ISSN 1330-3651
UDC/UDK 004.415.5

SOFTWARE PROCESS MEASURING MODEL

Ninoslav Slavek, Ivica Lukić, Mirko Köhler

Original scientific paper

In this paper the Software Process Measuring Model (SPMM) is described. SPMM is a method for software process assessment, quantitative measurement and improvement for software producing organizations (SPOs). It has been developed partly based on a renovation of the CMM/CMMI, Bootstrap and SPICE methods, standards ESA PSS 05, and ISO 90003. SPMM focuses on the software development process in software production enterprises. The article explains the central concept of gaining data about software engineering organizations with a thoroughly constructed questionnaire. It gives a ground to measure the quality maturity level of organization and its projects. The SPMM can be interpreted as a method for describing where an organization stands and what changes are to be recommended in the next steps. The main idea of the SPMM is to determine the process maturity profile of an SPO. The goals of a SPMM self-assessment are: a) to measure and develop an SPO maturity quality profile showing strengths and weaknesses of the SPO assessed, b) to derive the steps for improvement from the shown quality profile. The result of one day assessment in software production organization X (SPO X), and Project X within the SPO X which was held at the beginning of October 2010 is presented. The result of the assessment showed the total organization and methodology maturity levels of the Project X. The organization is on maturity level 2, 83. The methodology is on maturity level of 2, 48. The total maturity level of the organization of SPO X is on maturity level of 2, 42, and the methodology is on maturity level of 2, 57. The organization of the paper is as follows: after the introduction in section one, section two explains the reasons of the SPMM development. Section three depicts the SPMM development. The maturity level algorithm is explicated in the next section. Section five explains the evaluation of the SPO, the assessment results are in section six. The conclusion is given in section seven, and the list of literature in section eight.

Keywords: *software quality, software process, software process assessment, measuring model*

Model mjerenja softverskog procesa

Izvorni znanstveni članak

U ovom radu opisan je Model mjerenja softverskog procesa (MMSP). MMSP je metoda procjene softverskih procesa, kvantitativnog mjerenja i unapređenja procesa za organizacije koje se bave razvojem softvera (SPO). Metoda je razvijena dijelom na temelju poboljšanja metoda CMM/CMMI, Bootstrap i SPICE, i na standardima ESA PSS05 i ISO 90003. U žarištu MMSP-a je proces razvoja softvera u softverskim poduzećima. Članak objašnjava glavni koncept dobavljanja podataka o softverskim inženjerskim organizacijama i njihovim projektima pomoću temeljito izgrađenog upitnika. MMSP se može interpretirati kao metoda za opisivanje kakav je položaj organizacije i koje se promjene predlažu u slijedećim koracima. Osnovna ideja MMSP-a je utvrditi profil zrelosti procesa SPO-a. Ciljevi MMSP procjene su: a) izmjeriti i razviti profil zrelosti kvalitete procesa prikazom jakih i slabih strana procijenjenog SPŠO-a, b) derivirati korake za unapređenja iz prikazanog profila kvalitete procesa. Prikazan je rezultat procjene obavljene u jedan dan u organizaciji koja se bavi proizvodnjom softvera (SPO X) i Projekta X unutar SPO-a X koji je održan početkom listopada 2010. Rezultati procjene prikazuju ukupne organizacijske i metodološke razine za Projekt X. Organizacija je na razini zrelosti od 2, 83. Metodologija je na razini zrelosti od 2,48. Ukupna razina zrelosti za organizaciju SPO X je na razini zrelosti od 2,42, dok je metodologija na razini zrelosti od 2,57. Organizacija članka je sljedeća: nakon uvoda u poglavlju jedan, poglavlje dva objašnjava razloge razvoja sustava MMSP. Poglavlje tri opisuje razvoj MMSP-a. Algoritam razina zrelosti je prikazan u slijedećem poglavlju. Poglavlje pet objašnjava evaluaciju SPO-a, rezultati procjene prikazani su u poglavlju šest. Poglavlje sedam sadrži zaključak, popis literature je u poglavlju osam.

