Information Systems Project Management

Chapter 1: Introduction

What Is a Project?

- A **project** is "a temporary endeavor undertaken to create a unique product, service, or result" (PMBOK® Guide, Fifth Edition, 2013)
- Operations is work done to sustain the business
- Projects end when their objectives have been reached or the project has been terminated
- Projects can be large or small and take a short or long time to complete

Examples of IT Projects

- A team of students creates a smartphone application and sells it online
- A company develops a driverless car
- A government group develops a system to track child immunizations
- A company develops a new system to increase sales force productivity and customer relationship management that will work on various laptops, smartphones, and tablets.

Top Strategic Technologies

- Computing everywhere
- The Internet of things
- 3D printing
- Advanced, pervasive, and invisible analytics

Project Attributes

- A project
  - has a unique purpose
  - is temporary
  - is developed using progressive elaboration
  - requires resources, often from various areas
  - should have a primary customer or sponsor
    - The **project sponsor** usually provides the direction and funding for the project
  - involves uncertainty

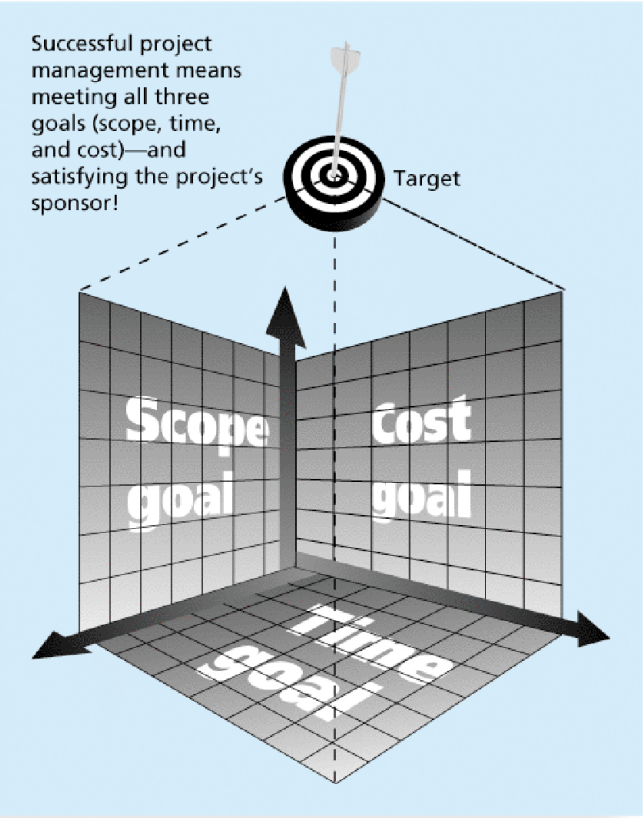
Project and Program Managers

- **Project managers** work with project sponsors, project team, and other people involved in a project to meet project goals
- **Program**: group of related projects managed in a coordinated way to obtain benefits and control not available from managing them individually (PMBOK® Guide, Fifth Edition, 2013)
- Program managers oversee programs; often act as bosses for project managers

Figure 1-1 The Triple Constraint of Project Management

Success criteria:

- Deliver the software to the customer at the agreed time.
- Keep overall costs within budget.
- Deliver software that meets the customer's expectations.
- Maintain a happy and well-functioning development team.



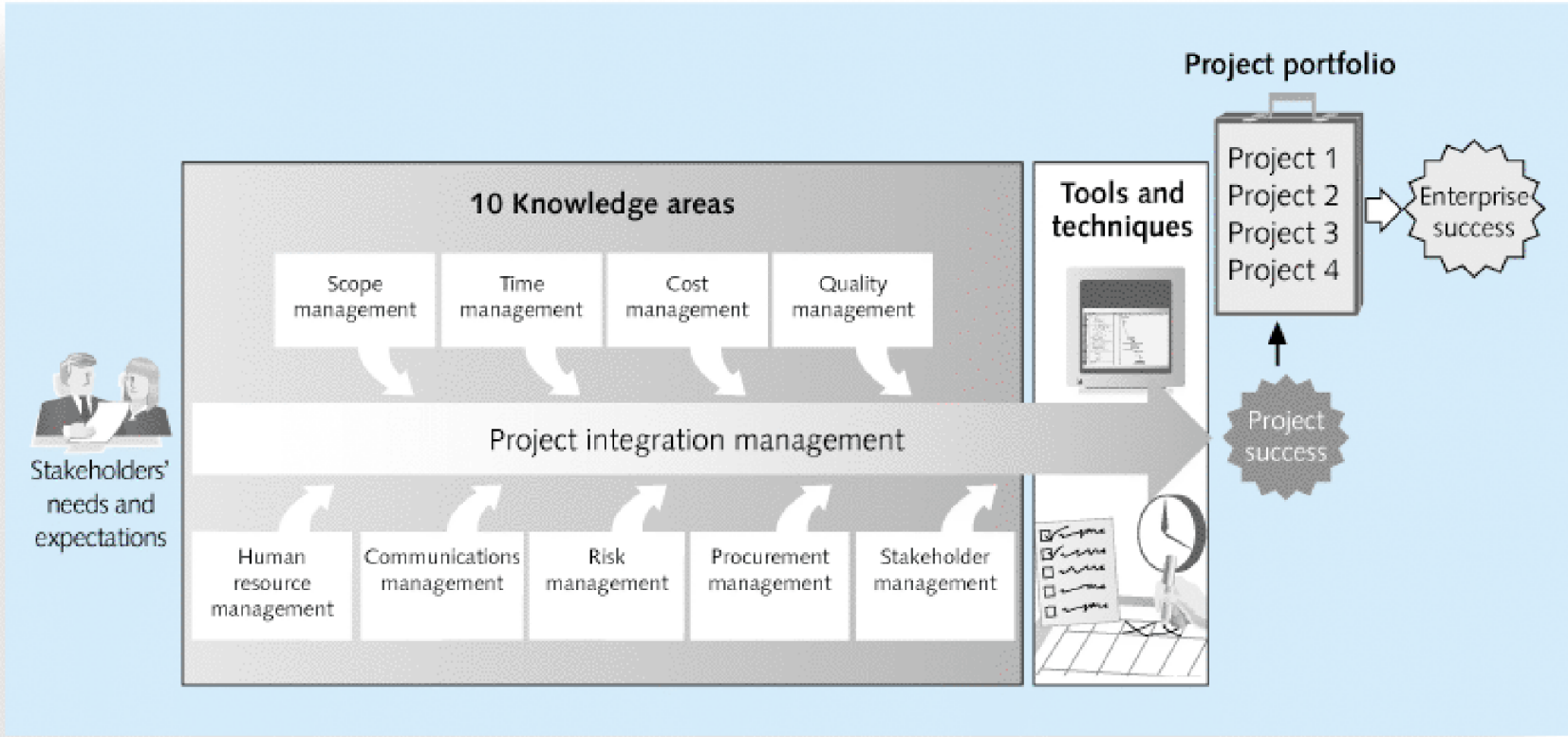
What is Project Management?

- **Project management** is "the application of knowledge, skills, tools and techniques to project activities to meet project requirements" (PMBOK® Guide, Fourth Edition, 2013)
- Project managers strive to meet the **triple constraint** (project scope, time, and cost goals) and also facilitate the entire process to meet the needs and expectations of project stakeholders

Advantages of Using Formal Project Management

- Better control of financial, physical, and human resources
- Improved customer relations
- Shorter development times
- Lower costs
- Higher quality and increased reliability
- Higher profit margins
- Improved productivity
- Better internal coordination
- Higher worker morale

Figure 1-2 Project Management Framework



Knowledge areas:

- Project scope management: involves defining and managing all the work required to complete the project successfully.
- Project time management includes estimating how long it will take to complete the work, developing an acceptable project schedule and ensuring timely completion of the project
- Project cost management: consists of preparing and managing the budget for the project
- Project quality management: ensures that the project will satisfy the stated or implied needs for which it was undertaken
- Project human resource management: concerned with making effective use of the people involved with the project.

Knowledge areas

- Project communications management: involves generating, collecting, disseminating and storing project information
- Project risk management: includes identifying, analyzing, and responding to risks related to the project
- Project procurement management: involves acquiring or procuring goods and services for a project from outside the performing organization
- Project stakeholder management: includes identifying and analyzing stakeholder needs while managing and controlling their engagement throughout the life of the project
- Project integration management: is an overarching function that affects and is affected by all of the other knowledge areas

Project Stakeholders

- **Stakeholders** are the people involved in or affected by project activities
- Stakeholders include
  - the project sponsor
  - the project manager
  - the project team
  - support staff
  - customers
  - users
  - suppliers
  - opponents to the project

IS projects types

- Software development project:
- Package implementation project
- System enhancement
- Consultancy and business analysis assignment
- System migration projects
- Infrastructure implementation projects
- Outsourcing and in-sourcing projects
- Disaster recovery projects
- Smaller IS projects

Software development project

- Group of people working together to specify, design, implement, test a software product => new software application
- Main problem: supplier and customer have different ideas about what is to be done => ambiguity
- Project manager should be:
  - Flexible in specifying the requirements
  - Having the skills to deal with stakeholders

Package implementation project

- Buy a software, install it, switch it on and use
- Problems:
  - For the customer to choose the best package that satisfies his requirements
  - For the supplier: modifying the package to satisfy the customers' needs
  - Integration problem
- Example: E-learning
- The project Manager's responsibilities:
  - Ensuring that the feature claimed by the supplier are really there
  - Package customization
  - Data migration
  - Connect with existing systems
  - User training

System enhancement

- To provide new features or functions or to meet external demands
- Small changes no need for a project => IT support team can do the job
- But large changes require a new project=> time, cost and effort
- Example: Registration system from old to new
- Manager responsibility:
  - keeping the existing system operational while work proceeds on the enhancements

Consultancy and business analysis assignment

- Here we investigate a business issue and propose solutions using information technology.
- Problem:
  - Difficult to estimate, plan for and control.
  - One may not quite sure what the problem is or where the suitable solutions maybe found
- Manager should be:
  - Flexible in determining the budget and timescale

Systems migration projects

- An existing operational system has to be moved to a new operating environment
- Issues for project manager:
  - May need to develop new software components (parts)
  - New platforms may not work exactly like the old ones => compatibility issues
  - Infrastructure implications
  - Retraining of users to enable them to utilize the new environment

infrastructure projects

- Includes ones to introduce or replace hardware, servers or PCs, for example, to put in place communications infrastructure and also the physical construction of things like computes.
- Manager responsibilities:
  - Maintain "business as usual " while putting in place the new infrastructure
  - Supplier management features

outsourcing (and in-sourcing) projects

- Why outsourcing:
  - Utilize external expertise
  - Difficulty in managing the IT estate internally
  - Reduce costs through lowing employees.
- Some issues involved:
  - Physical movement of some assets and infrastructure such as servers from one place to another.
  - Establishing who will have the ownership of assets, hardware and infrastructure.
  - Organizing the transfer of software licences from one party to another
  - Migrating people's contracts
  - Dealing with severance terms for people who don't wish to move
  - Recruiting new staff to replace people who have left

