KADIR HAS UNIVERSITY

GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

SOFTWARE COST ESTIMATION IN THE SOA AGE

MASTER OF SCIENCE THESIS

ASLI YILMAZ TAŞKIN

İSTANBUL, TURKEY

February, 2012

Student’s Full Name: ASLI YILMAZ TAŞKIN M.Sc. Thesis 2012

SOFTWARE COST ESTIMATION IN THE SOA AGE

ASLI YILMAZ TAŞKIN

Submitted to the Graduate School of Science and Engineering
in partial fulfillment of the requirements for the degree of
Master of Science
in
Computer Engineering

KADİR HAS UNIVERSITY
February, 2012

KADIR HAS UNIVERSITY GRADUATE SCHOOL OF
SCIENCE AND ENGINEERING

SOFTWARE COST ESTIMATION IN THE SOA AGE

ASLI YILMAZ TAŞKIN

APPROVED BY:

Assistant Prof. Dr. Arif Selçuk ÖĞRENCİ ____________________
Kadir Has University
(Thesis Supervisor)

Assistant Prof. Dr. Osman Kaan EROL ____________________
Istanbul Technical University

Assistant Prof. Dr. Taner ARSAN ____________________
Kadir Has University

APPROVAL DATE: / /2012

SOFTWARE COST ESTIMATION IN THE SOA AGE

Abstract

Many estimation models have been proposed to examine the accuracy of software cost estimation. We focus on the development cost of services in SOA (Service Oriented Architecture). We tried to find new factors that affect the development cost for SOA projects. We defined a new framework which may help to find the development cost in SOA.

The main point of our research is focused on this question: “What is the difference in SOA age?” Today, in SOA age, many researchers try to define new frameworks, because calculating cost estimation using old models such as COCOMO II gives inaccurate results when it comes to service-based systems. SOA implication, analysis and coding stages provide us to find new effective factors for calculating the cost estimation.

In our research, we defined a new framework to calculate development cost of a service. We used this framework in our experiments and we obtained different results. We found the effect of I/O and Complexity metrics on development cost estimation. We also took an advantage of COCOMO II model while we are creating our main metrics to calculate cost.

During our experiments we created 50 different clusters with 100 or 200 different service samples with Monte Carlo simulation. As a result of our experiments, we found the optimal service for all clusters. We have shown services’ development costs which are minimum and maximum in detailed graphs. Finally, we found that cluster size of granulated services does not affect the development cost excessively.

Key Words: Development Cost Estimation, SOA, Estimation Models, Monte Carlo.

SOA ÇAĞINDA YAZILIM MALİYETLENDİRMESİ

Özet

Günümüzde yazılım maliyetini hesaplamak için birçok yazılım maliyet tahminleme modeli geliştirilmiştir. Yaptığımız araştırmada, servis odaklı mimarilerdeki servislerin geliştirme maliyetine odaklanılmıştır. Çalışmalarımızda servis odaklı mimari projelerde yazılım geliştirme maliyetini etkileyecek yeni faktörler bulmaya çalıştık. Servisin geliştirme maliyetini servis odaklı mimarilerde bulmamıza yarayacak yeni bir yapı tanımladık.

Yaptığımız araştırmada, servis odaklı mimarideki farklılıkların neler olduğu konusuna odaklandık. Günümüz servis odaklı mimari çağında, birçok araştırmacı COCOMO II gibi eski maliyetlendirme modellerinin gerçeğe yakın olmayan sonuçlar verdiğinden dolayı yeni yapılar tanımlamaya ihtiyaç duymaktadırlar. Servis odaklı mimarilerin uygulama, analiz ve kodlama aşamaları maliyetlendirme hesaplamasında kullanacağımız yeni ve etkili faktörler bulmamıza yardımcı olmaktadır.

Tanımlamış oldugumuz geliştirme maliyetini hesaplayan yapıyı farklı deneylerde kullanarak değişik sonuçlar elde ettik. Bu deneyler sonucunda, Input/Output ve karmaşıklık faktörlerinin yazılım maliyetlendirmesine olan etkisini saptadık. Çalışmalarımız süresince, maliyet hesaplamasında kullandığımız faktörleri COCOMO II modelinden de faydalanarak tanımladık.

Yaptığımız deneyler süresince, Monte Carlo simülasyon programını kullanarak 50 kümeli (cluster) 100 – 200 örnekli empirik sonuçlar elde ettik. Deneylerimizin sonucunda her küme için optimal servisi bulduk. Bu servisleri karşılaştırmak üzere çizdiğimiz grafikler ile gösterdik.

Sonuç olarak, parçalanmış servislerin büyüklüklerinin yazılım maliyeti üzerindeki etkisinin cok fazla olmadığını saptadık.

Acknowledgements

During the development of my graduate studies at Kadir Has University several people have collaborated directly and indirectly with my research. Without their support it would be impossible for me to finish my work. That is why I wish to dedicate this section to recognize their support.

I want to start expressing a sincere acknowledgement to my advisor; Assistant Prof. Dr. Arif Selçuk Öğrenci. He gave me the opportunity to research under his guidance and supervision. I received motivation, encouragement and support from him during my studies.

I also want to thank the motivation and support I received from my husband. Special thanks I owe to my brother Assistant Prof. Dr. Alper Yilmaz for his support and guidance during my studies.

At last, but the most important I would like to thank to my family for their encouragement, inspiration and endless love.

Asli Yilmaz Taskin

Table of Contents

List of Tables

List of Figures

1. INTRODUCTION

In recent years, software started to manage our lives. Thus, many of companies have to arrange their budget to buy good software to use at their work. Today, software companies are trying to give the best services to customers by developing new services.

Software cost estimation is the process of predicting the effort required to develop a software system.[1] According to Hareton Leung and Zhang Fan [1]; “Accurate software cost estimates are critical to developers and customers. Understanding the costs may result in management approving proposed systems that then exceed their budgets, with under-developed functions and poor quality, and lapse of time. ”

Calculation of software estimation consists of some actions. These actions start with analysing and refining software requirements, SW architecture and programmatic constraints. Then, we have to define software elements. By using lines of code, we can calculate the estimated size of software according to existing methods. I/O size and complexity of service are important metrics for the cost estimation.

Identifying project risks, estimating their impact and revise estimates are other steps for finding cost estimation of software. But in our research we will focus on the development cost. So, we will consider mainly the granularity of a service in development cost estimation.

[1] Hareton Leung, Zhang Fan “Software Cost Estimation”, The Hong Kong Polytechnic University, Japan.

Validation and verification are important concepts for estimation models. Developing alternative effort, schedules and cost estimation help to improve accuracy of methods. In our research, we defined a new framework to calculate development cost of a service. We used this framework in our experiments and we obtained different results. We found the effect of I/O and Complexity metrics on development cost estimation. We also took an advantage of COCOMO II model while we are creating our main metrics to calculate cost.

2. GENERAL VIEW OF SW COST ESTIMATION

2.1 Software Cost Estimation

Software cost is an important concept in development of software. Up to today, many of researchers tried to find new models and look at new points of view to improve the accuracy of cost estimation. These methods use some target parameters such as human effort, size of project, timing and hardware and software costs during project development.

Project cost estimation and project scheduling are normally carried out together. The costs of development are primarily the costs of the effort involved, so effort computation is used in both the cost and the schedule estimate.[2]

Human effort is the dominant cost factor for cost estimation. Companies pay cost for software engineers to develop new projects. Sometimes engineers have to travel during the project while others working extra in the office. These all factors mean to extra cost to bosses.

Misestimating cost of the project may cause many problems. These problems affect budget directly.

During Development, managing the cost is very important. Cost estimation processes consist of some management steps. Some of these important steps are;

- Resources that will be used by the project can be determined. Extra resources mean extra payment. So, determining the resources and their cost to developers

[2] Ian Sommerville "Software Cost Estimation", Software Engineering 7th Edition, UK, 2004

will be definite before the project starts. Project cost is accurate if the resources are enough for real needs.

- According to Hareton Leung and Zhang Fan [1]; “Spending more time than determined may cause extra cost. During contract bidding, result in not winning the contract which can lead to loss of jobs.”

- Effort is an important concept of calculating the software cost estimation. Generally effort is measured in person in months of the workers. Workers may consist of programmers, analysts and the manager of projects. They all earn salary per unit time of staff involved. Multiplying the salary per unit time with estimated effort required may give us the cost of project. But, there are external factors that affect the cost of the project.

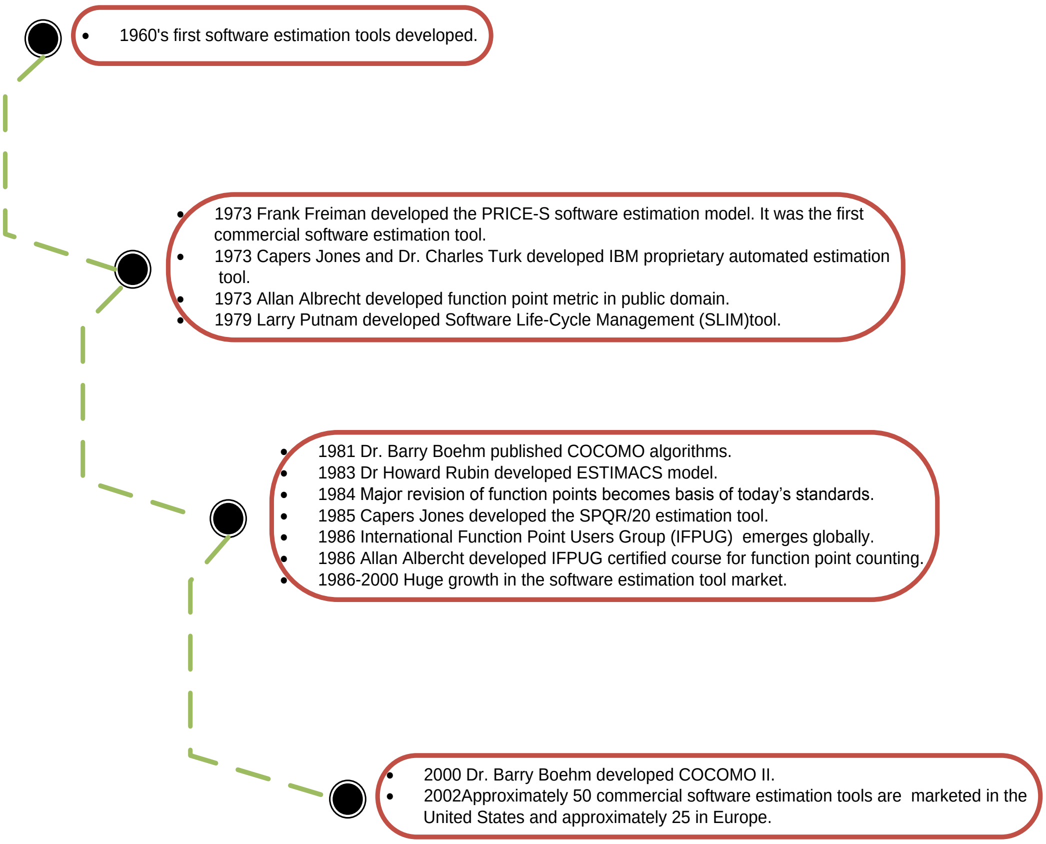
Except these steps there are some important attributes of a good software estimate[3]:

- It is conceived and supported by the project manager and the development team.
- It is accepted by all stakeholders as realizable.
- It is based on a well-defined software cost model with credible basis.
- It is based on a database of relevant project experience (similar processes, similar technologies, similar environments, similar people and similar requirements).
- It is defined with enough detail so that its key risk areas are understood and the probability of success is objectively assessed.

[3] W.Royce , “Software project management: a unified framework, Adisson Wesley”, 1998

2.2 Evolution of Software Cost Estimation

The evolution of software cost estimation tools have started to develop in early 1960’s. Today, there are a lot of papers that study methodology of cost estimation.
A time-line of the software estimation tools is shown in **Figure 1.**



- 1960's first software estimation tools developed.

- 1973 Frank Freiman developed the PRICE-S software estimation model. It was the first commercial software estimation tool.
- 1973 Capers Jones and Dr. Charles Turk developed IBM proprietary automated estimation tool.
- 1973 Allan Albrecht developed function point metric in public domain.
- 1979 Larry Putnam developed Software Life-Cycle Management (SLIM)tool.

- 1981 Dr. Barry Boehm published COCOMO algorithms.
- 1983 Dr Howard Rubin developed ESTIMACS model.
- 1984 Major revision of function points becomes basis of today’s standards.
- 1985 Capers Jones developed the SPQR/20 estimation tool.
- 1986 International Function Point Users Group (IFPUG) emerges globally.
- 1986 Allan Albercht developed IFPUG certified course for function point counting.
- 1986-2000 Huge growth in the software estimation tool market.

- 2000 Dr. Barry Boehm developed COCOMO II.
- 2002Approximately 50 commercial software estimation tools are marketed in the United States and approximately 25 in Europe.

Figure 1- Evolution of Software Cost Estimation

According to the NASA Johnson Space Center [4] , In the late 1940’s, the DoD, and the United States Air Force began a study of multiple scenarios concerning how the country should proceed into the age of jet aircraft, missiles and rockets. The Military saw a need for a stable, highly skilled group of analysts to help with the evaluation of such alternatives.

In 1950, Rand’s successful studies represented one of the most systematic studies of cost estimation in the airplane industry. Rand’s group proved to be prolific contributors to the art and science of cost analysis.

The Cost Estimating Relationship (CER) was found by Rand in the mid 1950’s. CER was a basic tool of cost estimation. Merging CER with the learning curve formed the foundation of parametric aerospace estimating. This finding is still used today.

[4] NASA Johnson Space Center , “Parametric Cost Estimating Handbook”, NASA, United States of America, 1995

According to National Aeronautics and Space Administration[6] “For the first time, cost analysts saw the promise of being able to estimate relatively quickly and accurately the cost of proposed new systems. Rand extended the methods throughout the 1950’s, and by the early 1960’s, the techniques were being applied to all phases of aerospace systems.[5]”

After 1970, there were more robust models such as COCOMO [Boehm 1981] Checkpoint [Jones 1997], PRICE-S [Park 1988], SEER [Jensen 1983], and SLIM [Putnam and Myers 1992]. Even though the researchers started to develop these robust models about the same time, they all faced the same problem.

This problem was about the size and importance of the software. Researchers found that if software grew in size and importance means grew in complexity, making it very difficult to calculate the cost of software development.

It was difficult to develop new parametric models that work efficiently and accurately for software development in all domains because of the changing form of software development.

To that end, most of the software estimation models have developed from 1981 to 1997 based on the leading efforts of researchers which are mentioned above.[6]

[5] National Aeronautics and Space Administration [1995] “Parametric Cost Estimating Handbook”, Department of Defense, USA

[6] Barry Boehm, Chris Abts, [Chulani 1998]“Software Development Cost Estimation Approaches-A Survey1”, Sunita Chulani IBM Research, University of Southern California, Los Angeles

2.3 Description of Basic Software Cost Estimation Models

2.3.1 PRICE-S Model

Lockheed Martin Life Cycle Cost Estimating Systems' PRICE-S is a proprietary, empirically-based cost model. Since it is a proprietary model, complete information about the internals of the model are unavailable.

Some details, however, are available about the model. Unlike other models, PRICE-S uses machine instructions, not source lines of code, as its main cost driver. You can hire consulting services from Lockheed Martin to exercise PRICE-S. However, if you need to use a consultant anyway to perform your estimates, the fact that PRICE-S is proprietary and requires a consultant to utilize may not be a problem to your organization. PRICE-S is one of the earliest and most successful models that have been developed.[7]

2.3.2 SLIM Model

Software Life Cycle Model (SLIM) is marketed by Quantitative Software (QSM). SLIM was developed in 1979 by Mr. Larry Putnam. Originally developed from analysis of ground-based radar programs, the SLIM tool has been expanded to include other types of programs. It can be customized for the user's development environment.

SLIM supports all phases of software development, except requirements analysis, as well as all sizes of software projects, but was especially designed to support large projects.

Success in using SLIM depends on the user's ability to customize the tool to fit the software development environment and to estimate both a Productivity Index (a measure of the software developer's efficiency) and a Manpower Build up Index (a measure of the software developer's staffing capability). SLIM also provides a life cycle option which extrapolates development costs into the maintenance phase [7].

[7] Boehm, B., Clark, B., Horowitz, E., Westland, C., Madachy, R., Selby, R. “Cost Models for Future Software Life Cycle Processes”,1995 http://sunset.usc.edu/COCOMOII/Cocomo.html

2.3.3 Basic COCOMO (COCOMO 81) Model

The Constructive Cost Model (COCOMO) is the best known and most popular cost estimation model. COCOMO was developed in the late 1970s and early 1980s by Barry Boehm (1981).

This early model consists of a hierarchy of three increasingly detailed models named Basic COCOMO, Intermediate COCOMO and Advanced COCOMO. These models were developed to estimate custom, specification-built software projects.[8]

2.3.4 COCOMO II Model

The COCOMO II research was started in 1994 at USC (The University of Southern California).

