Name suliman khan
ID 14282
Subject srs
Date 24/06/20

Question No: 02 (10)

Explain software requirements types.

Ans :

The most common types of software requirements are below.

1. *Business Requirements*

   - *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. *Market Requirements*

   - *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 bulleted list or table, and are usually less than 5 pages long.*

3. ***Functional Requirements (FR) – Use Cases***

   - *These cover the functionality of the product in detail use cases are one of the best ways of documenting functional requirements*
   - *Depending on the product being built, FRs can run several hundred pages.*

Requirements, which are related to functional aspect of software fall into this category.

They define functions and functionality within and from the software system.

Describe functionality or system services Depend on the type of software, expected users and the type of system where the software is used Functional user requirements may be high-level statements of what the system should do; functional system requirements should describe the system services in det

Examples -

- ***Search option given to user to search from various invoices.***

- ***User should be able to mail any report to management.***
- ***Users can be divided into groups and groups can be given separate rights.***
- ***Should comply business rules and administrative functions.***
- ***Software is developed keeping downward compatibility intact***.

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 projects make the mistake of not specifying these explicitly.***

Requirements, which are not related to functional aspect of software, fall into this category. They are implicit or expected characteristics of software, which users make assumption of.

ements Product requirements – Requirements which specify that the delivered product must behave in a particular way, e.g. execution speed, reliability etc. Organisational requirements – Requirements which are a consequence of organisational policies and procedures, e.g. process standards used, implementation requirements etc. External requirements – Requirements which arise from factors which are external to the system and its development process, e.g. interoperability requirements, legislative require

Non-functional requirements include -

1. **Security**
2. **Logging**
3. **Storage**
4. **Configuration**
5. **Performance**
6. **Cost**
7. **Interoperability**
8. **Flexibility**
9. **Disaster recovery**
10. **Accessibility**

5. **UI Requirements (UIR)**

interface specs are not considered "requirements" in traditional requirements management theory.

Phooey! In my opinion, UI specs are indeed requirements – and in fact should be considered an integral part of requirements for any software that has a UI.

6. **Domain requirements**

• The system's operational domain imposes requirements on the system. – For example, a train control system has to take into account the braking characteristics in different weather conditions.

• Domain requirements be new functional requirements, constraints on existing requirements or define specific computations.

• ***If domain requirements are not satisfied, the system may be unworkable***.

Question No: 03 (10)

State difference between system requirement engineering and software requirement engineering.

Ans :

- **system requirement engineering**

- ***Computer-based systems fall into two broad categories:***
  - ***User-configured systems where a purchaser puts together a system from existing software products***
  - ***Custom systems where a customer produces a set of requirements for hardware/software system and a contractor develops and delivers that system***
- ***System requirements engineering***
  - ***The requirements for the system as a whole are established and written to be understandable to all stakeholders***
- ***Architectural design***
  - ***The system is decomposed into sub-systems***
- ***Requirements partitioning***
  - ***Requirements are allocated to these sub-systems***
- ***Software requirements engineering***
  - ***More detailed system requirements are derived for the system software***
- ***Sub-system development***
  - ***The hardware and software sub-systems are designed and implemented in parallel.***
- ***System integration***
  - ***The hardware and software sub-systems are put together to make up the system.***
- ***System validation***
  - ***The system is validated against its requirements.***

Software Requirements engineering:

We should try to understand what sort of requirements may arise in the requirement elicitation phase and what kinds of requirements are expected from the software system.

Broadly software requirements should be categorized in two categories:

Functional Requirements

Requirements, which are related to functional aspect of software fall into this category.

They define functions and functionality within and from the software system.

Examples -

- *Search option given to user to search from various invoices.*
- *User should be able to mail any report to management.*
- *Users can be divided into groups and groups can be given separate rights.*
- *Should comply business rules and administrative functions.*
- *Software is developed keeping downward compatibility intact.*

Non-Functional Requirements

Requirements, which are not related to functional aspect of software, fall into this category. They are implicit or expected characteristics of software, which users make assumption of.

Non-functional requirements include -

- *Security*
- *Logging*
- *Storage*
- *Configuration*
- *Performance*
- *Cost*
- *Interoperability*
- *Flexibility*
- *Disaster recovery*
- *Accessibility*

Requirements are categorized logically as

- *Must Have : Software cannot be said operational without them.*
- *Should have : Enhancing the functionality of software.*
- *Could have : Software can still properly function with these requirements.*
- *Wish list : These requirements do not map to any objectives of software.*

Question No 04: (10)

Give five reasons why requirements negotiation is needed in software engineering

Ans ;

Requirements negotiation is needed in software engineering because

Is a decision making process .involves interaction and interdependency

Implies two parties with conflicts

1.***Introduction reqirements negotiation***

2. Negotiation Process

3. Dimensions of reqirements negotiation

4. **Compromise**

5. **Accommodate**

1.***Introduction reqirements negotiation***

It's difficult to get one customer to decide which of his requirements are most important Gaining agreement among multiplev customers with diverse expectations is even more challenging We need to negotiate requirementsv

