

An Efficient Approach for Agile Web Based Project Estimation: AgileMOW

Ratnesh Litoriya¹, Abhay Kothari²

¹Department of Computer Engineering, JIT University, Rajasthan, India; ²Department of CSE, SIMS, Indore, India.
Email: ratneshlitoriya@yahoo.com

Received April 3rd, 2013; revised May 3rd, 2013; accepted May 11th, 2013

Copyright © 2013 Ratnesh Litoriya, Abhay Kothari. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

ABSTRACT

There exist numerous sophisticated models and methods for estimating the size, cost and schedule of software projects. However, the ability to accurately predict the software cost of web based software for agile software projects is still dubious. The most appropriate web based software cost estimation in an agile software development environment is a big problem due to varying levels of customer requirement and different individual personnel capabilities. Also, the need to develop a separate model to estimate web based project is rising. In this research paper, we present AgileMOW, an approach to model a software cost estimation process which will evaluate effort and cost of software development for web based projects developed using Agile methodology with a number of constraints imposed by stakeholders and environmental characteristics, thereby satisfying multitudinous criteria by making use of COCOMO II model, a famous traditional algorithmic technique. Most importantly, the paper identifies the difference between conventional and web projects. The proposed estimation model enhances the level of visibility in the planning stages.

Keywords: Agile Software; COCOMO II; Estimation; Manifestoes; Web Projects

1. Introduction

Software projects generally follow the traditional development life cycle models [1]. Unfortunately these models do not result in successful product in terms of technology used and satisfaction of revised requirements at the time of product release. Software development effort estimation deals with the prediction of the probable amount of time and cost required to complete the specific development task. Predicting the estimates obtained at the early stages of development life cycle is inaccurate because of long duration between the signing of the project and its delivery, also not much details of the system is available at that time. The way we develop software is changing. Software is developed from requirements through Agile Web development; professionals join together for building blocks and reusable components using rapid application development process and continuous prototyping [2]. Things take place so quickly that it is tough to get a handle on their status and whether they are making suitable progress [3]. Web based projects are also hard to estimate, especially in agile environment with limited resources, software developers require to better predict the

time and effort is essential to pull off such projects successfully. The result motivates industries and developers to adapt more iterative and incremental agile models. Software estimation is mandatory for software developers and their companies because it can provide cost control delivery accuracy and many other benefits. There are three elements under software Web cost estimation, *i.e.* software metric, web cost estimation model and software cost estimation tool [4]. Presently many quantitative models of software cost estimation have been developed. Most of the software estimation models available are based on some forms of regression technique, these models have a mathematical foundation and are constructed by collecting data on completed software project and developing regression equation relating them.

In this paper we introduce AgileMOW, an adaptation of a software estimation model to a particular context of agile methodology of software development. In the Second section, we briefly discuss about the software cost estimation models. Section 3 presents agile based web development projects and its characteristics, Section 4 highlights about challenges in web cost estimation, Sec-

tions 5 and 6 present the Agile software development and their manifestoes along with problem statement. The proposed model for agile web based software effort estimation is given and described in the following section.

2. Software Cost Estimation Models

Software industries and developers always interest to know the time estimation of software projects at the time of inception of software development. Cost models were only based on a single parameter such as program size. These models were not accurate and estimation done by comparing similar projects that have already been developed. Now a day in the light of software crisis all over the world, software estimation is a big challenge due to it allows for financial and strategic planning. Software cost estimation techniques can broadly be classified under algorithmic and non algorithmic models [5] as shown in **Figure 1**.

Algorithmic models are based on the statistical analysis of historical data such as past projects. Non algorithmic techniques are based on new approach like expert judgment, price to win and machine learning.

2.1. Algorithmic Models

Few very popular algorithmic models includes CO-COMO II by Boehm’s, function point by Albert’s and SLIM by Putnam. All these models require inputs accurate estimate of specific attribute such as line of code (LOC), number of user screen, interface complexity etc. which are difficult to predict during the initial stage of software development. **Figure 2** illustrates some very popular algorithmic models. Calculating these models is hard due to inherent complex relationships between the related attributes. Despite attributes and relationships used to predict software development estimates could change over for different software development environment. The limitation of algorithmic models led to the introduction of non algorithmic technique.

