International Journal of Software Engineering
and Knowledge Engineering
Vol. 26, No. 1 (2016) 63–86

DOI: 10.1142/S0218194016500042

Accounting for Product Similarity in Software Project Duration Estimation

Semra Yilmaz Tastekin
Aselsan Inc., Mehmet Akif Ersoy Mahallesi
296 Cadde No. 16, Yenimahalle, Ankara, Turkey
syilmaz@aselsan.com.tr

Yusuf Murat Erten
Izmir Institute of Technology
Izmir, Turkey
muraterten@iyte.edu.tr

Semih Bilgen
Computer Engineering Department, Yeditepe University
Istanbul, Turkey
semih.bilgen@yeditepe.edu.tr

Received 7 August 2013
Revised 8 February 2015
Accepted 21 February 2015

We extend an existing model proposed for estimating project duration for industrial projects in general, to software intensive systems projects. We show, through nine different cases studies from different sectors, that product similarity, measured in terms of requirements reuse, can be incorporated into that model to improve its applicability in software intensive systems projects.

Keywords: Product development time; product similarity based on requirements reuse.

1. Introduction

Estimation of product development time is a critical activity at the outset of any software intensive system development project. Johnson and Kirchain [1] state that 70% to 90% of project costs are determined during these earlier stages. However, at this stage, little information on development details is available. Thus, it is not easy to determine the project development time correctly and inaccurate estimations can present risks in terms of project scheduling and resource allocation. Bashir and

Thomson [2] emphasize the importance of correct estimations, stating that average schedule overruns range from 41% to 258%.

If data on previous projects in the same domain is available this can be used for estimating development time. At the beginning of the project, requirements are defined and formally specified. Whenever possible, requirements from previous projects can be reused, a similarity analysis can be performed and this can be used as input for development time estimation.

In the literature, there are numerous studies investigating techniques to reduce development time and metrics to control it. Carter [3] discusses product portfolio optimization to reduce development time. Callahan and Moretton [4] and Filippini *et al.* [5] address reducing the development time in terms of project management. Johnson *et al.* [6] discuss the importance of market knowledge on new product development success. Lebcir and Choudrie [7] investigate the influence of product complexity on product development time. Griffin's model [8], derived essentially in the context of manufacturing industries, is a significant contribution to this field in that it applies reuse data quantitatively to development time estimation and obtains realistic results; albeit in a non-software specific environment.

This study investigates and extends the development time estimation model proposed by Griffin [8]. Her model has been developed based on measurements from 343 projects in different sectors, but to the best of the authors' knowledge, has not been applied on, nor adjusted specifically for, industrial software intensive systems projects. This is what the present study aims to do.

In the present study, we focus specifically on project duration and not more generally on cost or effort, because, while the literature on cost and effort prediction is rather abundant, prediction of project duration seems to be less studied, and yet, constitutes one of the significant factors in contract negotiations.

The present study aims to go beyond academic research to investigate the applicability of a software intensive systems development time estimation model in industrial organizations. Within this scope, the impact of requirements reuse on software development duration for different products in a similar domain is investigated. A requirements oriented similarity analysis is performed for different products in the same domain and the findings are used as an input to estimate the development time using Griffin's model. To assess the applicability of that model for industrial software development projects, nine cases from three different organizations have been studied. In each case study, based on system and software requirements and their re-use, similarity and newness of each product has been quantified. Duration data derived from the empirical case studies have been compared with the expected durations obtained using Griffin's model. According to the main functionalities of each product, their complexities have been determined. By taking into account the newness and complexity measures of each product, product development times have been estimated using Griffin's model. The results of those estimations have been compared with the actual durations of each project.

The case studies showed that estimated durations did not match the actual durations of software products whereas the duration estimations for system projects have been found to be compatible with realized durations. Therefore, an extension to Griffin's formulation is proposed for development time estimation specifically for software projects.

An earlier version of this study was presented at ICSEA 2012 [9]. In the present paper the work has been extended to include a detailed discussion, verification of product complexity measurement [10], an increased number of case studies and the development of a framework for the development time estimation process.

The rest of this article is organized as follows:

Section 2 briefly reviews the background of the problem of software product development time estimation and the product similarity concept. Section 3 poses the research problem and describes the research methodology. Characteristics of the study and validity of the case studies are discussed in this section. Section 4 presents the findings of the nine case studies. Section 5 discusses the results of case studies and formulates the proposed modification for software development time estimation. The software-specific case studies are re-analyzed with the proposed modification. Section 6 concludes the paper with an overview of the proposed process model for estimating product development time, a summary of the limitations of the study and suggestions for future work.

2. Background

We focus on the applicability of an existing product development time estimation model [8] to industrial software and system projects. This section reviews the concepts of product development time estimation, product similarity based on requirements reuse and product complexity.

2.1. *Product development time estimation*

It is generally accepted that it is difficult to formulate a generic model for development time estimation [11]. Griffin undertook a number of studies [8, 12–14] to determine the time spent on product development and the factors that effect this duration.

An earlier study by Griffin [12] proposes a formula for estimating the product development time with a limited data set. She classifies the factors which effect development time in four groups as:

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

Changes in the product and accordingly in the requirements will have a considerable effect on the development workload. Callahan and Moretton [4] observed that a factor that has a major effect on product development time is *Newness/uncertainty*.

Griffin hypothesizes [8] that the development cycle time increases with greater product complexity.

If organizations do not have formal development processes, the development time is longer compared to those with formal development processes. In an empirical study by Barczak *et al.* [15], based on the 2003 best practices survey of the Product Development & Management Association (PDMA) in the USA, about 150 organizations were analyzed and according to this study 15% of the firms did not have a formal development process.

The use of a cross functional team also affects development time. A study by Olson *et al.* [16] emphasizes that cooperation between specific functional departments associated with the new product being developed is important in increasing project performance.

Griffin [12] has defined Development Time (DT) and Concept To Customer (CTC) as two separate parameters. DT begins from the design and product development through to the introduction to the customer. CTC begins with concept development and continues to the specification definition until the introduction to the customer. Requirements engineering activities are covered within CTC. If DT is subtracted from the CTC this will give the time spent on requirements engineering activities such as business development, concept development and requirements definition.

