# Basic Concepts of Software Quality Assurance

**Software quality** is the degree to which a software product meets established requirements; however, quality depends upon the degree to which established requirements accurately represent stakeholder needs, wants, and expectations.

**Software quality assurance** is a set of activities that define and assess the adequacy of software processes to provide evidence that establishes confidence that the software processes are appropriate for and produce software products of suitable quality for their intended purposes. The SQA function may also be organizationally independent of the project, that is, free from technical, managerial, and financial pressures from the project. This definition may be characterized by the following:

* SQA is based on the planning and implementation of a series of activities that are integrated into all stages of the software development process.
* SQA refers to the software development products keeping the specified technical requirements and suitability for stakeholder’s intended use.
* SQA refers to the technical appropriateness of the development process.

An extended SQA definition was created considering the importance of the quality of the software operation and the important effect of schedule and budget keeping on the software quality product.

# The Principles of Software Quality Assurance (SQA

* **Customer Focus:** Organizations depend on their customers, and thus need to understand their current and future needs, fulfill their requirements, and achieve their satisfaction. Customer focus requires obsessive knowledge of what your customers need, and how to deliver. One good example is the Sephora. In 2016, the beauty brand uses technology to create a personalized experience with a comprehensive app, virtual try-on of makeup products and a strong online community for a seamless customer experience. Sephora recognized they needed to be more customer-focused.
* **Leadership:** An organization’s leaders should create an internal environment in which employees are involved in

achieving quality targets. Mark Zuckerberg is one of the most successful leaders in the world. One of his keys to success? Passion. Zuckerberg believes in hiring for passion over skill. This passion leaks into their motivation in other aspects of their work.

* **Involvement of People-employees:** The involvement of employees at all levels enable benefiting from their capabilities to promote software quality issues. Enterprises have now realized that they are nothing without their employees. This is why in the past few decades, entire departments and teams have been set-up to ensure employee involvement and participation in the company. Take Human Resources, for example. From being a purely transactional entity focused solely on administration, HR has now taken on a more transformational role.
* **Process Approach:** Managing activities and resources as processes results in their improved efficiency. The process approach includes establishing the organization’s processes to operate as an integrated and complete system. If you have a good process, you’ll have good results. Think about teaching writing - according to Graham Stanley, the process approach treats all writing as a creative act which requires time and positive feedback to be done well.
* **System Approach to Management:** Process management achieves higher effectiveness and efficiency through identification, analysis, and understanding of interrelated processes. An example is a system-oriented manager would only make decisions after they have identified the impact of these decisions on all other departments (such as the HR department, finance department, etc.) and the entire organization.
* **Continual Improvement:** Continual combined improvement of quality and processes’ effectiveness and efficiency performance are a permanent objective of the organization. It is also called continuous improvement which is an ongoing effort to improve products, services, or processes. For instance, Brainstorming sessions or think tanks open the door to continuous workplace improvements by bringing members of management together to resolve problems. Annual brainstorming sessions can consistently improve business operations.
  + **Factual Approach of Decision-making:** Decisions should be based on data and information. This allows for objective decision making that will lead to positive actions. Take for example, hiring a new employee to the company – HR and heads should have enough information about the candidate and decide whether it would be helpful to the company.
  + **Mutually Beneficial Supplier Relationships:** Understanding that an organization’s supplier relationships based on mutual benefits contributes to improved performance of the organization with regard to quality, efficiency, and effectiveness. Today, businesses are very interdependent on each other, their suppliers, and their customers. Decisions should not be made in isolation, as they can have an impact on others in the marketplace and may have a harmful effect on the organization.
    - **Data Necessary for Operating the Software System (Data):** The required data include lists of codes and parameters, and also standard test data. The purpose of the standard test data is to ascertain that no undesirable changes in the code or software data have occurred during bug corrections and other software maintenance activities and to support the detection of causes for any malfunctioning.

To summarize, it should be noted that software quality assurance refers to the quality of all components of the software product, namely the code, procedures, documentation, and the necessary operating data. Moreover, the composition of software product components varies significantly according to software development tools and methodology.

# Software Products

**Software Errors, Faults, and Failures .**

|  |  |
| --- | --- |
| **Software Errors** | Made by a software designer or programmer. An error may refer to a grammatical error in one or more of the code lines, or a logical error in carrying out one or more of the specification requirements. It should be emphasized that all errors are human  errors. |
| **Software Faults/ Software Defect** | These are software errors that cause improper functioning of the software in a specific application, and in rare cases, of the software in general. However, not all software errors become software faults. In many other cases, erroneous code lines will not affect the functionality of the software  (software faults are not caused). |
| **Software Failures** | These are results of software faults. These become software failures only when they are “activated,” that is when a user tries to apply the specific software section that is faulty. Thus, the  root of any software failure is a software error. It disrupts the use of the software. |

When we think about software, we imagine an accumulation of programming language instructions and statements, usually referred to as “code.” However, when referring to a professional software product, “code” by itself is not sufficient. Software products need to undergo defect corrections, and other maintenance services, which typically include user instruction, corrections, adaptations, and improvements of the software product during their life cycle.

