CONSTRUCTING A PRODUCTIVITY MODEL FOR SMALL ENHANCEMENTS

by

Gülşah Yıldızoğlu

B.S., Computer Engineering, Dokuz Eylül University, 2009

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Computer Engineering

Boğaziçi University

2013

ACKNOWLEDGEMENTS

This thesis would not have been possible without the help of several individuals who contributed and extended their valuable assistance in the preparation and completion of this study. I wish to express my deepest gratitude to all people who encouraged and supported me throughout the preparation of this thesis.

First and foremost I offer my sincerest gratitude to my thesis supervisor, Assoc. Prof. Tuna Tuğcu and Ph.D. Jean-Marc Desharnais for their guidance, they have supported me thoughout my thesis with their patience and knowledge. I also wish to express my sincere thanks to Assoc. Prof. Haluk Bingöl and Asst. Prof. Atay Özgövde for kindly accepting to be committee member.

I would also like to thank Ph.D. Alain Abran for all his support and precious feedback. He shared his individual ideas and comments about my master thesis and the article.

I also want to present my sincere gratitude to Ahmet Ferhat Şahin, Serkan Kırbaş, Mehmet Çağatay Çakır and Fatih Kızkun for their patience, invaluable support.

Most importantly, I would like to thank my family. They were always supporting me and encouraging me with their best wishes.

ABSTRACT

CONSTRUCTING A PRODUCTIVITY MODEL FOR SMALL ENHANCEMENTS

Software is in constant evolution and many approaches have been suggested to study software maintenance productivity. This master thesis aims to describe the process to design and implement a productivity measurement model for small functional enhancements to legacy software. Two motivations influence this research: (i) understanding the productivity of the software maintenance process to help manage the cost of maintenance, (ii) understanding the cost drivers that affect the software maintenance productivity. Another purpose of this research is to determine whether it is possible to construct productivity model(s) for small enhancements that is as good as development model. The case study, to show the feasibility of the application of the methodology, is coming from a large technology group that works for a bank organization. In this approach, it is showed that small enhancements can be measured by using the COSMIC measurement method that has the possibility to capture a lower level of granularity. Also, the quality of documentation for each small enhancement was studied. Because the person who has done the maintenance was at hand for this exercise, it was possible to complete the documentation. Therefore the quality of the documentation is high. Totally, 88 small enhancements are investigated and measured separately. The measurement of the functional size of each individual enhancement was based on the COSMIC ISO 19761 international standard. It was possible to produce a sound productivity model with this sample using all independent variables (R square of 0.75). The methodology used is showing that the experimentation was well controlled: within an enterprise for one major application for a period of time, design, program and implement by the same person, documented from the maintainer, measure within a control environment and verify by an expert.

ÖZET

KÜÇÜK FONKSİYONEL GELİŞTİRMELER İÇİN VERİMLİLİK ÖLÇÜM MODELİ TASARLAMAK

Yazılım sürekli evrim içindedir ve birçok yaklaşım yazılım bakım verimliliğini incelemek için öne sürülmüştür. Bu yüksek lisans tezinde küçük fonksiyonel geliştirmeler için bir verimlilik ölçüm modeli tasarlamak ve uygulamak için gerekli sürecin tanımlanması amaçlanmıştır. İkı motivasyon bu araştırmayı etkilemektedir: (i) Bakım maliyetini yönetmeye yardımcı olmak için yazılım bakım sürecinin verimliliğini anlamak, (ii) yazılım bakım verimliliğini etkileyen maliyet sürücülerini anlamak. Bu araştırmanın bir başka amacı, geliştirme modeli olarak küçük geliştirmeler için verimlilik modeli oluşturmanın mümkün olup olmadığının sorusuna yanıt bulmaktır. İlk olay çalışması, bir banka organizasyonunda çalışan büyük bir teknoloji grubundan geliyor. Bu çalışma metodolojinin uygulamaya uygulanabilirliğini gösteriyor. Bu yaklaşımda, düşük seviyedeki tanecikliği yakalayabilen COSMIC ölçüm yöntemi kullanılarak, küçük geliştirmelerin ölçülebildiği gösterilmiştir. Ayrıca, her küçük geliştirme için dökümantasyon kalitesi incelenmiştir. Küçük geliştirmeleri ölçümleyebilmek için yeterli bilgi yer aldığı için dökümantasyon kalitesi yüksektir. Tam olarak 88 küçük geliştirme incelendi ve ayrı ayrı ölçüldü. Herbir küçük geliştirmenin ölçüm boyutu COSMIC ISO 19761 uluslararası standardına dayanmaktadır. Tüm bağımsız değişkenler kullanılarak bu veri kümesi ile bir verimlilik modeli ü retmek mümkün olmuştur (R-square değeri 0,75). Kullanılan metodoloji deneyselliğin iyi kontrol edildiğini gösteriyor: Büyük bir uygulama bir gelişim içinde belli bir zaman diliminde aynı kişi tarafından tasarlanmış, programlanmış ve geliştirilmiş, bakım yapan kişi tarafından dökümante edilmiş, bir denetim ortamında ölçülmüş ve bir uzman tarafından doğrulanmıştır.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

%	Percent
$\sum$	Sum
#	Number

LIST OF ACRONYMS/ABBREVIATIONS

CMMI	Capability Maturity Model Integration
CR	Change Request
CFP	COSMIC Function Point
COCOMO	The Constructive Cost Model
DSI	Delivered Source Instructions
ERP	Enterprise Resource Planning
ESS	Explained Sum of Squares
FSM	Functional Size Measurement
FUR	Functional User Requirement
IEC	International Engineering Consortium
IFPUG	International Function Point Users Group
ISBSG	International Software Benchmarking Standards Group
ISO	International Organization for Standardization
IT	Information Technology
KDSI	Delivered Source Instructions in Thousand
MIS	Management Information Systems
MM	Maintenance Manager
SPSS	Statistical Package for the Social Sciences
TSS	Totam Sum of Squares

1. INTRODUCTION

The software life cycle can be divided into two distinct parts (ISO/IEC 12207:2008) [1] the initial development of software and the maintenance and use of software. Organizations have tendency to overlook the development aspect but neglect the maintenance until problem occurs [2–4]. In this study, software maintenance part is taken into consideration. Small enhancements are analyzed and measured using COSMIC measurement method [5] to answer it is possible to construct productivity model(s)for small enhancements.

In ISO/IEC 12207 maintenance means that all the activities required to support the software and to keep it satisfying the users' needs after the delivery. Some activities could start during the initial development of software but most activities are those following its delivery. In 1981 Boehm [6] defines maintenance as the process of modifying existing operational software while leaving its primary functions intact.

ISO/IEC 14764 [7] defines four categories to classify the nature of maintenance work requests: adaptive, corrective, preventive, perfective and user support.

ISO/IEC 14764 [7] classifies adaptive and perfective maintenance as enhancements, while the corrective and preventive maintenance categories as corrections. In this study especially adaptive and corrective maintenance is handled. It is possible to find change request (CR) in all of these categories.

Software Maintenance is request-driven and is not managed using project management techniques but, rather, queue management techniques [8]. Therefore management request are tackle differently by management and should be studied considering this to assess productivity of the maintenance within a specific organization.

Few researches have been done to study the improvement of software maintenance. A literature review published by Benestad and *et al.* [9] presents an overview of

Table 1.1. Maintenance Categories.

Category	**Description**
Adaptive	Modifications to adapt a software product to changes in data requirements and processing environments
Corrective	Reactive modification of a software product performed after delivery to correct the faults discovered Modification repair code to satisfy functional requirements
Preventive	Modification of a software product after delivery to detect and correct latent faults before they become operational faults
Perfective	Modification of a software product after delivery to detect and correct latent faults in the software product before they are manifested as failures

recent approaches. Authors analyze how to: improve individual changes, design maturity models to help with software maintenance process improvement initiatives [10,11], estimate the cost of maintenance [12], provide insights into the prerequisites needed before trying to measure software maintenance productivity [9,13] and analyze directly productivity models [2–4, 14–16].

The studied organization was targeted to have its processes CMMI [17] Level 3 a few years ago. Several studies were held in order to satisfy CMMI Level 3 requirements. But CMMI initiative could not be finalized because huge amount of effort was required by CMMI studies, especially in the measurement and verification aspect. By knowing that, it was decided to use an iterative incremental approach in this maintenance measurement research. The iterations are planned for short periods and the concrete results are defined and produced for each iteration. In this study 88 small enhancements will be measured and analyzed.

Our motivation is to held empirical studies in software maintenance. There are

not enough empirical data collected and analyzed for the maintenance phase of software [3] especially for small change requests, (an effort of 40 hours or less for 90% of the small enhancements). Therefore empirical studies are to be held in the organizations in order to make them aware that software maintenance is also part of the software life cycle and should be considered together with software development.

There are different types of perspective:

- Data processing group consists of programmers and analysts (because of complex set of qualitative data from multiple resources),
- Maintenance team to handle small enhancements,
- Project manager to estimate development effort of each enhancement by using calculated functional size.

Constructing a productivity model for software development is a well establishes activity within organizations. Many articles and books had been written on the subjects in the last 40 years starting with Barry Boehm COCOMO approach [6, 18–20]. Few maintenance researches had been done to construct a productivity model for small enhancements [13, 21, 22].

1.1. Contribution and Objectives of the Master Thesis

This master thesis aims to describe the process to design and implement a productivity measurement model for small enhancements. The motivations influence this research: (i) understanding the productivity of the software maintenance personnel to help manage the cost of maintenance, (ii) understanding the cost drivers that affect the productivity of the software maintenance personnel to influence the choice of the best available tool to maintain the software.

Another contribution of this research is to determine whether it is possible to construct productivity model(s) for small enhancements that is as good as development model. The case study, to show the feasibility of the application of the methodology,

is coming from a large technology group that works for a bank organization. In this approach, it is showed that small enhancements can be measured by using the COSMIC [5] measurement method that has the possibility to capture a lower level of granularity. Also, the quality of documentation for each small enhancement was studied. Because the person who has done the maintenance was at hand for this exercise, it was possible to complete the documentation. Therefore the quality of the documentation is high. Description of the requirements of each small enhancement is in the Annex B.

The main hypothesis of this study is the following: it is possible to construct a productivity model (or more than one) for small enhancements. This could be done within a banking organization measuring and analyzing 88 small enhancements following a methodology to do it. Because functional size measurement methods have been designed to measure software development or medium-large enhancement projects and they have not been used to measure very small change requests, it is important to choose a measurement method that gives an accurate size of each small change requests. For this reason the measurement method should provide a low level of granularity. COSMIC standard measurement method [5] was chosen for this capability. Small enhancements are mainly part of maintenance work to improve actual applications or systems. The use of the data from small enhancements will be slightly different as much of the productivity model, even if the objective is still provide the organization with elements of effective processes that ultimately improve the performance. As already mention the quality of the documentation used is high. This help to have reliable data for the measure of the size of the product.

1.2. Organization of the Thesis

The study is organized as follows. Chapter 2 introduces the different concepts and methods used in this study background. Chapter 3 explains the methodology and the design used. chapter 4 presents the data collected and results of the statistical analysis applied to data. Chapter 5 interprets the data presented and provides inferences from the analysis. Finally, Chapter 6 concludes by summarizing the study and the results of it and presents the future work.

2. BACKGROUND AND DEFINITIONS

In large organizations, as presented in Table 2.1 [15], most of the efforts are in software maintenance but at the same time it is still a rather neglected activity according to a number of authors (Brooks [2], Torchiano, Ricca, and De Lucia [3], Boehm [4], Koskinen [15], Kuhlmann [16], Bennet and Rajlich [23]). This is not a recent problem. In 1975 Brooks [2] states that over 90% of the costs of a typical system arise in the maintenance phase, and that any successful piece of software will inevitably be maintained. In 1976 Boehm [24] already predicts that most of the overall hardware-software dollar is going into software maintenance and this number is likely to grow for a long time, as organizations continue to add to the inventory of code via development at a faster rate than making code obsolete. In 1991 Abran [14] presents the empirical data from a two-year measurement effort in maintenance environment of an organization. In 2010 Desharnais [25], presents an empirical study with 2 years and half measurement effort about the productivity at maintenance phase of an organization. Desharnais used IFPUG measurement method to size the applications but not for the small change requests. In 1996 Basili *et al.* wrote there is not enough empirical study and data for maintenance phase of software [3] (Basili Software maintenance predictive process), even if before and after this remark some significant empirical studies were held on software maintenance include Swanson [26], Lehman [27], Genuchten [28], Arfa [29], Abran [14], Desharnais [25]and others.

The maintenance costs in Table 2.1 presents references only through 2000. However, what about maintenance cost after 2000. In the article [30] about the economics of software maintenance in the twenty first century, the cost percentage of maintenance is growing.

As presented in Table 2.2, most of the IT personnel are assigned to software maintenance. Software systems are simulation of the real world activities. So, when the world changes, the software need to change accordingly. Latest studies also show that software maintenance is predominant activity in software engineering. Maintenance

Table 2.1. Proportional Software Maintenance Costs for its Supplier.

Year	**Proportion of Software Maintenance Costs**	**Definition**	**Reference**
2000	>90%	Software cost devoted to system maintenance and evolution /total software costs	Erlikh (2000)
1993	75%	Software maintenance / information system budget (in fortune 1000 companies)	Eastwood (1993)
1990	>90%	Software cost devoted to system maintenance and evolution / total software costs	Moad (1990)
1990	60-70%	Software maintenance / total management information systems (MIS) operationg budgets	Huff (1990)
1988	60-70%	Software maintenance / total management information systems (MIS) operationg budgets	Port (1988)
1984	65-75%	Effort spent on software maintenance / total available software engineering effort	McKee (1984)
1981	>50%	Staff time spent on maintenance / total time (in 487 organizations)	Lientz and Swanson (1981)
1979	67%	Maintenance costs / total software costs	Zelkowitz et al. (1979)

Table 2.2. U.S Software Populations in Development and Maintenance.

Year	**Develoment Personnel**	**Maintenance Personnel**	**Total Personnel**	**Maintenance Percent**
2000	750,000	2,000,000	2,750,000	72.73%
2005	775,000	2,500,000	3,275,000	76.34%
2010	800,000	3,000,000	3,800,000	78.95%
2015	1,000,000	3,500,000	4,500,000	77.78%
2020	1,100,000	3,750,000	4,850,000	77.32%
2025	1,250,000	4,250,000	5,500,000	77.27%

cost is 90% of total cost of typical software [31].

In 2000, Bennett and Rajlich [23] wrote: 'Software maintenance and evolution are characterized by their huge cost and slow speed of implementation'. Koskinen [15] wrote in 2010: 'Software maintenance and evolution is a considerably understudied area while taking into account its cost effect'. About characteristic of maintenance cost, a document from Department of Computer Science at Tufts University [32] says: 'In order to increase the maintainability of software, we need to know what characteristics of a product affect its maintainability. There has been a great deal of speculation about what makes a software system more difficult to maintain. There are some program characteristics that are found to affect a product's maintainability'.

In 1983 Martin and McClure [33] already suggested the following factors: system size, system age, number of input/output data items, application type, programming language, and the degree of structure.

For this reasons, characteristics of the maintenance of the bank organization should be analyze to be introduced in the productivity model(s). Those characteristics are mainly 'technical' by nature. To analyze the productivity it will be then necessary to apply a measurement method that is independent of the technique, namely a

functional size method, that will be explain with the methodology. Finally, this study could not be held without considering what is available in term of data in this enterprise under study. In the organization studied, maintenance requests are received via an Enterprise Resource Planning (ERP) system. From there, the Maintenance Manager (MM) handles requests by deciding the priorities and assigns them to the parties who will resolve them. A specific team handles the maintenance requests received for one specific module. Maintenance Manager can reach the details of the problem and measurements like duration, priority for each single request. However, the Maintenance Manager cannot monitor the performance of the maintenance process as a whole. For this reason, in this study feedback loops are placed to the studied processes, completed with measurements and planning. Therefore the process can be checked to identify and change the parts that need improvements. Through these feedback loops, a learning mechanism will be able to apply and in smaller portions at a time the processes will evolve to the mature processes.

In the content of this section, the concepts, which are used to construct productivity model(s) for small enhancements, are define.

2.1. Enhancement

Enhancement is defined as additions and changes to existing applications [21,34]. Maintenance and enhancement are generally defined as activities, which keep systems operational and meet user needs [26]. Maintenance work can be further divided into either major enhancement projects or small maintenance work [21].

2.1.1. Major Enhancement Project

The work carried out to enhance (add or modify) existing applications is complex and requires a team effort. Usually a business case must be prepared, must be approved by senior management and the workload must be handled within a project structure. These major enhancement projects must be planned ahead of time, usually on an annual basis at budget time. The nature of the work carried out for major enhancements

can be classified as a variation of the development life cycle process [34, 35].

2.1.2. Small Enhancements

Small enhancements are carried out based on a service concept, while both major maintenance work and development projects are handled within a project management structure (Abran 1993 [35]).

The main characteristics of small enhancements are:

- limited scope and time required to complete them, usually with only one or two programmers (Abran 1993 [35]),
- not requiring a project management structure for the performance of the work and it is handled very differently [34, 35],
- work request come in more or less randomly and work request will require less than 60 person-days to complete [34, 35],
- work requests cannot be accounted for individually in the annual budget-planning process [34, 35],
- work requests are reviewed by operations committees which are responsible for assigning priorities. Priorities can be shifted around at any time, and any work request on a production problem takes priority over work in progress [34, 35],
- the operations committees must work within present annual budgets, and they must manage their queue of work requests (requests outstanding, completed, in progress, etc.) [34, 35],
- absence of structured planning and formal project reporting during the process [36].

2.2. Productivity and Unit Cost

In economics, productivity is indeed defined as the ratio of the output of a process to its inputs. In this study, productivity is the software product over the effort required to develop the software [36]. Example: number of function points per day.

$$Productivity = \frac{Outputs\ produced\ by\ process}{Inputs\ consumed\ by\ process}$$

Productivity improvement can be stated as a reduction in unit cost or as an increase in output per unit cost [37]. From this definition unit cost is just the inverse of the productivity and often used in an estimation model. It can be formulated as:

$$Unit\ Cost = \frac{Inputs\ consumed\ by\ process}{Outputs\ produced\ by\ process}$$

Measures of the output of the maintenance process are required for productivity studies of software maintenance. Ideally, different types of output of the maintenance process are therefore required [36].

2.2.1. Software Productivity Concepts of Input and Output

By increasing software's outputs, decreasing its inputs, or both we can improve the productivity of the software. However, this means that we need to provide meaningful definitions of the inputs and outputs of the software process.

Defining inputs: For the software process, providing a meaningful definition of inputs is a nontrivial but generally workable problem. Inputs to the software process generally comprise labor, computers, supplies, and other support facilities and equipment. However, one has to be careful which of various classes of items are to be counted as inputs [4].

Defining outputs: The big problem in defining software productivity is defining outputs. Most sources say that defining delivered source instructions (DSI) or lines of codes as the output of the software process is totally inadequate, and they argues that there are a number of deficiencies in using DSI. However, most organizations doing practical productivity measurement, still use DSI as their primary metric [4].

Who should measure software productivity [38]?

- Programmer self-report
- Project or team manager
- Outside analysts or observers
- Automated performance monitors

2.3. Productivity Model

Productivity model is a process improvement approach that provides organizations with the essential elements of effective processes that ultimately improve their performance [25]. Productivity model is used in organizations to improve quality and productivity such as developing more valuable products for lower costs. Object is request from product and process. We measure the product and we look at the process while measuring it. Well-known examples of productivity models are COCOMO and ISBSG productivity models.