In a later study based on a large dataset, Griffin [8] formulates development time as:

$$\begin{aligned}\mathrm{DT} = \alpha + \beta_{1\mathrm{DT}} \times \mathrm{PC} + \beta_{2\mathrm{DT}} \times \mathrm{NN} + \beta_{3\mathrm{DT}} \times (\mathrm{PC} \times \mathrm{FP}) \\ + \beta_{4\mathrm{DT}} \times (\mathrm{NN} \times \mathrm{XFT}) + {}_{\mathrm{DT}}\end{aligned} \tag{1}$$

$$\begin{aligned}\mathrm{CTC} = \alpha + \beta_{1\mathrm{CTC}} \times \mathrm{PC} + \beta_{2\mathrm{CTC}} \times \mathrm{NN} + \beta_{3\mathrm{CTC}} \times (\mathrm{PC} \times \mathrm{FP}) \\ + \beta_{4\mathrm{CTC}} \times (\mathrm{NN} \times \mathrm{XFT}) + {}_{\mathrm{CTC}}\end{aligned} \tag{2}$$

where α is the cycle time constant, PC and NN are product complexity and product *Newness/uncertainty*, respectively. FP and XFT show, respectively, whether formal processes or cross functional teams are used. is the error term. If a formal development process is not used, then FP = 0. The units of β_1 and β_3 are the months/function designed in the product. The units 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 the data collected from many companies are given in Table 1.

Since Griffin's work, cited more than 200 times and noted, for example by Dooley *et al.* [17], as one of the best on new product development time estimation, was based on measurements from 343 projects from 21 divisions of 11 companies in different sectors, her model has significantly contributed to establish the relationship between

Table 1. Coefficients used in the DT and CTC equations [8].

	α constant	β_1 PC	β_2 NN	β_3 PC $\times$ FP	β_4 NN $\times$ XFT
DT	8.4	4.2	0.09	−1.9	−0.09
CTC	10.4	3.7	0.16	0.1	−0.16

development time and product complexity, newness of product and use of a formal process. As mentioned by Bashir and Thomson [2], Griffin uses less subjective estimations in comparison to other studies and does not require a large amount of development detail which may not be available at the early phases of projects. Thus, the framework presented in the present study is based on the model proposed by Griffin [8].

2.2. *Product similarity based on requirements*

Reuse has traditionally been considered as a means for improving development productivity and quality [18], and it is widely accepted to lead to the introduction of faster, better and cheaper products into the market [19].

Engineers discover most of the software and hardware problems at the integration phase of projects. Isolation of the source of these problems at this stage can take time and this may affect the project duration. According to a study by Guo *et al.* [20], 50% of the total time and cost of a project is spent on testing. To minimize the number of faults detected during integration and test phases and avoid unnecessary delays in project delivery, reuse of components created during various phases of different projects plays an important role. Considering that 7–15% of total project resources are used for requirements engineering [21, 22], requirements-related phases of the development lifecycle should be realized as effectively as possible.

Beside such advantages of requirements reuse, there are some concerns about using existing requirements [19]. For example, existing requirements might not be completely developed, in which case it will not be possible to use them. Another concern is that if the existing requirements have not been updated, this would make it difficult to reuse them. Finally, if the requirements' quality is poor, their implementation will be difficult. Dieste *et al.* [22] point to the risk of getting requirements wrong when incremental development is used. In spite of these concerns, Chernak's empirical study [19] indicates that requirements reuse helps to reduce time to market as well as product cost. Moreover, according to Goldin and Matalon-Beck [23], requirements reuse reduces the development effort by 45%, development time by 33% and time-to-market by 60%.

Lee and Lee [24] propose a measurement model for product similarity to identify the products in the same family. Specifications or features of the product are defined and according to those specifications/features a product family relation is calculated between two product groups. A similar approach has been adopted in the present study. Requirements that define the products have been chosen as the main features of the product. Based on the contents of the requirements, a similarity analysis is conducted to ascertain the *Newness* of a current product compared to previous products. Product similarity analysis is based on the reused requirements of the products. Reuse rate is defined as the ratio of the number of existing requirements reused from the previously released requirements to the total number of requirements used to implement a given release [19]. For Griffin's model, the complement of Similarity is used for *Newness*. Both variables are formulated as below.

$$\begin{aligned}\text{Similarity} &= \text{Reuse Rate}\\ &= \frac{(\text{number of reused requirements from previous released requirements})}{(\text{total number of requirements})}\\ &\quad \times 100\% \end{aligned} \tag{3}$$

$$\text{Newness}(\%) = 100 - \text{Similarity} \tag{4}$$

In the present study, requirements are counted for running projects and completed (or close to completion) projects. For requirements similarity, requirements are identified according to their semantics instead of text-based similarity. Requirements engineers evaluated the content meaning of the existing requirements and determined if the selected requirements should be identified as reused or not. The compared projects are in the same domain, therefore requirements are defined by the engineering team responsible from both projects in the same domain. The term *same domain* means that the products, system or software in each case study are similar and share some common features. This implies that the requirements semantics are comparable among multiple projects. Thus, it is believed that the selected requirements can be equally comparable for the project in the scope of each case study.

2.3. *Product complexity*

Product complexity has effects on project determinants such as cost, duration and resource allocation [25–27].

Various authors from a wide-ranging spectrum of engineering viewpoints have considered product complexity according to their areas of research and adopted scope. Murmann [28] defines it as the number of parts in a product. Larson and Gobeli [29] define project complexity as the number of different disciplines or departments in a project. Meyer and Utterback [30] define complexity from the design aspect as the number of core technologies in a product. Novak and Eppinger [31] define product complexity as the number of product components and the level of coupling between these components plus the degree of product novelty. Similar concepts are also applied to determine software complexity. McCabe [32] introduced cyclomatic complexity, based on program structure. He developed a mathematical model to identify software modules to test and maintain software more easily. A similar definition based on software structure is introduced by Zuse [33]. He defines complexity as the difficulty to maintain, change and understand software. Other sources and different dimensions of product complexity are discussed by Kim and Wilemon [34] and Orfi *et al.* [27]. Such widely varying definitions of product complexity lead to difficulties in arriving at a universally accepted way of measuring it [27].

According to Griffin's research, complexity levels are between 1 and 11 for projects from different sectors. In the scope of the present study, Griffin's complexity definition has been preserved, after having verified it [10] by measuring the grey complexity of five products developed in a major defense contractor operating in the international market.

3. Research Methodology

This section poses the main research question and outlines the research method and data collection process.

3.1. *Research problem and method*

Project managers are aware of the impact of changes whether pre-planned or unexpected and therefore recognize the need to monitor the effort and time expended on the development of the project. However, without an accurate estimation of the factors that can affect projects, managers are unlikely to be able to predict the success or failure of the project. This observation leads to the following research problem:

> ***How can product similarity be reflected to development time estimation at the beginning of a software intensive system development project?***

The unit of analysis for our case study research is a software or system development project. The research method depicted in Fig. 1 has been employed. Initially an extensive literature survey has been conducted based on product development time and product similarities. Based on that survey, Griffin's model originally proposed for manufacturing industries has been chosen. Similarity analysis has been conducted between products to obtain the *Newness* value needed to estimate the development time. Since Griffin's model has not been applied to software products, it has been expected that a modification would be needed.

