



Generalized Agile Estimation Method

Shilpa Bahlerao[#], Maya Ingle^{*}

[#] Faculty of Computer Application

Acropolis Institute of Technology and management, Indore, 452016, India
Tel.:+917314730057, E-mail:Bhalerao.shilpa@gmail.com

^{*}Indore Institute of Computer Application

Rau, Indore,452001, India
Tel.:+917314010580, E-mail: maya_ingle@rediffmail.com

Abstract— Agile cost estimation process always possesses research prospects due to lack of algorithmic approaches for estimating cost, size and duration. Existing algorithmic approach i.e. Constructive Agile Estimation Algorithm (CAEA) is an iterative estimation method that incorporates various vital factors affecting the estimates of the project. This method has lots of advantages but at the same time has some limitations also. These limitations may due to some factors such as number of vital factors and uncertainty involved in agile projects etc. However, a generalized agile estimation may generate realistic estimates and eliminates the need of experts. In this paper, we have proposed iterative Generalized Estimation Method (*GEM*) and presented algorithm based on it for agile with case studies. *GEM* based algorithm various project domain classes and vital factors with prioritization level. Further, it incorporates uncertainty factor to quantify the risk of project for estimating cost, size and duration. It also provides flexibility to project managers for deciding on number of vital factors, uncertainty level and project domains thereby maintaining the agility.

Keywords— Agile software development, estimation, CAEA, *GEM*, vital factors, ergonomic, technological, story point.

I. INTRODUCTION

Software cost estimation (also known as software estimation) has been an important and difficult task since the evolution of the software. Many formal and informal methods have been proposed for software estimation. It is important for estimation methods to generate realistic software estimates to build the trust of customers as well as team members. Unrealistic estimates are major factors for either software project failure or decreasing the quality of the software [1]. The software estimation process becomes more cumbersome in case of Agile Software Development Process (ASDP) [2]. ASDP is a light weight process that addresses the volatile requirements at any stage of software and follows iterative and incremental development [3][4]. Thus, software estimation in case of ASDP involves some challenges such as uncertainty in requirements and more dependency on oral communication etc. Many agile estimation techniques have developed and are classified as non-algorithmic and Algorithmic methods[8]. Non-algorithmic methods are frequently used by agile practitioners and include planning poker, disaggregation

etc.[9][10]. These methods are used to derive the estimates on the basis of expert opinion and historical data. However, these methods are not useful in case of unavailability of both aforesaid factors. Further, these methods may generate the different estimates for same project depending on the intuition of the estimators [8]. On the contrary, Constructive Agile Estimation Algorithm (*CAEA*) is an iterative estimation method that incorporates vital factors for estimation of Cost, Size and Duration (CSD) of an agile project [2]. These vital factors being project specific and hence cover the related factors mainly; performance, configuration, complex processing, data transfer, security, multiple sites and operational ease. Algorithmic approach generates realistic estimates [8]. At the same time, this method has certain limitation in terms of number of vital factors, their prioritization and uncertainty of project. Therefore, there is a strong need to develop the generalized estimation based algorithm for agile projects.

In this paper, we have introduced Generalized Estimation Method (*GEM*) and presented algorithm based on *GEM* for agile projects with some case studies. The terminology used in our proposed method is described in Section II first. In Section III, *GEM* along with an algorithm is discussed. We

cover some case studies in Section IV. Finally, we conclude in Section V.

II. TERMINOLOGY

In our proposed *GEM*, some terms are frequently used. These terms are discussed in elaborated manner as follows:

A. Project Domain (PD)

Project domain is defined as an environmental class of projects that has specific attributes and may be developed for specific community/ environment. For example, military project requires higher reliability, security and is developed for defense applications. The classification of *PD* for CSD estimation of projects is shown in Table I.

B. Vital Factors (*v*)

Vital factors refer to the factors that affect the CSD of project in agile estimation and classified vital factors into main sub classes as project, sociological, technological and ergonomic classes as shown in Table II.

C. Weights (*w*)

It is defined as the value assigned to each vital factor depending upon its priority in a project from any project domain. As per the importance of the vital factors, various levels may be formed to accommodate the vital factors having same priority with appropriate weights.

D. Intensity levels (*I*)

Intensity level refers to influence/ impact of a particular vital factor in a project and assumes either low (L), medium (M) and high (H) intensity levels of a vital factor.

E. Uncertainty Factor (UF)

It deals with the level of uncertainty in requirements, technology and resource aspects related to project and its value lies between 0 and 1.

