



Estimation Tools and Techniques

Luigi Buglione and Christof Ebert

Estimating size or resources is one of the most important topics in software engineering and IT. You won't deliver according to expectations if you don't plan, and you can't plan if you don't know the underlying dependencies and estimates. This column is an overview of estimation. It covers estimation methods and provides an overview and evaluation of popular estimation tools. —*Christof Ebert*

AN ESTIMATE IS a quantitative assessment of a future endeavor's likely cost or outcome. People typically use it to forecast a project's cost, size, resources, effort, or duration. Today's software market, with its increasing reliance on external components and adapted code, has led to new kinds of technologies for estimation, a practice that has moved from mere size-based approximations to functional and component estimates. Standards are starting to evolve as well, because these estimates play such a crucial role in business and enormous amounts of money are at stake.

Unfortunately, people often confuse estimates with goals or plans. For instance, they schedule projects according to needs instead of feasibility, or commit to something "very urgent and important" before checking how this "urgency" relates to current commitments and capacity planning. In fact, most failures in software projects come from belatedly understanding and considering the important difference between goals, estimates, and plans. Figure 1 shows how they are related.^{1,2}

Two of the most important ingredients for proper estimation are people and historical data. They're interrelated more than you

might expect because most organizations lack historical data, thus they form estimates primarily through analogy and experience. It works if you have experienced people who periodically measure and put their estimates versus actual data into a historical database. Tools can help reducing the time and costs involved in data gathering, as well as supporting reports, risk management, and scenario analysis.

Estimation Technologies

Four types of estimation techniques are regularly used today in industry practice—namely, expert judgment, analogy, decomposition, and statistical (or parametric) methods. For their origins, we recommend both the overview literature^{1,2} and the specific detailed work.³⁻⁷

Expert judgment is based on the brainstorming of one or more experts who have experience with similar projects; a consensus mechanism then produces the estimate. Analogy estimation is based on comparing previous, similar activities, analyzing the most relevant project and service attributes, and trying to figure out the new project's effort and cost values through estimator experience. As with expert judgment, this



ADDITIONAL RESOURCES

Software measurement: <http://metrics.cs.uni-magdeburg.de>
 Project estimation overview, tools, and templates (NASA): <http://software.gsfc.nasa.gov/AssetsApproved/PA1.2.1.doc>
 Function point calculation: www.ifpug.org
 Function points counting manual: www.softwaremetrics.com/freemanual.htm
 Functional size measurement: www.semq.eu/leng/sizestfsm.htm
 Effort estimation overview: www.semq.eu/leng/sizestees.htm
 Estimation white paper for nonfunctional aspects and IT productivity: www.semq.eu/pdf/fsm-prod.pdf
 Estimation methods based on SLIM: www.qsm.com
 COSMIC function points, by the Common Software Metrics International Consortium: www.cosmicon.com
 Estimating maintenance projects: www.cs.jyu.fi/~koskinen/smcems.pdf
 The PREPARE project at the SIMULA Lab: <http://simula.no/research/prepare>



Therefore, we look only at commercial off-the-shelf tools and technologies.

Most tools here overlap in their underlying functionality, so the rest of this discussion doesn't indicate bias for a specific vendor. A very important requirement to analyze in an estimation tool is to have the opportunity to run benchmarks against best-in-class projects and browse within such data. That's why all major tools have recently included the International Software Benchmarking Standards Group (ISBSG) history database (www.isbsg.org), one of the most renowned public sources of information that's continuously maintained and updated. Table 1 summarizes a collection of estimation tools.

All these tools deal with lines of code; when testing to count function points (FPs), they often simply do a backfiring (see www.computer.org/portal/web/csdl/doi/10.1109/2.471193) according to a conversion table or by manually inserting the basic information for calculating FPs. In the first case, it's fundamental to control the LOC definition spread and apply it consistently across the organization and project group, to avoid the risk of having incomparable data and inserting mistakes in your decision-making process. Another negative side effect of the backfiring practice: recent studies demonstrate the advantages of using FP-based regression models using two or more base functional components (BFCs),⁷ while the backfiring process returns only the whole number of FPs, with no splits. The precondition for having a successful estimation using FSM methods such as FPA depends on the proper storage of the single BFC values: a simple percentage distribution based on data originally counted with an FSM method shouldn't be necessarily valid or particularly precise, reducing the reliability of such a project repository. In the second case (manually inserting FSM counts), the tool

