

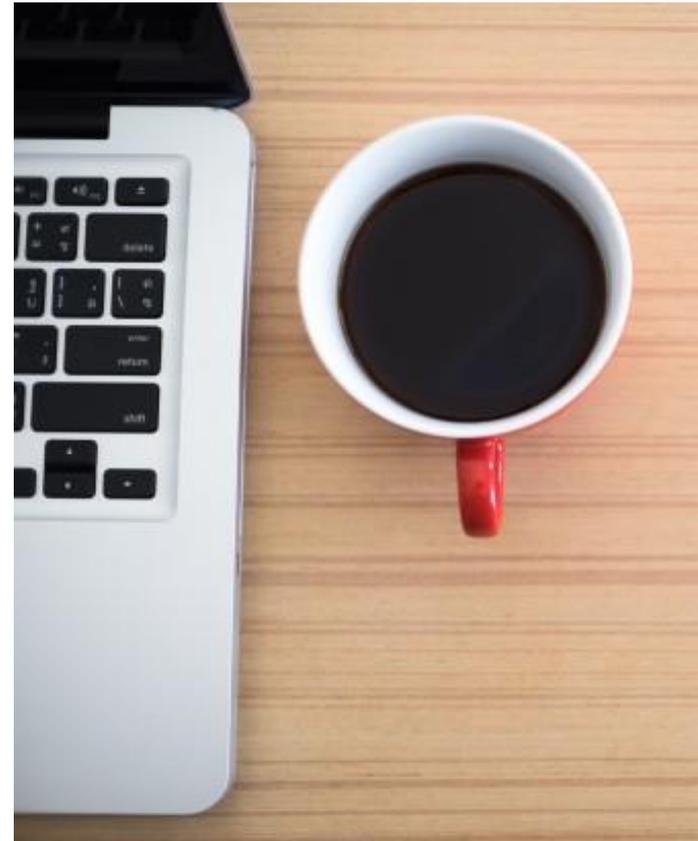
Course Topics

- ~~Introduction~~
- ~~Software Process Models~~
- Requirements Engineering
- Modeling
- Programming Languages
- Software Construction Techniques
- Testing
- Project Management
- Refactoring
- Ethical Issues

Lecture Objectives

- ✓ Requirements Engineering
 - User requirements
 - System requirements
 - Domain requirements

- ✓ Types of requirements
 - Functional
 - Non-functional





Programming Wisdom @CodeWisdom · 28 Nov 2016

“Without requirements or design, programming is the art of adding bugs to an empty text file.” - Louis Srygley

What is a Requirement?



- **Requirements** for a system are the descriptions of **what** the system should do and the **services** that it provides and the **constraints** on its operation.
- 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

What is a Requirement?

■ Requirements:

- May be given to the software engineers
 - Initial product/system requirements
 - For second and/or third follow-on release of a “planned” sequences of software product where a preliminary set of requirements are already established
 - Requirements provided as a part of a request for price quotation for a software development project
- Have to be established by software engineers
 - Users sometimes have an understanding of only the requirements related to their specific job tasks
 - The business rationale and goals are not always clear to individual user and needs to be established for prioritization reason
 - There may be contradicting and incomplete requirements stated by the users and customers

Requirements Engineering



- The process of finding out, analyzing, documenting and checking these services and constraints.
- The requirements themselves are the descriptions of
 - The system **services**
 - **Constraints** that are generated during the requirements engineering process

Types of Requirements

- 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's functions, services and operational constraints.
 - Defines what should be implemented so may be part of a contract between client and contractor.

- Why?
 - Different levels of requirements are useful because they communicate information about the system to different types of readers.

Types of Requirements

- A user requirement may be expanded into several system requirements.

