



CSE- 321

Software Engineering

Lecture : 06

Project Management & Requirement

Fahad Ahmed

Lecturer, Dept. of CSE

E-mail: fahadahmed@uap-bd.edu

Lecture Outlines

- ✧ **Software project management**
- ✧ **The 4 P's**
- ✧ **Management activities**
- ✧ **Project planning**
- ✧ **Risk management**
- ✧ **Requirements Engineering (RE)**
- ✧ **Functional and Non-functional Requirements**
- ✧ **Requirements Engineering Process**
- ✧ **Feasibility Studies**
- ✧ **Requirements Elicitation and Analysis**
- ✧ **Requirements Management**

4Ps in Marketing



Software Project Management : The 4 P's

- **People** — the most important element of a successful project :Developers, project managers, testers, ..etc.
- **Product** — the software to be built
- **Process** — the set of framework activities and software engineering tasks required to get the job done.
- **Project** — all work required to make the product a reality



Copyright © Deepak

3

Software Project Management : The 4 P's

- **People** — People of a project includes from manager to developer, from customer to end user. But mainly people of a project highlight the developers. It is so important to have highly skilled and motivated developers that the Software Engineering Institute has developed a **People Management Capability Maturity Model (PM-CMM)**.
- **Product** — Product is any software that has **to be developed**. To develop successfully, product objectives and scope should be established, alternative solutions should be considered, and technical and management constraints should be identified. Without this information, it is impossible to define reasonable and accurate estimates of the cost, an effective assessment of risk, a realistic breakdown of project tasks or a manageable project schedule that provides a meaningful indication of progress.

Software Project Management : The 4 P's

- **Process** — A software process provides the framework from which a **comprehensive plan** for software development can be established. A number of different tasks sets— tasks, milestones, work products, and quality assurance points— enable the framework activities to be adapted to the characteristics of the software project and the requirements of the project team.
- **Project** —The project includes all and everything of the total development process and to **avoid project failure**. Here, the project manager has to do some job.

Software Project Management

- Concerned with activities involved in ensuring that software is delivered **on time** and **on schedule** and in accordance with the **requirements** of the organisations developing and procuring the software.
- Project management is needed because software development is always subject to **budget and schedule constraints** that are set by the organisation developing the software.

Management activities

- ❖ Proposal writing.
- ❖ Project planning and scheduling.
- ❖ Project costing.
- ❖ Project monitoring and reviews.
- ❖ Personnel selection and evaluation.
- ❖ Report writing and presentations.

Project planning

- Probably the most **time-consuming** project management activity.
- **Continuous activity** from initial concept through to system delivery. Plans must be regularly **revised** as new information becomes available.
- Various **different types of plan** may be developed to support the main software project plan that is concerned with schedule and budget.

Project plan structure

Most plans should include the following sections:

- Introduction.
- Project organisation.
- Risk analysis.
- Hardware and software resource requirements.
- Work breakdown.
- Project schedule.
- Monitoring and reporting mechanisms.

