Product Development Time Improvement with Requirements Reuse

Semra Yilmaz Tastekin
Informatics Institute
Middle East Technical University
Ankara, Turkey
e-mail: syilmaz@aselsan.com.tr

Yusuf Murat Erten
Innova Inc.
Ankara, Turkey
e-mail: merten@innova.com.tr

Semih Bilgen
Electrical and Electronics Engineering
Middle East Technical University
Ankara, Turkey
e-mail: bilgen@eee.metu.edu.tr

***Abstract*—Product development time estimation is important for project management tasks. This study investigates the impact of requirements reuse on product development duration for different products in a similar domain. We propose an analytical tool to estimate the minimum time to be saved given the percentage of requirements reused from earlier projects. Empirical results of industrial case studies are used as inputs to this study. Three cases from different organizations have been studied for software and system development projects, which consist of hardware and software components. The results of the case studies are compared with a study in the literature on product development time. According to the industrial empirical results, a modified product development time estimation method is proposed for software projects.**

Keywords - requirements reuse; product development time.

I. Introduction

In most organizations, project deliverables are generally systems that combine hardware and software components. For system projects, which include hardware and software components, software is becoming an increasingly significant constituent [1][2] and project management is accordingly becoming more complex.

Engineers working in these projects discover most of the problems at the integration phase. Isolation of the source of these problems at this stage can take time and this may affect the project duration. According to [3], 50% of total time and cost of a project is spent for testing. So, minimizing possible system faults at earlier stages will minimize the test efforts. Since most defects are found in the integration and test phases, these phases are generally stressful for developers and testers since they are responsible for correcting the defects. Therefore, reuse of the components created during each phase of different projects has an important role in eliminating the defects in the product, correspondingly reducing the engineering effort in the project.

This study focuses on reducing some of the sources of defects hence the product development time via requirements reuse. The motivation behind the reuse of requirements items from previous projects is that these are already validated and accepted by end users in previous projects. In this study, three case studies are presented to demonstrate the benefits of requirements reuse for a system project (containing both hardware and software) and a software product. In the literature, there are some studies on the reuse of some components and they have generally been performed for software code reuse. One of the contributions of the present study is to provide empirical results, which demonstrate the reduction of project duration due to requirements reuse for industrial products. Another contribution is to show that the product development time estimations proposed earlier in the literature for manufacturing industries [19] can be applied to system and software projects. We also propose a modification to the current formulation to represent software-based projects more accurately [19].

This paper is organized as follows. Section II briefly reviews the software product line literature, focusing specifically on product development time. Section III discusses the impact of requirements reuse in the context of industrial case studies. Section IV presents a modified product development time proposal. Section V concludes the paper and suggests future directions for reuse in project management.

II. Related Work

This section reviews the literature on Software Product Line (SPL) and product development time.

A. Software Product Line

When reuse is applied in all stages of the product development cycle, this corresponds to the product line concept. The aim of the product line concept is to enhance the quality of the product and decrease the engineering cost. In the literature, there are many studies, which evaluate the establishment of software and system product lines in organizations [5][6][9][10].

The accepted SPL approach is based on two-life cycles, which are Domain Engineering (DE) and Application Engineering (AE). SPL mostly focuses on the DE activities [5], which create a common infrastructure for the related domain. On the other hand, AE is active when there is a new project. To produce a product, the required assets are selected from the common assets created by DE. For the remaining part of the product, new assets are created from scratch. Although SPL applies to all phases of projects life cycle, this study covers only the requirements definition phase. Thus, in the following

 ISBN: 978-1-61208-230-1

sections, requirements definition phase, which is a part of SPL, is studied.

B. Product Development Time

While preparing project proposals, little information is available concerning the development details. Before starting the development, it is important to have information about development time to estimate the cost of the project used for proposals.

In the literature, there is some information about the methods for estimating the duration for product development. Griffin [18] classifies the metrics, which affect the development time in four groups as;

- Changes during the product generation,
- The complexity of product,
- Whether a formal process is used in the organization, and
- Whether a cross functional team is used.

Changes in requirements have a considerable effect on the development workload ([11][20][21]). Therefore, it is important to define the requirements accurately to ensure minimum change during the development stage. The number of main functions the product performs gives the complexity of the product [19]. If organizations do not have formal development processes, the development time is higher compared to those with formal development processes. A study by Olson et al. [12] emphasizes that the use of a cross functional team is also an important parameter in increasing project performance.

