DA-IICT

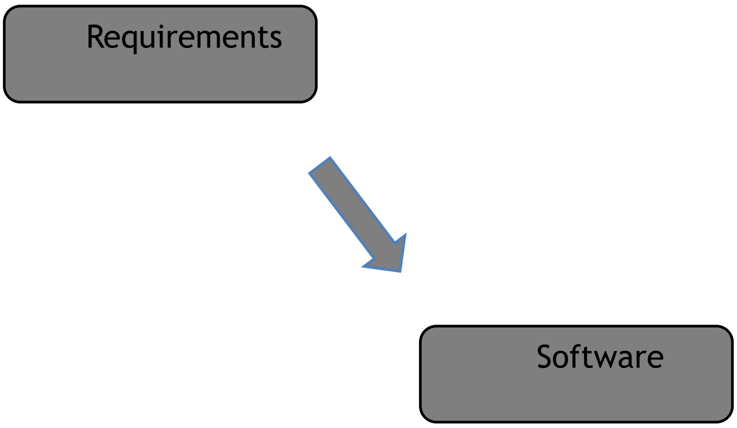
IT 314: Software Engineering

Software Requirements

"I'll go talk to the stakeholders and find out their requirements... in the meantime, you guys start coding."

Saurabh Tiwari

How to Start?



Requirements

Analysis

Design (Analysis + Design)

Code

Test

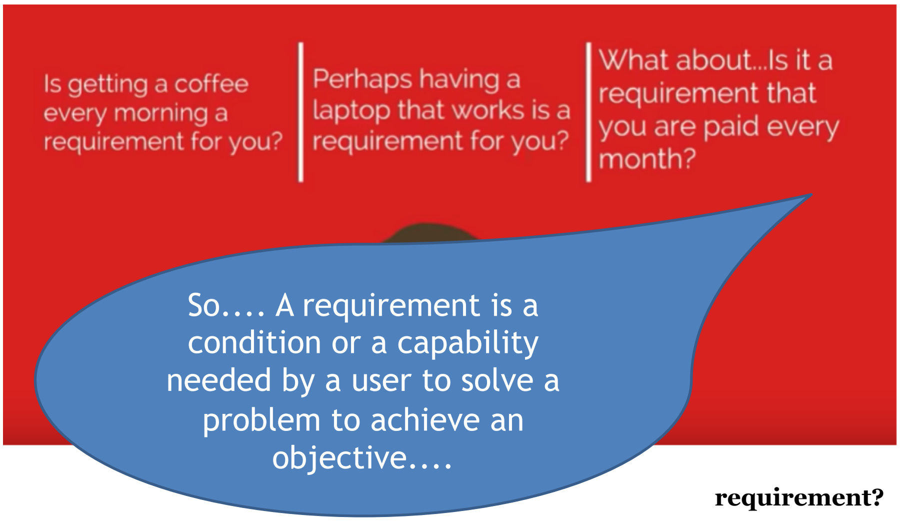
Deploy

Maintenance

Requirements Engineering
(Elicitation/gathering;
Requirements Analysis;
Requirements Specification/documentation;
Requirements Management/Traceability;
Requirements Validation [System Testing])

Analysis – Analysis of the problem domain

Design – System Design



Is getting a coffee every morning a requirement for you?

Perhaps having a laptop that works is a requirement for you?

What about...Is it a requirement that you are paid every month?

So.... A requirement is a condition or a capability needed by a user to solve a problem to achieve an objective....

requirement?

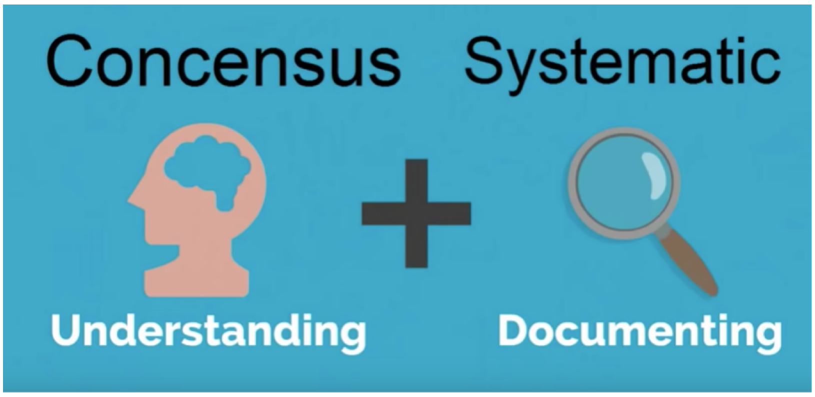
What is a requirement?

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

- *May be the basis for a bid for a contract - therefore must be open to interpretation*
- *May be the basis for the contract itself - therefore must be defined in detail*
- *Both these statements may be called requirements*

What is Requirements Engineering?



Requirements Engineering

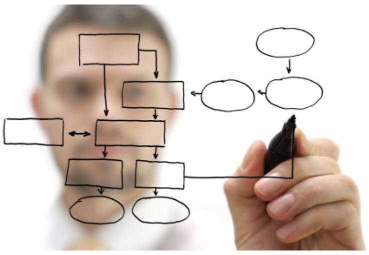
- The process of establishing the services that the customer requires from a system and the constraints under which it operates and is developed
- The requirements themselves are the descriptions of the system services and constraints that are generated during the requirements engineering process

RE: What is this Phase For?

- Major misconception
  - determining what client wants

"I know you believe you understood what you think I said, but I am not sure you realize that what you heard is not what I meant!"

- Must determine client's & user's needs
- Problems that needs to be sorted out during RE
  - Incompleteness
  - Ambiguity
  - Inconsistency



Requirements Abstraction

"If a company wishes to let a contract for a large software development project, it must define its needs in a sufficiently abstract way that a solution is not pre-defined. The requirements must be written so that several contractors can bid for the contract, offering, perhaps, different ways of meeting the client organization's needs. Once a contract has been awarded, the contractor must write a system definition for the client in more detail so that the client understands and can validate what the software will do. Both of these documents may be called the requirements document for the system."

- Davis

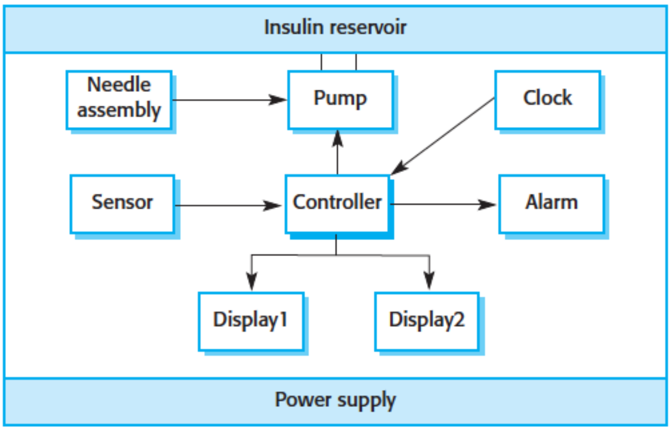
Case Studies

- A personal insulin pump
  - An embedded system in an insulin pump used by diabetics to maintain blood glucose control.
- A mental health case patient management system
  - A system used to maintain records of people receiving care for mental health problems.
- A wilderness weather station
  - A data collection system that collects data about weather conditions in remote areas.

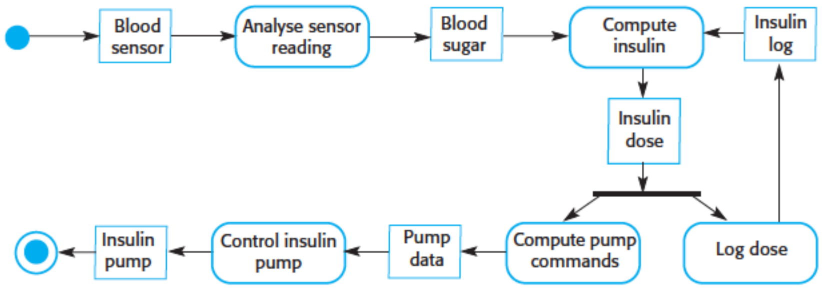
Insulin Pump Control System

- Collects data from a blood sugar sensor and calculates the amount of insulin required to be injected.
- Calculation based on the rate of change of blood sugar levels.
- Sends signals to a micro-pump to deliver the correct dose of insulin.
- Safety-critical system as low blood sugars can lead to brain malfunctioning, coma and death; high-blood sugar levels have long-term consequences such as eye and kidney damage.

