1. What are the types of system models?What are its perspectives?Explain

System modeling is the process of developing abstract models of a system, with each model presenting a different view or perspective of that system. It is about representing a system using some kind of graphical notation, which is now almost always based on notations in the Unified Modeling Language (UML). Models help the analyst to understand the functionality of the system; they are used to communicate with customers.

Models of both new and existing system are used during requirements engineering. Models of the existing systems help clarify what the existing system does and can be used as a basis for discussing its strengths and weaknesses. These then lead to requirements for the new system. Models of the new system are used during requirements engineering to help explain the proposed

requirements to other system stakeholders. Engineers use these models to discuss design proposals and to document the system for implementation.

Models can explain the system from different perspectives:

An external perspective, where you model the context or environment of the system.

An interaction perspective, where you model the interactions between a system and its environment, or between the components of a system.

A structural perspective, where you model the organization of a system or the structure of the data that is processed by the system.

A behavioral perspective, where you model the dynamic behavior of the system and how it responds to events.

The UML has many diagram types and so supports the creation of many different types of system model. Five types of UML diagrams that are the most useful for system modeling:

1. Activity diagrams, which show the activities involved in a process or in data processing.
2. Use case diagrams, which show the interactions between a system and its environment.
3. . Sequence diagrams, which show interactions between actors and the system and between system components.
4. Class diagrams, which show the object classes in the system and the associations between these classes.
5. State diagrams, which show how the system reacts to internal and external events.

2. Explain the following models with examples :

a. Context model

b. Data flow model

At an early stage in the specification of a system, you should decide on the system boundaries. This involves working with system stakeholders to decide what functionality should be included in the system and what is provided by the system's environment. These decisions should be made early in the process to limit the system costs and the time needed for understanding the system requirements and design.

Context models are used to illustrate the operational context of a system - they show what lies outside the system boundaries. Social and organizational concerns may affect the decision on where to position system boundaries. Architectural models show the system and its relationship with other systems. Context models normally show that the environment includes several other automated systems. However, they do not show the types of relationships between the systems in the environment and the system that is being specified.

System boundaries are established to define what is inside and what is outside the system. They show other systems that are used or depend on the system being developed. The position of the system boundary has a profound effect on the system requirements. Defining a system boundary is a political judgment since there may be pressures to develop system boundaries that increase/decrease the influence or workload of different parts of an organization.

In some cases, the boundary between a system and its environment is relatively clear. For example, where an automated system is replacing an existing manual or computerized system, the environment of the new system is usually the same as the existing system's environment.

For example, say you are developing the specification for the patient information system for mental healthcare. This system is intended to manage information about patients attending mental health clinics and the treatments that have been prescribed. In developing the specification for this system, you have to decide whether the system should focus exclusively on collecting information about consultations (using other systems to collect personal information about patients) or whether it should also collect personal patient information.

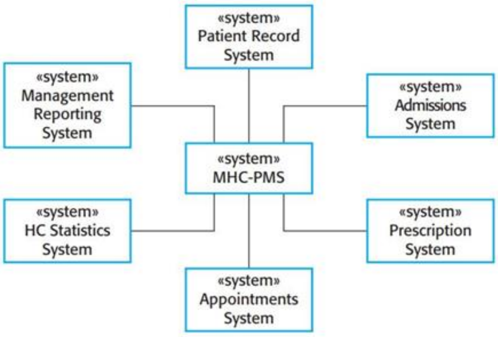
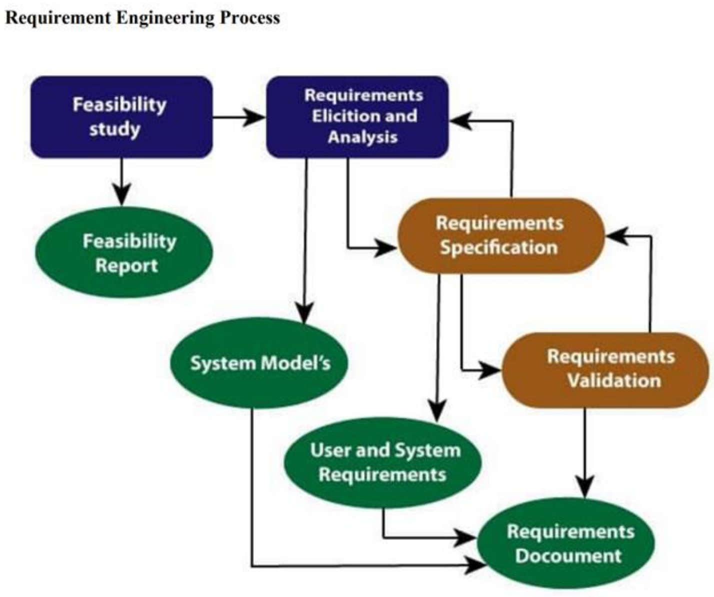


Figure 5.1 The context of the MHC-PMS

is a simple context model that shows the patient information system and the other systems in its environment. From Figure 5.1, you can see that the MHC-PMS is connected to an appointments system and a more general patient record system with which it shares data. The system is also connected to systems for management reporting and hospital bed allocation and a statistics system that collects information for research. Finally, it makes use of a prescription system to generate prescriptions for patients' medication.

Simple context models are used along with other models, such as business process models. These describe human and automated processes in which particular software systems are used.