Griffin [18][19] defines Development Time (DT) and Concept To Customer (CTC) as two separate parameters. DT begins from design up to the introduction to the customer and CTC begins with concept development and continues to specification definition until the introduction to the customer. Requirements engineering activities are covered within CTC. If DT is subtracted from CTC this will give the time spent for requirements engineering activities. DT and CTC formulations proposed for the manufacturing industries are given below [19].

$$DT = \alpha + \beta_{1DT} * PC + \beta_{2DT} * NN + \beta_{3DT} * (PC * FP) + \beta_{4DT} * (NN * XFT) + \epsilon_{DT} \quad (1)$$

$$CTC = \alpha + \beta_{1CTC} * PC + \beta_{2CTC} * NN + \beta_{3CTC} *(PC * FP) + \beta_{4CTC} * (NN * XFT) + \epsilon_{CTC} \quad (2)$$

where α is the cycle time constant, PC and NN are product complexity and product newness/uncertainty, respectively. If NN value increases, the change probability on the product during the development also increases. FP and XFT show whether formal processes and cross functional teams are used, respectively. If formal development processes are not used, then FP=0. ϵ is the error term. The unit of β_1 and β_3 are the months/function designed in the product. The unit of β_2 and β_4 are the months/percentage of change in the product. The estimation of the coefficients α and β, based on data collected from many companies, is given in Table I [19].

TABLE I. COEFFICIENTS USED IN THE DT AND CTC FORMULATIONS

	α	β_1	β_2	β_3	β_4
DT	8.4	4.2	0.09	-1.9	-0.09
CTC	10.4	3.7	0.16	0.1	-0.16

III. PROBLEM DEFINITION AND QUANTITATIVE DATA

The requirements phase will have the greatest impact on the subsequent steps in the project life cycle. If there are defects in the requirements phase, its negative effect is reflected on the later phases. Considering that 7-15% of the total project resources are used for requirements engineering [14][15], requirements-related phases of the development lifecycle should be realized as effectively as possible. Besides, the customer generally needs the product within the shortest possible time. This puts pressure on the project manager and engineers to find ways to shorten the project duration. This can be achieved by shortening the phases in the life cycle.

The requirements engineering stage can be shortened by effective reuse of requirements, which are generated at the initial step of the project. In the final step, test activities are performed to determine whether the system meets defined requirements. Once the system is accepted by the customer, all the requirements defined at the beginning of the project are tested and approved. So, reusing these requirements in other projects will enable easy acceptance of the same features in similar systems. This will increase the quality of the requirements and will reduce the project duration [8][13][22].

In the literature, there are many studies discussing guidelines for reuse but few compare project development time with reuse and without reuse. For example, Johnson [23] proposes a model, which integrates SPL with a software development environment; however, this study does not have measured results. Similarly, Catal and Diri [4] proposes a framework for software fault prediction in SPL but does not present an analysis due to the lack of fault data. There are several theoretical studies concerning software components reuse [6][7][8][16][17], but little research about hardware components. In the present study, the requirements engineering phase in software intensive systems projects is analyzed for two different organizations using a reuse approach. Following paragraphs present the quantitative data from these organizations. The results of the case studies are compared with the results of Griffin's study [19]. Of the three case studies the first involved the reuse of the requirements for a system with hardware and software functionalities. The second and third cases involved the reuse of the requirements for software products. Cross functional teams are not used for the projects in these case studies.

A. Characteristics of the Case Studies

To analyze the product development time using the requirements reuse approach, it was intended to gather data from different companies and the requirements engineering phase would be analyzed for three different organizations. Data is collected with a joint effort of the project technical managers and the authors. Unfortunately, there were some difficulties in gathering data from different companies. First, organizations generally do not keep the project related data in a systematic way. To overcome this difficulty, some interviews with the project technical managers were undertaken and the related data were collected using related documents and organizational database. Second, even when metrics were kept systematically, the organizations would prefer not to release this data for external use.

The data collected in this study is quantitative ([24]). The necessary data to perform the case studies include the total number of requirements, the number of requirements reused from a previous project and the duration of the requirements definition phase. These data are presented in tabular format below (Table II, Table III, Table V, Table VI, Table VII, Table VIII, Table X, Table XI).

The project related data are derived from the responses to the questions given below, following the methods as described:

Question 1: Are there any similar products which can be in the same domain or are derivative products in the company?