F. Story points (SP)

It defined as approximate lines of code in a story and also referred as size of the story. A story in agile project refers to a small piece of requirement.

G. New Story Points (NSP)

It is a measure of efforts required to develop a story after the inclusion of vital factors.

H. Size of Project (SOP)

SOP is term used for denoting the size of project and is measured in terms of *NSPs*.

I. Duration of Project (DOP)

Duration of project is denoted as *DOP* and is measured in time as either in weeks, months or years.

J. Velocity

It is number of story point developed by team in a specified time.

III. GENERALIZED ESTIMATION METHOD (GEM)

Generalized Estimation Method is an iterative algorithmic approach that follows an agile estimation process. Agile estimation process starts with identification of project domain class and vital factors on the basis of project behavior and initial information of user class. *GEM* is divided in two phases: Early Estimation (EE) and Iterative Estimation (IE). EE takes input of project domain, identified vital factors and risk involved in project and is

Table I: Project Domain Classification

Sr. no.	Domain	Sub-domain	Description
1	Critical software	System Software	Software that are used to control the devices such as telephone system, computer fall under this category. Operating System, compilers, tools are considered in this category.
		Military Software	Software developed for military organization
		Life saving software	Medical or scientific software that require higher accuracy such as space shuttle software, medical software
2	Application Software	Commercial Software	Software that are leased or marketed to client such as spread sheets, word processor etc.
		Information System	MIS systems that are produced in-house support such as payroll, attendance
		Out Sourced Software	A software that built under the contract to client organization
		Web application	Software that use client server technology
		End user application	Software for personal use

non-chargeable activity to develop faith and trust of customer. IE is chargeable activity and performed during iteration planning. It re-estimates the CSD of working software after identification of actual velocity, influence of vital factors in project and risk involved in requirements and resources. It is way of autocorrecting the estimates after updating requirements and feedback received from stakeholders. In following subsection, we will discuss algorithm based on *GEM* and its formal description.

A. GEM Based Algorithm and its Description

In *GEM* based algorithm, we assume that class of project domain (*PD*) and associated n vital factors v_1, v_2, \dots, v_n are properly identified. Further, intensity level I_i of i th vital factor is assigned either Low (L) or Medium (M) or High (H) value and the quantified values accordingly. Weights w_1, w_2, \dots, w_n associated with vital factors v_1, v_2, \dots, v_n are allocated in such a manner that sum of weights (*SUM*) of all vital factors for a project is 1. Using these weights and intensity levels, the priority factor $PF(v_i)$ for each vital factor and Unadjusted value *UV* are computed using equation (1) and equation (2) respectively. We have quantified the risk associated with project in the form of uncertainty factor (*UF*). The value of *UF* has been quantified as per the risk associated with project and may vary from 0 to 1. Every class of project consists of many functionalities corresponding to requirements. These may further be decomposed into m small independent pieces i.e. stories. Thus, it becomes possible for us to compute SP_j for each story j . Using equation (3), New Story Point (NSP_j) is computed for each m number of stories. With the help of

these NSPs and velocity of project development team, finally we compute SOP and DOP using equation (4) and (5).

$$PF(v_i) = w_i * I_i \text{-----(1)}$$

$$UV = \sum_{i=1}^n PF(v_i) \text{-----(2)}$$

$$NSP_j = SP_j + PD + UF * UV \text{----(3)}$$

$$SOP = \sum_{j=1}^m NSP_j \text{-----(4)}$$

$$DOP = SOP / velocity \text{-----(5)}$$

It is important to note that the above process is executed for estimating CSD of a particular case of a specified project domain application. We describe formally an algorithm based on GEM in this section as follows:

Algorithm:

// We assume that PD and vital factors are identified in a given project to be estimated. This algorithm inputs quantified intensity levels of identified vital factors, weights of vital factors, story points and uncertainty factor. Algorithm is executed till all requirements are covered //

Main()

{
do

{
Input $n, m, velocity$;

Step 1: // Assign quantified intensity levels I_1, I_2, \dots, I_n of vital factors v_1, v_2, \dots, v_n respectively. //

For ($i=0; i \leq n; i++$)

Input $I(i)$;

Step 2: //Assign weights w_1, w_2, \dots, w_n to vital factors v_1, v_2, \dots, v_n respectively such that sum of all weights should not exceed 1. //

$SUM = 0$;

For ($i=0; i \leq n; i++$)

