Experimental Study Using Functional Size Measurement in Building Estimation Models for Software Project Size

Nelly Condori-Fernandez
Universitat Politècnica de Valencia
Valencia, Spain
nelly@pros.upv.es

Luigi Buglione
ETS/Engineering.IT.
Rome, Italy
luigi.buglione@eng.it

Maya Daneva
University of Twente
Enschede, The Netherlands
m.daneva@utwente.nl

Olga Ormanjieva
Concordia University
Montreal, Canada
ormanj@cse.concordia.ca

Abstract — This paper reports on an experiment that investigates the predictability of software project size from software product size. The predictability research problem is analyzed at the stage of early requirements by accounting the size of functional requirements as well as the size of non-functional requirements. The experiment was carried out with 55 graduate students in Computer Science from Concordia University in Canada. In the experiment, a functional size measure and a project size measure were used in building estimation models for sets of web application development projects. The results show that project size is predictable from product size. Further replications of the experiment are, however, planned to obtain more results to confirm or disconfirm our claim.

Keywords: functional size measurement; experiment; empirical software engineering; software project estimation

I. INTRODUCTION

Project effort estimation at the stage of early requirements has been recognized as one of the most challenging processes in software engineering (SE). This challenge may aggravate in the context of software product development, as the commitment of product managers (and, ultimately, the product market success) depends on the project managers' ability to provide accurate estimates. While the IT industry has developed numerous effort estimation methods, it is a well-known fact that a huge gap exists between the initial effort estimation and the actual effort which has been required in reality for software projects. Moreover, to the best of our knowledge, almost no research has been done in the area of exploring the relationship which might exist between software product attributes and software project attributes at the requirements stage. The lack of knowledge about this relationship motivated the research that we present in the present paper. We make a step towards understanding the possible links between software product and project attributes. We think that if we increase our knowledge of the project

variables on which the effort estimation depends, we will be able to eventually reduce the gap between estimated effort and actual effort.

We have identified the functional size, the main variable in effort estimation formulas, as one of the possible factors contributing the above gap. Functional size measurement (FSM) methods [1] have been proposed as solutions to this problem. However, using a FSM method in the early requirements stage of a project poses a number of risks. First, contemporary ISO-standardized FSM approaches, as [2,3,4] take as inputs the functional requirements only and translate them into a (product) functional size. This, more often than not, leads to a larger Magnitude of Relative Error (MRE) in the early phases; that is, to the 'cone of uncertainty' phenomenon [11] where the earlier the estimation, the larger the MRE as compared to the final results. Second, the exclusion of the non-functional requirements (NFR), such as reliability and security, from the sizing process also contributes to the 'cone of uncertainty' phenomenon; indeed, in non-MIS projects, implementing the NFRs on a project can represent up to 50% of the overall project effort [6]. Third, the current effort estimation techniques use the *product* functional size of software and not the size of a software *project* as an independent variable [7]. Because the goal of the project manager is to take care of the whole project boundary, as per prominent Scope Management approaches [8], it makes good sense to evaluate a complementary view on the way a project can be sized. In this paper, we make a first step developing a deeper understanding of the possible relationship which might exist between software project size and software product size. An important unique aspect of this research is that the notion of product size, which we use, includes both functional and non-functional requirements. Furthermore, we apply the guidelines proposed by Wohlin [9] to design an experimental study. As research methodologists in SE suggest [18], analyzing the results of experiments involves learning and encapsulation of

knowledge, which in turn increases the researchers' ability to change and refine their solution proposals over time. In this paper, we report on an experiment which we performed, aiming at developing a solution to the above three issues.

The rest of the paper is organized as follows. Section II provides background on the concepts used and formulates our research hypotheses. Section III presents the plan and the execution of the experiment. Section IV reports on validity threats. Section V concludes the paper.

II. BACKGROUND ON PROJECT SIZE AND PRODUCT SIZE

The research in this paper draws on previously published results by other authors who suggest that if we allow for an early and project-tuned prediction of the product size, this could reduce the effect of the 'cone of uncertainty'[20][21]. Below, we describe the concepts of project size and product size and discuss possible relationships between these concepts.

