# SOFTWARE QALITY ASSURANCE (*ASIGURAREA CALITĂȚII PRODUSELOR SOFT*)

**SECU ANDREI, masterand, MI-191m**
Academia de Studii Economice din Moldova,
R. Moldova, Chișinău, str. Bănulescu-Bodoni 61, MD-2005,
e-mail : andrei.secu1997@gmail.com

***Abstract.*** *Calitatea unui produs este factorul care îi determină popularitate acestuia. Conceptul de asigurare a calității apare în anul 1950, în SUA, iar mai apoi datorită economiei concurențiale s-a extins rapid în toată lumea. Astăzi, acest aspect se bucură de o atenție sporită, fiind un proces pe larg apreciat și constant în creștere în toate ramurile industriale, asta datorită încrederii pe care consumatorii o atribuie brand-urilor ce asigură implementarea și monitorizarea cu succes a tuturor indicilor de calitate. Un impact sporit al asigurării calității este sesizat în cadrul Serviciile IT, care dețin o cotă impunătoare la nivelul pieței globale. Abordând metode de analiză și comparare, acest articol vine să prezinte conexiunea dintre asigurarea calității și livrarea produselor IT(soft).*
***Key words:*** *Calitate, Asigurarea calității, Produse soft, Standarde, Indici de calitate.*

**JEL CLASSIFICATION: O32 - Management of Technological Innovation and R&D**

## INTRODUCTION

The competitive economy is growing quickly. Having a look over all the daily activities can be noted that the IT services are involved in most of our tendencies. As we all know, human tendencies are priorities for business which nowadays is almost digital. A fault cost it's not something that can be estimated while it is not reproduced, in some cases, when we speak about human lives, money may not be enough resource to cover the result of a fault made by the business. Therefore, this is the reason why most companies are spending a huge amount of money to improve their digital services. According to spiceworks [1] portal, 44% of businesses plans aim to increase tech spenditure in 2020. More than 50% percentage of the tech business investments are oriented to align their activities with ISO 9001 standard – the international standard for Quality Management Systems.

## CONTENT

As it has been mentioned above, quality of the services is a key priority for business, thus there have been developed numerous approaches and methodologies designed for software development, to ensure high-quality products which meet the client needs. Some of the most popular software development life cycles are:

- Waterfall Model – the oldest and most straightforward of the structured SDLC methodologies, finish one phase, then move on to the next. No going back is allowed. Each stage relies on information from the previous stage and has its own project plan. Waterfall is easy to understand and simple to manage.
- V-Shaped Model - known as the Verification and Validation model, the V-shaped model grew out of Waterfall and is characterized by a corresponding testing phase for each development stage. Like Waterfall, each stage begins only after the previous one has ended.
- Iterative Model - is repetition incarnate. Instead of starting with fully known requirements, you implement a set of software requirements, then test, evaluate and pinpoint further requirements. A new version of the software is produced with each phase, or iteration. Rinse and repeat until the complete system is ready.
- Spiral Model - one of the most flexible SDLC methodologies, the Spiral model takes a cue from the Iterative model and its repetition; the project passes through four

phases over and over in a "spiral" until completed, allowing for multiple rounds of refinement.

- Agile Model - quickly delivers a working product and is considered a very realistic development approach. The model produces ongoing releases, each with small, incremental changes from the previous release. At each iteration, the product is tested.

By doing a exploratory analyze of the delivered IT produscts, we can constat that quality assurance has a consistent presence within all software development life cycles. The most effetive approach in providing quality is considered the agile one. According to Dr. Dvir Zohar, about 67% of IT projects are developed with the help of Agile methodology related to iterative software development model. The root of this statistic data is the way the errors are removed from the newly building software product. By adapting to Agile, the team is delivering small units of a functional application which go through an intense process of quality assurance that validates both functional and non-functional behavior of the newly delivered features. According to Agile Testing book [2] written by Lisa Crisn and Janet Gregory, till the product goes live, the quality assurance management will ensure the following non-functional testing types are performed:

- Security - done to ensure that the application has no loopholes which could lead to any data loss or threats. It is one of the important aspects of non-functional testing and if not performed properly, it can lead to security threats.
- Performance - evaluates the overall performance of the system.
- Load - evaluates whether the system's performance is as expected under normal and expected conditions.
- Stress - evaluates whether the system's performance is as expected when it is low on resources.
- Volume - evaluates the behavior of the software when a large volume of data is involved.
- Usability - evaluates the system for human use or checks if it is fit for use.
- Recovery - evaluates that the application terminates gracefully in case of any failure and the data is recovered appropriately from any hardware and software failures.
- Compatibility - Evaluates that the application is compatible with other hardware /software with minimum and maximum configuration.
- Scalability - done to verify if the application is capable enough to handle increased traffic, number of transactions, data volume, etc. The system should work as expected when the volume of data or change in the size of data is done.
- Reliability - done to verify if the application is reliable and is tested for a specific period of time in the defined environment. An application should give the same output as expected every time, only then it can be considered as reliable
- Portability - done to verify if in case a software/application is installed on a different system or on a different platform it should be able to run as expected i.e. no functionality should be affected because of a change in the environment.

A high-quality level of the above-described parameters will reduce production risk and cost associated with non-functional aspects of the product. On the other side, there is a functional testing which validates that the system behaves as intended. Functional validations are done on different application level in order to ensure that each component is fully operational and do not occur side effects. Referring to International Software Testing Qualifications Board[3], most common functional test levels used by the quality assurance are pointed out within the figure 1.

