COMP- 3110

ASSIGNMENT 2

Keerthana Madhavan, University of Windsor

"I, Keerthana Madhavan 104995097, confirm that I have not received any unauthorized assis- tance in completing this assignment. I acknowledge that a mark of zero may be assigned for copied work."

Professor:
Dr. Roohollah Etemadi

Table of Contents

Questions

Q1. Explain what are the differences between system requirements and business requirements?

System Requirements is a detailed requirement definition report that specifies what the new system should do and what properties it should have.[1] It is a more detailed descriptions of the software system functions, services, and operational constraints. The system requirement is broken down into two types: Functional and Non-Functional requirements.

Business Requirement generates the System Request that has the objective of the product and the vision and scope of the new to-be system. [2]

Q2. What are the trade-offs that software project managers must manage?

During the project selection process, the approval committee must be selective about where to allocate their resources, because the organization might have limited funds. This activity involves the trade-offs, in which the organization must give up something in return for something else to keep its portfolio well balanced.

The project management is involved in making trade-offs among three important concepts: the functionality of the system, the time to complete the project, and the cost of the project. The project managers should control and manage the three interdependent levers throughout the development of the system.[3]

The project managers task is to keep things balanced when there are trade-offs in the project. For example, if a project manager needs to readjust a deadline to an earlier date, then the only solutions are to decrease the functionality of the system or to increase the costs by adding more people or initiate an overtime plan. So, in this case you are giving cost of the project by adding more resources to complete the system. (increased cost is the trade-off).

---

[1] Dr. Roohollah's Notes – Chapter 5 System Analysis – Use Case Modeling

[2] Dr. Roohollah's Notes – Chapter 5 System Analysis – Use Case Modeling

[3] System Analysis Design (7th ed), Alan Dennis, Barbara Haley Wixom, David Tegarden, 2019.

Q3. What is scope creep and how can it be managed?

The most common reason for schedule, and cost overruns that is **scope creep**, occurs after the project is underway (you have to expect sudden changes to your project). The analyst will suffer from this because he/she may assume that the project will be safe from scheduling problems because of careful planning.

Scope Creep happens when new requirements are added to the project after the original project scope was defined already.[4] It can happen from for various reasons:

1. Users might suddenly add new requirements
2. Developers might discover new ways to develop the system
3. Senior manager might want to change the process.

To manage **scope creep**, the team must:

- Identify the requirements in the beginning of the project and to apply analysis techniques effectively.
- Use effective meetings and prototyping to better visualize and support customers' needs. It reduces about 5% of scope creep on a typical project.

Q4. What is timeboxing and why is it used in software projects?

Another approach to scope management is a technique called **timeboxing**. Some companies prefer meeting deadlines over delivering functionality of the product.

Incongruency exits in software development, that is 75 percent of a system can be provided quickly and the latter 25 percent of the functionality would demand most of the time. In this case, one may use timeboxing techniques (with RAD and agile methodologies).

Timeboxing technique sets a fixed deadline for a project, delivers the system by that deadline (even with less functionality). It prevents project team from being stuck on finishing a task to generate a perfect deliverable product.

The steps to implement timeboxing in a project include[5]:

1. Set the date for system delivery
2. Prioritize the most important functionality to be implemented in the system
3. Build the core of the system (ranked from high to low order)
4. Postpone functionality that cannot be provided within the time frame.
5. Deliver the system with core functionality
6. Repeat steps 3 through 5 to add refinements and enhancements to the system.

---

[4] System Analysis Design (7th ed), Alan Dennis, Barbara Haley Wixom, David Tegarden, 2019.

[5] Pg . 85 System Analysis Design (7th ed), Alan Dennis, Barbara Haley Wixom, David Tegarden, 2019.

**** Scrum agile development sets all of it timeboxes to 30 working days.

Q5. Explain what are functional and nonfunctional requirements?

Functional Requirement:

Functional-requirement is the new system capabilities and function to support its users to do their tasks. It is written for system developers and in a natural language so that system users and managers can understand them. There are two types of functional requirements:

1. Process-Oriented: A process the system must perform or do.
2. Information-oriented: information the system must contain

Non-Functional Requirement:

Non – Functional requirement contains important behavioral properties that new system must have. These are requirements that are not directly concerned with the specific services delivered by the system to its users.