2.2. Non Algorithmic Models

Non algorithmic models for software estimation came into the existence in early 90’s. Researchers of their field found some new approaches supported by soft computing;

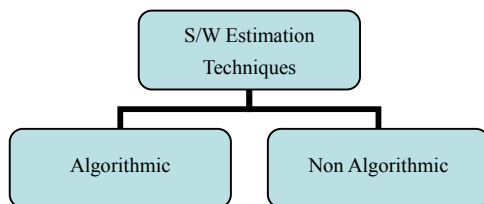


Figure 1. Software estimation classification.

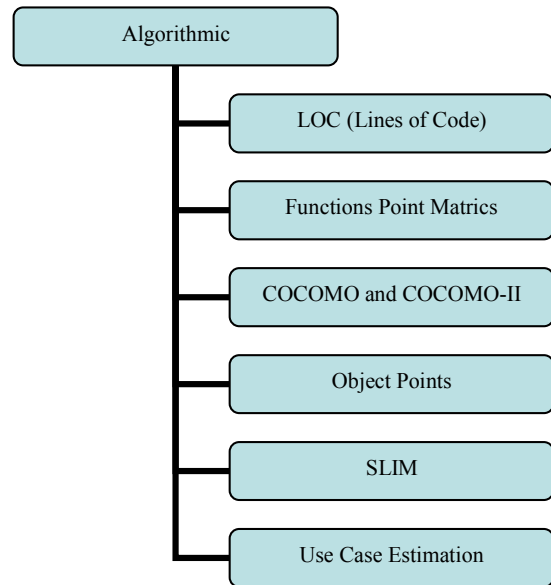


Figure 2. Algorithmic models.

they are artificial neural network, fuzzy logic and genetic algorithms.

Figure 3 summarize the non algorithmic techniques. Expert Judgment is a non algorithmic technique is carried out based on experience of project manager or a team of expert. Experience proves that model-based estimates do not perform considerably better than estimates exclusively based on expert judgment [6]. In Thumb Rule decision is taken based on personnel interest, it has certain drawbacks. In Delphi technique, no direct interaction is there among the experts, Coordinator look after the whole process. Wide Band Delphi technique introduced by Rand Corporation is a one to one interaction technique is carried out after mutual agreement among experts. Buy vs. Make decision is based on reusable Component based software development. In Parkinson’s Law the project cost is calculated on the basis of resources available in an organization. In pricing to win estimation technique the project cost is determined from the customer’s budget, however the approach is a business like but when detailed information is lacking it may be the appropriate strategy. PSP Probe Method introduced by Watt uses similar project and product work experiences when estimating future efforts. TSP Planning (Team software process) introduces team dynamics for planning, role definition and development phases over PSP trained and operating developers.

3. Agile Based Web Development Projects and Characteristic

During the past few years, estimators tried to estimate the cost and schedule for a Web based project. Apparently, developers use different high level technologies such as

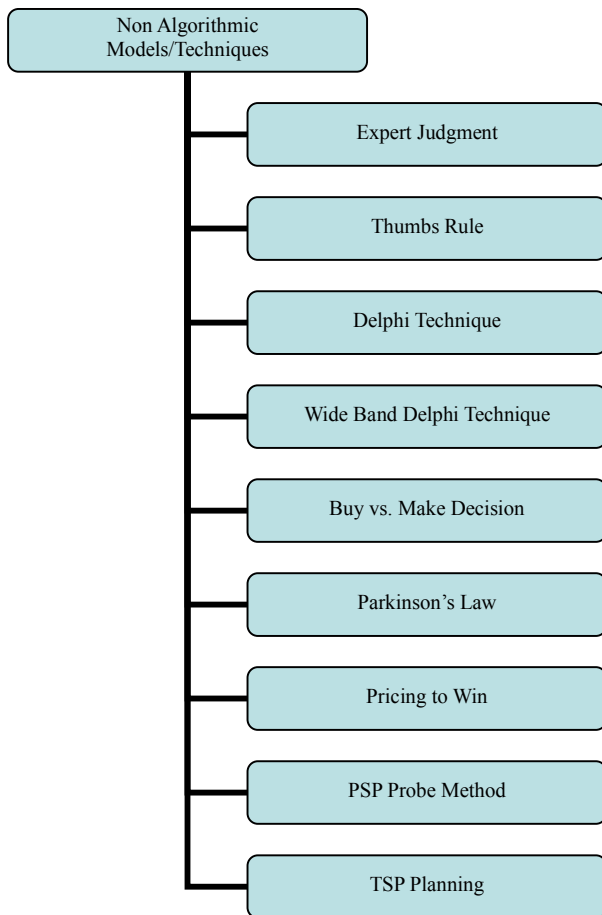


Figure 3. Non algorithmic mod.