3.Explain Requirement engineering process


Requirement Engineering Process
Feasibility study
Requirements Elicition and Analysis
Feasibility Report
Requirements Specification
System Model's
Requirements Validation
User and System Requirements
Requirements Doocument

The goal of the requirement engineering process is to create and maintain a system requirements document.The overall process includes four high-level requirements engineering subprocesses.These are concerned with assessing whether the system is useful to the business (feasibility study) ; discovering requirements (elicitation and analysis) ; converting these requirements into some standard form (specification) ; and checking that the requirements actually define the system that the customer wants (validation) .

These activities are concerned with the discovery,documentation and checking of requirements.The people involved develop a better understanding of what they want the software to do ; the organisation buying the system changes ; modifications are made to the system hardware,software and organisational environment.The process of managing these changing requirements is called requirement management .

Feasibility Study

For all new systems the requirement engineering process should start with a feasibility study.The input to the feasibility study is a set of preliminary business requirements an outline description of the system and how the system is intended to support business processes.The results of the feasibility study should be a report that recommends whether or not it is worth carrying on with the requirement engineering and system development process.If you are unsure whether your system solution will deliver the outcome you want,then a project feasibility study will help gain that clarity.During the feasibility study a variety of assessment methods are undertaken.The outcome of the feasibility study is a confirmed solution for implementation.

A feasibility study is a short focused study that aims to answer a number of questions:

1) Does the system contribute to the overall objectives of the organisation?
2) Can the system be implemented using current technology and within given cost and schedule constraints?
3) Can the system be integrated with other systems which are already in place?

Carrying out a feasibility study involves information assessment,information collection and report writing.

Requirement Elicitation and Analysis

After an initial feasibility study, the next stage of the requirements engineering process is requirements elicitation and analysis. In this activity, software engineers work with customers and system end-users to find out about the application domain,what services the system should provide, the required performance of the system,hardware constraints, and so on.

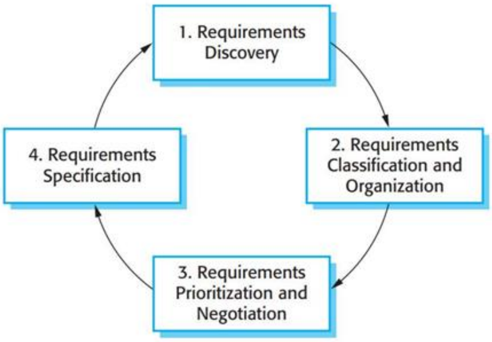

1. Requirements Discovery
2. Requirements Classification and Organization
3. Requirements Prioritization and Negotiation
4. Requirements Specification

Figure 4.13 The requirements elicitation and analysis process

Requirements elicitation and analysis may involve a variety of different kinds of people in an organization. A system stakeholder is anyone who should have some direct or indirect influence on the system requirements. Stakeholders include end users who will interact with the system and anyone else in an organization who willbe affected by it.

Requirements elicitation and analysis may involve a variety of different kinds of people in an organization. A system stakeholder is anyone who should have some direct or indirect influence on the system requirements. Stakeholders include end users who will interact with the system and anyone else in an organization who willbe affected by it.

The process activities are:

1. Requirements discovery - This is the process of interacting with stakeholders of the system to discover their requirements. Domain requirements from stakeholders and documentation are also discovered during this activity.
2. . Requirements classification and organization - This activity takes the unstructured collection of requirements, groups related requirements, and organizes them into coherent clusters.
3. Requirements prioritization and negotiation - Inevitably, when multiple stakeholders are involved, requirements will conflict. This activity is concerned with prioritizing requirements and finding and resolving requirements conflicts through negotiation. Usually, stakeholders have to meet to resolve differences and agree on compromise requirements.
4. Requirements specification - The requirements are documented and input into the next round of the spiral. Formal or informal requirements documents may be produced

Figure 4.13 shows that requirements elicitation and analysis is an iterative process with continual feedback from each activity to other activities. The process cycle starts with requirements discovery and ends with the requirements documentation. The analyst’s understanding of the requirements improves with each round of the cycle. The cycle ends when the requirements document is complete.

4. Explain Requirement validation. What are its techniques?

Requirements validation is the process of checking that requirements actually define the system that the customer really wants. Requirements validation is important because errors in a requirements document can lead to extensive rework costs when these problems are discovered during development or after the system is in service.

During the requirements validation process, different types of checks should be carried out on the requirements in the requirements document. These checks include:

1. Validity checks - A user may think that a system is needed to perform certain functions. However, further thought and analysis may identify additional or different functions that are required. Systems have diverse stakeholders with different needs and any set of requirements is inevitably a compromise across the stakeholder community.

2. Consistency checks - Requirements in the document should not conflict. That is, there should not be contradictory constraints or different descriptions of the same system function.

3. Completeness checks - The requirements document should include requirements that define all functions and the constraints intended by the system user.

4. Realism checks - Using knowledge of existing technology, the requirements should be checked to ensure that they can actually be implemented. These checks should also take account of the budget and schedule for the system development.

5. Verifiability - To reduce the potential for dispute between customer and contractor, system requirements should always be written so that they are verifiable. This means that you should be able to write a set of tests that can demonstrate that the delivered system meets each specified requirement.

There are a number of requirements validation techniques that can be used individually or in conjunction with one another:

1. Requirements reviews - The requirements are analyzed systematically by a team of reviewers who check for errors and inconsistencies.
2. Prototyping - In this approach to validation, an executable model of the system in question is demonstrated to end-users and customers. They can experiment with this model to see if it meets their real needs.
3. Test-case generation - Requirements should be testable. If the tests for the requirements are devised as part of the validation process, this often reveals requirements problems. If a test is difficult or impossible to design, this usually means that the requirements will be difficult to implement and should be reconsidered. Developing tests from the user requirements before any code is written is an integral part of extreme programming.