BENCHMARKING IS AN ESSENTIAL CONTROL MECANISM FOR MANAGEMENT

Ton Dekkers
President of the ISBSG (International Software Benchmarking Standards Group)
ton.dekkers@shell.com

Abstract For management, discussions on budget and outsourcing become more transparent when motivated by facts. Benchmarking, internal and external is the only way to make it achievable. In addition benchmarking requires more standardisation, another focus area in most large organisations.

1. INTRODUCTION

Performance and outsourcing has become issues in projects and service contracts. Main drivers for this are increasing business value (and often shareholder value), cutting cost and earlier time-to-market.
Knowing your performance is the best way to validate the expectations and benefits of improvements and outsourcing deals. Benchmarking repository data is objective reference for decision-making. To be able to benchmark the performance, some basic questions have to be answered:

- What is (software) benchmarking?
- Which performance indicators should be measured?
- What reference is available?

2. BENCHMARKING

Benchmarking is the process of continuous measuring and comparing activities and/or products with each other. The areas and conditions for comparison should be defined properly and unambigious. Another essential precondition is a repeatable measurement process. The availability of internal measurement data will make the benchmarking process more valuable.

Software benchmarking can be regarded as a specific domain. Internal benchmarking is relevant for detecting improvement possibilities and for the outsourcing discussion. External benchmarking will help to set goals for improvement, could lead to organisational restructuring and/or decision on outsourcing and outsourcing supplier.

At the moment the most benchmarking data available is on new development of tailor-made software. For 'run and maintain' activities the data is limited. A benchmark repository for business process packages acquisition and implementation is under construction.

3. MEASUREMENT MODEL

In manufacturing, the input-process-output model is quite common in business economics. Why not apply this model to software economics.

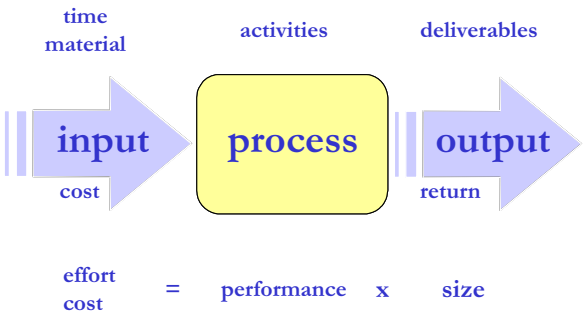


Figure 1. IPO/Nominal Measurement Model

In this case, the IT staff (effort) develops (process) the tailor-made software (deliverables). When the variables are defined, the performance can be measured. The model can be used for evaluation (with input and output known, the actual process performance can be determined) and prediction (with output and process performance, the input can be estimated).

Of course everybody agrees upon business or project managers that will mention that this project has some specific conditions that makes it different. This is reflected in the enhanced measurement model (figure 2).

