Unit III

3 Activity Planning and Risk Management

Syllabus

Objectives of Activity planning - Project schedules - Activities - Sequencing and Scheduling, Network Planning Models - Formulating Network Model - Forward Pass and Backward Pass Techniques.

Risk Management - Introduction, Risk Management, Risk Assessment, Risk identification, Risk Prioritization, Risk Planning, Risk control, Risk Strategies, Evaluating Risk to the schedule.

Study Risk Management Tools - SpiraPlan by Inflectra, Risk Management Studio, GRC Cloud.

Case study : Online Shopping System.

Contents

3.1 Introduction to Activity Planning
3.2 Project Schedules
3.3 Project Activities
3.4 Project Sequencing and Scheduling Activities
3.5 Network Planning Models
3.6 Formulating Network Model
3.7 Forward and Backward Pass Techniques
3.8 Risk Management
3.9 Risk Assessment
3.10 Risk Identification
3.11 Risk Prioritization
3.12 Risk Planning
3.13 Risk Control
3.14 Risk Strategies
3.15 Evaluating Risks to the Schedule
3.16 Study Risk Management Tools
3.17 Case Study : Online Shopping System

3.1 Introduction to Activity Planning

Introduction :

- However, a schedule outlining the beginning and end times of each activity is also required as part of a thorough project plan. With this, we will be able to :
  - Guarantee that the necessary resources will be accessible exactly when needed;
  - Avoid from conducting many activities that compete for the same resources at the same time;
  - Create a detailed timetable indicating which employees handle each task;
  - Create a detailed plan that may be used to gauge real success;
  - Create a forecast for timed cash flow;
  - Re-plan the project to address drift from the aim during its life.
- A plan must be expressed as a collection of objectives whose accomplishment or failure can be clearly measured in order to be effective. This is accomplished via the activity plan's provision of target start and finish dates for each activity (or a window within which each activity may be carried out).
- One of the reasons it is advisable to make sure that every project activity results in some sort of tangible output or "deliverable" is because the starts and completions of activities must be readily apparent. So, at least in part, monitoring the project's progress involves making sure that each activity's product is provided on schedule.
- It is improbable that everything will proceed as planned as a project develops. Recognizing errors, determining their causes and changing the plan to mitigate their impacts make up a large portion of project management tasks. The activity plan should include a way to assess the effects of missing any of the activity goal dates and instructions on how to change the plan most efficiently to get the project back on track.

3.1.1 Objectives of Activity Planning

- Activity planning has a variety of other objectives in mind in addition to supplying project and resource schedules, which can be summed up as follows.
- **Feasibility assessment :** A feasibility study is only an assessment of the feasibility of a proposed plan or project. Pay attention to possible problems during project implementation.

In order to determine whether the project is feasible after considering all important factors, that is, whether it is worth solving.

- **Resource allocation :** Resource allocation is the process of allocating and scheduling available resources as efficiently and economically as possible. Projects always need resources, but these resources are usually scarce. Therefore, it is the responsibility of the project manager to determine the appropriate time and allocate resources. These resources are within the project schedule. So what is the resource allocation in project management ? It is about managing and delegating resources throughout the project to ensure that it runs as evenly and successfully as possible.
- **Detailed costing :** After generating an activity plan and allocating particular resources, we will acquire extra complete estimates of prices and their timing.
- **Motivation :** Motivation to set goals and track progress against goals is an effective way to motivate employees, especially if they are initially involved in setting these goals.
- **Coordination :** The project plan offers a useful service means of communication and collaboration between teams, especially for large projects with multiple project teams. Can be used as needed, will not be affected by forced shutdown.
- The goal of activity planning and scheduling strategies is to complete the project in the shortest possible time at the lowest possible value or to fulfill an arbitrary deadline at the lowest possible value.
- A successful method to shorten the project duration is to hold activities in parallel. Obviously, we cannot complete all activities at the same time, some of which require completion of other activities to start and there may be resource constraints that limit the scope of work. However, activities planning will enable us to understand the cost of these restrictions in terms of extending the time frame and will let us understand how to reduce the time frame by relaxing these restrictions. For example, priority restrictions make it possible to start coding the program before the design is complete to ensure that we have a clear understanding of the potential impact on product quality.

3.1.2 Project Plan

- Planning is a process of continuous improvement and each iteration is more detailed and precise than the previous one. Iterations after iterations, focus and the plan's objective will change.

- Throughout the feasibility study and start-up of project period, the main objective of the plan is to assess the time and risk of non-compliance with the deadline for completion or exit within the budget. Develop an action plan to ensure the availability of resources and control of cash flow.
- From the entire project to the delivery of the final product to the customer, monitoring and re-planning should continue to correct any deviations that may hinder the achievement of time or cost goals.

Review Questions

1. What are the objectives of activity planning ? Explain in detail
2. Explain the importance of project plan.

3.2 Project Schedules

- In the case of large projects, detailed planning for subsequent stages will be postponed till the details of the necessary task is available at an earlier point in time.
- Before starting work on a project or perhaps in a bigger project phase, a project plan should be created to show the date when each activity should start and end and the amount of each resource. When a plan is improved to a level of detail it is called a project schedule.
- There are four phases to developing a project schedule.

1. **The initial stage** in creating a plan is deciding which activities must be completed and in what sequence they must be completed. On this basis, we can create an ideal activity plan, that is, if resources are not limited, ideally a time plan for each action can be taken. This is about making the perfect plan of activity. The activity plan is gradually created in steps 4 and 5, as shown in the Fig. 3.2.1.
2. The **second phase**, ideal activity plan then becomes the subject of activity risk analysis, the purpose of which is to identify potential problems that can come up with an activity plan that will almost certainly affect the allocation of resources.
3. The **third phase** is to allocate resources. Expected resources availability may limit how some activities can be taken and our ideal plan may need to take this into account.
4. The **last step** is scheduling. Once resources are allocated for each activity, we can create and produce a project schedule, which contains the start and end dates of the plan and an overview of the resource requirements for each activity.

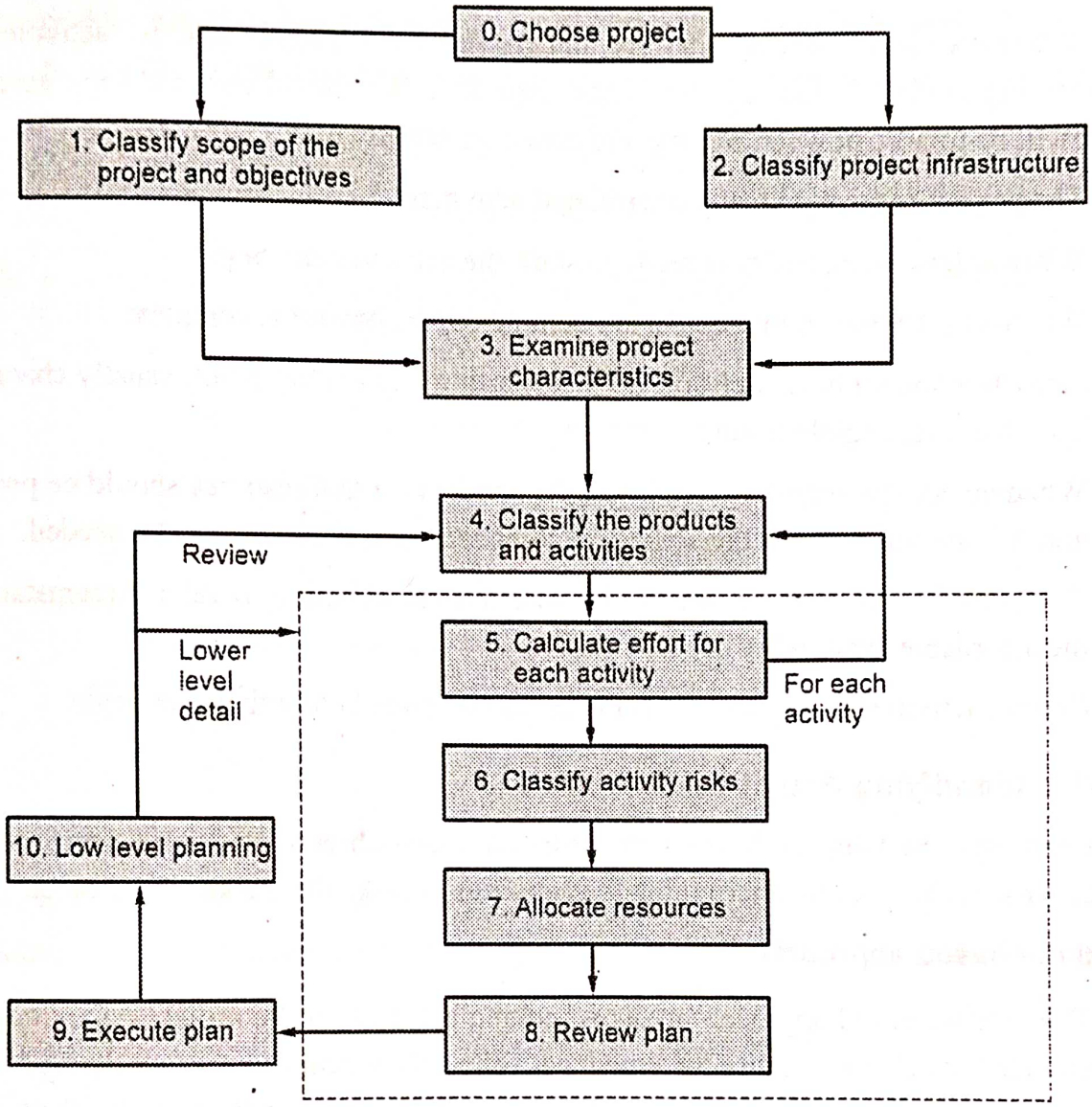


Fig. 3.2.1 : The activities are planned for step 4 and step 5

Review Question

1. Explain how project schedule is done in step wise.

3.3 Project Activities

- Project activities are actions taken by the project to achieve the set goals. These are very specific, practical and well-defined measures. By reading the description of the project activity, we can immediately describe the project in our mind effortlessly.
- Activities need to be defined to fulfill these standards. Any activity that does not fulfill these standards should be reviewed.

- It is worthwhile to assess what we understand by a project and its activities before attempting to identify the activities that comprise it. We should also add some assumptions that will be important when we begin to create an activity plan.
  - The project consists of many interrelated activities.
  - When at least one activity is ready to start, the activities can begin.
  - When all activities in the project are completed, the project is complete.
  - Activities should have a clearly defined starting and ending point, usually characterized by achieving tangible results.
  - When an activity requires a resource, the demand for that resource should be predictable and it is assumed that as the activity decomposes, it will continue to be needed.
  - The period of an pastime need to be forecastable assuming regular circumstances and the affordable availability of resources.
  - Certain activities may require other users to complete before they can begin.

3.3.1 Identifying Activities

- The activity-based approach, the product-based approach and the hybrid approach are the three basic techniques to determining the activities or tasks that make up a project.

1. Activity based approach

- The activity-based approach involves making a list of all the project's activities should include, which can be brainstormed with the entire project team or derived from an analysis of previous projects. For large projects, it may be helpful to break the project down into lifestyle milestones and view each milestone separately.
- The preferred method of creating a task list is to create a Work Breakdown Structure (WBS) instead of performing this operation all at once and there is an obvious risk of tasks being missed or calculated twice. This includes defining the main goal (or high level). It is necessary to finish the project first and then divide it down into sub-tasks. Fig. 3.3.1 shows an excerpt from the PSP, where the design task is split into three tasks and one of the tasks is split into two tasks.
- Activities should only be added to a branch in a structure if they directly benefit the job directly above it; otherwise, they should not be added to that branch. Any branch's duties should include everything needed to finish the task at the higher level at each level.
- It is important to take the final level of detail or depth of the structure into account while creating a WBS. A structure that is too deep will produce a lot of little, difficult-to-manage jobs, but one that is too shallow won't give us enough information to manage

our project. However, each branch should be divided at least sufficiently to allow each leaf to be allocated to a specific person or accountable division within the organization.

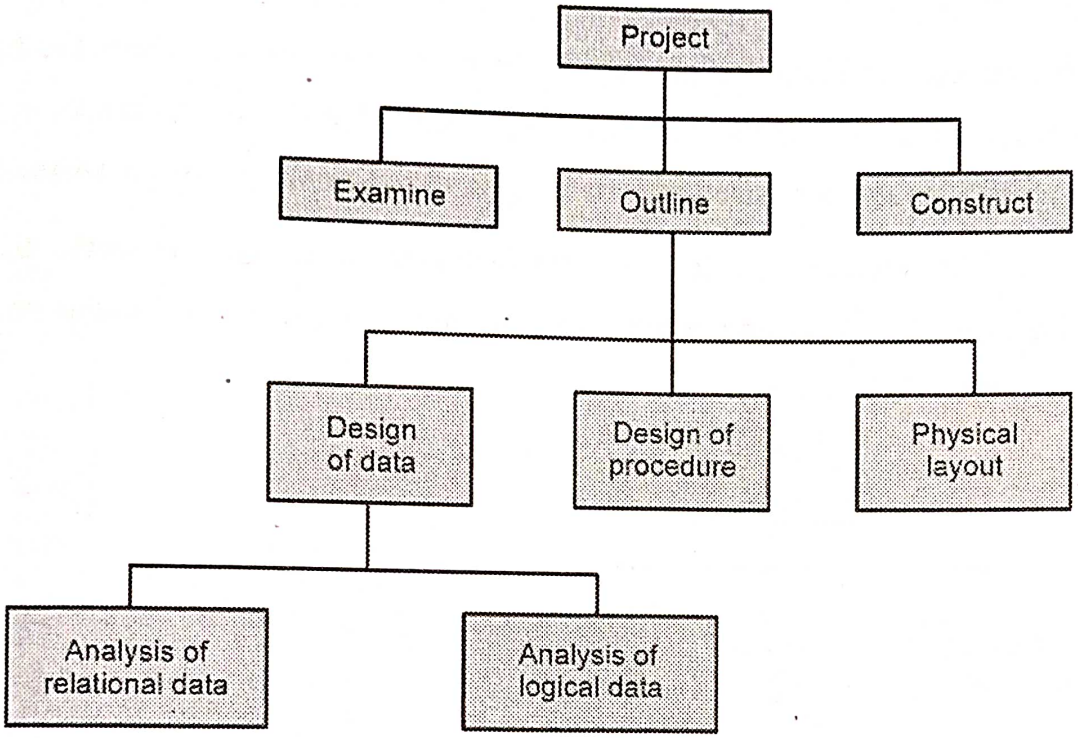


Fig. 3.3.1 : A fragment of an activity-based work breakdown structure

- **Advantage of WBS :** The WBS approach is said to have the advantage of being far more likely to provide a task catalogue that is comprehensive and made up of non-overlapping tasks. Keep in mind that the list of project activities is only represented by the structure's leaves; higher-level nodes are only aggregates of project activities.
- The WBS also indicates a framework that could be improved as the project moves forward. We may utilise a somewhat shallow or high-level WBS in the early stages of a project. This WBS can be updated as new information becomes available, usually throughout the project's analysis and specification phases.
- It is necessary to sequence the project's activities after they have been identified (whether or not a WBS was used for this purpose). This means determining which activities must be finished before others can begin.

2. Product based approach

- The product-based approach, used in PRINCE2. It involves creating a Product Flow Diagram(PFD) and a Product Breakdown Structure (PBS). The PFD details which additional items are needed as inputs for each product. By identifying the processes that change some products into others, it is simple to convert the PFD into an ordered list of activities. According to proponents of this strategy, it is less probable that a product will be missing from a PBS than that an activity may be left off of an unstructured activity list.

- This approach is especially suitable if using a methodology like SSADM or USDP (Unified Software Development Process), which explicitly lists each product required and the actions necessary to produce it for each phase or job. From the Fig. 3.3.2, for each stage of SSADM, the SSADM Reference Manual, for instance, offers a set of generic PBSs that can be used as a starting point for creating a project-specific PBS.
- The SSADM reference guide also provides generic activity networks that is a good starting point for designing project-specific action networks by using project-specific PBS and PFD derivatives.

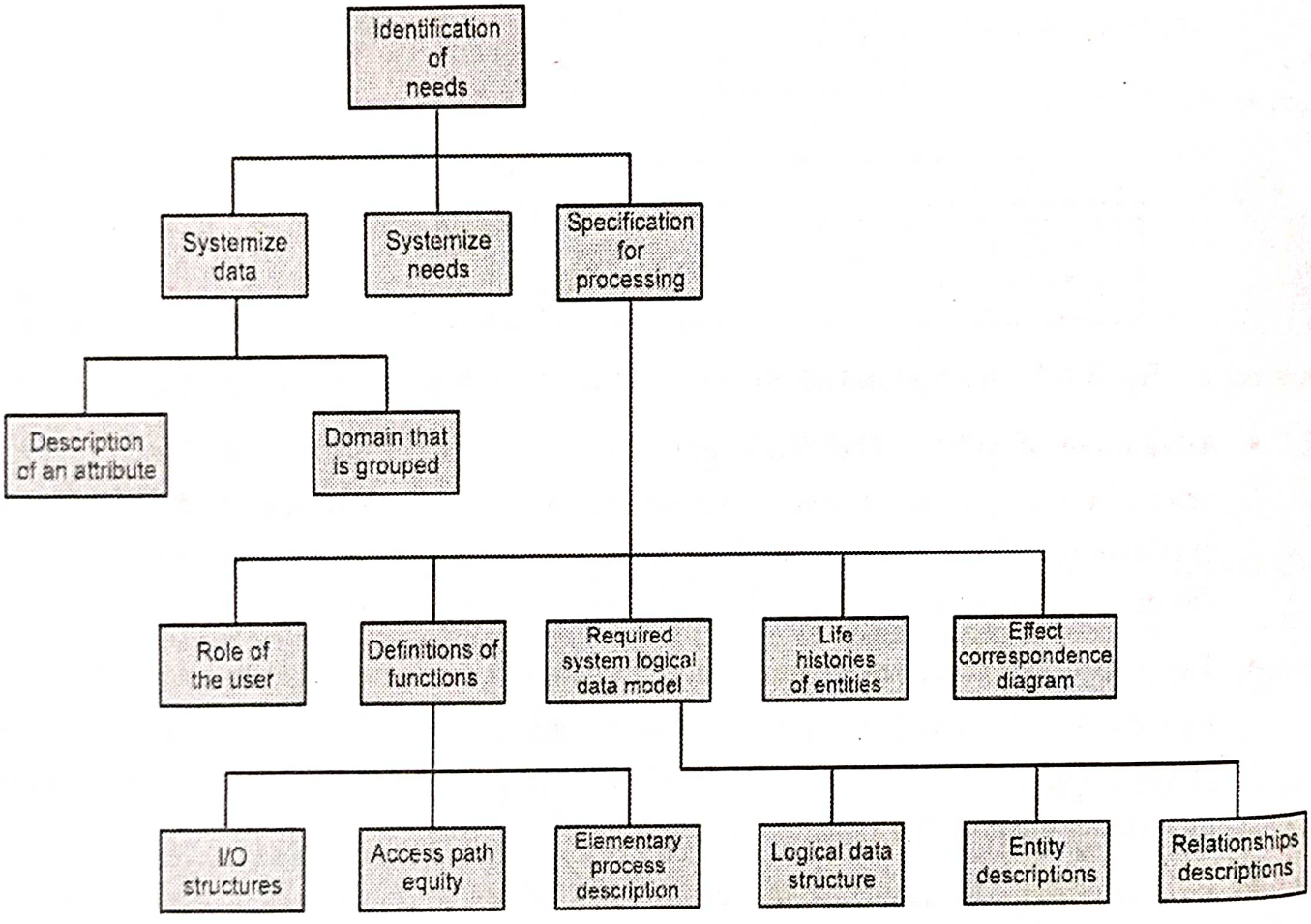


Fig. 3.3.2 : SSADM product breakdown structure

- Products are referred to in the USDP as artifacts (see Fig. 3.3.3) and the series of tasks required to create them is known as a workflow (see Fig. 3.3.4 for an example). Drawing an activity network from these operations requires some care. The iterative nature of processes is emphasised by USDP. As a result, it might not be possible to directly map a USDP process onto a single network action.

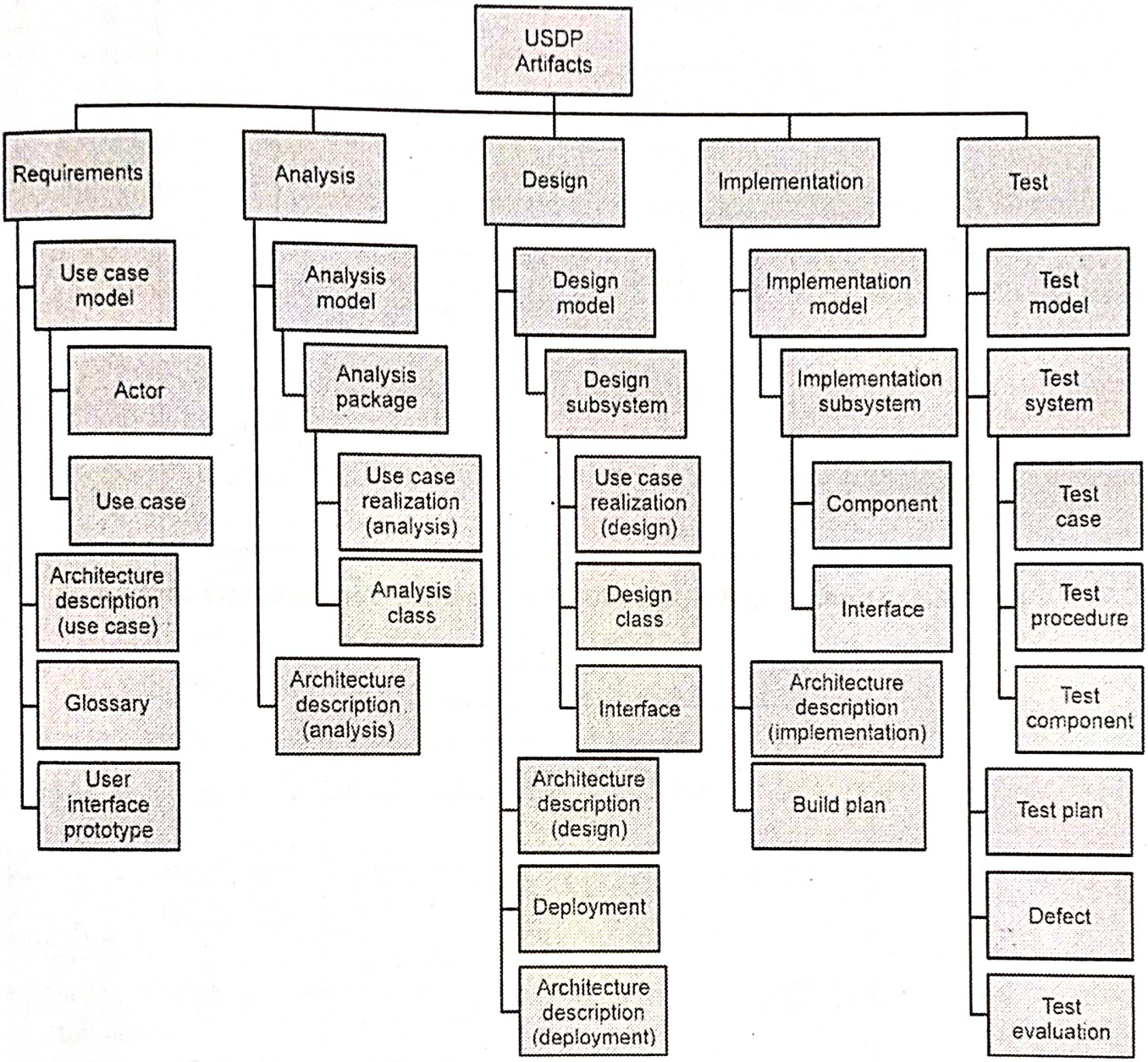


Fig. 3.3.3 : Structure of USDP products depending on artefacts

3. Hybrid based approach

- The WBS shown in Fig. 3.3.1 is based largely on structured activities. Or, possibly more frequently, the WBS can be based on the results of the project, as shown in Fig. 3.3.5 and the results of the project are based on a simple list of results. For each result, a series of activities are required to achieve the goal. Fig. 3.3.5 shows a flat WBS and it is likely that the introduction of additional layers of structured products and activities is useful in projects of any size. When the structure is related to the product or the activity may depend on the type of project and the specific development method used. In the same way that pure action-based PSPs, after recognizing actions, our task is to organize them.

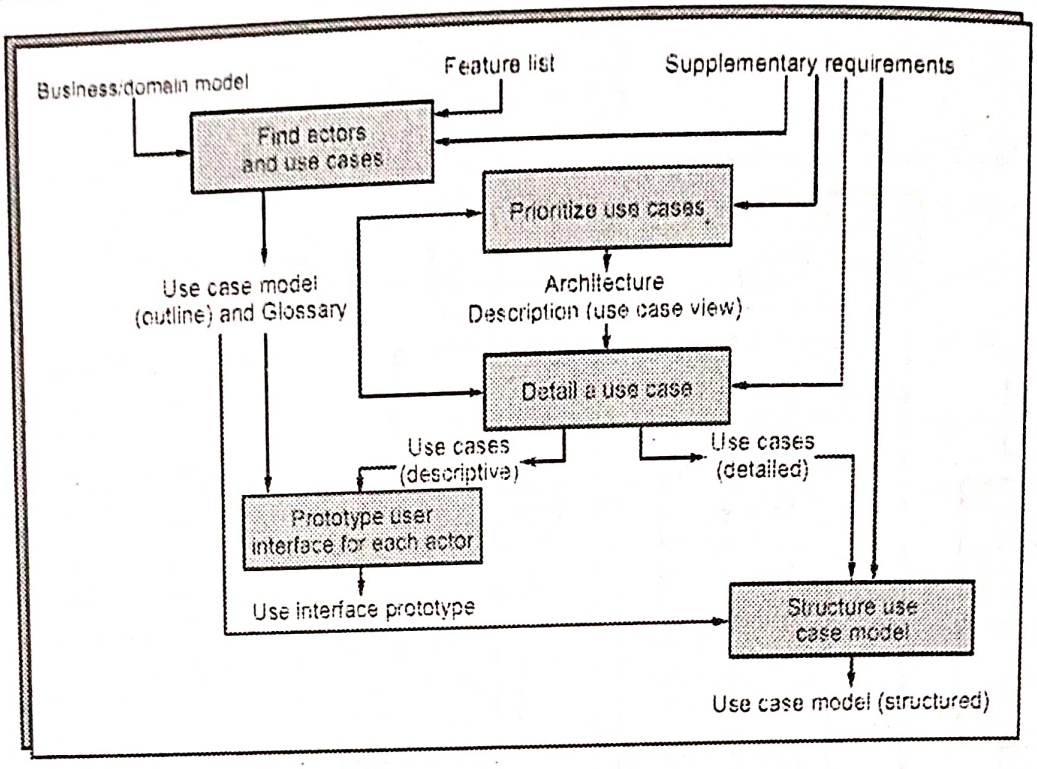


Fig. 3.3.4 : Organizing the procedure for the USDP requirements capture

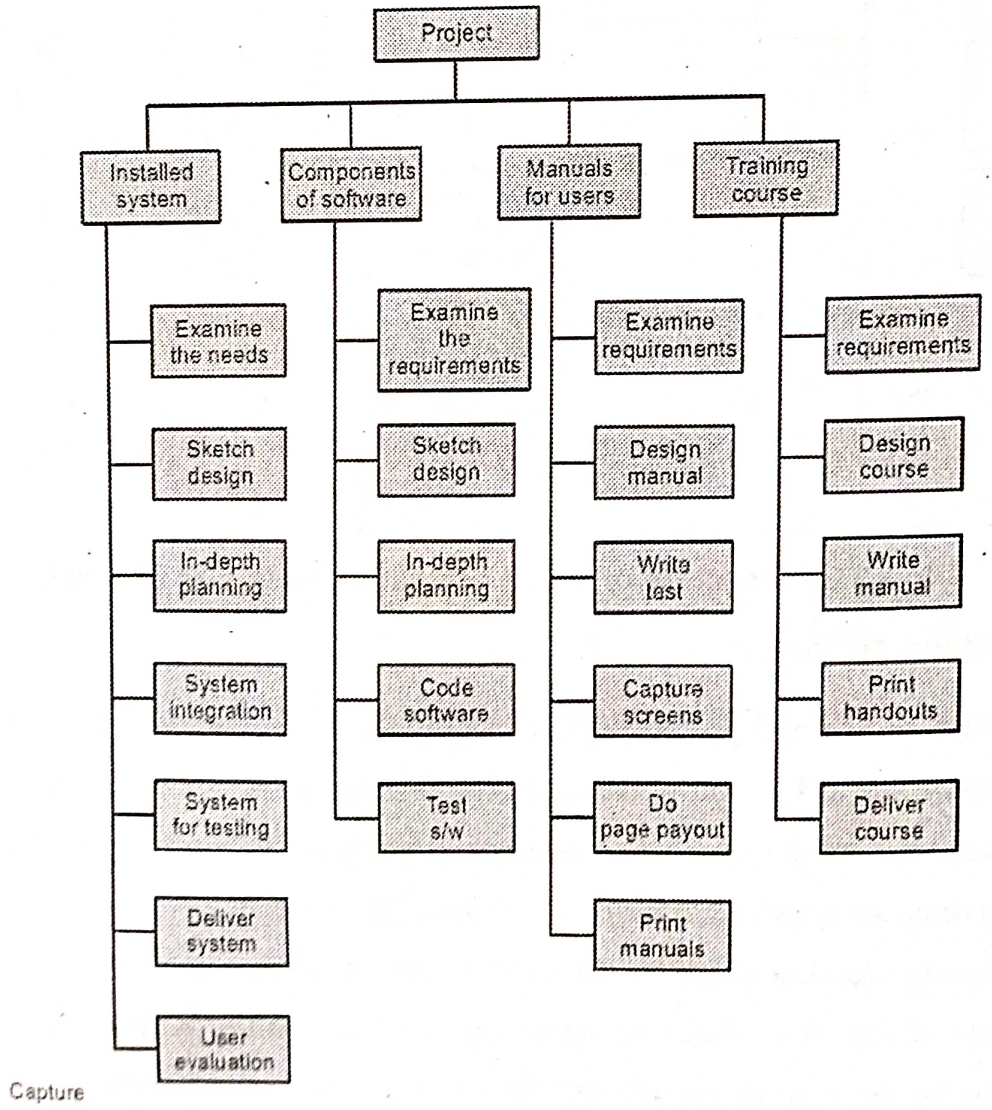


Fig. 3.3.5 : Hybrid work breakdown based on deliverables and activities

- The structure defines the number of layers in the structure and the type of each layer, which can be superimposed on the PSP. For example, IBM recommends using the following five levels in WBS in its MITP method :
  - Level 1 : Project
  - Level 2 : Deliverables at the second level, including software, manuals and training courses.
  - Level 3 : Components are the primary work items required to produce deliverables at Level 3, such as the modules and tests necessary to produce the system software.
  - Level 4 : Work-packages, which are substantial pieces of work or groups of linked tasks needed to create a component
  - Level 5 : Tasks of Level 5 are those that are typically the responsibility of a single person.

