AN AUTOMATED DEFECT DETECTION APPROACH FOR COSMIC FUNCTIONAL SIZE MEASUREMENT METHOD

A THESIS SUBMITTED TO THE GRADUATE SCHOOL OF INFORMATICS
OF THE MIDDLE EAST TECHNICAL UNIVERSITY

BY

GÖKÇEN YILMAZ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
IN
THE DEPARTMENT OF INFORMATION SYSTEMS

AUGUST 2012

Approval of the Graduate School of Informatics

____

Prof. Dr. Nazife Baykal
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

____

Prof. Dr. Yasemin Yardımcı Çetin
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

____

Prof. Dr. Onur Demirörs
Supervisor

Examining Committee Members

Assoc. Prof. Dr. Altan Koçyiğit	(METU, II)	______________________
Prof. Dr. Onur Demirörs	(METU, II)	______________________
Dr. Ali Arifoğlu	(METU, II)	______________________
Assist. Prof. Dr. Aysu Betin Can	(METU, II)	______________________
Dr. N. Alpay Karagöz	(INNOVA)	______________________

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 material and results that are not original to this work.

Name, Last name: Gökçen Yılmaz

Signature: _________________

ABSTRACT

AN AUTOMATED DEFECT DETECTION APPROACH FOR COSMIC FUNCTIONAL SIZE MEASUREMENT METHOD

Yılmaz, Gökçen
M.Sc., Department of Information Systems
Supervisor: Prof. Dr. Onur Demirörs

August 2012, 123 Pages

Software size measurement provides a basis for software project management and plays an important role for its activities such as project management estimations, process benchmarking, and quality control. As size can be measured with functional size measurement (FSM) methods in the early phases of the software projects, functionality is one of the most frequently used metric. On the other hand, FSMs are being criticized by being subjective.

The main aim of this thesis is increasing the accuracy of the measurements, by decreasing the number of defects concerning FSMs that are measured by COSMIC FSM method. For this purpose, an approach that allows detecting defects of FSMs automatically is developed. During the development of the approach, first of all error classifications are established. To detect defects of COSMIC FSMs automatically, COSMIC FSM Defect Detection Approach (DDA) is proposed. Later, based on the proposed approach, COSMIC FSM DDT (DDT) is developed.

Keywords: COSMIC, Functional Size Measurement, COSMIC Defect Categories, COSMIC FSM Defect Detection Approach, COSMIC FSM DDT.

ÖZ

COSMIC İŞLEVSEL BÜYÜKLÜK ÖLÇÜM METODU İÇİN BİR OTOMATİK HATA YAKALAMA YAKLAŞIMI

Yılmaz, Gökçen
Yüksek Lisans, Bilişim Sistemleri Bölümü
Tez Yöneticisi: Prof. Dr. Onur Demirörs

Ağustos 2012, 123 Sayfa

Yazılım büyüklük ölçümü yazılım proje yönetimi için temel oluşturur ve proje yönetim kestirimleri, süreç kıyaslama ve kalite kontrol gibi yazılım proje yönetimi aktiviteleri için önemli bir rol oynar. Büyüklük, İşlevsel Yazılım Büyüklüğü metotları ile yazılım projelerinin erken sahalarında ölçülebilir olduğu için, fonksiyonellik sıklıkla kullanılan metriklerden birisidir. Diğer bir taraftan, işlevsel büyüklük ölçümleri (İBÖ) öznel olmaları yönü ile tartışılmaktadır.

Bu tezin ana amacı, COSMIC metodu ile ölçülmüş işlevsel yazılım büyüklüklerindeki hata sayısını azaltarak, ölçümlerin doğruluğunu arttırmaktır. Bu amaçla, İBÖ'lerinin hatalarını otomatik olarak tespit edilmesini sağlayan bir yaklaşım öne sürülmüştür. Yaklaşımın geliştirilmesi sırasında, öncelikle hata sınıfları oluşturulmuştur. COSMIC büyüklük ölçümlerin hatalarını otomatik olarak yakalamak amacı ile COSMIC İBÖ Hata Yakalama Yaklaşımı öne sürülmüştür. Daha sonra, bu yaklaşıma göre, COSMIC İBÖ Hata Yakalama Aracı geliştirilmiştir.

Anahtar Kelimeler: COSMIC, İşlevsel Büyüklük Ölçümü, COSMIC Hata Kategorileri, COSMIC İBÖ Hata Yakalama Yaklaşımı, COSMIC İBÖ Hata Yakalama Aracı.

ACKNOWLEDGEMENT

I would like to offer my sincere appreciation to my advisor, Prof. Dr. Onur Demirörs for his guidance, continuous support, encouragement and patience throughout my thesis. I have learned much from his feedbacks.

I would also like to thank to Seçkin TUNALILAR who had spent her limited time to answer my questions. Her ideas enlightened me while I was developing defect detection procedures of my thesis.

I would like to express my special thanks to Barış ÖZKAN for his patience while answering my questions especially about the CUBIT.

I am grateful to my destiny friends, M. Erhan UYAR and Mina NABII for always helping me whenever I needed support. Mina NABII, thank you very much for the Grails sessions. M. Erhan UYAR, thank you very much for helping me to handle the huge amount of data.

My friends Özge GÜRBÜZ, Serhat PEKER, Ayşegül ÖZKAYA, and Davut ÇAVDAR were always with me whenever I felt overwhelmed. Thank you very much for your great friendship.

I would like to thank to my friends, Görkem DUMAN and Serkan ORHON for helping me to solve my programming problems with their high level of coding experience.

I would also thank to Emin Birey SOYER for accompanying me with his puzzle during my thesis writing process.

My colleague Şeyma KÜÇÜKÖZER helped me so much by conducting IS100 tasks smoothly with a little explanation. Thank you very much for allowing me to keep my thesis concentration on the highest level.

I would appreciate to staff of Informatics Institute for helping me in every bureaucratic task.

AND finally, I would like to express my deepest gratitude to my father Bekir YILMAZ, my mother Kadriye YILMAZ and my brother Çağrı YILMAZ for their endless support and love throughout my life. I am really happy to be your daughter and I owe to you much.

To my family…

TABLE OF CONTENTS

LIST OF TABLES

TABLE OF FIGURES

LIST OF ACRONYMS

FSM	Functional Size Measurement
İBÖ	İşlevsel Büyüklük Ölçümü
COSMIC	The Common Software Measurement International Consortium
EC	Error Category
DDT	Defect Detection Tool
DDA	Defect Detection Approach
MIS	Management Information Systems
FUR	Functional User Requirement
FP	Functional Process
DM	Data Movement
DG	Data Group
OOI	Object of Interest
DA	Data Attribute
DC	Defect Cause
RS	Reference Source
BFC	Base Functional Component
MPR	Measurement Process Related
MR	Measurement Related
SAR	Software Artifact Related
NI	Newly Identified
COSMIC MM	COSMIC Measurement Manual
COSMIC AG	COSMIC Accuracy Guideline
SLOC	Source Lines of Code
DDL	Defect Detection Logic

CHAPTER 1

INTRODUCTION

Software size measurements are fundamental input for software project management activities such as effort and cost estimation, project monitoring, project control and quality control. Thus, accuracy and reliability of software size measurements is very crucial.

Among various approaches, FSM methods are widely used in industry as it is easier to apply in the early phases of software life cycle. From the emergence of the term "functionality" many FSM methods such as The Common Software Measurement International Consortium (COSMIC) method (ISO/IEC, 2003b), IFPUG (ISO/IEC, 2003c), and MARK II (ISO/IEC, 2002c) and related standards have been proposed. For measuring the functional size of the software, these methods use software requirements specification artifacts as an input and maps requirements presented in natural language to the elements of the FSM methods. Thus, many researches showed that although FSMs are performed according to these well-known standards, there are many factors that cause variations in measurement results.

Important points of the related researches about increasing the reliability of FSMs are given in Chapter 2.

In this thesis it is hypothesized that increasing the reliability of FSMs can be provided by eliminating measurement defects in measurement reports. For this purpose we aimed to develop a tool that detects defects of FSMs automatically. In the context of the study the COSMIC FSM method is chosen, since it is one of the most widely used FSM method. The tool is planned to be developed integrated into CUBIT[1] which is a web based tool that has measurement data of many software projects.

[1] http://smrg.ii.metu.edu.tr/cubit/auth/login?targetUri=%2F

To achieve the aim of the study, first of all, we identified the error categories of COSMIC FSMs, and defined them clearly. In order to detect defects of COSMIC FSMs automatically; a COSMIC DDT is built. To improve the defect detection effectiveness of the Tool we proposed a COSMIC FSM DDA. Later, according to the proposed approach we improved the Tool.

Three case studies are conducted during the development of the tool and the approach. While first case study is the exploratory case study, second and third case studies are performed to validate the COSMIC FSM DDA and CUBIT COSMIC FSM DDT . Detailed information about the case studies is given under Chapter 4.

At the end, case studies showed that such an approach and tool allow measurers to find the exact point of defects and eliminate them from the measurement reports with a small amount of effort.

1.1 Problem Statement & Motivation

FSM methods are presumable, reliably performed once the Software Requirements Specification (SRS) work product is available. Although the SRS is written and FSMs are performed according to well-known standards, two different FSMs that belong to the same software system may vary.

In researches, that have been conducted to investigate the origins of the accuracy problems of FSMs, manual expert review mechanism has been used. However, since it is time consuming, it is not preferred as a defect detection technique.

Suggestions, that were derived to solve the subjectivity problem of FSMs, are not adequate to increase the reliability of FSMs. One of these solutions is a defect detection checklist that is given in the guideline “Guideline for Assuring the Accuracy of Measurements” (COSMIC, 2011). It allows measurers controlling their reports with respect to mostly encountered COSMIC errors. However, these errors are defined generally. An example that is taken from the guideline is given as follows;

“Check the relevant part of the measurements, OOI sub-types.”

Detecting defects by using this checklist is inefficient and time consuming. In order to detect defects of COSMIC FSMs; includes error classifications and their detailed definitions should be maintained of the COSMIC FSMs. In order to overcome all of these problems a technique is required to be developed to detect the defects of FSMs automatically.

1.2 Methodology

The aim of this thesis is to increase the reliability of the COSMIC FSM by detecting the defects of COSMIC FSMs, as well as reducing the time and effort spent to detect defects of measurement reports by automating the defect detection process.

As the starting point of this thesis, a case study is performed to explore the error categories of the COSMIC FSMs. The error categories are identified, based on the most commonly made COSMIC errors which were investigated in the researches (Top, Demirors, & Ozkan, 2009), (Ungan, Demirörs, Top, & Özkan, 2009), (COSMIC, 2011). After the identification of error categories, each of them is defined clearly. Their definitions are given in Chapter 3.

The error categories, that have the possibility of detecting its defects automatically, are determined. By using expert review mechanism, patterns of these error categories are investigated and prototype of the COSMIC FSM DDT for MIS applications is developed. CUBIT COSMIC FSM DDT for MIS applications is integrated to CUBIT.

To detect defects effectively, a COSMIC FSM DDA for MIS applications is developed. By updating the prototype based on the Approach, we finalized the development of CUBIT COSMIC FSM DDT for MIS applications. To identify the defect detection effectiveness of the Tool, it is utilized for three different groups of COSMIC FSMs.

Finally, we controlled if the tool is valid for COSMIC FSMs of different application domains, such as COSMIC measurements of Real-Time and Embedded software projects

1.3 Overview

Next chapters are organized as follows; in Chapter 2 literature review is given. Related researches and especially the Guideline for Assuring the Accuracy of Measurements (COSMIC, 2011) are explained in detail.

COSMIC FSM DDA for MIS applications is explained in detail and definition of each EC is given in Chapter3.

Chapter 4 introduces detailed information about case studies as case study plan, implementation and their results.

Lastly in Chapter 5, conclusions of the study and future works are given.

CHAPTER 2

LITERATURE REVIEW

This chapter of thesis presents the findings of the literature review related to software size measurement methods and reliability studies for improving the accuracy of size measurements. Since this thesis focuses on improving the reliability of FSMs that are measured by COSMIC FSM method, rules of COSMIC FSM method are explained in detail. Later, reliability studies about FSMs are given in the following sections.

2.1. Software Size Measurement Methods

Throughout the history of software size measurement; developing a reasonable, objective, reliable, repeatable and easy to apply software size measurement method has been the main objective. For quantifying the size of software, several metrics are introduced and used.

First commonly used metric is the Source Lines of code (SLOC). SLOC is the most traditional and most widely used size metric (Fenton & Pfleeger, 1998). Although measuring by LOC is easy and objective, it is technology dependent. SLOCs are language specific; this creates a difficulty in comparing the size measurements of applications written in different languages (Gencel & Buglione & Demirors, 2006). Moreover, as SLOC size is obtained only after the source code is developed, it failed to meet the software project management needs in the early stages of a software project. It is difficult to relate SLOC with Functional User Requirements (FUR) in the early stages of the development.

Second widely used metric is the Function Points introduced by Allan Albrecht in 1979 as an IBM researcher. Function Points quantifies the size of software in terms of functionality (Albrecht, 1979).

Later, Albrecht performed a study with Gaffney and improved the methodology in 1984 (A. J. Albrecht & Gaffney, J.E., 1983). During 1980s and 1990s, researchers proposed varieties of original method by improving or extending the domain application of original method (Symons, 2001).

As the evolution of those methods, conceptual inconsistency among theoretical aspects of FPA methods occurred. In order to establish association between fundamental concepts, in1998, ISO/IEC published a standard ISO/IEC 14143-1 (ISO/IEC, 1998) in which BFCs are defined and FSM term was first coined. Later, other parts of the standard were published (ISO/IEC, 1998), (ISO/IEC, 2002a), (ISO/IEC, 2003a), (ISO/IEC, 2002b), (ISO/IEC, 2004), (ISO/IEC, 2005a). In 1999, COSMIC-FFP v2.0 was published. The main intension was that it was applicable to data-driven, event-driven software and contained the fundamental concept basis introduced in ISO/IEC 14143-1 (ISO/IEC, 1998). Methods proposed during 80s and 90s were classified as first generation. These methods work for restricted types of software. With publication of COSMIC FFP v2.0 in 1999 second generation FSM methods arise. Those methods can be applied to a wide range of software application domains (COSMIC, 2007).

Currently, there are five FSM methods that are certified by the International Organization for Standardization (ISO). These are; ISO/IEC 19761 (ISO/IEC, 2003b), ISO/IEC 20926 (ISO/IEC, 2003c), ISO/IEC 24570 (ISO/IEC, 2005b), ISO/IEC 29881 (ISO/IEC, 2008) and ISO/IEC 20968 (ISO/IEC, 2002c). In this study COSMIC FSM Method is taken as basis, since it is one of the most widely used and certified FSM method by ISO /IEC. Important concepts of each method are presented in Table 1 (Gencel & Demirors, 2008) briefly, but COSMIC FSM (ISO/IEC, 2003b) will be explained in detail in section 2.2.

Table 1: Important concepts of FSM methods (Gencel & Demirors, 2008)

Method	Generation	Based on	Applicable domains	Type of component of functionality	BFCs	Unit
IFPUG	1st gen.	FUR	MIS and assumes the use of traditional software development methodologies such as structured analysis and design.	Elementary process	External inputs, External outputs, External Inquiries, External Interface Files, Logical Interface Files	IFPUG FP
Mk II	1st gen.	FUR	Only Business Application software	Logical transaction	Input Data Element Types, Data Entity Types Referenced, Output Data Element Types	Mk II FP
COSMIC	2nd gen.	FUR	Business Application software, Real-time software, Hybrids of these 2	Functional Process	Data Movements(Entry, Read, Write, eXit)	COSMIC FP
NESMA	2nd gen.	FUR	Business Application software	Elementary process	External inputs, External outputs, External Inquiries, External Interface Files, Logical Interface Files	NESMA FP

continued on the next page

continued from the previous page

Method	Generation	Based on	Applicable domains	Type of component of functionality	BFCs	Unit
FISMA	2nd gen.	FUR	Business Application software	Interactive End-User Navigation and Query Services	Function designator q1, Log-in, log-out functions q2, Function List q3, Selection lists q4, Data inquiries q5, Generation indicators q6, Browsing lists q7	FISMA FP
				Interactive End-User Input Services	1-functional i1, 2-functional i2, 3-functional i3	
				Non-interactive End-User Output Services	Forms o1, Reports o2, Emails or text, messages o3, Monitor screens o4	
				Interface Services to Other Applications	Messages in f1, Batch records f2, Signals in f3	
				Interface Services from Other Applications	Messages out t1, Batch records t2, Signals out t3	
				Data Storage Services	Entities/classes d1, Other records d2	
				Algorithmic & Manipulation Services	Security routines a1, Calculations a2, Simulations a3, Formatting alg. A4, Db cleaning a5, Other routines a6	

2.2. COSMIC FSM Method

COSMIC FSM DDA, which is indicated in the next chapter, makes some modifications on some of the basic concepts of COSMIC FSM Method (ISO/IEC, 2003b). For a better understanding of the Approach and its modifications, basic concepts of COSMIC FSM Method should be clarified. Thus, while COSMIC FSM Method will not be explained in detail, descriptions of important basic concepts of the Method will be explained in detail.

According to the COSMIC Measurement Method (ISO/IEC, 2003b), functional size is purely based on the functionality. It ignores any technical or quality requirements. COSMIC FSM is purely based on the software artifacts, especially Software Requirements Specification (SRS) document. COSMIC FSM uses 5 mandatory functional size components during the measurement process. These are; “Functional User Requirement” (FUR), “Functional Processes” (FP), “Object of Interest” (OOI), “Data Group” (DG) and “Data Movement” (DM). Method has one more optional component which is Data Attribute (DA). When measuring size of software projects, identifying FUR, FP, OOI, DG, and DM is mandatory. However, DA is left as an optional component for the measurement process.

COSMIC FSM (ISO/IEC, 2003b) has three phases as; “Measurement Strategy Phase”, “Mapping Phase” and “Measurement Phase. Definition of each BFC will be explained in detail in related phase.

In the Measurement Strategy Phase, first of all, the purpose and the scope of the measurement should be identified. Secondly, Functional Users should be identified and based on these; FURs should be extracted from the artifacts of the software to be measured. FUR is a subset of the User Requirements that describe what the software shall do, in terms of tasks and services. Functional Requirements include but not limited to: (COSMIC, 2005)

- Data transfer (for instance input customer data)
- Data transformation (for instance Calculate bank interest)
- Data storage (for instance Store customer order)
- Data retrieval (for instance List current employees)

Additionally, in the Measurement Strategy Phase, level of granularity should be determined.

In the Mapping Phase, the FUR must be mapped to three main concepts which are FP, OOI and DG, of the COSMIC MM.

FP is defined as: “an elementary component of a set of FURs comprising a unique, cohesive and independently executable set of DMs (cannot be sub divided). It is triggered by a DM (an Entry) from a functional user that informs the piece of software that the functional user has identified a triggering event. It is totally complete when it has executed.” (ISO/IEC, 2003b).

OOI is, from the point of view of the Functional Users, any “thing” that is identified to process and/or store data (ISO/IEC, 2003b).

DG is a distinct, non-empty, non-ordered and non-redundant collection of DAs related with one OOI (ISO/IEC, 2003b).

DA is the smallest part of information within an identified DG and carrying a meaning from the perspective of the software’s Functional User (ISO/IEC, 2003b).

For instance; a Work Order Management system that stores a lot of data about work orders and employees, e.g. employee_id, employee_name, employee_ role, workorder_id, workorder_name, workOrder_description and etc., has many OOIs.

“Employee” and “Work Order” are clearly OOIs to many functional users e.g. employees of the company. OOI “Employee” stores data about employees of the company and the OOI “Work Order” stores data about the work orders that are created within the company.

Each of the given attributes employee_id, employee_name, employee_ role, workorder_id, workorder_name, workOrder_description is an example of DAs.

Collection of these (two or more) DAs forms a group of DAs, and is an example of DG.

In the Measurement Phase, size is measured by decomposing the FPs into DMs and data manipulation. By knowing the FPs and DGs, the individual DMs of the FPs can be identified as the basis for measuring the functional size (ISO/IEC, 2003b). Data Manipulation is the anything that happens to data other than movement of the data. However, DMs of a FP are assumed also to represent the data manipulation of the FP (ISO/IEC, 2003b).

DM is a BFC which moves a single DG type. As depicted in Figure 1, BFC of COSMIC Measurement Method is DM which has 4 different subtypes as Entry, Read, Write, and eXit.

DMs cross the boundary between the functional user and the application. DM subtypes are described as follows: (ISO/IEC, 2003b)

- An Entry moves a DG from a functional user across the boundary into the FP where it is required.
- An eXit moves a DG from a FP across the boundary to the functional user that requires it
- A Read moves a DG from persistent storage within reach of the FP which requires it
- A Write moves a DG lying inside a FP to persistent storage

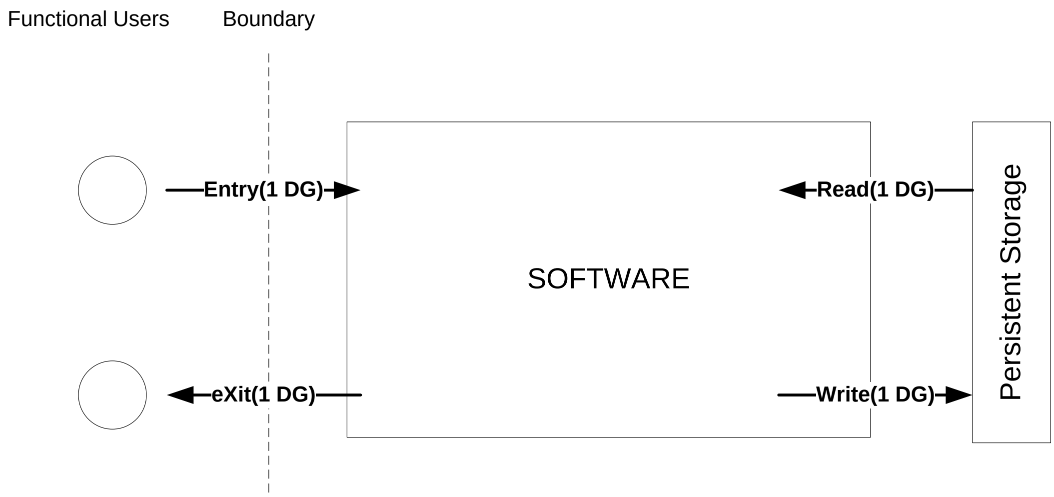


Figure 1: Relationship between DM, FPs and DGs and DM types (Top, 2008)

Functional size of the software is calculated in CFP units, by aggregating the functional sizes of individual DMs.

$$Size\ CFP(functional\ process_{i)}) = \sum size(Entries_i) + \sum size(eXists_i) + \sum size(Reads_{i)}) + \sum size\ (Writes_{i)})$$

2.2.1. Reliability of FSMs

In order not to underestimate or overestimate effort, cost and budget of software projects, functional size of applications should be measured accurately. Measurement errors are essential since they cause poor management which usually results in runaway projects(Glass, 2002).

Reliability of FSMs is one of the important topics of software engineering research area. Although FSM is a widely used software size measurement method, it is usually criticized as being subjective. Therefore a number of reliability case studies and reliability improvement research studies are conducted in order to investigate the origins of the problems that decrease the accuracy of FSMs.

Study of Low and Jeffery (1990)

In 1990, Low and Jeffery performed an experimental study about inter-rater reliability. This study was performed in order to highlight that two different measurers measuring the same software system by using the same measurement method, may results in different function point counts. According to the experiment results, consistency of function points counts "appears to be within the 30 percent reported by Rudolph." using the same method (Low & Jeffery, 1990).

Study of Kemerer and Porter (1992)

By using IFPUG method, Kemerer and Porter, conducted a two-phased case study to identify the sources and magnitudes of reliability of FP variations. The first phase was composed of a field survey to investigate the sources of the FP counting variations. Second phase of the study was designed as a detailed case study to estimate the magnitude of the effects of these sources.

Survey was developed based on the IFPUG Counting Practices Manual 3.0, and composed of 16 questions. Later this survey was mailed to 84 volunteer member organizations of IFPUG and the results were collected. If the 50% responses were different than rules of CPM 3.0, then the topic was selected for further study. 11 topics were selected as case study variants and to calculate the magnitude of the effect of these variants, 3 different software products were measured.

For each software product, based on each variant, FP variance analysis with respect to actual size was performed. After collecting results of each case, variants were clustered into 5 categories based on their magnitude of their effect on the FP counting as, consistent, likely, possible, unlikely sources of variation and topics that do not have likelihood as a source of variation.

The study proved that small number of factors had bigger effects on the reliability. According to the results, recommendations are given to improve the reliability of FP counting in organizations. Ambiguity of FP measurement conventions of the counting guideline should be resolved and counting guideline should be updated continuously based on the rapid technological improvements of the software systems. Results of this study also showed that FP variations can be eliminated by creating custom guidelines based on the standard guideline within the organizations (Kemerer & Porter, 1992).

Study of Kemerer (1993)

Based on the inter-rater reliability study of Low and Jeffery (1990), Kemerer (1993) conducted an experimental study on both inter-rater and inter-method reliability.

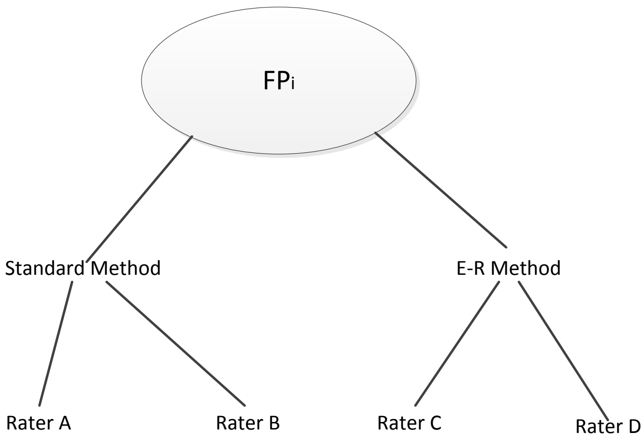


Figure 2: Overall study design (Kemerer, 1993)

For the experimental study a total of 4 raters were used, as depicted in Figure 2. While two of them were assigned to measuring the system by using standard IFPUG method, other two measured the same system by using E-R method. Inter-rater reliability of the standard method and the E-R method, and inter-method reliability of these two methods, were analyzed based on some statistical calculations. Inter-method results of these methods were similar to that was obtained across raters. After the observations were completed, it can be inferred that inter-rater and inter method reliability of FP measurement are high enough to continue further development and adoption of FP measurement.

Study of Silvia et al (2004)

In 2004, Silvia et al (2004) made an experimental study on evaluating the FSM methods OOmFP and IFPUG FPA with respect to their reproducibility and accuracy. Experiment was conducted among 22 students whose academic and measurement background levels were similar. Results of the experiment showed that OOmFP produces more consistent and reliable assessments than IFPUG FPA for OO systems.

Study of Turetken et al (2008)

In 2008, Turetken et al (2008) performed a study in which same software requirements specification document is given to three different teams which are composed of at least two expert measurers. Each team measures the software system by using different standard, IFPUG FPA, Mk II, COSMIC. However each method have different type measurement metric, result of each measurement were largely different from each other. As a result of the study, different assumptions and interpretations of measurers cause significant differences in measurement results.

Study of Ungan et al (2009)

In 2009, Ungan et al (2009) made an experimental study to identify the discrepancies in FSMs among individuals. Participants were the students of "Software Project Management" course at Middle East Technical University. Prior to the measurement they were given a six-hour COSMIC training, and to increase measurement experience level of the participants, small pilot project was given to them. Results of this study showed that variance of COSMIC measures are due to application of fundamental rules of measurement method and different interpretations and assumptions of individuals. This study showed that different knowledge and experience level of different measurers may result in the variations of the size measurement of the same software projects.

Study of Top et al (2009)

In order to identify reliability of COSMIC and observe frequently encountered COSMIC defects, O.Top (2009) conducted a case study among twelve industrial cases. Prior to measurements of industrial cases, participants performed measurements on a pilot study. Twelve industrial cases were measured by 5 different participants, and three of them took the pilot study. Results showed that measurers who performed pilot study made less measurement errors. Evaluation of the measurement results showed that knowledge and experience level of the measurers are two main factors that impact the accuracy of measurement results. Therefore COSMIC trainings should be given by emphasizing frequently encountered error types observed.

Study of Ungan et al (2010)

In 2010, Ungan et al (2010) made an experimental study as continuation of the first experiment. In this experimental study they gave suggestions about actions to be taken to improve the reliability of the COSMIC measurement results, especially about the findings that are identified in their first experimental study. According to the study, organizations would increase the reliability of COSMIC FSMs by giving detailed and advanced training programs through sample cases.

Guideline for Assuring the Accuracy of Measurements (2010)

As a reliability improvement study, COSMIC published a guideline called Guideline for Assuring the Accuracy of Measurements (COSMIC, 2011). For improving the reliability of COSMIC measurements, guideline includes actions to be taken both before and after the measurement. Actions to be taken before and after the measurement are defined as Error Prevention and Defect Detection, respectively.

For Error Prevention, prior to the measurement, pre-measurement actions to be taken are proposed in the guideline. These actions are related to Measurers, Software Artifacts and Measurement Process.

Defect Detection is composed of three activities; auditing measurements, auditing measurers and root cause analysis of defects. Additionally, for defect detection, a checklist which is composed of mostly encountered COSMIC errors is given. This checklist allows the measurers to control their measurement reports and to eliminate defects, if any exists.

However, since these most commonly made COSMIC defects are not clearly defined and error categories are not developed, all of the proposed defect detection suggestions and checklists in those researches are not adequate. Checklists contain general defects. They are not in a form that allows measurers to control and to detect defects of their measurement documents. Additionally, relationship between defects and defects' causes are not indicated.

Study of Yilmaz et al (2010)

In 2010, Yilmaz et al (2010) presented the early results of a study on the impact of the quality of software requirements specification work products on the FSMs. For the case study COSMIC FSM Method was selected. This study showed that software requirements artifacts are as much critical as measurement methods for reliability and accuracy of COSMIC FSMs.

Increasing the reliability of FSMs is related with eliminating the measurement defects from measurement reports. Checklists that were created for this purpose are not adequate to allow measurers investigating the exact point of defects. These checklists give general idea about the defects. Most commonly encountered errors given in the researches are not defined in detail. In most of the researches defect detection was performed by manually by using expert review mechanism which is time consuming. Thus, it is not preferred. In the literature there is not any research that proposes an approach to detect measurement defects automatically.