Disaster Recovery Projects

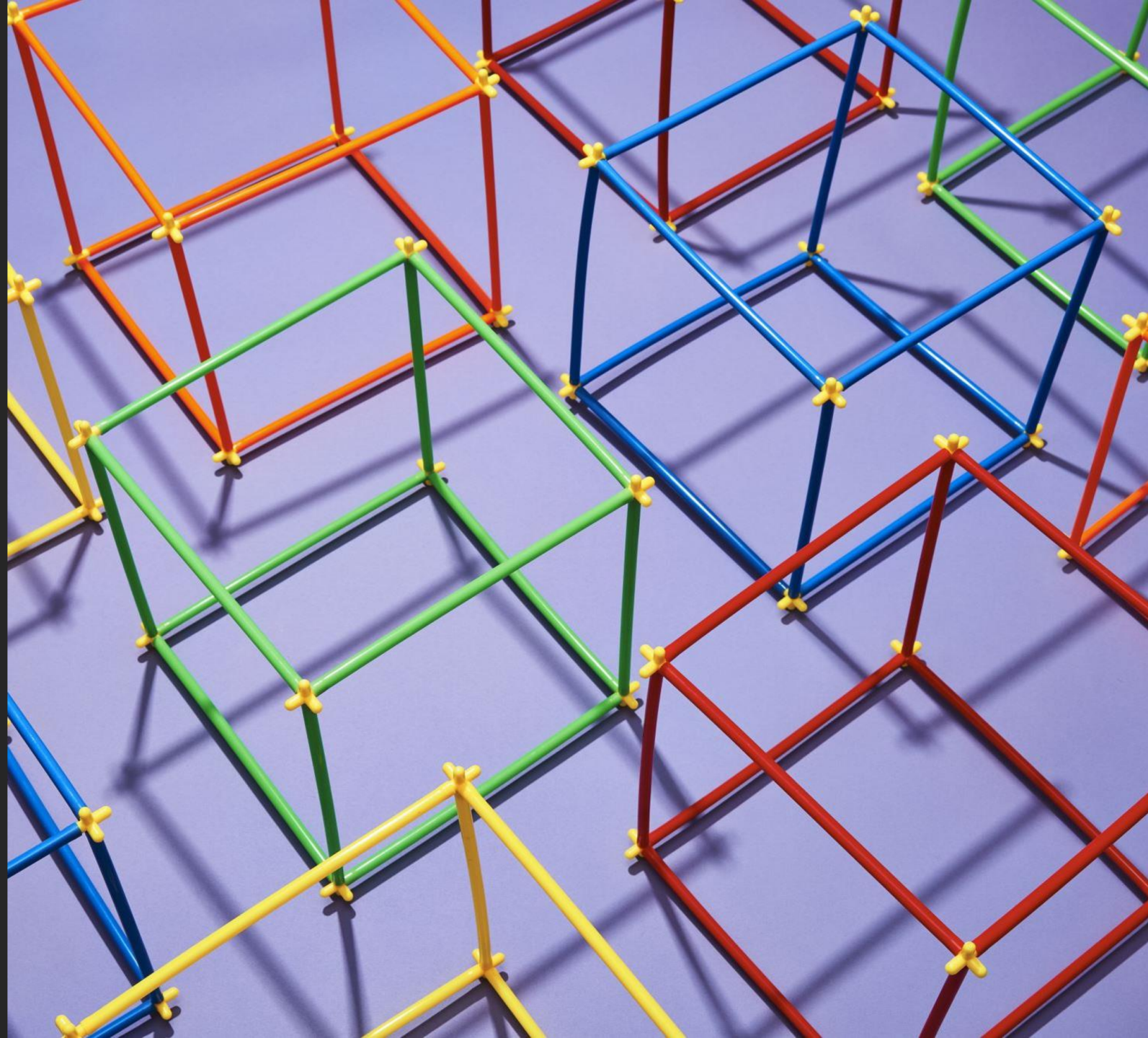
- A large scale failure in the system and the organization needs to get its systems back up and running as soon as possible.
- Examples: Fire, freezing, floods, terrorist attacks, power failure.
- Put in place adequate defensive measures to **prevent** the occurrence of the various threats.
- Pre-planning of the recovery process
  - Well thought out disaster recovery plan covering all possible scenarios.
  - Arrangements with suppliers of DR services and resources (business continuity service) such as alternative workplaces and data centres
  - Up to date list of key personnel with essential contact details.
  - Equipment stored and accessible if needed e.g. laptops and printers

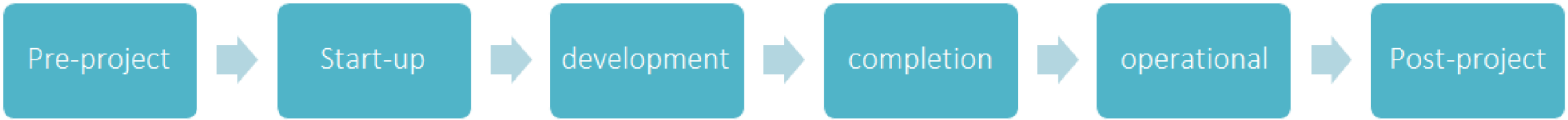
Smaller IS projects

- If a small project is needed, it is practical to adjust the project management approach to the size of the project, PID (project initiation document) could only be 2 pages long, simple bar chart showing the main steps.

INFORMATION SYSTEMS PROJECT MANAGEMENT

CHAPTER 2: PROJECT MANAGEMENT PROCESS MODEL





IS PROJECT MANAGEMENT PROCESS MODEL

PRE-PROJECT WORK

- Extensive discussions between the customer and supplier to establish the objectives and scope of the project and to agree on a suitable contractual framework within which the project will take place.
- This stage Includes:

1. The customer prepares a specification of his requirements and issues invitations to tender (ITTs) to suppliers whom he thinks may be able to carry out the project.
2. The suppliers will then respond with their tenders and the customer will subject these to a detailed evaluation process.

WHAT IS INVITATION TO TENDER ITT

Is the initial step in competitive tendering, in which suppliers and contractors are invited to provide offers for supply or service contracts, the ITT is one process in IT procurement.

An ITT document specifies all requirements of the organization, including goods, services and timelines, as well as the evaluation process that will be followed.

PROJECT START-UP

Project start-up covers the work that is carried out at the beginning of the project when the basic framework is put in place, it involves:

- What is to be carried out?
- Why is it being done?
- Who is going to do it?
- How and When it will be accomplished?

Products of Project start-up

- Project initiation document (PID)
- Project plan
- Quality plan
- Risk management plan
- Project organization structure
- Project administrative procedures

PID (PROJECT INITIATION DOCUMENT

- produced by the project manager and approved by the project sponsor
- Completing the sections of the OSCAR format will ensure that the most important project start-up issues are addressed

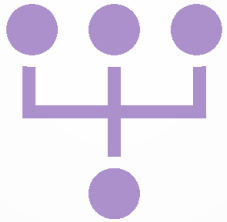
OBJECTIVES:

The business objectives: these are the business case for the project, for example ‘to increase our market share by 25 per cent’, or ‘to enable us to handle 25 per cent more patients without an increase in manpower’.

The project objectives: are narrower and specify exactly what the project itself is to deliver, for example ‘to implement an e-commerce trading system’, or ‘to implement an automated appointment scheduling system’.

SCOPE

Boundaries: which areas, departments or functions or included and/or excluded.



Activities: which tasks the project team is and is not undertaking.

Deliverables: what will be produced and/or handed over to the customer at the conclusion of the project.

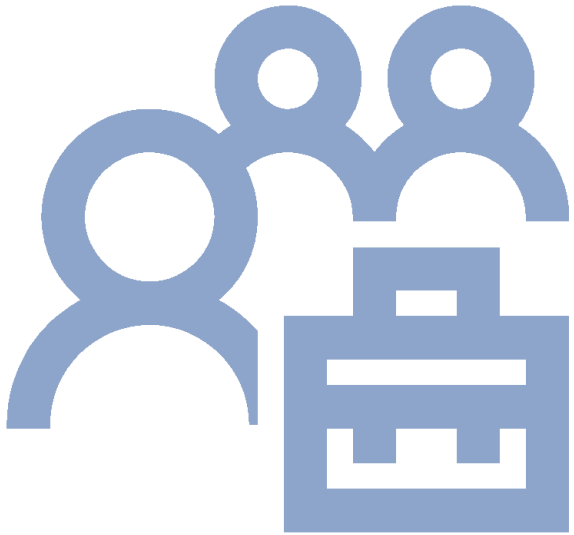
CONSTRAINTS

- methods and standards to be used.
- the hardware and software platforms that will be involved and any legislation or organizational policy that must be complied with.
- time, for example that a system must be available by some date.
- Resources, such as people, equipment or money, “these are probably best dealt with under their own heading”.

AUTHORITY

- Who is the customer for the project.
- Who can authorize it in the first place.

RESOURCES



- people, money, equipment and so on – required to execute the project

DEVELOPMENT STAGE

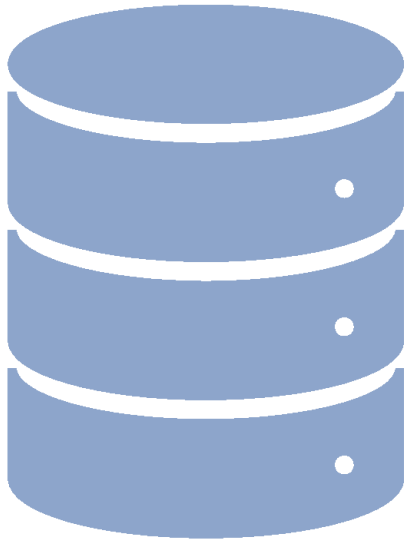
- This stage is where the supplier's work is carried out, although the customer's project manager will have an overall control of the project.
- Parts of the Development stage:
  - Requirements definition
  - Design
  - Implementation
  - Integration and testing
  - System testing

DEVELOPMENT STAGE

The main products resulting from the development stage are as follows

- Requirements specification
- Technical specification
- Module specifications
- Prototypes
- Completed and tested hardware and software modules
- Acceptable systems test results
- Factory acceptance certificate

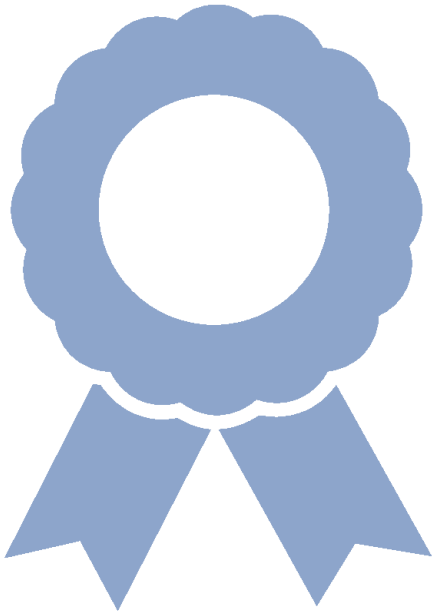
Note: factory acceptance means that the customer has tested the system on its development site and pronounced it to be satisfactory.



COMPLETION STAGE

- Delivery to the customer
- Training and familiarization
- Acceptance testing
  - Functionality testing: according to the requirements
  - Performance testing: response time, number transactions per second, number of users logged on
  - Interface testing: check that the system works well with other systems
  - Environmental testing: power consumption, heat dissipation, noise..
- Acceptance by the customer: the system is officially accepted by the customer through a signed certificate
- System commissioning: set the system in its final environment for live running
- Final customer takeover: formally accepts the system and the project comes to an end, minor faults are accepted to be corrected within a specified time.

COMPLETION STAGE



- Products of completion stage:
  - Site acceptance certificate
  - Trained staff
  - Commissioned system

OPERATIONAL STAGE

- The Operational stage takes over when **live running** begins.
- It is likely that the system will require maintenance and enhancement.
- **Faults** arise during live running that were **not discovered** during testing of the system
- Products of operation stage:
  - Fixes
  - Enhancements

POST PROJECT REVIEW

- The post-project review should address the following issues:
  - The technical methods and standards used, and how effective these proved.
  - Customer/supplier relationship issues.
  - Project risks – how effective were the methods used to identify, assess and manage risks.
  - Contractual issues – what they were and how they were resolved.
  - Stakeholder management issues.
  - Team resourcing issues.
  - Project performance against plans, with a view to updating and improving the planning and estimating methods used.

CREATING A PROJECT IN JIRA: EPICS, STORIES AND TASKS

Dr. Rana Yousef

EPICS

- In Jira, an epic is a large body of work that can be broken down into smaller, more manageable tasks or stories.
- Epics provide a way for teams to track progress on a high-level goal or initiative.
- Epics can be used to group related stories and tasks together, making it easier to manage and prioritize work.
- Epics can have multiple stories associated with them, and each story can be broken down into individual tasks or sub-tasks.
- In Jira, epics are represented by a unique issue type that can be used to track progress, assign tasks, and communicate updates to the team.

