

# Web Application Development

**Introduction to Software Engineering Concepts**

**Indian Institute of Information Technology  
Sri City, Chittoor**



# Today's Agenda

- Need for Software Development
- Introduction to Software and Application
- Fundamental Activities of Software Processes
- The Problem Searching
- Challenges in Web application development
- Requirement Specification
- User requirement and System requirements
- Case Study: MentCare system
- Functional and non-functional requirements
- State completion criteria

# Need for Software Development

- With the increasing complexity of systems at the place of work or in day-to-day needs.
- Current human centred work procedures are costly and inefficient.
- Information quickly goes out-of-date and manual entry data are often only available as hardcopy.
- This results in an extreme amount of wasted paper and other costs associated.

# Introduction to Software and Application

- *Software*- Software is a set of instructions or data that operates the hardware.
  - Software does not need user interaction for functioning.
  - Software is used as mediator between user and hardware.
  - All the software are not applications.
- *Application*- Application is a type of software that does a certain task.
  - Application need user interaction for functioning.
  - Application is used only by end users.
  - All the application are software.
- A web application (or web app) is application software that runs on a web server, unlike computer-based software programs that are run locally on the operating system (OS) of the device.
- Web applications are accessed by the user through a web browser with an active internet connection.

# Fundamental Activities of Software Processes

- Four fundamental activities are common to all software processes.
  1. **Software specification**, where customers and engineers define the software that is to be produced and the constraints on its operation.
  2. **Software development**, where the software is designed and programmed.
  3. **Software validation**, where the software is checked to ensure that it is what the customer requires.
  4. **Software evolution**, where the software is modified to reflect changing customer and market requirements.

# The Problem Searching

- Begin by choosing a significant real world problem that your audience can relate to.
- Show a concrete example of the chosen problem and need for the solutions.

- Defining the Problem:

Describe the problem background, need for the solution and how the proposed solution impacts.

This will serve as a "hook" to get the attention of your audience and convince them of the importance of the project.

- Next you want to focus in on the part of the problem you want to attack.
- So follow your example by focusing on a small but significant portion of the problem. This is the part that you are proposing to tackle.

# Problem searching and defining its scope: example

Title: Generating e-Pass to monitor the COVID symptoms and make safe work environment.

- Background: Many of the organisations do not have tracking about the essential on-site employees health status in this pandemic. This will add the burden to the organisation if any employee got tested positive and leads to the epicentre of the disease. So, apart from checking the temperature at the on-site (which is not enough). If the employees are allowed to self-monitor the symptoms and report it via web portal and the data is available to organisation, then the organisation can allow the symptomatic employee to either work-from-home or give him a time off until he becomes symptom free.

What we want or proposed solution requirement: A mobile or web application based e-pass portal, where employee assess his symptoms everyday before he comes to on-site and the assessed the data is also available to HR to keep track the employees.

- Category: Software/Web Application

# Problem searching and defining its scope: example

Title: Innovative online tool for protection of Child Rights

Problem Statement:

Build an app for tracking and monitoring status of children that stay in government recognized Child Care Institutions (CCI) and Specialized Adoption Agencies (SAA). Design an efficient biometric tracking tool that tracks daily attendance, as well as in and out movement of children from the respective institutions. Additionally, also tracks guardian visits in the system. Design a centralized system that enables tracking attendance and guardian visit across India. Ensure that this system also works in an offline manner.

For example, if database is synced once then it should allow marking of attendance and guardian visit without having to sync the database. However, the attendance should be uploaded automatically once internet connection is available. The monthly grant will be sent only if the data is synced. Provide login to District Child Protection Unit (DCPU) and Child Welfare Committees (CWC) to enable them to access the tracking information. Provide facility to transfer child from one organization to other seamlessly without losing the past tracking information. Depending on the attendance and number of visits by guardian, provide recommendation on adoption eligibility of the child based on JJ Act of 2015 and Adoption Guidelines of 2017. Based on the recommendation, flag the children with higher eligibility to be moved to the legal adoption pool. Currently, a lot of eligible children are not being moved to the legal adoption pool and hence, their right to good family is being violated.