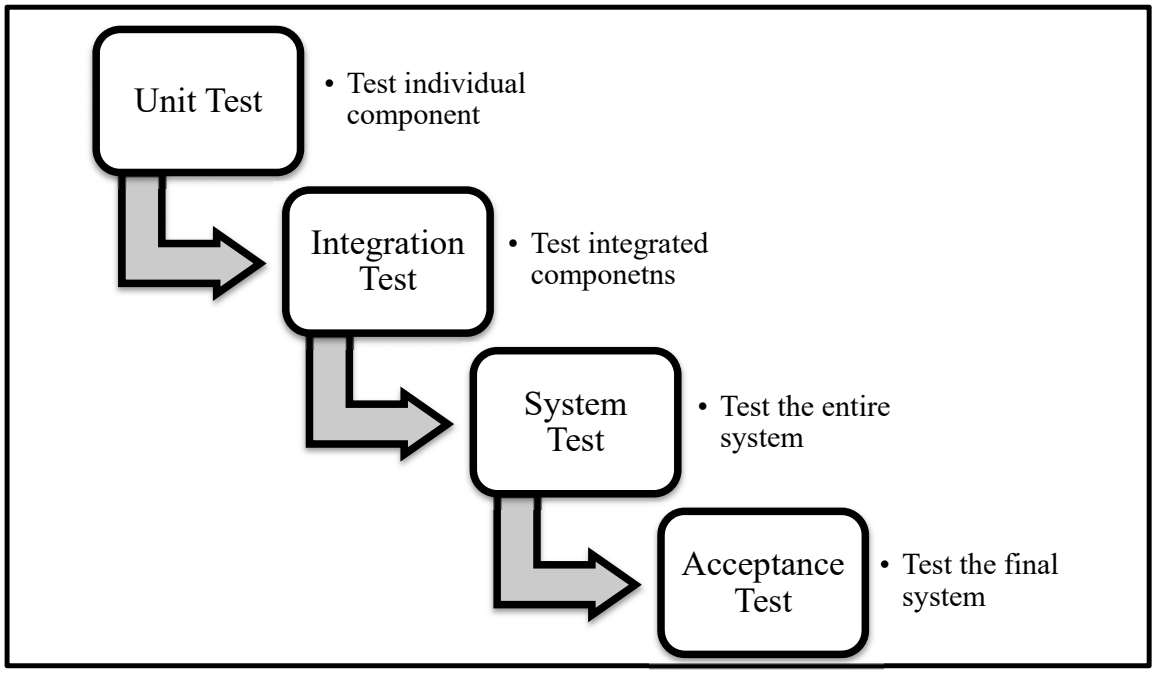


Figure 1: Software testing levels

- Unit test - smallest testable portion of system or application which can be compiled, liked, loaded, and executed. This kind of testing helps to test each module separately. The aim od unit testing is to test each part of the software by separating it. It checks that components are fulfilling functionalities or not. This kind of testing is performed by developers.
- Integration test - different software modules are combined and tested as a group to make sure that integrated system is ready for system testing. Integrating testing checks the data flow from one module to other modules. This kind of testing is performed by testers.
- System test - is performed on a complete, integrated system. It allows checking system's compliance as per the requirements. It tests the overall interaction of components. System testing most often the final test to verify that the system meets the specification.
- Acceptance test- is conducted to find if the requirements of a specification or contract are met as per its delivery. Acceptance testing is basically done by the user or customer. However, other stockholders can be involved in this process.

Going through all of the mentioned above levels the testing team should find out the most important defect/error that could become a production disaster. Each found defect is reported by providing the following details:

- Defect name
- Environment
- Description
- Precondition
- Steps to reproduce
- Attachments
- Priority
- Severity

After all the details are defined, the issues are going to be fixed by developers within the same realise based on the priority and the severity of the defects, which shows the importance and the impact of the issue.

In order to deliver a successful project, the quality assurance team have to cover the main functions of quality standards:

- Technology transfer – it includes product design document, as well as trial and error data and its evaluation.
- Validation is when the application meats the exit criteria.
- Documentation is characterized through distribution and archiving of documents.
- Assuring Quality of products
- Quality improvement plans

As a result of correct action execution, an organization can assure that the processes are efficient and effective in terms of quality standards defined for software products. This research shows that the common formula Plan-Do-Check-Act is effective event for software quality assurance processes. By planning correct testing actions, selecting a correct development environment and managing the above specified testing activities will bring success for the newly created software product.

## CONCLUSION

Quality Assurance is to check whether the product developed is fit for use and it's not going to cause additional costs for business. For that, an organization should have processes and standards to be followed, which need to be improved on a periodic basis by adapting to Plan-Do-Check-Act formula. To sum up, it concentrates mainly on the quality of product/service that the business is providing to the customers during or after implementation of the software.

### REFERENCES

1. https://www.spiceworks.com/ : Statistic data related to technology management;
2. Agile Testing, Lisa Crispin, Janet Gregory
3. https://www.istqb.org/ : International Software Testing Qualifications Board

**Conducător ştiinţific: ANDRONATIEV VICTOR, dr., conf. univ.,**
**Academia de Studii Economice din Moldova,**
**str. Bănulescu-Bodoni 61, MD-2005, Chişinău, R. Moldova,**
**e-mail: androvic@ase.md**