The approach of the developer's architectural activities: This approach opposes using existing techniques in software architecture agile environments and it looks to be the new perspective of these activities and the architecture is designed for the XP and development team that worked on XP does, of these activities must use in the development process.

2.1. Agility

Agility emphasizes on the ability of individuals to respond to changes and less attention to planning and providing a detailed plan. This does not mean that the methods considering agility do not plan, but plan is a way of dealing with change and gain the flexibility. In fact, when the environment changes so transcends, other methods are not responsive to the needs of propeller and in these circumstances, agile methods, are the only ways to success. Despite the challenges of software architecture that is agile methods, these methods are also considered to bring benefits, so the value is still as efficient and effective methods are introduced.

3. MATRIX REPRESENTATION OF QUALITY ATTRIBUTIONS AND ARCHITECTURAL SOLUTIONS

In this paper the relation between quality attribution and architecture solutions are shown in matrix form. The rows show the quality attributions and the columns show the solutions. The amount of each matrix entry is a number between +9 to -9 that on the basis of AHP[4] method the effect of solution in providing quality attribution is shown by a number from +1 to +9 (table.1)[14,24]. 1 shows the least positive effect and 9 the most positive effect in providing quality attribute. As in [22] attentions are paid to agreement and disagreement of some quality attributes e.g. efficiency increase will decrease reformation. So in this paper -1 to -9 is introduced for negative effect of solution on attribution. -1 shows the least negative effect and -9 the most negative effect of a solution on an attribution.

Table.1: Valence on the Bases of AHP Method

Numerical Effect	Verbal Effect
+1 to +9	Attribution provision from least to most with positive effect
-1 to -9	Attribution provision from least to most with negative effect

Every software attribution is of weight of 0 to 1 and indicates its importance in software. Also completion weight of all software attribution equals 1. (Attribution weight is normalized).

Table.2: Matrix representation of quality attribution and system responsibility solution

	Solution 1	Solution 2	…..	Solution m
Attribution 1				
Attribution 2				
……				
Attribution n				

4. PROBLEM DESCRIPTION

In XP, if the process is that each system requires n functional attribute and m approach is proposed to achieve this quality traits in software, So that each approach had a positive impact on one or more traits and trait negative Affect on one or more then we want to provide a numeric criteria in order to minimize the architecture designing risks and choose an approach among m proposed approach for reconstruction and development process architecture in XP.

4.1. Introducing Utilized Symbols

In this section the notation used in this paper have been introduced.

Table.3: Utilized Symbols in This Paper

n	The number of quality attribution for responsibility
m	The number of architecture solutions (decisions) for system responsibility
W	Matrix n * 1 weight of software quality attributions
W^T	Transposed matrix of W
X_i	Matrix m* n the amount of quality attribution via responsibility solutions of i
X_{IJ}	J_{th} vector of X_i matrix (shows the rate of j_{th} solution on quality attribution without counting attribution weight)
G_i	Matrix m * 1 shows the importance of responsibility solutions I system on software
G_{xi}	X entry of matrix G_i
μ_i	The mean of rating responsibility solutions I system
δ_i	Covariance of G_i vector
F (G_{xi})	Rating density implement on x responsibility solution I process on software
D_i	1*m matrix shows the importance of system I software architectural approach
D_{xi}	X element of D_i matrix

4.2. Proposed Solution

For the solution of the following procedures are used:
A. According to the fact that each solution is of a rating in providing quality attribution, the product of solution ratings in attributions' weight results in matrix G_i.

$$G_{i=W}{}^{T}X_i \qquad (1)$$

Matrix G_i shows the importance or rating of responsibility solutions I and is called rating vector. Each one of its entry indicates a solution in providing all quality attributions of responsibility and the more the amount is, the more the rating density is. If matrix Gi is calculated for several different solutions and same software, through their propriety the best solution is gained for each software responsibility.

If the mean of matrix G_i is called μi, the mean is calculated through formula (2). Covariance or difference of mean for G_i is showed by δ_i.

$$\mu_i=E(G_i) \qquad (2)$$

Possible density of G_{xi} is defined propriety of matrix G_i and calculated through formula (3).

$$f(G_{xi}) = \frac{e^{\frac{-(G_{xi}-\mu_i)^2}{2\sigma_i}}}{\sqrt{2\pi}} \qquad (3)$$

In formula (3), f (G_{xi}) is a numeric which shows the solution rating on the responsibility of X_i. In other words, f(G_{xi}) shows the suitability of solution which is called solution propriety and the bigger this numeric is, the more density is the solution.

