

The Software Engineering Body of Knowledge: A Benchmarking Tool for Organizational Process Assessment and Improvement – Case Study

Mohammad Zarour¹, Jean-Marc Desharnais² and Alain Abran²

¹College of Computer and Information Sciences, Department of Software Engineering P. O. Box 66833, Riyadh 11586 Kingdom of Saudi Arabia; mzarour@psu.edu.sa

²École de Technologie Supérieure, Department of Software Engineering and IT 1100 Notre-Dame west, Montréal, Québec H3C 1K3, Canada; jean-marc.desharnais@etsmtl.ca, alain.abran@etsmtl.ca

Abstract

The Guide to the Software Engineering Body of Knowledge (the SWEBOK Guide) represents the consensus on the knowledge that software engineers, and their organizations, should use whenever and wherever appropriate in software development. This paper presents an innovative use of this SWEBOK Guide as a benchmarking reference for software organizations interested in process improvement and looking for best practices. Process improvement approaches help organizations improve their processes and their performance. Before implementing improvements to existing processes, it is necessary to benchmark organization's practices already in place against a reference, identifying process weaknesses and looking for best practices that can contribute to process improvement according to corporate priorities. This paper presents two industry case studies illustrating the use of the SWEBOK Guide for benchmarking purposes and process improvements. This paper presents also quantitative results of productivity and quality analyses in both organizations and discusses the candidate linkages.

Keywords: Appraisal, Best Practices, Process Improvement, SWEBOK

1. Introduction

The Guide to the Software Engineering Body of Knowledge (SWEBOK) is a key project designed to contribute to the recognition of the software engineering as an engineering discipline: this project has been sponsored by the IEEE Computer Society and supported by a consortium of industrial sponsors.^{1,2} Later on, the SWEBOK was adopted as an ISO technical report (ISO/IEC TR 19759:2015).² The main goal of the SWEBOK was to develop an international consensus on the “generally

accepted knowledge” in the software engineering domain. This criterion of generally accepted knowledge, to be distinguished from advanced and research knowledge (on the grounds of maturity) and from specialized knowledge (on the grounds of generality of application). This definition comes from the Project Management Institute: “The generally accepted knowledge applies to most projects, most of the time and widespread consensus validates its value and effectiveness”.

ISO/IEC TR 19759:2015 standard has been developed based on SWEBOK version3 and documents the

*Author for correspondence

consensus on the structure of the software engineering knowledge¹. It consists of fifteen (15) Knowledge Areas (KAs): the first ten KAs represent what the ISO calls the primary processes in the ISO 12207 standard, another KA called Software engineering professional practice which is concerned with the knowledge, skills, and attitudes that software engineers must possess to practice software engineering³. The remaining four KAs are concerned with the educational requirements of software engineering. The scope of this research work is covered by the first ten KA as depicted in Table 1. For each knowledge area, there are a number of topics, and for each topic there are sub-topics – Note that the assessment tool used in this research has been developed prior to publishing SWEBOK V3 hence is based on SWEBOK:2004 version³.

Software products are produced through a development process; hence, software product quality is directly affected by that process. The efforts of software engineering to achieve high quality products developed in time and within budget, have been based on this principle. Accordingly, several model-based Software Process

Improvement (SPI) approaches (e.g. CMMI, bootstrap, ISO 15504 and ISO 9001) have been designed to improve productivity, product quality and customer satisfaction through process improvement⁴. The initiatives associated with these approaches all begin with an assessment of the organization's current processes. The results of this assessment provide data on the strengths and weaknesses of the organization's processes and provide guidelines with respect to which processes require improvement.

One of the key benefits of the SWEBOK Guide is that the knowledge and practices specified for each knowledge area can be used to design an assessment tool (realized as a questionnaire) to measure the coverage of organizational practices in use for each process with the generally accepted practices specified in the SWEBOK Guide. In this paper the SWEBOK Guide is used as a tool to conduct the process assessment.

The purpose of this article is to illustrate, practically, the use of the SWEBOK Guide for benchmarking purposes and process improvements through case studies. This paper explores also, quantitatively, the candidate

Table 1. Guide to the SWEBOK: Knowledge Areas²

SWEBOK Knowledge Areas (KA)	ISO 12207 Process Type
Software requirements	Primary Process
Software design	
Software construction	
Software testing	
Software maintenance	
Software configuration management	Support and Organizational Process
Software engineering management	
Software engineering process	
Software engineering models and methods	
Software quality	

linkages between productivity and quality analyses in both organizations. Henceforth, we can determine if this questionnaire is useful to evaluate the practices of the organization relative to the effort it takes to do it. The usefulness will be found by comparing the finding from the questionnaire with the finding of benchmarking results relative to productivity (development and/or maintenance) and quality of the product that was done at the same period. Accordingly, two case studies are presented, one at a public-sector organization and the second at a private sector organization. For both organizations the benchmarking study included as well the development of a quantitative estimation model for each organization and a quality benchmarking using their own data as well as data from an international repository of software projects, namely the ISBSG repository.⁵

Practices of software maintenance have not been covered for the first organization; the scope of the benchmarking was limited to software Development. Finally, there will be a discussion on resemblances and differences of the results between both organizations.

This paper is structured as follows: Section 2 presents the design of the assessment questionnaire and process benchmarking with the SWEBOK Guide. Section 3 presents the data collected in the two case studies. Section 4 presents the analysis of the data collected. Section 5 presents the discussion and Section 6 presents the conclusion and future works.

2. The Process Assessment Questionnaire based on the SWEBOK Guide

An assessment questionnaire was designed to address each SWEBOK topic and sub-topic within each Knowledge Area. For example, the question for Software Requirements Fundamentals is the following:

Q1. Does your method or development process define functional requirements and/or non-functional requirements?