- Organisation: Ministry of Women and Child Development

Credits: Smart India Hackathon 2020



# Problem searching and defining its scope: example

Problem statement: “Automate government working system of billing”- example- Measurement book in the department for any contract which is presently filled manually by JE and then SDO , accounts, E.E and then to the finance department, which takes approx. 2 months of time. To get this manual work through the system & make it easier to complete in 3-4 days.

- Organisation: Ministry of Micro, Small & Medium Enterprises

Credits: Smart India Hackathon 2020

# Problem searching and defining its scope: example

Problem statement: E –Marketplace (Like Amazon, flipkart) wherein tribals can promote, market and sell tribal produce such as handicrafts, arts, paintings, minor forest products etc. on line with provision of delivery and e-payment and promotional discounts.

- Organisation: Ministry of Tribal Affairs

Credits: Smart India Hackathon 2020

# Challenges in Web application development

1. Clearly defining of goals of the application.
  - Who are your intended users?
  - What experience do you want to give them?
  - What are your must-have design features?
  - What are your technical requirements?
2. Choosing the right tech stack.
3. User Experience (UX)
4. User Interface and Design:
  - Intuitive UI typically involves:
  - Clear navigation
  - Engaging visuals
  - Easy-to-read typography
5. Performance and Speed
6. Scalability
7. Security Threats