The characteristics of non-functional requirements:

1. **Operational:** the physical and technical environments in which the system will operate. Means that the system should be able to work on any web browser.
2. **Performance:** includes he speed, capacity, and reliability of the system
3. Security: for who authorized access to the system under what circumstances (why did they get authorized for).
4. **Cultural and Political:** factors that affect the system, should the system will be able to switch into different currencies.

Non-Functional requirements can further be broken down into three categories: Product Requirements, Organizational Requirements, and External Requirements.[6]

1. **Product Requirements:** specify of constrain the runtime behavior of the software. Like how fast the system must execute, how much memory it requires, reliability requirements, security requirements and usability requirements.
2. **Organizational Requirements:** These are broad system requirements derived from policies and procedures in the customer's and developer's organizations.
3. **External Requirements:** covers all requirements that are derived from factors external to the system and its development process.

[6] Pg. 109 Software Engineering 10e by Ian Sommerville

Q6. Using an example explain the differences between “include” and “extend” relationships between use case?

An include relationship represents the inclusion of the functionality of one-use case within another. It has an arrow drawn from the base use case to the used use case (←-----). Whereas, an extend relationship represents the extension of the use case to include optional behavior, it has an arrow drawn from the extension use case to the base use case. (----→)

Example of “extend” relationship:

Every time a patient makes an appointment, the patient is asked to verify their payment. But they can also make new payment arrangements, if they don’t have an existing payment. So, we can have a use case called MAKE PAYMENT ARRANGEMENTS that extends the MANAGE APPOINTMENTS use case. [7]

Example of “include” relationship:

So, if there is a use case called MANAGE SCHEDULE that performs routine tasks needed to maintain doctor’s office appointment schedules, that has two other shared use cases RECORD AVAILABILITY and PRODUCE SCHEDULE. So, to fulfill the two shared use cases, you use the “include” relationship, pointing towards MANAGE SCHEDULE use case.

[7] System Analysis Design (7th ed), Alan Dennis, Barbara Haley Wixom, David Tegarden, 2019.

Q7. What is an association class?

In UML Diagrams, an association class is a class that is part of an association relationship between two other classes.[8] An association class is identical to other classes and can contain operations, attributes, as well as other associations. An association class always has two or more and 1… many associations tot other classes.

For example, a class called Student represents a student and has an association with a class called Course. The Student can take the course. An association class is called take that further defines the relationship between the student and courses.

Association has two types: Aggregation and Composition

1. Aggregation – is a special type of association in which objects of one class is part of an object of the other class.
2. Composition – is a special type of aggregation in which objects of part class cannot exist independently; it is stronger relationship compared to aggregation. [9]

Q8. What are the main differences between domain modeling and behavioral modeling?

Domain Model

A domain model describes the **structures of the objects** that support the business processes in the problem. (no technical details) So, Domain model describes the concepts of the problem domain and their relationship: conceptual classes + relations

Behavioral Model

A behavioral model describes the **internal dynamic aspects** of an information system. The internal logic of the process without implementation details. And how the subset objects in the problem domain work together to form a collaboration to support each of the use cases. The behavioral model is more of in the object level. There are three UML Behavioral Diagrams: sequence diagram, communication diagram and behavioral state diagram.

---

[8] https://www.ibm.com/support/knowledgecenter/SSCLKU_7.5.5/com.ibm.xtools.modeler.doc/topics/cassnclss.html

[9] Dr. Roohollah's notes – Chapter 5 Analysis Domain Modeling.

A behavioral model is not used for all objects but for the most complex one to help simplify the algorithms for the methods.[10]

Q9. Explain what are the similarities and differences between a sequence diagram and communication diagram in the behavioral modeling?

A **sequence diagram** is a dynamic model that shows the explicit sequence of messages that are passes between objects in a defined interaction.

Communication diagrams are like sequence diagrams, it provides a view of the dynamic aspects of an object-oriented system. They show how the members of a set of objects collaborate to implement a use case or use-case scenario. Communication diagrams are very useful to show process patterns, and they are equivalent to sequence diagrams, but they also emphasize the flow of messages through a set to objects, but the sequence diagram focuses on the time ordering of the messages that is being passed.[11]

Both Sequence Diagrams and Communication Functions can be used to completely understand the dynamic activity of the system.

---

[10] Pg . 253 [10] System Analysis Design (7th ed), Alan Dennis, Barbara Haley Wixom, David Tegarden, 2019.

[11] Pg. 246 [11] Pg . 253 [11] System Analysis Design (7th ed), Alan Dennis, Barbara Haley Wixom, David Tegarden, 2019.