A. There are four possible answers:

- B. The two types of requirements are present,
- C. No functional requirement,
- D. No non-functional requirement
- E. None is present

If the answer is 'none', then it is not necessary to go to sub-topics. If the answer is 'two types of requirements are present' or if one of the two other answers is present, then there are further questions. For each sub-topic, there is a possibility of more than one question. For example, for 'Definition of Software Requirement' there are four questions:

1. Are all your software requirements formulated in a way to solve a problem?
2. Are your requirements verifiable?
3. Are your requirements verifiable within the available resources?
4. Are they uniquely identified so they can be subject to configuration control software and be monitored throughout the life cycle?

The total number of questions could be as high as 400 when all practices at the topic level are present within an organization. For this benchmarking exercise, equal weights are assigned to each question. The questionnaire was used in each organization and different groups of personnel were interviewed.

3. Summary of the Data Collected in the Two Case Studies

3.1 Process Coverage of SWEBOK Practices: Case Study 1

The assessment of the organization in case study 1 was conducted in a governmental organization having

Table 2. Coverage of SWEBOK practices – Case Study 1

SWEBOK Knowledge Areas	Coverage of SWEBOK practices
Software Design	89 %
Software Management	75 %
Configuration Management	74 %
Requirements Engineering	73 %
Software Construction	71 %
Software Testing	61 %
Tools and Methods	54 %
Software Engineering Process	44 %
Software Quality	38 %

more than 500 information technology employees (including external consultants); it was developing mostly Management Information System (MIS) software, with between 15 to 20 major software projects yearly. It took two days to conduct the assessment using the SWEBOK questionnaire. The practices from the maintenance knowledge area were not included within the scope of this assessment.

In addition, 16 software development and improvement projects were measured in terms of functional size, effort, and various environment qualifiers. The staff who developed these projects included both internal and external developers, distributed equally overall. The benchmarking results are used for interpretation purposes

The percentages of software engineering practices in use for each knowledge area are presented in Table 2, in decreasing order of coverage. A preliminary version of the benchmarking results for the first organization has

been published in the presented article.⁶ These percentages represent the ratio between the practices observed to be in general used in this organization, divided by the total number of practices listed in the questionnaire for each of the SWEBOK knowledge areas.

From the five knowledge areas related directly to the development life cycle (e.g., from requirements engineering up to software testing) the coverage of over 60% of the SWEBOK practices indicates a widespread use of organizational processes across this organization: this means that the software engineering processes in this organization have been deployed and are in use across this organization.

To help the organization understand quickly its position regarding different knowledge areas and sub-topics, graphical representations of these data are presented in Figures 1 and 2. On the nine (9) knowledge areas within the scope of the assessment (excluding maintenance) the

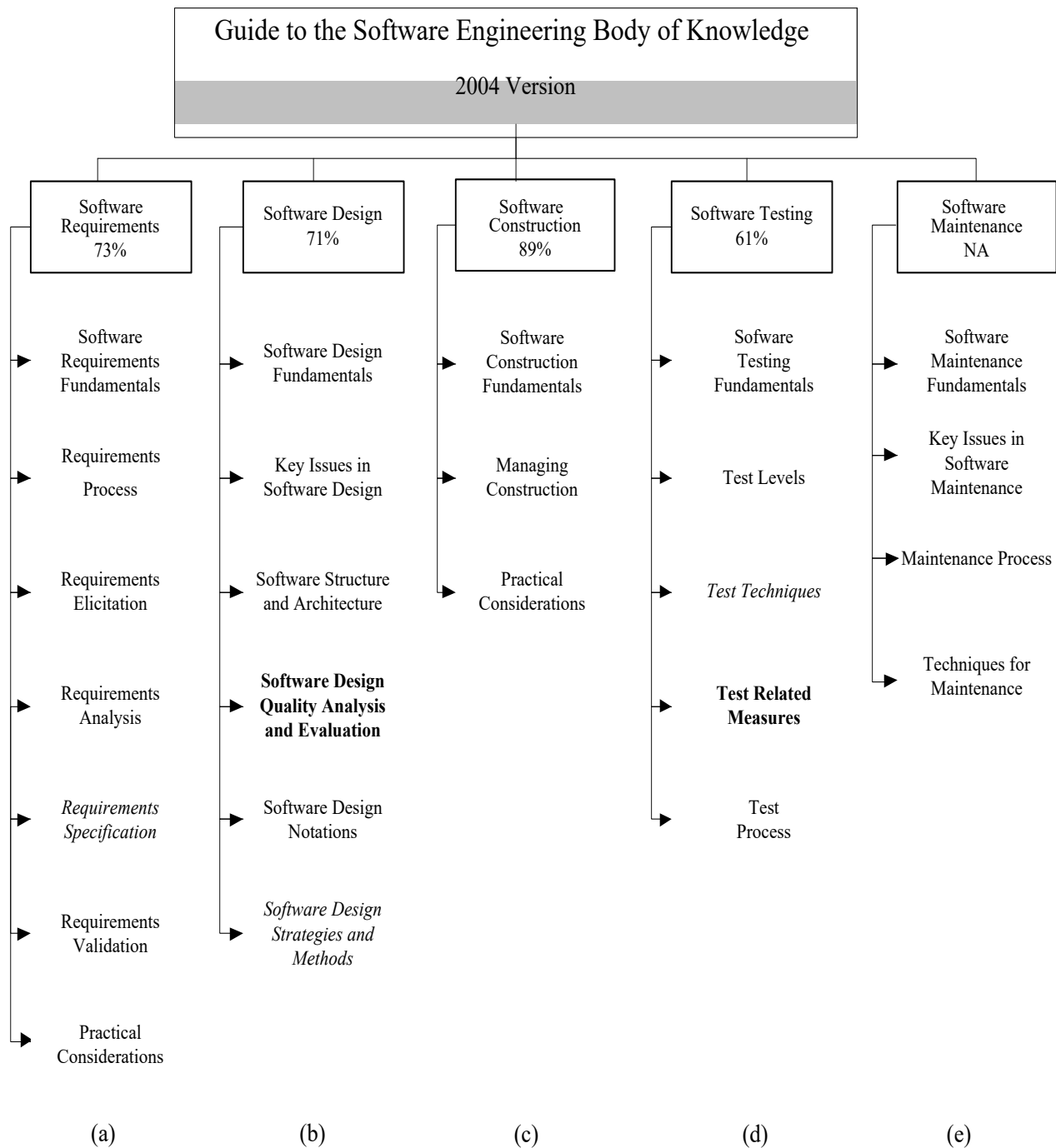


Figure 1. Coverage of SWEBOK practices in the Software Life Cycle knowledge areas – Case Study 1.

results show a generally good cover of each knowledge area in general:

- six knowledge areas covered more than 60% of the SWEBOK practices (software requirement, software design, software construction, software testing, software configuration management and software engineering management);
- two other knowledge areas covered between 40% and 59% (software engineering process and software engineering tools and models);
- one knowledge area achieved less than 40% (software quality).

