USE OF PROJECT SIMILARITY FOR SOFTWARE DEVELOPMENT TIME ESTIMATION

A THESIS SUBMITTED TO

THE GRADUATE SCHOOL OF INFORMATICS

OF

THE MIDDLE EAST TECHNICAL UNIVERSITY

BY

SEMRA YILMAZ TASTEKIN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPY

IN

THE DEPARTMENT OF INFORMATION SYSTEM

SEPTEMBER 2013

USE OF PROJECT SIMILARITY FOR
SOFTWARE DEVELOPMENT TIME ESTIMATION

Submitted by SEMRA YILMAZ TAŞTEKİN in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Information Systems, Middle East Technical University by,

Prof. Dr. Nazife Baykal ____________________
Director, Informatics Institute

Prof. Dr. Yasemin Yardımcı Çetin ____________________
Head of Department, Information Systems

Prof. Dr. Semih Bilgen ____________________
Supervisor, Electrical and Electronics Eng., METU

Assoc. Prof. Dr. Y. Murat Erten ____________________
Co-Supervisor, Innova Inc.

Examining Committee Members:

Prof. Dr. Onur Demirörs ____________________
Information Systems, METU

Prof. Dr. Semih Bilgen ____________________
Electrical and Electronics Eng., METU

Assoc. Prof. Dr. Y. Murat Erten ____________________
Innova Inc.

Assist. Prof. Dr. Erhan Eren ____________________
Information Systems, METU

Assoc. Prof. Dr. Altan Koçyiğit ____________________
Information Systems, METU

Date: 12 September 2013

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all materials and results that are not original to this work.

Name, Last name : Semra Yılmaz Taştekin

Signature :

ABSTRACT

USE OF PROJECT SIMILARITY FOR

SOFTWARE DEVELOPMENT TIME ESTIMATION

Yılmaz Taştekin, Semra
Ph.D., Department of Information System
Supervisor: Prof. Dr. Semih Bilgen
Co-Supervisor: Assoc. Prof. Dr. Murat Erten

September 2013, 145 pages

Accurate development time estimation is crucial for project management in general, and critical for software intensive systems projects, in particular. Before beginning the project, little information is available for development details. Therefore, development time may not be estimated correctly. If data on previous projects in the same domain is available, this can be used for development time estimations. At the beginning of the project, requirements are defined and requirements specification document is created as a formal document in the organizations. By using the reused requirements from previous projects, a similarity analysis can be performed and this analysis can be used for development time estimation.

This study investigates the applicability of a model that was proposed earlier for project management in general, in software intensive systems development projects. In this scope, the impact of requirements reuse on product development duration for different products in a similar domain is studied. Similarity analysis has been performed for different products in the same domain and the result of this analysis is used to estimate the development time. For development time estimation, Griffin's model [9] is used. For the applicability of Griffin's model for industrial companies, nine case studies from different organizations have been performed for software and system development projects which consist of hardware and software components. The results of the case studies are compared with Griffin's model.

According to the empirical results, a modification to Griffin's formulation for product development time is proposed for software projects. For the projects which include only software or both software and hardware, the proposed model will guide project managers to estimate project budgets more accurately.

Keywords: Product development time, product similarity based on requirements reuse

ÖZ

YAZILIM GELİŞTİRME SÜRESİ KESTİRİMİNDE

PROJE BENZERLİĞİNİN KULLANILMASI

Yılmaz Taştekin, Semra
Doktora, Bilişim Sistemleri
Tez Yöneticisi: Prof. Dr. Semih Bilgen
Yardımcı Tez Yöneticisi: Doç. Dr. Murat Erten

Eylül 2013, 145 sayfa

Geliştirme süresi kestirimi proje yönetimi açısından özellikle yazılım içerikli sistem projelerinde çok önemli bir husus olarak değerlendirilmektedir. Proje başlamadan önce ürünün geliştirme detayına yönelik çok fazla bilgi mevcut olmamaktadır. Bundan dolayı, geliştirme süresi doğru bir şekilde tahmin edilemeyebilir. Aynı alanda bulunan eski projeler var ise, bu projelerdeki veriler yeni projelerde kullanılabilmektedir. Projenin başında gereksinimler belirlenmekte ve gereksinim özellikleri dokümanı hazırlanmaktadır. Önceki projelerden kullanılan gereksinimlerin belirlenmesiyle benzerlik analizi yapılabilmekte ve bu analiz, geliştirme süresi kestirimi yapılmasında kullanılabilmektedir.

Bu tezde, daha önceden proje yönetimi kapsamında önerilen bir modelin, yazılım içerikli geliştirme projelerinde uygulanabilirliği çalışılmıştır. Bu kapsamda, aynı alanda bulunan farklı ürünler için gereksinimlerin tekrar kullanılmasıyla ürünlerin geliştirme süresine etkisi çalışılmıştır. Aynı alandaki farklı ürünler için benzerlik analizi yapılmış ve bu analizin sonuçları geliştirme süresi kestiriminde kullanılmıştır. Geliştirme süresi kestirimi için Griffin tarafından önerilen ürün geliştirme süresi modeli [9] kullanılmıştır. Griffin tarafından önerilen modelin endüstriyel organizasyonlarda uygulanabilirliğine yönelik olarak dokuz adet deneysel çalışma yapılmıştır. Bu deneysel çalışmanın kapsamında yazılım projeleri ve yazılım ve donanım ürünleri içeren sistem projeleri değerlendirilmiştir. Bu çalışmalardan elde edilen sonuçlar, Griffin tarafından önerilen model ile karşılaştırılmıştır.

Firmalardan elde edilen verilere göre, Griffin tarafından önerilen ürün geliştirme süresine ilişkin formülasyona, yazılım projelerine yönelik olarak bir modifikasyon önerilmiştir. Sadece yazılım içerikli veya yazılım ve donanım içerikli projelerde, önerilen model, proje yöneticilerinin projeyi daha doğru bütçelenmesinde yardımcı olacaktır.

Anahtar Kelimeler: Ürün geliştirme süresi, gereksinimlerin tekrar kullanılmasına dayalı ürün benzerliği

This thesis is dedicated to;

my son Ata,

my husband Ahmet

and my parents Ramazan & Tahide

ACKNOWLEDGEMENT

There are many people to thank for their support during my thesis study.

Firstly, I thank my thesis advisor Prof. Dr. Semih Bilgen. He provided valuable comments, guidance and support to complete this study in short duration. I would specially thank to my thesis co-advisor Assoc. Prof. Dr. Y. Murat Erten for his valuable time spent to me and for his contributed guidance to my study. Your time, advice and patience were important for me. I have learned from both of you how to approach academic problems, conduct research and complete the study within academic perspective. I could not have completed this study without your confidence and trust in me.

I am very grateful for the interest, suggestions and support of the committee members; Prof. Dr. Onur Demirörs, Assoc. Prof. Dr. Altan Koçyiğit and Assist. Prof. Dr. Erhan Eren.

I also would like to thank to Prof. Dr. Abbie Griffin for her support and prompt responses to my questions.

Thanks to my friends Seçkin Tunalılar, Oumout Chouseinoglou, Gökçen Yılmaz and Mina Nabi for their help and ideas to solve some important issues.

I am exceedingly grateful to my wonderful family, who has always encouraged me. Thank you for your love and support.

Thanks to my husband for sharing my life. You always supported me and helped me to complete this thesis.

Finally special thanks to my son, ATA, who always understood that I had to study even he wanted to spend time with me.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

COSMIC	:	Common Software Measurement International Consortium
CTC	:	Concept To Customer
DT	:	Development Time
FFP	:	Full Function Points
FPA	:	Function Points Analysis
FSM	:	Fucntional Size Measurement
FSU	:	Functional Size Unit
IEEE/EIA	:	Institute of Electrical and Electronics Engineers / Electronic Industries Associations
IFPUG	:	International Function Point Users Group
ISO/IEC	:	International Organization for Standardization / International Electrotechnical Commission
MIS	:	Management Information Systems
NN	:	Newness

PC	:	Product Complexity
PDMA	:	Product Development & Management Association
R&D	:	Research and Development
TTM	:	Time To Market
USA	:	United States of America
XFT	:	Cross Functional Team
WTTM	:	Weighted Time To Market

CHAPTER 1

INTRODUCTION

Important activities such as business development, contractual and technical reviews with the customer, project proposal preparation, submission of the proposal and resource allocation are performed well before the execution of project. In this process an essential element is that the product development time should be estimated. The development time estimate also forms the basis for the project cost. Without accurate development time estimation, assuring commitment to project activities would be very risky. Thus, estimation of product development time is a critical activity at the outset of any software intensive system development project.

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%. Similarly, Cuadrado-Garcia and Cuadrado-Gallego [3] discuss the importance of project prediction and monitoring activities in order to keep the project within the estimated cost and schedule. Therefore, project duration should not be underestimated as much as possible to complete projects as scheduled.

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. If an applicable and practical development time estimation model that benefits from requirements reuse is available, this would be very useful for industrial companies. By using more accurate duration data for development time, project managers will be able to complete the project proposals with fewer errors.

In the literature, there are numerous studies investigating techniques to reduce development time and metrics to control it. Some investigate organizational factors; others are related with people attributes and product determinants; yet other authors are concerned with technical issues. Carter [4] discusses product portfolio optimization to reduce development time. Callahan and Moretton [5] and Filippini et al. [6] address reducing the development time in terms of project management where supplier involvement, sales and marketing involvement in the development cycle. Callahan and Moretton [5] also discuss the impact of project experience on development time. Johnson et al. [7] discuss the importance of market knowledge on new product development success. Lebcir and Choudrie [8] investigate the influence of product complexity on product development time. Griffin's model [9], 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 [9]. Her model was developed based on measurements from 343 projects in different sectors, and has not been applied on, nor adjusted specifically for, industrial software intensive systems projects. This is what the present study aims to do.

This study focuses 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. Effort is an important determinant for development organizations to plan the projects. However during project execution customers usually emphasize the need for compliance of the schedule with the plan, rather than effort overruns, which may, if occur, possibly be absorbed by developers. Therefore, this study investigates duration of development projects rather than the issue of effort estimation and budget control.

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. Four of those case studies are involved with system products which include hardware and software components. The remaining five studies consider software development projects. In each case study, based on system and software requirements and their re-use, similarity and newness of each product was quantified. Duration data derived from the empirical case studies were compared with the expected durations obtained using Griffin's model. According to the main functionalities of each product, their complexities were determined. By taking into account the newness and complexity measures of each product, product development times were estimated using Griffin's model. The results of those estimations were 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 were found to be compatible with realized durations. Therefore, an extension to Griffin's formulation is proposed for development time estimation specifically for software projects. The proposed extension is based on the software similarity metric and is shown in the context of the presented cases to accurately reflect the effect of reuse on project duration.

For the industrial projects which may include only software or both software and hardware, it is believed that the proposed model will guide the project managers to estimate the project budget and schedule more accurately and correspondingly allocate the necessary resources to projects. So there will be less deviations from contractual commitments.

1.1 Thesis Outline

This thesis is organized as follows.

Chapter 2 briefly reviews the background of the problem of software product development time estimation, product similarity and product complexity concept. The proposed methods in the literature on these subjects are summarized.

Chapter 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.

Chapter 4 presents the findings of the nine case studies.

Chapter 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.

Chapter 6 concludes the paper with an overview of the proposed process model for estimating product development time. Limitations of the study and suggestions for possible future work are also presented.

CHAPTER 2

BACKGROUND

This study focuses on the applicability of an existing product development time estimation model [9] to industrial software and system projects. Griffin's model considers Newness and Complexity of the product to calculate the development time. Newness can be formulated if the similarity of the product to previous ones is known. Complexity is defined as the number of main functions which the product performs.

This section reviews the concepts of project parameters estimation, product development time estimation, product similarity based on requirements reuse and product complexity.

2.1 Project Parameters Estimation

For project proposal preparation and submission, product development time should be estimated however, since the customer generally requires the proposal to be submitted within a limited time this means that often there is insufficient information available concerning the development details. In order to accurately estimate the cost and schedule of a project it is important to have adequate information about the development time. The literature contains some estimation techniques including Delphi [10], Wideband Delphi ([11], [12]), Wisdom of Crowds [13], Planning Poker [14] and Proxy Based Estimating [15] which might be used in earlier project stages. For these methods, assumptions are made by team members to make progress on the decisions about development. Teams are established by members of the different departments that are included in the project work. Different people think differently and this may increase the accuracy of estimations.

The Delphi model was constructed by the RAND Corporation in the 1950s [16]. Estimations are based on input from experts and multiple iterations are undertaken to reach a consensus. The Wideband Delphi method was derived from the Delphi method, and essentially bases the task of estimation on a specialized team. The Wideband variant results from the inclusion of more participants in the estimating. To generate an estimate in Wideband Delphi, the project manager creates a team who decide on the estimation of the duration of the project development. As with the Delphi method this is an iterative process to achieve a consensus. The team members meet regularly to present their estimates and the meetings continue until the estimates of each team member are very close. In his book [13], Wisdom of Crowds, James Surowiecki writes about the estimation process stating that this technique involves a large group of people, such as the staff in a corporation or a group of researchers. The group members should understand the domain and be motivated to achieve the appropriate estimates. A variant of Wideband Delphi is Planning Poker described by James Grenning [14]. In this technique the estimators must think independently but they have to show their estimates (cards) at the same time. Proxy Based Estimating, in which project-specific experience of developers is significant, was introduced by Watts Humphrey [15]. In this system the history of the developers is used. For example, if a developer is building a component similar to one he has previously developed, his estimates are used to develop similar component.

Since these estimation techniques are essentially based on subjective approximations or experience of people, the input data for estimation might be imprecise or uncertain. Hence the need to reduce estimation risks remains. Of course, the estimates are very useful if there is no accurate data, however, business decisions and plans should rely on formal processes even if the team is able to produce very good estimations. It may not be possible to reach 100% accuracy but it is necessary to find ways to reduce the risks and errors to a minimum. In the following sub-section, proposed models for development time estimations are discussed.

2.2 Product Development Time Estimation

The starting point of the project is the release of the Request For Proposal document by the customer. The customer needs the budgetary and technical proposals from the bidders for selecting the most appropriate solution. The customer generally requires the proposals in limited durations. During this limited duration of project proposals, little information is available concerning the development details. There are some effort estimation methods that can be applied to software projects if the details of the software product such as functions points or lines of code are known. Before starting the product development, it is important to have the information about the development time to estimate the cost of the project used in the budgetary calculations in proposals. The estimation of product development time is also important for the efficient resource allocation in projects.

It is generally accepted that it is difficult to formulate a generic model for development time estimation [17]. A fundamental prerequisite for estimation of project duration in many cases is planning of all activities at a fine level of detail, which implies considerable progress in the project calendar. Most of the studies have been performed within the limitations of development activity classification which is referred as pattern, such as overlapping pattern, cycle pattern, communication pattern etc. These patterns designate the relationships between the activities during the design and development. Generally, a subset of those patterns is covered by studies in the literature. Furthermore, these studies require more detail about the development activities which may not be available during the project proposal time. Those are the number and definition of activities, information dependencies between the activities, degree of overlapping activity with another if available, starting time dependency of one activity to another etc. [17]. All these details necessitate fine planning of activities before the project proposal. At the beginning of the projects, these details might not be ready yet. Another shortcoming in the literature is the lack of case studies to validate the proposed development time models, such as [17].

In the following paragraphs, some of the proposed models are discussed.

Jun et al. [17] developed an estimation model for different types of relationships between activities. These relationships are called patterns and are classified according to information dependency, relation cardinality, degree of overlapping and type of collaboration. Information dependency shows the required outputs for one activity in relation to another. This pattern necessitates the definition of the detail of activities. Relation cardinality describes the relationship of two or more activities' relation to another activity. The degree of overlapping defines the start time of the activities. For example, to define the degree of activity, it is necessary to know the start time of activity B relative to the completion of activity A. The type of collaboration refers to the collaborations for the decision making process these might include; feedback, interaction, cycle, communication, branching and merging. To estimate the development time, a large amount of detail about the project should be obtained such as the number and types of activities, their start and end times, their inputs and outputs, dependency of each activity on others, iterations in the activities, repetition probability of each activity and the number of reworks have to be known. Accumulating this amount of detail is difficult, thus, this model does not seem very useful at earlier stages of projects.

Carter [4] proposed that the product development time be estimated using the organization's weighed average of the product category. According to Carter's taxonomy, products can be totally new, derivative or variant. By adjusting the product portfolio, the organization can reduce the development time or time-to-market. The Weighted Time To Market (WTTM) can be calculated as:

$$\mathbf{WTTM = Sum\ (Np \times TTMp + Nd \times TTMd + Nv \times TTMv)} \qquad \textbf{(Equation 1)}$$

where N_p, N_d and N_v show the percentage of the new, derivative and variant product in the portfolio of the organization, respectively; and TTM_p, TTM_d and TTM_v correspondingly show the average time-to-market (cycle time) of the categories of products in months.

For example, if a company has a portfolio mix of 40% new, 50% derivative and 10% variant products with a cycle time of 24, 8 and 6 months respectively, it will have an average WTTM of 14.2 months. This company can reduce the time-to-market by changing the portfolio percentages. This model is a simplified approach to define the time to market duration of organization and it does not consider the product characteristics explicitly. For different product features, it may not be applicable and may not give the correct duration information.

Griffin undertook a number of studies [9], [18], [19] and [20] to determine the time spent on product development and the factors that effect this duration. She proposed methods for determining the development time using the data collected from 343 projects from 21 divisions of 11 companies in different sectors such as chemical, communications, electro-mechanic and medical. Her proposed method was not applied to products consisting solely of software. She summarized the generalized results for the development time: if a product is new to the world, it takes approximately 53.2 months to develop; if it is new to the organization, it takes approximately 36 months to develop and a next generation project requires approximately 22 months for development [19].

An earlier study by Griffin [18] details the metrics for the measurement of the time spent on the development cycle and proposes a formula for 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. Therefore, it is important to determine the requirements accurately to ensure minimum change during development stage. And the studies [21], [22] and [23] also support this idea. Callahan and Moretton [5] observed that a factor that has a major effect on product development time is *Newness*/*uncertainty*.

The number of functions the product performs gives the complexity of the product [9]. Griffin hypothesizes that the development cycle time increases with greater product complexity, measured by the number of functions it performs. This hypothesis is also supported by Lebcir [24] who used the simulation results to show the effect of the product uncertainty and product newness on the development time. Simulations use product uncertainty, product newness, product inter-connectivity and product size to calculate the development size. There are some similar definitions in the literature for complexity. Larson and Gobeli [25] define the project complexity as the number of different disciplines or departments in the projects. Murmann [26] defines the product complexity as the number of part in the product. Meyer and Utterback [27] define the complexity as the number of core technologies in a product. Novak and Eppinger [28] define the product complexity as the number of product components plus coupling between these components and plus degree of product novelty.

If organizations do not have formal development processes, the development time is longer compared to those with formal development processes. Indeed, most organizations use formal processes during the product development cycle. In an empirical study by Barczak et al. [29], 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. [30] emphasizes that cooperation between specific functional departments associated with the new product being developed is important in increasing project performance.

Griffin [18] had 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.

If the organization has a formal development process, Griffin [18] formulates the DT for product complexity level 13 as;

$$\mathbf{DT=11.5 + 0.22*\Delta\ \%} \quad \textbf{months} \qquad \textbf{(Equation 2)}$$

If the organization does not have a formal development process, the DT [18] for product complexity level 13 is;

$$\mathbf{DT=17.2 + 0.36*\Delta\ \%} \quad \textbf{months} \qquad \textbf{(Equation 3)}$$

Where, Δ shows the changes in the product during development which can range from 0% to 100%. Study [18] suggests that there is a start-up time for development projects. This start-up time changes according to the complexity level of the product.

Without a formal process, an additional 5.7 months (from 11.5 months to 17.2 months) for the start-up time is required for product development. Moreover, the changes in the product denoted as Δ have more effect by an additional multiplication with 0.14.

In a later study based on a large dataset, Griffin [9] formulates the development time formulations as;

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

$$\mathbf{CTC = \alpha + \beta_{1CTC}*PC + \beta_{2CTC}*NN + \beta_{3CTC}*(PC*FP) + \beta_{4CTC}*(NN*XFT) + \epsilon_{CTC}} \quad \textbf{(Equation 5)}$$

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. ϵ 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. If the value of NN increases, the change probability of the product during the development process also increases.

Table 1 Coefficients Used in the DT and CTC Equations [9]

	α constant	**β_1 PC**	**β_2 NN**	**β_3 PC*FP**	**β_4 NN*XFT**
DT	8.4	4.2	0.09	-1.9	-0.09
CTC	10.4	3.7	0.16	0.1	-0.16

According to Equation (4), each main function (complexity) for the product requires 4.2 months (β_1). A 10% change in the product requires 0.9 months (β_2) in the expected duration of DT. The contribution of the *Complexity* and *Newness* to development time can be explained by an example with 50% new features of the product and 3 functions (complexity). The *Newness/uncertainty* contributes 4.5 months (β_{2DT}*NN = 0.09*50%) and complexity contributes 12.6 months (β_{1DT}*PC = 4.2*3) to the duration of the project development. In this example, the formal process will remove 5.7 months (β_{3DT}*(PC*FP) = -1.9*3*1) and a cross-functional team will remove 4.5 months (β_{4DT}*(NN*XFT) = -0.09*50%*1).

2.3 Analysis of Griffin's Study [9]

In Griffin's extensive range of research, she conducted some studies on new product development for example with Barczak et al. [29] and her own study [20] were PDMA Best Practice studies. Since Griffin's work, cited many academic publications and noted, for example by Dooley et al. [31], 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 development time and product complexity, newness of product and use of a formal process. Filippini et al. [6] considered that Griffin's model contributes to fill the gap on development time estimations. 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 [9].

It also covers the similarities from previous projects. For this purpose, a similarity analysis was performed based on reused requirements and the details of the analysis are given in the next sub-section.

Griffin's study [9] has been cited in many researches since it has been published. Following figures show the cumulative distribution of cited researches and number of citations over the years. These figures are retrieved from [32].

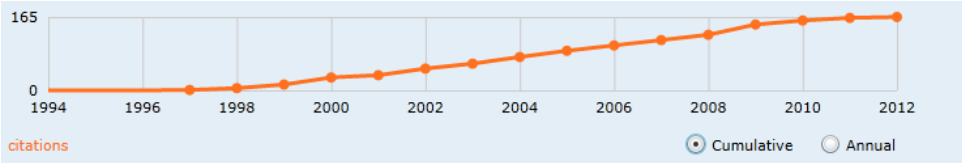

Figure 1 Cummulative Use of Griffin's Study over the Years [32]

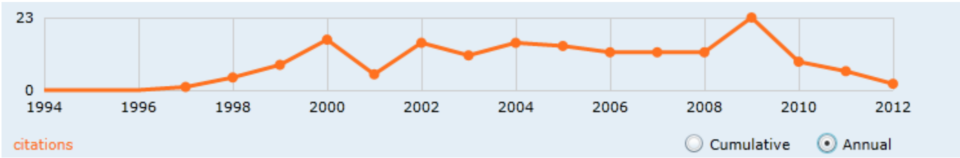

Figure 2 Number of Cited Researches of Griffin's Study Over the Years [32]

Griffin’s work [9] has been cited in the literature with different study subjects. The literature summary for some of the cited studies is given in Appendix A. As summarized in appendix, Griffin’s study was generally used on this field. Where some of the studies used her determinants in its hypothesis, others used her studies in the scope of the literature review and cited for the metrics which Griffin emphasized. Some of the studies stated the conformity with Griffin’s hypotheses. To the best of our knowledge, there is no study which improves the Griffin’s model for development cycle time.

2.4 Product Similarity based on Requirements

Reuse has traditionally been considered as a means for improving development productivity and quality [33] and it is widely accepted to lead to the introduction of faster, better and cheaper products into the market [34]. Furthermore, reuse of requirements can increase reuse in the later stages of projects [35]. Requirements or systems engineers need to ensure that the customer’s requirements are satisfied throughout the entire life cycle of the system.

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. [36], 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. Since requirements engineering process is followed by design and development processes, reuse in early stages of the life cycle will improve the requirements engineering process and support development with reuse [37]. Considering that 7-15% of total project resources are used for requirements engineering ([38], [39]), 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 [34]. 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. [40] point to the risk of getting requirements wrong when incremental development is used. In spite of these concerns, Chernak's empirical study [34] indicates that requirements reuse helps to reduce time to market as well as product cost. Moreover, according to Goldin and Matalon-Beck [41], requirements reuse reduces the development effort by 45%, development time by 33% and time-to-market by 60%. Reference [33] investigated code reuse and gave 51% and 24% defect reduction for two case studies.

Griffin's model [9] uses the NN (*Newness/uncertainty*) variable to estimate the product development time. According to that model, product *Newness* can be obtained in several ways. For manufactured products, *Newness* is defined as the ratio of the materials/parts list of the new product to that of a previously manufactured product. For formula based products, *Newness* can be calculated from formulary differences. In some cases, *Newness* is estimated by the team members. The measurement of *Newness* is an important issue in the more accurate estimate of the development time. It is obvious that a formal and objective metric for *Newness* is still needed.

In the literature, there are few studies that focus on *Newness*. Although Zhao [42] presents a scheme to classify products it does not provide a similarity measurement model. Ramdas et al. [43] propose a methodology to analyze the differentiation of physical products specifically in an existing product line of wrist watches. They define two types of similarity: objective and perceived. The former refers to the physical attributes of products and the latter is a subjective measurement of the overall similarity of two products. Politze and Dierssen [44] propose a model to measure the similarity between the functions of a product via subjective assessment of functional drivers. To use this method, it is necessary to define each function and to obtain the details of functions. Functional drivers require prior knowledge of the features of the functions which include; the reused parts, maturity of technology, realizations of functions and common functions used in all product variants. Lee and Lee [45] 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 were 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. Similarity can be defined as the feature or detail in which two items are alike whereas newness can be defined as the first of its kind. Similarity and newness of products are related to each other. The Similar and New functions create whole product. Product similarity analysis is based on the reused requirements of the products. As complementary, there is a strong link between product newness and corresponding new requirements to be developed [46]. Nine cases were examined with each case study containing two separate projects (1 and 2) in the same domain. To define the similarity of Project 2 to Project 1, the requirements specifications document is analyzed. To find an objective method for similarity, formal deliverables should be used. For this purpose, system/software requirements specifications document is used. For each case study, requirements are classified and number of requirements for each class is defined.

