Experiment with COSMIC V3.0: Case Studies in Business Applications

Sanae Saadaoui, Annick Majchrowski and Christophe Ponsard
CETIC, Charleroi, Belgium
{sanae.saadaoui, annick.majchrowski, christophe.ponsard}@cetic.be
http://www.cetic.be

Abstract

Software measurement currently plays a crucial role in software engineering. It helps analysing both quality and productivity of the developed or maintained software. COSMIC–FFP, the COmmon Software Measurement International Consortium Full Functional Point (CFFP) is a measurement methodology designed to estimate the functional size of software. While bundled with comprehensive guidelines, the practical application of COSMIC is not trivial and experience plays an important role for its successful application. The objective of this paper is to provide practitioners with hints and additional recommendations learnt from applying CFFP to significant industrial cases. More specifically this paper will focus on the latest COSMIC (COSMIC V3.0) which comes with new concepts and for which there is still not much experience returns from its application. This paper relies on two significant case studies covering different domains: a wide-scale web application for a European commission administration and a modular application to be used jointly by five Belgian federal and regional parliaments. Different measurement contexts are also considered especially initial estimation vs application enhancement.

Keywords

Functional Size Measurement, COSMIC V3, modular application, business application

1 Introduction

The COSMIC method is one of the largely used methods of sizing software estimation [4]. The interest in COSMSIC comes from the fact that it is based on objective criteria and rules which allows repetitiveness of measures. The method is innovative [8][9]because it is applicable early in the software lifecycle, already at specifications level. Another main advantage of the CFFP is its complete independence of software development technologies and methods. The recent versions of the method are largely generic and cover many software domains from business applications [5] to real time systems [13].

The aim of this paper is to present a set of practical recommendations that we hope may help COSMIC V3.0 practitioners. These recommendations are based on problems encountered in representative case studies that we think are typically encountered when applying the method, especially the latest evolution of COSMIC. Some among the problem discussed are:

- the choice of artefacts and documents to use for software size estimation,
- the choice of the level of specification details, especially when artefacts have different granularity level
- the impact of the structure of the document, especially related to the mapping

All those may have great impact on the estimation effort. Although already partially addressed by guidelines such as [5], we give here a more practical feedback based on two case studies related to large systems. The first one concerns the estimation of the size of a change to a large scale web application for a European commission administration. The second case study is a size estimation of a new modular access card system to connect five parliaments in Brussels.

The present paper only considers the business applications. However it presents observations bound to a large spectrum of contexts: measurement of new development software, change request size estimation, development effort estimation and government procurement procedure.

The paper is organized as follows: section 2 presents an overview of the COSMIC model V3.0. Sections 3 and 4 respectively describe two case studies using a similar structure: a first sub-section is dedicated to the context and technical aspects of the application. It is followed by a second sub-section giving highlights of how COSMIC was applied. This sub-section focuses only on key and relevant aspects and differs from one case study to the other. Section 5 resumes lessons learnt from the experiments and give some practical recommendations for future applications. Section 6 gives some conclusions and future work directions.

2 Overview of the COSMIC V3.0 model

The COSMIC-FFP [4][5][6] is a standardized method – ISO/IEC 19761:2003 - designed for measuring the functional size of software [4][5][6]. The method is applicable for data rich applications: Business application software, real-time software and hybrid application of the above. The measurement can be carried out at any phase of the life cycle of the software, whether the software already exists or not. The COSMIC V3.0 method is applied in three phases: the strategy phase, the mapping phase and the measurement phase.

During the **strategy phase** four key COSMIC concepts are defined: (1) the scope and (2) purpose of the measurement to identify the set of functional user requirements (FUR) to measure. The software may need to be broken down into several pieces; (3) the functional users - who will use the pieces of software – users may be humans or peer application and (4) the level of granularity which specify the level of details at which the measurement should be carried out.

At the **mapping phase**, the FUR is decomposed into functional processes. Each functional process comprises a unique set of data movements, at least two. A data movement is a functional sub-process that moves a data group. A data group is a set of attributes related to an object of interest. A data movement can be one of four types: an *ENTRY* (E), moves a data group from a functional user to the functional process; an *EXIT* (X), moves data from the functional process to the functional user; a *READ* (R), moves a data group from a persistent storage to the functional process and the *WRITE* (W), moves data from the functional process to a persistent storage.