2.3. Automation Possibility for Defect Detection in COSMIC FSMs

In the literature there are not any researches conducted on detecting defects of COSMIC FSMs automatically. COSMIC FSM DDT will be developed as an extension of CUBIT. Thus, in the next section, main functionality of the CUBIT will be explained briefly.

2.3.1. CUBIT

CUBIT is a web based tool to provide easy means to measure software projects, store benchmark and measurement data and analyze them in the future. According to the SRS of CUBIT application site is presented by application server and database system. Main three screen shots are presented for a better understanding of the CUBIT's functionalities.

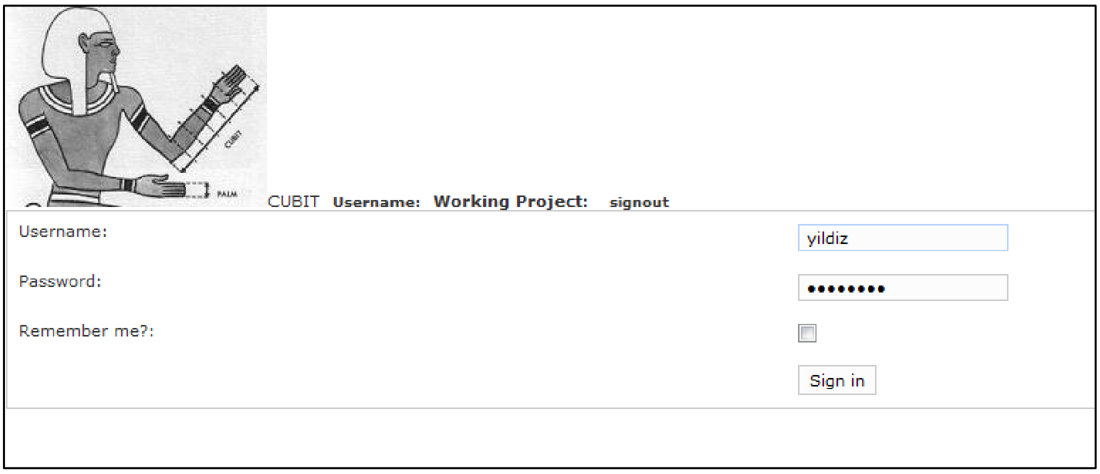


Figure 3: Login UI

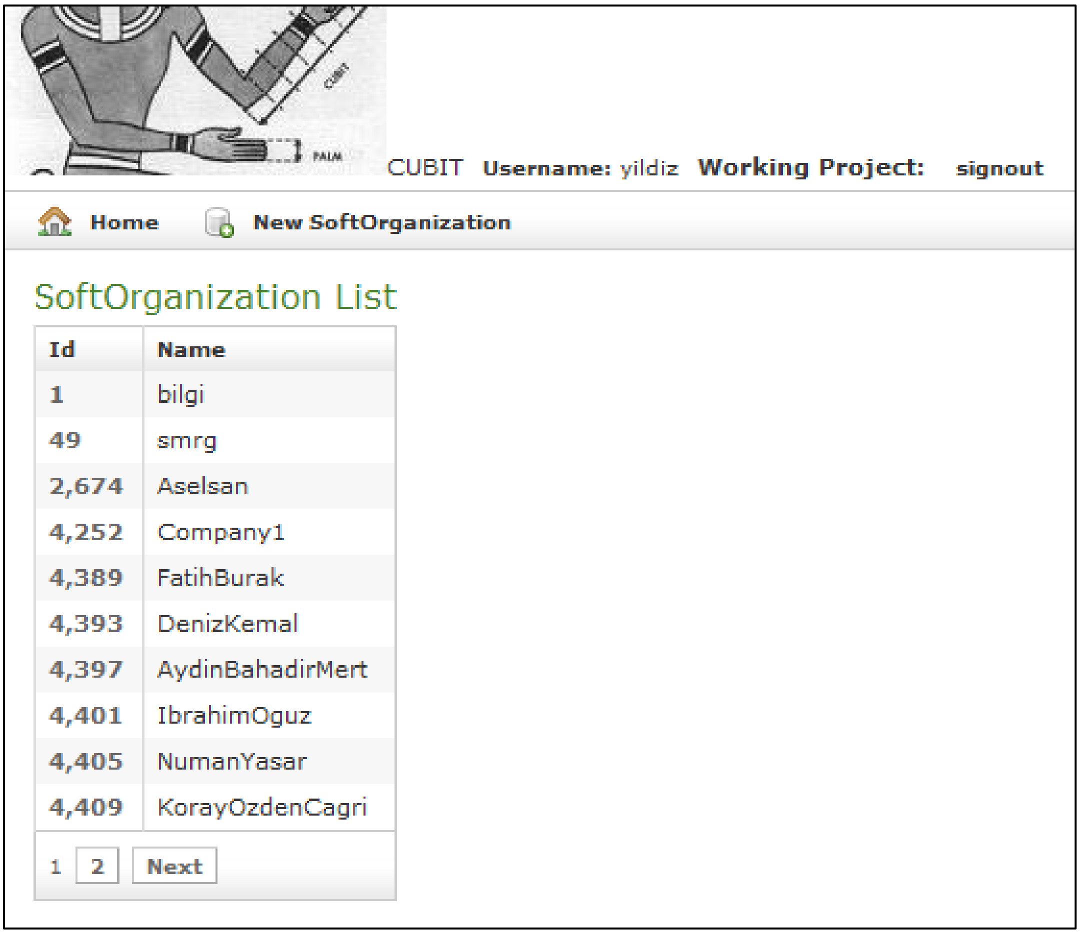
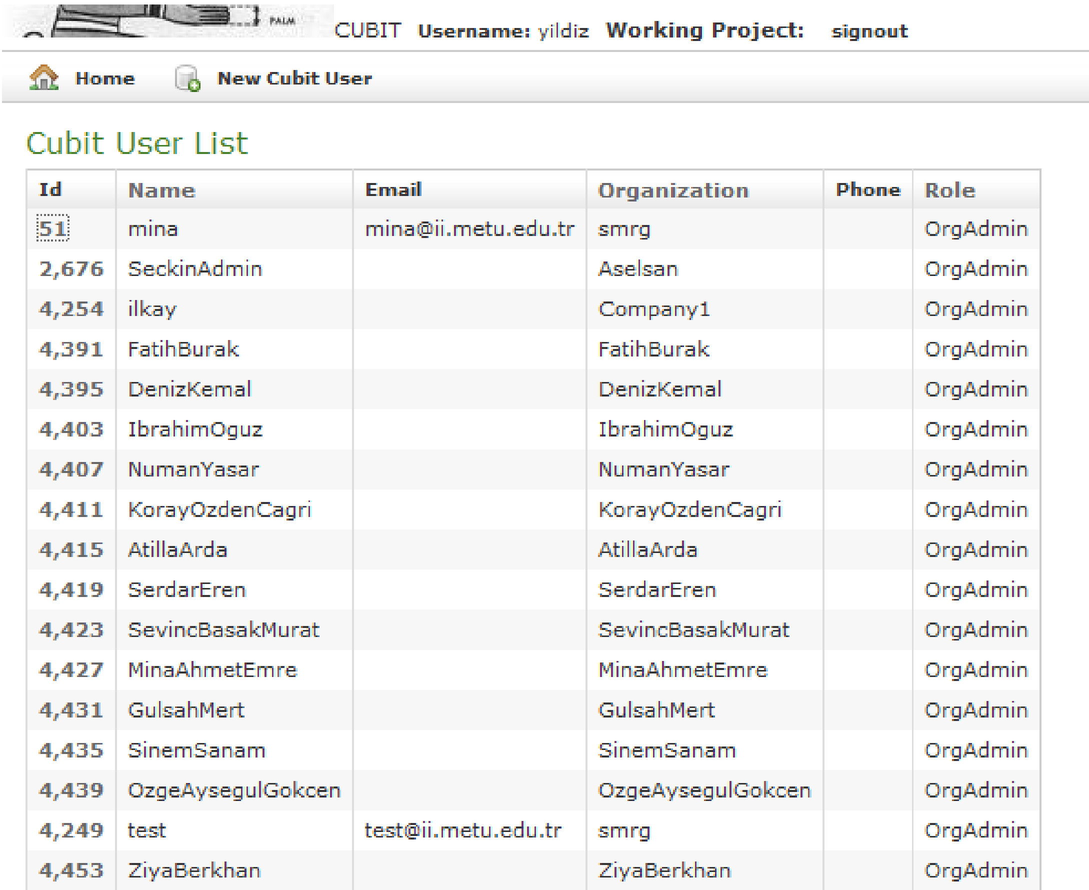


Figure 4: Manage Organization UI

In order to measure software projects, and store measurement and benchmark data by using CUBIT, first of all an organization should be created as depicted in Figure 4. For each organization only one organizational administrator, who is responsible for creating multiple users for the related organization, shall be created as shown in Figure 5.



Id	Name	Email	Organization	Phone	Role
51	mina	mina@ii.metu.edu.tr	smrg		OrgAdmin
2,676	SeckinAdmin		Aselsan		OrgAdmin
4,254	ilkay		Company1		OrgAdmin
4,391	FatihBurak		FatihBurak		OrgAdmin
4,395	DenizKemal		DenizKemal		OrgAdmin
4,403	IbrahimOguz		IbrahimOguz		OrgAdmin
4,407	NumanYasar		NumanYasar		OrgAdmin
4,411	KorayOzdenCagri		KorayOzdenCagri		OrgAdmin
4,415	AtillaArda		AtillaArda		OrgAdmin
4,419	SerdarEren		SerdarEren		OrgAdmin
4,423	SevincBasakMurat		SevincBasakMurat		OrgAdmin
4,427	MinaAhmetEmre		MinaAhmetEmre		OrgAdmin
4,431	GulsahMert		GulsahMert		OrgAdmin
4,435	SinemSanam		SinemSanam		OrgAdmin
4,439	OzgeAysegulGokcen		OzgeAysegulGokcen		OrgAdmin
4,249	test	test@ii.metu.edu.tr	smrg		OrgAdmin
4,453	ZiyaBerkhan		ZiyaBerkhan		OrgAdmin

Figure 5: Manage Users UI

Each organization may create more than one software project. Additionally, it allows users to keep information of project modules.

For measuring the software project and store its measurement data, FURs of the software project shall be created as the first step. By using different user interfaces of the CUBIT, IFPUG, COSMIC and UniFSM measurements can be facilitated, easily.

For the possibility of having different version of software projects, it allows to keep track of the sizes of different versions of the software projects. Additionally, it allows users to compare the counts to see the differences between the two versions of software projects. After the measurement is completed, it reports the measurement summary.

CUBIT allows calculating similarities between two projects as well as managing benchmark process. Each organization can create different benchmark questions based on the important attributes of their companies. Benchmark questions can be managed during the time. After definition of the benchmark questions, by answering the questions CUBIT allows organizations to store their benchmark data.

CHAPTER 3

COSMIC FSM DEFECT DETECTION APPROACH FOR MIS APPLICATIONS

As far as known from literature review, there are many reliability case studies and reliability improvement research studies conducted. In most of those studies, common problems and defects were observed manually by expert review method. Those studies showed that manual inspection is time consuming and highly depend on the quality of the requirements documentation and knowledge. Error categories are not clarified in the literature and there is not any defect detection approach that allows detecting these defects automatically. The main contribution of our study is to identify error categories and clarify their definitions and based on these error categories to detect defects of measurements automatically.

In this part of the thesis, CUBIT COSMIC FSM DDA and COSMIC FSM DDT for MIS projects and their implementations are explained. In the context of this thesis, to prevent confusion CUBIT COSMIC FSM DDT and COSMIC FSM DDA are referred as, Tool and Approach, respectively.

3.1. COSMIC FSM Defect Detection Approach

This section composed two subsections. In the first sub section identified error categories which constitute the basis of the Approach and their definitions are given, and in the second sub section COSMIC FSM DDA is explained accordingly.

3.1.1 Error Categories for MIS Projects

Accuracy of COSMIC FSMs are adversely proportional to number of defects found in the measurement reports. Decreasing the number of defects found in a COSMIC measurement directly increases the accuracy of the measurement report. Thus, in this thesis our main aim is to develop a methodology to detect the defects of COSMIC measurement reports, and by correcting these defects to increase the accuracy of COSMIC FSMs. Starting point of this thesis is to identify the error categories based on mostly encountered errors in COSMIC FSMs.

Most Most commonly made errors that are given in COSMIC AG (COSMIC, 2011) are not described in detail. It does not give the exact location of the defect in measurement report and does not allow measurers to correct the measurement defects. In this thesis, ECs of COSMIC FSMs concerning MIS applications are determined and defined in detail. Later, COSMIC FSM DDA is developed to detect defects concerning error categories with their exact location. By taking the COSMIC AG (COSMIC, 2011) and SMRG researches, (Top, Demirors, & Ozkan, 2009), (Ungan, 2010) as reference, we performed an expert review process for the 10 different measurement reports of the same software product. To identify error categories we controlled each report iteratively and clustered the defects. If a defect constitutes a pattern, we identified it as an EC. ECs are summarized in Table 2 and defined accordingly.

Table 2: Error Categories and Error Patterns

EC	DDL	Error Categories		RS	Scope
MR	SEM	EC1	FP Duplication	NI	Generic
MPR	SEM	EC2	Lack of List FP before Update FP	COSMIC AG	MIS
MPR	SEM	EC3	Lack of List FP before Delete FP	COSMIC AG	MIS
MPR	SEM	EC4	Lack of Retrieve FP before Update FP	NI	MIS
MR	SEM	EC5	Lack of DM type W in Add, Delete, Update FPs	NI	MIS
MR	SEM	EC6	Redundant DM type W in List FPs	NI	Generic
MR	SEM	EC7	Multiple occurrences of same DM within the same FP	NI	Generic
MR	SEM	EC8	Each FP should be composed of at least 2 DMs	COSMIC MM	Generic
MR	SEM	EC9	Each FP should contain at least 1 W/X DM	COSMIC MM	Generic
MR	SEM	EC10	Each FP should contain at least 1 E DM	COSMIC MM	Generic
MPR	SYT	EC11	List Fp might be included in Update/Delete FPs	NI	MIS
MR	SYT	EC12	Create/Delete/Update operations might be combined	NI	MIS
MR	SEM	EC13	DG Duplication	NI	Generic
MR	SYN	EC14	User interface components and System users are considered as DG/OOI	NI	Generic
MR	SYN	EC15	OOIs are named wrong	NI	Generic
MR	NA	EC16	Combined FP that belongs to the same FUR	NI	NA
MR	NA	EC17	Considering DMs for irrelevant OOIs	NI	NA
MR	NA	EC18	Assumed additional functionality	NI	NA
MR	NA	EC19	Dropdown list is not properly used	COSMIC AG	NA
MPR	NA	EC20	Measuring data manipulation within the system boundary	NI	NA
SAR	NA	EC21	Data that are not attributes of OOIs	COSMIC AG	NA
SAR	NA	EC22	OOI sub types are not considered	COSMIC AG	NA
SAR	NA	EC23	Errors in data analysis and OOI identification.	NI	NA

Definition of each error category (EC) is given as;

EC1 - FP Duplication: If two different FPs have exactly same (DG, DM) tuple and representing the same functionality of the system, one of these two FPs is redundant.

EC2 – Lack of List FP before Update FP: For a specific Update FP, if user forgets to measure its related List FP, measurement may have this type of defect. For instance;

List Work Order	Work Order Info	E
	Work Order Info	R
	Work Order Info	X
Update Work Order	Work Order Info	E
	Work Order Info	W
	Work Order Info	X

If "List Work Order" FP is forgotten to be measured, than there may be a defect concerning error category EC2; lack of list FP for "Update Work Order" FP.

EC3 – Lack of List FP before Delete FP: For a specific Delete FP, if user forgets to measure its related List FP, measurement may have this type of defect. For instance;

List Work Order	Work Order Info	E
	Work Order Info	R
	Work Order Info	X
Delete Work Order	Work Order Info	E
	Work Order Info	W
	Work Order Info	X

If "List Work Order" FP is forgotten to be measured, than there may be a defect concerning error category EC2; lack of list FP for "Delete Work Order" FP.

The important point for EC2 and EC3 is that for update and delete FPs that moves exactly same (DG, DM) tuple, only one list FP should be measured. Otherwise, a defect concerning EC1-Fp Duplication will occur. For the given examples "Update Work Order" and "Delete Work Order" FPs moves exactly same (DG, DM) tuple. Thus, only one list FP should be measured for both "Update Work Order" and "Delete Work Order". If measurers measure two list FPs separately, for delete and update FPs, this defect may turn into error category EC1-FP Duplication.

EC4 – Lack of Retrieve FP before Update FP: For a specific Update FP, if user forgets to measure its related Retrieve FP, measurement may have this type of defect. For instance;

Retrieve Work Order	Work Order Info	E
	Work Order Detail Info	R
	Work Order Info	X
Update Work Order	Work Order Info	E
	Work Order Info	W
	Work Order Info	X

If “Retrieve Work Order” FP is forgotten to be measured, than there may be a defect concerning error category EC2; lack of list FP for “Update Work Order” FP.

EC5 - Lack of DM type W in Add, Delete, and Update FPs: Add, Delete, and Update FPs write data on persistent storage. Thus, these FPs should consist W DM. If these FPs do not contain at least one W DM, defects occur concerning this error category.

EC6 - Redundant DM type W in List/Retrieve FPs: List, Retrieve FPs reads data from persistent storage with R DM and thus, usually do not contain W DM within the FP. If these FPs contain at least one W DM, defects occur concerning this error category.

EC7 - Multiple occurrences of same DM within the same FP: If a FP contains more than two DMs that move the same DG with the same DM type, then defects occur concerning this error category. For instance; a FP that has two DMs which move DG “Person Info” with same DM type “R” is given as follows;

Update Personnel	Person Info	E
	Person Info	R
	Person Info	R
	Person Info	X
	Error Message	X

One of these DMs is redundant within the FP.

EC8 - Each FP should be composed of at least 2 DMs: This error category is a measurement rule of COSMIC FSM Method and thus, if any defect is detected concerning this error category, it is counted as an error. Any FP should be composed of at least two DMs.

EC9 - Each FP should contain at least 1 W/X DM: This error category is a measurement rule of COSMIC FSM Method and thus, if any defect is detected concerning this error category, it is counted as an error. Any FP should contain at least one W or X DM.

EC10 - Each FP should contain at least 1 E DM: This error category is a measurement rule of COSMIC FSM Method and thus, if ant defect is detected concerning this error category, it is counted as an error. Any FP should be composed of at least one E DM.

EC11 - List FP might be included in Update/Delete FPs: As described in the error category EC4, in any FP for update and delete FPs related list FP should be measured. If list FP is measured within any update or delete FP not as a separate FP, a defect occurs concerning this category.

EC12 - Create/Delete/Update operations might be combined: Create, Delete, Update represents different functionalities of systems. They should be measured separately. If measurers combine these functionalities and measure them as a whole, defects concerning this error category exist.

EC13 - DG Duplication: DG composed of combinations of DAs concerning OOIs, if two different DG contains exactly the same combinations of DAs; defects concerning this error category exist.

EC14 - User interface components and System users are considered as DG/OOI: Since OOI and DG concepts are abstract; it is difficult to identify them for an inexperienced measurer. Measurers usually use user interface components for determining the names of DG and OOI. If measurers use user interface components such as command, menu, button, and screen for naming OOIs and DGs, defect concerning error category EC14 occurs.

EC15 - OOIs are named wrong: OOI usually represents tables in the database, and usually composed of phrases. If measurers use verbs such as select, search, save, find and query, for naming OOIs, defects concerning this error category exist. These verbs were identified in case study 1.

EC16 - Combined FP that belongs to the same FUR: FPs are identified based on FURs. If all of the functionality which is explained in one FUR is measured within less than the required amount of FP, defects concerning this error category occur.

EC17- Considering DMs for irrelevant OOIs: If a DM processes an action for an OOI that is unrelated with the system, defects of this category occur.

EC18 - Assumed additional functionality: If measurer measure a functionality that do not exist in SRS, defect concerning this error category exist.

EC19 - Dropdown list is not properly used: Dropdown usage composed of two DMs E and R, if measurer use dropdown in unsuitable place and with wrong template, defect concerning this error category exist.

EC20 - Measuring data manipulation within the system boundary: Data manipulations of complex algorithms such as back ground process a query are not measured, with respect to COSMIC FSM method. If measurers measure processes that are out of measurement system boundary, defects concerning this error category exist.

EC21 - Data that are not attributes of OOIs: If measurers add DAs into an OOI which is unrelated with the added DAs, defects concerning this category exist.

EC22 - OOI sub types are not considered: In some measurements, OOI and its related subtypes should be used. If measurers miss to identify subtype of an OOI and instead of subtype use OOI, defects related to this category occur.

EC23 - Errors in data analysis and OOI identification: As OOIs are generally identified based on E-R diagrams of systems. If E-R diagram do not represent the related system, defects may occur during OOI identification.

In the context of error categories determination, commonly-made COSMIC errors stated in COSMIC Guideline for Assuring the Accuracy of Measurements (COSMIC, 2011) and stated in previous researches (Turetken, Ozcan Top, Ozkan, & Demirors, 2008), (Top, Demirors, & Ozkan, 2009), (Ungan, Demirörs, Top, & Özkan, 2009), (Ungan, 2010) in SMRG group will be extended and some fundamental COSMIC rules of COSMIC Measurement Method (ISO/IEC, 2003b) is taken as reference.

After the identification of error categories, they are classified according to their existence as; Newly Identified (NI) error categories, the ones exists in COSMIC Guideline for Assuring the Accuracy of Measurements (COSMIC, 2011) (COSMIC AG), and error categories created based on some fundamental COSMIC rules of COSMIC Measurement Manual(ISO/IEC, 2003b) (COSMIC MM). 5 of the presented 23 error categories are under the category COSMIC AG, 15 of them are under the category NI and lastly 3 of them are under the category (COSMIC MM).

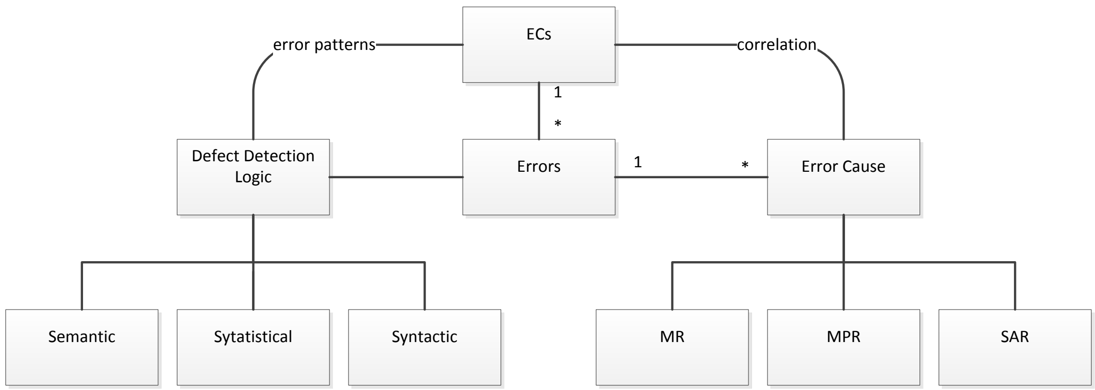


Figure 6: Error Categories & Error Causes &Defect Detection Logics

Additionally, causes of the ECs are divided into three as Measurer Related (MR), Software Artifact Related (SAR) and Measurement Process Related (MPR). MR error categories occur if either a measurer does not know how to measure a piece of a software or he/she makes a mistake when he/she knows to measure a piece of a software. SAR error categories are errors occur when a software artifact does not represent the system clearly. MPR error categories are errors based on the usage of measurement processes categories.

To detect defects of each category we developed algorithms which constituted the Tool. Each algorithm has a logic that maintains detecting defects of the related error category. These Defect Detection Logics (DDL) are divided into three categories as Semantic, Statistical and Syntactic as presented in Figure 6.

Semantic (SEM) logic checks errors semantically.

Statistical (SYT) logic checks errors based on statistical calculations.

Syntactic (SYN) logic checks the existence of any defects related with the syntactical errors.

3.1.2 COSMIC FSM Defect Detection Approach for MIS Applications

COSMIC FSM DDA is developed based on the ECs, which are defined in Section 3.1.1. Since, these error categories were identified by using COSMIC measurements of MIS applications, COSMIC FSM DDA can only be valid for MIS applications.

COSMIC FSM DDA constitutes a basis for the DDT. To detect defects concerning all of the error categories which are given in Table 1, some rules are created based on the COSMIC FSM Method. Figure 7 visualizes the relationship between the Approach and the COSMIC FSM Method (ISO/IEC, 2003b). These rules are related with the functional size components, which are defined in literature review in Section 2.2. COSMIC FSM DDA makes modifications on some of these functional size components.

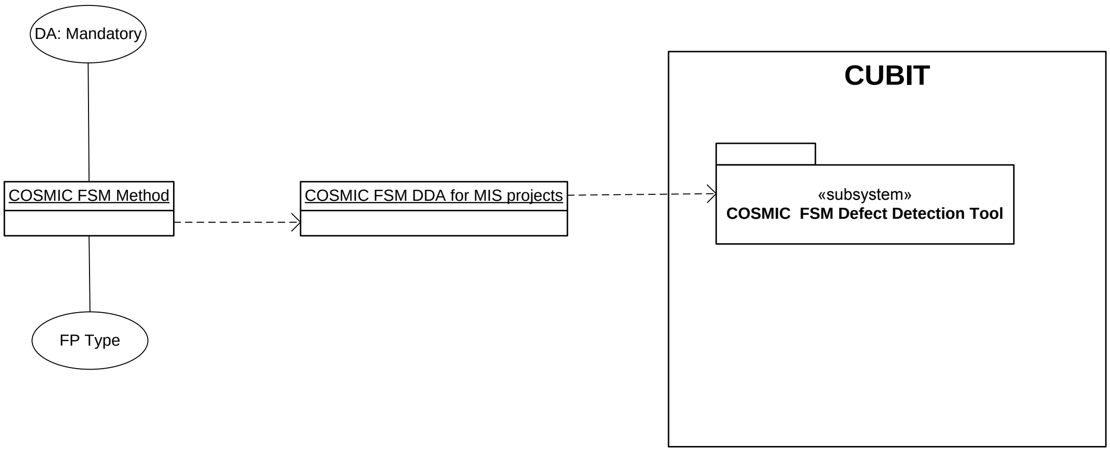


Figure 7: Relational representation of COSMIC FSM Method, the Approach and the Tool

First rule of the DDA is that according to measurement process of COSMIC Measurement method, identifying DAs are optional. Thus, measurers usually skip determining DAs in their size measurements. As depicted in Figure 7, the approach considers DA as one of the mandatory functional size components of COSMIC measurement process such as FURS, FPs, DMs, OOI and DGs.

Secondly, the DDA adds a newly identified functional size component called FP Type. There are 4 different subtypes of FP Type, “Create”, “Delete”, and “Update”, List” and “Retrieve”. In case measurements may include FPs that does not fit any FP type, we created one more FP Type as “Other”. Rules of the Approach and its benefits are explained in detail in the following paragraphs.

CUBIT COSMIC FSM DDA for MIS applications is composed of mandatory components FP, OOI, DM, DG, DA, and FP Type. During a measurement process, if “DA” and “FP Type” are missed to be identified, defects concerning some of the error categories, which is given Table 1, cannot be detected. We observed the impact of existence of DA and FP Type knowledge on the defect detection process, respectively. According to the findings, if DAs are not identified for a COSMIC FSM, then defects concerning “EC13 - DG Duplication” cannot be detected. FP Type knowledge affects the detect detection of error categories, “EC1 - FP Duplication”, “EC2 – Lack of List FP before Update FP”, “EC3 – Lack of List FP before Delete FP”, “EC4 – Lack of Retrieve FP before Update FP”, “EC5 - Lack of DM type W in Add, Delete, and Update FPs”, “EC6 - Redundant DM type W in List FPs”, “EC11 - List FP might be included in Update/Delete FPs”, and “EC12 - Create/Delete/Update operations might be combined”.

After introducing the DDA, CUBIT COSMIC FSM DDT for MIS projects is developed. Relationship between the DDT and the DDA is visualized in Figure 4. Detail information about the DDT is explained in section 3.2.

3.2. CUBIT COSMIC FSM Defect Detection Tool for MIS Applications

CUBIT COSMIC FSM DDT is developed based on the Approach, and thus contains rules of the Approach. In this section CUBIT COSMIC FSM DDT for MIs applications is explained in detail.

CUBIT COSMIC FSM DDT for MIS applications detects defects of COSMIC measurements with respect to error categories automatically. It is developed as a module of CUBIT which is explained in Chapter 2.

During the development of the tool based on the approach, some important points were controlled. Since according to the rules of the Approach, existing architecture of the CUBIT requires changes, structure of CUBIT E-R diagram and database, were controlled considering functional size components, DA and FP type

Since DA, exists in CUBIT database by default, we did not need to modify E-R diagram and database of the CUBIT. However, since component FP type is a newly created, CUBIT E-R diagram showed that and CUBIT database does not contain such a field, FP type. Therefore, CUBIT E-R diagram and database is modified by adding FP type field to corresponding tables.

The Tool is the collection of defect detection algorithms. There is a one to one relationship between algorithms and error categories. For each error category a specific algorithm is tried to be developed. In the context of the Defect-Detection Algorithms development, error patterns of which determinations are explained in detail in chapter 4, under case stuyd1, for each error category are taken as reference. As can be seen from Table 1, although 23 error categories were identified, for only 15 of them defect detection algorithms are created.

Algorithms can be classified under 3 defect detection approaches as, semantic, statistical and syntactical as depicted in Figure 6. DDL column of Table 2 shows the defect detection logics with respect to each error category

Error categories that can be detected by algorithms that use SEM logic are EC1, EC2, EC3, EC4, EC5, EC6, EC7, EC8, EC9, EC10, EC13, error categories which can be detected by algorithms that use statistical SYT logic are, EC11 and EC12, error categories which can be detected by algorithms that use syntactical SYN logic are EC14 and EC15. N/A indicates that related error categories cannot be detected automatically. SRS knowledge and manual inspection is required to detect defects in these error categories.

Tool gives two different types of results as Warning and Error. Errors are given for only the error categories EC8, EC9 and EC10 which were created based on COSMIC MM (ISO/IEC, 2003b). For the rest of the error categories, warnings are given to users as a summary. Users shall correct their measurements according to these results.

To detect defects related to functional size components FP, DM, OOI and DG, the Tool uses FP type and DA components as inputs. It summarizes the results related to FP & DM and results related to OOI & DG, in two different screens which are presented in Figure 10 and Figure 11, respectively.

