

# **Problems Review, High-Level Architecture, Identification Stakeholders and Their Requirements During the Implementation of Quality Control in the Case of Software Development**

Michail V. Druzhinin<sup>1</sup>, Alexander N. Kulemin<sup>2</sup>

<sup>1</sup> Ural Federal University, Yekaterinburg, Russia;

<sup>2</sup> ABC Solutions, Moscow, Russia

michaildruzhinin@gmail.com

Nowadays current software solutions happen on a short notice and with limited resources. The importance of software quality assurance stems from the pace of IT-companies development and their interest in the product's competitiveness. In that case developers may ignore the necessity of providing a certain level of software solutions exposing the project to risks.

It requires elaboration of key indicators which define the ability to work well and produce quality products. Key indicators characterize the dynamic of project development in measurable units used for planning of project during the entire life cycle of software. Indicators directly connected to mistakes found in the program – bugs. Detection and elimination of those defects in early stages of software development would cost the company much less than in later operational phases[1].

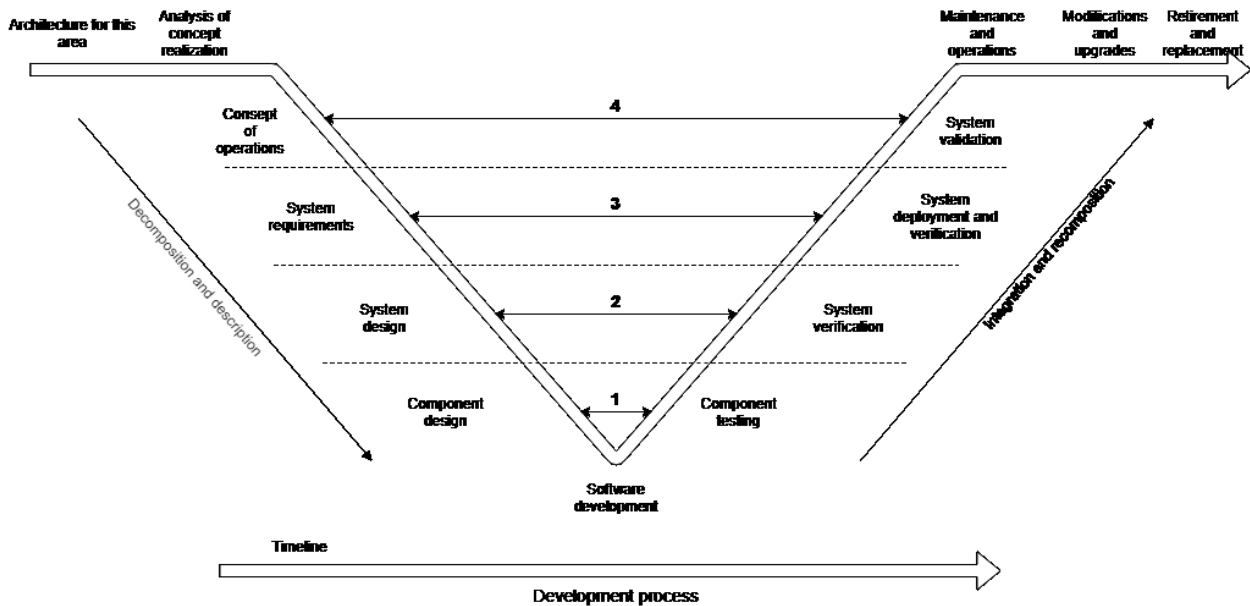
Testing allows aggregate information about mistakes that are already found in the process of development. Timely report about a defect to project managers, developers and analytics decreases the odds of repeating those mistakes which positively influences software quality and also provides with ability to control the course of development by redistributing resources in time.

Testing system is a set of regulated measures in the end of which there is an analysis of obtained results[2]. Set of measures consists of:

- Planning of testing;
- Analysis of testing results;
- Tests verification;
- Report verification; etc.

There are manual testing, via test engineer, and auto testing. Auto tests partly exclude time wasted on their replication and inclusion of human factor in testing process. Companies benefit from using auto tests, if the required tests are repeated and of the same type, because in that case there will be released resources for redistribution within the project.

In work itself object of the research is a software life cycle. Research goal is application of systematic approach to system design of quality control at the stage of determination of stakeholders.

