Global Journal on Technology

Issue 7 (2015) 98-108

Selected Paper of 4th World Conference on Innovation and Computer Sciences (INSODE-2014)
11-12 April 2014, Quality Hotel Rouge et Noir Conference Center, Roma, Italy

Selection of a functional sizing methodology: A telecommunications company case study

Selami Bağrıyanık*, Software Development Department, Turkcell Technology R&D, Bahçeşehir University, Maltepe, Istanbul, Turkey.

Adem Karahoca, Software Engineering Department, Faculty of Engineering, Bahcesehir University, Besiktas, Istanbul, Turkey.

Ersin Ersoy, Software Engineering Department, Faculty of Engineering, Bahcesehir University, Besiktas, Istanbul, Turkey.

Suggested Citation:

Bağrıyanık, S., Karahoca, A. & Ersoy, E. Selection of a functional sizing methodology: A telecommunications company case study, *Global Journal on Technology* [Online]. 2015, 07, pp 98-108. Available from: www.awer-center.org/pitcs

Received December 14, 2013; revised February 19, 2014; accepted March 03, 2014.
Selection and peer review under responsibility of Assoc. Prof. Dr. Özcan Asilkan, Akdeniz University, Turkey.
©2015 SPROC LTD. Academic World Education & Research Center. All rights reserved.

Abstract

Functional size measurement is a very powerful tool for information technology practitioners since it's output is an invaluable information and used for several purposes. For instance functional size is vital in measurement of productivity and quality. It's also an important indicator in software projects cost estimation. However usage of functional sizing methodologies is not widespread in software industry and rollout within companies is still a big challange. This study focuses on the selection process of a functional sizing methodology in the context of a telecommunications company. It investigates the factors influencing sizing methodology selection decision and experienced issues in the course of proof of concept project. A case study was conducted measuring more than 40 projects from a Turkish telecommunications company using COSMIC and IFPUG methodologies as two candidates. After evaluations, the selected method within company was COSMIC.

Keywords: functional size measurement, function point, COSMIC, IFPUG, software cost estimation.

* ADDRESS FOR CORRESPONDENCE: **Selami Bağrıyanık,** Software Development Department, Turkcell Technology R&D, Bahçeşehir University, Maltepe, Istanbul, Turkey. *E-mail address*: selami.bagriyanik@turkcell.com.tr

1. Introduction

Functional size measurement can be defined as measuring functionality of a software system in terms of information processing operations such as saving and retrieving data in a database system or publishing information to external systems via interfaces. It is in use in software industry since its first introduction at IBM by Albrecht's FPA method [1]. Several other methods were derived from the original method in last 30 years. Figure 1 shows a brief history of the most common variants mentioned [2].