COCOMO II was initially published in the Annals of Software Engineering in 1995 [Boehm *et al.* 1995]. There are three sub-models of COCOMO II model. These are Applications Composition, Early Design and Post-Architecture. Every sub-model can be unified in different ways for dealing with the today's and future's software practices workplace.

According to Barry Boehm and Chris Abts [7] "The Application Composition Model is used to calculate effort and schedule on projects that use Integrated Computer Aided Software Engineering tools for rapid application development. These projects are too scattered but sufficiently easy to be rapidly composed from interoperable components."

The Application Composition Model has several components. These are GUI builders, database or objects managers, middleware for distributed processing or transaction processing and domain components like financial or medical process control packages.

[8] The Data & Analysis Center for Software "Modern Empirical Cost and Schedule Estimation Tools", https://www.dacs.dtic.mil/techs/estimation

Boehm[9] shows that COCOMO II be used to identify the critical cost driver factor and estimate the cost difference that would result due to change in the critical driver factor.

[9] Boehm,B "Safe and Simple software cost analysis", IEEE Software, 2000

3. COCOMO&COCOMO II COST ESTIMATION MODELS BY USING COSTAR AND FUNCTION POINTS

3.1 Advantages of Using COCOMO

Today, thousands of people (SW project managers, etc…) use the COCOMO model to calculate the cost of the project. COCOMO is an open model. This is the basic difference of COCOMO model. There are some advantages of COCOMO model[10];

- COCOMO estimates are more repeatable than estimates made by methods relying on proprietary models.
- COCOMO can be calibrated to reflect your SW development environment, and to produce more accurate estimates.

3.2 Using Costar

Costar is the implementation of COCOMO. Costar helps to define software structure and it is used by small projects. Using the right COCOMO and COCOMO II definition and assumption contributes to the accuracy of the Costar.

Assume that, at the first time of coding in project, analyst and developers decided to start with 2000 line of code. At the coming days, these codes may be separated into little pieces. So, there will be a system and other subsystems. Costar is used while

[10] Web Site; www.softstarsystems.com/overview.htm

code is separated into little pieces. Method allows defining components of each subsystem. These processes keep on until project's needs are received.

3.3 Using Function Points

Function point is a method that is used by the developers and managers to calculate the cost estimation and it is developed by Allen Albrecht.

According to function points, estimation should be identified while forming the project analysis. Important items for Function Points are [11];

- External inputs
- External outputs
- Logical internal files
- External interface files
- External inquiries

Function Point provides to estimate source lines of code.
Costar converts the Function point count into an equivalent number of SLOC (source lines of code), and uses that in the COCOMO equations to make its estimates.

Function Point method separates all management tools in to several levels. These management tools may be project risks, abilities of all project members, etc…

Method puts these levels in order. All levels have special coefficient. By using SLOC and these coefficients, the spending person-month will be found.

4. PRINCIPLES OF THE COCOMO CALCULATION

4.1 Model Structure

According to Laren Lum, Michael Bramble, Jairus Hihn, John Hackney, Mori Khorrami and Erik Monson [11]; “Many parametric models compute effort in a similar way, where estimated effort is proportional to size raised to a factor.”

$$\boxed{\mathbf{E= [A\ (Size)^{B}\ (EM)]}} \qquad \mathbf{(1)}$$

The expression is given in Equation (1) is COCOMO estimation effort formula where;

E is an estimated effort in work-months.
A is a constant that reflects a measure of the basic organizational / technology costs.

Size is the equivalent number of new logical lines of code. Most parametric tools are able to compute the equivalent lines of code by using size and heritage percentage inputs. Size may change by the code growth according to the requirement evolution.

B is a scaling factor of size. It is a variable exponent whose values represent economies/ diseconomies of scale.

[11] Laren Lum, Michael Bramble, Jairus Hihn, John Hackney, Mori Khorrami, Erik Monson “Handbook for software cost estimation”, Jet Propulsion Laboratory, Psadena,California, 2003

EM is the product of a group of effort multipliers that measure environmental factors used to adjust effort (**E).** The set of factors comprising **EM** are commonly referred to as cost drivers because they adjust the final effort estimate up or down.

4.2 Software Sizing

The most important factor that affects the software cost is the software size of the project. “Line of Code” and “Function Points” is the most important metrics for calculating the cost estimation.

4.2.1 Inside the Line of Code (LOC)

Line of code is the number of lines of the delivered source code of the software. It is not possible to obtain the actual LOC before the project has completed.

There is a technique to calculate the cost size(S). The technique for calculating the size of project is called as PERT.

Actual size is calculated by taking the average of lowest possible size, highest possible size and most likely size. According to the PERT analysis technique, cost size is computed as;

$$\boxed{S= (S1 +Sh+4Sm) / 6} \quad \textbf{(2)}$$

The expression is given in Equation (2) is a Pert Analysis Technique Cost Size Equation where;

$\mathbf{S_1}$ is the lowest possible size,
$\mathbf{S_m}$ is the most likely size.
$\mathbf{S_h}$ is the highest possible size,

According to J.D. Aron[12] “PERT can also be used for individual components to obtain an estimate of the software system by summing up the estimates of all the components.”

[12] J.D. Aron “Estimating Resource for Large Programming Systems”, NATO Science Committee, Rome, Italy, 1969

Line of Code can be calculated with another method which uses unadjusted function-point counts (UFC). UFC equation is shown in section **0**

Linear formula for code-size estimation of the project is;

$$\boxed{LOC= a* UFC +b} \qquad \textbf{(3)}$$

The expression is given in Equation (3) is Pert Analysis Technique- Line of Code Equation where;

LOC is Line of Code,
UFC is Unadjusted Function Point Counts
a and b can be derived from previously completed project data and can be obtained using linear regression.

4.2.2 Inside the Function Points

According to the Function Points**;** there are 5 important classes to estimate source lines of code. These are[13];

- User-input types (data or control user-input types).
- User-output types (output data types to the user that leaves the system).
- Inquiry types (interactive inputs requiring a response).
- Internal file types (files [logical group of information] that are used and shared inside the system).
- External file types (files that are passed or shared between the system and other systems).

As we mentioned in section 3.3; Function Point method separates all management tools in to several levels. By calculating the unadjusted function-point counts (UFC); the complexity levels are shown in three groups; {1= Simple, 2=Medium, 3=Complex}.

[13] A.J Albrecht, and J. E. Gaffney, “Software Function, source lines of codes and development effort prediction: a software science validation”; IEEE Trans Software Eng. SE-9 pp. 639-648, 1983

According to Albrecht, the unadjusted function-point count (UFC) equation is given as;

$$\boxed{\text{UFC}=\sum_{i=1}^{5}\sum_{j=1}^{3} N_{ij}\, W_{ij}} \qquad \textbf{(4)}$$

The expression is given in Equation (4) is Albrecht's Unadjusted Function Point Count Equation where;

$\mathbf{W}_{ij}$ is the weight of type class i with complexity j.

$\mathbf{N}_{ij}$ is the number of class i with complexity j.

Also there is another equation for UFC;

$$\boxed{UFC=\sum_{i=1}^{5} n_i * W_i} \qquad \textbf{(5)}$$

The expression is given in Equation (5) is UFC Equation without complexity factor where;

W_i is the weight of i.

n_i is the number of items of variety i where i stands for the number of items 1, items 2 etc..

Function Points for the system can be calculated by the following empirical formula;

$$\boxed{\text{FP=UFP x (0.65+0.01xTCF)}} \qquad \textbf{(6)}$$

The expression is given in Equation (6) is Function Point Equation where;

(*)UFP is Unadjusted Function Point described in section **0.**

(*)TCF is Technical Factor described in section **0.**

4.2.3 Unadjusted Function Points (UFP)

The function point classes are the components of the system. The system components have five types which are external or logical inputs, outputs, inquiries, external interfaces to other systems and the logical internal files.

These components are further weighted as "simple", "average" or "complex" depending on their characteristics. Then the sum of all components is called as Unadjusted Function Points (UFP).[14]

4.2.4 The Technical Factors

Technical Factor describes the size of the technical complexity involved in the development and implementation of the system.[15] Technical Factors is calculated by the equation below;

$$\boxed{TCF = C_1 + C_2 \sum_{i=1}^{n} F_i} \qquad \textbf{(7)}$$

The expression is given in Equation (7) is Technical Factors Equation where;

$C_1 =$ 0.65

$C_2 =$ 0.01

F_i : is the factor valued from 0 to 5. 0 if it is irrelevant and 5 if it is essential.

[14] http://www.ii.metu.edu.tr/;"Software Quality Assurance".

[15] G. Karner ; "Resource Estimation for Objectory Projects"; Objective Systems; Torshamnsgatan, 1993

5. INTRODUCTION TO SERVICE ORIENTED ARCHITECTURE

5.1 Overview of SOA (Service Oriented Architecture)

A Service Oriented Architecture is a collection of services. These services communicate with each other by passing simple data. Also, communications between services could involve two or more services coordinating some activity.

Before understanding the ability of service oriented architecture, we have to define the meaning of a service. Service is a function that is well-defined, self-contained, and does not depend on the context or state of other services.[16]

Service is the endpoint of connection. The basic service oriented architecture is shown in **Figure 2**.

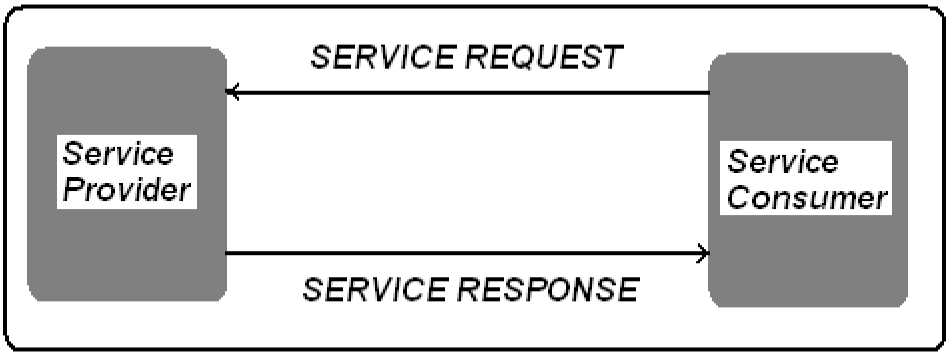


Figure 2- Basic Service Oriented Architecture

[16] http://www.service-architecture.com/web-services/articles/service oriented_architecture_soa_definition.html, *"SOA Definition"*

At the figure above, Service Consumer sends service request to Service Provider. The Service Provider sends back service response message to service consumer. These connections between Service provider and consumer is providing by the web services.

Services provider can also be services consumer and it may requests service from other services providers. These connections form the basic structure of Service Oriented Architecture.

There are 2 main reasons to choose SOA;

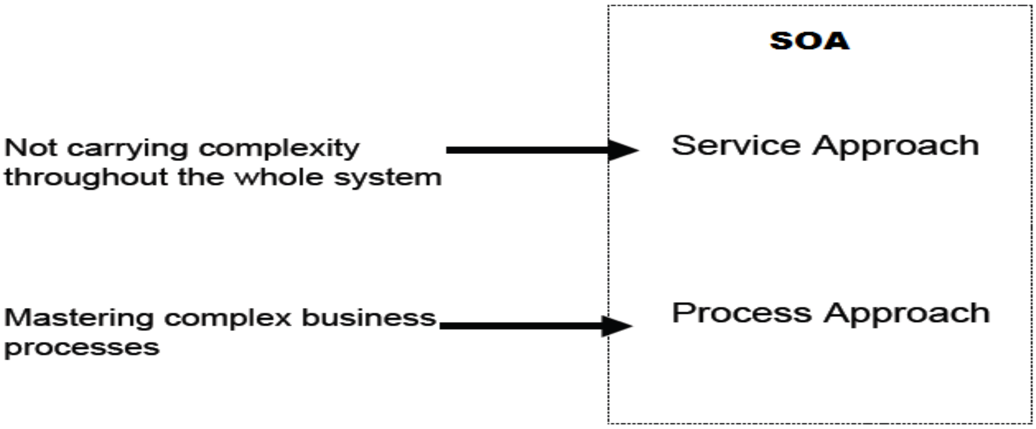

SOA
Not carrying complexity throughout the whole system
Service Approach
Mastering complex business processes
Process Approach

Figure 3-Why SOA?

<u>Service approach provides to:</u>

- Integrate sub-systems through services
- Ease uniqueness principle avoiding redundancies
- Build a core framework of services for future development

<u>Process approach provides to;</u>

- Separate business process from code
- Provide better visibility on business processes
- Ease maintenance and evolution of business processes

5.2 Evolution of Service Oriented Architecture

In the past, programmers had coded little units with a code pieces and system used these code pieces in lots of positions. In old years, it was enough for people to do their work. According to development of new technologies these methods became unserviceable.

In recent years, new communication age damaged the component based architecture. Programmers wanted to use the components of system at the remote machines. This idea exposed new software concepts. These are Distributed Component Object Model (DCOM) and Component Object Model (COM+).

After distributed systems are used, programmers coded little code pieces and put them into only one place to use. So, lots of applications used only one component. This procedure formed the application server.

The usage of software services continued to develop itself day by day. But there was another problem: The coded component runs only on platforms of the same type. Different type of platforms caused many problems. SOA developed to take up this problem to serve best service to users.

5.3 How is Service Oriented Architecture different?

For understanding the difference of SOA from other approaches; first we have to understand its structure. Once the structure is understood it is possible to compare SOA and other approaches to understanding and organizing Information Technology assets.

According to Matthew MacKenzie, Ken Laskey, Francis McCabe, Peter F Brown, Rebekah Metz, Booz Allen Hamilton[17] "SOA reflects the reality that ownership boundaries are a motivating consideration in the architecture and design of systems.

This recognition is evident in the core concepts of visibility, interaction and effect. However, SOA does not itself address all the concepts associated with ownership, ownership domains and actions communicated between legal peers.**"**

The service descriptions and service interfaces inside SOA automatically provide location references. This facilitates the reuse of frameworks and synergic systems which are developed externally.

Additionally, SOA helps applying the lessons which are learned from commerce to the organisation of Information Technology assets for facilitating the matching of facilities and necessities. Thus, entities which league together inside the context of a single interaction require the swap of some type of value.

According to Matthew MacKenzie, Ken Laskey, Francis McCabe, Peter F Brown, Rebekah Metz, Booz Allen Hamilton [22] "This is the same fundamental basis as trade itself, and suggests that as SOAs evolve away from interactions defined in a point-to-point manner to a marketplace of services; the technology and concepts can scale as successfully as the commercial marketplace."

We will see the effects of SOA on cost estimation in section 7.

[17] OASIS-Matthew MacKenzie, Ken Laskey, Francis McCabe, Peter F Brown, Rebekah Metz, Booz Allen Hamilton; "Reference Model for Service Oriented Architecture 1.0", Committee Specification, 2006

6. ADAPTATION OF SOFTWARE COST ESTIMATION FOR SOA PROJECT

Today, most companies ask the question “How much this SOA will cost?” for their projects. This is an important question to obtain the funding of the company.

Obtaining the cost of the project is starting with understanding the domain. Understanding the domain in detail includes;

- Number of Data Elements
- Complexity of Data Storage Technology
- System Complexity
- Service Complexity
- Process Complexity
- New Services needed
- Enabling Technology
- Applicable Standards
- Potential Risks

After this step we have to make a decision about required resources for the project and understand their cost.

6.1 Number of Data Elements & Complexity of Data Storage Technology

Defining a unit of data for processing directly effects the project cost estimation while you are creating the data structure of the system.

According to David Linthicum who is an internationally known application integration and service oriented architecture expert, this formula may help to find the data complexity depending on the number of data elements. [18]

$$\boxed{\begin{array}{c}\text{Cost of Data Complexity = (Number of Data Elements) *}\\ \text{(Complexity of the Data Storage Technology) * (Labor Units)}\end{array}} \quad \textbf{(8)}$$

The expression is given in Equation (8) is Cost of Data Complexity Equation where;

Number of Data Elements being the number of semantics you are tracking in your domain, new or derived.

Complexity of the Data Storage Technology, expressed as a percentage between 0 and 1 (0% to 100%).

Labor Unit, the amount of money it takes to understand and refine one data element Clearly, if the number of data elements increases then the complexity of the structure will increase as well.

6.2 System Complexity

System complexity is a composite measure of complexity inside procedures and between them. It measures the complexity of a system design according to procedure calls, data use and parameter passing.

[18] http://www.soainstitute.org/,David S. Linthicum; “How much will your SOA cost?”, 2007

The complexity of system is basically a time and design metric. Before the original implementation exists it is possible to measure the difficulty of creating a designed system.

Also, source code is another important factor for calculating system complexity. It is another way of calculating the complexity of the system.

According to Card and Agresti; [19] "System complexity is not suitable for the evaluation of how difficult it is to change an existing system".

System complexity contains two main metrics inside it. These metrics are structural complexity and data complexity. We can use these two main metrics while we are calculating the System Complexity.

