Using PSU for Early Prediction of COSMIC Size of Functional and Non-functional Requirements

Luigi Buglione[1], Olga Ormandjieva[2], and Maya Daneva[3]

[1] École de Technologie Supérieure (ETS) / Engineering.it
luigi.Buglione@eng.it
[2] Concordia University, Computer Science & Software Dept.,
ormandj@cse.concordia.ca
[3] University of Twente, Information Systems Group
m.daneva@utwente.nl

Abstract. The project effort calculation with a functional size measurement method such as COSMIC can only be properly performed after the "Requirements Analysis" phase in a Project Life Cycle. The goal of this research is to investigate an early and project-level tuned prediction of the product size with the intent to reduce the effect of the 'cone of uncertainty' phenomenon. The lack of size measurement methods which take into account the effect of the product non-functional requirements (NFR) on size also contributes to the above phenomenon. We propose to use the Project Size Unit (PSU) technique for predicting the product (FUR and NFR) size measured in COSMIC functional size units. Such early prediction will lower the cost of size counting the project and minimize the estimation error in the requirements phase. Furthermore, the PSU calculation procedure can be automated, which would further reduce the cost of size counting. The expected advantage of jointly using PSU and COSMIC is the ability to get early estimates of the whole project effort.

Keywords: Project Size, Prediction, COSMIC, Project Size Unit (PSU), Functional User Requirements (FUR), Non-Functional Requirements (NFR).

1 Introduction

Increasingly, business demands require anticipated size estimates, in order to define the needed effort and the related cost (and expected revenues) for a project. However, when dealing with a FSM method, the product functional size can be simply estimated at early stages and not fully counted, as also discussed in the IFPUG CPM [19]. Therefore, the size calculation with a FSM method such as COSMIC can only be properly conducted at the end of the "Requirements Analysis" phase in a Project Life Cycle, having at your disposal an "advanced" *information detail* about the implementation for the software to be developed. Even if there are guidelines about the usage at early stages of FSM methods, the more refined the FUR, the higher the number of functional size units (fsu) obtained. Again, at early stages several non-functional requirements (NFR) must be accomplished (i.e. architectural and setup

R. Dumke et al. (Eds.): IWSM / MetriKon / Mensura 2008, LNCS 5338, pp. 352–361, 2008.

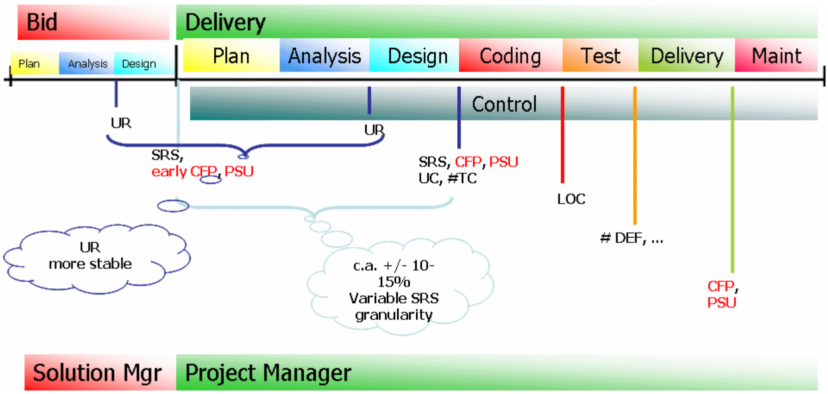

Fig. 1. Sizing measures and possible gathering moments during the SLC

tasks) and would not be considered in the product functional sizing. Since the goal for an estimator is to take care of the whole project boundary, as in Scope Management approaches, a complementary view on the way a project can be sized must be evaluated and introduced (see PMI's PMBOK [22] in the project domain and SouthernScope [23] and NorthernScope [24] in the software engineering domain). Figure 1 summarizes the moments and measures for typically sizing a project during the whole SLC, from the Bid phase on.

The current effort estimation techniques use the *product* functional size of software and not the size of a software *project* as an independent variable [9]. In the early estimation of the overall project effort, however, taking product FURs into account only translated into a (product) functional size, which definitely contributes to a larger MRE (Magnitude of Relative Error) in the early phases; that is, to the '*cone of uncertainty'* phenomenon [21] where the earlier the estimation, the larger the MRE as compared to the final results. The lack of size measurement methods which consider the effects of the product non-functional requirements (NFR) on size also contributes to the above phenomenon; in non-MIS projects. NFRs present a percentage of non-functional effort that can represent up to 50% of the overall project effort.