Review Question

1. Write a short note on activity.
2. Explain different approaches for identifying the activities that make up a project.
3. Define WBS. What are the advantages of WBS ?
4. Write a note on : Workflows and PFD.
5. Explain 5 levels of WBS.

3.4 Project Sequencing and Scheduling Activities

- During the course of the project, we need a schedule that clearly shows the planned time of each project activity and what resources we need. One way to express this type of plan is to use bar charts, as shown in Fig. 3.4.1.

Activity key :

- A : Overall design
- B : Specify module 1
- C : Specify module 2
- D : Specify module 3
- E : Code module 1
- F : Code module 3
- G : Code module 2
- H : Integration testing
- I : System testing

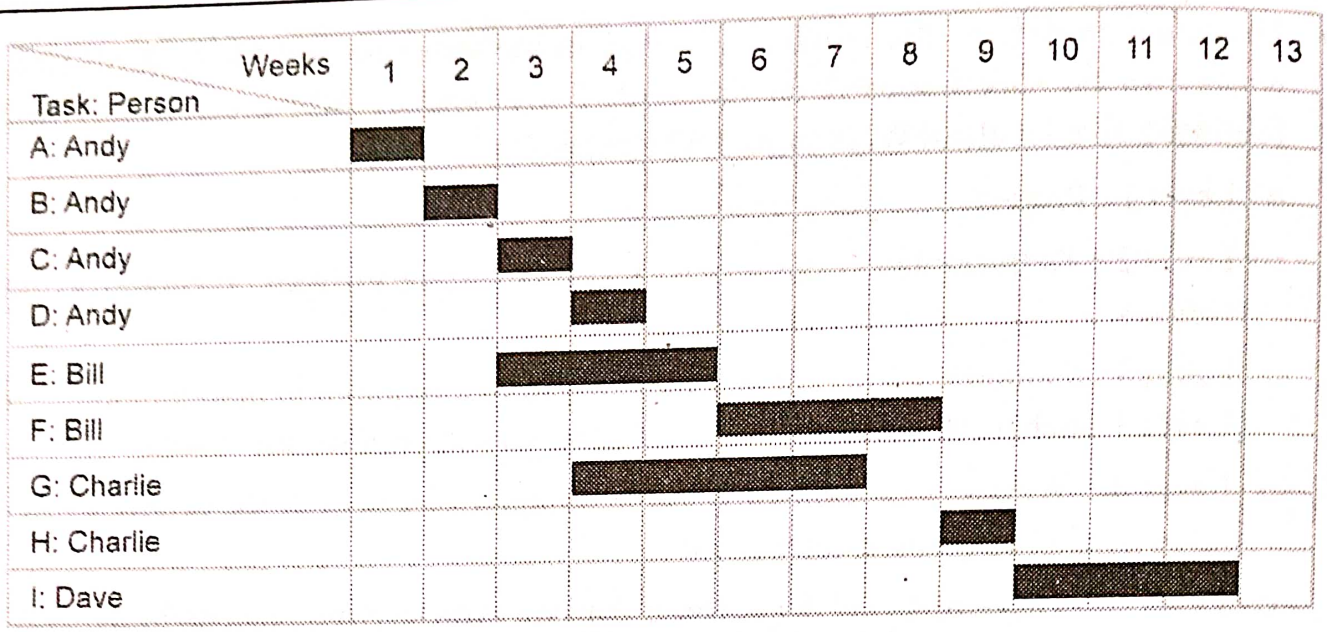


Fig. 3.4.1 : A project plan as a bar chart

- The chart was created while taking into account the resources that are available and the nature of the development process, which requires that some activities be finished before others can begin (for example, activity C follows activity B because Andy cannot work on both tasks at the same time). As a result, we have sequenced the tasks (that is, identified the dependencies among activities mandated by the development process) and scheduled them in the chart (that is, specified when they should take place). The activities' distribution among the employees and their availability had to be considered when creating the schedule.
- The bar chart does not explain the rationale behind specific choices. For instance, it is unclear why activity H won't begin until week 9. It might be because Charlie will be on vacation in week 8 or because it cannot begin until activity F has been finished.
- This combined sequencing-scheduling strategy may work well for small projects, especially if we want to assign specific tasks to specific people at the beginning of the planning process. On larger projects, it is preferable to divide these two activities: order the tasks in accordance with their logical relationships, followed by scheduling them while taking resources and other aspects into account.
- Network modelling is a scheduling technique that separates the logical and physical components of a project.

3.5 Network Planning Models

- Project activities and their interactions are modeled as networks in these project scheduling systems. Time flows from left to right in the network.

- CPM (critical path method) and PERT (project evaluation and reporting technique) are two of the most well-known of these methods, which were created in the 1950s (program evaluation analysis method).
- When visualizing the project as a network with activities represented as arrows joining circles or nodes that could represent the beginning or end of an activity or series of activities, both of these strategies used an activity-on-arrow approach. Precedence networks, a variant of these methods, have gained popularity more lately. In this method, activities are represented as nodes in activity-on-node networks and the links between the nodes stand in for precedence (or sequencing) requirements. The latter method offers more flexibility for representing specific situations simply while avoiding some of the drawbacks of the activity-on-arrow representation. The vast majority of currently accessible computer applications use this technique.
- These three approaches are highly similar and it must be acknowledged that many people refer to any or all of the approaches by the same name (especially CPM).

3.6 Formulating Network Model

- Representing the activities and their Interrelationships as a graph is the first step in building a network model. To do this with activity-on-node, we represent activities as nodes (boxes) in the network, with the connections between nodes signifying dependencies.

A. Constructing precedence networks

- It is worthwhile to spend some time thinking about certain guidelines for networks' construction before we examine how they are used.

1. **A project network should have only one start node :** Although drawing a network with more than one initial node is logically conceivable, doing so is not recommended because it could lead to confusion. In certain circumstances (for instance, when multiple activities may begin as soon as the project begins), it is typical to create a "start" activity that has no duration but might have a real start date.

2. **A project network should have only one end node :** A project can only be finished once and the end node marks its conclusion. It is feasible to depict a network with more than one end node, however doing so will almost likely cause confusion. It is typical to create a "finish" activity when more than one "final" action is required for a project to be completed.

3. **A node has duration :** An activity is represented by a node and activities typically take time to complete. But take note that there is no mention of durations in the network in Fig. 3.6.1. This network diagram just illustrates the project's logic the principles dictating the sequence in which tasks are to be completed.

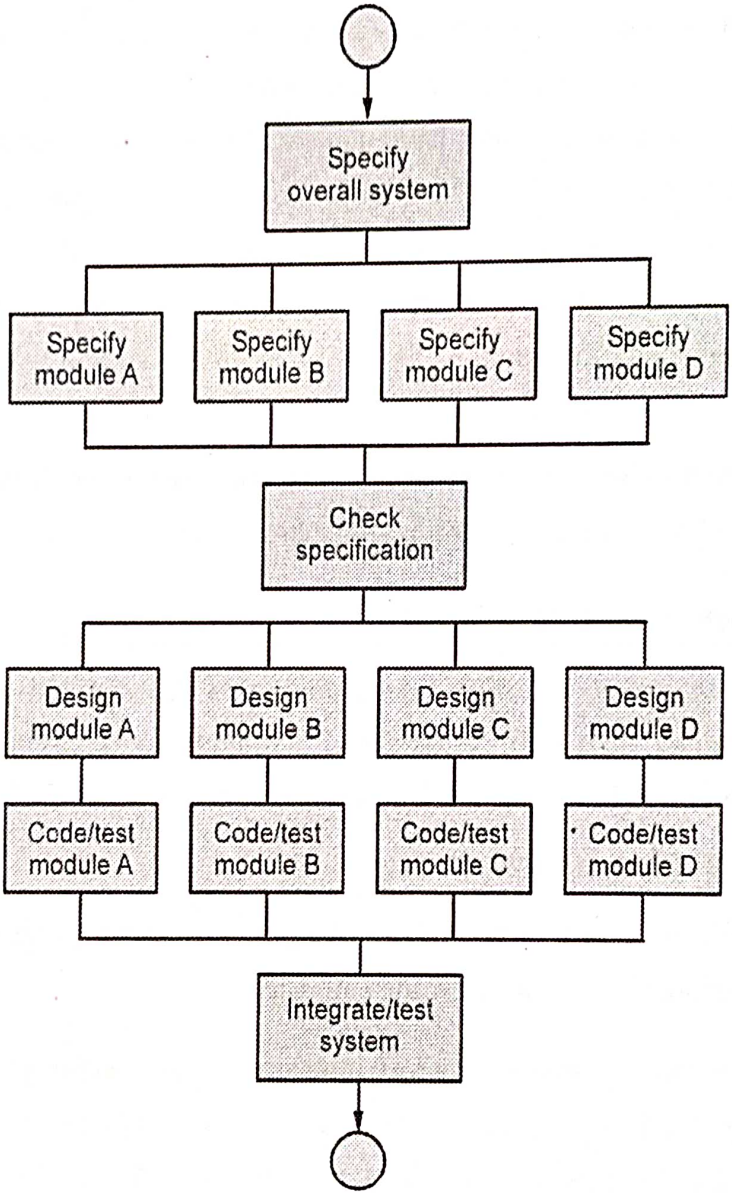


Fig. 3.6.1 : The project activity network for the IOE annual maintenance contracts has a checkpoint activity added to it

Fig. 3.6.2 shows how this network looks like a network with a critical path.

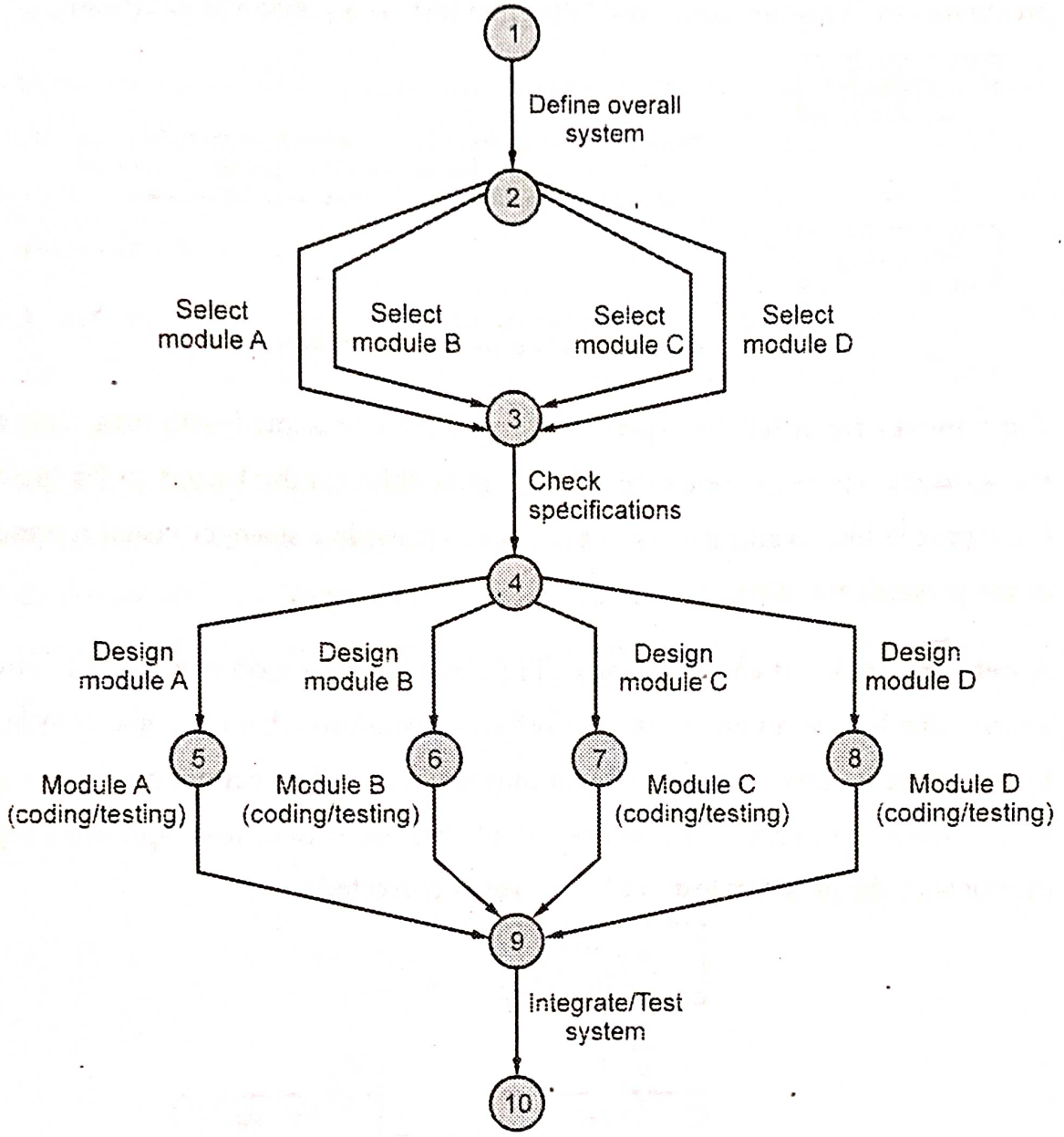


Fig. 3.6.2 : The IOE maintenance group accounts project activity network fragment represented as a CPM network

4. **Links normally have no duration :** The relationships between activities are represented by links. Installation in Fig. 3.6.3 cannot begin until program testing is finished. Coding and data take-on must both be finished before program testing can begin.

5. **Precedents are the immediate preceding activities :** According to Fig. 3.6.3, the activity "Program test" cannot begin until the activities "Code" and "Data take-on" have been done and the activity "Instal" cannot begin until the activity "Program test"

has been completed. Therefore, it can be claimed that "Code" and "Data take-on" are precursors of "Program test," and "Program test" is a precedent of "Install.

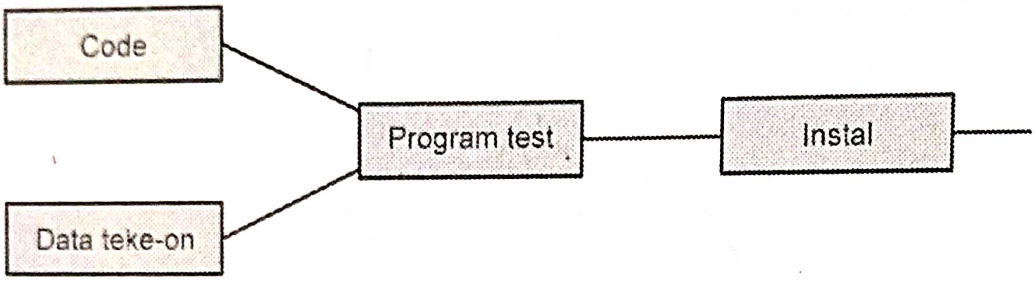


Fig. 3.6.3 : A precedence network's fragment

6. **Time moves from left to right :** Networks are represented with time flowing from left to right whenever practical. Rarely does this standard need to be broken, but some people like to add arrows to the lines to provide a stronger visual representation of the project's timeline.

7. **A network may not contain loops :** Fig. 3.6.4 shows a loop in the CPM network. In a sense, the loop is an error; since it reflects a condition that does not exist in reality. Loops in the iterative sense can occur in practice, but they cannot be directly mapped to the project network. It should be noted that the logical assumption in Fig. 3.6.4 cannot start the program test until the error is corrected.

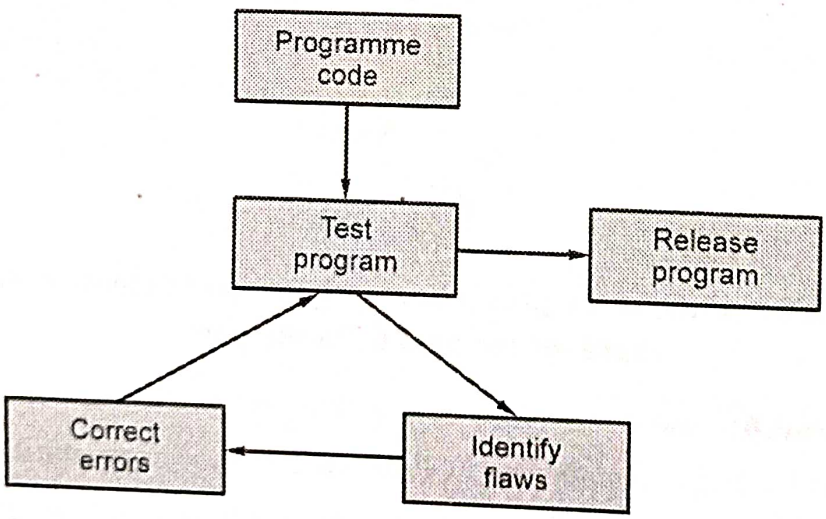


Fig. 3.6.4 : A loop represents on impossible sequence

We can draw a set of activities as a straight sequence and repeat it the right number of times if we know how often we plan to repeat a set of activities, such as a test-diagnose-correct sequence. If we don't know how often a sequence will occur, we can't estimate how long the project will take without using a different approach, such

reframing the entire sequence as a single activity and predicting how long it would take to finish.

Even though the loop in this tiny network fragment is obvious, very large networks can easily have intricate loops that are challenging to identify when they are first built. Thankfully, every network design software will find loops and send out error alerts when they do.

8. **A network should not contain dangles :** A dangling activity like "Write user manual" in Fig. 3.6.5, shouldn't exist since it could result in mistakes in further analysis. In fact, when activities are created as an afterthought, they frequently show logical problems. Fig. 3.6.5, should be revised to include a final completion activity if, as intended, the project is considered complete once the software has been deployed and the user manual has been prepared. This would be a more true depiction of what should occur in this scenario. In Fig. 3.6.6, the redrawn network is displayed.

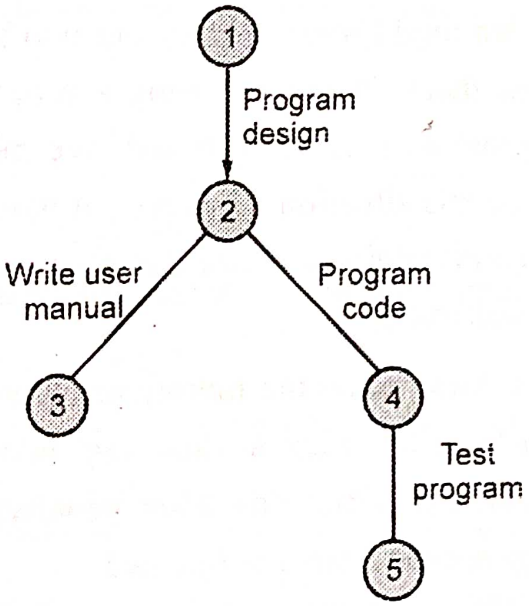


Fig. 3.6.5 : A dangle

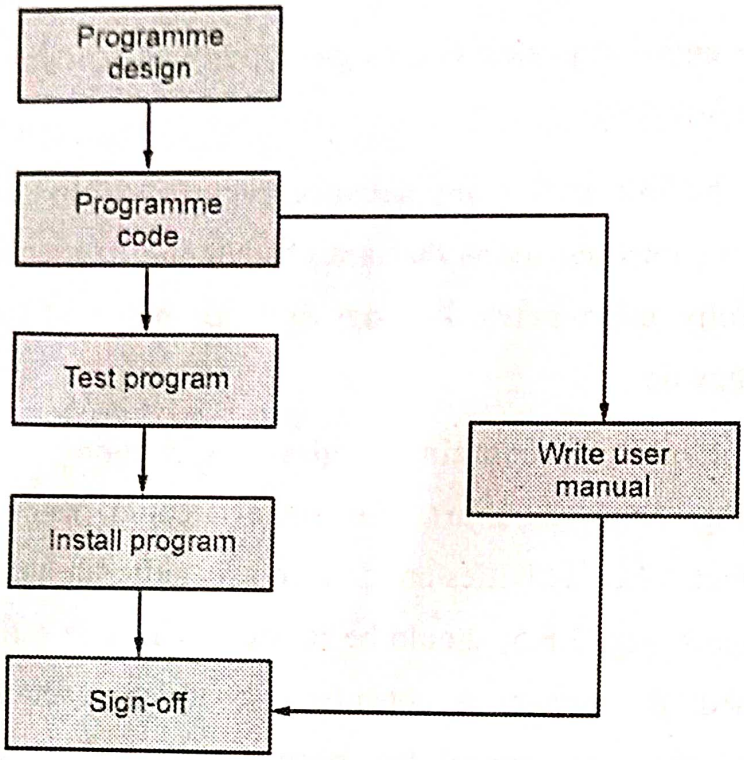


Fig. 3.6.6 : Resolving the dangle

B. Representing lagged activities

- In some circumstances, we might want to carry out two tasks simultaneously as long as there is a delay between them. When assessing a prototype, we may want to record changes made to a program as it is being tested. We may name the activity "test and document adjustments" in this situation. However, it would be hard to demonstrate that modification recording could begin, say, one day after testing began and terminate, say, a few days after testing was over.
- We depict the lag with a duration on the linking arrow as illustrated in Fig. 3.6.7 when actions can happen in parallel with a time lag between them. This means that documenting changes can begin the day after prototype testing begins and can be finished two days after prototype testing is finished.

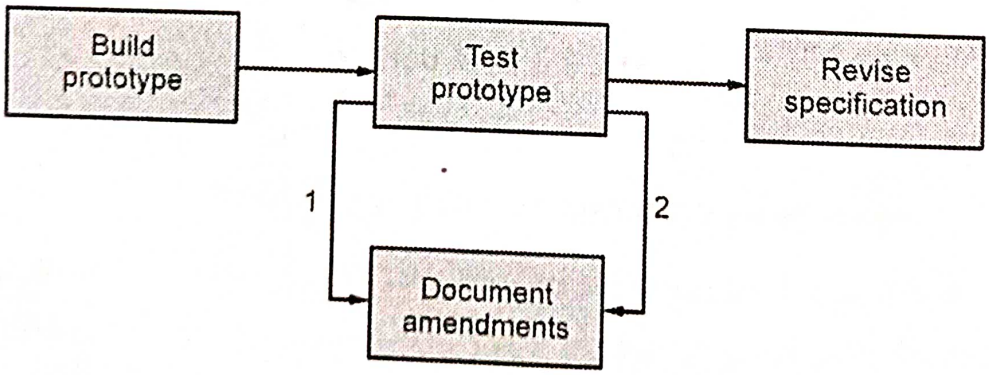


Fig. 3.6.7 : Lags Indication

C. Hammock activities

- Activities that are "hammocked" are those that, while having no duration by themselves, are considered to begin at the same time as the first activity and terminate at the same time as the last. They are typically employed to depict overhead expenses or other resources that will be used or incurred continuously throughout a set of operations.

D. Labelling conventions

Earliest start	Duration	Earliest finish
Activity label, activity description		
Latest start	Float	Latest finish

Table 3.6.1 : Labelling conventions

- From Table 3.6.1, for entering data on an activity on-node network, a variety of different conventions have been adopted.
- The activity label, which may also include a project code, is typically a code created to specifically identify the activity.
- Typically, the activity description will be a succinct activity name, like "Test take-on module."

Review Questions

1. Write a note on network planning models.
2. With the neat sketch explain formulating a network model.
3. Define a dangle. How to resolve it ?
4. How to construct precedence network ? Explain with suitable diagram.
5. Explain sequencing and scheduling activities with example.

3.7 Forward and Backward Pass Techniques

Introduction

- Two main goals of the critical path method are to design the project to be finished as quickly as feasible and to identify those activities where a delay in their execution is likely to effect the project's overall end date or the start dates of following activities.
- The method demands that we have an estimated time estimate for each action. After that, the network is examined by running a forward pass to determine the earliest possible start dates for activities and project completion and a backward pass to determine the latest possible start dates for activities and the critical path.

3.7.1 The Forward Pass

1. In the forward pass, the earliest date is recorded in the calculation. They are recorded in the network diagram of the activities and the activity table of the activity.
2. The forward pass is used to determine the earliest date on which each event and activity can be begun and completed. The earliest date for an event is the earliest date that all of the activities that it depends on may be accomplished.

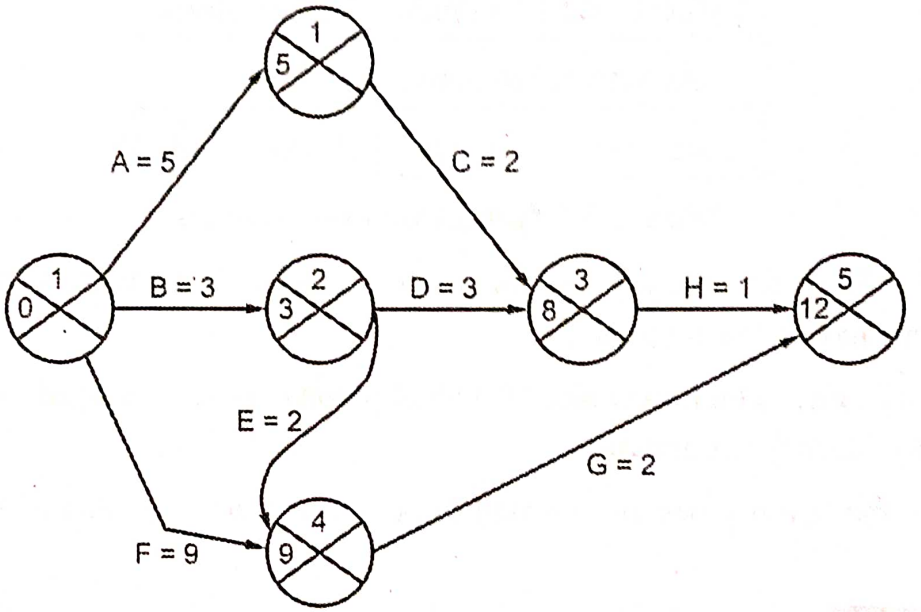


Fig. 3.7.1 : A CPM network after the forward pass

3. Dates are used to signify the conclusion of a period by convention, therefore the project is portrayed as beginning in week zero (or the beginning of week I).
4. The calculation of the forward pass and earliest start date is based on the following considerations. Because activities A, B and F can begin at any time, the earliest start date for event 1 is zero, as well as the earliest start date for these three activities.
5. Activity A will take five weeks to complete, thus the earliest it may be completed is week five (recorded in the activity table). As a result, the earliest we can complete event 2 is week five.
6. Activity B will take three weeks to complete, therefore the earliest it can be completed and the earliest we can complete event three is week three. Because activity F will take nine weeks to complete, the earliest it can be completed is week nine; but, because we have not yet estimated when activity E will be completed, we cannot say whether or not this is also the earliest date that we can complete event five. When both E and F have been finished, that is, week 9, event 5 can be performed (the later of 7 and 10).

7. We can also assume that event 4 will occur in week 8. This is the later of activity D's (week 7) and activity C's (week 7) earliest completion dates (week 8).
8. The earliest project completion date, Activity 5, is the end of week 12 and the last is week 10 (H is the earliest completed) and week 12 (G is the earliest completed).

- Fig. 3.7.1 and Table 3.7.1 shows the results of the forward pass.