Bar charts and activity networks

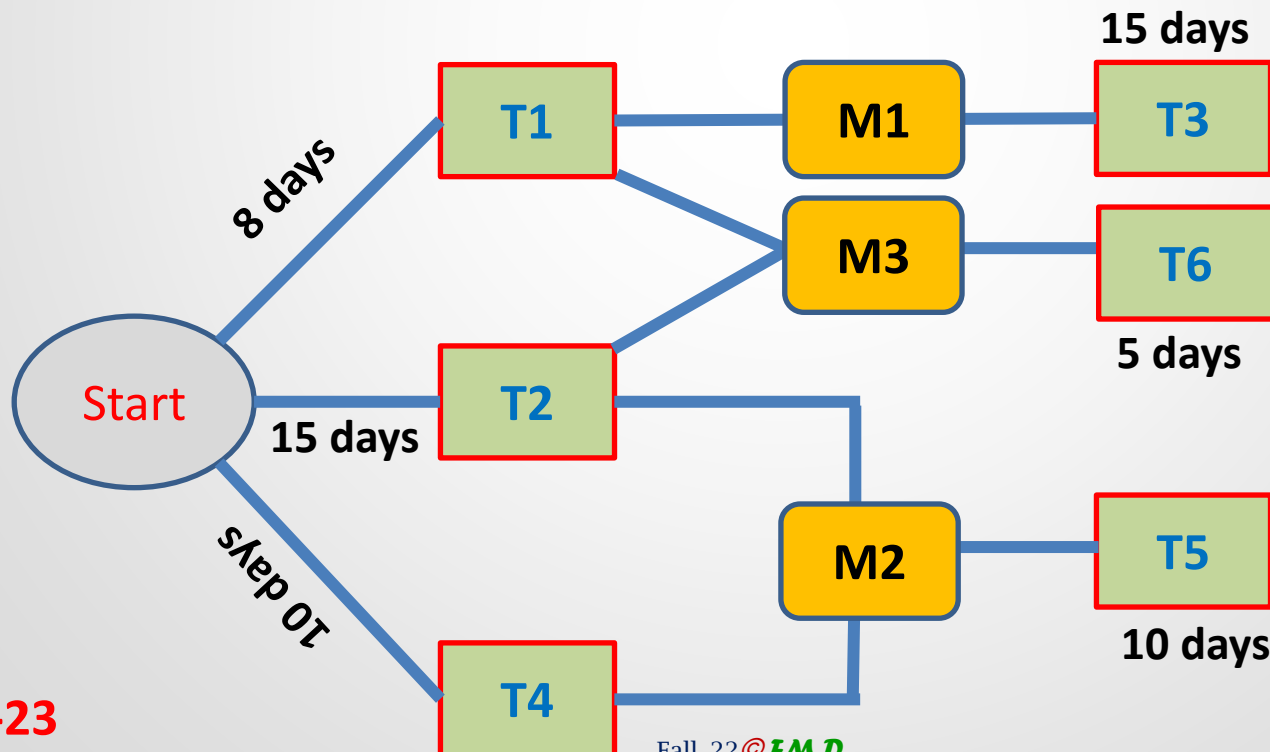
- Graphical notations used to illustrate the project schedule.
- Show project breakdown into tasks. Tasks should not be too small. They should take about a week or two.
- Activity charts show task dependencies and the critical path.
- Bar charts show schedule against calendar time.

Task durations and dependencies

Task	Duration (days)	Dependencies
T1	8	
T2	15	
T3	15	T1 (M1)
T4	10	
T5	10	T2, T4 (M2)
T6	5	T1, T2 (M3)
T7	20	T1 (M1)
T8	25	T4 (M5)
T9	15	T3, T6 (M4)
T10	15	T5, T7 (M7)
T11	7	T9 (M6)
T12	10	T11 (M8)