Ključne riječi: *kvaliteta softvera, softverski proces, procjena softverskog procesa, model mjerenja*

1 Introduction

The Software Process Measuring Model (SPMM) [14, 15, 16] has been developed at the Faculty of Electrical Engineering in Osijek, Croatia, by the Software Engineering Institute at the University of Osijek, and is based on the CMM (Capability Maturity Model) [3, 13], CMMI (Capability Maturity Model Integration) [2], Bootstrap [1, 2, 4, 10, 12] and SPICE model [3, 7, 8]. It has been developed in conformance with the emerging ISO standard [6] for software process assessment and improvement, and is aligned with the model with the European Space Agency-ESA PSS 05 standard [17]. The SPMM is an assessment method that can be applied to small and medium size Software Producing Organizations (SPOs), or software departments within large companies. Beside a general overview of SPOs and their projects, SPMM focuses on the quality of software production. It identifies individual attributes of an SPO or individual software project and has separate questionnaires for the assessment of an organization's quality system (SPO questionnaire) and the assessment of the software project within this organization (Project questionnaire). The purpose of SPMM is to provide a sort of algorithm for evaluating an SPO. This evaluation also provides a method for measuring improvements motivated by the need to improve the effectiveness, quality and competitiveness of SPOs. The main goals are to identify the best practices of an SPO, to establish process models (work flows, data flows, managerial aspects and resources) to identify strengths, weaknesses and bottlenecks, and to establish quality control function to check if the best practices are being efficiently implemented. SPMM characteristics are:

- it is oriented towards practical use,
- it can be used primarily for the first - party assessment (self - assessment), where it is performed internally inside the SPO by the SPO's own personnel in order to identify the SPO's own software process capability.
- it can be used also for second party assessment for capability determination, where external assessors are used to perform the assessment in the SPO, in order to evaluate the SPO's capability to fulfil specific contract requirements,
- it is easy to use and understand for the audited (SPOs).

The main features of the method are:

a) the assessment process: it is part of the improvement. Assessment results provide the main input for the improvement action plan and provide feedback from the improvement activities implemented. During assessment the organizational processes are evaluated to define each process whose capability evaluation is based on the SPMM process model.

b) the process model: it defines processes and capability levels. Process capability is measured based on five maturity levels according to Paulk at al. [13]:

- Level 1: Initial process – the software process is characterized as ad hoc, even chaotic. Few processes are defined, success depends on individual effort.
- Level 2: Repeatable Process – basic management processes are established to track cost, schedule, and functionality. The necessary process discipline is in place to repeat earlier successes on projects with similar applications.
- Level 3: Defined Process - the software process for both management and engineering activities is documented, standardized and integrated into an organizations-wide software process. All projects use a documented and approved version of the organization's process for developing and maintaining software.
- Level 4: Managed Process – detailed measures of the software process and product quality are collected. Both the software process and products are quantitatively understood and controlled using detailed measures.
- Level 5: Optimizing Process – continuous process improvement is enabled by quantitative feed back from the process and from testing innovative ideas and technologies.

c) rating, scoring, presentation: assessment results are the basis for software process planning if assessment data provides a good representation of the assessed company's capability.

d) process assessment guidelines: the guidelines support the identification of processes that have impact on the achievement of the company goals, improvement priorities are assigned to processes with low capability and high impact. Improvement targets and priorities are evaluated in order to help management to undertake the improvement effort. The structure of the SPMM process model is shown in Fig. 1.

2 Reasons of SPMM development

Beside the existing CMM, CMMI, Bootstrap, and SPICE methodology that are based on the third parties for carrying out assessments, reason for development of SPMM is to provide a framework for self-assessment of software engineering practice. The idea is that a company can have its own assessors who can make assessments of the projects at any time and can identify the core attributes of the characteristics of the company, and software projects. Software Engineering Institute at the University of Osijek has prepared a framework for self-assessment training that includes:

- producing a metrics assessment questionnaire
- development of a reference training curriculum
- setting reference certification criteria.

