

BASIC MANAGEMENT CHARACTERISTICS OF SOFTWARE ENGINEERING

Duško KOSTIĆ¹, Ljubiša JOSIMOVIĆ², Toma PETROV³

¹ Higher Technical School of Entrepreneurship, Belgrade, Serbia

² Polytechnic School, Požarevac, Serbia

³ JP Termoelektrane, Kostolac, Serbia

Abstract—Scientific and professional discipline that deals with all aspects of production is the software engineering. Primarily, it's related to models, methods and tools that are needed that on the cheaper and efficient way could produce high quality software ventures. The management of software engineering includes the possibility of maintenance, reliability and security, efficiency and usability of software that users expect from it with all its performance that follow the latest trends of the most contemporary technologies and processes.

Keywords—management, software engineering, software tools.

I. INTRODUCTION

THE field of software engineering management deals with the management and evaluation of software engineering. It is defined as the application of the management planning activities, coordination, measurement, monitoring, control and reporting due to systematic insurance, discipline and quantity.

Management of the software engineering can be done in the same way as for any other complex field, though there are specific circumstances relating to the management of software manufacture.

Generation of software engineering is the need for new requirements or change of existing ones, where the client perceives a lack of understanding of the complex content of software engineering process. From this point of view, the management of software engineering represents a very demanding and hardly sustainable business between the two contradictions, the degree of novelty and complexity of the software. All of this is followed by a major change in technology development.

In software engineering management activities are carried out at three levels:

- organizational or infrastructure management,
- project management,
- planning and control of measurement programme.

II. MANAGEMENT OF SOFTWARE ENGINEERING

If we analyze Figure 1, which represents the management scheme of software engineering according to SWEBOK-u (The Software Engineering Body of Knowledge) [1], we'll notice that there are six chapters that deal with the management of software engineering:

- decision to initiate the project of software engineering,
- activities undertaken for successful management of software engineering,
- accepted activities that occur during the process of software engineering,
- software quality assurance,
- completion of the software engineering project,
- development and implementation of effective control of measurement and evaluation in the organization of software engineering.

The management process represents activities aimed to ensure goals and standards of the organization, and the measurement and evaluation represent an assignment of values and meaning of aspects of software engineering and models. Management based on the measurements can change the aforementioned relationship.

