2009 International Conference on Computer Engineering and Applications
IPCSIT vol.2 (2011) © (2011) IACSIT Press, Singapore

Agile Estimation using *CAEA*: A Comparative Study of Agile Projects

Shilpa Bhalerao[1], Maya Ingle[1+]

[1] Devi Ahilya Vishwavidyalaya, Indore, M.P. India

Abstract. Agile estimation approach has been a subject of active interest during the recent years due to volatile requirements of projects. Constructive Agile Estimation Algorithm (CAEA) is an algorithmic approach for estimation of cost, size and duration of project. It incorporates various vital factors namely; performance, complex processing, configuration, security, data transfer, operation ease, project domain and multiple sites. All these factors require extra efforts in development of the software thereby increasing the cost, size and duration of the project. We have estimated 87 projects from various domains to study the usefulness of CAEA for agile estimation. These projects have been classified in eight categories based upon the number of vital factors having high intensity values. It has been observed that CAEA has generated the more realistic results and eliminated the need of experienced team members and historical data. It has been noticed that CAEA is more useful in case of higher number of vital factors having high intensity values. CAEA is a step toward integrating engineering practices in agile methodology in a lighter way.

Keywords: Agile methodology, agile estimation, Constructive Agile Estimation Algorithm (CAEA), story points, planning poker.

1. Introduction

Agile methodology has become the main stream of software development due to its ability to generate higher customer satisfaction. It is fast and flexible methodology that recommends to incorporating last minute changes and requirements provided by the customer at any stage of software development phase [1] [2]. The ability to accept last minute changes generated the need of different way to estimate the Cost, Size and Duration (CSD) of project [3][4][5]. Agile estimation approach is a subject of active interest during the recent years due to its usefulness in mobile and internet technology. It is categorized in non-algorithmic and algorithmic approach and both approaches use story points as a unit for CSD estimation of agile projects. Non-algorithmic approaches include expert opinion, analogy and disaggregation that require experts and historical data to estimate precisely [7] [8]. These approaches may generate different estimates for same project depending on intuition of estimators [15]. On contrary, algorithmic approach such as Constructive Agile Estimation Algorithm *(CAEA)* incorporates vital factors for story point evaluation and generates the estimation based upon this computed new story points [6]. It includes the vital factors such as performance, complex processing, security and multiple sites etc. that affect the CSD estimation of an agile project. It has been observed that algorithmic methods generate more realistic estimates and also eliminate the need of experts [6].

We have attempted to analyze the usefulness of *CAEA* in this paper by estimating various application domain projects of small size. Section 2 provides the background of the work by defining *CAEA* and terminology algorithm in brief. Section 3 highlights the case studies considered for analyzing the usefulness of *CAEA* and results and conclusion of the study are presented in Section 4.

+ S. Bhalerao. Tel.: + (9109826687066); fax: +(912761358).
E-mail address: (bhalerao.shilpa@gmail.com)
+ M. Ingle. Tel.: + (9109893278823); fax: +(912761358).
E-mail address: (maya_ingle@rediffmail.com).

2. Background

Our study is mainly concerned with estimation of various agile projects using CAEA. Firstly, we discuss CAEA informally in Section 2.1. Subsequently, the terminology used for the previously mentioned study has been presented in Section 2.2.

2.1 Constructive Agile Estimation Algorithm (CAEA)

This algorithm is based upon two phased constructive agile estimation process. These phases are namely; Early Estimation (EE) and Iterative Estimation (IE). The purpose of EE is to identify the scope of the project by identifying the upfront with just enough requirements. On the other hand, IE is an iterative activity that starts at the beginning of iteration to incorporate new requirements/ changes. These requirements/ changes may arise from the customers with detail information of system. EE and IE use story points for CSD estimation based on size of story. *CAEA* takes input as the intensity levels of vital factors that affect CSD of project and computes the New Story Points (NSPs). Vital factors, incorporated in *CAEA* are namely; project domain, performance, configuration, data transaction, complex processing, multiple sites, operation ease and security. Each vital factor is quantified in three intensity levels mainly; low, medium and high. For example, any project with high performance, high configuration or any two vital factors having high intensity level, requires higher efforts in design, coding of the software thereby increasing the CSD of the software [6]. It is preferred to map the intensity levels with mathematical series such as square series (1, 4, 9, …) since it provides realistic level of accuracy for complex and ill-defined project [10].

