

9.3 SOFTWARE RELIABILITY IMPROVEMENT TECHNIQUES

Good engineering methods can largely improve software reliability. Before the deployment of software products, testing, verification and validation are necessary steps. Software testing is heavily used to trigger, locate and remove software defects. Software testing is still in its infant stage; testing is crafted to suit specific needs in various software development projects in an ad-hoc manner. Various analysis tools such as trend analysis, fault-tree analysis, Orthogonal Defect classification and formal methods, etc, can also be used to minimize the possibility of defect occurrence after release and therefore improve software reliability.

After deployment of the software product, field data can be gathered and analyzed to study the behaviour of software defects. Fault tolerance or fault/failure forecasting techniques will be helpful techniques and guide rules to minimize fault occurrence or impact of the fault on the system.

9.4 DEFINITION OF QUALITY

Standards shall and can help to define terms like quality. Nevertheless, the means of expression used in standards are often not appropriate for the practice. This is also true for the definition of the ISO 8204 for quality: "Totality of characteristics of an entity that bears on its ability to satisfy stated and implied needs." That means: We require a quality software product to have certain characteristics that are related to requirements (of the user) and satisfy them. It is clear that the pair requirement and characteristic plays a central role in the definition of quality. Therefore, an object oriented model contributes to a better understanding for these notions

A quality factor is defined by "a non-functional requirement for a software program which is not called up by the customer's contract, but nevertheless is a desirable requirement which enhances the quality of the software program".

- Quality software product must. ✓
- Do what it should, and only this (no side effects, no bugs).
- Be usable for the intended audience. ✓
- Be maintainable for the programmers intended to do so.
- Not make the system it is running on less secure. ✓

9.5 SOFTWARE QUALITY ASSURANCE

Software quality assurance (SQA) consists of a means of monitoring the software engineering processes and methods used to ensure quality. The methods by which this is accomplished are many and varied, and may include ensuring conformance to one or more standards. SQA encompasses the entire software development process, which includes processes such as software design, coding, source code control, code reviews, change management, configuration management, and release management.

9.5.1 Software Quality Assurance Activities

Product evaluation and process monitoring are the SQA activities that assure the software development and control processes described in the project's Management Plan are correctly carried out and that the project's procedures and standards are followed. Products are monitored for conformance to standards and processes are monitored for conformance to procedures. Audits are a key technique used to perform product evaluation and process monitoring. Review of the Management Plan should ensure that appropriate SQA approval points are built into these processes. Product evaluation is an SQA activity that assures standards are being followed. Ideally, the first products monitored by SQA should be the project's standards and procedures. (SQA assures that clear and achievable standards exist and then evaluates compliance of the software product to the established standards.) Product evaluation assures that the software product reflects the requirements of the applicable standard(s) as identified in the Management Plan. Process monitoring is an SQA activity that ensures that appropriate steps to carry out the process are being followed. SQA monitors processes by comparing the actual steps carried out with those in the documented procedures.

The Assurance section of the Management Plan specifies the methods to be used by the SQA process monitoring activity. A fundamental SQA technique is the audit, which looks at a process and/or a product in depth, comparing them to established procedures and standards. Audits are used to review management, technical, and assurance processes to provide an indication of the quality and status of the software product.

The purpose of an SQA audit is to assure that proper control procedures are being followed, that required documentation is maintained, and that the developer's status reports accurately reflect the status of the activity. The SQA product is an audit report to management consisting of findings and recommendations to bring the development into conformance with standards and/or procedures.