B. Note that each approach has value in achieving a quality attribute. Approaches values multiplied by the weight traits, the resulting matrix gives D_i.

$$D_{i=W}{}^{T}X_i \quad (4)$$

D_i matrix shows the importance of attitudes and values of the matrix, we call it a vector. If D_i matrix calculates for several different architectures or for software, by fitting them, the aim of this study is that researchers will find the best D_i, will achieve.If the architecture D_i Average call μ the mean value of the matrix equation (5) is calculated. difference or average covariance of the covariance matrix Z are both arrays D_i and the formula (6) and (7) is calculated.

$$\mu = \left| \begin{matrix} E(D_{i1}) \\ \dots \\ E(D_{im}) \end{matrix} \right| \quad (5)$$

$$Z = \left| \begin{matrix} \sigma_{11} & \dots & \sigma_{1m} \\ \dots & \dots & \dots \\ \sigma_{m1} & \dots & \sigma_{mm} \end{matrix} \right| \quad (6)$$

$$\sigma_{ij=}\mathrm{Cov}(D_i,D_j) \quad (7)$$

D_i fitted probability density of the architecture definition D and formula (8) is calculated.

$$f(D_i) = \frac{1}{(2\pi)^{\frac{m}{2}}|detZ|^{\frac{1}{2}}} e^{\left[-\frac{1}{2}(D_i-\mu)^T Z^{-1}(D_i-\mu)\right|} \quad (8)$$

In formula (8), D_i formula (4) is calculated, Z^{-1} inverse matrix Z, Z are matrices determining the absolute. Also $f(D_i)$ is a number that the value of the software architecture represents the i. In other words, $f(D_i)$ of the suitability of the architecture that we call it a fitting architecture and architectural dense will be bigger. The solution to this is that, to reduce the risk of architectural design, the architect of the first developers to offer a basic architecture and the architect of the proposed architectures and using formula (8) to choose the most appropriate one. To develop the basic architecture of choice, every couple will have responsibility for program development at the end of each functionality, a simple conceptual models created and sent to the architect. Architecture models and gathers them together to present the conceptual structure of the system. The structure of the system, the qualitative features of the system is functional and qualitative scenarios as they storm a meeting of the minds of customers with at least some of the programmers. Overview of the scenario, the quality of the session, several approaches to achieve each of the functional characteristics of quality in software production, the proposed architecture. The architect of the proposed approaches and Specialty Table 3, each of the functional quality of the system using AHP, in the form of a matrix are closely related. Using formula 3 in the most appropriate approach for each value of vector and provides the functionality. The process of restructuring and reforming the system architecture are done with activities in XP.

4.3. G_i Normal Distribution Confirmation

If it is assumed that X_{ij} or every vector of j is a random sample, every entry of this vector will be of monotonous distribution in (+9, -9). So, X_{ij} are of monotonous distribution as random variables of n variables. On the bases of centric matter [28], each entry of G_i which is caused through formula (1) is a random variable that is produced from dependent and distributed variables so they will be of normal distribution. According to normal vector definition, G_i is normal probable distributed, of μ_i mean and is calculated through formula (9):

$$\mu_i = \frac{1}{m} (G_{i1} + G_{i2} + \ldots + G_{im}) \qquad (9)$$

δ_i calculating formula (10) and putting it in the formula (3) fitted to the proposed approaches to meet the quality characteristics of functionality assigned to each pair, the value f (G_{xi}) will be available.

$$\sigma i = \frac{1}{m-1} \sum_{j=1}^{p} \left(G_{ji}\_\mu_{i} \right)^{2} \qquad (10)$$

Note that the condition of formula (3), G_i is a normal distribution. The property of normal distribution [28], the probability density of random variables that are close to average, are more. So we can expect that the approach is more likely that the average density value is closer. Thus, the fitting approach, the approach that is closer to the average of the highest density and the average will be the answer. Normal distribution of D_i would also like to prove the normal distribution.

