Software quality control

Software quality control is the set of procedures used by organizations[1] to ensure that a software product will meet its quality goals at the best value to the customer,[2] and to continually improve the organization's ability to produce software products in the future.[1]

Software quality control refers to specified functional requirements as well as non-functional requirements such as supportability, performance and usability.[2] It also refers to the ability for software to perform well in unforeseeable scenarios and to keep a relatively low defect rate.

These specified procedures and outlined requirements lead to the idea of Verification and Validation and software testing.

It is distinct from software quality assurance which encompasses processes and standards for ongoing maintenance of high quality of products, e.g. software deliverables, documentation and processes - avoiding defects. Whereas software quality control is a validation of artifacts compliance against established criteria - finding defects.

Contents

Definition

Software quality control is a function that checks whether a software component, or supporting artifact meets requirements, or is "fit for use". Software Quality Control is commonly referred to as Testing.

Quality Control Activities

- Check that assumptions and criteria for the selection of data and the different factors related to data are documented.
- Check for transcription errors in data input and reference.
- Check the integrity of database files.
- Check for consistency in data.

- Check that the movement of inventory data among processing steps is correct.
- Check for uncertainties in data, database files etc.
- Undertake review of internal documentation.
- Check methodological and data changes resulting in recalculations.
- Undertake completeness checks.
- Compare Results to previous Results.

Software Control Methods

- Rome laboratory Software framework
- Goal Question Metric Paradigm
- Risk Management Model
- The Plan-Do-Check-Action Model of Quality Control
- Total Software Quality Control
- Spiral Model Of Software Developments
- Control management tool

Verification and validation

Verification and validation assure that a software system meets a user's needs.

Verification: "Are we building the product right?" The software should conform to its specification.

Validation: "Are we building the right product?" The software should do what the user really requires.

Two principal objectives are:

- Discovery of defects in a system.
- Assessment of whether the system is usable in an operational situation.

Verification and Validation of Methods

- Independent Verification and Validation (IV&V)
- Requirements Traceability Matrix (RTM)
- Requirements Verification Matrix
- Software Quality Assurance[1]

Testing

- Unit testing
- Functional testing
- Integration testing
- System testing
- Usability testing
- Software performance testing
- Load testing
- Installation testing

- Regression testing
- Stress testing
- Acceptance testing
- Beta testing
- Volume testing
- Recovery testing

See also

- Software quality management
- Software quality assurance
- Verification and Validation (software)
- Software testing

References

1. Clapp, Judith A, *Software Quality Control, Error Analysis, and Testing*, 1995 William Andrew In.
2. http://www.sqa.net/softwarequalitycontrol.html

- Wesselius, Jacco, "Some Elementary Questions on Software Quality Control"
- https://web.archive.org/web/20071023034030/http://satc.gsfc.nasa.gov/assure/agbsec5.txt

External links

- Software Engineering Body of Knowledge Ch. 11 Sec. 2.1 (http://www2.computer.org/portal/web/swebok/html/ch11#Ref2.1)