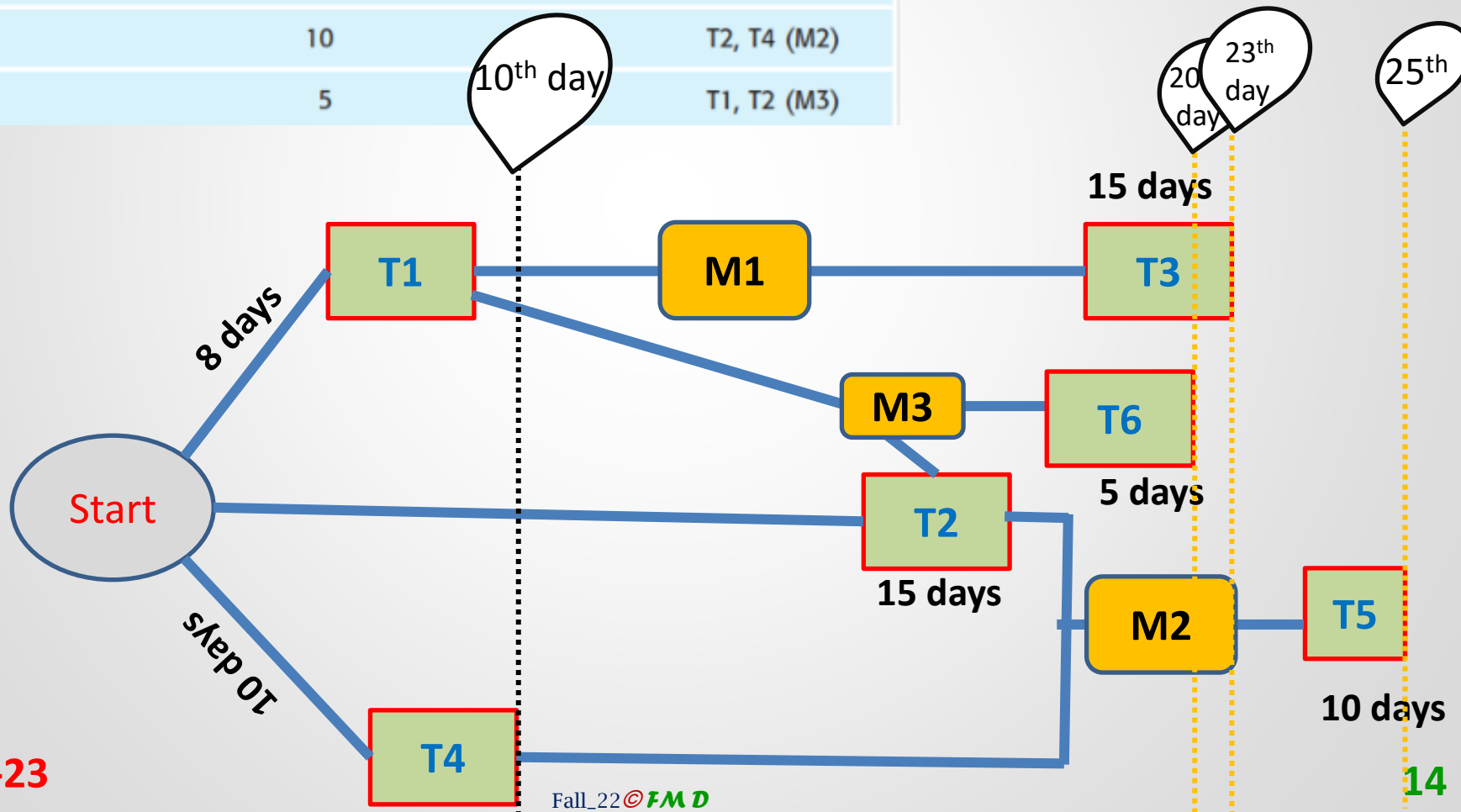
Activity network

Task	Duration (days)	Dependencies
T1	8	
T2	15	
T3	15	T1 (M1)
T4	10	
T5	10	T2, T4 (M2)
T6	5	T1, T2 (M3)