2.3.1. COCOMO

One of the most popular productivity model is Constructive Cost Model COCOMO [6,18–20]. This model is used to support software cost and estimation processes. It also gives information about the reason of the cost and schedule implication of software engineers ‘ and software managers’ development decisions, investment decisions, client negotiations and requested changes, risk management decisions, and process improvement decisions [20]. With COCOMO, software managers and software engineers understand the cost consequences of the decisions they will make in start-up, developing, and supporting a software product. And also COCOMO provides that what costs the model is estimating, and why it comes up with the estimates it does [6].

COCOMO is actually hierarchy of three models. One of them is intermediate level COCOMO. Intermediate COCOMO estimates the cost of a software product in the following way [6]:

(i) A nominal development effort is estimated as a function of the product’s size in

delivered source instructions in thousand (KDSI) and the project's development mode.

(ii) A set of effort multipliers are determined from the product's ratings on a set of 15 cost driver attributes.

(iii) The estimated development effort is obtained by multiplying the nominal effort estimate by all of the product's effort multipliers.

(iv) Additional factors can be used to determine dollar costs, development schedules, phase and activity distributions, computer costs, annual maintenance costs, and other elements from the development effort estimate.

First of all, the characteristics best fit the profile of the software project are determined. Next, the size of the product is estimated in source instructions (KDSI). And then, nominal development effort for the project is determined. So far, these steps are used to estimate nominal effort. Then, complexity rating and corresponding effort multiplier is determined. After that, estimated development effort is calculated (nominal effort X all of the product's effort multipliers).

2.3.2. ISBSG

As mentioned before, ISBSG [39] Development & Enhancement repository that has over 5600 projects and the ISBSG [40] Maintenance & Support repository has only 470 applications. This industry data can be used to estimate benchmark and improve the planning and management of projects. Application types are recorded in this repository and projects are grouped in major categories. Each project has the following types of information:

- Functional size (it can be from any of the FSMs)
- Effort in hours
- Different characteristics for each project (like language, domain, the experience of the IT people, etc.)

The projects in this database used as a reference for the effort estimations of projects.

In the following the steps are listed to calculate the effort per CFP value for a project by using ISBSG historical database:

(i) To choose typical projects from the historical database criteria like language, domain is defined.
(ii) According to specified criteria a filter is applied on historical database.
(iii) Per CFP lower, average and higher efforts are calculated and this gives unit effort.
(iv) R-square is calculated to determine the reliability of the results.
(v) Per CFP effort is decided to apply which may be a single or a range of unit effort.

2.4. Functional Size Measurement (FSM) Method

There is a standard that define what as a Functional Size Measurement (FSM) since the middle of the 90s [40]. All functional size measurement methods are based on the assumptions of a simplified model of software functionality that is intended to be reasonable on average for its intended domain of applicability. Caution is therefore needed when measuring, comparing or using sizes of very small pieces of software, and especially of very small changes to a piece of software, where the average assumption may break down [5].

Functional Size Measurement (FSM) methods have been designed to measure the functional size of software applications from a user's perspective. While it is being used extensively to measure software development or medium-large enhancement projects, it has not been used to measure very small functional enhancements: the current measurement structure (for example the use of tables) of some FSM methods does not allow it to discriminate small size increments [25]. The main characteristic of small functional enhancements is the limited scope and time required to complete them, usually with one or two programmers. Small functional enhancements consist of small additions or modifications to existing software [36]. COSMIC FSM method [5] (is able

to measure small enhancement because of the possibility to capture a lower level of granularity. Tables problems are inexistent because there are no tables but just the count of the number of data movement with COSMIC measurement method.

2.5. COSMIC Measurement Method

COSMIC measurement method is mainly used to measure the functional size of the functional processes, but recently the use of COSMIC measurement method is also considered to evaluate the quality of documentation in a separate document [5]. There is also a Measurement Manual [5] that explains how to apply the measurement method.

COSMIC is a measurement method [41, 42] that was first accepted by ISO/IEC JTCI SC7 in December 2002, as an International Standard [43]. It involves applying a set of models, principles, rules, and processes to the Functional User Requirements (FUR) of a given piece of software to produce a numerical value of a quantity which represents the functional size of the piece of software according to the COSMIC method [5, 43]. Only the FUR is taken into account when measuring a functional size. Functional User Requirements describes what the software must do for the functional users, which are senders of data to the system and intended recipients of data from the system, and it completely excludes any requirements that state how the software performs.

2.5.1. COSMIC Measurement Method Phases

COSMIC measurement method is designed to provide a standardized method of measuring a functional size of software from the functional domains commonly referred to as business application (or MIS) software(needed in support of business administration) and real-time software (controls events happening in real world). Measured functional sizes are independent of all technology decisions and implementation details of the software. So, COSMIC measurement method can be applied as soon as user requirements are determined.

The general COSMIC measurement process consists of their phases:

- The Measurement Strategy Phase: in which the Software Context Model (a piece of software carefully defined) is applied to the software to be measured. Four key parameters of software functional size (the purpose and scope of the measurement, the identification of functional users and the level of granularity) are addressed in this phase.
- The Mapping Phase: the Generic Software Model (identifies the components of the functionality) is applied to the software to be measured. Functional user requirement in the artifacts of the software to be measured is kept as input and FUR in the form of the Software Generic Model is obtained as output.
- The Measurement Phase: actual size of measurements is obtained in this phase. In the first step data movements are obtained, measurement function applied in the second step and in the last step measurement results are aggregated.

The relation among each phase is showed in Figure 2.1.

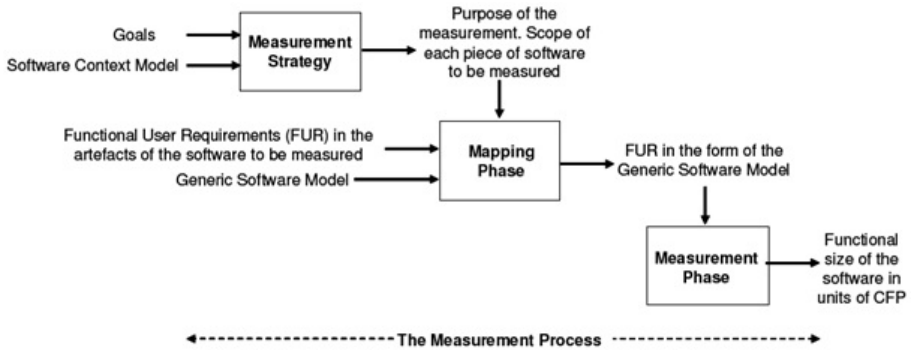


Figure 2.1. Structure of the COSMIC Measurement Method.

At the end of Measurement Phase, functional size of the software is obtained in terms of CFP. If all functional processes are not measured at the end of second step, the first step is iterated.

2.5.1.1. Identification of Data Movements. A data movement is a base functional component, which moves a single data group type which is distinct, non-empty, non-

ordered, and non-redundant set of data attributes where each included data attribute describes an aspect of the same object of interest [5]. There are four sub-types of data movement:

- Entry data movement (E): this moves a data group from a functional user to the functional process where it is required.
- Read data movement (R): this moves a data group from a persistent storage to the functional process where it is required.
- Write data movement (W): this moves a data group from a functional process to a persistent storage.
- Exit data movement (X): this moves a data group from a functional process to the functional user that requires it.

Figure 2.2 shows the relationships between data movement types:

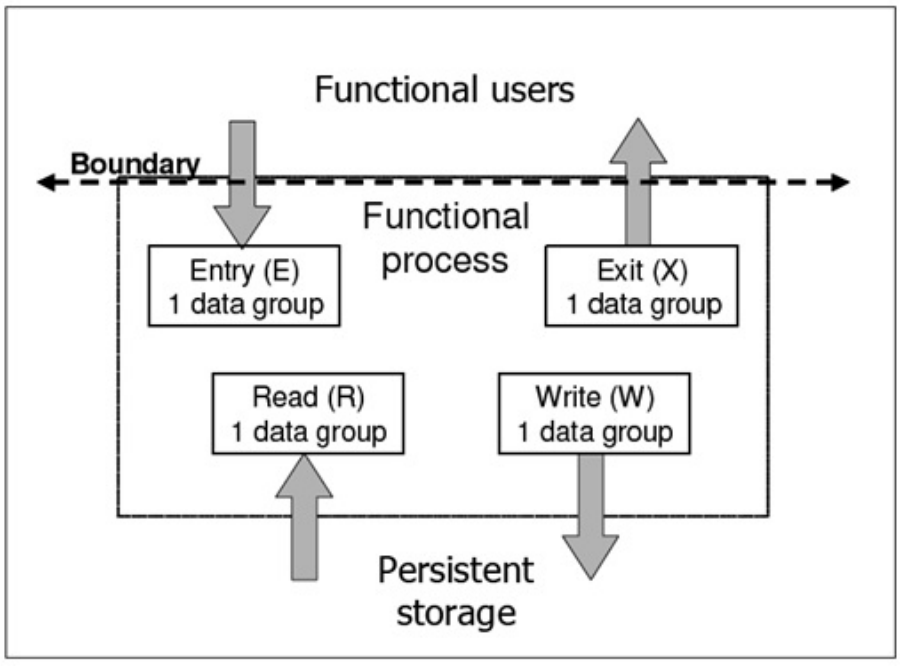


Figure 2.2. The four types of data movement and their relationships with the functional process and data groups.

Each data movement must be identified after functional processes are determined. As shown in Figure 2.2, there is strong relationship between the data movements

and data groups since data movements identified data group types towards related concept. In this document the word 'module' is also use, but not define in the context of COSMIC. This definition of module is use in the context of the organization where we took the measure of small enhancements. In computer programming, especially in older languages such as PL/1, the output of the language compiler was known as an object module to distinguish it from the set of source language statements, sometimes known as the source module. In mainframe systems such as IBM's OS/360, the object module was then linked together with other object modules to form a load module. The load module was the executable code that you ran in the computer [44].

2.5.1.2. Application of Measurement Function. In this step, COMSIC measurement function is applied to each data movement identified in functional processes where COSMIC measurement function represents a mathematical function, which assigns a value to its data movement based on the COSMIC measurement standard [43]. According to COSMIC measurement standard, size of one data movement is identified as one CFP. For this reason, each identified data movement is counted as one CFP after measurement function is applied.

2.5.1.3. Aggregation of Measurement Results. After measurement function is applied to each data movement, the results must be aggregated into on functional size value. The aggregation of the results consists of adding up all identified CFs of each functional process. This aggregation function is shown in Equation 2.1.

$$size(FP_i) = \sum size(Entry_i) + \sum size(Read_i) + \sum size(Write_i) + \sum size(Exit_i)$$

Where, i is the functional process number. The result of this equation shows the size of the i the functional process in terms of CFPs.

This formula can be applied to all identified functional processes in order to find the overall functional size of given small enhancement.

2.5.2. COSMIC Verification Method for Rating Documentation Quality

COSMIC method is used for measuring the functional size of the functional processes. However, it is also possible to identify the quality of the documentation of the functional processes by using COSMIC verification method [45]. This quality rating operation is a good support to the measurement, and it provides indirectly the capability of analyzing the quality of the measurement result (CFP). Quality analysis document is not a part of the Measurement Manual but another guideline [46]. After each functional process is measured in terms of CFP, the documentation for each functional process is checked based on a number of criteria. Thus, the reliability and acceptability of the measurement is identified. Since there is a rating for each functional process, documentation of each functional process must be analyzed separately. The rating scale for the documentation quality is from (a) to (e), and their descriptions according to COSMIC guide [45] are showed in Table 2.3. They are part of the methodology.

Table 2.3. Documentation Quality Rating Scale.

Rating	**Description**
A	The functional process is completely documented together with its data movements by type.
B	The functional process is documented but the description of the data moved is unclear. The input, output, stores and retrievals of each functional process are also described but not clearly enough to identify the number of data movements.
C	The functional process is identified only but their data movements are not.
D	The number of functional process is given but they are not specified.
E	The functional process is not mentioned in the artifacts but is implicit.

According to the available documentation of each functional process, quality of

the measurement result is analyzed, and this provides the opportunity to determine whether the measurement is reliable or not.

2.6. Granularity

In COSMIC measurement methodology to define level of granularity is important. The COSMIC manual [5] defines level of granularity as: "Any level of expansion of the description of a single piece of software (e.g. a statement of its requirements, or a description of the structure of the piece of software) such that at each increased level of expansion, the description of the functionality of the piece of software is at an increased and uniform level of detail".

Measurers should be aware that when requirements are evolving early in the life of a software project, at any moment different parts of the required software functionality will typically have been documented at different levels of granularity.

In this manual [5] there are two levels of granularity: higher level and lower level. In the initial stage of a software project, functional user requirements (FURs) are defined at a higher level. As the project progress, the FURs are defined revealing more detail at a lower level.

There are no standard scales for the various level of granularity at which software may be specified. For this reason, it is not easy to be sure those two statements of FUR are at the same level of granularity. It is impossible to know for sure that two functional size measurements may be compared. Measures may have to develop their own method of scaling measurements at one level of granularity to another.

To illustrate the problem let's think the plan of flat.

- Plan A shows only the rooms in the first floor of the flat
- Plan B shows the all floors in a flat
- Plan C shows all floors of the all flats in one site

The abstract concept of 'level of granularity' lies behind the scales of these different plans.

For software measurement, there is only one standard level of granularity that it is possible to define unambiguously. That is the level of granularity at which individual functional processes have been identified and their data movements defined. Measurements should be made at this level or scaled to this level whenever possible [5]. One of the rules in COSMIC measurement method about the level of granularity is that functional size measurement should be made at the functional process level of granularity.

2.7. Software Complexity

Basili [47] defines software complexity as a measure of the resources expanded by another system in interacting with a piece of software. Machines, people, other software, and even external environment are categories of systems that may interact with the software. The measures deal with execution time and memory space if the interacting system is a machine. If the interacting system is software, the measures might focus on the number of interfaces. If the interacting system is people, the measures are concerned with human efforts to comprehend, to maintain, to change, to test, etc., that software. If a software development project requires travel to another site, there are certain physical limitations and expenditures in travel time that must be considered. These are the constraints that external environment acts more like a set of them.

3. METHODOLOGY USED

The methodology of this study has the following steps to construct productivity model for small enhancements in software maintenance:

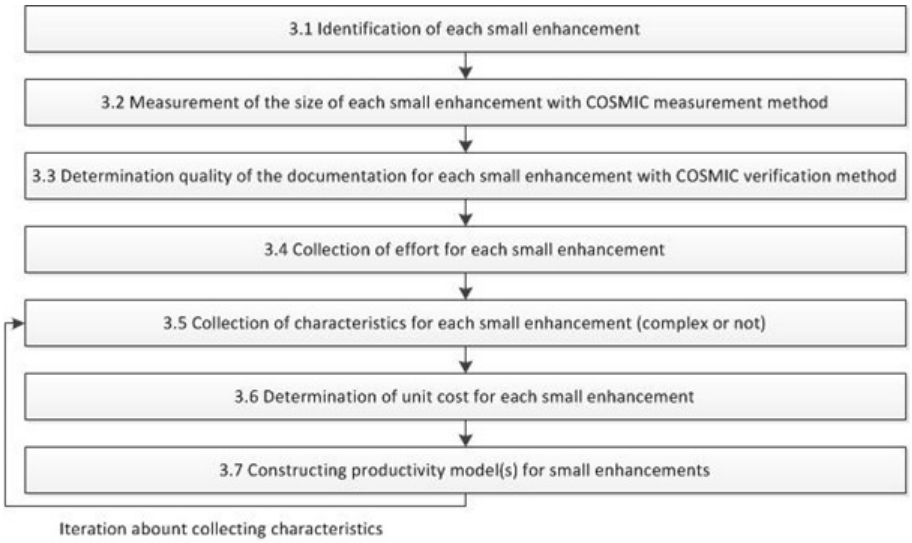


Figure 3.1. Steps of the Used Methodology.

3.1. Identification of Each Small Enhancement

First small enhancements need to be determined. The selected small enhancements have the following properties in common:

- Adaptive and corrective maintenance type only,
- The duration varies between half a day and 3 weeks,
- Less than twenty (20) CFPs (Cosmic Function Points),
- 88 small enhancements of a large application designed, programmed and implemented between April 2010 to December 2011,
- The small enhancements are from the change request system of the organization,
- Effort and duration must be available and reliable,
- Documentation is well known by the measurer,
- Domain must be management information system (MIS),

- Traditional languages in mainframe environment (e.g., COBOL language, PL/1 or tool like Cool:Gen using DB2 or Oracle database)

So, in this step all required information about small enhancements is provided to measure each of them. Finally, the small enhancements were found in the organization using change requests system. A change request (CR) is defined as "proposed changes to a product that is being maintained" by ISO/IEC 14764 standard [7]. In this study the data used for analysis is obtained basically from the CRs reported via Enterprise Resource Planning (ERP) system of the organization . ERP is using techniques and concepts for integrated management of businesses as a whole from the viewpoint of the effective use of management resources to improve the efficiency of enterprise management [48]. The documentation is well known because the measurer is also the person who design, program and implement each small enhancement. It is also the person how have to complete the effort and duration of each small enhancement.

The criteria exclude user support because not related specifically to the construction of software. Preventive and perfective were also excluded because it was not possible to find enough activities maintenance related to those categories. This left then adaptive and corrective.

3.2. Measurement of the Size of Each Small Enhancement with COSMIC Measurement Method

In this step, COSMIC [5] measurement method was applied to each small enhancement. In order to calculate the size of a small enhancement, firstly all of its functional processes are defined, and then functional processes were measured in terms of CFP with COSMIC measurement method. The maintainer who applied the measurement method is the one who has done the design, programming and implementation of each small enhancement. Afterwards, an independent functional measurement expert has verified the number of CFP obtained for each functional process based the documentation of each small enhancement (see examples in annex B).

Each data movement must be identified after functional processes are determined. A data movement is a base functional component [5]. In data movement identifying phase, Entry, Read, Write, Exit data movements of each functional process is specified. In this step for each data movement some specific principles and rules are applied.

Principles of Entry Data Movement [5]:

- An Entry shall not exit data across the boundary, or read, or write.
- One Entry is identified to the functional process for obtaining data. Any message from the functional process to the functional user seeking to retrieve the data should not be counted as an Exit. However, where a functional process must obtain some data from a functional user such as typing answer of a question, count one Exit for the request and one Entry for the return of the requested data.

Principles of Exit Data Movement [5]:

- An Exit shall not data across the boundary, or read, or write.

Principles of Read Data Movement [5]:

- A Read shall not receive or exit data across the boundary or write data.
- During a functional process movement or manipulation of constants or variables shall not be considered as Read data movements.
- A Read data movement always includes any "request to Read" functionality.

Principles of Write Data Movement [5]:

- A Write shall not exit or receive data across the boundary, or read data.
- Deletions of a data group from persistent storage shall be measured as a single Write data movement.
- Updating internal variables, manipulation or movement of data dose not persis-

tent shall not be counted as a Write data movement.

According to these principles Entry, Read, Write, and Exit data movements are counted and then these counts are aggregated.

In this approach, it is showed that small enhancements can be measured by using the COSMIC [5] measurement method that has the possibility to capture a lower level of granularity.

3.3. Determination Quality of the Information Provided for Each Small Enhancement with COSMIC Verification Method

The quality of documentation for each small enhancement is studied based on COSMIC verification process requirements [5]. This helped to verify the quality of the measure of the functional size measure obtained. Each identified small enhancement is rated by using Table 2.3. The determination of the quality of the documentation can be assessed based on facts that can be listed as:

- The presence or absence of a data model
- Description of the data moved to identify data movements
- Identification of each functional process

According to provided information used in the definition, each small enhancement needs to be rated for its documentation quality. The small enhancements used in this research are rated after their measurement. The definition is enough to define functional process, data model and identify data movements (entry, read, write, and exit). Because the person who has done the maintenance was at hand for this exercise, it was possible to complete the documentation. So, the quality of the documentation related to small enhancements used in this research is high.