Algorithm begins with identification of intensity levels of the various vital factors in terms of low, medium and high. Unadjusted Value (UV) is computed by adding up intensities of all aforesaid vital factors. *CAEA* uses work breakdown approach and decomposes the project in small tasks known as stories. Size based estimation of each story is performed and denoted as Story Point (SP). New Story Point (NSP) refers to story points computed after inclusion of vital factors and is computed by using equation (1) for each story. Further, Size of Project (SOP) and Duration of Project are computed using equation (2) and (3) respectively.

$$NSP = SP + 0.1 * UV \quad (1)$$

$$\text{Size of Project (SOP)} = \sum_{i=1}^{n} NSP_i \quad (2)$$

$$\text{Duration of project (DOP)} = SOP / velocity \quad (3)$$

where *SP* is story point of a story, *UV* is unadjusted value, NSP_i is new story point of i^{th} the story, *n* is total number of stories of project, *SOP* is size of project, *velocity* is number of *NSP* developed in a iteration.

2.2 Terminology

We have used some important terminology in our study. These are described with examples as follows:

2.2.1Application Project Domain

Our comparative study is mainly concerned with the agile projects of different application project domains such as web application, MIS projects and critical or military projects. These project domains are classified on the basis of number of tasks. For example, web application has less number of tasks as compared to military projects [13]. Therefore, fewer efforts are required to develop web application as compared to military projects. *CAEA* algorithm incorporates project domain as one of the important factor in CSD estimation of agile projects. Intensity levels have been assigned to web application as low, MIS application as medium and military or critical projects as high. *NSPs* computations for application project domains such as web application, MIS Projects and military projects have been denoted as *NSP1*, *NSP2* and *NSP3* respectively.

2.2.2Vital Factors and Intensity Levels

We have assigned the codes for vital factors performance, configuration, complex Processing, data transfer, security, multiple sites and operation ease as 1 to 7 respectively and shown in Table 1. Each of these vital factors possesses one of the intensity levels as low, medium or high as represented in Table 1.

2.2.3Categories of a Project

We have categorized the agile projects depending on the intensity levels of vital factors. Category I of agile projects is categorized by assigning the intensity levels of all vital factors as low where as category II possesses at least any one vital factor having high intensity. Category III contains the projects with any two vital factors having high intensity values etc. In this manner, we have considered total 87 agile projects of three aforesaid domains for the study of *CAEA*. A summary of agile projects with all categories and some of the cases used for the study is shown in Table 1. As shown in Table 1, category VIII includes projects with all vital factors having high intensity level and encoded as HHHHHHH.

3. CASE STUDIES AND RESULTS

We have identified eight categories of agile projects with varying number of cases in our study. In this section, we discuss the *NSP* computations using *CAEA* of eight categories of agile projects as follows:

3.1 CATEGORY I

This category includes projects of all aforesaid domains with all vital factors having low intensity as shown in Table 2. It is the smallest category having only one project from all specified application domains. Thus, the values of *NSP1, NSP2* and *NSP3* have been computed using *CAEA* as shown in Table 2. It has been observed that use of *CAEA* in such projects does not result in drastic changes in CSD estimation.