The sub-topics with the least coverage in this case study are (Figure 2):

- Software Configuration Auditing under Software Configuration Management,
- Software Engineering Measurement under Software Engineering Management, and
- Process and Product Measurement under Software Engineering Process.

In addition, the knowledge area of Software Quality is rated with the lowest level and with two of its three sub-topics (Software Quality Fundamentals and Software Quality Process) rated below 40% and its third one (Practical Considerations) rated between 40%-59%.

3.2 Process Coverage of the SWEBOK Practices: Case Study 2

The second case study is from a private sector organization with 75 information technology employees at the time of the assessment, and was quickly growing (over 125 IT employees 12 months later). The organization had two major applications (MIS type applications) with mainly

adaptive and corrective maintenance over the years. The assessment took a single day. The process evaluation covered the 10 knowledge areas (including the Maintenance) as well as the supplemented knowledge area proposed in 2009 to SWEBOK, that is, the candidate measurement knowledge area. In this second case study, the assessment results show that many practices from the SWEBOK knowledge areas were not covered: out of over 400+ assessment questions, only 163 have been answered. Table 3 present the details of the number of yes (30), number of no (106) and number of partial (27) and the percentage of yes (18%) on the number of questions. When the answer is yes one point is added. For Partial answer, half of a point is added, and for No, 0 point is added.

The percentage takes in account the sum of the points to the number of questions. For example, the Software Engineering Management knowledge area has 15 questions, and this organization for case 2 ended up with 8 Yes (8 points), 6 No (0 point) and 1 partial (= +.5 point), for a total of 8.5 (out of a maximum of 15), or a 57% coverage of SWEBOK practices (where each practice has been assigned an equal weight).

This organization did not have a single knowledge area with coverage of more than 60% of SWEBOK practices: Software Management coverage of practices is 57% while Software Construction and Software Maintenance coverage was at 50%. Others were at coverage of less than 40%. Software measurement gained 20% coverage of SWEBOK practice, but they were implemented in the months prior to the assessment. For the knowledge areas of Software Testing and Software Quality the results were even null 0%, see Table 3.

4. Analysis

The analysis is presented for each case study, including references to the benchmarking.