3.4. Collection of Effort for Each Small Enhancement

In the core banking system, the maintenance requests are received via an ERP system that captures change requests. The Maintenance Manager (MM) then looks at individual requests and decides first its priority and second assigns it to a maintainer who will assess and resolve them. The maintenance requests received for one specific module are handled by a specific team.

The team member who is responsible to solve the request enters activities for the tasks assigned to them weekly with the ID of the task correspond to a maintenance request. Then, effort is calculated from the activities by summing hours for each single request. In the process of collecting small enhancements, effort should be considered to be available and reliable. The small enhancements which do not have effort measure in ERP system are not used in this study.

Effort is acquired from the computerized reporting system of the bank and expressed in person-hours. Maintenance activities are differentiated by their related modules. While entering a maintenance request, the related module needed to be identified and selected from the configuration file. Then, activities in each module are grouped request by request. The last step is to order activities according to their effort in hours. The pre-requisites of [13] are met by the bank before attempting to measure the software maintainers' productivity.

3.5. Collection of Characteristics of Each Small Enhancement

The measures that will quantify the cost drivers selected are listed as:

- Functional size
- Maintenance categories: adaptive and corrective maintenance types are handled
- Effort required for each small enhancement (mentioned at Appendix A)
- Development Tool
- Batch and online applications

The chose measure for the unit of the functional size of each small enhancement is Common Software Measurement International Consortium (COSMIC) [45].

The software development tool used to maintain small enhancements is one of the selected characteristics to differentiate enhancements. The most frequently used development tools are PL/I and Cool:GEN in the company. So, the small enhancements used in this research are gathered according to development tool.

Three types of reach ability used in ERP system are batch, online and batch/online. The set of small enhancements includes these three types. Batch applications are submitted end of the day in a schedule. They work sequentially or in parallel. Online applications are mostly used in day time. Almost all of the online applications are developed by using Cool:GEN tool and most of the batch applications are developed by using PL/I, some of them are developed with COBOL.

3.6. Determination of Unit Cost for Each Small Enhancement

After the maintenance requests are ordered according to efforts, especially small enhancements with the effort less than 50 hours are selected to analyze.

The maintenance process has many inputs and outputs, which in turn has many characteristics that can be measured. Unit cost is formulated as the ratio of the inputs consumed by the process to outputs produced by the process. In calculation of functional size of each small enhancement with CFP [5] inputs can be considered as entry data movements and outputs are considered as exit data movements. This can be formulated as below:

$$Unit\ Cost\ Calculation\ of\ Small\ Enhancement = \frac{\#\ of\ Entry\ Data\ Movement(s)}{\#\ of\ Exit\ Data\ Movement(s)}$$

Also, number of functional points per hour is the productivity and unit cost is determined by dividing effort (input) required to develop small enhancements to

functional size (output) of each small enhancement.

$$Unit\ Cost\ of\ Small\ Enhancement = \frac{Effort}{Functional\ Size}$$

3.7. Constructing Productivity Model(s) for Small Enhancements

A productivity model is typically built with [49]:

- Data from projects completed
- When all information on a project is available and that there is no more uncertainty
- All of the software functions have been delivered and
- All of the number of hours for the project have been completed and measured

The ISBSG dataset is investigated by taking into consideration the conditions mentioned above and small enhancements are gathered in the light of these information.

As already mentioned, ISBSG has actually a lot less information about maintenance; most of the data is from the development of software. This research data set consists of maintenance data.

In ISBSG database each project has the following types of information [39]:

- Functional size can be from any of the functional size methods (FSMs)
- Effort in hours(normally this information is related to the development or enhancements part of the life cycle of a project)
- Different characteristics of projects like the domain, the language, the experience of the IT people, etc.

In this research, the selection criteria refer to conditions of acceptance of projects in the sample.

Traditional languages in mainframe environment (e.g PL/I or COBOL language with DB2 or Oracle database) will also be used as selection criteria.

The criteria of availability of the information regarding small enhancements were selected to allow greater reliability of the data and also the consistency in regards to information received for each project.

4. PRESENTATION OF THE DATA

This section defines the attributes of the dataset and provides a descriptive analysis of the dataset. The definitions of statistical concepts for descriptive analysis are in Annex C.

4.1. Definition of the Attributes of the Dataset

Data about software maintenance (small enhancements as define in the Introduction) of a Banking System were collect from April 2010 to December 2011. All the small enhancements in the data set are coming from the organization's ERP system.

ERP system receives the maintenance requests. Each request has the related sub module information, definition of it, type of the request batch or online. According to specified data selection criteria in Section 3.7, the small enhancements were collected from the different sub modules of ERP system. The most important modules were considered based on different criteria (Section 3.7).

In the data set, each small enhancement is presented with the following properties: number of type of data movements (read, write, exit and entry), functional size in CFP [5], effort in hours, type, ratio of effort to functional size, development tool used to enhance, module affect.

Adding to the size of the small enhancements and the efforts, there were some more attributes to group the small enhancements. These are $Write$, $Exit$, $Module$, $Development\ Tool$, $Quality$, $Batch$, $Online$, $Type$. These are used to determine their influence on the effort (to construct a productivity model).

$Write$ criteria shows that, if a small enhancement has two or more data group movement from a functional process to a persistent storage, it is numbered with nominal value "1", else it is numbered with "0". $Exit$ criteria usage is similar to $Write$. If a

small enhancement has two or more data group movement from a functional process to the functional user , it is numbered with nominal value “1”, else it is numbered with “0”. *Module* criteria indicates that if in the step of enhancement, more than one executable code that run in the system is affected, it is numbered with “1”, else “0”.

In organization, the most common used development tools are Cool:GEN and Pl/I. It is more difficult to develop with Cool:GEN than PL/I due to standards of language. So, the small enhancements developed with Cool:GEN has more effort value than developed with PL/I.

Collected small enhancements have three different program types; batch, online and batch/online. Mostly, small enhancements developed with PL/I has batch type, the rest developed with Cool:GEN has online type. There are few small enhancement has the type batch/online in data set, and these are developed with PL/I.

Batch mode programs are submitted at the end of day, online mode programs usually used at day time. Batch/Online programs are used at the end of day and also at day time according to entered working type flag.

Type criteria is formed the ratio of effort to functional size. If functional size of small enhancement is equal to effort, the ratio is 1 and it is labeled with “M”, else “N”.

4.2. Descriptive Statistics of the Dataset

Table 4.1. Used Attributes for Small Enhancements.

	N	
Tool	88	39
Online	88	41
Module	88	22
Type	78	51
Valid N (listwise)	78	

Table 4.1 shows the distribution of small enhancements related to each attribute (name criteria within SPSS tool). For example, there are 39 small enhancements that comply with the criteria Tool, 41 with the criteria Online, 22 with the criteria Module and finally 51 with the criteria Type, the later for only 78 small enhancements, not 88 (column N).

Table 4.1 is giving information about the distribution of data related to different attributes. Finally Valid N (listwise) indicates that 88 small enhancements are valid for all the attributes.

Table 4.2. Data Movement Counts of Small Enhancements.

	N	**Minimum**	**Maximum**	**Sum**	**Mean**
E	88	1	3	95	1.08
R	87	0	6	154	1.77
W	65	0	5	83	1.28
X	88	1	4	109	1.24
Valid N (listwise)	64				

Table 4.2 gives the number of total entry data movement, *Read* data movement, *Write* data movement and *Exit* data movement count. *Minimum* columns shows minimum number of each data movement, *Maximum* column shows the maximum number of each data movement, *Sum* column gives the sum of each data movement type, and *Mean* (sum /88) column gives the mean of each data movement type.

For *Entry* and *Exit* data movement minimum count is 1, which is not surprising. This shows that all small enhancements had *Entry* and *Exit* data movements. The data movement write is present 65% of the time. Therefore, maximum number of data movements belongs to *Read* and *Write* data movements. This shows the relation to write or give an output it is needed to read before.

Table 4.3 shows minimum and maximum number of functional size and effort

Table 4.3. Functional Size and Effort Presentation of Small Enhancements

Descriptive Statistics of effort and CFP.

	N	**Minimum**	**Maximum**	**Mean**	**Std. Deviation**	**Skewness**	
Effort	88	1	40	9.92	8.375	1.667	0.257
CFP	88	3	13	5.01	2.125	1.727	0.257
Valid N (listwise)							

related to small enhancements. The maximum effort is 40 person-hours, and minimum 1 person-hour. The maximum Functional size is 13 CFPs. Standard deviation [50] for effort is less than the Mean (average). This shows no large difference between the minimum and maximum effort. The Skewness [51] indicates that the tail on the left (1,7 or positive) and the Deviation is lower than the Mean. Generally, a lower deviation is a positive result because it shows a difference between minimum and maximum that is not too large. The value of 1,7 indicates that most of small enhancements are on the left side of the curve.

Table 4.4. Data Movement Counts of Small Enhancements.

	N	**Minimum**	**Maximum**	**Sum**	**Mean**	**Std. Deviation**
Unit Effort	88	0.33	4.00	158.42	1.8003	0.94111
Valid N (listwise)						

In Table 4.4, the unit effort (Effort per CFP) criteria is taken into consideration. Maximum ratio belongs to the small enhancement which has the maximum effort. Functional size of this small enhancement is closed to maximum.

In Table 4.5, *Online* type is numbered with "1" and *Batch* type is numbered

Table 4.5. Cross Tabulation between Development Tool and Online Criteria.

Count		**Online**		**Total**
		0	1	
Tool	0	47	2	49
	1	0	39	39
Total		47	41	88

with "0". The same method is used for development tool. "0" is used for Cool:GEN and "1" is for PL/I. The cross tabulation shows that all small enhancements developed with Cool:GEN has online mode and PL/I programming language is used to developed all batch mode small enhancements.

Table 4.6. Cross Tabulation between Module and Online Criteria.

Count		**Online**		**Total**
		0	1	
Module	0	30	36	66
	1	17	5	22
Total		47	41	88

In Table 4.6, the small enhancements that need to use two more modules are numbered with "1" (small enhancement needs to use more than one module) and "0" if the (small enhancement does not need to use more than one module). According to Table 4.6, in batch mode, the small enhancements are using more modules than in online mode. Table 4.6 shows that most of the small enhancements need to use only one related module.

In conclusion, for this study, functional size method (COSMIC) of 88 functional processes (minor change in application) was applied in 2011 measuring also the efforts (hours), the number of data movements by type for each functional process, the type of development tool and the number of steps to reach a module for each functional

process. Using those data the goal is to answer a main question: is it possible to obtain a productivity model for small enhancements? This is the topic of the next chapter.

5. ANALYSIS OF THE DATA

To analyze the data, regression statistics method was used. The acceptance of regression results hinges on diagnostic checking for the breakdown of classical assumptions. In this study the classical assumption to a regression model (effort and independent variables) is 70%. If there is a breakdown, then the estimation is unreliable and thus the interpretation is unreliable [52].

The most important part of analysis is to gather data. The dataset was presented and defined in Chapter 4. It was also mention that the quality of the documentation was high. To proceed for a regression analysis it is also necessary to follow a procedure. Figure 5.1 is showing the steps for regression analysis using a flow chart diagram. Firstly, main contribution is defined and which method will be used to solve the problem is selected. Then model is defined. After method and model decision, small enhancements data set is read into SPSS software. The read data is prepared for analysis. The first statistical analyses are descriptive (see chapter 4). From descriptive statistics no outlier were found.

This part also has a productivity analysis of the main attributes: PL/1 and Cool:Gen, batch and online, adaptive and corrective maintenance.

5.1. Regression Analysis

Next steps are about the regression analysis. This study will use this flow diagram for regression analysis using the following steps in this chapter:

(i) Choose which variables will be analyze
(ii) Run a regression analysis
(iii) Present the results in a Figure
(iv) Interpret the results
(v) Accept or reject the results (in the productivity model)

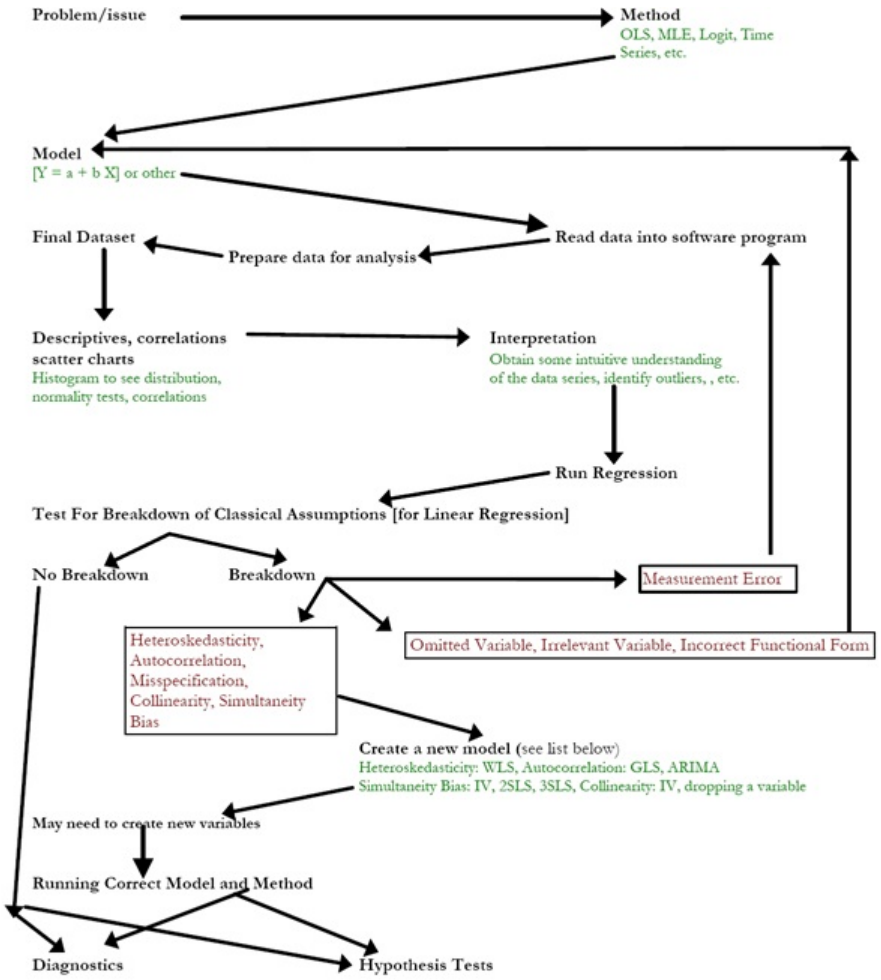


Figure 5.1. Flow Diagram for Regression Analysis.

(vi) Start again from new choices of variables

5.1.1. Regression Analysis with CFP

In the first regression analysis, functional size of small enhancements is selected as independent variable as showned in Table 5.1 and for productivity model test effort is dependent model. With this analysis, it is aimed to find relation between functional size and effort.

$$Effort = \text{function } (functional\ size\ in\ CFP)$$

Table 5.1. Variables Entered/$Removed^a$.

Model	Variables Entered	Variables Removed	Method
1	CFP^b		Enter

a.Dependent Variable: Effort

b.All requested variables entered.

Table 5.2. Model Summary.

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	0.870^a	0.757	0.754	4.155

a. Predictors: (Constant), CFP

In Table 5.2 model summary part, "Adjusted R Square" shows that 75.7% of the variance was explained and "R-square" shows that 75.4% of the variation (and not the variance) was explained. "Std. Error of the Estimate" is 4.155. If the Std.Error is more than 10% of the mean, it is high. In this model it is less that 10%.

When looking at the fit "ANOVA" in Table 5.3 , the goodness of fit of the model is 0.00. It can be concluded that the model fit the data (If "Sig. is greater than

Table 5.3. $ANOVA^a$.

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	4617.496	1	4617.496	267.420	0.000*b
	Residual	1484.948	86	17.267		
	Total	6102.443	87			

a.Dependent Variable: Effort

b. Predictors: (Constant), CFP

Table 5.4. $Coefficients^a$.

Model		Unstandardized Coefficients		Standard Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	-7.257	1.140		-6.366	0.000
	CFP	3.428	0.210	0.870	16.353	0.000

a.Dependent Variable: Effort

0.05, the model could not fit the data [52]). The "Sum of Squares" column holds the values for "Total Sum of Squares", "Explained Sum of Squares", "Not Explained Sum of Squares". Here, explained sum of squares is showed in row Regression and it equals to 4617.496. Residual row contains "Not Explained Sum of Squares". The model fit shows that the amount (4617.496) that could be explained by the model is greater than the amount (1484.948) that could not be explained by the model.

The R-square in "Coefficients" table is the ratio of "Explained Sum of Squares" to "Total Sum of Squares" (ESS/TSS). It captures the percent of deviation from the mean in the dependent variable "Effort" that could not be explained by the model. In this analysis, R-square is 0.757. For high R-square, ESS should be high compared to TSS.

The values "Sign." in "Coefficients" table are 0.00. It means that the estimate in column "B" can be asserted as true with a level 100%. This table provides information about effect of independent variable "CFP (functional size)" on the dependent variable "Effort". In software engineering, to say something R-square should be greater than 0.500. In this analysis it is 0.757. It is needed to analyze other properties of small enhancements to get better R-square values.

The Figure 5.2 is showing Effort in hours in the left and CFP below. The number of hours is from 1 through 39 hours with CFPs from through 13. There is no outlier.

5.1.2. Regression Analysis with Batch/Online

In the second regression analysis, functional size of small enhancements and batch/online properties are selected as independent variable and effort is dependent model. Table 5.5 shows the dependent and independent variables in this analysis. The usage of Batch/Online independent variable with functional size is to observe what is going on the results if two independent variables are used together.

$$Effort = \text{function } (functional\ size\ in\ CFP, Batch/Online)$$

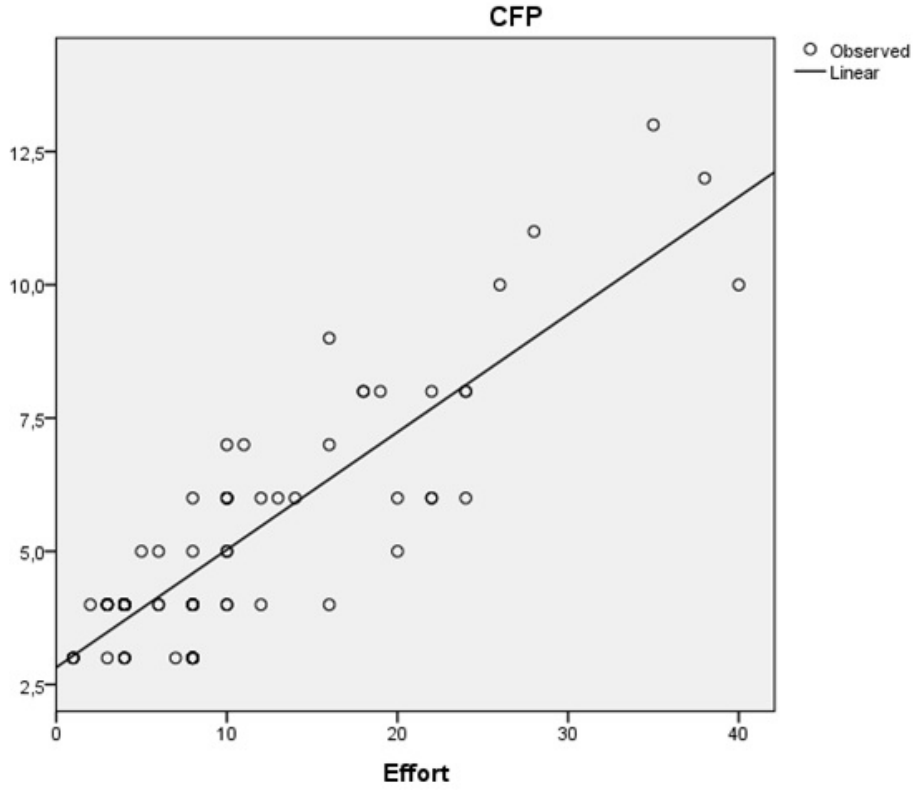


Figure 5.2. Regression with Effort and CFP for 88 small enhancements.