5. A CASE STUDY OF A PROBLEM

The experimental approach is proposed to fit a few basic architecture and functionality of the system architecture is examined in detail for their XP process to various aspects of the proposed method is more evaluated.This issue is related to a software manufacturer that currently produces commercial-industrial applications. Because of experience with this software, there's an overview of the analytical method for its production is to be done.

5.1. Quality Attribution of Software

For software [18] a series of under mentioned qualities are demanded:

- **Flexibility from Performance Points:** This quality attribution is called A_1, the architecture solution which can be attributed to this quality. Separation is representing service from data and is named D_1.

- **Applicability from Performance Velocity Point:** This attribution is called A_2. The solution that can be proposed to provide this attribution is replicate of data accessibility factors in different places (D_2) or utilizing for parallel processing of commands (D_3).

- **Applicant Accessibility Safety:** Safety is a basic factor in applicant confidence in trade software systems and factors that try to represent accessibility to applicants by hiding some data should be searched in wide level of architecture. This attribution is called A_3 and creating factors in order to control accessibility to information banks and proposed solution system is named D_4.

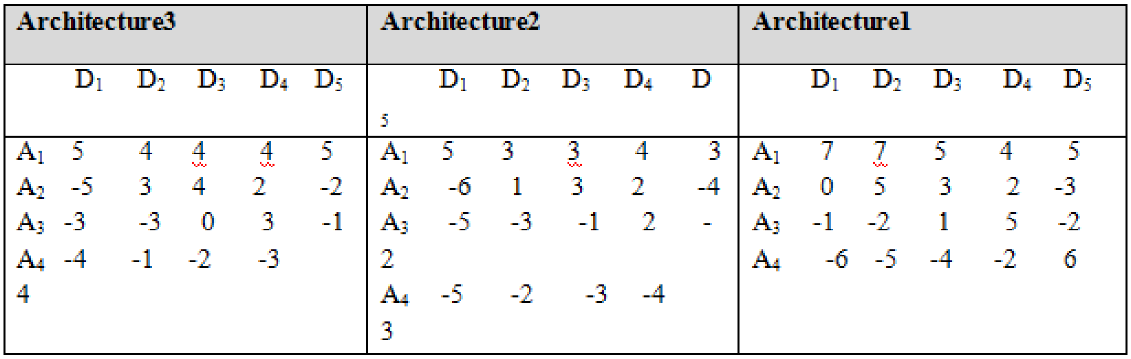
- **Time Cost:** This quality attribution is called A_4 and utilizing packs and completions of prepared codes as proposed solution is D_5.

5.2.Provide more functionality for a few basic architecture and process architecture composed

Based on quality traits expressed, three types of architecture first and second functional difference, considering that (assuming that all sent and received between the architect and the couple has been done.) That the experience of the authors and consult with experts for architectures the initial value of the matrix (Table 4) and for each functional, matrix of table 5 is presented. Each of the functional is based on capabilities of VB.NET Also expressed for traits, attributes the weight matrix W is represented in the W_1^T and W_2^T for the functionality of the architectures we consider. The amount of discretion and the importance of software quality attributes are selected.

W_1^T=[0.3,0.2,0.2,0.3]
W_2^T=[0.6,0.4]

Table.4: Rating Matrix of Candidate Architectures



Architecture3						**Architecture2**						**Architecture1**					
	D_1	D_2	D_3	D_4	D_5		D_1	D_2	D_3	D_4	D_5		D_1	D_2	D_3	D_4	D_5
A_1	5	4	4	4	5	A_1	5	3	3	4	3	A_1	7	7	5	4	5
A_2	-5	3	4	2	-2	A_2	-6	1	3	2	-4	A_2	0	5	3	2	-3
A_3	-3	-3	0	3	-1	A_3	-5	-3	-1	2	-2	A_3	-1	-2	1	5	-2
A_4	-4	-1	-2	-3	4	A_4	-5	-2	-3	-4	3	A_4	-6	-5	-4	-2	6

Table.5: Rating Matrix of Candidate Responsibilities

Responsibility 1						**Responsibility 1**					
	D_1	D_2	D_3	D_4	D_5		D_1	D_2	D_3	D_4	D_5
A_1	5	2	1	1	2	A_2	-6	3	1	2	2
A_3	-5	3	2	3	0	A_4	2	- 5	5	1	0

5.3. Solving Case Problem:

First, choose one of the proposed architectures, the D_i using the formula (4) architecture, then the average value μ, the formula (5) are calculated.