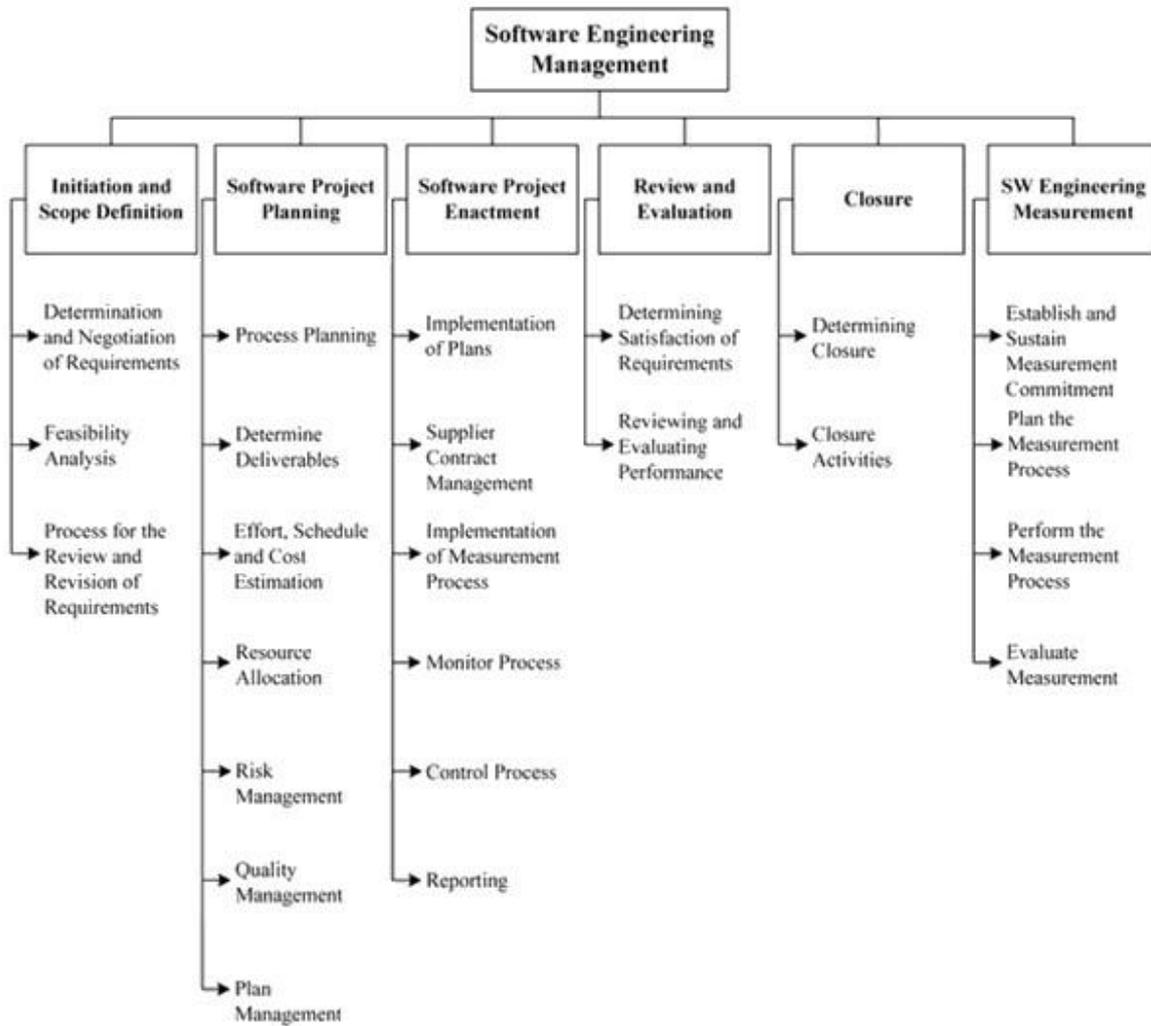


Fig. 1: The management scheme of software engineering according to SWEBOK (The Software Engineering Body of Knowledge) [1]

The methods of software engineering according to SWEBOK can be divided into three fields [1-5]:

- heuristic methods,
- formal methods,
- prototype methods.

All the above methods are dealt with informal approaches (heuristic), mathematically based approaches (formal) and various forms of prototype (prototype). We can not say that the mentioned methods are completely separated. Heuristics are further divided into structural, object-oriented, data-oriented and specialized methods in particular fields of application. Formal methods are dealt with mathematically-based methods of software engineering, notations and specification of languages, methods of specification transformation, verification and techniques of proof, while the prototype methods consider the various styles of prototypes, goals of prototypes and evaluation of prototype techniques.

Primarily, we will focus on tools for software configuration management, where they are divided into three categories: tools for tracking, version management and publishing tools. Improvement and tracking of

defect is crucial in solving a problem in a given software product. For this purpose are used the tools for management in the management of multiple versions of the product. Tools for packaging and publishing of software with the inclusion of installation tools involve commissioning activities and configuration of software installation product [1-5].

Tools for software engineering management are divided into three categories: planning and project monitoring, risk management and measurement. Evaluation of effort, estimation of costs and project schedule checking are updated by tools for planning and project monitoring. And, for identification, assessment and monitoring of risk are used the risk management tools.

Tools of software engineering process are divided into tools for modeling, management and software development environments. In the process of development are used the CASE tools that perform multiple functions and are used through the entire software engineering process. CASE tools can be for:

- configuration management,
- date modeling,
- transformation model,
- refactoring,
- UML modeling tools.

In addition to mention tools, there are tools for review and revision, or software quality tools that are divided into two categories: inspection and analysis tools.

III. OVERVIEW OF THE BEST TOOLS IN USE

At the very beginning were used the tools with simple solutions that were suitable for that stage of software development. As time passed, softwares were developed being involved in the whole process of development.

