An Empirical Analysis of Effort Distribution of Small Real-Client Projects

Ali Afzal Malik, Barry W. Boehm

Center for Systems and Software Engineering – University of Southern California
Los Angeles, CA 90089, U. S. A.

`alimalik@usc.edu, boehm@usc.edu`

Abstract. *This paper presents an analysis of the weekly effort distribution of 23 small real-client projects. The procedure used in collecting the effort data is described and the major findings are summarized. The results indicate the crests and troughs of project effort in relation to the major project milestones. Possible reasons for the pecularities in the effort distribution are also discussed. Moreover, an attempt is made to analyze the impact of project type on the effort distribution by comparing the effort profiles of web-based projects with non-web-based projects.*

Keywords: effort distribution, empirical study, project milestone, software cost estimation

1. Introduction

Software development effort is one of the most important outputs of a software cost estimation model like COCOMO II [Boehm et al. 2000]. While models like COCOMO II provide estimates of the distribution of effort among the major life cycle phases (e.g. Inception, Elaboration, etc.) they do not foretell the distribution of effort down to the level of a week.

While estimates of weekly effort distribution are almost impossible to obtain, it is much easier to get the actual weekly effort distributions of completed projects by, for instance, querying their effort logs. These weekly effort distributions enable us to conduct a much finer-grained analysis of these projects in terms of the variation in their effort relative to different phases and milestones.

An opportunity to conduct such an analysis for small real-client projects is provided by USC's annual two-semester-long series of graduate-level, team-project-based software engineering courses. The first course - Software Engineering I (SE I) – is offered in the Fall semester while the second course – Software Engineering II (SE II) – is offered during the Spring semester. SE I exposes the graduate students to the first two phases of RUP [Kruchten 2003] i.e. Inception and Elaboration and SE II allows them to experience the last two phases i.e. Construction and Transition.

Typical team strength of these small real-client projects is between 6 and 8 graduate students with two being off-campus students. These off-campus students are assigned the job of independent verification and validation (IV & V) of these projects. Typical clients include neighborhood organizations (e.g. California Science Center) and university departments (e.g. USC Libraries). This empirical study analyzes the weekly

effort distribution of 23 such projects [SE I 2008, SE II 2008] done at USC during the past few years.

The remaining paper is organized as follows. Section 2 compares and contrasts our work with the most relevant previous work in this area. Section 3 presents the methodology used for gathering and analyzing the weekly effort data. Section 4 contains a summary of the results while Section 5 presents a qualitative discussion on the salient aspects of these results. Finally, Section 6 concludes this empirical study with a brief summary of our major findings and mentions some of our plans for future work in this area.

2. Related Work

This empirical study is a continuation of the work reported by Zhihao Chen in [Chen 2005]. While the previous work analyzed the effort data of 29 small real-client projects done between Fall 2001 and Spring 2004 this study presents the analysis of 23 similar projects done between Fall 2005 and Spring 2008. While there are many similarities between this work and the previous, there are some important differences.

First and foremost, the projects selected for this empirical study are much more similar to each other. For instance, all projects in this study are custom-development projects (as opposed to being a mix of COTS-based and custom-development in the previous work). Secondly, the software development process followed by these 23 newer projects is a leaner version of the process used by the older 29 projects.

Another important difference is that our work analyzes the overall effort of the projects while the previous work analyzed the different components (e.g. Design, Implementation, etc.) and subcomponents (e.g. Modeling for SSAD, Component Prototyping, etc.) of the overall effort. Last, but not the least, we look at the impact of project type (web-based vs. non-web-based) on the effort distribution – something not considered in the previous work.

3. Methodology

We analyzed the weekly effort distribution of 23 small real-client projects done between Fall 2005 and Spring 2008. These projects are listed in Table 1. Each of these projects followed the same LeanMBASE/RUP [Boehm et al. 2005, Kruchten 2003] software development process and was completed in two semesters. For the purposes of this empirical study we restricted ourselves to analyzing custom-development projects only. In other words, none of the 23 projects we selected for this empirical study was a COTS-based project.