2.1 SPMM resources

The main sources for the establishment of the SPMM and SPMM questionnaire are:

Standard ISO 90003:2000 [6]. This standard cancels and replaces old ISO 9000-3: 1997, which has been updated for conformity with ISO 9001:2000. This International Standard provides guidance for organizations in the application of ISO 9001:2000 to the acquisition, supply, development, operation and maintenance of computer software and related support services. In general, it specifies requirements for a quality system where an organization:

a) needs to demonstrate its ability to consistently provide product that meets customer and applicable regulatory requirements, and
b) aims to enhance customer satisfaction through the effective application of the system including processes for continual improvement of the system and the assurance of conformity to customer and applicable regulatory requirements.

Standard European Space Agency-ESA PSS 05 [17] is concerned with all software aspects of a system. It does not make use of any particular software engineering method or tool mandatory. The standard describes the mandatory practices, recommended practices and guidelines for software engineering projects, and allows each project to decide on the best way of implementing them.

Bootstrap is the European project whose aim is to develop a method for software process assessment and improvement of Software Producing Units (SPUs). An SPU is a small or middle sized software producing company or a software department of a large company in which projects are performed to develop software products [1, 4]. The

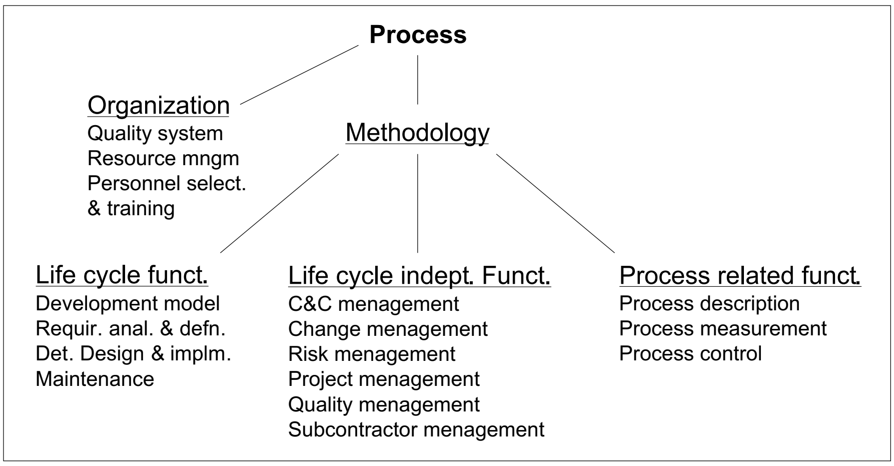

Figure 1 SPMM Process model

Bootstrap methodology is built on the original concept of the software Capability Model (CMM), and incorporates accepted standards ESA–PSS 05 and ISO 9000. The basis of the methodology is to provide a reliable and consistent benchmark on which to assess existing practices and processes and define a set of actions and initiatives which would improve quality, reliability, development costs and productivity. Bootstrap methodology covers two key areas:

1. the assessment which provides an in – depth critical study detailing the capability of each major aspect of software development environment,
2. an action plan for step by step evolution to provide both the organization and the process.

Bootstrap's hypothesis is that before investments are made in technology upgrade through products, the questions on how to build solutions i.e. methodology and methods, and how to organize development and maintenance of software have to be solved. The formula of priorities in level and schedule is: O>M>T [4]. Bootstrap has developed a method to determine the profile of an SPU, showing its strengths and weaknesses using software process measurement. It identifies individual attributes of an organization or individual project and has separate questionnaires for the assessment of an organization's quality system (Global questionnaire) and the assessment of the project within this organization (Project questionnaire). The global questionnaire investigates whether the organization is providing the necessary resources for the software systems development, while the project questionnaire investigates how effectively the projects are making use of these resources. According to the Bootstrap hypothesis, "attribute trees" are defined which cover three major aspects of the organization and projects, namely organization (O), methodology (M) and technology (T). The maturity level algorithm enables the calculation of a maturity level for each individual process quality attribute.

