



HAL
open science

Functional Size Estimation Technologies for Software Maintenance

Christof Ebert, Hassan Soubra

► **To cite this version:**

Christof Ebert, Hassan Soubra. Functional Size Estimation Technologies for Software Maintenance. IEEE Software, 2014, 31 (6), pp.24 - 29. 10.1109/ms.2014.138 . hal-04554322

HAL Id: hal-04554322

<https://hal.science/hal-04554322>

Submitted on 22 Apr 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/273393931>

Functional Size Estimation Technologies for Software Maintenance

Article in IEEE Software · November 2014

DOI: 10.1109/MS.2014.138

CITATIONS

16

READS

315

2 authors:



Christof Ebert

Vector Consulting Services

331 PUBLICATIONS 6,050 CITATIONS

SEE PROFILE



Hassan Soubra

École Centrale d'électronique

53 PUBLICATIONS 180 CITATIONS

SEE PROFILE



Functional Size Estimation Technologies for Software Maintenance

Christof Ebert and Hassan Soubra



ESTIMATING FUNCTIONAL software size is the key input for building software models. Unlike direct effort estimates, software size estimation gives a measure of the software product itself and can be used to build objective estimation models for predicting project effort and duration, estimating defects for quality and service cost predictions, and obtaining software development productivity ratios.

Although maintenance dominates software projects, the necessary technologies for estimating maintenance effort have been rather poorly used to date. Here, we examine estimation technologies based on the COSMIC (Common Software Measurement International Consortium, ISO/IEC 19761:2011) method as it's applied to maintenance projects.

Functional Change

Software projects in industry typically fall into two camps—new software development and software evolution—with the latter happening more frequently than the former. Functional change tends to dominate software evolution; it's driven by a change request to a piece of software's functional requirements that adds, modifies, or deletes tasks or functions. In practice, change requests can be complex because they might inte-

grate both functional and nonfunctional changes. Moreover, they're described at various levels of abstraction and with several connotations that can themselves be a mix of different elements.

Figure 1 shows the functional change request process between customer and supplier. Currently, the documentation of such change is poorly performed if at all. To establish a measurement process and reduce the possibility of human errors, functional change should be well documented and modeled.

Estimating a Change's Functional Size

An estimate is a quantitative assessment of the likely outcome of a future endeavor. It's usually applied to forecast project costs, size, resources, effort, or duration. Given that estimates are, by definition, not very precise, they should always include some indication of accuracy (such as plus or minus x percent).¹⁻³

Companies use estimation to more effectively manage their budgets and schedules. Typically, they use estimation in

- project management to estimate size and effort before a project starts to better allocate resources within the project;

- developing software products or services, such as when a supplier has to estimate the effort needed for a change request or project during a bid, before having all the technical specifications available;
- outsourcing software projects because estimation provides part of the answer to the “make, reuse, or buy” question that many companies face when optimizing their software and IT production; and
- productivity improvement because what gets measured can be improved. With fierce cost pressure and worldwide competition, productivity benchmarks based on estimation techniques pinpoint how to improve and stay competitive.

The dynamics of the software market, with its increasing use of external components and adaptation of code instead of writing it from scratch, has led to extended or new kinds of technologies for estimation. Gradually, estimation has moved away from being about size toward something that focuses more on functional estimates. Standards are evolving because estimates play a crucial role in business, and enormous amounts of money are at stake.