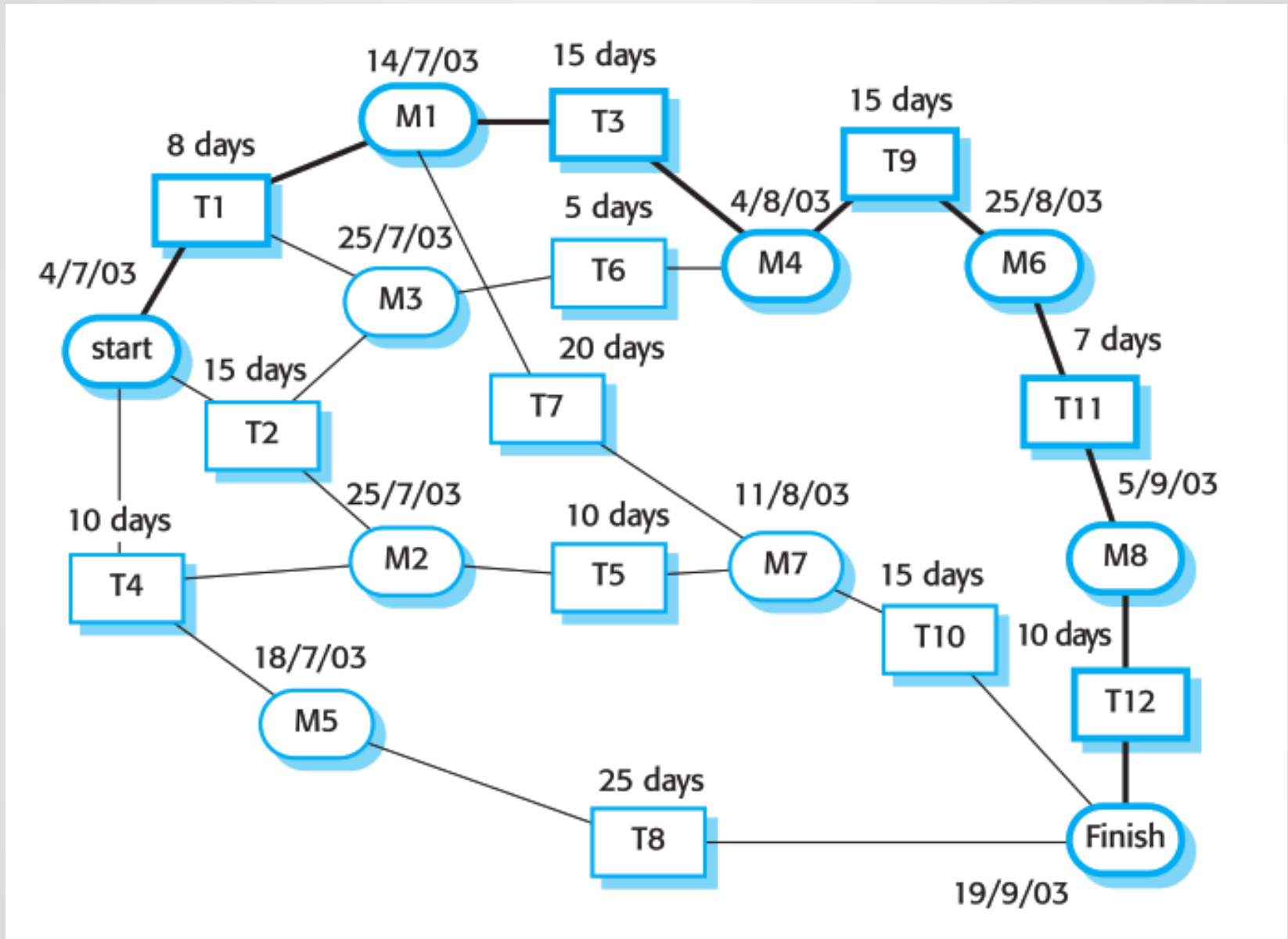


Activity network

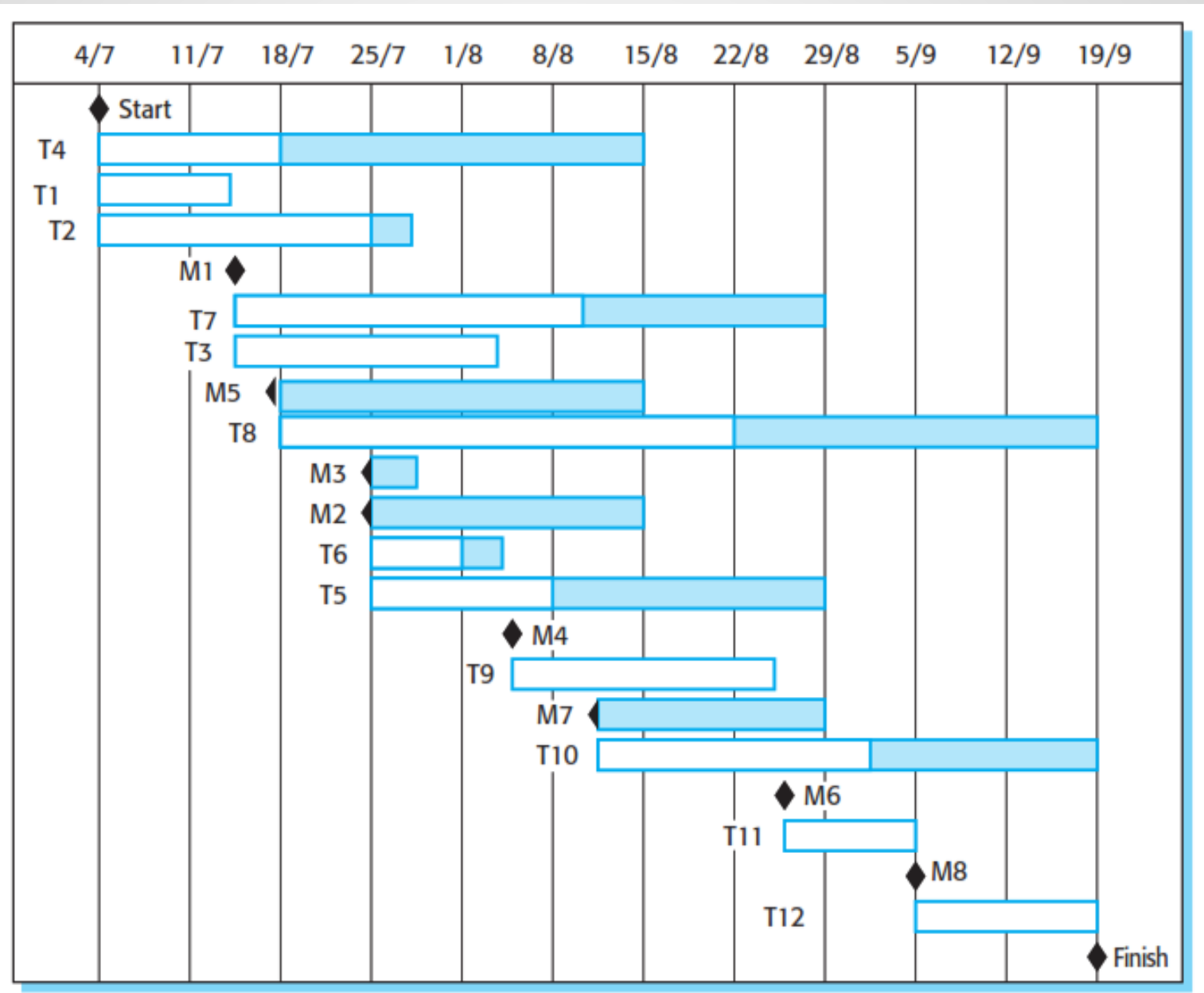
Task	Duration (days)	Dependencies
T1	8	
T2	15	
T3	15	T1 (M1)
T4	10	
T5	10	T2, T4 (M2)
T6	5	T1, T2 (M3)