The Capability Maturity Model (CMM) is a framework that describes the key elements of an effective software process. It describes an evolutionary improvement path from an ad hoc, immature process to a mature, disciplined process. The CMM covers practices for planning, engineering, and managing software development and maintenance. When followed, these key practices improve the ability of organizations to meet goals for cost, schedule, functionality, and product quality. This model defines five levels of process maturity plus an improvement framework for process maturity and as a consequence, quality and predictability.

The Capability Maturity Model Integration is a process improvement approach that provides organizations with the essential elements of effective processes [2]. CMMI best practices enable organizations to do the following:

a) more explicitly link management and engineering activities to their business objectives,
b) expand the scope of and insight into the product lifecycle and engineering activities to ensure that the product or service meets customer expectations,
c) incorporate lessons learned from additional areas of best practice (e.g., measurement, risk management, and supplier management),
d) address additional organizational functions critical to their products and services
e) comply with relevant ISO standards.

The CMMI is structured into five maturity levels, the considered process areas, specific goals (SG), generic goals (GG), specific practices (SP) and generic practices (GP).

3 SPMM development

The outline of the processes that were performed to develop SPMM is:

- Definition of metrics – metrics is defined as a measure of extend or degree to which a product or process possesses and exhibits certain quality characteristics. Metrics are instruments with the aims of: making software more reliable, supporting project management with process and product data, and achieving continual process improvement. SPMM focus on process metrics.
- Design of the information collection method – SPMM minimizes the number of questions and minimizes the high complexity of questions.
- Design of the evaluation method including weighting and scoring – the aim is to provide significant information (e.g. the process performed in the organization, the characteristics of the organization, etc.).

The evaluation should deliver general information and detailed information for analysis.

3.1 Extracts from Bootstrap questions

The main source for establishing SPMM process questionnaires is the Bootstrap questionnaire. There are some basic differences: (1) SPMM does not take into account technology questions as it is focused on processes for SPOs and projects, (2) some questions have been redesigned according to the new ESA PSS 05 and ISO 90003 standards.

The SPMM and Bootstrap maturity questionnaire are metrics used to quantitatively assess software process. Therefore it is possible to analyze these using general criteria for tentative metrics. Each question, when performing assessment, is answered on a four-point scale: absent represents 0 %, basic 33 %, significant 66 %, and extensive 100 %.

3.2 Influences

Influences from ESA PSS 05: the life cycle part of the SPMM questionnaire follows the structure suggested by the ESA PSS 05 standard. It describes a comprehensive and phase oriented software development model. This standard is based on the IEEE software engineering standards.

Influences from CMMI: it describes the five maturity levels of software processes, of which a number of key process areas are defined. Each key process area contains a number of key practices that have to be performed. The SPMM method evaluates each process attribute separately on a five-point maturity scale.

Influences from ISO 90003: it describes a number of software process quality attributes. SPMM can evaluate about 85 % of attributes.

4 SPMM maturity level algorithm

SPMM developed an algorithm that produces results, taking into account the following facts:
- whether the algorithm fits the complexity of software engineering,
- whether the algorithm minimizes dependence on individual assessors,
- whether the algorithm is based on steps instead of percentages and a scale with variable distances between the levels,
- whether the algorithm has enhanced evaluation capabilities as it has a questionnaire for the organization and for the project, and is able to compare the organization profile with the profile of the projects.