EXAMPLE: EPICS FOR A PHARMACY MANAGEMENT SYSTEM

Medication Management: This epic would include features such as medication dispensing, inventory management, and prescription filling. It would allow pharmacists to track medication usage, manage inventory levels, and ensure that prescriptions are filled accurately.

Patient Management: This epic would include features such as patient registration, medical history, and medication history. It would enable pharmacists to keep track of patient information, medication allergies, and any other medical conditions that may impact the dispensing of medication.

Reporting and Analytics: This epic would include features such as reporting on medication usage, inventory levels, and financial metrics. It would enable pharmacists to gain insights into their business and make data-driven decisions.

EXAMPLE: EPICS FOR A PHARMACY MANAGEMENT SYSTEM

Security and Compliance: This epic would include features such as user authentication, access control, and compliance with HIPAA regulations. It would ensure that patient data is secure and that the pharmacy is following all legal requirements for the handling of medical information.

Customer Relationship Management: This epic would include features such as loyalty programs, customer feedback, and communication tools. It would enable pharmacists to build stronger relationships with their customers and improve the overall customer experience.

Integration with Electronic Medical Records (EMR): This epic would include features such as integration with electronic medical records systems. It would enable pharmacists to access patient information directly from the EMR and streamline the medication dispensing process.

USER STORIES AND TASKS

- In Jira, a story is a user-centric requirement that describes a piece of functionality from the perspective of a user or customer. A story is typically written in a specific format, such as "As a [user persona], I want [functionality], so that [reason or benefit]."
- A task, on the other hand, is a piece of work that needs to be completed in order to deliver a user story or other piece of work. A task is typically written as a specific action that needs to be taken in order to move a project forward.
- In Jira, stories and tasks are both types of issues that can be tracked and managed as part of a project. Stories are typically used to represent user-facing requirements or features, while tasks are used to represent the work that needs to be done to deliver those features.

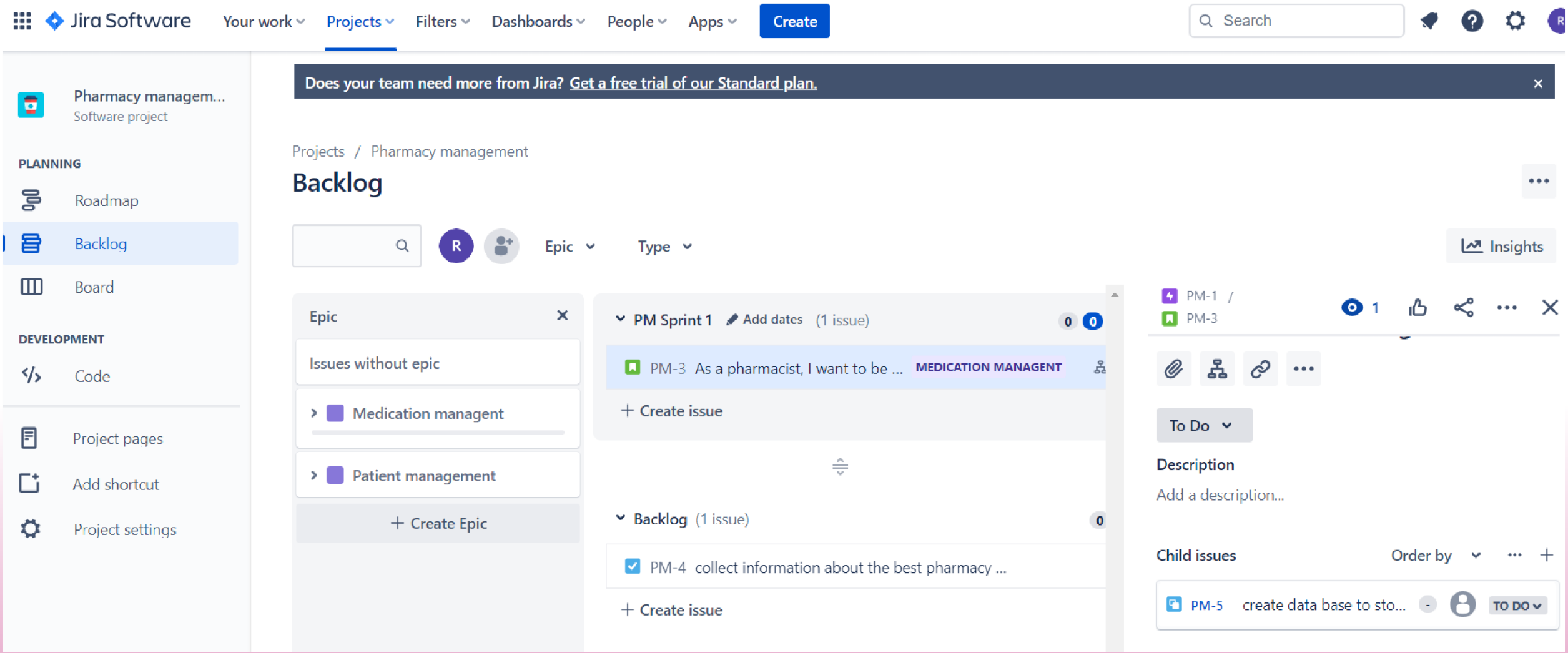
- As a pharmacist, I want to be able to track medication usage so that I can ensure that patients are receiving the appropriate medication and dosage.
  - Task 1.1: Create a database to store medication usage data
  - Task 1.2: Develop a system for pharmacists to record medication usage
  - Task1.3: Implement a notification system to alert pharmacists if a patient's medication usage is outside of normal parameters

- As a pharmacist, I want to be able to manage inventory levels so that I can ensure that the pharmacy has an adequate supply of medication.
- Task 2.1: Develop a system for tracking medication inventory
- Task 2.2: Implement automatic reordering when inventory levels fall below a certain threshold
- Task 2.3: Develop a system for tracking medication expiration dates

- As a pharmacist, I want to be able to fill prescriptions accurately and efficiently.
- Task 3.1: Develop a system for verifying patient information and medication dosage
- Task 3.2: Implement a barcode scanning system for medication dispensing
- Task 3.3: Develop a system for tracking prescription history and refills

- As a pharmacist, I want to be able to communicate with physicians to ensure that patients are receiving appropriate medication.
- Task 4.1: Develop a system for communicating with physicians
- Task 4.2: Implement a system for receiving electronic prescriptions
- Task 4.3: Develop a system for requesting medication changes or clarifications from physicians

- As a pharmacist, I want to be able to provide medication counseling to patients.
- Task 5.1: Develop a system for scheduling medication counseling sessions
- Task 5.2: Train pharmacists on medication counseling best practices
- Task 5.3: Develop a system for documenting medication counseling sessions


Jira Software
Your work
Projects
Filters
Dashboards
People
Apps
Create
Search
Pharmacy managem...
Software project
PLANNING
Roadmap
Backlog
Board
DEVELOPMENT
Code
Project pages
Add shortcut
Project settings
Does your team need more from Jira? Get a free trial of our Standard plan.
Projects / Pharmacy management
Backlog
Epic
Type
Insights
Epic
Issues without epic
Medication managent
Patient management
Create Epic
PM Sprint 1
Add dates
(1 issue)
PM-3 As a pharmacist, I want to be ...
MEDICATION MANAGENT
Create issue
Backlog (1 issue)
PM-4 collect information about the best pharmacy ...
Create issue
PM-1 /
PM-3
1
To Do
Description
Add a description...
Child issues
Order by
PM-5 create data base to sto...
TO DO

Project Initiation

Instructor: Dr. Rana Yousef

Project Initiation

- Initiating a project includes recognizing and starting a new project: starts with an idea or need (by the sponsor)
- The table shows the project initiation knowledge areas, processes, and outputs

Knowledge Area	Initiating Process	Outputs
Project Integration Management	Develop project charter	Project charter
Project Stakeholder Management	Identify stakeholders	Stakeholder register

Aspects of initiation phase

- Not every project survives the initiation phase
- There are six aspects of the initiation phase that a potential project must survive:
  - Goal
  - Scope
  - Feasibility study
  - Risks, assumptions and expectations
  - Project team
  - Project charter

Goal

- A project should deliver a benefit to the organization and fit its strategy.
- First, we must explain the project's purpose for the business, the business case, where all costs and potential gains will need to be carefully estimated, and all financial elements must be described, revenues, expenses, expected inflation, discounted cash flow.
- If the business case is approved, this often serves as the official sign off to start the project.

Scope

- The project manager and sponsor use the team's expertise when defining the scope to decide what is within and what is outside the project's scope
- So they set the boundaries, activities and deliverable

Feasibility Study

- An analysis of the goals, scope and resources to determine whether the success of the project is feasible.
- For now, we only ask: can the organization provide the needed budget, time, employees so the project can be executed?
- A more detailed analysis and preparation of resources is performed during the planning phase.

Risks, assumptions and expectations

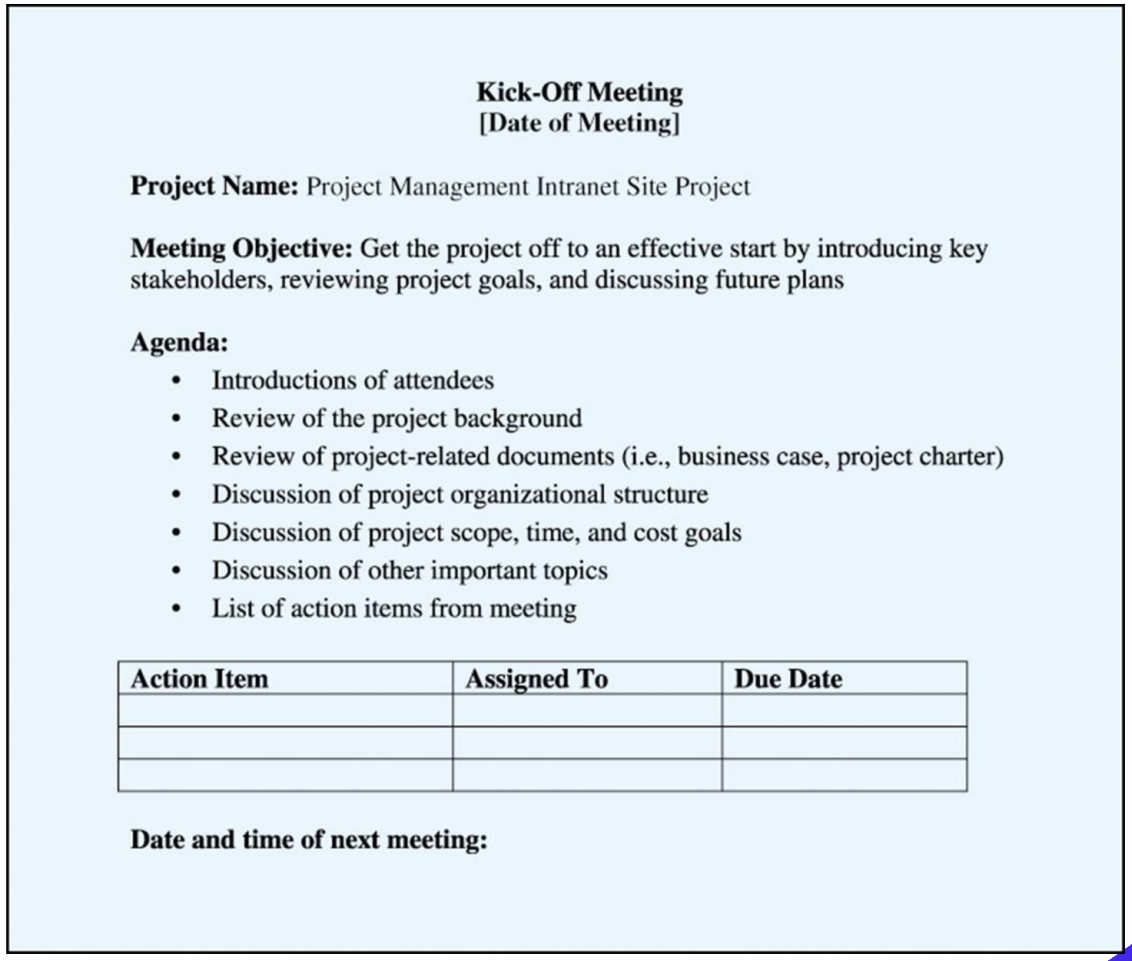
- Risks must be identified, analyzed and responded to.
- More risk details will be covered during the planning phase .
- But for now, one area that needs to be addressed is everyone's expectations.
- Does everybody share the same vision as each other?
- Are they expecting the same resources and results?