3.2 CATEGORY II

This category II deals with projects with any one vital factor as high intensity level and other vital factors with the combinations of low and medium intensity levels. Thus, projects with performance as high and all other factors such as configuration, complex processing, data transfer, security, multiple sites and operational ease at either medium or low except high and is example of this category. Table 3 shows the computation of UV and corresponding *NSPs* for some cases of category II projects using *CAEA*. It has been noticed that as any one vital factor in category I is changed from low to high, it becomes the case of category II. And hence, we observed that there is drastic change in *UV* values ranging from 7 to 15 thereby category II agile projects will always possess higher *NSP* values. As another example, HMMLLLL represents any agile project having high performance, medium configuration, medium data transfer and low values of multiple sites, operational ease and security and *UV* value for such a project is computed as 21 using *CAEA* as shown in Table 3. It has been observed that change in intensity level of one vital factor from low to high increases the size of a story to more than one point thereby affecting *DOP* and *SOP*.

3.3 CATEGORY III

CAEA takes input as two vital factors having high intensity levels and other vital factors with combination of intensity levels as low or medium for this category. It is encoded as HHXXXXX where X can take values L or M. For example, HHLLLLL, HHMLLLL, HHMMLLL, HHMMMLL, HHMMMML and HHMMMMM (when performance and configuration are having high intensity values) as shown in Table 4. UV and NSPs computation for aforesaid domains is also depicted in Table 4. It has been noticed that *NSP1*, *NSP2* and *NSP3* are computed through *UV* using *CAEA* as 28.3, 31.3 and 36.3 respectively of a story for case HHLLLLL of agile projects. We have analyzed that a small project with high performance; high configuration and medium intensity values of other vital factors has *UV* value as 38 as shown in Table 4. It has been interesting to study that lowest *UV* value in this category is the project with two vital factors having high intensity levels and others as low intensity levels (HHLLLLL) that is approximately average value of category II projects.

3.4 CATEGORY IV

CAEA algorithm uses high intensity values of any three vital factors with remaining factors at either low or medium intensity values as inputs for this category. The examples include the cases namely; HHHLLLL, LHHMLLH, LLHMMHH, MMMHHHL and MHMHMHM etc. A project with vital factors as high performance, high configuration, high data transfer; medium level of complex processing, operational ease, security and collocated team (i.e. HHHMMML) possesses UV values as 40. It has been observed that *NSPs* are increasing as intensity level of one vital factors increase from lower to medium of this category. It is evident from Table 5 that the lowest value of *UV* in this category is of the project is higher than the average value of previous category i.e. category III. The comparison of *UV* values of Table 4 and Table 5 shows that changing any four vital factors of category III from low to medium results in higher CSD estimation as compared to the lowest *UV* value of this category.

Table 1 Category Corresponding Cases of Agile Projects

Category No.	Intensity levels of Vital Factors							Cases of Agile Projects
	1	2	3	4	5	6	7	
I	L	L	L	L	L	L	L	LLLLLLL
II	H	L	L	L	L	L	L	HLLLLLL,HMLLLLL,HMMLLLL,HMMMLLL,HMMMMLL,HMMMMML,HMMMMMM
	H	M	L	L	L	L	L	
	H	M	M	L	L	L	L	
	H	M	M	M	L	L	L	
	H	M	M	M	M	L	L	
	H	M	M	M	M	M	L	
	H	M	M	M	M	M	M	
III	H	H	L	L	L	L	L	HHLLLLL,HHMLLLL,HHMMLLL,HHMMMLL,HHMMMML,HHMMMMM
	H	H	M	L	L	L	L	
	H	H	M	M	L	L	L	
	H	H	M	M	M	L	L	
	H	H	M	M	M	M	L	
	H	H	M	M	M	M	M	
IV	H	H	H	L	L	L	L	HHHLLLL, HHHMLLL,HHHMMLL,HHHMMML,HHHMMMM
	H	H	H	M	L	L	L	
	H	H	H	M	M	L	L	
	H	H	H	M	M	M	L	
	H	H	H	M	M	M	M	
V	H	H	H	H	L	L	L	HHHHLLL,HHHHMLL,HHHHMML,HHHHMMM
	H	H	H	H	M	L	L	
	H	H	H	H	M	M	L	
	H	H	H	H	M	M	M	
VI	H	H	H	H	H	L	L	HHHHHLL,HHHHHML,HHHHHMM
	H	H	H	H	H	M	L	
	H	H	H	H	H	M	M	
VII	H	H	H	H	H	H	L	HHHHHHL, HHHHHHM
	H	H	H	H	H	H	M	
VIII	H	H	H	H	H	H	H	HHHHHHH

