# Quality Control Approach in Developing Software Projects

**Viktorija Stepanova**

Department of Computer Science and Information Technology, Riga Technical University, Riga, Latvia

viktorija.stepanova@rtu.lv

## ABSTRACT

Achievements in a unified digital market raise a topical question on provision of high quality software. Mistakes correction and the quality control have impact on the project. The proposed method provides understanding on the software progress condition with its strong and weak sides.

Keywords: *Process Assessment, Process-Orientation, Six Sigma, Software Quality Management, Software Engineering.*

## 1. INTRODUCTION

As contemporary world is based on flows of data and information, information systems in this contemporary digital age is omnipresent. They are adjusted to comply with the needs of different fields. Fact sheet issued by the European Commission indicates that 75% of Europeans believe that digitalization has a positive impact on economy and 44% of the employees think that their jobs can be at least partly implemented by a robot or artificial intelligence [1]. In the recent years, software developers have spent approximately 30-40% of their budget on software verification and validation. There are many different definitions of software quality. IEEE's definition of 1983 can be mentioned as an example: "Software quality is combined characteristics of software that can satisfy particular needs of the client and complies with the specification" [2]. A German industry standard features the following software definition: "Quality comprises all the action's characteristics and substantial features which are related to implementation of particular requirements" [3]. In a broader sense the definition of quality pertains to requirement satisfaction or, in other words, "suitability for the aim" (Fig.1). The system can be illustrated input–process–output (IPO) model [4] that consists of software requirement from potential end-user (input), a series of actions made by project team in order to meet the aim (process), and a software produced (output).

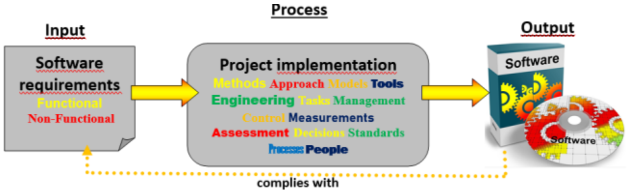

*Fig. 1. Software quality system*

Software quality can be divided into two groups:

- Outer quality that is assessed by the client or end user of the software
- Inner quality that pertains to the inner aspects of software that are assessed by the developer team.

Moreover, software quality is directly related to the software testing concept. Software testing is a quality control action with the aim of to assess software unit in accordance with the set system requirements. Whereas software quality control pertains to certain functional requirements and non-functional requirements. The aim of testing the functioning requirements is to check particular action or function code of the system. Testing of non-functional requirements pertains to software aspects that are not related to a particular function or user's action. For example, requirements for the product that defines such various characteristics of the software system to be developed as safety, high-speed functioning, transfer among many platforms, etc. Outer requirements are also included in the category of non-functional requirements and include within requirements for the system's interface with other systems, legislative requirements and requirements on data security.

Quality assessment must be included in every software development project. There are several ways how software testing can be included in the advancement of the project.

Implementation of testing at the end of the project could lead to reworking rather large phases of project by correcting mistakes, and the acquired experience will no longer have impact on the finished project, but it will positively affect future projects. Software quality control can be implemented in a particular project phase or continuous quality assessment methodology can be developed.

## 2. RELATED WORK

Many software life cycle models have been developed and they all include control or testing phase. Within the framework of this paper various software project's implementation models were reviewed and they are all summarized below:

Waterfall model [5] - This is one of the traditional types of software development in software engineering. It has 6 stages. In a waterfall model all the processes take place in a linear flow with detailed levels. Every progress stage begins only when the previous one is finished.

Agile methodology [6] - Agile development principles are based on adaptability of processes and clients' satisfaction. The software development project is implemented in small, successive stages. In the beginning of every step the plan of this stage and implementation sequence is set. The length of stages is usually set within margins of 1-2 months. The product is checked very often. The basis for agile methodology is interaction with the client and open communication with minimal documentation.

Prototype methodology [7] - Prototyping of the software allows understanding the client requirements at an early development stage. The software is developed gradually approaching the real version of the software with every iteration. When the prototype is finished, it is submitted to the client for assessment. The client carries out tests and gives their commentary to the developer, whether the given product complies with the client's requirements. After the final number of iterations the client receives the final version of software.

Rapid application development (RAD) [8]. This approach is directed to fast achievement of results. Models are directed to iterative and gradual delivery of working models to the client. Functional models are developed in parallel as prototypes and are integrated in a single product.

Spiral model [9] - Model combines the idea of iterative systematic progress with the waterfall model. It allows gradual release or gradual improvement of the product by using every repetition in a spiral. It comprises four stages, and in each stage the risks are assessed.

DevOps [10] is a body of models that automatizes the processes between the software development and IT teams to facilitate faster and safer development, testing and delivery of software. The aim of DevOps is to accelerate formation and installations of software build, as well as improving their quality.