Activity	Duration (Weeks)	Earlier start date	Latest earliest latest total start date finish fate finish date float
A	5	0	5
B	3	0	3
C	2	5	8
D	3	3	7
E	2	3	6
F	9	0	9
G	2	9	12
H	1	8	II

Table 3.7.1 : The activity table following the forward pass

- **The forward pass rule :** The earliest date of an event is the earliest end date of any activity that ends with the activity. If more than one event ends with the same activity, we will use the latest of the earliest end dates of these activities.

3.7.2 The Backward Pass

- The second stage is to go back and calculate the last date that each event can reach and start and stop each activity without delaying the project completion date. All follow-up actions must be started so that the project can be completed on time. We assume that the project's latest finish date is the same as the earliest finish date when computing the latest dates - that is, we want to finish the project as soon as possible.
- Fig. 3.7.2 shows our network and Table 3.7.2 shows the activity table after backward pass, because the forward pass records the activity data in the figure and the activity dates in the activity table.

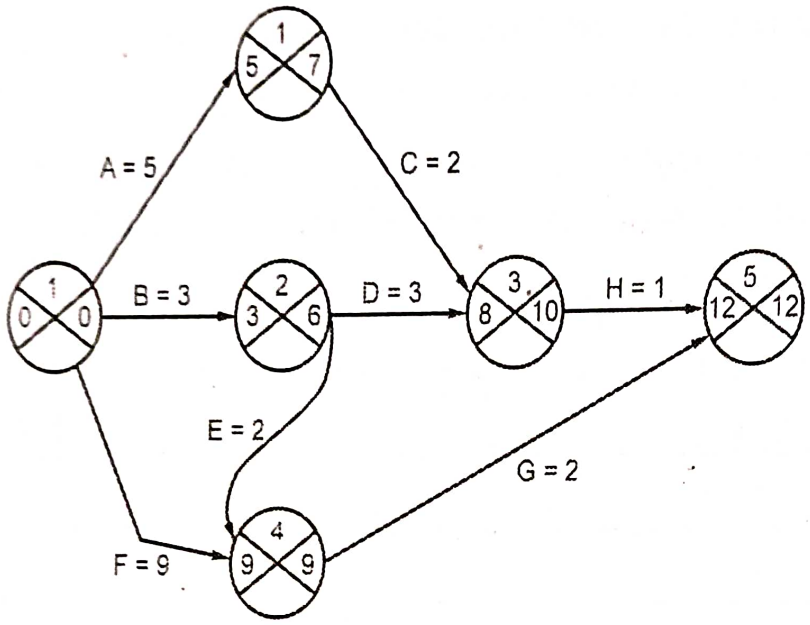


Fig. 3.7.2 : A CPM network after the backward pass

- The following are the most recent event dates :

1. Week 12 is supposed to be the latest date for node 6. The earliest date is the same as the current date.
2. The final date of activity 5 is week nine. Because activity G lasts for three weeks, if the project completion date is not exceeded, it should be completed in week twelve.
3. The final date for activity 4 is week ten, because if activity H lasts for one week and does not have to be completed before week twelve, then activity H does not have to start before week II.

Activity	Duration(Weeks)	Earlier start date	Latest	Earliest	Latest	Total
			Start date	finish date	finish date	Float
A	5	0	1	5	7	
B	3	0	2	3	6	
C	2	5	7	8	II	
D	3	3	6	7	II	
E	2	3	6	6	9	
F	9	0	0	9	9	
G	2	9	9	12	12	
H	1	8	10	II	12	

Table 3.7.2 : The activity table following the backward pass

4. The final date for activity 3 is the date when we can start activities D and E. Activity E does not have to end before week nine, nor is it required. So start before the 6th week. Activity D does not have to end before week 10. If it is three weeks, it does not have to start before week six. So the final date for event 3 is week six.
5. The final date for event 2 is week seven, because C, which lasts for two weeks, does not have to end before week ten. Unless there is an arithmetic error, the start event's earliest and latest dates must always be the same.
6. The deadline for event I is that we should be able to perform operation A (should begin from week one), operation B (must start from week two) and the deadline for operation F (must start from week 0) The date must start).

Difference between Forward pass and Backward pass

Forward pass	Backward pass
The forward pass is used to determine the earliest possible start and end dates for each activity.	The second step in analyzing a critical path network involves performing a backward pass to determine the earliest possible start and end dates for each activity without delaying the project's completion date
When an actual start date is known, actual dates may be used in the calculations. We can also utilize day or week numbers as an alternative.	Since we want to conclude the project as soon as possible, we make the assumption when calculating the latest dates that the project's latest conclusion date coincides with its earliest finish date.
A forward pass is a method for advancing across a network diagram to calculate the project's duration and identify its critical path or free float.	While a backward pass refers to going backward to the outcome in order to determine a late start or determine whether there is any slack in the process.
The procedure used in the critical path technique to choose the early start or early finish times for activities.	The procedure used to calculate the critical path method's late start or late finish times for activities.
We use forward pass to determine early finish means moving from an early start to a right turn so that the project will finish early. **Early Finish (EF) = ES + Duration** EF = 6 + 10 = 16 Days if Early Start is 6 days and Duration is 10 days.	We use backward pass, moving from late finish and subtracting from activity duration, to determine Late Start (LS). **LS = Duration – LF** If the duration is 10 days and the late finish is 30 days, then the late finish is 20 days.

3.7.3 Critical Path

1. Identifying the critical path

- Any time spent on the critical path will cause the project to be delayed. The slack is the difference between the earliest and latest dates for an event; it is a measure of how late an event can be without compromising the project's finish date. Any non-slack event is critical, because any delay in reaching that event will delay the completion date of the entire project. At least one path will always exist connecting these critical events through the network. This path is called the critical path (Fig. 3.7.3)

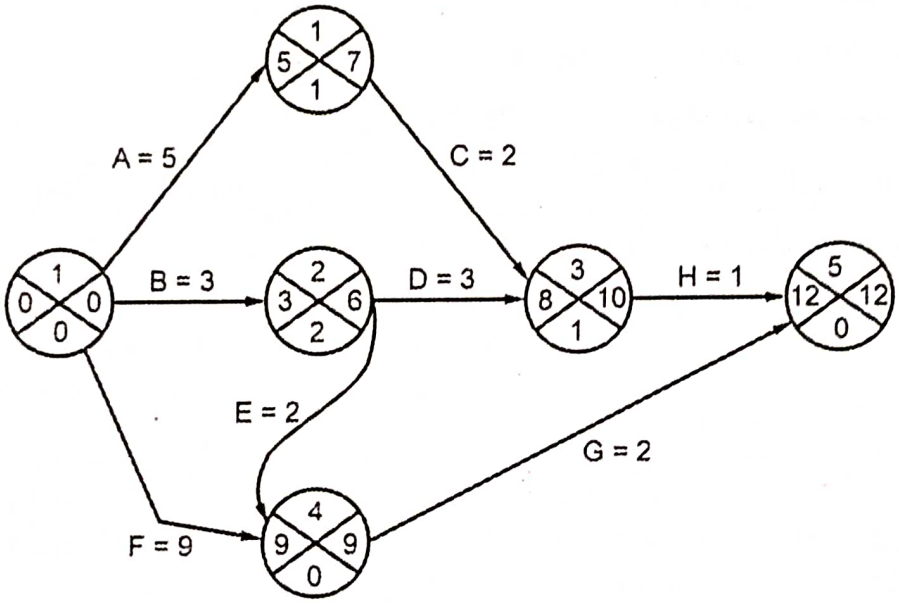


Fig. 3.7.3 : The critical path

2. The significance of the critical path is two-fold

- When it comes to project management, the next steps on the critical route must be given considerable attention in order to identify and correct the consequences of resource delays or unavailability as quickly as possible. If we want to shorten the project's overall time, we must shorten the critical path during planning. The crucial path is the network's longest path.

Remedies of critical path

- Shorten the construction period on the critical path.
- Changed task limits to allow greater planning flexibility.
- Breaks down key tasks into smaller tasks that different resources can handle at the same time.
- Check task relevance for greater planning flexibility.

3.7.4 Activity Float

- As the event slows down, the action remains the same. The cumulative list shown in Table 3.7.3. is the distinction between the first and last start dates of an activity (or the difference between the earliest and most recent end date). We can postpone the start or end of an event for an extended amount of time without influencing the project's completion date.

Activity	Duration (Weeks)	Earlier start date	Latest earliest latest start date finish date finish date			Total float
A	5	0	*m	5	7	mm
B	3	0	2	3	6	2
C	2	5	7	8	II	1
D	3	3	6	7	II	2
E	2	3	6	6	9	2
F	9	0	0	9	9	0
G	2	9	9	12	12	0
H	1	8	10	II	12	1

Table 3.7.3 : The activity schedule showing total float for each activity

- Only a certain amount of float may be available. Although the total float is displayed for each activity, it actually 'belongs' to a path that was only used once. Both Activities A and C have a total float of two weeks. If, on the other hand, activity A exhausts its float (i.e., it is not completed until week 8), activity B will have no float (it will have become critical). In such cases, publicizing entire float could be deceptive and damaging to the project's success! Other activity float metrics include the following :

1. **Free float :** The amount of time that an activity can be postponed without having an impact on other activities. It's calculated as the difference between the previous activity's earliest completion date and the next activity's earliest start date. This could be a better way to publicize the activities to the people who are actually doing them.
2. **Interfering float :** Total float and free float are not the same thing. This is a term that is frequently used, especially in conjunction with the term "free float." The interfering float tells us how far the activity can be delayed without delaying the project end date (even if it delays the start of future activities) if the free float has been exhausted (or if it is 0).

3.7.5 Shortening the Project Duration

- If we want to shorten the overall duration of the project, we usually consider shortening the duration of the activity. In many cases, this can be achieved by using more resources, working overtime or hiring more employees. We should try to save time. When trying to postpone the completion date of a project, it is obviously meaningless to reduce unimportant activities. Fig. 3.7.2 shows that we can complete the project in 12 weeks and then reduce the activity timeframe F by one week (to nine weeks).
- As we shorten the activity time of critical paths, we must constantly look for new critical paths and refocus our efforts where appropriate.
- One day, we will no longer be able to safely or cheaply shorten the duration of critical operations in order to advance the project completion date. The logical sequence of activities. Time saving is usually achieved by increasing the degree of parallelism in the network and eliminating bottlenecks (of course, always consider resource and quality constraints).

3.7.6 Identifying Critical Activities

- The critical route indicates actions that are crucial to the project's completion date; nevertheless, activities that are not on the critical path may become critical in the future.
- As the project progresses, the activity will inevitably consume its outstanding part, which will require periodic recalculation of the network.
- Once the activities on a particular route reach their total float, that route becomes a critical path and many previously unimportant activities suddenly become vital.
- Therefore, it is common practice to define "almost critical" paths of limited length. 10-20 % of the critical path duration or total inventory is less than 10 % of the unfinished project duration.
- It is important to identify critical and close to critical actions because they are most likely to cause delays in project completion.

Review Questions

1. What are forward and backward pass ?
2. What is activity table ? Explain activity table of forward pass.
3. Differentiate between forward and backward pass techniques.
4. What do you understand by critical path ?

5. What are the limitations of critical path ?
6. Write short notes on : i) Activity float ii) Shortening the project duration.
7. How to improve the critical path ?
8. Differentiate between free float and Interfering float.

3.8 Risk Management

- The process of minimizing any potential issues that can adversely affect a project's schedule is known as risk management. Any unforeseen circumstance that could have an impact on the people, procedures, technology and resources used in a project is considered a risk. Risks are events that could happen but we might not be able to predict when they will, as opposed to "problems," which are unavoidable occurrences. Project risk must be prepared for in order to be effectively managed given this unpredictability.
- It involves preparing for and dealing with challenges that could threaten the success of our project.

3.8.1 Risk

- Risk can be defined by the PM-BOK (Project Management Body of Knowledge) as "an uncertain event or circumstance that, if it occurs, has a positive or negative effect on a project's objectives."
- Risk is defined as "the possibility of being exposed to the negative effects of future events."
- The first definition covers instances where a future uncertainty actually works in our favour and gives us with an opportunity. This is where the two definitions diverge.
- Following are the main elements of a risk.

1. **It relates to the future :** The future is inevitably unpredictable. Some problems that seem clear once a project is completed, such as that expenses were underestimated or a new technology was excessively challenging to use, may not have been as obvious during planning.
2. **It involves cause and effect :** For example, A "cost over-run," for instance, might be noted as a danger, but the term "cost over-run" only refers to the damage itself and not its origin. Is it, for instance, a weak specification, an inexperienced staff member or an erroneous effort estimate ? For each risk, the cause (or hazard), such as "inexperienced employees," as well as a specific type of adverse result, such as "reduced productivity," should be specified.

- It's difficult to define where risk management stops and "normal" software project management begins. For instance, when deciding on the project's best overall strategy.

3.8.2 Categories of Risk

- Project risks are those that might hinder the project manager and the project team's ability to meet their goals.
- There is a chance that an application, even after successful implementation, would fail commercially. As a result, even if an e-commerce site may be set up appropriately to sell a product, clients may not utilize it due to the excessively high costs asked.
- The team responsible for the application's implementation is probably not directly responsible for handling these business risks. However, the business case for the project could suffer if any project objective were not met. An increase in development costs, for instance, can indicate that the revenue (or cost savings) produced by the delivered application no longer adequately justifies the higher investment.
- **There are various categories for risk.** For instance, a sociotechnical risk model developed by Kalle Lyytinen and his colleagues is depicted diagrammatically in Fig. 3.8.2.
- All the individuals participating in the creation of the application in issue are referred to in the box titled "**Actors**." High staff turnover might result in the loss of knowledge that is important to the project, which is a common concern in this industry.
- The "**Technology**" box in Fig. 3.8.2 includes both the technology used to implement the application and the technology included into the finished products. Here, risks may be related to the suitability of the technologies and potential flaws in them, particularly if they are novel.
- The term "**structure**" refers to management systems and structures, particularly those that impact planning and control. For instance, the implementation might call for user involvement in specific tasks, yet it's possible that management of the users' input is not clearly delineated.
- **Tasks** refer to the work that is scheduled. For instance, because it takes more time to assemble the numerous components, the intricacy of the work may cause delays.

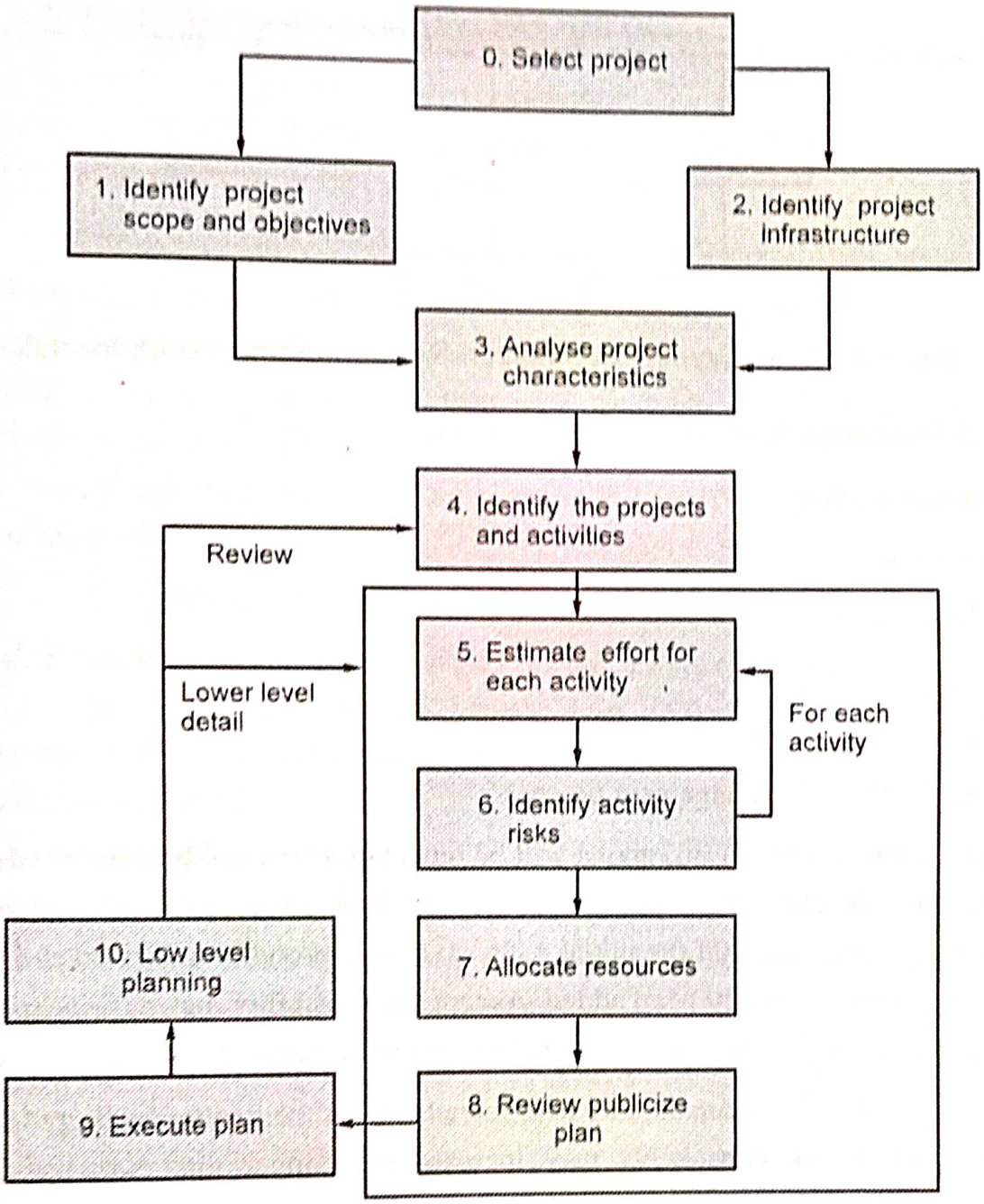


Fig. 3.8.1 : Steps 3 through 6 are where risk planning is mainly done

- All of the boxes in Fig. 3.8.2 are connected. Risks frequently result from the interactions between variables, such as those between technology and people. If a development technology is new, the developers could not be skilled at using it, which would cause outcomes to be delayed. The novelty of a new technology is actually a trait of its creators; after they become accustomed to it, the technology ceases to be "novel."

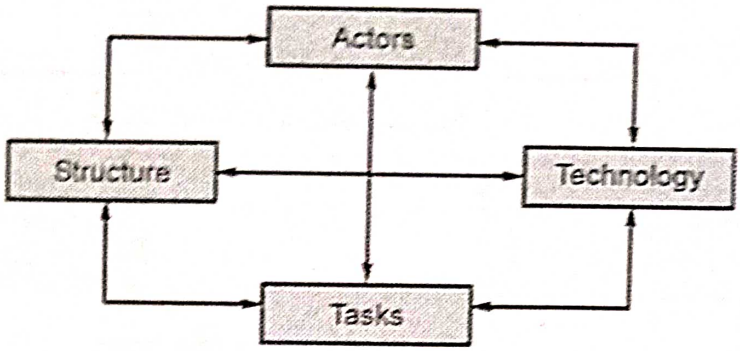


Fig. 3.8.2 : The Lyynen-Mathiassen-Ropponen framework for risk

3.8.3 A Framework for Dealing with Risk

- Risk identification,
- Risk analysis,
- Risk prioritizing,
- Risk planning and
- Risk monitoring

are all steps in the risk management process.

- Most likely, steps I through (iii) above will be repeated. Plans can be developed to lessen or eliminate hazards that could prevent a project from succeeding. The plans are then reevaluated to make sure that the initial risks have been properly mitigated and that no new hazards have unintentionally been added. Accept the possibility that staff inexperience with a new technology will cause software development to be delayed.
- It may be possible to hire consultants with knowledge of the new technology to reduce this risk. However, hiring consultants may increase the danger that expertise in the new technology will not be passed on to the permanent personnel, which would complicate software maintenance in the future. Additional risk-reduction initiatives can be planned when this additional danger has been recognized.

Review Questions

1. What is risk ? Explain risk management in detail.
2. Explain various categories for risk.
3. How to deal with risk for managing the risk process ?
4. What is a framework for risk management process ?
5. Explain different elements of risk.

3.9 Risk Assessment

- A major issue with risk identification is the possibly infinite number of risks. It is necessary to have a method for separating dangerous from likely threats. To achieve this, calculate the risk exposure for each risk using the following formula :

$$\textbf{Risk exposure} = \textbf{(Potential damage)} \times \textbf{(Probability of occurrence)}$$

- The potential damage would be evaluated as a monetary value using the most rigorous approach, however it may not always be the most useful one. Say a project was dependent on a data centre that was at risk from fire. A fresh computer configuration could conceivably be constructed for £500,000 in the event of a fire. There is also a possibility that the area where the computer is located has a 1 in 1000 chance or 0.001 percent of experiencing a fire.

$$\textbf{£500,000 3 0.001} = \textbf{£500}$$

- This value can be conceptualized roughly as the minimal amount of premium that an insurance company would demand. When the 1 in 1000 possibility of the fire occurring really happened, there would be enough money if 1000 companies, all in the same situation, each paid £500 to a fund.
- The amount of damage sustained is assumed to be constant in the calculation of risk exposure above. However, it frequently happens that there could be different levels of injury. For instance, as software development advances, more software is produced and if it were destroyed, it would take longer to recreate it.
- Some risks carry the possibility of both damage and gains. The testing of a software component was supposed to take six days, but it was finished in just three. Therefore, a team leader might believe it is acceptable to create a probability chart similar to the one in Fig. 3.9.1. This chart displays the likelihood that a task will be completed in four days (5%) followed by five days (10%) and so forth. There is a 65 percent possibility that the task will be completed on or before the seventh day, according to the cumulative probability for that day (65 percent).
- The majority of the time, clients would insist that we choose one of the days as the target. This target may be "aggressive," for example, only five days in the case above, but the chart indicates that there is an 85% probability that it will fail. Eight days, with a failure rate of only 15%, would be a more reliable estimate.
- The "loss" in Fig. 3.9.1 is really expressed in days rather than dollars. Days or another measure of personal effort are frequently employed in this context to represent financial losses in lieu of dollars.

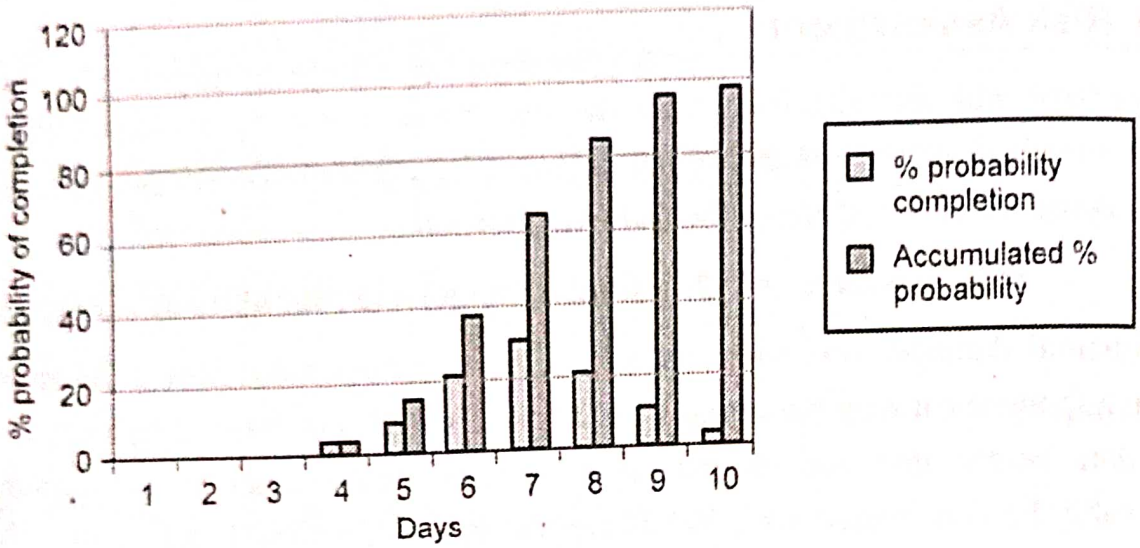


Fig. 3.9.1 : Probability chart

Ref	Hazard	Likelihood	Impact	Risk
R1	Changes to requirements specification during coding	8	8	64
R2	Specification takes longer than expected	3	7	21
R3	Significant staff sickness affecting critical path activities	5	7	35
R4	Significant staff sickness affecting non-critical activities	10	3	30
R5	Module coding takes longer than expected	4	5	20
R6	Module testing demonstrates errors or deficiencies in design	4	8	32

Table 3.9.1 : Part of Amanda's risk exposure assessment

- Most managers avoid making extremely exact predictions of loss or of the likelihood that something will happen because such numbers are frequently guessed. Because of this, Barry Boehm has suggested that relative scales in the range of 0 to 10 be used to evaluate both the risk losses and the probability. The notional risk exposure might then be calculated by multiplying the two numbers. Based on Amanda's IOE group accounts project, Table 3.9.1 offers an illustration of how this has been accomplished. Though more complex risk computations are not achievable, this value could be used to rank the importance of dangers.

- Boehm advises planners to concentrate their efforts on the top 10 threats based on risk exposure scores. The focus may be on fewer risks for smaller initiatives, such as the final-year projects of computing students.
- Even using approximate monetary values and probabilities in the range of 0 to 10 is not entirely satisfactory. The values are probably arbitrary and several analysts might choose alternative ones. Another strategy is to offer qualitative explanations of each risk's likelihood and potential impact; examples can be found in Tables 3.9.2 and 3.9.3. Associating each qualitative description with a range of values makes it easier for assessors to be consistent with one another.

Probability Level	Range
High	Greater than 50 % chance of happening
Significant	30-50 % chance of happening
Moderate	10-29 % chance of happening
Low	Less than 10 % chance of happening

Table 3.9.2 : Qualitative descriptors of risk probability and associated range values

Impact Level	Range
High	More than 30 % above budgeted expenditure
Significant	20 to 29 % above budgeted expenditure
Moderate	10 to 19 % above budgeted expenditure
Low	Within 10 % of budgeted expenditure

Table 3.9.3 : Qualitative descriptors of impact on cost and associated range values

- The potential amount of harm has been divided into groups in Table 3.9.3 based on how it would affect project expenses. Other tables could display how risks affect a project's length or the calibre of its deliverables.
- The project manager and project sponsor have some degree of control over whether the harm caused by a risk affects cost, time or the calibre of deliverables.
- R5 in Table 3.9.1, Amanda's list of risks, relates to modules taking longer to code than anticipated. As a result, the project would take longer and cost more money because more staff time would be required. Adding more software developers and distributing the remaining development work among them could be a solution. Costs will go up, but the scheduled completion date might be preserved. Another choice is to shorten the time and lower the cost of hiring people by doing less software testing before the software is made public. This may come at the expense of lower project deliverable quality.

- The risk exposure cannot be determined by multiplying the potential danger and probability of a risk when those terms are used as qualitative descriptors.
- The position of the risk in a matrix in this instance, as shown in Fig. 3.9.2, indicates the risk exposure. Different names for these matrices are probability impact grids and summary risk profiles.
- In Fig. 3.9.2, a tolerance line has zoned off part of the cells in the matrix's top right corner. The magnitude of the risks that are present in this area necessitates special consideration.

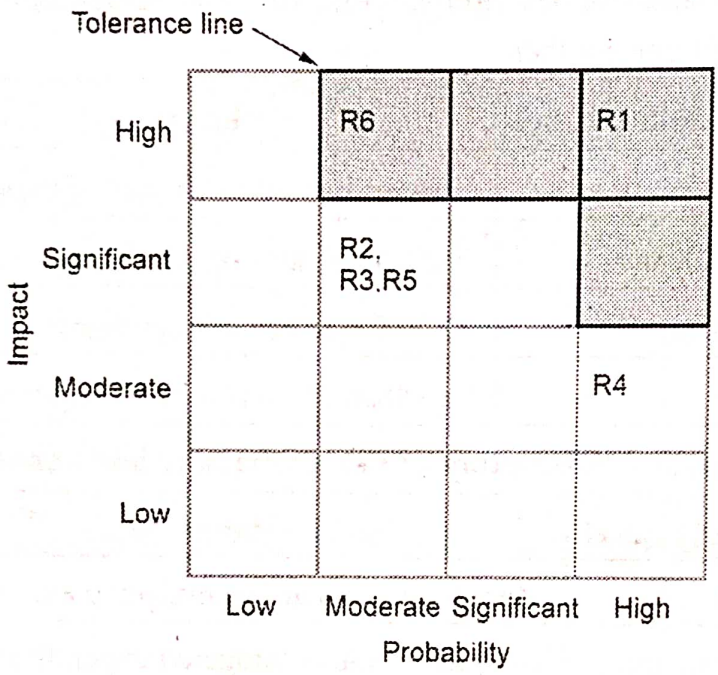


Fig. 3.9.2 : A probability impact matrix

3.10 Risk Identification

- Use of **checklists** and **brainstorming** are the two basic methods for identifying risks.

1. Checklists :

- The risks that have been discovered to arise frequently in software development projects are simply listed in checklists. A modified version of Barry Boehm's specialized list of software development hazards may be found in Table 3.10.1. A group of representative project stakeholders should ideally review a checklist outlining the risks that are relevant to their project. Frequently, the checklist offers potential countermeasures against each risk.