Table 5.5. Variables Entered/*Removed*[a].

Model	Variables Entered	Variables Removed	Method
1	Online, CFP^b		Enter

a.Dependent Variable: Effort

b.All requested variables entered.

Table 5.6. Model Summary.

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	0.873^a	0.761	0.756	4.139

a. Predictors: (Constant), Online, CFP

The analysis results in this trial are better than first analysis results, but with a non-significant improvement (,002). The R-square in "Coefficients" Table 5.8 is 0.761. The Model 2 has little enhancement compare to Model 1. The ESS ratio to TSS in Model 2 is bigger than the ESS ratio to TSS in Model 1.

When we look at the fit "ANOVA" in Table 5.7 , the goodness of fit of the model is 0.00 again. The model still fit the data. In this analysis, explained sum of squares showed in row Regression equals to 4646.417. Residual equals to 1456.027. The model fit shows that the amount (4646.417) that could be explained by the model is more than the amount (1456.027) that could not be explained by the model.

The values "Sign." in "Coefficients" table are still 0.000. So, the estimate in column "B" can be asserted as true with a level 100%.

In this trail, R-square is 0.761. As a result, in this regression analysis the chosen dependent variables functional size and batch/online have better results than regression analysis with only functional size dependent variable. It is needed to analyze other

Table 5.7. $ANOVA^a$.

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	4646.417	2	2323.208	135.624	0.000*b
	Residual	1456.027	85	17.130		
	Total	6102.443	87			

a.Dependent Variable: Effort

b. Predictors: (Constant), Online, CFP

Table 5.8. $Coefficients^a$.

Model		Unstandardized Coefficients		Standard Coeffi-cients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	-7.741	1.195		-6.478	0.000
	CFP	3.417	0.209	0.867	16.357	0.000
	Online	1.150	0.885	0.069	1.299	0.197

a.Dependent Variable: Effort

properties of small enhancements.

5.1.3. Regression Analysis with Tool

In the third regression analysis, functional size of small enhancements and tool properties are selected as independent variable and effort is dependent model.

$$Effort = \text{function } (functional\ size\ in\ CFP, Tool)$$

Table 5.9. Variables Entered/$Removed^a$.

Model	Variables Entered	Variables Removed	Method
1	Tool, CFP*b		Enter

a.Dependent Variable: Effort

b.All requested variables entered.

Table 5.10. Model Summary.

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	0.871^a	0.759	0.753	4.162

a. Predictors: (Constant), Tool, CFP

When analysis results in tis trial are investigated, the R-square in Table 5.10 "Model Summary" is 0.759. The Model 3 has little enhancement compare to Model 1. The ESS ratio to TSS in Model 3 is bigger than the ESS ratio to TSS in Model 1. The Model 3 and Model 2 results are similar. In Chapter 4, it is presented that most of batch modules are developed with Pl/I and online modules are developed with Cool:GEN. The comparison of second and third analysis supports this presentation.

Table 5.11. $ANOVA^a$.

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	4630.291	2	2315.145	133.673	0.000*b
	Residual	1472.153	85	17.319		
	Total	6102.443	87			

a.Dependent Variable: Effort

b. Predictors: (Constant), Tool, CFP

Table 5.12. $Coefficients^a$.

Model		Unstandardized Coefficients		Standard Coeffi-cients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	-7.534	1.186		-6.351	0.000
	CFP	3.415	0.210	0.867	16.226	0.000
	Tool	0.770	0.895	0.046	0.860	0.392

a.Dependent Variable: Effort

The result in this regression analysis shows that the effort of small enhancements changes according to selected development tool. However, the effect of Batch/Online property is greater than Development Tool property.

5.1.4. Regression Analysis with Exit

In this regression analysis, functional size of small enhancements and exit properties are selected as independent variable and effort is dependent model.

$$Effort = \text{function } (functional\ size\ in\ CFP, Exit)$$

Table 5.13. Variables Entered/$Removed^a$.

Model	Variables Entered	Variables Removed	Method
1	Exit, CFP*b		Enter

a.Dependent Variable: Effort

b.All requested variables entered.

Table 5.14. Model Summary.

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	0.879^a	0.772	0.767	4.046

a. Predictors: (Constant), Exit, CFP

When we look at the fit "ANOVA" in Table 5.15, the goodness of fit of the model is again 0.00. The model still fit the data. In this analysis, explained sum of squares showed in row Regression equals to 4711.145. Residual equals to 1391.298. The model fit shows that the amount that could be explained by the Model 4 is better than Model 2 and 3.

Table 5.15. $ANOVA^a$.

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	4711.145	2	2355.572	143.911	0.000*b
	Residual	1391.298	85	16.368		
	Total	6102.443	87			

a.Dependent Variable: Effort

b. Predictors: (Constant), Exit, CFP

Table 5.16. $Coefficients^a$.

Model		Unstandardized Coefficients		Standard Coeffi-cients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	-9.141	1.361		-6.717	0.000
	CFP	3.894	0.282	0.988	13.802	0.000
	Exit	-4.960	2.074	-0.171	-2.392	0.019

a.Dependent Variable: Effort

The R-square in "Model Summary" Table 5.14 is 0.772. The Model 4 has little enhancement compare to Model 2 and 3. The ESS ratio to TSS in Model 4 is bigger than the ESS ratio to TSS in Model 2 and 3.

The values "Sign." in "Coefficients" Table 5.16 are still 0.00. So, the estimate in column "B" can be asserted as true with a level 100%.

In this trail, R-square is 0.772. As a result, in this regression analysis the chosen independent variable Exit has more effect on effort compared to development tool and batch/online independent variables.

5.1.5. Regression Analysis with Module

In this regression analysis, functional size of small enhancements and module properties are selected as independent variable and effort is dependent model.

$$Effort = \text{function } (functional\ size\ in\ CFP, Module)$$

Table 5.17. Variables Entered/$Removed^a$.

Model	Variables Entered	Variables Removed	Method
1	Module, CFP*b		Enter

a.Dependent Variable: Effort

b.All requested variables entered.

In Table 5.20, the values "Sign." in "Coefficients" table are still 0.000. So, the estimate in column "B" can be asserted as true with a level 100%.

The R-square in "Model Summary" Table 5.18 is 0.786. The Model 5 has the best results so far. The Module property stands for small enhancement has effect on

Table 5.18. Model Summary.

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	0.887^a	0.786	0.781	3.917

a. Predictors: (Constant), Module, CFP

Table 5.19. $ANOVA^a$.

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	4798.261	2	2399.130	156.363	0.000*b
	Residual	1304.182	85	15.343		
	Total	6102.443	87			

a.Dependent Variable: Effort

b. Predictors: (Constant), Exit, CFP

Table 5.20. $Coefficients^a$.

Model		Unstandardized Coefficients		Standard Coeffi-cients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	-6.280	1.112		-5.648	0.000
	CFP	3.043	0.227	0.772	13.392	0.000
	Module	3.806	1.109	0.198	3.432	0.001

a.Dependent Variable: Effort

only one module or on more than one module. Development tool, Batch/Online small enhancement reachability property has effect on effort, but Module effect is more. This is an expected result. While developing small enhancement more than one module is considered and this causes to increase effort on development. Considering one module is easier than to consider more than one module. It requires having knowledge about other module(s), usability of entities related to other one(s).

5.1.6. Regression Analysis with Module and Exit

In this regression analysis, functional size of small enhancements, module and exit properties are selected as independent variable and effort is dependent model.

$$Effort = \text{function } (functional\ size\ in\ CFP, Module, Exit)$$

Table 5.21. Variables Entered/$Removed^a$.

Model	Variables Entered	Variables Removed	Method
1	Exit, Module, CFP^b		Enter

a.Dependent Variable: Effort

b.All requested variables entered.

Table 5.22. Model Summary.

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	0.893^a	0.798	0.790	3.835

a. Predictors: (Constant), Exit, Module, CFP

As shown in Table 5.22, the R-square in "Model Summary" is 0.798. The results are getting better when other independent variables are used together. Each independent variable has own effect on dependent variable effort. It is important to

Table 5.23. $ANOVA^a$.

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	4867.049	3	1622.350	110.311	0.000*b
	Residual	1235.394	84	14.707		
	Total	6102.443	87			

a.Dependent Variable: Effort

b. Predictors: (Constant), Exit, Module, CFP

Table 5.24. $Coefficients^a$.

Model		Unstandardized Coefficients		Standard Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	-7.968	1.339		-5.949	0.000
	CFP	3.470	0.297	0.881	11.666	0.000
	Module	3.555	1.092	0.185	3.256	0.002
	Exit	-4.275	1.977	-0.148	-2.163	0.033

a.Dependent Variable: Effort

use suitable relation among the independent variables to get better regression results.

5.1.7. Regression Analysis with Write and Module

In this analysis, functional size of small enhancements and Write properties are used in the regression analysis as independent variable and again "Effort" is dependent variable.

$$Effort = \text{function } (functional\ size\ in\ CFP, Write, Module)$$

Exit and *Write* properties are related to data movement. The border on the number of these data movements are defined and if the small enhancements have more *Exit* or *Write* data movement than the value of the border this property this property is associated with small enhancement.

Table 5.25. Variables Entered/*Removed*[a].

Model	Variables Entered	Variables Removed	Method
1	Module, Write, *CFP*[b]		Enter

a.Dependent Variable: Effort

b.All requested variables entered.

Table 5.26. Model Summary.

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	0.887[a]	0.787	0.780	3.930

a. Predictors: (Constant), Module, Write, CFP

Table 5.27. $ANOVA^a$.

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	4804.831	3	1601.610	103.679	0.000*b
	Residual	1297.613	84	15.448		
	Total	6102.443	87			

a.Dependent Variable: Effort

b. Predictors: (Constant), Module, Write, CFP

Table 5.28. $Coefficients^a$.

Model		Unstandardized Coefficients		Standard Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	-6.577	1.205		-5.458	0.000
	CFP	3.126	0.261	0.793	11.960	0.000
	Write	-0.932	1.429	-0.041	-0.652	0.516
	Module	3.916	1.125	0.204	3.480	0.001

a.Dependent Variable: Effort

Exit and Write properties are similar ones. So they are analyzed separately to observe which one has more effect on dependent variable effort with independent variable functional size. The results in Table 5.26, 5.27, 5.28 shows that Exit property (R-square is 0.798) has more effect than Write property (R-square is 0.787).

5.1.8. Regression Analysis with Multiple Variables

In this analysis, many properties of small enhancements are used in the regression analysis as independent variable and again "Effort" is dependent variable. The main stone of the productivity model is effort. So, it cannot be omitted from the regression.

$$Effort = \text{function } (functional\ size\ in\ CFP, Tool, Write, Exit, Module)$$

Table 5.29. Variables Entered/$Removed^a$.

Model	Variables Entered	Variables Removed	Method
1	Tool, Write, Exit, Module CFP^b		Enter

a.Dependent Variable: Effort

b.All requested variables entered.

Table 5.30. Model Summary.

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	0.899^a	0.809	0.797	3.770

a. Predictors: (Constant), Tool, Write, Exit, Module, CFP

Many independent variables are used in this trial. Each of the property has effect on dependent variable effort. And when they used together the best regression analysis results are gathered. R-square value is 0.809 as shown in Table A.1. The least Std. Error of Estimate is reached. It is adequate to say something to construct

Table 5.31. $ANOVA^a$.

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	4936.825	5	987.365	69.460	0.000*b
	Residual	1165.618	82	14.215		
	Total	6102.443	87			

a.Dependent Variable: Effort

b. Predictors: (Constant), Tool, Write, Exit, Module, CFP

Table 5.32. $Coefficients^a$.

Model		Unstandardized Coefficients		Standard Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	-8.589	1.394		-6.163	0.000
	CFP	3.410	0.323	0.865	10.564	0.000
	Write	-0.838	1.372	-0.037	-0.611	0.543
	Module	4.459	1.149	0.232	3.880	0.000
	Exit	-4.027	1.947	-0.139	-2.069	0.042
	Tool	1.823	0.861	0.109	2.118	0.037

a.Dependent Variable: Effort

the productivity model for small enhancements.

Table 5.33. R-square Results of Different Independent Variables.

	Module	**Exit**	**Online/Batch**	**Tool**	**Write**
CFP	0.782	0.772	0.761	0.759	0.757

The Table A.4 shows the R-square results of independent variables when they are used with independent variable functional size (CFP). The biggest R-square value belongs to group of functional size and module independent variables. However, in this table each independent variable has effect on dependent variable effort. So, regression analysis test is done by using all independent variables and better R-square value 0.809 is reached.

5.2. Unit Cost and Productivity Analysis

This research put in place the conditions that favor not only to reach the objective of a designing sound productivity models, but also to try to explain the maintenance costs using a number of cost drivers.

Table 5.34 shows that the average unit cost for all the data is 1.81 hours per CFP with a variance of 0.9 hours for 88 small enhancements.

Analysis of the unit cost from Table 5.34 shows that the lowest unit cost if for Adaptive maintenance (1,4), while corrective maintenance is almost the double (2,5). For the tools (PL/1 or Cool:Gen) it shows they have the same average than Batch and Online (1,6 and 2,0). When looking at the data it shows that the PL/1 is always associate with Batch and Cool:Gen with Online.

Table 5.34. Unit Cost per Variable.

Variables	**Average hours/CFPs**	**Variance**
All data	1.18	0.94
PL/1	1.6	1.05
Cool:Gen	2.0	0.73
Batch	1.6	1.05
Online	2.0	0.73
Adaptive	1.4	0.87
Corrective	2.5	0.79

6. CONCLUSION AND NEW RESEARCHES

When looking at the software life cycle, few researches have been done to study the improvement of software maintenance. There are not enough empirical data collected and analyzed for the maintenance phase of software especially for small change requests. Therefore empirical studies are to be held in the organizations in order to make them aware that software maintenance is also part of the software life cycle and should be considered together with software development.

Main scope of this research consists of small enhancements. 88 small enhancements are analyzed and measured using COSMIC measurement method [5] to answer it is possible to construct productivity model(s) for small enhancements and unit cost and productivity analysis takes part in the research to explain the maintenance costs using a number of cost drivers.

Steps of the proposed approach to construct productivity model are as follows:

(i) Identification of each small enhancement
(ii) Measurement of the size of each small enhancement with COSMIC measurement method
(iii) Determination quality of the documentation for each small enhancement with COSMIC verification method
(iv) Collection of effort for each small enhancement
(v) Collection of characteristics for each small enhancement
(vi) Determination of unit cost for each small enhancement
(vii) Constructing productivity model(s) for small enhancements

After all iterations about collecting characteristics are completed regression analysis is applied to small enhancements with different independent variables.

With this approach, it is showed that small enhancements can be measured by

using the COSMIC [5] measurement method. The regression analysis results show that it is possible to produce a productivity model with this sample using all independent variables (R-square of more than 0.75) of a specific large retail bank.

Regression analysis method was tried with different independent variables and dependent variable Effort. First analysis was with the CFP (Cosmic Function Point), R-square value was 0.757. In the second analysis, in addition to CFP, Batch/Online property was used. In this case R-square value (0.761) was better than the value in the first analysis. Tool property was used with CFP in the third trial. The R-squre value was better than first trial, worse than second trial. It was 0.759. That shows Batch/Online effect on Effort more than Tool property. Exit and CFP properties are used together in the fourth analysis and gathered better results than previous ones. The rest property Module was used with Exit and CFP, the experiment results were getting better when variety of independent variables increased. And via the last analysis with multiple independent variables, the best R-square value was reached, it was 0.809. The results show that it is possible to construct productivity model for small enhancements.

The productivity is 20% better using Cool:Gen tool Online, than the PL/1 tool in Batch mode (table 20). In this case it is not possible to determine if the increase of productivity is related to the tool or the mode (Online versus Batch), because they are link together. Productivity is also better (almost 60% increase) for adaptive maintenance than corrective maintenance.

There was good conditions to construct this productivity model: within an enterprise for one major application for a period of time, design, program and implement by the same person, documented from the maintainer, measure within a control environment and verify by an expert.

In the real world this type of situation is not common, especially if it is necessary to compare the productivity within different applications and enterprises. This is why new studies need to be done for different applications and different maintainers. It would permit to verify how unit cost would differ.

APPENDIX A: SMALL ENHANCEMENT MEASURES

The abbreviations used in table are Q: Quality, B: Batch, O: Online, M: Module. And C:Gen references for Cool:Gen

Table A.1. Small Enhancement Measures 1.

Num	E	R	W	X	CFP	Effort	Type	Ratio	Tool	Q	B	O	Write	Exit	M
1	1	1	1	1	4	4	M	1.00	PL/I	A	Y	N	0	0	1
2	1	1	1	1	4	4	M	1.00	PL/I	A	Y	N	0	0	1
3	1	1	1	1	4	4	M	1.00	PL/I	A	Y	N	0	0	1
4	1	2	2	1	6	22	N	3.67	PL/I	A	Y	N	1	0	1
5	1	2	2	1	6	20	N	3.33	PL/I	A	Y	N	1	0	1
6	1	5	1	1	8	18	N	2.25	PL/I	A	Y	N	0	0	0
7	1	1	0	1	3	4	M	1.33	PL/I	A	Y	N	0	0	0
8	1	1	1	1	4	4	M	1.00	C:Gen	A	N	Y	0	0	0
9	1	1	0	1	3	3	M	1.00	PL/I	A	Y	N	0	0	0
10	1	1	1	1	4	4	M	1.00	C:Gen	A	N	Y	0	0	0
11	1	1	0	1	3	4	M	1.33	C:Gen	A	N	Y	0	0	0
12	1	1	1	1	4	4	M	1.00	PL/I	A	Y	N	0	0	0
13	1	1	1	1	4	4	M	1.00	C:Gen	A	N	Y	0	0	0
14	1	1	1	1	4	4	M	1.00	C:Gen	A	N	Y	0	0	0
15	1	1	1	1	4	4	M	1.00	PL/I	A	Y	N	0	0	0
16	1	1	1	1	4	4	M	1.00	PL/I	A	Y	N	0	0	0
17	1	1	1	1	4	2	M	0.50	PL/I	A	Y	N	0	0	0
18	2	2	2	1	7	10	N	1.43	PL/I	A	Y	N	1	0	0
19	1	2	2	1	6	10	N	1.67	C:Gen	A	N	Y	1	0	0
20	1	2	2	1	6	10	N	1.67	C:Gen	A	N	Y	1	0	0
21	1	2	2	1	6	8	N	1.33	C:Gen	A	N	Y	1	0	0
22	1	1	1	1	4	3	M	0.75	PL/I	A	Y	N	0	0	0
23	1	1	1	1	4	3	M	0.75	C:Gen	A	N	Y	0	0	0
24	1	1	1	1	4	3	M	0.75	PL/I	A	Y	N	0	0	0
25	1	1	1	1	4	4	M	1.00	PL/I	A	Y	N	0	0	0

Table A.2. Small Enhancement Measures 2.