A. Project Size

Systematic reviews [Jor] on the topic of software estimation indicate that expert judgment is still the dominant estimation technique in practice today for software project size and effort. Estimation experts are often supported by means of checklists and group discussions. Recently, attempts were made to complement expert judgments with point-based techniques, for example, the Project Size Unit [15], to add up to a more objective project estimation process. In this paper, we use the PSU model as the approach to project sizing. From a project management viewpoint, it means considering the total sum of activities included in a work breakdown structure (WBS), trying to estimate the total amount of effort from the requirements in an early stage. As Figure 1 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.

We make the note that "Project Size" is a term not yet defined in the ISO/IEEE/PMI glossaries [8, 17, 19]. 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*" [7].

Figure 1. STAR Taxonomy: measurable entities

Unlike a FSM method, PSU needs an experiential/analogous estimate to produce a more refined estimate, compared with the 'organizational memory', that is the organization's 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 this 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. 'Riskier' means that there is a 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 and balancing the distribution of the forecasted effort against the several possible views (for example, by Software Life Cycle phase; by effort type; by task type, etc.). The PSU formula can be summarized as follows:

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

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 PHD. For detailed procedures, we refer interested readers to the PSU Measurement Manual [15]. By taking care of two main groups of requirements: functional user requirements (FURs) and NFRs, it is also possible to derive the final number of PSU as the sum of the PSU_{FUR} (calculated from the tasks derived by FURs) and PSU_{NFR} (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 v2.2 [16] showed a good PSU prediction capability using a standard weighting system. The periodical update of the weights (see formula (1)) 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 automate its calculation within any project management software tool that also helps elicitation and refinement of FUR [14]. For example, a first implementation under an Open Source Software (OSS) was done with GanttProject (www.ganttproject.org) v2.0.3 [13]. We make the note that plenty of project data and attributes stored within the software project management tool can be managed with an export utility in XML/CSV 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.

B. Product Size

In this research, we exploit sizing procedures which include both FURs and NFRs. Specifically, the sizing methods we use in this experiment are COSMIC [2] and the newly proposed COSMIC-extension, the so-called COSMIC-NFSM [12], which includes the NFRs on a project.

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. The COSMIC is applicable to NFRs decomposable into *operationalizations* which provide both design solutions for achieving the NFR goals and the basis for defining their indicators [12].

C. Predicting Project Size using CFP

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.

The hypotheses formulated from our research question are the following:

Hypothesis 1. In relation to functional requirements:

Null hypothesis, H_{10} . Project Functional Size cannot be estimated from Product Functional Size.

Alternative hypothesis, H_{11} . Project Functional Size can be estimated from Product Functional Size.

Hypothesis 2: In relation to non-functional requirements:

Null hypothesis, H_{20} . Project Non-Functional Size cannot be estimated from Product Non-Functional Size.

Alternative hypothesis, H_{21} . Project Non-Functional Size can be estimated from Product Non-Functional Size.

We think, that knowledge on these relationships will allow for: (1) reducing the size measurement effort at this early stage; and (2) improving the accuracy of size prediction of all NFR, including those which are not (yet) stated in measurable terms. As stated in section 2.1, PSU respects the additive property, thus the following equation is valid theoretically from the representational-theory-of-measurement point of view: $PSU = PSU_{FUR} + PSU_{NFR}$ hence the scale type of the PSU is at least interval (see [6] for more details on scale types and their characteristics). On the other side, the addition of the CFP size values ($CFP = CFP_{FUR} + CFP_{NFR}$) 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 $PSU = k * CFP + b$ ($k > 0$), which justify the following relations:

$$PSU_{FUR} = weight_{FUR_Size} * CFP_{FUR_Size} + b_{FUR_Size} \quad (2)$$

$$PSU_{NFR} = weight_{NFR_Size} * CFP_{NFR_Size} + b_{NFR_Size} \quad (3)$$

Theoretically, equations (2) and (3) mean that if we can run a regression analysis process, then we should be able to identify the relationships between the CFP_{FUR} , PSU_{FUR} and $weight_{FUR_Size}$ as well as between CFP_{NFR} , PSU_{NFR} and $weight_{NFR_Size}$.