hypertext markup language, Java applets and script, and many more to generate software for the Web in the flash of an eye. But these special category projects beat estimation processes, defy models, and makes size metrics obsolete. With the advent of Agile process model of software project development generating numerous Web objects and making them functional in just a few months, this accelerated pace raises pressing questions.

Table 1 emphasizes the move to Agile software development paradigm with component-based software development, systematic reuse, and visual technologies [7]. It discovers the move to quick-paced developments: Getting their software to market first is the top priority for firms doing business on the Web. The way of software development is shifting. Rather than develop software from requirements through the waterfall, Agile based Web development provides working software over comprehensive documentation professionals stick together building blocks and reusable components using iterative and incremental development, rapid application development methods and continuous prototyping. Agile Estimation Using Functional Metrics [8] is also a way of predicting web cost. Agile Web developments are also

difficult to estimate particularly in firms with limited resources.

4. Challenges in Web Cost Estimation

Web projects are different and web development using agile methodologies are even different. To underscore the challenges involved in the area of Agile Web estimation, **Table 2** demonstrates the comparison of conventional estimation approach and agile web cost estimation challenges [7].

The key challenges are accurately estimating size and duration. New modified size metrics are desirable to handle with Web objects like shopping carts, Java scripts, and building blocks like Cookies, ActiveX controls, and Component Object Model components. New modified duration-estimating equations are required to address the fact that the cube root laws utilized by most estimating models just do not seem to work for the Web.

5. Agile Software Development

Agile software development is a collection of software development methods based on iterative and incremental development, where requirements and solutions evolve through collaboration between self-organizing, cross-functional teams [9]. It promotes adaptive preparation, evolutionary development and delivery, a time-boxed iterative approach, and encourages rapid and flexible response to change. **Figure 4** shows the functioning of Agile development. It is a conceptual framework that promotes anticipated interactions throughout the development cycle.

5.1. Manifestos for Agile Software Development

In 2001 Manifesto for Agile Software Development to define the approach published and came into the existence now known as agile software development. The four manifestoes are as follows:

- 1) Individuals and interactions over processes and tools;
- 2) Working software over comprehensive documentation;
- 3) Customer collaboration over contract negotiation;
- 4) Responding to change over following a plan.

5.2. Agile Methods and Characteristics

There exist many specific agile development methods. Majority promote development, collaboration, teamwork, and process adaptability throughout the life-cycle of the project. Agile methods break tasks into small increments with least planning and do not directly engross long-term planning [11]. Iterations are short time frames that typically last from one to four weeks. Each iteration involves

Table 1. Characteristics of conventional versus agile web development projects.

Characteristics	Conventional Development	Agile Web Development
Key objective	Build quality software products at minimum cost	Bring worth products to market as rapidly as possible under varying requirements of clients
Project size	Medium to large (hundreds of team members)	Small (5 - 7 team members)
Costing	In millions	In thousands
Development approach employed	Classical, requirements-based, water fall or incremental delivery, uses cases, documentation-driven	Rapid application development, gluing building blocks less paper work, XP
Major engineering technologies used	Object oriented methods, generators, modern programming languages (C++), CASE tools, and so forth	Web object based methods, fourth- and fifth-generation technologies like JAVA. Net framework PHP etc.
Processes employed	Capability maturity model-based	<i>Ad hoc</i>
Products developed	Code-based systems, mostly new, some reuse, many external interfaces, often complex applications	Object-based systems, many reusable components (on line banking, web kiosk etc.), few external interfaces, moderately simple
People involved	Professional software engineers typically with 5+ years of experience in at least two application domains	Graphic designers, less experienced software engineers (2+ years), new appoints right out of school
Estimating technologies used	Algorithmic and non algorithmic techniques Analogy using historical data as its basis, SLOC or function point-based models, Work Breakdown Structure (WBS) approach for small projects	Analogy based upon current experience, "design-to-fit" based on available resources, WBS approach for small projects

Table 2. Challenges of agile web development.