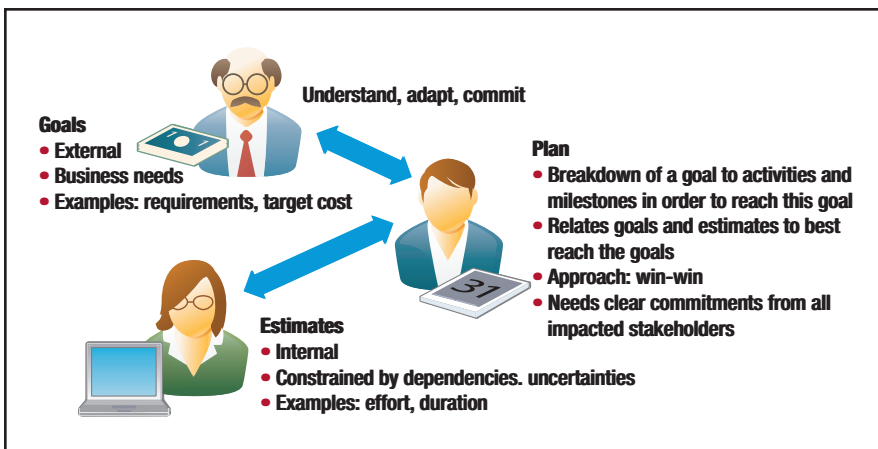


FIGURE 1. Goals, estimates, and plans interfere with each other. Start with realistic goals: estimate the underlying work, schedule, and effort. Then, relate goals to estimates with a feasible plan.

technique requires skilled people who can properly understand and see relationships and implicitly evaluate qualitative and quantitative figures to determine possible clusters of projects. Decomposition is a top-down estimation technique that tries to make a granular list of initially planned tasks. The more granular the tasks associated with a certain requirement in a work breakdown structure (WBS), the closer the planned effort is with its final value, thereby reducing the mean relative error and possible slippage in project deliverables. Statistical (parametric) models

are a set of related mathematical equations in which you define alternative scenarios by changing the assumed values of a set of fixed coefficients (parameters). Software project managers use such models or parametric estimation tools to estimate a project's duration, staffing, and cost.

Estimation Tools

Most estimation tools are proprietary due to the huge effort to consolidate underlying history databases. This explains the lack of mainstream open source software estimation tools.

TABLE 1

Software estimation tools.

Tools	Producer	LOC/FP based	Single (S)/ multi (M) nodes	Online	Proprietary	Number of projects	ISBSG r11 data	Complexity (high/ medium/low)	Size (S)/ estimation (E)	Price range (single license)
COCOMO family (http://csse.usc.edu/tools/COCOMOSuite.php)	USC	LOC	S	Y	N	n/a	N	L	E	Free
Comparative Estimating Tool (www.isbsg.org)	ISBSG	FP	S	N	Y	5,000+	Y	L	E	Low
CostXpert (www.costxpert.eu)	CostXpert Group	LOC	M	N	Y	n/a	N	H	S+E	Mid
Early Estimate Checker (www.isbsg.org)	ISBSG	FP	S	N	Y	5,000+	Y	L	E	Mid
Function Point Workbench (www.charismatek.com.au)	Charismatek	FP	S	N	Y	n/a	N	M	S+E	Low
FP Outline (www.totalmetrics.com)	Total Metrics	FP	S	N	Y	2,000+	N	L	E	Mid
KnowledgePlan (www.spr.com/spr-knowledgeplanr.html)	SPR	FP	S	N	Y	14,000+	U	M	S+E	Mid
PQM Plus (www.qpmg.com)	Q/P Management Group	FP	M	N	Y	n/a	N	M	S+E	Mid
Reality Checker (www.isbsg.org)	ISBSG	FP	S	Y	Y	5,000+	Y	L	E	Mid
SEER-SEM (www.galorath.com)	Galorath	LOC	S	N	Y	n/a	Y	H	S+E	Mid
SLIM (www.qsm.com/tools/index.html)	QMS	LOC	M	N	Y	8,000+	Y	H	S+E	High
SMRe (www.qpmg.com)	Q/P Management Group	FP	M	N	Y	n/a	N	M	E	Mid

wouldn't cut time and costs for functional analysts but would simply represent a project database.

Hints for Practitioners

Estimation implementation in a large organization might take two years. Gaining estimation experience and integrating it into project management processes plus the consequent introduction of IT measurements for continuing improvement might require another two years.

Implementation costs are often cited

as an argument against systematic and professional estimation. But considering the effort involved, you can easily predict that just one failed IT project will cost more than all the effort required to implement and support sound methods for estimation and software measurement. Here's a suggested "to do" list for making your case and improving your estimates:

- Data is the resource, information brings the value. The first step is to determine your own informa-

tion goal and establish the proper process to gather, verify, and validate data before their storage in historical databases. The second step is the analysis, bringing valuable information for the decision-making process.

- Collect your own data on a regular basis and at the right level of granularity. Estimation models based on multiple regression analysis are more effective than using a single independent variable. Thus, it's better to use multiple BFCs from

FP-based projects than the final FP value related to the project work effort.