III. THE EXPERIMENTAL STUDY

The goal of our study, according to the goal-driven approach [10][22], is to **analyze** the software size measurement

method based on the ISO 19761 standard *with the purpose of evaluating its ability to predict software project size; from the viewpoint of the researcher in the context of* SE undergraduate students in their third year of studies enrolled in the 2008 and 2009 “Software Measurement” and “Software Project” undergraduate SOEN courses at Concordia University, Montreal, Canada. The experiment addresses the following two research questions (RQ):

RQ1: In what extent does the product non-functional size determine the project size?

RQ2: In what extent does the product functional size determine the project size?

A. Selection of subjects

The selected subjects were 55 third-year Computer Science students enrolled in both 2008 and 2009 SOEN337 “Software Measurement” and SOEN390 “Software Project” undergraduate courses at Concordia University, Montreal, Canada). Prior to these courses, students have taken several courses on SE subjects. The experiment was organised as mandatory part of the SOEN337 “Software Measurement” course. The subjects received lectures on the COSMIC method, including its application to NFRs (according to the measurement procedure presented in [12]). The duration of the students training prior to the experiment was 8 hours. The teaching process of the both courses included group work by students. This meant that student groups were responsible for carrying out a group project and executing life cycle activities ranging from project planning, specification, analysis and design. The students were organized in eleven groups, each of which included 5 students.

B. Identification of variables

As research methodologists recommend [5,9], planning an experiment include the identification of (i) response variables (outcomes of the experiment as per Wohlin [9]), (ii) factors which might impact the response variables, and (iii) parameters, which we do not want to influence the experiment’s results. Project size and product size were identified as response variables. As these outcomes should be measurable, project size and product size were expressed in PSU and CFP units respectively.

Furthermore, two variables were identified as factors that could affect the response variables. These are: first, the chosen project size measurement method, namely PSU; and second, the product size measurement method. With respect to the latter, two treatments were considered: the COSMIC method [2] and the COSMIC-NFSM method [12]. Last, three variables were identified as parameters: (i) application domain: our study is carried out in the web-application domain only, (ii) experience using size measurement methods, and (iii) quality of requirements specification.

C. Instrumentation

The instruments used in this experiment included the training materials and the experimental object. The *training materials* were the following: a set of instructional slides on COSMIC [2], COSMIC-NFSM [12], COSMIC case studies,

and the COSMIC manual [2]. The *experimental object* was the Use-Case Model document that described in natural language the structure of an online exam management system. Each group worked on a subsystem of this system. The exam management application is meant to be used by instructors, students, coordinators, exam-markers, and administrators. Among other services, this software application 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.

D. Experimental procedure

The experiment was initiated by training the subjects in both the COSMIC and the COSMIC-NFSM methods. The time used for training was 8 hours (4 lecture hours and 4 hours of tutorials) distributed over 4 days.

After the training process, each group was given the same problem statement describing the respective subsystem of the online exam management system.

The experimental procedure is presented in Fig 2. It says that eleven student groups used both the COSMIC CFP and COSMIC NFSM to size the functional and non-functional requirements in their projects.

Figure 2. Illustration of the experimental procedure

As is shown in Fig 2, a WBS project and software requirements documentation were created by each group. In the initial planning activity step the students were asked to estimate the effort for each task entry in their WBS charts and, later, to record the corresponding actual effort. The documentation was written iteratively in the Software Project course (SOEN390) as part of the 6-milestones' deliverables of the students.

Each group of students also used both the COSMIC-CFP and COSMIC-NFSM methods to size the FR and the NFRs in their respective projects (See Table I.). The measurement was performed as an assignment in the Software Measurement class (SOEN337). We make the note that although two weeks were

allocated for this assignment, the students mostly worked the last night before the due date.

TABLE I. SUMMARY OF THE STUDENT PROJECT DATA

Groups 2008	PSU_{NFR}	CFP_{NFR}	PSU_{FUR}	CFP_{FUR}	Total PSU	Total CFP
A	60	5	53	75	113	80
B	43	3	99	220	142	223
C	29	5	85	204	114	209
D	25	9	162	168	187	177
E	56	5	115	152	171	157
Groups 2009						
F	250	27	133	133	383	160
G	63	3	103	115	166	118
H	55	26	69	109	124	135
I	70	11	70	103	140	114
J	44	3	43	139	87	142
K	108	8	127	147	235	155

The software requirements and the WBSs have been provided to an expert, namely the third author of this paper, who calculated the PSU_{FUR} and PSU_{NFR} values for each group (See Table I.). We must note that the expert and the students worked independently, without any communication between each other. The expert did not know the students. The students were unaware of the work of the expert.