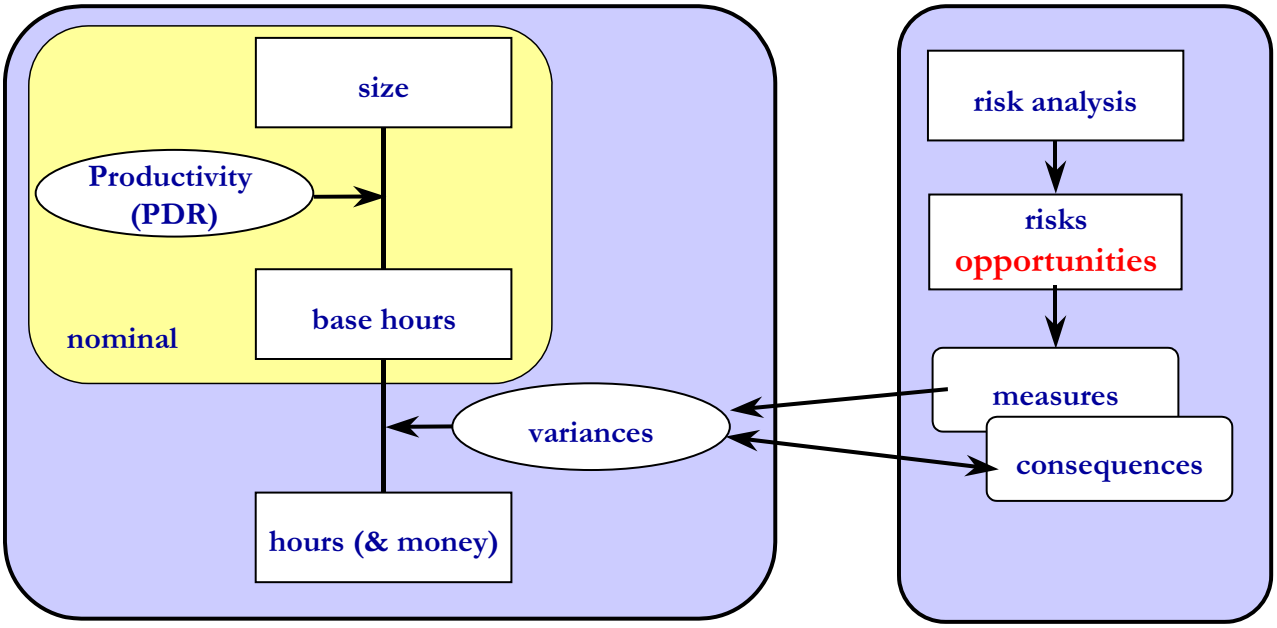


Figure 2. Enhanced Measurement Model

Performance Indicator	"Formula" Definition
Project Delivery Rate [Hours per unit]	Effort spent / size of the application Measures the rate at which a project delivers software functionality to the end user as a factor of the effort required to do so. It is defined as Project Work Effort (measured in hours), over Functional Size of the delivered software (measured in size units). Project delivery rate is used regardless of how the software is produced.
Speed of Delivery [Units per period]	Size of the application / elapsed time needed for delivery Measures the ability of a team to deliver a quantity of software over a period of time. It is defined as the Functional Size of the delivered software (measured in size units), over the Project Elapsed Time (measured in months).

Table 1. Definitions

When measuring performance, size matters. This requires proper size measurement. Fortunately the IEC/ISO standard 14143-1 defines the principles of a functional size measurement method. At the moment following methods are compliant (certified by IEC/ISO): Function Points Analysis according IFPUG (ISO 20926), Function Points Analysis according NESMA (ISO 24570), Mark II Function Points (ISO 20968) and COSMIC Full Function Points (ISO 19761). All mentioned is valid for these standards.

4. INTERNATIONAL SOFTWARE BENCHMARKING STANDARDS GROUP

The ISBSG is an International 'not-for-profit' organisation with 13 members. Members are software measurement associations like IFPUG (USA, Brazil, …), ASMA (Australia), GIFPU (Italy), NESMA (Netherlands), NASSCOM (India), CSPIU (China) and JFPUG (Japan).

Based on a questionnaire, data is collected from all over the world to fill the benchmarking repositories. Data available is almost completely related to tailor-made software. The "New & Enhancement Projects" Release 10 contains data of over 4,000 projects. The repository "Maintenance & Support" comprises 115 applications or programs. The data can be acquired directly by ISBSG or by the members associations.
Under construction are repositories for "business systems software package acquisition and implementation".

With this benchmark data it is possible to validate performance, estimates and proposals.

5. PRACTICAL USE OF A BENCHMARK REPOSITORY

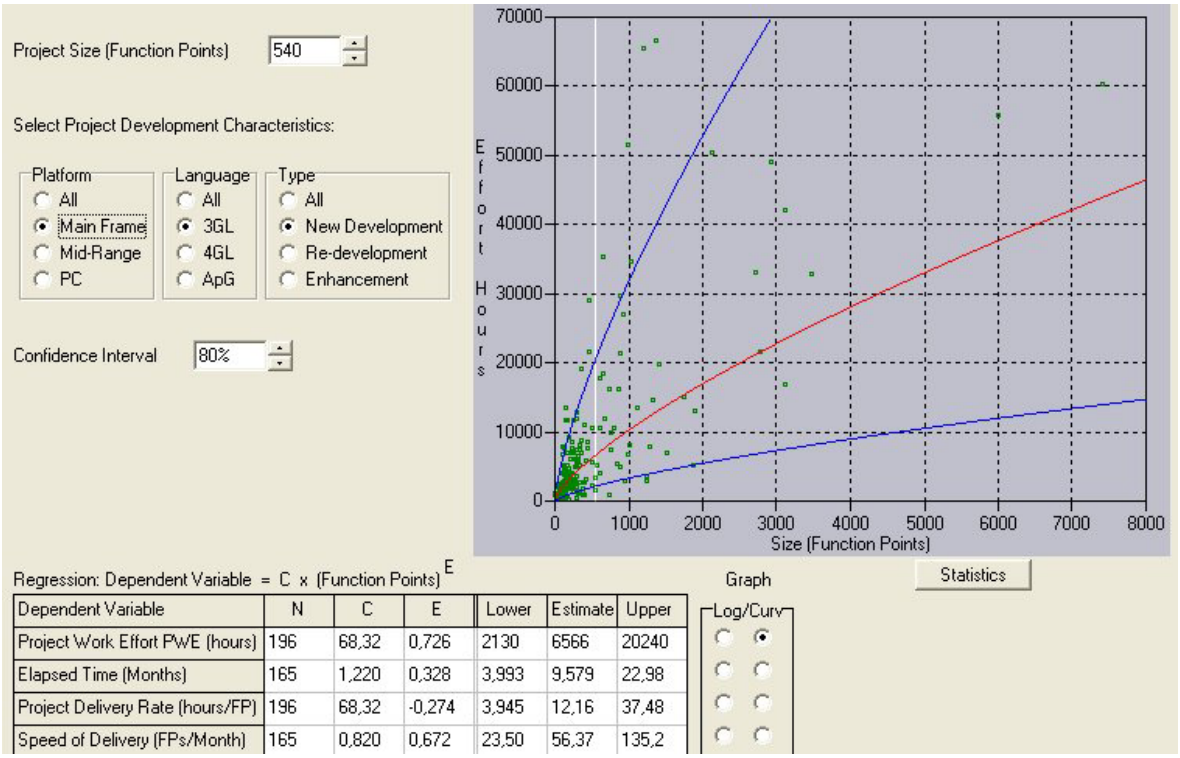
The validation with the use of ISBSG benchmarking data is shown in this real-life case. A request for proposal is sent out to the various suppliers. In the table the main project characteristics. The base for the size calculation is provided as part of the information package.