For the case studies, the requirements specifications documents are approved by customer for all projects, that is, System/Software Requirements Review phase has been completed for Projects 1 and 2 of all the case studies. Thus, it is doubtful to change the requirements of the projects. Project 1 has been completed or is near the completion stage depending on the case study. In this study, similarity is defined as the reuse rate of Project 2 to Project 1. Chernak [34] defines the reuse rate 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. For Griffin's model, the complement of *Similarity* is used for *Newness*. Both variables are formulated as:

$$\textbf{Similarity = Reuse Rate} = \frac{\textbf{(number of reused requirements from previous released requirements)}}{\textbf{(total number of requirements)}} \textbf{ x 100\%} \qquad \textbf{(Equation 6)}$$

$$\textbf{Newness (\%) = 100 – Similarity} \qquad \textbf{(Equation 7)}$$

In the present study, requirements were counted for running projects and completed (or close to completion) projects. For requirements similarity, requirements were identified according to their semantics instead of text based similarity. Requirements engineers evaluated the contents of each of the existing requirements and determined if the selected requirements should be identified as reused or not. During this evaluation it was also considered whether the selected requirements would lead to the implementation of the same functions in the product or not. If they did, these requirements were selected as similar requirements. Another issue was who evaluated the requirements and made decisions. The compared projects were in the same domain, therefore requirements were defined by the engineering team responsible from both projects in the same domain. This implied that the requirements semantics were comparable among multiple projects. Thus, it is believed that the selected requirements were comparable for the project in the scope of each case study. Basic approach to define the similarity in the scope of this study is shown in Figure 3.

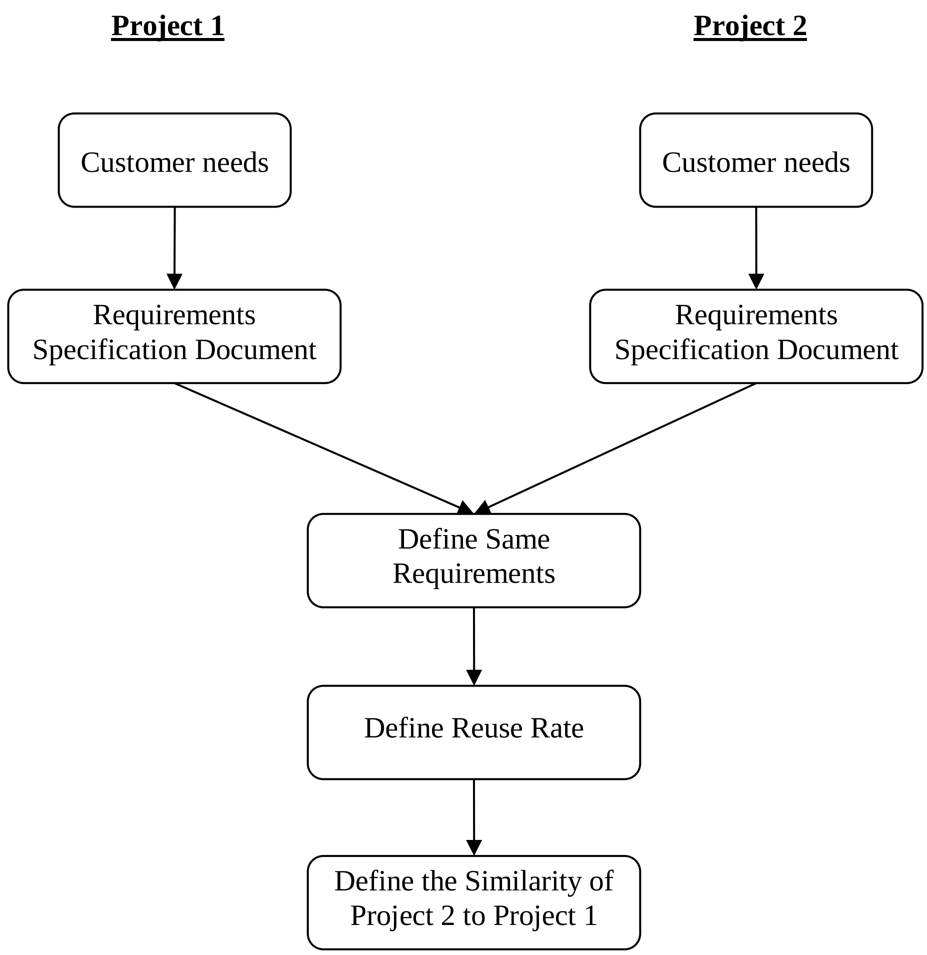

Figure 3 Determining the Product Similarity Process

Since the projects were in the same domain for each case study, it was not difficult to define the same requirements. Same requirements were mostly defined as the same written text in the specification documents. Ilyas and Küng [47] define requirements similarity as text-based or semantic based. They proposed a framework for text-based similarity. In the present study, similar requirements were mostly written with the same wording, but during the similarity analysis semantics were also explicitly considered. If the requirements had the same semantics but were written differently, they were identified as identical requirements. This judgment was performed by technical person involved in the projects.

The outcomes of the similarity analysis were then used to examine the applicability of an existing product development time model from the literature [9] to industrial products. In this way, the product development time estimation method was modified and the modified method was considered to be more applicable for the industrial software and system products. This current study was undertaken to find a more practical and applicable model for estimating the development time to make project budgeting tasks easier for project managers.

2.4.1 Importance of Requirements

During the project development lifecycle, each activity is very critical. If any phase cannot be completed as scheduled, this may result in the project not being completed as planned. The requirements phase, on the other hand, will have the greatest impact on the subsequent steps in the project lifecycle. If there are shortcomings in the requirements phase, its negative effect is reflected in the later phases. Requirements engineering is the first engineering activity that may fail during the system lifecycle. Failure in this stage will result in budget being exceeded, schedule delays, scope reduction, poor quality application and sometimes the cancellation of the project. On the other hand, success in this stage will result in, for example; the elimination defects, improvements in design, development and testing and will also decrease the cost. Moreover, some studies such as [48] indicate that about half of the factors associated with project or product success are requirements related.

The exponential increase of the cost of fixing a software defect depending on the development stage in which it was found is very well-known (e.g. [49]). For example, if a problem in the requirements is found only post-release, then it could cost 10–100 times more to fix than if it had already been found by the requirements review. Considering that 7-15% of the total project resources are used for requirements engineering [38], [39], the requirements-related phases of the development lifecycle should be realized as effectively as possible. The definition of the requirements is an important stage at the beginning, since the active development can only be started after this phase.

Moreover, requirements engineering activities have impact on different variables such as the project, product and company as mentioned in [39]. The project variable determines whether the project was completed on time, within budget, and met the requirements. The product variable determines the degree of product success, e.g., whether the product succeeds in fulfilling the needs of its intended customers. The company variable determines if the product has succeeded in the marketplace.

Requirements engineering becomes more complicated as the complexity of the products is increasing [48]. Therefore, it is continuously challenging by unstable requirements, product complexity and managing changes.

During the requirements definition phase, an important issue to be considered is the scope of the system. System development projects include the development of hardware components in addition to developing the software. The time taken to develop the hardware components cannot be ignored. First, time is needed for the procurement of materials and the development of electrical and mechanical components. Then, there is additional time for the qualification testing of the products which is performed to show the suitability of the product to the requirements such as; functional, electrical, mechanical and environmental (e.g. temperature, humidity, rain, fog, shock, vibration and electromagnetic compatibility) requirements. Such physical constraints are not considered in software products [5]. The software components are tested for function and performance however; the hardware components should also be tested for their compatibility with the environment in which they will be used. This activity takes additional time in the project lifecycle which is why the system projects take several years to complete. However, 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 duration of the project. This can be achieved by shortening of the phases in the lifecycle but it is also important issue to consider the final quality of product. Studies [18], [50], [51] and [52] show that trying to reduce the project duration inadvertently affects the product quality metric. Thus, the challenge is to maintain high product quality while reducing the duration of the project development.

The requirements engineering stage can be shortened by effective reuse of the requirements. Requirements are generated at the initial step of the project and followed by the development activities on the defined requirements. In the final step of the project, test activities are performed to determine whether the system meets all the defined requirements. Once the system is accepted by the customer, all the requirements defined at the beginning of the project are tested and approved by the stakeholders (customer, developers, systems engineers and test engineers). So, reusing these requirements will also facilitate the easy acceptance of the same features in similar systems produced in the subsequent projects. This will increase the quality attribute of the requirements, because already used and tested requirements will have positive impact on the test activities of the system and will reduce the project duration. The tested requirements always include the experience of implementation in previous projects. The increase in quality attributes by reuse has been confirmed by previous studies [33], [53] and [54].

All the stakeholders of the requirements should have the same understanding of the specific requirements. If the requirements are reused, developer will be familiar with them and they will be able to interpret the correct meaning of the requirements. This is defined as the internal quality of the system according to ISO/IEC 14598 [55]. In addition, the external quality represents the user’s view of the system. If the requirements are reused, customer will also be satisfied with the implementation of requirements in the system because they will be assured by the fact that reused characteristics of the system had been tested before and utilized by real users. Reusing the requirements at the beginning of the project will improve the requirements engineering activities of projects and will provide customer satisfaction.

Some of the requirements engineering issues are investigated in [56], [57], [58] and [59]. From these studies some of the critical problems that arise during the requirements engineering are summarized as;

- *Frequent changes in requirements.* Due to long development lifecycles of projects, customers will want to use that time to add new requirements to existing requirements.
- *Limited timeline.* The extent of the tasks undertaken in requirements engineering is generally underestimated when scheduling the project. Thus, requirement engineers tend to ignore important aspects of the system.
- *Poor requirements quality.* Requirements are generally misunderstood by the stakeholders. Users usually do not understand what they want and they do not understand the development process.
- *Inadequate requirement verification and validation.* Requirement defects are not identified during the requirements engineering process and requirements are not validated by the stakeholders.
- *Inadequate requirement process.* The requirements engineering phase is generally realized to be as-followed rather than as-documented.

All these problems necessitate good requirements definition in a limited time. This will lead to the use of existing materials in an organization. Thus, requirements reuse is an important technique to be employed. Through tested and re-used requirements, customer acceptance will be quicker and their satisfaction with the requirements will be higher, possibly leading to fewer requests for changes to the product. By using the existing requirements, time will be used efficiently. Reusing requirements in more projects in the same domain will complete the requirement process faster than expected.

The ISO/IEC 9126 [60] quality model includes the attributes of; functionality, reliability, usability, efficiency, maintainability and portability. If requirements are reused, most of these attributes will have been met in previous projects. This will improve the quality of the requirement phase. Requirements reuse will also motivate the reuse of development activities, such as the reuse of code, hardware components and existing modules. Since reused requirements are those that have been implemented and tested in other projects, this will also help improve the efficiency of the system. Requirements reuse will not only simplify the development and implementation activities but also reduce the number of defects that are identified later in the implementation [61].

Requirements reuse will also reduce the project and time-to-market duration. In the literature there are several theoretical studies concerning software components reuse. Study [37], [62] and [63] define some classification approaches to support the reusability in the organization repository. Study [64] focuses on the quality requirement while reusing the software components. Likewise studies [53], [65] and [66] discuss software components reuse in the projects. There is little research concerning hardware components but reusing hardware related assets are as least as important as the software related components. Reusing hardware related requirements will support the reutilization of hardware components which are the output from previous projects.

Reuse approaches for hardware should be considered carefully since the reuse of hardware may not be similar to software reuse. Different disciplines should be included in hardware reuse, such as mechanical, and electrical engineering [53]. For example, hardware used in one product may be used in another in the same domain. So they can be considered as common assets in an organization. The procurement of such components can be quicker if they have been used in different projects.

2.5 Project Complexity

Project complexity has effects on project determinants such as cost, duration and resource allocation [67], [68], [69]. If the project complexity estimation is not based on the accurate measurement models, project planning will be envisioned on inaccurate predictions. In the later phase of the development process, this will affect the software quality due to the wrong planning. Finally, it will be difficult to achieve the targets of project. It will also be difficult to manage the complexity if it is not measured.

As defined in the general context of engineering, by Griffin [9], Lebcir and Choudrie [8], Zhang and Luo [70], product complexity has a significant impact on product development time. It has been shown, definitively, that development duration is higher for products of higher complexity [9], [71]. Based on functional descriptions of products, some complexity measurement methods have been proposed in the literature [72], [73], [74] for software products. These are mostly derived from Function Points Analysis (FPA) or COSMIC Full Function Point (COSMIC-FFP) estimations. Similarly, software complexity measurement based on the cognitive functional size of software is proposed by Shao and Wang [75]. However, functional size is not the only concern in defining complexity and function point analysis alone is insufficient for complexity evaluation [73]. It has been argued and shown that software complexity should be assessed under multiple dimensions ([46], [76]). Complexity measurement should consider the technological and organizational factors besides the commonly used size measures.

To arrive at a comprehensive assessment of project complexity, grey measurement theory can be of use. Grey measurement focuses on problems involving small samples or poor data [77]. It deals with the known information to generate useful data. It is used to identify the objects into the different clusters. Grey measurement theory supports the decision making process by judging the grey clusters that the object belongs [78]. Zhang and Luo [70] propose a framework for product complexity based on grey measurement and present some indicators of system complexity. However, their framework cannot be applied to software products, because it covers some physical properties of the products under consideration. The present study deals exclusively with software products and proposes a software complexity evaluation model based on a modification of the framework proposed by Zhang and Luo [70].

There is no universally accepted definition of the dimensions of software product complexity [69]. As mentioned in Section 2.2, Murmann [26], Larson and Gobeli [25], Meyer and Utterback [27], Novak and Eppinger [28] define complexity in different aspects. Similar concepts are also applied to determine software complexity. McCabe [79] 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 [80]. 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 [81] and Orfi et al. [69]. Such widely varying definitions of product complexity lead to difficulties in arriving at a universally accepted way of measuring it [69].

Zhang and Luo [70] consider the different factors affecting complexity and propose a grey measurement model based on technology, size, organization and environment determinants. Each determinant includes indicators of complexity. According to the evaluations and grading for each indicator, they introduce what they call a “triangle whitenization function” to derive grey measurement complexity. *Technology, Physical Size, Organization* and *Environment* determinants comprise the first level of determinants and for each determinant, there are second level indicators. Second level indicators for the *Technology* determinant are Number of Technologies and Maturity of Technology Users. Second level indicators for the *Physical Characteristics* determinant are Number of Components, Volume and Density; for the *Organization* determinant the indicators are People, Department, Information Transfer and Resource Allocation. Finally, second level indicators for the *Environment* determinant are Number of Suppliers and Customers, Regulations and Standards and Market and Competitions. These indicators listed in [70] are applicable for system products as they incorporate physical characteristics, but for software products, they must be revised. That method is modified in the scope of current study to be applicable for software products. In the following subsections, the detail of modification and measurement method are presented.

2.5.1 Product Complexity Measurement Model

In this section, proposed grey measurement model based on the study by Zhang and Luo [70] is presented. For the grey measurement, first level determinants and second level indicators belonging to those determinants are defined. Each indicator is scored and a weight is calculated for each of them. After that, whitenization functions are measured and grey coefficient is derived from whitenization functions. Finally, a complexity evaluation can be performed by using the grey coefficients. This study contributes to the study of Zhang and Luo [70] by concerning the software characteristics which influence software product complexity.

2.5.1.1 Determination of Indicators

Similar to the model proposed by Zhang and Luo [70], a grey measurement model is described in this section. Their model does not reveal the factors affecting the software complexity. Therefore, the factors for software projects are analyzed and the indicators for software projects are modified. For system projects, indicators defined by Zhang and Luo [70] are used.

First level determinants for software complexity are *Software Characteristics*, *Technology*, *Organization and Environment*. Important factors affecting software complexity are classified as Functional Size and Interaction with Other Systems. These are determined as second level indicators of the *Software Characteristics.* Second level indicators of

Technology are Number of Technologies and Maturity of Technology Users. Second level indicators of *Organization* and *Environment* are as defined by Zhang and Luo [70]. The first level determinants and second level indicators of the system and software projects are presented in Table 2 and Table 3. In these tables, *X* is used to symbolize each indicator.

Table 2 First Level Determinants and Second Level Indicators of System Project Complexity

First Level Determinants	Second Level Indicators	Unit	Symbol	Criteria
Technology	Number of Technology	per	X1	number of technology developed in the product
	Maturity of Technology Users	1-4	X2	expert evaluation
Physical Characteristics	Number of components	per	X3	actual number of components
	Volume	m3	X4	actual volume
Organization	People	person	X5	number of people involved in project
	Departments	per	X6	number of departments involved in project
	Information transfer	per file	X7	files transfered (the amount of information transferring among the people and departments)
	Resource allocation	1-4	X8	expert evaluation
Environment	Number of suppliers and customer	per	X9	actual number of suppliers and customers
	Requlations and standards	per file	X10	requlations and standards to be followed
	Market and competitions	1-4	X11	expert evaluation

Table 3 First Level Determinants and Second Level Indicators of Software Project Complexity

First Level Determinants	Second Level Indicators	Unit	Symbol	Criteria
Software Characteristics	COSMIC FFP Size	cfsu	X1	COSMIC functional size
	Interactions with other system	1-4	X2	expert evaluation
Technology	Number of Technology	per	X3	number of technology developed in the product
	Maturity of Technology Users	1-4	X4	expert evaluation
Organization	People	person	X5	number of people involved in project
	Departments	per	X6	number of departments involved in project
	Information transfer	per file	X7	files transfered (the amount of information transferring among the people and departments)
	Resource allocation	1-4	X8	expert evaluation
Environment	Number of suppliers and customer	per	X9	actual number of suppliers and customers
	Requlations and standards	per file	X10	requlations and standards to be followed
	Market and competitions	1-4	X11	expert evaluation

Except for Software Characteristics, other determinants are as proposed by Zhang and Luo [70]. This study proposes a new determinant named Software Characteristics for complexity measurement. Therefore, it is necessary to clarify the rationale for using the indicators of Software Characteristics. The indicators of this determinant are Functional Size and Interaction with Other Systems. It is accepted that there is a relationship between the software size and its complexity ([75], [82]). FPA is a standard method for measuring software size and it can be used as a measure of software complexity in the early stages of development [73]. The functional complexity of software characterizes the dynamic performance of the product [72]. Therefore, since it affects the overall complexity of software products software functional size is specified as a second level indicator of Software Characteristics.

For functional size estimation of software, various approaches were evaluated by Gencel and Demirors [85]. The following four approaches have been approved by International Organization for Standardization as functional size measurement methods; International Function Point Users Group Function Points Analysis (IFPUG FPA), Common Software Measurement International Consortium-Full Function Points (COSMIC-FFP), Mk II FPA (Function Point Analysis) and NESMA FSM (Functional Size Measurement). IFPUG FPA and COSMIC-FFP are the main methods and widely used [86]. IFPUG FPA was developed for MIS. However even though it is applicable to MIS projects, it was necessary to extend its applicability to a larger set of projects [87]. COSMIC-FFP was developed as a second generation functional size measurement method to address this applicability issue to real-time, technical, system and MIS projects [87], [88]. Since COSMIC FFP can be widely applied, this method is used to define the functional size of software.

Another issue for determining software complexity is the interaction of the software with other systems. Tran-Cao and Lévesque [73] discuss the data dependency between functional processes in terms of the number of connections between them. Similarly, Lebcir [46] discusses the interdependency of software with other systems or subsystems. Ameri et al. [83] identify different dimensions of complexity and study the connections of product components. Dependency on other systems will increase the complexity of products. Therefore, interaction of the software with other systems should also be considered as an indicator.

2.5.1.2 Rating of Indicators

For the grey measurement of complexity, firstly each indicator is rated as proposed by Zhang and Luo [70]. Some of the indicators are qualitative such as: Interaction with Other Systems, Maturity of Technology Users, Resource Allocation, and Market and Competitions. These are evaluated by experts with similar project experience who take into account the product features. The evaluation criteria used in assessing qualitative indicator values are listed in Table 4. The evaluation criteria for qualitative indicators are rated between 1 and 4 in a similar way to that proposed by Zhang and Luo [70]. These criteria for each indicator are discussed below.

Table 4 Evaluation Criteria for Qualitative Indicators

Indicator	Score	Evaluation criteria
Interaction with other systems	4	A complex system and has more than five interfaces [84] with other systems
	3	A complex system and has five or less interfaces with other systems
	2	A moderate/complex system and does not have any interfaces with other system
	1	A simple system and does not have any interfaces with other systems
Maturity of Technology Users	4	New technology, proof of concept stage with limited testing
	3	New technology for team, only working prototypes developed and tested outside
	2	Technology was developed and tested outside the organization
	1	Use of mature technology in organization and team
Resource allocation	4	Lack of internal resources, need for a large scale resource integration
	3	Key resources available internally, need for a certain scale of resource integration
	2	Internal resources meeting basic requirements, need for some resources integration
	1	Enough internal resources, no need for resources integration
Market & competitions	4	Limit experience, new target market
	3	Basic marketing function, experience in marketing similar products
	2	Major marketing function, a major player in target market
	1	Complete marketing function, a leader in target market

The new indicator, Interaction with Other Systems was not defined by Zhang and Luo [70]. In the present study it was evaluated by technical experts. Similar to the qualitative criteria defined by Zhang and Luo, the evaluation criteria present general complexity evaluation of the software and interfaces to other systems/subsystems. The experts' evaluation was based on how they define overall system in terms of complexity and whether the product has interfaces with other systems.

Zhang and Luo [70] do not explicitly define the Technology dimension in their study. In the present study, Number of Technologies is rated as the number of technologies used in the development of the products. These can be programming languages such as Java, C++, ASP, or database management systems such as Oracle, SQL, or special software interfaces developed in the product such as military standard interfaces as MIL-STD-1553 for software products. For system products, these can be modulation technology, encryption method, frequency hopping, spreading, special antenna technology (such as phase array), power supply technology plus number of manufacturing technology (microwave production, mechanical technology such as deep-freezing, balistic protection etc).

In terms of the Maturity of Technology Users after defining the technologies in the product, maturity of each technology and team is evaluated and then a general assessment is performed by experts to evaluate the maturity. The technology life cycle has four stages; introduction, growth, maturity and decline as defined by Papageorgiou [89]. Reinhart and Schindler [90] present four similar stages for manufacturing technology. The present study adapts the four stages of technology maturity and uses them in the evaluation and rating. Maturity is evaluated for the product to be developed and the team to develop that product. For example, although Oracle might have been used in an organization over a period of time, if the team developing the product lacks experience with Oracle, this database management system should be evaluated as a technology used immaturely by team. If the team has experience with Oracle, then it is evaluated as a mature technology.

The People and Department indicators contain the number of people and departments, respectively, involved in the project. Information Transfer represents the amount of information that is transferred between people and departments. The number of files transferred is counted for this indicator. Resource Allocation is evaluated by experts in relation to the internal resources of organization.

To evaluate the Number of Suppliers and Customers, the suppliers and customers involved in the project are counted. Similarly, Regulation and Standards are specified and each document is added to the regulations and standards indicator. Market and Competitions is evaluated by experts to determine the position of organization in market.

2.5.1.3 Weight of Indicators

After scoring the indicators, their weights should be determined for grey measurement. The weight of each indicator is determined in the same way as in the study by Zhang and Luo [70] and their formulations are used in this study.

The first level determinants are rated according to the relative importance between adjacent ones. These are r_i indices. r_i indices of first level determinants are rated between 1 and 1.5. A greater level has more importance than the others. r_i' is the absolute importance of each first level determinant. Similarly, $r_{i,j}$ and $r_{i,j}$' are the indices for the second level indicators. Finally, f_i gives the weight of the first level determinants and η_j gives the weight of the second level indicators. The sum of the weights should be equal to 1. The relationships among these parameters are given below.

$$r_i' = r_i \cdot r_{i+1}' \qquad i=1,2,\ldots,m-1$$
$$r_i' = 1 \qquad i=m$$
(Equation 8)

$$f_i = r_i' \Big/ \sum_{i=1}^{m} r_i'$$
(Equation 9)

$$r_{i,j}' = r_{i,j} \cdot r_{i,j+1}'$$
(Equation 10)

$$\eta_j = f_i \cdot r_{i,j}' \Big/ \sum_{i=1}^{m} r_i'$$
(Equation 11)

To derive the weight of each indicator, firstly, the r_i, and $r_{i,j}$ indices for each determinant and indicator are rated by the experts. Subsequently, r_i', and $r_{i,j}$' indices are determined by using Equations (8) and (10). Then the weight of each indicator is calculated using the formulations given in Equations (9) and (11). In this study, data from an MIS project which includes the development of a Document Management System (DMS) is provided to exemplify the application of the measurement model. Table 5 shows the calculated weights of the indicators for DMS project as an example. r_i, and $r_{i,j}$ indices for each determinant and indicator are determined by project manager of that project.

Table 5 Determinants, Indicators and Weights of DMS Project

First Level Determinants	r_i	r_i'	f_i	**Second Level Indicators**	$r_{i,j}$	$r_{i,j}$'	η_j
Software Characteristics	1.5	2.7	0.403	COSMIC FFP Size	1.5	1.5	0.2418
				Interactions with other system		1	0.1612
Technology	1.5	1.8	0.269	Number of Technology	1.4	1.4	0.1567
				Maturity of Technology		1	0.1119
Organization	1.2	1.2	0.18	People	1.3	2.028	0.0617
				Departments	1.2	1.56	0.0475
				Information transfer	1.3	1.3	0.0395
				Resource allocation		1	0.0304
Environment	1	1	0.15	Number of suppliers and customer	1.2	1.68	0.0615
				Requlations and standards	1.4	1.4	0.0512
				Market and competitions		1	0.0366

2.5.1.4 Grey Clusters

Three level grey clusters are used in the measurement (k=1, 2, 3). These grey clusters define the complexity of project in terms of “low”, “moderate” and “high”. For each cluster type, a range is determined corresponding to each indicator as given in Table 6. Except for the indicators of the Software Characteristics, the other indicators use the same ranges as defined by Zhang and Luo [70]. For COSMIC FFP Size indicator, the study by Buglione et al. [91], who collected data from 25 projects and measured the COSMIC functional size of those projects, are used. Using their descriptive statistics, the range of low, moderate and high clusters are determined for the COSMIC FFP Size indicator as given in Table 6 together with the DMS project data are also provided.