Table 1. Projects summary

S#	Year	Project	Type
1	2005	Data Mining PubMed Results	Data mining
2	2005	USC Football Recruiting Database	Web-based database
3	2005	Code Generator – Template based	Stand-alone application
4	2005	Develop a Web Based XML Editing Tool	Web-based application
5	2005	EBay Notification System	Stand-alone application
6	2005	Rule-based Editor	GUI
7	2005	CodeCount™ Product Line with XML and C++	Code Counter Tool
8	2006	California Science Center Newsletter System	Web-based database
9	2006	California Science Center Event RSVP System	Web-based database
10	2006	USC Diploma Order/ Tracking Database System	Web-based database
11	2006	USC Civic and Community Relations (CCR) web application	Web-based database
12	2006	New Economics for Woman (NEW)	Web-based database
13	2006	Web Portal for USC Electronic Resources	Web-based GUI
14	2006	Early Medieval East Asian Tombs	Web-based database
15	2006	USC CONIPMO	Cost model
16	2006	An Eclipse Plug-in for Use Case Authoring	Stand-alone application
17	2007	USC COINCOMO	Cost model
18	2007	BTI Appraisal Projects	Stand-alone database
19	2007	LAMAS Customer Service Application	Web-based database
20	2007	BID review System	Stand-alone database
21	2007	Proctor and Test Site Tracking System	Web-based database
22	2007	E-Mentoring program	Web-based application
23	2007	Los Angeles County Generation Web Initiative	Web-based database

Since each of these projects was completed in two distinct semesters each with its own separate phases and milestones we broke the analysis of effort distribution into two roughly equal parts. The first part analyzes the effort distribution during the Inception and Elaboration phases (Fall semester) while the second one examines the effort distribution during the Construction and Transition phases (Spring semester).

One of the most important decisions in gathering the effort data was to select the right source of data. Among the two available options, one was the weekly progress reports [Progress Report Template 2004] submitted by the project teams and the other was the database of the Effort Reporting System [ER System 2007]. Even though it was easier to obtain data from the progress reports which were uploaded by teams on their project websites we decided to obtain data from the effort database. This was done due to two main reasons – completeness and accuracy. Progress reports are submitted by project managers after consultation with the other team members. Each team member (including

the project manager) logs his or her own effort (under various effort categories e.g. Design, Life Cycle Planning, Implementation, Deployment, etc.) in the Effort Reporting System and then manually reports the overall effort to the project manager. The project manager then compiles these individual reports and enters a single value for the team's weekly effort in the progress report. This approach is vulnerable to errors of misreporting and miscommunication. For these reasons we decided to query the effort database. Later on it was confirmed that even though this approach was time consuming (we had to run a separate SQL query for each semester of every project) it was much more accurate. For over half of the 23 project teams there was a significant difference between the effort they had reported in the weekly progress report and the effort they had logged in the Effort Reporting System. The second main reason to go with the effort database was the fact that not all teams had uploaded all of their progress reports on their website even though they had logged the effort values in the Effort Reporting System. The effort database, on the other had, contained complete effort information for every project.

After collecting the effort data for each project we analyzed the average effort for each week. Since each of these 23 projects had a lot in common (e.g. duration, size, software development process, etc.) the average turned out to be a good measure of the centrality of the effort data. As a matter of fact, the median and average are almost the same for each week in the Fall semester (see Figure 1) and very similar for weeks in the Spring semester (see Figure 2).

The variation in the weekly average effort was analyzed in relation to the major milestones such as those mentioned in [Boehm 1996]. Table 2 summarizes the major milestones we used for reference in each of the two semesters. Note that the Spring Break has also been included in these milestones to determine the impact on effort due to this mid-semester recess. The LCO, LCA, and RLCA milestones mark the due dates of their respective packages i.e. LCO package, LCA package, and RLCA package. The CCD milestone marks the end of the CCD while the IOC milestone indicates the last IOC working set. Since not all projects start and end at precisely the same week these milestones help in anchoring the effort data and aligning the weeks across different projects.

Table 2. Milestones in each semester

Semester	Milestone	Week
Fall (SE I)	Life Cycle Objectives (LCO)	7
	Life Cycle Architecture (LCA)	13
Spring (SE II)	Rebaselined Life Cycle Architecture (RLCA)	5
	Spring Break	8
	Core Capability Drivethrough (CCD)	10
	Initial Operational Capability (IOC)	14

We also tried to look at the variability in the distribution of average weekly effort based on the category of projects. Two broad categories of projects – web-based and non-

web-based – were considered. 13 out of the 23 projects were web-based (gray rows in Table 1).