Method Used for Answering Question 1: Discussions with different project technical managers from different companies are performed and the details of the projects are evaluated.

Question 2: Is there recorded data for the number of requirements for each project in the same domain?

Method Used for Answering Question 2: The System Requirements Documents for each project are used to obtain the necessary data.

Question 3: Is there recorded data for reused requirements?

Method Used for Answering Question 3: If the metrics are kept systematically, data is retrieved from organization database. If they are not kept in an organizational database, reused requirements are derived from the system requirements documents by technical personnel involved in the projects.

Question 4: Is there duration data for requirement definition phases?

Method Used for Answering Question 4: The enterprise resource planning systems of the companies are used to extract this data.

Question 5: What is the complexity level of the product to be studied?

Method Used for Answering Question 5: As defined in [19], the main functions of the products are defined to obtain the complexity level of the product with the help of technical managers of the projects. If there is critical technology to be developed within the scope of the product, this factor is also added to the complexity. For example, if the product has 4 main functions and includes 2 technological infrastructure developments, the complexity level is defined for this product as 6.

The data from the case studies are used for the following purposes;

- Deriving the realized duration for requirement definition phases of system products and software products,
- The comparison of durations for different projects in the same domain,
- Comparison of realized duration with the theoretically predicted product development time ([19]),
- Separately analyzing the software products and system products,
- Definition of any necessary modification to product development time proposed in [19] if required.

Using Griffin's study [19] and case studies data from current research, the following issues will be resolved at the end of this study;

- The product development time proposed in [19] was appropriate for system products which involve hardware and software components,
- But product development time proposed in [19] was not appropriate for software products,
- There was a need to modify the product development time prediction for more accurate estimations for software products.

B. *Case Study-1*

Project A1 and Project A2 in the same domain from Company A were analyzed. Some of the requirements of Project A1 have been reused in Project A2. Table II shows the number of requirements in Project A2. Project A2 had 183 requirements in total. 104 requirements of those were reused from Project A1. Remaining 79 requirements were created from scratch for Project A2. The realized duration for requirements engineering (RE) activities for both projects is given in Table III.

Change probability of 104 reused requirements was very low in Project A2, because they were tested, and approved by the customer previously. This implies that;

- 57% of total requirements (104 requirements) for Project A2 were almost fixed.
- 43% of total requirements (79 new requirements) could still be changed in Project A2.

By reusing the requirements, change probability in the product (NN) is minimized. While NN varies between 0% and 100%, by reusing the requirements it can be decreased in the range of 0% to 43% for Project A2. By using Griffin's CTC formulation (2), this situation indicates that for all possible changes in the requirements, the organizations would require an additional 16 months ($\beta_{2CTC} * NN = 0.16 * 100\%$) if requirements were not reused. On the other hand, the organization would only require an additional 6.88 months (0.16*43%) maximum when requirements were reused. So, change effect is reduced by 9.12 months for Case Study-1.

TABLE II. REQUIREMENTS USED IN PROJECT A2

	Total # of Req. in Project A2	Req. of Project A1 Reused in Project A2
Total	183	104 (57% of Project A2)

TABLE III. DURATION EXPENDED IN PROJECT A1 AND PROJECT A2

	Project A1	Project A2	Possible Impact of Reuse
Total Duration (months)	8	5	37 % decrease in duration

To estimate the time spent for RE activities, the calculations of DT and CTC given in Appendix-A for 100% and 43% cases are performed using (1) and (2). 100% indicates that product requirements/features were totally new. 43% indicates the amount of new requirements, and is taken as the change probability of the requirements. The complexity level of the product developed in the scope of Project A2 was taken as 6 based on the number of main functions the product has. Eventually, calculated time spent for RE activities is summarized in Table IV.

Even if the maximum change (43%) occurs in the requirements, there would be at least 22% decrease (from 18 months to 14.01 months) in the duration of RE activities. If the change in the requirements were less than 43% change, the improvement would be expected to be greater than 22%.

When this result is compared with the actual results of Case Study-1 in Table III, the decrease in the Project A2 shows agreement with these calculations. Griffin's formulation predicts at least 22% reduction in duration, likewise a reduction of 37% (more than 22%) was obtained. Thus, formulations proposed by Griffin for the estimation of project duration applies for the system, which involves both hardware and software.