	Conventional Approach	Agile Web-based challenges
Estimating process	Use analogy supplemented by lessons gathered from past experience	Job costing done <i>ad hoc</i> based on inputs from the developers.
Size estimation	SLOC or function points are used. Separate models are used for COTS and reused software.	Applications are built using templates and a variety of web-based objects (html, applets, components, building blocks).
Effort estimation	Effort is estimated via regression formulas customized by cost drivers.	Effort is estimated by breaking the job down into tasks and identifying what is needed to do the work.
Schedule estimation	Schedule is estimated using a cube root relationship with effort.	Schedule is estimated based upon analogy. Models typically estimate schedules high because cube root relationship doesn't hold.
Model calibration	Measurements from past projects are used to calibrate models.	Measurements from past projects are used to identify myths.

a cross functional team working in all functions: planning, requirements analysis, design, coding, unit testing, and acceptance testing. At the end of the iteration a working product is demonstrated to stakeholders. This minimizes overall risk and allows the project to adapt to changes quickly. Agile Methods are in Wide-Spread Use, few popular methods involves:

- 1) Extreme Programming (Kent Beck);
- 2) Scrum (Ken Schwaber, Jeff Sutherland, Mike Beedle);
- 3) Crystal (Alistair Cockburn);
- 4) DSDM (Arie van Bennekum);
- 5) Feature-Driven Development (Jeff De Luca);
- 6) Lean Development (Bob Charette);
- 7) Adaptive Software Development (Jim Highsmith).

6. Problem with Existing Models and Techniques

The estimators presently have not agreed on how to develop estimates for Web-based projects developed by agile methodology. The difficulty is that the characteristics of the Web-based projects that are listed in **Table 1** along with manifestoes of agile paradigm make it complicated for estimators to acclimatize and put already existing processes, metrics, and models to work functionally.

7. Proposed Model for Agile Based Web Development

Predictive methods focus on configuration of a programmed

plan to carry out the project. In contrast, adaptive methods focus on examining the current situation and finding the best solution in every step of performing the project [12]. The seed of web estimation for agile software projects (AgileMOW) is based on a multiple web object and multiple attribute approach. The architectural model illustrated in **Figure 5** describes the idea of arriving at the influencing factors required to imitate the agile environment. The cost drivers and scale factors described in COCOMO II is an appropriate combination set of agile environmental and people attributes. The agile manifestoes impose certain constraints on the estimates. In order to identify the agile attributes, certain inbuilt issues in this environment are to be considered. Agile methods consider the human/people factor to be more significant than the process factor [13]. We propose the new model AgileMOW, the Web Model for agile software methods, because it is an adaptation and extension of the webMO [7] and Cocomo II early design model.

The formulation of any predictable processes essentially requires the various components to be predictable. Though people are not predictable components, therefore there exists a difficulty in forecasting and quantifying web based software developed using agile for cost estimation. In addition, uncertainty, risk factors, emerging requirements and complicated issues are presented in agile as in any other conventional software development process. The estimation of cost of web projects in an agile environment needs a mapping from the qualitative province to a set of quantitative values. To address this mapping, the key quality-attributes affecting the agile web based software development are tabulated in **Table 3** along with their rating levels. Rating levels refer to a statistical measure that comparatively signifies the effect of the attribute on the product quality and completion

time respectively.

7.1. Size Estimation

Web-based projects are unsurprisingly small and intensive, so unavailability of an appropriate effort estimation model move forwards developers to make highly risky estimations. Moreover, the rapid evolution and growth of Web related technology, tools and methodologies makes historical information quickly old-fashioned. Size esti-