{

Input $w(i)$;

$SUM = SUM + w(i)$;

If ($SUM > 1$) then

{

Message "invalid input";

$SUM = SUM - w(i)$;

$i--$;

}

}

Step 3: //Find priority factor $PF(v_i)$ for each vital factor.//

For ($i=0; i < n; i++$)

$PF(i) = I(i) * w(i)$;

Step 4: //Compute Unadjusted Value (UV) for a project.//

$UV = 0$;

For ($i=0; i < n; i++$)

$UV = UV + PF(i)$;

Step 5: //Calculate the value of NSP for each story using UF where ($0 < UF \leq 1$). //

For ($j=0; j < m; j++$)

$NSP(j) = SP(j) + PD + UF * UV$;

Step 6: //Compute Size Of Project SOP. //

$SOP = 0$;

For ($j=0; j < m; j++$)

$SOP = SOP + NSP(j)$;
Step 7: // Calculate Duration of Project (DOP). //
 $DOP = SOP / velocity$;
}
while ($no_of_requirements > 0$)
}

Table II: Vital Factors Affecting Project Estimation

Sr. no	Classification	Vital Factors	Description
1	Project Specific	Configuration	Requirement of special hardware or software
		Complex processing	Special Scientific calculations
		Performance	characterized as execution time, designing and coding standards, accuracy in outcome etc. as per customer requirements
		Data Transfer	volume and frequency of data transfer from peripherals/ machine to remote machine/ peripheral.
		Security	Data security, authentications
		Operational ease	Facilities provided to user more than must have feature
		Multiple Sites	Software run on multiple sites
2	Sociological	Constraints	Special constraint required by user such as higher reliability, time constraint etc.
		Team Size	Number of professional working on team (full time)
		Team Experience	Experience person ratio in team
		Communication	Mode of communication in team
		Domain expertise	Experience in project domain
3	Technological	Language expertise	Team command on language
		Language	Procedural or object oriented
		Methodology	Particular agile method or hybrid practices
		Project management Techniques	Project tracking and monitoring
		Reusable material	Reusability components
4	Ergonomic	Automated tools	Use of automated tools
		Physical layouts	Open space or cubical
		Customer involvement	Full time customer, proxy customer
		Level of distraction in office	Frequency of distraction during the work

IV. CASE STUDIES

In this section, we have considered the small applications to analyze results obtained from *GEM*. In our study, we have concentrated on two aspects of agile estimation; firstly impact of assigning weights to aforesaid vital factors on CSD of agile project and secondly, impact of uncertainty factor on CSD estimation. In this section, we first describe the research setup for our case studies following the discussion of project cases with results of this study.

Our study included small projects of three *PD* classes namely; web application, MIS project and military projects. We have used square series for quantifying the efforts required for particular *PD* class such as web application possesses *PD* value as 1, MIS project as 4 and military project possesses *PD* value as 9 in our study. Our study concentrated on only project specific vital factors assuming that other vital factors are favourable for agile software development and do not have any effect on the CSD of projects. Table III depicts the projects categories with other parameters that are considered for our study. The vital factors with varying intensity levels are considered with their corresponding weights are shown in Table IV. We have attempted to identify the impact of uncertainty on CSD estimation by considering the various levels of uncertainty for the projects with same intensity level of vital factors. The uncertainty levels in terms of uncertainty factor are ranging from 0.2, 0.4, 0.6 and 0.8. Also, we assumed the team velocity as 8, SP value of a story as 7 and each project consists of 25 stories of same size. In study, we have vital factors at all three intensities levels are considered for assessing generalized results obtained from use of *GEM* based algorithm on various class of project domains. Thus, includes around more than 1500 projects of all three domains with varying intensity levels.

All projects considered for study are categorized in two cases. Case I deals with projects having same intensity level of all vital factors. Algorithm computed *NSP* values, *SOP* and *DOP* for project of three aforesaid domains projects of three domains for case I as shown in Table V. Thus, case I included three special cases namely; all vital factors at low intensity levels (i.e. LLLLLL), all vital factors at medium intensity level (i.e. MMMMM) and high intensity level (HHHHHH). *NSP*s values for low uncertainty projects range from 8.2 to 9.8 for web application whereas *NSP* values for higher uncertainty (i.e. 0.8) range from 8.8 to 15.2 for same *PD* class i.e. web application. It has been found that these *NSP* values are lower and upper limits of *NSP* values for the projects at same uncertainty level of particular *PD* class.