C. Case Study-2

For this case, three software projects from Company B were analyzed; Projects B2 and B3 used Project B1 as a baseline. Table V and Table VI show the number of requirements in Project B2 and Project B3. Project B2 had 376 requirements in total. 314 of those were reused from Project B1. Remaining 62 requirements were created from scratch for Project B2. Similarly, Project B3 had totally 323 requirements, 230 of those were reused from Project B1 and 93 new requirements were created. For the requirements engineering phase of both projects, the duration data are given in Table VII and Table VIII.

As shown in Table V, for Project B2;

- 84% of total requirements (314 requirements) were almost fixed and change probability of those was very low.
- 16% of total requirements (62 new requirements) could still be changed during the product cycle time.

Similarly for Project B3, as shown in Table VI;

- 71% of total requirements (230 requirements) were almost fixed and change probability of those was very low.
- 29% of total requirements (93 new requirements) could still be changed during the product cycle time.

When the requirements are reused, changes in product could be decreased in the range of 0% to 16% for Project B2 and 0% to 29% for Project B3. Using (2), if the requirements were not reused, the organizations would require an additional 16 months (0.16*100%). On the other hand organization would only require an additional 2.56 months (0.16*16%) maximum for Project B2 and 4.64 months (0.16*29%) maximum for Project B3 if requirements were reused.

Detailed calculations of DT and CTC are given for possible changes of 100%, 16% and 29% in Appendix-B using (1) and (2). 16% and 29% indicate those requirements, which were new and can be changed for Project B2 and Project B3, respectively. The complexity level of the products in Project B2 and B3 was taken as 3 again based on the number functions in the software. By using the results of calculations, the time spent for RE activities is summarized in Table IX.

However, calculated results are not similar to the actual results of Case Study-2 in Table VII and Table VIII. The decreases in both Project B2 and Project B3 were 34% in real life. But Griffin's formulation predicts at least 44% and 37% decreases in Project B3 and Project B4, respectively. This is explained and tackled in Section IV.

D. Case Study-3

Two software projects from Company A were studied as Project A3 and Project A4. In Project A4, some of the requirements from Project A3 were reused as well as some additional requirements included by the customer. Table X shows the number of requirements in Project A4. Project A4 had 342 requirements in total. 255 of those were reused from Project A3. For the requirements engineering phase of both projects, the data related to the duration are given in Table XI.

As shown in Table X, for Project A4;

- 75% of total requirements (255 requirements) were almost fixed and change probability of those was very low.
- 25% of total requirements (87 new requirements) could still be changed during the product cycle time.

By reusing the requirements, changes in product could be decreased in the range of 0% to 25% for Project A4.

TABLE IV. CASE STUDY-1: ESTIMATED TIME SPENT FOR RE WORKS

	RE works for 100% change	RE works for 43% change	% of Decrease in RE works
CTC-DT	18 months	14.01 months	$\geq$ 22%

TABLE V. REQUIREMENTS USED IN PROJECT B2

	Total # of Req. in Project B2	Req. of Project B1 Reused in Project B2
Total	376	314 (84% of Project B2)

TABLE VI. REQUIREMENTS USED IN PROJECT B3

	Total # of Req. in Project B3	Req. of Project B1 Reused in Project B3
Total	323	230 (71% of Project B3)

TABLE VII. DURATION EXPENDED IN PROJECT B1 AND PROJECT B2

	Project B1	Project B2	Possible Impact of Reuse
Total Duration (months)	7,5	5	34 % decrease in duration

TABLE VIII. DURATION EXPENDED IN PROJECT B1 AND PROJECT B3

	Project B1	Project B3	Possible Impact of Reuse
Total Duration (months)	7,5	5	34 % decrease in duration

 ISBN: 978-1-61208-230-1

TABLE IX. CASE STUDY-2: ESTIMATED TIME SPENT FOR RE WORKS

	RE works for 100% change	RE works for 16% and 29% change	% of Decrease in RE works
CTC-DT	13.5 months	7.62 months 8.53 months	$\geq$ 44% (16% change) $\geq$ 37% (29% change)

Using (2), if requirements were not reused, the organizations would require an additional 16 months (0.16*100%), while the organization would only require an additional 4 months (0.16*25%) maximum for Project A4 if requirements were reused.

Detailed calculations of DT and CTC are given for 100% and 25% changes in Appendix-C using (1) and (2). The durations for 100% and 25% changes is summarized in Table XII. The complexity level of the product was defined as 5.

