SYEDA TOOBA ALI

SE-053

PAPER: B

SE-053 Syeda Tooba Ali Date:

Question: 1

FUNCTIONAL VS NON-FUNCTIONAL:

Functional requirements are those requirements which describe the functionality of the system i.e what the system does and how it reacts to inputs and behaves in different situations.

Whereas

Non-functional requirements relate to the system as a whole and they capture the emergent behaviour of the system; including timing, performance, reliability, accuracy, security constraints e.t.c.

Examples Of NFRs:

- The statements like "Ease Of Use", "Rapid User Response", "Recovery from Failure" are Non-functional reqs.

★ Scenario-based examples:-

(1) After 3 hours of training, experienced controllers should be able to use all functions of the system. After training, the number of errors made by experienced controllers should not exceed an average of two per day.

SE_053 Syeda Tooba Ali Date:

② The system should process 3 transactions per second and the rate of failure should be no more than 2%.

SE - 053 Syeda Tooba Ali

Date:

Question: 02

PROBLEMS ENCOUNTERED DURING REQs ELICITATION:

Following are the problems usualy encountered during requirements elicitation:

1) Problems Of Scope:

Problems of scope ~~to~~ occur if:

(i) Unnecessary or extraneous information about design is provided.

(ii) The boundary of the system to be developed is not defined properly.

2) Problems of volatality:

As the time passes, requirements keep changing due to one reason or another. Throughout the system developement process, new requirements may evolve and the existing ones might change.

3) Problems Of Understanding:

They Include:

(i) Requirements are vague, unmeasurable and untestable e.g. "user friendly system" or "robust system".

(ii) Users have little or no understanding about technology & its limitations.

(iii) Analysts have little or no domain knowledge.

(iv) User do not understand their needs completely.

(v) Analysts and users speak different languages which may lead to ambiguity.

(vi) Different users have different perspectives; so conflicts may occur.

(vii) Obvious information can be omitted easily ~~which~~ without making its way to requirements document.

Question:03:-

DEFECTS WHICH PBR HELPS DETECT:

PBR can help detect the following Requirements defects:-

(1) MISSING INFORMATION:

* Any requirements related to functionality, design, performance e.t.c may be missing.

* How the system is going to behave for all inputs for all situations is not defined.

* Figures, references or any related diagrams may be missing. Different terms used or units may be undefined. A whole section of requirements document may be missing.

(2) AMBIGOUS INFORMATION:

The meanings of requirements are not clear or can be interpreted differently by different stakeholders.
(One word can have different meanings).

(3) INCONSISTENT INFORMATION:-

Requirements may conflict with one another

(4) INCORRECT FACTS:

Some requirements-asserted facts might not be true under specific conditions of the system.

(5) Extraneous INFORMATION:

Unnecessary/ Extra/ unused information may be provided.

(6) MISCELLENOUS DEFECTS:

PBR can also help identifying other errors for example; including ~~info~~ requirements in the wrong section in the document.

Question #04:-

NEGATIVE IMPACT OF REQs ERRORS:

Requirements errors may have the following negative impacts.

(1) The resulting software product might be useless and does not satisfy the real needs of users.

(2) If one requirement can be interpreted in multiple ways; it will lead to ambiguities and can cause disagreements between customers/users and developers. which will result in wastage of time and money; and can result in lawsuits as well.

(3) It can have negative effect on people; e.g

→ Customers and developers will be unsatisfied.

→ People will lose interest in automation of Processes.

→ A blame game might start among different people.