Table 2 NSP Computations for Category I Projects

Sr. No.	CASE	UV	NSP1	NSP2	NSP3
1	LLLLLLL	7	26.7	29.7	34.7

Table 3 NSP Computation for Category II Projects

Sr. no	Cases	UV	NSP1	NSP2	NSP3
1	HLLLLLL	15	27.5	30.5	35.5
2	HMMMMMM	33	29.3	32.2	37.3
3	HMMMMML	30	29	32	37
4	HMMMLLL	24	28.4	31.4	36.4
5	HMMMMLL	27	28.7	31.7	36.7
6	HMLLLLL	18	27.8	30.8	35.8
7	HMMLLLL	21	28.1	31.1	36.1

Table 4 NSP Computations for Category III Projects

Sr. no	Cases	UV	NSP1	NSP2	NSP3
1	HHLLLLL	23	28.3	31.3	36.3
2	HMMMMHM	38	29.8	32.8	37.8
3	MHMMHML	35	29.5	32.5	37.5
4	MMMHHLL	32	29.2	32.2	37.2
5	LHMLLLH	26	28.6	31.6	36.6
6	LLMMHHL	29	28.9	31.9	36.9

Table 5 NSP Computation for Category IV Projects

Sr. no	Cases	UV	NSP1	NSP2	NSP3
1	HHHLLLL	31	29.1	32.1	37.1
2	MHMHMHM	43	30.3	33.3	38.3
3	LHHMLLH L	34	29.4	32.4	37.4
4	HHHMMML	40	30	33	38
5	LLHMMHH	37	29.7	32.7	37.7

3.5 CATEGORY V

NSP1, *NSP2* and *NSP3* values have been computed on the basis of *UV* values for the cases with any four vital factors at high intensity values as shown in Table 6. For example, a single user military project with high intensity values of performance, complexity, configuration and security and low intensity values of data transfer operation ease is candidate project of this category (may be HHHHLLL as example).

It has been observed that the increment in each *SP* results in increase of *CSD* and generates the realistic *CSD* estimates of an agile project. It is also noticed that a project with any three vital factors having high and four vital factors having medium values (i.e.- MHHMHMM) has higher *UV* as compared to the project having four vital factors having high, two vital factor having low and remaining one vital factors having medium intensity (i.e. HHHHMLL). This fact reveals that increase in intensity level of any one vital factor from medium to high causes decrease in CSD estimated values of the project in which any two intensity levels of vital factors decreased from higher level to lower level.

3.6 CATEGORY VI

This category consists of projects with any five vital factors at high intensity levels and other vital factors at either low or medium intensity levels as shown in Table 7. A web application possessing aforesaid characteristics is an example of this category. For such projects we have observed that *CAEA* generates CSD estimation approximately twice as compared to previously size based story point estimation.

3.7 CATEGORY VII

This category includes projects having any six vital factors at high level and one vital factor is either low or medium intensity level. *UV* computation for this project category along

NSP computations for web based applications, MIS and critical projects are shown in Table 8. It is has been found that vital factors with high intensity levels increase the *UV* value by at least 5 points as compared to previous category of the projects.