Risk	Risk reduction techniques
Personnel shortfalls	Hiring the best individuals; matching candidates with jobs; developing teams; promoting training and career development; and early scheduling of important personnel
Unrealistic time and cost estimates	Standardization of procedures, multiple estimation methodologies, design to cost, incremental development, documentation and analysis of previous projects
Developing the wrong software functions	User surveys, early user guides, formal specification methodologies, improved software evaluation and prototyping
Developing the wrong user interface	Task analysis, prototyping and user interaction
Gold plating	Requirements Prototyping, cost-benefit analysis, design for cost and cleaning
Late changes to requirements	Stringent procedures for change management, a high threshold for change and progressive development (deferring changes)
Shortfalls in externally supplied components	Benchmarking, examinations, written specifications, agreements on contracts, quality control processes and certification
Shortfalls in externally performed tasks	Processes for quality control, a competitive design process or prototype and contract incentives
Real-time performance shortfalls	Benchmarking, technical analysis, prototyping, adjustment and simulation
Development technically too difficult	Technical evaluation, evaluation of costs and benefits, prototyping and personnel training and development

Table 3.10.1 : Software project risks and strategies for risk reduction

- When a project is finished, project management techniques like PRINCE2 frequently advise conducting a review to identify any issues that arose and the actions that were (or should have been) done to address or prevent them. In some instances, a checklist of organizational risks that is used with new initiatives could include these issues.

2. Brainstorming

- Once some sort of preliminary plan has been created, it is ideal to bring together the key stakeholder representatives. They then determine potential issues utilising their own understanding of various project components. This cooperative method could encourage a sense of project ownership.
- Since Brigette is aware that there are aspects of college administration that she is unaware of, brainstorming may be used with her Brightmouth payroll implementation project.
- She advises the project's principal stakeholders, including the personnel office and finance office officials, to get together and address the project's biggest risks.

Review Questions

1. What are the basic methods for identifying risks ?
2. Explain various categories for risk.
3. Explain software project risks and strategies for risk reduction.
4. Explain probability impact matrix.
5. Write a note on risk assessment.
6. Define probability chart. Explain in detail.

3.11 Risk Prioritization

- The process of determining which project risks are the most important so they may be dealt with first is known as risk prioritization. The likelihood of a risk and the possible harm it causes to the organization should be used to determine priority.
- To determine which risks are most important ?
- Our team should assign a risk level to each risk after reviewing the risks of a project and entering the risks into a risk matrix. The actual evaluation is done by the project team and risk management team; frequently, those judgments can be made more simply by employing risk levels that have been determined in advance.
- Typically, risk levels are described as :
  1. **Tolerable risk** is one that is thought to have little to no impact on the project's goals and whose likelihood of happening is anticipated to be negligibly low.
  2. **Low risk** means that there is just a slight chance that it will affect the project's objectives negatively and that there will only be a small amount of worry.

3. **Medium risk** project is one that has an impact on the project's goals, budget or schedule. The likelihood of it happening is high enough to justify tight control of risk factors causing it.

- When something is **high risk**, it is thought to be very likely to happen and the consequences will be detrimental to the project's goals, budget and schedule.

3.12 Risk Planning

- Risk acceptance;
- Risk avoidance;
- Risk reduction and mitigation;
- Risk transfer;
- The task is to choose how to respond to them when the major risks have been recognized and given priority. These options will be discussed :

1. **Risk acceptance :** The default action is to do nothing. During the risk prioritizing process, we will have already chosen to disregard some risks in favour of focusing on those that are more dangerous or likely to occur. In some cases, we may conclude that the costs of taking action to lessen the likelihood of a danger occurring outweigh the harm that would be caused.
2. **Risk avoidance :** It may be advisable to completely avoid some activities because they are so prone to accidents. We avoid entering the ocean if we are afraid of sharks. Given all the issues with creating software solutions from scratch, managers might opt to stick with current clerical procedures or purchase an off-the-shelf solution, for instance.
3. **Risk reduction :** In this case, we choose to move through with a course of action despite the risk while taking measures to lower the likelihood of the risk.

- Suppose that two of the employees who were supposed to work on Amanda's development project at IOE left for other employment. If this had been recognized as a concern, actions may have been taken to lessen the possibility of employee departures.
- For instance, the developers may have been promised significant incentives to be paid upon the project's successful completion.
- Table 3.12.1, Where one risk is a challenge integrating the data formats and communication protocols of various applications. It is possible that Brigette required the chosen package to make use of a well used data management system, such as Oracle, to facilitate smoother integration.

- Sometimes it's possible to tell risk reduction from risk mitigation. Risk reduction aims to lessen the possibility of the risk happening. Actions made to guarantee that a risk's effects are mitigated when it materializes are known as risk mitigation. For instance, regular data backups would lessen the effects of data corruption but not its likelihood.

Integration	Problems merging various applications' data formats and communication protocols
Upgrading	When the provider updates the package, it might no longer exactly fulfill the users' needs. If we continue using the outdated version, our risk losing the supplier's backing for the package.
No source code	We might not be able to improve the system if we wish to because we don't have access to the source code.
Suppliers failures or buyouts	The application's manufacturer can go out of business or be acquired by a competitor.

Table 3.12.1 : Fairley's four commercial of the-shelf (COTS) so ware acquisition risks

4. **Risk transfer :** Effectively, risk transfer occurs when we purchase insurance.

- The risk is shifted to another person or business in this situation. An illustration of this in software projects would be the outsourcing of a software development work to a third party organization for a set charge.
- We might anticipate that the supplier will provide a larger estimate to cover the possibility that the project will take longer than the 'average' anticipated time.
- On the other side, a well-established external company might have an edge in terms of productivity because its developers have experience with the kind of development that needs to be done. Prices would tend to decrease as a result of the necessity to compete with other experts in software development.

3.13 Risk Control

- Implementing risk response plans, monitoring identified risks, tracking residual risks, detecting new risks and assessing the efficiency of risk management procedures are all parts of the control risks process. The main advantage of this procedure is that it increases the effectiveness of the risk strategy over the course of the project life cycle, allowing risk responses to be constantly optimized. Fig. 3.13.1 illustrates the inputs, methods and results of this process. The process's data flow diagram is shown in Fig. 3.13.2.

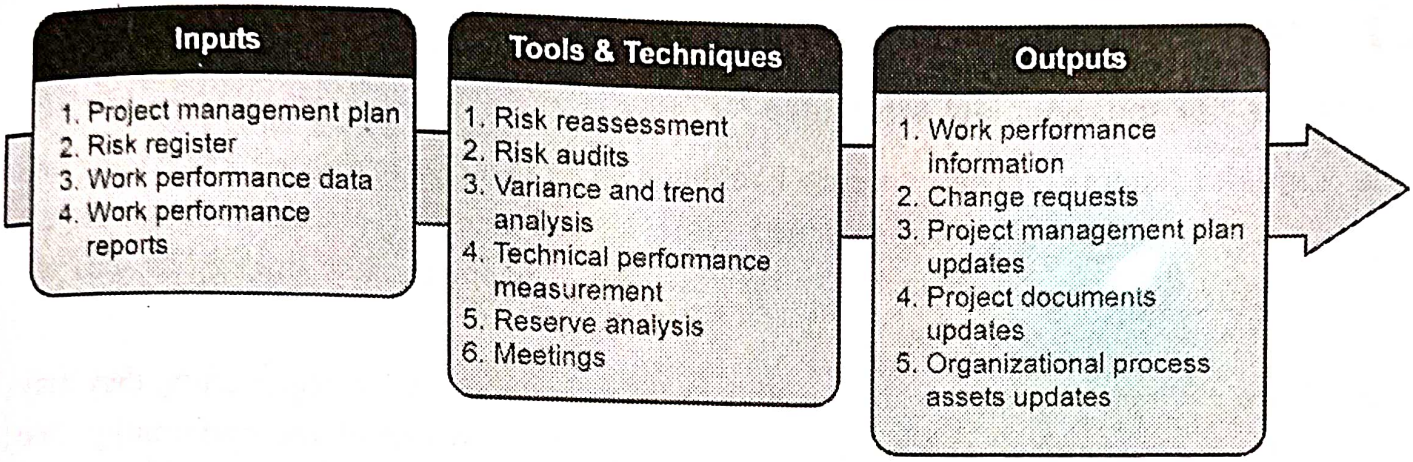


Fig. 3.13.1 : Control Risks : Inputs, tolls, techniques and outputs

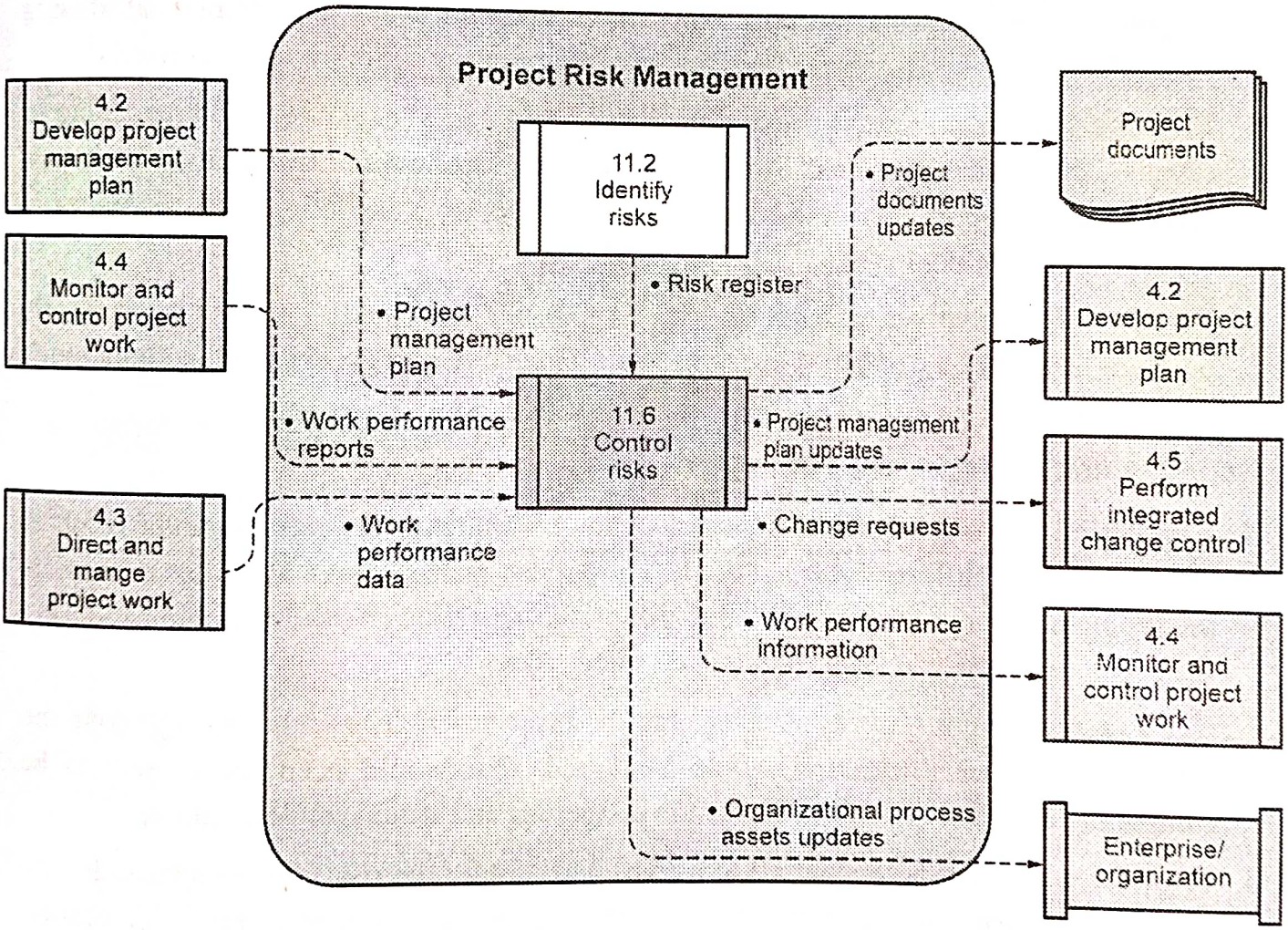


Fig. 3.13.2 : Data flow diagram of control risks

Review Question

1. Write a short note on : Risk control, risk prioritization, risk planning.

3.14 Risk Strategies

1. Positive risk management strategies

a. **Exploit :** Exploitation enhances the likelihood that a favourable risk will materialise and present an opportunity. We assigned sufficient and effective resources as the project manager to seize this chance. By making sure it happens, this strategy lessens the uncertainty related to a positive risk.

b. **Share :** When the project team is unable to fully capitalise on the opportunity, they may seek the assistance of another business. To get the most out of the opportunity, one organisation makes use of another's knowledge. Creating risk-sharing alliances, teams, corporations with special purposes or joint ventures are a few examples of sharing opportunities. In this, everyone benefits according to their actions and investments.

c. **Enhance :** Enhancing entails raising the risk's likelihood of occurring and broadening its effects. This is accomplished by figuring out how to affect the different risk triggers. Increasing an opportunity could involve giving a project's activities more resources in order to complete it sooner.

d. **Accept :** This involves utilising the favourable risk as it arises without aggressively pursuing it. It is comparable to receiving an offer that is quickly accepted without much planning.

2. Negative risk management strategies

a. **Avoid :** By getting rid of the cause, avoidance eliminates the risk. It might result in doing the action differently or not at all. The project manager may also alter or separate the problematic goal. Early knowledge gathering, better stakeholder communication and the use of expertise can all help to reduce some risks.

One example of this strategy is to change the project activity's scope or to prolong the schedule. Another illustration would be a risk that would need the project to be completely abandoned because it was too dangerous and would result in fatalities.

b. **Transfer :** The risk is transferred to a third party under the risk transfer approach. The risk is owned and carried by the third party, such as an insurance company or vendor, who is compensated to take or manage the risk on our behalf. The term "risk premium" refers to this payout. Contracts are entered into to assign risk liability to a third party.

Risk transfer lessens the risk's immediate impact on the project but does not eradicate it. An insurance policy, performance bonds, warranties, guarantees, etc. are a few transference mechanisms. The best method for reducing exposure to financial risk is this one. A risk's likelihood of occurrence is decreased by mitigation, which also lessens the risk's effects.

c. **Mitigate :** A risk's impact is minimised or its likelihood of occurrence is reduced through mitigation to acceptable levels. This strategy is founded on the core tenet that preventing a risk before it occurs is more effective than trying to remedy the problems after it has already occurred.

Utilizing cutting-edge technology or industry best practises to manufacture more defect-free products is an example of risk mitigation. The development of a prototype for mitigation may be necessary to gauge the level of risk. The risk impact reduction is targeted by finding the linkages that define the risk severity in the event that it is not possible to reduce the likelihood of the risk.

3. Contingent risk response strategies

- These strategies are only inferred when specific circumstances exist. These tactics can only be used under specific, predetermined circumstances. Before using these tactics, the team waits for enough warning signals. These warning signs may include missed deadlines, job items or milestones.
- These tactics include leveraging financial reserves, reallocating staff and putting in place workarounds in order to reduce loss, restore damage to the extent practicable and avoid a repeat of the incident.

Review Question

1. What are the different strategies of risk ? Explain any two strategies in detail.

3.15 Evaluating Risks to the Schedule

- From probability chart figure, this demonstrated the idea that the most realistic way to present a forecast of the amount of time needed to complete a task is as a graph of the likelihood of a range of numbers, with the most likely duration as the peak and the chances that the task will take longer or shorter shown as curves sloping down on either side of the peak. Thus, we can demonstrate that while a task may require five days, there is also a probability that it may require four, six, three, seven or more days.
- If a project's work takes longer than anticipated, we can hope that another task would go more quickly and make up for the delay. We will look at PERT, a method that accounts for the ambiguities in activity durations inside a project, in the sections that follow. We will also briefly discuss Monte Carlo simulation, a more potent and adaptable approach that addresses the same issue.

- The tendency for developers to adhere to the plan in practice, even when a task may be finished more rapidly, is a disadvantage of using methodologies like PERT. Even when tasks are finished earlier than anticipated, project managers aren't always fast to take advantage of the chance to begin subsequent activities ahead of schedule. The use of critical chain management will be investigated as a solution to this issue.

PERT Chart :

- PERT chart is popular visualization techniques in project management. This Chart is used to plan, control and administer the tasks that are required to complete a project. The PERT chart is a network diagram.
- The acronym for PERT Chart is (Program Evaluation and Review Technique). A PERT chart is a project management tool for planning, organizing and coordinating work in a project. It's a technique for analyzing the activities involved in completing a project, particularly the time required to perform each work and determining the shortest time required to finish the entire project.
- PERT was created to account for the uncertainty around work duration predictions. It was created in a world of high-cost, high-risk, cutting-edge initiatives, not different to many of today's huge software projects.
- The method is similar to CPM (Critical Path Method), except instead of utilizing a single estimate for each task's length, PERT requires three estimates.
- **Most likely time :** Under normal circumstances, how long would we expect the task to take ? The letter m will be used to denote this.
- **Optimistic time :** The quickest time we could expect to accomplish the activity, if nothing goes wrong. For this, we'll utilize the letter a.
- **Pessimistic time :** We'll name this b. the worst possible time, accounting for all reasonable scenarios but eliminating acts of God and warfare.
- PERT then uses the formula $t_e = \frac{(a + 4\,m + b)}{6}$ to integrate these three estimations into a single estimated duration, t_e.

Review Questions

1. How to evaluate risk for schedule ?
2. Explain PERT in detail.

3.16 Study Risk Management Tools

3.16.1 SpiraPlan by Inflectra

- Teams can access risks using SpiraPlan's centralised hub, which includes a module for recognising risks, regulating flaws, deciding actions and creating tasks that can be monitored to completion.
- A distinct artefact type with its own categories (business, technical, timetable, etc.), properties and workflows exists for risk in SpiraPlan. Users of the platform can analyse and classify risk using criteria including probability, impact and exposure.
- SpiraPlan is the best option for teams who need to maintain a validated system with risk workflow processes, including electronic signatures, because it has built-in support for risk Audit trails. Users can create risk reports in a variety of formats using the standard SpiraPlan reporting menu.
- SpiraPlan is the premier Enterprise Program Management platform from Inflectra, with an emphasis on risk management for businesses of all sizes and in all sectors.
- SpiraPlan, now in its sixth iteration, assists users in connecting strategic objectives with essential risk management strategies and supports enterprise-wide risk monitoring.
- With a complete set of tools for programme and portfolio management, release planning, resource management and risk management, this all-in-one solution integrates test management, bug tracking and requirements traceability.

3.16.2 Risk Management Studio

- One of the most useful and often used applications for risk management is this one.
- Gap analysis, risk assessment and management with therapy and business continuity manager are all included in one package.
- Because of its ISO 27001 certification, the threat library is rather large.
- The installation is simple and the annual price includes free upgrades and customer support.
- Since RM Studio is simple to learn, anyone can start using it right away like a pro.
- Excel sheets are still widely used by many of us in daily activities. This offers import and export capability for switching from Excel to RM Studio.
- In RM Studio, there is also support for reporting.

RM Studio through the hazards that lie ahead :

1. The integrated risk management framework solutions can be separately modified to better fit our particular business objectives even though they are designed to work together depending on organizational structure.
2. The risk owners and stakeholders can be effectively included in the strategy and decision-making process by using the IRMF tools for risk assessment and treatment, control implementation effectiveness, incident management and business continuity and web solution.
3. The Systems-Theoretic Process Assessments (STPA) module, which will be introduced at the end of this year and will further improve the IRMF solution, is at the cutting edge in terms of socio-technical systems risks and safety risk analysis.

3.16.3 GRC Cloud

- Resolver Systems created the top-notch risk management platform known as GRC Cloud.
- Resolver GRC Cloud enables efficient risk management, security management and incident management.
- The risk management tool enables the user to anticipate risks, monitor them once they are in the system and take necessary action when necessary.
- The risk score is used to prioritize the hazards in this risk assessment, which is based on the score. This provides a way to show the application's risk areas using a heat-map as well.
- There is a system in place for alerts and it operates automatically. The system has the ability to activate emails depending on the danger and timing of the occurrence.

Review Question

1. Explain any two study risk management tools.

3.17 Case Study : Online Shopping System

1. Online shopping risks

- Risks to online buying security include :
- **Identity fraud :** Cybercriminals typically do this by breaking into e-commerce websites and collecting users' login or credit card information. It gives them the option of using user identities to make fraudulent purchases or selling personal information to other online criminals.

- **False online retailers :** Sadly, not all online stores are reliable. Scammers may create fake websites made to resemble well-known, legitimate retailers. They imitate designs, layouts and trademarks to deceive unwary users into believing they are on a reliable website. They might sell affordable electronics, jewellery and clothing from well-known brands. Users occasionally receive the goods they have purchased, but they are typically fakes. Users occasionally receive nothing at all.
- **Data not encrypted :** On some websites, data is not encrypted. They are particularly susceptible to attacks if they lack an updated SSL certificate. Customers who share their credit card information and other sensitive information with websites whose URLs begin with HTTP rather than HTTPS run the risk of that website not being secure.
- **Violation of data :** When we shop online, we provide retailers private information like our address, contact information and bank or credit card information. In the event of a data breach, there is a chance that our information could be made public if hackers get illegal access to an e-commerce website.
- **Fake reviews :** Before making a purchase, many online customers check reviews. However, keep in mind that not all online reviews are reliable. Try to verify the source and pay attention to our gut feelings if a certain merchant consistently receives evaluations that appear too wonderful to be true or are lacking in information.
- **Fake apps :** Numerous reliable online merchants have specialised apps. Cybercriminals occasionally attempt to imitate them by making false versions. They want to collect our usernames, passwords, bank or credit card information and other personal information.
- **Unprotected WiFi :** Public Wi-Fi that isn't secure poses security hazards. One of these is the chance that hackers will stand between us and the connection point. The chance of hackers obtaining our personal information, including our contact information and credit card information, increases if our conduct online shopping on an unprotected Wi-Fi network.
- **Adware :** When accessing the internet, unwanted adverts may appear on the screen. This is known as malware. Adware, which stands for "advertising software," is designed to bring in money for its owner. Adware can be used for legitimate or fraudulent objectives by cybercriminals, such as attracting us to malicious websites where they attempt to steal our personal information. It's occasionally possible for an infection to start when we click the 'X' to close a pop-up advertisement.
- **Phishing :** Scammers do this by sending phoney emails that look to be from a reputable retailer. The emails typically include an attachment or a link that attempts to deceive the recipient into clicking it, which usually results in the installation of malware.

2. Risk management for online shopping system

- Online shopping is not without risk, though. So, yes, we do need to exercise caution. Fortunately, there is a straightforward risk management strategy we can use for online shopping. We must adhere to the five Ps when making online purchases.
- **Password :** Using secure passwords is the first rule of any online risk management procedure. The most typical error is using the same password across the board. That poses a serious security issue in addition to maybe making it simpler to memorize them. Use a unique, complicated password when creating logins for websites where we'll be making purchases. Make careful to password protect our devices with a unique password given how much online shopping is done on mobile.
- **Pricing :** When deciding which websites and vendors to purchase from, exercise caution. Pay attention to the cost. If a price appears to be too good to be true on an unfamiliar website, it probably is.
- **Protection :** Make sure the checkout is secure if we do locate something on a website we haven't used before. Additionally, this connects to conventional online risk management. Make sure the URL begins with "https" rather than "http" if we are entering personal information, but especially if we are entering financial information. It denotes the use of a secure server by the website.
- **Payment procedure :** If we can't eliminate a risk, we should endeavor to lessen the possible harm in risk management. One or both of those options are available with some payment methods. With 45 days of post-sale protection, PayPal provides excellent safety for online buyers. It's also suggested to use a prepay credit card or a low limit card designed specifically for online purchases if we don't utilize PayPal. In this manner, the harm caused if our information is compromised will be minimal.
- **Privacy :** Another that would be included in a list of common online risk management advice is the final P. Everyone values their privacy, but many people online give it away far too easily. The majority of online forms just ask for the bare minimum of information; it's crucial that we don't divulge more than is necessary. This is particularly valid in places where payment information is collected.
- A significant time saver is online purchasing. It may facilitate and even enhance holiday shopping. Just make sure we take the proper risk management precautions so we can take pleasure in the remainder of the holiday season.

3. Risk assessment for online shopping system

Strategic risks :

1. **Unpredictable consumer behavior** - Consumer traffic to online channels will undoubtedly surpass that to brick-and-mortar establishments. The scope of this change and the significance of brick and mortar locations are yet uncertain. It is now harder to predict how, where and from whom internet consumers will make purchases because lower switching costs have eroded brand loyalty and established consumer patterns. The post-pandemic, omnichannel retail environment is unpredictable, therefore e-commerce players need to be ready for a variety of potential future events.
2. **Reputational risk** - Due to the abundance of options and simple access to information, savvy online shoppers are acutely aware of the standing and core principles of e-commerce companies. A few very noticeable negative (or favourable) reviews can instantly and substantially change demand. One commercial has the power to ignite a backlash that spreads throughout social media. E-commerce players must construct their digital identities to appeal to a younger and more demanding generation of online consumers while protecting their reputation in this dynamic risk environment.
3. **Competitive pressures** - The low barrier to entry presents new e-commerce solutions with both an opportunity and a threat. Successful e-commerce merchants are increasingly attracting the attention of larger companies, like Amazon and Walmart, who benefited most from the e-commerce boom, in an era of increasing consolidation.

Operational risks :

1. Rising cyber threats are the most evident risk to e-commerce solutions. One ransom ware attack was successful every eight minutes on average last year, according to estimates of 65,000 attacks and experts expect this year to be even worse. Online retailers must invest in the necessary cyber security measures, such as using multifactor authentication, maintaining patched and current systems, installing detection and response tools, setting up a third-party incident response team in advance and cultivating a secure digital culture, to protect themselves and their customers as their businesses grow.
2. Supply - chain disruption - Since the start of the pandemic, supply-chain resilience has gained significant significance and visibility. This is especially true for e-commerce solutions. It is significantly more important to monitor and manage any supply-chain and distribution risks for a company model that depends on customers being able to order whenever they want and expecting next-day or even same-day delivery. Geopolitical hazards that could have an impact on suppliers (or supplier's supplier) are something that online retailers who buy their items from abroad must pay close attention too.

❑❑❑

Unit IV

4 Project Tracking, Monitoring and Control

Syllabus

Introduction : Project Tracking and Control, Monitoring and Control Processes, Collection of Project data, Partial Completion Reporting.

Data Collection Methods : Phone vs. Online vs. In-Person Interviews, Visualizing Progress, Visual Project Management, Kanban Boards, Project Calendars, Cost Monitoring, Four Steps in Project Cost Management, Earned Value Analysis, Project Tracking, Effective Approach to Track Projects.

Status Report : Four features of a Good Status Report, Change Control, Different factors of Change Control Process, Change Process Flow-Diagram, Software Configuration Management, Tasks in SCM Process, Participant of SCM Process.

Software Configuration Management Tools : Git, Team Foundation Server, Ansible, Managing Contracts, The Stages of Contract Management, Challenges of Contract Management, Benefits of Contract Management, Types of Contracts in Software Project Management.

Case study : Online Shopping System, track different versions of a software using Git tool.

Contents

4.1 Project Tracking and Control

4.2 Monitoring and Control Processes

4.3 Collection of Project Data

4.4 Partial Completion Reporting

4.5 Data Collection Methods

4.6 Visualizing Progress

4.7 Visual Project Management

4.8 Kanban Boards

4.9 Project Calendars

4.10 Cost Monitoring
4.11 Project Cost Management
4.12 Earned Value Analysis
4.13 Project Tracking
4.14 Effective Approach to Track Projects
4.15 Status Report
4.16 Four Features of Good Status Report
4.17 Change Control
4.18 Different Factors of Change Control Process
4.19 Change Process Flow Diagram
4.20 Software Configuration Management
4.21 Tasks in SCM Process
4.22 Participant of SCM Process
4.23 Software Configuration Management Tools
4.24 Git
4.25 Team Foundation Server and Ansible
4.26 Managing Contracts
4.27 The Stages of Contact Management
4.28 Challenges of Contract Management
4.29 Benefits of Contract Management Process
4.30 Types of Contract in Software Project Management

4.1 Project Tracking and Control

Introduction :

- Project tracking and control activities occur concurrently with project execution process group activities, so that the project is monitored and controlled while the project work is being completed by implementing the appropriate level of oversight and corrective action.
- Project tracking and controlling are processes that are required to track, review and regulate the project's progress and performance. It also identifies any areas where the project management method needs to be changed and initiates the necessary changes.
- The project is frequently inspected and measured against the project plan to ensure that cost, schedule and scope deviations are within acceptable limits and that risks and issues are continuously monitored and remedied as needed.

4.2 Monitoring and Control Processes

Introduction :

- After the work plan has been published and the project has started, progress should be monitored. This requires monitoring what is happening, comparing actual performance to the schedule and adjusting as needed. Revise plans and schedules as needed to get the project back on track as much as possible.

Creating the framework :

- The project management plan serves as the project's foundation. This document serves as a guide for tracking and controlling the project. Access to work performance information, performance reports and change requests will be required of a project manager.
- The inputs to monitoring and controlling project work are :

1. **Project management plan :** The project management plan is the primary source of information for the project's execution, monitoring and control. It is the plan, as well as any other subsidiary plans that may be required.
2. **Work performance information :** Information about project activities is known as work performance information. This information contains progress, deliverables, costs and quality assurance validations.
3. **Rejected change requests :** When examined in the context of determining how the project is progressing, reject change requests might be instructive.