IV. DATA SAMPLE

To answer our research questions, hypotheses H1 and H2 were formally tested by using the linear regression analysis. The data used for the regression analysis is presented in Table I. With respect to the project size estimation, based on non-functional size our results are shown in Table II and Table III. The data is plotted in Figure 3.

The coefficient of determination (R^2) indicates that the 41% of the variation in the project non functional size (PSU_{NFR}) can be explained by the variation in the non functional size of the product (CFP_{NFR}).

TABLE II. REGRESSION ANALYSIS MODEL: CFP_{NFR} AND PSU_{NFR}

R	R^2	Std. Error of the Estimate	Significance
0.64	0.415	50.64	0.033

TABLE III. REGRESSION ANALYSIS: COEFFICIENTS.

	Unstandard. Coefficients		Standard Coeff. Beta	t	Sig.
	B	Std. Error			
Constant	29.04	23.16		1.25	0.24
CFP_{NFR}	4.6	1.824	0.644	2.52	0.03

Figure 3. Possible relationship between CFP_{nf} and PSU_{nf}

As the level of significance of the linear regression analysis is medium ($p < 0.05$) we conclude that the Hypothesis H_{21} is true. It means that Project Non-Functional Size can be estimated from Product Non-Functional Size.

The linear regression equation obtained is the following:

$$PSU_{NFR} = 4.61 * CFP_{NFR} + 29.04 \quad (4)$$

With respect to the estimation of project size based on product functional size, considering all the observations for calculating the regression analysis model (see Table IV), a low coefficient of determination was obtained (0.14) with a null significance (0.24).

TABLE IV. REGRESSION ANALYSIS MODEL: CFP_{FUR} AND PSU_{FUR}

R	R ²	Std. Error of the Estimate	Significance
0.383	0.147	35.48	0.24

Figure 4. Searching for possible relationship between CFP_{FUR} and PSU_{FUR}

However, plotting the data, as presented in Fig. 4, we can clearly see that two data points could be excluded from the analysis. We did exclude these two observations, and generated the regression analysis model, which is presented in Table V and Table VI. As it can be seen in the table, the R^2 value is strongly improved with a medium significance level ($p < 0.05$).

TABLE V. REGRESSION ANALYSIS MODEL WITHOUT TWO OBSERVATIONS: CFP_{FUR} AND PSU_{FUR}

R	R ²	Std. Error of the Estimate	Significance
0.712	0.507	30.42	0.031

Therefore, we can conclude that the 50.7% of the variation in the project functional size (PSU_{FUR}) can be explained by the variation in the functional size of the product (CFP_{FUR}).

TABLE VI. REGRESSION ANALYSIS: COEFFICIENTS.

	Unstandard. Coefficients		Standard Coeff.	t	Sig.
	B	Std. Error	Beta		
Constant	-29.89	48.42		-0.62	0.55
CFP_{FUR}	1.003	0.373	0.712	2.69	0.031

The linear regression equation obtained is the following:

$$PSU_{FUR} = 1.003 * CFP_{FUR} - 29.897 \quad (5)$$

The constant coefficient is the intercept, which represents the estimated average value of project functional size when the product functional size equals zero. We think that one possible reason for explaining the form of this equation is the matter that we used the default weighting scheme presented in the PSU manual [15] for calculating the PSU numbers. The manual [15] recommends that, in cases when a large number of projects is available for analysis, the weighting scheme be calibrated based on these available project data. We think that the equation in (5) reflects the need to calibrate the weighting schema based on the data from our 11 projects. We, therefore, consider in our immediate future research, to gather new project data, to complete a refined analysis of the project data, and to adjust the weighting scheme. Another future action in this research direction will be the creation of subsets of homogeneous of projects, according to different criteria (e.g. productivity, the median effort per task in the WBS, etc.), replicating the study, and analyzing and comparing the new results from applying the sizing techniques to the homogenous projects subsets against the results from applying the techniques to the whole data sample.