Num	E	R	W	X	CFP	Effort	Type	Ratio	Tool	Q	B	O	Write	Exit	M
26	1	1	1	1	4	4	M	1.00	C:Gen	A	N	Y	0	0	0
27	1	1	1	1	4	4	M	1.00	PL/I	A	Y	N	0	0	0
28	1	1	1	1	4	3	M	0.75	PL/I	A	Y	N	0	0	0
29	1	1	1	1	4	4	M	1.00	PL/I	A	Y	N	0	0	0
30	1	0	1	1	3	4	M	1.33	C:Gen	A	N	Y	0	0	1
31	1	6	5	1	13	35	N	2.69	C:Gen	A	N	Y	1	0	1
32	1	5	4	1	11	28	N	2.55	C:Gen	A	N	Y	1	0	1
33	1	1		1	3	1	M	0.33	PL/I	A	Y	N	0	0	0
34	1	1		1	3	1	M	0.33	PL/I	A	Y	N	0	0	0
35	1	3	1	1	6	14	N	2.33	PL/I	A	Y	N	0	0	1
36	1	2		1	4	12	M	3.00	PL/I	A	Y	N	0	0	1
37	1	3	1	1	6	10	N	1.67	C:Gen	A	N	Y	0	0	0
38	1	2	1	1	5	6	M	1.20	C:Gen	A	N	Y	0	0	0
39	1	1	1	1	4	4	M	1.00	PL/I	A	Y	N	0	0	0
40	1	1	1	1	4	4	M	1.00	PL/I	A	Y	N	0	0	0
41	1	1	1	1	4	4	M	1.00	PL/I	A	Y	N	0	0	0
42	1	1	1	1	4	4	M	1.00	PL/I	A	Y	N	0	0	0
43	1	1	1	1	4	4	M	1.00	PL/I	A	Y	N	0	0	0
44	1	1	1	1	4	4	M	1.00	PL/I	A	Y	N	0	0	0
45	1	1	1	1	4	4	M	1.00	C:Gen	A	N	Y	0	0	0
46	1	1	1	1	4	8	M	2.00	C:Gen	A	N	Y	0	0	0
47	1	5	2	1	9	16	N	1.78	PL/I	A	Y	N	1	0	1
48	1	6	2	1	10	26	N	2.60	PL/I	A	Y	N	1	0	1
49	1	2	2	1	6	12	N	2.00	PL/I	A	Y	N	1	0	1
50	1	2	2	1	6	13	N	2.17	PL/I	A	Y	N	1	0	1

Table A.3. Small Enhancement Measures 3.

Num	E	R	W	X	CFP	Effort	Type	Ratio	Tool	Q	B	O	Write	Exit	M
51	1	1		1	3	1	M	0.33	PL/I	A	Y	N	0	0	0
52	1	1	1	1	4	10	M	2.50	PL/I	A	Y	N	0	0	0
53	1	4	3	2	10	40	N	4.00	PL/I	A	Y	N	1	0	1
54	1	1		2	4	16	M	4.00	C:Gen	A	N	Y	0	0	0
55	1	2	1	3	7	16	N	2.29	C:Gen	A	N	Y	0	1	0
56	1	1	1	1	4	8	M	2.00	C:Gen	A	N	Y	0	0	0
57	1	1	1	1	4	8	M	2.00	C:Gen	A	N	Y	0	0	0
58	3	1	3	1	8	24	N	3.00	C:Gen	A	N	Y	1	0	0
59	2	3	1	2	8	22	N	2.75	C:Gen	A	N	Y	0	0	0
60	2	3	1	2	8	18	N	2.25	C:Gen	A	N	Y	0	0	0
61	2	3	1	2	8	19	N	2.38	C:Gen	A	N	Y	0	0	0
62	2	6		4	12	38	N	3.17	C:Gen	A	N	Y	0	1	1
63	1	2		2	5	20	M	4.00	PL/I	A	Y	N	0	0	1
64	1	3		2	6	24	N	4.00	PL/I	A	Y	N	0	0	1
65	1	3		2	6	22	N	3.67	PL/I	A	Y	N	0	0	1
66	1	1	1	2	5	8	M	1.60	C:Gen	A	N	Y	0	0	0
67	1	4	1	2	8	24	N	3.00	PL/I	A	Y	N	0	0	1
68	1	3	1	2	7	11	N	1.57	PL/I	A	Y	N	0	0	0
69	1	1	1	1	4	10	M	2.50	PL/I	A	Y	Y	0	0	0
70	1		1	1	3	8	M	2.67	PL/I	A	Y	Y	0	0	0
71	1	2		1	4	6	M	1.50	PL/I	A	Y	N	0	0	0
72	1	2	1	1	5	10	M	2.00	C:Gen	A	N	Y	0	0	1
73	1	1		1	3	8	M	2.67	C:Gen	A	N	Y	0	0	0
74	1	1		1	3	8	M	2.67	C:Gen	A	N	Y	0	0	0
75	1	1		2	4	6	M	1.50	PL/I	A	Y	N	0	0	0

Table A.4. Small Enhancement Measures 4.

Num	E	R	W	X	CFP	Effort	Type	Ratio	Tool	Q	B	O	Write	Exit	M
76	1	1		1	3	8	M	2.67	C:Gen	A	N	Y	0	0	0
77	1	1		1	3	8	M	2.67	C:Gen	A	N	Y	0	0	0
78	1	1		1	3	7	M	2.33	C:Gen	A	N	Y	0	0	0
79	1	1		2	4	8		2.00	C:Gen	A	N	Y	0	0	0
80	1	3	1	1	6	10		1.67	PL/I	A	Y	N	0	0	1
81	1	2		1	4	4		1.00	PL/I	A	Y	N	0	0	0
82	1	1		1	3	8		2.67	C:Gen	A	N	Y	0	0	0
83	1	1		1	3	8		2.67	C:Gen	A	N	Y	0	0	0
84	1	2		2	5	10		2.00	C:Gen	A	N	Y	0	0	0
85	1	1	1	1	4	8		2.00	C:Gen	A	N	Y	0	0	0
86	1	1	1	1	4	8		2.00	C:Gen	A	N	Y	0	0	0
87	1	1		2	4	3		0.75	PL/I	A	Y	N	0	0	0
88	1	2		2	5	5		1.00	PL/I	A	Y	N	0	0	0

APPENDIX B: SMALL ENHANCEMENTS WITH DEFINITION

B.1. Changing General Ledger Numbers of Retail Credits

There is a ledger number for each product code. If the product code is changed, ledger number must be changed.

```
Trigger: Product code change
Entry: Product Code
Read: General Ledger (1 Read)
Write: General Ledger (1Write)
Error Message: 1 Exit (General Ledger number is not defined related to product code)

1(E) + 1(R) + 1(W) + 1(X) = 4CFP
Quality: A
Effort Enh. = 4 hours
Data Group: General Ledger
```

B.2. Changing General Ledger Numbers of Retail Credits

There is a ledger number for each product code. If the product code is changed, ledger number must be changed.

```
Trigger: Product code change
Entry: Product Code
Read: General Ledger (1 Read)
Write: General Ledger (1Write)
Error Message: 1 Exit (General Ledger number is not defined related to product code)

1(E) + 1(R) + 1(W) + 1(X) = 4CFP
Quality: A
Effort Enh. = 4 hours
Data Group: General Ledger
```

B.3. Changing General Ledger Numbers of Overdue Credits

There is a ledger number for each product code. If the product code is changed, ledger number must be changed.

```
Trigger: Product code change
Entry: Product Code
Read: General Ledger (1 Read)
Write: General Ledger (1Write)
Error Message: 1 Exit (General Ledger number is not defined related to product code)
```

```
1(E) + 1(R) + 1(W) + 1(X) = 4CFP
Quality: A
Effort Enh. = 4 hours
Data Group: General Ledger
```

B.4. Calculation Effective Interest Rate(EIR) of Retail Credits

From financial report table, effective interest rate calculated on the day before processing date is read. Then, commission amount is read from retail credits log table and this amount is extracted from previous net investment amount. After that, with specified formula effective interest rate is calculated for processing date. Calculated effective interest rate is inserted into financial report table with new processing date and financial report log record is inserted into log table.

```
Trigger: Effective interest rate calculate request
Entry: Retail credit (1 Entry)
Read: Financial Report,  Retail Credit Log (2 read)
Write: Financial report master record, financial report log (2Write)
Exit: Error Messages( Financial report  record is not found, Retail credit log is not found)
     (1Exit)

1(E) + 2(R) +2(W) + 1(X) = 6CFP
Quality: A
Effort Enh. = 22 hours
Data Group: Retail credit, Retail Credit Log, Financial Report, Financial Report Log
```

B.5. Calculating Adjustment Amount of Retail Credits

Accrued interest and commission amount is read from accruement table. Then this amounts are extracted effective interest amount calculated on processing date. Calculated adjustment amount is inserted into financial report table with new processing date by updating processing day record and financial report log record is inserted into log table.

```
Trigger: Adjustment calculate request
Entry: Retail credit (1 Entry)
Read: Financial Report,  Retail Credit Accruement Information(2 read)
Write: Financial report master record, financial report log (2Write)
Exit: Error Messages(Financial report  record is not found, Retail credit accruement
    commission and interest amount is not found )(1Exit)

1(E) + 2(R) +2(W) + 1(X) = 6CFP
```

```
Quality: A
Effort Enh. = 20 hours
Data Group: Retail credit, Retail Credit Accruement Information, Financial Report, Financial
    Report Log
```

B.6. Daily Adjustment Accounting

Calculated adjustment amount is read from financial report table. And the calculated amount is transferred into income general ledger number. Income general number only stores base currency amount. So, if the currency code of calculated adjustment amount is not base currency, currency conversion link is called and it is converted to base currency amount.

```
Trigger: Calculated adjustment amount
Entry: Financial report (1)
Read: Adjustment amount, Income general ledger, Current general ledger, Product code,
    Currency code information (5)
Write: Accounting transaction (1)
Exit: Error Messages(Product code related to adjustment amount is not defined, Current
    general ledger is not found, Income general number is not found, Accounting error
    message, Currency code is not found) (1 Exit)
```

```
1(E) + 5(R) + 1(W) + 1(X) = 8 CFP
Quality: A
Effort: 18 hours
Data Group: Financial report, Account(product code is read from account table), Product
    Ledger Relation, Accounting Currency
```

B.7. Changing Commission and Income General Numbers of Amortization Batch

In previous Standard, commission and general numbers were constant. With new Standard, commission and general numbers are decided to be related to product code.

New product codes are read from product general ledger relation table.

```
Trigger: Daily Amortization batch schedule time
Entry: Amortization record
Read: Product related general ledger info
Write: -
Exit: Error Messages(Commission ledger is not defined in product related general ledger
    table, income ledger is not defined in product related general ledger table) (1 Exit)

1(E) + 1(R) + 1(X) = 3 CFP
```

```
Quality: A
Effort: 4 hours
Data Group:  Amortization, Product Ledger Relation
```

B.8. Transfer of Opening Commissions related to Commercial Credits with payment plan number to the income system

In new system opening commissions are needed to be transferred to the income system from amortization table. A conversion batch is needed for this transfer.

```
Trigger: Conversion request
Read: Opening Commission
Entry: Commission record
Write: Income info (1 Write)
Exit: Error Messages(Income record has already exist, Opening commission is not found) (1
    exit)

1(E) +1(R) +  1 (W) + 1(X) = 4CFP
Quality: A
Effort: 4 hours (Development tool is different, it is developed in coolgen)
Data group:  Amortization record, Income info, Opening Commission
```

B.9. Transfer of Opening Commissions related to Commercial Credits with no payment plan number to the income system (PL/I batch is used)

In new system opening commissions are needed to be transferred to the income system from amortization table. A conversion batch is needed for this transfer.

```
Trigger: Conversion request
Read: (No read)
Entry: Commission record
Write: Income info (1 Write)
Exit: Income record has already exist (1 exit)

1(E) + 1 (W) + 1(X) = 3CFP
Quality: A
Effort: 3 hours (Development tool is different, it is developed in PL/I tool)
Data group:  Amortization record, Income info
```

B.10. Cancel amortization of opening commissions related to commercial credits with payment plan number to the income system

```
Trigger: F5
Entry: Opening Commission record
Read: Reading commission record from amortization table.
Write: Delete commission record from amortization table.
Exit: Opening commission record is not found (1Exit)

1(E) + 1(R) + 1(W) + 1(X) = 4CFP
Quality: A
Effort: 4 hours
Data group: Opening Commission record, Amortization record
```

B.11. Transfer of Periodic Commissions related to Commercial Credits with payment plan number to the income system

In new system periodic commissions are needed to be transferred to the income system from amortization table. A conversion batch is needed for this transfer.

```
Trigger: Conversion request
Read: (No read)
Entry: Commission record
Write: Income info (1 Write)
Exit: Income record has already exist (1 exit)

1(E) + 1 (W) + 1(X) = 3CFP
Quality: A
Effort: 4 hours (Development tool is different, it is developed in PL/I tool)
Data group: Amortization record, Income info
```

B.12. Cancel amortization of periodic commissions related to commercial credits with payment plan number to the income system

```
Trigger: F5
Entry: Periodic Commission record
Read: Reading periodic commission record from amortization table.
Write: Delete commission record from amortization table.
Exit: Periodic commission record is not found (1Exit)

1(E) + 1(R) + 1(W) + 1(X) = 4CFP
Quality: A
Effort: 4 hours
Data group: Periodic Commission record, Amortization record
```

B.13. Transfer of Opening Commissions related to Retail Credits with payment plan number to the income system

In new system opening commissions of retail credits are needed to be transferred to the income system from amortization table. A conversion batch is needed for this transfer.

```
Trigger: Conversion request
Read: Opening Commission
Entry: Commission record
Write: Income info (1 Write)
Exit: Error Messages(Income record has already exist, Opening commission is not valid)(1
    exit)

1(E) + 1(R) + 1 (W) + 1(X) = 4CFP
Quality: A
Effort: 4 hours (Development tool is different, it is developed in coolgen)
Data group:  Amortization record, Income info, Opening Commission
```

B.14. Cancel amortization of opening commissions related to retail credits with payment plan number to the income system

```
Trigger: F5
Entry: Opening Commission record
Read: Reading opening commission record from amortization table.
Write: Delete commission record from amortization table.
Exit: Opening commission record is not found (1Exit)

1(E) + 1(R) + 1(W) + 1(X) = 4CFP
Quality: A
Effort: 4 hours
Data group:  Opening Commission record, Amortization record
```

B.15. Transfer of Periodic Commissions related to Retail Credits with payment plan to the income system

In new system periodic commissions of retail credits with payment plan are needed to be transferred to the income system from amortization table. A conversion batch is needed for this transfer.

```
Trigger: Conversion request
Read: Periodic commission
Entry: Commission record
Write: Income info (1 Write)
```

```
Exit: Error Messages( Income record has already exist, Periodic commission is not valid) (1
    exit)

1(E) + 1(R) + 1 (W) + 1(X) = 4CFP
Quality: A
Effort: 4 hours (Development tool is different, it is developed in coolgen)
Data group:  Amortization record, Income info, Periodic Commission
```

B.16. Cancel amortization of periodic commissions related to commercial retail income system

```
Trigger: F5
Entry: Periodic Commission record
Read: Reading periodic commission record from amortization table.
Write: Delete commission record from amortization table.
Exit: Periodic commission record is not found (1Exit)
```

```
1(E) + 1(R) + 1(W) + 1(X) = 4CFP
Quality: A
Effort: 4 hours
Data group:  Periodic Commission record, Amortization record
```

B.17. Commercial Credits Commission Rediscount Create

In the old system, there is no rediscount system for commission of commercial credit. So, in a table commission records for rediscount is created.

```
Trigger: Commission record accrue
Entry: Commission record
Read: Commission Information to check validation
Write: Rediscount info
Exit: Error Messages(Rediscount info already exit, Commission record is not valid)

1(E) + 1(R) + 1(W) + 1(X) = 4CFP
Quality: A
Effort: 2 hours
Data group: Commission record, Rediscount info
```

B.18. Commercial Credits Commission Rediscount Function

Every day, rediscount of record is calculated by dividing the commission to rediscount time till the end of commission collection. The amount is transferred to the income general number by using parametric accounting link.

```
Trigger: Daily rediscount batch schedule time
```

```
Entry: Rediscount info, Commission
Read: Rediscount General ledger related to product code, Income ledger related to product
Write: Rediscount record update, Accounting transaction (2 Write)
Exit: Error Messages(Accounting error, Rediscount General ledger related to product code is
    not found, Income ledger related to product is not found, Rediscount record update error
    )(1 Exit)

2(E) + 2(R) + 2 (W) + 1 (X) = 7 CFP
Quality: A
Effort: 10 hours
Data group: Rediscount info, Rediscount General ledger related to product code, Income
    ledger related to product, Commission
```

B.19. Calling IFRS general link to calculate EIR, Adjustment and Effective amount in Commercial Credits Investment Rediscount

To call ifrs general link, income record is read and investment rediscount record of commercial credits is read from rediscount table. Financial report record is updated and financial report log record inserted.

```
Trigger: Investment Rediscount Creation
Entry: Investment Rediscount
Read:  Rediscount Information, Income record
Write: Financial report, Financial Report Log (2 Write)
Exit: Error Messages(Rediscount information is not valid, Income record is not found)(1 Exit
    )

1(E) + 2(R) + 2 (W) + 1(X) = 6 CFP
Quality: A
Effort: 10 hours
Data group: Investment Rediscount info, Income info, Financial report, Financial Report Log
```

B.20. Calling IFRS general link to calculate EIR, Adjustment and Effective amount in Retail Credits Investment Rediscount

To call ifrs general link, income record is read and investment rediscount record of retail credits is read from rediscount table. Financial report record is updated and financial report log record inserted.

```
Trigger: Investment Rediscount Creation for Retail Credits
Entry: Investment Rediscount
Read:  Rediscount Information of Retail Credits, Income record
Write: Financial report, Financial Report Log (2 Write)
Exit: Error Messages(Rediscount information is not valid, Income record is not found)(1 Exit
    )
```

```
1(E) + 2(R) + 2 (W) + 1(X) = 6 CFP
Quality: A
Effort: 10 hours
Data group: Investment Rediscount info, Income info, Financial report, Financial Report Log
```

B.21. Calling IFRS general link to calculate EIR, Adjustment and Effective amount in Retail Credits Commission Rediscount

To call ifrs general link, income record is read and commission rediscount record of retail credits is read from rediscount table. Financial report record is updated and financial report log record inserted.

```
Trigger: Commission Rediscount Creation for Retail Credits
Entry: Commission Rediscount
Read:  Commission rediscount Information of Retail Credits, Income record
Write: Financial report, Financial Report Log (2 Write)
Exit: Error Messages(Commission rediscount information is not valid, Income record is not
    found)(1 Exit)

1(E) + 2(R) + 2 (W) + 1(X) = 6 CFP
Quality: A
Effort: 8 hours
Data group: Commission Rediscount info, Income info, Financial report, Financial Report Log
```

B.22. Cancel of calling amortization create from periodic commission collection of retail credits

Amortization create link is omitted from the program. So, the read operations to call the link is deleted from program.

```
Trigger: Cancel request
Entry: Periodic Commission of Retail Credit(1Entry)
Read: Periodic commission to check it is valid (1 Read)
Write: Amortization record (1 Write)
Exit: Error Messages(Amortization record already exists, Periodic commission record is not
    valid) (1 Exit)

1(E) + 1(R) + 1 (W) + 1(X) = 4 CFP
Quality: A
Effort: 3 hours
Data group: Periodic Commission,  Amortization
```

B.23. Cancel of calling amortization create from opening commission collection of retail credits

Amortization create link is omitted from the program. So, the read operations to call the link is deleted from program.

```
Trigger: Cancel request
Entry: Opening Commission (1Entry)
Read: Opening commission to check it is valid (1 Read)
Write: Amortization record (1 Write)
Exit: Error Messages(Amortization record already exists, Opening commission record is not
    valid) (1 Exit)

1(E) + 1(R) + 1 (W) + 1(X) = 4 CFP
Quality: A
Effort: 3 hours
Data group: Opening Commission, Amortization
```

B.24. Cancel of calling amortization create from periodic commission collection of commercial credits