- Always keep in mind that the measurement's quality is only as good as the data used to make it. While it may take some time, being alert and evaluating this information in the long run will save the time and money. So, how do we shift through the nitty-gritty details and turn them into data ? So, The good news is that this isn't a unique issue; it affects all project managers and solutions are available shown in the Fig. 4.2.1.

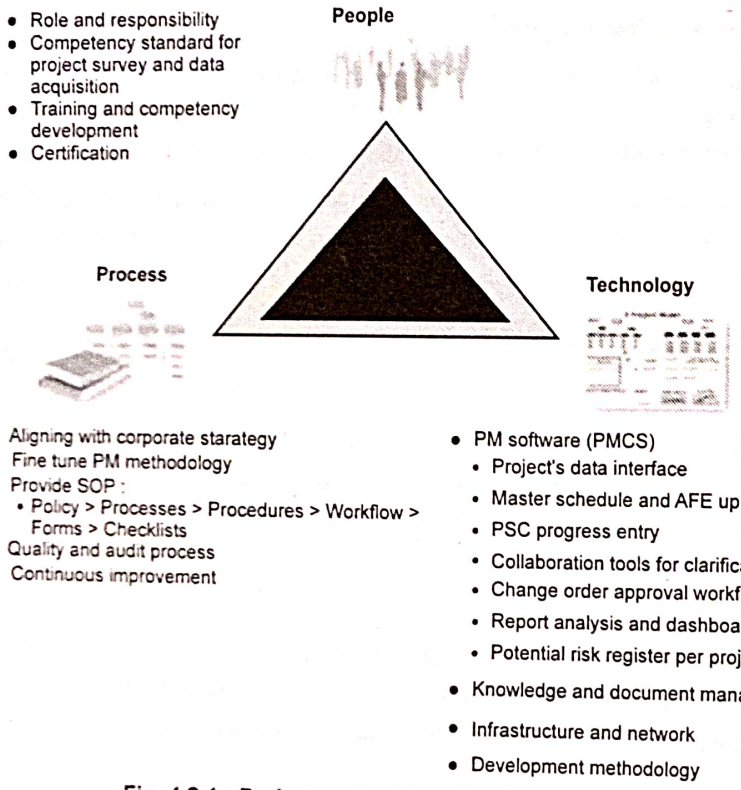


Fig. 4.2.1 : Project tracking and control solution framework

- Monitoring and regulating project operations using tools and procedures.
- The following are some of the tools and methods :
  - **Expert judgement :** Project managers and team members can use expert judgement to make judgements, such as whether to take corrective or preventive actions, based on current project information and previous experience with similar projects.
  - **Earned Value Technique (EVT) :** Earned Value Technique (EVT) is a method of calculating the current project schedule and cost performance for project managers. This data can then be used by project managers to predict future schedule and cost performance.

  - **Organizational project management :** The organizational project management approach offers project managers with detailed instructions and processes to enable effective monitoring and control during each stage of a project.
  - **PMIS (Project Management Information System) :** A PMIS helps us to keep track of and regulate variables like cost and resource utilization. A Project Management Information System (PMIS) can also help project managers calculate and manage earned value data, as well as request and update project data automatically.
- Using the right tools and strategies helps keep us informed about the status of our project.
- The following are the outputs of project work monitoring and control :
  - **Recommended corrective actions :** These are based on project work performance information. The project manager or team employs professional judgement to provide solutions to problems that have occurred by comparing this information to the project plan.
  - **Preventive actions recommendations :** These are based on project work performance data. The project manager or team utilizes professional judgement to identify strategies to prevent project risks by comparing this information to the project plan.
  - **Predictions :** Forecasts allow for the projection of successful project outcomes based on work performance data collected during the monitor and control project work process.
  - **Recommended defect repairs :** These are the results of project monitoring and control. When a product fails to satisfy quality standards, this output suggests what has to be done to fix it.
  - **Requested changes :** These are new actions that are required to accomplish the project's goals. The project manager or members of the project team frequently make requests in order to improve processes or solve difficulties.
- Every company is different and project monitoring and control experiences will vary depending on firm's project management approach and the project type.
- For defining projects and managing their implementation, there are a variety of frameworks available. IT project framework for Entrepreneurial Culture - Iterative System Development is depicted in the Fig. 4.2.2. The EC-ISD project framework is a realistic method for establishing a scalable structure that promotes discipline and best practices. It provides a high-level overview of the phases needed in any IT project, albeit the level of information required for each step will vary depending on the project's complexity and criticality. For smaller, less significant projects, all of the procedures in the first two columns, for example, might be done in a two - hour meeting with an email summary.

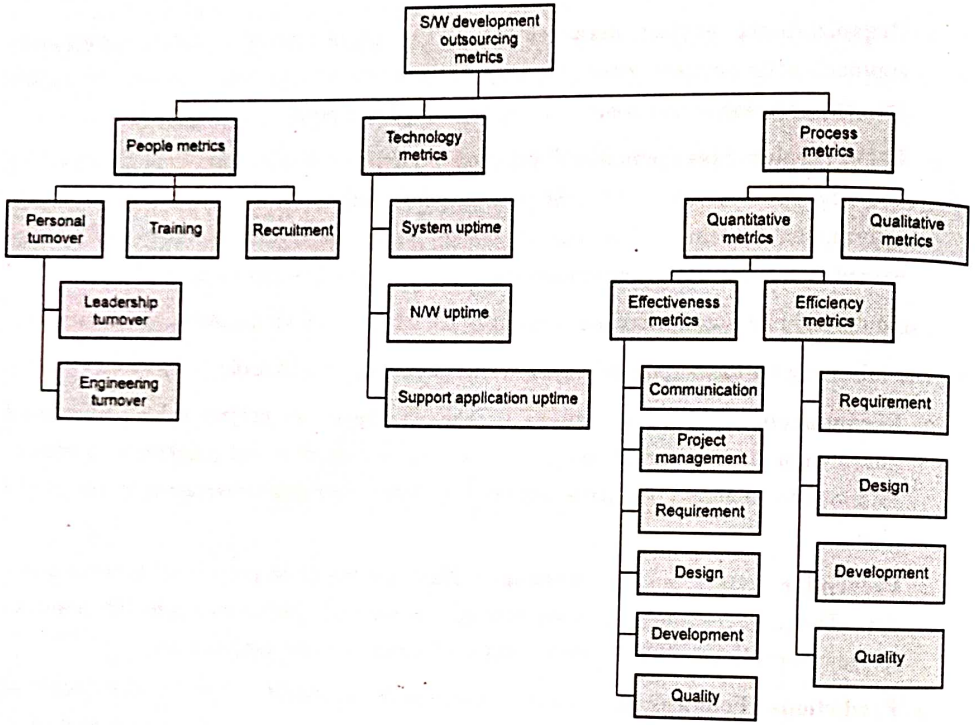


Fig. 4.2.2 : EC-ISD IT project framework

Review Questions

1. Explain in detail about creating the framework for monitoring and control.
2. What are the different tools and methods used for monitoring and regulating project operations ?

4.3 Collection of Project Data

- Any form of research project requires data collection. Inaccurate data gathering might skew a study's findings and lead to erroneous conclusions.
- For impact evaluation, there are a variety of data collection approaches. Quantitative data gathering methods are on one end of the spectrum, while qualitative data collection methods are on the other.
- "What data should I collect ?" is a question that many people have. The answer is to gather all of the information needed to create the metrics primitives and additional qualifiers.

- In most circumstances, the ideal response to the question of "who should gather the data ?" is the "owner" of the data. The data's "owner" is the person who has direct access to the data's source and in many situations, is also responsible for its generation. The proprietors of various types of data are depicted in Table 4.3.1.
- The following are some of the advantages of having the data owner gather the data :
- Data is captured in real time as it is generated, ensuring accuracy and completeness.
- Data owners are more likely to notice irregularities in data as it is being collected, which improves accuracy.
- Duplicate recording (once by the data recorder and twice by the data entry clerk) eliminates human error, increasing accuracy.

Owner	Data owned
• Management	• Project schedule, budgets
• Programmers	• Time per task, bugs, cause for bugs
• Tester	• Test case plan/execution/passed • Test coverage
• End users	• Real time problems • Time spent on application

Table 4.3.1 : Proprietors of various types of data

- There should be a training programme to ensure that the data collectors understand what to do and when to do. Appropriate procedures must be developed and documented in the creation of the training programme. These courses might be as short as an hour with simple collection mechanisms. The best outcomes will get from practical, interactive training, where the group works on actual data collecting situations.
- Without this training, support workers' hours can be spent to respond repeatedly to the same inquiries. Another advantage of the training is that it fosters a sharing of when and how the data are collected. This decreases the chance of invalid and incoherent data being collected.
- When the correct data are not correctly collected, it is not possible to achieve the goals of the measuring programme. Without proper data, data analysis is useless. The creation of a good data collection plan is therefore the cornerstone of any successful programme of measurements. The collecting of data should be :

- **Consistency :** The same person will gather the data in the same manner every time.
- **Unambiguous :** Two people taking the same measurement for the same item will get the same results.
- **Convenient :** Data gathering must be simple enough to not disrupt the individual collecting the data's working patterns. As a result, data collecting must become an integral component of the process rather than an afterthought.
- **Accessible :** In order for data to be useful and utilized, it must be easily accessible. This means that even if data is collected manually on forms, it must be entered into a metrics database at some point.

- It is widely agreed that as much of the data collection process as feasible should be automated. Standardized forms should be utilized at a minimum for data collection, but the data from these forms must be entered into a metrics database at some point if it is to be relevant in the long run.
- Another technique to inject human error into metrics values is to dump raw data and hand tally or calculate metrics. Even if the data is kept in a simple spreadsheet, automatic sorting, data extraction and calculation are all available and should be used. Using a spreadsheet or database also speeds up the production of measurements when compared to hand tallies.

4.3.1 Type of Tools

- Various data collection methods necessitate the use of appropriate recording forms. These are referred to as data collection tools or instruments. They are as follows :

Observation schedule

- Guide to conducting interviews
- Timetable for interviews
- Questionnaire sent by mail
- Scale of evaluation
- List of items to consider
- Data sheet/document schedule
- Institutions calendar

- Each of the instruments listed above is used for a specific type of data collection :
  - Schedule of observations for method of observation
  - Interview schedule and guidance for interviewing and questionnaire

- For example, a mail survey : The data collection tools translate the study objectives into particular questions / items, the answers to which will offer the data needed to meet the research goals. To do this, each question / item must transmit to the respondent the notion or collection of ideas required by the research objectives and each item must elicit a response that can be analyzed in order to meet the research goals.
- Below is a brief explanation of the various data collection tools.

Schedule of observations

- This is a format for recording observations of an object or phenomena. The things to be observed are chosen in accordance with the study's nature and goals. They're organized into categories and mentioned in the schedule in the order in which the observer would notice them.
- The schedule must be designed in such a way that it provides the necessary verifiable and quantitative data while also avoiding selective bias and misinterpretation of observed items. Simple and properly stated observation units are required to ensure precise and uniform recording.

Guide to interviewing

- Non-directive and depth interviews are conducted using this method. It does not provide a detailed list of items for which information must be obtained from a responder; rather, it focuses on the broad subjects or areas that will be discussed throughout the interview.
- During the interview, the interview guide serves as a helpful reference or prompter. It assists in focusing attention on important study points and obtaining similar results in different interviews conducted by the same or different interviewers.

Schedule of interviews and questionnaire sent by mail

- In surveys, both of these tools are commonly employed. Both are comprehensive lists of questions designed to collect information from responses. The main distinction between them is how they record replies. The respondent fills out a questionnaire while the interviewer writes up a timetable.

Scale of evaluation

- This is a form for capturing individual attitudes, aspirations and other psychological and behavioral characteristics, as well as group behavior.

Checklist

- This is the most basic of the devices. It is a prepared list of items that are relevant to an object or a specific task. Each item's presence or absence can be specified by checking "yes" or "no" or using a multipoint scale. The use of a checklist guarantees that all parts of the object, act or task are considered. Checklists contain phrases that the respondent understands and that express his opinions more briefly and succinctly than open-ended questions.
- It's a rudimentary device, but with proper pre-testing, it may be made less so. When used to examine specific hypotheses, it is at its best. It can be used on its own or as part of a timetable or questionnaire.

Data Sheet / Document Schedule

- This is a list of data that can be gleaned through documents, records and other sources.
- The components contained in the schedule are limited to those that can be uniformly obtained from a large number of case histories or other records in order to secure measurable data.

Institutional calendar

- This is used to conduct surveys of businesses, educational institutions, social and cultural organizations and other organizations. It will contain information on their profile, functions and performance in several categories.

4.3.2 Construction of Schedules and Questionnaires

Questionnaire vs. Schedule

- The most frequent data collection instruments are schedules and surveys. Both of these tools have a lot in common. Both feature a set of logically related questions; both aim to elicit responses from respondents; and in both cases, the content, response structure, question wordings, question order and other factors are the same for all respondents. So why are they referred to as "schedule" and "questionnaires," respectively ? This is due to the fact that the strategies they employ are different. A questionnaire is utilised for mailing, whereas a timetable is used for interviewing.
- Because of this difference in usage, there is a slight distinction between these two recording formats. In a face-to-face interview, the interviewer fills out a schedule, whereas the respondent fills out a questionnaire. As a result, two distinct terms are required.
- When used for interviewing, the tool is referred to as a schedule; when delivered to a respondent for completion and return, it is referred to as a questionnaire.

The construction process

- Except for a few minor modifications in mechanics, the process of creating a schedule and a questionnaire is nearly identical. This isn't a simple issue of making a list of questions that come to mind. It's a methodical procedure that takes a lot of time, effort and thought. It consists of the major steps listed below :

1. **Data must be determined :** Because an interview schedule or postal questionnaire is a tool for acquiring data for a specific study, it should be built logically from the information needed for that study.
2. **Development of "dummy" tables :** Developing "dummy" tables in which to present the data to be acquired is the best way to ensure that the information requirements are met.
3. **Establishing the respondents' level :** Who are the people who have responded to our survey ? Are they people who have specialist knowledge about the problem under investigation ? Or are they ordinary citizens ? What is their level of understanding and knowledge ? The choice of terms and concepts is determined by the respondents' degree of knowledge.
4. **Choosing a data collection method :** Should we do a face-to-face interview or send a letter ? The type of question structure we use is greatly determined by the communication medium we use.
5. **Instrument composition :** Following the determination of the data needed for the study, a basic outline of the instrument, identifying the many major categories of data, can be written.

   Second, the order in which these categories appear must be determined. The third step is to make a list of the questions that will be asked under each group category. All elements that are relevant to the 'data need' should be gathered.
6. **Evaluation of the draught instrument :** The researcher must thoroughly evaluate each question in the draught instrument in cooperation with other knowledgeable individuals.
7. **Pre-testing :** The new draught must be pre-tested in order to detect the instrument's flaws and make the necessary changes to correct them.
8. **Procedures / instructions specification :** Once the instruction has been finalised after pre-testing, the procedures or instructions for its use must be provided.
9. **Choosing a format :** The format should be appropriate for the research. The instrument should be organized into sections that correspond to various facets of the situation.

Question construction

- A survey instrument - interview schedules or questionnaire - is useful for collecting different categories of information, viz.,
  a) Factual information - Facts about the respondents : Sex, age, marital status, education, religion, caste or social class, income and occupation; as well as historical information and conditions.
  b) Psychological data such as attitudes, views and belief and
  c) Socioeconomic data expectancies and behavioral data, such as social participation, etc.
- Question construction can begin after determining the information need, as mentioned in the preceding article. There are four primary decision areas here.
- They are
  (a) Relevancy and content of the question,
  (b) Question language,
  (c) Response form and
  (d) Question order or sequence.

Relevance and content are two factors to consider

- Query should pass specific checks in order to be included in the instrument. Is it connected to the study's goals ? Is it likely to provide useful information in the case of a police investigation ?
- If it isn't already in the instrument, it should be.

Wording of the question

- This is a challenging task. In a schedule / questionnaire, the purpose of a question is to elicit specific information without distorting it. "Interrogating people is more like attempting to catch a specific elusive fish by casting different kinds of bait at different depths, without knowing what goes on underneath the surface," adds Oppenheim. Because words have different meanings depending on who we ask, the question designer should use terms that have the following characteristics :
  - Shared vocabulary
  - Uniformity
  - Exactness
  - Simplicity
  - Neutrality

- Some more problem areas are :
  a. Unwarranted assumptions,
  b. Personalization,
  c. Presumptions,
  d. Hypothesis,
  e. Questions in difficult cases.

Order of questions or sequence of questions

- The arrangement of questions in a timetable / questionnaire is just as significant as the language of the question. It has two main consequences. First, a suitable sequence can easily answer the questioner's task. Second, because of contextual effects, i.e. the impacts of previous questions on reply to later questions, the sequence could either produce or prevent biases.

Timetable and questionnaire mechanics

- In the designing of a schedule / questionary, in addition to question wording and question building, the mechanics of the form should be considered. The form mechanics has various aspects : form items, instructions, precoding, etc.
- **Form items :** For schedules and questionnaires, the following are mandatory :
  1. At the top of the main page should appear the name of the organization that collects data.
     The name, if applicable, of the sponsor of the study.
  2. In a huge type next to the organization's name on the first pages the title of the study should be shown. The title - e.g., "Consumer schedule;" - may be noted underneath this title.
  3. Cleat should be made of the confidentiality of the data.
  4. A place should be provided to write a date of completion on the form.
  5. Each tool copy can be provided a serial number.
  6. Number of pages of the instrument.

4.3.3 Measurement Scale and Indices

- In social science research, scales are designed to measure variables. In the last decades, researchers in sociology, psychology, education, psychiatry, ethics, behavior, economy, administration and other domains have drawn tens of thousands of scales.

4.3.4 Pilot and Preliminary Trials

Pilot study

- It's difficult to organize a big study or project without a thorough understanding of the topic matter, the target population and their degree of knowledge and understanding, among other things. What are the problems that need to be addressed ? What are the ideas that go along with the topic ? What are the options for putting these into action ? Which research approach is best ? What is the expected duration of the study ? What will the price tag be ? These and other such concerns necessitate a thorough understanding of the study's topic matter and scope. A preliminary research is carried out in order to get such pre-knowledge of the subject matter of a large study. A pilot study is what it's known as.

Preliminary trials

- A pre-test is a trial test of a single aspect of the study, such as the method of data collecting or data collection instrument - interview schedule, mailed questionnaire or measuring scale - while a pilot study is a full-fledged tiny investigation of a topic.

The requirement for Pre-testing

- The data collection instrument is created keeping the study's data requirements in mind. However, it cannot be perfected just through the designer's and other researchers' rigorous inspection. It should be put to the test. As a result, pre-testing a draught instrument is essential.
- Pre-testing entails putting the instrument through its paces with a small group of people before finalizing it.

Review Questions

1. What are the different ways of collecting data ?
2. What is check list ?
3. What is difference between schedule and survey ?
4. Explain in detail construction process of schedule for data collection.

4.4 Partial Completion Reporting

Collecting the data

- As a rule, managers will strive to interrupt down lengthy sports into extra controllable responsibilities of 1 or weeks' duration. However, gathering data on half completed tasks will still be required, especially predictions of how much work has to be done. It may be difficult to make such forecasts accurately.

- Where there may be a sequence of products, partial of entirety of sports is less complicated to estimate. Counting the number of document specs or display screen layouts produced, for example, can offer an inexpensive degree of progress. In a few cases, intermediate products may be used as in-interest milestones. The first sucessful compilation of a program, for example, is probably taken into consideration a milestone although it isn't a very last product.
- Many organizations use standard accounting systems with weekly timesheets to calculate the hours employees spend on their individual jobs. Labor hours posted to a project indicate the work performed and the cost of the project. However, the project doesn't say so Manage what has been produced or whether tasks are on time.

Time Sheet

Staff : John Smith Week ending : 30/3/07

Rechargeable hours

Project	Activity code	Description	Hours this week	% complete	Scheduled completion	Estimated completion
P21	A243	Code mod A3	12	30	24/4/07	24/4/07
P34	B771	Document take-on	20	90	6/4/07	4/4/07
Total recharged hours			32			

Non-rechargeable hours

Code		Description	Hours this week	Comment and authorization
Z99		Day in lieu	8	Authorized by RB
Total non-rechargeable hours			8	

Fig. 4.4.1 : A weekly timesheet and progress review form

- It is consequently common to evolve or beautify current accounting statistics series structures to satisfy the desires of task control. Weekly timesheets, for instance, are often tailored through breaking jobs right all the way down to pastime stage and requiring statistics approximately paintings executed further to time spent. Fig. 4.4.1 three suggests an instance of a record form, in this example soliciting for statistics approximately possibly slippage crowning glory dates in addition to estimates of completeness. Other reporting templates are possible. For example, instead of asking for an estimate of the percentage of tasks completed, some managers ask for the number of tasks completed.
- The amount of time we have already worked on the task and the estimated number of hours it will take to complete the task.

4.4.1 Red / Amber / Green (RAG) Reporting

- Avoiding asking for predicted completion dates in favor of asking for the team members' assessments of the chances of meeting the specified target date is one popular strategy for overcoming the objections to partial completion reporting.

Activity Assessment Sheet

Staff : Justin

Ref : IoE/P/13 Activity : Code and test module C

Week number	13	14	15	16	17	18	
Activity summary	G	A	A	R			
Component							Cpmments
Screen handling procedures	G	A	A	G			
File update procedures	G	G	R	A			
Housekeeping procedures	G	G	G	A			
Compilation	G	G	G	R			
Test data runs	G	G	G	A			
Program documentation	G	G	A	R			

Fig. 4.4.2 : A traffic-light assessment of IoE / P / 13

- The traffic-light system is one way to accomplish this. The steps in this are as follows :
  - Rrecognize the primary (first-level) components of a work that should be evaluated;
  - Dissect these essential components into their component parts (second level);
  - Rate each second-level component according to the following scale : Green for "on target," amber for "off target but recoverable" and red for "off target and only tough to recover from";
  - To arrive to first-level assessments, evaluate all the second-level assessments;
  - Create an overall assessment by reviewing the first- and second-level assessments.
- For instance, Amanda decides to review activity on the IOE project using a variation of the traffic-light system. She divides each task into a number of component components (deciding that additional division is not required in this instance) and asks the team members to complete a return at the conclusion of each week. Fig. 4.4.2 shows Justin's finished evaluation at the conclusion of week 16.
- Traffic-light assessment does not attempt to estimate work completed or quantify anticipated delays; it just shows the danger of non-achievement.
- The project manager utilizes these as a foundation for assessing the overall status of the project when all assessment forms for all activities have been completed. Any crucial task marked as amber or red calls for more thought and frequently necessitates a change to the project timeline. If non-critical operations are categorized as red, they are more likely to be seen as a concern, especially if all of their float is expected to be used up.

4.5 Data Collection Methods

Data collection :

- A review meeting is entirely a human endeavor, thus if the information describing the outcomes of the meetings is not correctly recorded, it could be lost. The information regarding the time the reviewers spent performing the review activity must be recorded in addition to the problems. To track faults throughout the project, a record of the defect data is required.
- The following are the several reports where the review data is gathered :

1. **Go over the preparations log :** A review preparation log is created by each reviewer. The information concerning problems the reviewer notices, their positions, their criticality and the overall time spent performing the evaluation of the work product are among the several items documented in it.

2. **Review log :** Only errors acknowledged by the author are recorded in the review log. A vital record, defect logs assist in tracing all defects through to closure.
3. **Review report summary :** The data from the review are compiled in this report, along with an overview of the review. It includes data on the overall defect count as well as the time allocated to each review process activity.

4.5.1 Phone vs. Online vs. In - Person Interviews

Phone interviews	Online interviews	In-person interviews
High confidence	Self manage, data accuracy	In-depth
Need to hire an agency	Need internet	Time consuming

Phone interviews

- **Pros :** High level of confidence in the information gathered, accessible to practically everybody.
- **Cons :** Costly, unable to administer oneself, requiring agency employment.

Online interviews

- **Pros :** Low cost, self-administer and very unlikely to include data errors.
- **Cons :** Not all of our consumers may have email addresses or be online and other customers may be reluctant to share information online.

In-Person interviews

- **Pros :** Comprehensive data with a high level of confidence.
- **Cons :** Costly, time-consuming and susceptible to being discounted as hearsay.

4.6 Visualizing Progress

- The project manager only needs to provide a few remarks to generate a project status report for the project and the project management report will be generated automatically, showing :
  - Milestones from the previous period and if they were met, milestones due the following time.
  - Action items that have not been completed by the deadline.
- The Project Manager's opinion on action items due to be completed in the next period.
- The report can be sent to a specific distribution list via e-mail. The project manager can decide on the time frame.

- The report's worth isn't as high as it appears. The benefit comes from ensuring that the team is required to examine due and outstanding items at least once per reporting period. This usually forces them to concentrate on accomplishing tasks and milestones in order to avoid being included in the report. This function can be used as a project status report template or as software for reporting period status.

Benefits of visualizing progress

- It is simple to understand ,as the information is presented in a single, consistent style.
- The project team is required to review progress on a regular basis.
- To create a report, only a small amount of effort is required.
- Periods can be customized to meet the needs of the firm.

Methods used for visualizing project progress

- A management must show the data collected regarding the project's development after gathering it. Some approaches for presenting the picture of a project and its future are :
  1. **Gantt chart :** A Gantt chart is essentially an activity bar chart with activity floats that shows the scheduled activity dates and duration.
  2. **Slip chart :** The slip chart provides a more vivid visual depiction of the operations. The more the slip line bends, the more deviation from the plan there is. A very jagged slip line suggests that rescheduling is required.
  3. **Time line :** A timeline is a way of keeping track of and illustrating how targets have changed during the course of a project.

The Gantt chart

- The Gantt chart is one of the earliest and most straightforward methods for monitoring project progress. Essentially, this is an activity bar chart with activity labels that shows the duration and dates of planned activities. The graphic shows reported progress (often by coloring activity bars) and a "today cursor" shows which tasks are ahead of or behind schedule immediately.
- Fig. 4.6.1 depicts a portion of Amanda's Gantt chart as of Tuesday, Week 17's final day. 'Code and test module D' has been finished earlier than expected and 'Code and test module A' likewise seems to be going more quickly than expected. The remaining two module's coding and testing are running behind schedule.

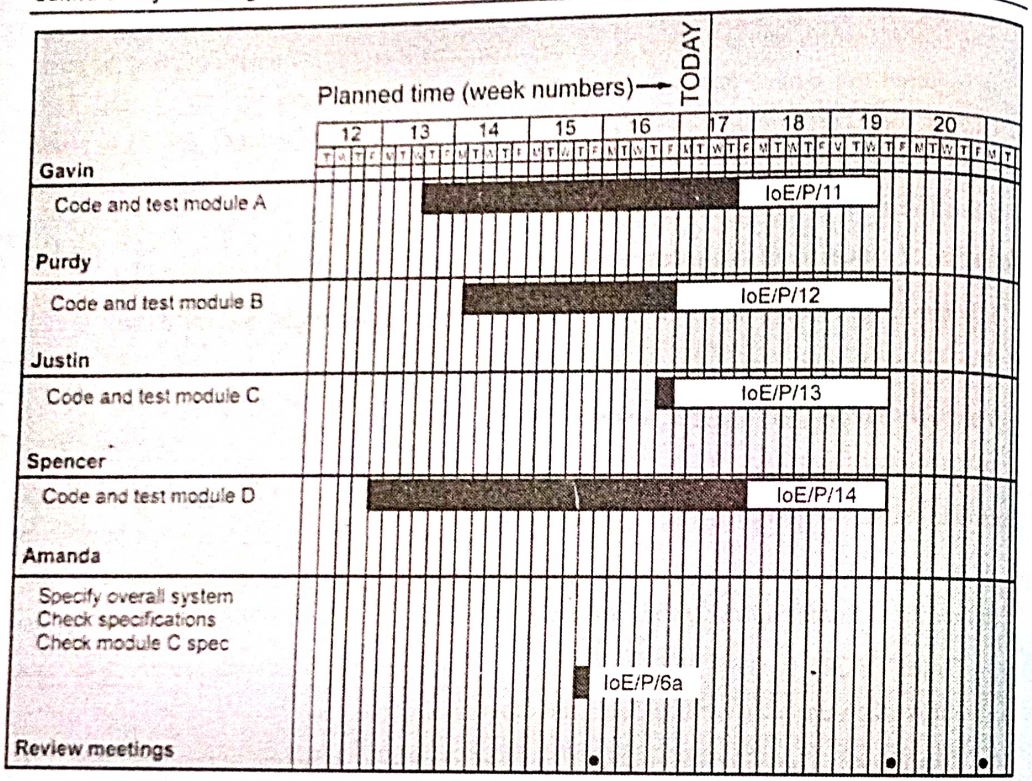


Fig. 4.6.1 : Part of Amanda's Gantt chart with the 'today cursor' in week 17

The slip chart

- A very similar alternative is a slip chart (Fig. 4.6.2), which some project managers prefer because they think it gives a more glaring visual indicator of which operations are not moving along according to plan : The more the slip line slants, the more deviation there is from the original plan. At regular intervals, new slip lines are inserted and as they accumulate, the project manager may see whether or not the project is progressing (as evidenced by following slip lines bending less). Rearranging the schedule is indicated by a highly jagged slip line.