Again, these results are not similar to the actual results of Case Study-3 in Table XI. The decrease in the Project A4 was 25% in real life. But Griffin's formulation predicts at least 32% decrease in Project A4.

The results of all case studies are tabulated in Table XIII.

IV. DISCUSSION

According to the empirical results of Case Study-1, Griffin's formulation for product development time is validated for system projects. But, empirical results of Case Study-2 and Case Study-3 show that it is necessary to modify Griffin's formulation for software projects. Software requirements may change more easily when compared to hardware requirements. The nature of software allows the customer to feel more comfortable while requesting changes. Thus, changes in software projects were more than expected. There are some decreases in project durations for Case Study-2 and Case Study-3, but these are less than what would be expected according to Griffin's formulation.

Changes in product features, i.e., in requirements are denoted by NN (newness/uncertainty) variable in (1) and (2). NN variable in [19] should be re-evaluated for software projects as this parameter should have a more significant effect on the software product development time according to the case studies performed in the scope of this study. By referring to these case studies, if the effect of NN variable is multiplied by at least 2.1 but not more than 3.4 for the cases where the product is not totally new, the results of Griffin's formulation show similarities with the real life results. Using a multiplier out of this range does not produce the realized results for the case studies. When the multiplication coefficient (referred as δ) is chosen to be close to 3.4, the decrease in the duration is going to be smaller. For values larger than 3.4, the results are similar to the case where requirements are not reused at all. So, δ is selected as 2.1 to see the best effect of requirements reuse. The projects for each case study selected from Company A and Company B are in the same domain and share the common requirements. Therefore, evaluations regarding the results of the case studies do not cover totally new products.

TABLE X. REQUIREMENTS USED IN PROJECT A4

	Total # of Req. in Project A4	Req. of Project A3 Reused in Project A4
Total	342	255 (75% of Project A4)

TABLE XI. DURATION EXPENDED IN PROJECT A3 AND PROJECT A4

	Project A3	Project A4	Possible Impact of Reuse
Total Duration (months)	6	4.5	25 % decrease in duration

TABLE XII. CASE STUDY-3: TIME SPENT FOR RE WORKS

	RE works for 100% change	RE works for 25% change	% of Decrease in RE works
CTC-DT	16.5 months	11.25 months	$\geq$ 32%

TABLE XIII. EXPECTED AND ACTUAL CHANGES IN DURATION OF RE ACTIVITIES FOR PROJECTS A2, B2, B3, A4 USIG GRIFFIN'S FORMULATION

Project	Max. Expected % of Change in Req.	Expected % of Duration Decrease in RE Works	Actual % of Duration Decrease in RE Works
A2	43%	$\geq$ 22%	37%
B2	16%	$\geq$ 44%	34%
B3	29%	$\geq$ 37%	34%
A4	25%	$\geq$ 32%	25%

The modified versions of (1) and (2) are proposed as below. The duration estimations include the engineering efforts during the requirements engineering phases. Other departments such as marketing, finance etc. are not included in the scope of the case studies. Therefore, this modification assumes that a cross functional team is not used in the organizations.

$$DT = \alpha + \beta_{1DT} * PC + \beta_{2DT} * \delta * NN + \beta_{3DT} * (PC * FP) + \epsilon_{DT} \quad (3)$$

$$CTC = \alpha + \beta_{1CTC} * PC + \beta_{2CTC} * \delta * NN + \beta_{3CTC} * (PC * FP) + \epsilon_{CTC} \quad (4)$$

where $2.1 \leq \delta \leq 3.4$.

Calculations for Case Study-2 of Company B are repeated in Appendix-D using (3) and (4). Equations (1) and (2) are used without any modification for the case when the product is totally new (NN=100%). Even if the maximum change (16%) occurred in the requirements for Project B2, the calculation indicates that there would be at least a 34% decrease in the duration (from 13.5 months to 8.85 months) of the RE activities. Real-life duration reduction, which was 34% as given in Table VII is in agreement with this result.

Similarly, if the maximum change (29%) occurred in the requirements for Project B3, the calculation indicates that there would be at least a 20% decrease in the duration (from 13.5 months to 10.76 months) of the RE activities. Consequently, real-life reduction of duration, which was 34% as given in Table VIII is in agreement with this result.