The aim of this research is to allow for an early and project-tuned prediction of the product size with the intent to reduce the effect of the 'cone of uncertainty'. Therefore, the research question is "How to predict product size for both FUR and NFR from project size before the analysis phase?"

We propose to use the PSU technique for predicting the product (FUR and NFR) size measured in COSMIC [2] [3] functional size units. Such early prediction will lower the cost of size counting the project and minimize the estimation error in the requirements phase. Furthermore, the PSU calculation procedure can be automated, which would further reduce the cost of size counting. Since calculating PSU takes less time, it can easily be used by small and medium enterprises (SMEs) who may not have the time or resources for learning and applying a FSM such as COSMIC. The expected advantage of jointly using PSU and COSMIC is the ability to get early estimates of the whole project effort with predicted CFP in the feasibility study phase. We also expect this would result in a better effort estimation approximation.

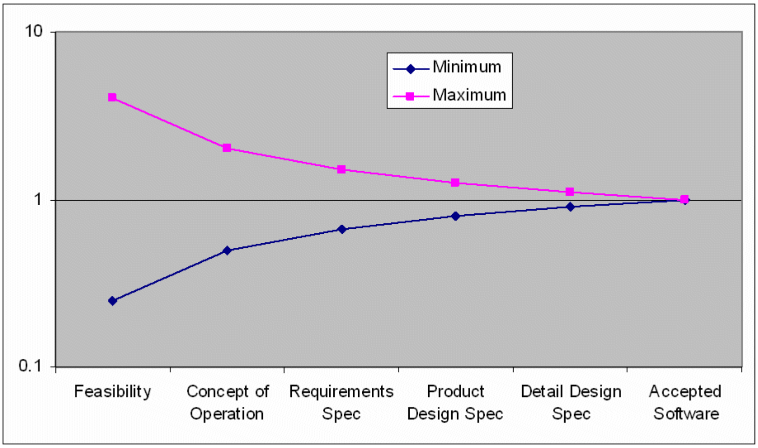

Fig. 2. The cone of uncertainty [21]

In what follows, we first provide background on PSU and COSMIC (see sections 2 and 3). The prediction formula and justification are presented in section 4. The approach is illustrated on student projects' data in section 5. The conclusions and the future research are outlined in section 6.

2 Overview of PSU (Project Size Unit)

PSU was first launched in 2003 as part of a Sw-CMM [16] level 3 certification process in an 80+staff-member organizational unit of a large multinational ICT company. One of the first challenges solved by means of the PSU was to accomplish those requirements from the Software Project Planning key process area which request the estimate efforts and costs (PP, Ac10), taking the overall project scope into account (PP, Ac2)[1]. Since the size calculated with a FSM method should directly refer solely to functional effort (and not to the overall project effort that also includes implicit and explicit project-level requirements, as well as product-level FUR and NFRs), a different process to size a project was put in action. The key idea was to move from the project boundary of the planned/executed activities to a scope extension that includes both FURs (Functional User Requirements) and NFRs.

From a Project Management viewpoint it means considering the whole sum of activities included in a WBS, trying to estimate the total amount of effort from requirements in an early stage. In fact, referring to ISO 9000 [15], the "quality" definition includes both explicit and implicit requirements, where activities and ensuing effort are generated by both and are therefore estimated and planned within the project boundary.

As Figure 3 shows, the goal of the PSU design was to define a new measure at the *project* level for approximating overall "project size" in the early stages.

[1] This also occurs with the newer CMMI-DEV v1.2 [14] model, where the old SPP key process area was simply renamed Project Planning (PP).

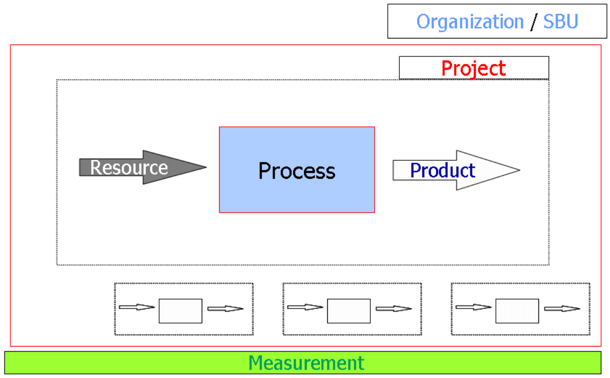

Fig. 3. STAR Taxonomy: measurable entities [13]

Fig. 4. Container (*project*) and content (*product*) [9]

"Project Size" is a term not yet defined in the ISO/IEEE/PMI glossaries. A proposal, according to the above premise, is to define it as "*the size of a software project, derived by quantifying the (implicit/explicit) user requirements referable to the scope of the project itself*" [9]. This term (and our definition) was proposed for inclusion in the next revision of the ISBSG Glossary of Terms [15].