Estimates are often confused with goals or plans. Projects are scheduled according to needs, but they aren't always in line with feasibility, and commitments are given to clients about something being “very urgent and important” before checking how this “urgency” relates to existing commitments and capacity planning. In fact, most failures in software projects come from not understanding and considering important gaps among goals, estimates, and plans.¹⁻³

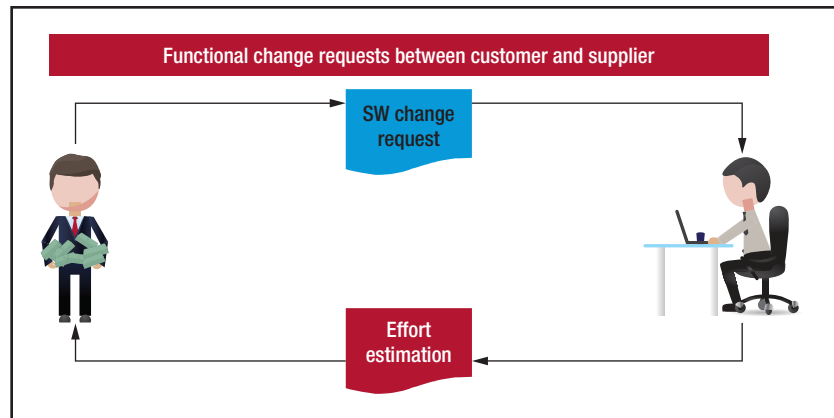


FIGURE 1. Functional change request between customer and supplier. The customer requests a change to existing software, and the supplier delivers an initial estimate as a basis for contracting.

To deliver more reliably and to better manage their projects, companies are interested in measuring and estimating the maintenance effort for their software change deliveries. Most companies use analogies or best guesses, but soon realize that the accuracy is insufficient and that there's no repeatability. So, they look for more reliable estimation technologies. The manufacturer, with this requirements measurement, can use benchmarks and negotiate with suppliers when developing its software products. For instance, during a tender phase, in which a software supplier provides an offer for a functional change or a maintenance activity, the client is interested in understanding the reasoning behind the change effort and any benefits. Customers often also want to benchmark their suppliers to find out whether their productivity is sufficient compared to other suppliers on the market.

Functional Size Measurement (FSM) is used in industry as a baseline on which to build estimates. It takes as input change requests and their analysis vis-à-vis the system in which the change must be incorpo-

rated. Based on this initial analysis of change impacts, an effort model helps derive the effort required to implement the underlying changes. Environmental factors such as software type, team distribution, or project repeatability help when adjusting the effort estimate. Obviously, the same change can cost a lot more effort if the supplier has insufficient processes or project management skills. Figure 2 shows an FSM procedure applied to a functional change to estimate the change's implementation effort. It takes the project requirements and change requests as inputs and yields functional change size for the overall effort estimation.¹

The COSMIC Method

COSMIC brings together software measurement experts from around the world; they built the method in the late 1990s to improve on traditional function point methods. The International Standards Organization classified COSMIC as a second-generation FSM method for both the business application and real-time domains. It's used worldwide in various industries, and unlike other

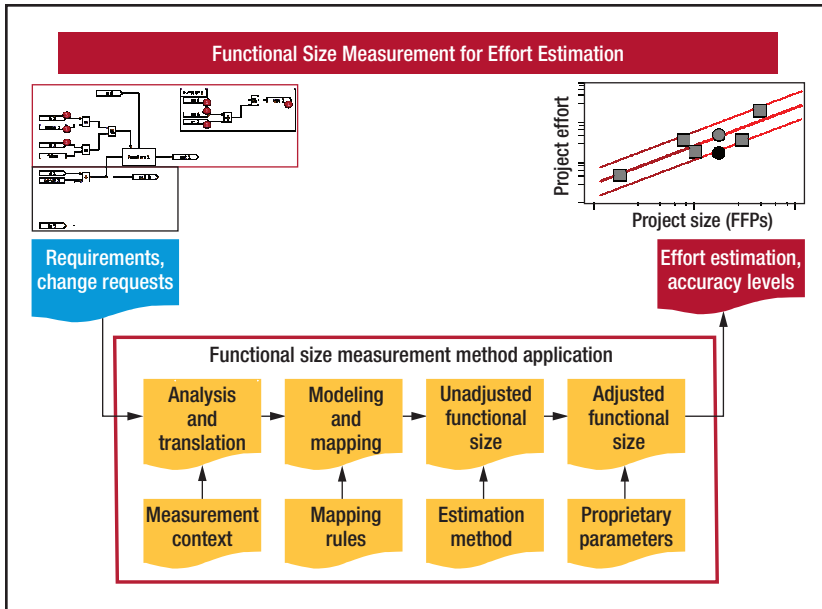


FIGURE 2. The functional size measurement (FSM) steps needed to generate a maintenance effort estimate. Effort estimation takes the functional change as an input, evaluates its impacts, and then maps to an effort model, which also considers environmental factors. The result is an effort estimate that’s adjusted to the supplier’s environment.