The number of reused requirements, requirements engineering duration and product complexity are the basic data collected in the case studies. Besides, other data are required to complete the case studies. For example, to derive the newness or similarity of the new product to previous products, it is necessary to know the total number of requirements. Moreover, to determine whether requirements are reused or not, it is also necessary to know the semantics of the requirements. For this purpose requirements are evaluated according to their context as well as content. All these issues lead to work with the engineers involved in the projects. Necessary data for the case studies have been gathered by joint efforts of project technical managers and one of the authors of this article.

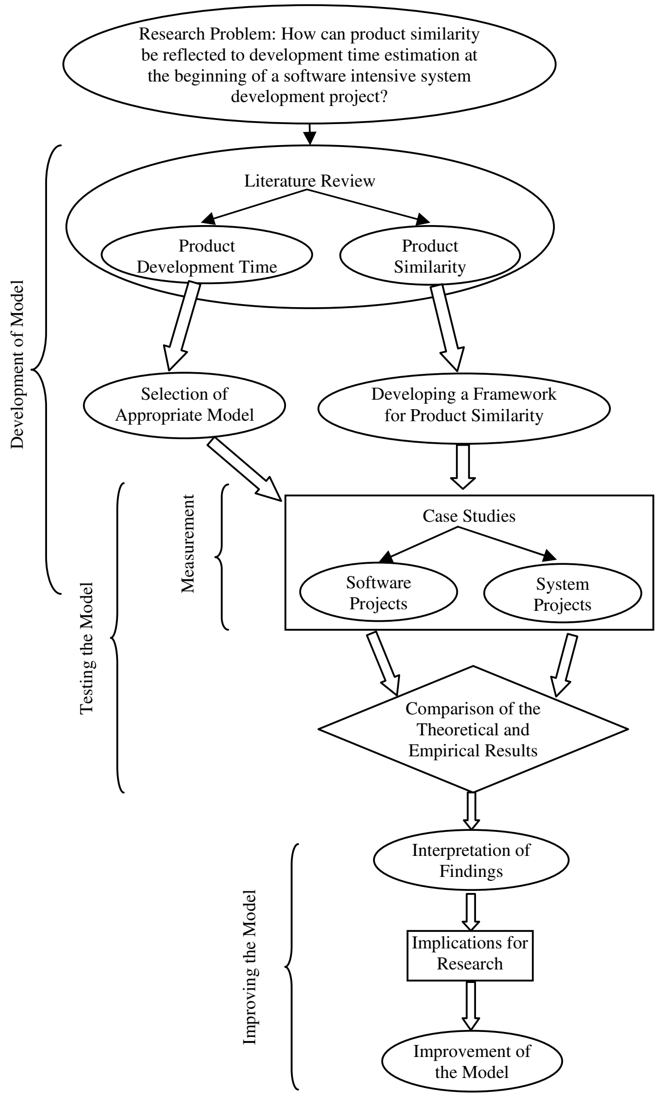

Fig. 1. Research method.

3.2. *Characteristics of the case studies*

Based on Runeson and Höst's classification [35], the characteristics of the present study are *explanatory* and *improving*. It is *explanatory* because it explains the necessity for project management to define product development time at the beginning of projects. It is also *improving* because the results show the shortcomings of a proposed model. It improves the applicability of Griffin's development time model to software projects.

Data collected in case studies can be *quantitative*, *qualitative* or a combination of both [35, 36]. The necessary data to undertake the reported case studies include the total number of requirements, reused requirements and duration of the requirements definition phase. This study also includes similarity analysis. Since this similarity framework requires a count of requirements, a *quantitative* approach is employed.

3.3. *Validity of the case studies*

The following criteria established by Cavaye[36] and Gibbert *et al.* [37] are assessed for the validity and reliability of the case studies:

- Internal validity: The metrics considered in this study are project complexity, newness of product and whether a formal process is used or not. The lack of interdependency among these is widely accepted as visible in the works of Callahan and Moretton [4], Olson *et al.* [16], Herstatt *et al.* [38], Michalek *et al.* [39], Schimmoeller [40] and Bonner *et al.* [41]. As we are investigating how product development time is affected with similarity of the new product to the ones developed in the past by the same company and having parallel degrees of complexity we believe that it is fairly safe to assume that the dominant factor causing improvement or degradation is the newness and complexity of the product.
- Construct validity: Data have been utilized from the interviews carried out by one of the authors. These data consist of the number of requirements and duration of the requirement definition phase. Also, the data for complexity level of the products have been derived from the evaluations of the author and project technical managers with a perspective gained by an objective assessment.
- External validity: To generalize our results for the estimation of product development time, it is appropriate to use the actual data from industry. In order to enhance generalizability of our results, multiple case studies from three different organizations have been undertaken. Since technical managers are responsible to all stakeholders for technical activities, they have been selected as the right sources for the necessary information.
- Reliability: Each case study has been conducted on the basis of an in-depth interview. Except for the complexity level, number of requirements and the duration of requirements engineering activities have been drawn from the organization's archives, and accordingly, it is not researcher-dependent, nor open to different interpretations. Because that information is also included in the contractual documentation such as requirements specifications documents and project schedule, it is sufficiently reliable. For the complexity level definition, an assessment has been performed for all of the products.

3.4. *Validity threats*

In this study, the following parameters have been used to estimate the product development time: product complexity, newness of product and whether a formal process is used during the life cycle of project. A threat for the validity of the presented findings is that different project managers may use different definitions for these terms. In particular, identification of the main functions of a given product may have subjective as well as objective aspects, hence possibly leading to inconsistent quantification of complexity. While this issue has been addressed via consistent definition of such terms throughout the cases studied, assuring an objective and consistent measurement of product complexity deserves future study.

Another threat arises from the fact that organizations may adopt formal process definitions, which may not be strictly applied in some projects. In this aspect, the actual use for a specific project, rather than the existence of formal process definitions, must be considered in development time estimation.

3.5. *Data collection*

To estimate product development time, data have been gathered from three different companies via the project technical managers and authors of this paper. Unfortunately, there were difficulties in gathering data from different companies. Firstly, organizations generally do not keep the project related data in a systematic way. To overcome this difficulty, interviews have been held with the project technical managers and the related data have been collected using relevant documents and the organization's database. Interviews were fully-structured, as defined by Runeson and Höst [35], all the questions have been prepared before the interviews and the authors determined the flow of the discussions. These questions are given below.

Question 1: Are there any similar products which can be in the same domain or are derivative products in the organizations?
Investigation approach: Discussions with different project technical managers from different companies have been performed and the details of the projects have been evaluated. During the evaluation, some project characteristics have been discussed. For instance, two projects should be in the same domain, that is, the projects should have some similar functionalities or features so that similarity between the projects could be determined. Another important issue has been to access personnel who could evaluate the semantics of the requirements and could define whether the requirements were reused or not.

Question 2: Are there recorded data for requirements of projects in the same domain?
Investigation approach: The System/Software Requirements Specification Documents for each project have been used to obtain the necessary data.