In June and July of 2017 software developers SMARTBEAR [11] carried out a global online survey and collected 5219 answers in total. The research indicates that the majority (63% of 3406 respondents) choose Agile methodology for software development, testing and quality control. While the Waterfall model is used by 7%, but DevOps – only by 5%. 23% of developers acknowledged that the choice of methodology relies on the project.

## 3. MATERIALS AND METHODS

A literature review was implemented to get an overview of existing tools and solutions, but an analyze of individual components and it's synthesis was used to connect elements in one IPO approach in managing software quality.

Within framework of this paper, is offered to define software development processes based on IEEE standard oriented on reusable components to produce additional value for the project. All defined processes should be assessed by processes' performance assessment system that is based on Standard ISO/IEC "Process measurement framework for assessment of process capability" [12]. To understand if the project complies with the specifications six-sigma method [13] was used.

## 4. QUALITY CONTROL APPROACH

Software life cycle models reviewed in this work include control or testing phase, but there are no tools or methods clearly defined to meet a particular aim. Usually project team members choose a method that is familiar to them or that will be easy to use. Some methods and tools can be combined. Within this work the input–process–output model was used for describing the structure of software project implementation order. It is possible to isolate six stages of process management [14]:

Stage 1: defining processes
Stage 2: assessing processes
Stage 3: process performance analysis
Stage 4: defining new aims
Stage 5: process improvement planning: Process's improvements must be planned in coordination with organization's aims
Stage 6: process improvement implementation: The set process's improvements must be implemented for fulfilling the aim.

Similarly, as business process management, quality control approach assumes the execution of the same six stages that allows developing integrated system for documenting, analysing and improving software quality. Information entered for accomplishing a task impacts execution of processes and system's outputs. Taking this into account several standards, methods, measurements were defined for each system's stage. The result of a particular stage could serve as inputs to the next stage. At the same time, each stage could have one or more inputs and outputs. The table below shows IPO model developed for software quality control within this work (Table 1.).

*Table 1: IPO model developed for software quality control*

| ***Input*** | ***Stage of quality control system*** | ***Output*** |
|---|---|---|
| 1. Software requirements<br>2. IEEE Standard for Information Technology. "1517-2010 System and Software Life Cycle Processes. Reuse Processes" | Stage 1<br>Processes must be clearly set and documented | Set of documented processes |
| 1. Set of documented processes<br>2. Software development processes' performance assessment system based on Standard ISO/IEC. 33020 | Stage 2<br>Process's performance must be checked by using quantifiable metrics | Project performance profile |
| 1. Project performance profile<br>2. Six sigma methodology | Stage 3<br>Process performance must be analysed with the help of various tools (graphical images, diagrams, causal relationship analysis and others) | 1. Cp, CPk indicators<br>2. Software project histogram |
| 1. Cp, CPk indicators<br>2. Software project histogram | Stage 4<br>Process's stability must be analysed and new aims and tasks must be set, if necessary | Conclusion whether the project complies specification:<br>1. Software (project complies specification)<br>2. New aim definition (project doesn't comply specification) |
| 1. Weakness<br>2. New aim | Stage 5<br>Process's improvements must be planned in coordination with organization's aims | Improvement plan |
| 1. New aim<br>2. Improvement plan | Stage 6<br>The set process's improvements must be implemented for fulfilling the aim. | Software |

## 4.1. Defining processes

Within the context of the developed quality control system, the first inputs are software development processes and user requirements and IEEE information technology standard 1517TM – 2010 System and Software Life Cycle Processes. Reuse Processes" [15]. In this paper "software of repeated use" is viewed as one of quality factors. By performing systematic repeated use, one must concentrate on the engineering technical principle application for all the repeatedly used actives that are involved in software development. Notion "repeated use" allows applying existing actives, setting possibilities of repeated use, and preparing repeatedly used processes.

Repeated use active can be a software design, specification, source code, documentation, test package, manual procedures etc. that are developed for use in several contexts. The developer can achieve additional gains from repeated use of the existing software component if it is modular and easily modifiable. From the processed information, the system produces the first output - a set of documented processes, that should serve as the basis for next actions:

- Software implementation processes
- Software requirement analysis processes
- Software architectural design processes
- Software detailed design processes
- Software construction processes
- Software integration processes
- Software qualification testing processes

Each defined process has sub-processes or operations that are necessary to develop, operate and maintain software or software elements for repeated use.