Insulin pump hardware architecture



Activity model of the insulin pump



Essential high-level requirements

- The system shall be available to deliver insulin when required.
- The system shall perform reliably and deliver the correct amount of insulin to counteract the current level of blood sugar.
- The system must therefore be designed and implemented to ensure that the system always meets these requirements.

A patient information system for mental health care

- A patient information system to support mental health care is a medical information system that maintains information about patients suffering from mental health problems and the treatments that they have received.
- Most mental health patients do not require dedicated hospital treatment but need to attend specialist clinics regularly where they can meet a doctor who has detailed knowledge of their problems.
- To make it easier for patients to attend, these clinics are not just run in hospitals. They may also be held in local medical practices or community centres.

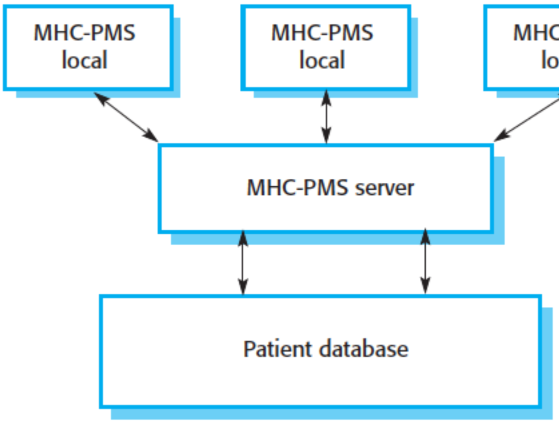
MHC-PMS

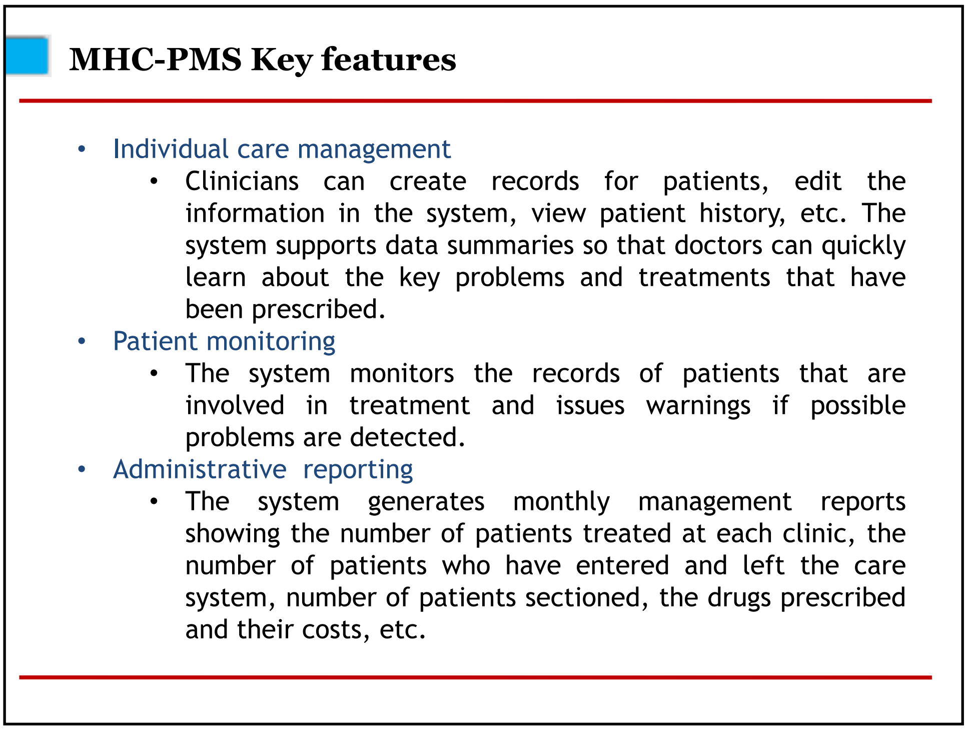
- The MHC-PMS (Mental Health Care-Patient Management System) is an information system that is intended for use in clinics.
- It makes use of a centralized database of patient information but has also been designed to run on a PC, so that it may be accessed and used from sites that do not have secure network connectivity.
- When the local systems have secure network access, they use patient information in the database but they can download and use local copies of patient records when they are disconnected.

MHC-PMS goals

- To generate management information that allows health service managers to assess performance against local and government targets.
- To provide medical staff with timely information to support the treatment of patients.

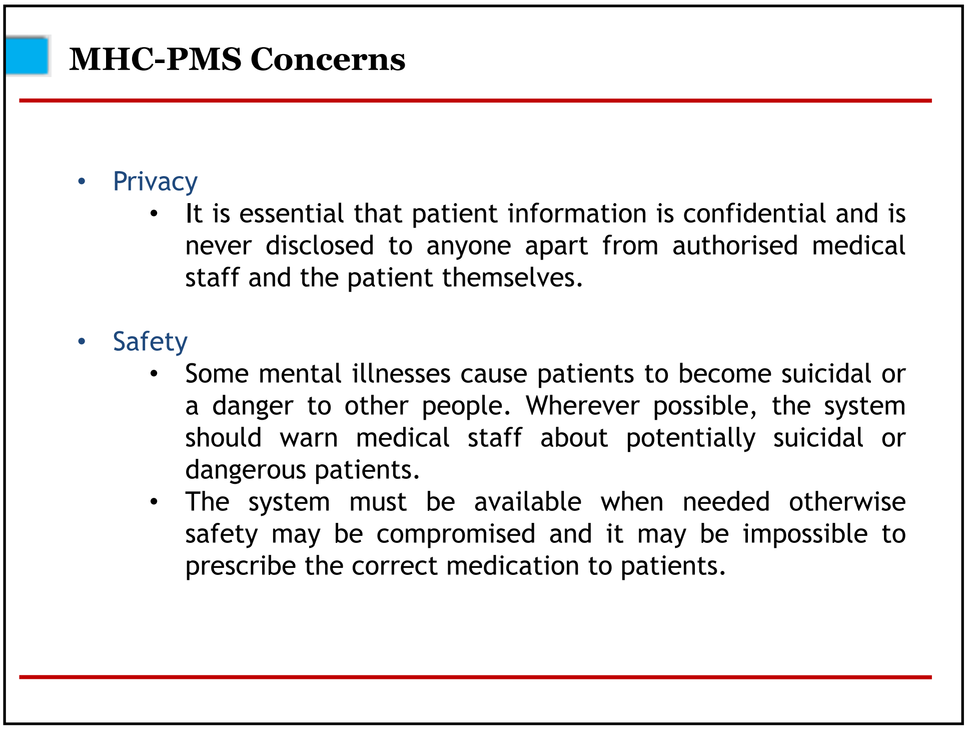
The organization of the MHC-PMS





MHC-PMS Key features

- Individual care management
  - Clinicians can create records for patients, edit the information in the system, view patient history, etc. The system supports data summaries so that doctors can quickly learn about the key problems and treatments that have been prescribed.
- Patient monitoring
  - The system monitors the records of patients that are involved in treatment and issues warnings if possible problems are detected.
- Administrative reporting
  - The system generates monthly management reports showing the number of patients treated at each clinic, the number of patients who have entered and left the care system, number of patients sectioned, the drugs prescribed and their costs, etc.



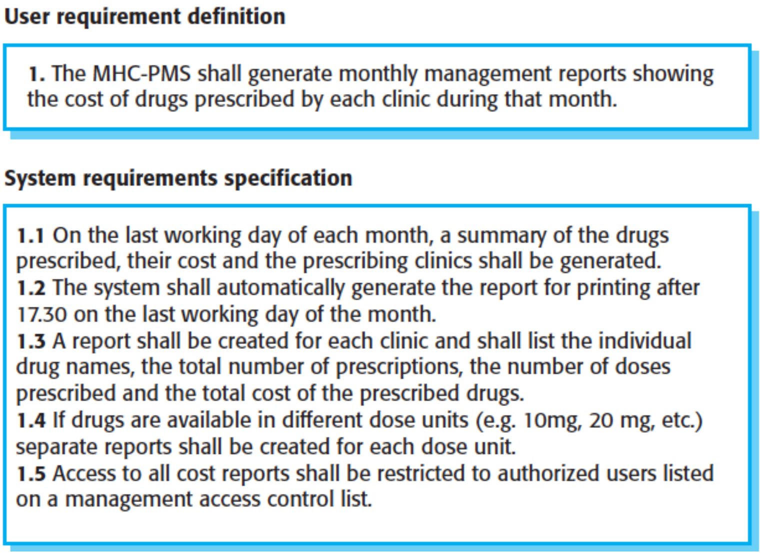
MHC-PMS Concerns

- Privacy
  - It is essential that patient information is confidential and is never disclosed to anyone apart from authorised medical staff and the patient themselves.
- Safety
  - Some mental illnesses cause patients to become suicidal or a danger to other people. Wherever possible, the system should warn medical staff about potentially suicidal or dangerous patients.
  - The system must be available when needed otherwise safety may be compromised and it may be impossible to prescribe the correct medication to patients.