User requirements definition

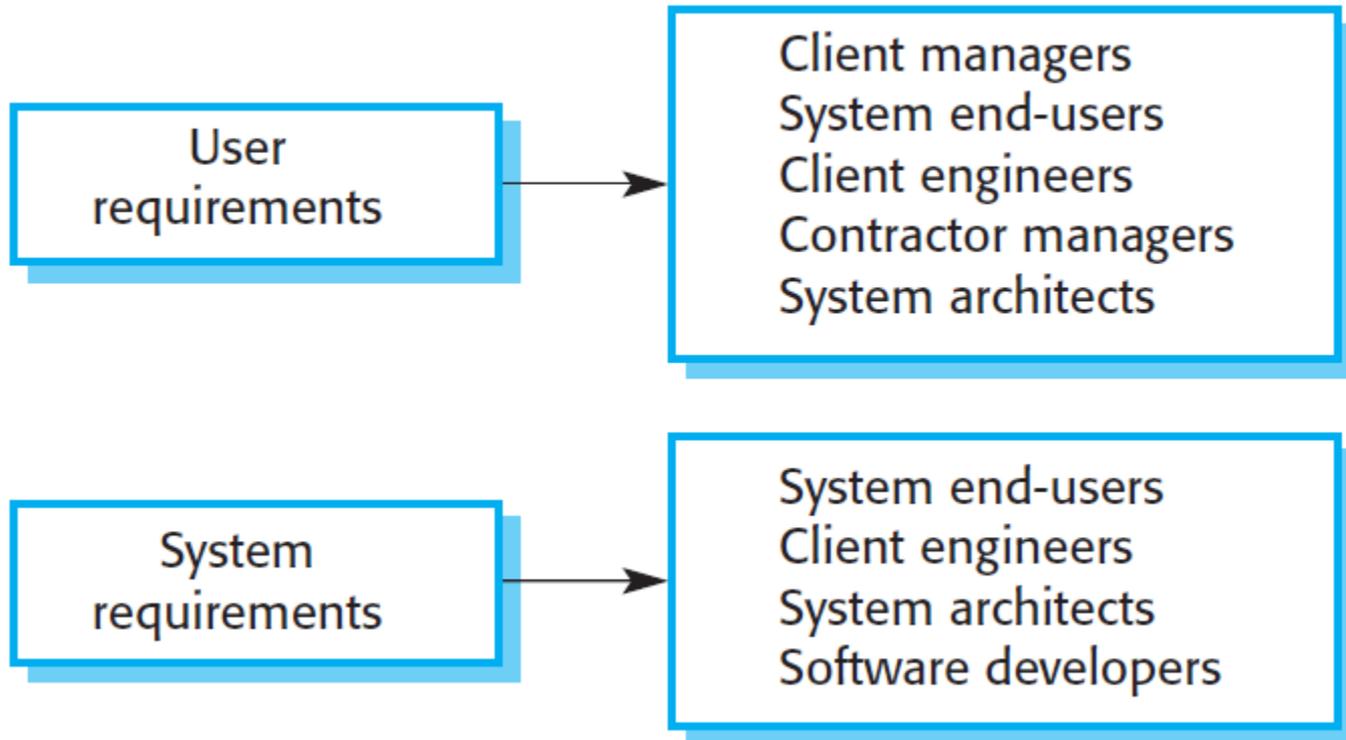
- 1.** The Mentcare system shall generate monthly management reports showing the cost of drugs prescribed by each clinic during that month.