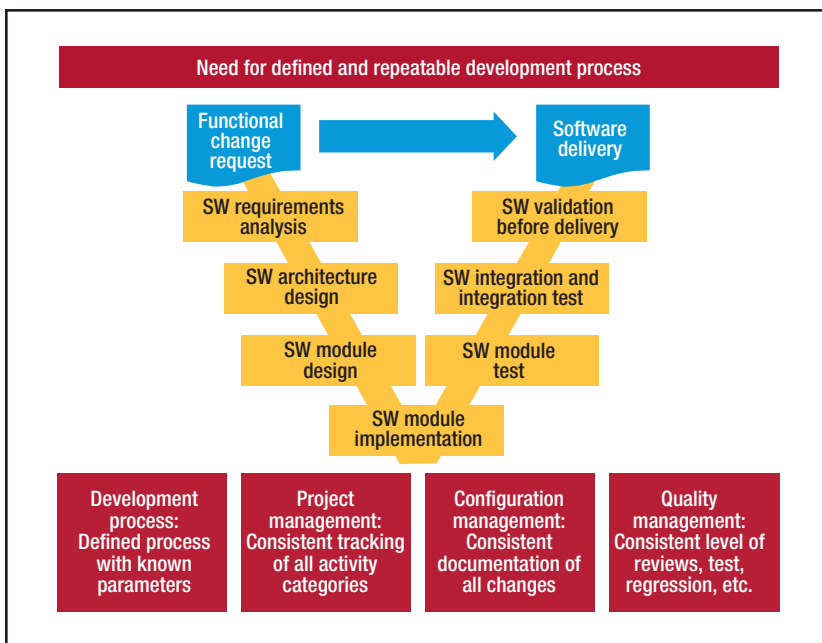


FIGURE 3. Need for a defined and repeatable development process. Without development process and project management competences, estimation can’t be improved because it lacks a baseline, and each project will evolve differently, even with the same requirements.

methods, it can be used in embedded systems.⁴⁻⁶ Although the current measurement guidelines include several refinements, the COSMIC method’s original principles have remained unchanged since they were first published in 1999.

The COSMIC method measures software’s functional user requirements (FUR). The result obtained is a numerical “value of a quantity” (as defined by the ISO) representing the software’s functional size. Functional sizes measured by the method are designed to be independent of any implementation decisions embedded in the operational artifacts of the software to be measured. This means that FUR can be extracted not only from actual software already developed but also from the software model before it’s implemented.

The measurement process consists of three phases:

- The measurement strategy phase, where a software context model is applied to the software to be measured.
- The mapping phase, where a generic software model is applied to the software to be measured.
- The measurement phase, where actual size measurement results are obtained.

Software functionality is considered within the functional flows of data groups. These dataflows can be characterized by four distinct types of data movements: two types of movements (E: ENTRY and X: EXIT) allow the exchange of data with users across a boundary, and the other two types (R: READ and W: WRITE) allow the exchange of data with persistent storage. The measurement result corresponds to the functional size of the measured software requirements

Estimation Tools with COSMIC FSM support.