Types of requirement

- User requirements
  - Statements in natural language plus diagrams of the services the system provides and its operational constraints. Written for customers
- System requirements
  - A structured document setting out detailed descriptions of the system services. Written as a contract between client and contractor
- Software specification
  - A detailed software description which can serve as a basis for a design or implementation. Written for developers

User and system requirements



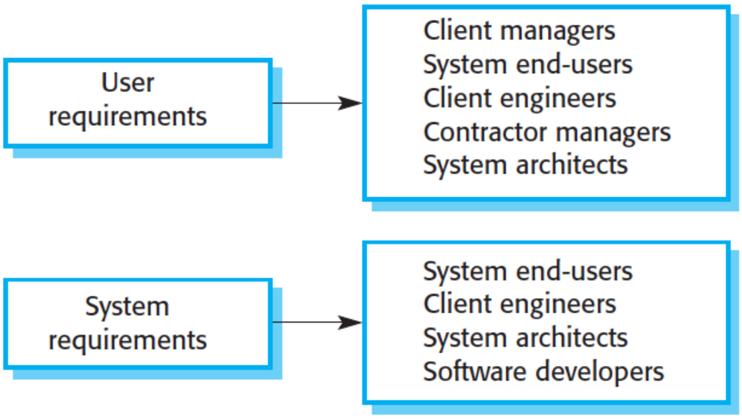
User requirement definition

1. The MHC-PMS shall generate monthly management reports showing the cost of drugs prescribed by each clinic during that month.

System requirements specification

1.1 On the last working day of each month, a summary of the drugs prescribed, their cost and the prescribing clinics shall be generated.
1.2 The system shall automatically generate the report for printing after 17.30 on the last working day of the month.
1.3 A report shall be created for each clinic and shall list the individual drug names, the total number of prescriptions, the number of doses prescribed and the total cost of the prescribed drugs.
1.4 If drugs are available in different dose units (e.g. 10mg, 20 mg, etc.) separate reports shall be created for each dose unit.
1.5 Access to all cost reports shall be restricted to authorized users listed on a management access control list.

Readers of different types of requirements specification



User requirements

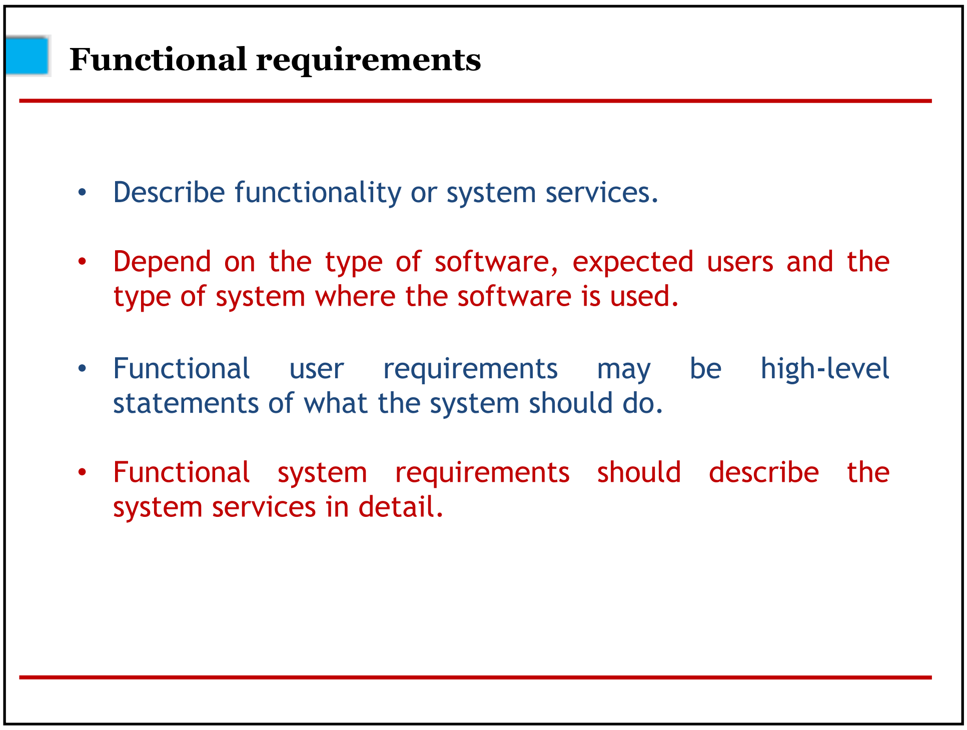
Client managers
System end-users
Client engineers
Contractor managers
System architects

System requirements

System end-users
Client engineers
System architects
Software developers

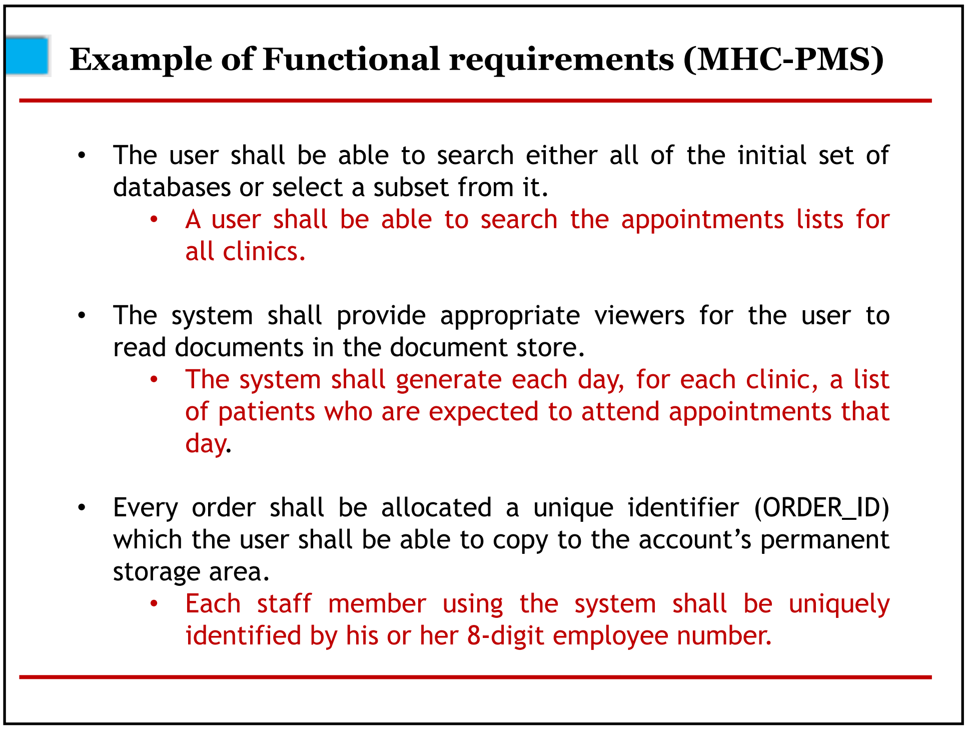
Functional & non-functional requirements

- Functional requirements
  - Statements of services the system should provide, how the system should react to particular inputs and how the system should behave in particular situations.
  - May state what the system should not do.
- Non-functional requirements
  - Constraints on the services or functions offered by the system such as timing constraints, constraints on the development process, standards, etc.
  - Often apply to the system as a whole rather than individual features or services.
- Domain requirements
  - Constraints on the system from the domain of operation



Functional requirements

- 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 detail.



Example of Functional requirements (MHC-PMS)

- The user shall be able to search either all of the initial set of databases or select a subset from it.
  - A user shall be able to search the appointments lists for all clinics.
- The system shall provide appropriate viewers for the user to read documents in the document store.
  - The system shall generate each day, for each clinic, a list of patients who are expected to attend appointments that day.
- Every order shall be allocated a unique identifier (ORDER_ID) which the user shall be able to copy to the account's permanent storage area.
  - Each staff member using the system shall be uniquely identified by his or her 8-digit employee number.