Table 6 Indicators and Grey Clusters of DMS Project

X_j	Actual Value	Weight η_j	Low Cluster	Moderate Cluster	High Cluster
X1	386	0.2418	[163,334]	[334,870]	[870,1090]
X2	3	0.1612	[1,2]	[2,3]	[3,4]
X3	7	0.1567	[2,5]	[5,15]	[15,30]
X4	1	0.1119	[1,2]	[2,3]	[3,4]
X5	9	0.0617	[5,20]	[20,50]	[50,150]
X6	2	0.0475	[1,5]	[5,10]	[10,30]
X7	12	0.0395	[5,20]	[20,50]	[50,150]
X8	1	0.0304	[1,2]	[2,3]	[3,4]
X9	3	0.0615	[1,10]	[10,30]	[30,100]
X10	5	0.0512	[2,5]	[5,10]	[10,20]
X11	2	0.0366	[1,2]	[2,3]	[3,4]

In Table 6, X_j's correspond to the symbols in Table 3 and Weight of each Indicator was calculated by using Equations (8) to (11) as given in Table 5. Actual values of each indicator are rated by the experts involved in project.

To find the complexity assessment, it is necessary to calculate the whitenization function, *f(x)*, for each grey cluster. This is performed by using the following equation as presented by Zhang and Luo [70].

$$f_j^k(x) = \begin{cases} 0 & x \notin [a_j^{k-1}, a_j^{k+2}] \\ \dfrac{x - a_j^{k-1}}{\lambda_j^k - a_j^{k-1}} & x \in [a_j^{k-1}, \lambda_j^k] \\ \dfrac{a_j^{k+2} - x}{a_j^{k+2} - \lambda_j^k} & x \in [\lambda_j^k, a_j^{k+2}] \end{cases} \quad \textbf{(Equation 12)}$$

Here, a^{k-1} and a^{k+2} are the values out of the cluster ranges, these values should satisfy $a^0<a^1$ and $a^{k+2}>a^s$, where s is the number of grey cluster. λ is the whiten value of k grey cluster and calculated as below:

$$\lambda_j^k = (a_j^k + a_j^{k+1}) / 2 \qquad \textbf{(Equation 13)}$$

Using (12) and (13), whitenization function of each grey cluster for DMS is calculated and given in Table 7, Table 8 and Table 9.

Table 7 Whitenization Function for Low Grey Cluster of DMS Project

X_j	Actual Value	Low $[a^1, a^2]$	λ_j^1	η_j	$f_j^1(x_j)$
X1	386	[163,334]	248.50	0.2418	0.779
X2	3	[1,2]	1.50	0.1612	0
X3	7	[2,5]	3.50	0.1567	0.696
X4	1	[1,2]	1.50	0.1119	0.667
X5	9	[5,20]	12.50	0.0617	0.72
X6	2	[1,5]	3.00	0.0475	0.667
X7	12	[5,20]	12.50	0.0395	0.96
X8	1	[1,2]	1.50	0.0304	0.667
X9	3	[1,10]	5.50	0.0615	0.545
X10	5	[2,5]	3.50	0.0512	0.769
X11	2	[1,2]	1.50	0.0366	0.667

Table 8 Whitenization Function for Moderate Grey Cluster of DMS Project

X_j	Actual Value	Moderate $[a^2, a^3]$	λ_j^2	η_j	$f_j^2(x_j)$
X1	386	[334,870]	602.00	0.2418	0.508
X2	3	[2,3]	2.50	0.1612	0.667
X3	7	[5,15]	10.00	0.1567	0.625
X4	1	[2,3]	2.50	0.1119	0
X5	9	[20,50]	35.00	0.0617	0.133
X6	2	[5,10]	7.50	0.0475	0.154
X7	12	[20,50]	35.00	0.0395	0.233
X8	1	[1,2]	2.50	0.0304	0
X9	3	[1,10]	20.00	0.0615	0.105
X10	5	[2,5]	7.50	0.0512	0.545
X11	2	[1,2]	2.50	0.0366	0.667

Table 9 Whitenization Function for High Grey Cluster of DMS Project

X_j	Actual Value	Moderate $[a^3, a^4]$	λ_j^3	η_j	$f_j^3(x_j)$
X1	386	[870,1090]	980.00	0.2418	0.08
X2	3	[3,4]	3.50	0.1612	0.667
X3	7	[15,30]	22.50	0.1567	0.114
X4	1	[3,4]	3.50	0.1119	0
X5	9	[50,150]	100.00	0.0617	0
X6	2	[10,30]	20.00	0.0475	0
X7	12	[50,150]	100.00	0.0395	0
X8	1	[3,4]	3.50	0.0304	0
X9	3	[30,100]	65.00	0.0615	0
X10	5	[10,20]	15.00	0.0512	0
X11	2	[3,4]	3.50	0.0366	0

To evaluate software complexity, the comprehensive grey coefficient, σ, was measured as given below. Three different complexities were calculated corresponding to whitenization function $f_j^k(x)$. Maximum of the comprehensive grey coefficients was selected as the complexity of project.

$$\sigma^k = \sum_{j=1}^{m} f_j^k(x_j) * \eta_j \qquad \textbf{(Equation 14)}$$

$$\sigma = max\{\sigma^k\} \qquad \textbf{(Equation 15)}$$

Table 10 summarizes the whitenization function calculated by using (12) for each of the grey clusters and corresponding grey coefficients obtained by (14). In this example, the software complexity of DMS product was found to be “Low” where the grey cluster “low” has the maximum grey coefficient.

Table 10 Whitenization Functions and Grey Coefficients

X_j	$f_j^1(x_j)$ (Low)	$f_j^2(x_j)$ (Moderate)	$f_j^3(x_j)$ (High)
X1	0.779	0.508	0.08
X2	0	0.667	0.667
X3	0.696	0.625	0.114
X4	0.667	0	0
X5	0.72	0.133	0
X6	0.667	0.154	0
X7	0.96	0.233	0
X8	0.667	0	0
X9	0.545	0.105	0
X10	0.769	0.545	0
X11	0.667	0.667	0
σ^k	**0.604**	**0.412**	**0.145**

2.5.1.5 Complexity Determinations of the Projects

The complexities of nine projects were measured according to the grey measurement method described in the previous section. Product related data was gathered by joint effort of project technical managers and author of this study. Actual values of each indicator for the nine projects are presented in Appendix B. Then, grey measurement is applied for each project. The results of grey measurement are given in Table 11.

Table 11 Complexities of Projects in the scope of Case Studies

Case Study	Product	σ^1 (Low)	σ^2 (Moderate)	σ^3 (High)	Complexity Measurement (according to grey measurement)	Complexity Level (main functions)
1	A2	0.569	**0.648**	0.117	Moderate	6
2	A3	0.532	**0.605**	0.150	Moderate	6
3	A5	0.548	**0.592**	0.194	Moderate	5
4	A7	**0.724**	0.240	0	Low	3
5	B2	0.468	**0.476**	0.143	Moderate	6
6	B4	0.509	**0.511**	0.109	Moderate	6
7	B6	0.115	0.222	**0.318**	High	7
8	C2	**0.604**	0.412	0.145	Low	3
9	C3	**0.701**	0.428	0.048	Low	3

Table 11 also gives the main functions of the nine products as defined by project technical managers. When a comparison is made between the grey complexities and main functions of the products, the number of main functions may linearly correspond to the grey complexity. Products A7, C2 and C3 have 3 main functions and these were measured as “Low” in complexity. Product A5 has 5 main functions and Products A2, A3, B2 and B4 has 6 main functions and they were measured as “Moderate” complexity. Finally, Product B6 having more functions compared to the other products and had a complexity correspondingly measured as “High”. This implies that the number of main functions is indicative of product complexity.

According to Griffin’s research, complexity levels are between 1 and 11. By using Griffin’s approach and the results of grey measurements for nine projects, it can be inferred that if a product has a number of main functions from 1 to 4, its complexity level might correspond to “low”. If it has 5 or 6 main functions, its complexity might be evaluated as “moderate” and finally 7 to 11 main functions correspond to “high” complexity.

CHAPTER 3

RESEARCH METHODOLOGY

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

3.1 Research Problem and Method

Since projects within industrial organizations are generally completed over budget and having exceeded the estimated scheduled project development time [92] the motivation for this study was to determine how to generate an accurate estimation of the product development time for industrial organizations. Dunham [92] summarizes the interlinking factors in estimating development time. There is a relationship between time-effort factors and cost-resources-specification factors. For example changes or out of control in specification will have an impact on the time or effort estimations in the project. 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 the project, 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 development project?

The unit of analysis for our case study research is a software or system development project. Multiple cases from three different companies were studied. The research method depicted in Figure 4 was employed. Initially an extensive literature survey was conducted based on product development time and product similarities. Based on that survey, Griffin's model [9] originally proposed for manufacturing industries was chosen and a case study approach to investigate its applicability in software intensive development projects was used. Furthermore, similarity analysis was 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 was expected that a modification would be needed.

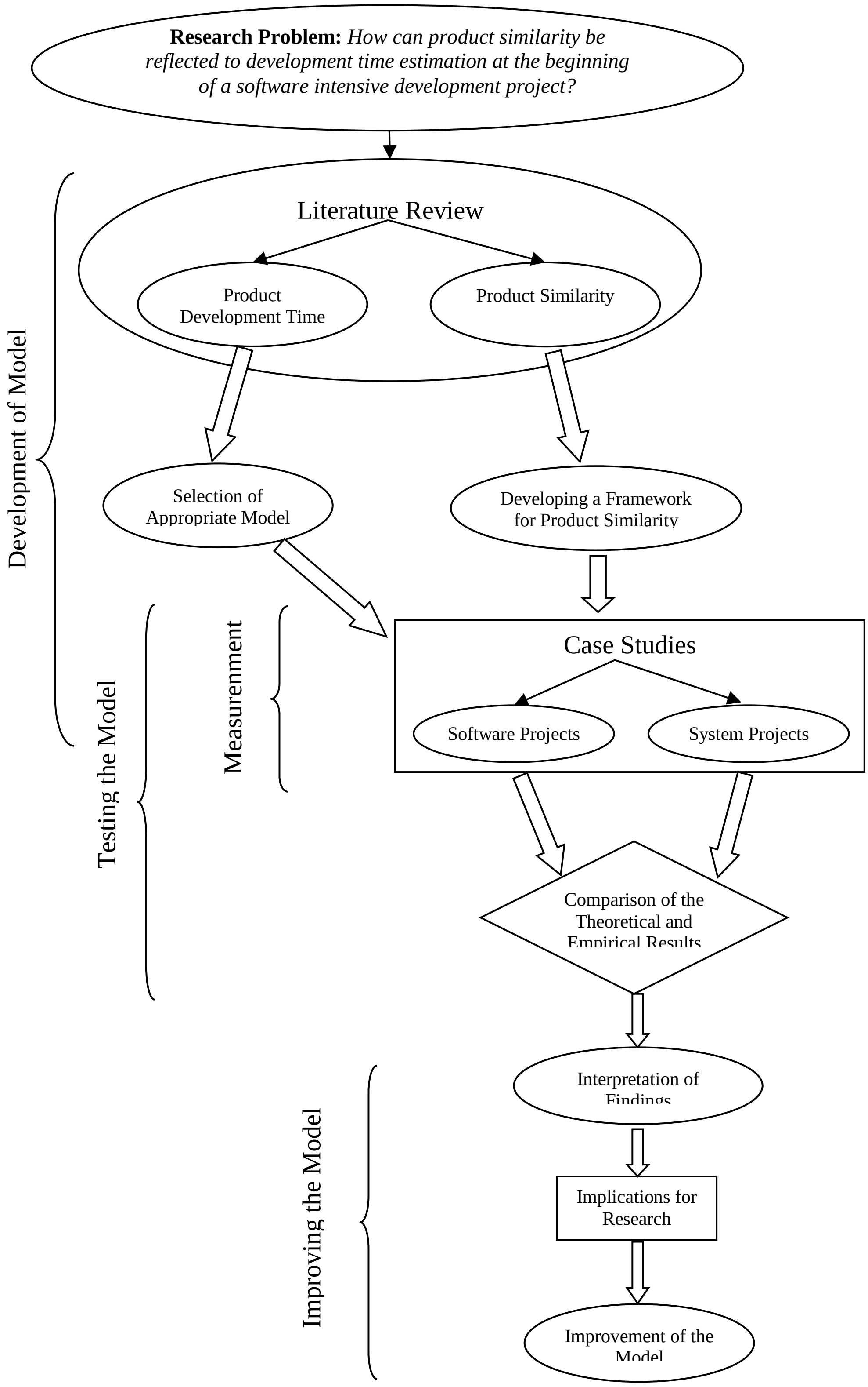

Figure 4 Research Method

Necessary data for the case studies, such as number of reused requirements, requirements engineering duration and data to derive product complexity, were gathered by joint effort of project technical managers and author of this study.

3.2 Characteristics of the Case Studies

Based on the classification in [93] the characteristics of research methodology can be:

- Exploratory: finding out what is happening, seeking new insights and generating ideas and hypotheses for new research.
- Descriptive: portraying a situation or phenomenon.
- Explanatory: seeking an explanation of a situation or a problem, mostly but not necessary in the form of a causal relationship.
- Improving: trying to improve a certain aspect of the studied phenomenon.

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. This study tries to find the answer to "how" one can estimate product development time. It is also *improving* because the results show the shortcomings of a proposed model and also show the necessity for improving the product development time estimation proposed by Griffin [9]. 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 ([93], [94]). 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 the similarity analysis of the products based on the number of similar requirements. Hence, since this similarity framework requires a count of requirements, a *quantitative* approach is employed. The results are the observed data and not open to interpretations. The author clearly knew in advance what information was needed from the participants in the case studies.

3.3 Validity of the Case Studies

The following criteria established by Gibbert et al. [95] and Cavaye [94] 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 or cross functional team is used or not. The lack of inter-dependency among those is widely accepted as visible in the works of Callahan and Moretton [5], Olson et al. [30], Herstatt et al. [96], Michalek et al. [97], Schimmoeller [98] and Bonner et al. [99]. As 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 team 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. According to the project characteristics, newness of the product could be evaluated in different manner. Newness is determined by analyzing the similarity of requirements. Different people could evaluate the similarity of the requirements and correspondingly similarity of the product. To minimize the effect of incorrect identification for product newness, for each case study it is noted that two projects were carried out by almost the same team.

- Construct validity: Data were utilized from the interviews carried out by the author. These data consist of the number of requirements and duration of the requirement definition phase. Also, the data for complexity level of the products were derived from the evaluations of the author and project technical managers. The main functions and technologies of the products were discussed with the project technical managers and complexity level of the product was defined from the 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. Theoretical studies may not provide reliable results since they are not based on actual data. Required data was collected from project technical managers. Since technical managers are responsible to all stakeholders for technical activities, they were selected as the right sources for the necessary information. Besides, for each case study two projects were selected for defining the similarity of first product to the previously developed one and for both projects it is noted that the project teams involved were almost the same. Therefore the approach for defining the newness of the product is only applicable for the projects carried by almost identical teams.

- Reliability: Each case study was conducted on the basis of an in-depth interview. Archival analysis was part of the data collection process. Except for the complexity level, number of requirements and the duration of requirements engineering activities were drawn from the organization's archives, and accordingly, it was 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 was sufficiently reliable. For the complexity level definition, an assessment was performed for all of the products.

3.4 Validity Threats

In this study, the following parameters were 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.

While comparing two products in terms of similarity, some project characteristics should not be ignored as well. Two projects might be in the same domain but the size of the projects might be different, this may lead to incorrect comparison of the projects. To overcome this issue, projects were selected for comparison only if they were comparable in effort, size and team characteristics.

3.5 Data Collection

Griffin's model requires the definition of the *Newness* variable to analyze the product development time however, she did not propose an objective method. In the present study, a framework to define the *Newness* of products according to a quantification of requirements reuse was provided. To estimate product development time, data was gathered from three different companies via the project technical managers and author of this study. Unfortunately, there were difficulties in gathering data from different companies.

Organizations generally do not keep the project related data in a systematic way. To overcome this difficulty, interviews were held with the project technical managers and the related data were collected using relevant documents and the organization's database. Interviews were fully-structured, as defined by Runeson and Höst [93], all the questions were defined before the interviews and the author determined the flow of the discussions. These questions are given in Table 12. During this period, instant messaging and telephone conversations were needed to clarify the detail of the data.

Table 12 Interview Questions for the Case Studies

Questions	**Investigation Approach**
Are there any similar products which can be in the same domain or are derivative products in the organizations?	Discussions with different project technical managers from different companies are performed and the details of the projects are evaluated.
Are there recorded data for the number of requirements for each project in the same domain?	The System/Software Requirements Specification Documents for each project are used to obtain the necessary data.
Are there recorded data for reused requirements?	If the metrics are kept systematically, data are retrieved from organization database. If they are not kept in the organization database, reused requirements are derived from the System Requirements Documents by the technical personnel involved in the projects.
Is there duration data for requirement definition phases?	The enterprise resource planning systems of the companies are used to extract this data.
What is the complexity level of product to be studied?	The main functions of the products are determined to obtain the complexity level of the product with the help of technical managers of the projects.

Data was incrementally collected. Number of requirements and the duration of requirements engineering activities were analyzed first. In the scope of that activity, product complexities were quantified considering the number of each product's main functions. This quantization of product complexity has been verified by correlating it with grey measurement results. In this context, software characteristics, technology, organization and environment determinants were evaluated. Each determinant included some indicators. For example, software characteristics were considered with two indicators as COSMIC FFP size [100] and interfaces of the products. 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. For that study, a second iteration was performed to collect the necessary data for grey measurement.

Even when 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 is limited in this study. The company data and related case study summary are given in Table 13.

Table 13 Summary of the Case Studies

Company	Sector of Products Developed	Case Study	Projects	Product Type
A	Military, Civilian	1	A1, A2	Hardware+Software
		2	A1, 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, C2	Software
		9	C1, C3	Software

The system products are consumer oriented products which include mechanical, electronics and software components for military usage. Software products for Case Study 3, 4 and 7 are military products whereas that for Case Study 8 and 9 are developed for government institutions.

All three companies are located in the same country and are private development organizations. Company A has more than 1,000 employees. It has research, development and manufacturing capabilities and its product family includes civilian and military products. Company B has over 500 employees and is similar to Company A in terms of facilities and capabilities however, it has foreign partner and mostly focuses on development projects. Company C has more than 700 employees. Its main area of interest is the development of software products for civilian and mostly government sector.

CHAPTER 4

CASE STUDY FINDINGS

The following paragraphs present the quantitative data from the three companies which are all located in the same country. The results from the analysis of the quantitative data from the three companies were compared with the results from Griffin's study [9]. 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. As of now, for each project, requirements definition and review phases are completed. That is, the data used in the case studies are the accepted number of requirements by stakeholders.

For the case studies the projects which covered requirement engineering activities cross functional teams were not used. Thus, the data only consisted of the engineering efforts and other departments such as marketing and finance were not included in the scope of the case studies.

The structure of the case studies is depicted in Figure 5.

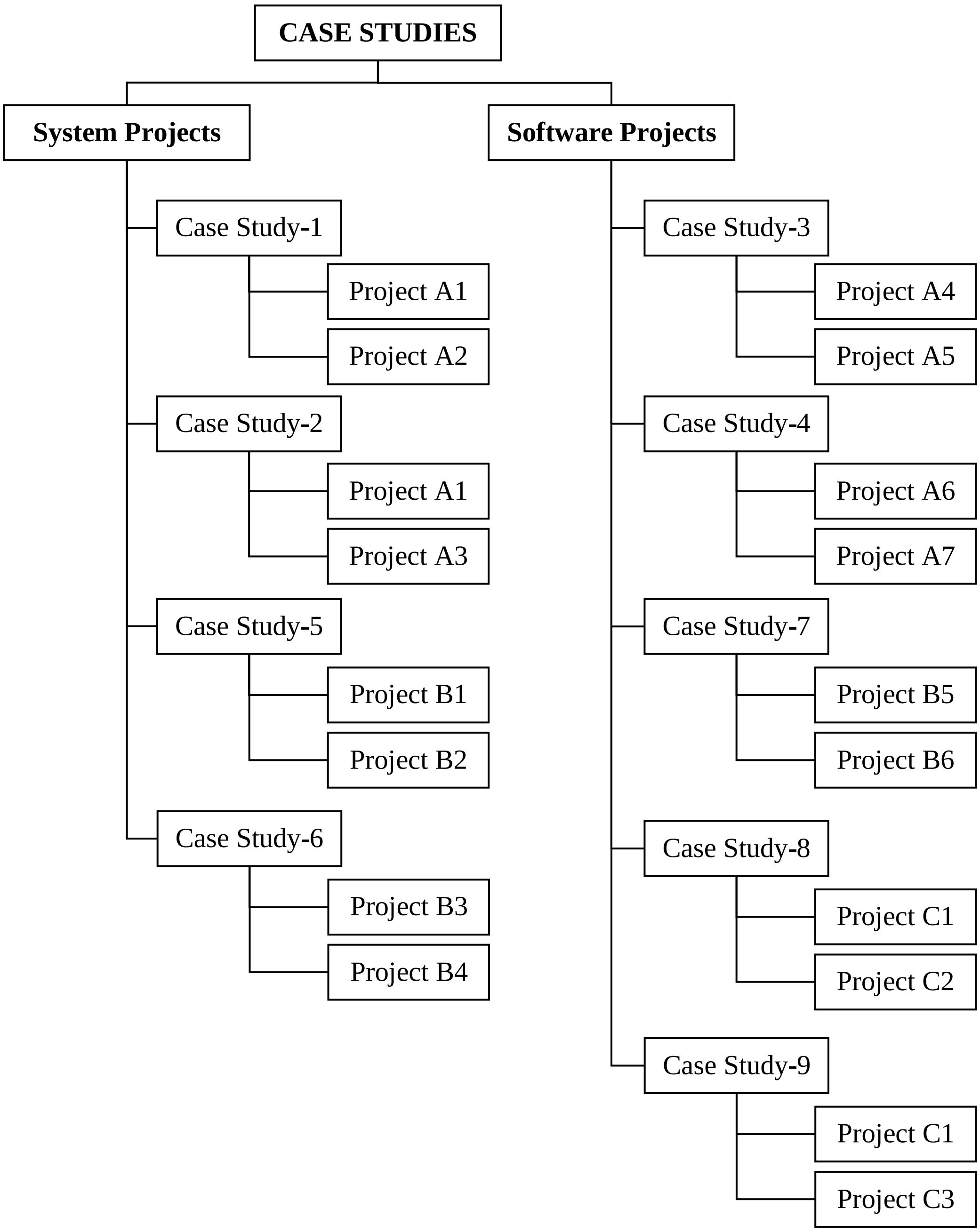

Figure 5 Structure of the Case Studies

The data from the case studies was 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 of system products and software products.
- 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 [9].
- Separately analyzing the software products and system products.
- Formulating the modification to be proposed for product development time estimation in software-intensive settings.

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 was formulated and validated for product development time estimation for software products.

4.1 CASE STUDY 1 (System Project)

4.1.1 General Description of Case Study 1

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 same new product family and including hardware and software components. Both projects were concerned with the design and development of a military communication product and included requirements analysis and definition, design, development, product integration, test and platform integration phases. Project A1 and Project A2 includes some complex algorithmic functions, user interface software and some hardware complexities due to the usage in military environment. These products are used in real time combat environment, they are not include any data storing functions except some simple parameters. Project A1 was 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.

The stakeholders of these projects were the project manager, systems engineers, software developers and hardware developers. The systems engineers classified the requirements and then reuse of these requirements was easily performed.

The systems engineering department in Company A defined the functional, electrical and interface requirements. For the mechanical/ergonomic requirements, the necessary standards were determined by mechanical engineers. For the environmental requirements, the required standards were determined and the necessary requirements were derived from these standards. For the integration requirement, a platform survey was performed and necessary requirements were defined.

4.1.2 Case Study 1 Quantitative Data (Realized)

Table 14 shows the classification and the number of the requirements in Project A2. The realized duration for requirements definition activities for both projects is given in Table 15.

Table 14 Number of Requirements Used in Project A2

Requirements Classification	**Total Number of Req. in Project A2**	**Req. of Project A1 Reused in Project A2**
Functional & Electrical & Interface Req.	115	71
Mechanical/Ergonomic Req.	29	12
Environmental Req.	28	19
Integration Req.	9	2
TOTAL	**183**	**104**

Table 15 Duration Expended in RE Works of Project A1 and Project A2

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

Table 14 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 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 Equation (5), for all possible changes in the requirements, if the requirements were not reused the organization would require an additional 16 months (β_{2CTC}*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 were reused. So, the change effect is reduced by 9.12 months for Case Study 1.

4.1.3 Case Study 1 Quantitative Data (Expected)

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 were performed using Equations (4) and (5). 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 was taken as 6 based on the number of main functions and technology the product possessed.

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

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

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

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

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

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

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

The calculated time spent for requirements engineering (RE) activities is summarized in Table 16. These durations are longer than the actual durations given in Table 15, because the estimated durations include other systems engineering activities at the beginning of the project, such as business and concept development.

Table 16 Case Study 1: Estimated Time Spent for RE Works

	Requirements Engineering Works for 100% Change	**Requirements Engineering Works for 43% Change**	**% of Decrease in Requirements Engineering Works**
CTC-DT	18 months	14.01 months	≥ 22%

4.1.4 Evaluation of Case Study 1

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 15, 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% was 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.

4.2 CASE STUDY 2 (System Project)

4.2.1 General Description of Case Study 2

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. Project A3 again includes some complex algorithmic functions, user interface software and some hardware complexities due to the usage in military environment. By the middle of 2012, the requirements of Project A3 had been defined and approved in the organization.