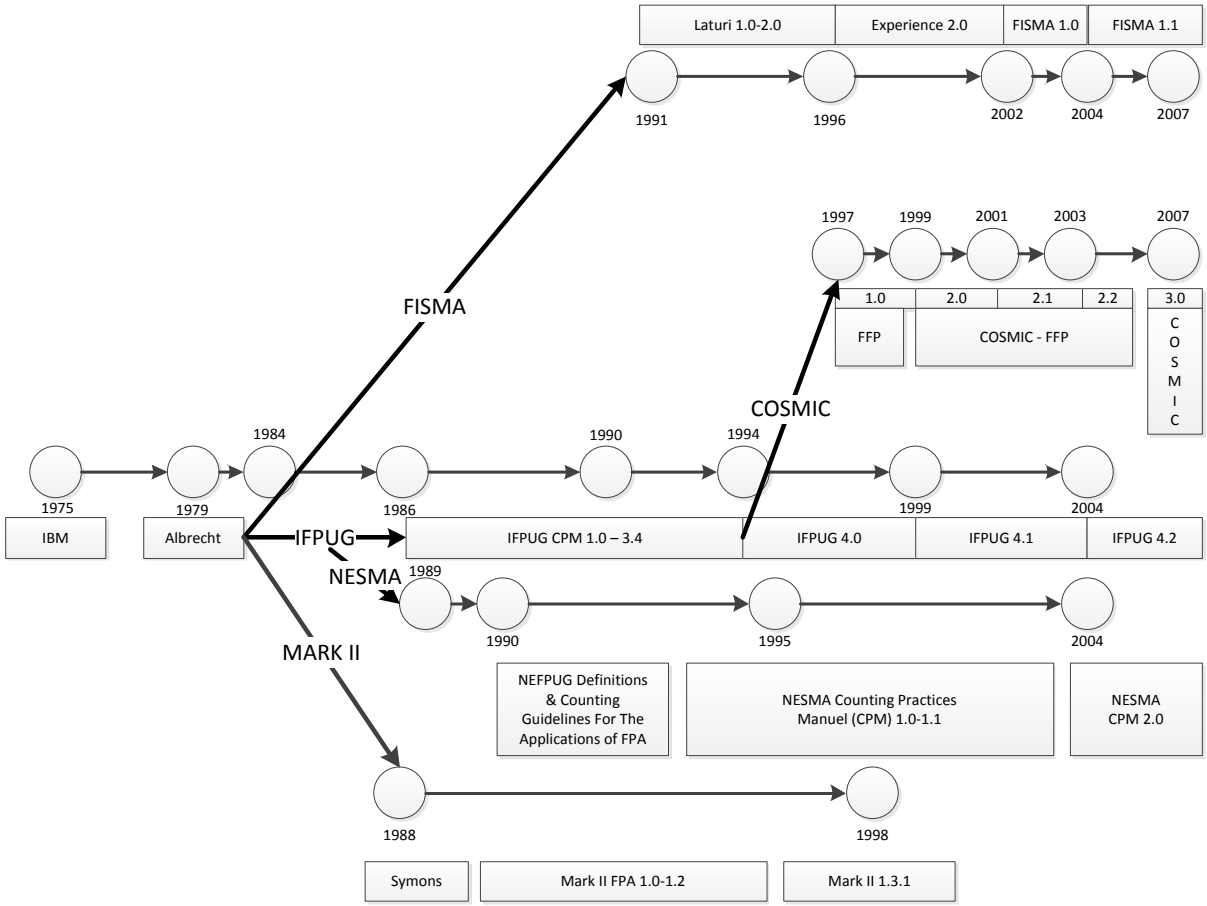

Figure 1. History of Functional Size Measurement (FSM) methods [2]

According to Capers Jones, there are a significant number of projects measured with function point based sizing methods in some important benchmarking organizations. Moreover, function point is the most reliable metric to measure software productivity and quality [3, 7]. However there are still some challenges in implementing a functional size measurement program for the software development companies since function point methods have some drawbacks. Some important issues are the cost of measurement and the difficulty to select a specific method among a lot of variants [3, 4]. On the other hand the issues mentioned are generic and can not be attributed to a specific method such as Netherlands Software Metrics Association (NESMA), International Function Point Users Group (IFPUG) or Common Software Measurement International Consortium (COSMIC) functional size measurement methods. Furthermore the company need, expectations from method and internal dynamics are other important factors for measurement projects to be successful.

In this case study, the selection process covers following steps:

1. Evaluation of existing functional size measurement (FSM) methods (long list)
2. Determining FSM short list (two methods) based on criterions which are important for the company
3. Constituting a proof of concept (POC) project team and getting necessary trainings
4. Determining the measurement scope (project list) of the POC project for the candidate methods in the short list.
5. Completing the measurements and evaluating results for the first candidate
6. Completing the measurements and evaluating results for the second candidate
7. Evaluating the results and choosing the best method
8. Final validation of the selected method

2. Material and Methods

2.1. Functional Size Measurement Methods

As pointed out in the book "Applied Software Measurement" [5], circa 2008, there are a lot of FSM methods in use and IFPUG, COSMIC, NESMA and Mark II variants dominate in the software industry. In the meantime, these four methods are ISO standards as well. The prominent FSM variants in alphabetical order is as follows: "

- 3D function points
- Backfiring function points
- COSMIC function points
- DeMarco "bang" function points
- Engineering function points
- Feature points
- Full function points
- Function points "light"
- IFPUG function points
- ISO function point standards
- Mark II function points
- Micro function points
- NESMA function points
- Object points
- Pattern matching and function points

- SPR function points
- Story points
- Unadjusted function points
- Use case points
- Web object points" [5]

It's assumed that the status regarding diversity of FSM variants described above is still the case at the moment. Detailed information regarding IFPUG and COSMIC methods will be given in following sections since they are the primary candidates used for this case study. The other methods' details will not be given.