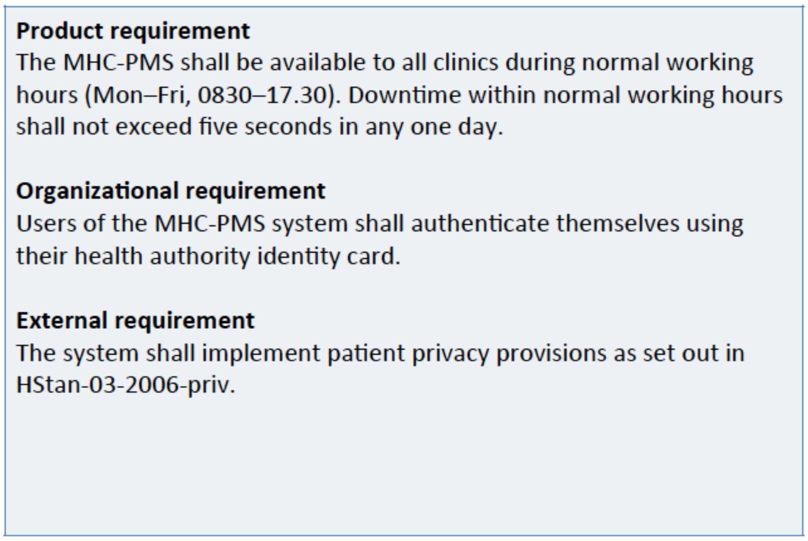
Requirements imprecision

- Problems arise when requirements are not precisely stated
- Ambiguous requirements may be interpreted in different ways by developers and users
- Consider the term ‘appropriate viewers’
  - *User intention - special purpose viewer for each different document type*
  - *Developer interpretation - Provide a text viewer that shows the contents of the document*

Requirements completeness and consistency

- In principle requirements should be both complete and consistent
- Complete
  - They should include descriptions of all facilities required
- Consistent
  - There should be no conflicts or contradictions in the descriptions of the system facilities
- In practice, it is impossible to produce a complete and consistent requirements document

Examples of NFR in the MHC-PMS



Product requirement
The MHC-PMS shall be available to all clinics during normal working hours (Mon–Fri, 0830–17.30). Downtime within normal working hours shall not exceed five seconds in any one day.

Organizational requirement
Users of the MHC-PMS system shall authenticate themselves using their health authority identity card.

External requirement
The system shall implement patient privacy provisions as set out in HStan-03-2006-priv.

Goals and requirements

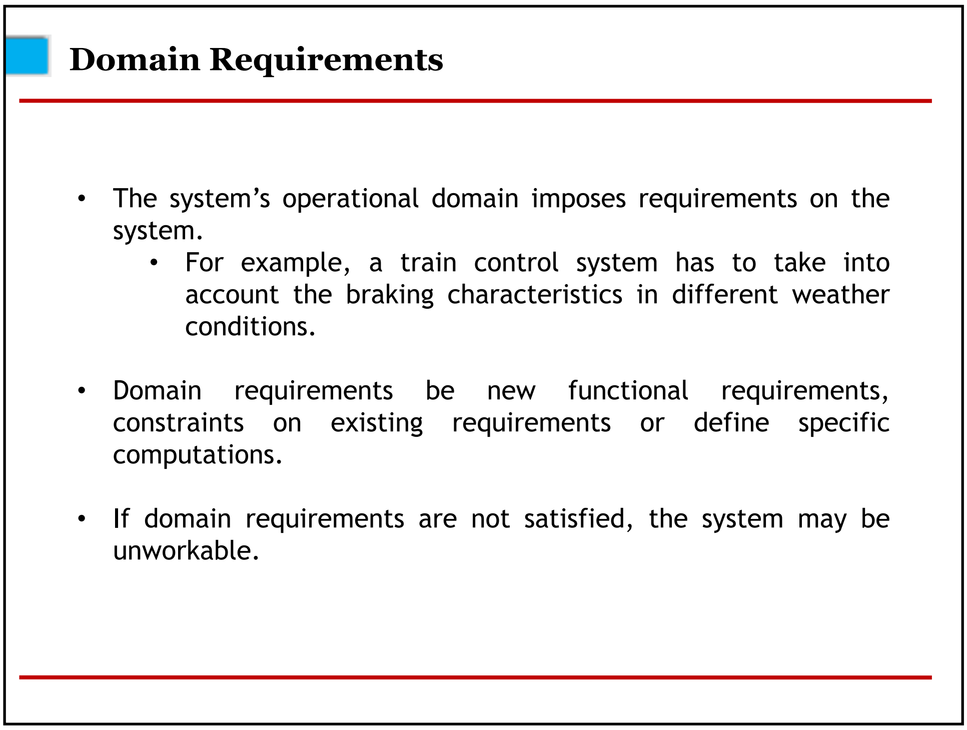
- Non-functional requirements may be very difficult to state precisely and imprecise requirements may be difficult to verify.
- **Goal**
  - A general intention of the user such as ease of use
- **Verifiable non-functional requirement**
  - A statement using some measure that can be objectively tested
- Goals are helpful to developers as they convey the intentions of the system users

Examples

- A system goal
  - The system should be easy to use by experienced controllers and should be organised in such a way that user errors are minimised.
- A verifiable non-functional requirement
  - Experienced controllers shall be able to use all the system functions after a total of two hours training. After this training, the average number of errors made by experienced users shall not exceed two per day.

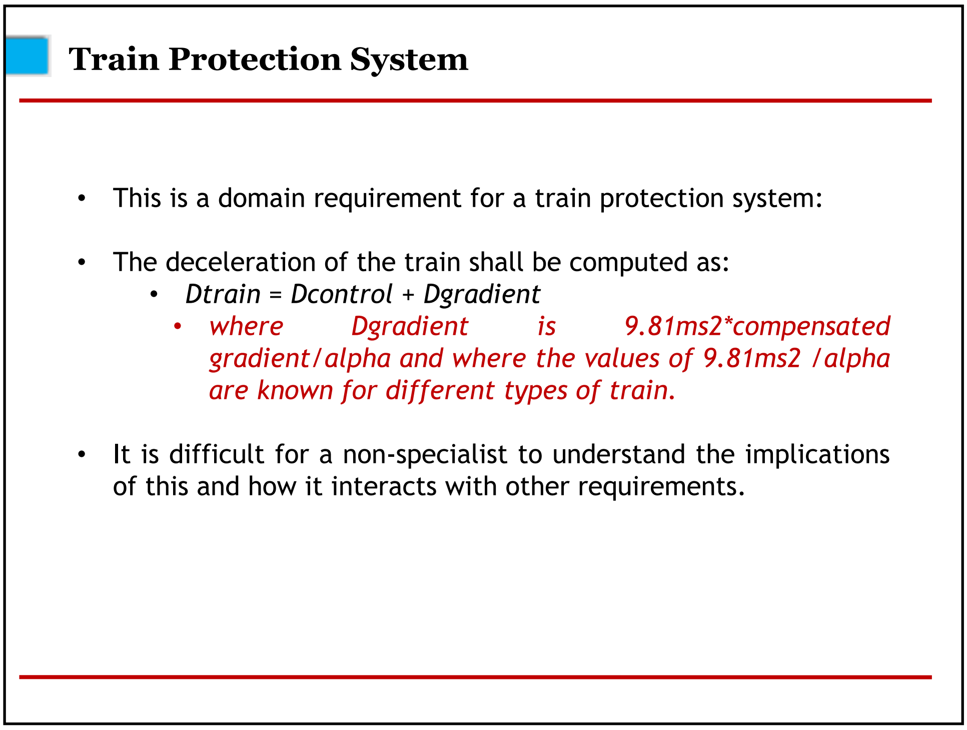
Metrics for specifying NFR

Property	Measure
Speed	Processed transactions/second User/event response time Screen refresh time
Size	Mbytes Number of ROM chips
Ease of use	Training time Number of help frames
Reliability	Mean time to failure Probability of unavailability Rate of failure occurrence Availability
Robustness	Time to restart after failure Percentage of events causing failure Probability of data corruption on failure
Portability	Percentage of target dependent statements Number of target systems