4.2.2 Case Study 2 Quantitative Data (Realized)

Table 17 shows the classification and the number of the requirements in Project A3. The realized duration for requirements definition activities for both projects is given in Table 18.

Table 17 Number of Requirements Used in Project A3

Requirements Classification	**Total Number of Req. in Project A3**	**Req. of Project A1 Reused in Project A3**
Functional & Electrical & Interface Req.	185	152
Mechanical/Ergonomic Req.	25	18
Environmental Req.	23	-
Integration Req.	-	-
TOTAL	**233**	**170**

Table 18 Duration Expended in RE Works of Project A1 and Project A3

	Project A1	**Project A3**	**Possible Impact of Reuse**
Total Duration (months)	8	4.5	44% decrease in duration

Table 17 shows that 73% of the requirements (170 requirements out of 233) of Project A3 were reused requirements. This implies that the change probability of 170 reused requirements was very low in this project, because they had previously been tested and approved by the same or a similar customer. Hence;

- 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.

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 27% for Project A3. Using Griffin's CTC formulation in Equation (5), for all possible changes in the requirements, if the requirements were not reused the organization would require an additional 16 months (β_{2CTC}*NN= 0.16*100%). On the other hand, the organization would only require an additional maximum of 4.32 months (0.16*27%) when all the common requirements were reused. So, the change effect is reduced by 11.68 months for Case Study 2.

4.2.3 Case Study 2 Quantitative Data (Expected)

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 27% cases were performed using Equations (4) and (5). 100% indicates that product requirements/features were totally new; 27% 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 A3 was taken as 6 based on the number of main functions and technology the product possessed.

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

$$CTC_{27} = 10.4 + 3.7*6 + 0.16*27\% + 0.1*6 = 37.52 \text{ months}$$

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

$$DT_{27} = 8.4 + 4.2*6 + 0.09*27\% - 1.9*6 = 24.63 \text{ months}$$

The time spent on requirements engineering for NN values of 100% and 27% would be;

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

$$CTC_{27} - DT_{27} = 37.52 – 24.63 = 12.89 \text{ months}$$

The calculated time spent for requirements engineering (RE) activities is summarized in Table 19. These durations are longer than the actual durations given in Table 18, because the estimated durations include other systems engineering activities at the beginning of the project, such as business and concept development.

Table 19 Case Study 2: Estimated Time Spent for RE Works

	Requirements Engineering Works for 100% Change	**Requirements Engineering Works for 27% Change**	**% of Decrease in Requirements Engineering Works**
CTC-DT	18 months	12.89 months	≥ 28%

4.2.4 Evaluation of Case Study 2

Even if a maximum change (27%) occurs in the requirements, there would be at least a 28% decrease (from 18 months to 12.89 months) in the duration of the RE activities. If the change in the requirements is less than 27%, the improvement would be expected to be greater than 28%.

When this result is compared with the actual findings of Case Study 2 in Table 18, 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% was 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)

4.3.1 General Description of Case Study 3

Another division of Company A had been using its own design and development processes and in 2011 they were certified at the CMMI Level 3 of maturity. Two software projects of this division were analyzed for this case: Projects A4 and A5. Project A4 covers the base product requirements some of which were modified according to customer requirements and for the platform requirement to be used in Project A5. Project A4 and Project A5 includes complex algorithmic functions and user interface software components. These products are used in real time combat environment, they do not include any data storing functions except some simple parameters. Important data are stored in other systems which have interfaces with Project A4 or Project A5. Project A4 was completed in 2011 and included design and development of a military product. For Project A5, the system requirements were defined and approved by the customer in 2012.

4.3.2 Case Study 3 Quantitative Data (Realized)

Table 20 shows the number of requirements in Project A5. Table 21 gives the requirements definition duration data for both projects.

Table 20 Number of Requirements Used in Project A5

Requirements Classification	**Total Number of Req. in Project A5**	**Req. of Project A4 Reused in Project A5**
Functional & Electrical & Interface Req.	334	252
Integration Req.	8	3
TOTAL	**342**	**255**

Table 21 Duration Expended in RE Works of Project A4 and Project A5

	Project A4	**Project A5**	**Possible Impact of Reuse**
Total Duration (months)	6	4.5	25% decrease in duration

As shown in Table 20, 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 change probability in the product could be decreased in the range of 0% to 25% for Project A5 by reusing the requirements. Using Equation (5), if the requirements were not reused, the organization would require an additional 16 months (0.16*100%), while the organization would only require an additional maximum of 4 months (0.16*25%) for Project A5 if the requirements were reused. Thus, the change effect would be decreased by 12 months for Project A5.

4.3.3 Case Study 3 Quantitative Data (Expected)

The calculations for CTC and DT are given for 100% and 25% changes using Equations (4) and (5). Similar to the Case Study 1, 100% indicates that all the product requirements/features are totally new and 25% is the rate of changes (i.e. NN) in the requirements. The complexity level of the product developed in the scope of Project A5 was assessed by organization staff as 5.

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

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

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

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

Time spent on RE works for 100% and 25% NN would be;

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

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

From the results of these calculations, RE durations are summarized in Table 22.

Table 22 Case Study 3: Estimated Time Spent for RE Works

	Requirements Engineering Works for 100% Change	**Requirements Engineering Works for 25% Change**	**% of Decrease in Requirements Engineering Works**
CTC-DT	16.5 months	11.25 months	≥ 32%

4.3.4 Evaluation of Case Study 3

If the maximum change (25%) occurred in the requirements for Project A5, the calculation indicates that there would be at least a 32% decrease in the duration (from 16.5 months to 11.25 months) of the RE activities. Again, if change in the requirements is lower, this rate would be expected to be greater than 32%.

These calculated results are not in agreement with the actual findings of Case Study 3 in Table 21. 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 DISCUSSION section below.

4.4 CASE STUDY 4 (Software Project)

4.4.1 General Description of Case Study 4

Two software projects were analyzed for this case: Projects A6 and A7. Project A6 covers the base product requirements some of which were modified according to customer requirements of Project A7. Project A6 and Project A7 includes user interface software components. These products do not include any data storing functions. Project A6 was completed in 2012 and included design and development of a military product. For Project A7, the system requirements were defined and approved by the customer in 2012.

4.4.2 Case Study 4 Quantitative Data (Realized)

Table 23 shows the number of requirements in Project A7. Table 24 gives the requirements definition duration data for both projects.

Table 23 Number of Requirements Used in Project A5

Requirements Classification	Total Number of Req. in Project A7	Req. of Project A6 Reused in Project A7
Functional & Electrical & Interface Req.	158	106
Integration Req.	9	-
TOTAL	**167**	**106**

Table 24 Duration Expended in RE Works of Project A6 and Project A7

	Project A6	Project A7	Possible Impact of Reuse
Total Duration (months)	4	3	25% decrease in duration

As shown in Table 23, for Project A7;

- 63% of total requirements (106 requirements out of 167) were almost fixed and their change probability was very low. This ratio is the similarity of A7 to A6.

- 37% of total requirements (61 requirements out of 167) could still be changed during the product cycle time. This ratio is the *Newness* of Project A7.

The change probability in the product could be decreased in the range of 0% to 37% for Project A7 by reusing the requirements. Using Equation (5), if the requirements were not reused, the organization would require an additional 16 months (0.16*100%), while the organization would only require an additional maximum of 5.92 months (0.16*37%) for Project A7 if the requirements were reused. Thus, the change effect would be decreased by 10.08 months for Project A7.

4.4.3 Case Study 4 Quantitative Data (Expected)

The calculations for CTC and DT are given for 100% and 37% changes using Equations (4) and (5). 100% indicates that all the product requirements/features are totally new and 37% is the rate of changes (i.e. NN) in the requirements. The complexity level of the product developed in the scope of Project A7 was assessed by organization staff as 3.

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

$$CTC_{37} = 10.4 + 3.7*3 + 0.16*37\% + 0.1*3 = 27.72 \text{ months}$$

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

$$DT_{37} = 8.4 + 4.2*3 + 0.09*37\% - 1.9*3 = 18.63 \text{ months}$$

Time spent on RE works for 100% and 37% NN would be;

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

$$CTC_{37} - DT_{37} = 27.72 – 18.63 = 9.09 \text{ months}$$

From the results of these calculations, RE durations are summarized in Table 25.

Table 25 Case Study 4: Estimated Time Spent for RE Works

	Requirements Engineering Works for 100% Change	**Requirements Engineering Works for 37% Change**	**% of Decrease in Requirements Engineering Works**
CTC-DT	13.5 months	9.09 months	≥ 33%

4.4.4 Evaluation of Case Study 4

If the maximum change (37%) occurred in the requirements for Project A7, the calculation indicates that there would be at least a 33% decrease in the duration (from 13.5 months to

9.09 months) of the RE activities. Again, if change in the requirements is lower, this rate would be expected to be greater than 33%.

These calculated results are not in agreement with the actual findings of Case Study 4 in Table 24. The decrease in Project A7 was actually 25% but Griffin's formulation predicts at least a 33% decrease in Project A7. 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 DISCUSSION section below.

4.5 CASE STUDY 5 (System Project)

4.5.1 General Description of Case Study 5

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 are related to the same product family of military communication equipment, and which include hardware and software components. Project B1 was completed in 2011. Project B1 and Project B2 includes some complex algorithmic functions, user interface softwares and some hardware complexities due to the usage in military environment. These products are used in real time combat environment, they are not include any data storing functions except some simple parameters. Project B2 was based on the product developed in the scope of Project B1. New requirements are added according to the product user and chosen platform. For Project B2 the system requirements were 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 will start at the end of 2013.

4.5.2 Case Study 5 Quantitative Data (Realized)

The number of the requirements in Project B2 is given in Table 26. The realized duration for the requirements definition activities for both projects is given in Table 27.

Table 26 Number of Requirements Used in Project B2

Requirements Classification	Total Number of Req. in Project B2	Req. of Project B1 Reused in Project B2
Functional & Electrical & Interface Req.	140	59
Mechanical/Ergonomic Req.	38	18
Environmental Req.	23	4
Integration Req.	11	10
TOTAL	**212**	**91**

Table 27 Duration Expended in RE Works of Project B1 and Project B2

	Project B1	Project B2	Possible Impact of Reuse
Total Duration (months)	5	3	40% decrease in duration

Table 26 shows that 43% of the requirements (91 requirements out of 212) of Project B2 were reused requirements. This implies that the change probability of the 91 reused requirements was very low in Project B2, since they had previously been tested and approved by the customer. To summarize:

- 43% of the total requirements (91 requirements) for Project B2 were almost fixed. This ratio is the similarity of B2 to B1.

- 57% of the total requirements (121 new requirements) could still be changed in Project B2. This ratio is the *Newness* of Project B2.

While NN varies between 0% and 100%, by reusing the requirements it can be decreased in the range of 0% to 57% for Project B2. Using Griffin's CTC formulation in Equation (5), this situation indicates that for all possible changes in the requirements, the organization would require an additional 16 months (β_{2CTC}*NN= 0.16*100%) if requirements were not reused. On the other hand, the organization would only require an additional maximum of

9.12 months (0.16*57%) when requirements were reused. So, the change effect would be reduced by 6.88 months for Case Study 5.

4.5.3 Case Study 5 Quantitative Data (Expected)

As described above, RE activities were contained within the duration of the CTC. To estimate the time spent on RE activities, the calculations for CTC and DT given below for 100% and 57% cases were performed using Equations (4) and (5). 57% 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 B2 was taken as 6 based on the number of main functions the product possessed.

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

$CTC_{57} = 10.4 + 3.7*6 + 0.16*57\% + 0.1*6 = 42.32$ months

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

$DT_{57} = 8.4 + 4.2*6 + 0.09*57\% - 1.9*6 = 27.33$ months

The time spent on RE works for 100% and 57% NN would be;

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

$CTC_{57} - DT_{57} = 42.32 – 27.33 = 14.99$ months

Subsequently, the time spent for RE activities is summarized in Table 28.

Table 28 Case Study 5: Estimated Time Spent for RE Works

	Requirements Engineering Works for 100% Change	Requirements Engineering Works for 57% Change	% of Decrease in Requirements Engineering Works
CTC-DT	18 months	14.99 months	≥ 17%

4.5.4 Evaluation of Case Study 5

In case where the maximum change (57%) occurs in the requirements, there would be at least a 17% decrease (from 18 months to 14.99 months) in the duration of RE activities. If it is less than 57%, the improvement would be expected to be greater than 17%.

When this result is compared with the actual findings of Case Study 5 as in Table 27, the decrease in Project B2 shows agreement with these calculations. Griffin's formulation predicts at least a 17% reduction in duration, in fact, a reduction of 40% (more than 17%) was obtained. Thus, similar to Case Study 1 and Case Study 2, the formulation proposed by Griffin for the estimation of project duration also applies to this case study which includes hardware and software components.

4.6 CASE STUDY 6 (System Project)

4.6.1 General Description of Case Study 6

This case study also consists of two system projects from Company B: Projects B3 and B4. They are within the same product family, including hardware and software components for a communication system. Project B3 and Project B4 includes some complex algorithmic functions, user interface softwares and some hardware complexities due to the usage in military environment. These product are used in real time combat environment, they are not include any data storing functions except some simple parameters. Project B3 was completed in 2012. Project B4 is in test phase and is expected to be completed in 2013.

4.6.2 Case Study 6 Quantitative Data (Realized)

The number of the requirements in Project B4 is presented in Table 29. The realized duration for the requirements definition activities for both projects is given in Table 30.

Table 29 Number of Requirements Used in Project B4

Requirements Classification	Total Number of Req. in Project B4	Req. of Project B3 Reused in Project B4
Functional & Electrical & Interface Req.	244	65
Mechanical/Ergonomic Req.	63	23
Environmental Req.	43	27
Integration Req.	44	31
TOTAL	**394**	**146**

Table 30 Duration Expended in RE Works of Project B3 and Project B4

	Project B3	Project B4	Possible Impact of Reuse
Total Duration (months)	5	4	20% decrease in duration

Table 29 shows that 37% of the requirements (146 requirements out of 394) of Project B4 were reused requirements. So, the change probability of 146 reused requirements was very low in Project B4. Briefly;

- 37% of total requirements (146 requirements) for Project B4 were almost fixed. This ratio is the similarity of B4 to B3.

- 63% of total requirements (248 new requirements) could still be changed in Project B4. This ratio is the *Newness* of B2.

Via requirements reuse, the change probability in the product can be decreased in the range of 0% to 63% for Project B4. Using Griffin's CTC formulation in Equation (5), for all possible changes in the requirements the organization would require an additional 16 months (β_{2CTC}*NN= 0.16*100%) if requirements were not reused. On the other hand, the organization would only require an additional maximum of 10.08 months (0.16*63%) when the requirements were reused. So, the change effect is reduced by 5.92 months for Case Study 6.

4.6.3 Case Study 6 Quantitative Data (Expected)

To estimate the time spent on the RE activities, the calculations for CTC and DT given below for 100% and 63% cases were performed using Equations (4) and (5). 57% 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 B4 was again taken as 6 based on the number of main functions.

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

$$CTC_{63} = 10.4 + 3.7*6 + 0.16*63\% + 0.1*6 = 43.28 \text{ months}$$

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

$$DT_{63} = 8.4 + 4.2*6 + 0.09*63\% - 1.9*6 = 27.87 \text{ months}$$

The time spent on RE works for 100% and 63% NN would be;

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

$$CTC_{63} - DT_{63} = 43.28 - 27.87 = 15.41 \text{ months}$$

As a result of these calculations, the time spent on RE activities is summarized in Table 31.

Table 31 Case Study 6: Estimated Time Spent for RE Works

	Requirements Engineering Works for 100% Change	**Requirements Engineering Works for 63% Change**	**% of Decrease in Requirements Engineering Works**
CTC-DT	18 months	15.41 months	≥ 14%

4.6.4 Evaluation of Case Study 6

If a maximum change (63%) occurs in the requirements, there would be at least a 14% decrease (from 18 months to 15.41 months) in the duration of the RE activities. If the change in the requirements was less than 63% change, the improvement would be expected to be greater than 14%.

When this result is compared with the actual findings of the Case Study 6 in Table 30, the decrease in Project B4 shows agreement with these calculations. Griffin’s formulation predicts at least 14% reduction in duration, in Project B4 a reduction of 20% (more than 14%) was obtained. Similar to Case Study 1, Case Study 2 and Case Study 5, the formulation proposed by Griffin for the estimation of project duration also applies to this case study which includes hardware and software components.

4.7 CASE STUDY 7 (Software Project)

4.7.1 General Description of Case Study 7

This case study involves two software projects from Company B; B5 and B6 which were in the same domain. Project B6 uses some of the requirements of Project B5. In the scope of Project B5, a commercial software product was developed. Project B6 includes the development of a similar product for military purposes. Project B5 and Project B6 includes complex algorithmic functions and user interface software components. These products are used in real time environment. Project B5 was completed in 2010. For Project B6, the software requirements were defined and approved by customer in 2012. Project B6 is expected to be completed in 2014.

4.7.2 Case Study 7 Quantitative Data (Realized)

Table 32 shows the number of requirements in Project B6. Again, for the requirements definition of both projects, the duration data are given in Table 33.

Table 32 Number of Requirements Used in Project B6

Requirements Classification	Total Number of Req. in Project B6	Req. of Project B5 Reused in Project B6
Functional & Electrical & Interface Req.	98	64
Integration Req.	36	18
TOTAL	**134**	**82**

Table 33 Duration Expended in RE Works of Project B5 and Project B6

	Project B5	Project B6	Possible Impact of Reuse
Total Duration (months)	5	4	20% decrease in duration

As shown in Table 32, for Project B6,

- 61% of total requirements (82 requirements out of 134) were almost fixed and change probability of those was very low. This ratio is the similarity of B6 to B5.

- 39% of total requirements (52 requirements out of 134) could still be changed during the product cycle time. This ratio is the *Newness* of B6.

By reusing the requirements, the changes in the product could be decreased in the range of 0% to 39% for Project B6. Using Equation (5), if requirements were not reused, the organization would require an additional 16 months (0.16*100%), while the organization would only require an additional maximum 6.24 months (0.16*39%) for Project B6 if requirements were reused. Thus, the change effect would be decreased by 9.76 months for Project B6.

4.7.3 Case Study 7 Quantitative Data (Expected)

Detailed calculations for CTC and DT are given for 100% and 39% changes using Equations (4) and (5). The rate of changes/newness in requirements is 39%. The product developed in the scope of the Project B6 had a complexity level of 7.

$CTC_{100} = 10.4 + 3.7*7 + 0.16*100\% + 0.1*7 = 53$ months

$CTC_{39} = 10.4 + 3.7*7 + 0.16*39\% + 0.1*7 = 43.24$ months

$DT_{100} = 8.4 + 4.2*7 + 0.09*100\% - 1.9*7 = 33.5$ months

$DT_{39} = 8.4 + 4.2*7 + 0.09*39\% - 1.9*7 = 28.01$ months

Time spent on RE works for 100% and 39% NN would be;

$CTC_{100} - DT_{100} = 53 – 33.5 = 19.5$ months

$CTC_{39} - DT_{39} = 43.24 – 28.01 = 15.23$ months

Using the results of these calculations, the duration for RE works is summarized in Table 34.

Table 34 Case Study 7: Estimated Time Spent for RE Works

	Requirements Engineering Works for 100% Change	**Requirements Engineering Works for 39% Change**	**% of Decrease in Requirements Engineering Works**
CTC-DT	19.5 months	15.23 months	≥ 22%

4.7.4 Evaluation of Case Study 7

Even if the maximum change (39%) occurred in the requirements for Project B6, the calculation indicates that there would be at least a 22% decrease in the duration (from 19.5 months to 15.23 months) of the RE activities. If change in the requirements is less, this rate would be expected to be greater than 22%.

These calculated results are not in agreement with the actual findings of Case Study 7 in Table 33. 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)

4.8.1 General Description of Case Study 8

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 from Company C were analyzed for this case study both projects were for government institutions. Project C1 and Project C2 are MIS projects and include user interface software and data storage functions. Project C1 began development in 2009. This project was the baseline for Project C2 and new customer requirements were added. Project C2 started at the beginning of 2012 and delivery is planned for the end of 2013.

4.8.2 Case Study 8 Quantitative Data (Realized)

Table 35 shows the number of requirements of Project C2. For the requirements definition activities of this project, the duration data are given in Table 36.

Table 35 Number of Requirements Used in Project C2

Requirements Classification	Total Number of Req. in Project C2	Req. of Project C1 Reused in Project C2
Functional & Electrical & Interface Req.	372	314
Integration Req.	4	-
TOTAL	**376**	**314**

Table 36 Duration Expended in RE Works of Project C1 and Project C2

	Project C1	Project C2	Possible Impact of Reuse
Total Duration (months)	7,5	5	34% decrease in duration

According to Table 35, for Project C2:

- 84% of total requirements (314 requirements out of 376) were almost fixed and the change probability of those was very low. This ratio is the similarity of C2 to C1.
- 16% of total requirements (62 requirements out of 376) could still be changed during the product cycle time. This ratio is the *Newness* of C2.

When the requirements are reused, the changes in product could be decreased in the range of 0% to 16% for Project C2. Using Equation (5), if the requirements were not reused, the organization would require an additional 16 months (0.16*100%). On the other hand, the organization would only require an additional maximum of 2.56 months (0.16*16%) for Project C2 if requirements were reused. Thus, the change effect would be decreased by 13.44 months.

4.8.3 Case Study 8 Quantitative Data (Expected)

Detailed calculations of CTC and DT are given below for possible changes of 100% and 16%. 16% indicates the requirements which were new and could be changed for Project C2. Again, the complexity level of the product developed in Project C2 was taken as 3 based on the number of functions in the software.

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

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

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

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

The time spent on RE works for 100% and 16% NN would be;

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

CTC_{16} - DT_{16} = 24.36 – 16.74 = 7.62 months

Table 37 gives summary of the time spent on RE works according to the results of these calculations.

Table 37 Case Study 8: Estimated Time Spent for RE Works

	Requirements Engineering Works for 100% Change	**Requirements Engineering Works for 16% Change**	**% of Decrease in Requirements Engineering Works**
CTC-DT	13.5 months	7.62 months	≥ 44 %

4.8.4 Evaluation of Case Study 8

Even if the maximum change (16%) occurred in the requirements for Project C2, calculations indicate that there would be at least a 44% decrease in the duration (from 13.5 months to 7.62 months) of the RE activities. In situations where there were fewer changes in the requirements, this rate would be expected to be greater.

However, the calculated result is not in agreement with the actual findings of Case Study 8 as shown in Table 36. The decrease in Project C2 was 34% in real life but Griffin's formulation predicts at least 44% decreases in Project C2.

4.9 CASE STUDY 9 (Software Project)

4.9.1 General Description of Case Study 9

A second case study from Company C was undertaken. In this case Project C3 used Project C1 which was defined in Case Study 8 as a baseline and new customer requirements were added to Project C3. This software product in the scope of Project C3 was also developed to be used by a government institution. Similar to previous case, Project C3 is MIS project and include user interface software and data storage functions. Project C3 started at the beginning of 2012 and the delivery is planned for the beginning of 2014.

4.9.2 Case Study 9 Quantitative Data (Realized)

Table 38 shows the number of requirements in Project C3. For the requirements definition activities of this project, the duration data are given in Table 39.

Table 38 Number of Requirements Used in Project C3

Requirements Classification	**Total Number of Req in Project C3**	**Req. of Project C1 Reused in Project C3**
Functional & Electrical & Interface Req.	311	230
Integration Req.	12	-
TOTAL	**323**	**230**

Table 39 Duration Expended in RE Works of Project C1 and Project C3

	Project C1	**Project C3**	**Possible Impact of Reuse**
Total Duration (months)	7,5	5	34% decrease in duration

As shown in Table 38;

- 71% of total requirements (230 requirements out of 323) were almost fixed and the change probability of those was very low. This ratio is the similarity of C3 to C1.

- 29% of total requirements (93 requirements out of 323) could still be changed during the product cycle time. This ratio is the *Newness* of C3.

By reusing the requirements, the changes in product could be decreased in the range of 0% to 29% for Project C3. Using Equation (5), if the requirements were not reused, the organization would require an additional 16 months (0.16*100%). On the other hand, the organization would only require an additional maximum of 4.64 months (0.16*29%) for Project C3 if requirements were reused. Thus, the change effect would be decreased by 11.36 months.

4.9.3 Case Study 9 Quantitative Data (Expected)

Detailed calculations of CTC and DT are given below for possible changes of 100% and 29%. Again, 29% indicates those requirements which were new and could be changed for Project C3. The complexity level of the product developed in Project C3 was taken as 3 based on the number of functions in the software.

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

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

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

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

The time spent on RE works for 100% and 29% NN would be;

CTC_{100} - DT_{100} = 37.8 – 24.3 = 13.5 months

CTC_{29} - DT_{29} = 26.44 – 17.91 = 8.53 months

The result of these calculations for the time spent on RE works is summarized in Table 40.

Table 40 Case Study 9: Estimated Time Spent for RE Works

	Requirements Engineering Works for 100% Change	**Requirements Engineering Works for 29% Change**	**% of Decrease in Requirements Engineering Works**
CTC-DT	13.5 months	8.53 months	≥ 37 %

4.9.4 Evaluation of Case Study 9

If the maximum change (29%) occurred in the requirements for Project C3, the calculations indicate that there would be at least a 37% decrease in the duration (from 13.5 months to

8.53 months) of the RE activities. In situations where there were fewer changes in the requirements, this rate would be expected to be greater.

However, the calculated result is not in agreement with the actual findings of Case Study 9 as shown in Table 39. The decrease in Project C3 was 34% in real life but Griffin's formulation predicts at least a 37% decrease in Project C3.

4.10 Summary of the Case Studies

A summary of all the case study results are presented in Table 41.