2.1.1. IFPUG Function Point

IFPUG method is the oldest function point method. Its first version was published in early 80s. [2] In this section a summary of the method will be given based on the IFPUG Counting Practices Manual version 4.1. [6] This summary will not be adequate to measure a software project. Counting manual should be studied thoroughly in order to measurement be done. IFPUG function point counting workflow begins with determining the type of function point count, scope of the measurement and application boundary. After that, data functions and transactional functions are measured and this yields unadjusted function point count. As a final step, adjustment factor value is determined and adjusted function point count is calculated.

Figure 2 shows the basic concepts of the method using a sample Human Resources application.

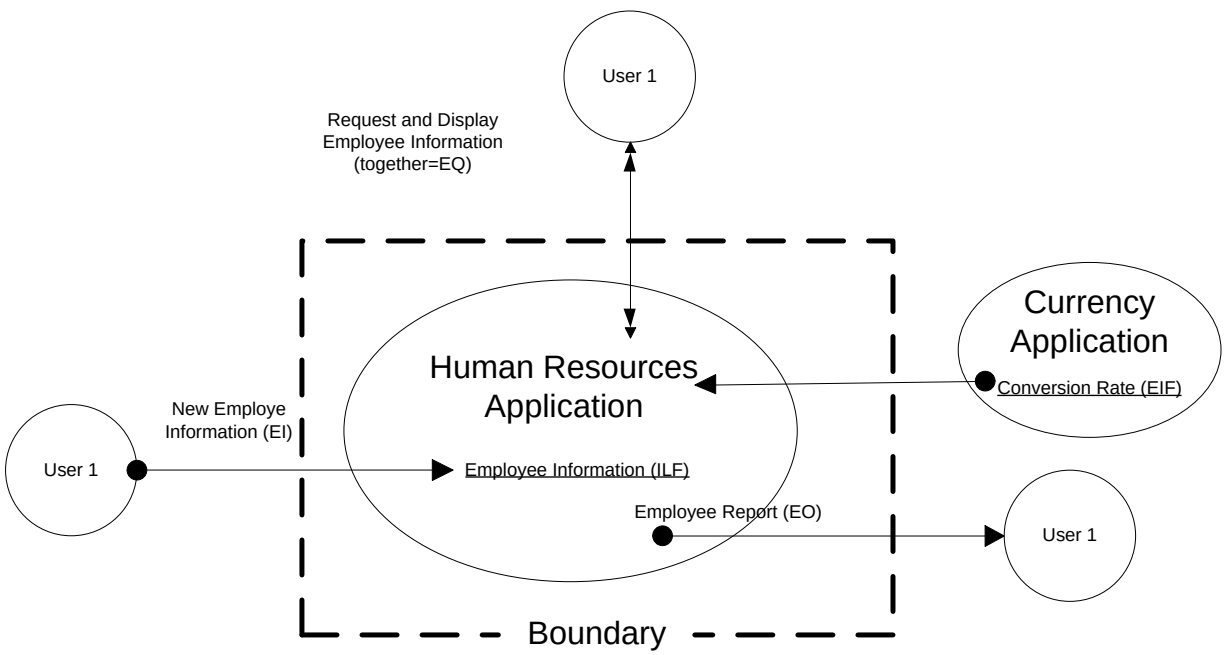

Figure 2. A Sample Application Exhibiting Basic IFPUG Concepts [6]

External inquiry (EQ), External Input (EI), External Output (EO), External Interface File (EIF) and Internal Logical File (ILF) are transactional and data functions. Boundary is another important basic concept of the method. It represents the virtual boundary between the user and the system. An ILF can be considered as a conceptual data entity perceived by the user and processed within the system. An EIF can be considered as a conceptual data entity perceived by the user and referenced by the system but processed within another system. An EI processes the data coming from outside of the system boundary. An EO and EQ transmits data outside the system's boundary. Each data and transaction function's contribution to total count is computed by complexity matrixes. An example matrix is given in Table 1. The other matrixes and translation tables will not be given. Please refer to manual for further details [6].