Tool	Description	URL
ISBSG Comparative Estimating Tool	Used for estimation purposes; produces estimates of effort, delivery rate, duration, and speed of delivery of software projects using function points based on either COSMIC or IFPUG. The estimates are generated using delivery rates and speed of delivery averages of a selection of projects in the ISBSG repository.	www.isbsg.org
Function Point Workbench	Estimation embedded in a project management tool. Supports software project sizing and evaluation. Designed to be scalable, and hence can be used for individual projects as well as in large-scale, distributed IT environments.	www.charismatek.com
KnowledgePLAN	Software estimation tool to create and refine detailed project plans for project management systems. The tool lets users size projects and estimate work, resources, schedule, and defects. A project's strengths and weaknesses can be evaluated as well, to determine their impact on quality and productivity.	www.spr.com/index.php
MeterIT-Cosmic	Support tool that gives support to metrics practitioners for CFP, using software requirement specifications as inputs. The tool supports both MIS and real-time software, giving the possibility of estimating size when software isn't completely defined.	www.telmaco.com/cosmic1.html
MeterIT-Project version	Software benchmarking and estimation tool for predicting projects' time/cost and managing strategies; in addition to the MeterIT-Converter tool, for converting the IFPUG, NESMA, and MKII functional sizes of software to CFP.	www.telmaco.com/meteritproject1.html
Scope	Estimation embedded in a project management tool. Supports IFPUG function point analysis (FPA) method. Usable at the initial stages of project development, allowing the evaluation of a decision's impact to include or exclude functionalities. Lets users keep track of each application's functional history by providing a documented audit trail of changes to software throughout its life.	www.totalmetrics.com
Siesta	Supports sizing and estimation by using the NESMA and COSMIC methods of software functional sizing.	http://metrieken.sogeti.nl
SLIM-Estimate	Estimation environment for cost, time, and effort. Helps satisfy a given set of system requirements and determine strategies for designing and implementing software. In addition to cost estimation, it provides configurability to accommodate different design processes (agile development, package implementation, model-based development).	www.qsm.com

(FUR) and is expressed in COSMIC function points (CFP).

Functional evolution or change is interpreted as any combination of additions of new data movements or of modifications or deletions of existing data movements. The CFP count for sizing any of these changes is the same—that is, one CFP is attributed to any data group movement that has been added, deleted, or modified. However, it's important to distinguish the various types of evolution

when making performance measurements and estimates.

FSM Technologies and Tools

The entirely manual application of FSM procedures is tedious and time-consuming for organizations with a large number of projects to measure within a very short time frame, both for project estimation purposes and for productivity studies. In addition, applying FSM manually to a very large set of requirements requires specialized expertise when

these requirements are complex. Recently, technologies and tools have emerged to facilitate automating requirements. They still need formal and defined descriptions depending on modeling language, but Table 1 provides an overview of recent tools that support FSM application.

Tools obviously can only be used once environmental constraints are analyzed and an appropriate mapping of the counting method and effort reporting has been achieved (see the sidebar). Figure 3 shows

USING THE COSMIC FUNCTIONAL SIZE MEASUREMENT IN A REAL-LIFE SETTING

Let's briefly look at a concrete case study of how changes are estimated using the functional sizing method. A leading systems supplier needed to build an estimation approach with tool support that could be used worldwide with the same consistency and repeatability. Christof Ebert's company VCS was asked to establish an improved project estimation, project control, and supplier management for software and IT development. The client also wanted to use benchmarks to assess his own productivity and that of his suppliers.

In a first step, we established an internal (and confidential) measurement database for the different development and maintenance projects of that client. We used COSMIC functional size measurement because it's widely known and standardized by ISO. Obviously, there was no out-of-the-box solution, so we developed with the client specific counting rules for the raw data by applying the ISO standard and COSMIC user guide and "translating" the counting rules to the client's environment, such as how to count data inputs, outputs in the specific modeling method used. We thus ensured a consistent application of the process during analysis of functional requirements and thus the mapping to a design proposal in the modeling language.