The SPMM Questionnaire is a subset of $N \times L \times S$, with $L = \{2, 3, 4, 5\}$ representing the set of levels, $S = \{n/a, 0, 0{,}33, 0{,}66, 1\}$ representing the set of possible scores and N representing the set of question numbers. Each evaluated question q is an element of set q and Q. The maturity level is a function that maps Q or a subset $V \subseteq Q$ in case of an individual attribute, onto a value between 1 and 5 on the maturity level scale.

$$M_L : V \Rightarrow [1,5],\ Q \subseteq N \times L \times S,\ V \subseteq Q. \tag{1}$$

The algorithm first calculates the number of steps and then the steps are mapped onto the dynamic scale obtaining a maturity level value.
- Calculation of the total number of steps:

#steps[i] = number of steps fulfilled on maturity level i, i = 2,...,5

$$\#\text{steps}\left[i\right] = \sum_{j=1}^{d[i]} \text{score}\left[x_{ji}\right] \tag{2}$$

$$\#\text{steps}_{\text{total}} = \sum_{i=2}^{5} \#\text{steps}\left[i\right] \tag{3}$$

- Implementation of function G. G represents a function that maps a number of steps onto the maturity level scale using the following algorithm:

If (#steps) =< d [2] then
ML: = 1 + (#steps/d[2])
else if (#steps=<d[2]+d[3]) and (#steps> d[2])then
ML: = 2 + (#steps- d [2]) / d [3]
else if(#steps=<d[2]+d[3]+d[4])and(#steps> d[2])
ML: = 3+ (#steps- d [2]-d [3]) / d[4]
else if (#steps=<d[2]+d[3]+d[4]+d[5])and(#steps> d[2]+d[3]+d[4]) then
ML: = 4 + (#steps-d [2]-d [3]-d [4]/d[5]
G: = ML rounded to a quarter; End.

4.1 Technical references

SPMM is a Web application. For that purpose, the MS SQL data base, SQL language, and IIL (Internet Information Server) were used, and for communication between the DB server and server and browser (Internet explorer) the PHP language PHP was used.

4.2 Software process assessment

Software process assessment helps software organizations improve themselves by identifying their critical problems and establishing improvement priorities, and its objectives are:
- to learn how the organization works
- to identify its major problems
- to involve its decision makers in the change process.

Assessment is conducted in three phases: preparation, on-site assessment, and findings and action plan recommendations. As SPMM is a self-assessment tool, a local assessment group should be established. The team members should be experienced software developers and at least one should have experience in each phase of the software process. Also, the assessment team should go through the SPMM's self assessment training.

At the assessment conclusion, the team prepares a report on its initial findings. The report is a composite summary of site status, together with findings in key areas. At the final assessment, the team presents a final report and recommendations to the site manager and staff. The recommendations should highlight the four items of highest priority. The action plan is then prepared. The organization should conduct follow-up assessments one year after the initial action plan is developed and improved for several reasons: a) to assess the progress that has been made, b) to provide milestones for completion of the actions from the prior assessment, c) to establish new priorities for continued improvement.

5 Evaluation of SPO

After the assessment, the data collected enables the evaluation of SPO.

The results are organizational attributes and methodology attributes including life cycle attributes, life cycle independent attributes and process related attributes of an SPO and separately of the project.

Interpretation of a process quality profile:
- Maturity level 1 – Characteristic: - unpredictable cost, schedule, and quality performance. Needed actions: planning, performance tracking, change control, quality assurance
- Maturity level 2 - Characteristics: - effective method is in place, cost and quality highly variable, control of schedules, informal and ad hoc process methods and procedures. Needed action: develop process standards and definitions, assign process resources, and establish methods.
- Maturity level 3 – Characteristics: - reliable costs and schedule, unpredictable quality performance, the effective methods are documented and standardized. Needed actions: establish process measurements and quantitative quality goals, plans, measurements, and tracking.
- Maturity level 4 – Characteristics: quantitative – reasonable statistical control over product quality. Needed actions: productivity plans and tracking, instrumented process environment.
- Maturity level 5 - Characteristics: quantitative base for continued capital investment in process automation and improvement. Needed actions: continued emphasis on

process measurement and process methods for error prevention.

6 Assessment results

At the beginning of October 2010 (06.10.2010) the SPO X (Company X) and Project X were subjected to one day SPMM check. The Company X is a software organization situated in different locations (Croatia, BRD, Austria, and China) with a distribution of the responsible project roles to these locations.