Amortization create link is omitted from the program. So, the read operations to call the link is deleted from program.

```
Trigger: Cancel request
Entry: Periodic Commission of Commercial Credit (1Entry)
Read: Periodic commission to check it is valid (1 Read)
Write: Amortization record (1 Write)
Exit: Error Messages(Amortization record already exists, Periodic commission record is not
    valid )(1 Exit)

1(E) + 1(R) + 1 (W) + 1(X) = 4 CFP
Quality: A
Effort: 3 hours
Data group: Periodic Commission,  Amortization
```

B.25. Calling income system link from periodic commission collection of retail credits

In new system periodic commissions are needed to be transferred in the income system from amortization table. So, when a periodic commission of a retail credit is collected from the customer, new records are created in the income table.

```
Trigger: Periodic commission collection
Read: Periodic commission read to check it is valid or not
Entry: Commission record
Write: Income info (1 Write)
Exit: Error Messages( Income record has already exist, Periodic commission record is not
    valid) (1 exit)

1(E) + 1(R) + 1 (W) + 1(X) = 4CFP
Quality: A
Effort: 4 hours
Data group:  Commission record, Income info
```

B.26. Calling income system link from opening commission collection of retail credits

In new system opening commissions are needed to be transferred in the income system from amortization table. So, when a opening commission of a retail credit is collected from the customer, new records are created in the income table.

```
Trigger: Opening commission collection
Read: Opening commission to check validation
Entry: Commission record
Write: Income info (1 Write)
Exit: Error Messages( Income record has already exist, Opening commission information is not
     valid) (1 exit)

1(E) + 1(R) + 1 (W) + 1(X) = 4CFP
Quality: A
Effort: 4 hours
Data group:  Commission record, Income info
```

B.27. Calling income system link from periodic commission collection of commercial credit

In new system periodic commissions are needed to be transferred in the income system from amortization table. So, when a periodic commission of a commercial credit is collected from the customer, new records are created in the income table.

```
Trigger: Periodic commission collection
Read: Periodic commission read to check it is valid or not
Entry: Commission record
Write: Income info (1 Write)
Exit: Error Messages(  Income record has already exist, Periodic commission record is not
    valid) (1 exit)

1(E) + 1(R) + 1 (W) + 1(X) = 4CFP
```

```
Quality: A
Effort: 4 hours
Data group:  Commission record, Income info
```

B.28. Cancel of calling amortization create from accrual system

Amortization create link is omitted from the program. So, the read operations to call the link is deleted from program.

```
Trigger: Cancel request
Entry: Accruement Information (1Entry)
Read: Accruement information is read to check it is valid (1 Read)
Write: Amortization record (1 Write)
Exit: Error Messages( Amortization record already exists, Accruement information is not
    valid) (1 Exit)

1(E) + 1(R) + 1 (W) + 1(X) = 4 CFP
Quality: A
Effort: 3 hours
Data group: Accruement Information,  Amortization
```

B.29. Calling commercial credits commission rediscount create link from accrual system

In new system for commercial credits commissions are needed to be stored in rediscount table for rediscount when it is accrued.

```
Trigger: Commission accrual
Read: Commission record check
Entry: Commission record
Write: Rediscount (1 Write)
Exit: Error Messages(Rediscount record has already exist , Commission record is not valid)(1
     exit)

1(E) + 1(R) + 1 (W) + 1(X) = 4CFP
Quality: A
Effort: 4 hours
Data group:  Commission record, Rediscount info
```

B.30. Connection between income system and amortization system for opening commission records of commercial credits with no payment plan

A collected commission can be returned to the customer. When opening commission collection of commercial credits with no payment plan, the commission is inserted into amortization table and also into income table. So there is needed a relation between amortization record and income record. A new interval table is designed for this connection.

```
Trigger: Opening commission collection
Read: (No read)
Entry: Commission record
Write: Amortization and income system relation (1 Write)
Exit: The amortization and income relation record has already exist (1 exit)

1(E) + 1 (W) + 1(X) = 3CFP
Quality: A
Effort: 4 hours (Development tool is different, it is developed in coolgen)
Data group: Commission record, Amortization and income system relation
```

B.31. Extra Commission Collection Facility for Commercial Credits

For commercial credits, extra commission collection facility is needed. According to entered account number and unit number, account information is read from general account table. Then, customer information is read from customer table. Account record has a field containing product code, with this code product table is read to get the name of the product related to entered account and unit number. After that, commission account related to entered main account is read from database. After the commission account is read, general information like amount and currency code of this account is read from main account table. After read operations are completed, the user enters how much commission is collected to the system. Then, updating main account table, account movements are registered and accounting operations are done. For accounting, general ledger related to product is read from ledger product relation table. Then, amortization record is created for extra commission of commercial credits. For this create operation amortization and income relation is read from database. And the last step, special transaction log is created for commercial credit.

```
Trigger: Extra commission collection request
Read: Main account, Related commission account, Commission account information, Product
    definition read, General ledger, Amortization and income relation (6 Read)
Entry: Account number and unit number of main account (1 Entry)
Write: Account update, Accounting movement registration, Account movement registration,
    Retail credit log, amortization record (5 write)
Exit: Error Messages(Account balance amount is not enough, accounting error messages,
    account movement registration error message, related commission account is not found
    related to main account, product definition is not found, Amortization and income
    relation is not found )(1 Exit)

1(E) + 6(R) + 5(W) + 1(X) = 13CFP
Quality: A
Effort: 35 hours
Data group: Main account, commission account, Product, General Ledger Information, Account
    movement record, Accounting record, Amortization income relation, Amortization record
```

B.32. Extra Commission Collection Facility for Retail Credits

For retail credits, extra commission collection facility is needed. According to entered account number and unit number, account information is read from general account table. Then, customer information is read from customer table. Account record has a field containing product code, with this code product table is read to get the name of the product related to entered account and unit number. After that, commission account related to entered main account is read from database. After the commission account is read, general information like amount and currency code of this account is read from main account table. After read operations are completed, the user enters how much commission is collected to the system. Then, updating main account table, account movements are registered and accounting operations are done. For accounting, general ledger related to product is read from ledger product relation table. And the last step, special transaction log is created for retail credit.

```
Trigger: Extra commission collection request
Read: Main account, Related commission account, Commission account information, Product
    definition read, General ledger (5 Read)
Entry: Account number and unit number of main account (1 Entry)
Write: Account update, Accounting movement registration, Account movement registration,
    Retail credit log (4 write)
Exit: Error Messages(Account balance amount is not enough, accounting error messages,
    account movement registration error message, related commission account is not found
    related to main account, product definition is not found) (1 Exit)

1(E) + 5(R) + 4(W) + 1(X) = 11CFP
Quality: A
```

```
Effort: 28 hours
Data group: Main account, commission account, Product, General Ledger Information, Account
    movement record, Accounting record
```

B.33. Discarding new Transactions from the First Level of Accounting Unload

The new defined transactions are omitted from first accounting unload. Because accounting using these transaction codes are generated after first unload of accounting unload. These transactions are needed to add into the second unload of accounting.

```
Trigger: New transaction definition
Read: Accounting movements
Entry: Transaction
Write: No write to the database
Exit: Accounting list without these transactions

1(E) + 1(R) + 1(X) = 3CFP
Quality: A
Data group: Transaction, Accounting record
Effort: 1 hour
```

B.34. Adding new Transactions to the Second Level of Accounting Unload

The new defined transactions are added to the second accounting unload. Because accounting using these transaction codes are generated before second unload of accounting unload.

```
Trigger: New transaction definition
Read: Accounting movements
Entry: Transaction
Write: No write to the database
Exit: Accounting list without these transactions

1(E) + 1(R) + 1(X) = 3CFP
Quality: A
Data group: Transaction, Accounting record
Effort: 1 hour
```

B.35. Loading accounting from excel file to the system without using general parametric accounting link

Accounting transactions are written into excel file and this file is loaded to the system. Account number, entered general number and unit number are controlled from database whether they are defined or not. Then accounting operations are done. If account is entered to the file account based movements are recorded to the database.

```
Trigger: Request job submitting registration
Entry: Accounting transactions
Read: Account control, General Ledger Number Control, Unit control
Write: Accounting, Account based movements
Exit: Error Messages(Accounting error message, Account movement write error)

1(E) + 3(R) + 1(W) + 1(X) = 6CFP
Quality: A
Effort: 14 hours
Data group: Accounting transaction, General Ledger, Unit, Account
```

B.36. Comparison of accounting movement and account movement to control trial balance (in two ways: from accounting to account movement and from account movement to accounting movement)

This comparison facility is needed to find the disparity between accounting and account movement to control trial balance.

```
Entry: Control request of trial balance (1 Entry)
Read: Accounting record, Account movement record (2 Read)
Write: List of disparity records (Accounting and account record) (Write is on the list not
    database)
Exit: Error Messages(No accounting record with the given process date, no account record
    with the given process date) (1 Exit)

1(E) + 2(R) + 1(X)= 4 CFP
Quality: A
Effort: 12 hours
Data group: Accounting record, Account record
```

B.37. Facility to insert general numbers according to product

To provide relation between general numbers with product code, a new facility is developed. With this facility general ledger related to product code is inserted and

this records are inserted into same table, if there is a relation, it is updated. Entered general ledger is controlled whether it is valid or not from general ledger table, and product codes are listed on the screen from product table.

```
Entry: Relation definition request (1 Entry)
Read: Product code, General Ledger, General Ledger and Product Relation (3 Read)
Write: General Ledger and Product Relation
Exit: Entered general ledger is not defined
```

```
1(E) + 3(R) + 1(W) + 1(X) = 6 CFP
Quality: A
Effort: 10 hours
Data group: Product code, General Ledger, General Ledger and Product Relation
```

B.38. Facility to insert old product codes and new products codes into product change parameter table

If a credit has overdue accounts, the related code of this credit account is changed with defined new product code. To trace the change, old product code and new product code is together stored in a log table.

```
Entry: Product code change
Read: Old Product code, New Product code
Write: Old and New Product Code
Exit: New product code is not defined

1(E) + 2(R) +1(W) + 1(X) = 5 CFP
Quality: A
Effort: 6 hours
Data Group: Old Product code, New Product code
```

B.39. NBR Opening Commission Report of Retail Credits

Opening commission records are read from table and listed for control of NBR. If opening commission is not found an error message is given to the user.

```
Entry: NBR request
Read: Opening commission of retail credits
Write: Formatted Opening commission to the list
Exit: No opening commission

1(E) + 1(R) + 1(W) + 1(X) = 4 CFP
Quality: A
Effort: 4 hours
Data Group: Opening commission
```

B.40. NBR Periodic Commission Report of Retail Credits

Periodic commission records are read from table and listed for control of NBR. If periodic commission of retail credits is not found an error message is given to the user.

```
Entry: NBR request
Read: Periodic  commission of retail credits
Write: Formatted Periodic commission to the list
Exit: No periodic commission

1(E) + 1(R) + 1(W) + 1(X) = 4 CFP
Quality: A
Effort: 4 hours
Data Group: Periodic commission
```

B.41. NBR Extra Commission Report of Retail Credits

Extra commission records are read from table and listed for control of NBR. If extra commission of retail credits is not found an error message is given to the user.

```
Entry: NBR request
Read: Extra commission
Write: Formatted Extra commission to the list
Exit: No extra commission

1(E) + 1(R) + 1(W) + 1(X) = 4 CFP
Quality: A
Effort: 4 hours
Data Group: Extra commission
```

B.42. NBR Opening Commission Report of Commercial Credits

Opening commission records are read from table and listed for control of NBR. If opening commission is not found an error message is given to the user.

```
Entry: NBR request
Read: Opening commission of commercial credits
Write: Formatted Opening commission to the list
Exit: No opening commission

1(E) + 1(R) + 1(W) + 1(X) = 4 CFP
Quality: A
Effort: 4 hours
Data Group: Opening commission
```

B.43. NBR Periodic Commission Report of Commercial Credits

Periodic commission records are read from table and listed for control of NBR. If periodic commission of commercial credits is not found an error message is given to the user.

```
Entry: NBR request
Read: Periodic  commission of commercial credits
Write: Formatted Periodic commission to the list
Exit: No periodic commission

1(E) + 1(R) + 1(W) + 1(X) = 4 CFP
Quality: A
Effort: 4 hours
Data Group: Periodic commission
```

B.44. NBR Extra Commission Report of Commercial Credits

Extra commission records are read from table and listed for control of NBR. If extra commission of commercial credits is not found an error message is given to the user.

```
Entry: NBR request
Read: Extra commission
Write: Formatted Extra commission to the list
Exit: No extra commission

1(E) + 1(R) + 1(W) + 1(X) = 4 CFP
Quality: A
Effort: 4 hours
Data Group: Extra commission
```

B.45. Facility to relate income codes with product codes

To list the related income codes in the commercial and retail credits screen for extra commission collect, income codes are stored id a relation table with product codes. So, if the credit is commercial, according to selected product code, corresponding income codes are listed.

```
Entry: Income code definition
Read: Product code
Write: Income code and product code relation
Exit: Income code and product code relation is defined before.
```

```
1(E) + 1(R) +1(W) + 1(X) = 4 CFP
Quality: A
Effort: 4 hours
Data Group: Product code, income code, Income code and Product code relation
```

B.46. List of income codes related to product codes

In extra commission screens (commercial credit extra commission and retail credit extra commission), related income codes to the selected product codes are listed. If there is no relation, error message is given.

```
Entry: Extra commission collection request
Read: Product code and income code relation
Write: List of relation in opening screen
Exit: No relation is found

1(E) + 1(R) +1(W) + 1(X) = 4 CFP
Quality: A
Effort: 8 hours (development tool is coolgen)
Data Group: Product code and income code relation
```

B.47. Impair flag update for risky credits

In an excel file for risky credits, impair flag is set or cleared. In excel file row, customer number, credit account number, unit number of the account belong to, impair flag and credit type information is written. So, in impair flag update program all of these information is checked.

```
Entry: Impair flag update request
Read: Main Account table, Commercial credit account table, Retail credit account table, Unit
     table, Customer information table (5 Read)
Write: Update of impair flag for retail credits, Update of impair flag for commercial
    credits (2 Write)
Exit: Error Messages(Account is not defined, the entered account is does not belong to
    retail credits, the entered account does not belong to commercial credits, the entered
    unit number is not defined, customer is not defined in the system )(1 Exit)
```

```
1(E) + 5(R) +2(W) + 1(X) = 9 CFP
Quality: A
Effort: 16 hours
Data Group: Main Account, Commercial credit account, Retail credit account, Unit, Customer
```

B.48. Calculating Effective Interest Amount of Commercial Credits

From commercial credit account table, the entered account information of the credit is read. Then from general account table, currency rate and product code of the credit account is read. From payment plan sequence table, last sequence number of payment plan of the credit and base money amount is read. Then, process main code, sub code, process reference and process sequence information is read from credit usage table. Then, cost amount is calculated with interest and commission amount read from commission loan table. From financial report table effective interest rate is read, and then via formula effective amount of retail credits is calculated. After that, calculated effective amount is inserted into financial report table with new processing date and financial report log record is inserted into log table.

```
Trigger: Effective interest amount calculate request
Entry: Commercial credit
Read: Commercial credit account, General account, Commercial Credit Log, Payment Plan
    Sequence, Credit Usage, Commission Loan (6 read)
Write: Financial report master record, financial report log (2Write)
Exit: Error Messages(Payment plan sequence is not defined, effective interest rate is not
    found, general account information is not found, the entered commercial credit is not
    defined in the system, commission loan is not found )(1 exit).

1(E) + 6(R) +2(W) + 1(X) = 10CFP
Quality: A
Effort Enh. = 26 hours
Data Group: Commercial credit account, General account, Payment plan Sequence, Credit Usage,
     Commission Loan , Financial Report
```

B.49. Calculation Effective Interest Rate(EIR) of Commercial Credits

From financial report table, effective interest rate calculated on the day before processing date is read. Then, commission amount is read from commission loan table and this amount is extracted from previous net investment amount. After that, with specified formula effective interest rate is calculated for processing date. Calculated effective interest rate is inserted into financial report table with new processing date and financial report log record is inserted into log table.

```
Trigger: Effective interest rate calculate request
Entry: Commercial credit (1 Entry)
Read: Financial Report,  Commission Loan(2 read)
Write: Financial report master record, financial report log (2Write)
```

```
Exit: Error Messages( Financial report  record is not found, commercial Loan information is
    not found) (1Exit)

1(E) + 2(R) +2(W) + 1(X) = 6CFP
Quality: A
Effort Enh. = 12 hours
Data Group: Commercial credit, Commercial Credit Log, Financial Report, Financial Report Log
```

B.50. Calculating Adjustment Amount of Commercial Credits

Accrued interest and commission amount is read from commercial credit accruement table. Then this amounts are extracted effective interest amount calculated on processing date. Calculated adjustment amount is inserted into financial report table with new processing date by updating processing day record and financial report log record is inserted into log table.

```
Trigger: Adjustment calculate request
Entry: Commercial credit (1 Entry)
Read: Financial Report,  Commercial Credit Accruement Information(2 read)
Write: Financial report master record, financial report log (2Write)
Exit: Error Messages( Financial report  record is not found, Commercial credit accruement
    commission and interest amount is not found )(1Exit)
```

```
1(E) + 2(R) +2(W) + 1(X) = 6CFP
Quality: A
Effort Enh. = 13 hours
Data Group: Commercial credit, Commercial Credit Accruement Information, Financial Report,
    Financial Report Log
```

B.51. Customer Account Report General Ledger Number Change

In customer account report, the field which includes investment general number is needed to change after IFRS system. This general number was constant, it is changed with a dynamic general ledger number. According to product code of the account, general number is read from the system.

```
Entry: Account
Read: General Ledger
Write: No Write for this change
Exit: Error Messages

1(E) + 1(R) + 1(X) = 3CFP
Effort: 1 hour
Quality: A
```

```
Data Group: Account, General Ledger
```

B.52. Recover Table Lock Escalation Problem

There was a job and in this job 5 different batches are running. Each of the batch is inserting and updating rediscount table every day and one of the batches is also updating account table. In each step, the same insert and update operation is repeated and other jobs in the bank get the error lock escalation of rediscount and account table. According to this error, instead of inserting and updating rediscount table in all steps, the records are written in a file and at the end of these steps, this file is loaded into rediscount table. And, the code block which is updating account table is closed because it is not really needed. Consequently, lock escalation error is recovered.

```
First functional process: Writing rediscount information into work file
Trigger: Batch
Read: Rediscount information
Exit: DSN File
Second functional process: Loading work file into rediscount table
Trigger: Batch
Write: File is loaded into rediscount table
```

```
1(E) + 1(R) + 1(W) + 1(X) = 4 CFP
Effort:10 hours
Data Quality: A
Data Group: Rediscount Information
```

B.53. Income Delivery Operations

Request: The income products of the customer is needed to deliver from one unit to another new unit. Firstly, according to sequence number(for delivery operation a sequence number is defined) the products which are to be delivered are selected with account number. Then, according to product id and unit num (these are get from product table) of the product, from daily income table, from cumm-income table and from income-return-info table the records which are related to unit num and product id of the product to be delivered are selected. After that, these records are updated with new unit number and new account number.

```
Trigger: Sequence Number entry
Entry: Sequence Number
```

```
Read: Product read, Daily Income Info Read, Cumulative Income Info Read, Income Return Info
    Read(4 Read)
Write: Update daily income info, update cumm income info and update income return info (3
    Write)
Exit: 1 Error Messages, 1 Report for which product is transferred

1(E) + 4(R) + 3(W) + 2(X) = 10 CFP
Effort: 40 hours
Data Quality: A
Data Group: Sequence Number, Product, Daily Income Info, Cumulative Income Info, Income
    Return Info
```

B.54. Pricing Service List

According to entered service criteria the pricing service codes which are defined in the system will be listed on the screen. From service, service codes are read.

```
Trigger: Service List Request
Entry: Service code(1 entry)
Read: Pricing Service codes (1 read)
Write: No write
Exit: Service code is not found, Service code list (2 exit)
```

```
1(E) + 1(R) + 2(X) = 4 CFP
Quality: A
Effort Enh. = 16 hours (developed in coolgen)
Data Group: Pricing Service
```

B.55. Service Definition Operation

On the screen firstly enter pricing service code is entered. If it is defined in the system, name of it is written on the screen, else space is written (no error message is returned back).