Table 41 Expected and Actual Changes in Duration of Requirements Engineering Activities for Projects A2, A3, A5, A7, B2, B4, B6, C2, C3 Using Griffin's Formulation

CS	Project	Product Type	Max. Expected % of Change in Req. (Newness)	Expected % of Duration Decrease in Requirements Engineering Works	Actual % of Duration Decrease in Requirements Engineering Works	Compatibility to Griffin's Formulation
1	A2	Hardware+Software	43%	≥ 22%	37%	Compliant
2	A3	Hardware+Software	27%	≥ 28%	44%	Compliant
3	A5	Software	25%	≥ 32%	25%	Not Compliant
4	A7	Software	37%	≥ 33%	25%	Not Compliant
5	B2	Hardware+Software	57%	≥ 17%	40%	Compliant
6	B4	Hardware+Software	63%	≥ 14%	20%	Compliant
7	B6	Software	39%	≥ 26%	20%	Not Compliant
8	C2	Software	16%	≥ 44%	34%	Not Compliant
9	C3	Software	29%	≥ 37%	34%	Not Compliant

CHAPTER 5

DISCUSSION

According to the findings of Case Studies 1, 2, 5 and 6, Griffin's formulation for product development time is validated for system projects which include hardware and software components. The results of her formulation are in line with the actual results of these cases. This demonstrates that Griffin's formulation [9] can be used for estimating development time of products 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, a modification to Griffin's formulation for software projects is proposed and it is shown that in its modified form, it can be used to accurately estimate the product development time.

Software requirements can be changed more easily than hardware requirements. The nature of software allows the customer to feel more comfortable while requesting changes. Since software changes generally do not affect the hardware, a change request can be met by amendments or modifications to software. However, hardware changes can have a greater impact on the project. Thus, the changes in software projects were more than expected. There were some decreases in project durations for Case Studies 3, 4, 7, 8 and 9, but these were less than expected according to Griffin's formulation.

Changes in product features and correspondingly in requirements, are denoted by the NN variable (*Newness/uncertainty*) in Equations (4) and (5). NN variable in Equations (4) and (5) must be re-evaluated for software projects. This parameter should have a more significant effect on the software product development time. 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 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 42.

Table 42 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	$1.2 \leq \delta \leq 2.5$
C2	$2.1 \leq \delta \leq 6.2$
C3	$1.2 \leq \delta \leq 3.4$

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 ([166], [167]) was used to reach a confidence interval for δ value. The bootstrap procedure takes the original data set and resamples it to form new data groups with the same size as the original data set. For each sample, the mean is computed. This process is repeated for a large number of times and a distribution of bootstrap mean is obtained. To reach the upper and lower bounds of bootstrap, an error margin is added/subtracted to/from the mean. Margin of errors is computed by a function of quantiles, standard deviation and number of samples in the data set.

For this purpose, minimum and maximum δ values were resampled 10.000 times by using XLSTAT tool [101]. The results are given in Table 43.

Table 43 Bootstrap resampling of δ value

Parameter	Lower Bound (Standard Bootstrap Interval)	Upper Bound (Standard Bootstrap Interval)
δ minimum	1.258	2.102
δ maximum	2.107	5.413

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 to use 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 δ value was selected as 2.102.

The proposed modified versions of Equations (4) and (5) are presented below. The duration estimations include the engineering efforts during the requirements 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.

$$\mathbf{DT = \alpha + \beta_{1DT}*PC + \beta_{2DT}* \delta*NN + \beta_{3DT}*(PC*FP) + \epsilon_{DT}} \quad \textbf{(Equation 16)}$$

$$\mathbf{CTC = \alpha + \beta_{1CTC}*PC + \beta_{2CTC}*\delta*NN + \beta_{3CTC}*(PC*FP) + \epsilon_{CTC}} \quad \textbf{(Equation 17)}$$

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

This proposal is not applicable for the case where the product is totally new. This case raises the issue of how to change the δ multiplier based on NN values. In addition to the NN values, there are some other determinants such as projects sizes, applicability of this model to different projects which are carried out at different times, the size of the hardware and software components in the projects. The functional relation of δ multiplier to those determinants on the project duration has been kept outside the scope of the present study, and might be investigated in the scope of future studies.

The revised calculation for the software project cases is repeated in the following subsections using the proposed formulation.

5.1 Case Study 3 According to the Proposed Formulation

The calculations for Case Study 2 (Project A5) of Company A are performed below using Equations (16) and (17). Since the evaluations regarding the results of the case studies do not cover totally new products, Equations (4) and (5) are used for NN=100%.

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

CTC_{25} = 10.4 + 3.7*5 + **2.102***0.16*25% + 0.1*5 = 37.8 months

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

DT_{25} = 8.4 + 4.2*5 + **2.102***0.09*25% - 1.9*5 = 24.63 months

Time spent on RE works is calculated as;

CTC_{100} - DT_{100} = 45.4 – 28.9 = 16.5 months

CTC_{25} - DT_{25} = 37.8 – 24.63 = 13.17 months

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 was 25% as given in Table 21 and is in agreement with the calculated result.

5.2 Case Study 4 According to the Proposed Formulation

The calculations for Case Study 4 (Project A7) of Company A are performed below using Equations (16) and (17). Since the evaluations regarding the results of the case studies do not cover totally new products, Equations (4) and (5) are used for NN=100%.

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

$CTC_{37} = 10.4 + 3.7*3 + \mathbf{2.102*}0.16*37\% + 0.1*3 = 34.24$ months

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

$DT_{37} = 8.4 + 4.2*3 + \mathbf{2.102*}0.09*37\% - 1.9*3 = 22.3$ months

Time spent on RE works is calculated as;

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

$CTC_{37} - DT_{37} = 34.24 – 22.3 = 11.94$ months

If the maximum change (37%) occurred in the requirements for Project A7, the calculation indicates that there would be at least a 12% decrease in the duration (from 13.5 months to 11.94 months) of the RE activities. The actual reduction in the duration was 25% as given in Table 24 and is in agreement with the calculated result.

5.3 Case Study 7 According to the Proposed Formulation

The calculations for Case Study 7 (Project B6) of Company B are given below using Equations (16) and (17). Again, Equations (4) and (5) are used without any modification for the case when the product is totally new (NN=100%).

$CTC_{100} = 10.4 + 3.7*7 + 0.16*100\% + 0.1*7 = 53$ months

CTC_{39} = 10.4 + 3.7*7 + **2.102***0.16*39% + 0.1*7 = 50.12 months

DT_{100} = 8.4 + 4.2*7 + 0.09*100% **-** 1.9*7 = 33.5 months

DT_{39} = 8.4 + 4.2*7 + **2.102***0.09*39% **-** 1.9*7 = 31.88 months

Time spent on RE works is calculated as;

CTC_{100} **-** DT_{100} = 53 – 33.5 = 19.5 months

CTC_{39} **-** DT_{39} = 50.12 – 31.88 = 18.24 months

In case where the maximum change (39%) occurred in the requirements for Project B6, the calculation indicates that there would be at least 7% decrease in the duration (from 19.5 months to 18.24 months) of the RE activities. The reduction in the real-life duration reduction was 20%, as given in Table 33, is in agreement with this result.

5.4 Case Study 8 According to the Proposed Formulation

The calculations for Case Study 8 (Project C2) of Company C are repeated below using Equations (16) and (17). Equations (4) and (5) are used without any modification for the case when the product is totally new (NN=100%).

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

CTC_{16} = 10.4 + 3.7*3 + **2.102***0.16*16% + 0.1*3 = 27.18 months

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

DT_{16} = 8.4 + 4.2*3 + **2.102***0.09*16% **-** 1.9*3 = 18.33 months

Time spent on RE works is calculated as;

CTC_{100} **-** DT_{100} = 37.8 – 24.3 = 13.5 months

CTC_{16} **-** DT_{16} = 27.18 – 18.33 = 8.85 months

If the maximum change (16%) occurred in the requirements for Project C2, 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. The 34% reduction in the real-life duration as given in Table 36, is in agreement with this result.

5.5 Case Study 9 According to the Proposed Formulation

The calculations for Case Study 9 (Project C3) of Company C are repeated below again using Equations (16) and (17).

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

CTC_{29} = 10.4 + 3.7*3 + **2.102***0.16*29% + 0.1*3 = 31.55 months

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

DT_{29} = 8.4 + 4.2*3 + **2.102***0.09*29% - 1.9*3 = 20.79 months

Time spent on RE works is calculated as;

CTC_{100} - DT_{100} = 37.8 – 24.3 = 13.5 months

CTC_{29} - DT_{29} = 31.55 – 20.79 = 10.76 months

Similarly, if the maximum change (29%) occurred in the requirements for Project C3, 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. If changes are less than 29%, this rate will be more than 20%. Consequently, the real-life reduction of the duration of 34% as given in Table 39 is in agreement with this result.

The summarized results of Case Studies 3, 4, 7, 8 and 9 using proposed formulations are given in Table 44.

Table 44 Expected and Actual Changes in Duration of Requirements Engineering Activities for Projects A5, A7, B6, C2 and C3 Using the Proposed Formulation

Case Study	Project	Max. Expected % of Change in Req. (Newness)	Expected % of Duration Decrease in Requirements Engineering Works	Actual % of Duration Decrease in Requirements Engineering Works	Compatibility to Modified Griffin's Formulation (Proposed Formulation)
3	A5	25%	≥ 20%	25%	Compliant
4	A7	37%	≥ 12%	25%	Compliant
7	B6	39%	≥ 7%	20%	Compliant
8	C2	16%	≥ 34%	34%	Compliant
9	C3	29%	≥ 20%	34%	Compliant

At the beginning of this study, the research problem was stated as "*How can product similarity be reflected to development time estimation at the beginning of a software intensive development project?"* By the finding of the case studies, an accurate model was proposed to estimate the product development time. Figure 6 outlines the proposed process of estimating the project development time that reflects the knowledge gathered from the reported case studies. The determinants of development time are usage of *formal development process*, *newness of product* and *complexity of product*. If formal process is used in the organization, it is set to 1 in the formulation, if not it is set to 0. *Newness* of the product is defined by using the framework proposed in this study. Requirements specification document should be created to complete this activity. Same requirements from the previous project are defined and then similarity of the product is specified by using the reuse rate (Equation (6)). This will bring the *Newness* of the product (Equation (7)). Finally, the complexity of product is defined by counting the main functions of product.

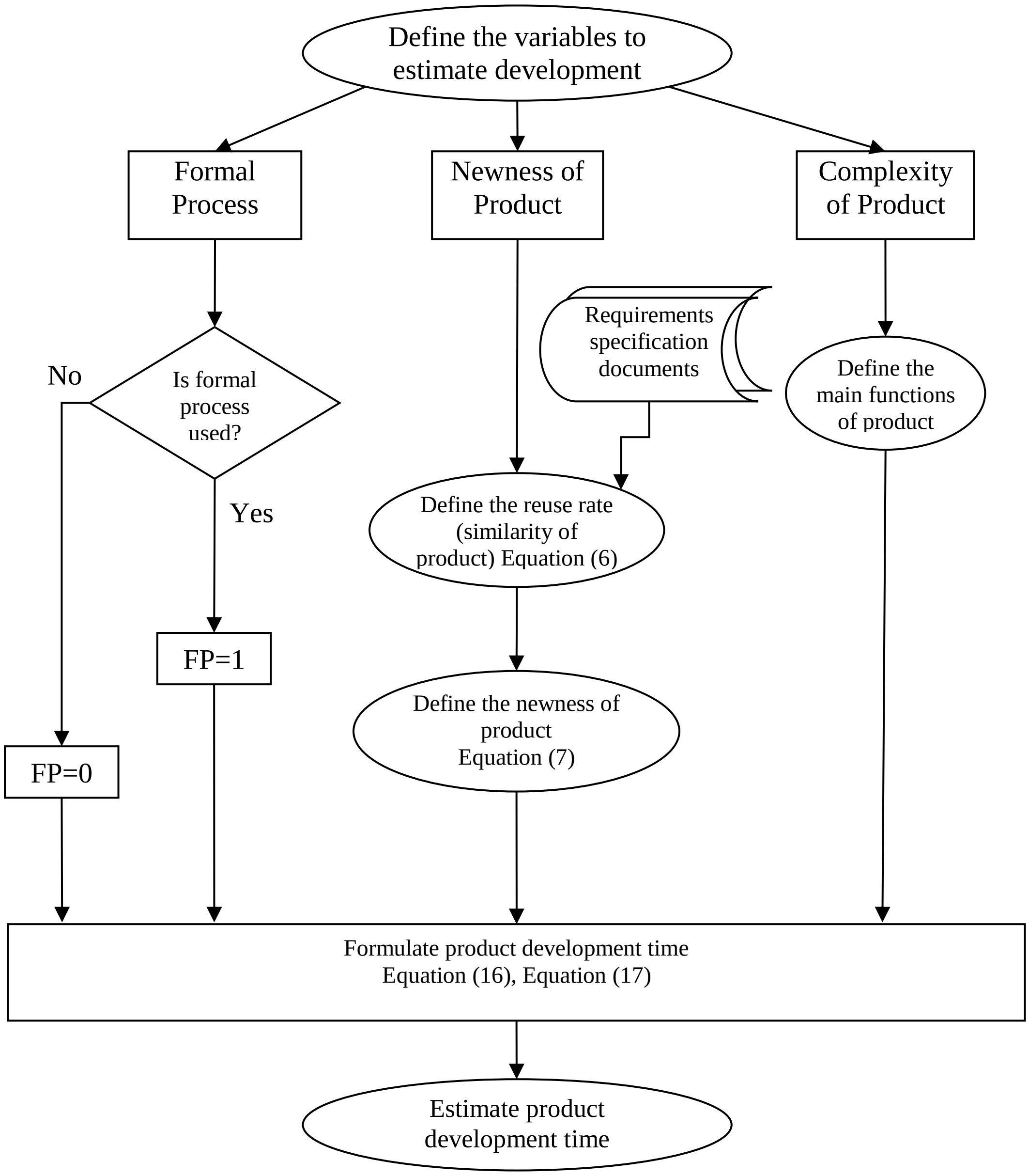

Figure 6 Product Development Time Estimation Process

CHAPTER 6

CONCLUSION

This study has shown that a modified product development time estimation model can be applied in software intensive systems development projects. A model in the literature [9] was examined and a modification has been proposed on the studied model for software products. 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, a modification to Griffin's formulation for software products was proposed.

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 all company personnel could access, systems engineers would choose to use these requirements in different projects. In the context of such an opportunity, engineers would be part of a common global view of requirements. Moreover, from another perspective the product would be developed within a common understanding of the requirements. In this study we quantify the similarity of different products in same domain according to the number of reused requirements in the products. After calculating the similarity of product, *Newness* is then derived from the similarity figures used in product development time estimations. For this investigation nine cases were studied and their results have been compared with a theoretical study. For this purpose the method suggested by Griffin [9] was used. Griffin’s formulations are derived from a large number of data sets for different companies. She collated data from 343 projects from 21 divisions of 11 companies in five industries.

Two projects were compared in Case Study 1. The requirements of the first project were created from scratch. For the second project, some of the requirements of the first project were reused. For the projects of Case Study 1, there were hardware and software components to be developed. The projects in Case Studies 2, 5 and 6 had a similar scope of products. They included the development of hardware and software components. By comparing the actual results of these case studies with the results of theoretical formulation proposed by Griffin [9], it is concluded that the proposed method by Griffin [9] can be applied to system projects which include hardware and software components.

For purely software projects five additional cases were studied as Case Study 3, Case Study 4, Case Study 7, Case Study 8 and Case Study 9. In Case Study 3, there were two software projects where the second project used some of the requirements from the first project. This was same for the Case Study 4 and Case Study 7. In Case Study 8 and Case Study 9, there were three projects, which are C1, C2 and C3, with the first project C1 being the baseline for the other two projects, C2 and C3. Some of the software requirements of the first project were used in the other two projects. According to the results of these software case studies, Griffin’s formulation does not yield the same results with real-life observations. This is because software requirements may change more easily when compared to hardware requirements, as the nature of the software allows the customer to feel more comfortable while requesting changes. Thus, there was some decrease in duration in the studied projects, but this was less than expected as foreseen in Griffin's formulation. Thus, it was necessary 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 (selected as 2.102), in which case comparable results to actual outcomes were obtained.

6.1 Contributions of the Study

The main contributions of the present study have been:

- The assessment of the application of a selected model on product development time to software projects and hardware and software systems projects in three different organizations by conducting industrial case studies.

- Nine industrial case studies present the requirements reuse approach to define product similarity in the same domain. Data from different development organizations are provided. Product similarity is used as an input to product development time estimations.

- 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.

- Griffin defines complexity directly in terms of the main functions of products. This study as presented in [165], has verified this approach by measuring the grey complexity of five software products and justifying the results with the main functions of the software products.

- Verification of Griffin's model for system projects is performed by using industrial case studies from different organizations.

- An extension to Griffin's project duration estimation model is proposed for software projects. This involves the use of data from previous projects. This data is incorporated into a mathematical model that facilitates a way of easily and more accurately estimating the project duration at earlier project stages.

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 had different customers. The proposed model can be applied for projects in similar domain. Projects should have similar requirements. According to the results of Chernak study [34], average reported reuse rate was 45 % and higher reuse rate in the range of 80-100 % is achievable in practice. Therefore, with such a high reuse rate in practice, we believe that this study will have some contributions for such projects.

6.2 Limitations

This study covers the requirements definition phase of the 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 does not analyze 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 the projects in different domains was not within the scope of this research. The modification proposed above is only considered to apply to projects within the same domain.

For each case study, two projects were used to find the similarity of one product to the previous one. This product similarity is based on the similarity of requirements. In some cases, requirements might be identical but the design decision based on the requirements might be different. Therefore, when selecting the projects it is noted that they have the same design methodology, same infrastructure and same design decisions, such as use of the programming languages and use of operating systems. The proposed model might not be applicable for projects which have such different characteristics.

Another limitation is that this study does not evaluate the complexity of requirements. All the requirements are considered to affect the complexity by the same degree irrespective of their nature.

6.3 Future Work

This study has covered only the requirements analysis phase. 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, beside requirements engineering. Similar studies may address design, implementation and testing phases explicitly.

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. The effect of some determinants such as projects' sizes and efforts, the specific time when projects are realized (as this might, especially for large time differences, may significantly influence technological infrastructures and working approaches), the size of the hardware and software components in the projects, are not included in the scope of this study. The functional definition of δ based on the different determinants might be studied as future work.

REFERENCES

[1] Michael D. Johnson & Randolph E. Kirchain. (February 2011). The Importance of Product Development Cycle Time and Cost in the Development of Product Families. *Journal of Engineering Design, Vol. 22. Number 2*, p. 87-112.

[2] Hamdi A. Bashir & Vince Thomson. (1999, May). Metrics for Design Projects: A Review. *Journal of Design Studies, Vol. 20, Issue 3*, p. 263-277.

[3] Jose L. Cuadrado-Garcia & Juan J. Cuadrado-Gallego (2011). Improve Tracking in the Software Development Projects. *Joint Conference of the 21st International Workshop on Software Measurement and the 6th International Conference on Software Process and Product Measurement,* p. 215-220.

[4] John Carter. (1996). Reducing Cycle Time by Optimizing Your Product Portfolio. *Western Electronic Show and Convention (WesCon)*. p.8-11

[5] John Callahan & Brian Moretton. (2001). Reducing Software Product Development Time. *Journal of Project Management, Vol. 9, Issue 1.* p. 59-70

[6] Roberto Filippini, Luigi Salmaso & Paolo Tessarolo. (2004, May). Product Development Time Performance: Investigating the Effect of Interactions between Drivers. *Journal of Product Innovation Management, Vol. 21, Issue 3,* p. 199-214.

[7] William H. A. Johnson, Zeljana Piccolotto & Roberto Filippini. (2009, May). The Impacts of Time Performance and Market Knowledge Competence on New Product

Success: An International Study. *IEEE Transactions on Engineering Management, Vol. 56, No. 2,* p. 219-228.

[8] Reda M. Lebcir & Jyoti Choudrie. (2011). Impact of Project Complexity Factors on Project Cycle Time: A System Dynamics Modelling Approach. *2nd International Conference on Construction and Project Management*. p. 166-170.

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

[10] Harold A. Linstone & Murray Turoff. (2002). *The Delphi Method: Techniques and Applications*. Addison-Wesley Pub. Co. ISBN 0-201-04294-0.

[11] Barry W. Boehm. (1981). *Software Engineering Economics*. Prentice Hall. ISBN 978-0138221225.

[12] Andrew Stellman & Jennifer Greene. (2005). *Applied Software Project Management*. Sebastopol, MA O'Reilly Media. ISBN 0-596-00948-8.

[13] James Surowiecki. (2005). *The Wisdom of Crowds*. Anchor Books. ISBN 0-385-72170-6.

[14] James Grenning. (2002, April). *Planning Poker*. Retrieved January 12, 2013, from

http://renaissancesoftware.net/files/articles/PlanningPoker-v1.1.pdf

[15] Watts Humphrey. (1995). *A Discipline for Software Engineering*. Addison Wesley. ISBN 978-0201546101.

[16] RAND Project. Retrieved January 13, 2013 from

http://www.rand.org/topics/delphi-method.html.

[17] Hong-Bae Jun, Hyun-Soo Ahn & Hyo-Won Suh. (2005, August). On Identifying and Estimating the Cycle Time of Product Development Process. *IEEE Transactions on Engineering Management, Vol. 52, No. 3.*

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

[19] Abbie Griffin. (2002, July). Product Development Cycle Time for Business-to-Business Products. *Industrial Marketing Management, Vol. 31, Issue 4*.

[20] Abbie Griffin. (1997, November). PDMA Research on New Product Development Practices: Updating Trends and Benchmarking Best Practices. *Journal of Product Innovation Management, Vol. 14, Issue 6*. p. 429-458.

[21] Andy J. Nolan, Silvia Abrahão, Paul C. Clements & Andy Pickard. (2011). Requirements Uncertainty in a Software Product Line. *15th International Software Product Line Conference*.

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

[23] Susan Ferreira, James Collofello, Dan Shunk & Gerald Mackulak. (2009, October). 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*.

[24] Mohamed Reda Lebcir. (2011). Impact of Project Complexity Factors on New Product Development Cycle Time. *University of Hertfordshire Business School Working Paper*. Retrieved May 10, 2012, from

https://uhra.herts.ac.uk/dspace/bitstream/2299/6040/1/S124.pdf

[25] Erik W. Larson & David H. Gobeli. (1989. May). Significance of Project Management Structure on Development Success. *IEEE Transactions on Engineering Management, Vol. 36, No. 2*, p. 119–125

[26] Philipp A. Murmann. (1994, June). Expected Development Time Reductions in the German Mechanical Engineering Industry. *Journal of Product Innovation Management, Vol. 11, Issue. 3*, p. 236–252.

[27] Marc H. Meyer & James M. Utterback (1995, November). Product Development Cycle Time and Commercial Success. *IEEE Transactions on Engineering Management, Vol. 42, Issue. 4*, p. 297–304.

[28] Sharon Novak & Steven D. Eppinger (2001, January). Sourcing by Design: Product Complexity and the Supply Chain. *Journal of Management Science, Vol. 47, No. 1*, p. 189–204.

[29] Gloria Barczak, Abbie Griffin & Kenneth B. Kahn. (2009, January). Perspective: Trends and Drivers of Success in NPD Practices: Results of the 2003 PDMA Best Practices Study. *Journal of Product Innovation Management, Vol. 26, Issue 1*.

[30] Eric M. Olson, Orville C. Walker, Robert W. Ruekert & Joseph M. Bonner. (2001, July). 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.* p.258-271

[31] Kevin Dooley, Anand Subra & John Anderson. (2002, March). Best Practices in New Product Development: Adoption Rates, Adoption Patterns, and Impact. *Journal of Innovation Management, Vol. 6, Issue 1.*

[32] Citations of Griffin's research "The Effect of Project and Process Characteristics on Product Development Cycle Time". Retrieved October 7, 2012, from

http://libra.msra.cn/Publication/2409003/the-effect-of-project-and-process-characteristics-on-product-development-cycle-time

[33] Wayne C. Lim. (1994). Effects of Reuse on Quality, Productivity, and Economics. *IEEE Software*. *Vol. 11, Issue 5.* p 23-30

[34] Yuri Chernak. (2012). Requirements Reuse: The State of the Practice. *IEEE International Conference on Software Science, Technology and Engineering*. p. 46-53.

[35] W. Lam, T.A. McDermid & A.J. Vickers. (1997, January). Ten Steps Towards Systematic Requirements Reuse. *3rd IEEE International Symposium on Requirements Engineering*. p. 6-15.

[36] Shaojie Guo, Weiqin Tong, Juan Zhang, Zongheng Li. (2011). An Application of Ontology to Test Case Reuse. *International Conference on Mechatronic Science, Electric Engineering and Computer*.

[37] Oscar Lopez Villegas & Miguel Angel Laguna. (2001). Requirements Reuse for Software Development. *5th IEEE International Symposium on Requirements Engineering*.

[38] Gerald Kotonya & Ian Sommerville. (1998). *Requirements Engineering Processes and Techniques.* John Wiley & Sons.

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

[40] Oscar Dieste, Natalia Juristo & Forrest Shull. (2008). Understanding the Customer: What Do We Know About Requirements Elicitation. *IEEE Software, Vol. 25, Issue 2,* pp.11-13, 2008.

[41] Leah Goldin & Michal Matalon-Beck. (2010). Reuse of Requirements Reduces Time to Market. *IEEE International Conference on Software Science, Technology & Engineering*.

[42] Xiaodong Zhao. (2009). Product Similarity. *Journal of Operations Management, Vol. 7, No. 1*. p. 58-64.

[43] Kamalini Ramdas, Oleksandr Zhylyevskyy & William L. Moore. (2010, November). A Methodology to Support Product Differentiation Decisions. *IEEE Transactions on Engineering Management, Vol. 57, No. 4*. p. 649-660.

[44] Daniel P. Politze & S. Dierssen. (2009, March). A Criteria Based Measure of Similarity between Product Functionalities. *19th CIRP Design Conference – Competitive Design*. p. 99.

[45] Joo Chang Lee & Heejung Lee. (2012). An Ontological Approach to Measure Similarity Between Different Product Information Model. *Proceedings of the Asia Pacific Industrial Engineering & Management Systems Conference*, p. 1575-1580.

