

Quality Management & Assurance of Critical Software-Based Systems

Good Afternoon

Software plays a crucial role in every walk of life, these days. From automated washing machines and microwaves to office work, it's used in almost every task performed by humans and machines.

The quality of Software-Based Systems plays a vital role in several domains of technology.

Systems ranging from nuclear power plants to driver-less cars need quality assurance & management.

Crashing or malfunctioning can severely affect people's lives.

The quality has to be assured with a requirement of a high level of accuracy.

It is important to ensure the qualitative parameters and analyzing the plausible outcomes after rigorous testing.

In this paper it highlights, a framework composed of, numerous, optimum software quality assurance practices, in development of such systems and elaborates the 9 phases which solidifies the quality of Critical Software-based System.

A category of software, named safety critical software (SCS), is responsible for automation of numerous real time systems. It's failure can affect people's lives & environment to a large extent, for example, aviation control systems etc.

SCS is one that has the potential to cause accidents. It depends on the decision making feature, that works during any unsafe condition occurs.

These kinds of systems are expected to have a high level of specific qualities such as

- reliability,
- availability,
- durability,
- security
- safety.

The list for the implementation of SRS for the accompanying the changes are as follows:

- Software Requirement Specification (SRS)
- Functional Specification (FS)
- Data Managerial Plan
- Configuration Managerial Plan

- Risk Assessing Plan
- Testing Plan
- User Manuals
- Backup and Recovery Plan
- Maintenance Plan F. Configuration Management

Because human safety is dependent on that. So, the system should be risk free and fail-proof.

Quality is defined as the level of user satisfaction. It is considered in every stage of software development & is a planned systematic pattern of all actions needed for providing an adequate confidence for the conforming the established technical requirement.

It directly impacts product's success as well as the safety.

The larger the software application, higher the number of phases of testing. This paper gives a framework for numerous, optimum software quality assurance practices, in development of such systems and elaborates the 9 phases which solidifies the quality of Critical Software-based System.

It begins with the main goal of the practices that are to be implemented on the software based system and determines those functional and non-functional techniques for the process verification and validation with a highlight to the backup & the recovery groundwork for the tackling with the disasters and unforeseen situations.

It is hard to grow huge, complex programming and to ensure that this product doesn't cause issues during activity. In the event that issues happen, these may cause cataclysmic impacts in specialized application areas.

This paper advocates a framework for the assurance of the conformed quality in the development process of a critical software-based system. These software applications are critical for the safety and reliability that need the top-most degree of accuracy and testing. Quality Assurance of the software is a key discipline that can affect the life of people and the environment, in turn, which is responsible for the checking of these type of features.

To conclude

there is a great disparity in development processes used for the development of the Safety Critical Systems; hence it is nearly impossible to develop a generic solution. This problem can be encountered with the integration of data-ware housing and data mining techniques. That would later serve the purpose of refining the presented approach according to the specific system under development.