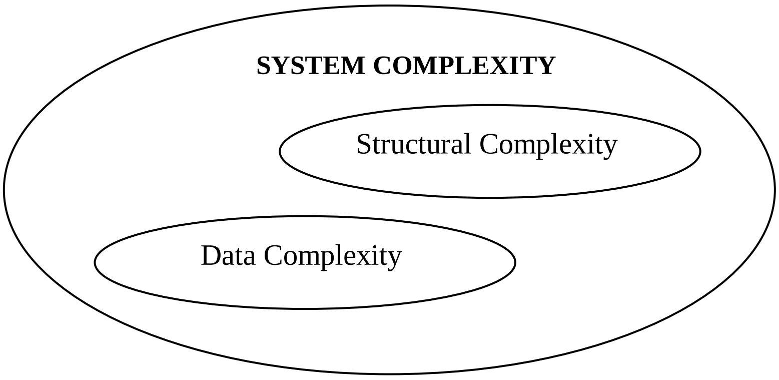


Figure 4- System Complexity and its metrics

Now, we will see how to calculate system complexity according to the structural complexity and data complexity.

6.2.1 Structural Complexity

Structural complexity is measured by the number of procedures. Structural Fan-in and Fan-out values vary the complexity result directly.

[19] Card, D. N. and W. Agresti, "Measuring Software Design Complexity." The Journal of Systems and Software 8, 3; 185-197. (Original definition of the metric),1998

Structural fan-in (**SFIN**) and fan-out (**SFOUT**) values are used to evaluate the interrelations between procedures and files. It is also possible to measure the complexity of the static structure of code which includes design and time factors.[20]

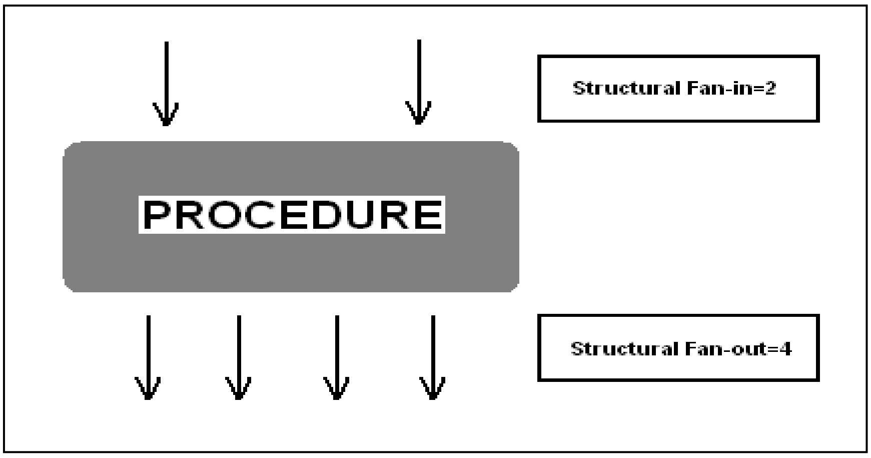


Figure 5- Simple call tree for one procedure

As you see in **Figure-5**, there are two coming calls to the procedure and when you control the SFOUT we can say that the procedure calls four other procedures to run in the system.

If you think there is more than one procedure in the system, the complexity of the structure will increase according to the SFIN and SFOUTs.

SFIN might be zero thus that means no procedure callers were found in the system. Being too many SFINs are not as desirable for a file.

6.2.2 Data Complexity

Data refers to a collection of information or facts usually collected as the result of experience, observation or experiment, or processes within a computer system, or a set of premises. For instance, data may consist of numbers and words.

[20] http://www.aivosto.com/project/help/pm-sf.html

Reading and writing business data in some form or order is the main responsibility of e-business applications. So, the storage, management and retrieval of the data always have been a tough problem in all organizations.

Using different mechanisms increases the cost of data depending on the usage of multiple databases.
According to SYS-CON Media's research [21] "Companies may have heterogeneous data environments with different schemas and they may contain redundant data elements. This data may be static reference data, such as personal customer information or geographical data, common business data, or common external data such as market data. This can lead to serious inefficiencies and consequently higher costs because of the overhead in accessing/updating data in multiple databases using different mechanisms."

There are some problems that affect the cost estimation of the SOA via data complexity.
These problems may be described as followed;

- Updating the data in multiple channels
- Difficulty of creating a common data spreading strategy for different technologies.
- Data may need to be synchronized between data repositories
- The requirements of the data access performance are increasing through the developing nature of business.
- The diversion of needs and usage patterns of data across systems

Data complexity directly affects the cost of project because of the time factor. If the user spends much more time than usual this means the structure of SOA is not work well. So, we have to develop a scalable strategy that can accommodate new applications with minimal turnaround time.

On the other hand, data has multiple channels of update. Hence, data has to be modified and controlled to make sure that updates are performed regularly.

[21] http://soa.sys-con.com, 2005

6.3 Service Complexity

Service is the work done by one person or group that benefits another people. The service complexity is related with the size of the project. The services that are given to the customers can be measured by all processes that are served in the system.
There might be several factors that affect the service complexity. We can say that, security is one of the main metric of the service complexity.

Today, companies want to provide a secure system service for their customers. So, if the security of the services increases the service complexity may increases as well.

6.4 Process Complexity

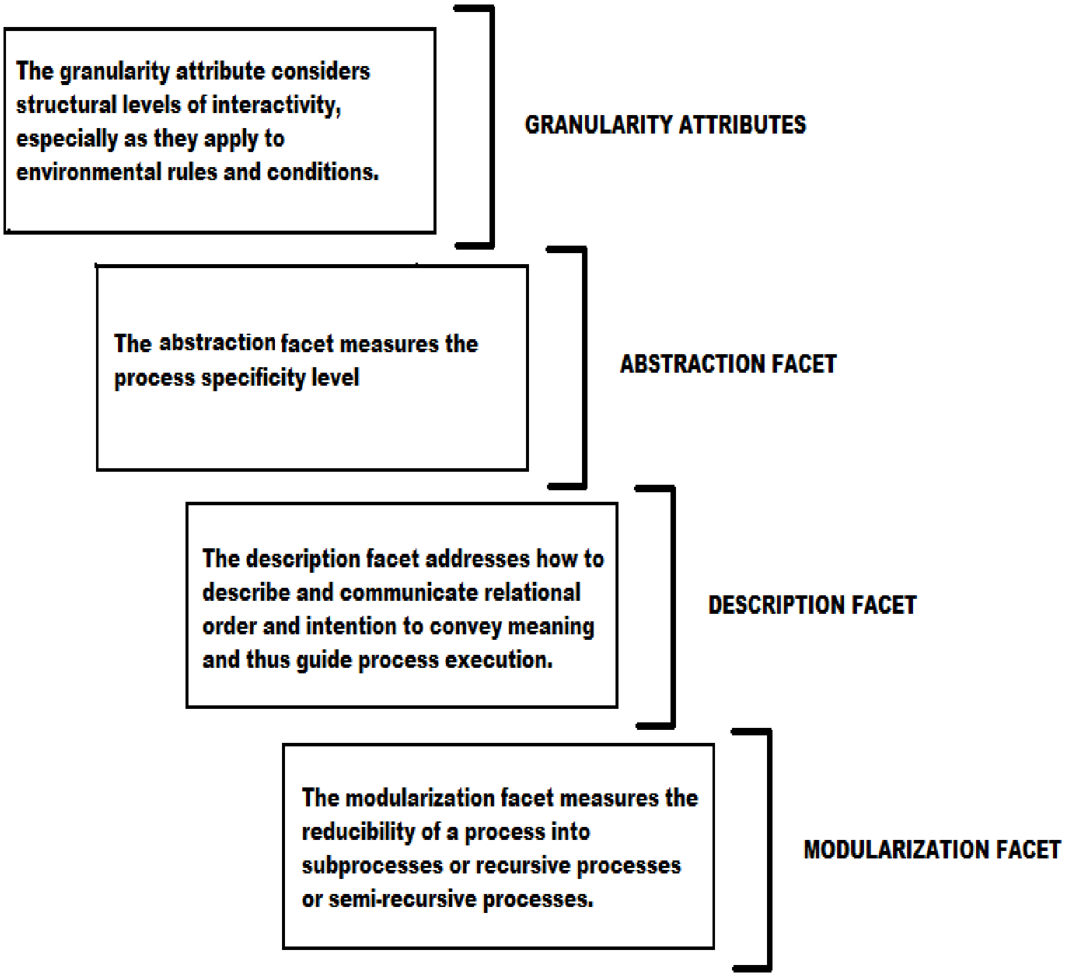
Process Complexity is another factor while calculating cost estimation of the system.

There are some metrics of the process complexity:

- a contents facet including coverage and granularity attributes
- an abstraction facet
- a description facet including form and notation attributes
- a modularization facet[22]

  Figure 6 shows the description of these metrics:

[22] C. Roland, "A comprehensive view of process engineering, Proceedings of the 10th international conference " and B. lecture notes in computer science, Italy, 1998



The granularity attribute considers structural levels of interactivity, especially as they apply to environmental rules and conditions.

The abstraction facet measures the process specificity level

The description facet addresses how to describe and communicate relational order and intention to convey meaning and thus guide process execution.

The modularization facet measures the reducibility of a process into subprocesses or recursive processes or semi-recursive processes.

Figure 6- Description of process complexity metrics.

6.5 New Services

Services are changing day by day. Thus, companies should be able to give new services to their customers to address much more people.

New technologies provide to add new customers to use your system. Needless to say that, adding new customers to the system takes some time. Increasing user number means the company of yours can earn much more money.

New standards bring new cost to your company and it is hard to raise the price of the services. This problem may be solved by reducing the operating costs.

According to the description above, by understanding the needs of your users, you can manage the cost of your system and give better service to customers. Otherwise, the cost of the system may increases.

6.6 Enabling Technology

Enabling technology is another metric for understanding the domain. Developing the domain of the system by using new technologies may decrease the development time of the project. So, the cost of the project will be decreased automatically.

6.7 Potential Risks

Beside all of these factors, we should think about risks during the project development. For example, economic crisis, decrease of the developer performance or problems during analysis process. Especially, analysing process takes too much time because of the misunderstanding between analyser and customer.

Unfortunately, most companies do not care about analysing process as much as it needs. Because, the manager of the project thinks that it is a loss of time to analyse deeply and they want to start development process immediately. However, analysing process is the main factor of the cost estimation.

Disregarding the analysing process may cause more loss of time unlike the manager's thinking. Therefore the risk of project might be increased.

Project risk is considered **low** when all processes involve fairly simple operations. Project risk is considered **medium** when the minority of the business processes under automation is complex, involving multiple steps, exchanges with external systems or significant validation/processing logic. Project risk is considered **high** when a majority of the business processes under automation are considered to be complex.

7. A NEW POINT OF VIEW TO SERVICE ORIENTED ARCHITECTURE COST ESTIMATION

The business conditions are changing day by day. SOA is developed to make IT more flexible. By using SOA, IT becomes adapted to changing business conditions.

SOA provides some advantages like simplifying integration, managing complexity, reducing costs and increasing reuse. As we mentioned, SOA is directly related with cost.

Steghuis [23]gives useful major reasons to answer the question “Why SOA is better in reducing costs?” These are:

- SOA implies a services mind-set and this anticipates shared use of services.
- SOA leads to monitoring and management of quality of the services so that experiences of the shared services can be exchanged.
- Services encapsulate complexity and isolate changes by only exposing their interface to the outside world.
- Standardisation of interfaces and the way they are exposed reduces cost and promotes reuse
- Services architecture focuses on the end-to-end lifecycle instead on single lifecycle stages.

On the other hand service granularity of SOA affects the whole system by changing its complexity. "Service granularity is the scope of functionality exposed by a service". [41]

[23] Claudia Steghuis, “Service granularity in SOA projects ” University of Twente, Netherlands, 2006

Modularity of a system can be reflected by its granularity. SOA needs to have well designed services to provide flexible business processes. Well-designed services provide low development cost.

Now, we can explore component based SOA systems to adapt new cost estimation facilities into SOA cost.

8. GENERAL VIEW OF COMPONENT BASED SOA SYSTEMS

8.1 Step by Step Cost Estimation in Component Based SOA Systems

Component based system is a branch of software engineering. This system consists of components that include data and functions inside it. These components are modular and cohesive, because they are semantically related with the contents of classes.

Each component may offer services from the system. The whole system finds the related interface that is offered by the component. We can say that interfaces are the signatures of related components. They find the services that will be used by the component. Component can use other component by using their interfaces.

While designing Component-Based System; first of all, we have to define the requirements of the system. These requirements help to generate the components of the system.

During planning the structure of the system, we should find new ways to make it flexible for using it in different platforms.

Decomposition of requirements is one of the most important processes before finding function points for each component. Because, using a lot of unnecessary components means much more line of code. So, it may affect the cost of the system. To determine complexity; for each component, we have to find Function Point items to calculate the cost estimation of service.

As we said before, we have to find external inputs, external outputs, logical internal files, external interface files and external inquiries to determine the complexity of the system.

If the number of components of the system increases than the UFP (Unadjusted Function Point) will increase as well. Function points will be affected by the value of UFP and technical factors. Technical factors are the total number of factors that affect the technical complexity.

UFP can be used to find the line of code. Line of code is the value that affects the cost of the system. Line of code will increase if the UFP increases. So, we have to compose the components of the system carefully.

Before determining the complexity of system, we will see how to decompose the SOA.

8.2 Evolution of Decomposition in Software

According to Boris Liblinsky; the first software decomposition approach introduced in the early 1960s was splitting mainframe applications into separate jobs, each implemented by a separate program. Later, as more insight into the program internals was gained, each program itself was split into modules or subroutines, according to their function.[24]

In 1970's decomposition adoption by introducing objects strengthened. These objects was implemented a model of real thing. Bust, on the other hand, abstractions provided by objects turned out to be too fine grained to have meaning on business level.

[24] Boris Lublinsky; "Defining SOA as an Architectural Style", www.ibm.com, 2007

To find better designed paradigm, researchers continued searching to find different approaches to decomposition in the late 1990's. Researchers were thinking that raising the level of abstraction and increasing granularity will help to fix the problems of object oriented systems.

According to component based architecture, software applications were much more flexible, better structured and more manageable. But, on the other hand it was decomposing the enterprise IT functionality.

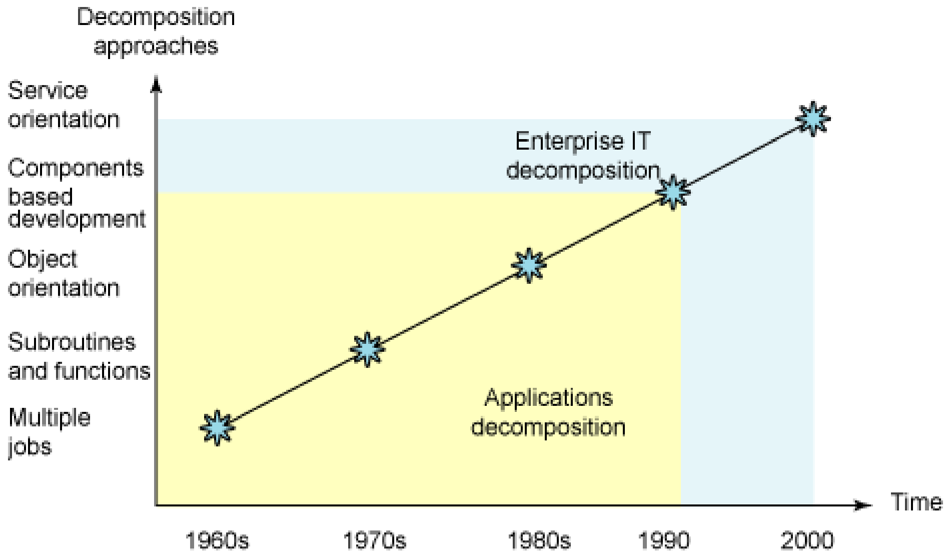


Figure 7- Evolution of decomposition approaches.[29]

9. APPLICATION SIZE USING FUNCTION POINT ANALYSIS (FPA)

Function point analysis (FPA), is one of the most popular and usable approaches for estimating the software size.

There are three main processes of FPA. These are;

- It quantifies the functionality requested by the customers and also provided functionality to the customers.
- Measure software development and maintenance independently of technology used for implementation.
- Across all projects and organizations, consistently measures the software development and maintenance.

User’s view of the system can change the size of the project according to their requirements. The inputs, reports, screens, stored data on the system will change the size of the application. We can say that user interactions may be one of the important inputs to change the size estimation of system.

FPA does not matter if it is an application written in .Net, Java etc… It is technology independent.

Adding FPA to our system is not expensive and provides to measure the size of the project to calculate the cost estimation of the system. FPA well works with use cases, so it is useful for object oriented software as well.

FPA measures the cost, duration, staffing size of the project. It seems to be easy to understand Cost per FP, FP's per hour and project defect rate metrics by using FPA.

The benefits of PFA are,

- Difficulties of estimation using Lines of Code can be avoided by using function point analysis.
- Sizing is important while determining the productivity of the system. FPs can be used to size software applications accurately.
- FPA is easily understood by every people, so it helps communicate sizing info to a user or customer.
- FPA is more productive than other methods, because it can be used to determine whether an environment, tool and language.
- The time or person is not important, they can be counted by different people at different times and they obtain the same measure within a reasonable margin of error.