3.8 CATEGORY VIII

Project case with all vital factors having high intensity is considered to the only case of this category (i.e. HHHHHHH) with *UV* value as 63 as shown in Table 9. It is evident that military project with high intensity of all vital factors has the highest *NSP* value in all the three project domains as well as all of the previous categories thereby increasing the CSD estimates

4. CONCLUSION

We have presented the study of various categories of projects using *CAEA*. The study included 29 *UV* computations with total 87 projects from three domains and corresponding *NSPs* estimates. A summary of *NSP* computations of these categories with some cases has been shown in Table 10. Similarly, *UV* values in each category have been depicted in Fig. 1. It has been noticed that as number of vital factors with high intensity increases, the *UV* range and corresponding *NSP* ranges decrease meaning that vital factors at high intensity level have more impact on CSD estimation a project as compared to other intensity levels. At the same time, we claim that *CAEA* is useful particularly at time of unavailability of historical data and expert estimators.

We have observed following facts and observation from the study:

- It has been noticed that *CAEA* computes estimation based upon intensity values of number of vital factors. Thus, CSD estimation of project having HLLLLLL intensity level is same as project having LLLLLLLH.
- *CAEA* algorithm is particularly useful and generates the realistic results when the number of vital factors having high intensity values increases. Thus, we can conclude that faith and trust among the customer on agile estimates will be increased.
- It is evident from the Fig.1 that with the increasing number of vital factors having high intensity values, the number of projects in that category decreases.
- Applicability of an algorithm is beneficial in absence of historical data and expert estimators. Thus, average project leader can provide the estimates by applying proposed algorithm.
- Algorithm generates higher deviations for projects having all or maximum number of vital factors having low intensity values and hence algorithmic method may be avoided for these types of projects.
- Algorithmic approach for estimation of agile projects leads to a step towards the engineering practices thereby establishing the fact that these methods are not ad-hoc methods.

Although, *CAEA* computes the CSD estimates on the basis of number of intensity levels of vital factors and not given preferences to type of vital factors. Still, it has computed the realistic estimates. This algorithm is a step towards developing a mathematical model for agile estimation. However, much work remains to be done to validate and extend this algorithm.

Table 6 NSP Computations for Category V Projects

Sr. no	Cases	UV	NSP1	NSP2	NSP3
1	HHHHLLL	39	29.9	32.9	37.9
2	MHHMHLH	45	30.5	33.5	38.5
3	Hj a t 15u 8i0w2 261qMHMHMH	48	30.8	33.8	38.8
4	HLLHMHH	42	30.2	33.2	38.2

Table 7 NSP Computations for Category VI Projects

Sr. no	Cases	UV	Nsp1	NSP2	NSP3
1	HHHHHLL	47	30.7	33.7	38.7
2	HMHHHHM	52	31.2	34.2	39.2
3	HHHMHLH	50	31	34	39

Table 8 NSP Computations for Category VII Projects

Sr. no	Cases	UV	NSP1	NSP2	NSP3
1	HHHLHHH	55	31.5	34.5	39.5
2	HHHHHHM	58	31.8	34.8	39.8

Table 9 NSP Computations for Category VIII Projects

Sr. no	Cases	UV	NSP1	NSP2	NSP3
1	HHHHHHH	63	32.3	35.3	40.3

Table 10 Summary of NSP Computation of various Categories

Category	Cases	NSP Computations Range		
		NSP1	NSP2	NSP3
I	LLLLLLL	26.7	29.7	34.7
II	HXXXXXX	27.5-29.3	30.5-32.3	35.5-37.3
III	HHXXXXX	28.3- 29.8	31.3-32.8	36.3- 37.8
IV	HHHXXXX	29.1-30.3	32.1-33.3	37.1-38.3
V	HHHHXXX	29.9-30.8	32.9-33.8	37.9-38.8
VI	HHHHHXX	30.7-31.2	33.7-34.2	38.7-39.2
VII	HHHHHHX	31.5-31.8	34.5-34.8	39.5-39.8
VIII	HHHHHHH	32.3	35.3	40.3