4. Results

Figures 1 and 2 summarize the weekly effort distribution of these 23 projects. Each of these two figures annotates the effort distribution with milestones (see Table 2) for reference. Both the average and the median effort distributions have been shown in these figures. Figure 1 shows the effort distributions in the Fall semester while Figure 2 shows the effort distributions for the Spring semester.
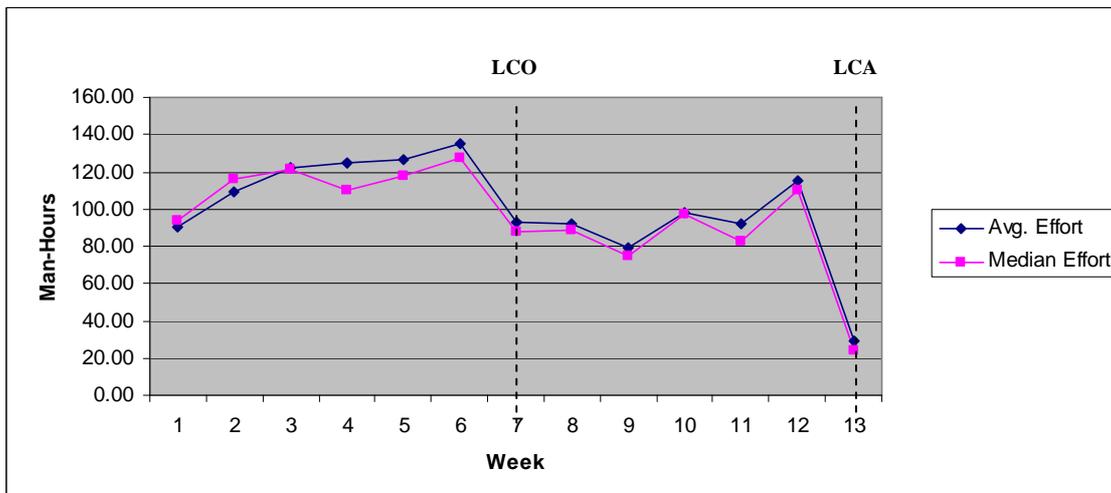

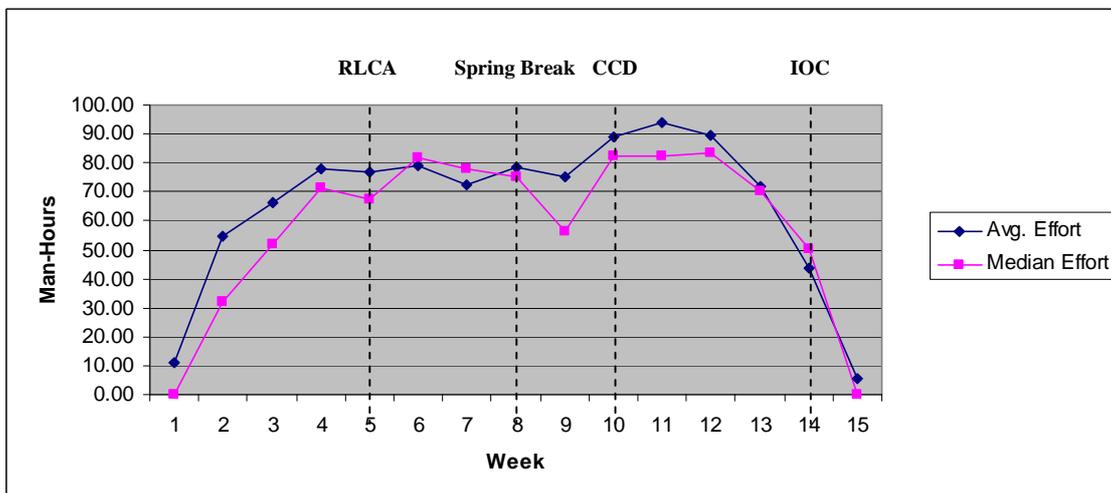**Figure 1. Weekly effort - Fall (SE I)**

Figure 2. Weekly effort - Spring (SE II)