The objectives of customers, users, or developers have to be reconciled to develop mutually acceptable agreements Stakeholders are not forced to agreev The result of negotiation is also tov understand why stakeholders disagree Identifying opposed interests is crucialv for project success Identified disagreements represent risksv that need to be addressed by managers

Requirements negotiation is an iterative process through which stakeholders make tradeoffs between requested system functionsv the capabilities of existing or envisionedv technology the delivery schedulev the cost

2. Negotiation Process

Theories for negotiation processes Game theoryv Economic modelsv Psycho-sociologic modelsv In game theory, the null sum principlev may apply The gains for one party imply losses forv others Strategies can be devised to obtain thev best outcome

Economic models define the basic factors for interactions among negotiators Negotiators behave rationallyν They focus on the degree of agreementν and its effectiveness Psycho-sociologic models assume thatν the information is imperfect Parties try to influence the others byν manipulating their expectations

The pre-negotiation phase includes definition of the negotiation problemν identification and solicitation of stakeholdersν elicitation of goals from stakeholdersν analysis of goals to find conflictsν The results of this phase are the issuesν and conflicts involved Each issue has a range of options Identifying the right people can accelerate the negotiation process

In the post-negotiation phase, stakeholders (or automated tools) analyze/evaluate the negotiation outcomesν suggest re-negotiation if necessaryν Find if a better solution is possible forν one party, without causing loss to others It can also involve quality assuranceν reviews of the negotiation results Ensure stakeholders' commitment over time

4.Dimensions of reqirements negotiation

A framework for requirements negotiation must address the following dimensions

1. conflict resolution strategy

2. collaboration situation of stakeholders

3. degree of tool support The dimensions cover key questions in ν requirements negotiation how are conflicts resolved? how do stakeholders collaborate?which tools are used to support the process?

Win–lose sound attractive to the winners, but it usually turns into lose–lose The traditional, adversarial, lawyer-oriented ν contracting mechanisms is no solution Flexibility to quickly renegotiate a new ν solution once unforeseen problems arise

Pro-active interventive support tools can coordinate the activities of stakeholders For example, they critique their actions ν or suggest what agreement to accept These tools access and use knowledgebases and intelligent software agents ν They monitor the negotiation process ν and the negotiators' individual activities

Passive support tools provide an infrastructure for negotiation and support all different collaboration situations All parties can express their preferences, ν communicate ideas, offer arguments, and share intermediate/final results Examples: email, chat, multimedia rooms ν Passive systems do not support the ν production of content with hints and guidance

Compromise

Many students of negotiation styles confuse the collaborative style with the compromising one. Unlike the "win-win" collaborative style, the compromising negotiation style follows a "I win/lose some, you win/lose some" model. When reaching the terms of the agreement, compromisers often relinquish some terms in favor of gaining others.

For example, if two governments are trying to reach a trade agreement, a compromiser might give the other government greater access to their country's dairy market to gain protections for digital media trade. Simply put, a compromising negotiation style is a form of bargaining. Compromisers split the agreement's value between the two parties versus finding a solution so that everyone benefits from an agreement's full value. A competitive negotiator can easily take advantage of a compromising negotiator.

A compromising negotiation style is most useful in situations where the opposite party is trustworthy, and the agreement is under a tight deadline. However, compromising will cause your company to lose out on collaborative partnerships and innovative solutions.

Accommodate

An accommodating negotiating style follows the "I lose, you win" model – which does not seem to be in a negotiator's best interest. Accommodating negotiators are the direct opposite of competitive negotiators. They focus on preserving relationships and building a friendly rapport by sacrificing some of their company's interests in favor of the opposite party's interests.

Accommodators tend to try to win people over by giving in to their requests. They tend to share more information than they should. They are often well-liked by their colleagues because of their kindness – but kindness doesn't work in every negotiation situation. Accommodating negotiation styles work best in situations where your company has caused harm to another and needs to repair a significant relationship. These negotiators are skilled at peacemaking between different bodies.

Question No: 01 (10)

Define requirements and define what the system (take example of any system) is required to do and what are the features and constraints under which it operates.

Ans :

Requirements:

May range from a high level abstract statement of a service or

A statement of a system constraint to a detailed mathematical functional specification

Requirements may be used for

- Abid for contact

  Must be open to interpretation the basis for the contact itself must be defined in detail.

  Both above statements may be called requirements.

Banking ATM system

- The example used here is an auto-teller system which provides some automated banking services.
- I use a very simplified system which offers some services to customers of the bank who pwn the system and a narrower range of services to other customers

- Services include cash withdrawal message passing send a message to request a service ordering a statement and transferring funds

System is required to do

The system shall maintain records of all library materials including books, serials, newspapers and magazines, video and audio tapes, reports, collections of transparencies, computer disks and CD-ROMs.

The system shall allow users to search for an item by title, author, or by ISBN.

The system's user interface shall be implemented using a World-Wide-Web browser.

The system shall support at least 20 transactions per second.