Project Charter

- It is a high-level document, meaning it's easy and quick to understand,
- Must contain all the fundamental information about the project.
- Charters are normally short and include key project information and stakeholder signatures

Kick-off Meetings

- It's good practice to hold a kick-off meeting at the beginning of a project so that stakeholders can meet each other, review the goals of the project, and discuss future plans



Kick-Off Meeting
[Date of Meeting]

Project Name: Project Management Intranet Site Project

Meeting Objective: Get the project off to an effective start by introducing key stakeholders, reviewing project goals, and discussing future plans

Agenda:

- Introductions of attendees
- Review of the project background
- Review of project-related documents (i.e., business case, project charter)
- Discussion of project organizational structure
- Discussion of project scope, time, and cost goals
- Discussion of other important topics
- List of action items from meeting

Action Item	Assigned To	Due Date

Date and time of next meeting:

Project Planning

Understanding the work

Why do we need a plan for IS projects?

- Developing an information system is complex as it involves various elements: hardware, software, data capture, user training, etc..
- The people involved in a project need to know exactly what their role is.
- Customers want to be confident that the developers know they are about.
- The project manager need to know whether the project is on schedule, ahead or behind and whether corrective action is needed.

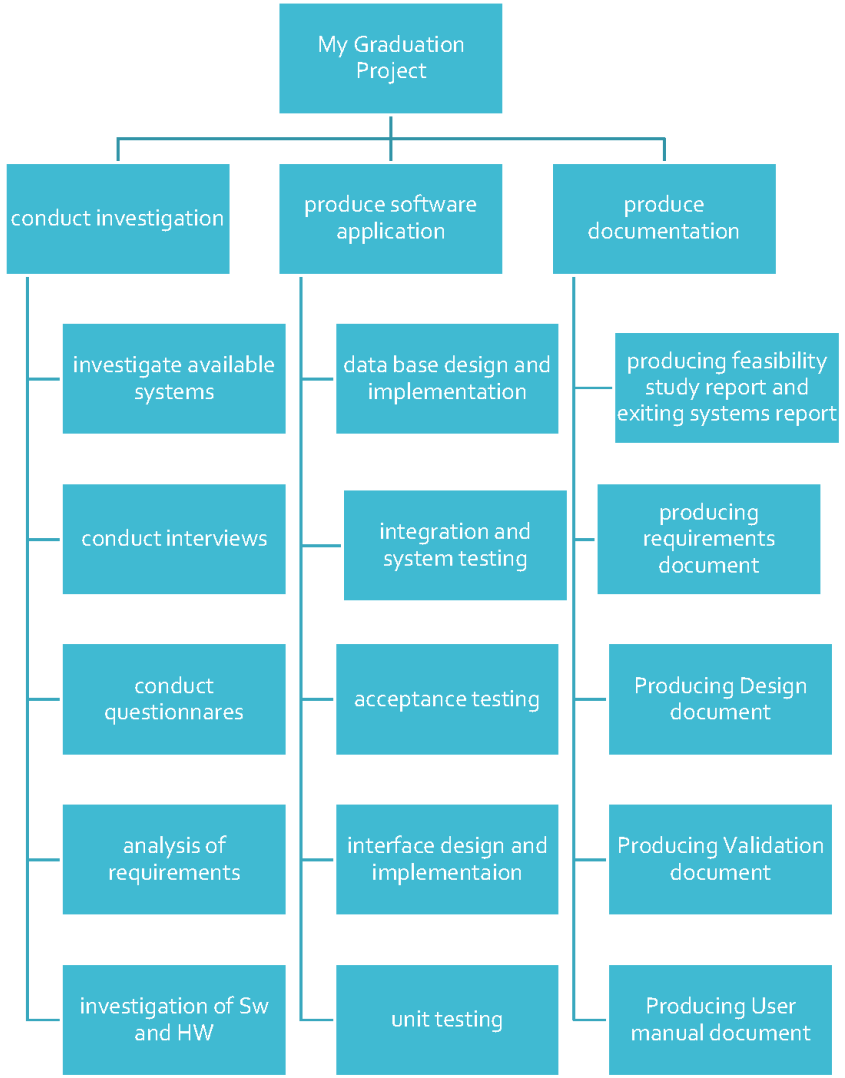
Breaking down the work

- The work breakdown structure (WBS)
- The product breakdown structure (PBS)

The work breakdown structure (WBS)

- More traditional approach, we keep breaking the project down progressively into smaller and smaller chunks until we end up with individual tasks or work packages that we can estimate sensibly and assign to team members.
- The work packages should be:
- Fairly atomic, that is, can't be subdivided or assigned to more than one person
- small enough to estimate with reasonable accuracy 0.5-2 days long

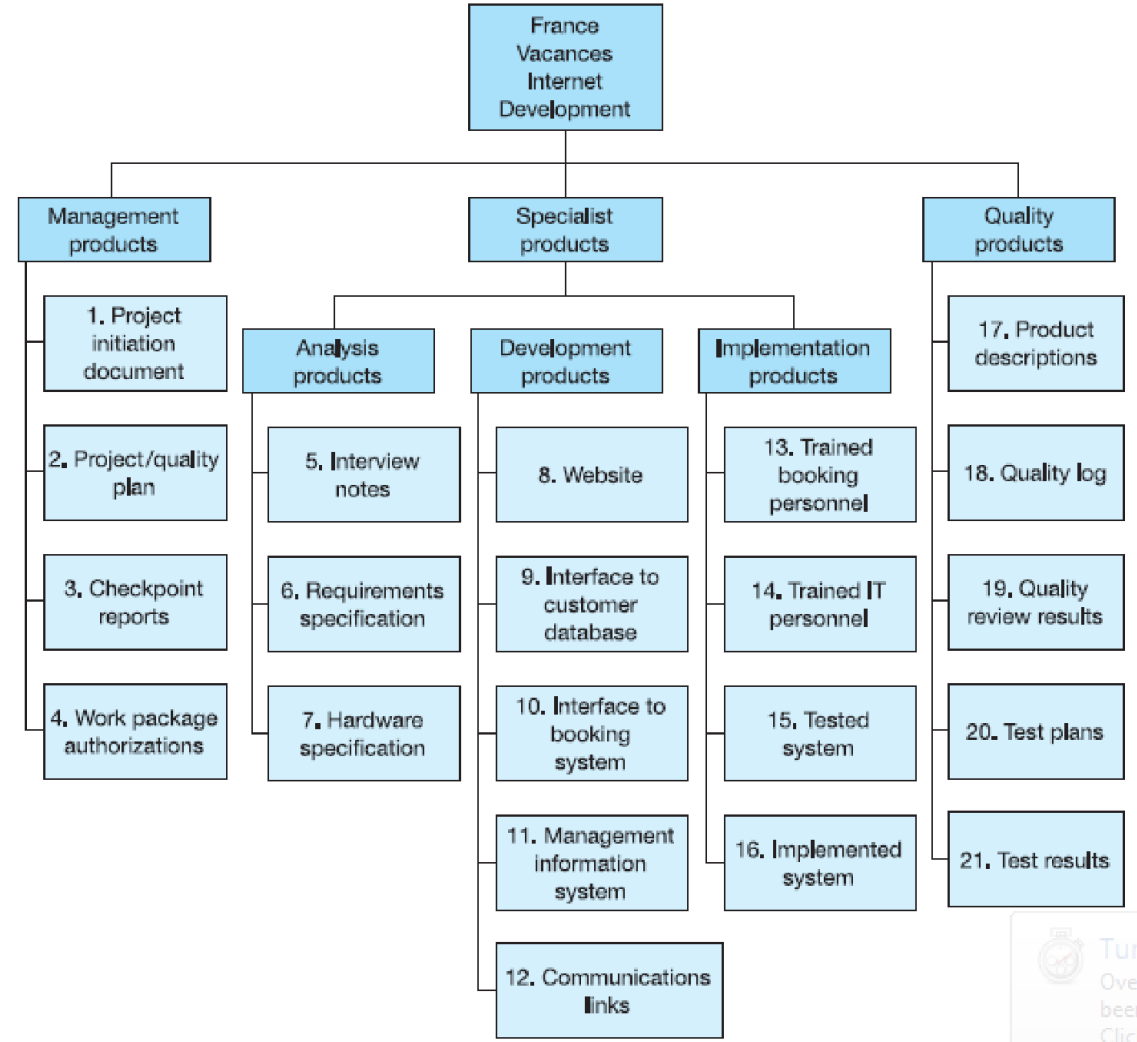
Example: CIS students Graduation project:



Product break down structure (PBS)

- This approach underpins the PRINSE2 project management method (https://en.wikipedia.org/wiki/PRINCE2)
- It ensures that the Project's focus is on what is to be achieved rather that how
- It is easier to consider what one has to develop
- It is less easy to forget something in the plans
- PBS works by progressively decomposing the project products into smaller products until a sensible, unitary product level is reached.
- In PRINCE2, the top level of products is known as 'project products'. These subdivide into two main categories: management products and specialist products.

Example


France Vacances Internet Development
Management products
Specialist products
Quality products
1. Project initiation document
2. Project/quality plan
3. Checkpoint reports
4. Work package authorizations
Analysis products
Development products
Implementation products
5. Interview notes
6. Requirements specification
7. Hardware specification
8. Website
9. Interface to customer database
10. Interface to booking system
11. Management information system
12. Communications links
13. Trained booking personnel
14. Trained IT personnel
15. Tested system
16. Implemented system
17. Product descriptions
18. Quality log
19. Quality review results
20. Test plans
21. Test results

Work Packages:

- sometimes it makes more sense to group a set of products together into a work package

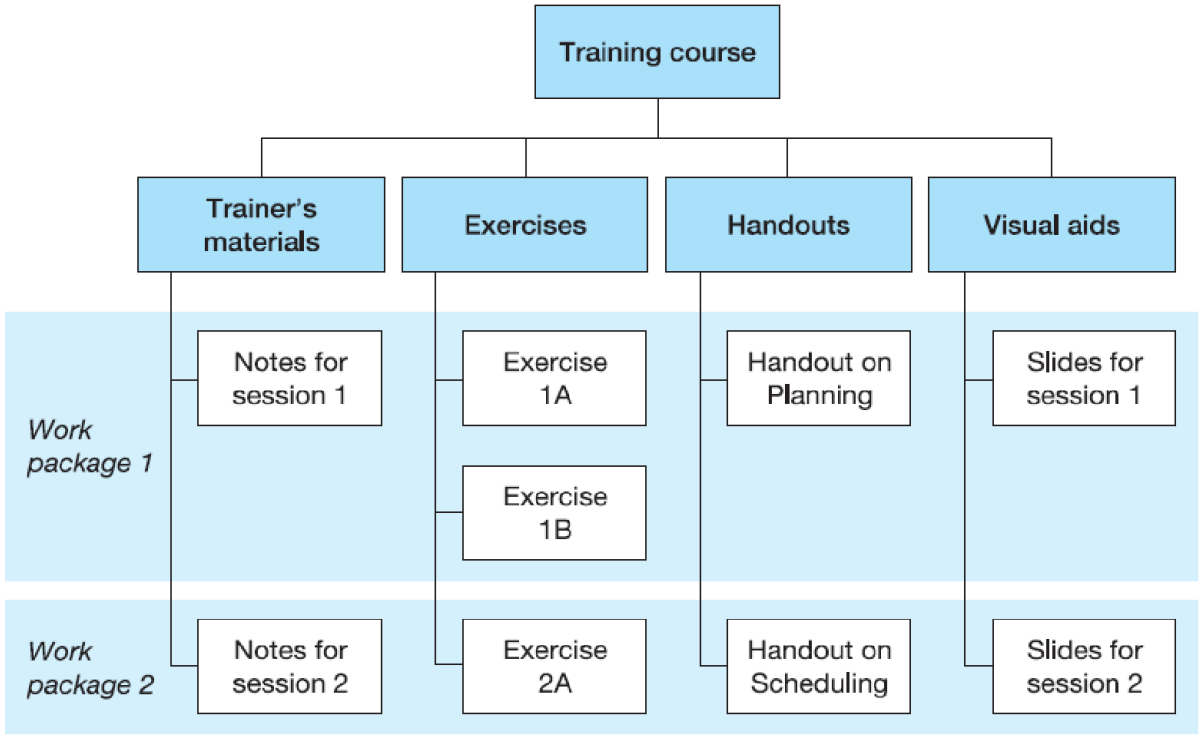


Figure 8.10 Work packages for a training course

Project Planning: Understanding the work

Understanding dependencies

Dependencies are fundamental to planning a project and, later, in understanding the effects of any problems encountered. In essence, understanding dependencies is simple. If activity B can begin only when activity A is complete, then we have a dependency.
We can analyse dependencies using a **network diagram** also known as a dependency diagram or **PERT chart.**

