



BEST PRACTICE Verification Techniques

Verification is the process of confirming that interim deliverables have been developed according to their inputs, process specifications, and standards. Verification techniques are listed below.

Feasibility Reviews

Tests for this structural element verify the logic flow of a unit of software (e.g., verifying that the software could conceivably perform after the solution is implemented the way the developers expect). Output from this review is a preliminary statement of high-level market requirements that becomes input to the requirements definition process (where the detailed technical requirements are produced).

Requirements Reviews

These reviews examine system requirements to ensure they are feasible and that they meet the stated needs of the user. They also verify software relationships; for example, the structural limits of how much load (e.g., transactions or number of concurrent users) a system can handle. Output from this review is a statement of requirements ready to be translated into system design.

Design Reviews

These structural tests include study and discussion of the system design to ensure it will support the system requirements. Design reviews yield a system design, ready to be translated into software, hardware configurations, documentation and training.

Code Walkthroughs

These are informal, semi-structured reviews of the program source code against specifications and standards to find defects and verify coding techniques. When done, the computer software is ready for testing or more detailed code inspections by the developer.

Code Inspections or Structured Walkthroughs

These test techniques use a formal, highly structured session to review the program source code against clearly defined criteria (System Design



Specifications, product standards) to find defects. Completion of the inspection results in computer software ready for testing by the developer.

Requirements Tracing

At each stage of the life cycle (beginning with requirements or stakeholder needs) this review is used to verify that inputs to that stage are correctly translated and represented in the resulting deliverables. Requirements must be traced throughout the rest of the software development life cycle to ensure they are delivered in the final product. This is accomplished by tracing the functional and non-functional requirements into analysis and design models, class and sequence diagrams, and test plans and code. The level of traceability also enables project teams to track the status of each requirement throughout the development and test process.

References

Guide – CSTE Common Body Of Knowledge, V6.1