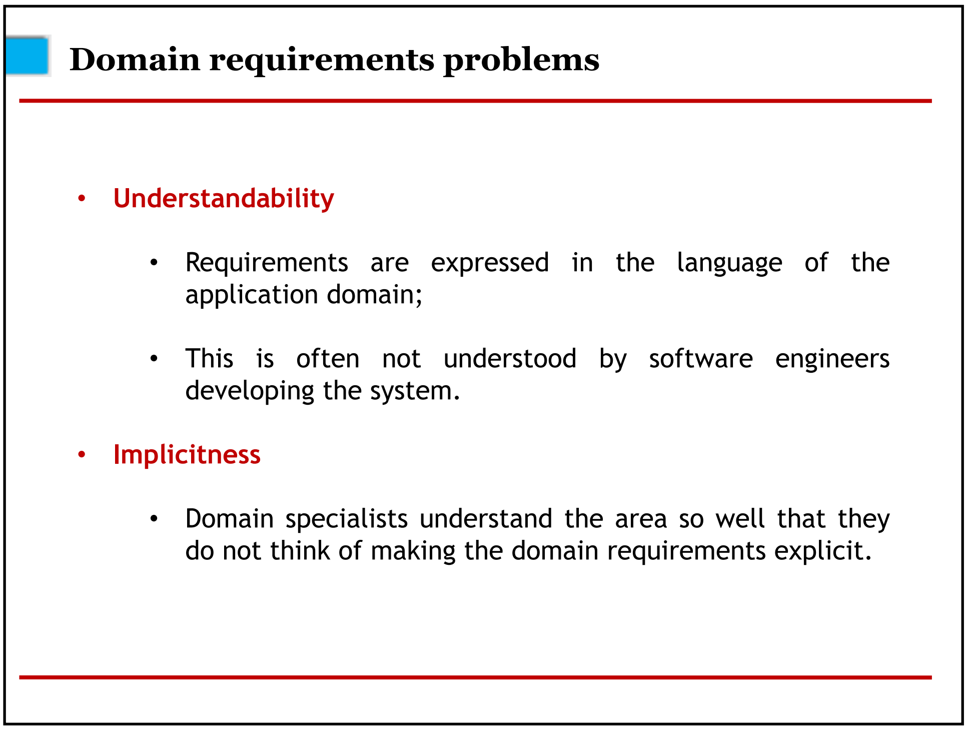
Domain Requirements

- 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.



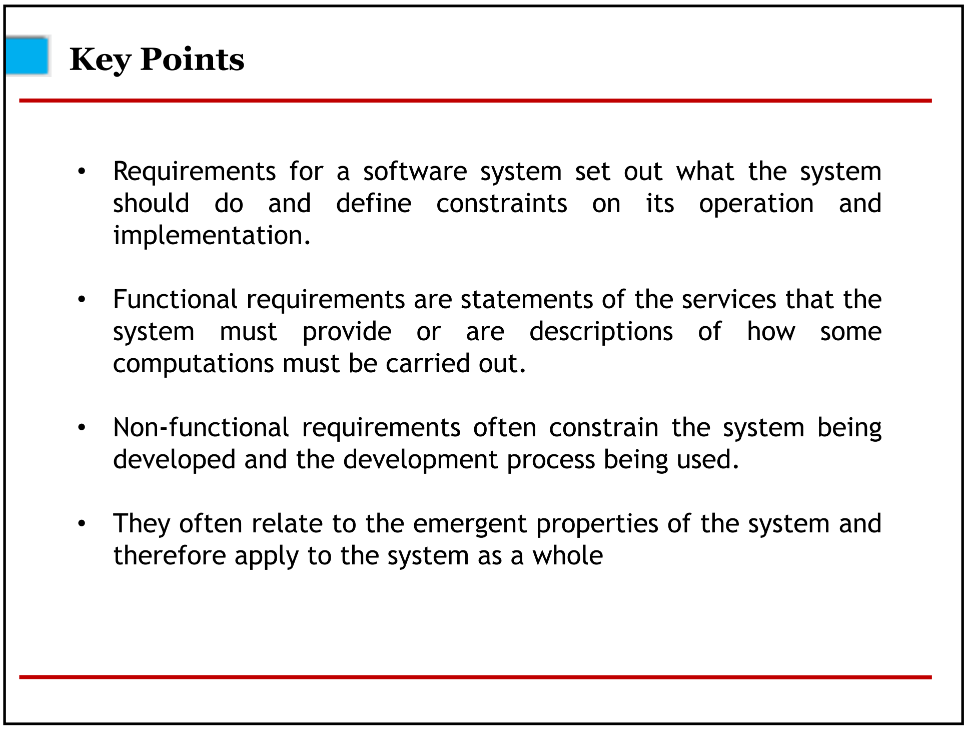
Train Protection System

- This is a domain requirement for a train protection system:
- The deceleration of the train shall be computed as:
  - *Dtrain = Dcontrol + Dgradient*
    - *where Dgradient is 9.81ms2*compensated gradient/alpha and where the values of 9.81ms2 /alpha are known for different types of train.*
- It is difficult for a non-specialist to understand the implications of this and how it interacts with other requirements.



Domain requirements problems

- **Understandability**
  - Requirements are expressed in the language of the application domain;
  - This is often not understood by software engineers developing the system.
- **Implicitness**
  - Domain specialists understand the area so well that they do not think of making the domain requirements explicit.



Key Points

- Requirements for a software system set out what the system should do and define constraints on its operation and implementation.
- Functional requirements are statements of the services that the system must provide or are descriptions of how some computations must be carried out.
- Non-functional requirements often constrain the system being developed and the development process being used.
- They often relate to the emergent properties of the system and therefore apply to the system as a whole

Example

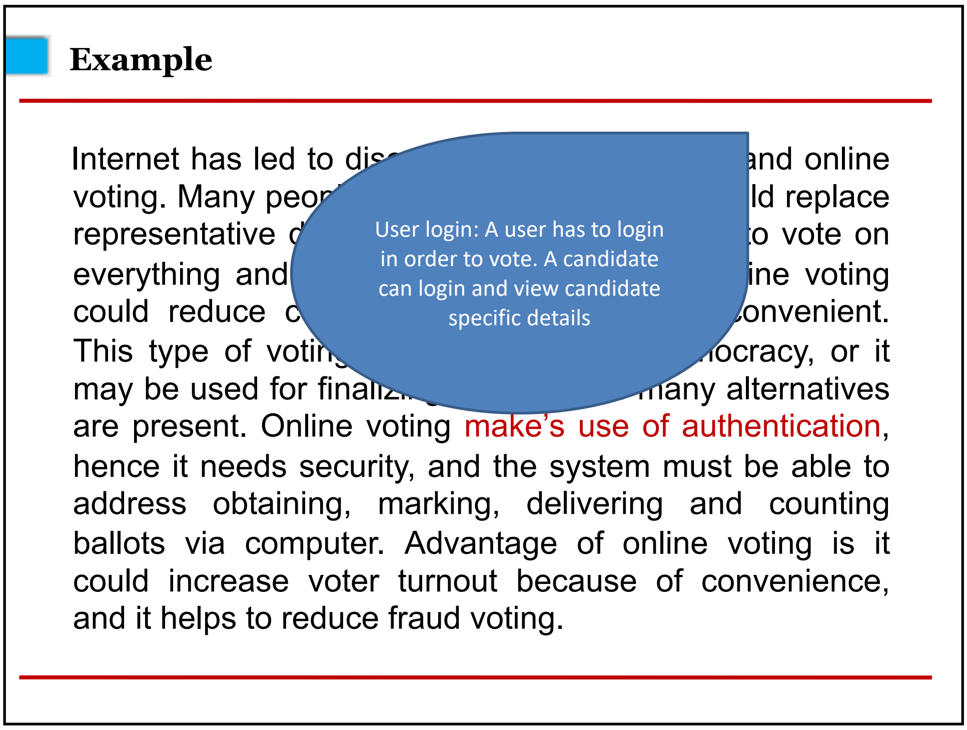
We show here how to extract functional requirements when a problem statement is given. The case under study is a online voting system.

Internet has led to discussion of e-democracy and online voting. Many peoples think that the internet could replace representative democracy, enabling everyone to vote on everything and anything by online voting .Online voting could reduce cost and make voting more convenient. This type of voting can be done for e-democracy, or it may be used for finalizing a solution, if many alternatives are present. Online voting make's use of authentication, hence it needs security, and the system must be able to address obtaining, marking, delivering and counting ballots via computer. Advantage of online voting is it could increase voter turnout because of convenience, and it helps to reduce fraud voting.