Question 3: Are there recorded data for reused requirements?
Investigation approach: If the metrics were recorded systematically, data have been retrieved from the organizational database. If they were not kept in such a formal repository, reused requirements have been derived from the System/Software Requirements Documents by the technical personnel involved in the projects. In this case, technical personnel have evaluated each requirement in the System/Software

Requirements Documents with the context they are defined. In the scope of each case study, two projects have been included, thus two System/Software Requirements Documents have been compared for each requirement. The functionality involved for each requirement has been interpreted in detail and if a requirement had the same meaning and if it required designing the same feature/object/activity with a requirement in the System/Software Requirements Documents of the other project, this requirement has been considered as reused.

Question 4: Is there duration data for requirement definition phases?
Investigation approach: The enterprise resource planning systems of the companies have been used to extract this data.

Question 5: What is the complexity level of product to be studied?
Investigation approach: The main functions of the products have been determined to obtain the complexity level of the product with the help of technical managers of the projects.

Data have been incrementally collected. Number of requirements and the duration of requirements engineering activities have been analyzed first. In the scope of that activity, product complexities have been quantified considering the number of each product's main functions. It has been shown that counting the main functions is sufficiently representative in terms of software characteristics, technology, organization and environment determinants and indicators of those determinants given in [10].

Even the metrics were recorded systematically, the organizations would not release the data for external use. Therefore, descriptive data about the companies who provided project related data has been limited in this study. The company data and related case study summary are given in Table 2. All three companies were located in the same country and were private development organizations.

Table 2. Summary of the case studies.

Company	Sector of products developed	Case study	Projects	Product type
A	Military, Civilian	1	A1[a], A2	Hardware + Software
		2	A1[a], A3	Hardware + Software
		3	A4, A5	Software
		4	A6, A7	Software
B	Military, Civilian	5	B1, B2	Hardware + Software
		6	B3, B4	Hardware + Software
		7	B5, B6	Software
C	Civilian	8	C1[b], C2	Software
		9	C1[b], C3	Software

[a] A1 is the baseline project for A2 in the Case Study 1 and A3 in Case Study 2. There were two projects derived from A1, these are A2 and A3, that is A1 in Case Study 1 and 2 is the same project.
[b] C1 is the baseline project for C2 in the Case Study 8 and C3 in Case Study 9.

4. Case Study Findings

Four of the nine case studies involved the reuse of requirements for a system with hardware and software. The remaining five cases involved the reuse of requirements for purely software products.

The data from the case studies have been used for the following purposes:

- Analyzing the similarity of products to previous products in same domain by studying the requirements of both products.
- Collecting the realized duration for requirement definition phases.
- The comparison of duration for two projects in the same domain for each case study.
- Comparison of the realized duration with the results of the method proposed earlier for product development time [8].
- Separately analyzing the software products and system products.
- Formulating the modification to be proposed for product development time estimation in software-intensive systems.

The outcomes of the case studies can be summarized as follows:

- Newness of products can be derived from the similarity of product to previously developed products. Similarity can be calculated from the number of reused requirements.
- Griffin's product development time estimation method is appropriate for system products which involve hardware and software components.
- Griffin's product development time estimation method is not appropriate for software products.

A proposal has been formulated and validated for product development time estimation for software products.

4.1. *Case study 1 (system project)*

Company A is a market leader for military products and systems. The division of the Company A that is the focus of this case study uses its own design and development processes and in 2013 they were certified at the CMMI Level 3 of maturity. In this division there were two different projects, Project A1 and A2 in the same product family and including hardware and software components. Project A1 had been completed in 2012. By the middle of 2012, the requirements of Project A2 had been approved by the customer and the pre-design phase had been completed.

Table 3 shows the number of the requirements in Project A2. The realized duration for requirements definition activities for both projects is given in Table 4. Tables 3 and 4 also include data from other case studies.

Table 3 shows that 57% of the requirements (104 requirements out of 183) of Project A2 were reused requirements. This implies that the change probability of 104

Table 3. Number of requirements used in projects.

	Project	Total number of requirements	Number of reused requirements	Newness (%)
Case Study 1	A2	183	104	43
Case Study 2	A3	233	170	27
Case Study 3	A5	342	255	25
Case Study 4	A7	167	106	37
Case Study 5	B2	212	91	57
Case Study 6	B4	394	146	63
Case Study 7	B6	134	82	39
Case Study 8	C2	376	314	16
Case Study 9	C3	323	230	29

Table 4. Durations expended in RE works for projects.

	Projects RE durations (months)		Possible impact of reuse
Case Study 1	$A1 = 8$	$A2 = 5$	37% decrease in duration
Case Study 2	$A1 = 8$	$A3 = 4.5$	44% decrease in duration
Case Study 3	$A4 = 6$	$A5 = 4.5$	25% decrease in duration
Case Study 4	$A6 = 4$	$A7 = 3$	25% decrease in duration
Case Study 5	$B1 = 5$	$B2 = 3$	40% decrease in duration
Case Study 6	$B3 = 5$	$B4 = 4$	20% decrease in duration
Case Study 7	$B5 = 5$	$B6 = 4$	20% decrease in duration
Case Study 8	$C1 = 7,5$	$C2 = 5$	34% decrease in duration
Case Study 9	$C1 = 7,5$	$C3 = 5$	34% decrease in duration

reused requirements was very low in this project, because they had previously been tested and approved by the same or a similar customer. Hence

- 57% of total requirements (104 requirements) for Project A2 were almost fixed. This ratio denotes the similarity of A2 to A1.
- 43% of total requirements (79 new requirements) could still be changed in Project A2. This ratio denotes the newness of Project A2.

By reusing the requirements, *Newness* of a product (NN) is minimized. While normally NN varies between 0% and 100%, by requirements reuse, this variation is decreased in the range of 0% to 43% for Project A2. Using Griffin's CTC formulation in Eq. (2), for all possible changes in the requirements, if the requirements had not been reused, the organization would have required an additional 16 months ($\beta_{2\mathrm{CTC}} * \mathrm{NN} = 0.16 * 100\%$). On the other hand, the organization would only require an additional maximum of 6.88 months (0.16*43%) when all the common requirements have been reused. So, the change effect is reduced by 9.12 months for Case Study 1.