System requirements specification

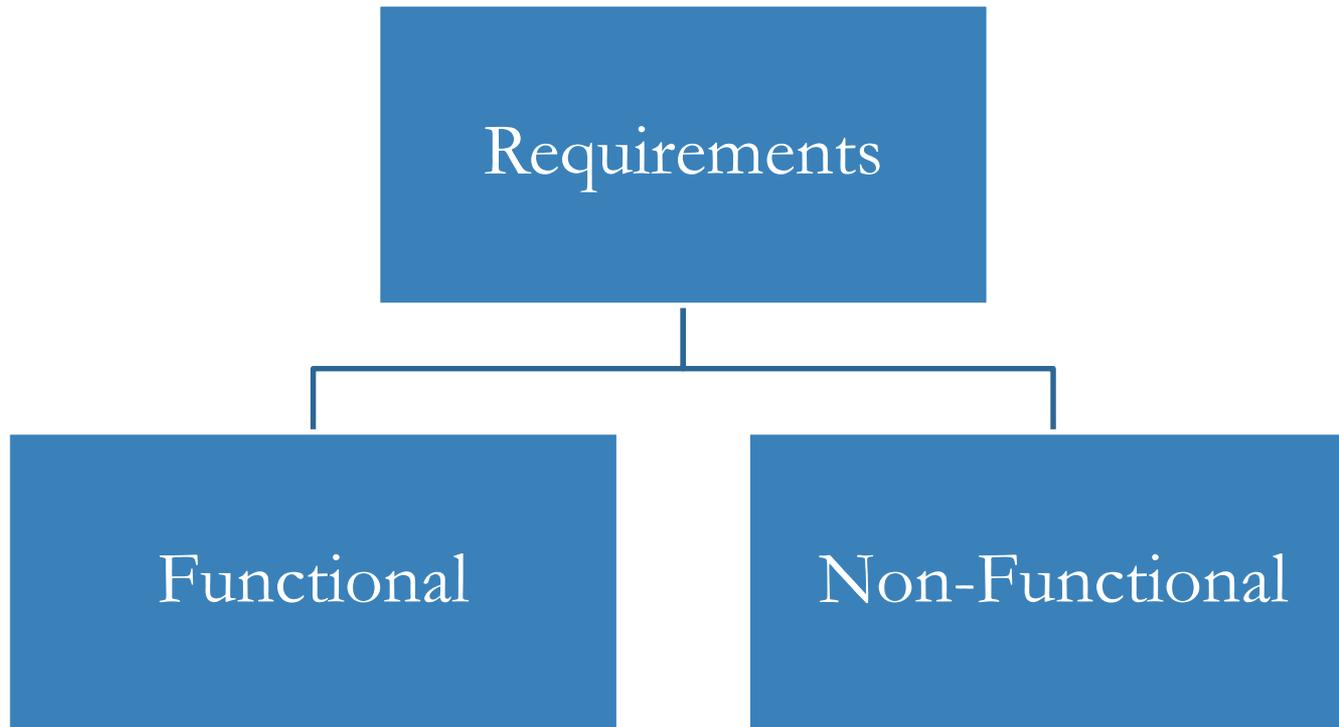
- 1.1** On the last working day of each month, a summary of the drugs prescribed, their cost and the prescribing clinics shall be generated.
- 1.2** The system shall generate the report for printing after 17.30 on the last working day of the month.
- 1.3** A report shall be created for each clinic and 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.

Different Types of Readers

- A user requirement may be expanded into several system requirements.



Functional and Non-Functional Reqs



Functional and Non-Functional Reqs



1- Functional requirements

- Statements of **services** the system should provide
- How the system should react to particular **inputs**
- How the system should **behave** in particular situations

2- Non-functional requirements

- **Constraints** on the services or functions offered by the system such as timing constraints
 - **Constraints** on the development process, standards, etc.
- 

Functional and Non-Functional Reqs

3- Domain requirements

- Requirements that come from the application domain of the system and that reflect characteristics of that domain.
- They may be functional or non-functional requirements.

In reality, distinctions between different types of Requirements is not clear-cut.

Functional and Non-Functional Reqs



User is concerned with security of the System. (appears as non-functional requirement)

The requirement can be refined as
'need to include user authentication' --
Functional Requirement

Functional Requirements (FR)



- Describe functionality or system services.
 - Depends on the 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 (Goals to achieve)

- Functional system requirements should describe the system services in detail

Examples of functional requirements

- User Functional Requirements
 - 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.

- System Functional Requirement
 - 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.

Non-Functional Requirements (NFR)



- These define system properties and constraints e.g.
 - reliability,
 - response time and storage requirements.
 - constraints on system implementation, such as I/O device capability, or data representation used in interfaces with other systems.

- Non-functional requirements may be more critical than functional requirements.