Activity network



Activity timeline



Risk management

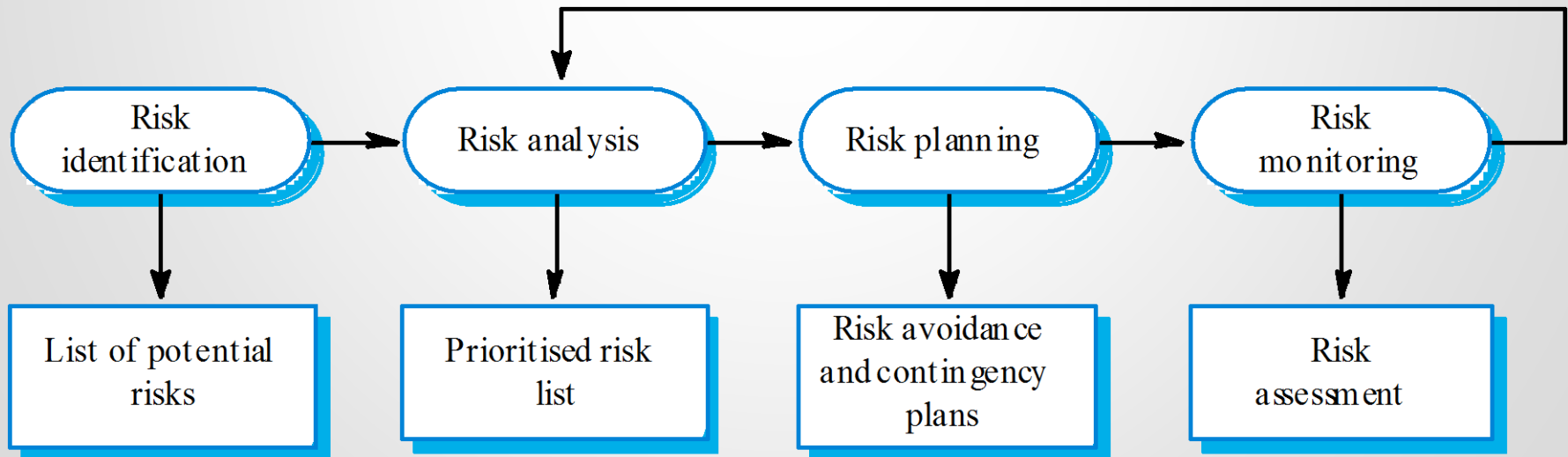
- Risk management is concerned with identifying risks and drawing up plans to minimise their effect on a project.
- A risk is a probability that some adverse circumstance will occur
 - Project risks affect schedule or resources;
 - Product risks affect the quality or performance of the software being developed;
 - Business risks affect the organisation developing or procuring the software.

Possible software risks

Risk	Risk type	Description
Staff turnover	Project	Experienced staff will leave the project before it is finished.
Management change	Project	There will be a change of organisational management with different priorities.
Hardware unavailability	Project	Hardware which is essential for the project will not be delivered on schedule.
Requirements change	Project and product	There will be a larger number of changes to the requirements than anticipated.
Specification delays	Project and product	Specifications of essential interfaces are not available on schedule.
Size underestimate	Project and product	The size of the system has been underestimated.
CASE tool under-performance	Product	CASE tools which support the project do not perform as anticipated.
Technology change	Business	The underlying technology on which the system is built is superseded by new technology.
Product competition	Business	A competitive product is marketed before the system is completed.

The risk management process

- **Risk identification**
 - Identify project, product and business risks;
- **Risk analysis**
 - Assess the likelihood and consequences of these risks;
- **Risk planning**
 - Draw up plans to avoid or minimise the effects of the risk;
- **Risk monitoring**
 - Monitor the risks throughout the project;



The risk management process

■ Risk identification

There are at least six types of risk that can arise:

- ❖ **Technology risks** : Risks that derive from the software or hardware technologies that are used to develop the system.
- ❖ **People risks**: Risks that are associated with the people in the development team.
- ❖ **Organisational risks**: Risks that derive from the organisational environment where the software is being developed.
- ❖ **Tools risks**: Risks that derive from the CASE tools and other support software used to develop the system.
- ❖ **Requirements risks**: Risks that derive from changes to the customer requirements and the process of managing the requirements change.
- ❖ **Estimation risks**: Risks that derive from the management estimates of the system characteristics and the resources required to build the system.

The risk management process

■ Risk analysis

During the risk analysis process, we have to consider each identified risk and make a judgement about the probability and the seriousness of it. There is **no easy way** to do this—we must rely on your own judgement and experience, which is why experienced project managers are generally the best people to help with risk management.

