

Quality Assurance of Embedded System

Contents

1	Introduction of quality assurance	2
2	Quality assurance in software	2
3	QA team	4
4	Goals in each stage of QA	4
5	Requirement analysis	5
6	Software specifications	5
7	Design	5
8	Coding	6
9	Test	6
10	Maintenance	6
11	SQA tools	8
12	Advantage and disadvantage of QA	8

1 Introduction of quality assurance

Quality assurance (QA) is any systematic process of determining whether a product or service meets specified requirements. QA establishes and maintains set requirements for developing or manufacturing reliable products.

QA establishes and maintains set requirements for developing or manufacturing reliable products. A quality assurance system is meant to increase customer confidence and a company's credibility, while also improving work processes and efficiency, and it enables a group or a company to better compete with others.

In terms of software development, QA practices seek to prevent malfunctioning code or products.

2 Quality assurance in software

Software quality assurance (SQA) systematically finds patterns and the actions needed to improve development cycles. Finding and fixing coding errors can carry unintended consequences; it is possible to fix one thing, yet break other features and functionality at the same time.

SQA has become important for developers as a means of avoiding errors before they occur, saving development time and expenses. Even with SQA processes in place, an update to software can break other features and cause defects – commonly known as bugs.

There have been numerous SQA strategies. For example, Capability Maturity Model Integration (CMMI) is a performance improvement-focused SQA model.

CMMI works by ranking maturity levels of areas within an organization, and it identifies optimizations that can be used for improvement. Rank levels range from being disorganized to being fully optimal.

Software development methodologies have developed over time that rely on SQA, such as Waterfall, Agile and Scrum. Each development process seeks to optimize work efficiency.

- **Waterfall** is the traditional linear approach to software development. It's a step-by-step process that typically involves gathering requirements, formalizing a design, implementing code, code testing and remediation and release. It is often seen as too slow, which is why alternative development methods were constructed.
- **Agile** is a team-oriented software development methodology where each step in the work process is approached as a sprint. Agile software development is highly adaptive, but it is less predictive because the scope of the project can easily change.
- **Scrum** is a combination of both processes where developers are split into teams to handle specific tasks, and each task is separated into multiple sprints.

To implement a QA system, first set goals for the standard. Consider the advantages and tradeoffs of each approach, such as maximizing efficacy, reducing cost or minimizing errors. Management must be willing to implement process changes and to work together to support the QA system and establish standards for quality.

3 QA team

A portion of careers in SQA include job options like SQA engineers, SQA Analyst and SQA test automation. SQA engineers monitor and test software through development. An SQA Analyst will monitor the implication and practices of SQA over software development cycles. SQA test automation requires the individual to create programs to automate the SQA process.

These programs compare predicted outcomes with actual outcomes. This work is used for constant testing.

4 Goals in each stage of QA

The goals of SQA in various stages of software development:

1. Requirement analysis
2. Software specifications
3. Design
4. Coding
5. Test
6. Maintenance

5 Requirement analysis

- Ensure that the system requested by the customer is feasible.
- Ensure that the customer's specified needs can indeed meet their true requirements.
- Avoiding misunderstandings between developers and customers.
- Provide users with appropriate software systems actually built to meet their stated needs.
- Ensure that the system requirements are clear, complete, and consistent, including functional, performance, and safety requirements.

6 Software specifications

- By establishing a requirement tracking document, ensure that the specifications are consistent with the system requirements.
- Ensure that the specifications appropriately improve the flexibility, maintainability, and performance of the system.
- Ensure that a testing strategy has been established.
- Ensure that a realistic development schedule has been established, including scheduled reviews.
- Ensure that a formal change procedure has been designed for the system.

7 Design

- Ensure that standards have been established to describe the design and that they are followed.

- Ensure appropriate control and documentation of changes made to the design.
- Ensure that coding begins only after the system design components have been approved according to agreed guidelines.
- Ensure that the review of the design is carried out according to schedule.

8 Coding

- Ensure that the code follows established style, structure, and documentation standards.
- Ensure that the code is properly tested and integrated, and that modifications to the coding module are appropriately identified.
- Check if the code writing follows the established schedule.
- Ensure that the code review is progressing according to schedule.

9 Test

- Ensure the establishment and adherence of testing plans.
- Ensure that the created test plan meets the requirements of all system specifications.
- Ensure that the software remains consistent with the specifications after testing and rework.

10 Maintenance

- Ensure consistency between code and documentation.

- Ensure monitoring of established change control processes, including the process of integrating changes into the product version of the software.
- Ensure that code modifications follow coding standards and are reviewed without damaging the entire code structure.

11 SQA tools

Software testing is an integral part of software quality assurance. Testing saves time, effort and cost, and it enables a quality end product to be optimally produced. There are numerous software tools and platforms that developers can employ to automate and orchestrate testing in order to facilitate SQA goals.

- **Selenium:** Selenium is an open source software testing program that can run tests in a variety of popular software languages, such as C#, Java and Python.
- **Jenkins:** Another open source program, called Jenkins, enables developers and QA staff to run and test code in real time. It's well-suited for a fast-paced environment because it automates tasks related to the building and testing of software.
- **Postman:** For web apps or application program interfaces (APIs), Postman will automate and run tests. It is available for Mac, Windows and Linux, and it can support Swagger and RAML formatting.

12 Advantage and disadvantage of QA

- **Pros:** Quality assurance helps ensure that organizations create and ship products that are clear of defects and meet the needs and expectations of customers. High-quality products result in satisfied customers, which can result in customer loyalty, repeat purchases, upsell and advocacy.

Quality assurance can lead to cost reductions stemming from the prevention of product defects. If a product is shipped to customers and a defect is discovered, an organization incurs cost in customer support, such as re-

ceiving the defect report and troubleshooting. It also acquires the cost in addressing the defect, such as service or engineering hours to correct it, testing to validate the fix and cost to ship the updated product to the market.

- **Cons:** QA does require a substantial investment in people and process. People must define a process workflow and oversee its implementation by members of a QA team. This can be a time-consuming process that impacts the delivery date of products. With few exceptions, the disadvantage of QA is more a requirement – a necessary step that must be undertaken to ship a quality product. Without QA, more serious disadvantages arise, such as product bugs and the market’s dissatisfaction or rejection of the product.