Thomas Fetcke, Alain Abran and Tho Hau Nguyen[25] have used "o-o Jacobson Approach". **Figure-8** gives details about their function model.

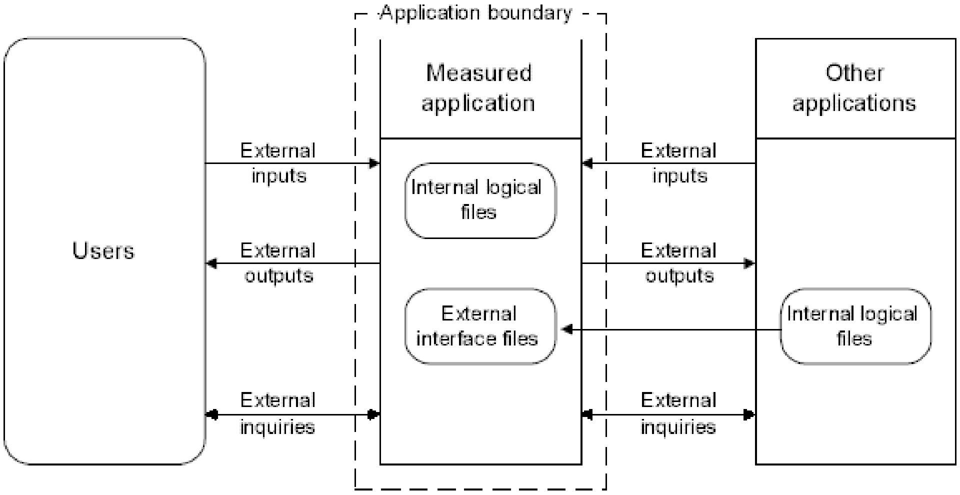


Figure 8- A high Level view of the abstract function point model with users and links to other applications.

[25] Thomas Fetcke, Alain Abran and Tho Hau Nguyen, **"**Mapping the o-o JacobsonApproach into Function Point Analyses**",** IEEE, 1998

9.1 Data Functions and Transactional Functions

As we mentioned in function point section; there are five main functions to be counted. Two of them are data functions and the other three are the transactional functions.

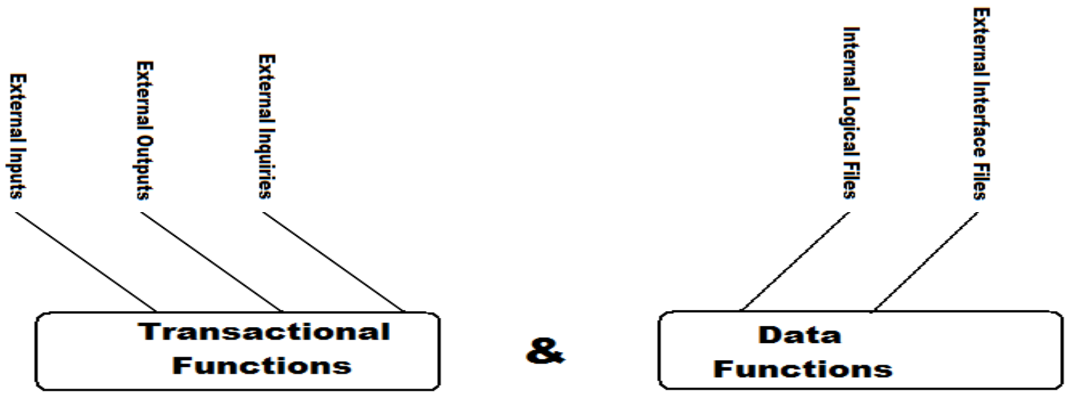


Figure 9- Data Functions and Transactional Functions

Using this terminology one of the researcher has drawn a figure to understand counted metrics during FPA.[26]

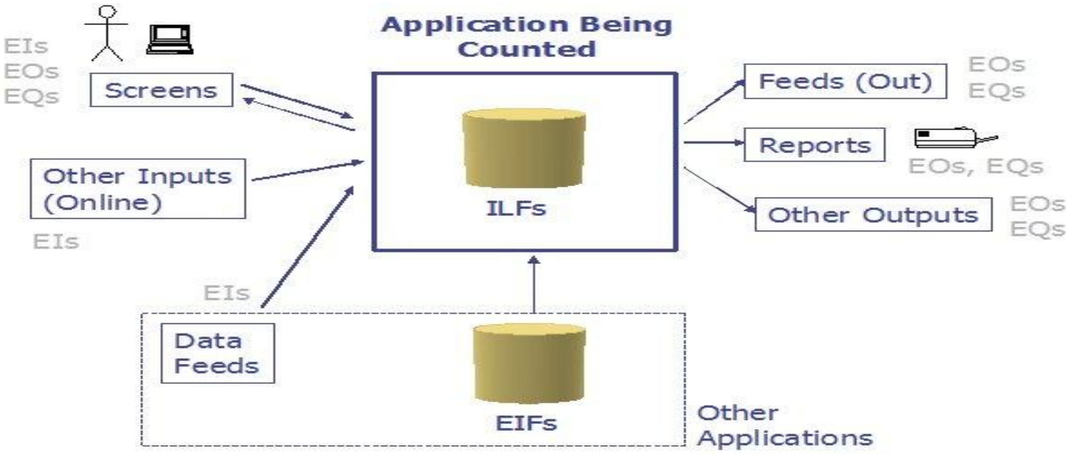


Figure 10- View of a software application from the eyes of a Function Point practitioner.

[26] Alvin J. Alexander; "How to Determine Your Application Size Using Function Point Analysis", 2004

9.1.1 Data Functions

(i) Internal Logical Files (ILF)

ILF represents the data stored in the system. For counting ILF we have to make sure that the data and group information are logical and user identifiable.

The group of data which is maintained through the beginning operation should be counted. Tables in a database and flat files can be an example to the ILFs.

First of all we have to determine the complexity level of Low (L), Average (A), or High (H). We can do this by counting number of Data Element Type (DET) and Record Element Type (RET).

RET is user recognizable data elements within ILF and EIF (External Interface File).**DET** is a unique, user recognizable and non-repeated field. You can find the points by complexity by checking number of DETs and RETs.

RETS	**Data Element Types (DETs)**		
	1-19	20-50	51+
1	L	L	A
2 to 6	L	A	H
6 or more	A	H	H

Table 1- ILF complexity matrix

For example, according to T**able-1**, we can say that, if there are 4 RETs and 25 DETs then, that system would bear (A) Average Complexity. This example is based on a real time project which is found on the internet.

Complexity	**Points**
Low	7
Average	10
High	15

Table 2 - ILF Complexity Matrix Weight.[27]

As you can see in table 2, a Low complexity ILF is worth 7 points, an Average ILF is worth 10 points, and a High is worth 15 in their example.

After finding complexity from the complexity table, we can determine the number of FPs that will be counted for this ILF. According to our example, the complexity is Average, so we can say that, the function points of this ILF will be 10FPs.

(ii) External Interface Files (EIF)

EIF represents the data that the application will use or refer. These data cannot be maintained by the existing system.

The main difference between EIF and ILF is that the EIF uses data maintained within the boundary of another application. ILF uses its own application. The data is not maintained within the boundary of another application. You can find the points by complexity by checking number of DETs and RETs.

As we mentioned on ILF section, checking the number of RETs and DETs will help us to find complexity and weight of the application. According to the tables above you can check the values and calculate complexity and weight.

[27] http://www.devdaily.com

RETS	Data Element Types (DETs)		
	1-19	20-50	51+
1	L	L	A
2 to 6	L	A	H
6 or more	A	H	H

Table 3- EIF Complexity Table [27]

According to the **table-3**, we can say that, if we count 3 RETs and 21 DETs that would be of (A) Average Complexity.

Complexity	Points
Low	5
Average	7
High	10

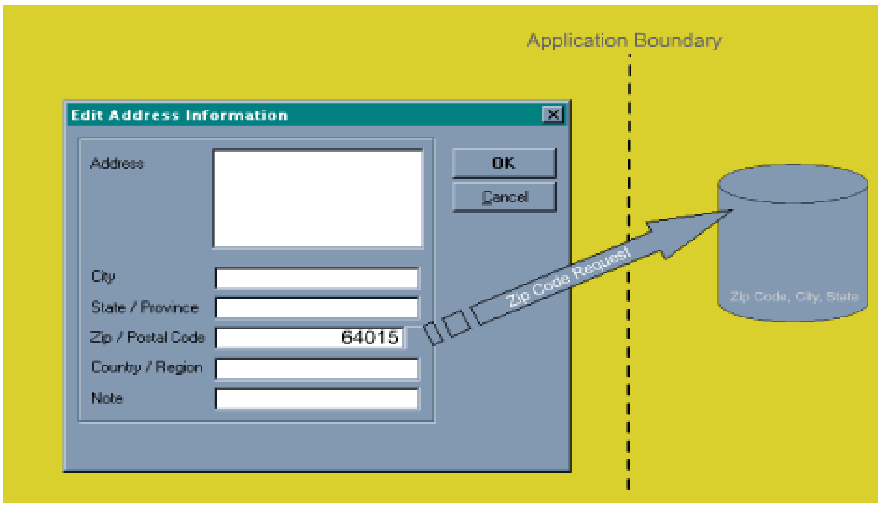
Table 4- EIF Complexity Matrix Weight [27]

According to our example, the complexity is Average, so we can say that, the function points of this ILF will be 7 FPs.

We found real examples which are really useful to understand the transactional and data files. [28]

Now we can consolidate the information about External Interface Files (EIF). Assume that our system goes to another boundary to find and take a Zip/Postal Code then gets full address detail from another system and fills our address detail fields automatically. You can find these examples in **Figure 11** and **Figure 12**.

[28] http://www.softwaremetrics.com/examples/



Edit Address Information

Address

City

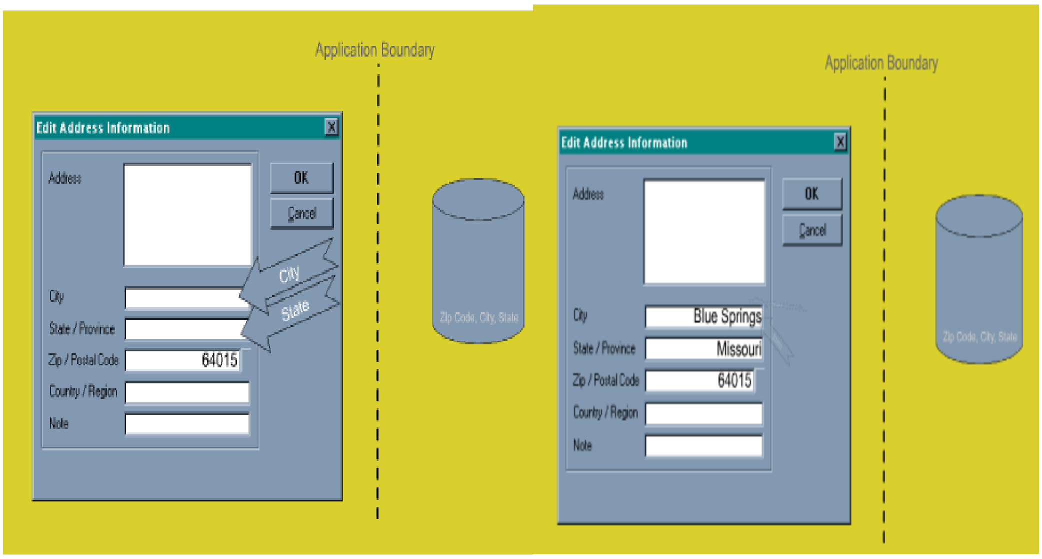
State / Province

Zip / Postal Code 64015

Country / Region

Note

Figure 11- EIF example Part-1: The service goes to another boundary to find and take a Zip/Postal Code.



Edit Address Information

Address

City

State / Province

Zip / Postal Code 64015

Country / Region

Note

Edit Address Information

Address

City Blue Springs

State / Province Missouri

Zip / Postal Code 64015

Country / Region

Note

Figure 12- EIF example Part-2: The service takes Zip Code and states City and State fields automatically.[27]

In this example, we can see the combined effect of an EQ and EIF.

Here, External Inquiry has 3 DETs. These are zip code, city and state fields.

It references just one FTR. Zip code field is an EIF, because it uses another application boundary. Now, we can say that the complexity of this EIF is **Low** and it points to 5 unadjusted function points.

Every EIF must have at least one transaction against it. Also, one of the EI, EO and EQ should refer to EIF.

9.1.2 Transactional Functions

According to Thomas Fetcke, Alain Abran and Tho Hau Nguyen[29] "There can be a one-to-one, one-to-many or many-to-many relation between deliverables visible to the user and transactions". Before describing transactional functions, it would be better to understand the **Figure-13** of these researchers.

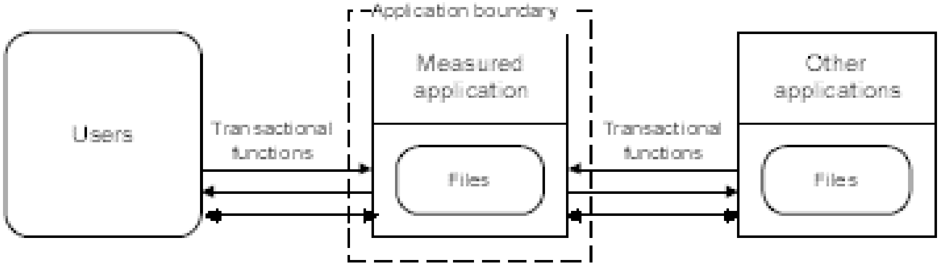


Figure 13- Identification of items within the counting boundary.

(i) External Inputs (EI)

External input processes data or control information which comes from outside the boundary of an application. Data entries and data or file feeds by external applications can be an example to the EI.

To find the complexity of the application we have to determine the DETs and FTRs (File Type Referenced). FTR can be either ILF or an EIF.

FTRs	**Data Element Types (DETs)**		
	1-4	5-15	16+
0-1	L	L	A
2	L	A	H
3 or more	A	H	H

Table 5- EI Complexity Table [27]

[29] Thomas Fetcke, Alain Abran and Tho Hau Nguyen, "Mapping the o-o JacobsonApproach into Function Point Analyses", IEEE,1998

For example, according to the **table-5**, we can say that, if we counted 4 FTRs and 12 DETs that would be a (H) High Complexity.

Complexity	Points/Weight
Low	3
Average	4
High	6

Table 6- EI Complexity Matrix Weight [27]

After finding complexity from the complexity table we can determine the number of FPs that will be counted for this EI. According to our example, the complexity is High, so we can say that, the function points of this EI will be 6 FPs.

We can give another example to consolidate the information about External Inputs. Here is an interface that shows Address Info page on the application.

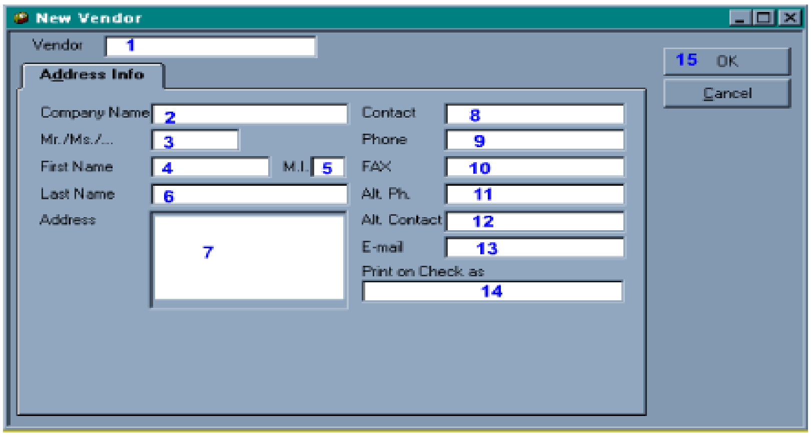


Figure 14- EI example [27]

This external input shown in **Figure-14**,

- Rated as a low EI
- Valued at 3 unadjusted FPs
- FTR is 0

- And has 15 Data elements(DETs) that create one ILF.[30]

You can confirm the complexity is **low** by checking **Table-5.**

According to this example, we have to know that each field that is saved or invokes the transaction will be counted as a data element. We can say that it is an External Input which has 15 Data elements(DETs) that create one ILF.

The OK button here will be counted as a DET. After using OK button, all information will be saved in an ILF file. Now we have a little more information about EI, DEF and ILF.

(ii) External Outputs (EO)

External output is a beginning process that sends data or control information outside the application boundary. According to the *International Function Point Users Group's research* [31] "reports created by the application being counted, where the reports include derived information can be an example to EO."

Allocating FPs to EOs is very similar to the process for EIs.

To find the complexity of the application we have to determine the DETs and FTRs (File Type Referenced) as an EO process.

FTRs	Data Element Types (DETs)		
	1-5	6-19	20+
0-1	L	L	A
2-3	L	A	H
4 or more	A	H	H

Table 7- EO Complexity Table [27]

For example, according to **Table-7**, we can say that, if we counted 0 FTRs and 1 DETs that would be a (L) Low Complexity.