As described above, requirements engineering activities are contained within the duration of the CTC. To estimate the time spent for requirements engineering activities, the calculations for CTC and DT given below for 100% and 43% cases have been performed using Eq. (1) and Eq. (2). 100% indicates that product requirements/features were totally new; 43% indicates the amount of new requirements, and the latter is taken as the change probability of the requirements. As shown in the calculations below, the complexity level of the product developed within the scope of Project A2 has been taken as 6, based on the number of main functions and the technology the product possessed.

$$\begin{aligned}
\mathrm{CTC}_{100} &= 10.4 + 3.7 * 6 + 0.16 * 100\% + 0.1 * 6 = 49.2 \text{ months} \\
\mathrm{CTC}_{43} &= 10.4 + 3.7 * 6 + 0.16 * 43\% + 0.1 * 6 = 40.08 \text{ months} \\
\mathrm{DT}_{100} &= 8.4 + 4.2 * 6 + 0.09 * 100\% - 1.9 * 6 = 31.2 \text{ months} \\
\mathrm{DT}_{43} &= 8.4 + 4.2 * 6 + 0.09 * 43\% - 1.9 * 6 = 26.07 \text{ months}
\end{aligned}$$

The time spent on requirements engineering for NN values of 100% and 43% would be;

$$\begin{aligned}
\mathrm{CTC}_{100} - \mathrm{DT}_{100} &= 49.2 - 31.2 = 18 \text{ months} \\
\mathrm{CTC}_{43} - \mathrm{DT}_{43} &= 40.08 - 26.07 = 14.01 \text{ months}
\end{aligned}$$

The calculated time spent for requirements engineering (RE) activities is summarized in Table 5. These durations are longer than the actual durations given in Table 4, because the estimated durations include other systems engineering activities at the beginning of the project, such as business and concept development.

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

When this result is compared with the actual findings of Case Study 1 in Table 4, the decrease in Project A2 shows agreement with these calculations. Griffin's formulation predicts at least a 22% reduction in duration, likewise a reduction of 37% has been obtained. Thus, this case study which involves both hardware and software components conforms to the formulation proposed by Griffin for the estimation of project duration.

Table 5. Estimated durations spent for RE works.

		RE works for 100% change	RE works	% of decrease in RE works
CTC-DT	Case Study 1	18 months	14.01 months for 43% change	$\geq 22\%$
	Case Study 2	18 months	12.89 months for 27% change	$\geq 28\%$
	Case Study 3	16.5 months	11.25 months for 25% change	$\geq 32\%$
	Case Study 4	13.5 months	9.09 months for 37% change	$\geq 33\%$
	Case Study 5	18 months	14.99 months for 57% change	$\geq 17\%$
	Case Study 6	18 months	15.41 months for 63% change	$\geq 14\%$
	Case Study 7	19.5 months	15.23 months for 39% change	$\geq 22\%$
	Case Study 8	13.5 months	7.62 months for 16% change	$\geq 44\%$
	Case Study 9	13.5 months	8.53 months for 29% change	$\geq 37\%$

4.2. *Case study 2 (system project)*

This case includes two different projects in the same division of Company A, Project A1 and A3. Project A1 is the same project defined in Case Study 1. By the middle of 2012, the requirements of Project A3 had been defined and approved in the organization.

Table 3 shows the number of requirements in Project A3. Table 4 gives the requirements definition duration data for both projects. As shown in Table 3, for Project A3;

- 73% of total requirements (170 requirements) for Project A3 were almost fixed. This ratio denotes the similarity of A3 to A1.
- 27% of total requirements (63 new requirements) could still be changed in Project A3. This ratio denotes the newness of Project A3.

The complexity level of the product developed in the scope of Project A3 has been assessed by organization staff as 6.

Similar calculations are performed as in Case Study 1, but details of the calculations have not been repeated for the rest of the case studies. Estimated RE duration is summarized in Table 5. When this result is compared with the actual findings, the decrease in Project A3 shows agreement with these calculations. Griffin's formulation predicts at least a 28% reduction in duration, likewise a reduction of 44% has been obtained. Thus, this case study which involves both hardware and software components conforms to the formulation proposed by Griffin for the estimation of project duration.

4.3. *Case study 3 (software project)*

Another division of Company A had been using its own design and development processes and in 2011 they had been certified at the CMMI Level 3 of maturity. Two software projects of this division have been analyzed for this case: Projects A4 and A5 which were in the same product family. Project A4 had been completed in 2011 and included design and development of a military product. For Project A5, the system requirements had been defined and approved by the customer in 2012.

Table 3 shows the number of requirements in Project A5. Table 4 gives the requirements definition duration data for both projects. As shown in Table 3, for Project A5;

- 75% of total requirements (255 requirements out of 342) were almost fixed and their change probability was very low. This ratio is the similarity of A5 to A4.
- 25% of total requirements (87 requirements out of 342) could still be changed during the product cycle time. This ratio is the Newness of Project A5.

The complexity level of the product developed in the scope of Project A5 has been assessed by organization staff as 5.

Estimated RE duration is summarized in Table 5. These calculated results are not in agreement with the actual findings of Case Study 3 in Table 4. The decrease in Project A5 was actually 25% but Griffin's formulation predicts at least a 32% decrease in Project A5. This observation, together with others in similar purely software development projects, as described in the remaining case studies, motivates our modification proposal to be presented Sec. 5, below.

4.4. *Case study 4 (software project)*

Two software projects have been analyzed for this case: Projects A6 and A7 which were in the same product family. Project A6 had been completed in 2012 and included design and development of a military product. For Project A7, the system requirements had been defined and approved by the customer in 2012.

Table 3 shows the number of requirements in Project A7. Table 4 gives the requirements definition duration data for both projects. The complexity level of the product developed in the scope of Project A5 has been assessed by organization staff as 3.

Estimated RE duration is summarized in Table 5. These calculated results are not in agreement with the actual findings of Case Study 4 in Table 4. The decrease in Project A7 was actually 25% but Griffin's formulation predicts at least a 33% decrease in Project A7.

4.5. *Case study 5 (system project)*

Company B has a design and development process which is in accordance with IEEE/EIA 12207. This case study analyzes Projects B1 and B2 which were related to the same product family of military communication equipment, and which included hardware and software components. Project B1 had been completed in 2011. Project B2 was based on the product developed in the scope of Project B1. New requirements were added according to the product user and chosen platform. For Project B2 the system requirements had been defined and approved by the customer in 2011. This project was in the development phase at the time of the study and the test phase would start at the end of 2013.

The number of the requirements in Project B2 is given in Table 3. The realized duration for the requirements definition activities for both projects is given in Table 4.

The complexity level of the product developed in the scope of Project B2 has been taken as 6 based on the number of main functions the product possessed.

Estimated duration for RE activities is summarized in Table 5. When this result is compared with the actual findings of Case Study 5 presented in Table 4, the decrease in Project B2 shows agreement with these calculations.

4.6. *Case study 6 (system project)*

This case study also consists of two system projects from Company B: Projects B3 and B4. Project B3 had been completed in 2012. Project B4 was in the test phase and was expected to be completed in 2013.

The number of the requirements in Project B4 is presented in Table 3. The realized duration for the requirements definition activities for both projects is given in Table 4.

The complexity level of the product developed in the scope of Project B4 has been taken as 6 based on the number of main functions.

As a result of calculations, the time spent on RE activities is summarized in Table 5. When this result is compared with the actual findings of the Case Study 6 in Table 4, the decrease in Project B4 shows agreement with these calculations.