Project X is a software development project for mobile phone handling characterised by the following data: time scaling for software development and integration testing approximately 16 months, product size about 600 KLOC, reuse from previous product approximately 35 %, project staffing included project manager, three software developers, three programmers, system tester, two code testers, and documentation person.

The check essentially comprises the steps:
- requesting and viewing the SPU and project documentation
- interviewing of SPO and project responsible.

In the framework of the interviews the following activities were performed:
- assessment of documents
- working through the questionnaires including assessment
- working throughout the assessment findings.

Based on the calculated maturity profiles four steps have to be followed: 1) Identification of MLs for the SPO and the Project. This is identification of all attributes which have the lower ML than the entire SPO or Project. These attributes represent areas for improvement. 2) Identification of strengths. This is the identification of attributes that have higher ML than the entire SPO and the Project. 3) Presentation. The SPO and the Project MLs are discussed in the feedback meeting on site. 4) Discussion of improvements. Based on the assessment findings the planning of improvement activities is discussed.

The result of the assessment in SPO X, and Project X are shown in tables bellow. Tab. 1 shows the total organization and methodology maturity levels for Project X. A maturity level of 3 (2,83) for the Organization means that the most effective methods are documented and standardized. A maturity level of 2 (2,43) for the Methodology means that the method is effectively used but weakly documented and standardized. Tab. 2 shows the total maturity levels for the Project X. The maturity level of Risk avoidance and management (1,82), and Software requirements (1,66) is weak. These attributes are the areas for improvement. Risk management is the process of identifying, assessing and documenting risk to the project. It includes the identification of possibly unstable parts of the specification and the existence of guidelines for fixing these instabilities. The low ML for this attribute means that these activities are not provided well. Software requirements serve as an input to the design specification. It must contain sufficient detail in the functional system requirements so that a project design solution can be devised. The ML for Process measurement is very high (3,90) which means that the measurements are provided for all projects, they are well documented and standardized. Tab. 3 shows the total organization and methodology maturity levels for SPO X, and Tab. 4 shows the maturity levels of SPO X.

Graphical presentation of the comparison of maturity profile for some attributes of SPO X (blue color) and Project X (red color) is shown in Fig. 2. For example, the interpretations of the comparison for some maturity profile of the SPO X and the Project X attributes are as follows: process description on level 2,66 is sufficiently provided by the SPO X, but not effectively used by the Project X which is on level 2,10. The process measurement is very high for the SPO X (3,90) but not followed by the Project X (2,21). The primary purpose of measurement is to provide insight into software processes and product so that an organization is better able to make decisions and manage the achievement of goals. The measurement is often equated with collecting and reporting data and focuses on presenting the numbers. SPO X develops, plans, implements, and improves a measurement process, but it is not sufficiently followed by the Project X. The project management for the SPO X is relatively weak (2,32), and high for the Project X. This suggests that upper management does not enough recommend the use of project management methods and tools. This caused project managers to react by themselves and to develop their own individual methods.

Table 1 Total Organization and Methodology Maturity Levels of the Project X

Attributes	ML
ORGANIZATION	2,83
METHODOLOGY	2,48

Table 2 Total Maturity Levels of the Project X

Attributes	ML
Coordination	2,86
Development	2,64
Testing, Verification & Validation	2,73
Support	3,31
Process description	2,10
Risk avoidance and management	1,82
Project management	2,21
Quality management	2,26
Process measurement	3,32
Process control	2,76
Software configuration & change management	2,41
Supplier management	2,98
Development model	2,66
Systems for special purpose	2,55
User requirements	2,32
Software requirements	1,66
Architectural design	1,99
Detailed design and implementation	2,43
Testing	2,32
Integration	1,99
Acceptance testing & transfer	2,32
Operation & maintenance	2,44