Another example from the real world is the one proposed in Figure 4. Looking at a glass filled with wine, the size of the content (in this case the amount of wine) is not the size of its container (the glass). Applying this image to the entities represented in Figure 3, how could the product's (variable) content size allow for estimation of the size of its container (project)?

Unlike a FSM method, PSU needs an experiential/analogous estimate to produce a more refined estimate, compared with the 'organizational memory' (Project Historical Database - PHD). The PSU-counting is based on the WBS project tasks by three types: management (M), quality (Q) and technical (T) tasks. The T-tasks refer to the primary processes, while the M/Q-tasks refer to the organizational and support processes.

Each task is characterized by its complexity, which is measured by the effort that task requires. The greater the effort required for a task, without any control/milestone in the middle, the more complex and consequently, the riskier it is, with higher probability to request a re-plan during the project lifetime. So, the tactic during the drafting for a WBS is to refine it at the right level trying to minimize high-complexity tasks as much as possible, balancing the distribution of the forecasted effort against the several possible views (by SLC phase; by effort type; by task type, etc.). The PSU formula can be summarized as follows:

$$PSU = \sum_{i=M,Q,T} \sum_{j=H,M,L} task_i * weight_j \tag{1}$$

where the weights' ranges can vary according to the organizational style and definition for creating projects' WBS and can be easily derived by regularly applying the Pareto Analysis on the Project Historical Database (PHD). For detailed procedures, we refer interested readers to the PSU Measurement Manual [3]. By taking care of (at least) two main groups of requirements (FURs and NFRs), it is also possible to derive the final number of PSU as the sum of the PSU_f (calculated from the tasks derived by FURs) and PSU_{nf} (calculated from the tasks derived by NFRs).

A recent case study using 33 projects that were also sized with IFPUG FPA v4.2 and COSMIC-FFP v2.2 [25] showed a good PSU prediction capability using a standard weighting system. The periodical update of the weighting system results in obtaining a better fit for newer estimates, moving away from the way estimators within the organization previously obtained results and further reducing episodes of the 'cone of uncertainty' as described above.

Again, since the input for calculating PSU are the tasks composing the project WBS, it is possible, as opposed to the FSM method, to easily automate its calculation under any project management software tool [18][2], even on the intensive human-based activity of elicitation and refinement of FUR. Plenty of project data and attributes stored within the software project management tool can be managed with an export utility in XML/CVS format in order to facilitate the creation and maintenance of the organizational PHD, moving progressively from experience/analogy-based estimates towards regression analysis-based ones.

3 Overview of COSMIC

The COSMIC measurement method conforms to all ISO requirements (ISO 14143-1 [20]) for functional size measurement. COSMIC focuses on the "user view" of functional requirements, and is applicable throughout the development life cycle, from the requirements analysis phase right through to the implementation and maintenance phases.

The process of measuring software functional size using the COSMIC method implies that the software functional processes and their triggering events be identified. These are available in the analysis phase. In COSMIC, the unit of measurement is the data movement, which is a base functional component that moves one or more data attributes belonging to a single data group. It is denoted by the symbol *CFP* (Cosmic Function Point). Data movements can be of four types: Entry, Exit, Read or Write. The functional process is an elementary component of a set of user requirements triggered by one or more triggering events, either directly or indirectly, via an actor.

[2] A first implementation under an Open Source Software (OSS) was done with GanttProject (www.ganttproject.org) v2.0.3 [26].

In [5] COSMIC is used for sizing NFR stated in verifiable terms. This means that NFR are stated in terms of crisp indicators with defined acceptable values; thus, it is possible to verify the satisfaction level of those NFR by comparing the acceptable values with the actual achieved values.

4 Predicting CFP with PSU

Prediction determines the likely future values of product measures based on existing measures of the same product. For the purpose of early size prediction we need to define a relationship between product size CFP and project size PSU by requirements type, that is, FUR and NFR. Such relations will allow for: (1) reducing the size measurement effort at this early stage; (2) allowing for accurate size prediction of all NFR, including those which are not (yet) stated in measurable terms.

