Software Quality Control in Agile: Goals and Business Values
Mind the Quality

When it comes to software products, users mostly value ease-of-use and achieving their goals at little to no cost. IT professionals value software which is easy to maintain, with simple extend and porting functionality – paying more attention to internal software characteristics. Business domain experts will give their vote to simplicity and accuracy of solving business problems, as well as the software product’s stability and reliability. All of these things and many other characteristics make what is known as “software quality.”

With agile methodology becoming increasingly popular and commonly implemented, the role of Quality Control processes has been modified. Apart from targeting quality characteristics artifacts produced by the software development process, Quality Control is now responsible for ensuring that agile development processes are solid and fast.

Software Quality Control: Aspects

The process of Quality Control and the way it fits the software development process to a great extent defines the quality of the end product and can often affect the production schedule, budget, and the scope of the delivered functionality, thus possibly affecting the client business opportunities. As you can see, using Quality Control to its fullest can make all the difference for your product.

For the maximum positive effect consider providing your Quality Control team with the following objectives:

- Focus on Software Product Quality Characteristics
  
  Quality Control encompasses Testing, Reviews of software work products and Quality Prevision procedures to identify if important quality characteristics are present.

Software Testing deals with the facts and establishes the extent of correspondence between the requirements set and the software product’s actual characteristics at a given time. Testing is usually the primary and most common method to verify software quality requirements. Proper identification of test levels and testing types is the basis for creating a testing strategy to be followed on a project. Reviews of software work products include project documentation and software code base. It is a formal or an informal static technique that gives prompt feedback on an artifact’s quality and initiates the quality improvements. Automated static reviews are run against codebase to check the coding convention conformance, technology usage verification, intended software architectural and design rules. As to the Quality Prevision, it encompasses
a set of prediction techniques, which are commonly based on the mathematical analysis of the software product’s code related data, as well as statistical analysis of the defects quantity, arrival, lifecycle, etc. The primary goal of Quality Prevision is to project the current quality level on up-coming changes.

▪ **Make sure quality is built into the development cycle**

As agile methodology promotes frequent and early software product releases in a short timeframe, there is no place for a classical way of identifying software defects and consequently correcting them. Thus, preventing defects is inevitable in agile development. By default, every developed artifact is considered to be faulty until the opposite is proven, therefore it is always tested as a result. It might take a long time to fix an identified defect and then to ensure it has no negative effect on the product so the measures to prevent defects from reoccurring are significant for agile. Make sure your Quality Control team implements Test Driven Development (TDD), Acceptance Test Driven Development (ATTD), Behavior Driven Development (BDD), continuous integration, code quality analysis, and peer reviews practices. This way it is possible to minimize the risk of defects occurring late in the lifecycle of an agile iteration, resulting in a considerable decrease in rework expenditures, which are otherwise commonly necessary.

▪ **Target your business needs: Quality Control as an instrument for the Product Owner**

To ensure a positive user experience and the product’s functionality at the same time, the product owner, developer and tester should be on the same page for the expected results. Performing the additional analysis of the requirements validity, as well as change impact analysis, Quality Control fills the role of the negotiator and proxy between the product owner and the development team and helps achieve this important goal. Since Quality Control individuals possess programming and technical background and skills, they can easily facilitate communication between the programmers developing the software and the customers identifying the product needs. Usually this is done with the help of the User Story Acceptance Criteria clarification by asking the right questions on positive, negative and edge cases of the user story definition and applying previous experience in the business domain. Each and every Acceptance Criteria is being covered by tests that are integrated into the entire product test suite and provide the Product Owner and Agile Team with an unbiased judgment on the quality of the User Story implementation.

The main Quality Control function in this respect is to help the Product Owner and Agile team synch up on the scope of work to be performed within an iteration and provides the Product Owner with tangible measures to make a decision whether a work item can be considered completed or not.