Similar calculations for Case Study-3 of Company A are performed in Appendix-E using (3) and (4). Again (1) and (2) are used for NN=100%. Even if the maximum change (25%) occurred in the requirements for Project A4, the calculation indicates that there would be at least a 20% decrease in the duration (from 16.5 months to 13.17 months) of the RE activities. The actual reduction in the duration, which was 25% as given in Table XI is in agreement with this result.

Summarized results of Case Study-2 and Case Study-3 using proposed formulations are given in Table XIV.

V. Conclusion and Future Work

The focus of this study is the reuse of requirements for different products in similar domains. The effects of requirement reuse for two different product types, one consisting of hardware and software, and the other purely software, have been investigated. For this investigation three case studies have been performed and their results have been compared with a theoretical study ([19]). It is very likely that different projects in the same domain have many common requirements and if these requirements were maintained and shared in a common database that employees could access, systems engineers would choose to use these requirements in different projects. In the context of such an opportunity, the product would be developed within a common understanding of the requirements. Besides, availability of applicable product development time estimations has an importance on the project management tasks. By using an applicable estimation model for industrial products, it should be possible to make project budget and resource allocation with minimum error. It might be difficult to work with minimum error at the beginning of the project with less information about the development details. From the case studies it is concluded that the proposed method [19] can be applied to system projects with hardware and software. However, it does not yield the same results with real-life software projects. This is because software requirements may change more easily when compared to hardware requirements. So, Griffin's formulation is revised for the software products and proposed formulation has more effect for the change probability of the product.

A. Future Work

This study showed the effects of requirements reuse on project duration based on empirical results of the case studies in which the data are collected from industry. This study covers only the requirements analysis phase. In terms of the future work, the duration can be further reduced by investigating reuse in the other phases of the project life cycle.

Table XIV. Expected and Actual Changes in Duration of RE Activities for Projects B2, B3, A4 Using the Proposed Formulation

Project	Max. Expected % of Change in Req.	Expected % of Duration Decrease in RE Works	Actual % of Duration Decrease in RE Works
B2	16%	$\geq$ 34%	34%
B3	29%	$\geq$ 20%	34%
A4	25%	$\geq$ 20%	25%

To enhance the validity of the δ value in proposed method, additional case studies can be performed. The effect of reuse can also be studied for the organizations in which cross functional teams are used.

Appendix

A. Calculations for Case Study-1

$$CTC_{100} = 10.4+3.7*6+0.16*100\%+0.1*6 = 49.2 \text{ months}$$

$$CTC_{43} = 10.4+3.7*6+0.16*43\%+0.1*6 = 40.08 \text{ months}$$

$$DT_{100} = 8.4+4.2*6+0.09*100\%-1.9*6 = 31.2 \text{ months}$$

$$DT_{43} = 8.4+4.2*6+0.09*43\%-1.9*6 = 26.07 \text{ months}$$

The time spent for requirements engineering is:

$$CTC_{100} - DT_{100} = 49.2 - 31.2 = 18 \text{ months}$$

$$CTC_{43} - DT_{43} = 40.08 - 26.07 = 14.01 \text{ months}$$

B. Calculations for Case Study-2

$$CTC_{100} = 10.4+3.7*3+0.16*100\%+0.1*3 = 37.8 \text{ months}$$

$$CTC_{16} = 10.4+3.7*3+0.16*16\%+0.1*3 = 24.36 \text{ months}$$

$$CTC_{29} = 10.4+3.7*3+0.16*29\%+0.1*3 = 26.44 \text{ months}$$

$$DT_{100} = 8.4+4.2*3+0.09*100\%-1.9*3 = 24.3 \text{ months}$$

$$DT_{16} = 8.4+4.2*3+0.09*16\%-1.9*3 = 16.74 \text{ months}$$

$$DT_{29} = 8.4+4.2*3+0.09*29\%-1.9*3 = 17.91 \text{ months}$$

The time spent for requirements engineering is:

$$CTC_{100} - DT_{100} = 37.8 - 24.3 = 13.5 \text{ months}$$

$$CTC_{16} - DT_{16} = 24.36 - 16.74 = 7.62 \text{ months}$$

$$CTC_{29} - DT_{29} = 26.44 - 17.91 = 8.53 \text{ months}$$

C. Calculations for Case Study-3

$$CTC_{100} = 10.4+3.7*5+0.16*100\%+0.1*5 = 45.4 \text{ months}$$

$$CTC_{25} = 10.4+3.7*5+0.16*25\%+0.1*5 = 33.4 \text{ months}$$

$$DT_{100} = 8.4+4.2*5+0.09*100\%-1.9*5 = 28.9 \text{ months}$$

$$DT_{25} = 8.4+4.2*5+0.09*25\%-1.9*5 = 22.15 \text{ months}$$

The time spent for requirements engineering is:

$$CTC_{100} - DT_{100} = 45.4 - 28.9 = 16.5 \text{ months}$$

$$CTC_{25} - DT_{25} = 33.4 - 22.15 = 11.25 \text{ months}$$

D. Calculations for Case Study-2 according to the proposed formulation (δ = 2.1)

$$CTC_{100} = 10.4+3.7*3+0.16*100\%+0.1*3 = 37.8 \text{ months}$$

 ISBN: 978-1-61208-230-1

$$CTC_{16} = 10.4{+}3.7{*}3{+}\mathbf{2.1}{*}0.16{*}16\%{+}0.1{*}3 = 27.18 \text{ months}$$

$$CTC_{29} = 10.4{+}3.7{*}3{+}\mathbf{2.1}{*}0.16{*}29\%{+}0.1{*}3 = 31.54 \text{ months}$$

$$DT_{100} = 8.4 + 4.2 * 3 + 0.09 * 100\% - 1.9 * 3 = 24.3 \text{ months}$$

$$DT_{16} = 8.4{+}4.2{*}3{+}\mathbf{2.1}{*}0.09{*}16\%{-}1.9{*}3 = 18.33 \text{ months}$$

$$DT_{29} = 8.4{+}4.2{*}3{+}\mathbf{2.1}{*}0.09{*}29\%{-}1.9{*}3 = 20.78 \text{ months}$$

Time spent for requirements engineering is calculated as;

$$CTC_{100} - DT_{100} = 37.8 - 24.3 = 13.5 \text{ months}$$

$$CTC_{16} - DT_{16} = 27.18 - 18.33 = 8.85 \text{ months}$$

$$CTC_{29} - DT_{29} = 31.54 - 20.78 = 10.76 \text{ months}$$

E. Calculations for Case Study-3 according to the proposed formulation (δ = 2.1)

$$CTC_{100} = 10.4{+}3.7{*}5{+}0.16{*}100\%{+}0.1{*}5 = 45.4 \text{ months}$$

$$CTC_{25} = 10.4{+}3.7{*}5{+}\mathbf{2.1}{*}0.16{*}25\%{+}0.1{*}5 = 37.8 \text{ months}$$

$$DT_{100} = 8.4 + 4.2 * 5 + 0.09 * 100\% - 1.9 * 5 = 28.9 \text{ months}$$

$$DT_{25} = 8.4{+}4.2{*}5{+}\mathbf{2.1}{*}0.09{*}25\%{-}1.9{*}5 = 24.63 \text{ months}$$

Time spent for requirements engineering is calculated as;

$$CTC_{100} - DT_{100} = 45.4 - 28.9 = 16.5 \text{ months}$$

$$CTC_{25} - DT_{25} = 37.8 - 24.63 = 13.17 \text{ months}$$

ACKNOWLEDGMENTS

The authors would like to thank the companies who have contributed the necessary data for the case studies to complete this work.

REFERENCES

[1] Matthew R. Kennedy and David A. Umphress, "An Agile Systems Engineering Process - The Missing Link", *CrossTalk, The Journal of Defense Software Engineering"*, May-June 2011.

[2] Richard Turner, "Toward Agile Systems Engineering Processes", *Cross Talk, The Journal of Defense Software Engineering*, April 2007.

[3] Shaojie Guo, Weiqin Tong, Juan Zhang, and Zongheng Li, "An Application of Ontology to Test Case Reuse", *International Conference on Mechatronic Science, Electric Engineering and Computer*, pp. 775-778, 2011.

[4] Cagatay Catal and Banu Diri, "A Conceptual Framework to Integrate Fault Prediction Sub-process for Software Product Lines", *2nd IFIP/IEEE International Symposium on Theoretical Aspects of Software Engineering*, pp. 99-106, 2008.

[5] Timo Kkl, "Standards Initiatives for Software Product Line Engineering and Management within the International Organization for Standardization", *Proceedings of the 43rd Hawaii International Conference on System Sciences*, pp. 1-10, 2010.