[46] Reda M. Lebcir. (2006). A Framework for Project Complexity in New Product Development (NPD) Projects. Business School Working Papers, Vol. UHBS 2006-1, University of Hertfordshire.

[47] Muhammad Ilyas & Josef Küng. (2009). A Similarity Measurement Framework for Requirements Engineering. *4th International Multi-Conference on Computing in the Global Information Technology*. p. 31-34.

[48] Brian Berenbach, Daniel Paulish, Juergen Kazmeier & Arnold Rudorfer. (2009). *Software & Systems Requirements Engineering: In Practice.* McGraw-Hill Osborne Media.

[49] Steve McConnell. (2004). *Code Complete* (2^{nd} ed.). Microsoft Press. ISBN 0-7356-1967-0, 2004.

[50] Roger J. Calantone & C. Anthony Di Benedetto. (2000, May). Performance and Time to Market: Accelerating Cycle Time with Overlapping Stages. *IEEE Transactions on Engineering Management, Vol. 47, No. 2*.

[51] B. J. Zirger & Janet L. Hartley. (1996, May). The Effect of Acceleration Techniques on Product Development Time. *IEEE Transactions on Engineering Management, Vol. 43, No. 2*.

[52] Siemens PLM Software. (2009). *Best Practice Brief: Establishing Effective Metrics for New Product Development Success*.

[53] Parastoo Mohagheghi & Reidar Conradi. (2008, June). 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.*

[54] William W. Thomas, Alex Delis & Victor R. Basili. (1997). An Analysis of Errors in a Reuse-Oriented Development Environment. *Journal of Systems and Software.* p. 211-224.

[55] ISO/IEC 14598. Information Technology Product Evaluation

[56] Donald Firesmith. (2007, January-February). Common Requirements Problems, Their Negative Consequences, and the Industry Best Practices to Help Solve Them. *Journal of Object Technology Vol. 6. No. 1*.

[57] Sohail Asghar & Mahrukh Umar. (2010, May). Requirement Engineering Challenges in Development of Software Applications and Selection of Customer-off-the-Shelf (COTS) Components. *Journal of Software Engineering. Vol. 1, Issue 1*.

[58] Humaira Naz & Muhammad Nadeem Khokhar. (2009). Critical Requirements Engineering Issues and Their Solution. *International Conference on Computer Modeling and Simulation*.

[59] Romi Satria Wahono. (2003). Analyzing Requirements Engineering Problems. *Proceedings of IECI Japan Workshop*.

[60] ISO/IEC 9126. Information Technology Product Quality

[61] Chethana Kuloor & Armin Eberlei. (2002). Requirements Engineering for Software Product Lines. *Proceedings of the 15th International Conference of Software and Systems Engineering and their Applications (ICSSEA'2002), Vol. 1*.

[62] Mohamed S. Shehata, Armin Eberlein & H.James Hoover. (2002). Requirements Reuse and Feature Interaction Management. *Proceedings of 15th International Conference on Software & Systems Engineering and their Applications*.

[63] Sami M. Homod & David C. Rine. (1999). Building requirements Repository Using Requirements Transformation Techniques to Support Requirements Reuse. *5th International Conference on Information Systems, Analysis and Synthesis*.

[64] Donald Firesmith. (2005). 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*.

[65] Mikael Svahnberg & Perolof Bengtsson. (2000). *Software Product Lines from Customer to Code*. Sweden: University of Karlskrona/Ronneby, Department of software Engineering and Computer Science Department, Research paper. Retrieved June 22, 2012, from

http://www.bth.se/fou/forskinfo.nsf/7172434ef4f6e8bcc1256f5f00488045/4a327c7ddcc7bec1c12568a3002cac25/$FILE/Research Report 1-00.pdf

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

[67] David P. Darcy, Chris F. Kemerer, Sandra A. Slaughter & James E. Tomayko. (2005, November). The Structural Complexity of Software: An Experimental Test. *IEEE Transactions on Software Engineering*, *Vol. 31, No. 11,* p. 982-995.

[68] Steven C. Braun & Udo Lindemann. (2008, December). The Influence of Structural Complexity on Product Costs. *IEEE International Conference on Industrial Engineering and Engineering Management*, p. 273-277.

[69] Nihal Orfi, Janis Terpenny & Asli Sahin-Sariisik. (2011). Harnessing Product Complexity: Step 1 - Establishing Product Complexity Dimensions and Indicators. *Journal of Engineering Economics, Vol. 56,* p. 59-80.

[70] Zhuo Zhang & Qunhui Luo. (2007, October). A Grey Measurement of Product Complexity. *IEEE International Conference on Systems, Man and Cybernetics*. p. 2176-2180.

[71] Morgan Swink. (2003, December). Completing Projects on-time: How Project Acceleration Affects New Product Development. *Journal of Engineering and Technology Management, Vol. 20, Issue 4,* p. 319-344.

[72] Alain Abran, Olga Ormandjieva & Manar Abu Talib. (2004). Information Theory-based Functional Complexity Measures and Functional Size with COSMIC-FFP. *Software Measurement Conference*.

[73] De Tran-Cao & Ghislain Lévesque. A Function Point Method for Software Complexity Measurement. Retreived 19 January 2013 from

http://132.208.137.12/etudiants/actuels/trancaode7.pdf

[74] De Tran-Cao, Alain Abran & Ghislain Lévesque. (2001). Functional Complexity Measurement. *International Workshop on Software Measurement*. p. 173-181.

[75] Jingqiu Shao & Yingxu Wang. (April 2003). A New Measure of Software Complexity Based on Cognitive Weights. *Canadian Conference on Electrical and Computer Engineering, Vol. 2,* Issue 2, pp. 1-6, April 2003.

[76] E. Da-Wei. (April 2007). The Software Complexity Model and Metrics for Object-Oriented. *IEEE International Workshop on Anti-Counterfeiting,* Security, Identification, pp. 464-469.

[77] Liu Sifeng, Jeffrey Forrest & Yang Yingjiel. (2011, September). A Brief Introduction to Grey Systems Theory. *IEEE International Conference on Grey Systems and Intelligent Services*. p. 1-9

[78] Rong Zhang, Liu Sifeng, & Liu Bin. (November 2007). An Extension for Grey Cluster Appraising Method Based on a Ranking Method of Interval Grey Number. *Proceedings of 2007 IEEE International Conference on Grey Systems and Intelligent Services*, pp. 313-317.

[79] Thomas J. McCabe. (December 1976). A Complexity Measure. *IEEE Transactions on Software Engineering, Vol. 2, No. 4*, pp. 308-320.

[80] Horst Zuse (December 1990). Software Complexity Measures and Methods. Walter De Gruyter.

[81] Jongbae Kim & David Wilemon. (2003, January). Sources and Assessment of Complexity in NPD Projects. *Journal of R&D Management*, *Vol. 33*, *Issue 1*, p. 15-30.

[82] Adrian Costea. (2007). On Measuring Software Complexity. *Journal of Applied Quantitative Methods*, *Vol. 2, No. 1,* p. 98-108.

[83] Farhad Ameri, Joshua D. Summers, Gregory M. Mocko & Matthew Porter. (2008, November). Engineering Design Complexity: An Investigation of Methods and Measures. *Journal of Research in Engineering Design*, *Vol. 19*, pp. 161-179.

[84] Usha Kumari & Shuchita Upadhyaya. (December 2011). An Interface Complexity Measure for Component-Base Software System. *International Journal of Computer Applications*, *Vol. 36*, *No.1*, pp. 46-52.

[85] Cigdem Gencel & Onur Demirors. (2008, June). Functional Size Measurement Revisited. *ACM Transactions on Software Engineering and Methodology, Vol. 17, Issue 3*, Article No 15.

[86] Gu Xunmei, Song Guoxin & Zheng Hong. (2006). The Comparison between FPA and COSMIC-FFP. *Proceedings of Software Measurement European Forum*.

[87] Kenneth Lind & Rogardt Heldal. (2010). On the Relationship between Functional Size and Software Code Size. *Proceedings of the ICSE Workshop on Emerging Trends in Software Metrics*, p. 47-52.

[88] Roberto Meli, Alain Abran, Vinh T. Ho & Serge Oligny. (2000, April). On the Applicability of COSMIC-FFP for Measuring Software Throughout Its Life Cycle. *ESCOM-SCOPE 2000*.

[89] Kostas Papageorgiou. (September 2011). How Your Technology Grows up? The 4 Stages of the Technology Life Cycle. *Technology Marketing blog*. Retrieved 31 March 2013 from

http://kanguro.fi/blog/how-your-technology-grows-up-the-4-stages-of-the-technology-life-cycle/

[90] Gunther Reinhart & Sebastian Schindler. (December 2010). A Strategic Evaluation Approach for Defining the Maturity of Manufacturing Technologies. *Journal of World Academy of Science, Engineering & Technology, Vol. 71*, pp. 730-735.

[91] Luigi Buglione, Filomena Ferrucci, Cigdem Gencel, Carmine Gravino & Federiva Sarro. (2010). Which COSMIC Base Functional Components are Significant in Estimating Web Application Development? - A Case Study. *Proceedings of the International Conference on Software Process and Product Measurement (MENSURA 2010)*, p.205-224.

[92] Michael Dunham. (2010). *Haut Tech Collaborative Software Development.* Retrieved October 6, 2012, from

http://blog.sciodev.com/2010/04/06/the-5-variables-of-project-estimation/

[93] Per Runeson & Martin Höst. (2009). Guidelines for Conducting and Reporting Case Study Research in Software Engineering. *Empirical Software Engineering, Vol. 14.* p. 131-164.

[94] Angele Cavaye. (1996). Case Study Research: A Multi-Faceted Research Approach for IS. *Information System Journal, Vol.6.* p. 227-242.

[95] Michael Gibbert, Winfried Ruigrok & Barbara Wicki. (2008). What Passes as a Rigorous Case Study?. *Strategic Management Journal, Vol. 29.* p. 1465-1474.

[96] Cornelius Herstatt, Christoph Stockstrom & Akio Nagahira. (2005). NPD-Process and Planning in Japanese Engineering Companies – Findings from an Interview Research. *Management of Technology and Innovation in Japan*. p. 249-265. Springer. ISBN: 10 3-540-25326-2.

[97] Jeremy J. Michalek, Fred M. Feinberg & Panos Y. Papalambros. (2005, January). Linking Marketing and Engineering Product Design Decisions via Analytical Target Cascading. *Journal of Product Innovation Management, Vol. 22, Issue 1*, p. 42-62.

[98] L. J. Schimmoeller. (2010, January). Success Factors of New Product Development Processes. *Journal of Advances in Production Engineering & Management, Vol. 5, No. 1*, p. 25-32.

[99] Joseph M. Bonner, Robert W. Ruekert & Orville C. Walker. (2002, May). Upper Management Control of New Product Development Projects and Project Performance. *Journal of Product Innovation Management, Vol. 19, Issue 3*, p. 233-245.

[100] COSMIC v.3.0.1, Measurement Manual, 2009, Common Software Measurement International Consortium.

[101] http://www.xlstat.com/en/

[102] Abdul Ali. (2000). The Impact of Innovativeness and Development Time on New Product Performance for Small Firms. Marketting Letters, Vol. 11, No 2, p. 151-163.

[103] Bryan A. Lukas & Ajay Menon. (2004, November). New Product Quality: Intended and Unintended Consequences of New Product Development Speed. *Journal of Business Research, Vol. 57, Issue 11*, p. 1258-1264.

[104] Fred Langerak & Erik Jan Hultink. (2008, September). The Effect of New Product Development Acceleration Approaches on Development Speed: A Case Study. *Journal of Engineering and Technology Management*, *Vol. 25*, *Issue 3*, p. 157-167.

[105] Fred Langerak & Erik Jan Hultink. (2005, February). The Impact of New Product Development Acceleration Approaches on Speed and Profitability: Lessons for

Pioneers and Fast Followers. *IEEE Transactions on Engineering Management, Vol. 52, No. 1*, p. 30-42.

[106] Bernardo A. Huberman & Dennis M. Wilkinson. (2005, December). Performance Variability and Project Dynamics. *Journal of Computational & Mathematical Organization Theory, Vol. 11*, Issue 4, p. 307-332.

[107] Christopher M. Schlick, Eric Beutner, Sönke Duckwitz & Torsten Licht. (2007). A Complexity Measure for New Product Development Projects. *IEEE International Engineering Management Conference.*

[108] David J. Closs, Mark A. Jacobs, Morgan Swink & G. Scott Webb. (2008, September). Toward a theory of competencies for the management of product complexity: Six case studies. *Journal of Operations Management, Vol. 26, Issue 5*, p. 590-610.

[109] Panos Fitsilis. (2009). Measuring the Complexity of Software Projects. *World Congress on Computer Science and Information Engineering*, p. 644-648.

[110] Akhilesh Bajaj, Sunder Kekre & Kannan Srinivasan. (2004, April). Managing NPD: Cost and Schedule Performance in Design and Manufacturing. *Journal of Management Science, Vol. 50, No. 4*, p. 527-536.

[111] Bryan A. Lukas, Ajay Menon & Simon J. Bell. (2002, July). Organizing for New Product Development Speed and the Implications for Organizational Stress. *Journal of Industrial Marketing Management, Vol. 31, Issue 4*, p. 349-355.

[112] Chen Jian'an & Hu Bei. (2007). Can Operational Centralization/ Decentralization Accelerate New Product Development? An Empirical Evidence from China. *Proceedings of IEEE International Conference on Grey Systems and Intelligent Services*.

[113] Christian Homburg, John P. Workman & Ove Jensen. (2000). Fundamental Changes in Marketing Organizations: The Movement Towards Customer-Focused Organizational Structure. *Journal of the Academy of Marketing Science, Vol. 28, No. 4*, p. 459-478.

[114] Christina Scott-Young & Danny Samson. (2008). Project Success and Project Team Management: Evidence from Capital Projects in the Process Industries. *Journal of Operations Management, Vol. 26, Issue 6*, p. 749-766.

[115] Christina Scott-Young & Danny Samson. (2009). Team Management for Fast Projects: An Empirical Study of Process Industries. *Journal of Operations & Production Management, Vol. 29, No.6*, p. 612-635.

[116] Dae Ryun Chang & Hang Cho. (2008, January). Organizational Memory Influences New Product Success. *Journal of Business Research, Vol. 61, Issue 1*, p. 13-23.

[117] Daniel J. Flint. (2002, July). Compressing New Product Success-to-Success Cycle Time: Deep Customer Value Understanding and Idea Generation. *Journal of Industrial Marketing Management, Vol. 31, Issue 4*, p. 305-315.

[118] Eric Fang & Robert W. Palmatier & Kenneth R. Evans. (2008, April). Influence of Customer Participation on Creating and Sharing of New Product Value. *Journal of the Academy of Marketing Science, Vol. 36, Issue 3*, p. 322-336.

[119] Edward F. McDonough, Kenneth B. Kahn & Abbie Griffin. (1999, November). Managing Communication in Global Product Development Teams. *IEEE Transactions on Engineering Management, Vol. 46, Issue 4*, p. 375-386.

[120] Gloria Barczak & David Wilemon. (2003, November). Team Member Experiences in New Product Development: Views from the Trenches. *Journal of R&D Management, Vol. 33, Issue 5*, p. 463-479.

[121] Ian P. McCarthy, Christos Tsinopoulos, Peter Allen & Christen Rose-Anderssen. (2006, September). New Product Development as a Complex Adaptive System of Decisions. *Journal of Product Innovation Management, Vol. 23, No. 5,* p. 437-456.

[122] Martin Hoegl & K. Praveen Parboteeah. (2006, March). Team Reflexivity in Innovative Projects. *Journal of R&D Management, Vol. 36, Issue 2,* p. 113-125.

[123] Nukhet Harmancioglu, Regina C. McNally, Roger J. Calantone & Serdar S. Durmusoglu. (2007, November). Your New Product Development (NPD) is only as good as Your Process: An Exploratory Analysis of New NPD Process Design and Implementation. *Journal of R&D Management, Vol. 37, Issue 5,* p. 399-424.

[124] Rajiv D. Banker, Indranil R. Bardhan & Ozer Asdemir. (2006, December). Understanding the Impact of Collaboration Software on Product Design and Development. *Journal of Information Systems Research, Vol. 17, Issue 4,* p. 352-373.

[125] Sara A. McComb, Stephen G. Green & W. Dale Compton. (2007, December). Team Flexibility's Relationship to Staffing and Performance in Complex Projects: An Empirical Analysis. *Journal of Engineering and Technology Management, Vol. 24, Issue 4*, p. 293-313.

[126] Andreas Pufall, Jan C. Fransoo & A.G. Ton de Kok. (2007). What Determines Product Ramp-up Performance?: A Review of Characteristics Based on a Case Study at Nokia Mobile Phones. Working paper. Technische Univeristeit Eindhoven.

[127] Debasish N. Mallick & Roger G. Schroeder. (2005, June). An Integrated Framework for Measuring Product Development Performance in High Technology Industries. *Journal of Production and Operations Management, Vol. 14, Issue 2*, p. 142-158.

[128] Devesh Verma & Kingshuk K. Sinha. (2002, September). Toward a Theory of Project Interdependencies in High Tech R&D Environments. *Journal of Operation Management, Vol. 20, Issue 5*, p. 451-468.

[129] Eric H. Kessler & Paul E. Bierly. (2002, February). Is Faster Really Better? An Empirical Test of the Implicaitons of Innovation Speed. *IEEE Transactions on Engineering Management, Vol. 49, No. 1*.

[130] Francisco-Jose Molina-Castillo & Jose-Luis Munuera-Aleman. (2009, October). New Product Performance Indicators: Time Horizon and Importance Attributed by Managers. *Journal of Technovation, Vol. 29, Issue 10*, p. 714-724.

[131] Fred Langerak, Serge Rijsdijk & Koen Dittrich. (2009, December). Development Time and New Product Sales: A Contingency Analysis of Product Innovativeness and Price. *Journal of Research in Marketing, Vol. 20, Issue 4*, p. 1-15.

[132] Gregory N. Stock & Mohan V. Tatikonda. (2004). External Technology Integration in Product and Process Development. *Journal of Operations & Production Management, Vol. 24, No. 7*, p. 642-665.

[133] Izabela Leskiewicz Sandvik & Kare Sandvik. (2003, December). The Impact of Market Orientation on Product Innovativeness and Business Performance. *Journal of Research in Marketing, Vol. 20, Issue 4*, p. 355-376.

[134] Jaakko Aspara, Henrikki Tikkanen, Erik Pöntiskoski & Paavo Järvensivu. (2011). Exploration and Exploitation Across Three Resource Classes: Market/Customer Intelligence, Brands/Bonds, and Technologies/Processes. *European Journal of Marketing, Vol. 45, Issue 4*, p. 596-630.

[135] Jurgen Mihm, Christoph Loch & Arnd Huchzermeier. (2003, June). Problem-Solving Oscillations in Complex Engineering Projects. *Journal of Management Science, Vol. 49, No. 6*, p. 733-750.

[136] Kathy A. Paulson Gjerde, Susan A. Slotnick & Matthew J. Sobel. (2002, October). New Product Innovation with Multiple Features and Technology Constraints. *Journal of Management Science, Vol. 48, Issue 10,* p. 1268-1284.

[137] Kyriakos Kyriakopoulos & Ko de Ruyter. (2004, December). Knowledge Stocks and Information Flows in New Product Development. *Journal of Management Studies, Vol. 41, Issue 8*, p. 1469-1498.

[138] Leslie Olin Morgan, Ruskin M. Morgan & William L. Moore. (2001). Quality and Time-to-Market Trade-offs when There Are Multiple Product Generations. *Journal of Manufacturing & Service Operations Management, Vol. 3, Issue 2,* p. 89-104.

[139] Ludwig Bstieler & Charles W. Gross. (2003). Measuring the Effect of Environmental Uncertainty on Process Activities, Project Team Characteristics, and New Product Success. *Journal of Business & Industrial Marketing, Vol. 18, No. 2*, p.146-161.

[140] Ludwig Bstieler. (2005, May). The Moderating Effect of Environmental Uncertainty on New Product Development and Time Efficiency. *Journal of Product Innovation Management, Vol. 22, No. 3*, p. 267-284.

[141] Marcus A. Rothenberger & Kevin J. Dooley. (1999). A Performance Measure for Software Reuse Projects. *Journal of Decision Sciences, Vol. 30, Issue 4*, p. 1131-1153.

[142] Marcus A. Rothenberger. (2003, December). Project-Level Reuse Factors: Drivers for Variation within Software Development Environments. *Journal of Decision Sciences, Vol. 34, Issue 1*, p. 83-106.

[143] Marcus A. Rothenberger, Kevin J. Dooley, Uday R. Kulkarni & Nader Nada. (2003, September). Strategies for Software Reuse: A Principal Component Analysis of

Reuse Practices. *IEEE Transactions on Software Engineering, Vol. 29, No.9*, p. 825-837.

[144] Michael D. Johnson. (2009, December). A Framework for Incorporating Time, Cost, and Fidelity Trade-offs among Design Assessment Methods in Product Development. *IEEE International Conference on Industrial Engineering and Engineering Management*, p. 578-582.

[145] Michael Tracey. (2004, September). A Holistic Approach to New Product Development: New Insights. *Journal of Supply Chain Management, Vol. 40, Issue 4*, p. 37-55.

[146] Michael Song & Morgan Swink. (2002, August). Marketing-Manufacturing Joint Involvement Across Stages of New Product Development: Effects on the Success of Radical vs. Incremental Innovations. *Academy of Management Proceedings & Membership Directory*.

[147] Mohan V. Tatikonda & Mitzi M. Montoya-Weiss. (2001, January). Integrating Operations and Marketing Perspectives of Product Innovation: The Influence of Organizational Process Factors and Capabilities on Development Performance. *Journal of Management Science, Vol. 47, No. 1*, p. 151-172.

[148] Mohan V. Tatikonda & Gregory N. Stock. (2003, November). Product Technology Transfer in the Upstream Supply Chain. *Journal of Product Innovation Management, Vol. 20, Issue 6,* p. 444-467.

[149] Mohan V. Tatikonda & Stephen R. Rosenthal. (2000, June). Successful Execution of Product Development Projects: Balancing Firmness and Flexibility in the Innovation Process. *Journal of Operations Management, Vol. 18, No. 4*, p. 401-425.

[150] Mohan V. Tatikonda & Stephen R. Rosenthal. (2000, February). Technology Novelty, Project Complexity, and Product Development Project Execution Success:

A Deeper Look at Task Uncertainty in Product Innovation. *IEEE Tarnsactions on Engineering Management, Vol. 47, No. 1*, p. 74-87.

[151] Morgan L. Swink & Roger Calantone. (2004, November). Design-Manufacturing Integration as a Mediator of Antecedents to New Product Design Quality. *IEEE Transactions on Engineering Management, Vol. 51, No. 4*, p. 472-482.

[152] Morgan Swink, Srinivas Talluri & Temyos Pandejpong. (2006, September). Faster, Better, Cheaper: A Study of NPD Project Efficiency and Performance Tradeoffs. *Journal of Operations Management, Vol. 24, No. 5*, p. 542-562.

[153] Paul Hong, Abraham Y. Nahm & William J. Doll. (2004). The Role of Project Target Clarity in an Uncertain Project Environment. *Journal of Operations & Production Management, Vol. 24, Issue 12*, p. 1269-1291.

[154] Robert W. Veryzer. (2005, January). The Roles of Marketing and Industrial Design in Discontinuous New Product Development. *Journal of Product Innovation Management, Vol. 22, Issue 1*, p. 22-41.

[155] Robert J. Trent. (2004, June). The Use of Organizational Design Features in Purchasing and Supply Management. *Journal of Supply Chain Management, Vol. 40, Issue 3*, p. 4-18.

[156] Subroto Roy, Kumaraswamy Sivakumar & lan F. Wilkinson. (2004, January). Innovation Generation in Supply Chain Relationships: A Conceptual Model and Research Propositions. *Journal of the Academy of Marketing Science, Vol. 32, No. 1*, p. 61-79.

[157] Susan Hart, Nikolaos Tzokas & Michael Saren. (1999). The Effectiveness of Market Information in Enhancing New Product Success Rates. *European Journal of Innovation Management, Vol. 2, Issue 1*, p. 20-35.

[158] Zhuo Zhang, Hongyuan Qin & Lin Wang. (2009, November). A Measurement of Organizational Complexity and Its Impact on Quality Economics - A Grey Perspective. *IEEE International Conference on Grey Systems and Intelligent Services*, p. 783-787.

[159] Donald Gerwin & Nicholas J. Barrowman. (2002, July). An Evaluation of Research on Integrated Product Development. *Journal of Management Science, Vol. 48, No.7,* p. 938-953.

[160] Ely Dahan & John R. Hauser. (2002, September). The Virtual Customer. *Journal of Product Innovation Management, Vol. 19, Issue 5*, p. 332-353.

[161] Karlos A. Artto & Kim Wikström. (2005, July). What is Project Business?. *Journal of Project Management, Vol. 23, Issue 5,* p. 343-353.

[162] Kathleen Kier Wheatley & David Wilemon. (1999). From Emerging Technology to Competitive Advantage. *Portland International Conference on Management of Engineering and Technology, Vol. 1*.

[163] Vish Krishnan & Karl T. Ulrich. (2001, January). Product Development Decisions: A Review of the Literature. *Journal of Management Science*, *Vol. 47*, *No.1*, p. 1-21.

[164] Larry J. Menor, Mohan V. Tatikonda & Scott E. Sampson. (2002, April). New Service Development: Areas for Exploitation and Exploration. *Journal of Operations Management, Vol. 20, Issue 2*, p. 135-157.