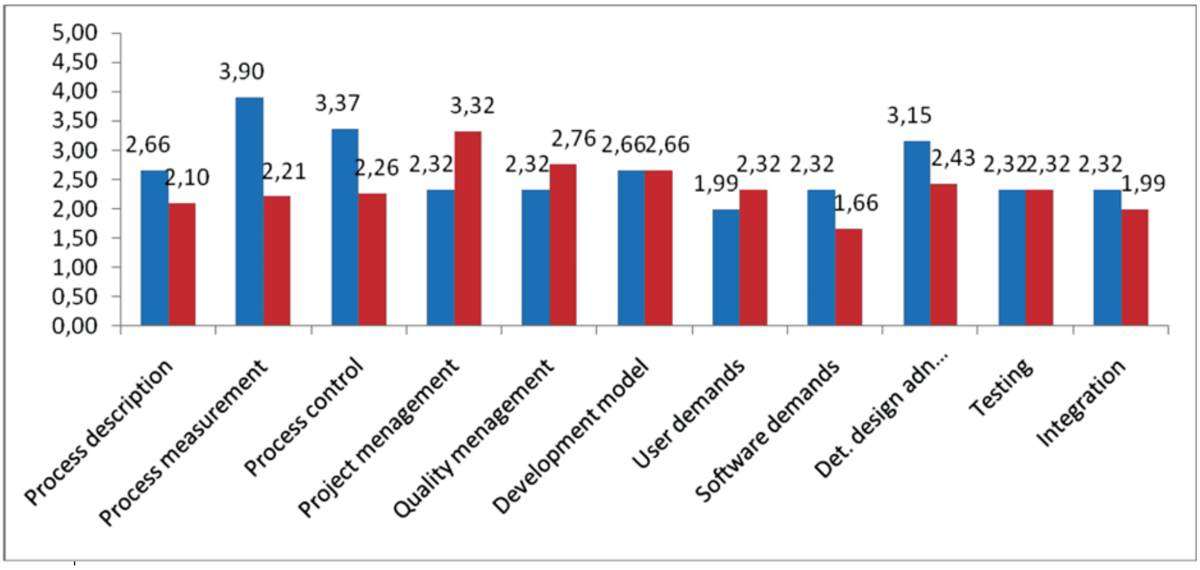

Figure 2 Comparison of the SPO X and Project X

Table 3 Total Organization and Methodology Maturity Levels of SPO X

Attributes	ML
ORGANIZATION	2,42
METHODOLOGY	2,57

Table 4 Maturity Levels of SPO X

Attributes	ML
Quality assurance	2,32
Resource management	2,43
Staff selection and training	2,56
Process description	2,66
Risk avoidance & management	2,15
Process measurement	3,90
Process control	3,37
Project management	2,32
Quality management	2,32
Configuration & change management	2,41
Supplier management	2,49
Development model	2,66
Systems for special purpose	2,49
User requirements	1,99
Software requirements	2,32
Architectural design	1,99
Detailed design & implementation	3,15
Testing	2,32
Integration	2,32
Acceptance testing	2,49
Operation & maintenance	2,99

7 Conclusion

SPMM is an assessment method for software producing organizations (SPOs). It focuses on the quality of software production identifying individual attributes of an SPO or individual project using separate questionnaires for the assessment. The purpose of SPMM is to provide a sort of algorithm for evaluating SPOs and a method for measuring improvements motivated by the need to improve the effectiveness, quality and competitiveness of SPOs. The main goals are to identify the best practices of an SPO, to establish process models to identify strengths, weaknesses and bottlenecks, and to establish a quality control function if best practices are being efficiently implemented. At the beginning of October 2010 the SPO X and its Project X were subjects of process assessment. The participants were the two assessors from the Faculty of Electrical Engineering Osijek, and the SPO X team consists of the senior manager, QA manager, project manager, software designer, and research methodologist. The SPMM assesses the state of SPO X and the Project X. It measures the software development process in two main areas: a) organization, b) methodology. SPMM identifies strengths, weaknesses, highlights areas for improvement, and makes recommendations to correct the weaknesses noted in the assessment. The interviews took two hours, the assessors recorded the data using SPMM questionnaire. After collecting the data the SPMM team performed an analysis using the SPMM algorithm. The SPMM check differentiates between questions of level 2, 3, 4, and 5, the results are shown in the tables above. The majority of the improvement potential listed below results from the viewing of project and the organization and discussions in the framework of the SPMM check during answering the questionnaire. The weak points for the Project X are: a) risk avoidance and management, b) process description, c) quality management, d) software requirements. The strong points are: a) process measurement, b) support, c) supplier management, and d) coordination. The weak points for the Organization X are: a) user requirements, b) architectural design, and c) risk avoidance and management. The future plans of the SPMM are to conduct the assessments for a lot of the software production companies in our country. Next goal is to produce the data base in which all the data of any SPMM assessment will be collected.