These risk estimates should not generally be precise numeric assessments. But should be based around a number of bands:

- The probability of the risk might be assessed as **very low (<10%), low (10–25%), moderate (25–50%), high (50–75%) or very high (>75%)**.
- The effects of the risk might be assessed as catastrophic, serious, tolerable or insignificant.

The risk management process

■ Risk Management strategies

Risk	Strategy
Organisational financial problems	Prepare a briefing document for senior management showing how the project is making a very important contribution to the goals of the business.
Recruitment problems	Alert customer of potential difficulties and the possibility of delays, investigate buying-in components.
Staff illness	Reorganise team so that there is more overlap of work and people therefore understand each other's jobs.
Defective components	Replace potentially defective components with bought-in components of known reliability.

The risk management process

■ Risk Management strategies

Risk	Strategy
Requirements changes	Derive traceability information to assess requirements change impact, maximise information hiding in the design.
Organisational restructuring	Prepare a briefing document for senior management showing how the project is making a very important contribution to the goals of the business.
Database performance	Investigate the possibility of buying a higher-performance database.
Underestimated development time	Investigate buying-in components, investigate the use of a program generator.

Requirements Engineering (RE)

- The requirements for a system are the descriptions of the services provided by the system and its operational constraints.
- The process of **finding out**, **analyzing**, **documenting** and **checking** these services and constraints is called **requirements engineering (RE)**.
- **User requirements**
 - Statements in **natural language** plus **diagrams of the services** the system provides and its operational constraints.
- **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.

Functional and non-functional requirements

■ Functional requirements (FRs)

Statements of services the system should provide, how the system should react to particular inputs and how the system should behave in particular situations.

- what the system should do.

Functional Requirements Example:

- Authentication of a user when he/she tries to log into the system.
- System shutdown in the case of a cyber attack.
- Verification email is sent to user whenever he/she registers for the first time on some software system.

Functional and non-functional requirements

■ Non-functional requirements(NFRs)

- ❑ **constraints on the services** or functions offered by the system such as timing constraints, constraints on the development process, standards, etc.
- ❑ Non-functional requirements **are not directly concerned with specific system** functions but specify required system properties or place constraints on the system or its development process.
- ❑ Non-functional requirements include -
 - ❑ Security
 - ❑ Logging
 - ❑ Configuration
 - ❑ Performance
 - ❑ Cost
 - ❑ Flexibility
 - ❑ Disaster recovery
 - ❑ Accessibility

Non-functional Requirements Example:

- Emails should be sent with a latency of no greater than 12 hours.
- Each request should be processed within 10 seconds.
- The site should load in 3 seconds when the number of simultaneous users are > 10000

Functional and non-functional requirements

Functional Requirements	Non-Functional Requirements
They define a system or its component.	They define the quality attribute of a system
It specifies, "What the system should do?"	It specifies, "How should the system fulfill the functional requirements?"
User specifies functional requirement.	Non-functional requirement is specified by technical peoples e.g. Architect, Technical leaders and software developers.
It is mandatory to meet these requirements.	It is not mandatory to meet these requirements.
It is captured in use case.	It is captured as a quality attribute.
Defined at a component level.	Applied to a whole system.
Helps you to verify the functionality of the software.	Helps you to verify the performance of the software.
Functional Testing like System, Integration, End to End, API testing, etc are done.	Non-Functional Testing like Performance, Stress, Usability, Security testing, etc are done.
Usually easy to define.	Usually more difficult to define.

Project: Travel and Tourism Management System

The travel Management System offers clients a varieties of **travel packages** all over in Europe on an Affordable cost such as Sea exploring, Forest hiking, sports trip, snow adventure, city wanderer, mountain biking etc. Switzerland is one of the richest countries for tourist places in the world but there is not even a single platform who provide all kind of mandatory services and different kind of services on a single travel Management System in PHP. We provide a feature like tour mentors to guide, providing schedule of all upcoming travel packages containing all details of trip like number of people who already booked their reservation on a specific trip, difficulty level of trip, rating of each individual location, feedback system of clients so that new tourist can get awareness of their concerning trip, rating of a places, number of stops toward journey for tourists to savor the happiness and enjoy a beauty of nature.