On the screen when "Pricing Service Group" is clicked all names of pricing service groups are listed and one of them is picked up. Then the relation between Pricing service and pricing service group is inserted into database.

```
Entry: Pricing Service Code
Read: Pricing Service, Pricing Service Group
Write: Pricing Service code related with Pricing Service Group
Exit: Pricing Service, Pricing Service Group, Pricing Service Error Messages

1(E) + 2(R) + 1(W) + 3(X) = 7 CFP
```

```
Quality: A
Effort Enh. = 16 hours
Data Group: Pricing Service, Pricing Service Group
```

B.56. Record Priority Screen

If more than one service code is related to one operation code, priority record list will be renewed and user will be warned to select record with priority.

```
Entry: Operation Code
Read: Pricing Service related to Operation code
Write: Pricing Service relation with Operation Code
Exit: Error Messages(Pricing service code is not defined)

1(E) + 1(R) + 1(W) + 1(X) = 4 CFP
Quality: A
Effort: 8 hours
Data Group: Operation Code, Pricing Service
```

B.57. Pricing Service code Product Code Relation

For pricing service code, to which product a special definition will be done is selected among choices in the list. There are two parameters, one of them includes products and other one includes relation with products in the list and products in the bank.

```
Entry: Pricing Service
Read: Product Code
Write: Pricing Service and Product code relation
Exit: Error message

1(E) + 1(R)+ 1(W) + 1(X) = 4 CFP
Quality: A
Effort: 8 hours
Data Group: Product, Pricing Service
```

B.58. Service Pricing

Prices for each existing pricing service codes is defined . There are 3 steps to define prices in the system. These are, defining standard prices, modifying standard prices on the basis of branch of industry and modifying personnel prices, defining reference

prices.

When pricing service code is clicked pricing service list is opened and a service is selected. There are 3 types of pricing criteria (standard, branch of industry, reference) on the service screen. According to selected criteria, after "List" button is clicked price list is opened on the list. When a price is selected from the list, modify button is enabled and according to selected price type one of these there window is opened: standard price definition, business of industry definition, reference definition (These windows are measured later). When save button is clicked changes are saved into the system.

```
Entry: Standard Price, Business of industry, Reference Definition
Read: Pricing Service
Write: Standard Price, Business of industry, Reference Definition
Exit: Error Message
```

```
3(E) + 1(R) + 3(R) + 1(X) = 8 CFP
Quality: A
Effort: 24 hours
Data Group: Pricing Service, Standard Price, Business of industry, Reference Price
```

B.59. Business of Industry Price Detail Entry

Attributes of business of industry record is read and written on the screen. Defined standard price details are read from database and listed on the screen . There are 4 different enterable price fields on the screen: charge amount, commission rate, minimum commission amount, and maximum commission amount. When a standard price is selected from the list, if there is a defined business of industry detail record, "update" and "delete" buttons are enabled else "add" button is enabled.

```
Entry: Business of Industry Price Code, Business of Industry Price record Detail
Read: Business of Industry Price record, Standard Price Detail record, Business Industry
    Price Detail
Write: Business of Industry Price record Detail(Insert/Update)
Exit: Error messages, Standard Price Detail List

2(E) + 3(R) + 1(W) + 2(X) = 8 CFP
Quality: A
Effort: 22 hours
Data Group: Business of Industry Price, Standard Price Details, Business Industry Price
    Detail
```

B.60. Reference Price Detail Entry

Reference price code in entered. Defined standard price details are read from database and listed on the screen. There are 4 different enterable price fields on the screen: charge amount, commission rate, minimum commission amount, and maximum commission amount. When a business of industry price is selected from the list, if there is a defined reference price detail record, “update” and “delete” buttons are enabled else “add” button is enabled.

```
Entry: Reference Price Code, Reference Price record Detail
Read: Reference Price Code, Reference Price Detail record, Business of industry detail
Write: Reference Price Detail(Insert/Update)
Exit: Error messages, Standard Price Detail List
```

```
2(E) + 3(R) + 1(W) + 2(X) = 8 CFP
Quality: A
Effort: 18 hours
Data Group: Reference Price, Business of industry detail, Reference Price Detail
```

B.61. Special Price Detail Entry

Special price code in entered. Defined reference price details are read from database and listed on the screen. There are 4 different enterable price fields on the screen: charge amount, commission rate, minimum commission amount, and maximum commission amount. When a reference price is selected from the list, if there is a defined special price detail record, “update” and “delete” buttons are enabled else “add” button is enabled.

```
Entry: Special Price Code, Special Price record Detail
Read: Special Price Code, Special Price Detail record, Reference Price Detail
Write: Special Price Detail(Insert/Update)
Exit: Error messages, Reference Price Detail List

2(E) + 3(R) + 1(W) + 2(X) = 8 CFP
Quality: A
Effort: 19 hours
Data Group: Special Price, Reference Price Detail, Special Price Detail
```

B.62. Charge Commission Amount Querying

Pricing service code is entered then pricing service code details are written on the screen. Then channel types and price types are read from database and listed into different dropdown lists. When group code button is clicked, different group codes are listed on another window and one of them is selected.

In the second step, in accounting information part there are two lists: Credit account info, debit account info.

The third part is credit information part. When card number button is clicked, card information is listed on the screen. In this part, there is a enterable file reference field.

```
Entry: Pricing Service, Reference (2 Entry)
Read: Pricing Service, Channel Type, Price Type, Group Code, Credit Account, Debit Account
    (6 Read)
Write: (No write)
Exit: Price Type List, Channel Type List, Group Code List, Error Messages (4 Exit)

2(E) + 6(R) + 4(X) = 12 CFP
Effort: 38 hours
Second Functional Process: Charge Commission Amount Calculation
```

The last step is calculation part. When calculate button is clicked, many controls are done.

(i) One control is for channel. If no pricing service code is defined for the selected channel, error message is given = 1Enrty + 1Read +1Exit(error message) = 3CFP, 2 hours

(ii) For pricing svc code, if a none defined amount interval is entered error message is given. 1Enrty + 1Read +1Exit(error message) = 3CFP, 3 hours

(iii) For pricing service code if entered credit interval is not defined, an error message is given 1Enrty + 1Read +1Exit(error message) = 3CFP, 1 hours

(iv) If currency codes of products and pricing service codes are incompatible or currency codes of debtor and creditor are incompatible, error message is given. 2En-

try (Product, Pricing Service) +2Read + 1Exit(Error Message) = 5CFP, 4 hours

2Entry (Creditor, Debtor) +2Read + 1Exit(Error Message) = 5CFP

```
Total: 3 + 3 + 3 + 5 + 5 = 19 CFP
Quality: A
Effort: 10 hours
Data Group: Pricing Service, Debit Account, Credit Account, Price Type, Channel Type, Group
    Code, Reference
```

B.63. Calculation of Profitability of Customer for Company

With customer number, customer information table is read. Then, for each customer pension contracts are gathered from the system. After that, from revenue table expenses are read. At the same time, fund collection of customer is read from account table. Then, these amounts are rated to predefined segments. Then, this information list is given to the user as output.

```
Trigger: Profitability of Customer Calculation Request
Entry: Customer
Read: Customer info, Account info, Revenue, Fund
Write: (No write)
Exit: VIP report list, Error Messages
(Batch)

1(E) + 2(R) + 2(X) = 5 CFP
Quality: A
Effort: 20 hours
Data Group: Customer, Account, Revenue, Fund
```

B.64. Extract of Account Report

Extract of account records that are needed to be printed in due date are read from extract documentation table. From these documents, by using document id pension contract table is read and then with the customer id general customer information table is read. Then, extract of account reports are delivered to the printed address or email of customer.

```
Trigger: Extract of Account Report Due Date
Entry: Extract of Account
Read: Extract of Account, Customer info, Pension Contract
Write: (No write)
Exit: Extract of Account Print List, Error Messages
(Batch)
```

```
1(E) + 3(R) + 2(X) = 6 CFP
Quality: A
Effort: 24 hours
Data Group: Extract of Account, Customer info, Pension Contract
```

B.65. Group Revenue List

From internet or the operation request, group revenue lists are given user between two specified dates. Firstly group number is entered and according to this number group table is read. Then, pension contracts related to group certificated are read from the system. After that, due list is found related to pension contracts with the specified dates. As output, these fields are merged in a report as listed to the user.

```
Trigger: Group Revenue List Request
Entry: Group number
Read: Group, Pension Contract, Due List
Write: (No write)
Exit: Group Revenue List, Error Messages
(Online)

1(E) + 3(R) + 2(X) = 6 CFP
Quality: A
Effort: 22 hours
Data Group: Group, Pension Contract, Due List
```

B.66. Campaign Parameter Entry Screen Change (Cool:GEN)

According to credit variety, new fields are needed to add to the campaign parameter entry screen. To fill the content of the fields, application product table is needed to read for main and sub product codes.

```
Trigger: F5
Entry: Campaign parameter
Read: Application product
Write: Campaign Parameter (Insert/Update)
Exit: Error Message, Product List

1(E) + 1(R) + 1(W) + 2(X) = 5 CFP
Effort: 8 hours
Quality: A
Data Group: Campaign parameter,  Application product
Development Tool: Pl/I
Program Type: Cics Batch
```

B.67. Closing Accounts According to Criteria

In account table, some commercial credit accounts have no balance amount. If this accounts are still open in case of due date is less than current date, they are needed to closed by updating open-close flag to closed. In other case, if the due date of account is bigger than current date and the credit is a commercial credit with payment plan and balance amount is zero, the account is needed to be closed with commission and interest accounts. And also, if the commercial credit has overdue accounts and some of the overdue accounts balance amount equals to zero, all overdue accounts are needed to be closed.

```
Trigger: Cleaning of the Account Request
Entry: Account (1 Entry)
Read: Account, Account Overdue Relation, Commercial Credit Account Relation, Payment Plan (4
     Read)
Write:  Account Information Update (1 write)
Exit: Error Message, Closed Account List (2 Exit)

1(E) + 4(R) + 1(W) + 2(X) = 8 CFP
Effort: 24 hours
Quality: A
Data Group: Account, Account Overdue Relation, Commercial Credit Account Relation, Payment
    Plan
Development Tool: PL/I
```

B.68. General Ledger Number Update

According to new IFRS system, general ledger numbers related to accounts are varied. So, old general numbers are needed to be updated according to the criteria. Overdue accounts are needed to control if there is record on overdue account relation table. If it has, the account is updated with a specified general number, if not updated with another one.

```
Trigger: General Ledger Number Update Request
Entry: Account (1 Entry)
Read: Account, Account Overdue Relation, Product Code General Ledger Number Relation (3 Read
    )
Write:  Account Update (1 write)
Exit: Error Message, Updated Account List (2 Exit)
```

```
1(E) + 3(R) + 1(W) + 2(X) = 7 CFP
Effort: 11 hours
Quality: A
Data Group: Account, Account Overdue Relation, Product Code General Ledger Number Relation
```

B.69. Adding new fields to the Campaign Parameter Entry Screen (PL/I)

According to credit variety, new fields are needed to add to the campaign parameter entry screen. To fill the content of the fields, application product table is needed to read for main and sub product codes.

```
Trigger: F5
Entry: Campaign parameter
Read: Application product
Write: Campaign Parameter (Insert/Update)
Exit: Error Message

1(E) + 1(R) + 1(W) + 1(X) = 4 CFP
Effort: 10 hours
Quality: A
Data Group: Campaign parameter, Application product
Development Tool: Pl/I
Program Type: Cics Batch
```

B.70. Adding new fields to the Campaign Product Parameter Entry Screen

According to credit variety, new fields are needed to add to the campaign product parameter entry screen. Personnel flag attribute is added to the screen and entered value is sent to the validation process. Then if it is validated, the record is inserted or updated in the campaign product parameter table.

```
Trigger: F5
Entry: Campaign product parameter
Read: No read
Write: Campaign Product Parameter (Insert/Update)
Exit: Error Message
```

```
1(E) + 1(W) + 1(X) = 3 CFP
Effort: 8 hours
Quality: A
Data Group: Campaign Product Parameter
Development Tool: Pl/I
Program Type: Cics Batch
```

B.71. Customer Report Interest Amount Setting

In customer record, amount sign is needed to update according to specified general ledger numbers. If the general number is defined for expense, interest amount is listed on the report with the sign minus. If it is for income, the amount is listed with the sing plus. So, income and expense is differentiated. Specified general ledger numbers are read from parameter table.

```
Trigger: Customer Account Report Request
Entry: Daily Account
Read: Parameter Read, Daily Account Read
Write: No database update
Exit: Customer Account Report

1(E) + 2(R) + 1(X) = 4 CFP
Effort: 6 hours
Quality: A
Data Group: Parameter, Daily Account, Customer Account
Development Tool: Pl/I
```

B.72. Adding new Accounting Case To the Income Return Process

In income return process, if extra commission is collected and all of it returned on the same day, there was no need for income return accounting. Because, income process has not started yet. However, extra commission collection program starts accounting process while collecting the commission. So, when it is returned back to the customer, income return accounting is needed to make trial balance consistent.

```
Trigger: Income Return Request
Entry: Income Information
Read: General Ledger Number Related to Product Code, Income Information
Write: Accounting Transaction
Exit: Error Message
```

```
1(E) + 2(R) + 1(W) + 1(X) = 5 CFP
Effort: 10 hours
Quality: A
Data Group: Income Information, General Ledger Number Related to Product Code, Accounting
    Transaction
Development Tool: Cool:Gen
```

B.73. Retail Credits Interest Discount Control Link

If the entered customer has loan interest discount, the rate is returned. Loan interest discount table includes this rate.

```
Trigger: Control Request for Interest Discount
Entry: Customer Information
Read: Loan Interest Discount
Write: No Write
Exit: Loan Interest Discount

1(E) + 1(R) + 1(X) = 3 CFP
Effort: 8 hours
Quality: A
Data Group: Customer Information, Loan Interest Discount
Development Tool: Cool:Gen
```

B.74. Customer Transfer Check Link

The link controls if the customer information on an existing customer number is transferred to a new customer number or not. If it is transferred, error message is given to the user.

```
Trigger: Control Request for Customer Transfer
Entry: Customer Information
Read: Customer Transfer
Write: No Write
Exit: Error Message

1(E) + 1(R) + 1(X) = 3 CFP
Effort: 8 hours
Quality: A
Data Group: Customer Information, Customer Transfer
Development Tool: Cool:Gen
```

B.75. Cost Matrix Special Price Definition Log List

Price Definition Log records are listed according to criteria.

```
Trigger: Daily Batch Schedule
Entry: Special Price
Read: Special Price Log
Write: No Write
Exit: Special Price Log List, Error Message

1(E) + 1(R) + 2(X) = 4 CFP
Effort: 6 hours
```

```
Quality: A
Data Group:  Special Price, Special Price Log
Development Tool: Pl/I
```

B.76. Accounting Transaction Group List Service

From accounting transaction table, balance amount is summed grouping by each general ledger number and list is given to the service caller.

```
Trigger: Accounting Transaction Service Call
Entry: General Ledger Number
Read: Accounting Transaction
Write: No Write
Exit: Accounting Transaction Group List

1(E) + 1(R) + 1(X) = 3 CFP
Effort: 8 hours
Quality: A
Data Group:  General Ledger, Accounting Transaction
Development Tool: Cool:Gen
```

B.77. Account Plan Service

Account plan table is read according to entered organization code and reconsolidation id number. And the read information is given as input to the service caller.

```
Trigger: Account Plan Service Call
Entry: Account
Read: Account Plan
Write: No Write
Exit: Account Plan Information List
```

```
1(E) + 1(R) + 1(X) = 3 CFP
Effort: 8 hours
Quality: A
Data Group:  Account, Account Plan
Development Tool: Cool:Gen
```

B.78. Accounting Transaction List Performance Enhancement

If some fields in the criteria list is not empty, these fields are given as equal to the where condition. A new read process is added.

```
Trigger: Performance Request
```

```
Entry: Accounting Transaction Criteria
Read: Accounting Transaction
Write: No Write
Exit: Accounting Transaction List

1(E) + 1(R) + 1(X) = 3 CFP
Effort: 7 hours
Quality: A
Data Group:  Accounting Transaction
Development Tool: Cool:Gen
```

B.79. Simulation facility to calculate EIR, adjustment and effective amount

According to given account and unit number; EIR, adjustment amount and effective amount calculation links are called from the client. Calculated EIR, adjustment and effective amount values are showed on the screen.

```
Entry: Account
Read: Account
Write: No write
Exit: Error Messages, Financial Report

1(E) + 1(R) + 2(X) = 4CFP
Quality: A
Effort: 8 hours
Data Group: Account, Financial Report
```

B.80. Valuable Fund Tax Transfer

For each valuable fund, cost is read from parameter table. Also, general ledger number and currency information is read from parameter table. After that trial balance records are fill into cursor. Last step is accounting process. If general ledger number and currency code together defined in the parameter table, balance amount is summed for each currency code and general ledger number and transferred to the new general ledger.

```
Trigger: Valuable Fund Tax Transfer
Entry: Unit Information
Read: Trial Balance, Parameter Information for cost, Parameter information for General
    Ledger Number and Currency Code
Write: Accounting Transaction
Exit: Accounting Transaction List
```

```
1(E) + 3(R) + 1(W) + 1(X) = 6 CFP
Effort: 10 hours
Quality: A
Data Group:  Accounting Transaction, Unit
Development Tool: Pl/I
```

B.81. General Ledger Number Update for Unit

General ledger number and account information is integrated on the batch process. Parametric approach is used. In parameter table, general ledger numbers and account which are wanted to update re defined, then the records with these read values are updated on the trial balance. Update process exist, only parameter read added.

```
Trigger: General Ledger Number Update Request
Entry: Parameter Type
Read: Parameter Info for Account, Parameter Info for General Ledger Number
Write: No write
Exit: Error Message

1(E) + 2(R) + 1(X) = 4 CFP
Effort: 4 hours
Quality: A
Data Group:  Parameter Info
Development Tool: Pl/I
```

B.82. Credit Read Service

According to entered application id number credit information is read and returned as output.

```
Trigger: Credit Read Service Call
Entry: Credit application id number
Read: Credit
Write: No Write
Exit: Credit information

1(E) + 1(R) + 1(X) = 3 CFP
Effort: 8 hours
Quality: A
Data Group:  Credit
Development Tool: Cool:Gen
```

B.83. Retail Credits Master Information Read Service

According to entered application id number master retail credit information is read and returned as output.

```
Trigger: Master Information Read Service Call
Entry: Retail Credit application id number
Read: Master
Write: No Write
Exit: Master  information

1(E) + 1(R) + 1(X) = 3 CFP
Effort: 8 hours
Quality: A
Data Group:  Master Information
Development Tool: Cool:Gen
```

B.84. Parametric Accounting Detail Read Service

According to entered accounting key parametric detail information is read and the list is returned to the service caller.

```
Trigger: Parametric Accounting Detail Read Service Call
Entry: Accounting Key
Read: Parameter Read Check the Key, Parametric Detail Read
Write: No Write
Exit: Parametric Detail List, Error Messages (Accounting key is not defined)

1(E) + 2(R) + 2(X) = 5 CFP
Effort: 10 hours
Quality: A
Data Group:  Accounting Parameter, Accounting Parametric Detail Information, Error Message
Development Tool: Cool:Gen
```