▪ **Responding to changes frequently and efficiently**

Agile methodology at its core is a proactive approach to product development and promotes a flexible response to changes. Even though Quality Control is reactive and focuses on finding existing defects (any nonconformance of software deliverables, such as programming errors; function faults; misconfiguration issues; users’ documentation defects, etc.), in agile it is specifically targeted at ensuring quick reaction to any changes.

In agile methodologies the product is being developed iteratively – new functionality, as well as other quality attributes (stability, performance, usability, etc) stratify on current and existing quality. It means that new changes are introduced often in the iteration – continuously, up to 24 hours per day in geographically distributed agile teams. From a quality perspective, every change is considered potentially harmful to a
product under development. To identify if a change is indeed harmful, it is necessary to have it thoroughly tested alongside all impacted areas (very often the whole product). Many tests have to be done frequently and quickly. Improving the scope and time of testing can be done by means of automation and early testing. The Quality Control function in this respect is to establish, maintain and own the test automation processes and the testing culture within the project, so that the development team can ensure a timely response on code changes they perform.

Along with Business Analysis, Business Requirements Management, Project Management, Release Management, Programming Disciplines, etc., Quality Control is focused on performing a large and vital task: to achieve and maintain product success. As a result, underestimating the input of Quality Control in the end customer’s satisfaction may be quite dangerous and destructive to the company’s business advancement.

**SoftServe’s Quality Control Consulting Service**

Having years of experience in successful Agile QC process implementation, SoftServe offers consulting in quality control practices including test automation. Recognizing the importance of Software Quality Control for the product’s and company’s success, SoftServe created a special service package to help its clients assess and improve the quality control for both ongoing projects and those being developed. SoftServe helps companies define the most important goals and values for the product in question, and creates the best viable strategy on how to achieve the established business objectives with the help of the most suitable Quality Control practices.

SoftServe consulting in Quality Control consists of the following stages:

1. Setting goals
2. Analyzing capabilities
3. Designing solutions
4. Delivering solutions

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Activities:</strong>&lt;br&gt;• Questionnaires&lt;br&gt;• Interview with QC stakeholders&lt;br&gt;• Quality Control goals definition and prioritization&lt;br&gt;• Documentation analysis&lt;br&gt;• A quick expert review, selection of quality control assessment areas</td>
<td><strong>Activities:</strong>&lt;br&gt;• Gap analysis&lt;br&gt;• Discover areas for improvement and prioritize them according to the product requirements and the impact on the end-user&lt;br&gt;• Define the benefits from introducing improvements into the project</td>
<td><strong>Activities:</strong>&lt;br&gt;• Define the solution for assessment scope findings&lt;br&gt;• Assess the recommendations implementation risks&lt;br&gt;• Design the solution</td>
<td><strong>Activities:</strong>&lt;br&gt;• Evaluate Benefits, Opportunities, Costs and Risks of the proposed QC Solution&lt;br&gt;• Present the Quality Control Assessment Report</td>
</tr>
</tbody>
</table>
Selecting the correct set of practices and their effective implementation, result in a higher level of product quality and contributes to the organization’s success.

**Software Quality Control: Values**

SoftServe Quality Control Consulting Service provides the following business values:

- The quality control process becomes predictable
- The quality control team’s skills are effectively applied
- Defects are successfully identified on the basis of the product specification and the worldwide quality standards
- Faster testing with test execution time reduced
- The tests’ quality improves and the product test coverage extends
- The customer’s specified issues are covered first (prioritization)

Integrating a solid quality control methodology into the process of software development is the best practice in every situation. SoftServe’s experience has proven that with the right Quality Control practices, you can increase your software quality and lower the total cost of ownership. Customers benefit from mitigating the risks related to misunderstanding or miscommunication of business requirements, possible system failures, and others. We incorporate Quality Control principles into every project to ensure that we develop work flawlessly and satisfy every functional, technical, and architectural requirement. Whether you’re a startup business, an enterprise, or a product development group, quality control is essential to your long-term success.