The system facilities which are available to public users shall be demonstrable in 10 minutes or less.

Very general requirements which set out in broad terms what the system should do.

Functional requirements which define part of the system's functionality.

Implementation requirements which state how the system must be implemented.

Performance requirements which specify a minimum acceptable performance for the system.

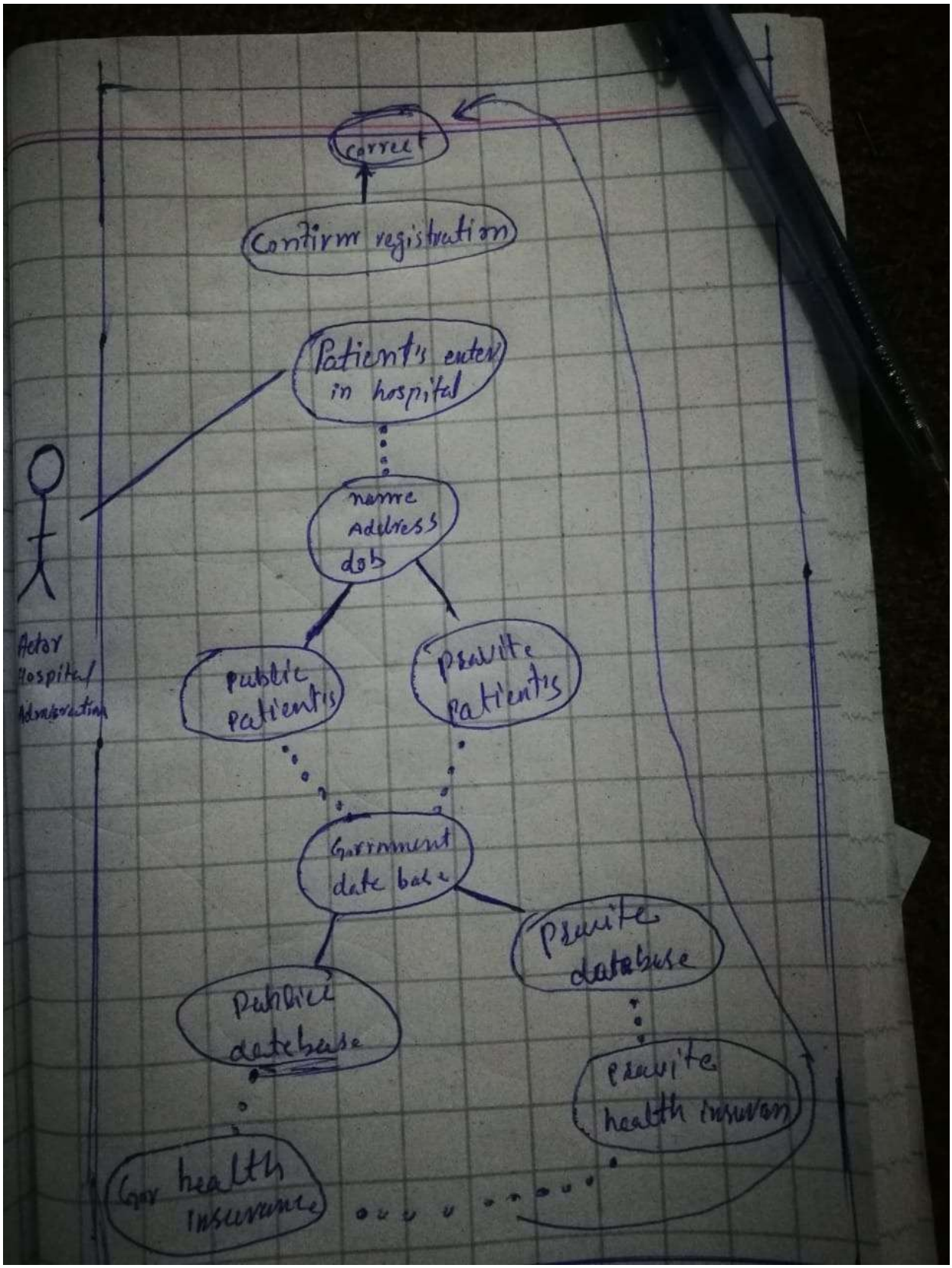
Usability requirements which specify the maximum acceptable time to demonstrate the use of the system.

Question No 05: (10)

Identify the **actors** and the **objects** in the following scenario to register a patient in a hospital management system and draw a **use case diagram**:

The administrator enters the patient's name, address, date of birth and emergency contact detailsinto the system. If the patient has only public health insurance, the administrator enters the patient'smedicare number, and the system verifies this with government health database. If the patient alsohas private health insurance, then the administrator enters also the patient's private healthinsurance details, and the system verifies these details with the private health insurance system.When these details are verified as correct, the system saves the patient's details and confirms theregistration.

Ans


correct
Confirm registration
Patients enter in hospital
name
Address
dob
Actor
Hospital
Administration
public patients
private patients
Government date base
Private database
Public database
Private health insuran
Gov health insurance