We soon realized that maintenance effort reporting was rather heterogeneous and so had to build a consistent effort reporting with clear rules for mapping of tasks, task allocation, overtime, and so on. Both estimation and effort reporting needed clear rules, such as focus on the changed part only, no mix of different software variants in the reporting database because they have different effort drives and environmental constraints, ensuring that the same tasks are detailed, and determining functional size consistently at the same level of detail. On this basis, we achieved an estimation accuracy of 10 to 20 percent deviation between effort estimate during the project's analysis phase and the final change effort, which is suitable for most industry projects.

the environmental needs to be considered when establishing estimation. There's no automatic standard for counting because each modeling approach and environment needs specific "translations." Insufficient data quality and environmental constraints require experienced counting to avoid errors. Thus, the maturity and competences in using the development process, project management, configuration management, and quality management all

impact estimation results and must therefore be stabilized to achieve sufficient estimation accuracy.


Companies can use FSM to estimate development effort, manage project scope changes, measure productivity, benchmark, and normalize quality and maintenance ratios. We've introduced estimation for software changes here, which is increasingly

demanding both for better project estimation and also for supplier management.

We close with some observations about estimation reality:

- Consider business impacts. Clearly distinguish goals, estimates, and plans. Estimation helps mitigate business risks, so target estimation accuracy in line with your business needs.
- Establish repeatability. Immature processes will always create ad hoc behaviors with unpredictable results. Establish a robust process to report and store data.
- A fool with a tool remains a fool. Don't use models you don't know. Never use a tool to camouflage insufficient process ("the tool says so"). Carefully introduce an estimation tool and evaluate different tools, considering needs and cost. Consider total cost of ownership. Provide adequate training and coaching on estimation principles.
- Garbage in, garbage out. Tools don't provide value if the information is missing. Clarify the underlying data collection and estimation approach. Verify and validate data before storing it in historical databases. Use standard measurements.
- Use estimation to grow. Continuously improve. Don't stay with the same parameters for longer than a year, and benchmark with other companies to focus your improvements. Challenge results and improve your efficiency each year.

We achieved with client companies on this basis an estimation accuracy of 10 to 20 percent between initial effort estimation during the

analysis phase and the final effort to implement the change, which in most cases is sufficient. Such improvement in effort estimation typically takes around a year, during which it's necessary to build a history database, analyze measurements, and set up the estimation process. Naturally, the requirements as outlined above, such as sufficient process maturity and project management competences, are prerequisites to improve the accuracy of estimating maintenance effort. 

References

1. C. Ebert and R. Dumke, *Software Measurement*, Springer, 2007.
2. L. Buglione and C. Ebert, "Estimation Technologies and Tools," *IEEE Software*, vol. 28, no. 3, 2011, pp. 15–18.
3. C. Ebert, *Global Software and IT: A Guide to Distributed Development, Projects, and Outsourcing*, John Wiley & Sons, 2012.
4. "Software Engineering—COSMIC: A Functional Size Measurement Method," ISO/IEC 19761:2011, Int'l Standards Org., 2011; www.iso.org/iso/catalogue_detail.htm?csnumber=54849.
5. A. Abran et al., *The COSMIC Measurement Manual*, v. 3.0.1, Common Software Measurement Int'l Consortium; www.cosmicom.com.
6. H. Soubra et al., "Design of a Functional Size Measurement Procedure for Real-Time Embedded Software Requirements Expressed Using the Simulink Model," *Proc. 2011 Joint Conf. 21st Int'l Workshop Software Measurement and the 6th Int'l Conf. Software Process and Product Measurement*, 2011, pp. 76–85.

CHRISTOF EBERT is managing director at Vector Consulting Services, is a senior member of IEEE, and serves on the *IEEE Software* editorial board. Contact him at christof.ebert@vector.com.

HASSAN SOUBRA is an assistant professor and member of the embedded-systems research team at ESTACA—Engineering School, France. Contact him at hassan.soubra@estaca.fr.



Selected CS articles and columns are also available for free at <http://ComputingNow.computer.org>.

IEEE computer society

PURPOSE: The IEEE Computer Society is the world's largest association of computing professionals and is the leading provider of technical information in the field.