D_1=[0.1,1.2, 1.1, 2, 2.3]
D_2=[-2.2,-0.1, 0.4 ,0.8,0.6]
D_3=[-1.3,0.9,1.4,1.3,2.1]
μ=[-1.133,0.667,0.967,1.367,1.667]

Based on values calculated, plot the values of D_i, or the value of the solutions proposed by the architecture with the architecture of the mean μ, in Figure.1 is drawn. As can be seen, the architecture a maximum difference of positive relative to the architecture in there. The architecture has three closest values were averaged to architecture and the architecture of the second largest negative difference is in the architecture.

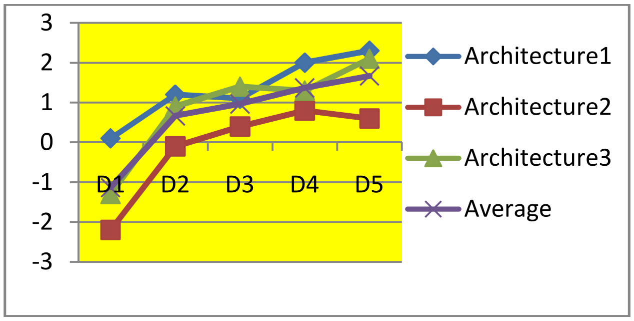


Fig.1: The Rating Graph of Responsibility 2 Solutions

If the values of Z and f (D_i) are according to formulas (8) and (5) following amounts are to be calculated.

$$Z = \begin{bmatrix} 1.343 & 0.505 & 0.28 & 0.466 & 0.664 \\ 0.505 & 0.463 & 0.202 & 0.263 & 0.419 \\ 0.28 & 0.202 & 0.263 & 0.145 & 0.292 \\ 0.466 & 0.419 & 0.292 & 0.345 & 0.863 \\ 0.664 & 0.419 & 0.292 & 0.345 & 0.863 \end{bmatrix}$$

The density values of each of the above architectures will be:

Table.6: The Density of Responsibility Solutions

$F(D_1)$=0.076	$F(D_2)$=0.068	$F(D_3)$=0.084
F(μ)=0.161		

As can be seen and Architecture 3 has the highest density value of 0.084 is the lowest density 3 .The proposed architecture is the architecture of the three best choices. With these qualities, as the architect of the architecture need three basic architecture selection and system developers can provide. Then the evolution of development work, the following activities performed.

At first the functional G_i software using the formula (1) and then calculated the average value μ_i them, according to formula (2) are calculated

$\mu_{1=1.56}$ و G_1=[1 2.4 1.4 1.8 1.2]

$\mu_{2=0.48}$ و G_2=[-2.8 -0.2 2.6 1.6 1.2]

Based on the calculated values, or value of the approach vector diagram G_{xi} values with average values for each functional form μ_i (1) and (2) is drawn. As can be seen two of the functional approach to a more positive difference to the average. The third approach is to have the closest values to the mean. In the second approach, a maximum difference from the average itself is negative.

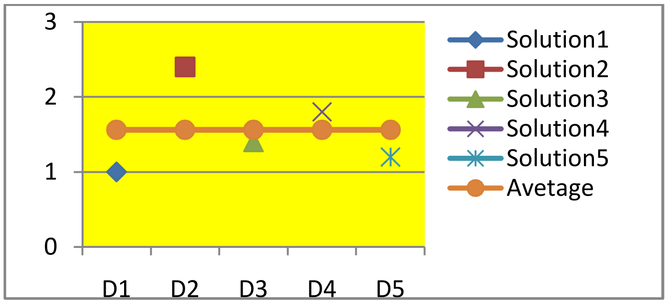


Fig.2: The Rating Graph of Responsibility 1 Solutions

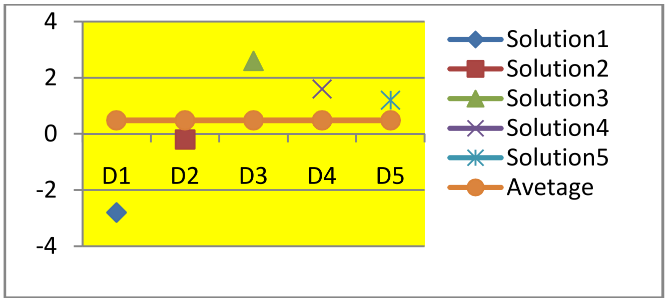


Fig.3: The Rating Graph of Responsibility 2 Solutions

If the values and f (G_{xi}) are according to formulas (10) and (3) are calculated, the results are the following values:

σ_2= 4.32و σ_1=0.308

The value of the density functional approach for each of the above is:

Table.7: The Density of Responsibility Solutions

Responsibility 1	Responsibility 2
$f(G_1)=0.248$ $f(G_2)=0.131$ $f(G_3)=0.392$ $f(G_4)=0.374$ $f(G_5)=0.331$	$f(G_1)=0.119$ $f(G_2)=0.397$ $f(G_3)=0.244$ $f(G_4)=0.353$ $f(G_5)=0.384$

As can be seen in the functional approach, the third highest density of 0.392 and a second approach has a lower density would be worthless. The second approach is the second highest density functional approach a value of 0.397 and 0.119 is the lowest density value. The second approach 2 and approach the task from the task from a choice of three approaches have proposed. So the two functional architecture of this system will be as follows:

Table.8: Chosen Architecture of System Represented Responsibilities

Architecture	
Responsibility 1	**Responsibility 2**
D_3	D_2
A_1 1 A_3 2	A_2 3 A_4 -5

As it is shown in table.7, according to rating density amounts solution 3 of responsibility 1 and solution 2 of responsibility 2 are as the best solutions and form table.8 system architecture.After it became clear that the functional architecture, the architect of the original architecture and architecture combine to create credit. Finally, the architecture is composed of a meeting of the minds of the storm with minimal client, architect, with some programmers and system performance will be held. At the meeting with your clients meet the qualitative characteristics of offers and if you do not consent, participants consult with each other to engage in tactics to achieve it.

6. CONCLUSIONS AND FUTURE PLANS

The following achievements can be stated that the proposed approach can be our future work:

- Smart software for a variety of functionality possible, so most of the qualitative characteristics of the proposed method are based on, fit the desired functional approach to identify baseline architecture to calculate.
- This method can be used with the base level of functionality and quality attributes of the matrix to calculate unknowns. For this purpose, the function f (G_{xi}) and matrix to calculate the value obtained using numerical methods based on the architecture and the functionality offered by the minimum values have been calculated.
- Discover and eliminate the weaknesses of the product architecture continuously in the XP process easily done.

7. REFERENCE

[1] Rumpe, B. and Schroeder, A., "Quantitative Survey on Extreme Programming Projects", Proceedings of International Conference on Extreme Programming and Flexible Processes in Software Engineering, pp. 95-100, 2002.

[2] Beck, K., Extreme Programming Explained: Embrace Change, 1st ed., AddisonWesley Professional, 2000.

[3] Beck, K. and Andres, C.,"Extreme Programming Explained: Embrace Change", 2nd ed., Addison-Wesley Professional, 2004.3.

[4] Nord R., Tomayko, J. and Wojcik, R., "Integrating Software-Architecture-CentricMethods into Extreme Programming (XP) ", Technical Report, Software.2002

[5] Arisholm, E., Gallis, H., Dybå, T. and Sjøberg, D. I. K., "Evaluating Pair Programming with Respect to System Complexity and Programmer Expertise",IEEE Transaction on Software Engineering, vol. 33, no. 2, pp. 65-86, 2007.

[6] Vanhanen, J. and Abrahamsson, P.,"Perceived Effects of Pair Programming in anIndustrial Context", Proceedings of 33rd EUROMICRO Conference on SoftwareEngineering and Advanced Applications, pp. 211-218, 2007.

[7] Cohn, M., "User Stories Applied for Agile Software Development", Addison Wesley,2004.

[8] Beck, K., Boehm, B., "Agility through Discipline: A Debate, Computer", vol. 36, no.6, 2003.

[9] Turk, D., France, R., Rumpe, B., "Limitations of Agile Software Processes",Proceedings of 3rd International Conference on Extreme Programming and Flexible Processes in Software Engineering.