During **measurement phase**, data movements of each functional process are first identified. Next, the COSMIC measurement function is applied to each data movement. Each data movement is allocated 1 CFP (COSMIC Function Point). CFP is the unit of COSMIC measurement. The functional size of a functional process is the sum of sizes of its constituent data movements. The size of the measured piece of software is the sum of the sizes if its constituent functional processes.

3 *Case Study 1 : Changes on a business application*

3.1 Project description

The project is realized in a consulting software development context where the client is DGTaxud – the General Direction of Taxation and Customs Union, one of the directions of the European Commission. The software development team is SBS - Siemens Business Services. Contact modalities require size estimation based on COSMIC and this estimation was contracted out to third party experts: CETIC – Centre of Excellence in Technologies of Information and Communication, an independent research centre.

The purpose of this project was to implement some change requests on a Service Management tool application supporting the essential processes included in ITIL (Information Technology Infrastructure Library) framework [14]. The changes to develop on this Web-based application are registered in a context of evolutionary maintenance of the application.

The two of the eleven ITIL processes impacted by the changes are: (1) The Incident Management Process is responsible for the management of all incidents (corresponding to calls) addressed by users to Service Desk management and (2) The Report Management Process is responsible for the management of all reports created by Report Managers and executed by Report Executors. For every Report, Reports managers define access rights for the Report Executors.

The changes to be developed can be summarized as followed:

- Changes concerning the creation of an incident. In the new incident creation screen, four new fields are added to support four new objects which also have an impact (modification and/or deletion) on other existing items including changes to their functionalities.
- Changes concerning reporting management. Templates of the different reports and export files have to be updated to take into account the new objects.

3.2 COSMIC application highlights

This section presents how COSMIC was applied and describes some key concepts as "*level of granularity*" and "*functional users*", "*functional processes*", "*data groups*" and "*data movements*" . We have focused on difficulties and facilities encountered when applying them.

Identification of the FURs

Before applying COSMIC method, we have to identify the Functional User Requirements (FUR) of the piece of software to be measured. Two input artefacts were used for extracting the FUR. The first describes the changes on the Incident Management process and the second on the Report management Process. Extracting FUR is a crucial task which was very hard to perform due to the scope of the changes on the other ITIL processes. New impacted changes were analysed using relative artefacts and initial artefacts before changes. As suggested by COSMIC V0.3 [5], to measure changes, we first measure the existing functional processes. Secondly, we check how they will be functionally changed. Last, the changes are mesured.

Thus, an **important number of input artefacts** (five large artefacts) were used. They are written in **different forms** for **different kind of readers** as end-users or analysts or developers. The two documents relating changes are most *developer-dedicated*. They are composed by a set of existing screen forms with their fields (and all possible values) and the detailed related changes - new, modified, deleted fields -. They also describe changes related to processed functionalities induced by these fields and some technical considerations. The third used document relating the specifications of the existing application are rather *analyst-dedicated* and contains a global description of all supported ITIL processes. The two last used documents are *end-users-dedicated* and contain a more detailed description of the functionalities for all ITIL data processes manipulated by the application. Note that all these differences between documents have contributed to complicate the extraction of the existing FUR and changed ones.

Strategy phase

When having identified the FUR, we defined the real context of our measure. The four key concepts of this phase being the purpose, the scope, the functional users and the level of granularity of the measurement had to be identified before starting the measure.

The **functional users** defined as "*the senders or the intended recipient of data in the FURs* [4]) were easily identified. In V3.0, no difference is made between developer and end-user viewpoints as in version 2.2. [7]. Distinguishing between the two viewpoints in our case would have been difficult due to large amount of available documentation.