As depicted in Figure 8, new tab “Show FP verification” and “Show OOI verification” are added to CUBIT Tool. “Show FP verification” and “Show OOI verification” tabs triggers the Tool for defect detection with respect to FP and/or DM and OOI and/or DG, respectively.

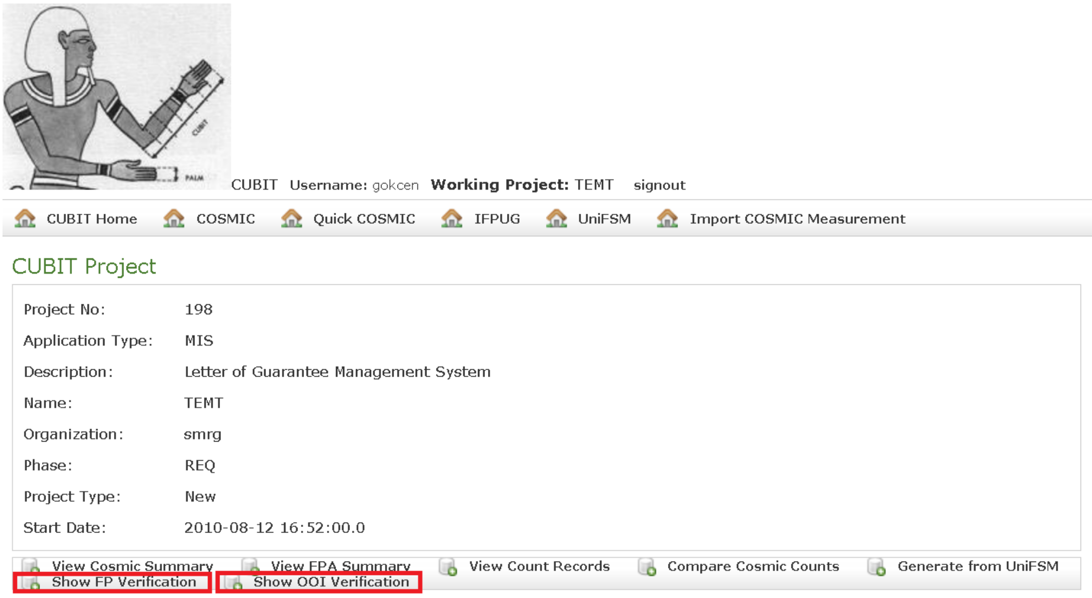


Figure 8: GUI 1-CUBIT COSMIC FSM DDT for MIS projects

Figure 9 presents the Use case model of the Tool, and use case definition of defect detection process is explained in detail accordingly in Table 3.

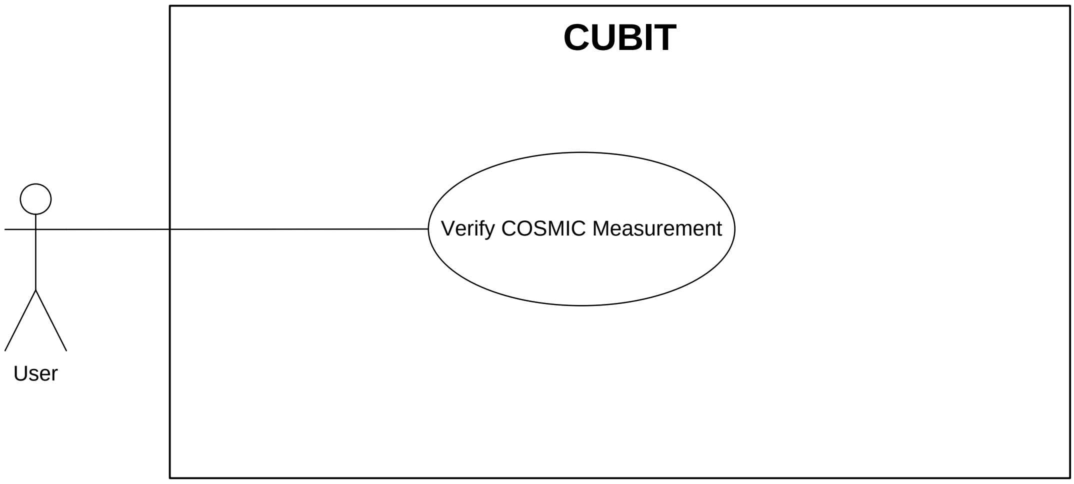
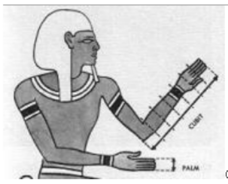


Figure 9: Use Case Model of COSMIC FSM Defect Detection Process

Table 3: Use case of defect detection of COSMIC FSMs

Use Case Section	Comment
Use Case Name	UC1: Detect Defects of COSMIC FSM
Scope	Requirement Management Tool
Level	User Goal
Primary Actor	User
Stakeholders & Interests	User
Priority	1
Preconditions	Measurement of related project should be selected
Success Guarantees	User successfully lists detected defects for COSMIC measurement of selected project.
Main Success Scenario	1- User wants to verify COSMIC measurement of selected project. 2- System displays the summary information of selected project (Figure1) 3- User selects FP verification. System displays FP verification results. (Figure 2)
Extensions	3. a. User selects OOI verification. 4- System displays OOI verification results(Figure 3)
Special Requirements	1- None
Technology and Data Var. List	None
Frequency of occurrence	Whenever a COSMIC FSM is facilitated by CUBIT.
Miscellaneous	None

Figure 10 present the summary report GUI of detected defects related to FPs and DMs. Figure 11 present the summary report GUI of detected defects related to OOIs and DGs.



CUBIT **Username:** gokcen **Working Project:** Budget-BM **signout**

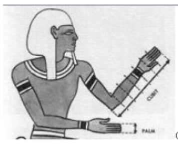
COSMIC List

1- Existing number of R DM is 50, List FP is 3
- Expected number of R DM is 44, List FP is 5
- While Existing R is equal to Expected R, existing List is less than expected List, List FP may be defined as part of FP Update/Delete.

2- Existing number of W DM is 20, Create FP is 7, Update FP is 5, Delete FP is 0
- Expected number of W DM is 24, Create FP is 6, Update FP is 4 Delete FP is 4
- Existing values are less than the expected ones,Create/Update/Delete operations might be combined.They should be measured as seperately.

Name	Description	Data Movements		Count	Verification Results
FP1	List Budget Category	BudgetCategoryInfo	E	3	
		BudgetCategoryInfo	R		
		BudgetCategoryInfo	X		
FP3	Retrieve Budget Category	BudgetCategoryInfo	E	5	Warning, there are multiple occurences of same data dovements BudgetCategoryInfo within the same FP.
		UpperBudgetCategoryInfo	R		
		BudgetCategoryInfo	X		
		BudgetCategoryInfo	R		
		BudgetCategoryInfo	X		
FP2	Add Budget Category	BudgetCategoryInfo	E	5	
		UpperBudgetCategoryInfo	R		
		UpperBudgetCategoryInfo	X		
		BudgetCategoryInfo	W		
		Error/Confirmation	X		

Figure 10: GUI 2-FP Verification Report



CUBIT **Username:** gokcen **Working Project:** Budget-BM **signout**

Object of Interest Definition

Name	Attributes	Data Groups	Verification Result
BudgetCategory		BudgetCategoryInfo UpperBudgetCategoryInfo	
BudgetItem		BudgetItemInfo UpperBudgetItemInfo	
BudgetItemAccount		BudgetItemAccountInfo shouldBeDeletedSpentMoneyInfo	
Unit		UnitInfo	
BudgetRequestForm		BudgetRequestFormExcelFileInfo BudgetRequestFormStatusInfo PreviousBudgetRequestFormInfo BudgetRequestFormInfo	
FundRequestForm		FundRequestFormInfo FundRequestFormExcelFileInfo FundRequestFormStatusInfo	
Period		ActivePeriodInfo PeriodInfo	
UnitBudgetItem		UnitBudgetItemInfo	
UnitFundItem		UnitFundItemInfo	
BudgetMovementDetail		BudgetMoveInfo	
QueryBudgetRequestForm		QueryBudgetRequestFormParameters	Warning, OOI names should composed of nouns.
QueryFundRequestForm		QueryFundRequestFormParameters	Warning, OOI names should composed of nouns.

Figure 11: GUI 3-OOI Verification Report

3.2.1. Defect Detection Algorithms

Figure 12 presents the flowchart of COSMIC FSM Verification. For the verification process, first of all COSMIC FSM of related project should be selected. For the selected project user starts the measurement verification process and then verification report are displayed.

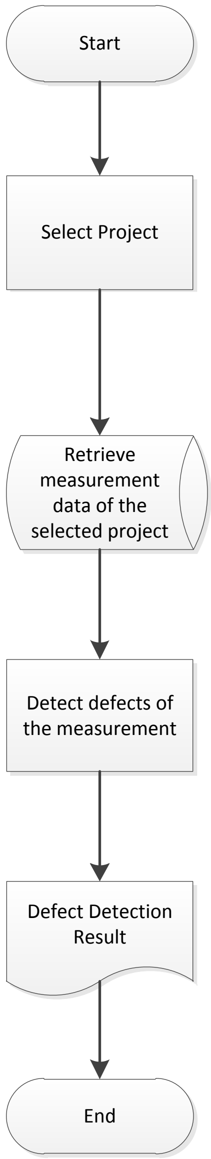


Figure 12: COSMIC FSM Verification Overview

Figure 13 explains the verification overview in detail. Process starts with EC1 and continues until the last error category EC15. For the related EC there are two options;

- If a defect is detected for related EC, system gives a Warning/Error message on the verification report and continues with the next EC.
- If no defect is detected, system directly goes to next EC.

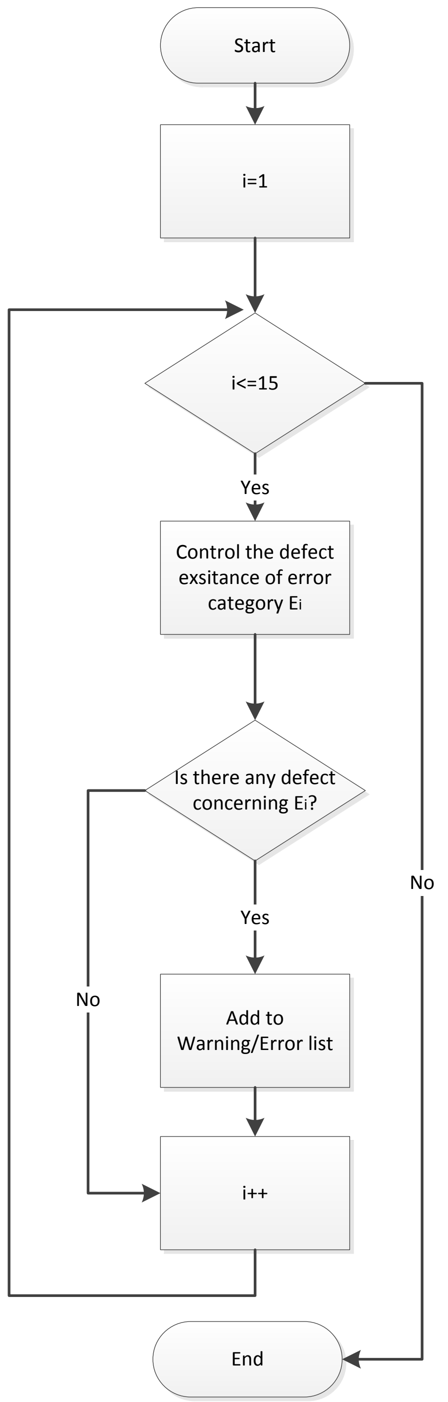


Figure 13: Measurement Verification Process

3.2.1.1. Algorithm for EC1

Figure 14 gives the detailed information about defect detection algorithm of EC1. Algorithm of EC1, is seeking for whether there is any FP duplication in the selected project or not. If it detects same (DG, DM) tuples for different FPs, it creates a warning message.

Logic of the algorithm contains two loops. Both outer and inner loops run on the same FP array. However, inner loop starts from the second element of the same FP array. First element of the outer array equals to 1^{st} FP of the FP array, first element of the inner array equals to 2^{nd} element of the FP array.

During one control system compares two FPs. For each outerFP, system checks each element of the inner array. The Tool continues its process until the control is completed for the last element of the outerFP array.

Details of the algorithm of EC1 are visualized in Figure 14. Each DM of the outerFP is compared to each DM of the innerFP. If two DMs have exactly same (DG, DM) tuple, Counter is increased. The Tool gives a warning message if the following conditions are provided;

```
Start
    If(outerDM=innerDM & outerDG =innerDG)
        Counter++
        If(Counter=size of outerFP & Counter=size of innerFP &
        outerFPtype=innerFPtype)
            print warning
        End if
    End if
End
```

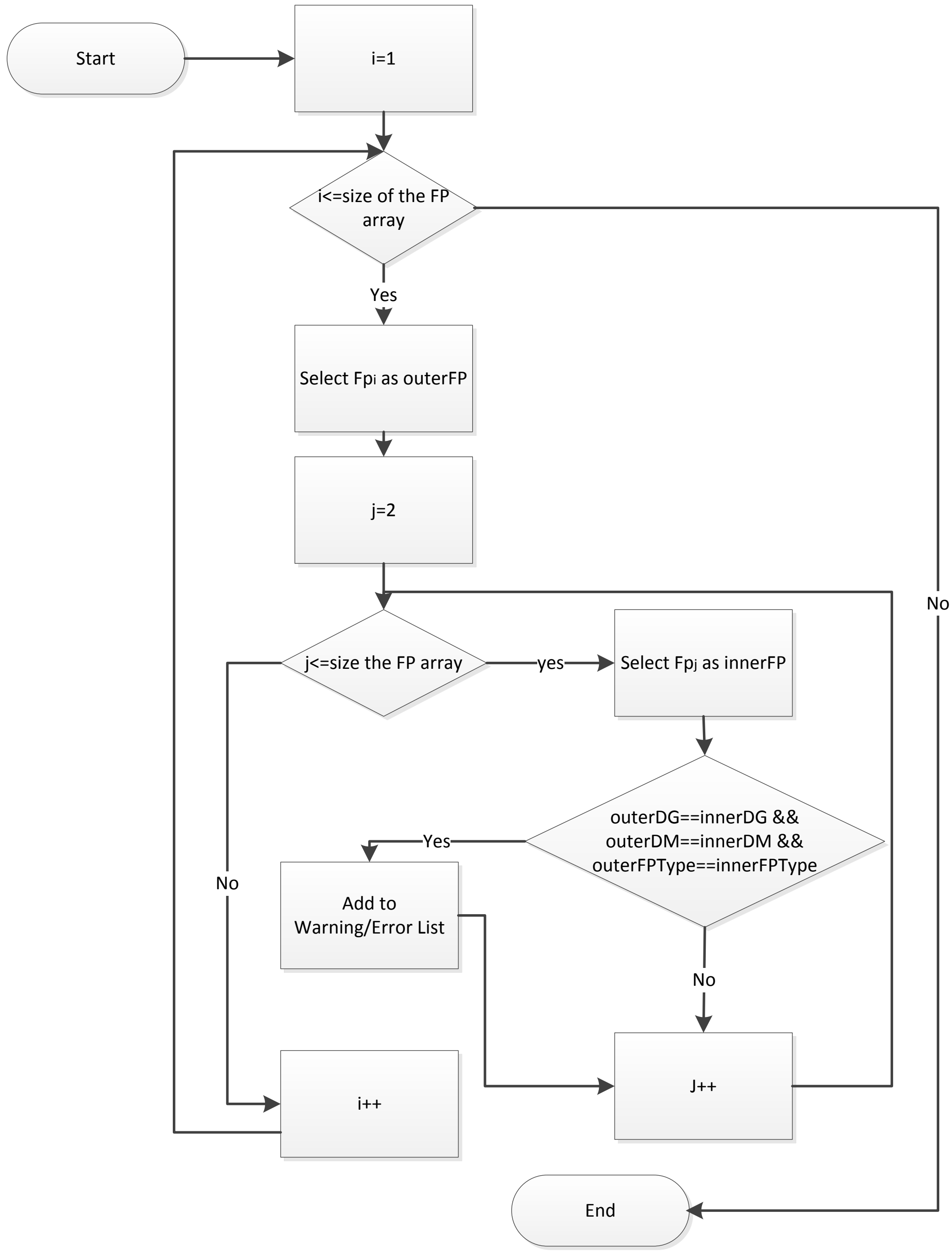


Figure 14: EC1 - Duplicate FP

3.2.1.2. Algorithm for EC2

For detecting EC2, type of FPs should be entered. With lack of FP knowledge, system cannot detect the existence of the defect concerning EC2.

Logic of the algorithm contains two loops. Both outer and inner loops run on the same FP array. However, inner loop starts from the second element of the same FP array. First element of the outer array equals to 1st FP of the FP array, first element of the inner array equals to 2nd element of the FP array. For each FP with type "update" in the FP array, the Tool controls whether its "list" FP exists or not. In order to do that, for instance;

Figure 15 visualizes the algorithm of EC2. The tool keeps "update" FPs in the outerFP and "list" FPs in the innerFP. For instance;

outerFP = Update Role Info

innerFP =List Role Info

It controls whether the innerFP, which is List Role Info, is the list FP of the outerFP, which is Update Role Info, by controlling the OOIs of the outerFP and innerFP. The Tool gives a warning message, if the following conditions for the related FPs are not provided;

```
Start
    If(outerFPtype=update & innerFPtype =list)
        If(OOI for outer and innerFp are the same)
            If(outerFp  contains  DM "W" & innerFp  contains DM "R")
                listCounterUpdate++
            End if
        End if
    End if
End
```

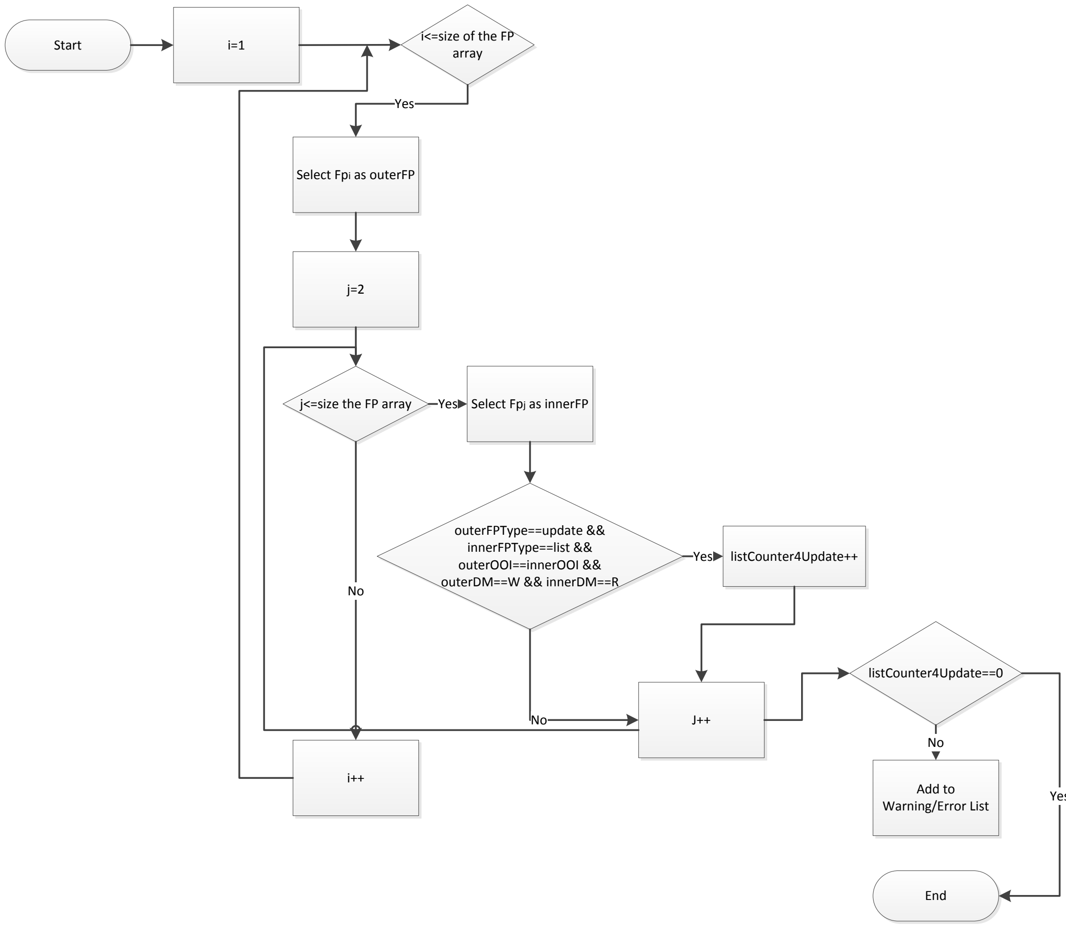


Figure 15: EC2 - Lack of List FP before Update FP

3.2.1.3. Algorithm for EC3

For detecting EC3, type of FPs should be entered. With lack of FP knowledge, system cannot detect the existence of the defect concerning EC3.

Logic of the algorithm contains two loops. Both outer and inner loops run on the same FP array. However, inner loop starts from the second element of the same FP array. First element of the outer array equals to 1st FP of the FP array, first element of the inner array equals to 2nd element of the FP array. For each FP with type "delete" in the FP array, the Tool controls whether its "list" FP exists or not. In order to do that, for instance;

Figure 16 visualizes the algorithm of EC3. The tool keeps "delete" FPs in the outerFP and "list" FPs in the innerFP. For instance;

outerFP = Delete Role Info

innerFP =List Role Info

It controls whether the innerFP, which is List Role Info, is the list FP of the outerFP ,which is Delete Role Info, by controlling the OOIs of the outerFP and innerFP. The Tool gives a warning message, if the following conditions for the related FPs are not provided;

Start

If(outerFPtype=delete & innerFPtype =list)

If(OOI for outer and innerFp are the same)

If(outerFp contains DM "W" & innerFp contains DM "R")

listCounterDelete++

End if

End if

End if

End

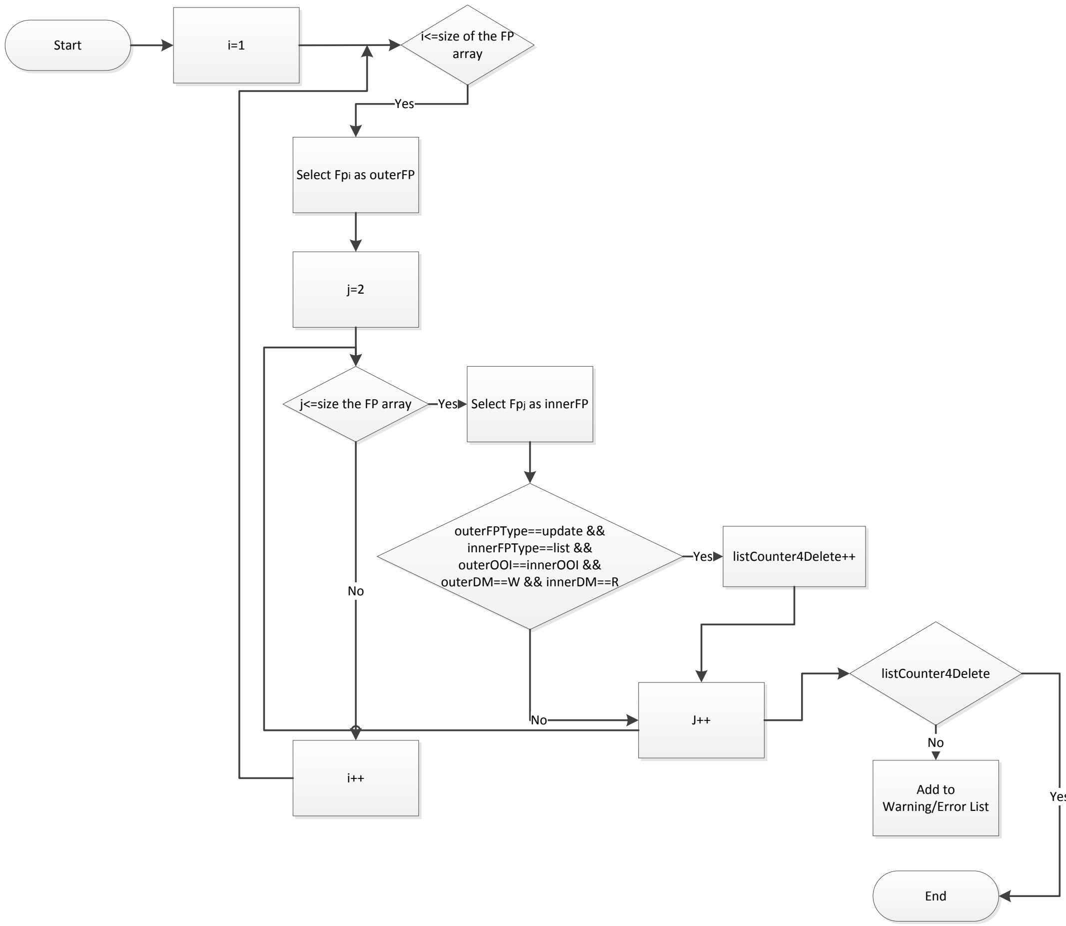


Figure 16: EC3 - Lack of List FP before Delete FP

3.2.1.4. Algorithm for EC4

For detecting EC4, type of FPs should be entered. With lack of FP knowledge, system cannot detect the existence of the defect concerning EC4.

Logic of the algorithm contains two loops. Both outer and inner loops run on the same FP array. However, inner loop starts from the second element of the same FP array. First element of the outer array equals to 1^{st} FP of the FP array, first element of the inner array equals to 2^{nd} element of the FP array. For each FP with type "update" in the FP array, the Tool controls whether its "retrieve" FP exists or not. In order to do that, for instance;

Figure 17 visualizes the algorithm of EC4. The tool keeps "update" FPs in the outerFP and "retrieve" FPs in the innerFP. For instance;

outerFP = Update Role Info

innerFP =Retrieve Role Info

It controls whether the innerFP, which is List Role Info, is the list FP of the outerFP, which is Delete Role Info, by controlling the OOIs of the outerFP and innerFP. The Tool gives a warning message, if the following conditions for the related FPs are not provided;

```
Start
        If(outerFPtype=update & innerFPtype =retrieve)
                If(OOI for outer and innerFp are the same)
                        If(outerFp  contains  DM "W" & innerFp  contains DM "R")
                                retrieveCounterUpdate++
                        End if
                End if
        End if
End
```

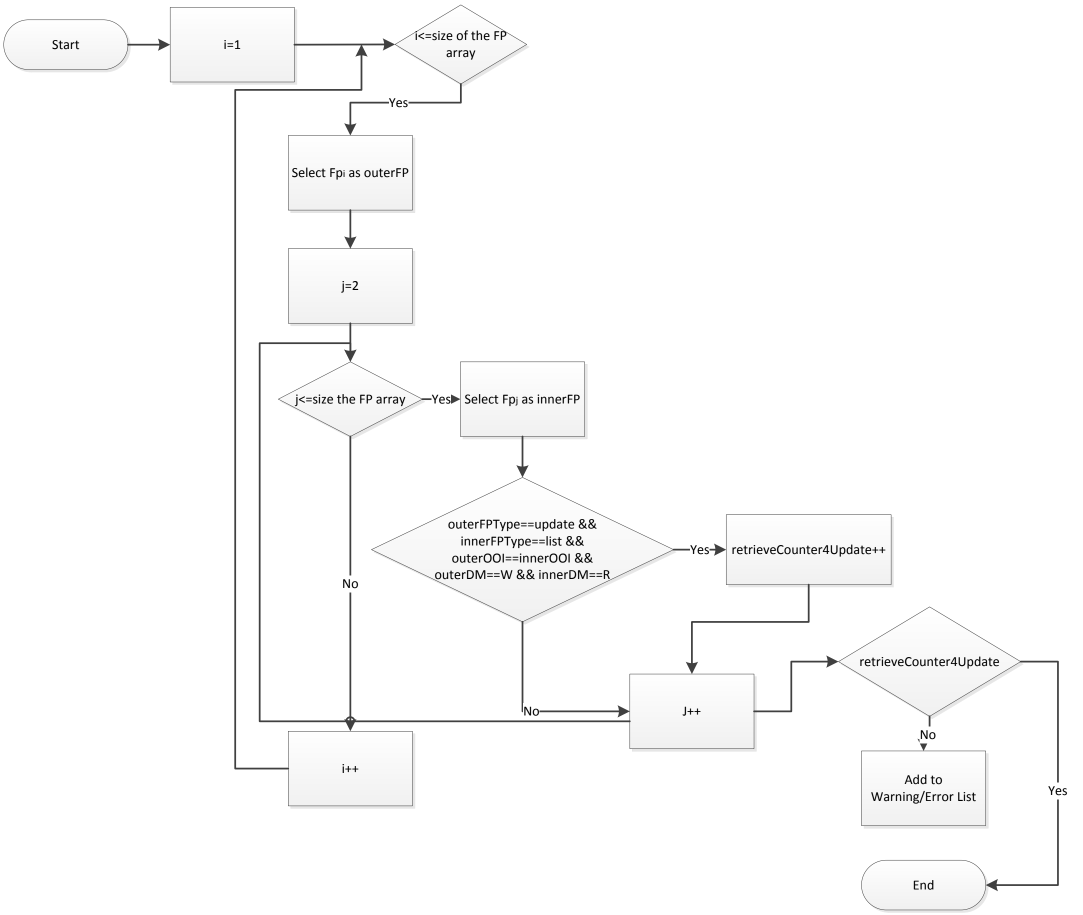


Figure 17: EC4 - Lack of Retrieve FP before Update FP

3.2.1.5. Algorithm for EC5

For detecting EC5, type of FPs should be entered. With lack of FP knowledge, system cannot detect the defect existence of EC5.