4.1 Case Study 1

4.1.1 SWEBOK Analysis

In case study 1, the software quality knowledge area had

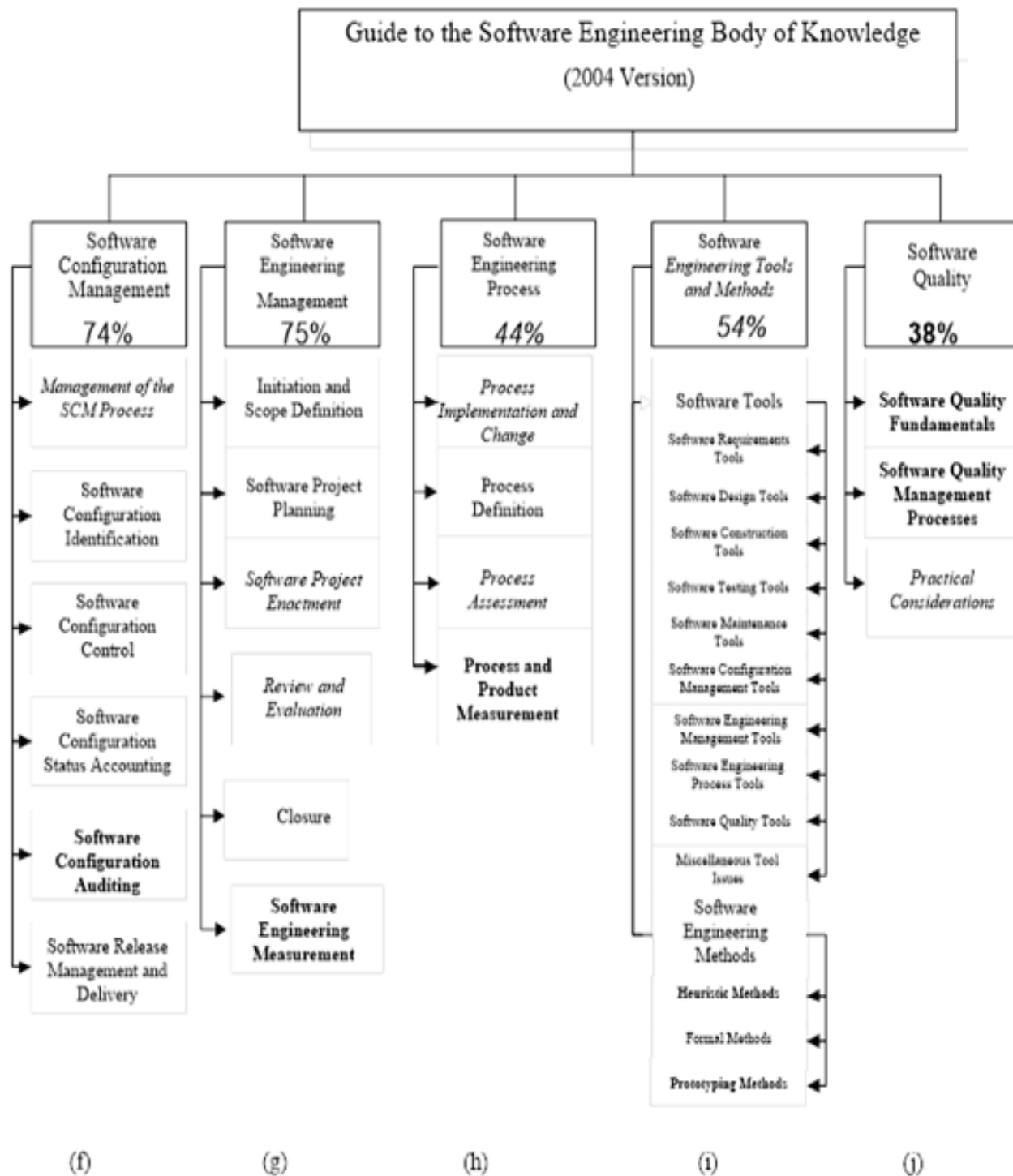


Figure 2. Support to Software Life Cycle knowledge areas – Case Study 1.

by far the lowest coverage of best practices with only 38%. Figure 3 shows the sub categories (highlighted in bold) that contribute to this lowest rate:

- In the Software Quality Fundamentals, there is almost no processes related to: Culture and Ethics, Value and Cost of Quality and Quality Improvement. Two of them are related to the measurement.

- In the Software Quality Management Processes the percentage is even lower with only 12% mainly because of absence of Verification and Validation and Reviews and Audits.

The coverage is much less extensive in support areas, such as Tools and Methods, Software Engineering Process, and Software Quality, than in development knowledge areas. For this reason, the organization did not have any

Table 3. Coverage of SWEBOK practices - per KA in decreasing order – Case study 2

Knowledge Areas	Questions answered	YES	NO	PARTIAL	Coverage of practices (e.g. Yes)
Software Engineering Management	15	8	6	1	57%
Software Construction	14	4	4	6	50%
Software Maintenance	12	4	4	4	50%
Software Requirements	30	7	18	5	32%
Software Configuration Management	16	3	9	4	31%
Software Design	19	2	12	5	24%
Software Engineering models & methods	7	1	5	1	21%
Software Measurement	5	1	4	0	20%
Software Engineering Process	9	0	8	1	6%
Software Testing	28	0	28	0	0%
Software Quality	8	0	8	0	0%

measurable information on the effectiveness of the implementation of their practices and the benefits derived, that is, quantitative information to support the decision-making process. The organization had over the recent years implemented many new practices but it had not implemented any process to check their implementation nor

their effectiveness. This is confirmed by observation that the practices in Practical Considerations and Software Quality Measurement sub-topics were almost nonexistent in this organization.

This can be illustrated using Figure 4: in the organization of case study 1, each project must use the corporate

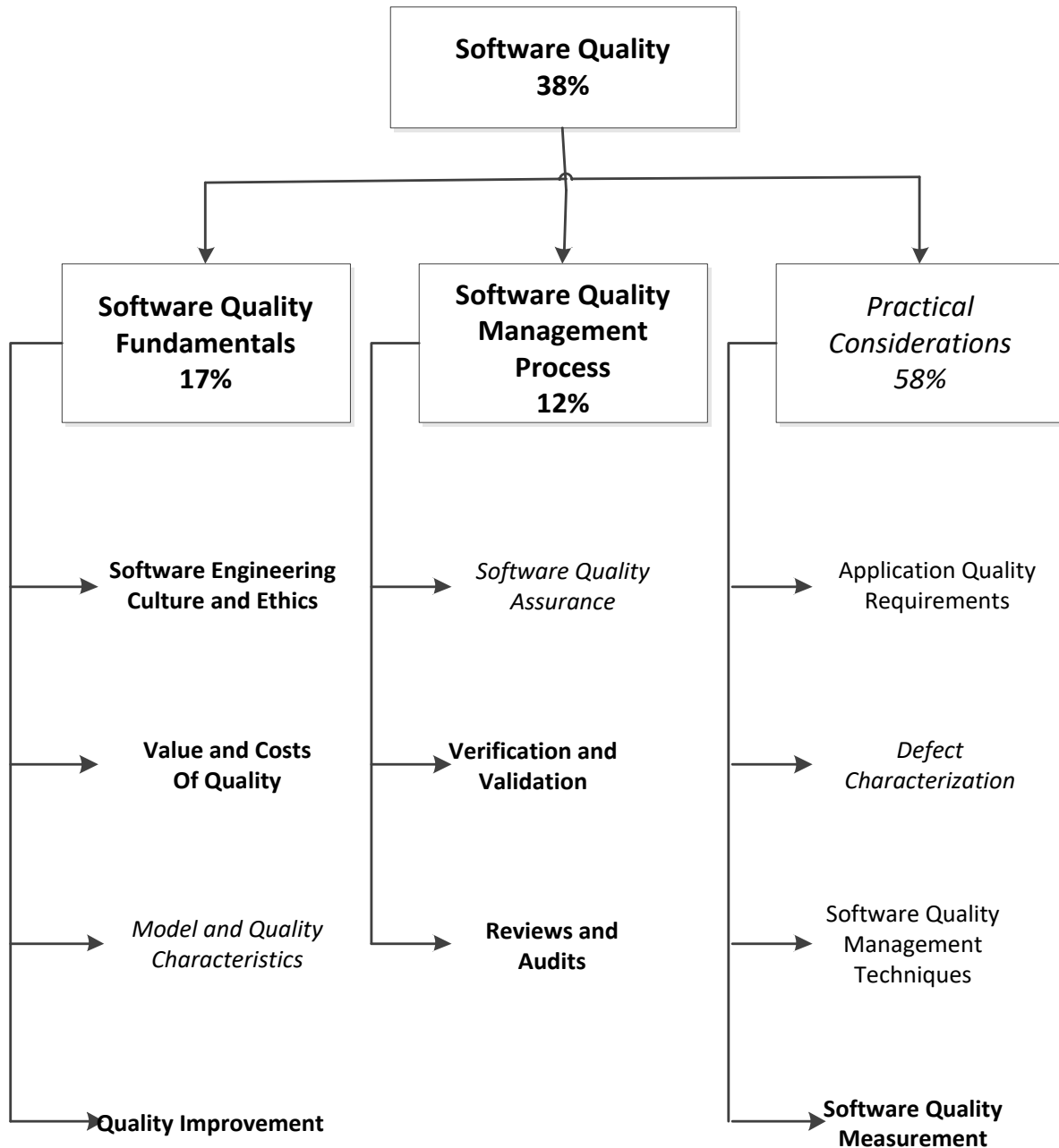


Figure 3. Software quality best practices.