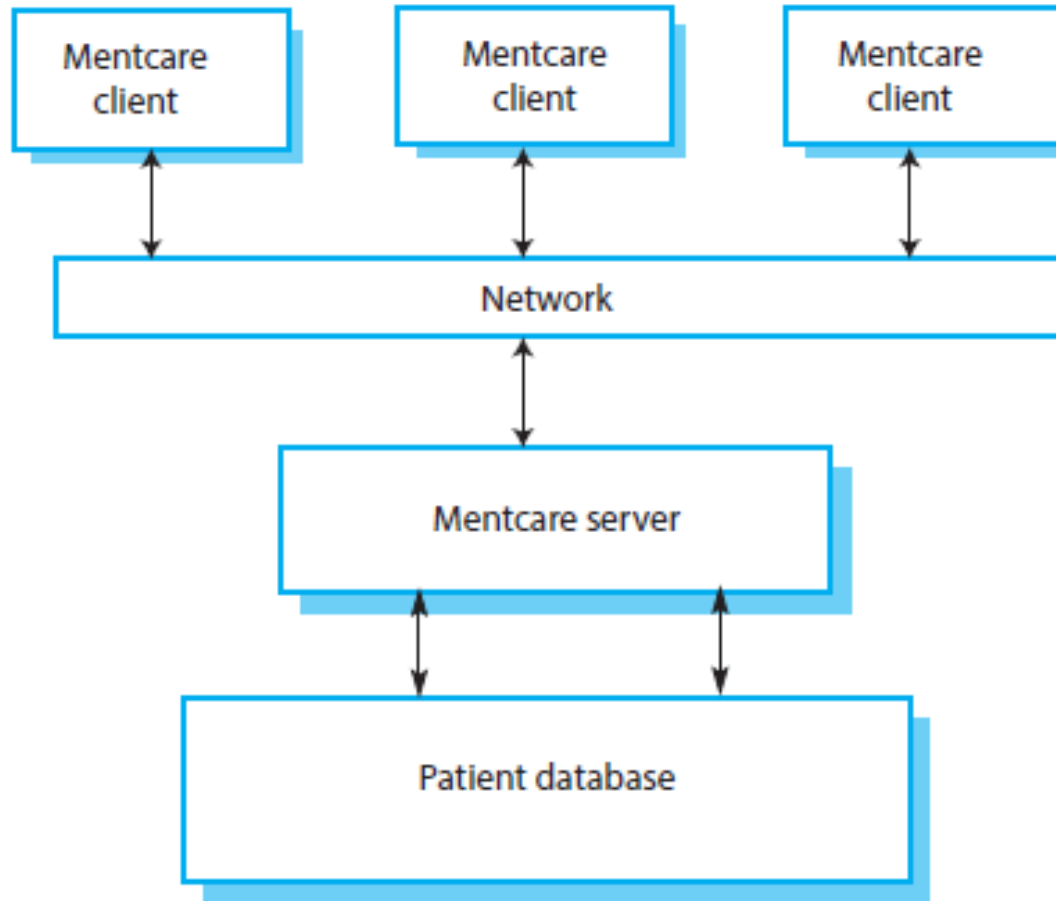
# Requirement Specification

- The requirements for a system are the descriptions of the services that a system should provide and the constraints on its operation.
- These requirements reflect the needs of customers for a system that serves a certain purpose such as controlling a device, placing an order, or finding information.
- The process of finding out, analysing, documenting and checking these services and constraints is called requirements engineering (RE).

# User requirement and System requirements

1. User requirements are statements, in a natural language plus diagrams, of what services the system is expected to provide to system users and the constraints under which it must operate.
  - The user requirements may vary from broad statements of the system features required to detailed, precise descriptions of the system functionality.
2. System requirements are more detailed descriptions of the software system's functions, services, and operational constraints.
  - The system requirements document (sometimes called a functional specification) should define exactly what is to be implemented. It may be part of the contract between the system buyer and the software developers.

# Case Study: MentCare System



Block diagram of “A patient information system for mental health care”.

# Case Study: MentCare System

## User and System Requirements

### User requirements definition

- 1.** The Mentcare system 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 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, 20mg, etc.) separate reports shall be created for each dose unit.
- 1.5** Access to drug cost reports shall be restricted to authorized users as listed on a management access control list.

# Functional and non-functional requirements

Software system requirements are often classified as functional or non-functional requirements.

- ***Functional requirements:*** These are statements of services the system should provide, how the system should react to particular inputs, and how the system should behave in particular situations.
  - In some cases, the functional requirements may also explicitly state what the system should not do.
- ***Non-functional requirements:*** These are constraints on the services or functions offered by the system. They include timing constraints, constraints on the development process, and constraints imposed by standards.
  - Non-functional requirements often apply to the system as a whole rather than individual system features or services.



# Functional and non-functional requirements

- In reality, the distinction between different types of requirements is not as clear-cut as these simple definitions suggest.

For example:

“A user requirement concerned with security in MentCare system, such as a statement limiting access to authorized users, may appear to be a non-functional requirement.”

However, when developed in more detail, this requirement may generate other requirements that are clearly functional, such as the need to include user authentication facilities in the system.

# Functional requirements

- The functional requirements for a system describe what the system should do.
- These requirements depend on the type of software being developed, the expected users of the software, and the general approach taken by the organization when writing requirements.
- When expressed as user requirements, functional requirements should be written in natural language so that system users and managers can understand them.
- Functional system requirements expand the user requirements and are written for system developers.
- They should describe the system functions, their inputs and outputs, and exceptions in detail.

# Functional requirements

- Ideally, the functional requirements specification of a system should be both complete and consistent.
  - Completeness means that all services and information required by the user should be defined.
  - Consistency means that requirements should not be contradictory.

Note: “In practice, it is only possible to achieve requirements consistency and completeness for very small software systems.”

*Reason: Large/complex systems involves lot of stakeholders and it is to make mistakes and omissions while preparing. Also in some cases some of the stake holders have different/inconsistent needs.*

# Functional requirements: MentCare system as a Case Study

- A user shall be able to search the appointments lists for all clinics.
- The system shall generate each day, for each clinic, a list of patients who are expected to attend appointments that day.
- Each staff member using the system shall be uniquely identified by his or her eight-digit employee number.
- **Information requirements** specify the information needed and how it is to be delivered and organized.
  - Therefore, an information requirement for the Mentcare system might specify what information is to be included in the list of patients expected for appointments that day.