In the modern world, software is virtually inevitable and ubiquitous. Today, software is a key to the success of many computer systems and also differentiator factor of organizations that possess it. Software has become an essential component in business decision-making and basis in scientific research and engineering problem solving. Also it represents a significant component in industrial, transportation, medical, telecommunications, military and a number of other types of systems.

Software engineering is linked and relies on the discipline of computer science, computer engineering, management, project management, quality management, software ergonomics and systems engineering. It is defined as the application of systematic and disciplined approach to the development, management and maintenance of software. It represents an engineering discipline that deals with all aspects of software production and involves knowledge, tools and methods for defining software requirements, software design, software construction, software testing and software maintenance [6].

The term itself Software Engineering has come to life during the conference on Software Engineering held 1968th in Garmish (Germany), by the Chairman of the Conference F.L. Bauer.

Software engineering involves important fields, as follows:

- Business running and IT
- software development methodologies and frameworks
- development costs
- duration of development
- risks in software development
- quality thinking incorporation in the software development process
- testing
- management of development teams
- project management
- project reporting

Table 1 summarizes the list of the best tools in use in several stages during the development.

Table 1: Overview of the best tools

Name of tools	Manufacturer
Application Development Management	Serena Software
Atlassian - Integrated ALM Suite	Atlassian
Borland Management Solutions	Borland
Caliber Analyst - Requirements Definition and Management	Borland
CASE Spec	GODA Software
Change	Telelogic (IBM)
Codendi	XEROX
Collaborative ALM environment	Artisan Software Tools
Collaborative modeling tool suite	Artisan Software Tools
DeployLX Software Protection System	XHEO
DevSuite	TechExcel
DOORS	Telelogic (IBM)
Endeavour software factory	Info Support
HP Quality Center	HP
Integrated and collaborative project management add-on - EZ!PM	Easy!Software
Kovair Global Lifecycle	Kovair
Lighthouse	Artifact Software
MKS Integrity	MKS Inc.
Neuma CM+	Neuma Technology
Parasoft Concerto	Parasoft
Plastic SCM	Codice Software
Polarion ALM	Polarion Software Inc.
QA Wizard Pro - Automated testing	Seapine Software
Rally	Rally Software
Rational Team Concert	IBM
Sauce Labs - cloud-based, concurrent Selenium test execution	Sauce Labs
SCM Anywhere	Dynamsoft
SCM4ALL	IKAN
Silk Suite - Test Management, Functional and Performance Testing	Borland
Synergy CM	Telelogic (IBM)
Team Focus - Project Management	Borland
TOMOS - SaaS solution for	TOMOS Software

ALM targeted at analysts,
developers, testers, project
managers

Usecase Help www.Usecase.com

Visual Studio Team System Microsoft

IV. CONCLUSION

Software process is a set of activities with the results whose aim is the development of software, with basic activities within software process such as: specification, shaping, implementation, verification and validation, as well as maintenance. The model is an idealized representation of software process that are carried out in desired mode and connect the main activities.

CASE tools (*Computer Aided Software Engineering*) are software packages that provide automatic support for specific activities within the software process. They are made in accordance with a specific software development methodology, are used for implementation, contain possibilities for overview of corresponding diagrams and are used for the production of appropriate documentation.

REFERENCES

- [1] IEEE Computer Society, Software Engineering Coordinating Committee, Guide to the Software Engineering Body of Knowledge, 2004, (www.swebok.org).
- [2] M. Dorfman and R.H. Thayer: Software Engineering, IEEE Computer Society Press, 2002.
- [3] M. Dorfman and R.H. Thayer, eds.: Software Engineering, IEEE Computer Society Press, 2002.
- [4] S.L. Pfleeger: Software Engineering: Theory and Practice, second ed., Prentice Hall, 2001.
- [5] Sommerville, Software Engineering, seventh ed., Addison-Wesley, 2005.
- [6] <http://www.it-akademija.com/ITAkademija-Software-Engineering-Program> 618 2 77 219.