Figure 4. Evaluation and Control.

development process (i.e. the production process in the lower box in Figure 4), but it typically does not collect any data for its evaluation and control. This means that the specific tasks are carried out by the technical staff in each phase of the development process (e.g. requirements, design, construction and testing), but there is no quantitative information provided to the management: this means that in this organization management does not request such information as a basis for decision making to analyze the performance of the processes implemented, to figure out the weaknesses and to figure out the necessary adjustments to the development processes based on quantitative information collected at each development stages.

This is illustrated in Figure 4 whereas the bottom part represents each of the development processes at each development phase with its inputs and outputs, and whereas the top portion represents the control portion (or management portion) which takes as input the quantitative information on the performance of the processes,

and as output, the management decision on what changes to make to the processes to optimize their performance.

In the SWEBOK Guide, for each knowledge area the necessary topics and best practices are listed to support collecting quantitative data and help making decision based on such quantitative data.

In this case study, it can be observed that this organization has a very good coverage of the SWEBOK practices at the process level (that is, they have technical expertise and technical processes related to this expertise) but an almost total lack of the management control functions which require an evaluation control function fed by measures from the processes and the feedback on the form of decision for making adjustment to the processes based on the quantitative data collected within processes as they are executed within each on-going development processes.

» Strengths and Weaknesses

The main strengths observed in the organization can be summarized as follows:

- The organization has strengths in documentation: it has generally very good documentation of requirements, better than the industry in general.
- Its software quality, based on the defect density 30 days after implementation, is better than that of the industry (based on information available from both industry and the organization).
- Comparison of the organization's practices with the SWEBOK practices guide (ISO TR-19760) showed, in general, good presence of practices in most areas.

The analyses also identified three areas of improvement opportunities:

- Early identification of project risks with a potentially twofold negative impact on project effort.
- Increase in project management efficiency:
 - Improvement to the productivity analysis and building of current productivity models;
 - Improvement to the estimation process;
 - Improvement to the management of change requests.
- Process improvement:
 - Establishment of mechanisms for monitoring and evaluating processes;
 - Reduction in fixed effort (establishment of predefined selection rules in the project context);
 - Monitoring of the impact of new technologies and new development processes.

This all suggest the need of better monitoring and control.

4.1.2 ISBSG Benchmarking

Benchmarking is the process by which measurement results of a specific entity are compared with the measurement results of similar entities. Traditional benchmarking models in software engineering are typically based first on the concept of productivity, defined as a single ratio of outputs to inputs, and on combining the productivity ratio with various additional factors.⁷ Benchmarking can be performed either internally with data collected within the organization, or externally with data collected outside the organization or available from multi-organizational datasets.^{5,8-11}

For external benchmarking, the following criteria were used to select a benchmarking repository:

1. A repository of projects representative of software applications providing financial services, from both the private and public sectors.
2. A repository of projects from various countries.
3. A repository with information available at the data field level (not at the aggregate level).

For this study, the repository of the International Software Benchmarking Standards Group – a nonprofit organization was selected as it met all these criteria: 3,854 relevant projects were available in the ISBSG at the time of the study.⁵ The selection of projects relevant to the context of this study was based on attributes similar to the environment of the organizations:

1. Projects with a similar size range (0 to 700 FP).
2. Projects developed in a development environment using 3rd Generation Programming Languages (3GL).

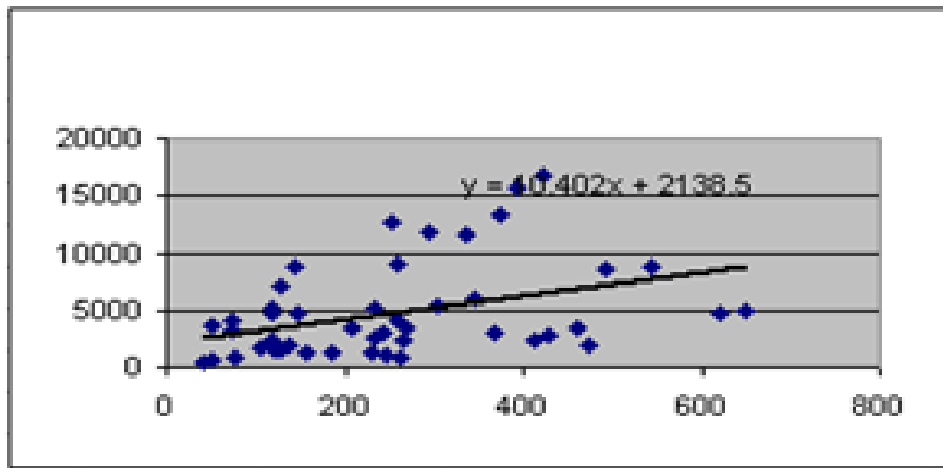


Figure 5. 3GL ISBSG governmental projects.