Project size	540 function points
Domain	Business application
Language	Cobol
Platform	Mainframe
Constraints	Duration: 10 months Cost: 1,000,000 Euro
An average hourly rate of 100 Euro is used.	

Table 2. Case characteristics

For a quick assessment of the characteristics, the reality checker is used (included in the ISBSG repository package). The screenshot shows the results based of the matching repository data (figure 3). A summery is given in Table 3.

The assessment uses the nominal measurement model. The expected cost (effort * hourly rate) are in range with the bandwidth of the repository selection. The duration however is less in balance. Next steps are assessing the expectations with the private data set and verifying data sets used for the benchmark with the project specific circumstances. Applying the identified variances in the enhanced measurement model will give calibrated results to validate the expectations and constraints.



Dependent Variable	N	C	E	Lower	Estimate	Upper
Project Work Effort PWE (hours)	196	68,32	0,726	2130	6566	20240
Elapsed Time (Months)	165	1,220	0,328	3,993	9,579	22,98
Project Delivery Rate (hours/FP)	196	68,32	-0,274	3,945	12,16	37,48
Speed of Delivery (FPs/Month)	165	0,820	0,672	23,50	56,37	135,2

Figure 3. Reality Checker v3.0 - Release 9

	Estimate	Reality Checker v3.0 – R9
Project size	540 function points	540 function points
Domain	Business application	-
Language	Cobol	3 GL
Platform	Mainframe	Mainframe
Constraints	Duration: 10 months Cost: 1,000,000 Euro	Duration: 9.5 - 23.0 months Cost: 656,000 - 2,024,000 Euro

Table 3. Example project - Reality Checker v3.0 - Release 9

Commercial tools can give similar support. Parametric estimating tools like the SLIM suite (QSM), KnowledgePlan (SPR), the SEER suite (Galorath) and Experience Pro (STTF) are based on similar benchmarking data sets. Their advantage is a more professional approach on estimating where ISBSG is focussing on the data set. The advantage of the ISBSG data set is the possibility of creating your own peer group and having access to all underlying data elements. The professional tools only provide a 'black box' external peer group benchmarking. When an internal data set is available, then the underlying data is available.

Whatever solution is chosen, using internal or external benchmark repositories for validating performance will validate the business expectations.

REFERENCES

[1] ISBSG, "The ISBSG Estimation, Benchmarking & Research Suite (release 9)", International Software Benchmark Standards Group, 2005, http://www.isbsg.org.au.

[2] ISO, "Information Technology - Software Measurement - Functional size measurement: ISO/IEC 14143-1", International Organization for Standardization 1998.

[3] IFPUG, "Function Point Counting Practices Manual, version 4.2, International Function Point Users Group, 2004, http://www.ifpug.org.

[4] NESMA, "Definitions and counting guidelines for the application of function point analysis A practical manual, version 2.2", Netherlands Software Measurement user Association, 2004 (in Dutch), http://www.nesma.org.

[5] UKSMA, "MK II Function Point Analysis, Counting Practices Manual, version 1.3.1", United Kingdom Software Metrics Association, 1998, http://www.uksma.co.uk.

[6] COSMIC, "COSMIC FFP Measurement Manual 2.2", Jan. 2003, http://www.lrgl.uqam.ca/cosmic-fpp.

.