In Case II, *GEM* based algorithm uses input as various intensity level of various vital factors. Thus, HLMHLMH is a project case configured with high performance, low configuration, medium complexity, high data transfer, low security, medium multiple sites and high operational ease. The *NSP* computations of random combinations of intensity levels of projects have been shown in Table VI. It is evident from the Table VI that *NSP* values at various intensity values in the specified range of case I i.e. *NSP* values of case II are higher as compared to projects with all vital factors at low intensity and lower as compared to projects with all vital factors at high intensity. Also, it has been noticed that the

UF plays a role of catalyst in increasing *SOP* and *DOP* of the project. We found in our study that a project case with low uncertainty MMMHMM possesses higher *NSP* and *SOP* values due to maximum vital factors of higher prioritization level at higher intensity level.

Table III: Projects with their Parameters and Attributes

Sr. No	Parameters	Project Attribute Values
1	Project domain	Application Software :Web application, MIS project, Critical Software :military project
2	Vital factors	Performance (v_1), configuration (v_2), complexity (v_3), data transfer (v_4), security (v_5), multiple sites (v_6), operational ease (v_7)
3	Intensity level	Low (L) 1, Medium (M) 4, High (H) 9
4	Uncertainty	Depending on uncertainty level of projects

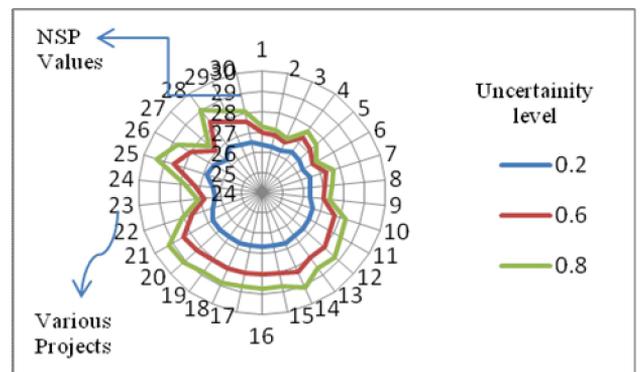


Fig. 1 *NSP* Computations for Web Based Project at various uncertainty levels

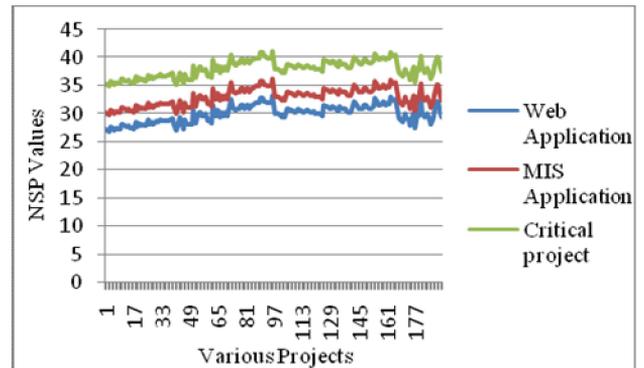


Fig. 2 *NSP* Computations for various Projects at same Uncertainty level 0.8

V. RESULTS AND CONCLUSION

CSD estimation of agile projects is a challenging task due to its principles and practices. Major challenge that lies in estimation is volatile requirements and ergonomic requirements of the ASDP. With the help of *GEM* based algorithm, early estimation process may be executed on the basis of vital factors with corresponding prioritized intensity levels, uncertainty levels and project domains by envisioning

Table IV: Vital Factors and Associated Weights

	Level I			Level II		Level III	
	Performance	Complexity	Configuration	Data	Security	Multiple	Operation
Weight	0.2	0.2	0.2	0.15	0.15	0.05	0.05