[165] Semra Yilmaz Taştekin, Yusuf Murat Erten & Semih Bilgen. (September 2013). Software Product Complexity Estimation Using Grey Measurement. *39th Euromicro Conference on Software Engineering and Advanced Applications.*

[166] Kesar Singh & Minge Xie. Bootstrap: A Statistical Method. Rutgers University. Retrieved September 15, 2013, from

http://www.stat.rutgers.edu/home/mxie/rcpapers/bootstrap.pdf

[167] R-Tutorial, Calculating Confidence Interval. Clarkson University, Depatment of Mathematics. Retrieved September 15, 2013, from

http://www.cyclismo.org/tutorial/R/confidence.html

APPENDICES

APPENDIX A: CITED RESEARCHES OF GRIFFIN'S STUDY [9]

Primary Focus of the Studies: Reducing development times	
[102]	This study emphasizes the product innovativeness affect the product success, that is, success of a product that is ahead (very innovative) or behind (late in the market) its time will suffer. That study investigates the relationship between product innovativeness and development time and market performance. According to the analysis, introducing highly innovative products too early or less innovative product too late will not meet the profit expectation of the firms. Griffin's study is referred as an empirical study.
[103]	This study determines the effect of reducing the development time on product quality. In this study, Griffin's definition for the product development time is used.
[104], [105]	These studies investigate nine development acceleration methods on development speed. In the scope of these studies, supplier involvement, user involvement, speeding activities, reduction of part in product, training/rewarding employees, implementing support system, stimulating inter-functional cooperation, emphasizing customer value and simplifying the organizational structure determinants were discussed. Griffin's study is referred to state the effect of inter-functional coordination on development cycle.
[5]	This study investigates the determinants of software development project duration. Authors used a development time in the literature as a base, proposed for physical products. They figured out that newness is an important determinant on software development time. They also figure out that coordination between different departments reduces the development time. They have defined some determinants for software development times and they came up with some hypothesis. These determinants are planning, supplier involvement, sales and marketing involvement, build frequency (design iterations), testing and project leader power. They analyzed the correlation between each

	determinant. Griffin's study was used as the basis for product development time. Authors found that previous experiences on product influence the development time as suggested by Griffin.
[71]	This article addresses reducing the project development time strategies regarding the project complexity, management approaches and design integration. Author brings the relationship between development time and product newness, product complexity, technological novelty, organizational complexity, management support, goal explicitness, activity overlaps, manufacturing involvement, quality function, use of computer aided design. Griffin's study is cited for the relationship between development time and product complexity and product newness.
Primary Focus of the Studies: Product complexity	
[106]	The effect of system size on completion time is studied. Complexity is analyzed by the number of tasks and the number of interactions in projects. This is compared with the Griffin's study.
[107]	This study analyzes the project complexity for coupled or uncoupled tasks in the projects. As a result of the study, if the tasks coupling increases, complexity also increase correspondingly. There is a case study for a sensor project to show the complexity analysis, but there were not sufficient data which include the subjective estimates for the measurements. Griffin's study is cited for the effect of number of functions embodied in the product.
[108]	This study investigates the product complexity management for product portfolio with six case studies. Product portfolio complexity was associated with external environment and organizational factor such as portfolio strategy, organization decision process and design and decision support system. Griffin's study is referred for the definition of complexity.
[81]	This study examines the importance of complexity factor on successfully managing the new products development. Authors study some source of complexity for new projects. Complexity definitions in the literature are studied and a template for complexity evaluation is constructed. According to this template complexity is assessed for R&D, engineering, manufacturing and marketing departments for different source of complexity such as technological, market, development,

	marketing, organizational, inter-organizational and other factor. Griffin's definition for complexity is cited in that study.
[109]	This paper presents a summary of identified sources for software project complexity based on the Project Management Body Of Knowledge (PMBOK) with Patterns of Complexity. Author focuses on the knowledge areas defined in the PMBOK. These are project integration management, scope management, time management, cost management, quality management, human resource management, communication management, risk management and procurement management. Griffin's study is cited for the relationship between product and project complexity.
Primary Focus of the Studies: Organizational factors (project team, processes etc.)	
[19]	This study analysis the product development time for different type of the products, such as new to the world, new to the firm, next generation product and incremental product. Griffin's study is referred to use the important parameters for the development time. It is also referred as an empirical study.
[110]	This study highlights the importance of the interdependencies between design and manufacturing process in the organization. Study mainly focuses on the design phase and important factors which affect the project performance. These factors are specialization, managerial level and customer interaction. The study covers the missile system projects. Griffin's study has been referred in the scope of cross-functionality.
[103], [111]	Bureaucratic structure of the organization is studied. In this scope, the effect of formalization, centralization and formal/informal control on the product quality is analyzed in [103]. The effect of same factors on the development speed is studied in [111]. Formalization creates the rules, codes, and instructions to define roles, authority relations, and procedures in an organization. Centralization is related with the decision-making authority and responsibilities among the people. Formal control covers the written procedures.
[112]	The relationship between decision-making power and development speed is studied for Chinese firms. Determinants in the development time model in Griffin's study are referred.
[113]	Changes in the marketing organizations are studied. There are some variables which are used to compare the changes in the organization. These are organization's

	structure, coordination of the activities, organizational culture and distribution of power.
[114]	Effect of project team management on the project cost and schedule, correspondingly project success is studied. Authors propose a project team model regarding the social psychological, cognitive, ecological approaches. Griffin's study is referred to state the effect of cross-functional team to development cycle.
[115]	The impact of project team management on the speed of project execution and project construction for capital projects is studied. Capital project is defined as improving the production capacity in response to market demand by establishing necessary plant and infrastructure. Griffin's study is cited to state the importance of the people management factor and the effect of cross-functionality on the development time.
[96]	Planning practices for innovation projects in Japanese companies were analyzed. According to that analysis, details of formal processes in companies were discussed. The relationship of development cycle time and formal processes is referred from Griffin's study.
[116]	This study analyzes the impact of organizational memory and information sharing on product development performance. The relationship between formal development process and new product success is stated from Griffin's study.
[117]	This study proposes a process which includes the customer value for new products development. According to this process, deep customer understanding will produce more successful ideas. Griffin's study is referred to state that formal processes reduce product development cycle time.
[118]	This study discusses the importance of customer participation in product development. This will improve the organization's processes by information sharing and coordination. Griffin's study is referred to support that study. Griffin suggests that a formalized development process involving the customer is more likely to improve the effectiveness of the product development process.

[119]	This study investigates the management of communication between the development teams. The impact of the country, location and culture of the organization on communication was analyzed.
[120]	This study discusses the factors which influence the product development team members. This study focuses on the actual experiences of the team members who perform the development activities. Griffin's study is referred with her suggestion of rewarding the team members.
[121]	This study proposes a complex adaptation system for new products decision making processes. This framework complements the existing frameworks in the literature. Griffin's study is referred to state the importance of some metrics on innovation.
[97]	This study proposes a method to formalize the coordination between marketing and design departments. Authors propose to use a well-known approach in design engineering in the marketing community. In such a way engineering and marketing departments are integrated more closely. Griffin's highlight on the use of cross-functional team is cited in that study.
[99]	This study examines the formal control processes for the project team and analyzes the impact of this control on new product development team performance. The relationship between the use of cross-functional team and the speed of the projects is cited from Griffin's study.
[122]	This paper studies the team reflexivity and impact of that on the team efficiency. Team reflexivity is defined as the adaption of the team on the changing environments.
[123]	This paper analyzes the implementation of new product development processes by regarding the senior management involvement, business case content, customer interactions and cross-functional integration. Authors explore the combination of those elements and their consequences for increase in productivity. Authors conducted 3 case studies for incremental and radical product types. Griffin's study is referred for citing the formal processes and cross-functional integration
[124]	This paper studies the impact of collaborative design team on the product quality, cycle time and cost. During the collaboration between the teams, the use of information technologies is emphasized.

	Griffin's study is referred to cite the scope of complexity.
[125]	This paper studies the flexibility issue and its impact on team performance. Project complexity is also concerned relating to team flexibility. Griffin's study is cited for the impact of the team with high quality technical skills on the project success.
Primary Focus of the Studies: Other factors affect the product/business performance	
[126]	This study analysis the important characteristics on the product ramp-up performance. These are product architecture, product development process, logistics system, manufacturing capability and external environment. Ramp-up is defined as the phase which starts by introducing the product with low volume, and after developing the confidence, it continuous by increasing the volume of the product. Griffin's study is referred in the scope of the product architecture which defines the product newness.
[127]	This study proposes a framework for measuring the product development performance. Authors used some metrics studied in different researches for development, manufacturing and marketing. This study referred to some of study by Griffin. These studies are related with the marketing metrics for the product performance.
[128]	This study investigates the concurrent project management. Study developed a framework for the interdependencies between projects in the organization and its impact on the project performance. Interdependencies were classified as resources, technology and market interdependencies, but the framework includes only resource and technology. The relationship between newness of the product and development time is quoted from Griffin's study.
[129]	The effect of innovation speed on project cost, quality and project success is investigated. Innovation speed is defined as the time beginning from the conceptual design to introduction of the product to customer. Griffin's study is cited to refer that, outcomes, processes and structure for new and incremental products are different.
[130]	This study reviews the way for defining product performance for short-term and long-term.

	Griffin's study is used as literature data for product performance dimensions.
[131]	The effect of the product development time on product sales was studied. New products and incremental products were studied. The price determinant was discussed for the product sales. Griffin's study is referred to state the difficulty of new products regarding the planning and implementation.
[132]	This study examines acquiring a technology from external sources and integrating it into a product or process. Griffin's study is used as literature data for product complexity sub-dimension.
[133]	Influence of product innovativeness on business performance is studied. Innovativeness is classified as new to the firm and new to the market. Griffin's study is referred for dealing with the team organization.
[134]	This study explains the exploration and exploitation strategies for innovative projects on different dimensions such as market and customer knowledge, brands and bonds with technology dimension.
[135]	This study proposes a model for solving the design problems by regarding the project size, project team coordination and management control. The relationship between the project complexity and the project cycle time is cited from Griffin's study.
[136]	This paper studies product innovation as either frontier or incremental. Authors discuss the decision making process in new product development. The metrics which affects the development cycle time is mentioned and referred as Griffin's study.
[137]	This paper studies the procedural and declarative memory in the organization and their impact on product outcomes such as financial performance or creativity. The relationship between the formal processes and product quality is cited from Griffin's study.
[138]	This article studies the time-to-market concern regarding the quality for single version and multiple version products. This study compares the profits for multiple version products and single version products.

	Griffin's research results are used as assumptions in that study.
[139]	This study examines the environmental factors that affect the new product development success. For the varying market and technological conditions, management strategies and development approaches are discussed. Griffin's suggestion to use the cross-functional team is used as a determinant for new product success.
[140]	Similar to [139], this article studies the environmental factors in two different countries, Australia and Canada. Again, the use of cross-functional team is cited by referring Griffin's study.
[141], [142], [143]	[141] examines the project performance with code-reuse. Authors developed project performance measurement framework depending on productivity of new code and reusing code, quality of reuse decision and value of reuse to the company. Success factors for software reuse and reuse strategies are discussed in [142] and [143], respectively. The relationship between the newness and development time is cited from Griffin's study.
[144]	This paper presents a framework for the assessment of product design process performance. Author focuses on the tradeoffs between lead time, cost and risk during the development. Assessment is based on a simple product's development process. Griffin's study is cited in the literature review section of the paper.
[145]	This paper studies the participation of different parties during the product development time and studies the influence of these parties on the project performance. These parties are manufacturing, purchasing, logistic departments in the organization and participation of suppliers and customer. This study is based on the US manufacturing companies. Griffin's study is cited for the inclusion of the members from different functional departments.
[146]	This article studies the involvement of marketing and manufacturing departments during the product development for radical and incremental products. Griffin's study is cited for the benefits of cross-functional team.
[147]	This paper studies the involvement of marketing and operations departments during

	the product development and studies the influence on the project performance. Authors analyze the managerial decisions under different project conditions. Griffin's study is cited for the formal processes and its effect on project success.
[148]	This article studies the procurement of some part of a product from outside of the company and integrating this part into a new product. Authors concern the technology uncertainty and complexity for the transfer of product. Griffin's study is referred for the definition of product newness and product complexity.
[149], [150]	[149] addresses the project management aspects during the product development based on the project types. Authors focus on the project planning and mostly on project execution phases. They bring the relationship between project execution and project formality, project management autonomy, resource flexibility, technology novelty. [150] studies the relationship between technology novelty and project complexity in detail. Griffin's study is cited for the relationship between formal processes, product complexity and development time.
[151]	This paper examines the technological novelty, organization complexity and design-manufacturing integration on product design quality. Quality is defines as the fitness for customer use. Authors suggest some managerial practices. Griffin's study is cited for the product complexity.
[152]	This study addresses the relationship between the project efficiency and market success, project management experience, management commitment, explicit project goals, collaborative work environment, project team collocation, design-manufacturing integration, activity overlap. Griffin's study is used as a source for the measurements.
[153]	This study presents the relationship between uncertain elements of the projects and the project outcomes. Uncertainty is studied for project environment, project target clarity and teamwork. Authors suggest some implications for successful projects. Griffin's study is cited for the use of cross-functional team.
[154]	This paper investigates the marketing and design disciplines into the development processes for radical new products. Author brings some propositions regarding the industrial design, marketing, product discontinuity, formal processes and recognition and appreciation among R&D managers.

	Griffin's study is cited for the relationship between the project success and cycle time and development processes.
[155]	This study examines the organizational processes such as assessing and selecting the structure, formal communication system, labor division, coordination, control, authority and responsibility for procurement and supply organizations. Griffin's study is cited to support this study for the use of cross-functional team.
[6]	This paper studies the impact of development process, organizational mechanism and strategic capabilities on the project time. The interaction between these determinants is also discussed. Griffin's study is cited for the impact of team, product newness and product complexity on cycle time. It is considered that her model contributes to fill the gap on development time estimations.
[156]	This paper proposes a model for the generation of innovation in buyer and seller relationship by studying the external and internal factors. Authors discuss the managerial implications for a better management of innovation. Griffin's study is cited in the literature review part.
[157]	This paper studies the use of market information during pre-development and development phases for the success rate of new products. This study concerns the product type, such as innovative product, extended product etc. The relationship between the newness of product and activities in product development process is also discussed. Griffin's study is referred for the relationship between product newness and cycle time.
[7]	This paper studies the relationship between the market knowledge and product success. For the product success, product effectiveness and time performance are concerned. Griffin's study is cited in the literature review part.
[158]	This paper studies the impact of organization complexity on the product cost. Organizational complexity is concerned as organizational structure, managerial processes, resource allocation and business rules. Griffin's study is cited to use the definition of complexity.

Primary Focus of the Studies: Literature research or review paper	
[159]	This study evaluates the researches on integrated product development. Integrated product development includes overlapped or interacted activities in new product development process. This study also evaluates the integrated product development and project performance. It suggests extending researches on project teams, portfolio management and collaboration between firms. Statistical data from Griffin's study have been referred in the study.
[160]	This study reviews the inclusion of customer feedback during the product development stage through the web page application. Customer virtually works with the project team. Griffin's study is referred to state the effect of customer input for product changes during the development.
[2]	This study reviews the metrics for engineering design projects. These metrics are classified as feasibility assessment, design effort, design effort distribution with time and duration. Griffin's study is explained as a proposed study which investigates the duration metrics for design projects. This study cites Griffin's study as using less subjective estimations when compared to other studies. Besides, this study states that Griffin's definition for product complexity does not give a good picture for product complexity because it assumes all functions are equally difficult to develop.
[161]	This paper studies the business and project concept. In this context, project management approaches are discussed. Author review the literature and they classify the researches in different clusters. Griffin's study is cited under "accelerating new product development" cluster.
[162]	This article studies the emerging technologies management to make the organization's ideas more competitive. Griffin's suggestion about the use of cross-functional team is referred in that article.
[31]	This paper studies the literature about the best practices for new product development. Authors study the impact of best practices on new product development processes regarding the customer requirements, concept generation, concept selection, concept design, product strategy, design process, manufacturing and marketing activities, product improvement etc. Griffin's study is cited as one of the previous best practices. The use of cross-

	functional team is summarized.
[163]	This study reviews the literature for decision making process on product development. Decisions are categorized in four groups: concept development, supply-chain design, product design, production ramp-up and launch. Griffin's study is cited in the literature researches.
[164]	This paper studies the design and development issue for new services. Authors provide a review of literature on new service development subject. Some future research areas are highlighted in the study.
[98]	This paper examines the cross-functional team, upper-management support and organizational structure and their effect on the project performance. Griffin's study is used as literature source for cross-functional team and organizational structure.

APPENDIX B: PRODUCT COMPLEXITY MEASUREMENTS FOR CASE STUDIES

CASE STUDY 1 (PROJECT A2):

Table 45 Case Study 1 Determinants, Indicators and Actual Values

First Level Determinants	Second Level Indicators	Unit	Symbol	Actual Value
Technology	Number of Technology	per	X1	9
	Maturity of Technology	1-4	X2	2
Physical Characteristics	Number of components	per	X3	18
	Volume	m3	X4	1
Organization	People	person	X5	23
	Departments	per	X6	9
	Information transfer	per file	X7	21
	Resource allocation	1-4	X8	2
Environment	Number of suppliers and customer	per	X9	4
	Requlations and standards	per file	X10	7
	Market and competitions	1-4	X11	2

Table 46 Case Study 1 Weights of Indicators

First Level Determinants	r_i	r_i'	f_i	Second Level Indicators	$r_{i,j}$	$r_{i,j}'$	η_j
Technology	1.5	2.535	0.3885	Number of Technology	1.5	1.5	0.2331
				Maturity of Technology		1	0.1554
Physical Characteristics	1.3	1.69	0.259	Number of components	1.4	1.4	0.1511
				Volume		1	0.1079
Organization	1.3	1.3	0.1992	People	1.4	2.548	0.0761
				Departments	1.4	1.82	0.0544
				Information transfer	1.3	1.3	0.0388
				Resource allocation		1	0.0299
Environment	1.2	1	0.1533	Number of suppliers and customer	1.3	1.95	0.0672
				Requlations and standards	1.5	1.5	0.0517
				Market and competitions		1	0.0344

Table 47 Case Study 1 Indicators and Grey Clusters

X_j	Actual Value	Weight (η_j)	Low $[a^1, a^2]$	Whiten (λ_j^1)	Moderate $[a^2, a^3]$	Whiten (λ_j^2)	High $[a^3, a^4]$	Whiten (λ_j^3)
X1	9	0.2331	[2, 5]	3.50	[5, 15]	10.00	[15, 30]	22.50
X2	2	0.1554	[1, 2]	1.50	[2, 3]	2.50	[3, 4]	3.50
X3	18	0.1511	[10, 30]	20.00	[30, 80]	55.00	[80, 300]	190.00
X4	1	0.1079	[0.2, 0.5]	0.35	[0.5, 1]	0.75	[1, 3]	2.00
X5	23	0.0761	[5, 20]	12.50	[20, 50]	35.00	[50, 150]	100.00
X6	9	0.0544	[1, 5]	3.00	[5, 10]	7.50	[10, 30]	20.00
X7	21	0.0388	[5, 20]	12.50	[20, 50]	35.00	[50, 150]	100.00
X8	2	0.0299	[1, 2]	1.50	[2, 3]	2.50	[3, 4]	3.50
X9	4	0.0672	[1, 10]	5.50	[10, 30]	20.00	[30, 100]	65.00
X10	7	0.0517	[2, 5]	3.50	[5, 10]	7.50	[10, 20]	15.00
X11	2	0,0344	[1, 2]	1.50	[2, 3]	2.50	[3, 4]	3.50

Table 48 Case Study 1 Grey Clusters of Product Complexity

X_j	Weight (η_j)	$f_j^1(x)$	$f_j^2(x)$	$f_j^3(x)$
X1	0.2331	0.522	0.875	0.229
X2	0.1554	0.667	0.667	0
X3	0.1511	0.9	0.178	0
X4	0.1079	0	0.889	0.333
X5	0.0761	0.72	0.6	0.038
X6	0.0544	0.143	0.933	0.267
X7	0.0388	0.773	0.533	0.013
X8	0.0299	0.667	0.667	0
X9	0.0672	0.727	0.158	0
X10	0.0517	0.462	0.909	0.2
X11	0.0344	0.667	0.667	0
Grey Coefficient		**σ^1 (Low)**	**σ^2 (Mod.)**	**σ^3 (High)**
		0.569	**0.648**	0.117

CASE STUDY 2 (PROJECT A3):

Table 49 Case Study 2 Determinants, Indicators and Actual Values

First Level Determinants	Second Level Indicators	Unit	Symbol	Actual Value
Technology	Number of Technology	per	X1	10
	Maturity of Technology	1-4	X2	1
Physical Characteristics	Number of components	per	X3	36
	Volume	m3	X4	1.28
Organization	People	person	X5	19
	Departments	per	X6	8
	Information transfer	per file	X7	18
	Resource allocation	1-4	X8	2
Environment	Number of suppliers and customer	per	X9	3
	Requlations and standards	per file	X10	7
	Market and competitions	1-4	X11	2

Table 50 Case Study 2 Weights of Indicators

First Level Determinants	r_i	r_i’	f_i	Second Level Indicators	$r_{i,j}$	$r_{i,j}$’	η_j
Technology	1.5	2.94	0.4027	Number of Technology	1.5	1.5	0.2416
				Maturity of Technology		1	0.1611
Physical Characteristics	1.4	1.96	0.2685	Number of components	1.5	1.5	0.1611
				Volume		1	0.1074
Organization	1.4	1.4	0.1918	People	1.4	2.548	0.0733
				Departments	1.4	1.82	0.0523
				Information transfer	1.3	1.3	0.0374
				Resource allocation		1	0.0288
Environment	1.3	1	0.137	Number of suppliers and customer	1.4	1.96	0.0616
				Requlations and standards	1.4	1.4	0.044
				Market and competitions		1	0.0314

Table 51 Case Study 2 Indicators and Grey Clusters

X_j	Actual Value	Weight (η_j)	Low [a^1, a^2]	Whiten (λ_j^1)	Moderate [a^2, a^3]	Whiten (λ_j^2)	High [a^3, a^4]	Whiten (λ_j^3)
X1	10	0.2416	[2, 5]	3,50	[5, 15]	10,00	[15, 30]	22,50
X2	1	0.1611	[1, 2]	1,50	[2, 3]	2,50	[3, 4]	3,50
X3	36	0.1611	[10, 30]	20,00	[30, 80]	55,00	[80, 300]	190,00
X4	1.28	0.1074	[0.2, 0.5]	0,35	[0.5, 1]	0,75	[1, 3]	2,00
X5	19	0.0733	[5, 20]	12,50	[20, 50]	35,00	[50, 150]	100,00
X6	8	0.0523	[1, 5]	3,00	[5, 10]	7,50	[10, 30]	20,00
X7	18	0.0374	[5, 20]	12,50	[20, 50]	35,00	[50, 150]	100,00
X8	2	0.0288	[1, 2]	1,50	[2, 3]	2,50	[3, 4]	3,50
X9	3	0.0616	[1, 10]	5,50	[10, 30]	20,00	[30, 100]	65,00
X10	7	0.044	[2, 5]	3,50	[5, 10]	7,50	[10, 20]	15,00
X11	2	0.0314	[1, 2]	1,50	[2, 3]	2,50	[3, 4]	3,50

Table 52 Case Study 2 Grey Clusters of Product Complexity

X_j	Weight (η_j)	$f_j^1(x)$	$f_j^2(x)$	$f_j^3(x)$
X1	0.2416	0,435	1	0,286
X2	0.1611	0,667	0	0
X3	0.1611	0,733	0,578	0,038
X4	0.1074	0	0,764	0,52
X5	0.0733	0,827	0,467	0
X6	0.0523	0,286	0,978	0,2
X7	0.0374	0,853	0,433	0
X8	0.0288	0,667	0,667	0
X9	0.0616	0,545	0,105	0
X10	0.044	0,462	0,909	0,2
X11	0.0314	0,667	0,667	0
Grey Coefficient		**σ^1 (Low)**	**σ^2 (Mod.)**	**σ^3 (High)**
		0.532	**0.605**	0.150

CASE STUDY 3 (PROJECT A5):

Table 53 Case Study 3 Determinants, Indicators and Actual Values

First Level Determinants	Second Level Indicators	Unit	Symbol	Actual Value
Software Characteristics	COSMIC FFP Size	cfsu	X1	578
	Interactions with other system	1-4	X2	3
Technology	Number of Technology	per	X3	8
	Maturity of Technology	1-4	X4	1
Organization	People	person	X5	17
	Departments	per	X6	5
	Information transfer	per file	X7	19
	Resource allocation	1-4	X8	2
Environment	Number of suppliers and customer	per	X9	4
	Requlations and standards	per file	X10	4
	Market and competitions	1-4	X11	2

Table 54 Case Study 3 Weights of Indicators

First Level Determinants	r_i	r_i'	f_i	Second Level Indicators	$r_{i,j}$	$r_{i,j}'$	η_j
Software Characteristics	1.5	2.52	0.3938	COSMIC FFP Size	1.5	1.5	0.2363
				Interactions with other system		1	0.1575
Technology	1.4	1.68	0.2625	Number of Technology	1.4	1.4	0.1531
				Maturity of Technology		1	0.1094
Organization	1.2	1.2	0.1875	People	1.4	2.744	0.0724
				Departments	1.4	1.96	0.0517
				Information transfer	1.4	1.4	0.037
				Resource allocation		1	0.0264
Environment	1	1	0.1563	Number of suppliers and customer	1.3	1.95	0.0685
				Requlations and standards	1.5	1.5	0.0527
				Market and competitions		1	0.0351

Table 55 Case Study 3 Indicators and Grey Clusters

X_j	Actual Value	Weight (η_j)	Low $[a^1, a^2]$	Whiten (λ_j^1)	Moderate $[a^2, a^3]$	Whiten (λ_j^2)	High $[a^3, a^4]$	Whiten (λ_j^3)
X1	578	0.2363	[163, 334]	248.50	[334, 870]	602.00	[870, 1090]	980.00
X2	3	0.1575	[1, 2]	1.50	[2, 3]	2.50	[3, 4]	3.50
X3	8	0.1531	[2, 5]	3.50	[5, 15]	10.00	[15, 30]	22.50
X4	1	0.1094	[1, 2]	1.50	[2, 3]	2.50	[3, 4]	3.50
X5	17	0.0724	[5, 20]	12.50	[20, 50]	35.00	[50, 150]	100.00
X6	5	0.0517	[1, 5]	3.00	[5, 10]	7.50	[10, 30]	20.00
X7	19	0.037	[5, 20]	12.50	[20, 50]	35.00	[50, 150]	100.00
X8	2	0.0264	[1, 2]	1.50	[2, 3]	2.50	[3, 4]	3.50
X9	4	0.0685	[1, 10]	5.50	[10, 30]	20.00	[30, 100]	65.00
X10	4	0.0527	[2, 5]	3.50	[5, 10]	7.50	[10, 20]	15.00
X11	2	0.0351	[1, 2]	1.50	[2, 3]	2.50	[3, 4]	3.50