Figure 18 explains the algorithm of EC5 in detail. For each FP in the FP array list, system controls the FP type and gives a warning message if the following condition is not maintained;

Start

    If(outerFPtype=create | outerFPtype=delete | outerFPtype=update)
        If(FP contains DM W)
            DMWCount++
        End if
    End if

End

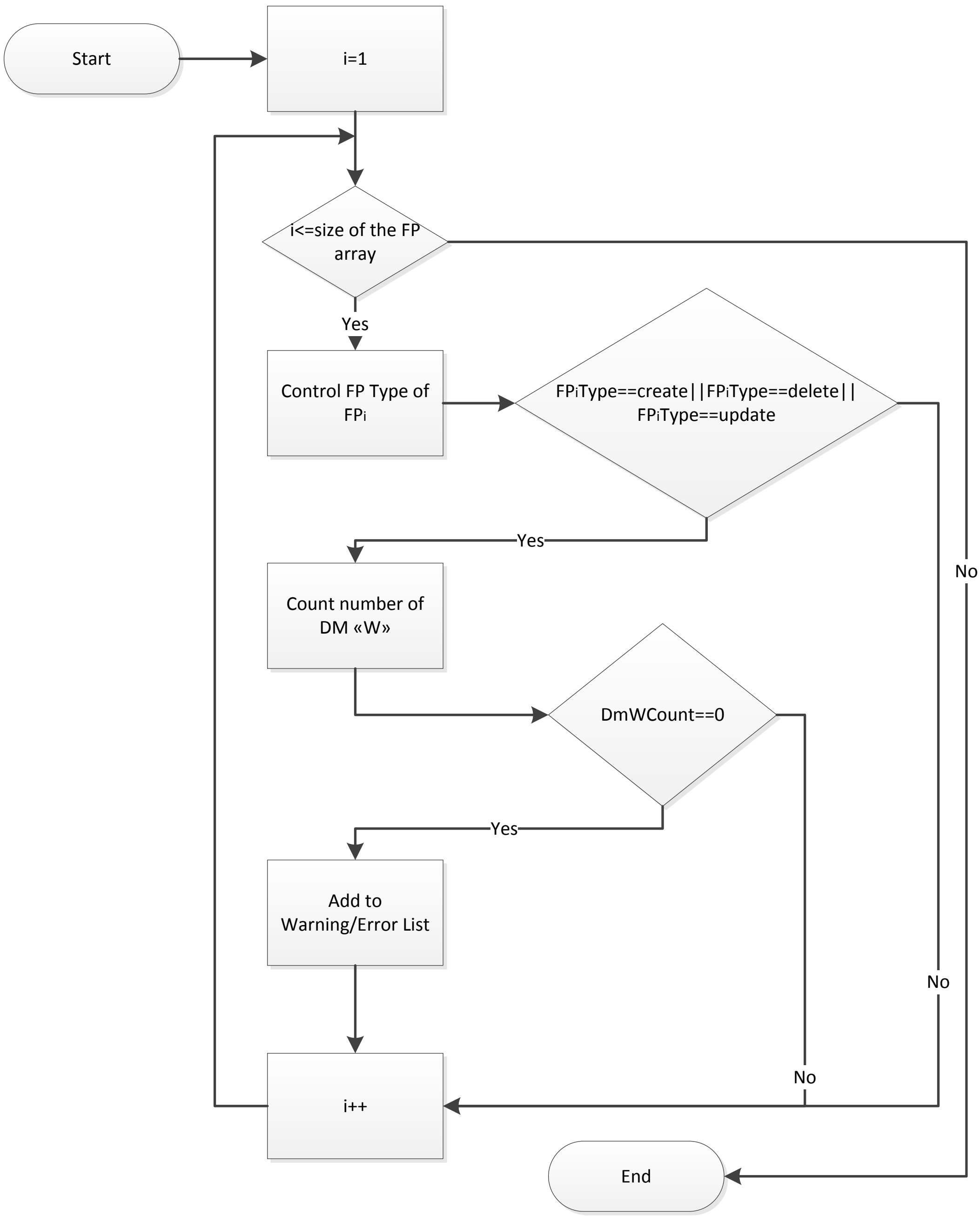


Figure 18: EC5 - Lack of DM type W in Add, Delete, and Update FPs

3.2.1.6. Algorithm for EC6

For detecting EC6, type of FPs should be entered. With lack of FP knowledge, system cannot detect the existence of error with error category EC5.

Figure 19 explains the algorithm of EC6 in detail. For each FP in the FP array list, system controls the FP type and gives a warning message if the following condition exists;

Start

    If(outerFPtype=list | outerFPtype=retrieve)

        If(FP contains DM W)

            DMWCount++

        End if

    End if

End

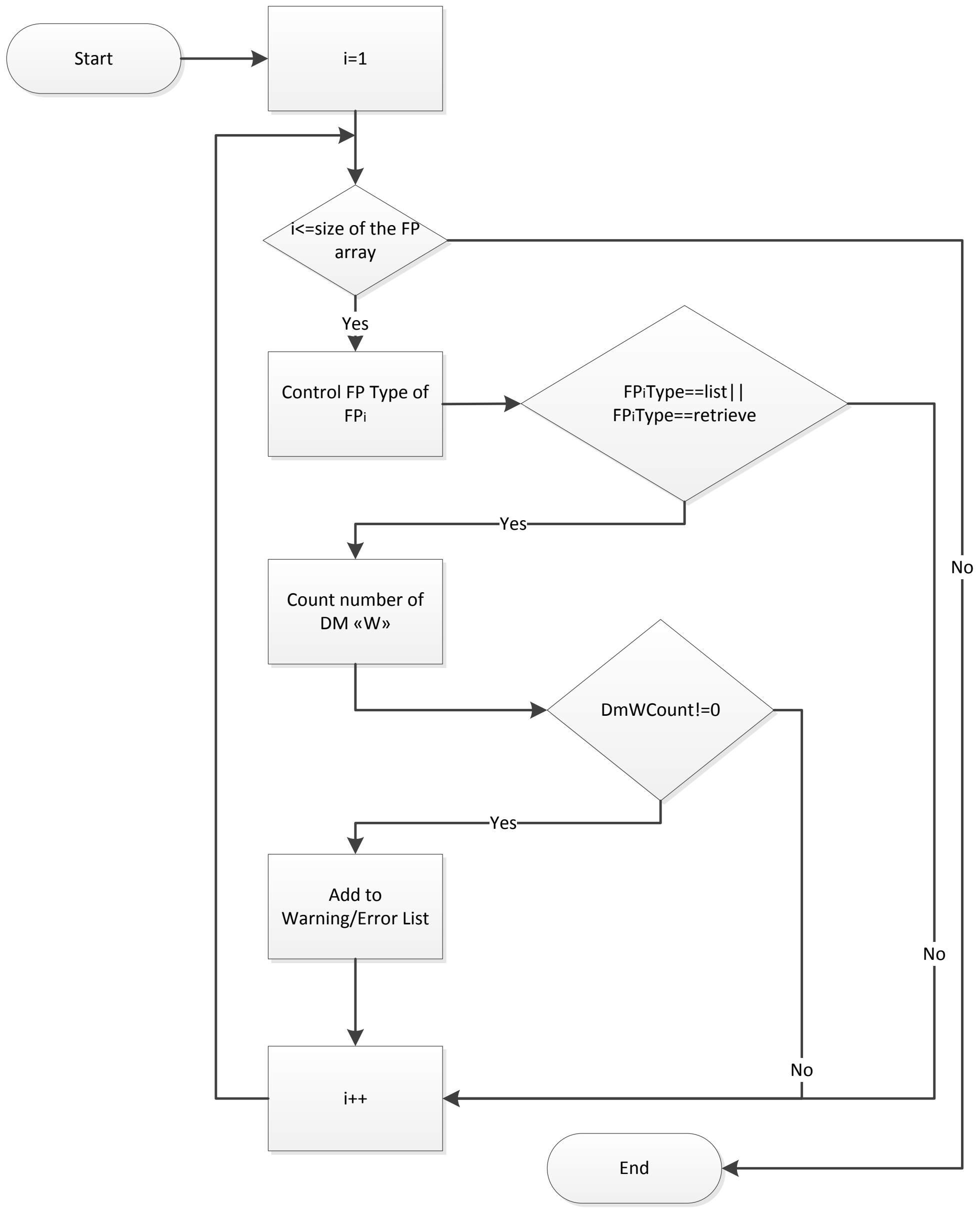


Figure 19: EC6 - Redundant DM type W in List FPs

3.2.1.7. Algorithm for EC7

Figure 20 explains the algorithm of EC7 in detail. For each FP in the FP array list, system controls each DM with the other DMs that belong to the same FP. System gives a warning message if simcounter is bigger than or equal to two.

```
Start
    If(DG4outerDM==DG4innerDM && Label4outerDM==Label4innerDM)
        simcounter++
    End if
End
```

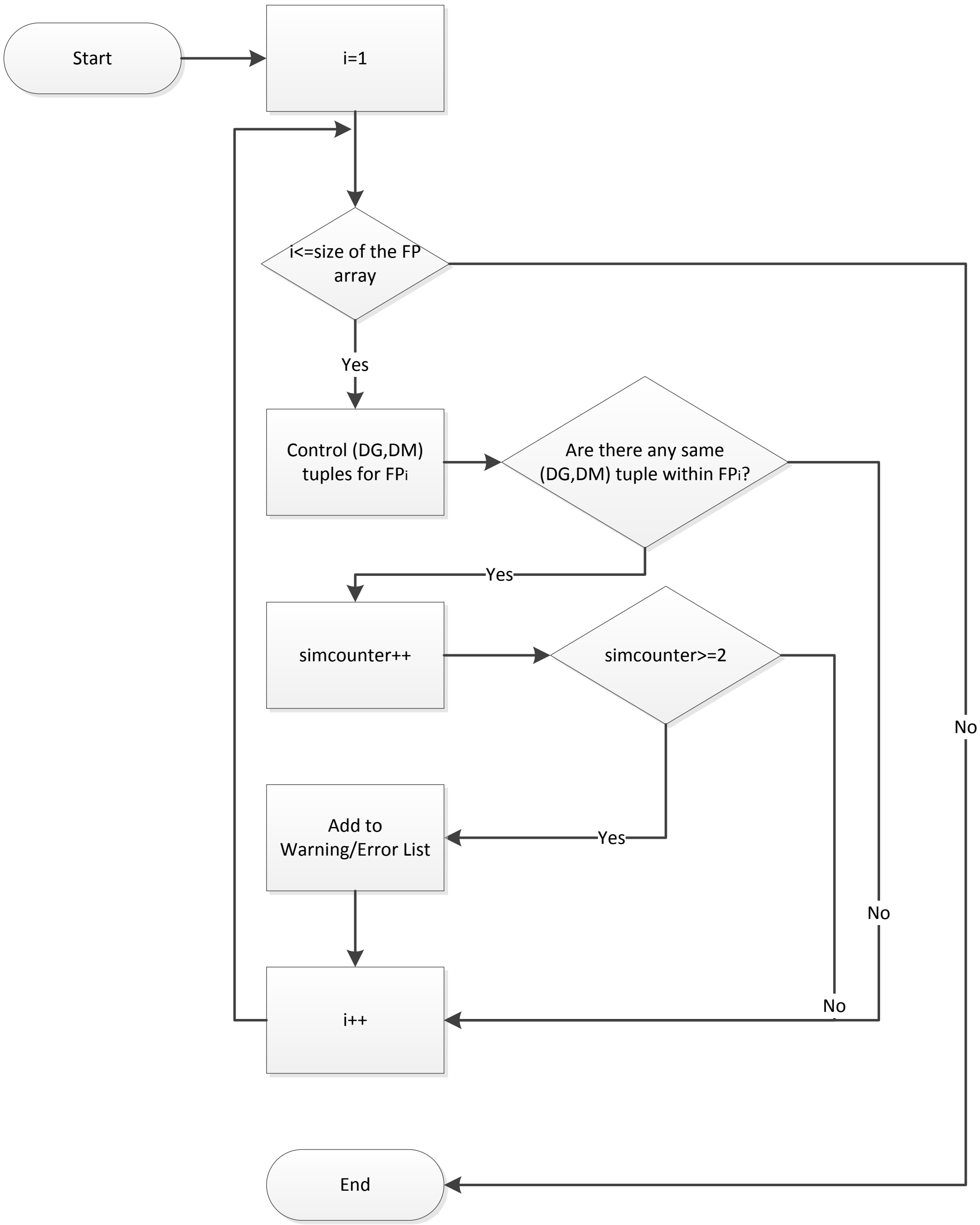


Figure 20: EC7 - Multiple occurrences of same DM within the same FP

3.2.1.8. Algorithm for EC8

Figure 21 explains the algorithm of EC8 in detail. For each FP of the FP array, system counts the number of DMs. If the FP has less than 2 DMs, then system gives an error message.

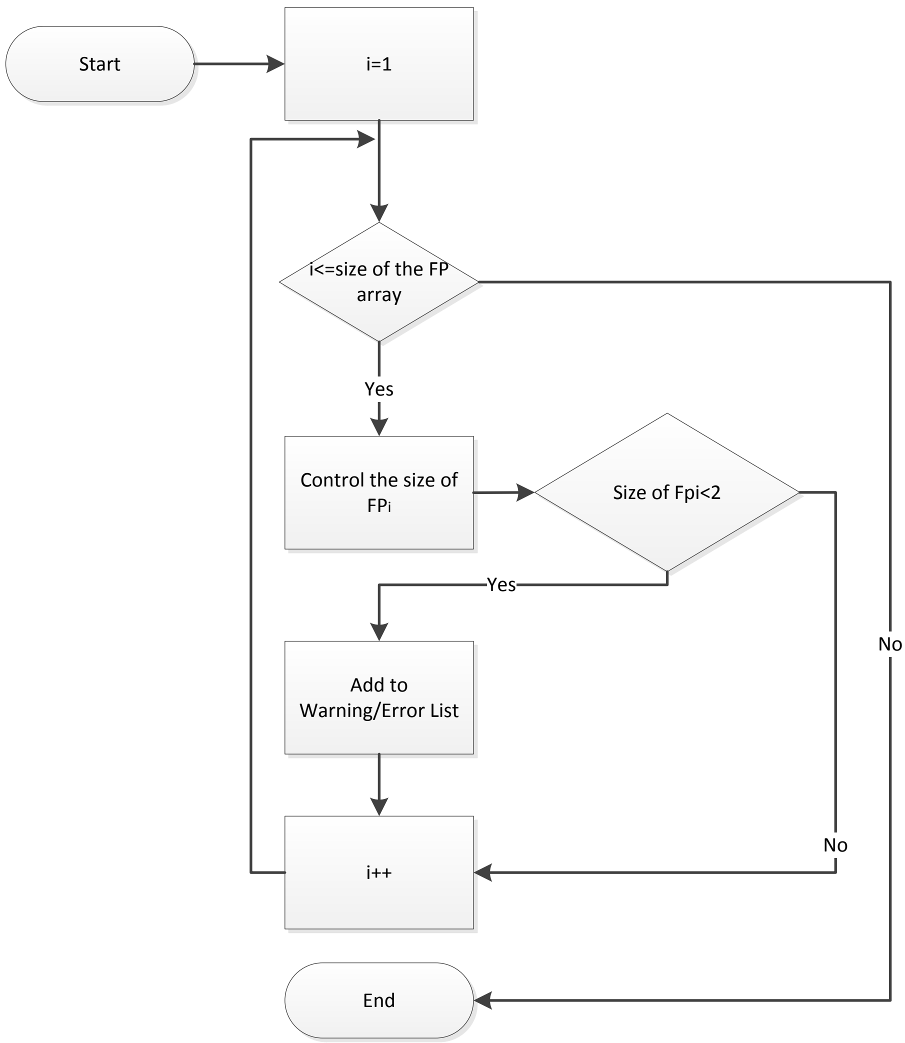


Figure 21: EC8 - Each FP should be composed of at least 2 DMs

3.2.1.9. Algorithm for EC9

Figure 22 explains the algorithm of EC9 in detail. For each FP of the FP array, system controls the type of each DM. The system gives a warning message if the flag is equal to zero.

Start

If(it.type.label=W | it.type.label =X)

flag++

End if

End

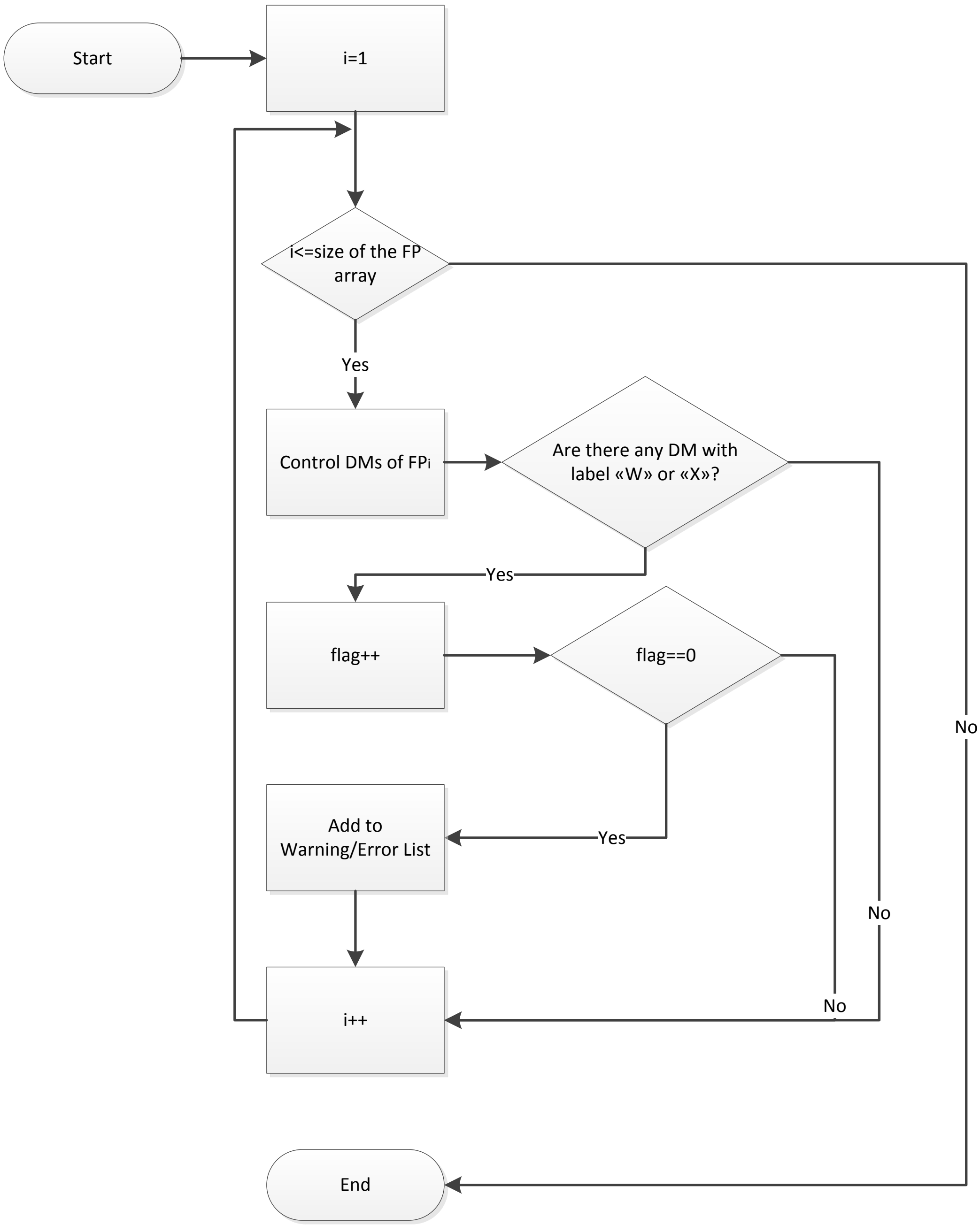


Figure 22: EC9 - Each FP should contain at least 1 W/X DM

3.2.1.10. Algorithm for EC10

Figure 23 explains the algorithm of EC10 in detail. For each FP of the FP array, system controls the type of each DM. The system gives a warning message if the eCounter is equal to zero.

Start

If(it.type.label =E)

eCounter++

End if

End

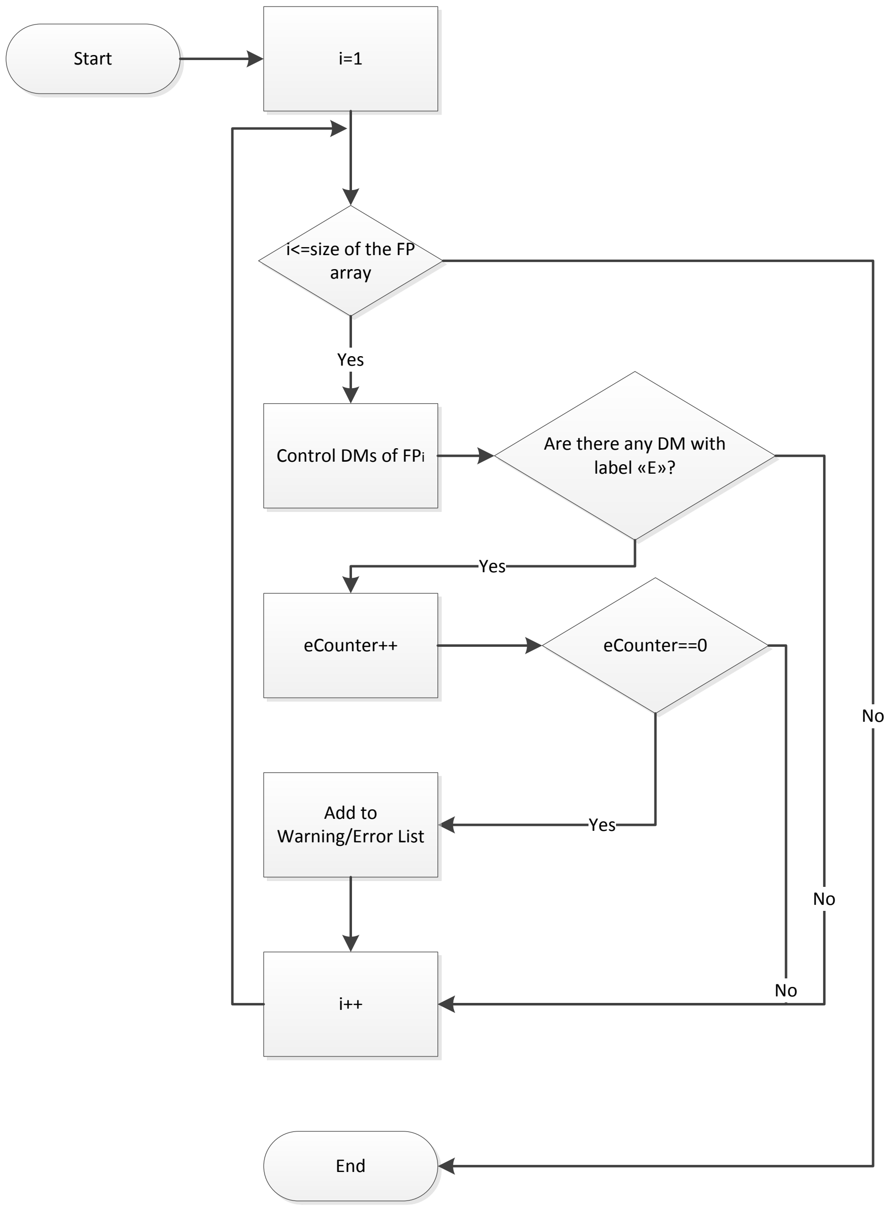


Figure 23: EC10 - Each FP should contain at least 1 E DM

3.2.1.11. Algorithms for EC11 and EC12

For error categories EC11 an EC12, the tool gives general warnings to users. Exact point of defect cannot be identified for these error categories.

As given in Table 2, defects of EC11 and EC12 are being detected based on statistical algorithms. For these calculations, the tool gathers required data from CUBIT database.

In order to make correct calculations and give meaningful warnings to users, the tool get data concerning measurements of software projects of which application type is the same as the selected work project. Based on those findings it only gives warnings to users where the problem may occur.

Figure 24 explains the algorithm of EC11 in detail. System gives a warning message if the following condition is maintained;

```
Start
    If(expectedR<=dmRCount  &  listCount<expectedlist)
        print warning
    End if
End
```

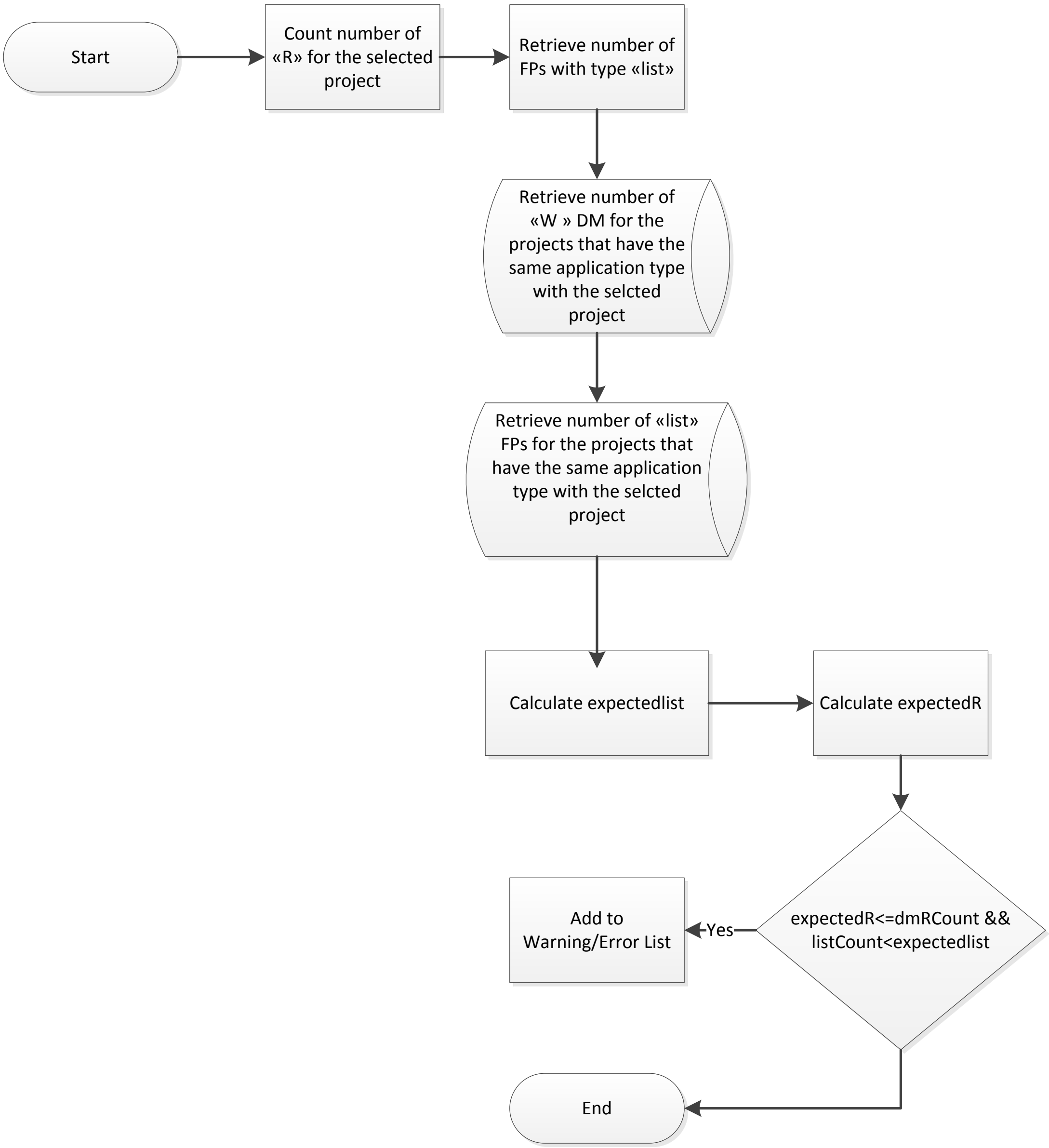


Figure 24: EC11 - List FP might be included in Update/Delete FPs

Figure 25 explains the algorithm for EC12 in detail. System gives a warning message if the following condition is maintained;

Start

If(dmWCount<= expected & (createCount<expectedcreate | updateCount<expectedupdate | deleteCount<expecteddelete)

print warning

End if

End

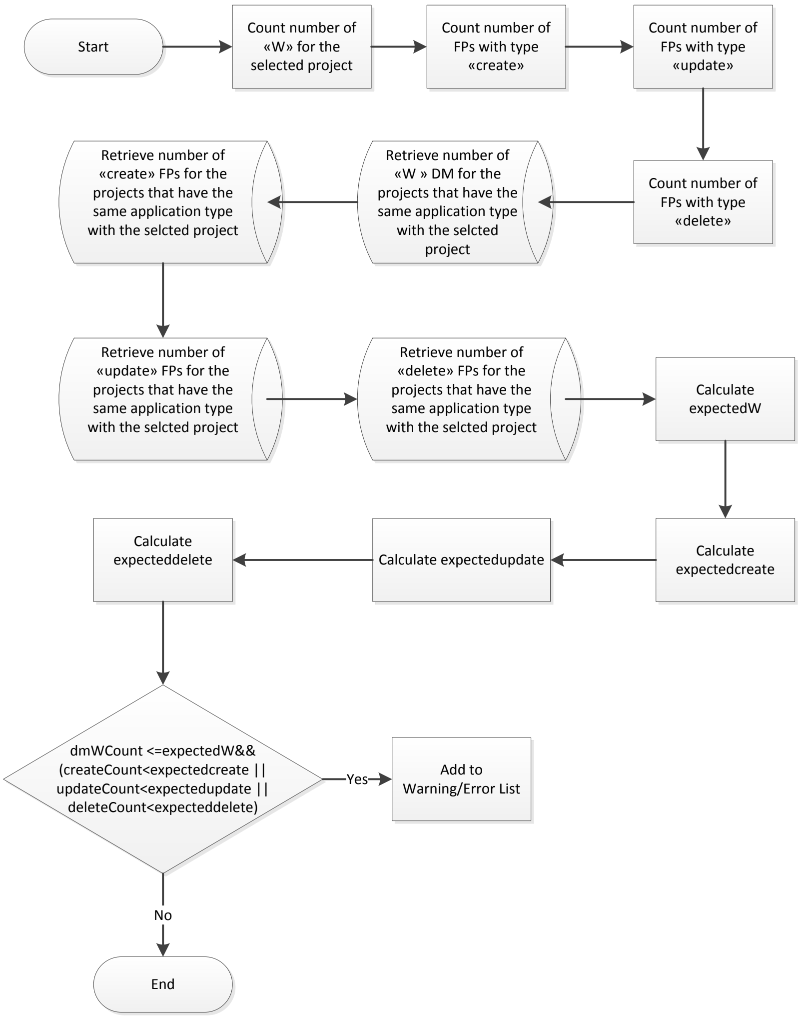


Figure 25: EC12 - Create/Delete/Update operations might be combined

3.2.1.12. Algorithm for EC13

In order to detect EC13 DAs should be entered. If DA field is null, system cannot detect existence of defects in error category 13.