The **level of granularity** is defined as "*any level of expansion of the description such that at each increased level of expansion, the description of the functionality is at an increased and uniform level of detail*" [4]. In applying the definition, we found that the level of details of the documentation was strongly linked to the COSMIC concept of level of granularity: the highest level of details corresponding to the lowest levels of granularity. Since the available input artefacts were dedicated to different readers, functionalities were described at different levels of details: developer-dedicated documents being more detailed than end-user documents. In such situation, COSMIC V3.0 says that "*The recommended and unambiguous way to measure is to make the measurement with the same level of granularity and ideally with the level of granularity at which the functional processes have been identified"* [4]. The level of details in the used changes artefacts corresponds to a **functional sub-process level of granularity**. Additionally, as the scope was to measure the changes of a piece of software, we granted more importance to the level of details of changes than of existing functions. These were our motivations to **choose the lowest level of granularity** for our measurement.

Mapping phase

In this phase, the identified FUR is mapped to the COSMIC concepts such as "*functional processes"*, "*functional sub-processes"*, "*data groups"*. Focuses was put on the easiness in deriving functional processes and data groups from the available artefacts.

To identify the **functional processes**, we followed one of the approaches described in [5] stating that *"It is sometimes helpful first to break down the FUR into their elementary parts or the smallest units of activity that are meaningful to the users (for example: definitions of screen, or report layouts)"*. In our case, the artefact relating changes for Incident Management was precisely composed by a set of screens and was composed of report layouts. We observed that defining **screens and reports layout** in input documentation is a good practice to identify the functional processes.

To identify the **data groups**, we have not applied a data analysis method, as recommended [5]. We derived data groups by interpreting the fields and the columns as data groups. We observed that documentation containing directly the COSMIC concepts remains naturally the most interesting approach.

Measurement phase

In the context of a software evolution, we used the following approach to identify the **data movements** of each functional process: (1) consider the old situation and measure the data movements for the old functional process, (2) consider the new situation and measure the data movements by noting the changes (Added, Modified, Deleted, No change), and (3) the total functional size of the changed functional processes equals the sum of the data movements of all functional processes different from "No change".

4 Case study 2 : A modular business application, the PACXS project

4.1 Project description

The PACXS (Parliamentary Access Control eXchange System) is a project which aims to unify the access control of five Belgian parliaments, both at regional and federal level. The application should guaranties a better secured access to all assemblies infrastructure. It should also ease the administration and improve usability due to the large number of people involved in several assemblies currently requiring multiple credentials. This requires fined grained sharing of information of access control information between those assemblies. It should also integrate with existing access control systems.

The application to develop had to be distributed into different logical modules grouping a number of components composed of a set of related functionalities. The main modules are: the access management system, the badge management system and exchange management system. The modularity aspect of the application was central to the project to allow assemblies to continue to work separately and independently while integrating the required modules for a reliable exchange of information. The system is designed around a number of mandatory modules, beyond those; each assembly is free to integrate additional shared modules depending on her needs and existing infrastructure. This can support two kind of deployment: a minimal one just relying on the core (targeting parliament with a full system in place) and a complete system (targeting parliaments with a partial system in place). The estimation is done for both kinds of modules with a clear interface on the interaction with the existing systems. The elaboration of the functional and technical description of the application went through the following steps:

- Functional requirements document describing the high level functionalities of the system using a goal-based approach [13]
- Use case document describing business scenarios (between the user and the system) and more technical scenarios (at message exchange level), based on a UML approach.
- A user interface document describing screen mock-ups
- A component model document describing component functional interfaces and interactions
- A detailed technical analysis describing all the functionalities of the system and of messages

4.2 COSMIC application highlights

The highlights presented here cover mainly the choice of artefacts from which to derive key concepts like Functional User Requirement (FUR), functional processes and data groups and data movements.

Purpose of the measurement

The purpose of this measurement is to estimate the development effort for the development of the new modular application PACXS. The firm responsible for the development of the system will be selected through a Government Procurement procedure. The overall objective of this measurement is indeed to help choosing the adequate development firm.

Identification of Functional User Requirements

Functional User requirements are a set of user requirements that explains what the software will perform in term of tasks and services. The FUR was extracted from the specification document. The FUR is composed of all requirements on business cases and all those relative to messages exchange. The non functional requirements are not part of the FUR.