Once we have estimated the effort involved in each activity, we can use the network to establish another important feature of the project. Let us suppose that we have estimated the effort for each activity as follows:

- Conduct interviews 8 days
- Investigate other systems 4 days
- Analyse requirements 3 days
- Investigate packages 8 days
- Investigate hardware 5 days
- Produce report 5 days

We can analyse dependencies using a network diagram, also known as a dependency diagram or PERT chart. A network diagram for our example project is shown in Figure 8.12. This diagram has been drawn using a format known as activity-on-arrow, which means that the lines represent project tasks and the circles the connections between tasks. From the diagram, we can read the following:

Once the project starts, we have two activities – Conduct interviews and Investigate other systems – that can proceed in parallel.

But the results of both activities have to be brought together before we can start the next three activities – Analyse requirements, Investigate packages and Investigate hardware. We show this bringing together by using a 'dummy' activity, one with zero duration, indicated by the dotted line.

The three activities are then brought together – again using two dummy activities – before we can start the last task of our project, Produce report.

This very simple structure has already told us one important thing about our project: we can use more than one person on it if we wish, working independently until such time as their work must be brought together.

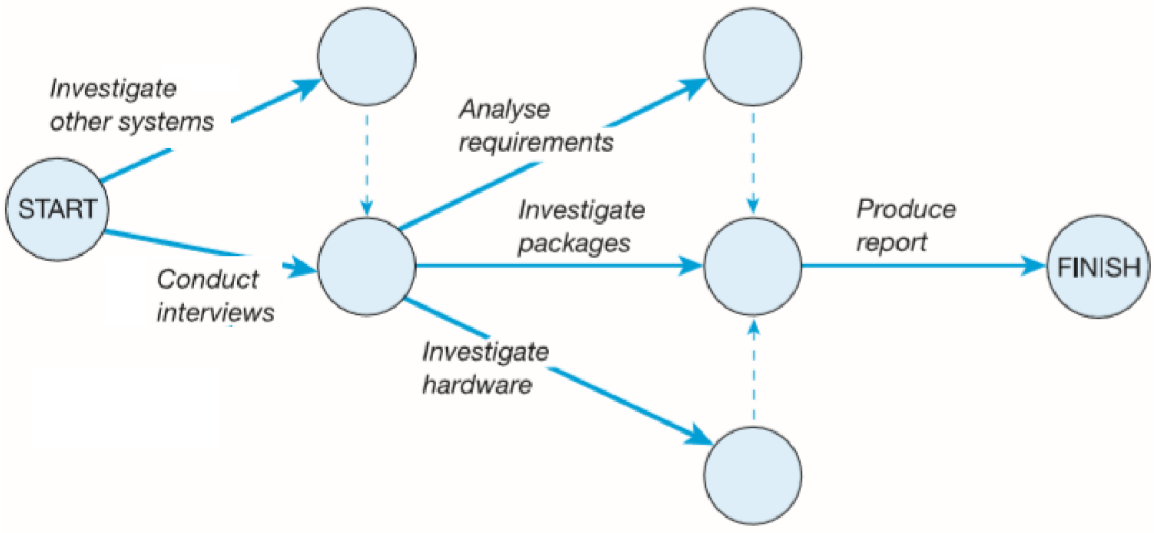


Figure 8.12 Network diagram (activity-on-arrow format)

Another method of producing a network diagram is a convention known as 'activity-on-node', where boxes represent the activities and the arrows simply represent the dependencies between them.

With the activity-on-node method, we analyse the critical path by making two passes through the model. In the forward pass, we establish the earliest start time (EST) and earliest finish time (EFT) for each activity, and we do this by addition, as shown in Figure 8.16.

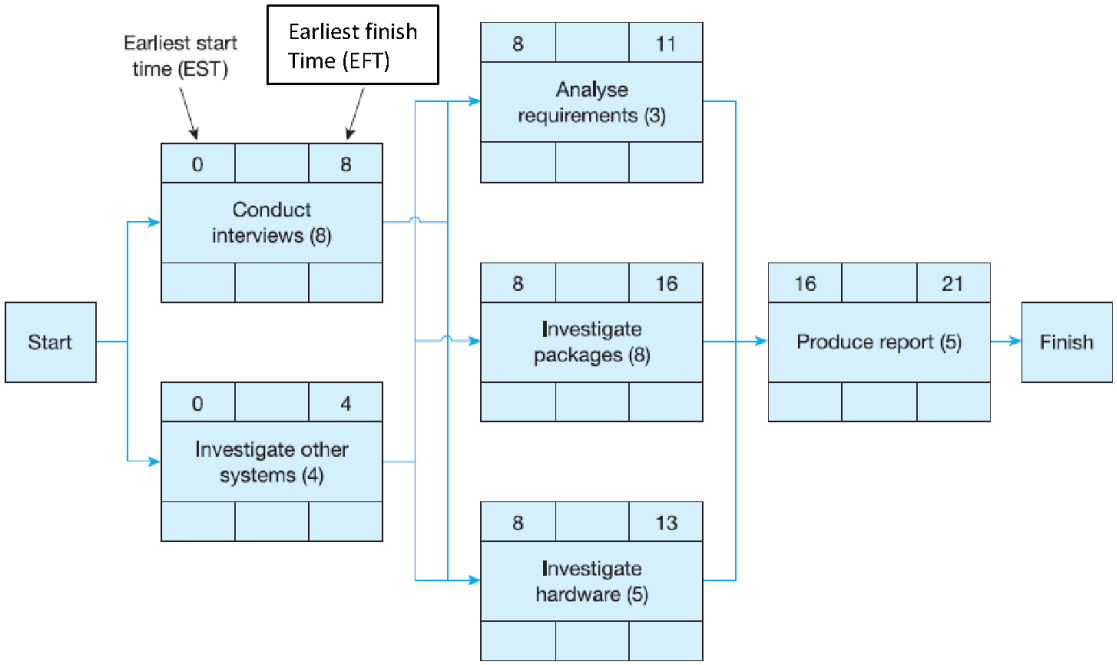


Figure 8.16 Network diagram after forward pass

Next, we perform a backward pass through the model to establish the latest start time (LST) and latest finish time (LFT) for each activity, this time by subtracting durations from our EFT for the whole project of 21 days. This is illustrated in Figure 8.17.

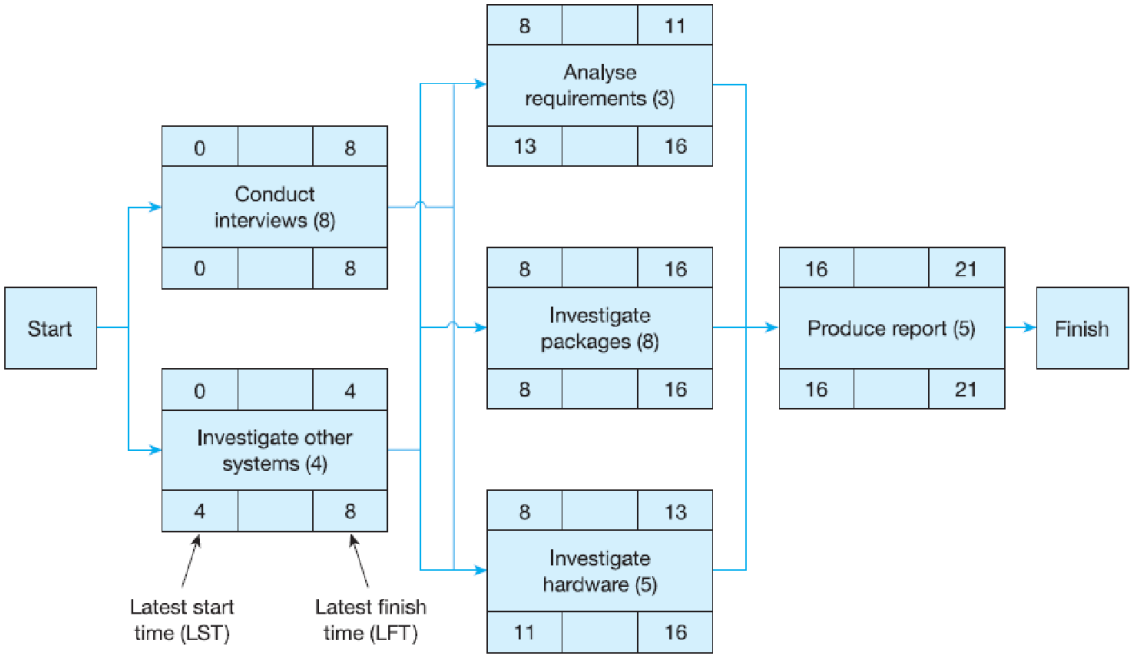


Figure 8.17 Network diagram after backward pass

We are now able to calculate the critical path by subtracting the EST from the LST for each activity or – which amounts to the same thing – by subtracting the EFT from the LFT. If there is no remainder from the subtraction, the activity is on the critical path; if there is a remainder, then this is equivalent to the float or slack there is in the activity. Figure 8.18 shows our network diagram with the critical path highlighted.