Projects that met these criteria were further classified into two groups:

A. Projects from governmental organizations:

In this group, 48 projects were found to have similar attributes as mentioned above. These ISBSG projects are presented graphically in Figure 5 (project size vs. effort), along with the corresponding model that best represents this dataset using a linear regression model:

$$\text{Effort} = 10.4 \text{ hours/FP} \times \text{project size} + 2138 \text{ hours}$$

B. Financial institutions (insurance and banks) in the private sector:

In this group, 119 projects were found to have similar attributes as mentioned above. The set of projects selected is represented in Figure 6, along with the following regression model:

$$\text{Effort} = 16.4 \text{ hours/FP} \times \text{project size} + 360 \text{ hours}$$

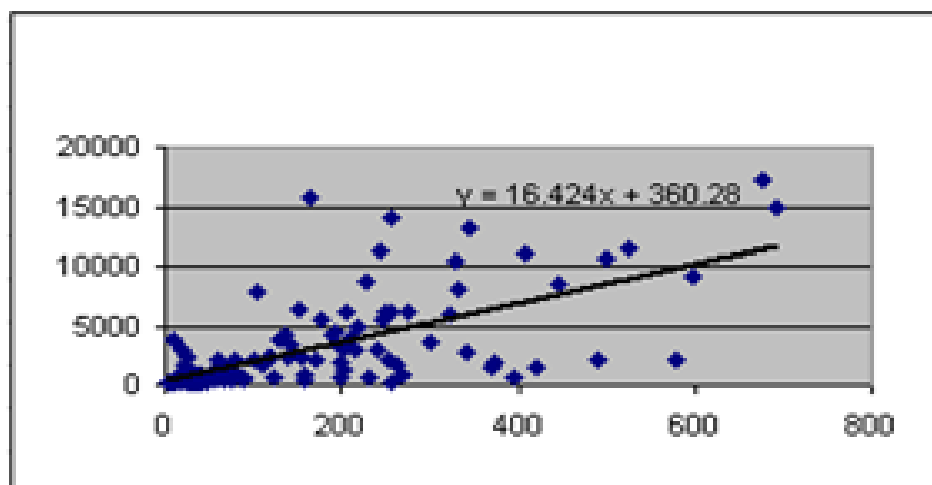


Figure 6. 3GL ISBSG projects from financial institutions.

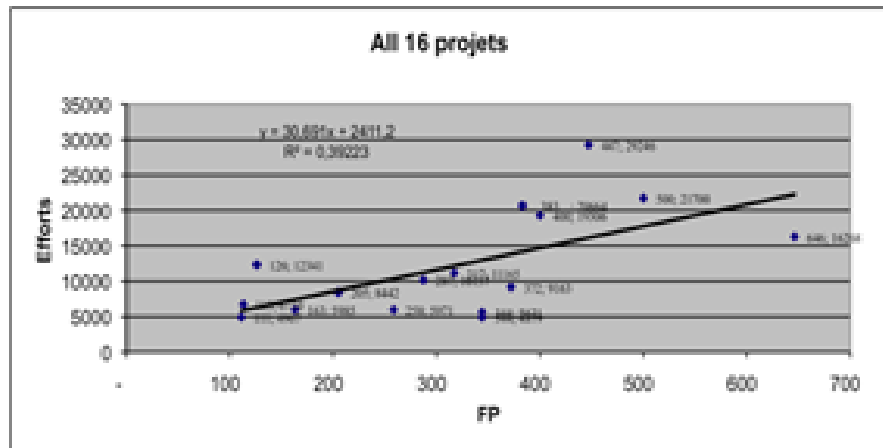


Figure 7. The organization's production model – Case Study 1.

The dispersion of the data points for all 16 projects in the organization is illustrated in Figure 7, together with the regression model:

$$\text{Effort} = 30.7 \text{ hours/FP} \times \text{project size} + 2411 \text{ hours.}$$

The coefficient of determination (R^2) of this model is relatively low, at 0.39.

The practical interpretation of the above equation is as follows for this organization:

- Fixed effort = 2411 hours
- Variable effort = 30.7 hours/FP

The possible reasons for the rather high fixed and variable unit effort figures have been discussed with the managers, and the following points were raised:

- The acquisition process is highly procedural and time-consuming.
- The projects have tight constraints and very extensive documentation requirements.
- The projects require lengthy consensus building procedures.

Table 4. Summary of fixed and variable effort

Public Organization	Org. (1)	ISBSG: Government (2)	ISBSG: Financial (3)
Number of projects	16	48	119
Fixed effort (hours)	2411	2138	360
Variable effort (hours/FP)	30.7	10.4	16.4

- The projects required a relatively high number of inspections.

Table 4 presents a summary of the fixed and variable costs of the models for both the organization (column 1) and the two reference groups from the ISBSG repository (columns 2 and 3):

- 1 The organization fixed effort of 2411 hours is:
 - comparable to the fixed effort of governmental institutions (2138 hours), and
 - 7 times higher than that of private financial sector organizations, at 360 hours.
- 2 The variable effort of 30.7 hours/FP is:
 - Almost triple that of the reference group of ISBSG governmental institutions, at 10.4 hours/FP, and
 - Almost double that of ISBSG private sector financial institutions, at 16.4 hours/FP.
3. To come up with more accurate productivity models, the 16 projects have been divided into two groups:
 - the most productive (n 11 projects) (below the regression line), and

- least productive (n = 5 projects) (above the regression line).

Next, a productivity model is developed for each group (see Table 5). Using the 11 most productive projects from the governmental site, the variable effort is very similar to the ISBSG financial projects at 17.1 hours. This is fairer for this governmental institution, because ISBSG claims that the projects received are the most productive. On the other hand, the variable effort of the 11 most productive projects in the organization is still higher than that of the ISBSG governmental projects by around 0.64%. The fixed effort of the 11 most productive projects is much higher than that for ISBSG groups both the two sets of governmental and from governmental financial.

4.1.3 Analysis summary

The higher variable cost in this organization can be explained by the different factors identified previously, including a high number of peer reviews, software process, documentation and requirements acquisition. Although these procedures are within SWEBOK knowledge areas that are untailored, excessive usage can be considered as a factor that hinders productivity: for instance, should the 11-step inspection process be carried through formally inspect half page of requirements specification in comparison to the ones requiring the inspection of a set of 30 pages of requirements?