The timeline

- The charts that have been described thus far have the drawback of not clearly displaying the project completion date slippage over the course of the project. We can forecast the project's future progress by analyzing and comprehending tendencies in the project's past performance. For instance, if productivity hasn't been as great as expected so far on a

project, it's possible that the project's completion date would be delayed even more unless measures are done to make up for or increase productivity.

- The timeline chart is a tool for documenting and visualizing how targets have changed during the course of the project.
- A timeline chart for Brigette's project at the conclusion of the sixth week is shown in Fig. 4.6.3. The horizontal axis is used to plot planned time and the vertical axis is used to plot elapsed time. The lines meandering down the chart show the dates on which the various activities are supposed to be completed. For example, at the beginning of the project, the activity "analyze existing system" was supposed to be finished by the Tuesday of week 3, the activity "obtain user requirements" by the Thursday of week 5, the activity "issue tender" was supposed to be finished by the Tuesday of week 9 and so on.

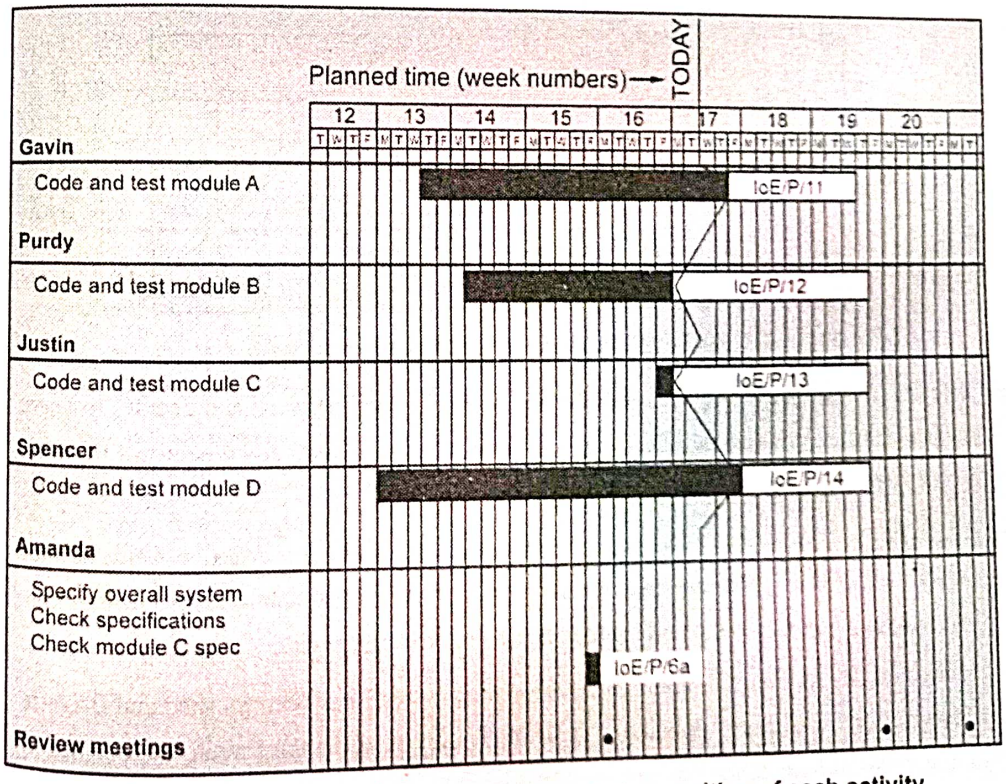


Fig. 4.6.2 : The slip chart emphasizes the relative position of each activity

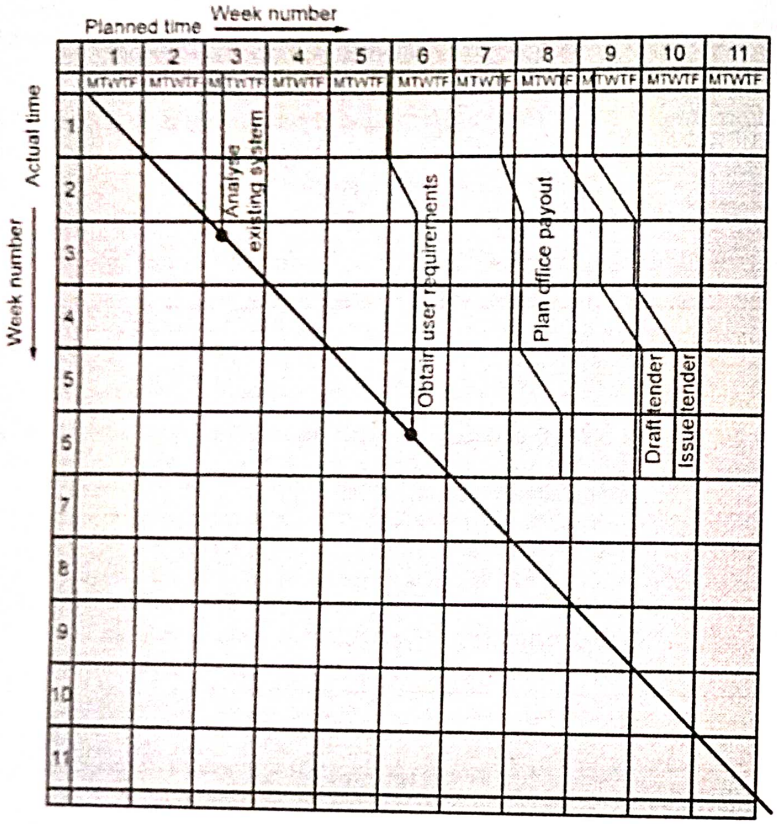


Fig. 4.6.3 : Brigette's timeline chart at the end of week six

- Brigette analyses these target dates at the conclusion of the first week and decides to leave them as they are, thus on the actual time axis, lines are drawn vertically downward from the target dates to the end of week 1.
- Brigette concludes at the end of week two that "get user needs" will not be finished until Tuesday of week six; she consequently diagonally stretches that action line to reflect this. The completion goals for the other activities are similarly postponed.
- By the Tuesday of the third week, "analyze existing system" has been finished and Brigette adds a blob to the diagonal timeline to show it. By the end of the third week, she decides to stick with the current goals.

- She extends "draught tender" and "issue tender" by an additional three days at the conclusion of week 4.
- Take note that two activities were finished before the end of week 6 while three remained unfinished. She has already made three target date revisions and the project as a whole is currently seven days behind schedule.
- The timeline chart is helpful both during project execution and for post-implementation analysis. Analyzing the timeline chart and the causes of the changes can reveal estimation errors or other mistakes that, with this knowledge, could be avoided in the future.

Review Questions

1. What are the different methods used in visualizing progress explain in detail ?
2. What are the benefits of visualizing progress ?
3. What is Gantt chart, slip chart and timeline ? Explain in detail.
4. What are different data collection methods ?
5. What is Red / Amber / Green (RAG) reporting ?

4.7 Visual Project Management

- The ability to effectively communicate with our team on both the small details and the big picture of our project is one of the biggest and most crucial problems in project management. After all, it's doubtful that individuals will work on the proper tasks in the proper order if they are unaware of the importance of each task and its timely completion to the project's success. Visual project management is one of the finest ways to communicate with our team.
- By using tools to help everyone involved visualize the project's needs and progress, visual project management is a technique for organizing and visualizing processes over traditional projects.
- When it comes to information delivery, "visual project management brings up information given in a way that anyone can consume it at a time, place and manner that is convenient to them."

4.8 Kanban Boards

- An agile project management tool called a kanban board was created to better visualise work, reduce work-in-progress and increase productivity (or flow). Both agile and DevOps teams can use it to help them build order in their everyday work.

- The Japanese word for "visual signal" is "kanban." Our labor is frequently imperceptible and intangible whether we work in the service or technology industries. With the aid of a kanban board, we can make our work visible, share it with others and keep everyone informed.

Elements of a kanban board

1. **Visual signal :** The visual cards on a kanban board are one of the first things we'll notice (stickies, tickets or otherwise). All of the tasks and projects that are part of a Kanban team are listed on cards, often one per card. Each card for an agile team might include one user story. These visual cues assist coworkers and stakeholders rapidly comprehend what the team is working on after they are posted on the board.
2. **Columns :** The columns are yet another distinguishing feature of the kanban board. Each column denotes a particular task that together make up a "workflow." Cards move along the workflow until they are finished. Workflows might be very straightforward, such as "To Do," "In Progress," or "Complete," or they can be very complex.
3. **Work in progress limits :** Limits for Work In Progress (WIP) The number of cards that can ever be in a column at once is the WIP limit. There can never be more than three cards in a column with a three-card WIP limit. Before fresh cards may enter that stage of the workflow when the column is "maxed-out," the team must swarm on those cards and move them ahead. These WIP restrictions are essential for identifying workflow bottlenecks and maximising throughput. WIP limitations provide us with a forewarning if we have taken on too much work.
4. **Commitment point :** Board backlogs are common for Kanban teams. Ideas for projects that the team can pick up when ready are posted here by clients and coworkers. The decision to proceed with a project is made at the commitment stage, when the team decides to pursue the idea.
5. **Delivery point :** The workflow of a kanban team ends at the delivery point. When the consumer has the product or service in their hands, it is the delivery point for the majority of teams. The team's objective is to move cards as quickly as possible from the commitment point to the delivery location. Lead Time is the amount of time that has passed between the two. Kanban teams are always refining their processes to cut down on lead times as much as feasible.

4.9 Project Calendars

- The working days and shifts that are open for scheduled activities are listed on a project calendar. It distinguishes between time periods that are available to finish scheduled activities and time periods that are not. Available time periods are expressed as days or portions of days. To account for varying work periods for specific tasks when calculating the project schedule, a schedule model could need more than one project calendar. The project schedules might be revised.

4.10 Cost Monitoring

- In addition to being essential in and of itself, expenditure monitoring is a crucial part of project control since it shows how much work has gone into (or at least been charged to) a project. A project may be finished on schedule, but only because more money than anticipated was allotted for the project's activities. A easy way to compare actual and projected spending is by using a cumulative expense chart like the one in Fig. 4.10.1.

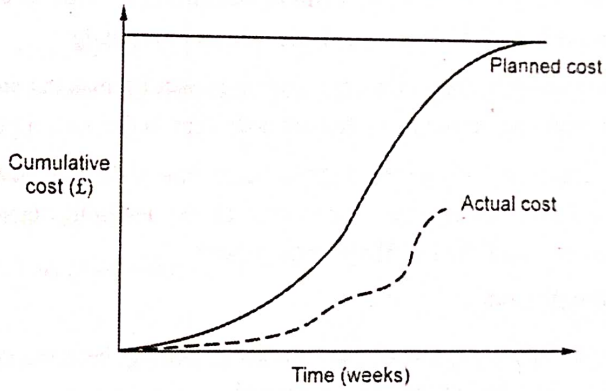


Fig. 4.10.1 : Tracking cumulative expenditure

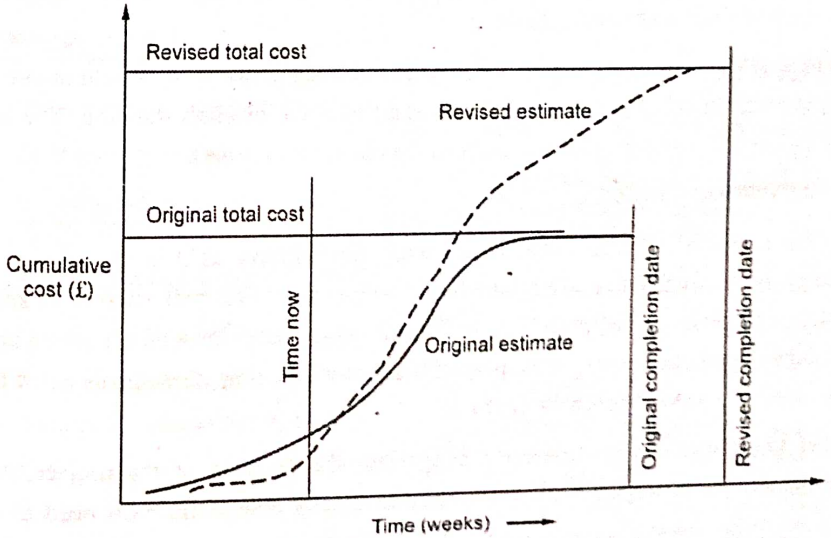


Fig. 4.10.2 : The cumulative expenditure chart can also show revised estimates of cost and complete on date

- Cost charts are significantly more helpful if we include expected future costs, which are computed by multiplying the costs currently incurred by the estimated costs of unfinished work.
- When using a computer-based planning tool, cost schedule revision usually happens automatically after actual spending has been recorded. The revised cost schedule is provided in Fig. 4.10.2, which shows the extra information that becomes accessible. In this example, it is clear that the project is behind schedule and over budget.
- To stay inside the project budget, we'll need precise project estimations and a solid project budget. Estimate costs and determine budget are both steps in the project planning process.
- The project will almost certainly never be delivered on time if the true costs are not kept in mind while it is being implemented. According to the PMBOK, there are numerous approaches for monitoring and controlling project costs :

1. Budgeting for the project

- Making a budget for the project at hand should preferably be done at the start of the planning session. This budget will be used to aid us with all payments and charges that we will incur during the project's life cycle. As a result, creating this budget necessitates extensive research and critical thinking.
- We must always provide room for revisions in our budget, just as we would in any other budget, because costs may not remain constant over the project's duration. The key to making a profit from a project is to stick to the budget at all times.

2. Keeping expenses tracked

- It's just as crucial to keep track of all actual expenditures as it is to use any other strategy. It's preferable to make a time-based budget here. This will assist us in keeping track of a project's budget at each phase. Actual expenditures must be compared against the budget's periodic targets. If the project will last a long time, these goals could be set on a monthly, weekly or even yearly basis.
- Rather to having a single complete budget for the duration of the project, this is considerably easier to deal with. If we need to do any new work, we'll need to make some estimates to see if it can be done within our budget.

3. Time management effortlessly

- Time management is another useful strategy. Although this strategy can be used in a variety of management situations, it is particularly useful in project cost management.

- The reason for this is that if we don't meet our project deadlines, the cost of our project will continue to rise; the longer the project drags on, the greater the costs will be, effectively resulting in a budget overrun.
- In order to ensure that work is finished on time, the project manager would need to remind his or her team of the project's key deadlines on a regular basis.

4. Project change management

- Another important strategy is project change control. Change control systems are required to account for any potential changes that may occur over the project's duration.
- This is because each modification to the project's scope will have an influence on the deliverables' dates, therefore the adjustments may raise project costs by increasing the amount of effort required.

5. Earned value applicability

- Similarly, using the accounting technique known as 'Earned Value' to determine the value of the work that has been completed thus far is quite beneficial.
- This is especially beneficial for large projects since it allows us to make any last - minute changes that are critical to the project's success.

Review Questions

1. Write a note on cost monitoring.
2. Define the following terms :
   a. Visual project management b. Kanban boards c. Project calendars

4.11 Project Cost Management

- The methods listed below for project cost management are summarized in Fig. 4.11.1.

1. **Plan cost management** is the process of establishing the guidelines, instructions and records for budgeting, managing and controlling project costs.
2. **Estimate costs :** Develop an approximate estimate of the financial resources required to accomplish project operations.
3. **Determine budget :** To produce an authorized cost baseline, the anticipated expenses of several activities or work packages are combined.
4. **Control costs** is the process of keeping track of the project's progress in order to update project costs and handle adjustments to the cost baseline.

- These operations communicate with one another and with operations in other knowledge areas.

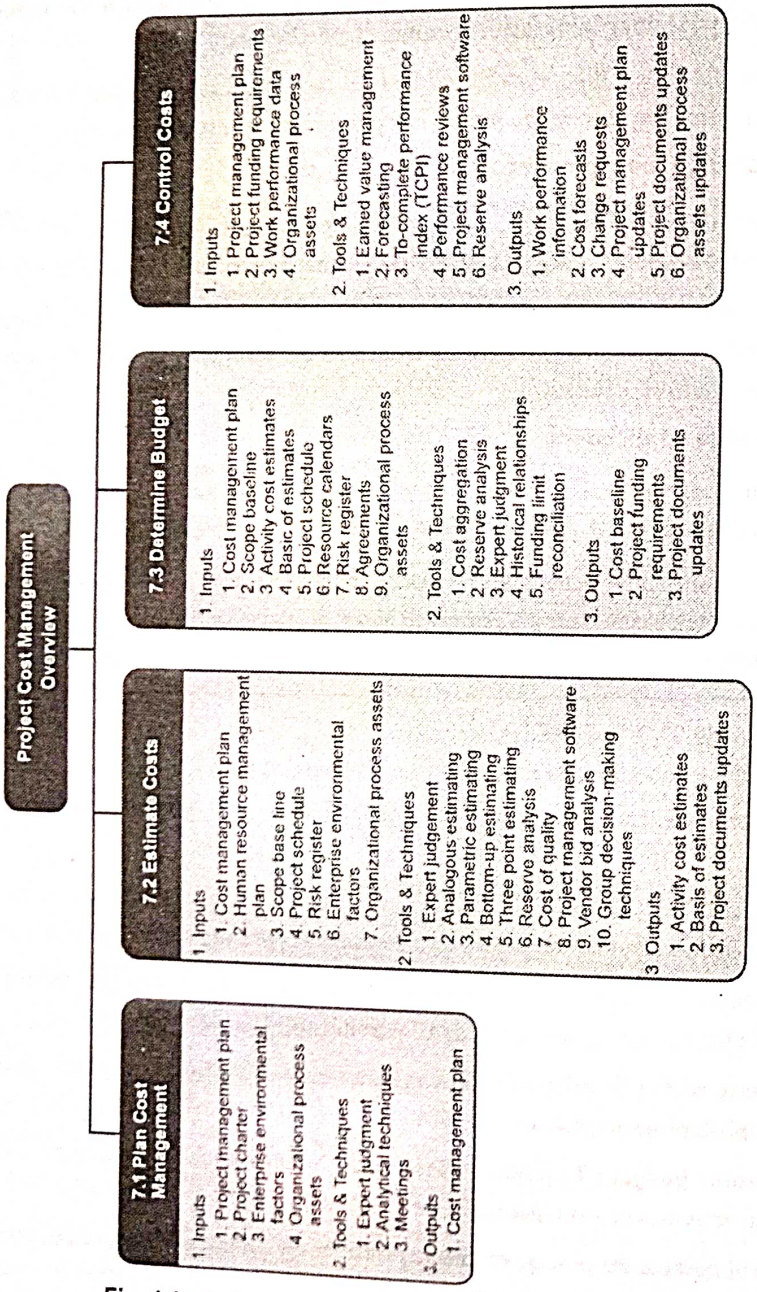


Fig. 4.11.1 : Project cost management overview

Plan cost management :

- The process of setting up the rules, guidelines and documentation for budgeting, managing and controlling project expenses is known as plan cost management. This process' main advantage is that it offers direction and guidance on how the project costs will be handled throughout the project. Fig. 4.11.2 shows the inputs, tools and procedures and outputs of this process. The process's data flow flowchart is shown in Fig. 4.11.3.
- The cost management plan contains a description of the cost management procedures as well as the tools and methods that go along with it. A part of the project management plan is the cost management plan.

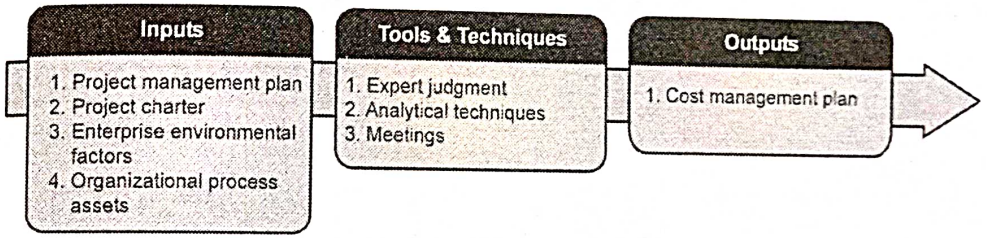


Fig. 4.11.2 : Plan cost management : Inputs, tools and techniques and outputs

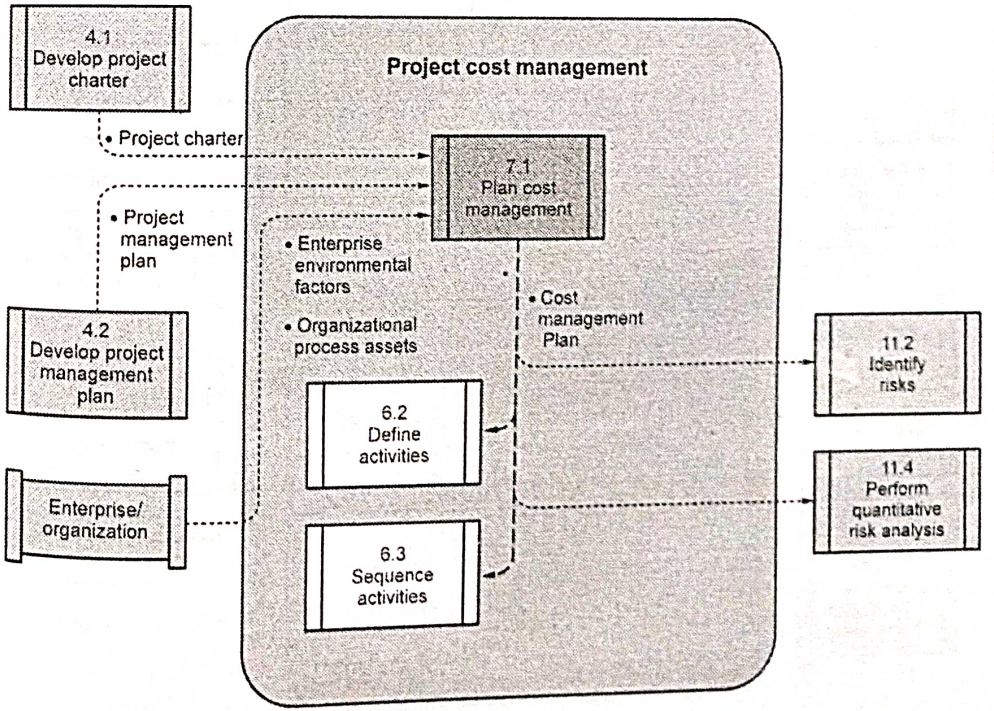


Fig. 4.11.3 : Plan cost management : Data flow diagram

Estimate costs :

- Estimating an approximate amount of the financial resources required to accomplish project operations is the process of estimating costs. The main advantage of this procedure is that it establishes the cost necessary to accomplish the project's work. Fig. 4.11.4 shows the inputs, tools and procedures and outputs of this process. The process's data flow diagram is shown in Fig. 4.11.5.

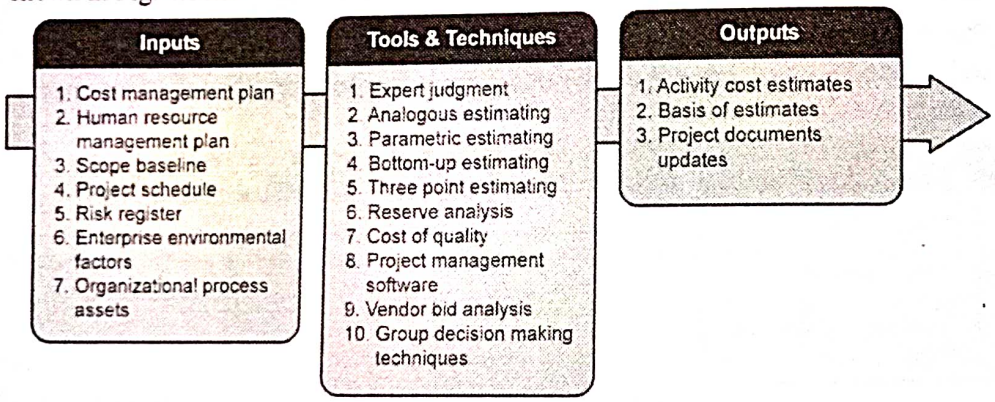
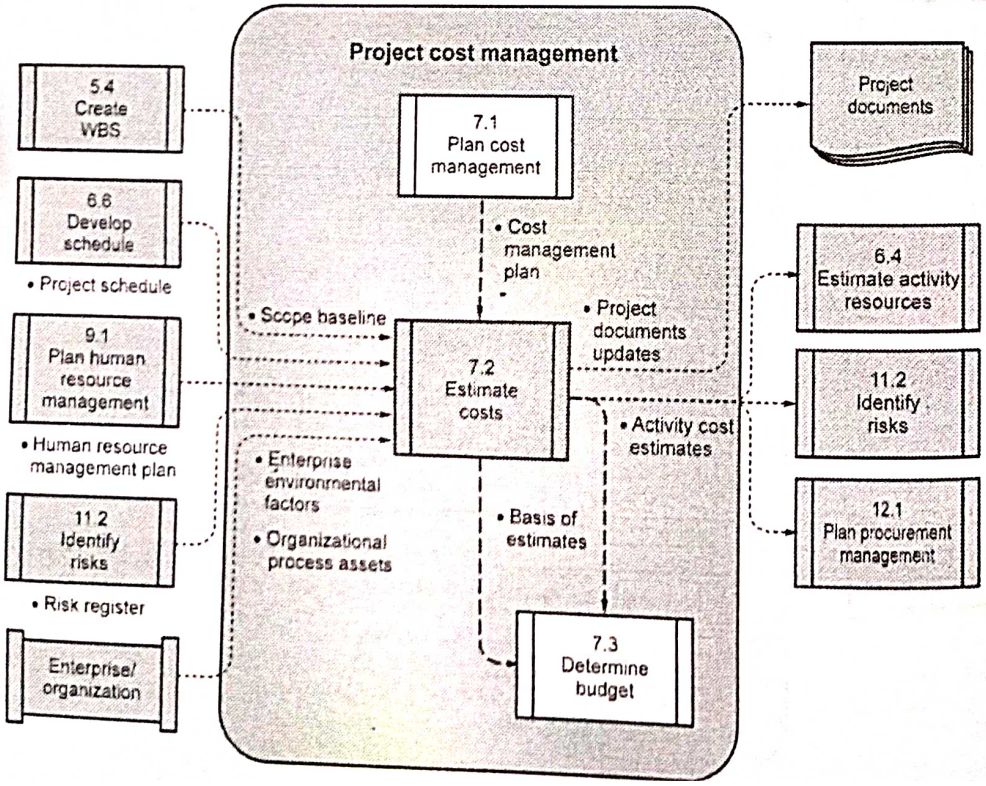


Fig. 4.11.4 : Estimate costs : Inputs, tools and techniques and outputs



- Cost estimates are a prediction based on the facts available at the time. Estimated costs involve the selection and evaluation of costing options for starting and finishing the project. To get the best prices for the project, cost trade-offs and hazards like make vs. purchase, buy vs. lease and resource sharing should be taken into account.
- Cost estimates are typically represented in units of some currency (e.g., dollars, euros, yen, etc.), but in some cases, other units of measure are used to assist comparisons by removing the impacts of currency fluctuations, such as staff hours or staff days.
- Cost estimates should be evaluated and adjusted during the project to take into account new information as it becomes available and hypotheses are tested. As a project moves through the project life cycle, its estimation accuracy will rise.

Determine budget :

- To develop an authorized cost baseline, the process of determining budget involves adding up the expected costs of several work packages or individual activities. This procedure's main advantage is that it establishes the cost baseline by which project performance can be tracked and managed. Fig. 4.11.6 shows the inputs, tools and procedures and outputs of this process. The process's data flow diagram is shown in Fig. 4.11.7.
- All the funds authorized to carry out the project are included in the project budget. The time-phased project budget is the version of the cost baseline that has been approved, but management reserves are not included.