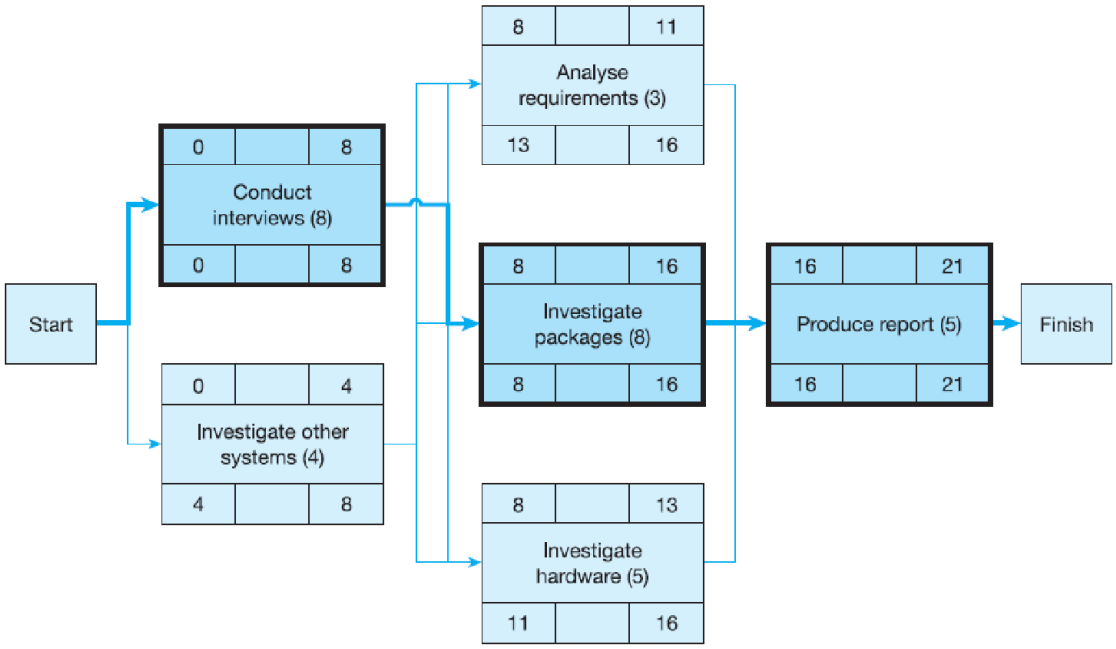


Figure 8.18 Network diagram with critical path activities highlighted

Project Planning: Scheduling and resourcing (Part 1)

Instructor: Dr. Rana Yousef

Scheduling

Effort and Elapsed time

Suppose that we have estimated a task as requiring 20 days' effort. Assuming that we have only one person available to do the work, then – if our person has no absence during the period- this 20 days' work will take 20 elapsed days to perform. If we have two people available, and the work can be partitioned, then the 20 days' work can be accomplished in 10 elapsed days. With four people, it would take 5 elapsed days. In practice, partitioning of activities is not usually as straightforward as this. In producing our project schedule, it is vital that we keep the distinction between effort and elapsed time in mind. Usually, the project manager cannot do much about the effort required to perform an activity, since the amount of work is inherent in the task itself. But the project manager can and must seek to influence the elapsed time by committing the right amount of resources to each task.

Developing the schedule

To develop our initial schedule, we might decide to see how long the project would take with one analyst assigned to the work.

Let's go back to our example where we have identified the following tasks and durations:

- Conduct interviews 8 days
- Investigate other systems 4 days
- Analyse requirements 3 days
- Investigate packages 8 days
- Investigate hardware 5 days
- Produce report 5 days

Figure 10.2 shows that the elapsed time for the project would be the sum of all of the activities – 33 days.

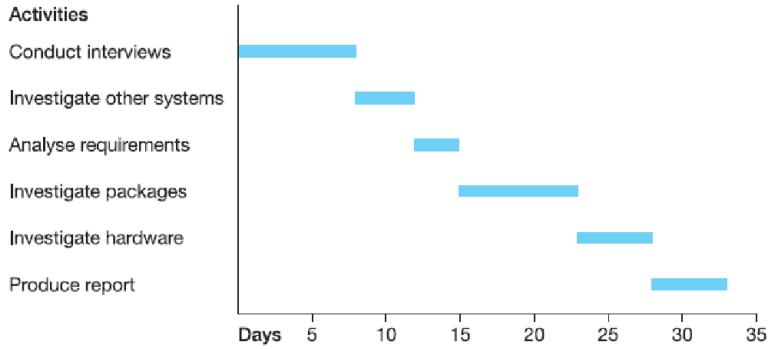


Figure 10.2 Schedule for one-person team

we have to examine our network to see if any activities can be performed in parallel. We find that:

- ⇨ Conduct interviews can be progressed in parallel with Investigate other systems.
- ⇨ Analyse requirements can be done in parallel with both Investigate packages and Investigate hardware.

If we have a second analyst available, then we can take advantage of this parallelism, as shown in Figure 10.3. In this plan, we have used one analyst on Conduct interviews and the other on Investigate other systems. we have shortened the elapsed time of the project to 21 days.

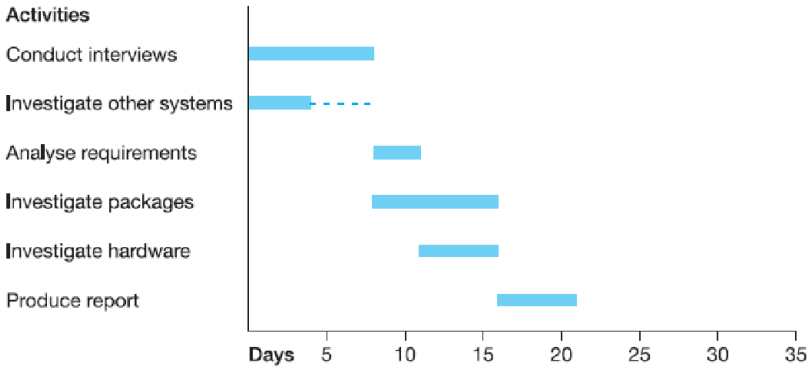


Figure 10.3 Schedule for two-person team showing parallel activities

if we had a third analyst available, we could partition the work again and shorten the timescale still further Figure 10.4 illustrates how we could do this.

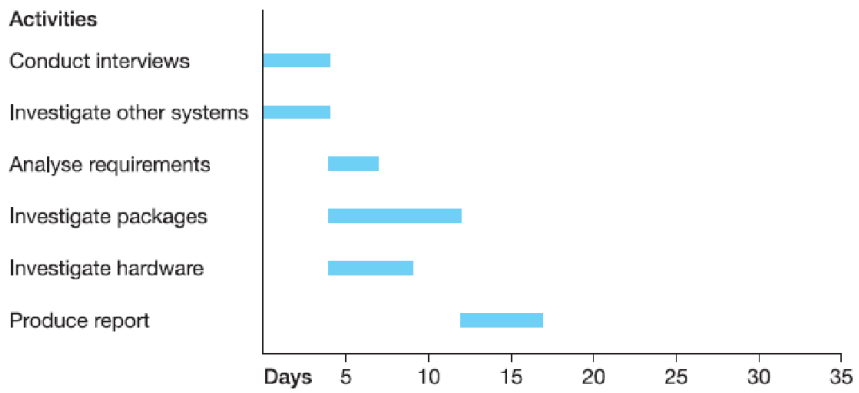


Figure 10.4 Schedule for three-person team

We have used two people on Conduct interviews, so the 8 days' effort now takes only 4 elapsed days. Similarly, we have used a different analyst on each of Analyse requirements, Investigate packages and Investigate hardware but in this case we do not gain anything since the elapsed time becomes the 8 days' effort of the longest of these activities, Investigate hardware. As a result we can now offer the customer the report in 17 elapsed days.

Although the feasibility study is a simple example, it does illustrate the approach that is used whatever the size of the project. An initial schedule is created and then it is adjusted and revised until the project manager is sure that it is realistic and provides a reasonable balance between the effective use of resources and the achievement of an acceptable end-date. There are, however, some other factors to consider in developing the project schedule.

Scheduling considerations:

Suppose you have an activity for an analyst to produce a report comparing eight different relational databases. Tasks included are:

1. Study some background material and decide what evaluation criteria will be used. => 4 days
2. Read the technical literature about each of the eight databases and note how each performs against the evaluation criteria. => 4 days
3. Place the results of all the evaluations side by side on a table and compare them =>4 days
4. Write a report documenting the findings => 4 days

One analyst will require : 16 days

If two analysts are doing this job?

We can't just say 16/2 = 8 days, why?

Task 1 => 4 days

Task 2 =>2 days

Task 3 => 2 days (provide that 2 pcs are available, otherwise -if one pc is available 4 days are required)

Task 4 = > 3 days (2 days writing the report and 1 day for discussion and negotiation)

Total 11 days.

Rule of thumb:

We can't divide the number of days required for the tasks on the number of worker.

A full time job for a worker is not 5 days per week. Why?

Total working days per year (52 * 5) = 260

Holidays = 33 days

Other non-working days (training, sickness) = 15

- ⇨ 212 days
- ⇨ Per month 212/12 = 17.66 =18 days
- ⇨ Per week 212/52 = 4.07
- ⇨ We say that full time job is 4 days per week.

If we adjust the schedule in Figure 10.4 on this basis, we get a more realistic plan as shown in figure 10.5

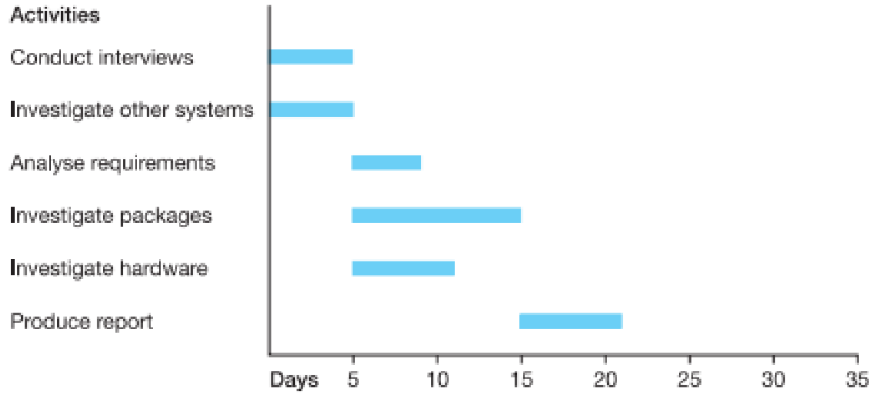


Figure 10.5 **Schedule adjusted for four days per week availability**

Project Management Tools

MICROSOFT PROJECT

Microsoft Project

Microsoft Project is a project management software program developed by Microsoft. It allows users to plan and manage projects, track progress, and collaborate with team members. Microsoft Project is widely used in a variety of industries, including construction, engineering, and IT.

Microsoft Project is available as a desktop application or as a cloud-based service through Microsoft 365. It is designed for use by project managers and other professionals who need to plan, execute, and track projects.

Features of Microsoft Project

1. Task management: Microsoft Project allows users to create tasks and organize them into a project plan.
2. Resource management: Users can assign resources such as people, equipment, and materials to tasks, and track their availability and usage.
3. Gantt charts: Microsoft Project can create Gantt charts, which provide a visual representation of a project's schedule and progress.
4. Budget management: Users can create and manage project budgets, including tracking expenses and costs.
5. Collaboration: Microsoft Project allows team members to collaborate on projects by sharing files, calendars, and project information.

Features of Microsoft Project

6. Timeline view: This feature provides a high-level overview of a project's schedule, making it easier to see the overall timeline and identify potential issues.
7. Critical path analysis: Microsoft Project can identify the critical path of a project, which is the sequence of tasks that must be completed on time in order to ensure the project is completed on schedule.
8. Risk management: Users can identify potential risks to a project and develop strategies to mitigate those risks.
9. Reporting: Microsoft Project includes a variety of reporting options, including Gantt charts, resource allocation reports, and budget reports.
10. Integration with other tools: Microsoft Project can integrate with other tools such as Microsoft Excel, Microsoft Teams, and SharePoint, allowing for easier collaboration and data sharing.
11. Customization: Microsoft Project can be customized to meet the specific needs of a project or organization, such as creating custom fields or workflows.

Project Planning: Scheduling and resourcing (Part 2)

Instructor: Dr. Rana Yousef

Milestones

Points in the project where you assess progress and/or deliver some deliverable.

+ve:

- ⇨ They provide useful control points at which we can evaluation progress and adjust our plans for the rest of the project as necessary
- ⇨ They can be used to illustrate progress to the customer
- ⇨ They may be intermediate sign-offs or stage payments linked to the achievement of milestones.

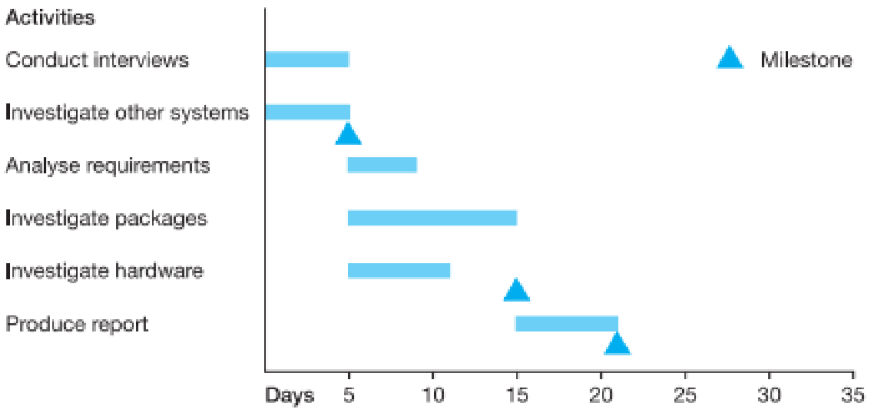


Figure 10.6 **Bar chart showing project milestones**

Showing overhead tasks on schedule:

Overhead tasks are project management and administrative tasks, meetings ,…

We present it as an extra line for project management at the end of the schedule.