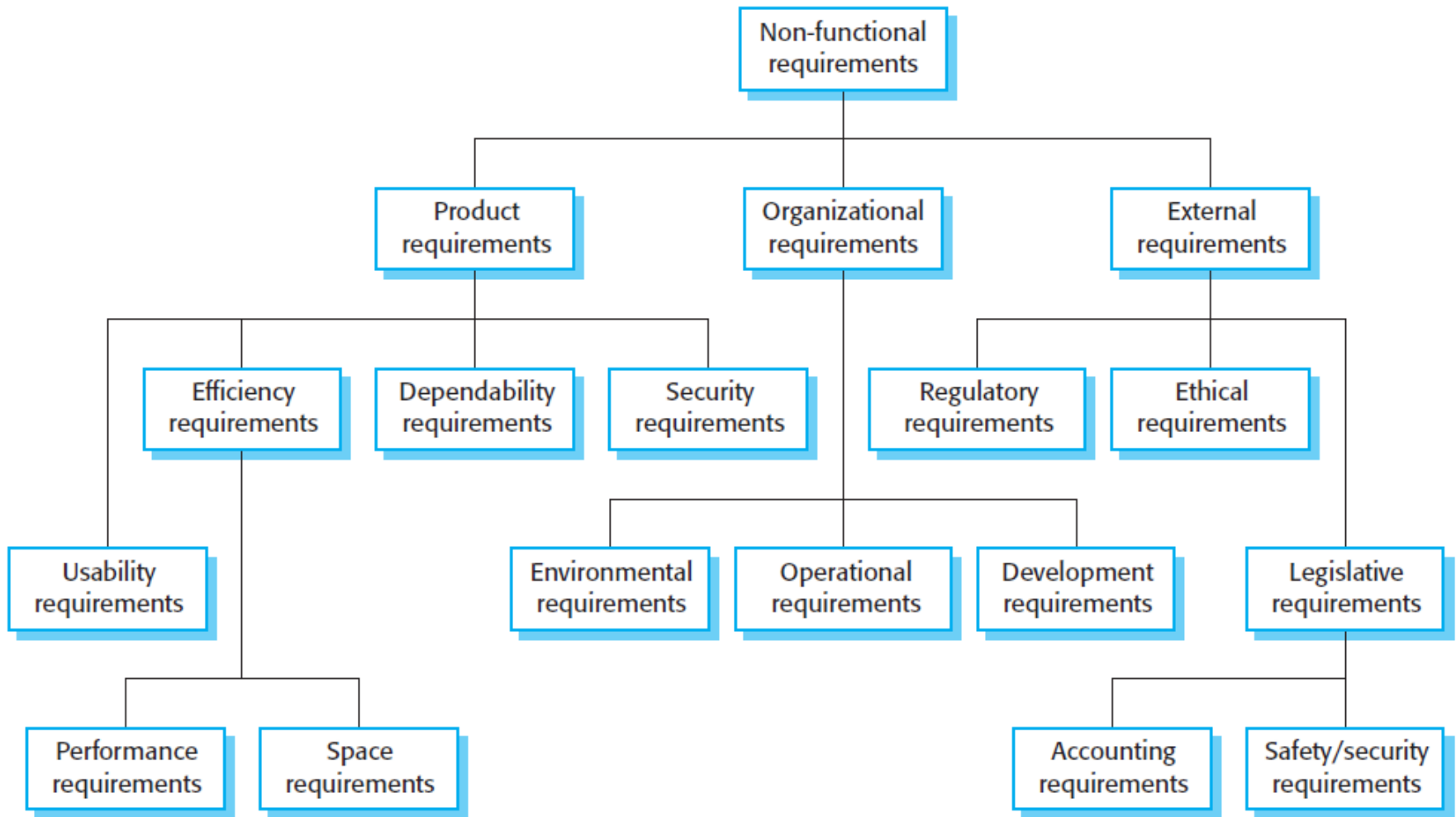
# Non-functional requirements

- Non-functional requirements, as the name suggests, are requirements that are not directly concerned with the specific services delivered by the system to its users.
- These non-functional requirements usually specify or constrain characteristics of the system as a whole.
- They may relate to emergent system properties such as reliability, response time, and memory use.
- Also, they may define constraints on the system implementation, such as the capabilities of I/O devices or the data representations used in interfaces with other systems.
- Non-functional requirements are often more critical than individual functional requirements.