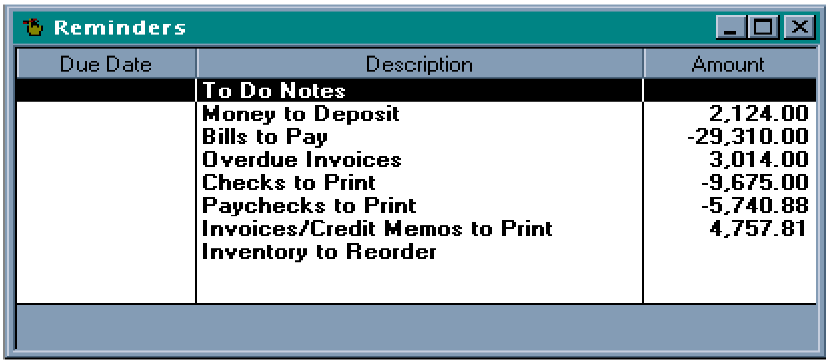
[30] http://www.softwaremetrics.com/examples/ei.htm
[31] www.ifpug.org

Complexity	Points/Weight
Low	4
Average	5
High	7

Table 8- EO Complexity Matrix Weight [27]

According to our example, the complexity is Low, so we can say that, the function points of this EO will be 4 FPs.

Now, we can consolidate the information about External Outputs by giving a real example. This is a 3 data elements (DET's) interface. Also it references 4 FTR's (file types referenced) and would be rated as an **Average** and valued at 5 unadjusted function points.[32] You can confirm the complexity **Average** by checking **Table-8.**



Due Date	Description	Amount
	To Do Notes	
	Money to Deposit	2,124.00
	Bills to Pay	-29,310.00
	Overdue Invoices	3,014.00
	Checks to Print	-9,675.00
	Paychecks to Print	-5,740.88
	Invoices/Credit Memos to Print	4,757.81
	Inventory to Reorder	

Figure 15- EO example [27]

We can say that, the Amount column is a derived data. It is not stated on FTR. Description and Due date data elements can be found on FTR. According to this information we can say that, External Outputs are the calculated data.

The due date does not take apart in this example. So, the due date counted as a DET here.

[32] http://www.softwaremetrics.com/examples/eo.htm

(iii) External Inquiries (EQ)

According to the description of the International Function Point Users Group [30] "An external inquiry (EQ) is an elementary process that sends data or control information outside the application boundary".

To find the complexity of the application we have to determine the DETs and FTRs (File Type Referenced).

FTRs	Data Element Types (DETs)		
	1-5	6-19	20+
0-1	L	L	A
2-3	L	A	H
4 or more	A	H	H

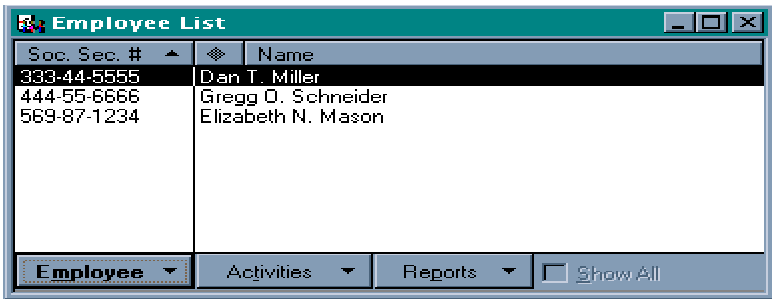
Table 9- EQ Complexity Table [27]

For example, according to the **table-9**, we can say that, if we counted 3 FTRs and 15 DETs that would be a (A) Average Complexity.

Complexity	Points/Weight
Low	3
Average	4
High	6

Table 10- EQ Complexity Matrix Weight [27]

According to our example, the complexity is Average, so we can say that, the function points of this EQ will be 4 FPs. Now, we will consolidate the information about External Inquiries by giving a real example.



Soc. Sec. #	Name
333-44-5555	Dan T. Miller
444-55-6666	Gregg O. Schneider
569-87-1234	Elizabeth N. Mason

Figure 16- EQ example [27]

This EQ has a Name field that has three different data elements. These are First name, Initial name and Last name. (ex: Dan T. Miller).

Social Security number is another data element and comes from Employee (ILF). So we can say that this is a 4 data elements (DET’s) interface. Also it references 1 FTR's (file types referenced) and would be rated as a **Low** complexity and valued at 3 unadjusted function points. You can confirm the complexity of this EQ by checking **Table-9.** The buttons that are named as an Employee, Activities and Reports will be another transactions.

As we mentioned before, it is not possible to get accurate results by using COCOMO models. Today, technology is changing day by day. As a result of this, we have to describe new framework which is useful in SOA environment.

After we got some important information about function point analyses, now we can start to describe the problem of using COCOMO II in SOA environment and Work Breakdown Structure that we use in our development cost framework.

10. THE PROBLEM OF USING COCOMO II IN SOA PROJECTS

In SOA based structures, we can say that there are a lot of loosely coupled services within the project. All of these services have their own communication protocols and well defined interfaces. Each service contains technical details which are technologically independent. It means that the platform and development language are stand-alone in SOA projects.

The main problem of using COCOMO II cost estimation model in SOA based structure is that, although COCOMO II model has scale factors and some other effort multipliers, it is hard to calculate the most reasonable cost estimation for service oriented architecture.

According to Stajanovic and Dahanayake[33]; COCOMO II model is not adequate to estimate effort needed while reusing Service oriented resources. In SOA framework, there are black box services which can be reused and also there are white box services which are ported from legacy systems.

In the research of Sommerville[34], it is mentioned that the black box reuse in the scope of project the difference between code and service level reuse is important. If the code level component is not for reuse, it should be described and understood by using reverse engineering.

[33] Stojanovic Z. and Dahanayake, A. “Service-Oriented Software System Engineering: Challenges and Practices”. Hershey, PA: IGI Global, Apr. 2005.

[34] Sommerville, I. Software Engineering, 8th ed., London: Addison Wesley, Jun. 2006.

According to Oladimeji, Folorunso, Taofeek, Adejumobi[35]; the reusable and loosely coupled service can be reused with service discovery techniques, such as semantic annotation and quality of service.

[35] Oladimeji Y. L., Folorunso O., Taofeek A.A, Adejumobi A. I,Nigeria,September, "A Framework for Costing Service-Oriented Architecture (SOA) Projects Using Work Breakdown Structure (WBS)", 2011, Global Journals Inc. (USA)

11. WORK BREAKDOWN STRUCTURE (WBS)

Work breakdown is a concept that separates operations into small manageable sections. It is a hierarchical decomposition of the project/scenario to be performed by the project team. All sub-categories within the work breakdown structure provide more details and definitions.

The key point of using WBS in the subject project is to process all sub-categories for finding the overall effort and cost estimation of the project. Finding total cost and effort starts from the smallest piece of sections then continues to upper stages. In this way, the total effort and cost estimations can be calculated without any leakage.

Today a lot of companies use WBS structure to manage their projects. This hierarchical structure provides to manage all sections deeply for project team to understand these sections clearly.

In the figure below, it is possible to see the structure of WBS.

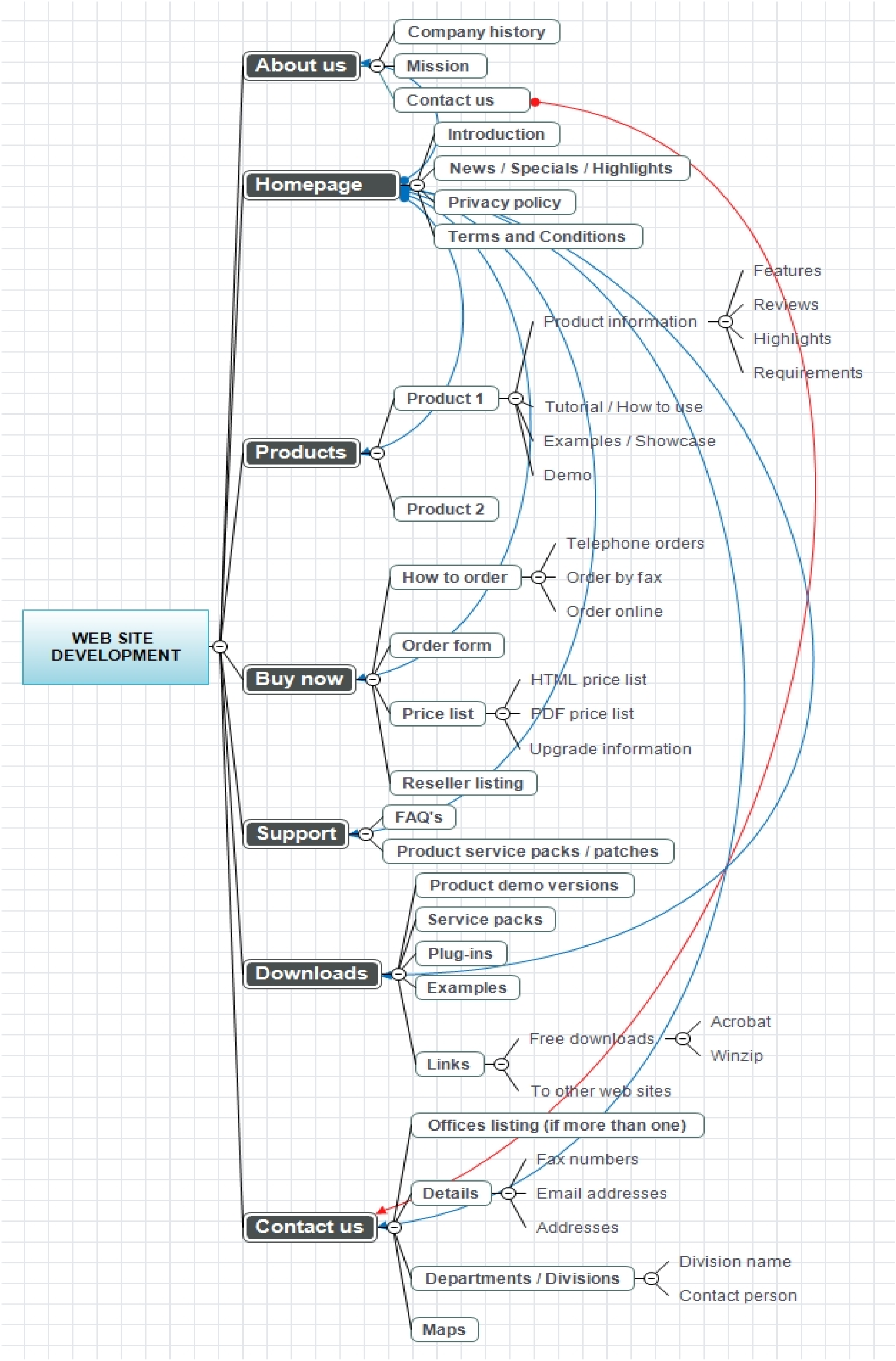


Figure 17- Example of WBS

In Figure 17**;** as the top level represents the entire “Web Site Development” project, in the lower levels it is possible to see components' parts in detail. While creating a new work breakdown structure, it is essential to make all sub-categories' outcomes and deliverables measurable.

According to the structure of WBS, project team identifies the main functional deliverables. After this process, project team divides these functional deliverables into small sub-deliverables.

The granularity of all sub-deliverables must be assignable to one person. All tasks within the sub-categories need to be completed by a specific time and effort that is given by the project managers.

In cost estimation projects, using this structure, it is possible to assign all main processes to specific departments which make projects more manageable.

It is expected that the costs and efforts may change during the project life cycle while the main objectives remain constant.

Second and other levels of the WBS structure must be created carefully by the managers as well as the planners of the project.

Upper levels must be completed before creating the bottom levels. It means that, creating WBS structure starts from up levels and continues to bottom levels which are totally reverse to the logic of the cost and effort estimation process in WBS.

11.1 Benefits of using Work Breakdown Structure

As it is mentioned in the previous section, WBS makes project more manageable with its detailed structure.

- By using WBS in the project management, it is possible to divide project into sub-categories which makes easier the calculation of the budget for every departments throughout the company. By dividing cost and time estimates to particular sections it becomes easier to develop project schedule and budget.

WBS helps project budget plans and schedule to be created effectively.

- By using WBS it is easier to control people who work within the project by controlling assigned tasks for each person. Responsibilities can be assigned to people easily.
- Changes during the project life cycle it is possible to add new tasks and sub-categories to the project. In this way, it is possible to update project changes while working on the project.
- Well-developed WBS can be used in the future projects.

11.2 Cost Estimation Using Work Breakdown Structure in SOA

WBS helps project managers to estimate the cost of projects easily. The performance and budget estimations can be found by assigning pre-planned time-based cost estimation to all WBS elements.

In today's world, SOA became more popular in the technology sector. Thus, the calculation of big projects became tough.

For the medium and small scaled projects COCOMO and other cost estimation models were able to calculate the most reasonable results in cost estimation. Then they updated main COCOMO model into COCOMO II model for enabling old model to SOA projects.

Stroulia and Tansey[36] have tried to use COCOMO II in their research to make estimation of cost for creating and migrating services.

[36] Tansey, B. and Stroulia, E. "Valuating Software Service Development: Integrating COCOMO II and Real Options Theory," The First International Workshop on the Economics of Software and Computation, IEEE Press, May 2007, pp. 8-8, doi: 10.1109/ESC.2007

At the end of their research, they indicated that the cost estimation with COCOMO II should be extended to be able to find the most reasonable results in new characteristics of SOA based development and COCOMO II model was inadequate for calculating the total cost estimation.

12. SIZE AND COST ESTIMATION OF SERVICE ORIENTED ARCHITECTURE (SOA)

All cost estimation models up to now find the cost by using case points, function points, line of codes etc.

In SOA methodology, the sizing method has to be customized. Function points can be described as functionalities provided to the user by using inputs, outputs, inquires, and files.

SOA based software architectures differ from other traditional software programmes. In SOA based project, the service function should show a real-world self-contained business activity.

There are problems that occur when applying IFPUG to software system size measurement. For example, the effort of wrapping legacy code and data to work as services cannot be assigned to any functional size. [40]

12.1 Previous Researches & Frameworks about Size and Cost Estimation of SOA

According to Linthicum[37] time is one of the main factors that affects cost estimation in SOA based software. It has been said that the equations below can be used for project management as well as project costing method;

[37] Linthicum, D., "How Much Will Your SOA Cost?" http://www.soainstitute.org/articles/article/article/how-muchwill-your-soa-cost.html, March 2007.

- The Cost of the SOA = (Cost of Data Complexity+ Cost of Service Complexity + Cost of Process Complexity + Enabling Technology Solution)

Then, he gives the equation of “Cost of Data Complexity”;

- Cost of Data Complexity = (Number of Data Elements) * (Complexity of Data Storage Technology) * (Labor Units)

-Where the Complexity of the Data Storage Technology: is shown as a percentage from 0% to 100% where Relational is 30%, Object-Oriented is 60% and ISAM is 80%

-The Labor unit: is the amount of money it takes for understanding and refine one data element.

There is no additional information about Linthicum's equations that proves the results of these formulations are reliable.

In the research of Conte, Iorio, Meli, and Santillo[38]; they claimed that, the size of the SOA based software can be found by using the service points instead of function points. Thus, the size estimation of SOA based project can be calculated by summing up the size of service points.[39]

$$Size = \sum_{i-1}^{n}(P_i \times P) \qquad \textbf{(9)}$$

Equation 9- Size Estimation of SOA based project

The expression is given in Equation 9 is Size of SOA based project Equation where;

Pi : Infrastructure Factor (This is an experimental value)

[38] Conte M., Iorio T., Meli R., and Santillo,“An early and quick approach to functional size measurement methods”. Software Measurement European Forum (SMEF). Rome, Italy, 2004.

[39] Liu, J., Xu, Z., Qiao, J., and Lin, S. “A Defect Prediction Model for Software Based on Service Oriented Architecture using EXPERT COCOMO,” Chinese Control and Decision Conference (CCDC), IEEE Press, Jun. 2009, pp. 2591-2594

P: is an estimated size of a single service.

Infrastructure factor can be described as a set of technology, management processes and supporting infrastructure.

Some services can be created from existing resources while others can be built from scratch. There are also existing services. All of these describe the type of services. Different types of services make harder to calculate the size estimation in SOA based projects.

After the research of Conte, Iorio, Meli, and Santillo, O'Brien described SMAT-AUS framework in his research[40]. They give the details of this framework and its effects on the scope, cost and effort of services. SMAT-AUS framework discloses technical, social, cultural and organizational dimensions of SOA project. Service mining, service development, service integration and SOA application development are classified as separate SOA project types when applying SMAT-AUS framework. They used several methods, cost models, templates and also functions to assist cost and effort estimation process for each project time. Then, they used that project times to calculate the total cost of SOA based project.

Oladimeji, Folorunso, Taofeek, Adejumobi created a new framework by using work breakdown methodology while calculating the cost estimation of the SOA based project. They allocated SOA based software development cost estimation into sub-categories as shown below [40],

- The cost of service discovery
- The cost of service migration (service wrapping)
- The cost of service development
- SOA Cost Estimation(component service)
- The cost of service integration for component services in the highest level (top level) service.