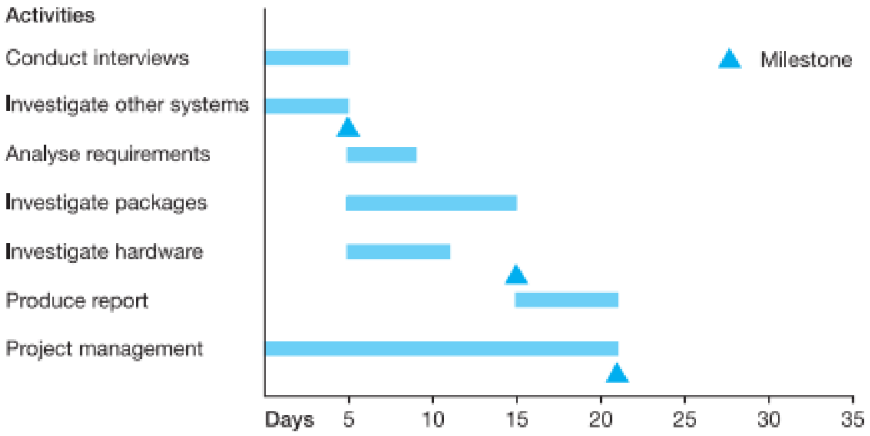


Figure 10.7 Bar chart showing project management as continuous activity over project

10.3 Developing resource plans

There are three types of resources:

1. Work (people)
2. Material (eg hardware)
3. Cost (services, travelling, training)

Figure 10.8 shows the bar chart for the feasibility study project and underneath it, a resource histogram which shows the deployment of project resources, in this case the analysts who will work on the study. It shows that for the first nine elapsed days of the project, we need three analysts. The requirement then drops down to two analysts for two days for two days and finally we need one analyst only for the last ten days. This gives us a total of 41 analyst-days:

Total number of analyst-days = (9*3) + (2*2) + (10*1) = 41

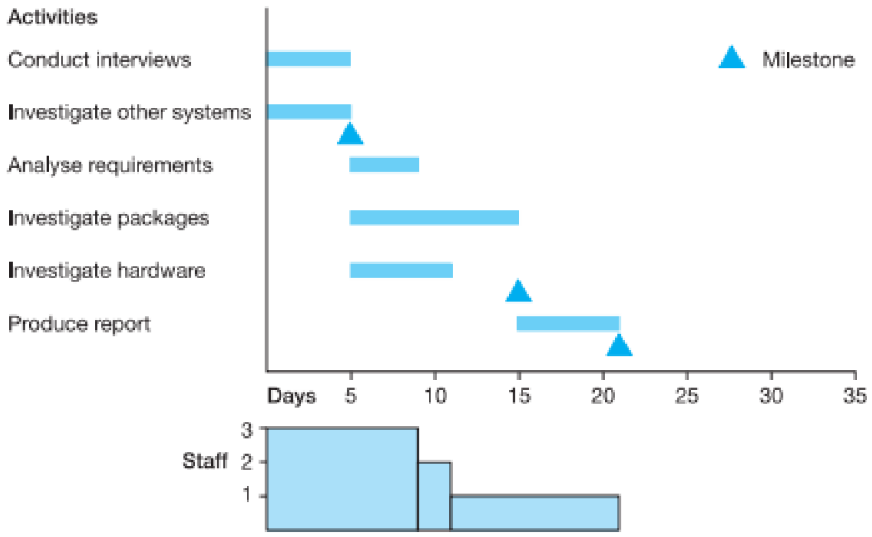


Figure 10.8 **Bar chart with resource histogram**

Budget is the sum of all resource costs

BUDGET FOR: NEW CUSTOMER CONTACT SYSTEM									
Expenditure code and heading		Monthly figures							Totals
		Mar	Apr	May	Jun	Jul	Aug	Sep	
A	Direct labour	50	50	70	90	120	70	30	480
B	Subcontract work		30	30	60	60	30		210
C	Hardware	100				200			300
D	Software	30				60			90
E	Telecommunications	10				60			70
F	Travel	3	3	1	1	3	2	1	14
G	Accommodation and subsistence	2	2	1	1	2	2	1	11
H	Project-specific training	10							10
I	Support services					2	6	5	13
J	Consultancy support	2	2	2	2	6	2	1	17
Contingency (10%) – items B–J only		*16*	*4*	*3*	*6*	*39*	*4*	*1*	*74*
Monthly totals:		207	87	104	154	513	112	38	1289

Figure 10.13 Example budget for an IT project

Contingency

To decide on the contingency percent for the project we must consider the following factors:

- ⇨ How tightly the requirements are defined
- ⇨ The confidence we have in our estimates
- ⇨ The degree of innovation in the project

Contingency is expressed in two ways:

1. Additional funds built into the project budget
2. Additional time built into the project schedule.

Documenting the plan

The plan must be documents to be read by the following:

- ⇨ Senior customer staff
- ⇨ IT management
- ⇨ Team members

What should be included in the plan document:

- ⇨ Introduction
- ⇨ Authorization and amendment record
- ⇨ Distribution
- ⇨ Related documents
- ⇨ Overview of the project
- ⇨ Products and deliverables
- ⇨ Milestones
- ⇨ Organization and responsibilities
- ⇨ Monitoring and control
- ⇨ Quality control
- ⇨ Reporting
- ⇨ Review and approvals
- ⇨ Risks to the project
- ⇨ Project schedule
- ⇨ Tasks description

Project Estimation

Instructor: Dr. Rana Yousef

Introduction

Project Estimating is a critical part of project planning, involving a quantitative estimate of project costs, resources or duration. Estimating IS projects is not glorious. Too many projects have gone badly over time and exceeded their budgets and the blame has often been put on the original estimates. Before we look at estimating methods, it might be as well to consider the special features of IS projects that make estimating for them so difficult

Why does IS projects difficult to estimate?

- IS projects are one-off projects (there is always something new)
- Requirements need not be specified at the start of the project
- it is seldom performed by professional estimators. Generally, estimates are prepared either by the project manager, by salespeople or by any staff who are spare at the time

Estimating Methods:

1. Analogy Method

The analogy method is one of the oldest, but one of the most reliable methods and depends on finding a project similar to the current one which has already been undertaken in the organization. The similarity should ideally extend to:

- The type of business involved and the general scope of the system
- The overall size of the applications.
- The technical methods, standards and languages used.

+ve: quickly

-ve:

If the new system is more complex => underestimation => possible loss.

If the older system is more complicated or wider in scope => overestimation => uncompetitive bid

2. Direct Estimation based on project breakdown

most detailed estimating technique and depends upon having a breakdown of the work to be performed (WBS or PBS)

once a detailed list of the project tasks is available, the estimator, or preferably several estimators so their results can be cross-checked, review the tasks and assess the effort to perform each. The effort required for each project stage and for the project as a whole is then arrived at through summation.

+ve: most reliable results
-ve: it is not always possible to use it

- Insufficient information from the beginning
- Takes great deal of time and effort

Generally used in developing plans for the immediate stage or sub-stage of a project

3. Delphi technique

Obtaining estimates from qualified people and then synthesizing them to produce the final estimate.

1. Each estimator is given a specification of the work, and is asked to provide their estimate for it.
2. Estimates are then summarized anonymously and the summary is circulated to each estimator
3. Estimators reconsider their own estimates in the light of the summary and provide a revised estimate if they wish.

+ve

The principle involved here is that, by keeping the estimates anonymous, personal disagreements are kept out of the process. In addition, the technique avoids a possible outcome of a round-table discussion which is that the person who shouts loudest, rather than the person with the best estimate, will win the day. Individual estimators can reconsider and revise their ideas in the light of other people's estimates without public loss of face.

4. CoCoMo

COCOMO (Constructive Cost Estimation Model) was proposed by Boehm [1981]. According to Boehm, software cost estimation should be done through three stages: Basic COCOMO, Intermediate COCOMO, and Complete COCOMO.

Basic COCOMO Model The basic COCOMO model gives an approximate estimate of the project parameters. The basic COCOMO estimation model is given by the following expressions:

Effort = $a1 \times (KLOC)^{a2}$ PM

Tdev = $b1 \times (Effort)^{b2}$ Months

Where

• KLOC is the estimated size of the software product expressed in Kilo Lines of Code,

• a 1, a2, b1, b2 are constants for each category of software products,

• Tdev is the estimated time to develop the software, expressed in months,

• Effort is the total effort required to develop the software product, expressed in person months (PMs).

The effort estimation is expressed in units of person-months (PM). It is the area under the person-month plot. It should be carefully noted that an effort of 100 PM does not imply that 100 persons should work for 1 month nor does it imply that 1 person should be employed for 100 months, but it denotes the area under the person-month curve

definition of organic, semidetached, and embedded systems are elaborated below.

Organic: A development project can be considered of organic type, if the project deals with developing a well understood application program, the size of the development team is reasonably small, and the team members are experienced in developing similar types of projects.

Semidetached: A development project can be considered of semidetached type, if the development consists of a mixture of experienced and inexperienced staff. Team members may have limited experience on related systems but may be unfamiliar with some aspects of the system being developed.

Embedded: A development project is considered to be of embedded type, if the software being developed is strongly coupled to complex hardware, or if the stringent regulations on the operational procedures exist.

Estimation of development effort

For the three classes of software products, the formulas for estimating the effort based on the code size are shown below:
Organic : Effort = $2.4(KLOC)^{1.05}$ PM Semi-detached : Effort = $3.0(KLOC)^{1.12}$ PM
Embedded : Effort = $3.6(KLOC)^{1.20}$ PM
Estimation of development time

For the three classes of software products, the formulas for estimating the development time based on the effort are given below:
Organic : Tdev = $2.5(Effort)^{0.38}$ Months Semi-detached : Tdev = $2.5(Effort)^{0.35}$ Months Embedded : Tdev = $2.5(Effort)^{0.32}$ Months

Example:

Assume that the size of an organic type software product has been estimated to be 32,000 lines of source code. Assume that the total salaries for programmers be 15,000 JD/month. Determine the effort required to develop the software product and the nominal development time.

From the basic COCOMO estimation formula for organic software: Effort = $2.4 \times (32)^{1.05}$ = 91 PM Nominal development time = $2.5 \times (91)^{0.38}$ = 14 months

Salaries cost required to develop the product = 14 x 15,000 = 210,000

Intermediate COCOMO

Cost drivers for **intermediate COCOMO** cost estimation method

Cost Drivers	Ratings					
	Very Low	Low	Nominal	High	Very High	Extra High
Product attributes						
Required software reliability	0.75	0.88	1.00	1.15	1.40	
Size of application database		0.94	1.00	1.08	1.16	
Complexity of the product	0.70	0.85	1.00	1.15	1.30	1.65
Hardware attributes						
Run-time performance constraints			1.00	1.11	1.30	1.66
Memory constraints			1.00	1.06	1.21	1.56
Volatility of the virtual machine environment		0.87	1.00	1.15	1.30	
Required turnabout time		0.87	1.00	1.07	1.15	
Personnel attributes						
Analyst capability	1.46	1.19	1.00	0.86	0.71	
Applications experience	1.29	1.13	1.00	0.91	0.82	
Software engineer capability	1.42	1.17	1.00	0.86	0.70	
Virtual machine experience	1.21	1.10	1.00	0.90		
Programming language experience	1.14	1.07	1.00	0.95		
Project attributes						
Application of software engineering methods	1.24	1.10	1.00	0.91	0.82	
Use of software tools	1.24	1.10	1.00	0.91	0.83	
Required development schedule	1.23	1.08	1.00	1.04	1.10	

The cost driver values should be added to the basic COCOMO equations to adjust the effort.