Table 1 Functional Complexity Matrix [6]

	1 to 19 DET	20 to 50 DRT	51 or more DET
1 RET	Low	Low	Average
2 to 5 RET	Low	Average	High
6 or more RET	Averate	High	High

After all data and transaction function's complexity values are determined and unadjusted function point values are found the summation can be multiplied by adjustment factor to find the final IFPUG adjusted function point total [6].

2.1.2. COSMIC Function Point

COSMIC method is a relatively new method compared to IFPUG. Its first publicly known version was published in 1998. [8] In this section a summary of the method will be given based on the COSMIC Measurement Manual. This summary will not be adequate to measure a software project. Measurement Manual should be studied thoroughly in order to measurement be done.

In Figure 3 COSMIC measurement process is shown. The process is composed of three basic phases: Measurement Strategy, Mapping and Measurement Phase. In the first phase purpose, scope and level of granularity are determined and functional users are identified. In mapping phase, functional processes, data groups and data attributes (optional) are identified. The last phase identifies data movements and applies measurement function to all functional processes identified in the previous phase. The output is functional size of the project in Cosmic function point unit.

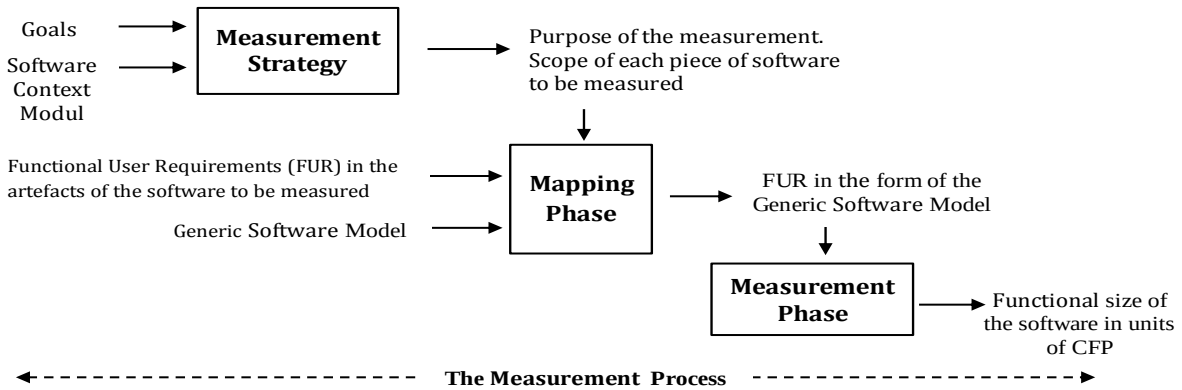

Figure 3. COSMIC Measurement Process [8]

Figure 4 exhibits some basic concepts of COSMIC method. Entry (E), Exit (X), Read (R) and Write (W) are data movement types. Data movements carry information called data groups within functional processes. E represents a data movement entered into the system. X represents a data movement to outward direction. R is a fetch of data from persistent storage while W is a data write to persistent storage [8].