Example

Internet has led to dis and online
voting. Many peop ld replace
representative d o vote on
everything and ine voting
could reduce c onvenient.
This type of votin nocracy, or it
may be used for finaliz many alternatives
are present. Online voting make's use of authentication, hence it needs security, and the system must be able to address obtaining, marking, delivering and counting ballots via computer. Advantage of online voting is it could increase voter turnout because of convenience, and it helps to reduce fraud voting.

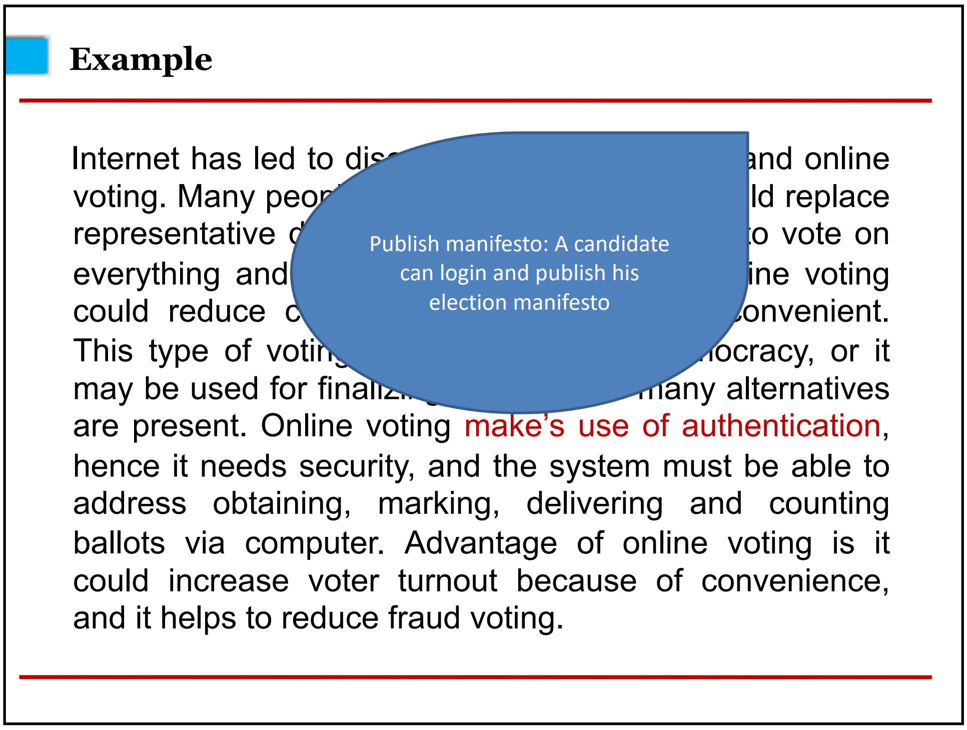
User registration: A candidate or a user wishes to vote online has to register with the system by providing his details



Example

Internet has led to dis and online
voting. Many peop ld replace
representative d o vote on
everything and ine voting
could reduce c onvenient.
This type of voti ocracy, or it
may be used for finaliz many alternatives
are present. Online voting make's use of authentication, hence it needs security, and the system must be able to address obtaining, marking, delivering and counting ballots via computer. Advantage of online voting is it could increase voter turnout because of convenience, and it helps to reduce fraud voting.

User login: A user has to login in order to vote. A candidate can login and view candidate specific details



Example

Internet has led to dis and online
voting. Many peop ld replace
representative d o vote on
everything and ine voting
could reduce c onvenient.
This type of voti ocracy, or it
may be used for finaliz many alternatives
are present. Online voting make's use of authentication, hence it needs security, and the system must be able to address obtaining, marking, delivering and counting ballots via computer. Advantage of online voting is it could increase voter turnout because of convenience, and it helps to reduce fraud voting.

Publish manifesto: A candidate can login and publish his election manifesto

Example

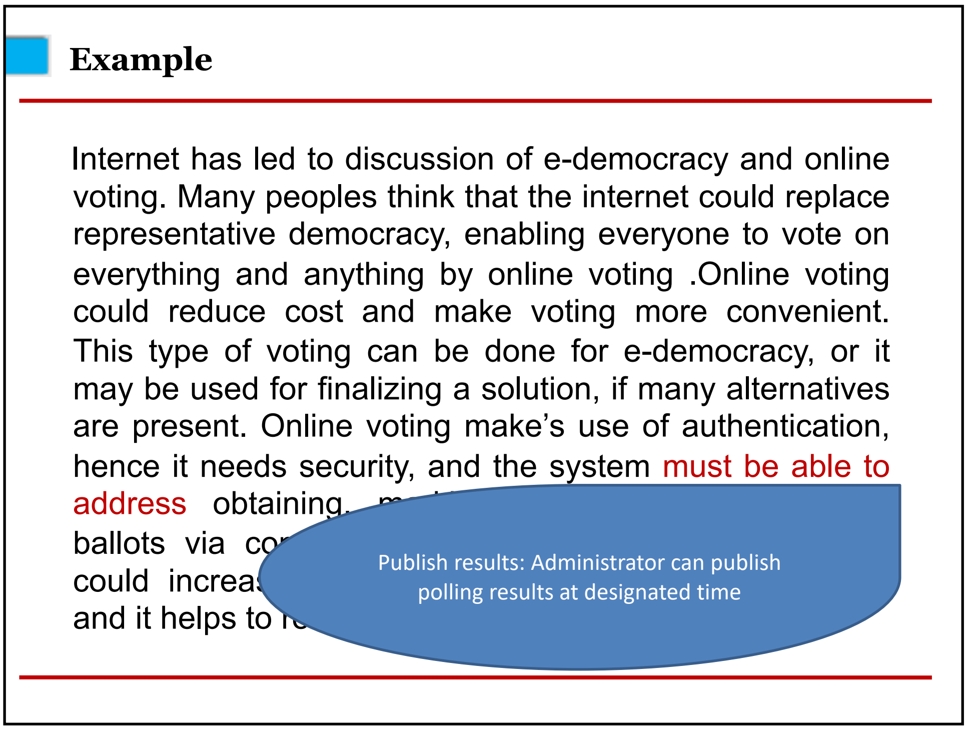
Internet has led to discussion of e-democracy and online voting. Many peoples think that the internet could replace representative democracy, enabling everyone to vote on everything and anything by online voting .Online voting could reduce cost and make voting more convenient. This type of voting can be done for e-democracy, or it may be used for finalizing a solution, if many alternatives are present. Online voting make's use of authentication, hence it needs security, and the system must be able to address obtaining, m
ballots via co
could increas
and it helps to r

Vote: A user can cast his vote in favor of a particular candidate

Example

Internet has led to discussion of e-democracy and online voting. Many peoples think that the internet could replace representative democracy, enabling everyone to vote on everything and anything by online voting .Online voting could reduce cost and make voting more convenient. This type of voting can be done for e-democracy, or it may be used for finalizing a solution, if many alternatives are present. Online voting make's use of authentication, hence it needs security, and the system must be able to address obtaining, m
ballots via co
could increas
and it helps to r

Count votes: System must be able to count votes received by each candidate based on polling