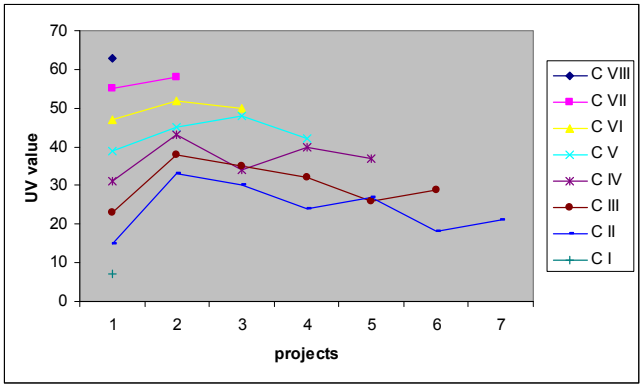

Fig. 1: UV Values of Various Project Categories

5. REFERENCES:

[1] K. Beck, Extreme Programming Explained, Pearson Education Low price Edition Asia, 2006.

[2] A. Cockburn, Agile Software Development, Pearson Education, Asia, Low Price Edition, 2007.

[3] P. Abrahamsson, O. Salo, J. Ronkainen and J. Warsta, "New Directions on Agile Methods : A Comparative Analysis", Proc. IEEE 25th International conference on Software Engineering 2003.

[4] T. Gruschke, Empirical Studies on Software Cost Estimation: Training of Effort Estimation Uncertainty Assessment Skills, Proc. 11th IEEE International Software metric Symposium.

[5] Information and tools about planning poker available at www.planningpoker.com.

[6] S. Bhalerao, M. Ingle, Incorporating Vital Factors in Agile Estimation through Algorithmic Method, International Journal of Computer Science and Applications Vol. 6, issue 1 pp.85-97.

[7] Information about Agile Estimation at http://blog.versionone.net/blog

[8] M. Cohen, Agile Estimation and Planning, Pearson Low Price Edition Asia 2006.

[9] P. Abrahamsson, "Extreme Programming: First Results from Controlled Case Study", Proc. 29th EUROMICRO Conference New Wave of System Architecture 2003.

[10] A. Cabri, M. Griffith,,Earn Value and Agile Reporting, Proc. Agile 2006 Conference(AGILE'06).

http://www.memory.com.uy/noticias/0247/Effort%20Estimation%20in%20Agile%20Software%20Development.pdf.

[11] J. Elie, Case Study: Using Agile Software Development at the Educational Credit Management Cooperation, Agile Journal, Jul.2008.

[12] C. Fedrick, Case study: How Douglas County, CO Cut the Project Timeline in Half, Agile Journal 2007

[13] T. Gibb, "Software Project Management: Adding Stakeholders Metric to agile Project", Upgrade e- journal http://upggrade-cepis.org 2003.

[14] W. Jawadekar, Software Engineering Principles and Practices, Tata McGraw-Hill First Edition 2005.

[15] C. Jones, Estimating Software Costs Bringing Realism to Estimating, Tata McGraw-Hill, 2nd Edition 2007.

[16] 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.

[17] B. Kane, Estimating and Tracking Agile Project, Student Paper published in PM World 2007, Vol. XI issue V.

[18] H. Leung, Z. Fan, Software Cost Estimation, ftp://cs.pitt.edu/chang/handbook/42b.pdf

[19] M. Mohammed, Project Management Measures in Small Enterprises, Master Thesis 2005, Department of Computer Science, Umea University, Sweden.

[20] R. Pressman, Software Engineering A Practitioner Guide, McGraw- Hill 6th Edition 2006.

[21] J. Shore, S. Warden, The Art of Agile Development,, O'Reily Publication, First Edition 2007.

[22] C. Stiendl, P. Krogdahl, Estimation in Agile Projects, Proc. Best Practices, Project Estimation Conference at IBM Academy of Technology 2005.