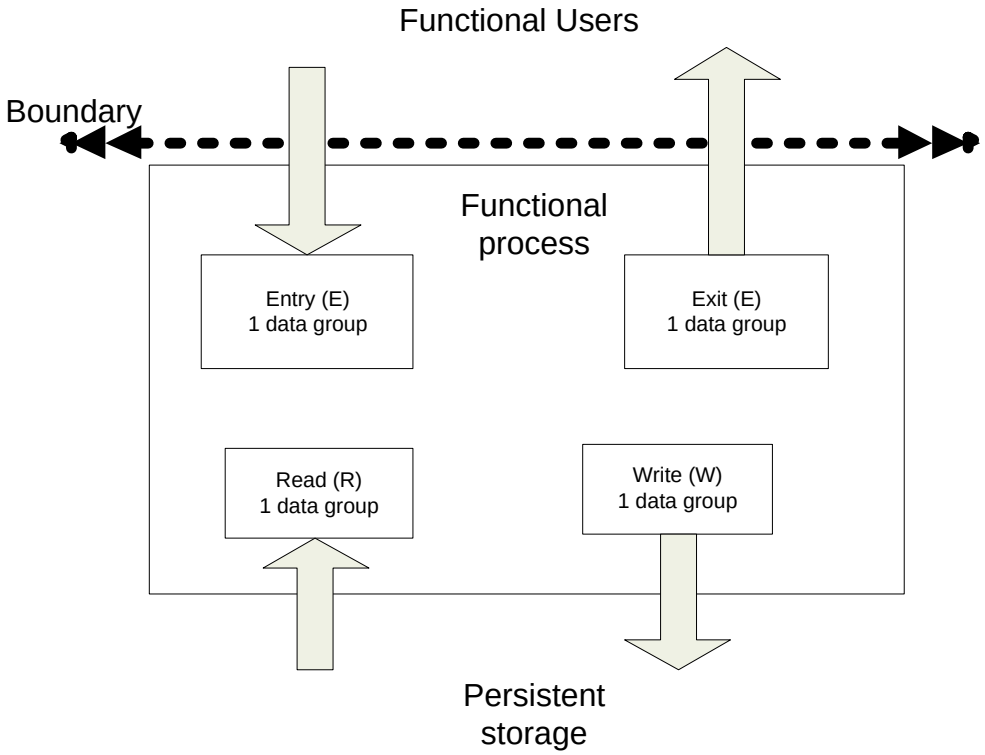

Figure 4. Basic Cosmic Function Point Concepts [8]

Each data movement inside a functional process, for a data group account for 1 Cosmic Function Point. Functional process size is computed by counting all data group movements flowing inside of it. Similarly the size of a project's requirements scope is the summation of the sizes of all functional processes inside the scope [8].

2.2. Weighted Decision Matrix

Weighted decision matrix which is developed by Stuart Pugh is a method used for Multiple criteria decision analysis (MCDA). [9] A weighted decision matrix can be constructed using a set of weighted criterions that drive the decision and decision options which are scored against each criteria and summed to have a total score for each option. Finally the scores are ranked and the top ranked option is selected. [9]

3. Evaluation and Selection of a Functional Size Measurement Method and Results

In the beginning, the main motivation in Information and Communication Technologies (ICT) Software Delivery Departments of company was a need for a systematic cost estimation model. However even it's a secondary goal, measuring productivity and quality in a more objective manner is also important for the ICT. Therefore the solution should address or support those two aims as well.

Since FSM output is a common denominator for several software delivery metrics, it can both serve as a basis to cost estimation and computing productivity and quality. Therefore the original cost estimation problem turned out to a functional size measurement (FSM) method selection problem in the first place.

Thus a functional size measurement method selection initiative was started by the management. The selection process has eight steps as listed in the introduction section. A detailed explanation for each step is given as follows.

3.1 Evaluation of Existing FSM Methods

There are at least 20 methods for FSM as stated in section 2.1. It's not feasible to try all these methods in a proof of concept study. Thus this long list should be filtered out to a short one.

3.2 Determining FSM Short List

In this step the aim was to select two methods among all FSM methods based on following preliminary criterions which are important for the company:

a. Method should be a widespread method in the world especially both in USA and in Europe.

b. Method should be applicable to telecom software products and applications, especially business support systems (BSS).

c. Documentation and training facilities should be available via local consultancy and web resources for the method.

d. Benchmarking measurement data measured by the method should be available .

e. Method should be an ISO standard.

f. Simplicity, measurement cost level and extra benefits and advantages provided by the method. Although it's a very important factor for the company, this criteria is not included in this phase and left to proof of concept study since it needs an experimental study to be validated.