Functional and non-functional requirements

Functional Requirements of Travel Management System

Admin Requirements:

- ❖ The admin can add new packages.
- ❖ The admin can update any package.
- ❖ The admin can delete any package.
- ❖ The admin can search any package
- ❖ The admin can also change his name or password from the dashboard settings.

End User Requirements:

- ❖ The end users book tour plan according to their choice.
- ❖ User share their feedback with comments.
- ❖ The user can sign up after filling all the fields mentioned in the sign up form.
- ❖ The user can login after validation of his/her details from the database.
- ❖ The end users can reserve their seats for tour plan which they choose or like.

Functional Requirements of Travel Management System

System Requirements:

- ❖ Travel management system offer logout functionality to end users.
- ❖ Travel management system will only accept a valid login details to enroll on a travel Management System in PHP.
- ❖ Travel management system will provide password recovery facility.
- ❖ Travel management system will redirect the user to whats-app whenever the whats-app icon is pressed for online payment purposes

Functional and non-functional requirements

Non Functional Requirements of Travel Management System

Database Security:

An unauthorized person cannot access the panel and database, do not read and write the information. It should maintain the security of the client's payment method.

Reservations Requirement:

Travel management system should reserve a travel package in maximum 30 to weekly evaluation by the project guide.

Reliability Requirement:

Travel management system should provide a reliable environment to both customers and owner.

Admin should be able to upload delete update new packages without any error.

Functional and non-functional requirements

Non Functional Requirements of Travel Management System

Usability Requirement:

The travel Management System in PHP is designed for user friendly environment and ease of use.

Availability:

The travel Management System in PHP should be available for 24 hours because it offers international tourists reserved packages from different countries so it should be available for 24 hours.

Efficiency Requirement:

When an online package of travel implemented customer can have reserved packages in an efficient manner.

LANGUAGES & TOOLS

- ❖ PHP or ASP.NET
- ❖ HTML
- ❖ CSS
- ❖ JavaScript
- ❖ Mysql-XAMPP
- ❖ VsCode

Requirement related terms

- ❑ **Prototyping** is building user interface without adding detail functionality for user to interpret the features of intended software product. It helps giving better idea of requirements.
- ❑ **Brainstorming:** An **informal debate** is held among various stakeholders and all their inputs are recorded for further requirements analysis.
- ❑ A **user story** is an informal, natural language description of features of a software system. They are written from the perspective of an end user or user of a system, and may be recorded on index cards, Post-it notes, or digitally in project management software.

A **feasibility study** decides whether or not the proposed system is worthwhile.

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 organization?
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?

Feasibility study implementation

Based on information assessment (what is required), information collection and report writing

Questions for people in the organization

- What if the system wasn't implemented?
- What are current process problems?
- How will the proposed system help?
- Is new technology needed? What skills?
- What will be the integration problems?
- What facilities must be supported by the system?
- What is the risk associated with cost and schedule?
- What are the potential disadvantages/advantages?
- Are there legal issues?
- Are there issues linked with the fact that this is an offshore project?

Problems of requirements analysis

- Stakeholders don't know what they really want
- Stakeholders express requirements in their own terms
- Different stakeholders may have different requirements
- Organizational and political factors may influence the system requirements
- The requirement changes during the analysis process.

Enduring and volatile requirements

Enduring and volatile requirements

- **Enduring requirements:**

These are the **stable** requirements derived from the core activity of the customer organization. E.g. a hospital will always have doctors, nurses, etc. May be derived from domain models

- **Volatile requirements:**

These are requirements which **change** during development process or when the system is in use. In a hospital, requirements may be derived from government health-care policies.



Thanks to All