As stated in section 2, PSU respects the additive property, thus the following equation is valid theoretically from the representational theory of measurement point of view: $PSU=PSU_f + PSU_{nf}$ hence the scale type of the PSU is at the least interval.

On the other side, the addition of the CFP size values ($CFP=CFP_f+ CFP_{nf}$) is also theoretically valid because COSMIC size has a unique unit of measurement, the CFP. Thus the COSMIC size measure is at least on the ratio scale. Consequently, the admissible transformation between the size units CFP and PSU is of type **M'=k*M+b (k>0)** [10]**,** which justify the following relations:

$$PSU_f = DataMovementPSU_{fSize} * CFP\_f + b1 \tag{2}$$

$$PSU_{nf} = DataMovementPSU_{nfSize} * CFP\_nf + b2 \tag{3}$$

For further discussion on the scale types and the representational theory of measurement, see [10].

The CFP_f and CFP_{nf} can be predicted in the planning phase of the new project from the actual values of PSU_f and PSU_{nf} and the DataMovementPSUfSize and $DataMovementPSU_{nfSize}$ derived using regression analysis of the PSU and CFP data, where $DataMovementPSU_{fSize}$ and $DataMovementPSU_{nfSize}$ serve as adjustment factors related to the project size.

The $DataMovementPSU_{fSize}$, $DataMovementPSU_{nfSize}$, b1 and b2 can also be derived from the PHD data on CFP_f , CFP_{nf} , PSU_f and PSU_{nf} by using Monte Carlo simulation [11]. Monte Carlo simulation is a problem-solving technique used to approximate the probability of certain outcomes by running multiple trial runs, called simulations; using random variables. We chose it for this research, because JPL and THAAD [27] recommended its use as a solution to deal with uncertainty in software project estimation. For example, these researchers and the last author [12] have deployed it to approach the inherent uncertainty of cost factors. The key advantage of Monte Carlo simulation is that - by collecting samples of the output variables for each run, it helps the estimation analysts produce an overall picture of the combined effect of different input variables' distribution on the output of a model.

To deploy the Monte Carlo simulation in our solution proposal, we first have to ascribe a particular distribution type to the input variables in the model (that is, to CFP_f, CFP_{nf}, PSU_f and PSU_{nf}). When we run the model, the distribution attached to each input variable will be randomly sampled and the result entered into the model. Repeatedly running the model many times (for example 10 000 times) and collecting samples of the output variables for each run will produce an overall picture of the combined effect of different input variables' distribution on the output of the model. The results of the simulation are in the form of a histogram showing the likelihood of obtaining certain output values for the set of input variables and attached distribution definitions.

While our solution proposal is common sense and sounds intuitive, our results of its use are theoretical and require empirical validation. The proposed method for predicting the product (FUR and NFR) size measured in COSMIC [2] [3] functional size units from the PSU data is illustrated on five student web application development projects.

5 Illustration

The approach described in this paper is illustrated on project data collected on a one-term software project given to five teams formed by third-year undergraduate students in the software engineering program at Concordia University. Each team was given the same problem statement describing an online exam management system that can be used by instructors, students, coordinators, markers, and administrators. Among other services, this software allows i) instructors to manage the question pool, the grades, and conduct exams, ii) students to write real and practice exams, view marks, and register for an exam, iii) markers to grade specific sections of an exam, and iv) administrators to manage courses and user accounts.

In the initial planning activity step the students were asked to estimate the effort for each task entry in their WBS charts and later record the corresponding actual effort. The above data collected by the students served as an input to the PSU_f and PSU_{nf} calculation process, where each task was classified as M/Q/T and assigned the corresponding complexity based on the task's effort estimation and risk assessment and using 4-level complexity schema (H/MH/ML/L).