4.7. *Case study 7 (software project)*

This case study involves two software projects from Company B; B5 and B6 which were in the same domain. In the scope of Project B5, a commercial software product had been developed. Project B6 includes the development of a similar product for military purposes. Project B5 had been completed in 2010. For Project B6, the software requirements had been defined and approved by the customer in 2012. Project B6 was expected to be completed in 2014.

Table 3 shows the number of requirements in Project B6. Again, for the requirements definition of both projects, the duration data are given in Table 4.

The product developed in the scope of the Project B6 had a complexity level of 7.

Using the results of the calculations, the duration for RE works is summarized in Table 5. These calculated results are not in agreement with the actual findings of Case Study 7 in Table 4. The decrease in Project B6 was 20% in the case study but Griffin's formulation predicts at least a 22% decrease in Project B6.

4.8. *Case study 8 (software project)*

Company C is a leading software company. Their software design and development activities are performed in accordance with ISO/IEC 15504 maturity model Level 2. Two software projects which were in the same product family from Company C have been analyzed for this case study. Both projects were carried out for government institutions. Project C1 began development in 2009. Project C2 started at the beginning of 2012 and delivery was planned for the end of 2013.

Table 3 shows the number of requirements of Project C2. For the requirements definition activities of this project, the duration data are given in Table 4.

Again, the complexity level of the product developed in Project C2 has been taken as 3 based on the number of functions in the software.

Table 5 presents a summary of the time spent on RE works according to the results of the calculations. However, the calculated result is not in agreement with the actual findings of Case Study 8 as shown in Table 4. The decrease in Project C2 was 34% in real life but Griffin's formulation predicts at least 44% decrease in Project C2.

4.9. *Case study 9 (software project)*

A second case study from Company C has been undertaken. In this case Project C3 used Project C1 which was defined in Case Study 8 as a baseline and new customer requirements have been added to Project C3. C1 and C3 were in the same product family. This software product in the scope of Project C3 was also developed to be used by a government institution. Project C3 had started at the beginning of 2012 and delivery was planned for the beginning of 2014.

Table 3 shows the number of requirements in Project C3. For the requirements definition activities of this project, the duration data are given in Table 4.

The complexity level of the product developed in Project C3 has been taken as 3 based on the number of functions in the software.

The result of these calculations for the time spent on RE works is summarized in Table 5. However, the calculated result is not in agreement with the actual findings of Case Study 9 as shown in Table 4. The decrease in Project C3 was 34% in real life but Griffin's formulation predicts at least a 37% decrease in Project C3.

A summary of all the case study results are presented in Table 6.

Table 6. Expected and actual changes in duration of RE activities for projects A2, A4, B2, B4, B6, C2, C3 using Griffin's formulation.

Project	Product type	Max. expected % of change in req.	Expected % of duration decrease in RE works	Actual % of duration decrease in RE works	Compatibility to Griffin's formulation
A2	Hardware + Software	43%	$\geq$ 22%	37%	Compliant
A3	Hardware + Software	27%	$\geq$ 28%	44%	Compliant
A5	Software	25%	$\geq$ 32%	25%	Not Compliant
A7	Software	37%	$\geq$ 33%	25%	Not Compliant
B2	Hardware + Software	57%	$\geq$ 17%	40%	Compliant
B4	Hardware + Software	63%	$\geq$ 14%	20%	Compliant
B6	Software	39%	$\geq$ 22%	20%	Not Compliant
C2	Software	16%	$\geq$ 44%	34%	Not Compliant
C3	Software	29%	$\geq$ 37%	34%	Not Compliant

5. Discussion

According to the findings of Case Studies 1, 2, 5 and 6, Griffin's formulation for product development time has been validated for system projects which include hardware and software components.

The findings of Case Studies 3, 4, 7, 8 and 9 disagree with the estimates based on Griffin's formulation. In this section, we propose a modification to Griffin's formulation for software projects and show that in its modified form, it can be used to accurately estimate the product development time.

Software requirements can change more easily and more often than hardware requirements. Software changes generally do not affect the hardware, a change

Table 7. Possible values of δ for software projects.

Project	Multiplication coefficient
A5	$1.7 \leq \delta \leq 4$
A7	$1.4 \leq \delta \leq 2.7$
B6	$2 \leq \delta \leq 2.5$
C2	$2.1 \leq \delta \leq 6.2$
C3	$1.2 \leq \delta \leq 3.4$

request can be met by amendments to software. However, hardware changes can have a greater impact on the project.

We propose that the NN variable in Eq. (1) and Eq. (2) must be re-evaluated for software projects. By referring to our case studies, if the effect of the NN variable is multiplied by at least 2.1 but not more than 2.5 (which is the range common to all of the cases considered) for the cases where the product is not totally new, the results of Griffin's formulation agree with real-life results. The multiplication coefficient used in this study is denoted as δ. The values for this coefficient for each software case study are given in Table 7.

If an estimate is to be performed for a new project then with the information at hand, the best value to be used for δ would be 2.1. Using a δ which is larger than the maximum value (2.5 in this case), the effects of requirement reuse diminishes. Using a δ lower than the minimum value, on the other hand, leads to the same results as using Griffin's original formula.

Since the number of samples, 5, is small, bootstrap sampling has been used to reach a confidence interval for δ value. For this purpose, minimum and maximum δ values have been resampled 10.000 times by using XLSTAT [42]. The results are given in Table 8.

As seen from the bootstrap resampling results, upper bound of minimum δ value is 2.102 and lower bound of maximum δ value is 2.107. That is;

$$[1.258, 2.102] \leq \delta \leq [2.107, 5.413]$$

This result supports our suggestion for using the values of δ between 2.102 and 2.107, which is the intersection interval for all bootstrap resamples. According to this approach, it is necessary to use the best value for δ within that range, so we selected δ value as 2.102.

The proposed modified versions of Eqs. (1) and (2) are presented below. The duration estimations include the engineering efforts during the requirements

Table 8. Bootstrap resampling of δ value.

Parameter	Mean (bootstrap)	Standard deviation (bootstrap)	Lower bound (standard bootstrap interval)	Upper bound (standard bootstrap interval)
δ minimum	1.679	0.152	1.258	2.102
δ maximum	3.755	0.595	2.107	5.413

Table 9. Expected and actual changes in duration of RE activities for projects A4, B6, C2 and C3 using the proposed formulation.