Figure 26 explains the algorithm for EC13 in detail. For each OOI, the tool compares DGs with each other. If two different DG of the same OOI contains exactly the same DAs, the tool creates a warning message.

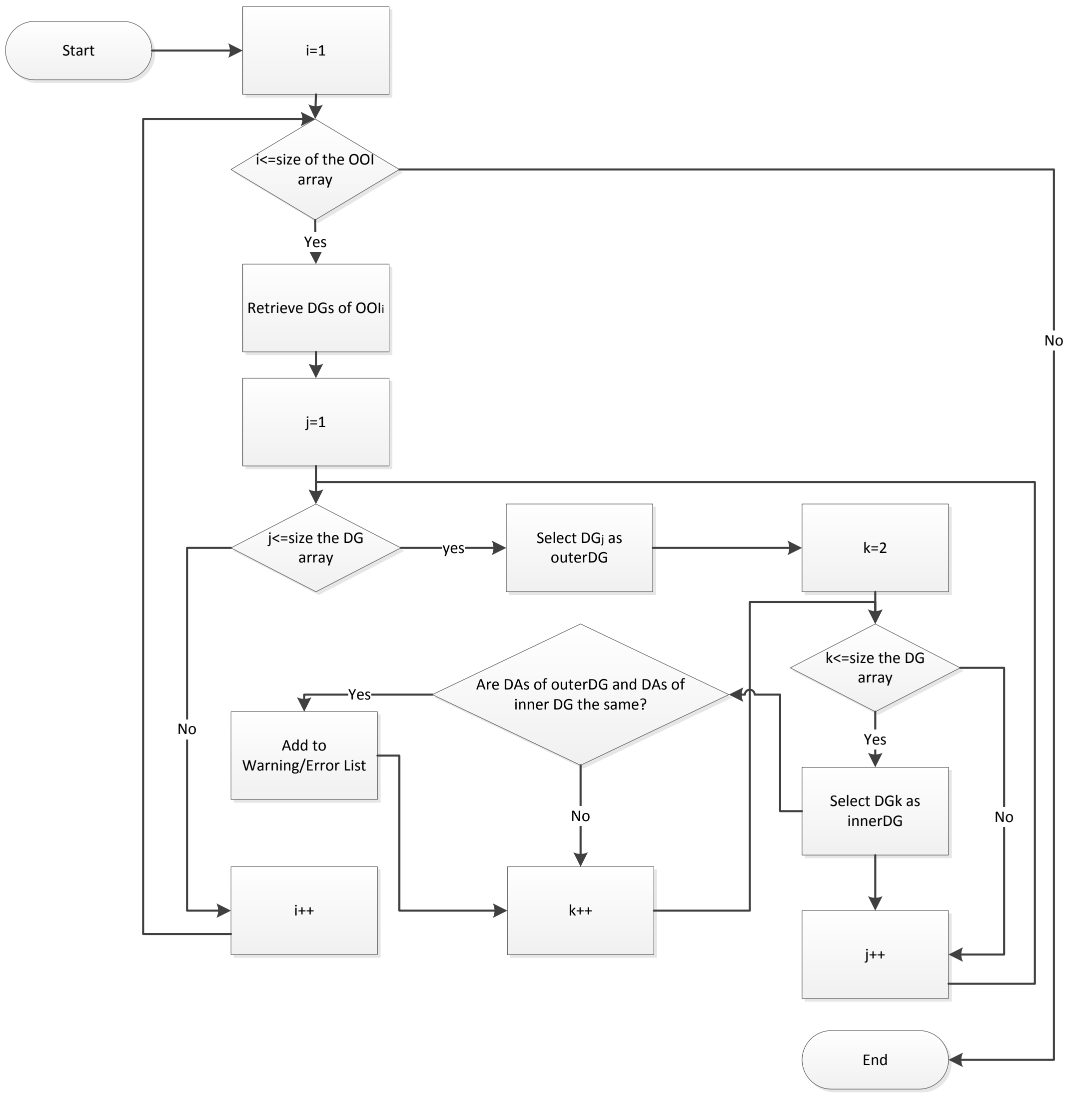


Figure 26: EC13 - DG Duplication

3.2.1.13. Algorithm for EC14

For each DG and OOI, system gives a warning message, if the DG or OOI names contain keywords “command”, “button”, “menu” or “screen”. Figure 27 explains the algorithm for EC14 in detail.

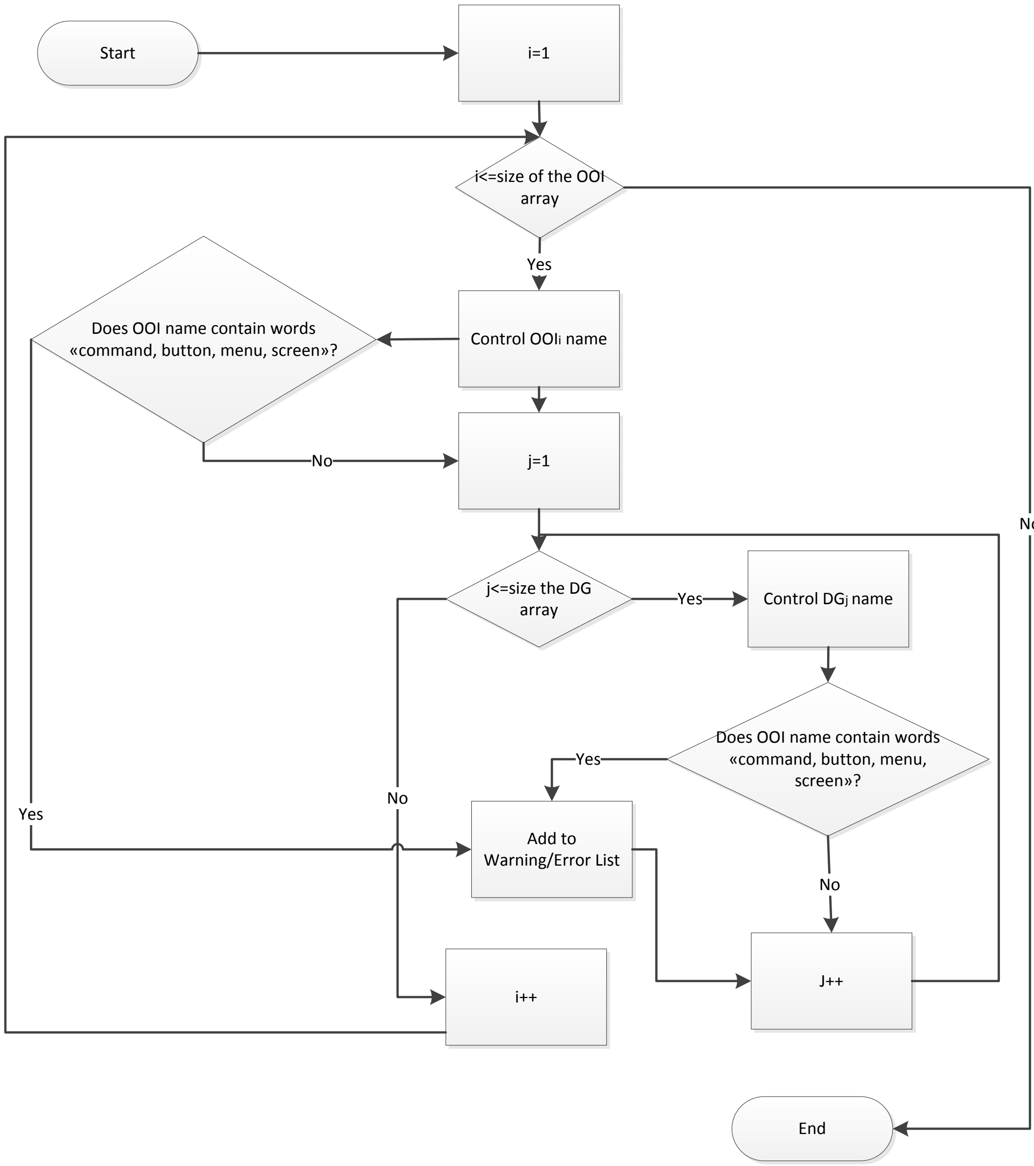


Figure 27: EC14 - User interface components and System users are considered as DG/OOI

3.2.1.14. Algorithm for EC15

For each OOI, system gives a warning message, if OOI name contains keywords “select”, “search”, “save”, or “query”. Figure 28 explains the algorithm for EC15 in detail.

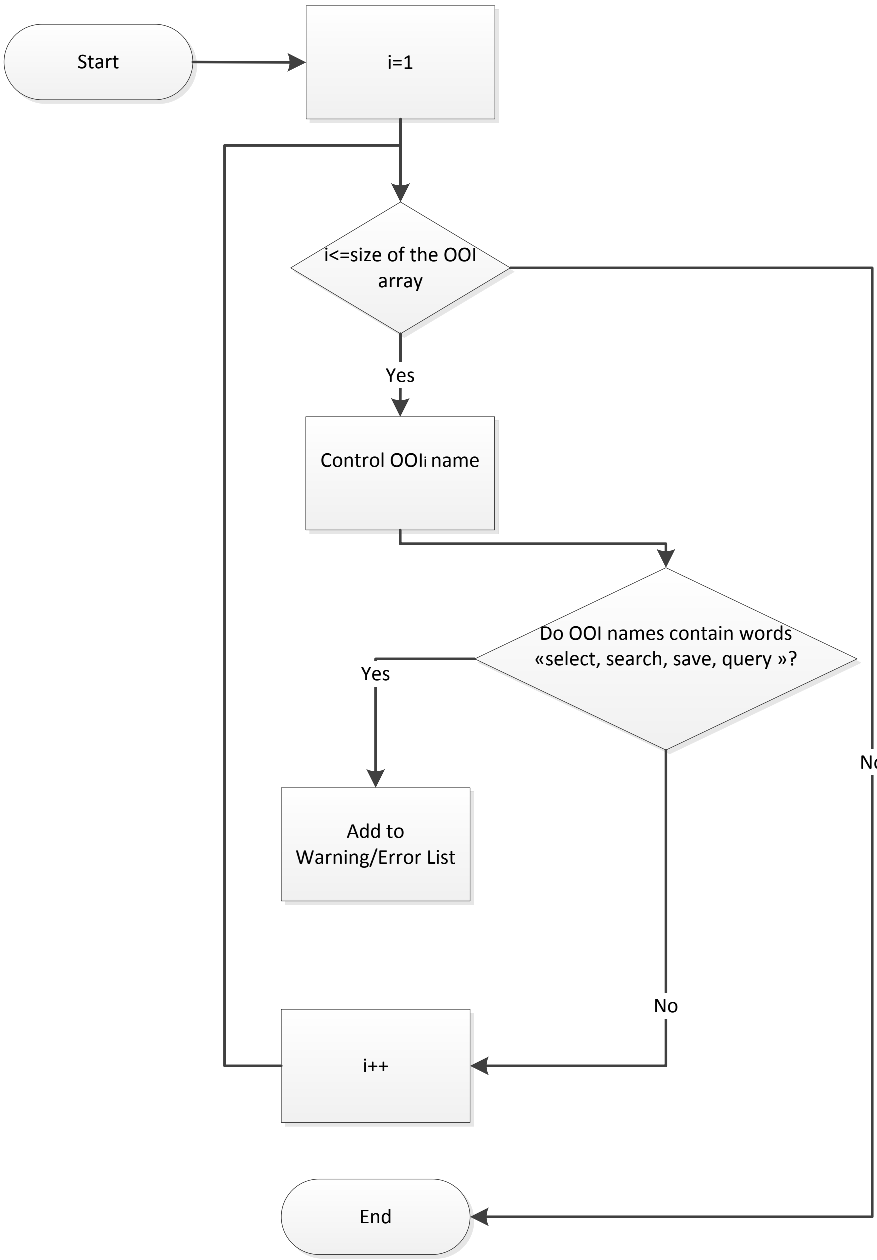


Figure 28: EC15 - OOIs are named wrong

CHAPTER 4

CASE STUDY

In this chapter, plan, implementation and results of explanatory and validation case studies are introduced in detail.

4.1. Research Questions

The main aim of this thesis is to investigate the automation possibility of detecting defects concerning COSMIC size measurements in MIS software applications by developing and using a COSMIC FSM DDA. In accordance with this purpose four research questions are created as follows:

Research Question 1: Can most commonly made COSMIC errors be classified into error categories?

The aim of the 1st research question is identifying the most commonly made errors in COSMIC Measurements, and classifying these errors into error categories.

Research Question 2: Can a prototype for MIS applications called CUBIT COSMIC FSM DDT be developed for detecting defects automatically?

Aim of the 2nd research question is to develop a prototype which is called CUBIT COSMIC FSM DDT for MIS applications to detect COSMIC measurement defects automatically.

Research Question 3: Can an approach for MIS applications called COSMIC FSM DDA be developed and based on the Approach, prototype be improved to generate more efficient defect detection results?

Main aim of the 3rd research question is to develop a COSMIC FSM DDA for MIS applications. After improving the prototype of CUBIT COSMIC FSM DDT based on this approach, to determine whether the Tool creates correct defect detection results or not.

Research Question 4: Is COSMIC FSM DDA extensive enough to work on different software application domains?

By answering the 4th research question, our main aim is to determine whether the COSMIC FSM DDA for MIS Applications is able to produce reliable results for COSMIC measurements which belong to different type of software projects such as RT and Embedded software projects.

Following sections are developed to answer these research questions. Section 4.2 is Case Study Design which gives overall information about all of the case studies. Section 4.3, 4.4 and 4.5 explain detailed information about Case Study 1, 2 and 3, respectively.

4.2. Case Study Design

First and second research questions will be answered in the Case Study 1. To answer the first research question, by utilizing expert review mechanism, error categories are identified and clearly defined. Based on these error categories, the prototype of the CUBIT COSMIC FSM DDT for MIS Applications is developed. To answer the second research question, effectiveness of the Prototype correctness should be observed by comparing the results of expert review process and results found via prototype. Therefore first and second research questions should be answered within the same case study.

Third research question requires a single case study and will be answered in the Case Study 2. To improve the effectiveness of the Prototype, a COSMIC FSM DDA for MIS Applications is developed. Based on the approach Prototype is improved and it is called CUBIT COSMIC FSM DDT for MIS Applications. Effectiveness of the Tool correctness is required to be determined by comparing three different defect detection results which are found by utilizing three different methods, the Expert Review, the Prototype and the Tool.

To identify the effectiveness of the Tool correctness on COSMIC measurements which belong to different application domains, another case study is needed. Therefore, fourth research question will be covered in Case Study 3.

Consequently, 1st and 2nd research questions will be answered in Case Study 1, 3rd research question in Case study 2 and 4th research question in Case study 3. Therefore, we have designed a total of three case studies to answer all of the research questions.

Case selection criteria, detailed case study design and implementation for each case study, will be given under each case study section.

4.3. Case Study 1: Exploration of the Error Categories and development of prototype of CUBIT COSMIC FSM DDT for MIS Applications

Case study 1 forms the starting point of development of both CUBIT COSMIC FSM DDT and COSMIC FSM DDA for MIS applications which were explained in Chapter 3. One of the aims of the 1st case study is to identify error categories of COSMIC measurements and to define them clearly. After identification and clarification of the error categories, a prototype of COSMIC FSM DDT for MIS applications will be developed and will be tested, and results will be analyzed.

In this section, selection criteria and background information of the selected cases are given. Plan, implementation and results of case study 1 are explained in detail.

4.3.1 Case Selection Criteria

For the 1st case study, we have selected 10 different COSMIC FSM cases of HRWOMS. There are several reasons of selecting these cases.

First of all, COSMIC measurements of HRWOMS belong to the same software project Human Resource Management System. To identify error categories easily, different COSMIC FSM measurements that belong to the same MIS software project are required to be selected.

Secondly, expert review technique will be utilized for errors categorization. Thus, SRS knowledge is required. Since we have the SRS of HRWOMS, these cases were selected.

Thirdly, since measurements are conducted by inexperienced measurers, possibility of having a wide variety of error types is higher than that of industrial cases.

Finally, most of the cases in industry, SRS creation and size measurement of software projects are performed by different work groups. As, SRS creation and COSMIC measurement of HRWOMS project were performed by different participants, it fits to a real life software project life cycle. Detailed information about SRS creation and size measurements of HRWOMS is given under Section 4.3.2.

4.3.2 Background information about the selected cases

Selected 10 measurement cases of HRWOMS were performed by the students of the "Software Project Management" graduate course in Middle East Technical University. For the measurement process SRS were given to the groups of students and they were asked to measure the size of the system by using COSMIC FSM method (ISO/IEC, 2003b). Every group was given the same SRS document which was prepared by the students of the graduate course "Information Systems Project" graduate course, before.

HRWOMS is a MIS application and a web based Human Resource Management System. The system includes four user types, manage and report personnel information, work orders, effort records, trainings, and announcements.

Before the measurement, the students were trained in COSMIC method. The training was composed of two 3-hour lectures and an interactive measurement workshop. At the end of the workshop students were given a small pilot project for measurement. Students formed 10 groups of 2 or 3. The level of industrial and academic experience was similar between randomly selected 2 groups.

At the end of the measurement process, measurement reports of HRWOMS were collected in the form of spreadsheet.

4.3.3 Case Study Plan

To realize the aims of the Case Study 1, we have two activities as;

- To identify error categories of COSMIC FSMs concerning MIS Applications, and to detect defects based on these error categories
- To identify possibility of prototype development concerning CUBIT COSMIC FSM DDT for MIS Applications and to determine the effectiveness of the Prototype correctness and effort savings

4.3.3.1. Activity 1: Exploration of the Error Categories and Defect Detection based on the Error Categories

As depicted in Figure 29, at the end of the Activity 1, a total of two case products will be gathered.

- Error categories
- Based on each Error Category, detected defects belong to each measurement case

Figure 29: Inputs & outputs of The Expert Review Process

First of all, Error categories will be determined. Then, based on these error categories, defects of each measurement case will be found. For both of these products the Expert Review Mechanism will be facilitated.

In order to get these outputs, some inputs should be prepared to be used during the Expert Review. Thus, prior to the Expert Review process, a preparation process is required. In the context of Activity 1, two processes will be performed,

- Preparation to Expert Review Process
- Expert Review Process

Preparation to Expert Review Process

A total of two documents will be prepared to be used in the Expert Review as;

- A reliable COSMIC measurement of HRWOMS which is referred as the “Key”,

- A defect marking template to control each measurement case with respect to the Key.

Prior to the Expert Review process, first of all, a reliable measurement of HRWOMS is required, and thus as depicted in Figure 30, based on SRS of HRWOMS, a reference measurement called the “Key” will be generated by the author of this thesis. Later, prepared Key will be peer reviewed by a group of experts and will be updated according to results of this review. Reviewers will be selected from Software Management Research Group (SMRG) researchers who have at least 3 years FSM experience.

Since measurement cases are submitted in spreadsheet format, to make comparison process easier, the Key is prepared in the form of spreadsheet.

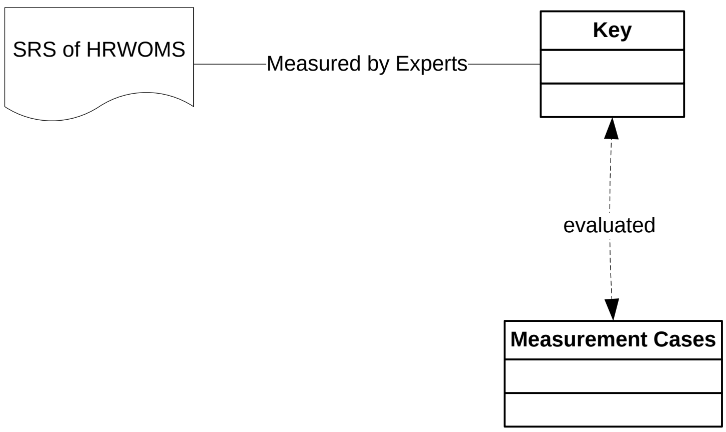


Figure 30: Flow of the Key Preparation and the Evaluation for Error Categories Identification

Secondly, to evaluate each measurement case with respect to the Key easily, a defect marking template will be created.

Both the reference key and the defect marking template will be facilitated for both of the Error Categories identification and the defect detection.

Effort spent for the preparation of the Key and the defect marking template will be collected in Tempus, which is a web based effort collection tool and being used by the members of SMRG in Middle East technical University.

Expert Review Process

As mentioned before, two case products will be collected at the end of Activity 1. These are the Error Categories and the defects of the measurement cases.

As depicted in Figure 30, by using defect marking template, each measurement case will be evaluated based on the Key in order to identify error categories.

To identify error categories for each measurement case, we will perform several activities as;

- Counting missing and redundant FPs
- Counting missing and redundant DMs
- Controlling OOIs and DG.

After collecting the indicated information, defects of each measurement will be detected and recorded in defect marking template. Findings belong to each measurement case; will be recorded in different results documents, separately.

Later, for the error categories identification each results document will be compared with each other and we will try to cluster the defects. If the same defect is encountered in more than two different measurement cases, we identify it as an error category.

By collecting all of this information, and taking the researches, which are explained in Chapter 3, as reference, error categories will be determined and defined clearly. Definition of each error category is given in detail, in Chapter 3.

After identification of the error categories, defects concerning error categories will be identified. Based on each error category and the Key, each measurement document will be reviewed again. Found defects will be recorded in a spreadsheet.

Effort spent for error categories identification and defect detection based on the error categories, will be collected in Tempus.

4.3.3.2. Activity 2: Prototype development possibility for the Tool

In the context of activity 2 we will perform the following processes;

- Investigation of the possibility of developing a prototype of COSMIC FSM DDT for MIS applications and the prototype development
- Determination of the effectiveness of prototype correctness and effort savings

Investigation of prototype development possibility

To develop a prototype of CUBIT COSMIC FSM DDT for MIS applications, firstly, error patterns will be investigated for each error category. To form an error pattern for each error category, defects found at the end of activity 1 will be analyzed for each measurement case. If same patterns are introduced in at least two different measurement cases, pattern represents the error category. If an error category does not constitute a pattern and its defects require SRS knowledge to be detected, we cannot create an algorithm for the error category. Error categories, that have NA in DDL column in Table 2: Error Categories and Error Patterns , cannot be detected automatically.

After the identification of error patterns, for prototype development, defect detection patterns will be written and will be peer- reviewed by an expert who works in industry and has COSMIC FSM experience. Later, they will be turned into defect detection algorithms. Collection of algorithms forms the prototype. Prototype will be developed integrated into CUBIT which is explained in Chapter 3, and for the development grails programming language and Intellij idea will be used.

Effort spent for error patterns identification and for the development of the prototype concerning CUBIT COSMIC FSM DDT will be collected in Tempus.

Determination of effectiveness and effort savings of prototype

To determine the effectiveness of the prototype correctness as depicted in Figure 31: Overview of determination of prototype , based on each error category results found via the prototype will be compared to that of the expert review process.

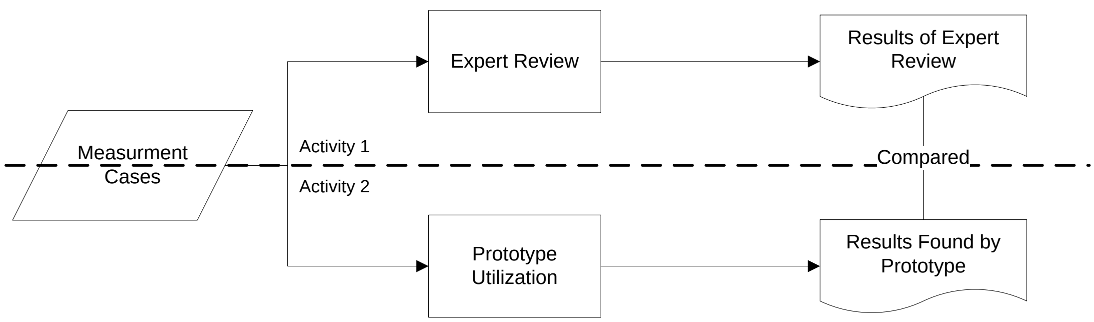


Figure 31: Overview of determination of prototype effectiveness

- Defects found in the expert review process will be gathered from Activity 1.
- To gather the results of the prototype, measurement cases will be uploaded into CUBIT.
- To prevent any data loss while uploading the measurement cases, for each measurement case, spreadsheet and CUBIT formats will be checked with each other.
- Later, for each measurement case, prototype will be facilitated. As results of expert review are in the form of spreadsheet, results found by prototype will be recorded in a spreadsheet.
- To compare the results of the expert review and the results found by the prototype easily, they will be united in one spreadsheet.
- To determine the effectiveness of the prototype correctness, based on the each error category, results will be analyzed.

Effort spent for uploading measurement cases into CUBIT and controlling them, and the prototype utilization will be recorded manually.

Lastly, effort spent for defect detection by expert review mechanism, which is performed in activity 1, and defect detection by utilizing prototype will be compared to calculate effort savings of the prototype.

4.3.4 Case Study Implementation

As illustrated in Section 4.3.3 in the context of Case Study 1, we performed two activities. Implementation information of each activity is given in the following sections.

4.3.4.1. Activity 1: Expert Review Process: Exploration of the Error Categories and Defect Detection based on Error Categories

To achieve the goals of the activity 1, firstly, several documents were prepared in order to be used in the expert review.

By using expert review method, error categories were identified and later defects of each measurement case were detected based on the error categories.

Preparation to Expert Review Process

First of all, prior to the expert review process, SRS of HRWOMS was discussed and measured by the author of this thesis. Later, a group of COSMIC FSM experts reviewed the measurement. According to findings of this review, measurement was updated and was named as the “Key”. It was used as the basis for the error categories determination and defects detection of each measurement case.

Key was generated in spreadsheet format and consists of 83 FPs and 366 DMs.

Secondly, in order to use in the identification of error categories, a defect marking template was created based on the Key.

Effort spent for the preparation of the Key and the defect marking template, was collected in Tempus and they are presented in Section 4.3.5.

Expert Review Process

To identify error categories, we checked each measurement case several times. As explained in Section 4.3.4.1, we counted missing & redundant FPs and DMs. OOIs and DGs were controlled manually.

Later, defects of each measurement case were detected and recorded in defect marking template. Defects of each measurement were marked in a different defect marking template. Then each results document was compared with each other several times. By checking results documents iteratively, we tried to cluster the defects and identify error categories. If the same defect was encountered in more than two different measurement cases, we identified it as an error category. By using these results and the most commonly made COSMIC errors of which researches are given in the literature review, error categories were identified and clearly defined. Definition of each error category is given in Chapter 3.

After determination of each error category, each measurement document was reviewed based on the error categories and the Key. Results were recorded in a spreadsheet which is called as “Results of Expert Review” within Case Study 1 to prevent any confusion. Effort spent for error categories determination and defect detection based on error categories is presented in Section 4.3.6.

4.3.4.2. Activity 2: Prototype of CUBIT COSMIC FSM DDT

In the context of the Activity 2, we aimed

- To investigate the error patterns for each error category and based on these error patterns to develop a prototype of CUBIT COSMIC FSM DDT for MIS applications
- To calculate the effectiveness of the prototype correctness.

Investigation of prototype development possibility

First of all, to investigate the development possibility of a prototype for CUBIT COSMIC FSM DDT for MIS applications, we tried to find out the error pattern for each error category. To search for a pattern for each error category; defects found at the end of activity 1 were analyzed for each measurement case. If same patterns are introduced in at least two different measurement cases, we said that a pattern exists for the specific error category. If an error category do not constitute a pattern and requires SRS knowledge to be detected, defect detection procedure of the error category cannot be developed. DDL of these error categories were indicated as NA in Table 2. Defects of them can only be detected by manually.

After determination of the error patterns, defect detection patterns were peer-reviewed by a COSMIC FSM measurement expert who works in industry. After the updates with respect to peer-reviews, the defect detection procedures were generated. Later, to develop the prototype following products were generated, respectively;

- The DDLs
- The Defect Detection Algorithms
- The Prototype of the Tool

DDLs that are derived from defect detection procedures are presented in Table 2. After the development of the DDLs were established. By using grails programming language, each algorithm is coded integrated into CUBIT. Logic of each defect detection algorithm is presented in Chapter 3.

Effort spent for error patterns identification and for defect detection procedures creation is presented in Section 4.3.5.

Determination of the effectiveness and effort savings of the prototype

To determine the effectiveness of prototype correctness, the Results of Expert Review and results found via the prototype are required to be compared. Results of the expert review issued at the end of Activity 1, thus we gathered them from Activity 1.

To have the results of the prototype, procedures which are explained in Section 4.3.3.2 were followed. First of all, prior to the prototype utilization, spreadsheets of measurement cases were uploaded into CUBIT. To prevent any data loss during the uploading, for each measurement case, spreadsheet and CUBIT formats were checked with each other. Later, the prototype was facilitated for each measurement case respectively. Findings of the prototype utilization were recorded in a spreadsheet and called as "Results of the Prototype" and were taken as reference values for comparison process.

During facilitating the prototype, we encountered with an exception related with the error categories EC11 "List FP might be included in Update/Delete FPs" and EC12 "Create/Delete/Update operations might be combined". Algorithms of error categories EC11 and EC12 have SYT DDL. It performs several calculations on the data which are gathered by queries from the CUBIT database. Thus, the prototype did not find the exact number of the defects concerning error categories EC11 and EC12. It only detected whether any defect might have existed in the measurement case or not. However, in the results of the Expert Review, exact number of the defects concerning error categories EC11 and EC12 were counted and presented in Figure 32 in Section 4.3.5. In order to compare the results found by the prototype with the results of the Expert Review, they should be in the same unit. Therefore we did not take the exact number of defects found in the expert review as reference; instead we took the number of measurement cases in which the defect was detected.

Results of Expert Review and Results Found by Prototype was united in one spreadsheet and compared to each other. Results are presented, and findings are discussed in Section 4.3.5.

Effort spent for uploading and controlling measurement cases, the prototype utilization are presented in Section 4.3.5.

At the end of Activity 2, effort spent for detecting defects by the expert review mechanism and that of defect detection by utilizing the prototype was compared for calculating effort savings and results of the comparison process are discussed in Section 4.3.5.