Scope of the measurement

The singularity of the application to be developed is its modularity. Estimating the effort for the whole system as one piece of software was not an option, it was explicitly required to produce module level estimations for the following reasons:

- estimate the distribution of development costs between partners in a mutualisation perspective
- produce several partial call for tenders
- to some extend: identify integration effort of existing modules in some parliaments

Choice of the artefacts

COSMIC estimation needs to define concepts at the logical level. In first place, we used the requirement document for a preliminary estimation. The document is well structured because relying on goal orientated requirements. It contains both functional and non functional requirements. Only functional requirement are measurable by COSMIC. Even though complete in a functional point of view, the estimation was not satisfactory because such functionalities are described at too high level and do not allow to uncover all functional processes. For this purpose, the use case document which relates all business diagrams scenarios was used. However it did not provide detailed enough information on the messages exchange. Hence for these aspects, the most detailed technical document was used.

Identification of functional users

The PACXS system is located at the architectural layer. Two kinds of users were identified. (1) Human users - technical staff responsible for the management and administration of the system – and (2) non human users which are the other modules exchanging information with the estimated one and external applications - local system of each assembly, badge provider application and local human resource applications. Those were easily identified at the requirement level document because those goal-oriented documents have a clear description of the responsibility assignments.

Level of granularity and functional processes identification

The FUR of software may be expressed at different levels of granularity. From more general specifications to more detailed requirements. For an accurate measurement, COSMIC requires measurement to be done at the functional process level. For higher levels, the results are approximations [1][4][5].

The PACXS measurement was performed at the functional process level. The particularity of this system is that it is composed of the access card management sub-system and the exchange management sub-system. For the first sub-system, the functional processes describe the different business scenarios from the human user point of view. For the second one, it was difficult to define functional process level without referring to technical specifications. It is well known that the functional size of software typically increases when going from a high granularity level to a lowest one. A well established level of granularity at which to estimate regularly is crucial for software management and improvement. It was quite challenging to find a common level of granularity to apply COSMIC to all components because the work was carried out by two different people with quite different expertise (analyst vs technical). An identified improvement here is to force the same level of granularity using common templates and possibly, earlier monitoring by the estimation team.

PACXS size measurement and development effort estimation

In the measurement phase, each module was sized by adding the data movements of all its functional processes. Each assembly can then compute the effort of its own development by considering only the modules she is interested in. Mutualisation can also occur to share costs. It is important to notice that the functional size of the whole PACXS system is different from the added sizes of all modules measured separately. Sizing separately the modules takes into account more interactions and then more data movements.

To produce development efforts from the functional size, some kind of productivity coefficient is required. As for this specific context, (first common eGoverment procurement) no historical data were available, the approach was to calculate the rate factor for effort estimation based on statistical data from the ISBSG repository [3]. An advantage of applying the estimation at module level is that it allows to apply different rates depending on the nature of the modules considered,. Another issue requiring more investigation is the “novelty” and “regularity” aspects: some modules are completely new and very specific (business process level) while other are based on well known techniques and show systematic patterns. Hence, the productivity for the later modules is expected to be higher than the former ones.

5 Lessons learnt and some recommendations

This section presents the summary of the lessons learnt based on the experiment of applying COSMIC V3.0 on the two proposed case studies:

- Large amount of artefact may be not productive. Fewer ones but highly and homogeneously structured at process level are more relevant for COSMIC measurement
- Identifying the FUR is not a trivial operation and depends on the provided artefacts. It is recommended to used carefully structured documents. A well structured document is a one that (1) distinguishes between functional and non functional requirements; (2) describes the links from high level specification to lowest level ones and (3) highlights the inputs and outputs related to each specification. When producing such a document it is necessary to get specification details grouped in the same section to avoid redundancy when measuring. Goal-oriented requirements templates combined with use-case documents proved efficient in this.
- Mapping the FUR from the artefacts to the COSMIC concepts is not a trivial operation. A more accurate template of artefacts would be profitable to reduce mapping time. It would also provide a more objective interpretation method to derive from the input document elements COSMIC key concepts. These templates could be structured in sections correlated with expected COSMIC concepts. Such templates are currently being elaborated.
- Input artefact containing screens and report layouts facilitate in a significant way the identification of the functional processes and sub-processes during the mapping phase.
- An accurate size measurement depends closely on the way the strategic phase was conducted. The results of the measurements are strongly impacted by the purpose, scope and level of granularity. Different contexts measurement leads to different functional sizes. It is strongly advised to conduct this phase very carefully.