Case study	Project	Max. expected % of change in Req.	Expected % of duration decrease in RE works	Actual % of duration decrease in RE works	Compatibility to modified Griffin's formulation (Proposed formulation)
3	A5	25%	$\geq 20\%$	25%	Compliant
4	A7	37%	$\geq 12\%$	25%	Compliant
7	B6	39%	$\geq 7\%$	20%	Compliant
8	C2	16%	$\geq 34\%$	34%	Compliant
9	C3	29%	$\geq 20\%$	34%	Compliant

engineering phases. Other departments such as marketing and finance are not included within the scope of the case studies. Therefore, this modification is undertaken for the case where a cross functional team is not used in the organization.

$$\mathrm{DT} = \alpha + \beta_{1\mathrm{DT}} \times \mathrm{PC} + \beta_{2\mathrm{DT}} \times \delta \times \mathrm{NN} + \beta_{3\mathrm{DT}} \times (\mathrm{PC} \times \mathrm{FP}) + {}_{\mathrm{DT}} \tag{5}$$

$$\mathrm{CTC} = \alpha + \beta_{1\mathrm{CTC}} \times \mathrm{PC} + \beta_{2\mathrm{CTC}} \times \delta \times \mathrm{NN} + \beta_{3\mathrm{CTC}} \times (\mathrm{PC} \times \mathrm{FP}) + {}_{\mathrm{CTC}} \tag{6}$$

where $2.1 \leq \delta \leq 2.5$ (for the most reliable result δ is selected as 2.102).

The revised calculation for the software project cases is repeated using the proposed formulation. The calculations for software projects are performed below using Eq. (5)

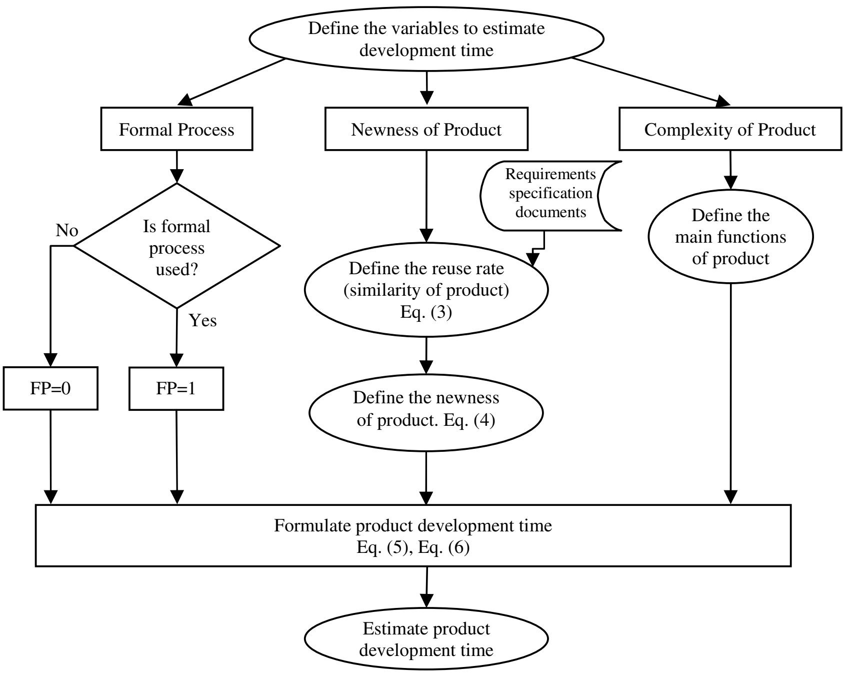

Fig. 2. Product development time estimation process.

and Eq. (6). Since the evaluations regarding the results of the case studies do not cover totally new products, Eqs. (1) and (2) are used for NN = 100%. The summarized results of software projects using the proposed formulation are given in Table 9.

At the beginning of this study, the research problem has been stated as "***How can product similarity be reflected to development time estimation at the beginning of a software intensive system development project***?" By the finding of the case studies, an accurate model has been proposed to estimate the product development time. Figure 2 outlines the process of estimating the project development time that reflects the knowledge gathered from the reported case studies. *Newness* of the product is defined by using the framework proposed in this study. Finally, the complexity of product is defined by counting the main functions of product.

6. Conclusion

Nine cases of software intensive systems projects have been studied, and it has been observed that Griffin's development duration estimation formula can be applied to systems projects involving both hardware and software components, while it has to be modified for purely software projects. Based on the findings of the case studies carried out, we have proposed to modify Griffin's formulation for software products. This is achieved by multiplying the *Newness/uncertainty* variable by coefficient δ which is between 2.1 and 2.5, in which case estimations have been observed to agree with actual project durations.

It is very likely that different projects in the same domain have many common requirements and if these requirements are maintained and shared in a common database to which all company personnel can access, systems engineers will choose to use these requirements in different projects. In this study we quantify the similarity of different products in same domain according to the number of reused requirements in the products. Newness is then derived from the similarity figures used in product development time estimations.

The results of this study are generalizable across development organizations, because the organizations considered in the case studies have been selected from different locations in the same country and they produce products in different sectors and the firms have different customers.

6.1. *Main contributions of the study*

The main contributions of the present study have been as follows:

- Griffin's product development time estimation model has been assessed via industrial case studies in three different organizations in the context of software projects and hardware and software systems projects.
- Nine industrial case studies have been conducted to present the requirements reuse approach to define product similarity in the same domain.

- A product similarity framework has been proposed to derive the newness of the product. This similarity framework is based on the requirements reuse among products in same domain.
- An extension to Griffin's project duration estimation model has been proposed for software projects. This involves the use of data from previous projects. This data has been incorporated into a mathematical model that facilitates a way of easily and more accurately estimating the project duration at earlier project stages.

6.2. *Limitations*

This study has focused on the requirements definition phase of projects which includes the efforts of a technical team. The projects covered in this phase do not include the efforts of non-technical departments such as marketing and finance. Therefore, this study has not addressed the effects of cross functional teams on product development times.

The products in each case study were in the same domain and each project used some of the requirements from previous projects. Thus, requirements reuse among different domains was not within the scope of this research.

Another limitation is that this study has not evaluated the complexity of requirements. All the requirements have been considered to affect the complexity by the same degree irrespective of their nature.

Finally, as stated at the beginning of the paper, having conducted qualitative research consisting of nine case studies carried out in different sectors, while we claim a certain level of generalizability of our extension to Griffin's method, our proposal remains a hypothesis deserving and needing verification in extensive and quantitative studies.

6.3. *Future work*

It is possible that the duration of a project can be further reduced by investigating reuse in the other phases of the project life cycle.

In the present work, the products considered in the case studies had different levels of complexity ranging from 3 to 7. Future work could address different products with wider variances of complexity to test the formulation.

To enhance the validity of the δ value in the proposed modification of Griffin's formulation, additional case studies can be performed and the effect of reuse can also be studied for organizations in which cross functional teams are used.

Acknowledgments

The authors are grateful to the companies who contributed the necessary data for the case studies to enable the completion of this research. The authors would like to thank Abbie Griffin for her explanatory responses to the questions.

