

Computers II Lesson 5

5.0 Requirements Discovery

Requirements discovery sometimes called requirements elicitation is the process of gathering information about the required system and existing systems, and distilling the user and system requirements from this information.

Sources of information during the requirements discovery phase include:

- Documentation
- System stakeholders
- Specifications of similar systems
- You interacting with stakeholders through interviews and observation
- Scenarios and prototypes to help stakeholders understand what the system will be like.

Stakeholders:

- End-users of a system
- Managers
- Regulators who certify the acceptability of the system.

For example, system stakeholders for the mental healthcare patient information system include:

1. Patients whose information is recorded in the system.
2. Doctors who are responsible for assessing and treating patients.
3. Nurses who coordinate the consultations with doctors and administer some treatments.
4. Medical receptionists who manage patients' appointments.
5. IT staff who are responsible for installing and maintaining the system.

6. A medical ethics manager who must ensure that the system meets current ethical guidelines for patient care.

7. Healthcare managers who obtain management information from the system.

8. Medical records staff that are responsible for ensuring that system information can be maintained and preserved, and that record keeping procedures have been properly implemented.

Use cases are a requirements discovery technique.

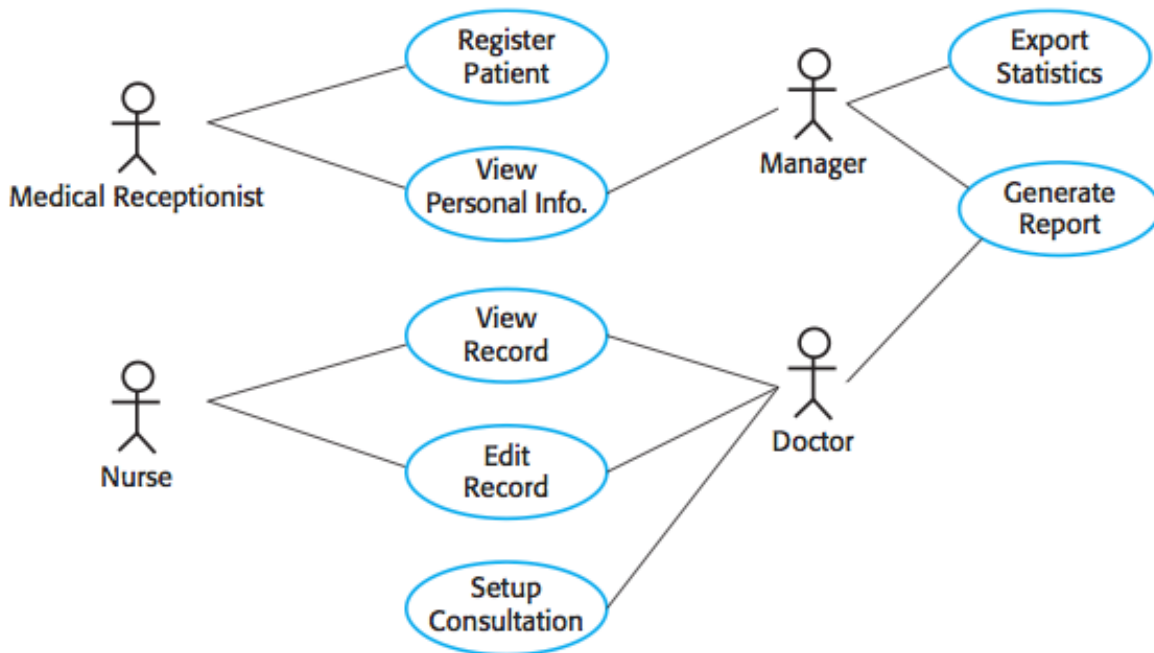
Each use case should be documented with a textual description.

A use case identifies:

- The actors involved in an interaction
- The type of interaction.

This is then supplemented by additional information describing the interaction with the system. The additional information may be a textual description or one or more graphical models such as UML sequence or state charts.

- Actors in the process, who may be human or other systems, are represented as stick figures.
- Each class of interaction is represented as a named ellipse.
- Lines link the actors with the interaction.



5.1 Interviewing

Formal or informal interviews with system stakeholders are part of most requirements engineering processes.

In these interviews, the requirements engineering team puts questions to stakeholders about the system that they currently use and the system to be developed.

Requirements are derived from the answers to these questions. Interviews may be of two types, but are generally a mixture of both types:

1. Closed interviews, where the stakeholder answers a predefined set of questions.
2. Open interviews, in which there is no predefined agenda. The requirements engineering team explores a range of issues with system stakeholders and hence develop a better understanding of their needs.

Interviews are good for getting an overall understanding of what stakeholders do, how they might interact with the new system, and the difficulties that they face with current systems. People like talking about their work so are usually happy to get involved in interviews.

Issues with Interviews:

1. All application specialists use terminology and jargon that are specific to a domain. It is impossible for them to discuss domain requirements without using this terminology. They normally use terminology in a precise and subtle way that is easy for requirements engineers to misunderstand.
2. Some domain knowledge is so familiar to stakeholders that they either find it difficult to explain or they think it is so fundamental that it isn't worth mentioning. For example, for a librarian, it goes without saying that all acquisitions are catalogued before they are added to the library. However, this may not be obvious to the interviewer, and so it isn't taken into account in the requirements.

Effective interviewer characteristics:

1. They are open-minded
2. Avoid pre-conceived ideas about the requirements
3. Willing to listen to stakeholders
4. Willing to change their mind about the system
5. Prompt the interviewee to get discussions going using a springboard question, a requirements proposal, or by working together on a prototype system.
6. Saying to people 'tell me what you want' is unlikely to result in useful information. They find it much easier to talk in a defined context rather than in general terms.

5.2 Scenarios

People usually find it easier to relate to real-life examples rather than abstract descriptions. They can understand and criticize a scenario of how they might interact with a software system.

Scenarios can be particularly useful for adding detail to an outline requirements description. They are descriptions of example interaction sessions. Each scenario usually covers one or a small number of possible interactions.

- A scenario starts with an outline of the interaction.

A scenario may include:

1. A description of what the system and users expects when the scenario starts.
2. A description of the normal flow of events in the scenario.
3. A description of what can go wrong and how this is handled.
4. Information about other activities that might be going on at the same time.
5. A description of the system state when the scenario finishes.

Business, organizational, and technical changes inevitably lead to changes to the requirements for a software system. Requirements management is the process of managing and controlling these changes.

Once a system has been installed and is regularly used, requirements inevitably emerge. End-users have experience of a system; they will discover new needs and priorities.

There are several reasons why change is inevitable:

- 1.The business and technical environment of the system always changes after installation. New hardware may be introduced, it may be necessary to interface the system with other systems, business priorities may change (with consequent changes in the system support required), and new legislation and regulations may be introduced that the system must necessarily abide by.
- 2.The people who pay for a system and the users of that system are rarely the same people. System customers impose requirements because of organizational and budgetary constraints. These may conflict with end-user requirements and, after delivery; new features may have to be added for user support if the system is to meet its goals.
- 3.Large systems usually have a diverse user community, with many users having different requirements and priorities that may be conflicting or contradictory. The final system requirements are inevitably a compromise between them and, with experience, it is often discovered that the balance of support given to different users has to be changed.