PHD data collection. The initial start-up hypothesis of the PSU calculation process assumed the weights of the tasks by experience, before evaluating the projects' WBS structure. The low correlation with PSU_f showed that such weighting assignment is unreliable for planning purposes. The weights were recalculated to incorporate the teams' WBS structures and the complexity distribution under the representational constraint weight(L)<weight(ML)<weight(MH)<weight(H). The weights adjustment resulted in a strong statistical relationship (R^2=0.69) between the PSU_f, PSU_{nf} and the tasks evaluation which proved the formula adequate for planning purposes. Additional analysis was carried out with a feasible subdivision into 5 effort ranges for classifying complexity, instead of the previously considered 4-level weighting system (H/MH/ML/L), but verification of the estimation validity of such a model was slightly lower than using 4 levels (with the same data). As a result, it was concluded that i) the 4-level weighting system is statistically more appropriate for planning purposes in this project, and ii) such a system and the PSU data could be used for comparison with

Table 1. Summary of the Student Project Historical Data

Groups	PSU_{total}	PSU_{nf}	PSU_f	CFP_{total}	CFP_{nf}	CFP_f
A	102	60	42	68	15	53
B	77	43	34	131	32	99
C	28	21	7	114	29	85
D	40	25	15	184	22	162
E	97	56	41	147	32	115

COSMIC data as explained in the rest of this section. The size of the FURs developed by each team was measured using the COSMIC method as CFP_f. The size CFP_{nf} of NFR where calculated as described in [6]. Table 1 presents the summary of the PSU and CFP calculations.

Regression analysis results. Formulas (4) and (5) calculated on the project historical data for the 5 projects listed in Table 1 describe the statistical dependencies between the pairs CFP_f, PSU_f and CFP_{nf} , PSU_{nf} :

$$PSU_f = -0.148 * CFP_f + 43 \tag{4}$$

$$PSU_{nf} = -0.47 * CFP_{nf} + 53.28 \tag{5}$$

6 Conclusions and Prospects

This paper aims to resolve one of the major issues, namely the challenge of predicting CFP of FUR and (more importantly) NFR moving from project scope knowledge captured in PSU estimates. A significant outcome is that the FUR and NFR functional size can be predicted from the PSU earlier in software planning, which will help managers in realistically scheduling project milestones. Moreover, the productivity analysis can be performed precisely from the predicted CFP size and the estimated effort. Other advantages of using PSU in this research are:

- *general-purpose*; PSU can be used on all kinds of projects (i.e. service, building, performing arts...).
- can be *automated* under various project management tools, also integrating other useful project information for an XML export easily creating the organizational PHD [18], since the measurable entities are tasks.

In future, we plan to investigate its applicability in real-life project settings. We are also aware of related validity concerns [10] and plan a series of case studies to test our approach, to properly evaluate its validity and to come up with an improved version of it.

References

[1] Albrecht, A.J.: Measuring Application Development Productivity. In: Proc. Joint SHARE/GUIDE/IBM Application Development Symposium, pp. 83–92 (1979)
[2] IEEE Std 830-1998: IEEE Recommended Practice for Software Requirements Specifications, Software Engineering Standards Committee of the IEEE Computer Society (1998)
[3] Abran, A., Desharnais, J.-M., Oligny, S., St-Pierre, D., Symons, C.: COSMIC FFP – Measurement Manual (COSMIC implementation guide to ISO/IEC 19761:2003). École de technologie supérieure – Université du Québec, Montréal (2003), http://www.cosmicon.com
[4] ISO/IEC 19761. Software Engineering – COSMIC-FFP – A Functional Size Measurement Method. International Organization for Standardization – ISO, Geneva (2003)
[5] Albrecht, A.J., Gaffney, J.E.: Software Function, Source Lines of Code, and Development Effort Prediction: A Software Science Validation. IEEE Trans. Software Eng. SE-9(6), 639–648 (1983), http://www.bfpug.com.br/Artigos/Albrecht/Albrecht_Gaffney.pdf
[6] Kassab, M., Ormandjieva, O., Daneva, M., Abran, A.: Non-Functional Requirements: Size Measurement and Testing with COSMIC-FFP. In: Cuadrado-Gallego, J.J., Braungarten, R., Dumke, R.R., Abran, A. (eds.) IWSM-Mensura 2007. LNCS, vol. 4895, pp. 168–182. Springer, Heidelberg (2008)
[7] Kassab, M., Ormandjieva, O., Daneva, M., Abran, A.: Towards a Scope Management of Non-Functional Requirements in Requirements Engineering. In: Proceedings of MeReP: Workshop on Measuring Requirements for Project and Product Success, Palma de Majorca, Spain, November 6, 2007, pp. 88–99. University of Heidelberg Press (2007), http://www-swe.informatik.uni-heidelberg.de/home/events/MeRePDocs/
[8] Buglione, L.: Project Size Unit (PSU) - Measurement Manual, version 1.21 (November 2007), http://www.geocities.com/lbu_measure/psu/psu-mm-121e.pdf
[9] Buglione, L.: Some Thoughts on Productivity in ICT projects, WP-2008-01, White Paper, version 1.2 (March 2008), http://www.geocities.com/lbu_measure/fpa/fsm-prod-120e.pdf
[10] Buglione, L.: Improving Estimation by Effort Type Proportions. Software Measurement News 13(1), 55–64 (2008), http://ivs.cs.uni-magdeburg.de/sw-eng/us/giak/SMN-08-1.htm
[11] Fenton, N., Pfleeger, S.L.: Software Metrics: A Rigorous and Practical Approach, 2nd edn. International Thompson Computer Press (1997) ISBN 0-534-95425-1
[12] Daneva, M.: Approaching the ERP Project Cost Estimation Problem: an Experiment. In: 1st International Symposium on Empirical Software Engineering and Measurement, p. 500. IEEE Press, New York (2007)
[13] Buglione, L., Abran, A.: ICEBERG: A Different Look at Software Project Management, IWSM 2002 in Software Measurement and Estimation. In: Proceedings of the 12th International Workshop on Software Measurement (IWSM 2002), Magdeburg, Germany, October 7-9, 2002, pp. 153–167. Shaker Verlag (2008) (2008-05-23), http://www.lrgl.uqam.ca/publications/pdf/757.pdf