Goals and NFR



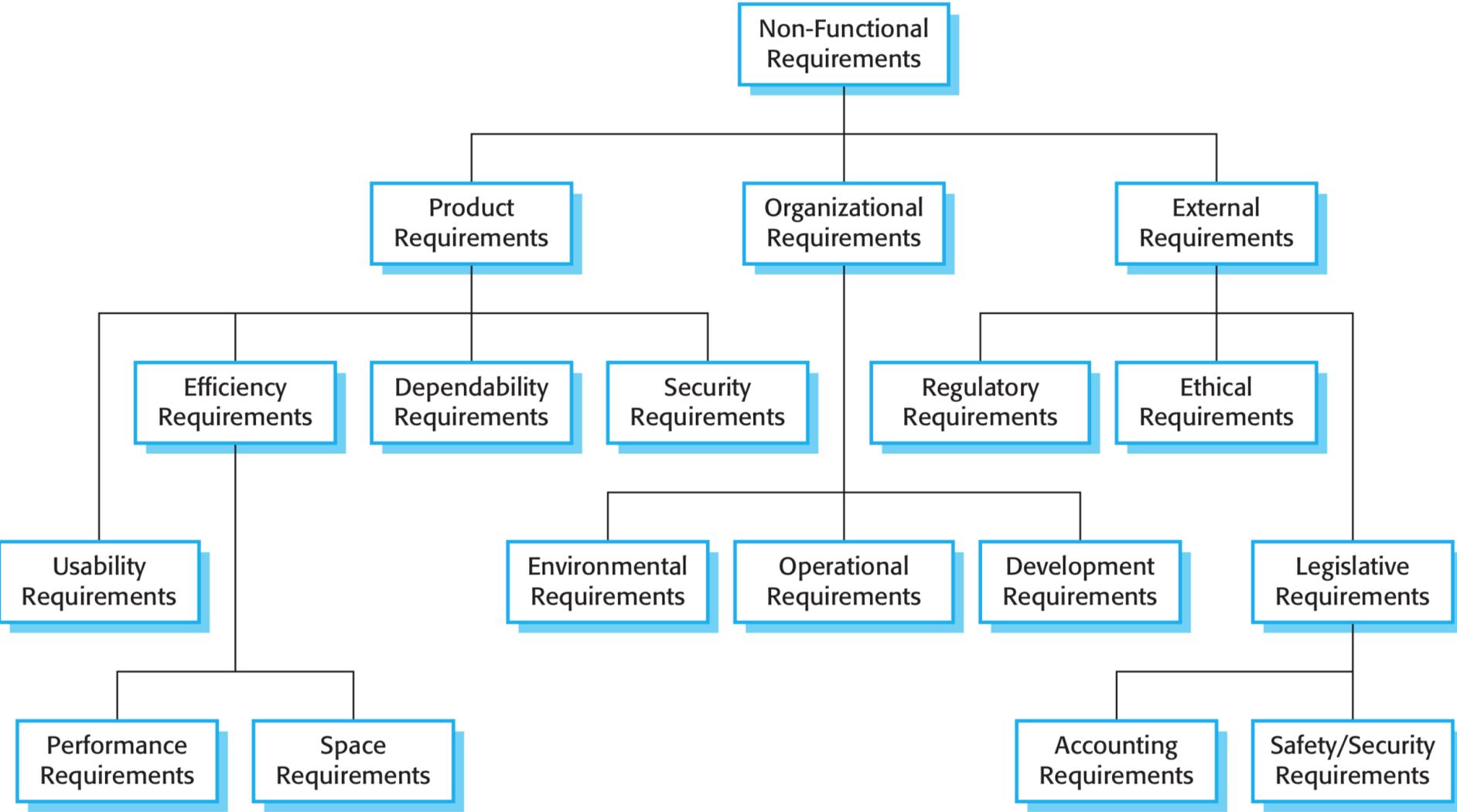
- Non-functional requirements may be very difficult to state precisely and imprecise requirements may be difficult to verify.
 - Goal
 - A general intention of the user such as “ease of use”.
 - Verifiable non-functional requirement
 - A statement using some measure that can be objectively tested.
 - 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.





