Email: knock@italics.in
Call:(+91) 9818886748

Non Functional Requirements In Software Projects

Non-Functional Requirements (NFRs) in software projects even though they are either overlooked or less emphasized form a critical component in meeting the desired quality standards and delivering on time, among other things. In fact non-functional requirements are crucial right from the development stage to final delivery of the software project making it essential for the IT decision makers to take these into account for any software project while making decisions.

Traditionally referred to as 'ilities', the changing business dynamics and stifling IT spending have transformed the business scenario to widen the spectrum of non-functional requirements to now include aspects hitherto treated as insignificant. Moreover, the need to take all the non-functional requirements into account for every software project has gained enormous significance mainly because if accounting for various non-functional requirements leads to quality of work and scheduled execution then avoiding the same can have significant adverse impacts as well.

For CIOs and CTOs taking care of the complexities that underlie the non-functional requirements is a desirable goal which needs to be accomplished. Beginning with understanding the concept of non-functional requirements, its various varieties as well as the impacts these have on the software projects and subsequently developing a workable and effective NFR framework helps IT decision makers in not only executing software projects efficiently but also bringing in financial viability.

Types of Non Functional Requirements

Sometimes referred to as 'softgoals', the non-functional requirements in software projects can be categorized into software product requirements, organizational requirements and external requirements. Most of the requirements collectively referred to as 'ilities' have mostly been taken into account while executing any software project. Keeping with the

times though the scope of non-functional requirements has expanded to account for the changing global market order, system properties as well as constraints to include;

- **Usability,** which defines the ease of interaction between the software and the user.
- **Reliability,** to check for the probability of failure-free operation of the software.
- **Interoperability,** to enable the software to work under various environments.
- **Scalability,** so that the software can work flawlessly and can be upgraded in the future when work volume increases.
- **Security,** for data and information protection.
- **Efficiency,** to get optimum outcomes out of the processes and operations.
- **Quality,** for maintaining desired levels of quality all along.
- **Complexity,** to take care of multimedia, tier and algorithm requirements.
- **Reusability,** to create a source code which can easily add functionalities.
- **Predictability,** to be sure of the timeliness.
- **Project Management Methodology,** to know beforehand the way the project will be handled.
- **Leadership and Team Dynamics,** essential for successful completion of the project.
- **Multiple Development Sites,** become significant in the existing global delivery model.