While Figures 1 and 2 show the average (and median) weekly effort distribution for all 23 projects combined Figures 3 and 4 show the average weekly effort distribution separately for web-based and non-web-based projects. The two effort distributions for the Fall semester are shown in Figure 3 while the two effort distributions for the Spring semester are shown in Figure 4.
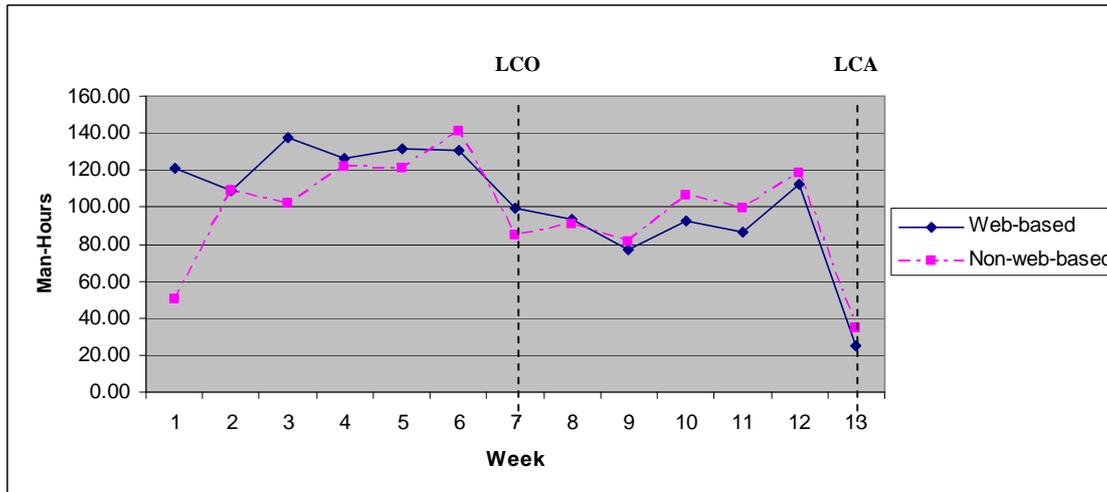
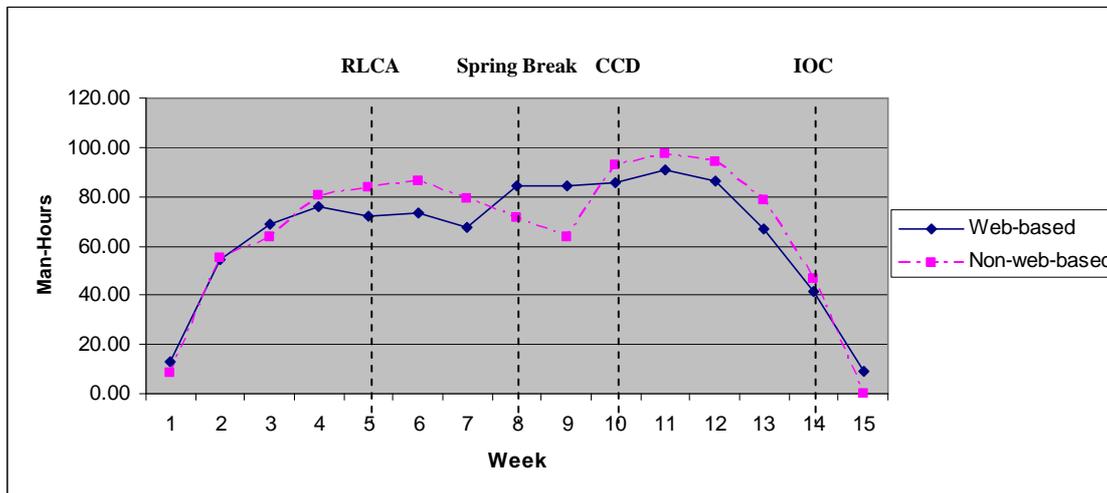**Figure 3. Average weekly effort by project type - Fall (SE I)**

Figure 4. Average weekly effort by project type - Spring (SE II)

5. Discussion

The results shown in the previous section indicate a number of important points. A quick glance at Figures 1 and 2 reveals that there is no significant difference between the average and the median weekly effort distributions. As mentioned earlier in Section 3 (Methodology) this is probably due to a large number of similarities between the projects. Either of the two (average or median), therefore, can be used for analyses. We have used the average effort values. Thus all subsequent references to effort are actually references to average effort.

Figure 1 indicates that the effort in the first few weeks of the project is significantly higher. In fact, it gradually rises till it reaches a peak right before the LCO milestone. This could, primarily, be due to the steep learning curve for these projects at the time of Inception. Another possible reason for this high effort during the initial period could be the fact that teams have a tendency to over-specify the artifacts in the LCO

package. This may be due to the incentive structures – students try to get good grades by playing safe and over-specifying the information in the project artifacts.

Another phenomenon displayed by Figures 1 and 2 is the "deadline effect". Effort peaks around the due dates of LCO package, LCA package, RLCA package, and CCD. The only exception is the last IOC working set milestone. The effort around this milestone drops significantly. This may be due to the reason that the project team members are busy with preparing for and taking the final exams for the other courses they are enrolled in.