Table 2 Decision Matrix for Long List of FSM Methods

FSM Options / Criterias	Weight	3D	Back firin	COSMIC	DeMarco	Engineerin	Feature Points	Full FP	Light	IFPUG	Mark II	Micro	NESMA	Object Points	Pattern Matchin	SPR	Story Point	Unadjusted	Use case	Web Object	FISMA
Widespread	0,3	1	6	8	2	2	3	4	4	10	5	1	7	4	2	6	6	7	6	7	7
Applicable to Telco Business	0,3	1	6	9	8	5	10	8	8	8	8	8	8	6	4	6	7	8	6	5	8
Documentation and Training	0,2	2	5	10	2	2	3	5	7	8	6	5	8	6	5	8	8	8	6	5	8
Benchmarking	0,1	1	5	7	1	1	2	3	3	10	5	2	7	3	3	8	3	8	4	6	5
ISO Standard	0,1	1	1	10	1	1	1	1	1	10	10	1	10	1	1	1	1	10	1	1	10
Total Score	**1**	**12**	**52**	**88**	**36**	**27**	**48**	**50**	**54**	**90**	**66**	**40**	**78**	**46**	**32**	**61**	**59**	**79**	**53**	**53**	**76**

Detail information regarding 20 candidate FSM methods in the long list is searched in literature [2, 5, 10, 11, 12, 13, 14, 15] and evaluated in a weighted decision matrix as shown in Table 2. Weights are determined based on the importance of the criteria for the selection. Scores for each method are determined using the theoretical information captured from the literature review. Methods whose total score value is above 70 point are marked with green colour in the table. These are COSMIC,

IFPUG, NESMA, Unadjusted Function Point and FISMA. Although all these last five candidates in short list performed very well, COSMIC and IFPUG are compliant with the criterions by far the best. Hence IFPUG and COSMIC methods were selected as final candidates to be tested with a proof of concept experiment within the company.

3.3. Constituting a Proof of Concept (POC) Project Team and Getting Necessary Trainings

A POC project team composed of twelve developers and software testers from different business domains constituted with voluntarily employees after announcement to delivery teams in the company. Firstly an introduction IFPUG training has been given to project team by a senior architect who learnt the IFPUG methodology from the official manual. Fundamentals, counting practices explained and put into practice on real development projects. IFPUG training was given internally since there is no local training company support for IFPUG. However COSMIC method training was taken from a local company based in Turkey.

3.4. Determining the Measurement Scope of the POC Project for the Candidate Methods

In order to measure a project, it is assumed that it is necessary to have high level information about the project. For that reason, selected sample projects were those that are known by the project team members. Additionally POC projects were selected based on following criterions:

- Project should be completed to estimate average man-day cost for each FP.
- At least one POC project member should have high level information about the project.
- An infrastructure project should be included in the scope to see whether they can be measured.
- A project with change requests in requirements should be included to see if the change requests can be measured or not.

3.5 Completing the Measurements and Evaluating Results for IFPUG Method

Eleven projects were measured and following conclusions were reached by the team:

- Measurement effort is approximately 1% of the total project effort.
- Infrastructure projects' size is zero function point since they have no new, modified or deleted functionality in their scope. Example project was a database upgrade project and scope of this project is upgrading the Oracle database used by some applications.
- It's observed that projects with change requests in their scope could be measured without a problem. Change request (scope creep) size could be measured separately as well.
- IFPUG is an easy method. However measurer's experience and domain knowledge is important for accurate results.
- Complexity matrices (Table 1) were interpreted as subjective by the team.
- Terminology is old and model is more relevant for the monolithic systems such as mainframe systems.

3.6. Completing the Measurements and Evaluating Results for COSMIC Method

This time, eleven projects in the POC scope were measured using COSMIC method by the team and following conclusions drawn:

- Measurement effort is approximately 0.4 % of the total project effort.
- Infrastructure projects' size is zero function point since they have no new, modified or deleted functionality in their scope. Example project was a database upgrade project and scope of this project is upgrading the Oracle database used by some applications.
- It's observed that projects with change requests in their scope could be measured without a problem. Change request (scope creep) size could be measured separately as well.
- COSMIC is an easier method than IFPUG. However measurer's experience and domain knowledge is still important for accurate results.
- COSMIC doesn't contain subjective models as in IFPUG Complexity matrices (Table 1). Scalability is provided while data model is getting more complex in COSMIC.
- Terminology and model is compatible with the new generation concepts such as object orientation, service oriented architecture and multi layered applications.