V. VALIDITY EVALUATION

We addressed the various threats [9] to the validity of the results in this experiment. First, **conclusion validity** checks if the findings of the study are correct by evaluating how the data analysis was executed and how the appropriate statistical tests were selected. We consider the threats to conclusion validity to be relatively small for this experiment. The PSU procedure is tested in industrial trials, and robust statistical tests are used. The COSMIC FSM is well tested in other studies comparing different FSM techniques [16]. In addition, the PSU procedure was executed by the same expert for all eleven projects. From the perspective of this expert, the time spent for estimating PSU was as he expected to be. Last, but not least, we must make the note that, despite the matter that all subjects received

the same training level and instructions on how to proceed with the COSMIC process, we did not verify the level to which the students absorbed the knowledge that their teacher (the fourth author of this paper) taught. It is possible that the different learning levels by different students have an impact on how correct the COSMIC procedures were applied. Because we do not know how well the students learnt the two COSMIC procedures, we acknowledge that this poses a threat to conclusion validity.

Next, **internal validity** [9] investigates if the observed relation between the treatment and outcome is really caused by the treatment or if there are other factors that the researcher may not be aware of. The common threats associated with internal validity are history, maturation, instrumentation, mortality and social threats [9]. History and maturation threats are reduced by scheduling the experiment on the same day and by using the same problem statement to produce the use cases which were used as input for each treatment. Furthermore, we believe that the threat to validity due to experiment instrumentation is minimal, because the experiment is conducted on requirements specifications of the same system. The subjects are expected to be familiar with the systems because all students were developing the system as part of their course load in SOEN390. However, quality of functional requirements specification was not verified; and we acknowledge that it could affect the functional size measurement. Mortality and social threats like compensatory rivalry and resentful demoralization do not apply since each subject applies both methods and none of the subjects have dropped out of the experiment.

Next, we acknowledge the following threats to **construct validity**. The material used in the experiment such as data collection forms as well as measured variables are the same as the ones used in the series of the COSMIC counting experiments conducted by other researchers, thus they are reliable [16]. The identified threats are mostly social in nature. We think that the threat to constructive validity due to “hypothesis guessing” [9] is excluded, since there was no risk that the students may try to guess the expected outcome of the experiment and act on it. The students were unaware of that their data would be analyzed in conjunction with PSU. Another possible threat to construct validity, which we identified, is experimenter expectancies, especially about the relationship between project functional size (that is, PSU_{FUR}) and COSMIC CFP_{FUR}. This threat is counteracted by involving an independent researcher (Condori-Fernandez) in the data analysis process. This independent researcher will be also included in the revised experiment design which we plan as our next step.

Last, but not least, we addressed threats to **external validity**, which is mainly concerned with the generalizability of the study findings. A predominant threat in the case of the experiment presented in this paper is the interaction of the setting and the treatment, i.e. there is a threat that the size and complexity of the requirements document used in the experiment will not be representative of the requirements specifications used in industry. This threat is addressed by finding a reasonable size for the requirements specifications

used in the experiment while considering restrictions put forward by the time budget of the experiment. To our best knowledge, the effects identified in this study should be applicable to larger documents as well. The increased size and complexity of the requirements document may increase the time it takes to perform the counting and, then, this should be true for both of the counting techniques involved (these are PSU and COSMIC).

VI. CONCLUSIONS AND FUTURE WORK

The results presented in this paper show that there is a causal relationship between PSU and CFP. For both regressions analysis models, a level of medium significance was obtained. We believe, we can improve this significance if we have more observations. In addition, as said in Section V, we are also interested in improving our experimental design by carrying out a refined analysis of the collected data and use it to adjust the PSU weighting scheme. PSU was born as a ‘living’ project-management based technique for internal usage into an organization and the periodical re-calibration of weights has the aim to reduce progressively the ‘cone of uncertainty’ and not only to serve external benchmarking purposes with a fixed set of weights. This is an incentive for us to replicate our study with other groups of students taught by the same teacher in the next course. In the current academic year, we plan to use the same two courses at the same university as the setting in which to execute our revised experiment design.