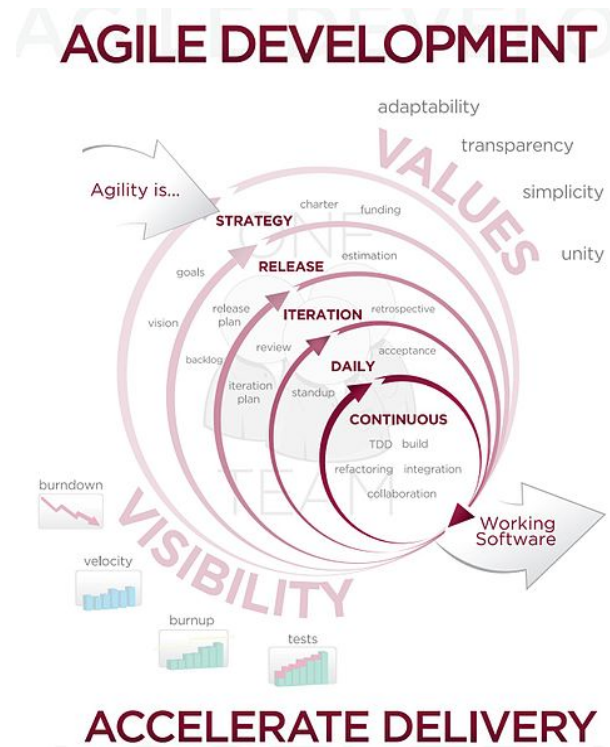


Figure 4. AGILE development poster [10].

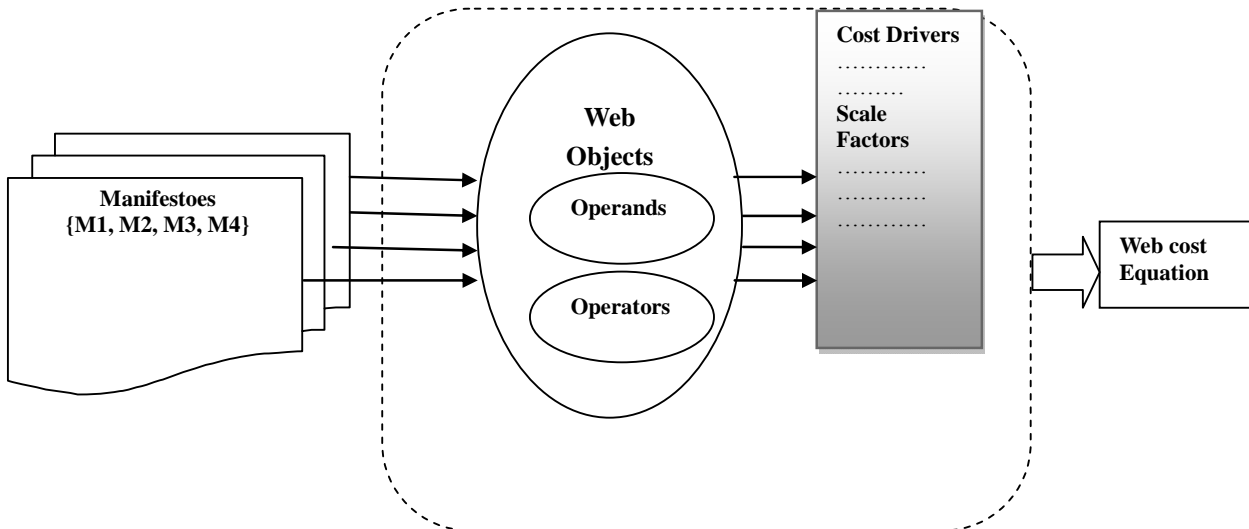


Figure 5. Proposed architecture of AgileMOW.

Table 3. Attributes and ratings relevant to agile manifestoes.

High priority attributes	Rating level	Low priority attributes	Rating level
Communication skills		Process maturity	
Proximity of team		Tool availability	
Feedback		Tool familiarity	
Courage		Conservativeness	
Managerial skills		Following process mod	
Consistent working		Training	
Technical ability		Planning	
Debugging capability		Pages of documents	
Reliability		Project complexity	
Function points		Other expenditure	
Ease of use		Documentation resources	
Early delivery 5		Documentation period	

mation of web development for agile methodology can be expressed in terms of, web objects [7]; size metrics computes size by considering every one of the various elements that make up the web application into account. We have used Halstead equation for Volume (*i.e.*, a proposed measure of size that is language independent and is related to calculate the length of the program in terms of operands and operators) as follows:

$$S = (W_1 + W_2) \cdot \log_2 (w_1 + w_2)$$

where W_1 = total occurrence of Web Operand estimator;

W_2 = total occurrences of Web Operator estimator;

w_1 = number of unique Web Operands estimator;

w_2 = number of unique Web Operators estimator.