[40] O' Brien, L. "A Framework for Scope, Cost and Effort Estimation for Service Oriented Architecture (SOA) Projects" Software Engineering Conference" IEEE Press, April 2009.

Total cost estimation can be found by summing all of these elements. Li and Keung described another framework that uses divide and conquer approach[41]. They used several metrics within their framework study. By using divide and conquer approach they tried to find the overall solution by dividing size of the SOA project problems into sub-problems. By decomposing "problem of size N" into "sub-problems" they tried to solve each sub-problem.

According to principle of divide and conquer approach, it is possible to produce "sub-solution" for each sub-problem. By composing each sub-solution, it is possible to find the overall solution. Li and Keung's work concentrating on cost estimation for service integration. They focused on the development process in their research by using 4 different service types. These types are [46];

- ***Available Service*** (basic service type): This is the service already existing. For instance, service provided by a third party or inherited from legacy SOA based systems.
- ***Migrated Service*** (basic service type), is the service to be generated through modifying or wrapping reusable traditional software component(s).
- ***New Service*** (basic service type), is the service to be developed from scratch.
- ***Combined Service:*** This is the service which is created by the combination of "Available Service", "Migrated Service", "New service." or other combined services.

[41] Li, Z., Keung Jacky "Software Cost Estimation Framework for Service-Oriented Architecture Systems using Divide-and-Conquer Approach" Sydney, Australia IEEE Press, September 2010

13. HOW TO DESCRIBE FUNCTION POINTS IN SOA BASED PROJECT'S COST ESTIMATION

13.1 A New Solution For Cost Estimation in SOA

After the needs of SOA projects' cost estimation increased, some researchers have analysed new frameworks and new calculations to become familiar with the current technologies. When it comes to services, it was hard to use function points in their calculations. Thus, they studied on projects to find the cost of the services not only the requirements on the project. They described new frameworks which are usable for different types of services.

In our study we will use work breakdown structure while calculating the development cost of the SOA project. We will try to find development cost for optimal granularity of services by using the work breakdown structure (WBS). We will use WBS while we are calculating the development cost for every single services that we created by Monte Carlo Simulation method. Then we will see if this framework gives us detailed cost of service development results in a SOA project.

In service oriented architecture it is faster to develop and implement IT and it lowers the development cost. The important question is that: “How services should be modelled to make them flexible?” There is no accurate solution to answer this question.

14. SERVICE GRANULARITY IN SOA

SOA based-projects provide reusability and composability of services within the project. A high level of granularity provides system to be flexible and extendible. At Papazoglou et al.'s research[42], he described service granularity as the unit of modularity of a system. Modularity means the amount of functionality of service.

Another researcher Feuerlicht and Wijayaweera[43] mentioned that, coarse grained services implement high level business functions while fine grained services implement a single atomic operation which exchange limited amounts of data in the SOA project.

Today, most researchers describe the importance of finding a right service granularity in SOA while others mention the importance of optimum service granularity.

[42] Papazoglou M., Van den Heuvel W. "Service-Oriented Design and Development Methodology". International Journal of Web Engineering and Technology 2006

[43] Feuerlicht, G., Wijayaweera, A. "Determinants of Service Reusability, New Trends in Software Methodologies, Tools and Techniques", IOS Press, 2007

15. FINDING DEVELOPMENT COST IN A SOA PROJECT

As we mentioned before, there were several approaches to the SOA cost estimation. Some of them tried to find overall cost while others try to find new solutions for management cost or development cost.

Current research shows the cost estimation of SOA project is still argumentative. By considering previous studies carefully, we will try to find development cost in our research while using work breakdown structure to simplify the estimation of development cost.

We will do our research based on 3 metrics and we will use these service points in our calculation:

- **Cluster Size**
- **Structural Complexity**
- **Number of Input and Output (I/O)**

In the next section, we will describe these metrics in detail.

In our case study, we will have several services. We will cluster these services into number of groups. The number of groups can be decided by the Monte Carlo simulator user. Clustering these services into sub-groups will give us different grouping sizes and different development costs with new services. We will start to calculate development cost according to WBS work flow. We will use mentioned metrics: Cluster size, Structural complexity and number of I/O while we are calculating the development cost of new services.

15.1 Cluster Size Metric

Cluster size will be measured as a number of services that is included by one cluster. By grouping clusters, we will obtain different fine grained or coarse grained service sets which give us different development costs according to their cluster size. By clustering services into several groups it is possible to create new services with different sizes.

Using work breakdown method in our research, cluster groups which are created by using Monte Carlo simulation programme may include one or more services. For instance, if we try to cluster services into two groups, it is possible for first group to include just one service while another group includes all of the other services throughout the process.

This will change the cluster size and service complexity. We will give examples of different scenarios to have different values which help us to find the optimal service clustering.

By creating new services with clustering may give us different development costs. For instance; if we put all services into one group, it will increase the line of code (LOC). Thus, complexity will be increased as well as development effort will be higher. Furthermore, if cluster size rises, number of I/O will be decreased oppositely.

15.2 Structural Complexity Metric

For calculating the service development cost, we can use structural complexity as another metric. We can say that, line of code can be the main factor of the structural complexity metric. We will use LOC size while we are calculating the complexity.

15.2.1 Line of Code (LOC)

In our case study Line of Code will be measured as a number of lines that is written by developers per service. The development effort will be increased when the Project's line of code increases. Increase of the LOC will affect the development cost. More LOC means more effort and more complexity. LOC may be different in different programming languages for the same program.

For instance, developing in C may be shorter than developing in Cobol. Therefore, programming language may affect the number of lines while estimating the project cost.

15.3 Number of Input and Output (I/O) Metric

I/O size can be measured by the number of input and output within the service. Having more I/O means more complexity of a service. In other words, the development cost will be less if there is not a lot of I/O in the service.

The table below shows how metrics may affect each other and our main service metrics.

IF	**Cluster Size**	↑	*THEN*	**LOC**	↑	*EXPECTED RESULT*	**DEVELOPMENT COST**	↑
IF	**Cluster Size**	↑	*THEN*	**I/O Size**	↓	*EXPECTED RESULT*	**DEVELOPMENT COST**	↓
IF	**LOC**	↑	*THEN*	**Structural Complexity**	↑	*EXPECTED RESULT*	**DEVELOPMENT COST**	↑
IF	**Structural Complexity**	↑	*THEN*	**LOC**	↑	*EXPECTED RESULT*	**DEVELOPMENT COST**	↑
IF	**I/O Size**	↑	*THEN*	**Structural Complexity**	↑	*EXPECTED RESULT*	**DEVELOPMENT COST**	↑

Table 11- How metrics affect each other

16. CASE STUDY: FINDING OPTIMAL SERVICE GRANULARITY BY CALCULATING DEVELOPMENT COST IN A SOA PROJECT

In our case study we will try to find the optimal service granularity by calculating the development cost. Development cost of a new service will be calculated by using Service line of code, service I/O size, and structural complexity.

We will try to create new services by granulating services into several groups. These groups will be named as "cluster". We will create 100 samples of each cluster to find different results. According to these results our simulation programme will find the optimal service which is produced with the lowest development cost.

Now we will give some information about Monte Carlo Method that we used during generating random numbers in our experiments for finding the optimal service granularity.

17. MONTE CARLO METHOD

Monte Carlo Methods are based on the use of random numbers and probability statistics to investigate problems. Monte Carlo Methods are used in many areas from finance to physics. Obviously, they are applied in variety ways from field to field. But, definitely, to call something as a Monte Carlo experiment you need to investigate the problems.

A large system can be sampled in a number of random configurations by using Monte Carlo Methods. So, the data which has been extracted can be used to define the system as a whole.

We use Monte Carlo Method in our experiments while sampling clusters with random services. Also, we distribute services into clusters with random services and random number of services. The main patterns of Monte Carlo Method may contain following steps:

STEP 1: Define possible inputs
STEP 2: Generate random inputs from possible combinations over the set.
STEP 3: Performing deterministic computation on the given inputs.
STEP 4: Accumulate the results.
Assume that we will calculate the cost of a project in SOA.

For estimating the cost, we will probably need to know the factors which may affect the cost. These factors can be defined by user manually.

In the next step, we should generate inputs randomly from the domain. The inputs and defined factors will be used by Monte Carlo Simulation to generate different variations which helps to obtain possible conditions or states.

On the following step, we should perform a deterministic computation using our inputs. Different states may give us different results while estimating the software cost.

In the final step, we should accumulate our individual inputs. To bring every little data together, we have to calculate every possible domain and the possible cost scenarios.

This simple sample describes how to use Monte Carlo methods in our case study. As we talked about cost calculation, we can also make other calculations by using different frameworks or methods.

17.1 History of Monte Carlo Method

The development of Monte Carlo method started in 1940's. Many reserchers used Monte Carlo Methods in their researches.

Neumann, Metropolis and other researchers started to systematic development of Monte Carlo Method after years. They did some statistical sampling using Computing Techniques. After years, in 1953, Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller described Metropolis Algorithm. [44]

Stanislaw Ulam used Monte Carlo Method for calculating complicated mathematical integrals in the theory of nuclear chain reactions.

[44] Nucleonica, " Introduction to Monte Carlo method", European Commission Directorate –General Joint Research Centre, 2008

Rosenthal used Monte Carlo algorithms for parallel computing in 2000. They showed that the Monte Carlo algorithms are useful in parallel computing.[45]. At the end of their experiments, they found that is not hard to run Monte Carlo and Markov chain Monte Carlo in parallel computing.

Researchers believe that, Monte Carlo method will become popular in the future when computers become more numerous and better networked. [46]

[45] J.S. Rosenthal, “Analysis of the Gibbs sampler for a model related to James-Stein estimators” Statics and Computing, 6, 269-275, 1996

[46] J. S. Rosenthal, “Parallel computing and Monte Carlo algorithms”, Far East Journal of Theoretical Statistics 4, 207–236, 2000

18. CASE STUDY: FINDING DEVELOPMENT COST OF A SERVICE

18.1 Motivation

In our research we will try to find optimum service granularity by clustering services into small pieces. By using Monte Carlo Simulation method we will generate random services with different inputs and outputs then we will be able to calculate the development cost of every single service to find optimal service granularity with accurate results. In our framework, we will use service points instead of using function points. You can find the metrics that we have used during our experiments in **Section 15**. We will use these metrics in our framework and calculations.

At the end of our research we will try to find the optimal clustering size that gives the most reliable development cost of a service and we will be able to see the effect of granularity on service development cost.

18.2 SOA Development Cost Framework & Scenario

Framework that we created is based on three main metrics Cluster size, Structural complexity and I/O size. Now we will describe how we used these service points in our framework.

We assumed that there is a scenario with 50 services. We assigned random complexity value for all services which can be a decimal number. Actually, this complexity value corresponds to LOC size for each service.

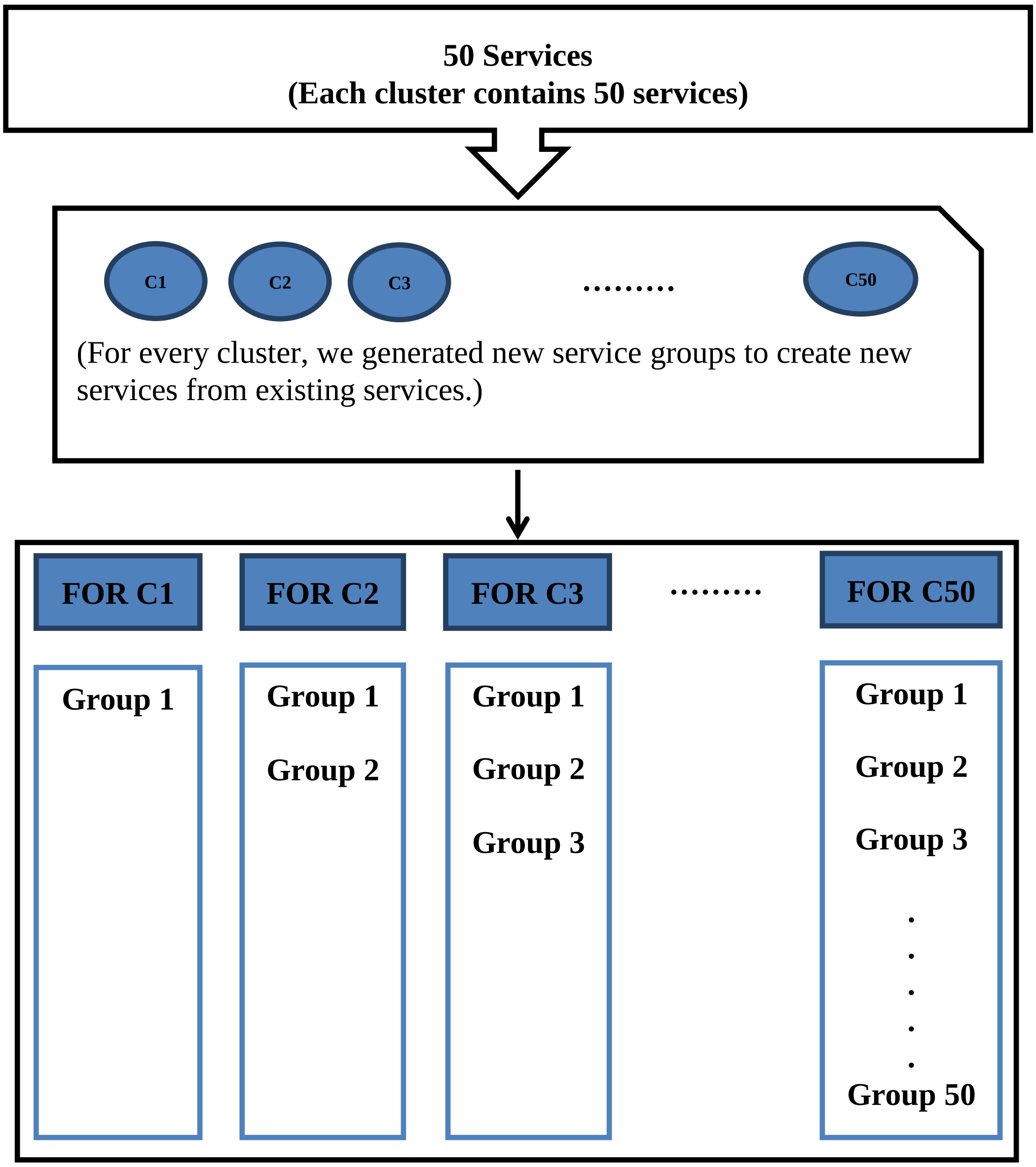

50 Services
(Each cluster contains 50 services)
C1
C2
C3
....
C50
(For every cluster, we generated new service groups to create new services from existing services.)
FOR C1
FOR C2
FOR C3
....
FOR C50
Group 1
Group 1
Group 2
Group 1
Group 2
Group 3
Group 1
Group 2
Group 3
Group 50

Figure 18- Service Clustering

As you see in **Figure 18**, we have used 50 services for service clustering. All clusters contain one or more groups that use our existing services which are distributed randomly into these groups. We assumed that there are 30 inputs and 25 outputs in our project. **Figure 19** illustrates the main steps of our framework.