Table 5. Governmental projects least and most productive – Case Study 1

Samples/ Regression coefficients	All 16 projects	Least productive: 5 projects	Most productive: 11 projects
Fixed effort (hours)	2411	8257	3208
Variable effort (hours/FP)	30.7	34.4	17.1

4.2 Case Study 2

4.2.1 SWEBOK Analysis

Due to confidentiality constraints, only the summary data for case study 2 can be presented - see table 3:

1. Software engineering management knowledge area has the highest score (57 %)
2. Software testing and software quality have the lowest score (0%).
3. Software Testing KA: the organization has zero percent because they do not have any systematic test technique and the existing test techniques are not based on of the best practices proposed by SWEBOK.
4. Not using any standard or recommended practices is the cause of low level percentage for Software Engineering Quality KA. The adopted assessment questionnaire, as part of the benchmarking, has confirmed this weakness showing a low score on user satisfaction in test results.
5. Software Requirements and Software Design are at 32% and 24% respectively. This can be illustrated by the quality of the documentation that is sig-

nificantly lower than the first organization. It was observed also by the usage of different requirements methods, design and construction. Also, there is no change request procedure in the organization. For a small organization with only two major applications the usage of one requirement method should be considered. For the design phase, the organization had already started to develop a unique methodology but it needs more work to help improve the quality of the design. Similarly, for software construction, there is only a partial written specific procedure to address the software construction.

6. There was no evidence of Software Configuration Management (SCM) activity in the organization. The purpose of SCM is to 'systematically controlling changes to the configuration, and maintaining the integrity and traceability of the configuration throughout the system life.'² Knowing interactions between different parts of the software is important when users ask improvement to the applications (new enhancement). The user satisfaction questionnaire shows that most clients were not satisfied with the documentation provided with new enhancements and with the test scenarios for the new enhancements.

Table 6. Number of Hours per Category for Three Years

Categories/ Year	1st year	%	2nd year	%	3rd year	%
Adaptive	564	14.3%	3476	30.4%	6145	37.1%
Corrective	2458	62.5%	6268	54.8%	7831	47.3%
User Support	909	23.1%	1686	14.8%	2585	15.6%
Total	3931		11430		16561	

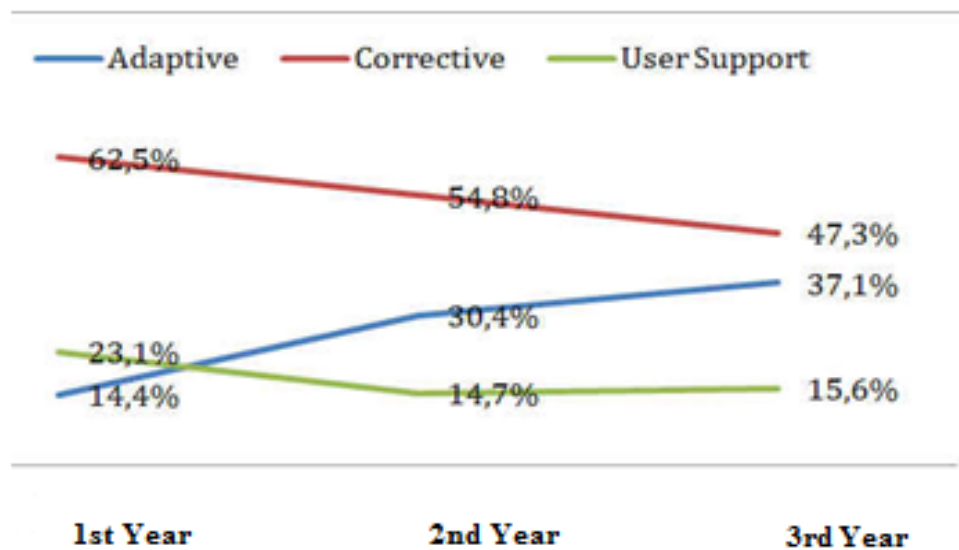


Figure 8. Relative effort for different types of maintenance.

- Maintenance knowledge area is at 50%. The effort to improve the maintenance processes in the organization has led to some improvements. Table 6 summarizes the number of hours spent in adaptive, corrective and user support for three consecutive years.

An analysis of the above data is showing that the relative cost of corrective maintenance has gone down from 62% in the first year to 54% in the second year and finally to 47% in the third year. However, the third-year corrective maintenance (47%) is still higher than adaptive (new enhancements) maintenance (37%) by 10% (Figure 8). Using the total efforts only, the benchmarking showed increase of costs for maintenance. From first to second year the cost of maintenance has increased by a factor of three (3) and from second year to the third year by a factor of (1.4) and by four (4) compared to the first year. This can be explained partly by the increase of the number of

clients and the increase of the functionalities provided to the clients.

Although for this organization the levels of practices in most of the knowledge areas have been rated as low, the maintenance indices are improving!

4.2.2 ISBSG Benchmarking

A productivity benchmark of this organization's projects (MIS development projects) with similar projects (MIS, similar sectors, size between 100 and 800 function points) in the ISBSG repository has been conducted. Table 6 presents a summary of the fixed and variable costs of the models for both the organization and the similar projects from the ISBSG repository. It shows that organization 2 has a slightly better productivity than that in the industry.

Figure 9 show the productivity model for organization 2. The constant cost is more than 250 hours and the variable cost are 10.5 hours/FP. The variable effort of 10.5 is close to the variable efforts of similar type of organization (small organization developing MIS projects) at 11.9.

Table 7. Summary of fixed and variable effort

Private organization	Org.	ISBSG: Small Enterprise MIS
Number of projects	5	239
Fixed effort (hours)	252	278
Variable effort (hours/FP)	10.5	11.9

4.2.3 Analysis Summary

The questionnaire showed that the employees in the organization are relatively young, fresh from universities and used new design and programming techniques on mainly two major applications. The higher productivity of organization 2 can be due to either the new techniques adopted by young employees or the better procedures and practices implemented because of the process improvement. This issue needs further investigation to clarify the influencing factor.