## 4.2. Process Assessment

The set of documented processes is the input information for the next stage. To assess these processes process performance self-evaluation method was developed (table 1). It is based on Standard ISO/IEC 33020 "Information technology. Process assessment. Process measurement framework for assessment of process capability"[12]. This standard allows implementing software quality self-assessment, ensuring the basis for process improvement and process quality determination, determining software profile with process ratings (levels). For process development five process performance levels (PPL) were defined. For quantifiable metrics, every condition has a set value from 0 to 1 (Table 2).

*Table 2: Processes process performance self-evaluation method*

| ***PPL*** | ***Value*** | ***Description*** |
|---|---|---|
| 0 | 0 | Process is not defined |
| 1 | 0.1 | Process is defined |
| 2 | 0.2 | Process is defined and documented, but the action is not yet started |
| | 0.3 | Process is started |
| 3 | 0.4 | Process is implemented |
| | 0.5 | Process is implemented and the results are documented |
| 4 | 0.6 | Process is assessed |
| | 0.7 | Process is assessed and results are documented |
| 5 | 0.8 | Process optimization requirements are defined |
| | 0.9 | Process is optimized |
| | 1 | Process is optimized and documented |

Process assessment implementation produces the output of a process – the project performance profile with process average value and ratings (Fig. 2), where process value is an average of all operations performed in each defined process.

| Process PPL | Process 1 Software Launch process | Process 2 Software Requirement Analysis | Process 3 Software Architecture Design | Process 4 Detailed Software Design | Process 5 Software Structure | Process 6 Software Integration | Process 7 Software Qualification Testing |
|---|---|---|---|---|---|---|---|
| 0 | 0 | | | | | | |
| 1 | | | | | | | |
| 2 | | | | | | | |
| 3 | | | | | | | |
| 4 | 0,74 | 0,7 | 0,62 | 0,68 | 0,7 | 0,6 | 0,7 |
| 5 | | | | | | | |

*Fig. 2. Project performance profile*

Subheadings should be as the above heading "2.1 Subheadings". They should start at the left-hand margin on a separate line.

## 4.3. Process Assessment

The next stage of quality control approach is the performance analysis. It is a body of calculations that are used to assess whether the system processes comply with the specifications. There are many methods of quality assessment, but as soon as the manufacturing process becomes manageable, the question arises: "To what extent does the process behavior comply with the specifications?"

To resolve this task, six sigma methodology was selected. It is based on the approach of continuous processes' improvement and decrease the number of defects. For the process control restrictions are set and their breach is regarded as an undesirable event of the process. The farther the value of processes is from the average value and the closer it is to the control restriction, the smaller the number of defects. The smaller the standard deviation (σ sigma), the more stable and better is the result.

Delivering the software to the client the minimal processes' value should be 0.5 (LSL = the lowest specification margin), but the maximal – 1 (USL = the highest specification margin) (Fig.3a), when all the development processes are finished, optimized and documented. The outputs of this analysis are two quality indicators: Cp and Cpk and project histogram that provides a visual display of acquired data (Fig.3b).

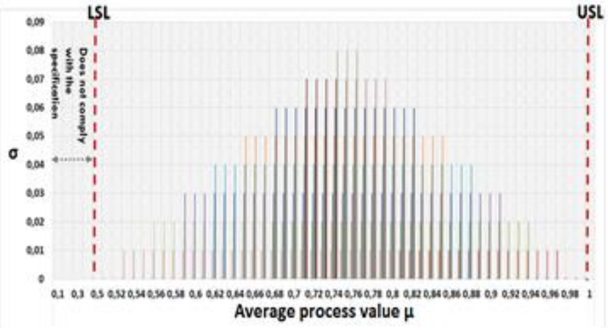

*Fig. 3a. Projects specification margin*

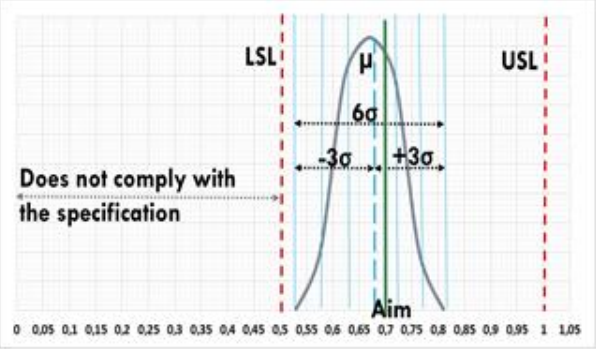

*Fig. 3b. Project performance histogram*

Cp index indicates the potential capacity of the project and shows how many times 6 sigmas can be contained in a range. To assess how far the average value of processes is from the specification Cpk indicator is used. Two formulas were used for calculations [16]:

$$Cp = (USL-LSL) / 6\sigma \quad (1)$$

$$Cpk= \min ((USL- \mu) /3\sigma; (\mu-LSL) /3\sigma) \quad (2)$$

## 4.4. Defining New Aims

During the implementation of the project its execution is monitored to determine whether the project is progressing towards the set aims. The acquired results (Cp and Cpk) indicate the appropriateness, suitableness and efficiency of each process in regard to the aims. Conclusion whether the project complies specification gives rise to the output of this stage. To satisfy software specification, the process' quality indicators should be to or larger than 1. The higher the Cpk value, the better the process complies with the requirements. If the results are far from the aims and the process does not comply with the specification, corrective measures should be implemented.