3.7. Evaluating the Results for the Candidates

Final decision is primarily based on the simplicity of the method, measurement cost introduced by the method and extra benefits and advantages provided by the method if any. As mentioned in section 3.6 the COSMIC method was evaluated as a simpler method compared to IFPUG. A similar observation regarding simplicity is made in a COSMIC and IFPUG case study conducted within Renault. [19] Moreover manual measurement effort experienced in IFPUG is approximately two times bigger than the effort in COSMIC. In the POC, it's also observed that the methods can be used for requirements analysis quality measurement and improvement as well. After that observation, a literature search is done and a formal COSMIC method usage for requirements analysis quality improvement was found in a Phd thesis and supported our observations.[18] The importance of the quality of the functional requirements is pointed out in case study within Siemens AT as well. [20] Regarding extra benefits and advantages, COSMIC's terminology and model is compatible with modern software engineering practices such as multi layered and service oriented architectures and object orientation, etc. COSMIC advantages compared to IFPUG is supported in a research project conducted in the same company as well [16].

As a result of the superiorities mentioned above, COSMIC method is selected as the functional sizing methodology for the company.

3.8. Final Validation of the Selected Method

Previous steps are conducted by a team composed of developers and test engineers. There are also requirements engineers (analysts) in the company ecosystem and they are potential users of the method as well. Hence a final validation step is planned to validate the COSMIC method from the requirements analyst perspective as well. A project team composed of six analysts is constituted from different domains and trained for the COSMIC method. They measured 37 small and mid sized projects from different business domains. The outputs of this last stage confirmed the results reached in section 3.6 and 3.7. as well. This stage also revealed the importance of the quality of the requirements' content one more time. That is, if the requirements are not complete, have some gaps or defects then the measurement is also a quality validation and improvement process.

4. Conclusion & Future Research

Functional size measurement is necessary in order to measure several key metrics in software development life cycle. It's also a strong indicator for software cost estimation. This study focused on the selection process with a two phased strategy: A theoretical information based short list candidate selection phase and an experimental phase to test the selected candidates in the short list. First of all, based on a weighted decision matrix and theoretical literature information a short candidate list (IFPUG and COSMIC) is chosen among twenty methods. After that, both methods were tried using completed real projects in the company. Project measurements were done by developers, testers and analysts to capture the perception from all disciplines. Finally, having survived all previous stages COSMIC is selected as the most appropriate technic for the company due to three more reasons: it's simpler and cheaper, it can also be used for quality improvement and last but not least, it's compatible with current software engineering technologies, models and practices.

Further research regarding following topics are considered to be done in the future:

- Automation of functional size measurement process using software requirements analysis artifacts
- A preliminary research conducted within company revealed that the function point based software cost estimation approach can be better than traditional estimation methods such as COCOMO. [17] As a further research, software cost estimation by data mining methods using Cosmic base functional components and some additional factors such as developer expertise, used technology etc will be done.
- Software requirements quality improvement using Cosmic will be considered.

Acknowledgements

We would like to thank the software developers, system analysts, architects, managers and all the other staff of the company who contributed this research by their voluntarily support.

References

[1] Albrecht, A., J., & Gaffney, J., E. (1983). Software Function, Source Lines of Code, and Development Effort Prediction: A Software Science Validation, *IEEE Transactions on Software Engineering, SE-9*(6), 639 - 648.