[14] CMMI Product Team, CMMI for Development, Version 1.2, CMMI-DEV v1.2, CMU/SEI-2006-TR-008, Technical Report, Software Engineering Institute (August 2006) (2008-05-23), `http://www.sei.cmu.edu/publications/documents/06.reports/06tr008.html`

[15] IEEE, Software & Systems Engineering Vocabulary (SEVOCAB), IEEE Computer Society & ISO/IEC JTC1/SC7 (2008-05-23), `http://pascal.computer.org/sev_display/index.action`

[16] Paulk, M.C., Weber, C.V., Garcia, S.M., Chrissis, M.B., Bush, M.: Key Practices of the Capability Maturity Model Version 1.1, Software Engineering Institute, CMU/SEI-93-TR-025 (February 1993) (2008-05-23), `http://www.sei.cmu.edu/pub/documents/93.reports/pdf/tr25.93.pdf`

[17] ISBSG, Glossary of Terms, version 5.9.1, International Software Benchmarking Standards Group (28/06/2006) (2008-05-23), `http://www.isbsg.org/html/Glossary_of_Terms.doc`

[18] Buglione, L.: Project Size Unit (PSU) – Calculation feature in Project Management tools - Requirements, v1.0, PSU-AU-1.00e (December 2006) (2008-05-23), `http://www.geocities.com/lbu_measure/psu/psu.htm`

[19] IFPUG, Function Points Counting Practices Manual (release 4.2), International Function Point User Group (January 2004) (2008-05-23) , `http://www.ifpug.org`

[20] ISO/IEC, International Standard 14143-1 - Information Technology - Software Measurement - Functional Size Measurement - Part 1: Definition of Concepts (February 2007)

[21] Boehm, B.: Software Engineering Economics. Prentice-Hall Inc., Englewood Cliffs (1981)

[22] Project Management Institute, A Guide to the Project Management Body of Knowledge, 3rd edn. (2004), ANSI/PMI 99-001-2004, ISBN 1-930699-45-X

[23] Victoria Government, SouthernScope (2007) (2008-05-23), `http://www.egov.vic.gov.au/index.php?env=-innews/detail:m1816-1-1-8-s-0:n-832-1-0-`

[24] FISMA, NorthernScope (2007) (2008-05-23), `http://www.fisma.fi/in-english/scope-management`

[25] Buglione, L., Cuadrado-Gallego, J.J., Gutiérrez de Mesa, J.A.: Project Sizing and Estimating: A Case Study using PSU, IFPUG and COSMIC. In: Proceedings of IWSM/Metrikon/Mensura 2008, Munich, Germany, November 18-19, 2008. Springer, Heidelberg (2008)

[26] Biagiotti, C.: Migliorare gli aspetti di stima e pianificazione di un progetto attraverso la customizzazione di un tool OpenSource di Project Management, University of Perugia, Tesi di Laurea, Perugia, Italy (July 2007)

[27] McDonald, P., Giles, S., Strickland, D.: Extensions of Auto-Generated Code and NOSTROMO Methodologies. In: Proc. of 19th Int. Forum on COCOMO, Los Angeles, CA (2001)