Example

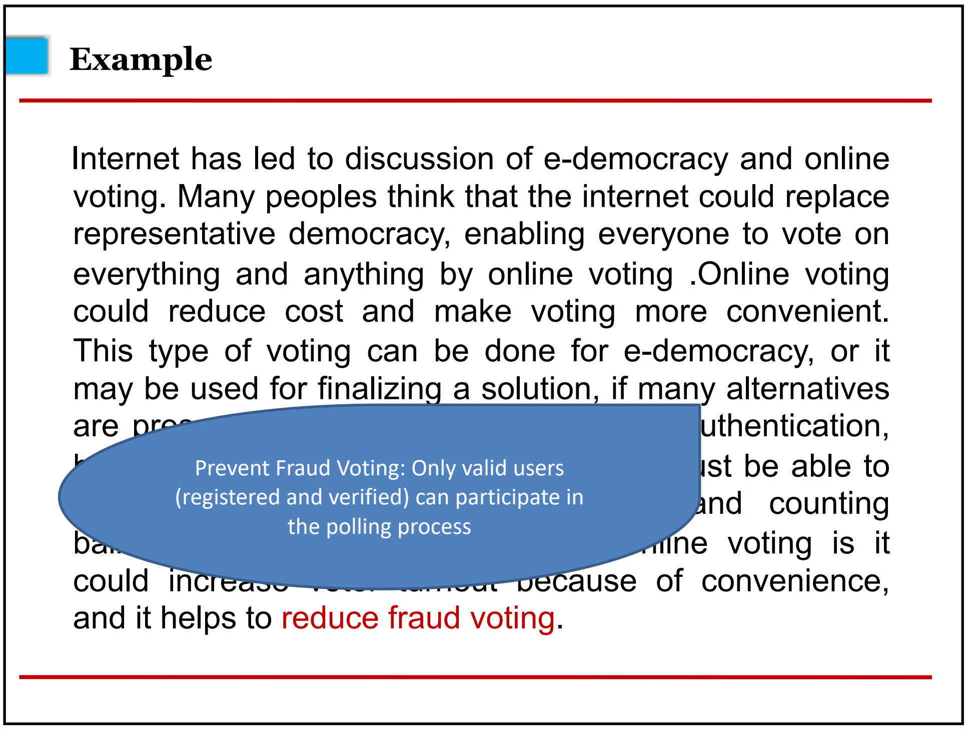
Internet has led to discussion of e-democracy and online voting. Many peoples think that the internet could replace representative democracy, enabling everyone to vote on everything and anything by online voting .Online voting could reduce cost and make voting more convenient. This type of voting can be done for e-democracy, or it may be used for finalizing a solution, if many alternatives are present. Online voting make's use of authentication, hence it needs security, and the system must be able to address obtaining, m

ballots via co

could increas

and it helps to r

Publish results: Administrator can publish polling results at designated time



Example

Internet has led to discussion of e-democracy and online voting. Many peoples think that the internet could replace representative democracy, enabling everyone to vote on everything and anything by online voting .Online voting could reduce cost and make voting more convenient. This type of voting can be done for e-democracy, or it may be used for finalizing a solution, if many alternatives are pre

uthentication,

ust be able to

and counting

ba

nline voting is it

could increase voter turnout because of convenience, and it helps to reduce fraud voting.

Prevent Fraud Voting: Only valid users (registered and verified) can participate in the polling process

Example

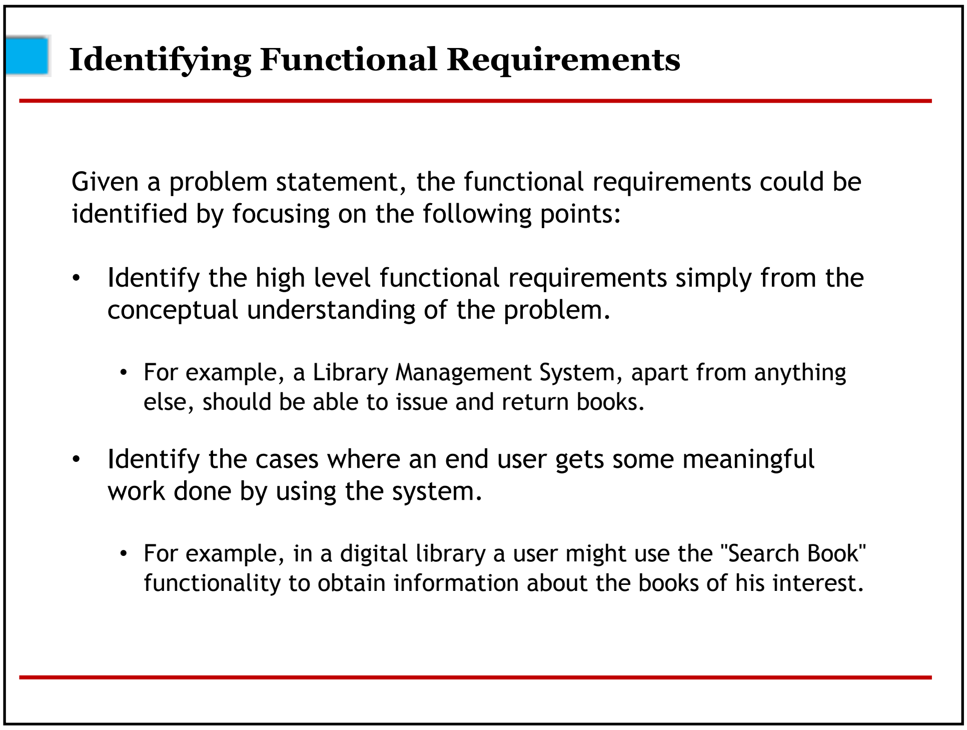
Internet has led to discussion of e-democracy and online voting. Many peoples think that the internet could replace representative democracy, enabling everyone to vote on everything and anything by online voting .Online voting could reduce cost and make voting more convenient. This type of voting can be done for e-democracy, or it may be used for finalizing a solution, if many alternatives are pre uthentication, ust be able to and counting ba nline voting is it could increase because of convenience, and it helps to reduce fraud voting.

Only one vote: A valid user can vote only once

Example

Internet has led to discussion of e-democracy and online voting. Many peoples think that the internet could replace representative democracy, enabling everyone to vote on everything and anything by online voting .Online voting could reduce cost and make voting more convenient. This type of voting can be done for e-democracy, or it may be used for finalizing a solution, if many alternatives are present. Online voting make's use of authentication, hence it needs security, and the system must be able to address obtaining, marking, delivering and counting ballots via computer. Advantage of online voting is it could increase voter turnout because of convenience, and it helps to reduce fraud voting.

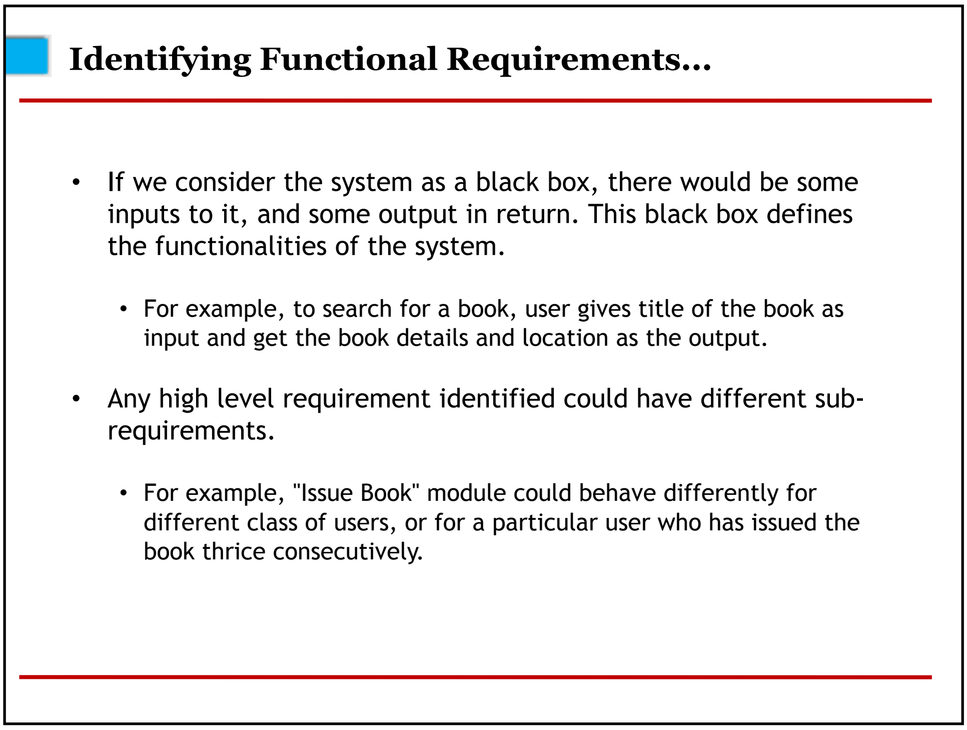
There could be others like email notifications, error handling and so on. Similarly, one has to identify the non functional requirements also. For example, the system must remain accessible to thousands of users at a time.