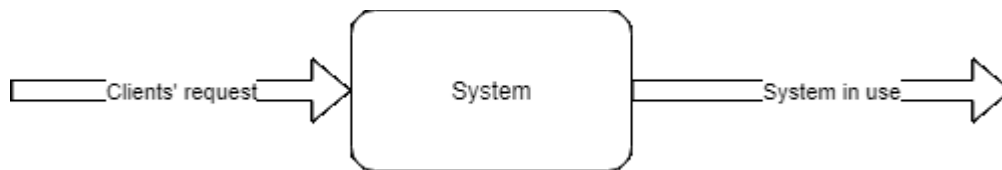


**Figure 1 - V–diagram of a life cycle[3]**

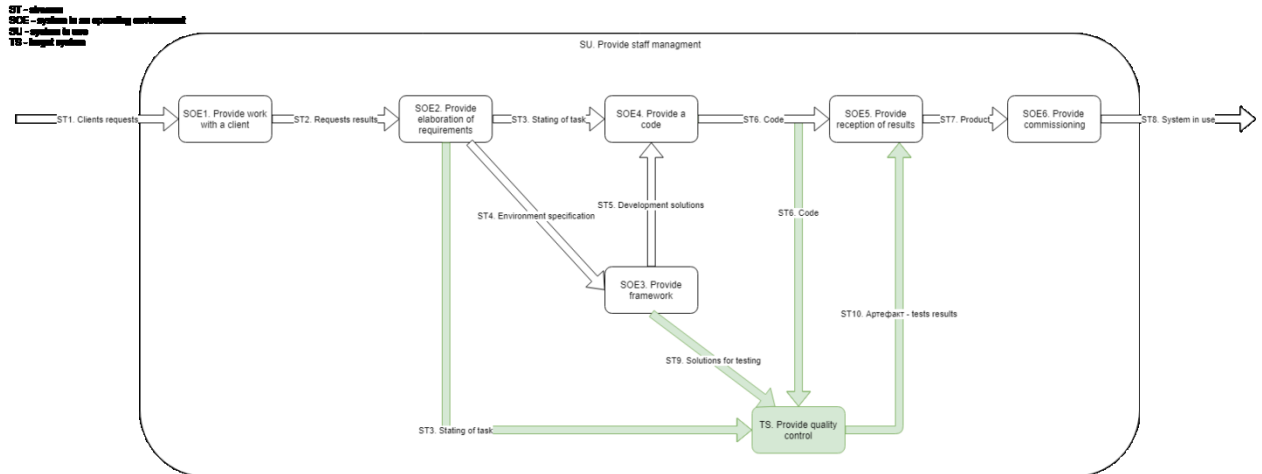
Software lifecycle presented in V–diagram implies four levels of verification:

1. Program and methodology of testing units
2. Subsystem verification plan
3. System verification plan
4. System validation plan

While designing quality control system (QCS) level one is taken into account. For identification of stakeholders on the testing level we'll consider QCS as a “black box” in the operational context.



**Figure 2 - “Black” box**



**Figure 3 - Operational context QCS**

Systems exchange with QCS streams will affect quality control process, which means in SOE there are interested parties. Table 1 shows stakeholders and their roles[4].

**Table 1. Target system stakeholder**

ID	Stakeholder	Business role
SH1	System customer	Business owner
SH2	Employer and project investor	Head of an IT-company
SH3	System user	System user
SH4	Analyst	Representative of analytics department
SH5	DevOps engineer	Representative of infrastructural solutions department
SH6	Software developer	Representative of software development department

**Table 2. System stakeholders' requirements.**

ID	Stakeholder	Requirement	Problem
P1	Customer of ZOZO system	Staff management	Resources distribution
P2	Project investor	Зарабатывать в нише систем WFM	—
P3	System user	Planning of working hours	Stating personal preferences
P4	Analyst	Verification of technical project	Verification of tasks given///
P5	DevOps engineer	Provide constant system deployment (про DevOps CI/CD)	Instability of assembly
P6	Software developer	Understand how to reproduce a mistake	Missing the mistakes

Quality defines the ability to satisfy users' needs within project limitations.

Function of conducting quality control in software development is carried out by testers. To fixate a certain level of quality there are different metrics such as: ratio of successful tests to all of the tests. Metrics allow assessing product's quality, its pace of development and adjusting the course of development.

Quality assurance – is a set of measures for the entire software development lifecycle including measures for achieving a certain level of quality. Starting from the step of determining stakeholders' requirements, where QA engineers validate if realization of the requirements is possible in the context of the current state of the system. Based on that, a decision can be made about advisability of system modification. To make this decision it is necessary to analyze the system ability to change and predict options of product quality development. This leads to changing tests scripts and update of testing documentation.

Requirements of different stakeholders need further design of model-oriented business-analysis which includes defining correlation between business-function and business-functional connection, defining business-structures and business-interfaces for their detailing and defining goals of every interested party. During specification of requirements new stakeholders may appear that were not there during “black box” approach. Establishing goals of an interested party improves understanding while designing the system, which directly affects product quality. This approach helps define main and secondary goals which give more flexibility while making decisions on distributing resources.

#### **References:**

1. ISO/IEC 9126-1:2001 Software engineering – Product quality – Part 1: Quality model.
2. ISO 9000:2015 Quality management systems — Fundamentals and vocabulary – Part 3: Terms and definitions.
3. Aleksandr Kosyakov, Uilyam N. Svit, Semyuel Dzh. Seymur, Stiven M. Bimer (2014) Systems Engineering: Principles and Practice. Moscow: DMK Press.
4. Stakeholder Needs and Requirements. (2015, June 29). *SEBoK*, . Retrieved 11:20, June 6, 2018  
from [http://www.sebokwiki.org/w/index.php?title=Stakeholder\\_Needs\\_and\\_Requirements&oldid=51430](http://www.sebokwiki.org/w/index.php?title=Stakeholder_Needs_and_Requirements&oldid=51430).