**ST.XAVIER’S COLLEGE**

Maitighar,Kathmandu



**Theory assignment #6**

**Submitted By:**

Yogesh Gautam

BIM 6th Sem ‘B’

014BIM062

**Submitted To:**

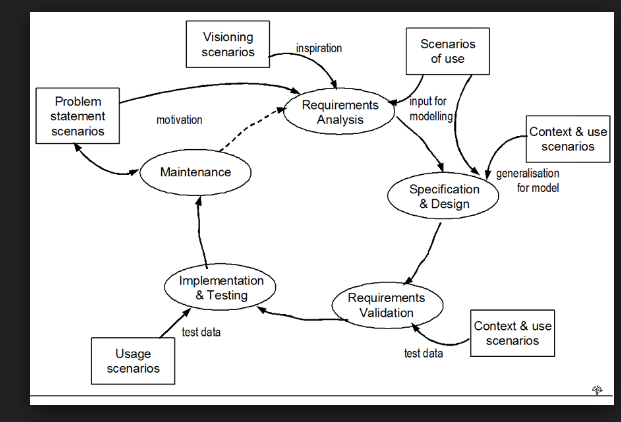
Bal Krishna Subedi

Lecturer

Department of Computer Science

Requirements analysis (requirements engineering)

Requirements analysis, also called requirements engineering, is the process of determining user expectations for a new or modified product. These features, called requirements, must be quantifiable, relevant and detailed. In software engineering, such requirements are often called functional specifications. Requirements analysis is an important aspect of project management.

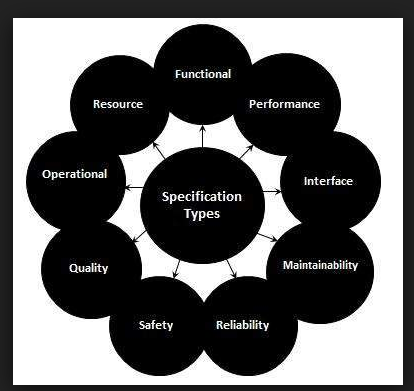


# Software requirements specification

A **software requirements specification** (**SRS**) is a description of a software system to be developed. It lays out  and non-functional requirements, and may include a set of use cases that describe user interactions that the software must provide.

Software requirements specification establishes the basis for an agreement between customers and contractors or suppliers (in market-driven projects, these roles may be played by the marketing and development divisions) on what the software product is to do as well as what it is not expected to do. Software requirements specification permits a rigorous assessment of requirements before design can begin and reduces later redesign. It should also provide a realistic basis for estimating product costs, risks, and schedules.[[1]](https://en.wikipedia.org/wiki/Software_requirements_specification#cite_note-1) Used appropriately, software requirements specifications can help prevent software project failure.[[2]](https://en.wikipedia.org/wiki/Software_requirements_specification#cite_note-2)

The software requirements specification document enlists enough and necessary requirements that are required for the project development.[[3]](https://en.wikipedia.org/wiki/Software_requirements_specification#cite_note-Pressman-3) To derive the requirements, the developer needs to have clear and thorough understanding of the products to be developed or being developed. This is achieved and refined with detailed and continuous communications with the project team and customer till the completion of the software.



Functional vs. non-Functional Requirements

**1) Objectives**

Functional testing is a type of black-box testing which is performed to check if the software application meets the predefined requirements and specifications of the business. The functional testing results helps managers to take crucial decisions related to the software’s promotion and release. But non-functional testing is performed with the objective to assess the user experience delivered by the application. The testers perform various types of tests against the non-functional requirements to check the amount of time required by the software to respond to user requests or perform specific actions.

**2) Types of Testing**

The testers need to perform a variety of tests to assess the features and functionality of the software. While assessing the functional requirements, they have to perform unit, system, integration, interface, smoke, sanity and user acceptance testing. Likewise, the QA professionals have to perform various tests to assess the non-functional requirements of the application. They have to perform security, usability, performance, load, installation, configuration, compatibility, interoperability, maintainability, and recoverability testing to check the application’s user experience. However, the types of functional and non-functional testing required by individual projects differ.

**3) Focus on Identifying Bugs or Defects**

Often the functionality of an application is impacted by bugs or defects. So the testers need to perform functional testing to identify and eliminate all bugs to make the software meet business requirements exactly. Thus, various types of functional testing aim to identify the bugs or flaws in a software application. On the other hand, the testers are not required to identify defects of bugs in an application while performing non-functional testing. While performing non-functional testing, they have to focus on the software’s user experience and behaviour in different conditions.

**4) Criterions or Parameters**

Functional testing and non-functional testing are performed based on different criterions or parameters. While assessing the functional requirements, the testers have to keep in mind both positive and negative criterions. However, they have to submit valid inputs to check if the software generates the desired output and behaves in the expected manner. They also need to submit invalid inputs and perform invalid operations to check the application’s behaviour in unexpected conditions. On the other hand, while performing non-functional testing they have to consider the criterions that are not considered while performing the positive and negative functional tests.

**5) Test Automation**

A business can reap a number of benefits by automating both functional and non-functional software testing efforts. The test automation tools will help QA professionals to perform a variety of tests repeatedly and frequently. Also, the tools will help the managers to obtain more accurate and reliable information about the functional and non-functional aspects of the software. However, it is always important for the business to pick the right automation tool for each type of functional and non-functional tests. The testers can use the tools to perform load, performance, compatibility and stress testing repeatedly. But they have to write custom test scripts to perform positive and negative functional tests under various conditions.

It is also important for a business to focus on both functional and non-functional requirements to make the software testing strategy more comprehensive and effective. However, the business must clearly define the functional and non-functional requirements to make it easier for testers to perform all required tests.