Table 56 Case Study 3 Grey Clusters of Product Complexity

X_j	Weight (η_j)	$f_j^1(x)$	$f_j^2(x)$	$f_j^3(x)$
X1	0.2363	0,47	0,945	0,378
X2	0.1575	0	0,667	0,667
X3	0.1531	0,609	0,75	0
X4	0.1094	0,667	0	0
X5	0.0724	0,88	0,4	0
X6	0.0517	0,714	0,615	0
X7	0.037	0,827	0,467	0
X8	0.0264	0,667	0,667	0
X9	0.0685	0,727	0,158	0
X10	0.0527	0,923	0,364	0
X11	0.0351	0,667	0,667	0
Grey Coefficient		**σ^1 (Low)**	**σ^2 (Mod.)**	**σ^3 (High)**
		0.548	**0.592**	0.194

CASE STUDY 4 (PROJECT A7):

Table 57 Case Study 4 Determinants, Indicators and Actual Values

First Level Determinants	Second Level Indicators	Unit	Symbol	Actual Value
Software Characteristics	COSMIC FFP Size	cfsu	X1	296
	Interactions with other system	1-4	X2	2
Technology	Number of Technology	per	X3	3
	Maturity of Technology	1-4	X4	1
Organization	People	person	X5	9
	Departments	per	X6	3
	Information transfer	per file	X7	6
	Resource allocation	1-4	X8	1
Environment	Number of suppliers and customer	per	X9	1
	Requlations and standards	per file	X10	2
	Market and competitions	1-4	X11	2

Table 58 Case Study 4 Weights of Indicators

First Level Determinants	r_i	r_i’	f_i	Second Level Indicators	$r_{i,j}$	$r_{i,j}$’	η_j
Software Characteristics	1.4	2.184	0.3674	COSMIC FFP Size	1.4	1.4	0.2143
				Interactions with other system		1	0.1531
Technology	1.3	1.56	0.2624	Number of Technology	1.3	1.3	0.1483
				Maturity of Technology		1	0.1141
Organization	1.2	1.2	0.2019	People	1.3	1.56	0.0662
				Departments	1.2	1.2	0.0509
				Information transfer	1	1	0.0424
				Resource allocation		1	0.0424
Environment	1	1	0.1682	Number of suppliers and customer	1.2	1.56	0.068
				Requlations and standards	1.3	1.3	0.0567
				Market and competitions		1	0.0436

Table 59 Case Study 4 Indicators and Grey Clusters

X_j	Actual Value	Weight (η_j)	Low [a^1, a^2]	Whiten (λ_j^1)	Moderate [a^2, a^3]	Whiten (λ_j^2)	High [a^3, a^4]	Whiten (λ_j^3)
X1	296	0.2143	[163, 334]	248.50	[334, 870]	602.00	[870, 1090]	980.00
X2	2	0.1531	[1, 2]	1.50	[2, 3]	2.50	[3, 4]	3.50
X3	3	0.1483	[2, 5]	3.50	[5, 15]	10.00	[15, 30]	22.50
X4	1	0.1141	[1, 2]	1.50	[2, 3]	2.50	[3, 4]	3.50
X5	9	0.0662	[5, 20]	12.50	[20, 50]	35.00	[50, 150]	100.00
X6	3	0.0509	[1, 5]	3.00	[5, 10]	7.50	[10, 30]	20.00
X7	6	0.0424	[5, 20]	12.50	[20, 50]	35.00	[50, 150]	100.00
X8	1	0.0424	[1, 2]	1.50	[2, 3]	2.50	[3, 4]	3.50
X9	1	0.068	[1, 10]	5.50	[10, 30]	20.00	[30, 100]	65.00
X10	2	0.0567	[2, 5]	3.50	[5, 10]	7.50	[10, 20]	15.00
X11	2	0.0436	[1, 2]	1.50	[2, 3]	2.50	[3, 4]	3.50

Table 60 Case Study 4 Grey Clusters of Product Complexity

X_j	Weight (η_j)	$f_j^1(x)$	$f_j^2(x)$	$f_j^3(x)$
X1	0.2143	0.924	0.303	0
X2	0.1531	0.667	0.667	0
X3	0.1483	0.857	0.125	0
X4	0.1141	0.667	0	0
X5	0.0662	0.72	0.133	0
X6	0.0509	1	0.308	0
X7	0.0424	0.48	0.033	0
X8	0.0424	0.667	0	0
X9	0.068	0.182	0	0
X10	0.0567	0.571	0	0
X11	0.0436	0.667	0.667	0
Grey Coefficient		**σ^1 (Low)**	**σ^2 (Mod.)**	**σ^3 (High)**
		0.724	0.240	0

CASE STUDY 5 (PROJECT B2):

Table 61 Case Study 5 Determinants, Indicators and Actual Values

First Level Determinants	Second Level Indicators	Unit	Symbol	Actual Value
Technology	Number of Technology	per	X1	7
	Maturity of Technology	1-4	X2	1
Physical Characteristics	Number of components	per	X3	88
	Volume	m3	X4	0,9
Organization	People	person	X5	10
	Departments	per	X6	5
	Information transfer	per file	X7	40
	Resource allocation	1-4	X8	1
Environment	Number of suppliers and customer	per	X9	3
	Requlations and standards	per file	X10	21
	Market and competitions	1-4	X11	1

Table 62 Case Study 5 Weights of Indicators

First Level Determinants	r_i	r_i'	f_i	Second Level Indicators	$r_{i,j}$	$r_{i,j}$'	η_j
Technology	1.5	2.73	0.3985	Number of Technology	1.5	1.5	0.2391
				Maturity of Technology		1	0.1594
Physical Characteristics	1.4	1.82	0.2657	Number of components	1.5	1.5	0.1594
				Volume		1	0.1063
Organization	1.3	1.3	0.1898	People	1.3	2.535	0.0689
				Departments	1.3	1.95	0.053
				Information transfer	1.5	1.5	0.0408
				Resource allocation		1	0.0272
Environment	1.4	1	0.146	Number of suppliers and customer	1.3	1.82	0.063
				Requlations and standards	1.4	1.4	0.0484
				Market and competitions		1	0.0346

Table 63 Case Study 5 Indicators and Grey Clusters

X_j	Actual Value	Weight (η_j)	Low $[a^1, a^2]$	Whiten (λ_j^1)	Moderate $[a^2, a^3]$	Whiten (λ_j^2)	High $[a^3, a^4]$	Whiten (λ_j^3)
X1	7	0.2391	[2, 5]	3,50	[5, 15]	10,00	[15, 30]	22,50
X2	1	0.1594	[1, 2]	1,50	[2, 3]	2,50	[3, 4]	3,50
X3	88	0.1594	[10, 30]	20,00	[30, 80]	55,00	[80, 300]	190,00
X4	0,9	0.1063	[0.2, 0.5]	0,35	[0.5, 1]	0,75	[1, 3]	2,00
X5	10	0.0689	[5, 20]	12,50	[20, 50]	35,00	[50, 150]	100,00
X6	5	0.053	[1, 5]	3,00	[5, 10]	7,50	[10, 30]	20,00
X7	40	0.0408	[5, 20]	12,50	[20, 50]	35,00	[50, 150]	100,00
X8	1	0.0272	[1, 2]	1,50	[2, 3]	2,50	[3, 4]	3,50
X9	3	0.063	[1, 10]	5,50	[10, 30]	20,00	[30, 100]	65,00
X10	21	0.0484	[2, 5]	3,50	[5, 10]	7,50	[10, 20]	15,00
X11	1	0.0346	[1, 2]	1,50	[2, 3]	2,50	[3, 4]	3,50

Table 64 Case Study 5 Grey Clusters of Product Complexity

X_j	Weight (η_j)	$f_j^1(x)$	$f_j^2(x)$	$f_j^3(x)$
X1	0.2391	0.696	0.625	0.114
X2	0.1594	0.667	0	0
X3	0.1594	0	0.865	0.363
X4	0.1063	0.154	0.933	0.267
X5	0.0689	0.8	0.167	0
X6	0.053	0.714	0.615	0
X7	0.0408	0.267	0.957	0.25
X8	0.0272	0.667	0	0
X9	0.063	0.545	0.105	0
X10	0.0484	0	0	0.4
X11	0.0346	0.667	0	0
Grey Coefficient		**σ^1 (Low)**	**σ^2 (Mod.)**	**σ^3 (High)**
		0.468	**0.476**	0.143

CASE STUDY 6 (PROJECT B4):

Table 65 Case Study 6 Determinants, Indicators and Actual Values

First Level Determinants	Second Level Indicators	Unit	Symbol	Actual Value
Technology	Number of Technology	per	X1	8
	Maturity of Technology	1-4	X2	1
Physical Characteristics	Number of components	per	X3	61
	Volume	m3	X4	0,7
Organization	People	person	X5	8
	Departments	per	X6	5
	Information transfer	per file	X7	56
	Resource allocation	1-4	X8	1
Environment	Number of suppliers and customer	per	X9	3
	Requlations and standards	per file	X10	24
	Market and competitions	1-4	X11	1

Table 66 Case Study 6 Weights of Indicators

First Level Determinants	r_i	r_i'	f_i	Second Level Indicators	$r_{i,j}$	$r_{i,j}$'	η_j
Technology	1.5	2.73	0.3985	Number of Technology	1.5	1.5	0.2391
				Maturity of Technology		1	0.1594
Physical Characteristics	1.4	1.82	0.2657	Number of components	1.4	1.4	0.155
				Volume		1	0.1107
Organization	1.3	1.3	0.1898	People	1.3	2.535	0.0689
				Departments	1.3	1.95	0.053
				Information transfer	1.5	1.5	0.0408
				Resource allocation		1	0.0272
Environment	1.4	1	0.146	Number of suppliers and customer	1.3	1.95	0.064
				Requlations and standards	1.5	1.5	0.0492
				Market and competitions		1	0.0328

Table 67 Case Study 6 Indicators and Grey Clusters

X_j	Actual Value	Weight (η_j)	Low $[a^1, a^2]$	Whiten (λ_j^1)	Moderate $[a^2, a^3]$	Whiten (λ_j^2)	High $[a^3, a^4]$	Whiten (λ_j^3)
X1	8	0.2391	[2, 5]	3,50	[5, 15]	10,00	[15, 30]	22,50
X2	1	0.1594	[1, 2]	1,50	[2, 3]	2,50	[3, 4]	3,50
X3	61	0.155	[10, 30]	20,00	[30, 80]	55,00	[80, 300]	190,00
X4	0,7	0.1107	[0.2, 0.5]	0,35	[0.5, 1]	0,75	[1, 3]	2,00
X5	8	0.0689	[5, 20]	12,50	[20, 50]	35,00	[50, 150]	100,00
X6	5	0.053	[1, 5]	3,00	[5, 10]	7,50	[10, 30]	20,00
X7	56	0.0408	[5, 20]	12,50	[20, 50]	35,00	[50, 150]	100,00
X8	1	0.0272	[1, 2]	1,50	[2, 3]	2,50	[3, 4]	3,50
X9	3	0.064	[1, 10]	5,50	[10, 30]	20,00	[30, 100]	65,00
X10	24	0.0492	[2, 5]	3,50	[5, 10]	7,50	[10, 20]	15,00
X11	1	0.0328	[1, 2]	1,50	[2, 3]	2,50	[3, 4]	3,50

Table 68 Case Study 6 Grey Clusters of Product Complexity

X_j	Weight (η_j)	$f_j^1(x)$	$f_j^2(x)$	$f_j^3(x)$
X1	0.2391	0.609	0.75	0.171
X2	0.1594	0.667	0	0
X3	0.155	0.317	0.976	0.194
X4	0.1107	0.462	0.909	0.133
X5	0.0689	0.64	0.1	0
X6	0.053	0.714	0.615	0
X7	0.0408	0	0.817	0.45
X8	0.0272	0.667	0	0
X9	0.064	0.545	0.105	0
X10	0.0492	0	0	0.1
X11	0.0328	0.667	0	0
Grey Coefficient		**σ^1 (Low)**	**σ^2 (Mod.)**	**σ^3 (High)**
		0.509	**0.511**	0.109

CASE STUDY 7 (PROJECT B6):

Table 69 Case Study 7 Determinants, Indicators and Actual Values

First Level Determinants	Second Level Indicators	Unit	Symbol	Actual Value
Software Characteristics	COSMIC FFP Size	cfsu	X1	2320
	Interactions with other system	1-4	X2	4
Technology	Number of Technology	per	X3	27
	Maturity of Technology	1-4	X4	3
Organization	People	person	X5	60
	Departments	per	X6	7
	Information transfer	per file	X7	111
	Resource allocation	1-4	X8	1
Environment	Number of suppliers and customer	per	X9	4
	Requlations and standards	per file	X10	87
	Market and competitions	1-4	X11	1

Table 70 Case Study 7 Weights of Indicators

First Level Determinants	r_i	r_i’	f_i	Second Level Indicators	$r_{i,j}$	$r_{i,j}$’	η_j
Software Characteristics	1.4	2.352	0.3774	COSMIC FFP Size	1.5	1.5	0.2264
				Interactions with other system		1	0.151
Technology	1.4	1.68	0.2696	Number of Technology	1.5	1.5	0.1617
				Maturity of Technology		1	0.1078
Organization	1.2	1.2	0.1926	People	1.4	2.352	0.0704
				Departments	1.2	1.68	0.0503
				Information transfer	1.4	1.4	0.0419
				Resource allocation		1	0.0299
Environment	1	1	0.1605	Number of suppliers and customer	1.2	1.8	0.0672
				Requlations and standards	1.5	1.5	0.056
				Market and competitions		1	0.0373

Table 71 Case Study 7 Indicators and Grey Clusters

X_j	Actual Value	Weight (η_j)	Low [a^1, a^2]	Whiten (λ_j^1)	Moderate [a^2, a^3]	Whiten (λ_j^2)	High [a^3, a^4]	Whiten (λ_j^3)
X1	2320	0.2264	[163, 334]	248.50	[334, 870]	602.00	[870, 1090]	980.00
X2	4	0.151	[1, 2]	1.50	[2, 3]	2.50	[3, 4]	3.50
X3	27	0.1617	[2, 5]	3.50	[5, 15]	10.00	[15, 30]	22.50
X4	3	0.1078	[1, 2]	1.50	[2, 3]	2.50	[3, 4]	3.50
X5	60	0.0704	[5, 20]	12.50	[20, 50]	35.00	[50, 150]	100.00
X6	7	0.0503	[1, 5]	3.00	[5, 10]	7.50	[10, 30]	20.00
X7	111	0.0419	[5, 20]	12.50	[20, 50]	35.00	[50, 150]	100.00
X8	1	0.0299	[1, 2]	1.50	[2, 3]	2.50	[3, 4]	3.50
X9	4	0.0672	[1, 10]	5.50	[10, 30]	20.00	[30, 100]	65.00
X10	87	0.056	[2, 5]	3.50	[5, 10]	7.50	[10, 20]	15.00
X11	1	0.0373	[1, 2]	1.50	[2, 3]	2.50	[3, 4]	3.50

Table 72 Case Study 7 Grey Clusters of Product Complexity

X_j	Weight (η_j)	$f_j^1(x)$	$f_j^2(x)$	$f_j^3(x)$
X1	0.2264	0	0	0
X2	0.151	0	0	0.667
X3	0.1617	0	0.15	0.4
X4	0.1078	0	0.667	0.667
X5	0.0704	0	0.783	0.5
X6	0.0503	0.429	0.923	0.133
X7	0.0419	0	0.339	0.89
X8	0.0299	0.667	0	0
X9	0.0672	0.727	0.158	0
X10	0.056	0	0	0
X11	0.0373	0.667	0	0
Grey Coefficient		**σ^1 (Low)**	**σ^2 (Mod.)**	**σ^3 (High)**
		0.115	0.222	**0.318**

CASE STUDY 8 (PROJECT C2):

Table 73 Case Study 8 Determinants, Indicators and Actual Values

First Level Determinants	Second Level Indicators	Unit	Symbol	Actual Value
Software Characteristics	COSMIC FFP Size	cfsu	X1	386
	Interactions with other system	1-4	X2	3
Technology	Number of Technology	per	X3	7
	Maturity of Technology	1-4	X4	1
Organization	People	person	X5	9
	Departments	per	X6	2
	Information transfer	per file	X7	12
	Resource allocation	1-4	X8	1
Environment	Number of suppliers and customer	per	X9	3
	Requlations and standards	per file	X10	5
	Market and competitions	1-4	X11	2

Table 74 Case Study 8 Weights of Indicators

First Level Determinants	r_i	r_i'	f_i	Second Level Indicators	$r_{i,j}$	$r_{i,j}$'	η_j
Software Characteristics	1.5	2.7	0.403	COSMIC FFP Size	1.5	1.5	0.2418
				Interactions with other system		1	0.1612
Technology	1.5	1.8	0.2687	Number of Technology	1.4	1.4	0.1567
				Maturity of Technology		1	0.1119
Organization	1.2	1.2	0.1791	People	1.3	2.028	0.0617
				Departments	1.2	1.56	0.0475
				Information transfer	1.3	1.3	0.0395
				Resource allocation		1	0.0304
Environment	1	1	0.1493	Number of suppliers and customer	1.2	1.68	0.0615
				Requlations and standards	1.4	1.4	0.0512
				Market and competitions		1	0.0366

Table 75 Case Study 8 Indicators and Grey Clusters

X_j	Actual Value	Weight (η_j)	Low $[a^1, a^2]$	Whiten (λ_j^1)	Moderate $[a^2, a^3]$	Whiten (λ_j^2)	High $[a^3, a^4]$	Whiten (λ_j^3)
X1	386	0.2418	[163, 334]	248.50	[334, 870]	602.00	[870, 1090]	980.00
X2	3	0.1612	[1, 2]	1.50	[2, 3]	2.50	[3, 4]	3.50
X3	7	0.1567	[2, 5]	3.50	[5, 15]	10.00	[15, 30]	22.50
X4	1	0.1119	[1, 2]	1.50	[2, 3]	2.50	[3, 4]	3.50
X5	9	0.0617	[5, 20]	12.50	[20, 50]	35.00	[50, 150]	100.00
X6	2	0.0475	[1, 5]	3.00	[5, 10]	7.50	[10, 30]	20.00
X7	12	0.0395	[5, 20]	12.50	[20, 50]	35.00	[50, 150]	100.00
X8	1	0.0304	[1, 2]	1.50	[2, 3]	2.50	[3, 4]	3.50
X9	3	0.0615	[1, 10]	5.50	[10, 30]	20.00	[30, 100]	65.00
X10	5	0.0512	[2, 5]	3.50	[5, 10]	7.50	[10, 20]	15.00
X11	2	0.0366	[1, 2]	1.50	[2, 3]	2.50	[3, 4]	3.50

Table 76 Case Study 8 Grey Clusters of Product Complexity

X_j	Weight (η_j)	$f_j^1(x)$	$f_j^2(x)$	$f_j^3(x)$
X1	0.2418	0.779	0.508	0.08
X2	0.1612	0	0.667	0.667
X3	0.1567	0.696	0.625	0.114
X4	0.1119	0.667	0	0
X5	0.0617	0.72	0.133	0
X6	0.0475	0.667	0.154	0
X7	0.0395	0.96	0.233	0
X8	0.0304	0.667	0	0
X9	0.0615	0.545	0.105	0
X10	0.0512	0.769	0.545	0
X11	0.0366	0.667	0.667	0
Grey Coefficient		**σ^1 (Low)**	**σ^2 (Mod.)**	**σ^3 (High)**
		0.604	0.412	0.145

CASE STUDY 9 (PROJECT C3):

Table 77 Case Study 9 Determinants, Indicators and Actual Values

First Level Determinants	Second Level Indicators	Unit	Symbol	Actual Value
Software Characteristics	COSMIC FFP Size	cfsu	X1	415
	Interactions with other system	1-4	X2	2
Technology	Number of Technology	per	X3	7
	Maturity of Technology	1-4	X4	1
Organization	People	person	X5	10
	Departments	per	X6	2
	Information transfer	per file	X7	11
	Resource allocation	1-4	X8	1
Environment	Number of suppliers and customer	per	X9	3
	Requlations and standards	per file	X10	5
	Market and competitions	1-4	X11	2

Table 78 Case Study 9 Weights of Indicators

First Level Determinants	r_i	r_i'	f_i	Second Level Indicators	$r_{i,j}$	$r_{i,j}$'	η_j
Software Characteristics	1.5	2.7	0.403	COSMIC FFP Size	1.5	1.5	0.2418
				Interactions with other system		1	0.1612
Technology	1.5	1.8	0.2687	Number of Technology	1.4	1.4	0.1567
				Maturity of Technology		1	0.1119
Organization	1.2	1.2	0.1791	People	1.3	2.028	0.0617
				Departments	1.2	1.56	0.0475
				Information transfer	1.3	1.3	0.0395
				Resource allocation		1	0.0304
Environment	1	1	0.1493	Number of suppliers and customer	1.2	1.68	0.0615
				Requlations and standards	1.4	1.4	0.0512
				Market and competitions		1	0.0366

Table 79 Case Study 9 Indicators and Grey Clusters

X_j	Actual Value	Weight (η_j)	Low [a^1, a^2]	Whiten (λ_j^1)	Moderate [a^2, a^3]	Whiten (λ_j^2)	High [a^3, a^4]	Whiten (λ_j^3)
X1	415	0.2418	[163, 334]	248.50	[334, 870]	602.00	[870, 1090]	980.00
X2	2	0.1612	[1, 2]	1.50	[2, 3]	2.50	[3, 4]	3.50
X3	7	0.1567	[2, 5]	3.50	[5, 15]	10.00	[15, 30]	22.50
X4	1	0.1119	[1, 2]	1.50	[2, 3]	2.50	[3, 4]	3.50
X5	10	0.0617	[5, 20]	12.50	[20, 50]	35.00	[50, 150]	100.00
X6	2	0.0475	[1, 5]	3.00	[5, 10]	7.50	[10, 30]	20.00
X7	11	0.0395	[5, 20]	12.50	[20, 50]	35.00	[50, 150]	100.00
X8	1	0.0304	[1, 2]	1.50	[2, 3]	2.50	[3, 4]	3.50
X9	3	0.0615	[1, 10]	5.50	[10, 30]	20.00	[30, 100]	65.00
X10	5	0.0512	[2, 5]	3.50	[5, 10]	7.50	[10, 20]	15.00
X11	2	0.0366	[1, 2]	1.50	[2, 3]	2.50	[3, 4]	3.50

Table 80 Case Study 9 Grey Clusters of Product Complexity

X_j	Weight (η_j)	$f_j^1(x)$	$f_j^2(x)$	$f_j^3(x)$
X1	0.2418	0.732	0.574	0.125
X2	0.1612	0.667	0.667	0
X3	0.1567	0.696	0.625	0.114
X4	0.1119	0.667	0	0
X5	0.0617	0.8	0.167	0
X6	0.0475	0.667	0.154	0
X7	0.0395	0.88	0.2	0
X8	0.0304	0.667	0	0
X9	0.0615	0.545	0.105	0
X10	0.0512	0.769	0.545	0
X11	0.0366	0.667	0.667	0
Grey Coefficient		**σ^1 (Low)**	**σ^2 (Mod.)**	**σ^3 (High)**
		0.701	0.428	0.048

CURRICULUM VITAE

Semra Yılmaz Taştekin was born in Adiyaman in 1976. She graduated from Gaziantep University, Department of Electrical and Electronics Engineering in 1997. She received her M.S. degree from METU, Informatics Institute in 2003. She started her PhD education in Informatics Institute in 2003. She is System Design Leader of IFF (Identification of Friend or Foe) Projects at Aselsan Inc.

Publications:

[1] Semra Yilmaz. Performance Evaluation of Routing Protocols in Wireless Ad Hoc Networks with Service Differentiation. M.Sc. Thesis, 2003.

[2] Semra Yilmaz & Yusuf Murat Erten. (June, 2008). TCP Performance in Wireless Ad Hoc Networks. *8th International Symposium on Computer Networks.*

[3] Semra Yilmaz Taştekin, Yusuf Murat Erten & Semih Bilgen. (November 2012). Product Development Time Improvement with Requirements Reuse. *7th International Conference on Software Engineering Advances.*

[4] Semra Yilmaz Taştekin, Yusuf Murat Erten & Semih Bilgen. (September 2013). Software Product Complexity Estimation Using Grey Measurement. *39th Euromicro Conference on Software Engineering and Advanced Applications.*

[5] Semra Yilmaz Taştekin, Yusuf Murat Erten & Semih Bilgen. Accounting for Product Similarity in Software Project Duration Estimation. *Submitted to International Journal on Software Engineering and Knowledge Engineering.*

TEZ FOTOKOPİSİ İZİN FORMU

ENSTİTÜ

Fen Bilimleri Enstitüsü ☐

Sosyal Bilimler Enstitüsü ☐

Uygulamalı Matematik Enstitüsü ☐

Enformatik Enstitüsü ☑

Deniz Bilimleri Enstitüsü ☐

YAZARIN

Soyadı : YILMAZ TAŞTEKİN
Adı : SEMRA
Bölümü : BİLİŞİM SİSTEMLERİ

TEZİN ADI (İngilizce) : Use of Project Similarity for Software Development Time Estimation

TEZİN TÜRÜ : Yüksek Lisans ☐ Doktora ☑

1. Tezimin tamamından kaynak gösterilmek şartıyla fotokopi alınabilir. ☐
2. Tezimin içindekiler sayfası, özet, indeks sayfalarından ve/veya bir bölümünden kaynak gösterilmek şartıyla fotokopi alınabilir. ☐
3. Tezimden bir (1) yıl süreyle fotokopi alınamaz. ☑

TEZİN KÜTÜPHANEYE TESLİM TARİHİ :