# Non-functional requirements

- System users can usually find ways to work around a system function that doesn't really meet their needs. However, failing to meet a non-functional requirement can mean that the whole system is unusable.
  - For example, if an aircraft system does not meet its reliability requirements, it will not be certified as safe for operation; if an embedded control system fails to meet its performance requirements, the control functions will not operate correctly.
- Non-functional requirements arise through user needs because of budget constraints, organizational policies, the need for interoperability with other software or hardware systems, or external factors such as safety regulations or privacy legislation.

# Classification of non-functional requirements



Types of non-functional requirements

# Functional requirements

- ***Product requirements:*** *These requirements specify or constrain the runtime behaviour of the software.*

Examples include performance requirements for how fast the system must execute and how much memory it requires; reliability requirements that set out the acceptable failure rate; security requirements; and usability requirements.

- ***Organizational requirements:*** *These requirements are broad system requirements derived from policies and procedures in the customer's and developer's organizations.*

Examples include operational process requirements that define how the system will be used; development process requirements that specify the programming language; the development environment or process standards to be used; and environmental requirements that specify the operating environment of the system.



# Functional requirements

- ***External requirements:*** *This broad heading covers all requirements that are derived from factors external to the system and its development process.*

Example these may include regulatory requirements that set out what must be done for the system to be approved for use by a regulator, such as a nuclear safety authority; legislative requirements that must be followed to ensure that the system operates within the law; and ethical requirements that ensure that the system will be acceptable to its users and the general public.

