

7.3 RELIABILITY

Most important and dynamic characteristic of software is its reliability. Software Reliability is the probability of failure-free software operation for a specified period of time in a specified environment. Software Reliability is also an important factor affecting system reliability. It differs from hardware reliability in that it reflects the design perfection, rather than manufacturing perfection. The high complexity of software is the major contributing factor of Software Reliability problems. Software Reliability is not a function of time - although researchers have come up with models relating the two. The modeling technique for Software Reliability is reaching its prosperity, but before using the technique, we must carefully select the appropriate model that can best suit our case. Measurement in software is still in its infancy. No good quantitative methods have been developed to represent Software Reliability without excessive limitations. Various approaches can be used to improve the reliability of software, however, it is hard to balance development time and budget with software reliability.

7.4 VERIFICATION AND VALIDATION

Verification is a quality process that is used to evaluate whether or not a product, service, or system complies with a regulation, specification, or conditions imposed at the start of a development phase. Verification can be in development, scale-up, or production. This is often an internal process.

Validation is the process of establishing documented evidence that provides a high degree of assurance that a product, service, or system accomplishes its intended requirements. This often involves acceptance and suitability with external customers.

It is sometimes said that validation ensures that 'we built the right thing' and verification ensures that 'we built it right'. Building the right thing refers back to the user's needs; while 'building it right' checks that the documented development process was followed. In some contexts, it is required to have written requirements for both as well as formal procedures or protocols for determining compliance.

V&V is intended to be a systematic and technical evaluation of software and associated products of the development and maintenance processes. Reviews and tests are done at the end of each phase of the development process to ensure software requirements are complete and testable and that design, code, documentation, and data satisfy those requirements.

7.4.1 Verification and Validation Planning

V&V is an expensive process. For some large systems, such as real-time systems with complex non-functional constraints, half the system development budget may be spent on V & V. Careful planning is needed to get the most out of inspections and testing and to control the costs of the verification and validation process. The planning of validation and verification of a software system should start early in the development process. Test planning is concerned with setting out standards for the testing process rather than describing product tests. Test plans also provide the information to staff to get an overall picture of the system tests and to place their own work in this context. Test plans also provide information who is responsible for ensuring that appropriate hardware and software resources are available to the testing team.

The structure of software test plan:

- The testing process
- Requirements traceability
- Tested items
- Testing schedule
- Test recording procedures
- Hardware and software requirements
- Constraints

Like other plans, the test plan is not a static document. It should be revised regularly as testing is an activity that is dependent on implementation being complete. If a part of the system is incomplete, it can not be delivered for integration testing.