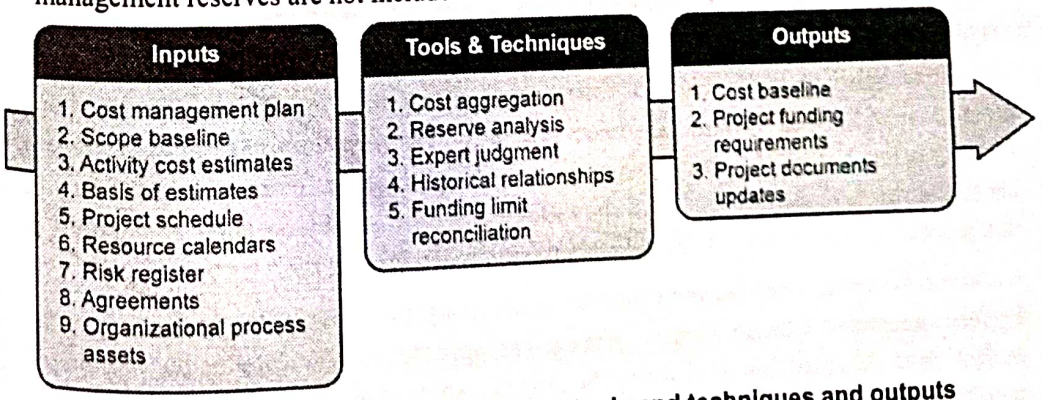


Fig. 4.11.6 : Determine budget : Inputs, tools and techniques and outputs

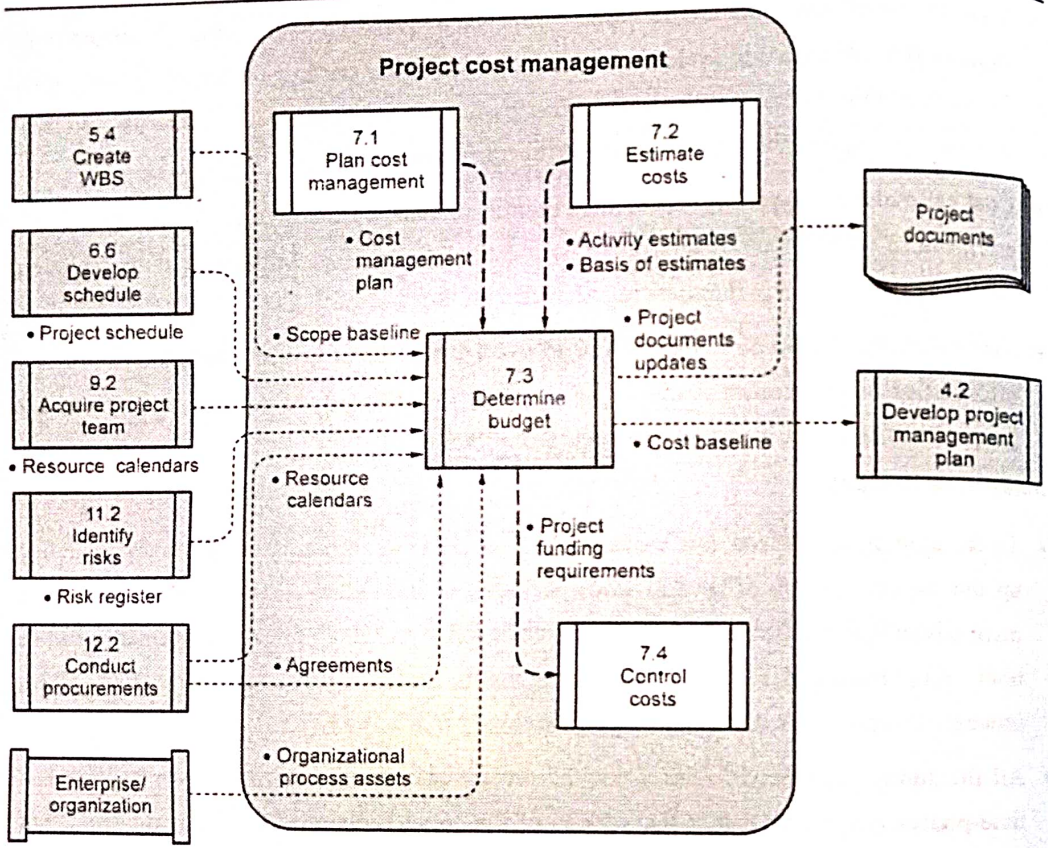


Fig. 4.11.7 : Determine budget data flow diagram

Control costs :

- In order to update the project costs and manage changes to the cost baseline, control costs is the process of keeping track of the project's progress. The main advantage of this approach is that it offers a way to identify deviations from the plan so that corrective action may be taken and risk is reduced. Fig. 4.11.8 shows the inputs, tools and procedures and outputs of this process. The process's data flow diagram is shown in Fig. 4.11.9.
- Knowing the actual costs incurred thus far is necessary for updating the budget. Only the Perform Integrated Change Control process can approve any increase to the authorized budget. With the exception of enabling the project team to stay within the approved money, monitoring the use of funds without consideration to the worth of work completed has little value to the project. Analysis of the connection between the financial resources used for a project's expenditures and the actual work being done for those expenditures constitutes a significant portion of cost control efforts. The administration of the agreed cost baseline and adjustments to that baseline is the key to efficient cost control.

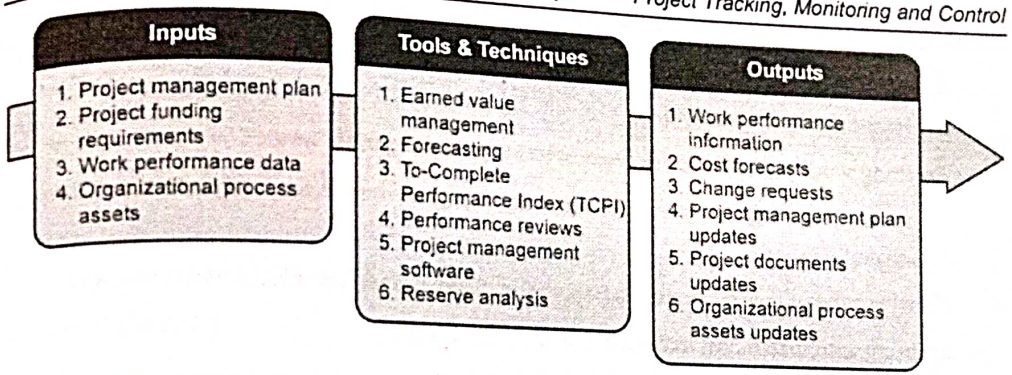


Fig. 4.11.8 : Control Costs : Inputs, tools and techniques and outputs

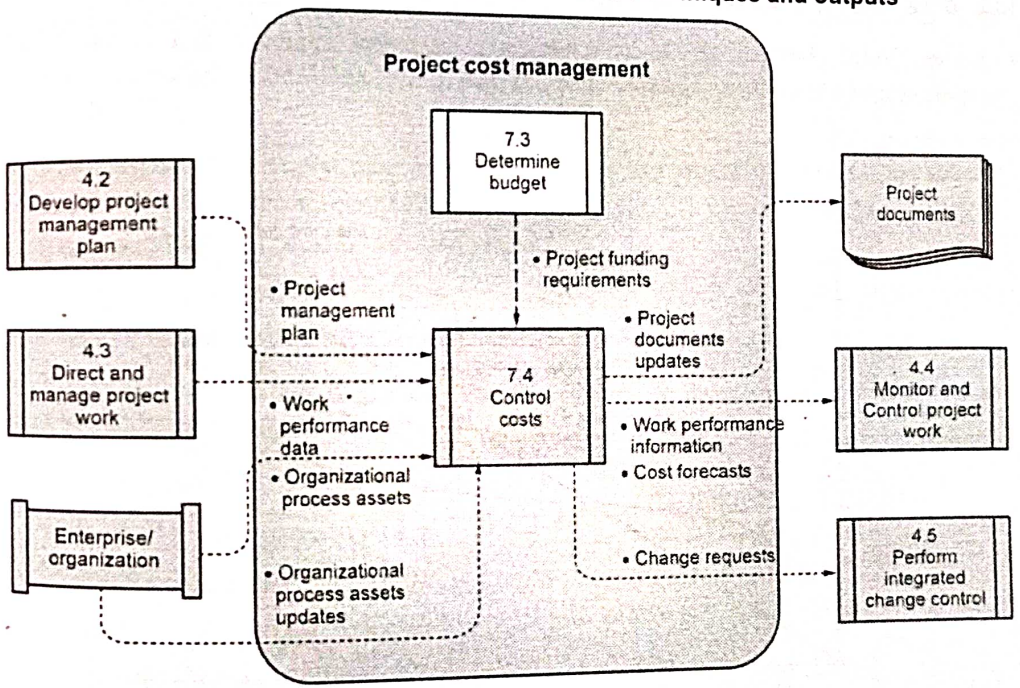


Fig. 4.11.9 : Control costs data flow diagram

- Project cost control includes :
  - Influencing the variables that result in modifications to the baseline of permitted costs;
  - Making sure that all change requests are handled quickly;
  - Controlling the real changes as they happen;
  - Making sure that cost expenditures for the project do not go over the allocated money by period, by WBS component, by activity and overall;

- Keeping track of cost performance in order to identify and comprehend deviations from the established cost baseline;
- Tracking work performance in relation to budgetary expenditures;
- Preventing unauthorized modifications from being accounted for in reported resource or expense utilization;
- Notifying the proper parties of any approved changes and their associated costs; and
- Limiting anticipated cost overruns to acceptable levels.

4.12 Earned Value Analysis

- Earned Value Analysis (EVA) is a project management approach that is used for monitoring and controlling purposes, shown in the Fig. 4.12.1.

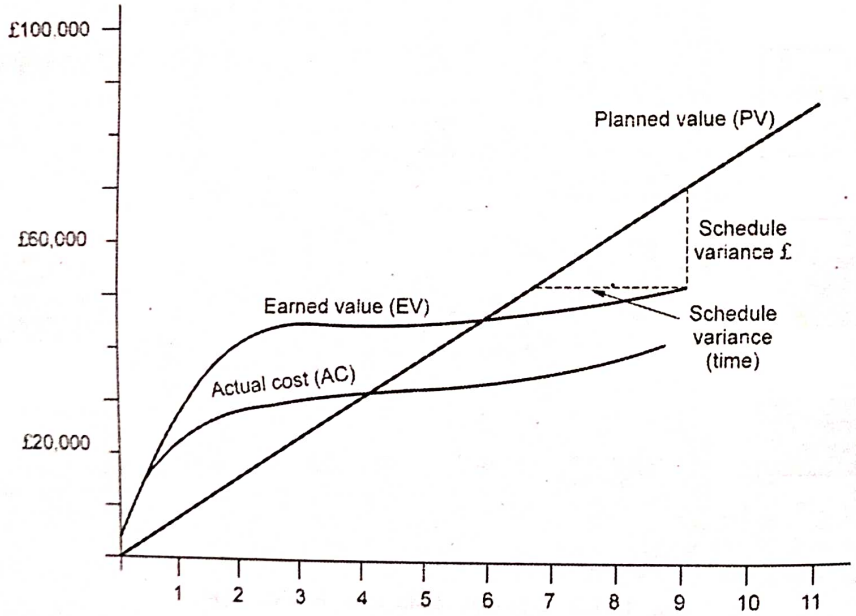


Fig. 4.12.1 : Earned value analysis

- The Earned Value Analysis comprises the indicators
  - Earned Value (EV),
  - Planned Value (PV),
  - Actual Cost (AC) and
  - Budget at Completion (BAC)

Earned Value(EV)

- The projected value is the percentage of the budget (excluding management reserve) allotted to the activities or periods under consideration.

Planned Value (PV)

- This metric represents the project's progress. It's worth is equal to the total of the budgeted and permitted funds for the work that has already been done.

Actual Cost (AC)

- The AC stands for the amount paid for the work done.

Budget At Completion (BAC)

- The total budget is the allowed value (plus the sum of expected costs) of a project's scope at the end.
- The purpose of an earned value analysis is to assist and support the control cost process. Earned Value Management (EVM) uses the outcomes of this study to analyse variations, trends and forecasts based on the EVA results.
- Earned value analysis is a suggested approach of the following procedures for projects adopting the PMI methodology :
- Work on the project should be monitored and controlled.
  - Schedule of control,
  - Cost-cutting and
  - Control the procurement process.

Ex. Earned Value

Task

Specify module 5 days

Code module 8 days

Test module 6 days

At the beginning of day 20, Planned Value (PV) = 19 days

If everything but testing completed, EV = 13 days

Schedule variance = EV – PV i.e. 13 – 19 = – 6

Schedule performance indicator (SPI) = EV/PV

13/19 = 0.68

Actual Cost (AC) is also known as actual cost of work performed

In previous example, if

'Specify module' actually took 3 days (planned 5)

'Code module' actually took 4 days (planned 8)

Actual cost = 7 days

Cost Variance (CV) = EV · AC i.e. 13 · 7 = 6 days

Cost Performance Indicator (CPI) = EV/AC i.e. 13 / 7 = 1.86

Positive CV or CPI > 1.00 means project under budget or the work is completed better than planned.

Common method of assigning Earned Value (EV)

0 / 100 technique

- A task is assigned a value zero until it is completed On completion its value will be 100 of the budgeted value.

50 / 50 technique

- At the starting 50 of the budgeted value Upon completion 100 % (remaining 50 %) of the budgeted value.

75 / 25 technique

- At the starting 75 of the budgeted value Upon completion 25 of the budgeted value.

The milestone technique

- Value is given based on the achievement of the milestones.

Percentage complete

- Value will be assigned based on the objective measurement of the work completion 0 / 100 technique is preferred for software development.

4.12.1 The Baseline Budget

- The creation of the baseline budget is the first step in setting up an earned value analysis. The project plan serves as the foundation for the baseline budget, which depicts the expected rise in earned value over time. Earned value may be quantified in monetary terms, although it is more typical to assess earned value in person-hours or workdays for staff-intensive projects like software development. Amanda's baseline budget shown in the Fig. 4.12.2.

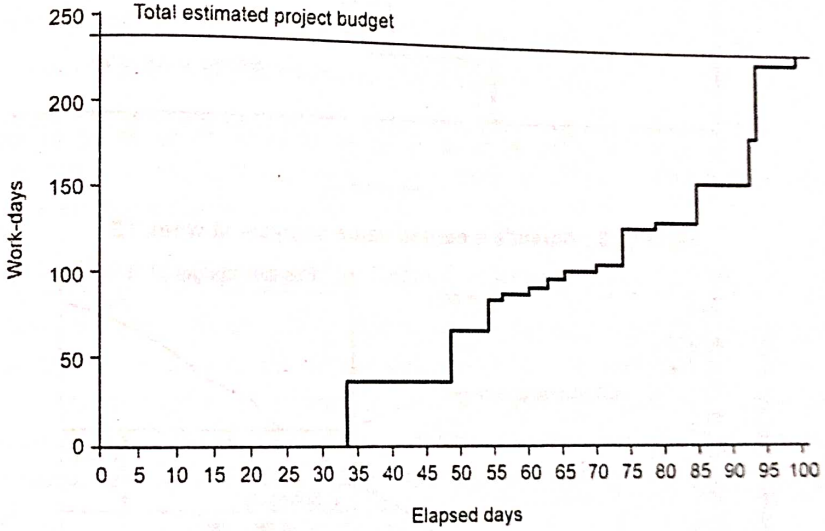


Fig. 4.12.2 : Amanda's baseline budget

4.12.2 Monitoring Earned Value

- The next step is to monitor earned value as the project develops once the baseline budget has been created. This is achieved by keeping track of how activities are completed (or activity starts and milestone achievements in the case of the other crediting techniques).
- In addition to recording EV, each task's real cost can also be gathered (AC). This is sometimes referred to as the actual cost of the work done (ACWP). Fig. 4.12.2, which in this instance records the data as percentages of the entire budgeted cost, illustrates this.
- The following performance metrics, which can be displayed directly or inferred from the earned value chart, are also shown in Fig. 4.12.4.

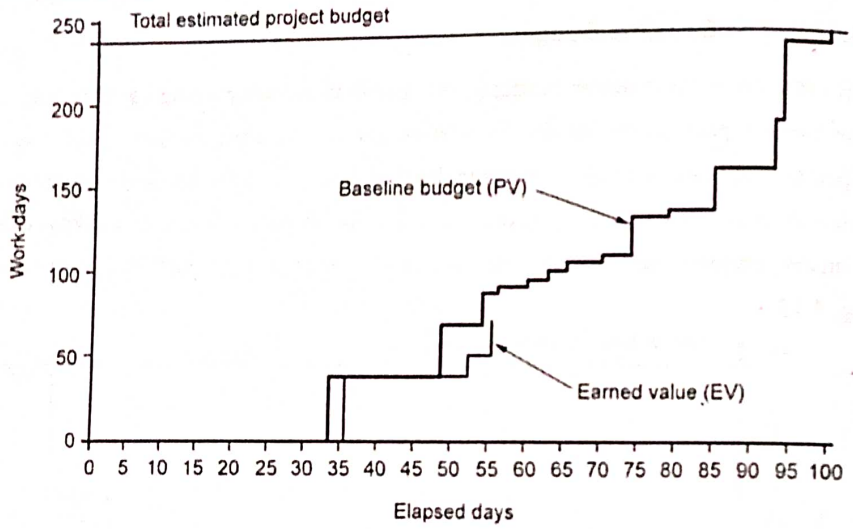


Fig. 4.12.3 : Amanda's earned value analysis at week 12

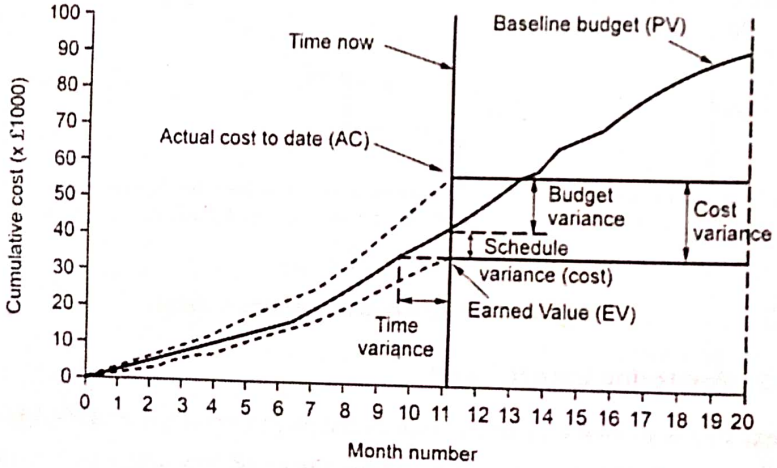


Fig. 4.12.4 : An earned value tracking chart

Review Questions

1. Explain the earned value assigning methods.
2. What is earned value ?
3. Explain in detail earned value analysis.
4. Write a note on : Project cost management.

4.13 Project Tracking

- To ensure that the project runs smoothly, create a reporting system that keeps us up to date on the many variables that explain how the project is progressing in comparison to the original plan.
- The following are characteristics of a reporting system :
  - Provides timely, complete and accurate status information;
  - Does not add enough overhead time to be inefficient.
  - Is easily accepted by the project team and top management.
  - Warns of impending problems in time for action to be taken.
  - Those who have a need to know will find it simple to comprehend.
- The majority of project management software packages allow us to tailor basic reports to match even the most precise requirements.

4.14 Effective Approach to Track Projects

- Tracking the progress of each project is one of the most crucial facets of a project manager's job. It aids in goal-setting and attention among staff members and it keeps project managers informed of workplace activities. Helping people stay on task, keeping team members interested and holding everyone accountable are the first steps of good leadership. Here are only a few efficient methods for project managers to monitor project progress.

1. **Create a project outline :** A effective technique to track project progress is to work with team members to establish a project overview. Each team member can contribute ideas for creating achievable project goals and can also learn what is expected of them both individually and collectively. A excellent first step is to take the time to meet down with employees and develop a thorough outline that includes team goals, milestones and Key Project Indicators (KPIs).
2. **Establish goals and milestones :** It's wise to start by taking into account a worker's particular skill set when deciding how to monitor a project's development. From one individual to the next and from one project to the next, measuring progress might seem very different. The better a project manager understands each employee's abilities and limitations, the greater understanding of how to keep them accountable he or she will have. Team happiness is also greatly influenced by the establishment of specific goals and milestones for each team member. It's critical to emphasise each team member's function and how it will advance the group's success. Even when achieving more modest

objectives and making incremental strides toward them, it's crucial to always keep the overall picture in mind.

3. **Check in regularly :** Never underestimate the importance of checking in when deciding how to monitor a project's progress. Establishing trust between project managers and employees through a brief, informal talk at the start of each work session can assist to highlight the crucial distinction between employees feeling cared for and checked up on and employees feeling criticised or checked up on. The smooth operation of a project is always dependent on effective communication. Instead than continuing to construct unmotivating or unhelpful frameworks, it is preferable to check in with the employee and attempt to determine why they are struggling to meet goals or complete time-sensitive activities.
4. **Establish clear deadlines :** Always keep the big picture in mind when tracking our progress. Workers can stay on task and accomplish tasks without becoming unduly stressed or overwhelmed if deadlines are made clear to them. Once everyone is aware of the project's timeframe, monitoring project progress becomes much simpler. While some project managers like to set a date for each milestone or goal, others prefer to work with a single end deadline. Whatever the selected approach, keeping track of each project should be a simple matter of adhering to the plan as long as each deadline is clearly defined and team members are aware of what they are working towards.

4.15 Status Report

Types of Project Status Reports

- The following are the five different types of project status reports :

1. **Reports for the current period :** These reports only cover the most recently completed period.

   They provide updates on activities that were open or slated for work throughout the time period. Reports may emphasize completed tasks as well as the difference between expected and actual completion dates.
2. **Cumulative reports :** These reports detail the project's progress from the start to the end of the reporting period. Because they illustrate trends in project progress, they are more informative than current period reports.
3. **Exception reports :** Exception reports detail deviations from the original plan. Typically, these reports are intended for senior management.

4. **Stoplight reports :** Stoplight reports are a version of the previous report kinds that can be used on any of them. On the top right of the first page of the project progress report, stickers of various colors are placed. The different colors of stickers serve the following purposes :

   a) **Green sticker :** It indicates that the project is on track and that everything appears to be going according to plan. Senior managers will see this sticker and know that everything is going according to plan and they won't even need to read the linked report.

   b) **Yellow sticker :** This indicates that the project has met a problem or has slipped behind schedule. That's a signal to upper management that the project isn't progressing as planned, but we've got a back-up plan in place.

   c) **Red stickers :** This indicate that a project has gotten out of hand. Red reports should be avoided at all costs because they indicate that a problem has arisen on the project and we don't have a fix in place or even a suggestion for top management.
5. **Reports on variance :** Variance reports reflect discrepancies between what was intended and what really occurred. Typical variance reports are snapshots of the status of an entity being tracked in time (the current period). The majority of variance reports leave out information on how the project got to that point.

4.15.1 How and What Information to Update

- As input to each of these report types, activity managers and the project manager must report the progress made on all of those activities that were open for work during the period of time covered by the status report. Recall that our planning estimates of activity duration and cost were based on little or no information. Now that we have completed some work on the activity, we should be able to provide a better estimate of the duration and cost exposure. This reflects itself in a re-estimate of the work remaining to complete the activity. That update information should also be provided.
- The following is a list of what should be reported in actuality :

1. Determine a certain time and day of the week when all updated information will be available is expected to be submitted.
2. Actual work completed within this time period should be reported.
3. Keep track of the past and recalculate the amount of time left (in-progress work only).
4. Actual start and end dates of activities that began or ended within the reporting period.

5. As evidenced in the record days of length accomplished of so far, working re-estimated duration in the amount of days to completion.
6. Report on the amount of time and effort spent (hours per day) and the amount of time left (in-progress work only).

4.15.2 Variances

- Deviations from the plan are known as variances. Consider a variance to be the difference between what was expected and what happened. Variations can be divided into two categories :

Positive variances

- Positive variances are variations from the plan that show an event that occurred ahead of time or that an actual cost was lower than anticipated. The project manager will be pleased with this type of variation.

Negative variances

- Negative variances are variations from the plan that indicate a scenario that is behind schedule or that an actual cost is higher than the projected cost. Negative variances aren't always bad news, just like positive variances aren't always good news. For example, we may have overspent because we completed more work than anticipated within the report period. However, by overpaying during this time, we could have completed the work for less money than we had anticipated. By glancing at the variance report, we won't be able to know.

4.16 Four Features of Good Status Report

- A project's status report will typically include the following :
  - The task that was finished
  - The strategy for what's coming
  - The project's budget and schedule in summary
  - A list of actions to do
  - Any issues or risks and what steps are being taken to address them
- A project status report's genuine significance extends beyond its function as a channel of communication. It also offers a written account of the project's past. This provides us with historical information so we can design a project in the future without making the same mistakes or running into the same problems.

- Historically, project status reports took a lot of time to write since they cover so many different areas. Thankfully, cutting-edge project management tools like Project Manager speed up the crucial reporting process.

Review Questions

1. Explain in detail variance in terms of project control.
2. List and explain different types of project status reports.
3. What is project tracking ? What are the effective approaches to track the project ? Explain in detail.

4.17 Change Control

- Changes are unavoidable when developing and maintaining a product. People make mistakes, consumers require adjustments and the product's operating environment changes. People also continue to improve their understanding of the problem and their abilities to solve it. It's a common adage in software development that solving one problem will lead to the creation of additional ones.
- The goal of change control is to have complete control over all product change requests and completed modifications. Any configuration item must be able to be distinguished from its predecessor in terms of changes.

Inputs

- An event triggers the control of change. A longing for change might be referred to as an event, yet It isn't necessary to express it as a specific wish. An event is any observation made in this context. Something unexpected, uncomfortable or outright wrong occurs while using the thing to be configured For example, it may be :
  - During the inspection of a document, a misspelling was discovered.
  - During a walk-through of some source code, a coding mistake was discovered.
  - During the course of the project, the customer comes up with a new idea, which leads to an enhancement request.
  - An error was discovered during the integration test.
  - A desire to expand or improve the finished product after it has been put into service.
  - A request to a helpdesk for assistance with a problem relating to the use of a system.
  - An upgrade to a new version of the middleware necessitated a change in the code.

- Supporting a system that may or may not be backwards compatible.

Outputs

- Documented events and change requests derived from these are the result of change control.
- Events. Both should be kept in a secure database so that relationships between them can be tracked.
- Change requests and configuration items can be kept up to date in a reliable manner.
- Change requests may be referred to configuration management, but this is uncommon.
- Where configuration management must be done in a structured manner.

4.17.1 Change Control Procedures

- The following steps could make up a straightforward change control approach for operational systems :

1. One or more users may feel that a system needs to be changed and request that a change request be forwarded to the development team.
2. The change request would be evaluated by the user management and if approved, it would be forwarded to the development management. Requests for Change (RFCs) between the client community and the management of the developers must go through a single authorized route. Before the RFC is generated, there would be some filtering among the client community to make sure the proposed change does, in fact, offer a benefit.
3. One individual would be responsible for receiving and processing RFCs in the development area. They would assign a staff member to examine the request and report on the cost and practicality of implementing the modification. As part of this, the developer would evaluate the goods that might be impacted by the modification.
4. The development representative would then present the findings to the user management, who would then determine whether to proceed given the cost estimate.
5. The RFCs would be prioritized for action by someone or some group who would represent the key stakeholders, including users, developers and the project sponsor.

   The Project Board or an equivalent should be responsible for this in the end. However, a sizable number of RFCs would have a restricted scope. If all of these changes needed to be approved at the highest level, this may result in a bureaucratic backlog. Therefore, it can be decided to provide a smaller group of engaged stakeholder representatives the

authority to review and approve changes up to a specific amount of spending. Even though they might not be termed that, this group would assume the job of a change control board. Giving the project managers permissions to accept modest adjustments (as long as they are documented with an RFC, etc.) as long as they do not go over budget and delivery deadlines is a further step. Thus, it may be said as a general rule that the further up in the control hierarchy an amendment must be submitted, the greater it is. But it goes beyond just a size issue.

Very many modest adjustments made repeatedly could have a significant cumulative impact on project development and may require the attention of higher management. The project manager might create an exception report if there are a lot of modifications.

6. One or more developers are permitted to make copies of the master products that will be updated once an RFC has been given the go-ahead to be put into effect. To accomplish this, use the configuration librarian.
7. The copies have been changed. If the components were software, this would entail changing the code, recompiling it and testing it.
8. After the user management has been notified that the development of new versions of the product is complete, copies of the program will be made available for user acceptance testing.
9. The user will allow the products' operational release if they are satisfied that they are adequate. The configuration items' master copies will be changed.

4.17.2 Changes in Scope of a System

- With IS development initiatives, it's typical for the system size to steadily grow. User-requested adjustments to requirements are one reason for this.
- A project's scope needs to be closely watched and managed. Recalculating the system size in terms of SLOC or function points at important milestones is one way.

4.17.3 Change Control Activities

- A change management process is a mini-development project in and of itself. A written and controlled life cycle for an event registration should broadly follow the phases. Each phase should be outlined in depth, including the company's responsibilities and particular activities shown in the Fig. 4.17.1.

- Depending on the kind of events to be handled, a corporation may need to specify many types of life cycles.

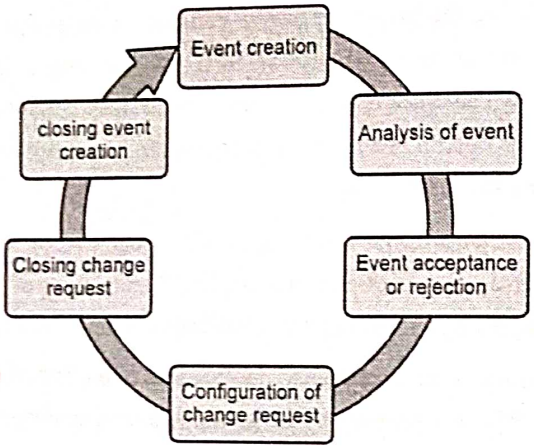


Fig. 4.17.1 : Phases of change control activities

- **Event creation :** The registration page for the event has been developed, as well as a description for the event.
- **Event analysis :** The configuration item(s) that may be affected by potential changes are identified and the scope of these changes is calculated.
- **Event acceptance or rejection :** If the event registration is approved, a change request for each configuration item that is affected is created.
- **Configuration of change request :** The update is implemented after a new configuration item is recognized and generated. Feedback is supplied to the configuration control board during the acceptance and storage of the new item.
- **Change request closing :** After the change has been implemented and accepted, the request can be closed.
- **Event creation closing :** When all change requests are completed, the event registration can be closed.