B.85. Rediscount Information Update Service

According to entered rediscount type code and rediscount record key, rediscount information table is read and the status of the record related to the key information is updated.

```
Trigger: Rediscount Information Update Service Call
Entry: Rediscount Information (type code and record key)
Read: Rediscount Information
Write: Rediscount Information Status Update
Exit: Error Messages(Rediscount information is not found)
```

```
1(E) + 1(R) + 1(W) + 1(X) = 4 CFP
Effort: 8 hours
Quality: A
Data Group:  Rediscount Information, Error Message
Development Tool: Cool:Gen
```

B.86. Amortization Information Update Service

According to entered amortization type code and amortization record key, amortization information table is read and the status of the record related to the key information is updated.

```
Trigger: Amortization Information Update Service Call
Entry: Amortization Information (type code and record key)
Read: Amortization Information
Write: Amortization Information Status Update
Exit: Error Messages(Amortization information is not found)
```

```
1(E) + 1(R) + 1(W) + 1(X) = 4 CFP
Effort: 8 hours
Quality: A
Data Group:  Amortization Information, Error Message
Development Tool: Cool:Gen
```

B.87. Commission Report new fields request for Retail Credits

With new request credit type, commission period type and collection type of retail credits are needed to be listed in the commission report. Credit type and commission period type is constant for retail credits. These are not needed to be read from the system. Commission period type is related to commission type. If commission period type is set to 'D' it means commission period type is daily, otherwise it means monthly administration.

```
Entry: Retail Credit Account
Read: Retail Credit Commission
Write: No Write for this change
Exit: Error Messages, Retail Credits Commission Report List

1(E) + 1(R) + 2(X) = 4CFP
Effort: 3 hours
Quality: A
Data Group: Retail Credit Account, Retail Credit Commission
```

B.88. Commission Report new fields request for Commercial Credits

With new commission report request; credit type, commission period type and collection type of commercial credits are needed to be listed in the commission report. Collection type is related to status code of the commission. According to predefined status codes, collection type is set as collected or uncollected. Credit type code is related to payment plan number of the credit. If the credit has payment plan number, credit type is set to IFRS credit, if it no it is set to out of IFRS. Commission period type is read from amortization table. If commission period type is set to 'D' it means commission period type is daily, otherwise it means monthly administration.

```
Entry: Commercial Credit Account
Read: Commercial Credit Commission, Payment Plan
Write: No Write for this change
Exit: Error Messages, Commercial Credits Commission Report List
```

```
1(E) + 2(R) + 2(X) = 5CFP
Effort: 5 hours
Quality: A
Data Group: Commercial Credit Account, Commercial Credit Commission, Commercial Payment Plan
```

APPENDIX C: REGRESSION EXPLAINED IN SIMPLE TERMS

C.1. Statistics Formulas and Definitions

This is the common vocabulary when using descriptive statistics to present the data:

- Valid N (listwise) - This is the number of non-missing values.
- N - This is the number of valid observations for the variable. The total number of observations is the sum of N and the number of missing values.
- Minimum - This is the minimum, or smallest, value of the variable.
- Minimum - This is the minimum, or smallest, value of the variable.
- Mean - This is the arithmetic mean across the observations. It is the most widely used measure of central tendency. It is commonly called the average. The mean is sensitive to extremely large or small values.
- Std. - Standard deviation is the square root of the variance. It measures the spread of a set of observations. The larger the standard deviation is, the more spread out the observations are.
- Variance - The variance is a measure of variability. It is the sum of the squared distances of data value from the mean divided by the variance divisor. The Corrected SS is the sum of squared distances of data value from the mean. Therefore, the variance is the corrected SS divided by N-1. We don't generally use variance as an index of spread because it is in squared units. Instead, we use standard deviation.
- Skewness - It measures the degree and direction of asymmetry. A symmetric distribution such as a normal distribution has a skewness of 0, and a distribution that is skewed to the left, e.g. when the mean is less than the median, has a negative skewness.

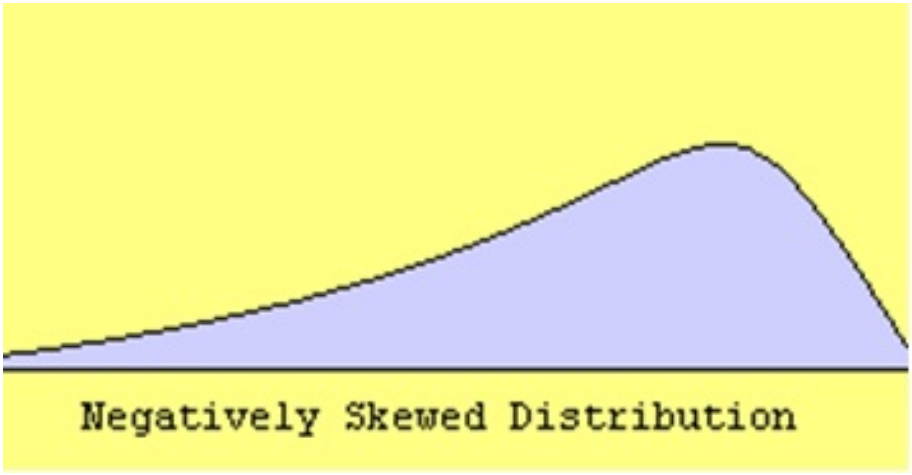


Figure C.1. Negatively Skewed Distribution.

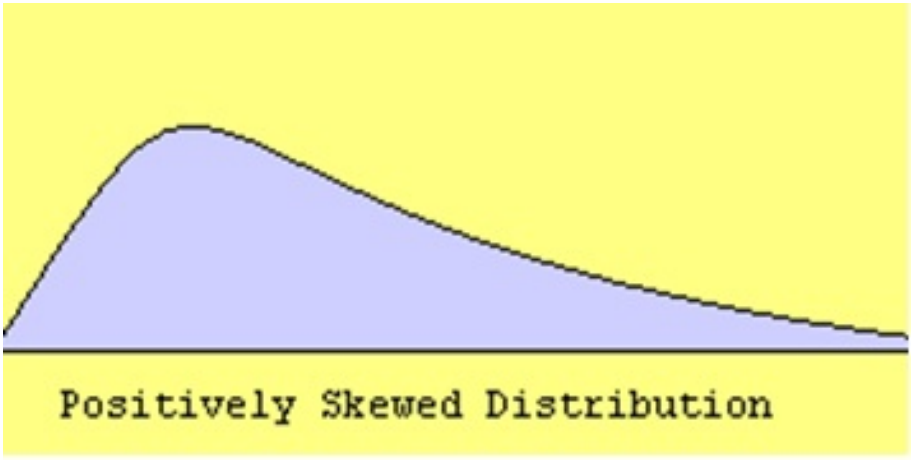


Figure C.2. Positively Skewed Distribution.

C.2. Model ANOVA

In this model, the last column Sig. shows the goodness of fit of the model. If this number < .01 then the model is significant at 99%, if it <.05 then the model significant at 95%, and if it <.1 the model is significant at 90% [53]. Significance implies the acceptance of the model. The lower this number is, the better the fit.

To compare different models, the column F is used. If this number is not significant, then it cannot be said that the model 1 is any better than the model 2 [53].

The Sum of Squares column holds the amount that could be explained by the model or not. Total shows total deviations in the dependent variable, Regression shows amount of the total deviations that could be explained by the mode and Residual shows the amount that could not be explained by the model.

Regression = Total Sum of Squares – Explained Sum of Squares

ANOVA[a]

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	54514.39	5	10902.88	414.262	.000[b]
	Residual	52295.48	1987	26.319		
	Total	106809.9	1992			

a. Dependent Variable: WAGE

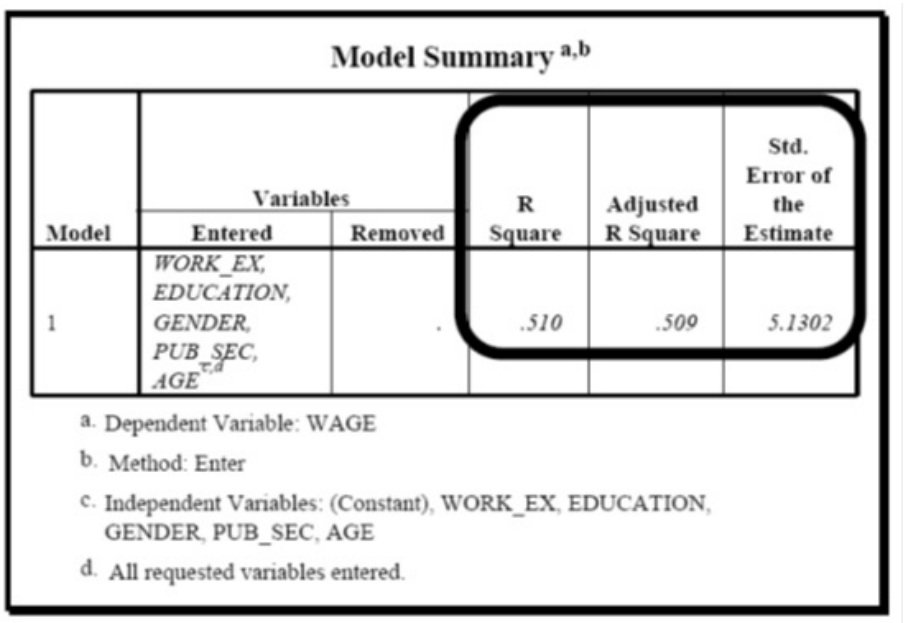
b. Independent Variables: (Constant), WORK_EX, EDUCATION, GENDER, PUB_SEC, AGE

Figure C.3. Model fit ANOVA.

C.3. Model Summary

This is the common vocabulary when using Model Summary to make regression analysis:

- Adjusted R-square: measures the proportion of the variance in the dependent variable that was explained by the variations in the independent variable.
- R-square: measures the proportion of the variation in the dependent variable that was explained by variations in the independent variable.
- Std Error of Estimate: measures the dispersion of the dependent variables estimate around its mean.
- Heteroskedasticity: In statistics, a collection of random variables is heteroskedastic if there are sub-populations that have different variabilities than others.



Model Summary [a,b]

Model	Variables Entered	Variables Removed	R Square	Adjusted R Square	Std. Error of the Estimate
1	*WORK_EX, EDUCATION, GENDER, PUB_SEC, AGE* [c,d]	.	*.510*	*.509*	*5.1302*

a. Dependent Variable: WAGE

b. Method: Enter

c. Independent Variables: (Constant), WORK_EX, EDUCATION, GENDER, PUB_SEC, AGE

d. All requested variables entered.

Figure C.4. Model Summary.

C.4. Coefficients Table

The table "Coefficients" provides information on the confidence with which for each estimate the estimate can be supported. If the values "Sign." Is less than 0.05, the estimate in column "B" can be asserted as true with a 95% level of confidence. If this value is more than 0.1 then the coefficient estimate is not reliable because it has too much variance.

Coefficients[a]

Model		Unstandardized Coefficients		t	Sig.	95% Confidence Interval for B	
		B	Std. Error			Lower Bound	Upper Bound
1	(Constant)	-1.820	.420	-4.339	.000	-2.643	-.997
	AGE	.118	.014	8.635	.000	.091	.145
	EDUCATION	.777	.025	31.622	.000	.729	.825
	GENDER	-2.030	.289	-7.023	.000	-2.597	-1.463
	PUB_SEC	1.741	.292	5.957	.000	1.168	2.314
	WORK_EX	.100	.017	5.854	.000	.067	.134

a. Dependent Variable: WAGE

Figure C.5. Coefficients.

REFERENCES

1. *ISO/IEC, 12207:2008-Systems and Software Engineering - Software Life Cycle Processes*, 2008.

2. Brooks, F., "The Mythical Man-Month. Addison-Wesley", Chapel Hill, 1995.

3. Torchiano, M., F. Ricca, and A. D. Lucia, "Empirical Studies in Software Maintenance and Evolution", *ICSM 2007 IEEE International Conference on Software Maintenance*, 2007.

4. Boehm, B. W., "Improving Software Productivity", *IEEE Transactions on Software Engineering*, Vol. 20, pp. 43–57, 1987.

5. "Measurement Manual v3.0.1 (The COSMIC Implementation Guide for ISO/IEC 19761: 2003)", *The Common Software Measurement International Consortium*, 2009.

6. Boehm, B. W., "Software Engineering", *IEEE Transactions on Computers*, Vol. 25, pp. 1226–1241, 1976.

7. *ISO/IEC, 14764:2006, Software Engineering - Software Life Cycle Processes – Maintenance*, 2006.

8. Boehm, B. W., "Software Engineering Economics", *IEEE Transactions on Software Engineering*, Vol. SE-10, pp. 4-21, 1984.

9. Benestad, H., B. Anda, and E. Arisholm, "Understanding Software Maintenance and Evolution by Analyzing Individual Changes", *Journal of Software Maintenance and Evolution: Research and Practice*, Vol. 21, pp. 349–378, 2009.

10. April, A., A. Abran, and R. Dumke, "Software Maintenance Management: Evaluation and Continuous Improvement", *Wiley-IEEE Computer Society Press*, pp.

314, 2008.

11. Kajko-Mattsson, M., "Corrective Maintenance Maturity Model (CM3): Maintainer's Education and Training", *ICSE '01 Proceedings of the 23rd International Conference on Software Engineering*, pp. 610–619, 2001.

12. April, A. and A. Abran, "Estimation Models for Software Maintenance Based on Functional Size", *Journal of Software Technology*, Vol. 9, 2006.

13. April(1), A., A. Abran(2), and R. R. Dumke(3), "Software Maintenance Productivity Measurement: how to asses the readiness of your organization", (1)École de Technologie Supérieure, Montréal, Canada, (2)École de Technologie Supérieure, Montréal, Canada, (3)Otto von Guericke University of Magdeburg, Germany.

14. Abran, A. and H. Nguyenkim, "Analysis of Maintenance Work Categories Through Measurement", *Proceedings. Conference on Software Maintenance*, 1991.

15. Koskinen, J., "Software Maintenance Costs", Departmant of Computer Science and Information Systems, University of Jykäskylä, Jyväskylä, 2010.

16. Kuhlmann, U., "Maintenance Activities in Software Process Models: Theory and Case Study Practice", University of Koblenz Landau, Campus Koblenz, Faculty of Computer Sciences, Institue of Software Engineering, 2003.

17. Deming and W. Edwards, "Out of the Crisis", MIT Center for Advanced Engineering Study, Cambridge, 1986.

18. Boehm, B. W., "Software Cost Estimation with COCOMO II", Prentice Hall PTR Upper Saddle River, NJ, USA, 2000.

19. Boehm, B., B. Clark, and E. Horowitz, "Cost Models for Future Software Life Cycles Processes: COCOMO 2.0", Chris Westland USC Center for Software Engineering, Ray Madachy USC Center for Software Engineering and Litton Data Systems, Richard Selby UC Irvine and Amadeus Software Research.

20. Boehm, B., R. Valerdi, J. A. Lane, and A. W. Brown, "COCOMO Suite Methodology and Evolution", University of Southern California.

21. Abran, A. and P. N. Robillard, "COCOMO Suite Methodology and Evolution", .

22. Boehm, B. W., "A Spiral Model of Software Development and Enhancement", TRW Defense Systems Group.

23. Bennett, K. and V. Rajlich, "Software Maintenance and Evolution: A Roadmap", Research Instititue for Software Evolution University of Durhan UK, Department of Computer Science Wayne State University Detroit.

24. Boehm, B. W., "Improving Software Productivity", TRW, 1987.

25. Desharnais, J., F. Pare, M. Maya, and D. St-Pierre, "Implementing a Measurement Program in Software Maintenance An Experience Report Based on Basili's Approach", *IFPUG 1997 Spring Conference*, 1997.

26. Lientz, B. P., E. B. Swanson, and G. E. Tompkins, "Characteristics of Applications Software Maintenance", Vol. 21, pp. 466–471, 1978.

27. Lehman, M. M., "Software System Maintenance and Evolution in an Era of Reuse, COTS, and Component-Based Systems", *keynote delivered at ICSM*, Oxford, 1999.

28. Genuchten, M. V., G. Brethouwer, T. V. den Boomen, and F. Heemstra, "An Empirical Study of Software Maintenance", *Information and Software Technology*, 1992.

29. Arfa, L., A. Mili, and L. Sekhri, "An Empirical Study of Software Maintenance", *Conference on Software Maintenance*, 1991.

30. Jones, C., "The Economics of Softare Maintenance in the Tweenty First Century", February 2006.

31. de Souza, S. C. B., N. Anquetil, and K. M. de Oliveira, "A Study of the Documentation Essential to Software Maintenance", 2005.

32. "Software Maintenance As Part of the Software Life Cycle", Comp180: Software Engineering, Prof. Stafford, Department of Computer Science Tufts University, 2003.

33. Martin, J. and C. McClure, "Software Maintenance: The Problem and Its Solutions", Prentice Hall, Englewood Cliffs, NJ, 1983.

34. Abran, A. and H. Nguyenkim, "Analysis of Maintenance Work Categories Through Measurement", *IEEE Conference on Software Maintenance*, pp. 104–113, Sorrento, Italy, October 1991.

35. Abran, A. and H. Nguyenkim, "Measurement of the Maintenance Process from a Demand-based Perspective", *Journal of Software Maintenance : Research an Practice*, Vol. 5, pp. 63–90, Sorrento, Italy, 1993.

36. Maya, M., A. Abran, and P. Bourque, "Measuring the Size of Small Functional Enhancements to Software", Universite du Quebec a Montreal.

37. Behrens, C. A., "Measuring the Productivity of Computer Systems Development Activities with Function Points", *IEEE Transaction on Software Engineering*, Vol. 9, November 1983.

38. Scacchi, W., "Understanding and Improving Software Productivity", Institue for Software Research, University of California, 2005.

39. ISBSG, "ISBSG official site", 2010, `http://www.isbsg.org`, accessed at February 2010.

40. *ISO/IEC, 14143-1:1997 - Information technology - Software measurement - Functional size measurement - Definition of concepts*, 1997.

41. Fenton, N. E. and A. Kitchenham, "Validating Software Measures", *Journal of Software Testing-Verification and Reliability*, Vol. 1, pp. 27–42, 1991.

42. Finkelstein, L., "A Review of Fundamental Concepts of Measurement", *Journal of Software Testing-Verification and Reliability*, Vol. 13, pp. 55–64, 2008.

43. *ISO/IEC, 19761:2003-Software Engineering-COSMIC-FFP-A Functional Size Measurement Method, International Organization for Standardization*, 2003.

44. IBM, "IBM official site", 2012, `http://publib.boulder.ibm.com/infocenter/zos/basics/index.jsp?topic=/com.ibm.zos.zappldev/zappldev_126.htm`, accessed at May 2011.

45. "The COSMIC Functional Size Measurement Method Version 3.0.1 Guideline for Assuring the Accuracy of Measurements Version 0.92", *The Common Software Measurement International Consortium*, 2011.

46. "The Cosmic Funtional Size Measurement Method Version 3.0.1, Guidline for Assuring the Accoracy of th Measurements, Version 1.0", *The Common Software Measurement International Consortium (COSMIC)*, 2011.

47. Basili, V., "Quantitative Software Complexity Models: A panel summary In V.R. Basili, Ed., Tutorial on Models and Methods for Software Management and Engineering", *IEEE Computer Society Press*, Los Alamitos, Calif, 1980.

48. ASQ, "The Global Voice of Quality", 2011, `http://asq.org/learn-about-quality/project-planning-tools/overview/pdca-cycle.html`, accessed at July 2010.

49. Desharnais, J. M., "Productivity Model and Benchmarking", *Course Notes*, 2011.

50. Mathworld, "Standard Deviation", 2012, `http://mathworld.wolfram.com/StandardDeviation.html`, accessed at Jun 2011.

51. Mathworld, "Skewness", 2012, `http://mathworld.wolfram.com/Skewness.html`, accessed at Jun 2011.

52. "Regression explained in simpler terms", *A Vijay Gupta Publication, SPSS for Beginners*.