NAME : Muhammad Younis Afridi

I.D : 12701

BS (SE)

Course : Software Requirement Specification.

Instructor : Asma Khan

Question No: 1

Ans 3 Requirement; In product development and process optimization, a requirement is a singular documented physical or functional need that a particular design, product or process aims to satisfy. It is commonly use in a formal sense in engineering design, including for example in system engineering, Software engineering or enterprise engineering.

OR

- It may range from a high-level abstract statement of a service or of a system constraint to a detailed mathematical functional specification.
- This is inevitable as requirements may serve a dual function

P.T.O

Requirement engineering is the process of establishing the services that the customer requires from a system and the constraints under which it operates and is developed. The requirements themselves and are the description of the system services and constraints that are generated during the requirements engineering process. requirements may range from a high-level abstract statement of a services or of a system constraint to a detailed mathematical functional specification.

System requirements: Detailed description of what the system should do including the software system's functions, services and operational constraints. The system requirements document should define exactly

P.T.O

what is to be implemented. It may be part of the contract between the system buyer and the software developers.

Gathering software requirements is the foundation of entire development project. Hence it must be clear, correct and well-defined.

A complete software requirement specification must be.

- Clear.
- Correct
- Consistent
- Coherent.
- Comprehensible
- modifiable
- verifiable.
- prioritized.
- unambiguous.
- Traceable
- Credible source.

users of requirement document (example)

System customers → Specify the requirement and read them to check that they meet their needs. Customers specify changes to the requirements.

Managers → use the requirements document to plan a bid for the system and to plan the system development process.

System engineering → use the requirements to understand what system is to be developed.

System Test engineers → use the requirement to develop validation tests for the system.

System maintenance engineer → use the requirements to understood the system and the relationships between its parts.

P.T.O

2 Question NO: 2

Ans Different Types of software requirements.

1. Business Requirements (BR)

- These are high-level business goals of the organization building the product, or the customer who commissioned the project.
- These are usually provided as a single page of high-level bullets.

2. Marked Requirements (MR)s

- These drill down into BRs, but still are high-level. In addition to business goals, they also outline market needs.
- These are usually provided as a prioritized bullet list or table, and usually less than 5 page long.

P.T.O

3. Function requirements (FR) - use case :

- These cover the functionality of the product in detail. use case are one of the best ways of documenting functional requirements.

- Depending on the product being built, FR, can run several hundred pages.

4. Non-functional Requirements (NFR) :

- These are not related to the "functionality" of the product - but cover goals such as reliability, Scalability, Security integration etc.

- Many project make the mistake of not specifying these explicitly.

5. UI requirements (UIR) :

- user interface Specs are not considered "requirement" in traditional requirements

P.T.O

management theory.

- Phooey! In my opinion. UI specs are indeed requirement and in fact should be considered an integral part of requirments for any Software that has a UI.

Question NO: 3

Ans 3 System requirment Engineering:

System requirments deals with the activities which attempt to understand the exact needs of the users of the Software to be developed and so translate such needs into precise and unambiguous statements which will subsequently be used in the development of the System.

Requirement Engineering is becoming the key issue for the development of Software Systems that meet the expectations of their customers and user's are delivered on time and developed within budget.

Software requirement Engineering:

Software requirment Engineering is the process of establishing the service that the customer require from a system and the constraints under which it operates and is developed. The requirements themselves are the descriptions of the system services and constraints that are generated during the requirements engineering process. Requirement may vary from a high level abstract statement of a service or of a system constraint to a detailed mathematical functional specification.

Question NO # 4

Ans # 4

1. Know Your Goals :

- Create a proposed with your goals in mind.
- outline what you want to see pricing for.

2. Monitor your data :

- Known and understand your software usage data
- knowing your numbers demonstrates strength in a negotiation.
- Having your data organized could act as an ~~debt~~ audit deterrent.

3 Do your Homework :

- Dont get locked-in find out if it's

P.T.O

possible to switch to alternative products.

- Be aware of using this as your only bargaining chip, as publishers will call your bluff.
- Research how much similarly sized companies are paying for the same product.

4. Talk to stakeholders:
   - Don't talk to the vendor too much, but keep stakeholders updated.
   - Include stakeholders from IT, procurement, legal, and finance on the negotiation process.
5. Expect:
   The complete vendor playbook
   How to squeeze your customers

P.T.O

Question NO : 5

Ans 5

Actors :

Administractor, Government Health database, private health insurance system.
(The last two are external system).

Objects :

- patient
- Administractor.
- Address
- Emergency Contact
- public Health insurance.
- private health insurance.
- Registration.

Assumptions

Address contain street number, Street name, Suburb, and postcode and So is large enough to be an object.

use case diagram