Table V: Statistics for Special Projects- Case I

Sr.no	Uncertainty	Vital Factors						NSP Computation			SOP (NSPs)			DOP(weeks)			
		v ₁	v ₂	v ₃	v ₄	v ₅	v ₆	v ₇	W	MIS	M	W	MIS	M	W	MIS	M
1	0.2	L	L	L	L	L	L	L	8.2	11.2	16.2	205	280	405	25.62	35	50.62
		M	M	M	M	M	M	M	8.8	11.8	16.8	220	295	420	27.5	36.25	52.5
		H	H	H	H	H	H	H	9.8	12.8	17.8	245	320	445	30.62	40	55.62
2	0.4	L	L	L	L	L	L	L	8.4	11.4	16.4	210	285	410	26.25	35.62	51.25
		M	M	M	M	M	M	M	9.6	12.6	17.6	240	315	440	30	39.75	55
		H	H	H	H	H	H	H	11.6	14.6	19.6	290	365	490	36.25	45.62	61.25
3	0.6	L	L	L	L	L	L	L	8.6	11.6	16.6	215	290	415	26.87	36.25	51.87
		M	M	M	M	M	M	M	10.4	13.4	18.4	260	335	460	32.5	41.87	57.5
		H	H	H	H	H	H	H	13.4	16.4	21.4	335	410	535	41.87	51.25	66.85
4	0.8	L	L	L	L	L	L	L	8.8	11.8	16.8	220	295	420	27.5	36.87	52.5
		M	M	M	M	M	M	M	11.2	14.2	19.2	280	355	480	35	44.37	60
		H	H	H	H	H	H	H	15.2	18.2	23.2	380	455	580	47.5	56.87	72.5

Table VI: Statistics computed for Case II projects

Sr. no	Uncertainty factor (UF)	Vital Factors						NSP Computation			SOP(NSPs)			DOP(weeks)			
		v ₁	v ₂	v ₃	v ₄	v ₅	v ₆	v ₇	W	MIS	M	W	MIS	M	W	MIS	M
1	0.2	L	L	L	H	H	L	L	8.68	11.68	16.68	217	292	417	27.12	36.5	52.12
		M	M	M	H	H	L	L	9.04	12.04	17.04	226	301	426	28.25	37.62	53.25
		H	L	M	L	H	M	L	8.91	11.91	16.91	222.75	297.75	422.75	27.84	37.21	52.84
		M	M	M	H	H	M	M	9.10	12.10	17.10	227.5	303.5	427.5	28.43	37.81	53.43
		H	M	L	H	L	M	H	8.99	11.99	16.99	224.75	299.75	424.75	28.09	37.48	53.09
2	0.4	L	L	L	H	H	L	L	9.36	12.36	17.36	234	309	434	29.25	38.62	54.25
		M	M	M	H	H	L	L	9.6	12.6	17.6	240	315	440	30	39.37	55
		H	L	M	L	H	M	L	9.82	12.82	17.82	245.5	320.5	445.5	30.68	40.06	55.58
		M	M	M	H	H	M	M	10.08	13.08	18.08	252	327	452	31.5	40.87	56.25
		H	M	L	H	L	M	H	9.98	12.98	17.98	249.5	324.5	449.5	31.87	40.56	56.18
3	0.6	L	L	L	H	H	L	L	10.04	13.04	18.04	251	326	451	31.37	40.75	56.37
		M	M	M	H	H	L	L	11.12	14.12	19.12	278	353	478	34.75	44.12	59.75
		H	L	M	L	H	M	L	10.73	13.73	18.73	268.25	343.25	468.25	33.5	42.90	58.53
		M	M	M	H	H	M	M	11.3	14.3	19.3	282.5	357.5	482.5	35.3	44.68	60.31
		H	M	L	H	L	M	H	10.97	13.97	18.97	274.25	349.25	474.25	34.28	43.65	59.28
4	0.8	L	L	L	H	H	L	L	10.72	13.72	18.72	268	343	468	33.5	42.87	58.5
		M	M	M	H	H	L	L	11.2	14.2	19.2	280	355	480	35	44.37	60
		H	L	M	L	H	M	L	11.64	14.64	19.64	291	366	491	36.37	45.75	61.37
		M	M	M	H	H	M	M	12.16	15.16	20.16	304	379	504	38	47.37	63
		H	M	L	H	L	M	H	11.96	14.96	19.96	299	374	499	37.37	46.75	63.37

upfront. These estimates are useful in identifying the early scope of project and establishing the trust of customer. On the other hand, an iterative estimation forces to improve the estimates after knowing actual team efficiency and clarity in requirements. We also presented two cases concerned with three project domains for study purpose. Further, we computed certain statistics such as size and duration of these project cases at various uncertainty levels. We have analyzed cost estimation of a particular project domain at various uncertainty level as well as cost estimation of various project domain at same uncertainty level as shown in Fig. 1 and Fig. 2. We observed following some interesting observation and facts as follows:

1) Iterative estimation provides the facility to improve the estimates after monitoring and tracking previous iteration thereby providing the scope of improvement in estimates and the actual facts to user and team.

2) GEM based algorithm eliminates the need of historical data and experts. Therefore, an average project manager can estimate more precisely.