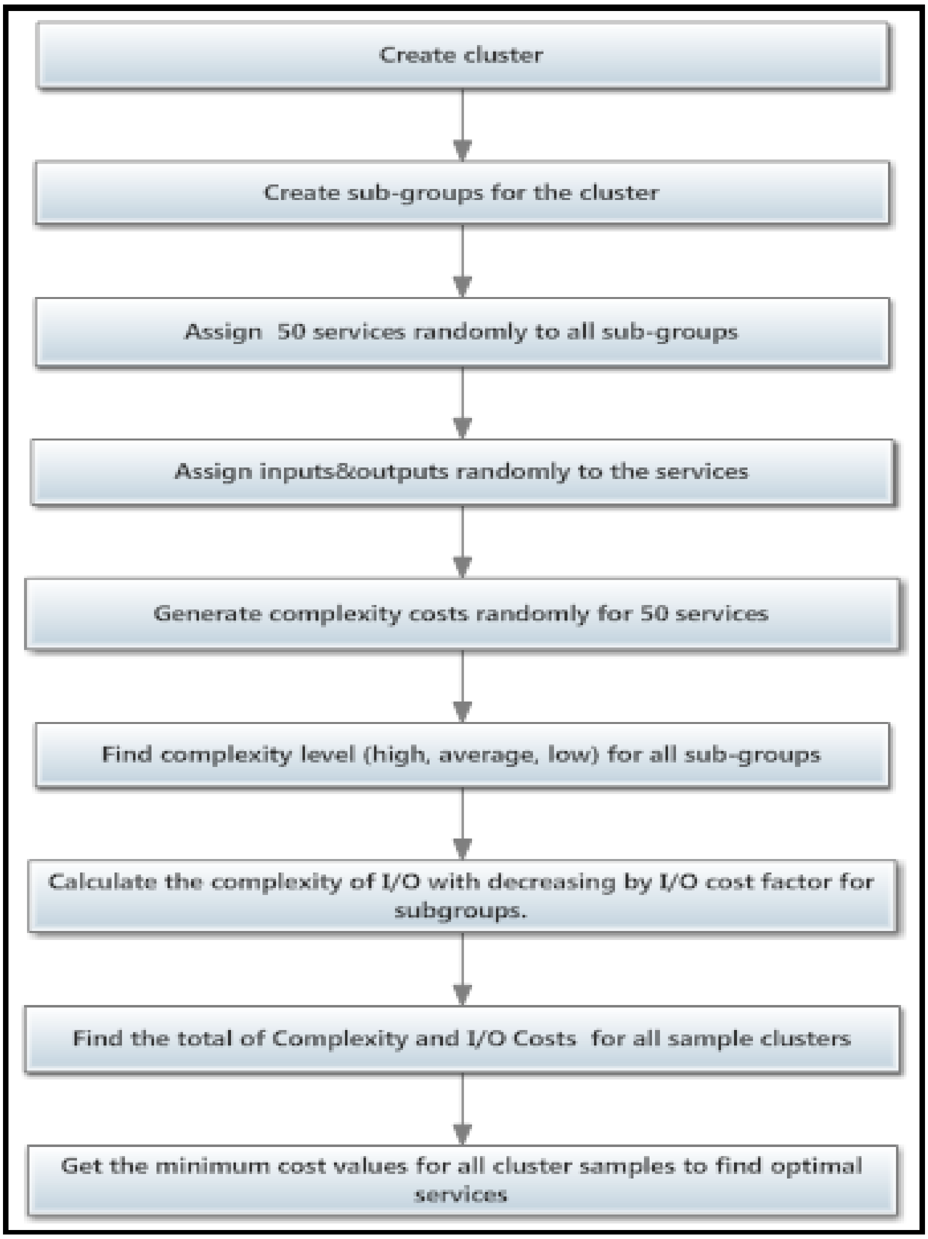

Create cluster
Create sub-groups for the cluster
Assign 50 services randomly to all sub-groups
Assign inputs&outputs randomly to the services
Generate complexity costs randomly for 50 services
Find complexity level (high, average, low) for all sub-groups
Calculate the complexity of I/O with decreasing by I/O cost factor for subgroups.
Find the total of Complexity and I/O Costs for all sample clusters
Get the minimum cost values for all cluster samples to find optimal services

Figure 19- Main steps of our framework

18.2.1 How to Estimate Complexity

We created 3 complexity categories for calculating the complexity of services. These are low, high and average.

	Low	**Average**	**High**
Number of services in cluster	1-5	6-45	46-50
Complexity Factor	1.10	1.13	1.15

Table 12- Complexity Ranges and Factors

First of all, we check the number of services inside the cluster. Then the complexity of this cluster is found if it is low, average or high. For instance, in cluster-4 there are 4 groups. The total of services is always 50. When we look inside these groups, we can see that the existing services are distributed into these four groups randomly. Each group contains different or the same number of services. We can describe it clearly;

Cluster 4

Group1 {22, 1, 30, 2, 37, 3, 46, 4, 32, 5, 47}
Group 2 {21, 6, 48, 7, 8, 23, 24, 30, 31, 25}
Group 3 {35, 9, 40, 10, 11, 41, 12, 13, 49, 14, 34, 15, 38, 16, 42, 17, 18, 43, 19, 20 28, 45, 29, 33, 39, 44}
Group 4 {50, 26, 27, 36}

As you see above, **Group 1** contains 11 services which make cluster complexity "**Average**". If the complexity of a group is average then we have to use 1.13 as a complexity factor of this group. As we said before, we assigned random complexity value for all services which can be a decimal number between 2 and 4. For our example in Group 1, assume that;

For 22 the complexity value is 2
For 1 the complexity value is 3.3
For 30 the complexity value is 2.1
For 2 the complexity value is 4
For 37 the complexity value is 3.8
For 3 the complexity value is 2.5
For 46 the complexity value is 3
For 4 the complexity value is 2
For 32 the complexity value is 3.7
For 5 the complexity value is 2.2
For 47 the complexity value is 3.9

If you look into the groups of cluster 4,

The Complexity Factor of Group 1 is 1.13 (Average)
The Complexity Factor of Group 2 is 1.13 (Average)
The Complexity Factor of Group 3 is 1.13 (Average)
The Complexity Factor of Group 4 is 1.10 (Low)

So the Complexity of Group 1 will be;

Group 1 Complexity= (Total Service Complexity Value of Group 1)* Complexity Factor of Group 1

Group 1 Complexity= (2+ 3.3+ 2.1 + 4+ 3.8 +2.5+ 3+ 2+ 3.7+2.2+3.9) * 1.13 = **36,725**

If we do the same calculations for Group 2, Group 3, Group 4; we will find different group complexities. Then we can find the total complexity of Cluster 4. So we can calculate it like this:

CLUSTER 4 TOTAL COMPLEXITY= ((Total Service Complexity Value of Group 1* Complexity Factor of Group 1) + (Total Service Complexity Value of Group 2* Complexity Factor of Group 2) + (Total Service Complexity Value of Group 3* Complexity Factor of Group 3) + (Total Service Complexity Value of Group 4* Complexity Factor of Group 4))
These calculations continue until all cluster’s complexities are found.

We generate 100 samples for each cluster to find optimal service with reliable development cost. So, we will have 5000 samples in our experiments.

We choose the cluster with minimum complexity in 100 samples. Think that, Sample 5 has the minimum complexity in our 4th Cluster. We do this process for all cluster samples. We take Sample 5 and other samples with the minimum complexity into account while we are calculating total development cost. This information can be found in the experiment section of this paper.

18.2.2 How to Estimate I/O Cost

In the framework we defined, there are 4 service types. These are;

- NO Input OR Output
- NO Input JUST Output
- JUST Input NO Output
- BOTH Input AND Output

We assigned different inputs and outputs for all services that we defined at the beginning of our project. We have used 4 different service types as we mentioned before. Some of our services do not contain input and output. This provided us to find different I/O costs during our experiments.

Also, we determined an I/O factor which affects the I/O cost of a service. There are some probabilities while clustering services into several groups. For instance, some groups may have services which have same inputs. On such an occasion, there is no point to define the same inputs or outputs twice. Thus, this will be a profit for reducing our development cost. We should take into consideration this issue while we are calculating the I/O cost. So, we thought that we have to make our calculations in this order:

Firstly, we find the total of all inputs and outputs within the cluster. Then we compare all inputs and outputs within the group to know if there are same inputs or outputs in different services. For instance;

Cluster 4 for Group1 {11, 24}
For service 11, input set is {1, 5} and output set is {10, 21, 7, 12}

For service 24, input set is {1, 13, 5, 7} and output set is {20, 10}
Now, we will compare service 11 and service 24's I/O values.

Common I/O for service 11 and 24 is; Input {1; 5} and Output {10}

We do this process for all services within the group. The Input and Output factors decrease if there is common I/O in the group.

In our first experiment, an initial factor of Input and Output is 0, 90. We decrease this factor if there are common inputs or outputs. So, at the end of calculations, input and output values can be different.

Now, we will use the same example that we used above;
There were two common inputs and one output in Group 1, {1; 5} for inputs and {10} for output.

Input Cost Factor calculation for Group 1:

Common Inputs: {1; 5}→ 2 common inputs.

Input Decrease Factor= 0, 1

If the common inputs are more than one; for each common inputs that follow the first common input, we divide decrease factor by multiple of two (twin or couple).

0, 90- (0, 1 + (0, 1÷2)) = 0, 75

Group 1 Input Cost= (Total inputs of Group 1) *(Input Cost Factor)

= 6* 0, 75= 4, 5

If there were 3 common inputs then we have to calculate it with this equation;

0, 90- (0, 1 + (0, 1÷2) + (0,1÷4)) = 0, 725

Output Cost Factor calculation for Group 1:

Common Outputs: {10}→ 1 common output.

Output Decrease Factor= 0, 1

If the common outputs are more than one; for each common outputs that follow the first common output, we divide decrease factor by multiple of two (twin or couple).

0, 90- (0, 1) = 0, 80

Group 1 Output Cost= (Total outputs of Group 1) *(Output Cost Factor)

= 6* 0, 80= 4, 8

Group 1 I/O Cost= Group 1 Input Cost + Group 1 Output Cost=

= 4, 5 + 4, 8= 9, 3

We should perform the same calculation for all groups within the cluster to find the total cluster I/O Cost.

18.3 Prerequisites

During our experiments we have used:

- Method 1 : Monte Carlo Model
- Method 2 : Work Breakdown Structure
- Computer Processor: Intel Core i5, 2.27 GHz
- RAM: 6GB
- Computer Operating System: Windows 7, 64-bit
- Server (For experiments): Intel Xeon 2.40 GHz, 4GB RAM. Operating System is Windows 2008 64-bit
- Microsoft Excel for reporting
- Development environment: MS Visual Studio 2010
- Programming Language : C#
- Database: SQL Server 2008

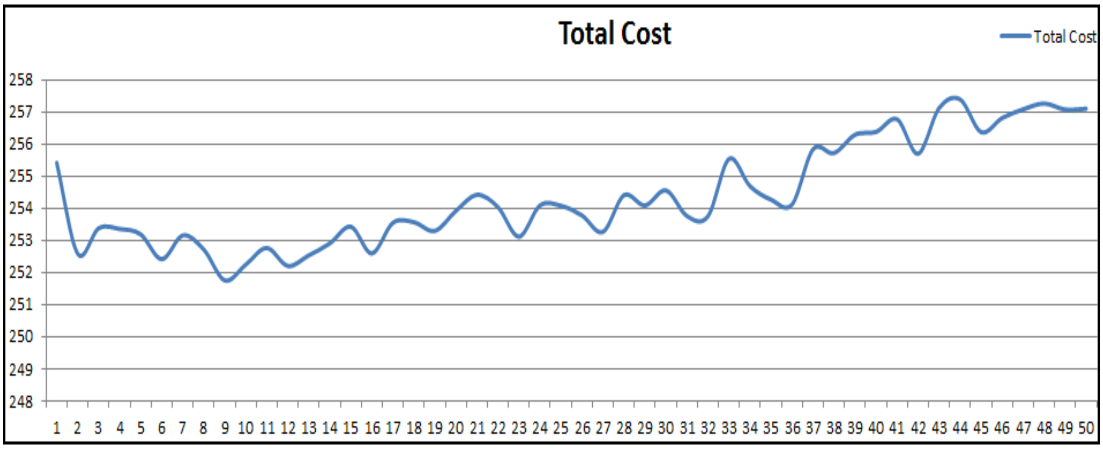
18.4 Experiment 1

Duration: 9 Hours (Generating clusters and distribute services into clusters)

Description: This example illustrates the effect of granularity on service development cost.

Parameters & Factors:

Number of Samples	100		
No of Services	50		
Complexity Cost Range	2-4		
Complexity Cost Factor	1		
Complexity Levels	LOW	AVG	HIGH
	1.1	1.13	1.15
IO Cost Factor	1		
IO Cost Levels	If the common inputs or outputs are more than one; for each common inputs and outputs that follow the first common input or output, we divide decrease factor by multiple of two (twin or couple) for each common inputs and outputs.		



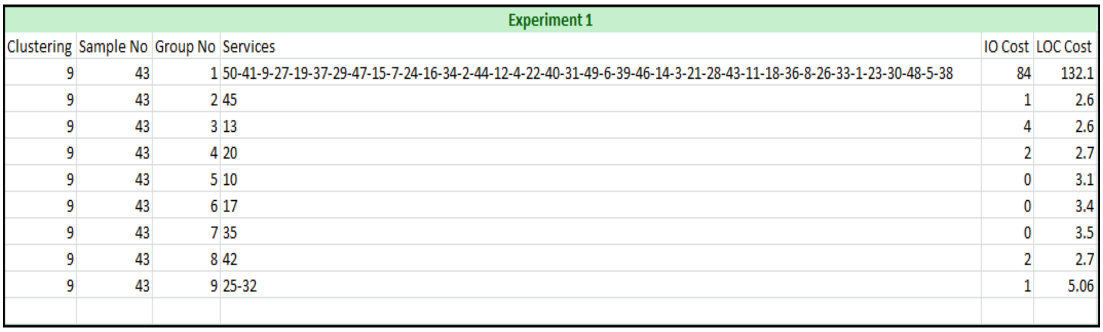
Graph 1- Experiment 1 (Total Development Cost of 50 clusters)

In graph 1, x axis illustrates the clusters that we have used during our experiments while y axis shows the total development cost of these clusters.

Result:

As a result we can say that;

- Cluster 9 is the optimal cluster which has the lowest development cost with 251, 76.
- Cluster 44 has the highest development cost with 257, 39.
- The difference between these two clusters is 5, 63.
- Development cost line goes crinkly. So, we can say that clustering services affect the development cost but there is not a significant difference between clusters to take into account.



Experiment 1					
Clustering	Sample No	Group No	Services	IO Cost	LOC Cost
9	43	1	50-41-9-27-19-37-29-47-15-7-24-16-34-2-44-12-4-22-40-31-49-6-39-46-14-3-21-28-43-11-18-36-8-26-33-1-23-30-48-5-38	84	132.1
9	43	2	45	1	2.6
9	43	3	13	4	2.6
9	43	4	20	2	2.7
9	43	5	10	0	3.1
9	43	6	17	0	3.4
9	43	7	35	0	3.5
9	43	8	42	2	2.7
9	43	9	25-32	1	5.06

Figure 20-Optimal Services of Experiment 1 with I/O and LOC Costs

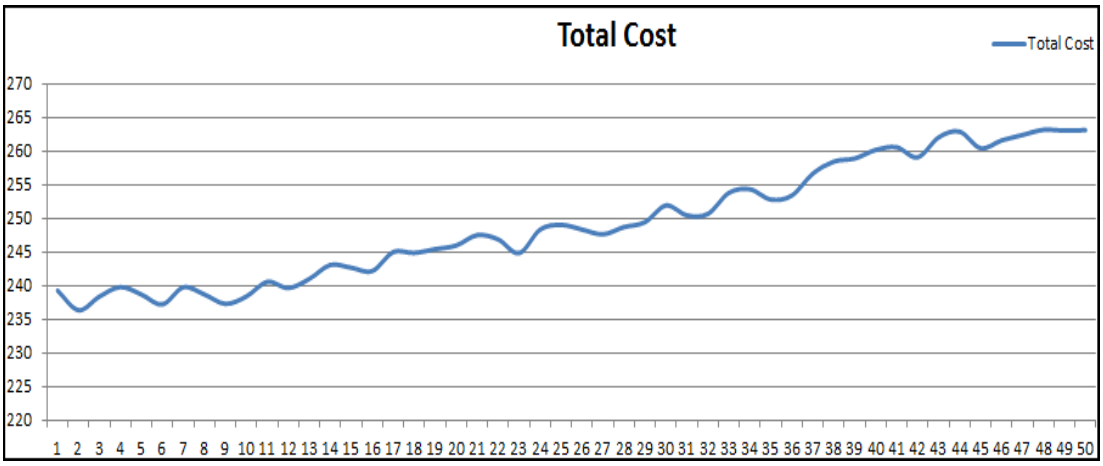
18.5 Experiment 2

Duration: 9 Hours (Generating clusters and distribute services into clusters)

Description: This example illustrates the effect of granularity on service development cost. We changed the I/O Cost levels calculation method.

Parameters & Factors:

No of Samples	100		
No of Services	50		
Complexity Cost Range	2-4		
Complexity Cost Factor	1		
Complexity Levels	LOW	AVG	HIGH
	1.1	1.13	1.15
IO Cost Factor	1		
IO Cost Levels	If the common inputs or outputs are more than one; for each common inputs and outputs that follow the first common input or output, for the first 3 inputs and outputs decrease factor is 0, 1. If there are more common inputs and outputs (more than 3), we divide decrease factor by multiple of two (twin or couple) for each common inputs and outputs.		



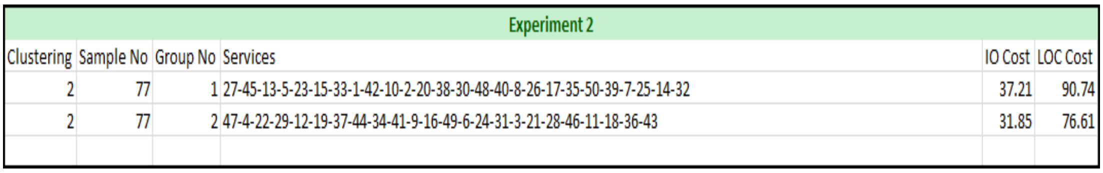
Graph 2- Experiment 2 (Total Development Cost of 50 clusters)

In graph 2, x axis illustrates the clusters that we have used during our experiments while y axis shows the total development cost of these clusters.

Result:

As a result we can say that;

- Cluster 2 is the optimal cluster which has the lowest development cost with 236, 41.
- Cluster 48 has the highest development cost with 263, 16.
- The difference between these two clusters is 26, 75.
- Development cost line increases constantly. So, we can say that clustering services do not affect the development cost significantly.