[6] Sholom Cohen, "Guidelines for Developing a Product Line Concept of Operations", Technical Report. CMU/SEI-99-TR-008, August 1999.

[7] John K. Bergey, Gary Chastek, Sholom Cohen, Patrick Donohoe, Lawrence G. Jones, and Linda Northrop, "Software Product Lines: Report of the 2010 U.S. Army Software Product Line Workshop", Technical Report, CMU/SEI-2010-TR-014, June 2010.

[8] Parastoo Mohagheghi and Reidar Conradi, "An Empirical Investigation of Software Reuse Benefits in a Large Telecom Product", *ACM Transactions on Software Engineering and Methodology*, Vol. 17, Issue 3, Article No 13, June 2008.

[9] Frank Dordowsky, Richard Bridges, and Holger Tschpe, "Implementing a Software Product Line for a complex Avionics System", *15th International Software Product Line Conference*, pp. 241-250, 2011.

[10] Kentaro Yoshimura, Jun Shimabukuro, Takatoshi Ohara, Chikashi Okamoto, Yoshitaka Atarashi, Shinobu Koizumi, Shigeru Watanabe, and Kazumi Funakoshi, "Key Activities for Introducing Software Product Lines into Multiple Divisions: Experience at Hitachi", *15th International Software Product Line Conference*, pp. 261-266, 2011.

[11] Andy J. Nolan, Silvia Abraho, Paul C. Clements, and Andy Pickard, "Requirements Uncertainty in a Software Product Line", *15th International Software Product Line Conference*, pp. 223-231, 2011.

[12] Eric M. Olson, Orville C. Walker, Robert W. Ruekert, and Joseph M. Bonner, "Patterns of Cooperation During New Product Development Among Marketing, Operations and R&D: Implications for Project Performance", *Journal of Product Innovation Management*, Vol 18, Issue 4. pp. 258-271, July 2001.

[13] Wayne C. Lim, "Effects of Reuse on Quality, Productivity, and Economics", *IEEE Software*, Vol. 11, Issue 5. pp. 23-30, 1994

[14] Gerald Kotonya and Ian Sommerville, *Requirements Engineering Processes and Techniques*, John Wiley & Sons Press, 1998.

[15] Tony Gorschek and Alan M. Davis, "Requirements Engineering: In Search of the Dependent Variables", *Information and Software Technology*, Vol. 50, No. 1-2. pp. 67-75, January 2008.

[16] Oscar Lopez Villegas and Miguel Angel Laguna, "Requirements Reuse for Software Development", *5th IEEE International Symposium on Requirements Engineering*, 2001.

[17] Donald Firesmith, "Achieving Quality Requirements with Reused Software Components: Challenges to Successful Reuse", *2nd International Workshop on Models and Processes for the Evaluation of off-the-shelf Components*, 2005.

[18] Abbie Griffin, "Metrics for Measuring Product Development Cycle Time", *Journal of Product Innovation Management*, Vol. 10, No. 2, pp. 112-125, March 1993.

[19] Abbie Griffin, "The Effect of Project and Process Characteristics on Product Development Cycle Time", *Journal of Marketing Research*, Vol. 34, No. 1, pp. 24-35, February 1997.

[20] He-Biao Yang, Zhi-Hong Liu, and Zheng-Hua Ma, "An Algorithm for Evaluating Impact of Requirement Change", *Journal of Information and Computing Science*, Vol. 2, No. 1, 2007.

[21] Susan Ferreira, James Collofello, Dan Shunk, and Gerald Mackulak, "Understanding the Effects of Requirements Volatility in Software Engineering by Using Analytical Modeling and Software Process Simulation", *Journal of Systems and Software*, Vol. 82, Issue 10, pp. 1568-1577, October 2009.

[22] William W. Thomas, Alex Delis, and Victor R. Basili, "An Analysis of Errors in a Reuse-Oriented Development Environment", *Journal of Systems and Software*, pp. 211-224, 1997.

[23] Mark Johnson, "From Engineering to System Engineering to System of Systems Engineering", *IEEE SMC Third International Conference on System of Systems Engineering*, pp. 1-6, 2008.

[24] Per Runeson and Martin Host, "Guidelines for Conducting and Reporting Case Study Research in Software Engineering", *Journal of Empirical Software Engineering*, Vol. 14. pp. 131-164, 2009.