### 4.5. Improvement Planning

The project's weakness that should be transformed to a new aim is defined as the input to the improvement planning stage. The project manager must decide what direction the project will take, what specifically the software developers must do to achieve the set aims. The aims must include the result, and actions to achieve the aim. Project process improvement should be planned in coordination with new aims and should be recorded in improvement plan that is the output of stage.

### 4.6. Improvement Planning

Software developer team should perform the established process improvements for achieving new aim. These activities are critical for achieving the aim. The obtained outputs should increase the software quality to satisfy end-users needs.

## 5. CONCLUSIONS

For software quality control continuous quality assessment methodology was developed that is based on several standards and project management conceptions. Project manager of the quality assessment result achieves understanding on the software progress condition with its strong and weak sides. This understanding offers an opportunity to completely finish and optimize software development processes for successful commercialization and gaining profit from ICT introduction to market. The authors' further research will be devoted to the testing and evaluating this approach on several software projects.

## REFERENCES

[1] European Commission, "Digital Single Market: Commission calls for swift adoption of key proposals and maps out challenges ahead" [Online]. Available: https://ec.europa.eu/digital-single-market/en/news/digital-single-market-commission-calls-swift-adoption-key-proposals-and-maps-out-challenges [Accessed: May 30, 2017].

[2] German Institute for Standardization, "DIN EN ISO 9000:2015-11, Quality management systems - Fundamentals and vocabulary (ISO 9000:2015); German and English version EN ISO 9000:2015", German Institute for Standardization, 2015.

[3] ANSI/IEEE 729-1983, "IEEE Standard Glossary of Software Engineering Terminology", Standard [Online]. Available: https://standards.ieee.org/findstds/standard/729-1983.html [Accessed: October 5, 2017].

[4] T.T. Raa, "Input–output economics: theory and applications: featuring Asian economies", World Scientific, 2009.

[5] W. R. Winston, "Managing the Development of Large Software Systems", Technical Papers of Western Electronic Show and Convention (WesCon), August 25–28, Los Angeles, USA, 1970.

[6] K. Wortman, B. Duncan and E. Melin, "Agile methodology for spacecraft ground software development: A cultural shift", Aerospace Conference, IEEE, 2017, pp. 1-3.

[7] D.A. Fern and S.E. Donaldson, "Tri-Cycle: a prototype methodology for advanced software development", System Sciences, Vol. II: Software Track, Proceedings of the Twenty-Second Annual Hawaii International Conference, 1989, pp. 377-386.

[8] V. Deepa and R. R. Geetha, "Rapid development of applications in data mining", Green High Performance Computing (ICGHPC), IEEE International Conference,2013, pp. 1-4.

[9] W. Boehm, "A spiral model of software development and enhancement", Newsletter ACM SIGSOFT Software Engineering Notes Homepage archive, Volume 1, Issue 4, 1986, pp. 14-24.

[10] T. Laukkarinen, K.Kuusinen and T. Mikkonen, "DevOps in Regulated Software Development: Case Medical Devices", IEEE Conference Publications, IEEE/ACM 39th International Conference on Software Engineering: New Ideas and Emerging Technologies Result,2017, pp. 15-18

[11] "State of Testing Survey 2017 I Industry Report" [Online]. Available: https://smartbear.com/resources/ebooks/the-state-of-testing-2017-industry-report/ [Accessed: September 17, 2017].

[12] Standard ISO/IEC. 33020. "Information technology. Process assessment. Process measurement framework for assessment of process capability (2)", Switzerland: ISO/IEC, 2015.

[13] D.C. Montgomery, & W.H. Woodall (2008), "An Overview of Six Sigma. International Statistical Review", Journal International Statistical Review 76 (3),2008, pp. 329-346.

[14] "Project Management Concepts. Process Based Management" [Online]. Available: https://www.tutorialspoint.com/management_concepts/process_based_management.htm [Accessed: May 30, 2017].

[15] IEEE Standard for Information Technology. "1517-2010 System and Software Life Cycle Processes. Reuse Processes", USA: IEEE, 2010.

[16] Y.Wooluru, D.R. Swamy and P.Nagesh, "The Process Capability Analysis – A Tool for Process Performance Measures and Metrics - A Case Study", International Journal for Quality Research 8(3), 2014, pp. 399-416.