Experiment 2

Clustering	Sample No	Group No	Services	IO Cost	LOC Cost
2	77	1	27-45-13-5-23-15-33-1-42-10-2-20-38-30-48-40-8-26-17-35-50-39-7-25-14-32	37.21	90.74
2	77	2	47-4-22-29-12-19-37-44-34-41-9-16-49-6-24-31-3-21-28-46-11-18-36-43	31.85	76.61

Figure 21- Optimal Services of Experiment 2 with I/O and LOC Costs

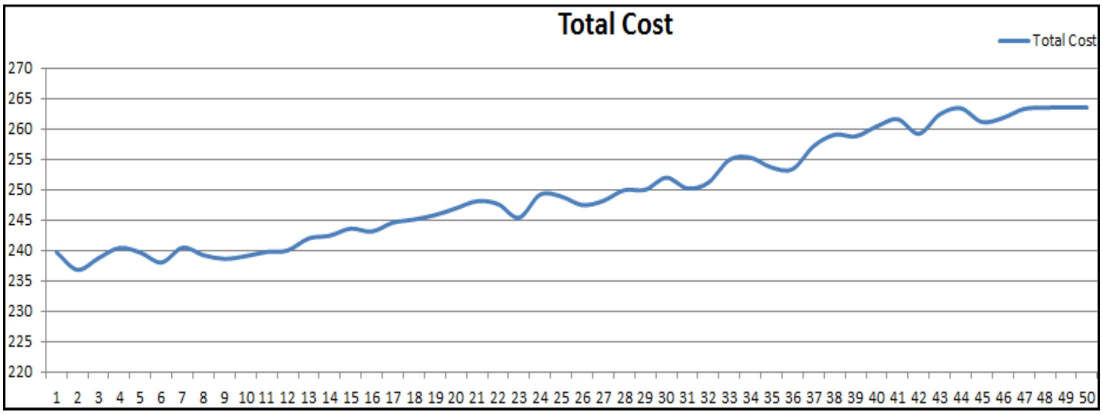
18.6 Experiment 3

Duration: 9 Hours (Generating clusters and distribute services into clusters)

Description: This example illustrates the effect of granularity on service development cost. We changed the complexity cost range.

Parameters & Factors:

No of Samples	100		
No of Services	50		
Complexity Cost Range	1-5		
Complexity Cost Factor	1		
Complexity Levels	LOW	AVG	HIGH
	1.1	1.13	1.15
IO Cost Factor	1		
IO Cost Levels	If the common inputs or outputs are more than one; for each common inputs and outputs that follow the first common input or output, for the first 3 inputs and outputs decrease factor is 0, 1. If there are more common inputs and outputs (more than 3), we divide decrease factor by multiple of two (twin or couple) for each common inputs and outputs.		



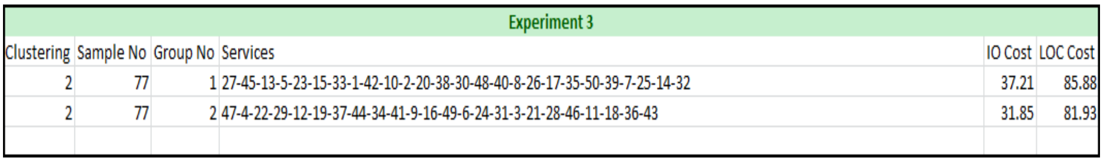
Graph 3- Experiment 3 (Total Development Cost of 50 clusters)

In graph 3, x axis illustrates the clusters that we have used during our experiments while y axis shows the total development cost of these clusters.

Result:

As a result we can say that;

- Cluster 2 is the optimal cluster which has the lowest development cost with 236, 87.
- Cluster 49 has the highest development cost with 263, 55.
- The difference between these two clusters is 26, 68.
- Development cost line increases constantly. So, we can say that clustering services do not affect the development cost significantly.



Experiment 3					
Clustering	Sample No	Group No	Services	IO Cost	LOC Cost
2	77	1	27-45-13-5-23-15-33-1-42-10-2-20-38-30-48-40-8-26-17-35-50-39-7-25-14-32	37.21	85.88
2	77	2	47-4-22-29-12-19-37-44-34-41-9-16-49-6-24-31-3-21-28-46-11-18-36-43	31.85	81.93

Figure 22- Optimal Services of Experiment 3 with I/O and LOC Costs

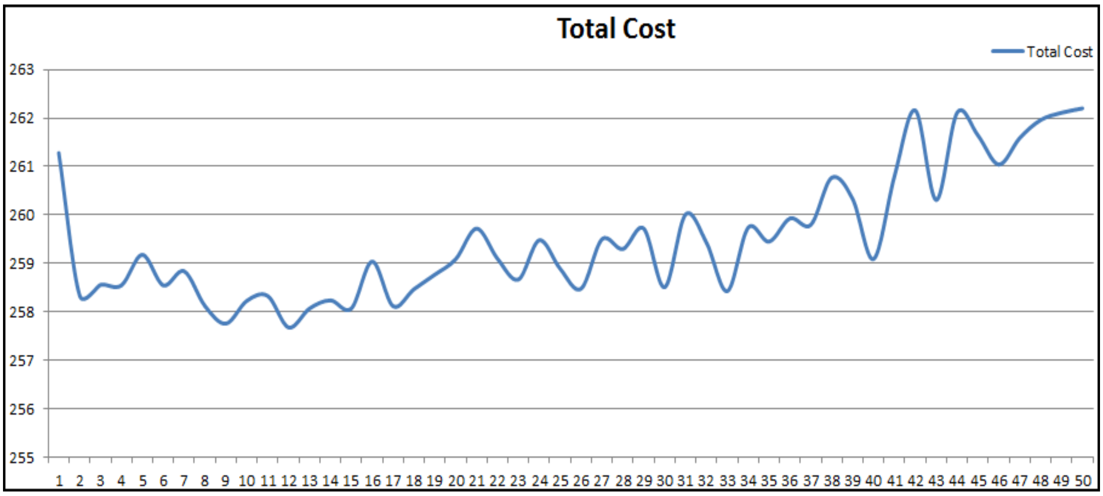
18.7 Experiment 4

Duration: 18 Hrs (Generating clusters and distribute services into clusters)

Description: This example illustrates the effect of granularity on service development cost. We changed the complexity cost range and number of samples. We repeated Experiment 1 with 200 samples.

Parameters & Factors:

No of Samples	200		
No of Services	50		
Complexity Cost Range	2-4		
Complexity Cost Factor	1		
Complexity Levels	LOW	AVG	HIGH
	1.1	1.13	1.15
IO Cost Factor	1		
IO Cost Levels	If the common inputs or outputs are more than one; for each common inputs and outputs that follow the first common input or output, we divide decrease factor by multiple of two (twin or couple) for each common inputs and outputs.		



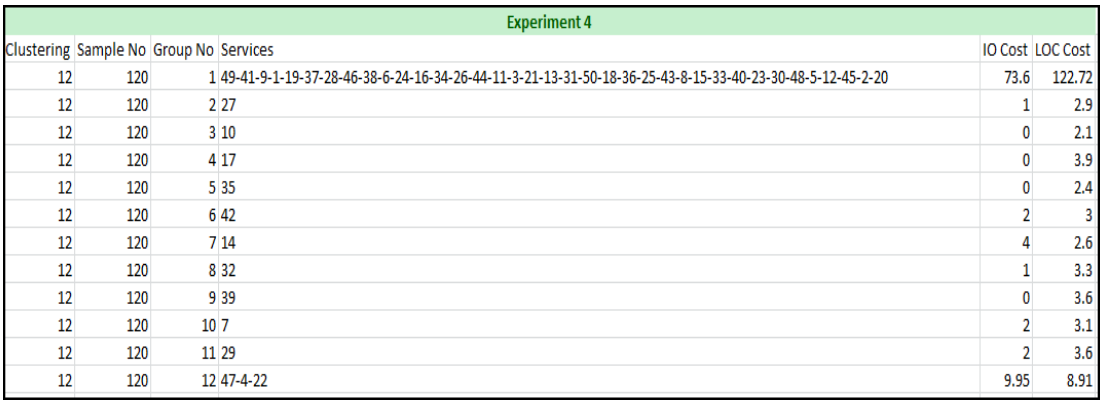
Graph 4- Experiment 4(Total Development Cost of 50 clusters with 200 samples)

In graph 4, x axis illustrates the clusters that we have used during our experiments while y axis shows the total development cost of these clusters.

Result:

As a result we can say that;

- Cluster 12 is the optimal cluster which has the lowest development cost with 257, 68.
- Cluster 50 has the highest development cost with 262, 2.
- The difference between these two clusters is 4, 52.
- The maximum development cost is in cluster 50. So, we can say that clustering services do not affect the development cost significantly.



Experiment 4					
Clustering	Sample No	Group No	Services	IO Cost	LOC Cost
12	120	1	49-41-9-1-19-37-28-46-38-6-24-16-34-26-44-11-3-21-13-31-50-18-36-25-43-8-15-33-40-23-30-48-5-12-45-2-20	73.6	122.72
12	120	2	27	1	2.9
12	120	3	10	0	2.1
12	120	4	17	0	3.9
12	120	5	35	0	2.4
12	120	6	42	2	3
12	120	7	14	4	2.6
12	120	8	32	1	3.3
12	120	9	39	0	3.6
12	120	10	7	2	3.1
12	120	11	29	2	3.6
12	120	12	47-4-22	9.95	8.91

Figure 23- Optimal Services of Experiment 4 with I/O and LOC Costs

19. DISCUSSION AND CONCLUSIONS

During our experiments, we found empirical results for different samples. The framework that we created in our research can be adapted to other real-time projects to get reliable results.

At the end of our experiments we found an optimal service granularity with different development costs. It is possible to see the effect of I/O Cost and Complexity metrics while calculating the development cost of a service.

According to our results, we can say that; I/O Cost and Complexity metrics do not affect the development cost of a service excessively when you generate different service granularities.

Of course there are possible factors which may decrease the cost estimation in SOA projects but in our case study, we tried to show the development cost of service granularities.

We noticed that the Work Breakdown Structure is not only for management processes but also useful for calculating development cost of a service. We have used this structure while we are calculating the complexity and I/O costs of a service. We defined Complexity and I/O cost as a sub-problems of the project. Then we found solutions to these problems by calculating the development cost for all services.

In the future, researchers should create new frameworks which are compatible with service oriented architecture. Defining new frameworks with different service points may help us to calculate more reliable development cost in our SOA projects.

References

[1] Hareton Leung, Zhang Fan "Software Cost Estimation", The Hong Kong Polytechnic University, Japan.

[2] Ian Sommerville "Software Cost Estimation", Software Engineering 7th Edition, UK, 2004

[3] W.Royce, "Software project management: a unified framework, Adisson Wesley", 1998

[4] NASA Johnson Space Center, "Parametric Cost Estimating Handbook", NASA, United States of America, 1995

[5] National Aeronautics and Space Administration "Parametric Cost Estimating Handbook", Department of Defense, USA, 1995

[6] Barry Boehm, Chris Abts, [Chulani] "Software Development Cost Estimation Approaches-A Survey1", Sunita Chulani IBM Research, University of Southern California, Los Angeles, 1998

[7] Boehm, B., Clark, B., Horowitz, E., Westland, C., Madachy, R., Selby, R. "Cost Models for Future Software Life Cycle Processes", 1995 http://sunset.usc.edu/COCOMOII/Cocomo.html

[8] The Data & Analysis Center for Software "Modern Empirical Cost and Schedule Estimation Tools", https://www.dacs.dtic.mil/techs/estimation

[9] Boehm,B "Safe and Simple software cost analysis", IEEE Software, 2000

[10] Web Site; www.softstarsystems.com/overview.htm

[11] Laren Lum, Michael Bramble, Jairus Hihn, John Hackney, Mori Khorrami, Erik Monson "Handbook for software cost estimation", Jet Propulsion Laboratory, Psadena,California, 2003

[12] J.D. Aron "Estimating Resource for Large Programming Systems", NATO Science Committee, Rome, Italy, 1969

[13] A.J Albrecht, and J. E. Gaffney, "Software Function, source lines of codes and development effort prediction: a software science validation"; IEEE Trans Software Eng. SE-9 pp. 639-648, 1983

[14] http://www.ii.metu.edu.tr/; "Software Quality Assurance".

[15] G. Karner ; "Resource Estimation for Objectory Projects"; Objective Systems; Torshamnsgatan, 1993

[16]http://www.service-architecture.com/web-services/articles/service-oriented_architecture_soa_definition.html, "SOA Definition"

[17] OASIS-Matthew MacKenzie, Ken Laskey, Francis McCabe, Peter F Brown, Rebekah Metz, Booz Allen Hamilton; "Reference Model for Service Oriented Architecture 1.0", Committee Specification, 2006

[18] http://www.soainstitute.org/,David S. Linthicum; "How much will your SOA cost?" 2007

[19] Card, D. N. ,W. Agresti, "Measuring Software Design Complexity." The Journal of Systems and Software 8, 3; 185-197. (Original definition of the metric), 1998

[20] http://www.aivosto.com/project/help/pm-sf.html

[21] http://soa.sys-con.com, 2005

[22] C. Roland, "A comprehensive view of process engineering, Proceedings of the 10th international conference" and B. lecture notes in computer science, Italy, 1998

[23] Claudia Steghuis, “Service granularity in SOA projects” University of Twente, Netherlands, 2006

[24] Boris Lublinsky; “Defining SOA as an Architectural Style”, www.ibm.com, 2007

[25] Thomas Fetcke, Alain Abran and Tho Hau Nguyen, **“**Mapping the o-o Jacobson Approach into Function Point Analyses**”,** IEEE, 1998

[26] Alvin J. Alexander; “How to Determine Your Application Size Using Function Point Analysis”, 2004

[27] www.devdaily.com

[28] http://www.softwaremetrics.com/examples/

[29] Thomas Fetcke, Alain Abran and Tho Hau Nguyen, **“**Mapping the o-o JacobsonApproach into Function Point Analyses**”,** IEEE, 1998

[30] http://www.softwaremetrics.com/examples/ei.htm

[31] www.ifpug.org

[32] http://www.softwaremetrics.com/examples/eo.htm

[33] Stojanovic Z. and Dahanayake, A. “Service-Oriented Software System Engineering: Challenges and Practices”. Hershey, PA: IGI Global, Apr. 2005.

[34] Sommerville, I. Software Engineering, 8th ed., London: Addison Wesley, Jun. 2006.

[35] Oladimeji Y. L., Folorunso O., Taofeek A.A, Adejumobi A. I,Nigeria,September, “A Framework for Costing Service-Oriented Architecture (SOA) Projects Using Work Breakdown Structure (WBS)”, Global Journals Inc. (USA), 2011

[36] Tansey, B. and Stroulia, E. “Valuating Software Service Development: Integrating COCOMO II and Real Options Theory,” The First International

Workshop on the Economics of Software and Computation, IEEE Press, May 2007, pp. 8-8, doi: 10.1109/ESC.2007

[37] Linthicum, D., "How Much Will Your SOA Cost?" http://www.soainstitute.org/articles/article/article/how-muchwill-your-soa-cost.html, March 2007.

[38] Conte M., Iorio T., Meli R., and Santillo,"An early and quick approach to functional size measurement methods". Software Measurement European Forum(SMEF). Rome, Italy, 2004.

[39] Liu, J., Xu, Z., Qiao, J., and Lin, S. "A Defect Prediction Model for Software Based on Service Oriented Architecture using EXPERT COCOMO," Chinese Control and Decision Conference (CCDC), IEEE Press, Jun. 2009, pp. 2591-2594

[40] O' Brien, L. "A Framework for Scope, Cost and Effort Estimation for Service Oriented Architecture (SOA) Projects" Software Engineering Conference" IEEE Press, April 2009.

[41] Li, Z., Keung Jacky "Software Cost Estimation Framework for Service-Oriented Architecture Systems using Divide-and-Conquer Approach" Sydney, Australia IEEE Press, September 2010

[42] Papazoglou M., Van den Heuvel W. "Service-Oriented Design and Development Methodology". International Journal of Web Engineering and Technology 2006

[43] Feuerlicht, G., Wijayaweera, A. "Determinants of Service Reusability, New Trends in Software Methodologies, Tools and Techniques", IOS Press, 2007

[44] Nucleonica, "Introduction to Monte Carlo method", European Commission Directorate –General Joint Research Centre, 2008

[45] J.S. Rosenthal, "Analysis of the Gibbs sampler for a model related to James-Stein estimators" Statics and Computing, 6, 269-275, 1996

[46] J. S. Rosenthal, “Parallel computing and Monte Carlo algorithms”, Far East Journal of Theoretical Statistics 4, 207–236, 2000

[47] S. Brin, L. Page, R. Motwami and T. Winograd, “The PageRank citation ranking: bringing order to the Web”, Stanford University Technical Report 1998, http://dbpubs.stanford.edu:8090/pub/1999-66.

Curriculum Vitae

Aslı Yılmaz Taşkın was born on 22 July 1984, in Izmir. She received her B.Sc. Degree in Computer Engineering in 2006 and M.Sc Degree in 2012 in Computer Engineering both from Kadir Has University. She worked as a research assistant at the Department of Computer Engineering of Kadir Has University from 2007 to 2008. She also worked in a private software company as a system analyst. After years, she started to run her own company in the United Kingdom.