5. Discussion

The SWEBOK-based assessment and benchmarking covered the development part for case study 1, while case study 2 covered the development part as well as the Maintenance KA. Various differences have been noticed as follows:

- In case study 1, this organization has systematically implemented more practices in development

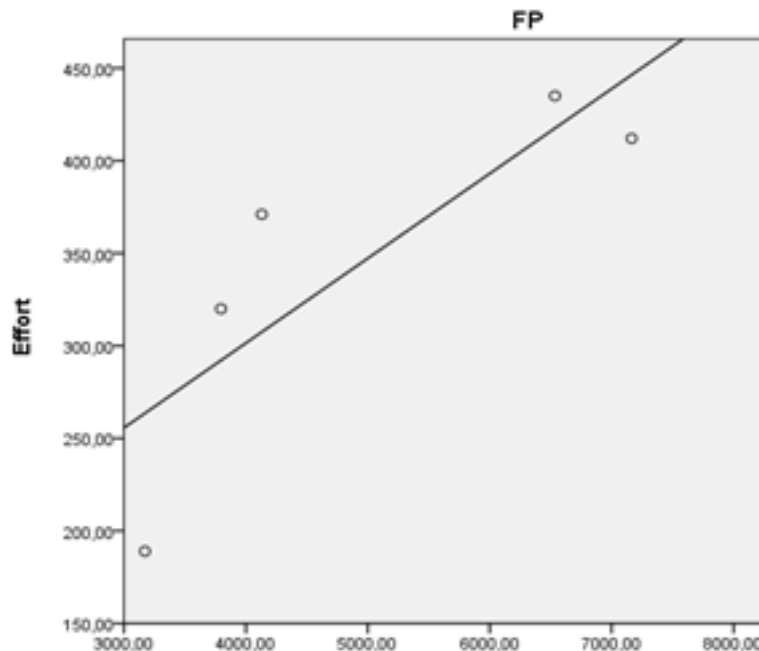


Figure 9. The organization 2 productivity model.

than in the organization in case study 2 as depicted in Table 2 and 3, mainly when looking the last column 'Coverage of practices';

- The productivity of the organization in case study 1 is low compared to the industry in general and for two third of the projects in average for this type of organizations, while the other third is higher (see tables 5 and 7). One of the different reasons could be the lack of control on the costs.
- The productivity of the organization in case study 2 is slightly higher to this type of industry but not significantly enough to conclude considering only 5 projects were measured in the organization.
- Large public IT organizations (500+) have more procedures than small private small organizations (less than 100 employees). This explain partly why the productivity of the organization in case study 1 is lower than that in case study 2, with considerably more practices;
- The private organization has no knowledge area with more than 59% of best practices;
- Software quality for the organization in case study 1 (identified by the implementation of SWEBOK practices in the Quality KA) is low (38%), but for that in case study 2 no related practices were implemented even with several minimal informal activities;
- From the results of the SWEBOK questionnaire, only 20% of the practices in case study 2 had been implemented, but over the previous few months prior to the usage of the questionnaire:–
- A score of 50% and higher could be interpreted as good in a small organization because not all

of these practices might be relevant and formally implemented for small organization's assessments.

6. Conclusion and Future Works

This research shows that it is possible to do useful appraisal using SWEBOK in one or two days. The results of SWEBOK assessment give enough information to find which goals the organization should seek to improve their processes. While benchmarking with ISBSG gives estimation of the projects effort, it does not justify that cost if it is higher than that in the industry, or what improvements are needed to improve the productivity. This part can be covered by SWEBOK assessment. Although it was impossible to link conclusively the SWEBOK questionnaire with the benchmarking data, we can conclude that the SWEBOK assessment can properly complement the benchmarking results to draw a complete picture about cost estimation and necessary process improvements of an organization. To understand more the relationship between benchmarking and SWEBOK assessment, more organizations' assessments and analyses using the SWEBOK questionnaire are needed.

7. References

1. Technical Committee: ISO/IEC JTC 1/SC 7 Software and systems engineering. ISO/IEC TR 19759:2015 - Software Engineering -- Guide to the software engineering body of knowledge (SWEBOK); 2015.
2. Bourque P, Richard EF. Guide to the software engineering body of knowledge. 3rd Edition. IEEE Computer Society Press; 2014.
3. Abran A, Moore JW. Guide to the software engineering body of knowledge (SWEBOK): 2004 version. IEEE Computer Society Press; 2004.
4. Grady RB. Successful software process improvement. 1st Edition. Prentice Hall PTR; 1997.
5. ISBSG: Software Project Benchmarking. [Online]. Available from: <http://isbsg.org/>. [Accessed: 09-Nov-2017].
6. Abran A, Desharnais JM, Zarour M, Demirörs O. Productivity - Based Software Estimation Models and

- Process Improvement : an Empirical Study. *Int Journal of Adv Softwar.* 2015; 8(1&2):103–14.
7. Buglione L, Abran A. Performance Calculation and Benchmarking using the ISBSG Release 10 Data Repository. In *Proceedings 5th Software Measurement European Forum*; 2008. p. 175–92.
 8. Cheikhi L, Abran A, Buglione L. ISBSG Software Project Repository & ISO 9126: An Opportunity for Quality Benchmarking. *UP GRADE.* 2006; 7(1): 46–52.
 9. Chiew V, Wang Y. Software Engineering Process Benchmarking. In *Product Focused Software Process Improvement (PROFES)*. Springer, Berlin, Heidelberg; 2002. p. 519–31. Crossref.
 10. Lokan C, Wright T, Hill P, Stringer M. Organizational benchmarking using the ISBSG Data Repository. *IEEE Softw.* 2001; 18(5): 26–32. Crossref.
 11. Maxwell KD, Forselius P. Benchmarking software development productivity. *IEEE Softw.* 2000; 17(1):80–8. Crossref.