* A software product comprises components, required to ensure the operational success of the services provided by the product. The components include:
  + **Computer programs (the codes):** The computer programs activate the computer system to perform the required applications. It is a collection of instructions that can be executed by a computer to perform a specific task.
  + **Procedures:** Procedures define the order and schedule within which the software or project programs are performed, the method for handling common malfunctioning of software products, and so on.
  + **Documentation:** The purpose of the documentation is to instruct or support new software product version developers, maintenance staff, and end-users of the software product.

# The Causes of Software Errors

* **Faulty Requirements Definition:** Usually prepared by the client, is one of the main causes of software errors. It is also the root causes of software errors. The most common errors of this type are:
  + Erroneous definition of requirements
  + Lack of essential requirements
  + Incomplete requirements definition
  + Inclusion of unnecessary requirements
* **Client–developer Communication Failures:** Misunderstandings resulting from a defective client–developer communication are additional causes for errors that prevail in the early stages of the development process. They must be aware of all signs of lack of communication, on both sides. Examples of these situations are:
  + Poor understanding of the client’s instructions
  + Written changes during development
  + Oral changes during developments
  + Lack of attention by the client to developer questions
  + Lack of attention by developers dealing with requirements and changes
* **Deliberate Deviations from Software Requirements:** An action that often causes software errors. This situation occurs when the developer incorrectly interprets a requirement and develops the software based on his own understanding. The most common situations of deliberate deviations are:
  + Reuse previous work to save time
  + Omit part of the required functions in an attempt to better handle time or budget pressures.
  + Ignore seemingly minor features, which sometimes are quite major.
* **Logical Design Errors:** Software errors can enter the system when professionals designing the system; system architects, software engineers, system analysts, and so on formulate the software requirements into design definitions. It is also a common source of errors; typical logical errors include:
  + Erroneous algorithms
  + Process definitions that contain sequencing errors
  + Erroneous definition of boundary conditions
  + Omission of required system state
  + Illegal operation of the software system
* **Coding Errors:** There is a wide range of reasons cause programmers to make coding errors. Some coding errors include:
  + Misunderstanding the design documentation
  + Linguistic errors in programming languages
  + Errors in data selection
  + Data processing errors
* **Noncompliance with documentation and coding instructions:** Even if the quality of the “noncomplying” software is acceptable, difficulties will inevitably be presented when trying to understand it. In other words, future handling of this software (by development and/or maintenance teams) is expected to substantially increase the rate of errors.
* **Shortcomings of the Testing Process:** Shortcomings of the testing process affect the error rate by leaving a greater number of errors undetected or uncorrected. These shortcomings result from:
  + Incomplete test plans
  + Failure to document, report detected errors and faults
  + Failure to fix the errors due to time constraints
  + Incomplete testing of software error corrections
  + Incomplete corrections of detected errors due to negligence
* **User Interface and Procedure Errors:** User interfaces direct users in areas such as the performance of input and output activities, and data collection and processing. Procedures direct users with respect to the sequence of activities required at each step of the process. User interface and procedure errors may cause processing failures even in cases of error-free design and coding.
* **Documentation Errors:** The documentation errors of concern to the development and maintenance teams are those found in the design, software manuals, documents, user manual and in the “help” displays incorporated in the software. Typical errors of this type are:
  + Errors in the design documents
  + Errors in the documentation within the software user manual
  + Listing of non-existing functions
  + Many error messages are meaningless.

**References:**

Galin, D. (2018). *Software quality assurance – Concepts and practice: IEEE Computer Society, Inc.*

Laporte, C. and April, A. (2018). *Software quality assurance: IEEE Computer Society, Inc.*

Nine Causes of Software Errors (n.d.) Retrieved from https://coggle.it Involvement of people employees (n.d.). Citing sources. Retrieved from

https://[www.efrontlearning.com](http://www.efrontlearning.com/)

Leadership (n.d.). Citing sources. Retrieved from https://[www.forbes.com](http://www.forbes.com/) Continuous improvement (n.d.). Citing sources. Retrieved from

https://smallbusiness.chron.com

System approach to management (n.d.). Retrieved from https://[www.slideshare.net/](http://www.slideshare.net/) Process approach (n.d.). Citing sources. Retrieved from https://the9000store.com Mutually beneficial supplier relationships (n.d.) Citing sources. Retrieved from

https://ebrary.net