Defining an accurate level of granularity is a key success for defining a development effort methodology and an efficient effort estimation model.

6 Conclusion

In this paper we presented some lessons learnt together with contextualized and practical recommendations collected when applying COSMIC V3.0 method on two important case studies. The observations show that analysing documents to extract the FUR and level of granularity is very demanding and time consuming. A well structured standard template for requirement documents may save time and reduce considerably the strategic phase. A common document helps defining a standard level of granularity and then defining a strategy for software size comparison and the establishment of an effort estimation model. The result is an improvement in the software development process. We are actually working on elaborating a template. We hope practitioners will find this set of recommendations useful for their own day to day work with COSMIC.

Literature

[1] Charles Simons, Advancing Functional Size Measurement – which size should we measure? Software Measurement European forum (SMEF 2007), Roma, Italy, May 9-11, 2007

[2] Alan Abran, Peter Fagg, Roberto Meli, Charles Simon, ISO transportation and Clarifications of the COSMIC FFP Method f Functional Sizing, IWSM 2002

[3] ISBSG. The International Software Benchmarking Standard Group. Available at: http://www.isbg.org

[4] The COSMIC Functional Size Measurement Method Version 3.0: Measurement Manual. September 2007. Available at: http://www.lrgl.uqam.ca/cosmic-ffp

[5] Guidelines for sizing Business Applicatin Software using COSMIC-FFP, version 1.0. December 2005

[6] The COSMIC Functional Size Measurement Method Version 3.0: Method Overview. September 2007. Available at: http://www.lrgl.uqam.ca/cosmic-ffp

[7] COSMIC-FFP Measurement Manual, Version 2.2, January 2003. Available at: http://www.lrgl.uqam.ca/cosmic-ffp

[8] Adel Khelifi, Alain ABran, Luigi Buglione. A System for References for Software Measurements with ISO 19761 (COSMIC-FF). IWSM/MetrKon 2004

[9] Christophe J. Lokan. Function Points. Advances in Computers. M. Zelkwitz (ed), volume 65, chp 7. Academic Press, 2005

[10] R. Bridges. Estimating with Confidence: Applying COSMIC Method for Estimation in the Avionics Industry, European SEPG, June 11-14, 2007. richard.bridges@eurocopter.com

[12] Charles Symons. Benefits to the UK software industry of successfully exploiting metrics, and the importance of the COSMIC-FFP method to producing credible metrics. December 2006. Available at www.compaid.com/caiinternet/ezine/symons-cosmic.pdf

[13] van Lamsweerde, Axel, Requirements Engineering - From System Goals to UML Models to Software Specifications", Wiley, January 2009.

[13] Charles Symons. Sizing and estimating for Real-time Software – The COSMIC FFP method. Software Tech September 2006, vol. 9, number 3.

[14] A. Hoshtein, R. Zarnekow, W. B Brenner. ITIL as common practice reference model for IT service management: formal assessment and implications for practice. IEEE International Conference on e-Technology, e-Commerce and e-Service, 2005

Author CVs

Sanae SAADAOUI

Dr Sanae Saadaoui is a senior researcher at the Centre of Excellence in Information and Communication technologies (CETIC). She is expert in software measurement and process improvement. She is involved in European technology transfer research projects and local industrial services.

Annick Majchrowski

Annick Majchrowski is currently working in the role of researcher/senior project manager at the Centre of Excellence in Information and Communication Technologies in Belgium. As researcher, Annick works on the domain of software and system engineering where she focuses on the quality of software products and software processes. Her research interests are in software process assessments and improvements. As senior project manager, Annick is responsible for service missions in companies with focus on quality management in IT domain.