4.3.5 Results

4.3.5.1. Results of Activity 1

Error categories that were identified at the end of the activity 1 are presented in Table 4: Error Categories and their aspects. Definition of each category is given in Chapter 3. Error categories were classified based on three aspects,

- DDL
- Defect Cause (DC)
- Reference Source (RS)

Table 4: Error Categories and their aspects

DC	DDL	Error Categories		RS
MR	SEM	EC1	FP Duplication	NI
MPR	SEM	EC2	Lack of List FP before Update FP	COSMIC AG
MPR	SEM	EC3	Lack of List FP before Delete FP	COSMIC AG
MPR	SEM	EC4	Lack of Retrieve FP before Update FP	NI
MR	SEM	EC5	Lack of DM type W in Add, Delete, Update FPs	NI
MR	SEM	EC6	Redundant DM type W in List FPs	NI
MR	SEM	EC7	Multiple occurrences of same DM within the same FP	NI
MR	SEM	EC8	Each FP should be composed of at least 2 DMs	COSMIC MM
MR	SEM	EC9	Each FP should contain at least 1 W/X DM	COSMIC MM
MR	SEM	EC10	Each FP should contain at least 1 E DM	COSMIC MM
MPR	SYT	EC11	List Fp might be included in Update/Delete FPs	NI
MR	SYT	EC12	Create/Delete/Update operations might be combined	NI
continue on the next page				

continued from the previous page				
DC	**DDL**	**Error Categories**		**RS**
MR	SEM	EC13	DG Duplication	NI
MR	SYN	EC14	User interface components and System users are considered as DG/OOI	NI
MR	SYN	EC15	OOIs are named wrong	NI
MR	NA	EC16	Combined FP that belongs to the same FUR	NI
MR	NA	EC17	Considering DMs for irrelevant OOIs	NI
MR	NA	EC18	Assumed additional functionality	NI
MR	NA	EC19	Dropdown list is not properly used	COSMIC AG
MPR	NA	EC20	Measuring data manipulation within the system boundary	NI
SAR	NA	EC21	Data that are not attributes of OOIs	COSMIC AG
SAR	NA	EC22	OOI sub types are not considered	COSMIC AG
SAR	NA	EC23	Errors in data analysis and OOI identification.	NI

Based on the error categories, total number of defects found in the expert review are summarized Figure 32.

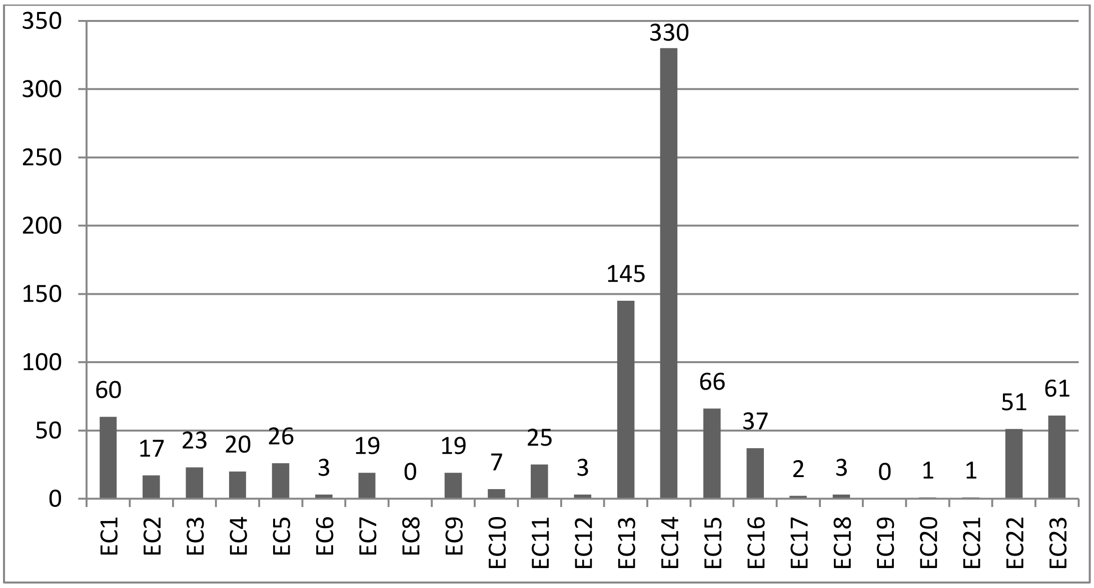


Figure 32: Total size of errors based on error categories.

Figure 33 represents the percentage of errors based on the Error Causes, MR, MPR and SAR. As can be inferred from Figure 33, errors are mostly related with Measurer. In order to increase the reliability of FSM, theoretical knowledge about COSMIC FSM Method should be practiced by workshops, pilot project etc.

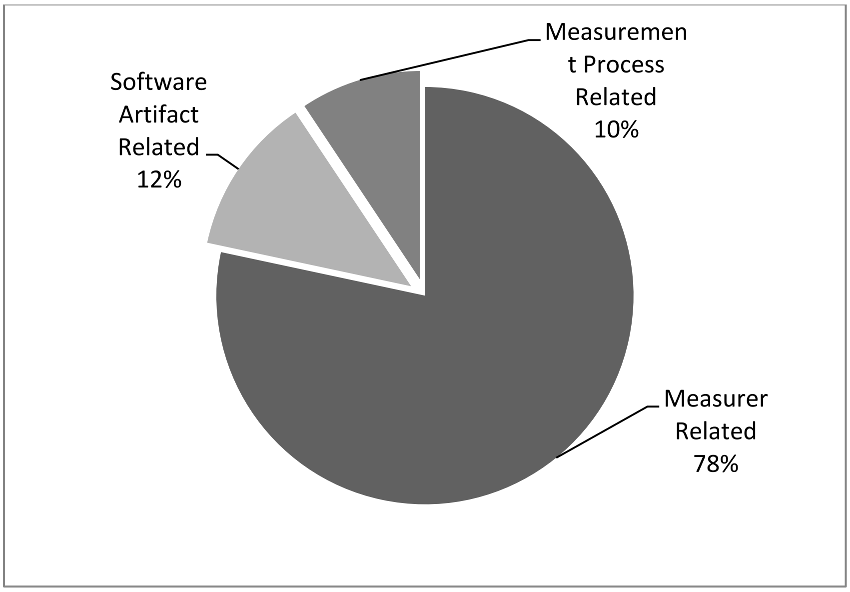


Figure 33: Percentage of errors based on error causes.

Figure 34 presents the percentage of error categories with respect to DDL. As can be inferred from the Figure 34, amount of the errors which are detected by SYN logic are much more than the errors which are detected by SEM and SYT DDLs.

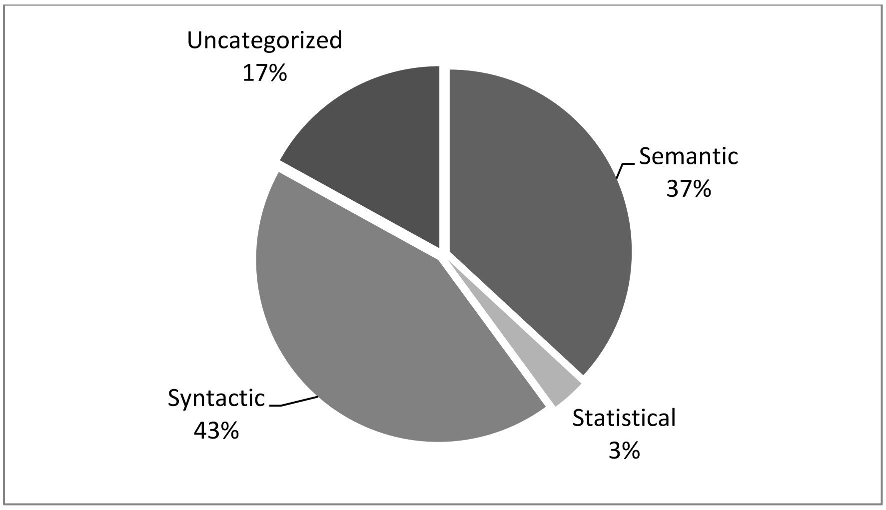


Figure 34: Percentage of error based on the DDLs

4.3.5.2. Results of Activity 2

Figure 32 and Table 4: Error Categories and their aspects contain only the categories that can be automatically detected by the Tool. EC1 to EC15 can be detected automatically.

For error categories EC1 to EC15 results found based on expert review and by the prototype are summarized in Figure 35: Total number of errors found by Prototype and Expert Review. Since defect detection algorithms of EC11 and EC12 are based on some statistical calculations, the prototype cannot investigate the exact number of defects concerning EC11 and EC12. It only gives warning about the possibility of the existence of defects concerning EC11 and EC12. While results for EC1-EC15, except from EC11 and EC12, were presented as total number of errors found in 10 different measurement report, results for EC11 and EC12 were presented as total number of measurement reports in which defect was detected. Thus, since errors found based on EC11 and EC12 were counted in different unit, different interpretations were required to represent them. EC11 and EC12 were excluded from Chart 3 and presented in Table 5.

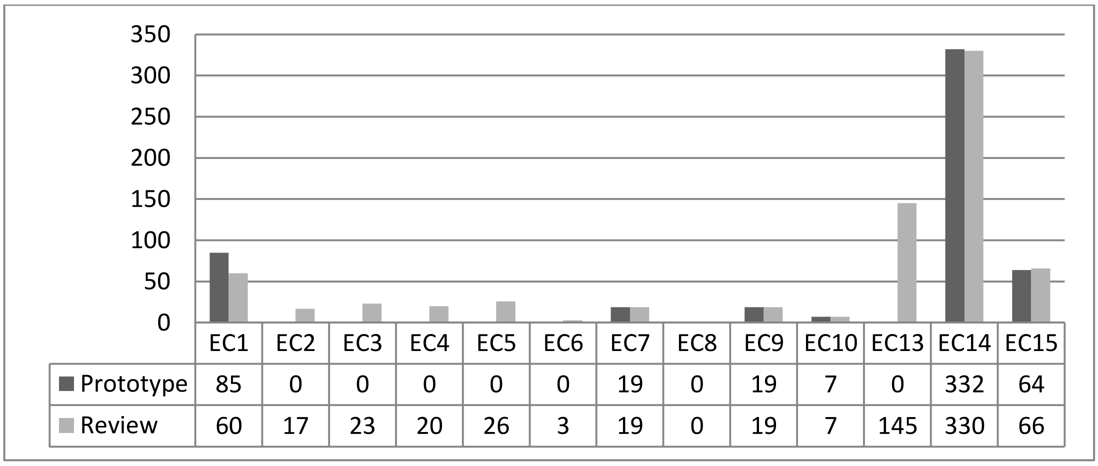


Figure 35: Total number of errors found by Prototype and Expert Review

Table 5: Results for EC11 and EC12

EC	Review	Prototype
EC11	7	0
EC12	3	0
Total	**10**	**0**

Table 6 presents the number of same defects that are found both in expert review and by prototype, total number of defects which are redundantly detected and that are missed to be detected by prototype.

Table 6: Number of Same & redundant & missing defects of the Prototype wrt the Expert Review

EC	Number of Defects found by Review	Number of Defects found by Prototype	Number of Same Defects	Number of Redundant Defects	Number of Missing Defects
EC1	60	85	53	32	7
EC2	17	0	0	0	17
EC3	23	0	0	0	23
EC4	20	0	0	0	20
EC5	26	0	0	0	26
EC6	3	0	0	0	3
EC7	19	19	16	3	3
EC8	0	0	0	0	0
EC9	19	19	19	0	0
EC10	7	7	7	0	0
EC11	7	0	0	0	7
EC12	3	0	0	0	3
EC13	145	0	0	0	145
EC14	330	332	322	10	8
EC15	66	64	64	0	2
Total	**745**	**526**	**481**	**45**	**264**

Table 7 presents the effectiveness of the Prototype correctness.

Table 7: Effectiveness of the Prototype wrt the Expert Review

EC	Effectiveness	Percentage of Missing Defect	Percentage of Redundant Defect
EC1	88%	12%	38%
EC2	0%	100%	NA
EC3	0%	100%	NA

continue on the next page

continued from the previous page			
EC	**Effectiveness**	**Percentage of Missing Defect**	**Percentage of Redundant Defect**
EC4	0%	100%	NA
EC5	0%	100%	NA
EC6	0%	100%	NA
EC7	84%	16%	16%
EC8	N/A	N/A	N/A
EC9	100%	0%	0%
EC10	100%	0%	0%
EC11	0%	100%	NA
EC12	0%	100%	NA
EC13	0%	100%	NA
EC14	98%	2%	3%
EC15	97%	3%	0%

For EC1, prototype detects same defects with 88% effectiveness. It missed to detect defects with 12% and detects 38% more defects. To investigate origin of the problems that caused redundant and missing defects, we checked each group's measurement reports manually. When two different FP which have exactly same (DG, DM) tuple, the prototype detects it as a defect. For instance; two different FPs that has same (DG, DM) tuple are given as follows;

Update Personnel	Person Info	E
	Person Info	R
	Person Info	X
Add Personnel	Person Info	E
	Person Info	R
	Person Info	X

These two FPs are different than each other. However since algorithm of "Duplicate FP" error category search for FPs that has exactly same (DG, DM) tuples, it detects the given tow FPs as duplication and gives a warning message.

For the error category EC2, the prototype detected no defects based on the error categories. For detection of defects concerning error category EC2 "Lack of list FP before update FP" requires, FP type knowledge is required. The prototype cannot infer the type of the FPs from the patterns of (DG, DM) tuples. FP type knowledge should be added as a filed. As there is not any attribute FP type in the existing CUBIT database, it missed to detect defects with 100%. Since there are no redundant defects, it is represented in Table 7: Effectiveness of the Prototype wrt the Expert Review as "NA".

The prototype failed to detect defects concerning error categories EC3, EC4, EC5, and EC6 same as the error category EC2.

Defects of EC7 were detected in 8 different measurement documents. Prototype can detect same defects with 84% effectiveness. It missed to detect defect with 16% and it detects 16% more defects.

Prototype detects a defect concerning EC7, if a FP has more than or equal to 2 DMs which move same DG with same DM type. For instance, a FP that has two DMs which move DG "New Password" with same DM type "E" is given as follows;

Change Password	Triggering Entry	E
	New Password	E
	New Password	E
	Error Message	X

(New Password, E) DMs move same DG "New Password" with same DM type. According to the findings of the prototype, one of these DMs is redundant and it gives a warning message. However, according to the SRS of the related system, this situation is not a defect. Thus, we did not count this as a defect in the expert review.

Since no defects of error category EC8 detected in the measurement cases, we could not calculate the effectiveness of prototype correctness based on the error category EC8.

Defects of EC9 were detected in 3 different measurement documents, and prototype detects 100% same defects with expert review results. As presented in Table 2, error category EC9 is created based on a fundamental rule of COSMIC FSM Measurement Method (ISO/IEC, 2003b), and thus detecting the defects concerning EC9 is very objective and do not require any SRS knowledge and judgment.

Defects of EC10 were detected in 4 different measurement documents, and prototype detects 100% same defects with expert review results. Prototype works on EC10 efficiently because of the same reason with EC9.

The prototype failed to detect defects concerning error categories EC11, and EC12 same as the error category EC2.

It failed to detect defects concerning error category EC13 “DG Duplication”. If two different DGs contain exactly the same DAs; defects concerning this error category exist. Since DA attributes do not exist in the existing CUBIT database, there are not any defects detected.

For the error category EC14 prototype can detect 98% same defects with expert review results. It missed to detect defects with 2% and detects 3% more defects.

Prototype seemed to detect redundant defects concerning EC14. Since algorithm of EC14 checks syntactical correctness of the DG and OOI naming, controlling each OOI and DG by manually may result in inaccurate results. As presented in Table 6, defects concerning these error categories are repetitive and were done frequently, during manual inspection some of the defects were missed and some were identified as defects, inaccurately.

For the error category EC15, prototype detects same defects with %97 effectiveness. It missed to detect defects with 3%. To identify the reasons of missing defects, we examined each measurement case and we recognized that some of OOI data of measurement case 10 were skipped while uploading measurement cases into CUBIT.

Effort spent for Case Study 1 is presented in Table 8.

Table 8: Effort spent for Expert Review Process & Prototype utilization

ITEM	**Effort (mins)**	**Effort (hours)**
Key Preparation	860.00	14.33
For expert review, preparation of defect detection template based on Key	510.00	8.50
Evaluation and error categories identification	2440.00	40.67
Defect detection based on error categories	2910.00	48.50
Statistical calculations	240.00	4.00
Expert Review (Total)	**6960.00**	**116.00**
Pattern identification and creation of defect detection procedures	840.00	14.00
Uploading measurement cases into CUBIT and controlling them.	1200.00	20.00
Facilitating prototype	20.00	0.33
Prototype (Total)	**2060.00**	**34.33**

Effort spent for controlling measurement cases by expert review method equals to 116 person*hours. For the prototype control we spent time for importing measurement cases into CUBIT and thereafter checking them. For prototype utilization we spent a little time. Effort spent for the prototype control equals to 34.33 person*hours. Thus, prototype can detect defects with less effort.

We conclude that for a significant number of error categories (EC7, EC9, EC10, EC14, and EC15) the error detection process can be automated. The case study demonstrated that the prototype can detect 67% of defects that are detected by the experts. For prototype utilization, measurers require significantly less effort and domain knowledge.

Additionally, we observed that the prototype detected no defects concerning error categories EC2, EC3, EC4, EC5, EC6, EC11, EC12, and EC13. The case study showed that in order to detect the defects of these error categories automatically, further researches are required, and the prototype should be improved.

4.4. Case Study 2: Improving the Prototype based on the Approach and Determination of the effectiveness of the Tool Correctness

Detailed information about Case Study 2 is given under this section.

4.4.1. Overview

Case study 1 showed that tool detected no defects based on the error categories EC2, EC3, EC4, EC5, EC6, EC11, EC12, and EC13. Additionally, although COSMIC FSM manual (ISO/IEC, 2003b) is unique, most of the industrial software projects are measured in different formats. For these two reasons, to make the prototype work more efficiently among a wide variety of measurement formats, the prototype should be improved based on the COSMIC FSM DDA for MIS applications. Main aim of the Case Study 2 is, after improving the prototype according to COSMIC FSM DDA for MIS applications, to investigate the effectiveness of the CUBIT COSMIC FSM DDT correctness on a wide variety of measurements of MIS software projects. Effectiveness of the tool correctness will be identified on three different groups which are composed of COSMIC measurements of different software projects. We will the results of the expert review found by the prototype and found by the tool for each group.

CUBIT COSMIC FSM DDT for MIS applications is the improved version of the prototype. It is improved based on the COSMIC FSM DDA for MIS applications which is explained in detail, in Chapter3. Differences between the prototype and the tool are visualized in Chapter 3.

4.4.2. Case Selection Criteria

For each group, measurements of MIS software projects should be selected, since the tool is developed based on MIS software projects.

Effectiveness of the tool correctness will be tested on;

- Group 1: 10 different COSMIC measurements of HRWOMS which were used in Case Study 1
- Group 2: 11 different COSMIC measurements of CUBIT application
- Group 3: COSMIC Measurements belong to 5 different industrial MIS applications

Group 1: Measurements of HRWOMS

First of all, effectiveness of the tool correctness is required to be compared with both of the results concerning expert review and results found by the prototype. In the context of the Case Study 1, measurement cases of HRWOMS were used to identify the effectiveness of the prototype correctness. To achieve this goal results of Expert Review Process and results of the Prototype utilization were identified. Therefore, results of the Expert Review Process and the results found by the Prototype are available from Case Study 1. By using these currently presented results, identifying the effectiveness of the Tool correctness on HRWOMs measurement cases come to the first place. Hence, we selected measurements of HRWOMS as measurement cases of Group 1.

Group 2: Measurements of CUBIT

Effectiveness of the Tool correctness is required to be tested on different measurement cases. Since measurements of CUBIT are performed by inexperienced measurers, measurement cases might have more defects than real-life cases. Additionally, for FP type and DA identification, SRS is required. For these reasons we selected 11 different COSMIC measurements of the CUBIT tool. Background information about case products of Group 2 are explained in detail in section 4.4.3.

Group 3: Measurements of industrial projects

The tool should be tested on measurements concerning industrial MIS projects. We selected 5 COSMIC FSM of the projects from 19 industrial projects as case group 3. Since the rest 14 projects are measured by a group of experts and possible error rates would be less to identify the effectiveness of the tool correctness. Selected measurements belong to KMTS, TEMT, ABYS, BM and MOMENTUS. Additionally, we have the SRSs of these software projects. Background information about these software projects and their measurements are explained in detail in section 4.4.3.

4.4.3. Background about the measurement cases of each group

Group1:

Background information of HRWOMS measurement cases were given under Section 4.3.2.

Group2:

CUBIT is a web based MIS software measurement tool built to facilitate COSMIC and IFPUG measurements. It allows measurers to process measurements as well as to calculate similarities between software projects and to perform benchmarking processes.

Measurement the size of the CUBIT tool was conducted as part of the graduate course offered in Informatics Institute in Middle East Technical University, which is "Software Project Management" the participants of which were students. Prior to the measurement participants were given a 3-hour long COSMIC FSM (ISO/IEC, 2003b) training. Later, they were given a small pilot project and they formed 11 different groups composed of 2 or 3. The level of industrial and academic experience was similar between randomly selected 2 groups.

For the measurement process, SRS of CUBIT was given to each group and they were asked to measure the system by using COSMIC FSM (ISO/IEC, 2003b) method in a week. Measurement documents were collected in spreadsheet format.

Group3:

Kırtasiye Malzemeleri Takip Sistemi (KMTS) - Stationary Requisition System- is an industrial MIS software project developed to manage the requests of stationary material purchase of departments throughout an approval workflow, in electronic environment (Urgun, 2008).

Teminat Mektupları Takip Sistemi (TEMT) - Letters of Credit Follow up System- is an industrial MIS project, which helps to manage transactions of Letters of Credit among the departments. Measurements of these software projects were performed in the context of a technical report by the same inexperienced measurer (Urgun, 2008).

Abone Bilgi Yönetim Sistemi (ABYS)-Subscription Information Management System is an industrial MIS software project developed to provide management interfaces for water subscribers. Bütçe Modülü (BM) - Budget Module is built to facilitate budgeting process automatically. Both of these software projects were measured by the same inexperienced measurer (Ergüden, 2008).

MOMENTUS was developed to manage finance applications of an organization. It was measured by an inexperienced measurer (Şentürk, 2008).

Measurements of industrial projects were collected in spreadsheet format.

4.4.4. Case Study Plan

We will begin Case study 2 with measurement cases of Group1, and continue with Group 2 and Group 3, respectively. We have a total of 2 sub case studies under the Case Study 2. Case Study 2A is for Group1 and Case Study 2B is planned for Groups 2 and 3. In the context of the Case Study 2, several activities are required to be performed. Activities to be performed for Group 1 measurement cases will be explained in Case Study 2A. For Group 1 measurement cases, some of these activities were performed in the context of the Case Study 1. Therefore for Group 1, we will gather these data from Case Study 1. And, thus we have a total of 2 sub case studies under the Case Study 2 for Group1 and for Groups 2 and 3. Activities to be performed for Group 2 and 3 measurement cases will be explained in Case Study 2B.

4.4.4.1. Case Study 2A

Group 1: Measurements of HRWOMS

As depicted in Figure 36, effectiveness of the Tool correctness will be identified on the measurement cases of Group 1 by comparing the results found by the Tool with the results found by the prototype and results of expert review.

For Group 1 a total of 2 activities will be performed. These are;

- The tool Utilization
- Comparing the Results found by the Tool with Results found by the Prototype and Results of Expert Review.

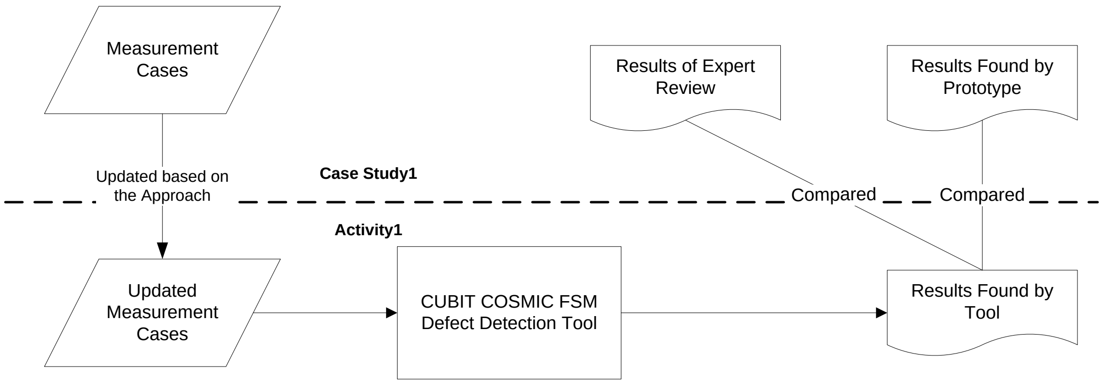


Figure 36: Activities to be conducted on Group 1

Activity 1: Tool Utilization

Results of Expert Review and Results found by the Prototype will be gathered from Case study 1. For finding results of the Tool utilization, measurement cases will be updated based on the Approach which is explained in detail in Chapter 3. First of all for each measurement case,

- FP type of each FP will be identified and they are modified in the CUBIT database.
- DAs will be identified and will be entered into CUBIT database.
- Tool will be facilitated and results found by the Tool will be recorded in a spreadsheet

For the identification of FP types and DAs, SRS of HRWOMS will be used. Effort spent for determining FP Type of each FP and identifying DAs of each OOI for each measurement case will be noted down. Additionally, total effort spent to utilizing the Tool will be calculated.

Activity 2: Comparison of the Results

As depicted in Figure 36; the effectiveness of the Tool correctness will be controlled based on results of the given two comparisons;

- Results found by the Tool – Results of Expert Review
- Results found by the Tool – Results found by the Prototype

While results found by the prototype and results of expert review will be collected from Case Study 1, results found by the Tool will be collected at the end of Activity 1.

To compare the results easily, results found by the Tool and results of Expert Review will be united in one spreadsheet and results found by the Tool and results found by the Prototype will be united in one spreadsheet. For each comparison,

- Same defects that are found by the Tool
- Redundant defects that are found by the Tool
- Missing defects that are missed by the Tool

will be identified. To determine effectiveness of the Tool correctness, results will be analyzed based on each error category. Percentage of same defects, redundant defects and missing defects will be calculated.

4.4.4.2. Case Study 2B

For Groups 2 and 3 as depicted in

Figure 37, a total of 4 activities will be performed. These are;

- The Prototype Utilization,
- The Tool Utilization,
- The Expert Review,
- Comparison of the Results.

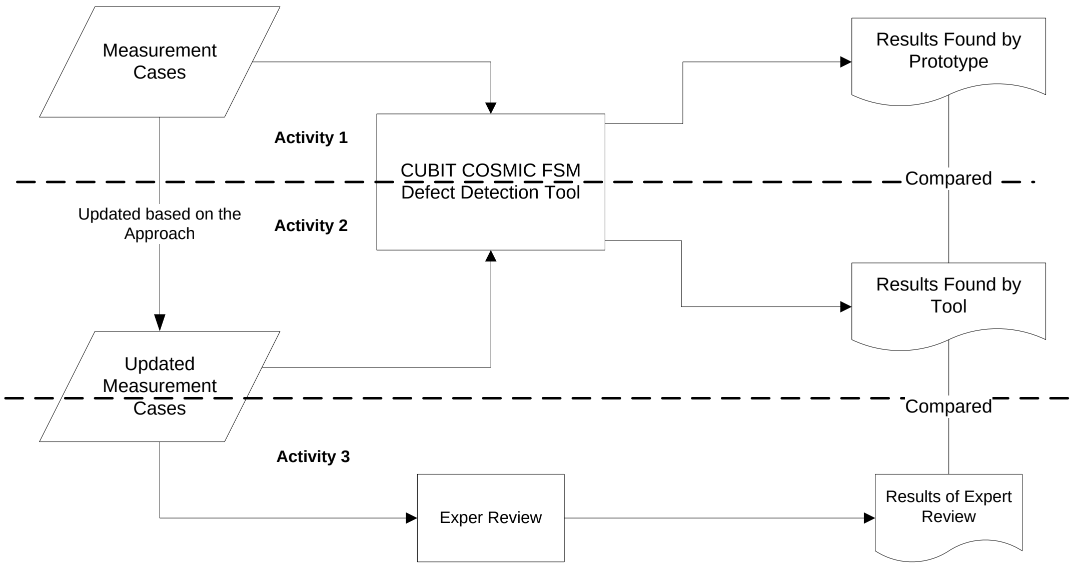


Figure 37: Activities to be conducted on Group 2 & Group 3

Activities to be conducted for each group will be explained separately under each Group's section.

Group 2: Measurements of CUBIT

Activity 1: Prototype Utilization

In order to gather defect detection results by using the prototype, measurement cases are required to be in CUBIT. Since measurement cases of Group 2 were performed by using spreadsheet, we will upload them into CUBIT. In order to spent less effort for uploading each measurement case into CUBIT, spreadsheet import functionality of CUBIT will be used. Later, to prevent any data loss, CUBIT and spreadsheet formats of each measurement case will be controlled with the CUBIT format. At the end, for each measurement case prototype will be facilitated, respectively and results found by prototype will be recorded in a results' spreadsheet based on the each measurement case.

Activity 2: Tool Utilization

To find results by the Tool for Group 2, we will perform the same procedures which are explained for Group 1. To identify the FP types and DAs, SRS of CUBIT will be used. Effort spent for determining FP Type of each FP and identifying DAs of each OOI for each measurement case will be noted down. Additionally, total effort spent to utilizing the Tool will be calculated.

Activity 3: Expert Review

After measurement cases are updated with respect to the Approach, each measurement case will be controlled by the author of this thesis. Prior to the Activity 3, a reference measurement key will be prepared. For the key preparation SRS of CUBIT will be taken as reference for the expert review.

Each measurement case of Group 2 will be controlled based on the references which are given as follows;

- Definition of each error category which are presented in Chapter 3,
- Reference Key of CUBIT,
- SRS knowledge.

Expert review process will be conducted based on the error categories iteratively. Existence of any defects concerning error category EC1 will be controlled through measurement case 1 and measurement case 11. After completing the control for error category EC1, process will continue with the next error categories.

Whenever any defect is detected, it will be noted down based on each measurement case and at the end they will be summarized into a spreadsheet with respect to its error category. Effort spent for detecting defects by utilizing expert review process will be noted down.

Activity 4: Comparison of the Results

To determine the effectiveness of the Tool correctness on the measurement cases of Group 2 we will perform the exactly same procedures that are explained for Activity 2 of Group 1.

Group 3: Measurements of industrial projects

Activity 1: Prototype Utilization

Since measurement cases of Group 3 were measured by using CUBIT, prior to the prototype utilization, each measurement case will be uploaded into CUBIT. However, for each measurement case, measurement in CUBIT and its spreadsheet format will be controlled with each other. Later, for each measurement case prototype will be facilitated and results found by prototype will be recorded in a spreadsheet.