Identifying Functional Requirements

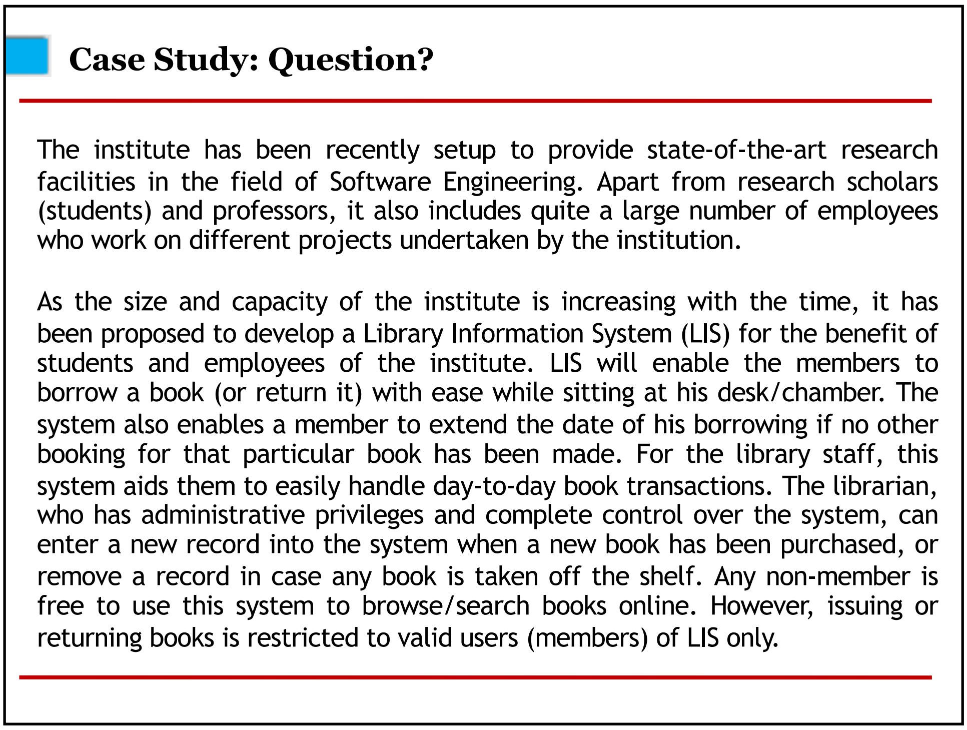
Given a problem statement, the functional requirements could be identified by focusing on the following points:

- Identify the high level functional requirements simply from the conceptual understanding of the problem.
  - For example, a Library Management System, apart from anything else, should be able to issue and return books.
- Identify the cases where an end user gets some meaningful work done by using the system.
  - For example, in a digital library a user might use the "Search Book" functionality to obtain information about the books of his interest.



Identifying Functional Requirements...

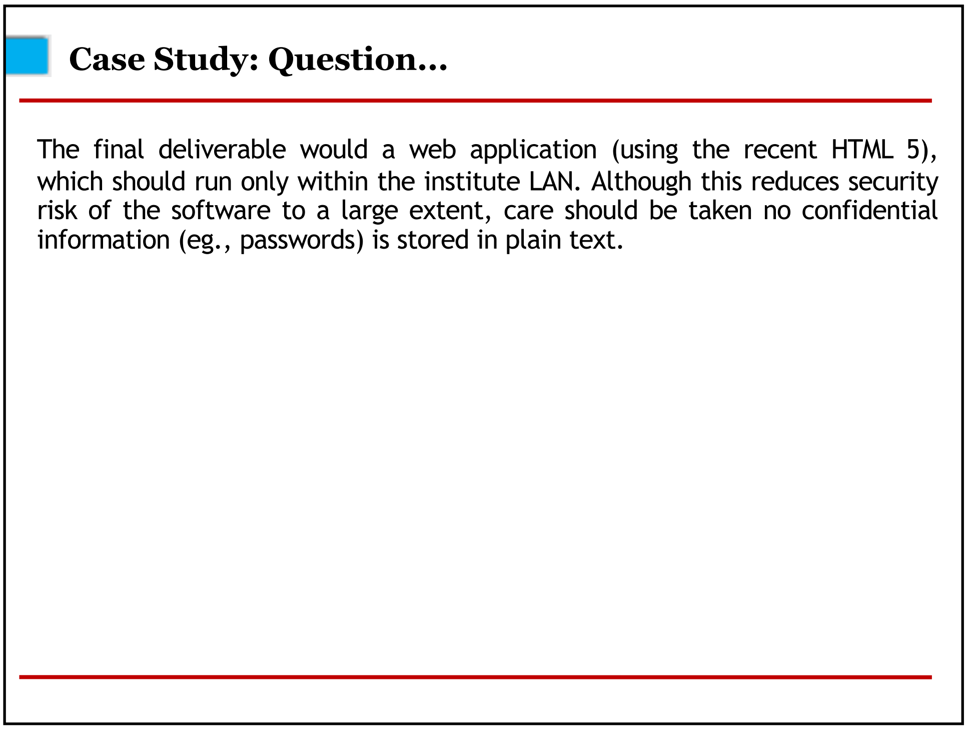
- If we consider the system as a black box, there would be some inputs to it, and some output in return. This black box defines the functionalities of the system.
  - For example, to search for a book, user gives title of the book as input and get the book details and location as the output.
- Any high level requirement identified could have different sub-requirements.
  - For example, "Issue Book" module could behave differently for different class of users, or for a particular user who has issued the book thrice consecutively.



Case Study: Question?

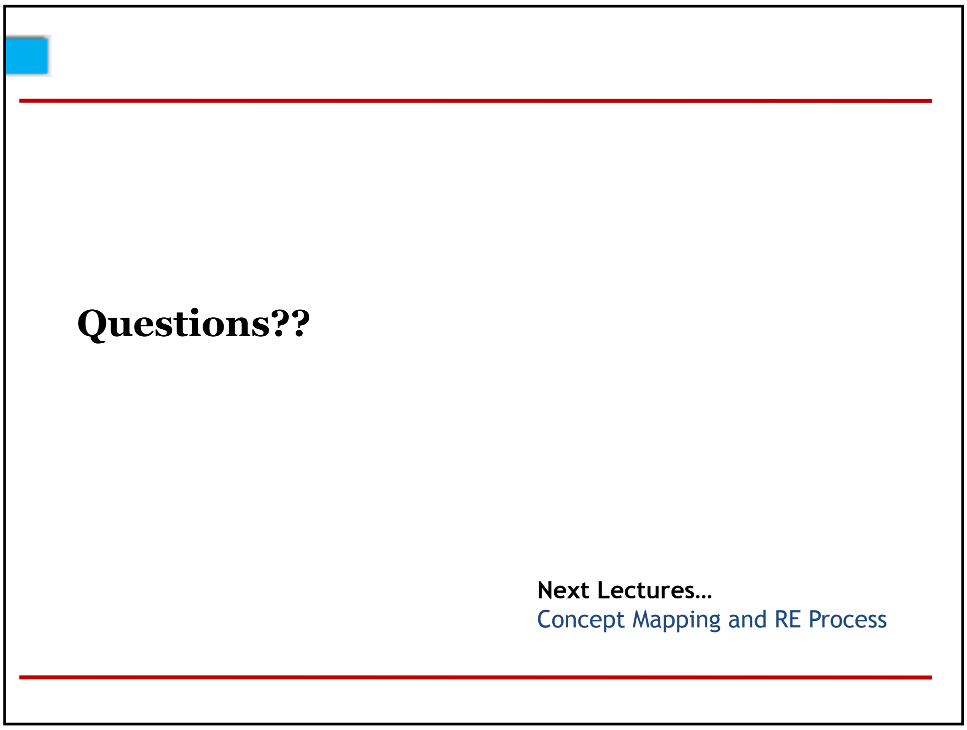
The institute has been recently setup to provide state-of-the-art research facilities in the field of Software Engineering. Apart from research scholars (students) and professors, it also includes quite a large number of employees who work on different projects undertaken by the institution.

As the size and capacity of the institute is increasing with the time, it has been proposed to develop a Library Information System (LIS) for the benefit of students and employees of the institute. LIS will enable the members to borrow a book (or return it) with ease while sitting at his desk/chamber. The system also enables a member to extend the date of his borrowing if no other booking for that particular book has been made. For the library staff, this system aids them to easily handle day-to-day book transactions. The librarian, who has administrative privileges and complete control over the system, can enter a new record into the system when a new book has been purchased, or remove a record in case any book is taken off the shelf. Any non-member is free to use this system to browse/search books online. However, issuing or returning books is restricted to valid users (members) of LIS only.



Case Study: Question...

The final deliverable would a web application (using the recent HTML 5), which should run only within the institute LAN. Although this reduces security risk of the software to a large extent, care should be taken no confidential information (eg., passwords) is stored in plain text.



Questions??

Next Lectures…
Concept Mapping and RE Process