It is quite surprising to see that there is no major change in effort during the mid-semester Spring recess. The effort during this recess stays almost the same as the effort spent during adjacent weeks. This may be explained by reasoning that the students utilize the time of the recess to finish the Construction phase of the project so that they can start the Transition phase on time.

A comparison of the effort distribution for web-based and non-web-based projects is shown in Figures 3 and 4. As shown by these two figures there is not much of a difference between the effort distributions for these two main categories of projects. Apart from a few exceptions, the two effort distributions rise and fall almost synchronously. This may be due to the fact that both types of projects undergo a stage called "win-win negotiation" at the start of the project wherein the success-critical stakeholders come up with a prioritized list of requirements in order to facilitate the completion within two semesters. Mid-semester rebaselining of projects also achieves the same goal.

6. Conclusions and Future Work

The above analysis of the weekly effort distribution of 23 small real-client projects has brought to light some important points. We now know (for such projects) when the effort peaks and plummets and what could be the possible reasons for this. Moreover, our evidence suggests that the effort distribution stays almost the same irrespective of the type of the project (web-based or non-web-based). Hopefully, teams undertaking similar projects in the future will benefit from these results and consider them while planning their projects.

While the results obtained so far offer valuable insights, the analysis of effort distribution of these small real-client projects is by no means complete. As stated earlier, this empirical study has concentrated solely on analyzing the weekly effort distribution of small custom-development projects. The next logical step is to conduct the same analyses on similar-sized COTS-based projects. Since COTS-based projects involve a significantly different set of activities vis-à-vis custom-development projects [Boehm et al. 2003] one would expect their effort distribution to be significantly different. An empirical study of such projects will help us in determining the exact differences.

Another aspect worth investigating is the evolution of effort distribution as various changes were made to the software development processes. Prior evidence [Koolmanojwong and Boehm 2007] suggests that a switch to the light-weight version of the MBASE [Boehm et al. 2004] software development process – LeanMBASE [Boehm et al. 2005] – played a key role in reducing the effort required to produce the

documentation of such small real-client projects. A thorough comparison of the weekly effort distribution before and after such process shifts will enable us to focus on some novel aspects of such changes. For instance, not only will we be able to see the big picture and determine the impact on the overall effort of such projects but also the micro-level variations in the effort spent while reaching different project milestones.

7. References

Boehm, B. (1996). "Anchoring the Software Process", *IEEE Software 13(4)*, pages 73–82.

Boehm, B., Abts, C., Brown, A., Chulani, S., Clark, B, Horowitz, E., Madachy, R., Reifer, D., and Steece, B. (2000), Software Cost Estimation with COCOMO II, Prentice Hall.

Boehm, B., Brown, A., Port, D., et al. (2004). "Guidelines for Model-Based (System) Architecting and Software Engineering (MBASE)", Center for Software Engineering, University of Southern California.

Boehm, B., Klappholz, D., Colbert, E., et al. (2005). "Guidelines for Lean Model-Based (System) Architecting and Software Engineering (LeanMBASE)", Center for Software Engineering, University of Southern California.

Boehm, B., Port, D., Yang, Y., Bhuta, J., and Abts, C. (2003). "Composable Process Elements for Developing COTS-Based Applications", *2003 International Symposium on Empirical Software Engineering (ISESE'03)*.

Chen, Z. (2005). "Software Engineering Graduate Project Effort Analysis Report", Technical Report, Center for Software Engineering, University of Southern California.

ER System (2007), http://greenbay.usc.edu/csci577/tools/ER/ER_Instructions.pdf

Koolmanojwong, S. and Boehm, B. (2007). "An Empirical Study on MBASE and LeanMBASE", *1st International Symposium on Empirical Software Engineering and Measurement (ESEM'07)*.

Kruchten, P. (2003), The Rational Unified Process: An Introduction, Addison-Wesley.

Progress Report Template (2004), http://sunset.usc.edu/classes/cs577a_2004/assignments/Team/Weekxx_PR_F04a_Txx.xls

SE I (2008). Links to websites of all past semesters of Software Engineering I (CSCI 577A) course at USC, http://sunset.usc.edu/csse/courseroot/course_list.html#577a

SE II (2008). Links to websites of all past semesters of Software Engineering II (CSCI 577B) course at USC, http://sunset.usc.edu/csse/courseroot/course_list.html#577b