3) It provides flexibility on number and type of vital factors, uncertainty level, project domains and quantification of aforesaid parameters depending on project behavior and team makeup.

4) Algorithm always generates different estimates for projects with different level of uncertainty. For example, CSD estimates for different uncertainty levels of web applications have been denoted in different colors in Fig. 2. Algorithm generates different statistics/ estimates for the projects with high intensity levels of different prioritization level.

5) Algorithm also resolves the limitations of CAEA by associating weights to each of the vital factors and at various uncertainty factors.

6) A dominating role of prioritization has been observed in case of low uncertainty on other hand impact of uncertainty is high in case of high uncertainty.

REFERENCES

- [1] T. Gruschke, Empirical Studies on Software Cost Estimation: Training of Effort Estimation Uncertainty Assessment Skills, Proc. 11th IEEE International Software metric Symposium.
- [2] S. Bhalerao, M. Ingle, "Agile Estimation using CAEA: A Comparative Study of Agile Projects", Proceeding of International Conference on Computer Engineering and Application (ICCEA09), Philippines pp.78-84.
- [3] K. Beck, *Extreme Programming Explained*, (Pearson Education Low price Edition Asia, 2006).
- [4] A. Cockburn, *Agile Software Development*, (Pearson Education, Asia, Low Price Edition, 2007)
- [5] P. Abrahamsson, O. Salo, J. Ronkainen and J. Warsta, New Directions on Agile Methods : A Comparative Study, . IEEE 25 International conference on Software Engineering, 2003
- [6] Lecture notes Helisinki University of technology Software Classification.
- [7] Information and tools about planning poker available at www.planningpoker.com.
- [8] S. Bhalerao, M. Ingle, Incorporating Vital Factors in Agile Estimation through Algorithmic Method, *International Journal of Computer Science And Applications 2009, Vol. 6, issue 1.*
- [9] Information about Agile Estimation at <http://blog.versionone.net/blog>
- [10] M. Cohen, *Agile Estimation and Planning*, (Pearson Low Price Edition Asia 2006).

- [11] P. Abrahamsson, Extreme Programming: First Results from Controlled Case Study, Proc. 29th EUROMICRO Conference New Wave of System Architecture 2003.
- [12] A. Cabri, M. Griffith, Eam Value and Agile Reporting, Proc. Agile 2006 Conference (AGILE'06).
<http://www.memory.com.uy/noticias/0247/Effort%20Estimation%20in%20Agile%20Software%20Development.pdf>.
- [13] J. Elie, Case Study: Using Agile Software Development at the Educational Credit Management Cooperation, *Agile Journal, Jul.2008*.
- [14] C. Fedrick, Case study: How Douglas County, CO Cut the Project Timeline in Half, *Agile Journal 2007*.
- [15] T. Gibb, Software Project Management: Adding Stakeholders Metric to agile Project, Upgrade e- journal <http://upggrade-cepis.org2003>
- [16] W. Jawadekar, *Software Engineering Principles and Practices*, (TataMcGraw-Hill First Edition 2005).
- [17] C. Jones, Estimating Software Costs Bringing Realism to Estimating, (Tata McGraw-Hill, 2nd Edition 2007).
- [18] M. Jorgenson, M. Shepperd, A Systematic Review of Software Development Cost Estimation Studies, *IEEE Transaction on Software Engineering 2007, Vol. 33, NO. 1, pp. 33-53*.
- [19] B. Kane, Estimating and Tracking Agile Project, Student Paper published in PM World 2007, Vol. XI issue V.
- [20] H. Leung, Z. Fan, Software Cost Estimation, <http://cs.pitt.edu/chang/handbook/42b.pdf>
- [21] M. Mohammed, *Project Management Measures in Small Enterprises*, (Master Thesis 2005, Department of Computer Science, Umea University, Sweden).
- [22] R. Pressman, *Software Engineering a Practitioner Guide*, (McGraw- Hill 6th Edition 2006).
- [23] J. Shore, S. Warden, *The Art of Agile Development*, (O'Reilly Publication, First Edition 2007).
- [24] C. Stiendl, P. Krogdahl, Estimation in Agile Projects, Proc. Best Practices in Project Estimation Conference at IBM Academy of Technology 2005.
- [25] L. Williams, L. Layman, W. Krabs, Extreme Programming evaluation framework for Object Oriented Languages version 1.4 , NCSU Computer Science TR, 2004.