[2] Bundschuh, M., & Dekkers, C. (2008). *The IT Measurement Compendium*. Springer – Verlag, Berlin, 2008, 326

[3] Jones, C. (2014). *Function Points as a Universal Software Metric*, [Online] Received January 1, 2014 from: http://namcookanalytics.com/wp-content/uploads/2013/07/Function-Points-as-a-Universal-Software-Metric2013.pdf

[4] Huskins, M., Kaplan, J. & Krishnakantan, K. (2014). *Enhancing the efficiency and effectiveness of application development*, [Online] Received January 1, 2014 from: http://www.mckinsey.com/insights/business_technology/enhancing_the_efficiency_and_effectiveness_of_application_development

[5] Jones, C. (2008). *Applied Software Measurement Global Analysis of Productivity and Quality*, 3rd edition. Mc Graw Hill, USA, 2008, 104-159.

[6] International Function Point User Group (IFPUG). *Function Point Counting Practices Manual Release 4.1*, [Online] Received February 16, 2014 from: https://www.assembla.com/code/gxpoints/subversion/nodes/40/Documentos/IFPUG_Counting_Practices_Manual_4.1.pdf

[7] Jones, C. (2010). *Software Engineering Best Practices Lessons from Successful Projects in the Top Companies*. Mc Graw Hill, USA, 376-379.

[8] Common Software Measurement International Consortium. *The COSMIC Functional Size Measurement Method Version 3.0.1 Measurement Manual*, [Online] Received January 1, 2014 from: http://www.cosmicon.com

[9] Wikipedia. *Decision Matrix Method*, [Online] Received January 1, 2014 from: http://en.wikipedia.org/wiki/Decision-matrix_method

[10] Total Metrics. *Methods for Software Sizing How to Decide Which Method to Use*, [Online] Received January 1, 2014 from: http://www.totalmetrics.com/function-point-resources/downloads/R185_Why-use-Function-Points.pdf

[11] Jones, C. (2007). *Estimating Software Costs: Bringing Realism to Estimating*. McGraw-Hill, USA, 23 - 32

[12] Demirors, O., & Gencel, C. (2009). Conceptual Association of Functional Size Measurement Methods, *IEEE Software*, 71-78.

[13] Software Measurement Services Ltd. *Everything You Always Wanted to Know About Software Measurement*, [Online] Received January 1, 2014 from: http://www.cosmicon.com/portal/public/AllAboutFPA.pdf

[14] The International Software Benchmarking Standards Group Limited. *Functional Sizing Methods*, [Online] Received January 1, 2014 from: http://www.isbsg.org/ISBSGnew.nsf/WebPages/AC308EF2C87A3A2BCA25745F002CAE29

[15] Software Engineering Measurement & Quality. *Functional Size Measurement (FSM)*, [Online] Received January 1, 2014 from: http://www.semq.eu/leng/sizestfsm.htm

[16] Ersoy, E. (2012). *Bilişim Sistemi Projelerinde Kaynak ve Zaman Gereksinimlerinin Hesaplanması: IFPUG ve COSMIC Yöntemleri Arasında Bir karşılaştırma.* Master of Business (MBA) Project, İstanbul Bilgi Üniversitesi, 2012, 35-46.

[17] Bağrıyanık, S. (2010). *Yazılım Uygulamalarının Geliştirilmesinde Kaynak ve Zaman İhtiyaçlarının Öngörülmesi: Bir Oracle – Siebel Müşteri İlişkileri Yönetimi Çözümü İçin Uygulama.* Master of Business (MBA) Project, İstanbul Bilgi Üniversitesi, 2010, 34-50.

[18] Trudel, S. (2012). *Using the Cosmic Functional Size measurement Method (ISO 19761) as a Software Requirements Improvement Mechanism*. Phd Thesis, École De Technologie Superieure Universite Du Quebec, 2012, 151-170.

[19] Stern, S. (2011). Practical Experimentations with the COSMIC Method in the Automotive Embedded Software Field, In Dumke R. Abran A. (ed) *COSMIC Function Points Theory and Advanced Practices.* CRC Press, Florida, 237-246.

[20] Schweikl U., Weber, S., Foltin, E., & Dumke, R. (2011). Applicability of Full Function Points at Siemens AT, In Dumke R. Abran A. (ed) *COSMIC Function Points Theory and Advanced Practices.* CRC Press, Florida, 2011, 228-237.