[10] Nord, R. and Tomayko, J., 'Software Architecture-Centric Methods and AgileDevelopment', IEEE Software, vol. 23, no. 2, pp. 47- 53, 2006

[11] Cockburn, A., Agile Software Development,Addison-Wesley Professional, 2002

[12] Jensen, R. N., Møller, T., S¨onder, P. and Tjørnehøj, G., 'Programming; Introducing"Developer Stories"', Proceedings of 7th International Conference on Agile Processes and Extreme Programming in Software Engineering, pp. 164 – 168,2006.

[13] Levin, L., 'Reflections on Software Agility and Agile Methods: Challenges, Dilemmas, and the Way Ahead', Technical Report, Software Engineering Institute, Carnegie Mellon University, 2005.

[14] Leffingwell, D., Scaling Software Agility: Best Practices for Large Enterprises,Addison-Wesley Professional, 2007.

[15] Beck, K., Boehm, B., 'Agility through Discipline: A Debate', Computer, vol. 36, no.6, 2003.

[16] West, D., Metaphor, Architecture, and XP, Agile Alliance, 2002.

[17] Bass, L., Klein, M. and Moreno, G., "Applicability of General Scenarios to the Architecture Tradeoff Analysis Method", Technical Report, Software EngineeringInstitute, Carnegie Mellon University, 2001.pp. 43-46, 2002.

[18] Len Bass,Paul Clements,Rick Kazman, Software Architecture in Practice , Second Edition, Addison Wesley.2003.

[19] L. Hohmann, Beyond Software Architecture:Creating and Sustaining Winning Solutions,Addison Wesley,2003.

[20] Bosch,Design & Use of Software Architectures: Adopting and evolving a product-line approach, Addison-Wealey,2000.

[21] L.Lundberg and et al,Quality Attributes inSoftware Architecture Design , Proceedings of the LASTED 3td International Conference on Software Engineering and Applications,October,1999.

[22] L.Hohmann, "Beyond Software Architecture:Creating and Sustaining Winning Solutions",Addison Wesley ,2003.

[23] Bredemeyer, D., 2010. The Role of the Architect in Software and Systems Development, Last visited 15April, 2010 http://www.bredemeyer.com/Architect/RoleOfTheArchitect.htm

[24] L.Zhu, A. Aurum.I. Gorton, R.Jeffry, "Tradeoff and sensitivity analysis in software architecture evaluation using analytic hierarchy process", Software Quality Journal, 13 ,PP. 357_375,2005.

[25] J.A.McCall Quality factors, in Encyclopedia if Software Enginering, J.L. Marciniak (ed), John Wiley & Sons New York,pp.958969,1994.

[26] K.LEE, A top-down approach to Quality driven architecture engineering of software systems IEICE TRANS.INF.& SYST, Vol.88 D,NO.12,December 2005.

[27] M.Poyhonen,R.P . Hamalainen, "On the convergence of multiattribute weighting methods",European Journal of Operational Research 129,PP. 569_585,2001.

[28] A.Papoulis, S . Pillai, "probability Random Variables and Stochastic Processes",4th Edition, McGrawHill,2002.

[29] Zafar Karimi,Hasan Rashidi" Encoding Text Messages through Turbulence Functions " Conference on Computers, Makoo,2011.

[30] Shaw, M. and Clements, P., 'The Golden Age of Software Architecture', IEEE Software, vol. 23, no. 2, pp. 31-39, 2006.

[31] Wojcik, R., Bachmann, F., Bass, L., Clements P., Merson, P., Nord, R. and Wood,B., 'Attribute-Driven Design (ADD), Version 2.0', Technical Report, Software Engineering Institute, Carnegie Mellon University, 2006.

[32] Barbacci, M. R., Ellison, R., Lattanze, A. J., Stafford, J. A., Weinstock, C.B. and Wood, W.G., 'Quality Attribute Workshops (QAWs), Third Edition', Technical Report, Software Engineering Institute, Carnegie Mellon University, 2003.

[33] Nord, R., Barbacci, M., Clements, P., Kazman, R., O'Brien, L. and Tomayko, J.,'Integrating the Architecture Tradeoff Analysis Method (ATAM) with the Cost Benefit Analysis Method (CBAM)', Technical Report, Software Engineering Institute, Carnegie Mellon University, 2003.

Under Page

Extreme Programming[1]

Quality Attribute Workshop[2]

Attribute Driven Design[3]

Analytic Hierarchy Process[4]