Activity 2: Tool Utilization

To find results by the Tool for Group 3, we will perform the same procedures which are explained for Group 1. To identify the FP types and DAs, SRS of each measurement case will be used.

Activity 3: Expert Review Process

For activity 3, we will perform the same procedures with that of Group 2.

Activity 4: Comparison of the Results

To determine the effectiveness of the Tool correctness on the measurement cases of Group2 we will perform the exactly same procedures that are explained for Activity 2 of Group 1.

4.4.5. Case Study Implementation

4.4.5.1. Case Study 2A

Group 1: Measurements of HRWOMS

Activity 1: Tool Utilization

Results of expert review and results found by prototype issued at the end of Case Study 1, thus we gathered them from Case Study 1. To have the results of the Tool, procedures which are explained in section 4.4.4 were followed. First of all, to facilitate the Tool for each measurement case, FP type of each FP was identified and they were modified in the CUBIT database. Later, DAs were identified and entered into CUBIT. Determination of DAs and entering them into CUBIT were more complex and time consuming than identification of FP Types and their upload process into CUBIT. Later, the tool was facilitated for each measurement case, respectively and results were recorded in a spreadsheet which is referred as “Results found by the Tool for Group 1”. Effort spent for determining FP types and DAs and their update process in CUBIT is presented in Section 4.4.6. Additionally, total effort calculated for the tool utilization is given in Section 4.4.6.

Activity 2: Comparison of the Results

As explained in Case Study Plan, we performed two comparisons to identify the effectiveness of the Tool correctness;

- Results found by the Tool – Results of Expert Review
- Results found by the Tool – Results found by the Prototype

For each comparison, same defects and redundant defects that are detected by the tool and defects that were missed by the tool were identified. Results were collected in a spreadsheet and analyzed. Summary of the results are presented, and findings are discussed in Case Study 2A Results.

4.4.5.2. Case Study 2B

Group 2: Measurements of CUBIT

Activity 1: Prototype Utilization

Since measurement cases of Group 2 were measured by utilizing spreadsheet, we uploaded them into CUBIT by using import functionality of CUBIT Tool. Later, for each measurement case, we controlled the measurement in CUBIT with its spreadsheet format. Later, for each measurement case, the prototype was facilitated and results found by prototype were recorded in a spreadsheet which is referred "Results found by the Prototype for Group 2".

Activity 2: Tool Utilization

To have the results of the tool, procedures which are explained in section 4.4.4 were followed. First of all, to facilitate the tool for each measurement case, FP type of each FP was identified and they were modified in the CUBIT database. Later, DAs were identified and entered into CUBIT. Tool was facilitated for each measurement case, respectively and results were recorded in a spreadsheet which is referred as "Results found by the Tool for Group 2". Effort spent for determining FP types and DAs and their update processes in CUBIT is presented in Results section. Additionally, total effort calculated for the tool utilization is given in Section 4.4.6.

Activity 3: Expert Review

Prior to the expert review a reference key of CUBIT application was prepared and was named as Reference Key of CUBIT (R-Key of CUBIT) in this study.

Based on the each error category, each measurement case of Group 2 was controlled based on the references;

- Definition of each error category which are presented in Chapter 3,
- R-Key of CUBIT,
- SRS knowledge.

Flow of expert review process was conducted based on the error categories and measurement cases iteratively. It started with the error category EC1. Each measurement case was firstly controlled based on the error category EC1. Whenever any defect is detected, it is noted down based on each measurement case. After completion of searching defects concerning error category EC1 through all measurement cases, process proceeded with the next error category, and continued until the control of last error category was completed.

In order to make evaluation process easier, noted defects were summarized into a spreadsheet which is referred “Results of the Expert Review for Group 2”. Effort spent for detecting defects by utilizing expert review process is presented in Case Study 2B Results for Group 2.

Activity 4: Comparison of the Results

After gathering results at the end of activities 1, 2 and 3, same procedures that were explained in Activity 2 in Section 4.4.4 were followed. Summary of the results are presented, and findings are discussed in Case Study 2B Results for Group 2.

Group 3: Measurements of industrial projects

Activity 1:

Since measurement cases of Group 3 were measured by using spreadsheet, prior to the prototype utilization, each measurement case was uploaded into CUBIT. To prevent any data loss, CUBIT and spreadsheet formats were controlled for each measurement case .Later, for each measurement case the prototype was facilitated and the results found by prototype were recorded in a spreadsheet.

Activity 2: Tool Utilization

For activity 2, we performed the same procedures that were followed for Group 2.

Activity 3: Expert Review

For the expert review process of Group 3, we performed same procedures for the expert review of measurement cases concerning Group 2. However, since each measurement case belongs to different systems, we used 5 different SRS and 5 different reference measurements. Thus, preparation for expert review process for Group 3 took longer than Group 2. Prior to the expert review of each measurement case, 5 different reference measurement of each system, TEMT, KMTS, ABYS, MOMENTUS and BM, was created.

Based on the each error category, each measurement case of Group 3 was controlled based on the references;

- Definition of each error category which are presented in Chapter ,
- Reference measurements for related systems,
- SRS of each system.

Flow of expert review process was conducted as same as the process that was realized for Group 2. Found defects were noted and summarized into a spreadsheet which is referred "Results of Expert Review for Group 3". Effort spent for detecting defects by utilizing expert review process is presented in Case Study 2B Results for Group 3.

Activity 4: Comparison of the Results

For activity 4, we performed the same procedures that were processed of Group 2. Results of measurement cases concerning Group 3 are given in section 4.4.6.

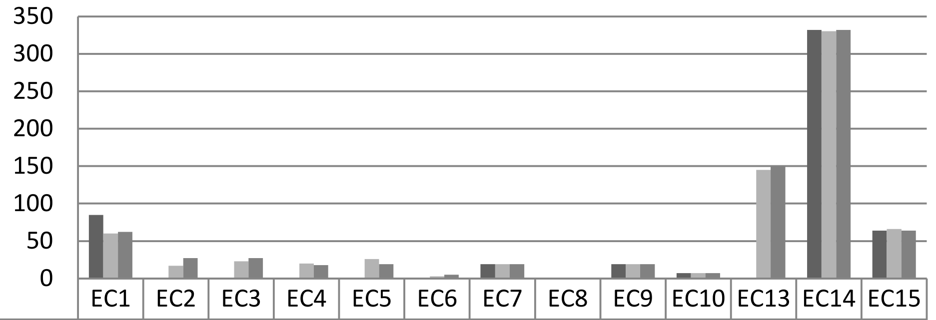
4.4.6. Results

4.4.6.1. Case Study 2A Results

Group 1

Results found in Group 1 cases concerning EC1 to EC15, except from EC11 and EC12, and are summarized in Figure 38. It contains Results of Expert Review, results found by prototype and results found by the Tool. In figure representation these are named as Review, Prototype, and Tool, respectively.

For error categories EC11 and EC12 results found based on Expert Review, results found by the Tool and that of by the Prototype are summarized in Table 9, separately because of the same reason that was explained in Section 4.3.5.



	EC1	EC2	EC3	EC4	EC5	EC6	EC7	EC8	EC9	EC10	EC13	EC14	EC15
Prototype	85	0	0	0	0	0	19	0	19	7	0	332	64
Review	60	17	23	20	26	3	19	0	19	7	145	330	66
Tool	62	27	27	18	19	5	19	0	19	7	150	332	64

Figure 38: Total number of defects belongs to Group 1 based on the Expert Review, the Prototype and the Tool

Table 9: For EC11 and EC12, results of Group 1 based on controls of Review, Prototype and Tool

EC	Review	Prototype	Tool
EC11	7	0	2
EC12	3	0	7
Total	**10**	**0**	**9**

Findings of the comparison concerning “Results of Expert Review – Results found by the Tool” are presented in Table 10.

Table 10: Same & redundant & missing defects of the Tool wrt Expert Review

EC	Number of Defects found by Review	Number of Defects found by Tool	Number of Same Defects	Number of Redundant Defects	Number of Missing Defects
EC1	60	62	53	9	7
EC2	17	27	13	14	4
EC3	23	27	22	2	4
EC4	20	18	14	3	7
EC5	26	19	19	0	7
EC6	3	5	3	2	0
EC7	19	19	16	3	3
EC8	0	0	0	0	0
EC9	19	19	19	0	0
EC10	7	7	7	0	0
EC11	7	2	1	1	6
EC12	3	7	2	5	1
EC13	145	150	145	5	0
EC14	330	332	322	10	8
EC15	66	64	64	0	2
Total	**745**	**758**	**700**	**54**	**49**

Based on the Group 1, effectiveness of the Tool correctness with respect to results of the Expert Review is presented in Table 11. It has been seen from Table 8 that the Tool missed to detect several defects and detected some redundant defects .Percentage of missing defects and redundant defects are presented in Table 11.

Table 11: Effectiveness of the Tool wrt the Expert Review

EC	Effectiveness	Percentage of Missing Defects	Percentage of Redundant Defects
EC1	88%	12%	15%
EC2	76%	24%	52%
EC3	96%	17%	8%
EC4	70%	35%	18%
EC5	73%	27%	5%
EC6	100%	0%	40%
EC7	84%	16%	16%
EC8	N/A	N/A	N/A
EC9	100%	0%	0%
EC10	100%	0%	0%
EC11	14%	86%	50%
EC12	67%	33%	43%
EC13	100%	0%	3%
EC14	98%	2%	3%
EC15	97%	3%	0%

For EC1, the Tool detects same defects with 88% effectiveness. It missed to detect defects with 12% and detects 15% more defects. Effectiveness of the Tool correctness is the same as the effectiveness of the prototype correctness, which is found at the end of Cases study 1. However, in Case study 1, while, percentage of redundant defects found by the prototype 38%, here; percentage of redundant defects found by the Tool is 15%. The main reason of this difference is related with improvement of prototype based on the Approach. By adding FP type component for each FP, measurement cases were modified. While the Prototype does not check the FP types, the Tool controls the FP types during detecting defects concerning error category EC1.

For instance; two different FPs that has same (DG, DM) tuple are given as follows;

Update Personnel	Person Info	E
	Person Info	R
	Person Info	X
Add Personnel	Person Info	E
	Person Info	R
	Person Info	X

Although, (DG, DM) tuples of these two PFs are exactly the same, they are different than each other. First FP is an “update” FP, second one is a “create” FP. If FP types of these FPs are entered, the Tool checks FP types and does not count these same (DG, DM) tuples as a defect and does not create a warning message.

Because of this reason, number of redundant defects which were found by the Prototype is more than the number of redundant defects which were found by the Tool.

For EC2, the Tool can detect same defects with 76% effectiveness. It failed to detect 52% defects of expert review results, and it detects 24% more defects.

In order to identify the reasons of redundant defects, we controlled each measurements case manually, and compared the findings of the Tool with results of Expert Review process.

Redundant defects of the Tool are because of the corruption of DGs belongs to “list” and “update” FPs. For instance;

List CV	User info	E
	User info	R
	User info	X
Update CV	User info	E
	Updated user info	W
	User info	X

For each (update, DG, W) tuple of “update” FP, the Tool controls the “list” FPs, and there should be a (list, DG, R) tuple. In this example, for (update, Updated user info, W) tuple, there is a (list, User info, R). Here the problem is the DG. While DG in list FP is “User info”, DG in update FP is “Updated user info”. For both of these FPs, DG should be the same, as “User info”. In expert review, we did not count this corruption as a defect and we said that for FP “Update CV”, there is a list FP, which is called “List CV”. However, the

Tool counted it as a defect. Because of this reason the Tool detects redundant defects with respect to the Expert review results.

Reason of missing to detect some of the defects is that for some of the "update" FPs, DG which is written in persistent storage may be listed in a "list" FP that represents another functionality of the system. For instance;

View Employee Schedule	Triggering Entry	E
	Employee Info	E
	Employee Info	R
	Schedule Info	R
	Schedule Info	X
Update Employee Information	Employee Info	E
	Employee Info	W
	Error Confirmation	X

In this example, FP type of the first FP is the "list", FP type of the second FP is the "update". For (update, Employee Info, W), there is a (list, Employee Info, R). Since the Tool controls the existence of these two tuples, it does not create an error. However, View Employee Schedule FP is not the "list" FP of, Update Employee Information. In expert review we took it as a defect concerning error category "Lack of list FP before update FP.". Because of this reason, the Tool missed to detect %52 defects of expert review.

For EC3, the Tool can detect same defects with 96% effectiveness. It missed to detect defect with 17% and it detects 8% more defects. Reasons of missing and redundant defects are because of the same reasons with the EC2.

For EC4, the Tool can detect same defects with 70% effectiveness. It missed to detect defect with 35% and it detects 18% more defects. Reasons of missing and redundant defects are because of the same reasons with the EC2.

For EC5, the Tool can detect same defects with 73% effectiveness. It missed to detect defect with 27% and it detects 5% more defects.

Some of the FPs cannot be fit into any type of FPs; we cluster these kind of FPs as FP type "other". Some of these "other" FPs requires DM with type "W". We can infer this requirement from SRS. For instance;

Approve Leave of Employee	Employee Info	E
	Leave Info	R
	Leave Info Status	W
	Error Confirmation	X

In this example type of the FP “Approve Leave Info” is the “other”, and it requires a DM (Leave Info Status, W). We gather this requirement from SRS. Since, the Tool gives warning for lack of W DM for the FP types “create”, “delete” and “update”, it does not give a warning for this FP “Approve Leave of Employee”.

For EC6, the Tool can detect same defects with 100% effectiveness. It detects 40% more defects.

In some of list FPs, W DM might be included. According to this, redundant defects found by the Tool are not defects with respect to the SRS of HRWOMS. Therefore they were not counted in as defects in expert review process.

For EC7, the Tool can detect same defects with 84% effectiveness. It missed to detect defect with 16% and it detects 16% more defects.

For EC8 no defects were detected in any of the measurement case based on expert review, the Tool and the Prototype. Thus, effectiveness of the Tool correctness cannot be identified for error category EC8.

Defects of error category EC9 were detected in 3 different measurement documents and the Tool detects same defects with %100 effectiveness. As far as known from Table 2, error category EC9 is created based on a fundamental rule of COSMIC FSM Measurement Method (ISO/IEC, 2003b), and thus detecting the defects concerning EC9 is very objective and do not require any SRS knowledge.

Defects of EC10 were detected in 4 different measurement documents and the Tool detects same defect with %100 effectiveness. Prototype works on EC10 efficiently because of the same reason with EC9.

For EC11, the Tool can detect same defects with 14% effectiveness. It missed to detect defects with 50% and detects 86 % more defects.

Defect Detection Algorithm of EC11 detects defects based on statistical calculations which are performed on raw data gathered from CUBIT database by clustering projects according to their application type. Adding new COSMIC measurements into CUBIT, may affect the results concerning error category EC11. The Tool cannot identify defects exactly; it only notifies the user for possibility of having defects concerning EC11. During expert review process, to detect defects concerning EC11 SRS knowledge was used and defects were identified exactly. Because of this reason the Tool missed to detect some of the defects that were identified in the Expert Review.

For EC12, the Tool can detect same defects with 67% effectiveness. It missed to detect defect with 33% and it detects 43% more defects. Missing and Redundant defects concerning EC12 are because of the same reason with EC11.

For EC13, the Tool can detect same defects with 100% effectiveness. It detects 3% more defects.

Controlling defects concerning EC13 manually requires high concentration. To identify the reason of redundant defects, each measurement case was controlled manually. And it has been seen that some of the defects are skipped during expert review process.

For EC14, the Tool can detect same defects with 98% effectiveness. It missed to detect defect with 2% and it detects 3% more defects.

The Tool seemed to detect redundant defects concerning EC14. However, since algorithm of EC14 checks syntactical correctness of the DG and OOI naming, and controlling each OOI and DG by manually may result in inaccurate results. As presented in Table 4, defects concerning these error categories are repetitive and were done frequently, during manual inspection some of the defects were missed and some were identified as defect inaccurately.

For the error category EC15, prototype detects same defects with %97 effectiveness. It missed to detect defects with %3. To identify the reasons of missing defects, we examined each measurement case and we recognized that some of OOI data of measurement case 10 were skipped while uploading measurement cases into CUBIT.

Results of “Results found by the Prototype – Results found by the Tool” control are presented in Table 12. Based on the Group 1, effectiveness of the Tool correctness with respect to the Prototype is presented in Table 13.

Table 12: Total numbers of same & redundant & missing defects of the Tool wrt the Prototype

EC	Number of Defects found by Prototype	Number of Defects found by Tool	Number of Same Defects	Number of Redundant Defects	Number of Missing Defects
EC1	84	60	62	0	22
EC2	0	18	0	18	0
EC3	0	25	0	25	0
EC4	0	17	0	17	0
EC5	0	20	0	20	0
EC6	0	5	0	5	0
EC7	19	19	19	0	0
EC8	0	0	0	0	0
EC9	18	18	18	0	0
EC10	7	7	7	0	0
EC11	0	2	0	2	0
EC12	0	7	0	7	0
EC13	0	150	0	150	0
EC14	331	331	331	0	0
EC15	64	64	64	0	0
Total	**523**	**743**	**501**	**244**	**22**

Table 13: Effectiveness of the Tool wrt the Prototype

EC	Effectiveness	Percentage of Missing Defect	Percentage of Redundant Defect
EC1	74%	26%	0%
EC2	NA	NA	100%
EC3	NA	NA	100%
EC4	NA	NA	100%
EC5	NA	NA	100%
EC6	NA	NA	100%
EC7	100%	0%	0%
EC8	N/A	N/A	N/A
EC9	100%	0%	0%
EC10	100%	0%	0%
EC11	NA	NA	100%
EC12	NA	NA	100%
EC13	NA	NA	100%
EC14	100%	0%	0%
EC15	100%	0%	0%

For error category EC1, the Tool detected 74% same defects with the defects which were found by the Prototype. . However it missed to detect 26% defects. To identify the reason of these missing defects, results belongs to the Prototype and the Tool were analyzed for each measurement case, respectively. The main reason is that the Prototype does not support the COSMIC FSM DDA for MIS applications. For error category EC1, the Prototype gives warning whenever two Fps have exactly (DG, DM) tuples, it does not control FP types.

The Prototype found faulty defects by detecting two different FPs with exactly same (DG, DM) tuples as a defect. An example was given to clarify the same situation in Activity 2 results of Case Study 1.

For error categories EC2, EC3, EC4, EC5, EC6, EC11, EC12 and EC13 the Tool detects 100% more defects. Defects concerning these error categories can only be detected with FP type and DA knowledge. Without FP type component, error categories EC2, EC3, EC4, EC5, EC6, EC11 and EC12, without DA component error category EC13 cannot be identified.

For error category EC8 the Tool no defect was found by the Prototype and the Tool. And thus effectiveness of the tool correctness cannot be identified for error category EC8.

For error categories EC7, EC9, EC10, EC14 and EC15 the Tool can detect same defects with 100% effectiveness. Since detection of these error categories does not require the Approach components which are FP type and DA, we did not make any improvement in the Prototype concerning these error categories. Therefore, defects which were detected by the Prototype and that of by the Tool are exactly the same.

Total effort spent for the Tool Utilization is presented in the Table 14

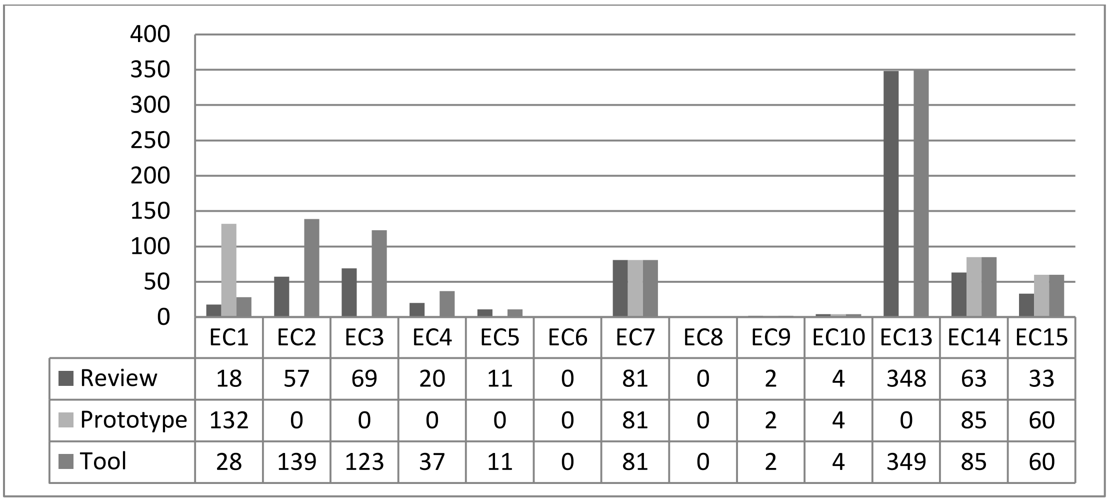
Table 14: Effort spent for the Tool utilization

ITEM	Effort (mins)	Effort (hours)
FP Type identification and CUBIT update	180.00	3.00
DA identification and CUBIT update	315.00	5.25
The Tool utilization (for 10 measurement case)	30.00	0.50
The Tool (Total)	**525.00**	**8.75**

4.4.6.2. Case Study 2B

Group 2:

For measurement cases of Group 2, total number of defects found in Expert Review, by the Prototype and by the Tool is presented in Figure 39. In figure representation these are named as Expert Review, Prototype and Tool, respectively. Error categories EC11 and EC12 are excluded from Figure 39 because of the same reason which is explained in Section 4.3.5.



	EC1	EC2	EC3	EC4	EC5	EC6	EC7	EC8	EC9	EC10	EC13	EC14	EC15
Review	18	57	69	20	11	0	81	0	2	4	348	63	33
Prototype	132	0	0	0	0	0	81	0	2	4	0	85	60
Tool	28	139	123	37	11	0	81	0	2	4	349	85	60

Figure 39: Total number of defects belongs to Group 2 based on the Expert Review, the Prototype and the Tool

Table 15: Results of Group 2 cases for EC11 and EC12

EC	Review	Prototype	Tool
EC11	5	0	2
EC12	1	0	3
Total	**6**	**0**	**5**

Results of “Results found in the Expert Review – Results found by the Tool” control are presented in Table 16. Based on the Group 2, effectiveness of the Tool correctness with respect to the Expert Review is presented in Table 17.

Table 16: Total numbers of same & redundant & missing defects of the Tool wrt the Expert Review based on Group 2

EC	Number of Defects found by Review	Number of Defects found by Tool	Number of Same Defects	Number of Redundant Defects	Number of Missing Defects
EC1	18	28	18	10	0
EC2	57	139	57	82	0
EC3	69	123	69	54	0
EC4	20	37	20	17	0
EC5	11	11	11	0	0
EC6	0	0	0	0	0
EC7	81	81	81	0	0
EC8	0	0	0	0	0
EC9	2	2	2	0	0
EC10	4	4	4	0	0
EC11	5	2	2	0	3
EC12	1	3	0	3	1
EC13	348	349	348	1	0
EC14	63	85	63	22	0
EC15	33	60	33	27	0
Total	**712**	**924**	**708**	**216**	**4**

Table 17: Effectiveness of the Tool correctness wrt the Expert Review based on Group 2

EC	Effectiveness	Percentage of Missing Defect	Percentage of Redundant Defect
EC1	100%	0%	36%
EC2	100%	0%	59%
EC3	100%	0%	44%
EC4	100%	0%	46%
EC5	100%	0%	0%
EC6	N/A	N/A	N/A
EC7	100%	0%	0%
EC8	N/A	N/A	N/A
EC9	100%	0%	0%
			continue on the next page

continued from the previous page			
EC	**Effectiveness**	**Percentage of Missing Defect**	**Percentage of Redundant Defect**
EC10	100%	0%	0%
EC11	40%	0%	60%
EC12	0%	100%	100%
EC13	100%	0%	0%
EC14	100%	0%	26%
EC15	100%	0%	45%

For error category EC1, the Tool detects same defects with 100% effectiveness. It detected 36% more defects than detects which were found in the Expert Review. For instance in measurement case 10, two FPs for which the Tool detects duplication are as follows;

Delete User	User ID	E
	Error Confirmation	R
	User Info	X
	Error Confirmation	R
Delete FURs	FUR Deletion	E
	Error Confirmation	R
	FUR Info	X
	Error Confirmation	R

For these two FPs, defects concerning error category EC7 which is multiple occurrences of the same DM, were detected. "Delete User" and "Delete FURs" FPs are different than each other. However, the Tool gave faulty warning concerning error category EC1. When the reason of this faulty warning was explored, it has been seen that existence of EC7 defects corrupted the defect detection algorithm of error category EC1 and the Tool detected faulty defect.

Defect Detection Algorithm is explained in Chapter 3. During controlling each DM of outerFP with the each DM of innerFP, counter was increased wrong because of the multiple occurrences of DM "Error Confirmation".

innerFP = Delete User

outerFP= Delete FURs

(FP type, DG, DM type)

During controlling the DM "Error Confirmation" outerFP against each DM of innerFp;

(Delete, User ID, E)	(delete, FUR Deletion, E)	Counter =0
	(delete, Error Confirmation, R)	Counter =0
	(delete, FUR Info, R)	Counter =0
	(delete, Error Confirmation, R)	Counter =0
(Delete, Error Confirmation, R)	(delete, FUR Deletion, E)	Counter =0
	(delete, Error Confirmation, R)	Counter =1
	(delete, FUR Info, R)	Counter =1
	(delete, Error Confirmation, R)	Counter =2
(Delete, User Info, X)	(delete, FUR Deletion, E)	Counter =2
	(delete, Error Confirmation, R)	Counter =2
	(delete, FUR Info, R)	Counter =2
	(delete, Error Confirmation, R)	Counter =2
(Delete, Error Confirmation, R)	(delete, FUR Deletion, E)	Counter =2
	(delete, Error Confirmation, R)	Counter =3
	(delete, FUR Info, R)	Counter =3
	(delete, Error Confirmation, R)	Counter =4

The tool gives a warning message when counter equals size of innerFp and outerFP. In this situation because of the multiple occurrence of DM "Error Confirmation", counter became 4 which is equals to the size of innerFp and outerFP.

For error category EC2, the Tool can detect same defects with 100% effectiveness and detected 59% more defects than detected in Expert Review.

In order to identify the reasons of redundant defects, we controlled each measurements case manually, and compared the findings of the Tool with results of the Expert Review process.

It has been seen that reason of the redundant defects found in Group 2 is the same with the reason that caused redundant defects in Group 1. Reason is explained in Section 4.4.5.1.

For error category EC3, the Tool can detect same defects with 100% effectiveness and detected 44% more defects than detected in Expert Review. Reason of redundant defects is because of the same reason with the EC2.

For error category EC4, the Tool can detect same defects with 100% effectiveness and detected 46% more defects than detected in Expert Review. Reason of redundant defects is because of the same reason with the EC2.

For error category EC5, the Tool can detect same defects with 100% effectiveness.

No defects found concerning error category EC6 in the Expert Review, by the Prototype and by the Tool. Therefore, for error category EC6, effectiveness of the Tool correctness cannot be identified on Group 2.

For error category EC7, the Tool can detect same defects with 100% effectiveness.

For error category EC8 no defects were detected in any of the measurement case of Group 2 based on expert review, the Tool and the Prototype. Thus, effectiveness of the Tool correctness cannot be identified for error category EC8.

For error category EC9, the Tool can detect same defects with 100% effectiveness. As far as known from Section 4.3.6 , error category EC9 is created based on a fundamental rule of COSMIC FSM Measurement Method (ISO/IEC, 2003b), and thus detecting the defects concerning EC9 is very objective and do not require any SRS knowledge.

The Tool detected same defects with %100 effectiveness for error category EC10. The Tool works on error category EC10 efficiently because of the same reason with EC9.

For EC11, the Tool can detect same defects with 40% effectiveness and detected 60 % more defects.

Defect Detection Algorithm of EC11 detects defects based on statistical calculations which are performed on raw data gathered from CUBIT database by clustering projects according to their application type. Adding new COSMIC measurements into CUBIT, may affect the results concerning error category EC11. The Tool cannot identify defects exactly; it only notifies the user for possibility of having defects concerning EC11. During expert review process, to detect defects concerning EC11 SRS knowledge was used and defects were identified exactly. Because of this reason the Tool missed to detect some of the defects that were identified in the Expert Review.

For EC12, the Tool could not detect the same defects with the defects found in the Expert Review. It missed to detect defect with 100% and it detects 10% more defects. Missing and Redundant defects concerning EC12 are because of the same reason with EC11.

For EC13, the Tool can detect same defects with 100% effectiveness.

For EC14, the Tool can detect same defects with 100% effectiveness and detected 26% more defects.

The Tool seemed to detect redundant defects concerning EC14. However, since algorithm of EC14 checks syntactical correctness of the DG and OOI naming, and controlling each OOI and DG by manually may result in inaccurate results. As presented in Table 16, defects concerning these error categories are repetitive and were done frequently, during manual inspection, controlling defects concerning EC14 manually requires high concentration. Therefore, some of the defects were missed and some were identified as defect inaccurately.

For the error category EC15, prototype detects same defects with 100% effectiveness and detected 45% more defects. Reason of redundant defects concerning error category Ec15 is because of the same reason with the EC14.

Results of "Results found by the Prototype– Results found by the Tool" control are presented in Table 18. Based on the Group 2, effectiveness of the Tool correctness with respect to the Prototype is presented in Table 19.

Table 18: Total numbers of same & redundant & missing defects of the Tool wrt the Prototype based on Group 2

EC	Number of Defects found by Prototype	Number of Defects found by Tool	Number of Same Defects	Number of Redundant Defects	Number of Missing Defects
EC1	132	28	28	0	104
EC2	0	139	0	139	0
EC3	0	123	0	123	0
EC4	0	37	0	37	0
EC5	0	11	0	11	0
EC6	0	0	0	0	0

continue on the next page

continued from the previous page

EC	Number of Defects found by Prototype	Number of Defects found by Tool	Number of Same Defects	Number of Redundant Defects	Number of Missing Defects
EC7	81	81	81	0	0
EC8	0	0	0	0	0
EC9	2	2	2	0	0
EC10	4	4	4	0	0
EC11	0	2	0	2	0
EC12	0	3	0	3	0
EC13	0	349	0	349	0
EC14	85	85	85	0	0
EC15	60	60	60	0	0
Total	**364**	**924**	**260**	**664**	**104**

Table 19: Effectiveness of the Tool correctness wrt the Prototype based on Group 2

EC	Effectiveness	Percentage of Missing Defect	Percentage of Redundant Defect
EC1	21%	79%	0%
EC2	NA	NA	100%
EC3	NA	NA	100%
EC4	NA	NA	100%
EC5	NA	NA	100%
EC6	N/A	N/A	N/A
EC7	100%	0%	0%
EC8	N/A	N/A	N/A
EC9	100%	0%	0%
EC10	100%	0%	0%
EC11	NA	NA	100%
EC12	NA	NA	100%
EC13	NA	NA	100%
EC14	100%	0%	0%
EC15	100%	0%	0%

For error category EC1, the Tool detects 21% same defects with the defects found by the Prototype. It missed to detect 79% defects which were found by the Prototype.