# Non-Functional requirements: MentCare system as a Case Study

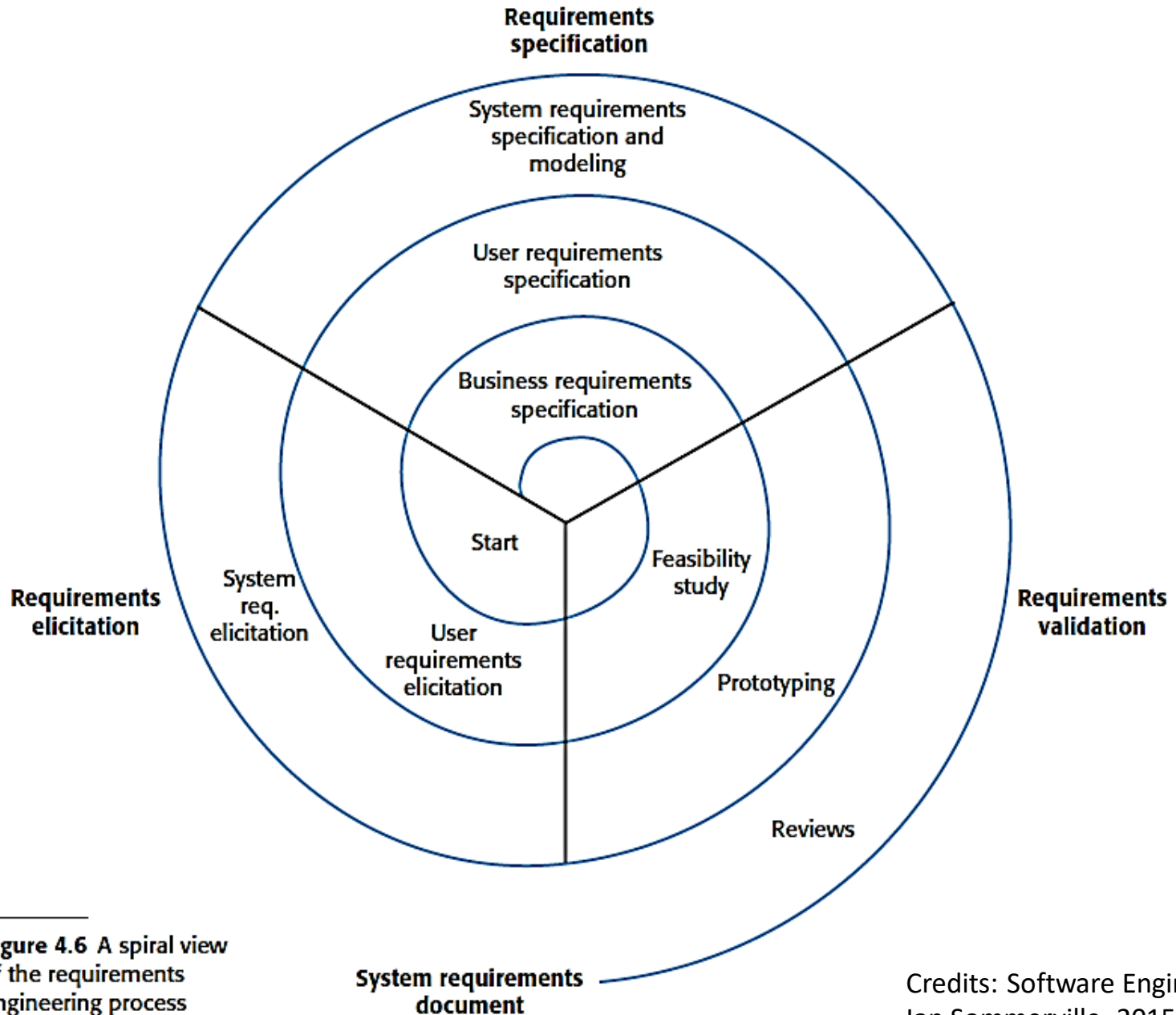
- **Product requirement:**
  - The Mentcare system shall be available to all clinics during normal working hours (Mon–Fri, 08:30–17:30).
  - Downtime within normal working hours shall not exceed 5 seconds in any one day.
- **Organizational requirement:**
  - Users of the Mentcare system shall identify themselves using their health authority identity card.
- **External requirement:**
  - The system shall implement patient privacy provisions as set out in “Privacy and the Right to Information Act, 2005: Govt. of India”

# Metrics for Non-Functional requirements specification.

Property	Measure
Speed	Processed transactions/second User/event response time Screen refresh time
Size	Megabytes/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

Different Metrics/properties for specifying non-unctional requirements.

# Requirements Engineering Process



**Figure 4.6** A spiral view of the requirements engineering process

# State completion criteria

- Completion criteria are the criteria by which you or anyone else can determine if a task was completed properly.
- Completion criteria can be a simple checklist or a more comprehensive set of standards or protocols. Whatever will be used to determine if the task was successfully completed should be defined in advance, in the completion criteria for the task.
- Criteria you set at the beginning of a project that determines when it is safe to stop testing. Exit criterion is connected to the test coverage, test case design technique adopted, risk level of the product varies from one test level to another.

## State completion criteria

- Specified coverage has been achieved.
- No Showstoppers or critical defects.
- There are very few known medium or low-priority defects that don't affect the usage of the product.
- If Exit criterion has not met, the test cannot be stopped.
- The Exit criterion has to be revamped or the time should be extended for testing based on the quality of the product.
- Any changes to the test completion criterion must be documented and signed off by the stakeholders.
- The test ware can be released upon successful completion of exit criteria.

# Example – Project Success Criteria: MentCare System

- Some example for testing and validating the a number of features of the Mentcare system:
  1. Authentication by logging on to the system.
  2. Downloading and uploading of specified patient records to a laptop.
  3. Home visit scheduling.
  4. Encryption and decryption of patient records on a mobile device.
  5. Record retrieval and modification.
  6. Links with the drugs database that maintains side-effect information.
  7. The system for call prompting.
  8. Security and Data Privacy check.

**Thank You: Question?**