Non-functional classifications

- **Product requirements** - system must behave in a particular way
e.g.
 - execution speed, memory usage, usability requirements, reliability requirements (set out the acceptable failure rate), etc.
- **Organisational requirements** – derived from organizational policies and procedures, e.g.
 - process standards used, implementation requirements, development environment, etc.
- **External requirements** - factors which are external to the system e.g.
 - Regulatory requirements, interoperability requirements, legislative requirements, etc.

Examples of Non-functional requirements

- **Product requirement**
 - The user interface for LIBSYS shall be implemented as simple HTML without frames or Java applets.

- **Organisational requirement**
 - The system development process and deliverable documents shall conform to the process and deliverables defined in XYZCo-SP-STAN-95.

- **External requirement**
 - The hotel management system should comply with the Ministry of Tourism guidelines and standards as documented in ...

Examples of Non-functional requirements



- **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.

 - **Organisational 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 HStan-03-2006-priv.
- 

Metrics for Product requirements

Property	Measure
Speed (Efficiency)	<ul style="list-style-type: none"> Processed transactions/second User/event response time Screen refresh time
Size (Efficiency)	<ul style="list-style-type: none"> Mbytes Number of ROM chips
Ease of use (Usability)	<ul style="list-style-type: none"> Training time Number of help frames
Reliability	<ul style="list-style-type: none"> Mean time to failure Probability of unavailability Rate of failure occurrence Availability
Robustness	<ul style="list-style-type: none"> Time to restart after failure Percentage of events causing failure Probability of data corruption on failure
Portability	<ul style="list-style-type: none"> Percentage of target dependent statements Number of target systems

Domain requirements



- Derived from the application domain and describes system characteristics and features that reflect the domain.
 - Domain requirements can be new functional requirements, constraints on existing requirements or define specific computations.
 - If domain requirements are not satisfied, the system may be unworkable.
- 

Library system domain requirements

- There shall be a standard user interface to all databases which shall be based on the **Z39.50 standard**.



- Because of copyright restrictions, some documents must be deleted immediately on arrival. Depending on the user's requirements, these documents will either be printed locally on the system server for manually forwarding to the user or routed to a network printer.

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.

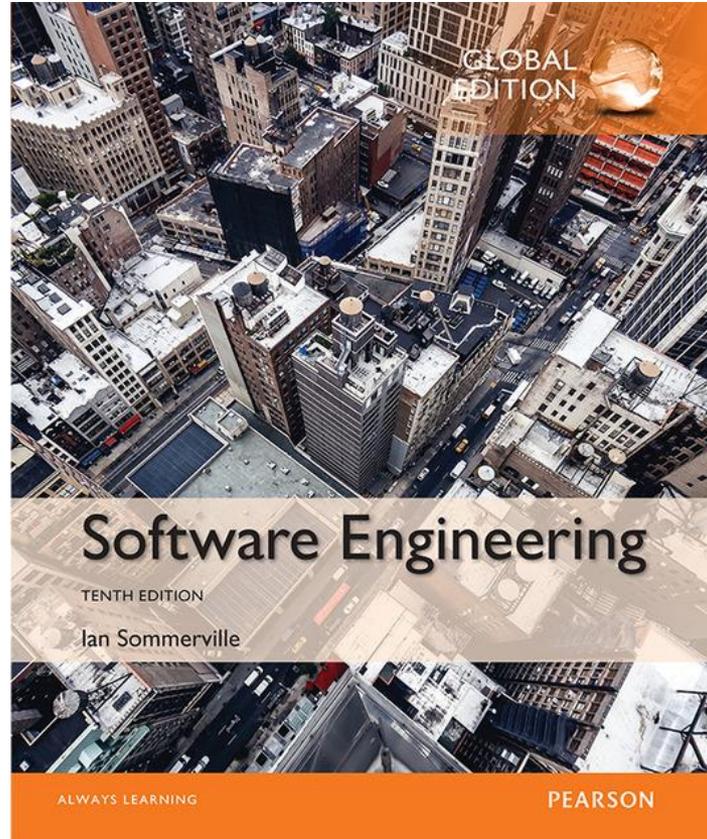
Key Points



- Software requirements is a process establishing the services that the customer requires from a system; and the constraints under which it operates and is developed.
- Software requirements
 - Level of details
- Functional & Non Functional Requirements