- Reporting isn't a second-level process—the decision-making process comes from information, not from single data. Report presentation can dramatically change the way the original data would suggest to take a certain action.
- Estimation should be viewed as cause-effect logic. What is forecast for a certain phenomenon will affect other processes. Without clarifying the series of links, the risk is to over- or underestimate more than expected.
- Target an estimation accuracy in line with business needs. Many of our clients improved their precision 10 to 20 percent, which in most cases is sufficient.
- Use estimation tools to grow. If not properly applied, parametric models could implicitly reduce an organization's willingness to grow and mature.

The ultimate tip is to avoid becoming complacent with formulas for your own

productivity and cost drivers. Challenge them and improve your efficiency each year with focused improvements.

From our own experience, estimation and measurement are insufficiently used across companies. Later, when projects are facing problems, it's too late to restart a proper estimation. We thus recommend starting off on the right foot with the next project and introducing or improving your estimation with a closed loop of estimation, planning, measuring, and periodically improving your estimates. That said, you can and should immediately strive to improve your data quality, estimation precision, and, of course, your efficiency. In the famous words of E. Deming, "In God we trust. All others bring data." ☞

References

1. S. McConnell, *Software Estimation: Demystifying the Black Art*, Microsoft Press, 2006.
2. C. Ebert and R. Dumke, *Software Measurement: Establish, Extract, Evaluate, Execute*, Springer, 2007.
3. P. Hill, ed., *Practical Software Project Estimation: A Toolkit for Estimating Software Development Effort & Duration*, McGraw-Hill, 2010.
4. C. Jones, *Estimating Software Costs: Bringing Realism to Estimating*, McGraw-Hill, 2007.
5. L.H. Putnam, "A General Empirical Solution to the Macro Software Sizing and Estimating Problem," *IEEE Trans. Software Eng.*, vol. 4, no. 4, 1978, pp. 345–361.
6. B.W. Boehm et al., *Software Cost Estimation with COCOMO II*, Prentice Hall, 2000.
7. L. Buglione and C. Gencel, "The Significance of IFPUG Base Functionality Types in Effort Estimation," *Proc. 5th IFPUG Int'l Software Measurement & Analysis Conf.*, 2010; www.optionbrasil.com.br/evento/isma5_2010/programacao.asp?X=&d=16.

LUIGI BUGLIONE is a process improvement and measurement specialist at Engineering.IT SpA, Italy, and associate professor at the École de Technologie Supérieure (ETS) in Montréal. Contact him at luigi.buglione@eng.it.

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



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

IEEE Software (ISSN 0740-7459) is published bimonthly by the IEEE Computer Society. IEEE headquarters: Three Park Ave., 17th Floor, New York, NY 10016-5997. IEEE Computer Society Publications Office: 10662 Los Vaqueros Cir., Los Alamitos, CA 90720-1314; +1 714 821 8380; fax +1 714 821 4010. IEEE Computer Society headquarters: 2001 L St., Ste. 700, Washington, DC 20036. Subscription rates: IEEE Computer Society members get the lowest rate of US\$54 per year, which includes printed issues plus online access to all issues published since 1984. Go to www.computer.org/subscribe to order and for more information on other subscription prices. Back issues: \$20 for members, \$173.33 for nonmembers (plus shipping and handling).

Postmaster: Send undelivered copies and address changes to *IEEE Software*, Membership Processing Dept., IEEE Service Center, 445 Hoes Lane, Piscataway, NJ 08854-4141. Periodicals Postage Paid at New York, NY, and at additional mailing offices. Canadian GST #125634188. Canada Post Publications Mail Agreement Number 40013885. Return undeliverable Canadian addresses to PO Box 122, Niagara Falls, ON L2E 6S8, Canada. Printed in the USA.

Reuse Rights and Reprint Permissions: Educational or personal

use of this material is permitted without fee, provided such use: 1) is not made for profit; 2) includes this notice and a full citation to the original work on the first page of the copy; and 3) does not imply IEEE endorsement of any third-party products or services. Authors and their companies are permitted to post the accepted version of IEEE-copyrighted material on their own web servers without permission, provided that the IEEE copyright notice and a full citation to the original work appear on the first screen of the posted copy.

Permission to reprint/republish this material for commercial, advertising, or promotional purposes or for creating new collective works for resale or redistribution must be obtained from IEEE by writing to the IEEE Intellectual Property Rights Office, 445 Hoes Lane, Piscataway, NJ 08854-4141 or pubs-permissions@ieee.org. Copyright © 2011 IEEE. All rights reserved.

Abstracting and Library Use: Abstracting is permitted with credit to the source. Libraries are permitted to photocopy for private use of patrons, provided the per-copy fee indicated in the code at the bottom of the first page is paid through the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923.