In the previous example if we state that size of application database is high and analyst capability is low, then the equation will be as follows:

Effort = 1.08 X 1.19 X 2.4 x $(KLOC)^{1.05}$
Effort =

5. COCOMO II

Depends mainly on function points to find the lines of code

(not only function point but object points, use case points)

How to calculate function points?

Compute the Unadjusted Function Point Count (UFC)

Function Count

Item	Item Description	Complexity	Count	Weight	Weighted Count
1	Number of User Inputs	Simple		3	
		Average		4	
		Complex		6	
2	Number of User Oututs	Simple		4	
		Average		5	
		Complex		7	
3	Number of User Inquiries	Simple		3	
		Average		4	
		Complex		6	
4	Number of Files	Simple		7	
		Average		10	
		Complex		15	
5	Number of External Interfaces	Simple		5	
		Average		7	
		Complex		10	
Total Weighted Function Count (FC)					0

Compute value adjustment factor (**VAF**) based on 14 general system characteristics (**GSC**).

General System Characteristic		Brief Description
GSC 1	Data communications	How many communication facilities are there to aid in the transfer or exchange of information with the application or system?
GSC 2	Distributed data processing	How are distributed data and processing functions handled?
GSC 3	Performance	Was response time or throughput required by the user?
GSC 4	Heavily used configuration	How heavily used is the current hardware platform where the application will be executed?
GSC 5	Transaction rate	How frequently are transactions executed daily, weekly, monthly, etc.?
GSC 6	On-Line data entry	What percentage of the information is entered On-Line?
GSC 7	End-user efficiency	Was the application designed for end-user efficiency?
GSC 8	On-Line update	How many ILF's are updated by On-Line transaction?
GSC 9	Complex processing	Does the application have extensive logical or mathematical processing?

GSC 10	Reusability	Was the application developed to meet one or many user's needs?
GSC 11	Installation ease	How difficult is conversion and installation?
GSC 12	Operational ease	How effective and/or automated are start-up, back-up, and recovery procedures?
GSC 13	Multiple sites	Was the application specifically designed, developed, and supported to be installed at multiple sites for multiple organizations?
GSC 14	Facilitate change	Was the application specifically designed, developed, and supported to facilitate change?

1. Weight each **GSC** on a scale of 0 to 5 based on whether it has no influence to strong influence.

1. Compute the **FPC** as follows.

   FPC = **UFC** * (0.65 + (sum(**GSC**) * .01))

Convert FPC to lines of source code (SLOC)

Language	SLOC per Function Point
1GL Default Language	320
2GL Default Language	107
3GL Default Language	80
4GL Default Language	20
Assembler	320
C	128
Basic	107
Cobol	107
C++	53
Java 2	46
Visual Basic 6	24
Delphi	18

HTML 4	14
SQL	13

6. Pert Estimating:

Most optimistic => 5 days

Most likely => 7 days

Most pessimistic => 12 days

(5 + 4*7 + 12)/ 6 = 7.5 days

9.4 Estimating supporting activities

Some activities may not be considered within the project's task and hence take more time that what estimated. For example: review tasks, rework after review, familiarization of business or techniques.
How to deal with them?

1. Either add a task (could be quality control)
2. Consider a percentage of the overall project duration (10% of all required time)
3. Determine a fixed time for these tasks (e.g one day per week)

Two types of supporting activities:

1. Proportional activities: team leading (supervision), program and operations documentation, quality control, quality assurance, customer review, reviewing third party work, post implementation review.
2. Explicit activities: user documentation, staff technical training, familiarization, data creation, conversion and migration.

9.6 Practical experiences with estimating

1. Building up metrics: it is the best way to take the uncertainty out of estimating, however, it takes time and effort, and may not necessarily be applicable for all kinds of projects.
2. Using standard project structures: follow a development process: requirements, design and implementation, testing and validation
3. Getting more than one view: for example Delphi method (more one person to estimate the effort and time is better than only one person)
4. Qualifying estimates: state all assumption used in estimation
5. Documenting the estimates: to compare results of different estimating processes, to find out any errors (late tasks), to fine-tune estimating process for future projects.
6. Estimating and risk analysis.

Managing Risks

Instructor: Dr. Rana Yousef

Risk Management Process:

Figure 15.1 The risk management process

Risk identification:

It is more easily said than done, because:

- Each project is a unique project
- Many areas in which risk could arise and it is difficult to locate them all.

So it is valuable to get a second or third opinion from experienced project managers.

After identifying the risk we must describe them so that it is clear exactly what each risk is about and how we can find the suitable solution. For example: “Poor staff performance” risk, must be described as either:

- Staff don’t work at the pace assumed in preparing the estimates
- Staff don’t grasp and conform to the developer’s programming standards
- Staff are difficult to manage with inexperienced team leaders

One way of identifying risks or prompting questions that will help to identify risks is to use a form of risk breakdown structure to uncover more details about risks.

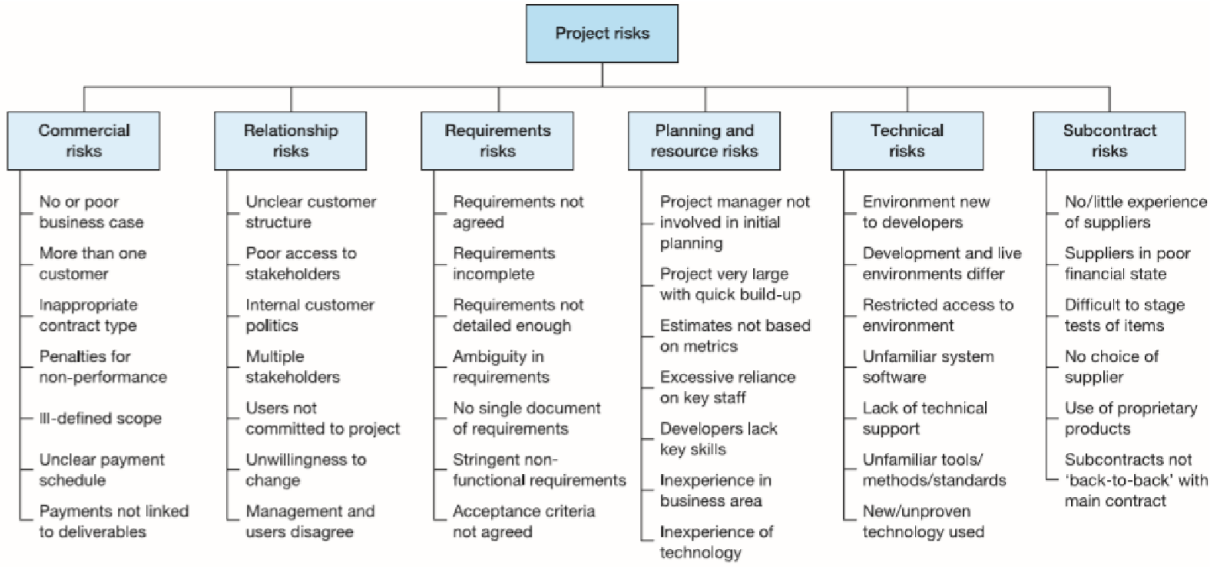


Figure 15.2 Risk breakdown structure

Risk assessment:

We assess the risk for their impact (seriousness), likelihood (probability to happen)

Assessment of impact could be related to the time/cost/quality criteria as:

- Large impact: could extend project by more than 10%
- Moderate impact: could extend project by 5-10%
- Small impact: could extend project by less than 5%

The Likelihood could be given a rough numerical value like:

- High probability: greater than 30%
- Medium probability: 10-30%
- Low probability: less than 10%

Now we can compare the risks to decide which ones need the closest management attention. The most important risks are those with a large impact and a high probability of occurrence. A useful way of highlighting the important risks is to use a risk map.

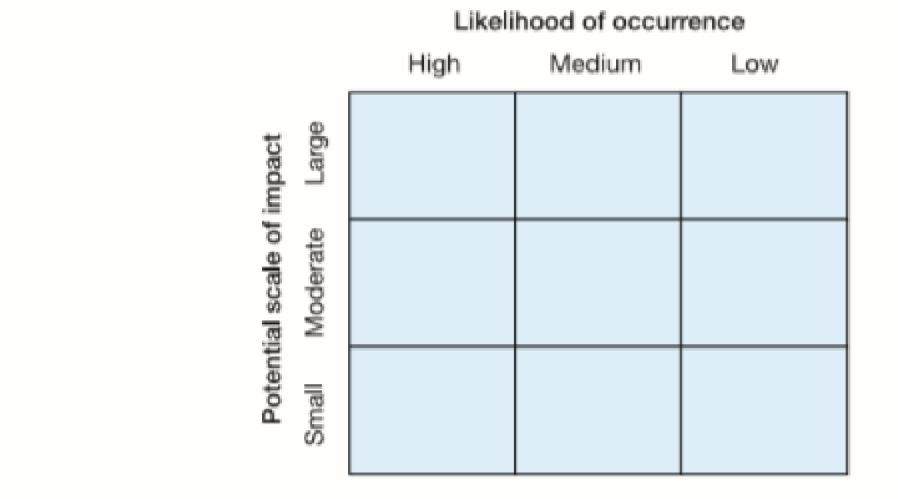


Figure 15.3 Risk map

Risk actions:

So far, we have identified the risks and quantified their effects. However, this is rather useless unless some actions are taken to deal with the risks. In essence, there are four main responses to risk:

- **Acceptance.** It may be that there are no feasible countermeasures, or that these are more expensive than suffering risk to occur. In that case, the only rational response may be to let the risk happen but also, perhaps, to build in time or budget contingency to deal with the effects of the risk if it occurs.
- **Avoidance.** This involves the things we can do to try to prevent the risks from occurring (in other words, dealing with the likelihood).
- **Mitigation.** This includes the steps we can take to reduce the impact of the risks if they occur (in other words, dealing with the impact).
- **Transfer.** This involves making the impact of the risk, if it occurs, fall on someone else. Taking out household insurance, for example, does not reduce the likelihood of your house being burgled but, if it is, the impact is felt by the insurance company.

Example:

Mitigation actions for “staff are not available” risk could be:

- Ensuring that we use only staff of whom we have experience
- Setting prospective staff a short test to assess their speed of work
- Conducting searching interviews to discover their attitude to standards

Risk management planning and control:

Risk management is an ongoing process. There needs to be a procedure to revisit the risk register regularly and to reassess the status of each risk. The review of risks in undertaken at regular progress meetings.

The risk management plan should set out:

- A statement of the *scope and intensity* of the risk management to be applied to the project. Risk management, like other project management tasks, must be tailored to the size, value and complexity of the individual project.
- An explanation of the *risk management cycle* to be used on the project, showing how and when risk reviews will be carried out and whether they will be a separate process or part of the ongoing project monitoring work.
- *Roles and responsibilities* – who will be in charge of the risk management process and the mechanism by which risks will be reviewed and controlled.
- A description of the *products* of risk management – for example, a regular risk assessment report prepared for senior management.

The risk register

It is an important document that could be word processor file, spreadsheet or database, contains:

- A *reference* – each risk needs a unique identifier, perhaps keyed to the phase, task or product on which it impacts.
- A *title and description* of the risk.
- The *current status* of the risk – for example, candidate (identified but not yet quantified), live, or closed.
- *Potential impacts* – there may be more than one of these and, for each, you need to record a description and assessment of its likelihood and scale of impact.
- *Risk owner* – the person who will be responsible for carrying out the identified risk actions (see below).

- *Actions* – the avoidance, mitigation and transfer actions that have been identified.
- *Action log* – a record of the progress made in discharging the risk actions.

Risk ownership

Part of the process of risk identification is to decide who should be the owner of each risk. The owner should be someone who:

- Has sufficient information concerning the risk
- Has the necessary resources
- Possesses the authority to do something about the risk