8 References

[1] Cachia, R.; Maiocchi, M.; Bicego, A. Project report "Bootstrap ESPRIT Project No 5441", Bootstrap Institute, Milano, 1993.

[2] Chrisis, M. B.; Konrad, M.; Shrum, S. CMMI - Guidelines for Process Integration and Product Improvement, Addison – Wesley, Reading, USA, 2006.
[3] Ebert, C.; Dumke, R. Software Measurement, ISBN 978-3-540-71648-8, Springer, Berlin, NY, 2007.
[4] Hasse, V.; Koch, G.; Messnarz, R. Process and Product Measurement, IIG – Report – Series, Institutes for Information Processing, Graz, 1993.
[5] Humphrey, W. Managing the Software Process, Addison-Wesley Publishing Company, ISBN: 0-201-18095-2, USA, 1989.
[6] ISO 90003, Guidelines for the application of ISO 9001:2000, Standard, USA, 2000.
[7] ISO 15504 SPICE, Assessment Instrument Standard, Assessment Instrument Product Description.
[8] ISO 15504-5:2006 Information technology – Process assessment, Part 5; An examplar Process Assessment Model, Standard, ISO 2006.
[9] ISO/IECT/JTC1/SC7/VVG10, 1994. Standard
[10] Kuvaja, P.; Simila, J.; Krzanik, L.; Bicego, A.; Koch, G.; Saukkonen, S. Software Process Assessment and Improvement: The Bootstrap Approach, Blackwell Puiblishers, Oxford, 1994.
[11] Laird, L. M.; Brennan, M. C. Software Measurement and Estimation: A Practical Approach. IEEE Computer Society Press, 2006.
[12] Messnarz, R. Design of a Quantitative Quality Evaluating System (QUES), Dissertation G – Report – Series, Institutes for Information Processing, Graz, 1993.
[13] Paulk, M. C.; Curtis, B.; Chrissis, M. B. Capability Maturity Model for Software, SEI, CMUS/SEI-91-TR-24, Pittsburg, USA, 1991.
[14] Slavek, N.; Lukić, I. Improvement of the SPMM Method, Science in Practice, 2008.
[15] Slavek, N.; Jović, F.; Blažević, D. Toward Software Process and Software Product Quality Integration, CADAM, Rijeka, 2007.
[16] Slavek, N.; Jović, F.; Blažević, D. Quality Factors for the Real-Time Embedded and Safety-Critical Software. EUROSIM, 2004.
[17] Standard ESA PSS 05, 1996.
[18] Tayntor, C. B. SixSigma Software Development. CRC Press, 2003.

Authors' Addresses

Doc. dr. sc. Ninoslav Slavek, dipl. ing. el.
Sveučilište J. J. Strossmayera u Osijeku
Elektrotehnički fakultet
Kneza Trpimira 2b
31000 Osijek, Hrvatska
e-mail: ninoslav.slavek@etfos.hr

Ivica Lukić, dipl. ing. el.
Sveučilište J. J. Strossmayera u Osijeku
Elektrotehnički fakultet
Kneza Trpimira 2b
31000 Osijek, Hrvatska

Mirko Köhler, dipl. ing. el.
Sveučilište J. J. Strossmayera u Osijeku
Elektrotehnički fakultet
Kneza Trpimira 2b
31000 Osijek, Hrvatska