7.2. Effort Estimation

Size estimation is just the primary step in developing a model that precisely estimates Web development costs in agile paradigm and schedule. The key issues revolve around the formation of the mathematical equations for effort. Though the traditional cube-root relationship between effort and duration in most estimation models does not seem to accurately predict Web development schedules for agile environment due to people centric approach. The proposed effort model AgileMOW is a mix of expert judgment and data from different academic projects using regression analysis. Its mathematical formulation rests upon parameters from both the Cocomo II and Soft Cost-OO software cost-estimating models along with manifesto attributes of agile methodologies. We have taken the value range of exponents from webMO model proposed by Donald J. Reifer. Equation 2 shows the AgileMOW model for estimating equations for effort (in per-

son-months) as noted the resulting effort estimation model has 24 manifesto attributes and fixed power laws.

$$\text{Web Effort } [E] = A \prod \text{cfwa}_i (\text{size})^{P_1}$$

where A is constant.

P_1 is power factor; cfwa is agile manifesto attributes for agile web development; size is the number of web objects.

Cost factor of this formula will be decided based on the effect of manifesto attributes on cfwa and which have been named as cfwa in this formula.

8. Conclusions

The market fruition compels clients to adopt agile development paradigm for very short-term Web projects. Very often the results are deficient, unreliable and hard to maintain applications that fail to meet the software industry needs, generally, the agile web effort estimation is not able to predict and help to avoid these problems.

In this paper, we aim to address the numerous open issues in web development particularly in context to agile software development by analyzing data from completed agile Web development projects. Though agile is relatively a new and less mature scenario for developers, they spend time on trying to estimate the web development effort pragmatically and reliably, they usually have very little or almost none historical information. Agile characteristics tend to make web estimations less reliable regarding both time and cost. In order to get speedy and reliable effort estimations of Agile based Web development projects, this paper presented the AgileMOW model. It does not substitute the expert estimator, but it offers a tool for achieving a more accurate estimation,

based on agile manifestoes and web characteristics.

REFERENCES

- [1] L. Lazić, *et al.*, “Two Novel Effort Estimation Models Based on Quality Metrics in Web Projects,” *WSEAS Transactions on Information Science and Applications*, Vol. 7, No. 7, 2010, pp. 923-934.
- [2] Ziauddin, S. K. Tipu and S. Zia, “An Effort Estimation Model for Agile Software Development,” *Advances in Computer Science and Its Applications (ACSA)*, Vol. 2, No. 1, 2012, pp. 314-324.
- [3] S. F. Ochoa, *et al.*, “Estimating the Development Effort of Web Projects in Chile,” *Proceedings of the First Latin American Web Congress (LA-WEB 2003)*, 10-12 November 2003, pp. 114-122.
- [4] T. C. Hooi, Y. Yusoff and Z. Hassan, “Comparative Study on Applicability of WEBMO in Web Application Cost Estimation within Klang Valley in Malaysia,” *IEEE 8th International Conference on Computer and Information Technology Workshops*, Sydney, 8-11 July 2008, pp. 116-121.
- [5] S. Kumari and S. Pushkar, “Comparison and Analysis of Different Software Cost Estimation Methods,” *International Journal of Advanced Computer Science and Applications*, Vol. 4, No. 1, 2013, pp. 153-157.
- [6] M. Ruhe, *et al.*, “Cost Estimation for Web Applications,” *Proceedings of the 25th International Conference on Software Engineering*, Portland, 3-10 May 2003, pp. 285-294.
- [7] D. J. Reifer “Web Development: Estimating Quick-to-Market Software,” *IEEE Software*, Vol. 17, No. 6, 2000, pp. 57-64.
- [8] Vice President David Consulting Group, “Agile Estimation Using Functional Metrics,” Thomas Cagley.
- [9] Mendes and Mosley, “Web Cost Estimation: An Introduction,” pp. 182-190.
- [10] http://commons.wikimedia.org/wiki/File:Agile_Software_Development_methodology.jpg
- [11] “Agile Estimation: Characteristics, Techniques and Complexities,” *Tools Journal*, pp. 1-4.
- [12] B. Nasr-Azadani and R. M. Doost, “Estimation of Agile Functionality in Software Development,” *Proceedings of the International Multi-Conference of Engineers and Computer Scientists*, Hong Kong, 19-21 March 2008, pp. 1338-1341.
- [13] S. Chandrasekaran, R. Lavanya and V. Kanchana, “Multi-Criteria Approach for Agile Software Cost Estimation Model”.