Read

Chapter 4



References



- Ian Sommerville, “Software Engineering”, 10th Edition, Addison-Wesley, 2015.
 - Timothy C. Lethbridge and Robert Laganière, “Object-Oriented Software Engineering: Practical Software Development using UML and Java”, 2nd Edition, McGraw Hill, 2001.
 - R. S. Pressman, Software Engineering: A Practitioner’s Approach, 10th Edition, McGraw-Hill, 2005.
- 

Question exercise

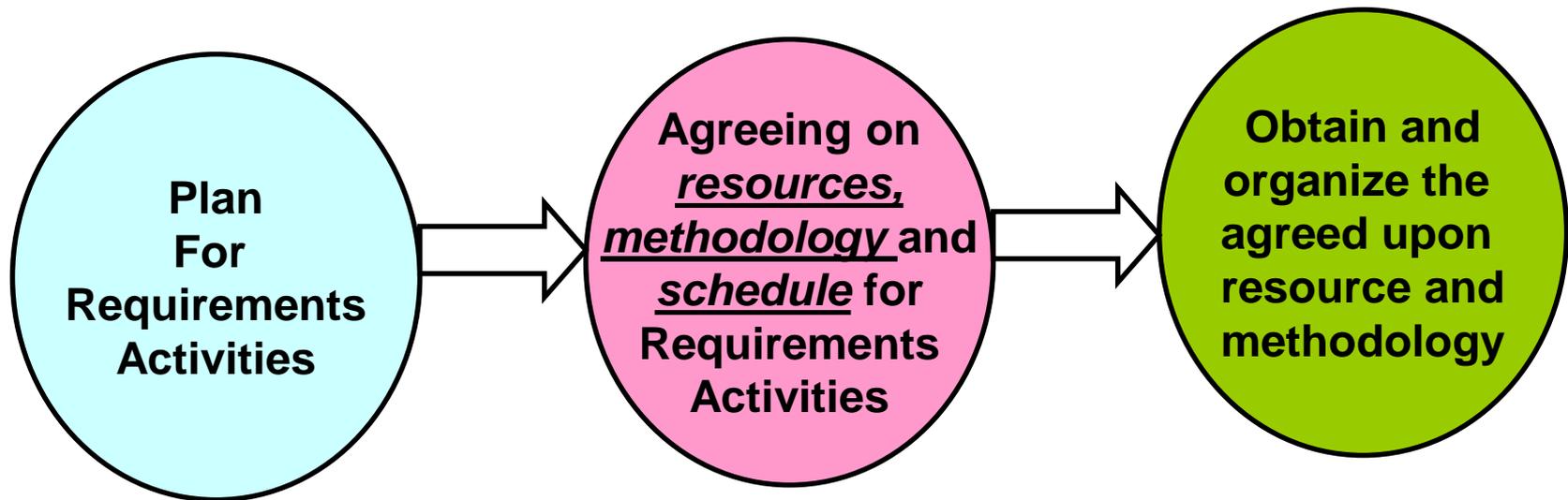


- KFUPM Portal

- Discuss requirements
 - Functional
 - Non functional

Extra Slides

Preparation for Requirements Engineering



1. Prior to actually performing the requirements engineering activities, it is important to plan for the resources, methodology and time needed to perform this crucial step in software engineering.
2. Some organizations even perform requirements engineering as a separate, stand-alone activity and price it separately, with the option of folding the cost into the whole project if they get the call to complete the software project.