MEMBERSHIP: Members receive the monthly magazine *Computer*, discounts, and opportunities to serve (all activities are led by volunteer members). Membership is open to all IEEE members, affiliate society members, and others interested in the computer field.

COMPUTER SOCIETY WEBSITE: www.computer.org

Next Board Meeting: 16–17 November 2014, New Brunswick, NJ, USA

EXECUTIVE COMMITTEE

President: Dejan S. Milojicic

President-Elect: Thomas M. Conte; **Past President:** David Alan Grier; **Secretary:** David S. Ebert; **Treasurer:** Charlene ("Chuck") J. Walrad; **VP, Educational Activities:** Phillip Laplante; **VP, Member & Geographic Activities:** Elizabeth L. Burd; **VP, Publications:** Jean-Luc Gaudiot; **VP, Professional Activities:** Donald F. Shafer; **VP, Standards Activities:** James W. Moore; **VP, Technical & Conference Activities:** Cecilia Metra; **2014 IEEE Director & Delegate Division VIII:** Roger U. Fujii; **2014 IEEE Director & Delegate Division V:** Susan K. (Kathy) Land; **2014 IEEE Director-Elect & Delegate Division VIII:** John W. Walz

BOARD OF GOVERNORS

Term Expiring 2014: Jose Ignacio Castillo Velazquez, David S. Ebert, Hakan

Erdogmus, Gargi Keeni, Fabrizio Lombardi, Hironori Kasahara, Arnold N. Pears

Term Expiring 2015: Ann DeMarle, Cecilia Metra, Nita Patel, Diomidis Spinellis,

Phillip Laplante, Jean-Luc Gaudiot, Stefano Zanero

Term Expiring 2016: David A. Bader, Pierre Bourque, Dennis Frailey, Jill I. Gostin,

Atsuhiko Goto, Rob Reilly, Christina M. Schober

EXECUTIVE STAFF

Executive Director: Angela R. Burgess; **Associate Executive Director & Director,**

Governance: Anne Marie Kelly; **Director, Finance & Accounting:** John Miller;

Director, Information Technology & Services: Ray Kahn; **Director, Membership**

Development: Eric Berkowitz; **Director, Products & Services:** Evan Butterfield;

Director, Sales & Marketing: Chris Jensen

COMPUTER SOCIETY OFFICES

Washington, D.C.: 2001 L St., Ste. 700, Washington, D.C. 20036-4928

Phone: +1 202 371 0101 • **Fax:** +1 202 728 9614 • **Email:** hq.ofc@computer.org

Los Alamitos: 10662 Los Vaqueros Circle, Los Alamitos, CA 90720

Phone: +1 714 821 8380 • **Email:** help@computer.org

Membership & Publication Orders

Phone: +1 800 272 6657 • **Fax:** +1 714 821 4641 • **Email:** help@computer.org

Asia/Pacific: Watanabe Building, 1-4-2 Minami-Aoyama, Minato-ku, Tokyo 107-

0062, Japan • **Phone:** +81 3 3408 3118 • **Fax:** +81 3 3408 3553 • **Email:** tokyo.ofc@computer.org

IEEE BOARD OF DIRECTORS

President: J. Roberto de Marca; **President-Elect:** Howard E. Michel; **Past President:**

Peter W. Staecker; **Secretary:** Marko Delimar; **Treasurer:** John T. Barr; **Director &**

President, IEEE-USA: Gary L. Blank; **Director & President, Standards Association:**

Karen Bartleson; **Director & VP, Educational Activities:** Saurabh Sinha; **Director**

& VP, Membership and Geographic Activities: Ralph M. Ford; **Director & VP,**

Publication Services and Products: Gianluca Setti; **Director & VP, Technical**

Activities: Jacek M. Zurada; **Director & Delegate Division V:** Susan K. (Kathy) Land;

Director & Delegate Division VIII: Roger U. Fujii

revised 23 May 2014