To identify the reason of these missing defects, results belongs to the Prototype and the Tool were analyzed for each measurement case, respectively. The main reason is that the Prototype does not support the COSMIC FSM DDA for MIS applications. For error category EC1, the Prototype gives warning whenever two Fps have exactly (DG, DM) tuples, it does not control FP types.

The Prototype found faulty defects by detecting two different FPs with exactly same (DG, DM) tuples as a defect. An example was given to clarify the same situation in Activity 2 results of Case Study 1.

For error categories EC2, EC3, EC4, EC5, EC11, EC12 and EC13 the Tool detects 100% more defects than the defects found by the Prototype. Defects concerning these error categories can only be detected with FP type and DA knowledge. Without FP type component, error categories EC2, EC3, EC4, EC5, EC11 and EC12, without DA component error category EC13 cannot be identified.

For error categories EC6 and EC8 no defect was found by the Prototype and the Tool. Therefore for error categories EC6 and EC8, effectiveness of the Tool correctness cannot be identified by using measurement cases of Group 2.

For error categories EC7, EC9, EC10, EC14 and EC15 the Tool can detect same defects with 100% effectiveness. Since detection of these error categories does not require the Approach components which are FP type and DA, we did not make any improvement in the Prototype concerning these error categories. Therefore, defects which were detected by the Prototype and that of by the Tool are exactly the same.

For Group 2 measurement cases total effort spent for utilizing the Prototype, the Tool and for the Expert Review process is given in the Table 20.

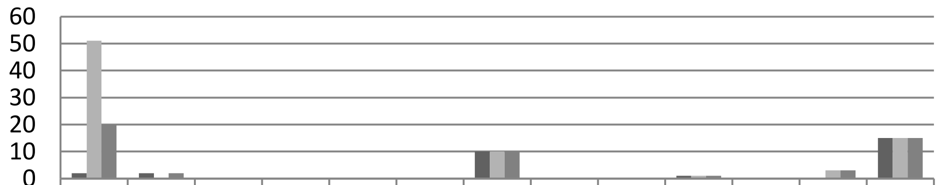
Table 20: Effort spent for the Expert Review & the Prototype & the Tool utilization

ITEM	**Effort (mins)**	**Effort (hours)**
Uploading measurement cases into CUBIT and controlling them.	72.00	1.20
Facilitating prototype	33.00	0.55
continue on the next page		

continued from the previous page		
ITEM	**Effort (mins)**	**Effort (hours)**
The Prototype (Total)	**105.00**	**1.75**
FP Type identification and CUBIT update	330.00	5.50
DA identification and CUBIT update	380.00	6.33
The Tool utilization (for 10 measurement case)	33.00	0.55
The Tool (Total)	**743.00**	**12.38**
Expert Review (Total)	**486.00**	**8.10**

Group 3:

For measurement cases of Group 3, total number of defects found in Expert Review, by the Prototype and by the Tool is presented in Figure 40. In figure representation these are named as Review, Prototype and Tool, respectively. Error categories EC11 and EC12 are excluded from Figure 40 because of the same reason which is explained in Section 4.3.5.



	EC1	EC2	EC3	EC4	EC5	EC6	EC7	EC8	EC9	EC10	EC13	EC14	EC15
Review	2	2	0	0	0	0	10	0	0	1	0	0	15
Prototype	51	0	0	0	0	0	10	0	0	1	0	3	15
Tool	20	2	0	0	0	0	10	0	0	1	0	3	15

Figure 40: Total number of defects belongs to Group 3 based on the Expert Review, the Prototype and the Tool

Table 21: Results of Group 3 cases for EC11 and EC12

EC	Review	Prototype	Tool
EC11	0	0	2
EC12	0	0	1
Total	**0**	**0**	**3**

Results of “Results found by the Expert Review– Results found by the Tool” control are presented in Table 22. Based on the Group 3, effectiveness of the Tool correctness with respect to the Expert Review is presented in Table 23.

Table 22: Total numbers of same & redundant & missing defects of the Tool wrt the Expert Review based on Group 3

EC	Number of Defects found by Review	Number of Defects found by Tool	Number of Same Defects	Number of Redundant Defects	Number of Missing Defects
EC1	2	20	2	18	0
EC2	2	2	2	0	0
EC3	0	0	0	0	0
EC4	0	0	0	0	0
EC5	0	0	0	0	0
EC6	0	0	0	0	0
EC7	10	10	10	0	0
EC8	0	0	0	0	0
EC9	0	0	0	0	0
EC10	1	1	1	0	0
EC11	0	2	0	2	0
EC12	0	1	0	1	0
EC13	0	0	0	0	0
EC14	0	3	0	3	0
EC15	15	15	15	0	0
Total	**30**	**54**	**30**	**24**	**0**

Table 23: Effectiveness of the Tool correctness wrt the Expert Review based on Group 3

EC	Effectiveness	Percentage of Missing Defect	Percentage of Redundant Defect
EC1	100%	0%	90%
EC2	100%	0%	0%
EC3	N/A	N/A	N/A
EC4	N/A	N/A	N/A
EC5	N/A	N/A	N/A
EC6	N/A	N/A	N/A
EC7	100%	0%	0%
EC8	N/A	N/A	N/A
EC9	N/A	N/A	N/A
EC10	100%	0%	0%
EC11	NA	NA	100%
EC12	NA	NA	100%
EC13	N/A	N/A	N/A
EC14	0%	0%	100%
EC15	100%	0%	0%

For error category EC1, the Tool detects 100% same defects with the defects found in Expert Review and detected 90% more defects than defects found in the Expert Review.

To identify the reason of these redundant defects, results belongs to the Expert Review and the Tool were analyzed for each measurement case, respectively. For error category EC1, the Tool gives warning whenever two FPs which have exactly (DG, DM) tuples and FP types.

FP types for some FPs cannot be determined. Therefore, they are categorized as "other". When two FPs with type ""other, have exactly the same (DG, DM) tuple, the Tool detects them as a defect concerning FP Duplication. For instance;

Import Subscribers' Bank Payment File	Payment Info	E
	Payment Info	W
	Error/Confirmation	X
Cancel Payment	Payment Info	E
	Payment Info	W
	Error/Confirmation	X

FP types of these two FPs are “other” and have exactly same (DG, DM) tuple, and thus the Tool gave a warning. However, these two FPs represent different functionalities of the system. To sum up, the Tool gave 90% redundant warning concerning error category EC1.

The Tool detected 100% same defects with defects found in the Expert Review process.

For error categories EC3, EC4, EC5, EC6, EC8, EC9 EC11, EC12 and EC13 no defect was found by the Tool and in the Expert Therefore for error categories EC6 and EC8, effectiveness of the Tool correctness cannot be identified by using measurement cases of Group 3.

For error categories EC7, EC10, and EC15 the Tool can detect same defects with 100% effectiveness.

For error category EC14 the Tool did not find the same defects. Defects found by the Tool were 100% different than the defects found in the Expert review.

Defect detection algorithm of the error category EC14, controls OOI and DG names based on the same specified user interface components which were given in Chapter 3 while defining error category EC14. In measurement case “Momentus”;

Retrieve MenuBar	user info	E
	menuInfo	R
	MenuBar	X

Measurer name DG as menuBar, word menu is semantically different than the user interface component “menu”. Therefore, during the Expert Review processes, usage of the word “menu” did not count as a defect. However, the Tool counted it as a defect. Because of that reason for error category EC14, effectiveness of the Tool correctness based the Group 3 is 0%.

Results of “Results found by the Prototype– Results found by the Tool” control are presented in Table 24. Based on the Group 2, effectiveness of the Tool correctness with respect to the Prototype is presented in Table 25.

Table 24: Total numbers of same & redundant & missing defects of the Tool wrt the Prototype based on Group 3

EC	Number of Defects found by Prototype	Number of Defects found by Tool	Number of Same Defects	Number of Redundant Defects	Number of Missing Defects
EC1	51	20	20	0	31
EC2	0	2	0	2	0
EC3	0	0	0	0	0
EC4	0	0	0	0	0
EC5	0	0	0	0	0
EC6	0	0	0	0	0
EC7	10	10	10	0	0
EC8	0	0	0	0	0
EC9	0	0	0	0	0
EC10	1	1	1	0	0
EC11	0	2	0	2	0
EC12	0	1	0	1	0
EC13	0	0	0	0	0
EC14	3	3	3	0	0
EC15	0	15	15	0	0
Total	**65**	**54**	**49**	**5**	**31**

Table 25: Effectiveness of the Tool correctness wrt the Prototype based on Group 3

EC	Effectiveness	Percentage of Missing Defect	Percentage of Redundant Defect
EC1	39%	61%	0%
EC2	NA	NA	100%
EC3	N/A	N/A	N/A
EC4	N/A	N/A	N/A
EC5	N/A	N/A	N/A
EC6	N/A	N/A	N/A
EC7	100%	0%	0%
EC8	N/A	N/A	N/A
EC9	N/A	N/A	N/A
			continue on the next page

continued from the previous page			
EC	**Effectiveness**	**Percentage of Missing Defect**	**Percentage of Redundant Defect**
EC10	100%	0%	0%
EC11	NA	NA	100%
EC12	NA	NA	100%
EC13	N/A	N/A	N/A
EC14	100%	0%	0%
EC15	100%	0%	0%

For error category EC1, the Tool detects 39% same defects with the defects found in Expert Review and missed to detect 90% defects.

The main reason is that the Prototype does not support the COSMIC FSM DDA for MIS applications. For error category EC1, the Prototype gives warning whenever two Fps have exactly (DG, DM) tuples, it does not control FP types.

The Prototype found faulty defects by giving warning for two different FPs which have exactly same (DG, DM) tuples as a defect. An example was given to clarify the same situation in Activity 2 results of Case Study 1.

During the controls of Group 3 cases product, we did not encounter defects concerning error categories EC3, EC4, EC5, EC6, EC8, EC9, and EC13. Therefore, for these error categories results of Group 3 did not give any additional supportive data to prove the effectiveness of COSMIC FSM DDA for MIS software projects.

For error categories EC7, EC10, EC14 and EC15 the Tool can detect same defects with 100% effectiveness.

Group 3 case products were measured more precisely when compared to Group 1 and Group 2 case products. Thus, results of Group 1 and Group 2 case products had wider variety of defects.

For further observation, we need to find out whether the COSMIC FSM DDA is efficiently applicable to measurements of software project with different application domains.

4.5. Case Study 3

COSMIC FSM DDA and CUBIT COSMIC FSM DDT were developed based on MIS software projects. We need to find out whether approach works efficiently on different application domains or not.

4.5.1. Case Selection Criteria

In the context of case study 3, measurements of software projects with application types RT and Embedded are selected.

Most of the software projects in CUBIT database have MIS application type. Therefore, we did not have many alternative measurements of software projects that have different application type rather than MIS. COSMIC measurements of 24 RT and 10 embedded software projects are selected for cases study 3.

We have chosen these projects for several reasons as;

First criterion for selecting those projects is that these projects were measured by an inexperienced measurer.

They all belong to the same organization. Since they are confidential and their measurements are stated by encrypting the functionalities of projects, gathering information about the projects should be easier. As it is easy to reach the owner of the software products, we have chosen these case products.

4.5.2. Case Study Plan

We plan to have a similar case study design with case study 2. Firstly, case products will be controlled in their default format. Secondly, they will be updated according to COSMIC FSM DDA and will be controlled again. At the end of each control, results will be collected and analyzed with each other. However, this time application domains of software projects are different than MIS. Prior to second control, we estimate that we will have difficulty in identifying FP types of each FP in each case product. In addition to this, since these projects are confidential, we do not have their SRS. Thus, this will prevent us to investigate DAs.

Since we do not have SRSs of case products, for the FP type determination we plan to talk with the owner of the projects.

We will have some difficulties in applying the case study design of 2nd case study, since starting from development of CUBIT COSMIC FSM DDT, until the case study 3; we have worked on MIS software projects. These difficulties and possible alternatives will be explained in case study implementation.

4.5.3. Case Study Implementation

Selected case products, first of all controlled with their default format results are recorded for comparing them with the results of second control.

Prior to the second control, for each measurement case FP types for each FP should be identified and related measurement case should be modified in the CUBIT database. During the FP type determination, FP types that are determined based on MIS application cannot be mapped to FPs belong to RT and Embedded software projects' measurements. For FP type determination process, we interviewed with owner of the cases. At the end of these interviews, we concluded that FP types of MIs software projects' measurements cannot be mapped to FPs of RT and Embedded software projects' measurements. New FP types should be determined for RT and Embedded software projects' measurements. As we could not reach SRSs, DAs are set as null. Therefore, second control cannot be performed because of lack of DA knowledge and FP type violation.

4.5.4. Results

In 11 measurements of RT software projects and 5 measurement documents of embedded software projects, the Tool does not detect any defect. Number of defects found with respect to error categories is summarized in Chart 7.

For error categories EC1, EC7, EC10, EC14 and EC15, defects are found by the Tool. Error categories EC2, EC3, EC4, EC5, EC6, EC11 and EC12 that require FP type's knowledge cannot be detected.

In case study 2, we have tested the Tool and the Approach on a total of 3 different groups of measurements. Within each group, results for the error categories EC7, EC8, EC9, EC10, EC14 and EC15 were consistent. Results of case study 3 for error categories EC7, EC10, EC14 and EC15 are 40, 3, 2 and 1, respectively. Combining all of these findings, error categories EC7, EC10, EC14 and EC15 can be detected in COSMIC measurements of software projects that have different type of application domain than MIS.

Since FP types cannot be mapped into FPs of measurements concerning RT and Embedded applications, from point of view case study 2 results, 20 found defects for error category EC1 may include redundant warnings. These redundant warnings may mislead users.

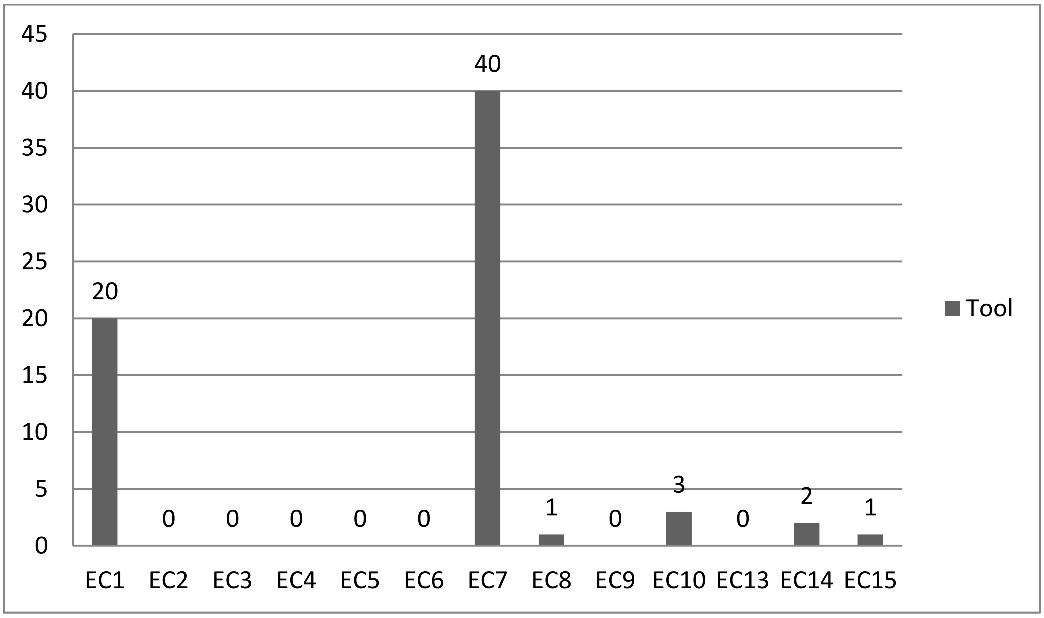


Figure 41: Results for measurements of RT /Embedded software projects

According to our findings error categories should be classified into two as, MIS specific and General. Findings are summarized in Table 26. Tool can detect defects concerning error categories EC7, EC8, EC9, EC10, EC13, EC14 and EC15 without application domain constraint. However, the Approach can detect defects concerning EC1, EC2, EC3, EC4, EC5, EC6, EC11 and EC12. Detection of these error categories depends on FP types. Therefore if FP types are identified for RT and embedded software projects, these error categories can be detected by the Approach.

Table 26: Error categories wrt availability of defect detection concerning application domains

General	MIS Specific
EC7	EC1
EC8	EC2
EC9	EC3
EC10	EC4
EC13	EC5
EC14	EC6
EC15	EC11
	EC12

4.6. Validity Threats

Expert review processes of Case Study 1, 2 and 3 were conducted by the author of this thesis; it may include some defects itself. To minimize the errors, an expert who has COSMIC FSM certification and at least 3 years measurement experience helped to the author for critical points.

In Case Study 2, for the prototype and the tool control, while uploading measurement cases, data loss may occur. To prevent data loss, measurement cases in CUBIT were controlled with their spreadsheet formats.

Defects of FSMs can be detected effectively by using COSMIC FSM DDA for MIS applications if and only if FP type information of FPs are supplied. Identified error categories and CUBIT COSMIC FSM DDT for MIS was developed based on MIS projects. Facilitating the tool on broader measurement cases may not create similar results. As depicted in results of case study 3, the approach is not valid for the measurements of real-time and embedded software projects.

CHAPTER 5

CONCLUSION AND FUTURE WORK

This Chapter introduces the conclusions and discussion in the first section, future work in the second section.

5.1. Conclusions

In this study we aimed to develop a COSMIC FSM DDA for MIS Applications and to use it for detecting COSMIC FSM measurement defects automatically. For this reason, three case studies were conducted within the context of this thesis. The first case study was an exploration case study. Most commonly made COSMIC errors found in the literature were classified into error categories and clearly defined by using COSMIC measurements of HRWOMS. Additionally, error categories of which defects can be detected by automatically were determined. Based on these error categories, the prototype of the CUBIT COSMIC FSM DDT for MIS applications was developed. Effectiveness of the prototype correctness was observed on the same set of measurement cases with the cases used in expert review process.

Results of the Case Study 1, which were given in Section 4.3.5, showed that the prototype was not adequate to detect the defects of COSMIC measurements of MIS Applications. For this reason a COSMIC FSM DDA is developed and based on the approach the prototype is improved. To detect the effectiveness of the improved CUBIT COSMIC FSM DDT for MIS Applications, second case study is performed on three different sets of measurement cases.

Third case study is performed for observing the behavior of the tool by utilizing it on COSMIC measurements belongs to different application domains. For this reason the Tool was facilitated on COSMIC measurement cases belong to RT and Embedded software project.

Table 27: Performance of the Tool wrt results of the Expert Review and the Prototype based on three different groups of measurement cases.

Performance of the Tool based on		Percentage of same Defects	Percentage of Redundant Defects	Percentage of Missing Defects
Group 1	Expert Review	94%	7%	7%
	Prototype	96%	33%	4%
Group 2	Expert Review	99%	23%	1%
	Prototype	71%	72%	29%
Group 3	Expert Review	100%	44%	0%
	Prototype	75%	9%	48%

In the context of thesis the tool correctness was measured based on the three different groups of measurement cases, with respect to the results of the prototype and the expert review. Table 27 presents the performance of the tool with respect to the expert review and the prototype.

When the results of the three case studies are combined we can conclude as;

For error categories EC2, EC3, EC4, EC5 and EC6, the prototype did not detect any defects because of the lack of FP type knowledge during the prototype control. On the other hand, the tool detected defects concerning these error categories. Error categories EC2, EC3, EC4, EC5 and EC6 cannot be detected, unless measurement cases are compatible with COSMIC FSM DDA.

For the error category EC1, the prototype gave faulty redundant warnings. On the other hand, component FP type allowed the tool to detect defects of error category EC1 more efficiently.

For error categories EC7, EC9, EC10, EC14 and EC15 the tool can detect 100% same defects with the defects found in the Expert Review and defects found by the prototype. Detected the defects of these error categories is independent from the approach and the application domain of the software projects.

Defects concerning error category EC8, were not encountered in measurement cases of each group. Therefore, for error category EC8, the effectiveness of the tool correctness could not be identified. However, since error category EC8 is from COSMIC FSM measurement method such as error categories EC9 and EC10, its defects are thought to be detected independently from the approach.

The Tool gave warnings about the possibility of having defects based on the error Category EC11 and EC12, since it creates warnings based on statistical calculations. The Tool does not give the exact point of defects concerning error categories EC11 and EC12.Additionally; results showed that, these error categories can only be detected with additional component FP type.

To detect defects concerning error category EC13 component DA attribute should be mandatory for the COSMIC FSM measurements.

Moreover, in the context of the case study 1 and 2, for group 1 measurement cases, while detecting defects with expert review mechanism requires more effort, the tool completes detection process more quickly.

In the context of the case study 2, prior to the utilization of the tool by using group 2 measurement cases, FP type and DA identification was time consuming. Only SRS knowledge was not adequate, for each measurement case, we needed to investigate the FPs, OOIs and DGs. Thus, for the tool utilization, more effort was spent when compared to effort spent for the expert review.

FP type and DA components are thought to be identified during the measurement process, defect detection process will only be composed of the time elapsed for the utilization of the tool.

As a result of this thesis, defects of number error categories can be detected automatically without SRS knowledge with less effort. These error categories are; EC1, EC2, EC3, EC4, EC5, EC6, EC7, EC9, EC10, EC13, EC14, EC15. Since EC11 and EC12 are general warnings and do not give the exact point of defect, these error categories constitute a basis for the possible errors to be done. Error categories EC1, EC2, EC3, EC4, EC5, EC6, EC11 and EC12 can only be detected if and only if component, FP type of the COSMIC FSM DDA is included in COSMIC measurements.

Defects of error category EC13 can be detected if and only if DA component of the approach is supplied. However, detecting defects concerning error category EC13 is independent from software domain type. It can be detected in not only measurements of MIS applications but also measurements of RT and Embedded software projects.

5.2. Future Work

Case Study 1 and 2 were performed by using COSMIC measurements of MIS Applications. In Case Study 3, COSMIC measurements of RT and Embedded software projects were used. Results of the Case Study 1 and 2 showed that defects of error categories EC1, EC2, EC3, EC4, EC5, EC6, EC11 and EC12 can be detected if and only if the FP type component of the COSMIC FSM DDA is supplied. Additionally, Case Study 3 showed that FP types, which are Cerate, Delete, Update, List and Retrieve, are only valid for COSMIC measurements of MIS software projects not for RT and Embedded software projects. In order to detect defects of COSMIC measurements concerning software projects which have different type of application domains, different FP types are required. To make the approach work among broader measurements belong to different application domain types, further researches are required.

In order to see the behavior of the tool on COSMIC measurements concerning various types of application domains, it is required to be utilized for the COSMIC measurements of different software projects. Additionally, the approach and the tool are developed for detecting defects of COSMIC FSMs. There are various FSM methods such as IFPUG, UniFSM, being used by different groups. COSMIC FSM DDA should be extended for controlling measurements that are measured by using different FSM methods.

Furthermore, to verify the measurement automatically, software artifacts such as whole SRS and E-R diagram of the software project can be supplied as an input into the CUBIT.

.

REFERENCES

Albrecht, A. (1979). Measuring Application Development Productivity. *Proc. of IBM Application Development Symp.* (pp. 83–92).

Albrecht, A. J., & Gaffney, J.E., J. (1983). Software Function, Source Lines of Code, and Development Effort Prediction: A Software Science Validation. *IEEE Transactions on Software Engineering*, *SE-9*(6), 639 – 648. doi:10.1109/TSE.1983.235271

ISO/IEC (1998). 14143-1: Information Technology - Software Measurement - Functional Size Measurement - Part 1: Definition of Concepts.

ISO/IEC (2002a). 14143-2: Information Technology - Software Measurement - Functional Size Measurement - Part 2: Conformity Evaluation of Software Size Measurement Methods to ISO/IEC 14143-1:1998.

ISO/IEC (2003a). 14143-3: Information Technology - Software Measurement - Functional Size Measurement - Part 3: Verification of Functional Size Measurement Methods.

ISO/IEC (2002b). 14143-4: Information Technology - Software Measurement - Functional Size Measurement - Part 4: Reference Model.

ISO/IEC (2004). 14143-5: Information Technology- Software Measurement - Functional Size Measurement - Part 5: Determination of Functional Domains for Use with Functional Size Measurement.

ISO/IEC (2005a). 14143-6: Guide for the Use of ISO/IEC 14143 and related International Standards.

Gencel, C., & Demirors, O. (2008). Functional size measurement revisited. *ACM Trans. Softw. Eng. Methodol.*, *17*(3), 15:1–15:36. doi:10.1145/1363102.1363106

Fenton, N. E., & Pfleeger, S. L. (1998). *Software Metrics: A Rigorous and Practical Approach* (2nd ed.). Boston, MA, USA: PWS Publishing Co.

Gencel, C., Buglione, L., Demirors, O., Efe, P., 2006, *A Case Study on the Evaluation of COSMIC-FFP and Use Case Points.* 3rd Software Measurement European Forum (SMEF 2006), pp. 121-138.

Ungan, E. (2010). Evaluation of Reliability Improvements for COSMIC Size Measurement Results.

The Common Software Measurement International Consortium (COSMIC): 2005 Guideline for Sizing Business Application Software, Version 1.0

Low, G. C., & Jeffery, D. R. (1990). Function points in the estimation and evaluation of the software process. *IEEE Transactions on Software Engineering*, *16*(1), 64 –71. doi:10.1109/32.44364

Symons, C. (2001). *Come Back Function Point Analysis (Modernized) – All is Forgiven!),* in Proc. of the 4th European Conf. on Software Measurement and ICT Control, , Germany, 413-426.

ISO/IEC (2003b). 19761: COSMIC Full Function Points Measurement Manual, v. 2.2.

ISO/IEC (2003c). IS 20926: Software Engineering - IFPUG 4.1 Unadjusted FSM Method - Counting Practices Manual.

ISO/IEC (2002c). 20968:2002: Software Engineering - MkII Function Point Analysis - Counting Practices Manual, (2002).

ISO/IEC (2005b). IS 24570: Software Engineering – NESMA Functional Size Measurement Method v.2.1 – Definitions and counting guidelines for the application of Function Point Analysis.

ISO/IEC (2008). 29881:2008 Information Technology-- Software and systems engineering— FISMA 1.1 functional size measurement method.

The Common Software Measurement International Consortium (COSMIC):2007, Method Overview v3.0.

Khalikov, F., Yilmaz, G., Ozkan, B., Ungan, E., Top, O.O., & Nabii, M., (2011). Software Requirements Specification of CUBIT

Kemerer, C. F. (1993). Reliability of function points measurement: a field experiment. *Commun. ACM*, *36*(2), 85–97. doi:10.1145/151220.151230

Kemerer, C. F., & Porter, B. S. (1992). Improving the reliability of function point measurement: an empirical study. *IEEE Transactions on Software Engineering*, *18*(11), 1011 –1024. doi:10.1109/32.177370

Turetken, O., Ozcan Top, O., Ozkan, B., & Demirors, O. (2008). The Impact of Individual Assumptions on Functional Size Measurement. *Proceedings of the International Conferences on Software Process and Product Measurement*, IWSM/Metrikon/Mensura '08 (pp. 155–169). Berlin, Heidelberg: Springer-Verlag. doi:10.1007/978-3-540-89403-2_14

Top, O. O., Demirors, O., & Ozkan, B. (2009). Reliability of COSMIC Functional Size Measurement Results: A Multiple Case Study on Industry Cases. *Software Engineering and Advanced Applications, Euromicro Conference* (Vol. 0, pp. 327–334). Los Alamitos, CA, USA: IEEE Computer Society.

Abrahao, S., Poels, G., & Pastor, O. (2004). Assessing the reproducibility and accuracy of functional size measurement methods through experimentation. *2004 International Symposium on Empirical Software Engineering, 2004. ISESE '04. Proceedings* (pp. 189 – 198). Presented at the 2004 International Symposium on Empirical Software Engineering, 2004. ISESE '04. Proceedings. doi:10.1109/ISESE.2004.1334906

Ungan, E., Demirörs, O., Top, Ö., & Özkan, B. (2009). An Experimental Study on the Reliability of COSMIC Measurement Results. In A. Abran, R. Braungarten, R. Dumke, J. Cuadrado-Gallego, & J. Brunekreef (Eds.), *Software Process and Product Measurement*, Lecture Notes in Computer Science (Vol. 5891, pp. 321–336). Springer Berlin / Heidelberg. Retrieved from http://www.springerlink.com/content/2045132k85641w04/abstract/

Ungan, E. (2010). Evaluation of Reliability Improvements for COSMIC Size Measurement Results.

The Common Software Measurement International Consortium (COSMIC): 2011 Guideline for Assuring the Accuracy of Measurements, Version 1.0.

Top, O. O. (2008), *FUNCTIONAL SIMILARITY IMPACT ON THE RELATION BETWEEN FUNCTIONAL SIZE AND SOFTWARE DEVELOPMENT EFFORT*. Middle East Technical University, Ankara, Turkey.

Glass, R. L. (2002). *Facts and Fallacies of Software Engineering* (1st ed.). Addison-Wesley Professional.

Yilmaz, G., Ungan, E., Demirors, O. (2010), *The effect of the quality of software requirements document on the functional size measurement.* [Online].Available: http://www.uksma.co.uk/conference2011/presentations/05GokcenYilmazTheEffectoftheQualityofSoftwareonFSM.pdf

Ergüden, E. (2008). *APPLICATION OF UNIFICATION MODEL FOR FUNCTIONAL SIZE MEASUREMENT METHODS: A CASE STUDY İN WEB APPLICATIONS.*

Urgun, E. (2008). *DETAILED EVALUATION COSMIC FFP AND IFPUG FPA CONVERSION APPROACHES.*

Şentürk, O. (2008). *A Case Study on the Unification Model for Functional Size Measurement Methods: Enterprise Solutions Web Application.*