References

1. M. Johnson and R. Kirchain, The importance of product development cycle time and cost in the development of product families, *J. Eng. Design* **22**(2) (2011) 87–112.
2. H. A. Bashir and V. Thomson, Metrics for design projects: A review, *J. Des. Stud.* **20**(3) (1999) 263–277.
3. J. Carter, Reducing Cycle Time by Optimizing Your Product Portfolio, Western Electronic Show and Convention (WesCon), 1996, pp. 8–11.
4. J. Callahan and B. Moretton, Reducing software product development time, *Int. J. Proj. Manag.* **9**(1) (2001) 59–70.
5. R. Filippini, L. Salmaso and P. Tessarolo, Product development time performance: Investigating the effect of interactions between drivers, *J. Prod. Innov. Manag.* **21**(3) (2004) 199–214.
6. W. H. A. Johnson, Z. Piccolotto and R. Filippini, The impacts of time performance and market knowledge competence on new product success: An international study, *IEEE Trans. Eng. Manag.* **56**(2) (2009) 219–228.
7. R. M. Lebcir and J. Choudrie, Impact of project complexity factors on project cycle time: A system dynamics modelling approach, in *2nd Int. Conf. Construction and Project Management*, 2011, pp. 166–170.
8. A. Griffin, The effect of project and process characteristics on product development cycle time, *J. Mark. Res.* **34**(1) (1997).
9. S. Yilmaz-Tastekin, Y. M. Erten and S. Bilgen, Product development time improvement with requirements reuse, in *7th International Conference on Software Engineering Advances*, Nov. 2012.
10. S. Yilmaz-Tastekin, Y. M. Erten and S. Bilgen, Software product complexity estimation using grey measurement, in *39th Euromicro Conference on Software Engineering and Advanced Application.*
11. H. B. Jun, H. S. Ahn and H. W. Suh, On identifying and estimating the cycle time of product development process, *IEEE Trans. Eng. Manag.* **52**(3) (2005).
12. A. Griffin, Metrics for measuring product development cycle time, *J. Prod. Innov. Manag.* **10**(2) (1993).
13. A. Griffin, PDMA research on new product development practices: Updating trends and benchmarking best practices, *J. Prod. Innov. Manag.* **14**(6) (1997) 429–458.
14. A. Griffin, Product development cycle time for business-to-business products, *Ind. Mark Manag.* **31**(4) (2002).
15. G. Barczak, A. Griffin and K. B. Kahn, Perspective: Trends and drivers of success in NPD practices: Results of the 2003 PDMA best practices study, *J. Prod. Innov. Manag.* **26**(1) (2009).
16. E. M. Olson, O. C. Walker, R. W. Ruekert and J. M. Bonner, Patterns of cooperation during new product development among marketing, operations and R&D: Implications for project performance, *J. Prod. Innov. Manag.* **18**(4) (2001) 258–271.
17. K. Dooley, A. Subra and J. Anderson, Best practices in new product development: Adoption rates, adoption patterns, and impact, *J. Innov. Manag.* **6**(1) (2002).
18. W. C. Lim, Effects of reuse on quality, productivity, and economics, *IEEE Softw.* **11**(5) (1994) 23–30.
19. Y. Chernak, Requirements reuse: The state of the practice, in *IEEE International Conference on Software Science, Technology and Engineering*, 2012, pp. 46–53.
20. S. Guo, W. Tong, J. Zhang and Z. Li, An application of ontology to test case reuse, in *International Conference on Mechatronic Science, Electric Engineering and Computer*, 2011.

21. G. Kotonya and I. Sommerville, *Requirements Engineering Processes and Techniques* (John Wiley & Sons, 1998).
22. O. Dieste, N. Juristo and F. Shull, Understanding the customer: What do we know about requirements elicitation, *IEEE Software* **25**(2) (2008) 11–13.
23. L. Goldin and M. Matalon-Beck, Reuse of requirements reduces time to market, in *IEEE International Conference on Software Science, Technology & Engineering*, 2010.
24. J. C. Lee and H. Lee, An ontological approach to measure similarity between different product information model, in *Proc. of the Asia Pacific Industrial Engineering & Management Systems Conference*, 2012.
25. D. P. Darcy, C. F. Kemerer, S. A. Slaughter and J. E. Tomayko, The structural complexity of software: An experimental test, *IEEE Trans. Softw. Eng.* **31**(11) (2005) 982–995.
26. S. C. Braun and U. Lindemann, The influence of structural complexity on product costs, in *IEEE International Conference on Industrial Engineering and Engineering Management*, 2008, pp. 273–277.
27. N. Orfi, J. Terpenny and A. Sahin-Sariisik, Harnessing product complexity: Step 1 — Establishing product complexity dimensions and indicators, *J. Eng. Econ.* **56** (2011) 59–80.
28. P. A. Murmann, Expected development time reductions in the German mechanical engineering industry, *J. Prod. Innov. Manag.* **11**(3) (1994) 236–252.
29. E. W. Larson and D. H. Gobeli, Significance of project management structure on development success, *IEEE Trans. Eng. Manag.* **36**(2) (1989) 119–125.
30. M. H. Meyer and J. M. Utterback, Product development cycle time and commercial success, *IEEE Trans. Eng. Manag.* **42**(4) (1995) 297–304.
31. S. Novak and S. D. Eppinger, Sourcing by design: Product complexity and the supply chain, *J. Manag. Sci.* **47**(1) (2001) 189–204.
32. T. J. McCabe, A complexity measure, *IEEE Trans. Softw. Eng.* **2**(4) (1976) 308–320.
33. H. Zuse, *Software Complexity Measures and Methods* (Walter De Gruyter, 1990).
34. J. Kim and D. Wilemon, Sources and assessment of complexity in NPD projects, *J. R&D Manag.* **33**(1) (2003) 15–30.
35. P. Runeson and M. Höst, Guidelines for conducting and reporting case study research in software engineering, *Empir. Softw. Eng.* **14** (2009) 131–164.
36. A. Cavaye, Case study research: A multi-faceted research approach for IS, *Inf. Syst. J.* **6** (1996) 227–242.
37. M. Gibbert, W. Ruigrok and B. Wicki, What passes as a rigorous case study? *Strateg. Manag. J.* **29** (2008) 1465–1474.
38. C. Herstatt, C. Stockstrom and A. Nagahira, NPD-Process and planning in Japanese engineering companies — Findings from an interview research, *Management of Technology and Innovation in Japan*, 2005, pp. 249–265.
39. J. J. Michalek, F. M. Feinberg and P. Y. Papalambros, Linking marketing and engineering product design decisions via analytical target cascading, *J. Prod. Innov. Manag.* **22**(1) (2005) 42–62.
40. L. J. Schimmoeller, Success factors of new product development processes, *J. Adv. Prod. Eng. Manag.* **5**(1) (2010) 25–32.
41. J. M. Bonner, R. W. Ruekert and O. C. Walker, Upper management control of new product development projects and project performance, *J. Prod. Innov. Manag.* **19**(3) (2002) 233–245.
42. http://www.xlstat.com/en/.