We also plan to investigate further the effect of other factors such as project difficulty and combine them into a multiplicative regression model, which may improve significantly the goodness of fit of the project size estimation model.

ACKNOWLEDGMENT

The authors would like to thank Concordia University students enrolled in SOEN337 and SOEN390 courses for their contribution to the research results presented in this paper. The second author thanks also NWO for their financial support under the Quadread and CARES projects.

REFERENCES

- [1] Bundschuh, M., C. Dekkers, The IT Measurement Compendium: Estimating and Benchmarking Success with Functional Size Measurement, Springer, 2008.
- [2] COSMIC, The Common Software Measurement International Consortium. *The COSMIC Functional Size Measurement Method*. Version 3.0.1. May 2009, URL: www.cosmicon.com.
- [3] UKSMA, United Kingdom Software Metrics Association. *MK II Function Point Analysis Counting Practices Manual*. Version 1.3.1, URL: <http://www.eee.metu.edu.tr/~bilgen/MARK%20II%20FP%20Guide.pdf>
- [4] FISMA, Finish Software Management Association. *FISMA 1.1, Functional Size Measurement Method*. 2008, URL: http://www.fisma.fi/wp-content/uploads/2008/07/fisma_fsmm_11_for_web.pdf
- [5] Juristo N., Moreno A, Basics of Software Engineering Experimentation, Kluwer, Boston, 2001.
- [6] Fenton, N., Pfleeger, S.L.: Software Metrics: A Rigorous and Practical Approach, 2nd ed. International Thompson Computer Press, 1998.

- [7] Buglione, L.: Some Thoughts on Productivity in ICT projects, WP-2008-02, White Paper, version 1.2 (July 2008), URL: <http://www.semq.eu/pdf/fsm-prod-120e.pdf>
- [8] Project Management Institute, A Guide to the Project Management Body of Knowledge, 4th ed. 2008, ANSI/PMI 99-001-2008.
- [9] Wohlin, C., P. Runeson, M. Höst, M.C. Ohlsson, B. Regnell and A. Wesslén, *Experimentation in Software Engineering: An Introduction*, Kluwer, Boston, 2000.
- [10] ISO/IEC, IS 15939:2007 - *Systems and software engineering — Measurement process*, International Organization for Standardization, 2007.
- [11] McConnell, S., *Software Estimation: Demystifying the Black Art*, Microsoft Press, 2006.
- [12] 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.
- [13] 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
- [14] Buglione, L.: Project Size Unit (PSU) – Calculation feature in Project Management tools - Requirements, v1.0, PSU-AU-1.00e (December 2006) (2009-12-14), URL: <http://www.semq.eu/pdf/psu-au-100e.pdf>
- [15] Buglione, L.: Project Size Unit (PSU) - Measurement Manual, version 1.21 (November 2007), <http://www.semq.eu/pdf/psu-mm-121e.pdf>
- [16] 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.
- [17] IEEE, Software & Systems Engineering Vocabulary (SEVOCAB), IEEE Computer Society & ISO/IEC JTC1/SC7, http://pascal.computer.org/sev_display/index.action
- [18] Wieringa, R., Design Science Research In Information Systems and Technologies, Proc. of the 4th Int. Conference on Design Science Research in Information Systems and Technology, ACM Press, 2009.
- [19] ISBSG, Glossary of Terms, version 5.10.2, International Software Benchmarking Standards Group (2009-12-14), URL: <http://www.isbsg.org/>
- [20] Boehm B., *Software Engineering Economics*, Prentice-Hall, 1981
- [21] Little T., Schedule Estimation and Uncertainty Surrounding the Cone of Uncertainty, *IEEE Software*, May/June 2006, Vol 23 No.3, pp.48-54
- [22] Basili V., Caldiera G., Rombach H.D., Goal Question Metric Approach, *Encyclopedia of Software Engineering*, pp. 528-532, John Wiley & Sons, Inc., 1994.
- [23] Magne Jørgensen, M. J. Shepperd: A Systematic Review of Software Development Cost Estimation Studies. *IEEE Trans. Software Eng.* 33(1): 33-53 (2007)