

Functional and Non Functional Property

Functional Property is related to functional requirement, Functional requirements describe what the system should do

Non functional property denotes a feature of software that is not covered by functional properties of software . **It includes**

Reliability: Reliability is the ability of a system to perform its required functions under stated conditions

Performance Performance requirements concern the speed of operation of a System. Types of performance requirements:

1. Response requirements (how quickly the system reacts to a userinput)
2. Availability requirements (is the system available for service when requested by end-users)

Security Security requirements are included in a system to ensure:

- ✓ <Unauthorised access to the system and its data is not allowed
- ✓ < Ensure the integrity of the system from accidental or malicious Damage

Usability Usability is the ease with which a user can learn to operate, prepare inputs for, and interpret outputs of system or component

Supportability concerned with the ease of changes to the system after deployment.

1. Adaptability: The ability to change the system to deal with additional application domain concepts
2. Maintainability: The ability to change the system to to fix defects)
3. Portability: The ease with which a system or component can be transferred from one environment to another

Architecture Structure and Views

View :- A representation of a set of elements

In construction, there are blueprints of

- Plan
- Different sites of construction
- Electrical wiring
- Plumbing

Each of these views specifies a single entity (i.e. the construction) from a different perspective (used by a different person)

Structure :- The set of elements itself. You can divide software architecture structures into 3 groups:

- 1) module structures,
- 2) component-and-connector structures
- 3) allocation structures.

Module Structures: There is less emphasis on how the resulting software manifests itself at runtime. Module structures allow us to answer questions such as What is the primary functional responsibility assigned to each module? What other software elements is a module allowed to use? What modules are related to other modules by generalization or specialization (i.e. inheritance) relationships? ‘

Component-and-connector structures: Here the elements are runtime components and connectors (which are the communication vehicles among components). Component-and-connector structures help answer questions such as What are the major executing components and how do they interact? What are the major shared data stores? Which part of the system are replicated? How does data progress through the system? What parts of the system can run in parallel? How can the system’s structure change as it executes?

Allocation structures Allocation structures show the relationship between the software elements and the elements in one or more external environments in which the software is created and executed. They answer questions such as What processor does each software element execute on? In what files is each element stored during development, testing, and system building?