Review Question

1. Explain in detail change control process.

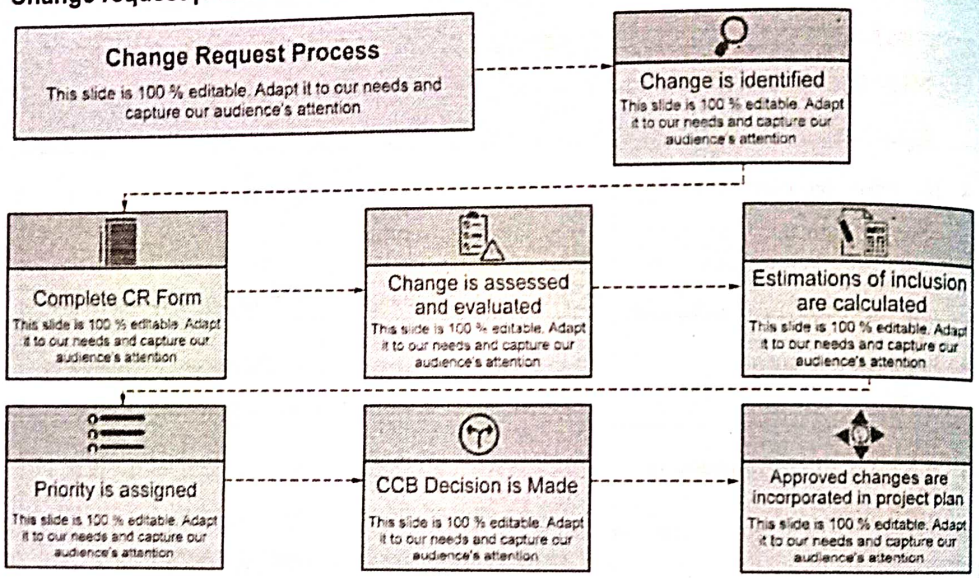
4.18 Different Factors of Change Control Process

- There are several factors that a change control process should take into account.

Steps in change control process	Action taken in change control
• Initiation and control of change requests	• Change requests should be uniform and subject to management scrutiny. • Request for a change or information to be continued
• Impact assessment	• Ensure that all proposals for change are evaluated in a methodical manner to identify potential effects.
• Control and Documentation of changes	• A change log should be kept that includes the date, the identity of the person who made the modifications and the changes that were made. • Only those who are authorized should be able to make modifications. • There should be a method for reverting to an earlier version.
• Documentation and Procedures	• The processes and related paperwork should be updated whenever system modifications are made.
• Authorized maintenance	• Controlling system access rights is necessary to prevent unwanted access.
• Testing and User signoff	• Software should undergo rigorous testing.
• Version control	• Production source code should be under control to ensure that only the most recent version is updated.
• Emergency changes	• The adjustment should be reported as soon as feasible after obtaining verbal approval.

4.19 Change Process Flow Diagram

Change request process flowchart with project plan



Review Question

1. What are the factors of change control process ? Explain change process flow diagram.

4.20 Software Configuration Management

- SCM is concerned with tracking and controlling software changes. Because each work product must be accessed and modified by multiple people during the software development process, a robust configuration management system is essential to avoid several difficulties.
- Problems that may occur if a proper SCM is not used
  - Concurrent access
  - Undoing changes
  - System accounting
  - Handling variance
  - Accurate determination of project status
  - Preventing unauthorized access to the work products

Configuration Management Process

- Configuration management is carried out using the two main tasks listed below.
  - **Configuration identification :** The process of determining which components of the system should be retained under configuration management is known as configuration identification.
  - **Configuration control :** The term "configuration control" refers to the process of ensuring that changes to a system go smoothly.

Changes to a work product that are under configuration control

- When a developer needs to make a change to a work product, he or she submits a reserve request.
  - After the reserved instructions have been executed successfully.
  - In the local environment, a private copy of the work product is created.
- Along with the modifications to the directory.
- The configuration must be restored once the changes were performed.
  - Depository for management information.
  - Permission is required to restore the modified work product.
- A Change Control Board is a group of people who are in charge of making changes to a system (CCB).
- The CCB is usually made up of members of the development team members.

4.20.1 Context in Which Configuration Management is Necessary

- The work products are altered as development activities are carried out during the development phase.
- The work products vary during the maintenance phase as a result of numerous improvements and adaptations, such as bug fixes. Thus, both throughout the development phase and the maintenance phase, the state of the work products changes continuously. The configuration of the software product refers to the status of all work products at any given time. Software configuration management focuses on efficiently monitoring and managing a software product's configuration over its entire life cycle. A configuration management tool must be deployed for configuration management to be effective.

- As a result, we can state that a tool is used in a project to implement the various configuration management concepts. There are various tools for configuration management; some are open source software that don't require any licensing costs, while others are paid products.
- Version control and baseline creation are examples of configuration management procedures. Terms like version, revision, variant and baseline must be understood in full before we can talk about configuration management.

4.20.2 Purpose of Software Configuration Management

- It is crucial to properly manage the configuration of the work items in a project for a number of reasons.
- The following are some of the serious issues that could arise from not using a suitable configuration management system.
  - **Problems Associated with Concurrent Access :** Controlling access to many deliverable items is probably the most crucial function of configuration management. If rigorous rules are not maintained regarding the updating and archiving of various work items, a number of issues may arise. Let's imagine that numerous developers are working on a program module that is only kept in a single copy. When making modifications to the various components of the same work product at the same time, two developers may accidentally overwrite each other's changes.
  - Rolling back changes : It becomes simple to reverse some aspects of a revision or even go back in time to a specific version of development. A change is exceedingly challenging to reverse without a good configuration management system in place.
  - System accounting system refers to documenting who made specific configuration item changes, what changes were made and when those changes were made. Understanding why changes were made, determining whether some changes are redundant and evaluating the effectiveness of various versions will all be aided by knowing the what, who and when of changes. In some cases, it could be necessary to go back to an earlier baseline if a modification was made in an unjustified or improper manner. Users may want to contrast some software versions from today with versions from yesterday or last year. This becomes a straightforward task since a configuration management system maintains track of each version and revision.

  - Handling with variants : As we've already mentioned, it's frequently required to produce versions. This makes it difficult to maintain track of all variants, their versions and revisions without a configuration management system. Additionally, the availability of software product varieties leads to some odd issues. Let's say we have different versions of the same module and we discover a bug in one of them. After then, all versions and revisions must be rectified. We shouldn't have to make the patch in each and every version and revision of the software independently in order to do it effectively. When one application is changed, it should be reflected in all pertinent versions and revisions.
  - Accurate project status determination : A project manager typically uses a configuration management tool to carry out the configuration management activity. A configuration management tool also assists in tracking numerous deliverable objects so that the project manager can quickly and clearly ascertain the project's present status. The developer can control changes to the various components thanks to the configuration management tool.
  - Protecting work products from unauthorized access : Implementing a regulated modification process is made easier by configuration management. As a result, it is now possible to stop illegal alterations from being made to the work products.

4.20.3 Configuration Management Process

- The following two primary actions constitute configuration management :
  - Determine which components of the system should remain under configuration management by doing configuration identification.
  - Configuration control is a process used to guarantee that system modifications go smoothly.
- We give a summary of these two actions in the section that follows.
  - Identification of configuration project managers typically categorize the outputs of a software development process.
  - Regulated, pre-controlled and uncontrolled, which are the three main groups. Work products that are subject to configuration control are known as controlled work products. To change these, the team members must adhere to some specific protocols. Although pre-controlled work products are not currently under configuration control, they will be in the future. Work products that are not under control will not be subject to configuration control. Both controlled and pre-controlled work items fall within the category of controllable work products.

- Typical examples of controllable work products are as follows :
  - Document defining the requirements
  - Design documents
  - The system's construction tools, including compilers, linkers, lexical analyzers, parsers, etc.
  - Each module's source code
  - Test cases
  - Problem reports
  - Configuration control
- The component of a configuration management system that most directly impacts developers' daily activities is configuration control. Only approved changes to the controlled objects are permitted and unauthorized changes are prevented, thanks to configuration control. Some team members may be given authorization by the project manager to access or make changes to particular work deliverables.
- A developer can obtain a private copy of a regulated work product, such as a code module, using a reserve action in order to modify it (See Fig. 4.20.1). Only one team member may reserve a module at any given moment thanks to configuration management tools. Once a work product has been reserved, no one else is able to reserve it again until the reserved module has been restored. Thus, the issues with concurrent access are resolved by preventing multiple developers from reserving a module at once.

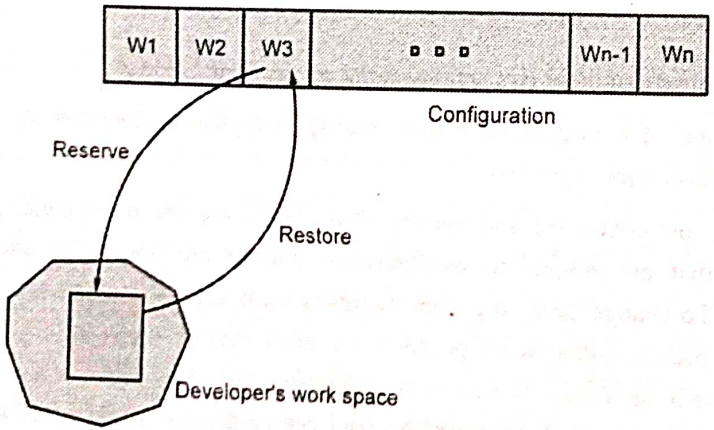


Fig. 4.20.1 : Work product modifications under configuration management

4.21 Tasks in SCM Process

- **The Tasks in the SCM Process** course includes :
  - Configuration identification
  - Change control
  - Version control
  - Configuration auditing
  - Reporting
- The core of SCM is configuration identification. It lays the groundwork for the SCM process's subsequent tasks. Software Configuration Items (SCIs) are identified through a process known as configuration identification. This process also involves defining the basis for identifying SCIs, the method to depict the relationship between SCIs, the identification scheme for naming SCIs, identifying SCIs, identifying the baselines to be established and the corresponding SCIs and the procedure for obtaining SCIs from the project repository.
- A defined process is used to manage change requests in the SCM process's change control task from the time they are initiated until the change is completed and made available to users. The evaluation of a change request, the implementation of the change and the verification and release of the change are all parts of the change control process.
- To manage numerous versions of a configuration and the SCIs that make it up, the **version control** job of the SCM process is used. Version control principles are applicable to all types of versions of software products, including revisions, variations and variants.
- Software Quality Assurance (SQA) is the role of the **configuration auditing task** of the SCM process. It is carried out professionally and objectively to make sure that the modifications have been implemented correctly and that quality has been upheld. Functional Configuration Audits (FCA), Physical Configuration Audits (PCA) and Formal Qualification Reviews are the three different forms of configuration audits (FQR).
- The **reporting task** of the SCM process informs those who might be impacted by the changes in a timely manner about the status of the requested modifications and the SCIs. The developers, the project manager and senior management may be those making modification requests.

4.22 Participant of SCM Process

1. Configuration manager :

- The person in charge of recognizing configuration elements is the configuration manager.
- CM makes sure the team adheres to the SCM methodology
- He/she must accept or reject requests for changes.

2. Developer :

- The developer must alter the code in accordance with change requests or customary development operations. He is in charge of keeping the code configuration.
- The developer needs to review the modifications and resolve any problems.

3. Auditor :

- The auditing professional is in charge of SCM audits and reviews.
- The release must be consistent and comprehensive.

4. Project manager :

- Ensure that the product is created within a specific amount of time.
- Keeps track of development's progress and pinpoints problems with the SCM procedure
- Produce reports on the condition of the software system.
- Ensure that procedures are followed while developing, modifying and testing

5. User

- To make sure he has the most recent version of the software, the end user should be familiar with the essential SCM terminologies.

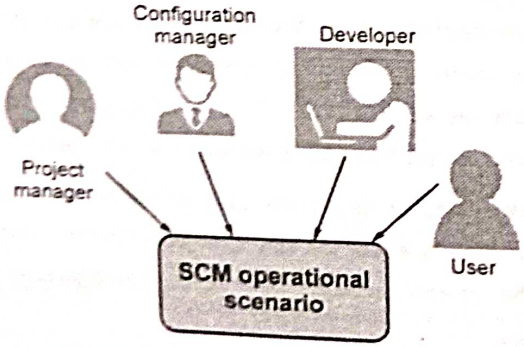


Fig. 4.22.1 : Participant of SCM process

4.23 Software Configuration Management Tools

- SCCS (Source Code Control System) and RCS (Revision Control System) are two popular configuration management tools available on most UNIX systems. They are used for controlling and managing different versions of text files (but not binary files) and provide an efficient way of storing versions that minimizes the amount of occupied disc space.
- Binary files cannot be handled by SCCS or RCS (i.e., executable files, documents, files containing diagrams, etc.). Versions can be stored effectively with SCCS and RCS, using the least amount of disk space possible. If a module called MOD exists in three different versions - MOD1.1, MOD1.2 and MOD1.3 - SCCS and RCS will keep MOD1.1 in its original form together with the modifications that turned MOD1.1 into MOD1.2 and MOD1.3 became MOD1.3. Deltas are the adjustments that must be made in order to upgrade each baseline file to the newest version. Storage capacity is the primary justification for storing the deltas rather than the entire revision files.
- Both SCCS and RCS include options for checking components in and out (also known as reserve and restore procedures), as well as the ability to place limits on the group of people who can produce new versions. Individual developers examine and alter components. They check in the modified module into SCCS or RCS when they have completed all necessary modifications to a component and after these modifications have been reviewed.

4.24 Git

- Best suited for teams with high-value intellectual software code or applications, such as banking software and defence-related software.

Features :

- Installation of Git software on-site.

Pros :

- Simple to learn.
- Offers distributed versioning support.
- Setup on-premises is simple and safe.
- Has no single point of failure because each Git pull locally copies the whole repository.
- The integrity of the code is ensured by the checksums of all files and commits.

Cons :

- Developers must rely on community support because it is free.
- Managing an on-premise configuration can be challenging, especially when other connections like CI tools are involved.

Pricing :

- Free to use and open-source.

4.25 Team Foundation Server and Ansible

- **Team foundation server :** Microsoft Corporation created Team Foundation Server (TFS), an Application Lifecycle Management (ALM) tool. TFS is a great option for teams using Agile Software Development and Waterfall processes because it provides simple and adaptable tools for planning and monitoring projects and individual tasks.
- **Ansible :** Ansible is a platform for configuration management that automates networking, servers and storage. Configuring these components with Ansible makes challenging manual operations repeatable and less error-prone.

4.26 Managing Contracts

- The management of contracts with customers, vendors, partners or staff is known as contract management or contract administration.

Types of contract

- Contracts can be classified majorly in two categories
  - Acquiring software from external supplier
  - Acquiring software based upon payment

Acquiring software from external supplier :

- Further classified into three categories
  - A bespoke system created specially for the customer.
  - Off the shelf bought 'as it is'.
  - Customized Off The Shelf (COTS) - A core system is customized to meet needs of a particular customer.

Acquiring software based upon payment

1. **Fixed price contracts**

   Customer benefits :

   - Known spending;
   - Supplier incentivized to be cost effective;

   Loss :

   - Supplier will increase price to fulfil contingencies;
   - Difficult to amend requirements;
   - Upward pressure on the cost of modifications;
   - Threat to system quality

2. **Time and materials contracts**

   Customer benefits :

   - Requirements are simple to adjust
   - Product quality might be aided by a lack of pricing pressure.

   Loss :

   - Liability of the customer. The buyer bears the entire risk.
   - Linked to criteria that aren't well specified or are constantly changing.

3. **Fixed price per delivered unit**

   Customer benefits

   - Customer awareness of how prices are calculated.
   - Comparability of multiple pricing schedules.
   - Developing functionality can be accounted for.
   - A supplier incentive to be cost-effective.

   Loss :

   - Problems with software size measurement may necessitate the use of a separate FP counter.
   - Changing (rather than new) requirements : how do we charge ?

Typical terms used in contract

- Definition Form of agreement lease, licence, Sale.
- Goods and services to be supplied.
- Ownership of software.
- Environment.
- Acceptance standards.
- Time table.
- Price and payment method.
- Legal requirements.

4.27 The Stages of Contact Management

- Contract management is a complex monitoring process that includes execution, vendor selection, issue identification and control, tracking and processing of contracts from pre-award to conclusion. Contract management techniques, when effectively implemented, ensure that budgets and capabilities are in line with project goals.
- The best contract management integrates project management and control seamlessly throughout the company, always incorporating team members for input and outcomes and closely monitoring contractors for performance and deadlines.

The stages of contract management

- Contract management entails more than just drafting and approving contracts. It consists of a number of stages that guide the process to a successful end. Any missing steps can lead to delays and errors later on. Here's a rundown of the five most important stages :

1. **Create :** The contract management system must be able to include standardized procedures with specific features tailored to the organization's objectives. The first step is to determine the sort of contract and who will be in charge of each duty. While developing an overview of prospective problems and hazards, the planning process should take into account company resources, objectives and team member strengths and limitations.
2. **Negotiate :** The contract should be designed in a way that matches the organization's needs and beliefs, in order to enable the two parties create confidence. As part of the negotiating process, line items might be discussed, altered or eliminated.

3. **Approve :** Approval normally entails a number of signatures. Before the final deal is made, a number of managers, departments and even contractors may need to sign off on the details.
4. **Finalize :** The signing of contracts between businesses is the last step before the project can begin. It's critical to get signatures from a variety of parties and bodies immediately, especially when distance is an issue, to minimize delays in the process.
5. **Manage :** Changes can still happen once the project starts. Revisions must be meticulously monitored and notified to all relevant parties. Deadlines, audits, revenue and expenses must all be kept track of, completed and communicated with the rest of the team.

4.28 Challenges of Contract Management

- Contract management processes can be difficult to manage, especially when done manually. The following are some specific challenges :
  - **Execution :** Without adequate administration, paperwork can quickly become misplaced and final approvals and signatures can take considerably longer than they should.
  - Contract tracking and auditing can be challenging once contracts have been signed and passed on to other employees who may not comprehend the intricacies that must be tracked. Budget information, e-mails, meeting minutes and modifications are all stalled in transit before being addressed. Project delays reduce profitability since time is money.
  - **Revisions :** It's just as vital to manage change before a contract is approved as it is after it's been performed. Larger projects usually involve national or international teams, posing challenges to deadlines and teamwork. Original documentation updates might take days, if not weeks, to complete. Furthermore, if teams work from various versions of the same document, incorrect conclusions may be made. Any of these possibilities could result in higher risk, missing deadlines, errors and even legal action.
  - **Contractual compliance :** Contractual responsibilities are non-negotiable. There may be regulatory or compliance problems that must be followed to the letter from a legal aspect. Failure to comply can result in serious consequences.

Stages in contract placement

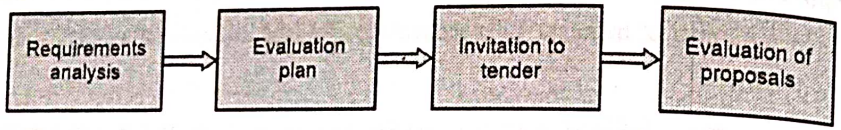


Fig. 4.28.1 : Stages in contract placement

- **Requirement :** All the requirements are documented and which consist of following things :
  - Description of the current system and its surroundings
  - Plans or strategies for the future
  - Features that are required or desired by the system
  - Deadlines
- Software functions with required inputs and outputs.
- Other apps with which software must be compatible must follow certain standards.
- For example, there are quality requirements that must be met. Timeliness of responses
  - Bidders are needed to provide further information.

Evaluation plan :

- This include different methods for evaluation of proposals
  - Reading proposals
  - Interviews
  - Demonstrations
  - Site visits
  - Practical tests
- Need to assess value for money for each desirable feature.

Example :

- Feeder file saves data input
- 6 hours a month saved
- Cost of data entry at RM20 an hour
- If cost of feature RM1000, would it be worth it ?

Invitation To Tender (ITT)

- Note that the bidder is responding to the ITT with an offer.
  - Acceptance of an offer creates a contract;
  - The customer may require additional information
  - Different technical solutions to the same problem

Evaluation of proposals

- Verify that the paper meets all of the requirements. Supplier interviews, demonstrations, site visits and practical tests are all part of the process.

4.29 Benefits of Contract Management Process

- Communication, response time, goal alignment, transparency and accountability are all improved when contracts are managed properly. Good contract management can help we track and improve these and other project performance metrics :
  - **Efficiency :** Contract management streamlines contract adherence and can help businesses save money. All essential documentation may be discovered and tracked in one location, providing greater transparency for team members from other departments as well as contractors operating offsite. Automation makes documents and modifications more accessible, which can shorten the time it takes to respond to changes, additions or issues. Positive contracting experiences may help lock in better raw material pricing and availability, as well as lower service expenses in the future.
  - **Risk reduction :** When a contract is established, important company objectives and goals are identified. A good contract management strategy establishes expectations around those priorities and ensures that contract commitments are kept. The benefit has a wide range of applications, including keeping businesses accountable to performance clauses and ensuring vendor compliance. Regulations are always changing and staying on top of them necessitates ongoing monitoring to guarantee that our projects remain compliant. Assets, commitments and provisions must all be tracked to reduce the risk of financial loss. A contract management tool's tracking feature constantly monitors performance and compliance in real time.
  - **Building relationships :** When vendors and subcontractors have a favorable contract experience, they become long-term business partners. Finding good staff is critical for future initiatives, especially in the construction business.

- Contract management technologies organize and centralize documents and processes, allowing for easier access to corporate data and analytics. Standardized reporting and record - keeping ensures accuracy and transparency, resulting in actionable insights. Projects are less likely to become stuck due to an unanticipated challenge or compliance issues when contracts are managed transparently.

Review Questions

1. What are the different types of contracts ?
2. List the various typical terms of a contract ?
3. Write short notes on contract management ?
4. Explain the different stages in contract placement.
5. What are the different challenges in contract management ?
6. What are benefits of CMP ?
7. State different SCM tools
8. Define : Git, Team foundation server and ansible.

4.30 Types of Contract in Software Project Management

- The necessary external resources could take the shape of services, like staff working under short-term contracts to complete certain project tasks. Brigette may engage temporary workers at Brightmouth College to enter the necessary employee information for the new payroll system. IOE may build applications internally but supplement the regular personnel with contract programmers. The contractor may operate the new system for the client in addition to supplying it. For instance, Brightmouth College might decide against purchasing a package and hire a payroll services company to handle the payroll process instead.
- On the other hand, a contract might be made for a finished software package. This might be :
  - A specialized system developed just for one client;
  - An off-the-shelf product purchased "as is"; this is also known as shrink-wrapped software.
  - Customized Off-The-Shelf (COTS) software, in which a basic system is adjusted to a client's requirements.

- In English law, when equipment is purchased, a contract for the delivery of goods is typically involved. When it comes to the provision of software, this may be seen as the provision of a service (i.e., the writing of the software) or the issuance of a licence (i.e., permission to use the software, which remains the supplier's property). There will be legal repercussions from these discrepancies.
- Another method of categorizing contracts is according to how the payment to suppliers is determined.
  1. Fixed price contracts;
  2. Time and materials contracts;
  3. Fixed price per delivered unit contracts.

1. Fixed price contracts :

- In this instance, the cost is established at the time of contract signing. The client is aware that this is the final price they will pay, barring any modifications to the terms of the contract.
- The customer's requirement must be established up front for this to work. In other words, a full requirements analysis must have been completed before the contract to build a software system was signed. Once development has begun, the customer cannot alter their specifications without renegotiating the contract's pricing.
- These are the advantages of this approach :
  - Recognized consumer spending. The customer has a known cost as long as the requirements are accurate and remain unchanged.
  - Supply chain inspiration. The supplier is driven to do their work efficiently.
- The disadvantages comprise :
  - Higher prices to account for uncertainty. Any estimations that are inaccurate are taken on by the provider. The supplier will increase the price by a margin to offset the impact of this risk.
  - Challenges in changing needs. During development, it may become necessary to modify the requirements' scope, which could lead to conflict between the supplier and the customer.
  - Increased pressure on change's price. The supplier will attempt to offer the lowest price when competing with other potential suppliers. If new requirements are made after the contract has been signed, the provider is well-positioned to demand a premium price for them.

- Risk to system integrity. The software's quality could be compromised in order to reach a specified price.

2. Time and materials contracts :

- In this kind of contract, the customer is charged a certain amount per unit of work, such as per employee hour.
- Based on their existing understanding of the customer's requirements, the supplier may offer an initial cost estimate; nevertheless, this estimate will not serve as the foundation for the final payment. For completed work, the supplier typically sends monthly invoices to the client.
- These are the benefits of this strategy :
  - It's simple to adjust requirements. This may be a suitable mode of payment when a project has a research emphasis and its course is subject to change as choices are investigated.
  - No downward price pressure. Better quality outputs could be encouraged by the absence of pricing pressure.
- These are the drawbacks of this strategy :
  - Customer responsibility. Risks resulting from inadequately defined or changing needs are assumed by the client.
  - Lack of supplier incentives. The supplier has little motivation to work efficiently or to manage the deliverables' scope.
- This strategy is unpopular with customers because it gives the impression that the provider has been issued a blank check. However, this kind of contract might actually be involved in the hiring of contract development workers.

3. Fixed price per delivered unit contracts :

- This frequently goes hand in hand with Function Point (FP) counting. At the beginning of the project, the system's size is computed or estimated. The size could be calculated in terms of lines of code, however FPs can be obtained more quickly from requirements files. Also included is a price per unit. The ultimate price is then calculated by multiplying the unit price by the quantity of units. A typical price schedule is shown in Table 4.30.1.

Function point count	Function design cost per FP	Implementation cost per FP	Total cost per FP
Up to 2,000	$ 242	$ 725	$ 967
2,001 - 2,500	$ 255	$ 764	$ 1,019
2,501 - 3,000	$ 265	$ 793	$ 1,058
3,001 - 3,500	$ 274	$ 820	$ 1,094
3,501 - 4,000	$ 284	$ 850	$ 1,134

Table 4.30.1 : A schedule of charges per function point

- For larger systems, the company that created this table actually charges a greater amount per FP. For instance, a system that will be put into use has 2600 FPs. A total of 2000 * $967, 500 * $1,019 and 100 * $1,058 would be required.

These are the benefits of this strategy :

- Customer comprehension The buyer may see how the pricing is determined and how it will change as their needs change.
- Ability to compare pricing plans can be contrasted.
- Emerging capabilities The risk of adding functionality is not borne by the supplier.
- Supplier effectiveness The supplier is nevertheless motivated to provide the essential functionality in a way that is economical (unlike with time and materials contracts).
- Life-cycle period. The needs need not be unambiguously stated at the outset. Thus, the project's analysis and design phases can both be covered by the development contract.

These are the drawbacks of this strategy :

- Measurement difficulties with software size. Adopting a verbose coding style can easily increase the number of lines of code. There may be arguments regarding the proper FP count in circumstances when the supplier or the consumer are unfairly favored by the FP count in circumstances when the supplier or the consumer are unfairly favored by the FP counting regulations. Users in particular are quite likely to be unfamiliar with the idea of counting regulations. Users in particular are quite likely to be unfamiliar with the idea of FPs, therefore they may require further training. Using a separate FP counter may be the answer to these issues.
- Modifying specifications Some requested changes might have a significant impact on the existing code while without adding to the total FP count. It normally takes more work to implement a modification that is made later in the development cycle than one that is made early.

- Australia has made the recommendation that the charge be adjusted according to the stage at which they are requested to lessen the final issue; see Table 4.30.2.

	Pre-acceptance testing handover	Post-acceptance testing handover
Additional FPs	100 %	100 %
Changed FPs	130 %	150 %
Deleted FPs	25 %	50 %

Table 4.30.2 : Examples of additional charges for changed functionality

- There are other payment methods and variations on those alternatives. Any additions or modifications to the requirements might be paid per FP, with the cost of implementing a specification being set. It is possible to negotiate a contract where the final price includes a fixed percentage for labor plus an amount that depends on the real cost of acquired components when the contractor has purchased equipment, the price of which may vary.
- Another way to initially categorize contracts is in accordance with the method of contractor selection, particularly.
  - Open
  - Restricted
  - Negotiated.

Open tendering process :

- There has been a global movement to remove obstacles that prevent companies from one country from providing goods and services to firms in another. Examples of this include the World Trade Organization's (WTO) and the European Union's initiatives to make sure that public institutions do not unjustly favor neighborhood firms. One of the agreements under the WTO's supervision relates to government procurement and establishes guidelines for the tendering procedure. When a public entity is the client, an open tendering procedure might be required.
- Any supplier may submit a proposal to provide the goods and services in this situation. All offers that adhere to the original requirements in the invitation to tender must be taken into account and assessed uniformly. This evaluation process can be time - and money - consuming with large projects.

Restricted tendering process :

- Only suppliers that have been invited by the customer have submitted bids in this instance. In contrast to an open tendering procedure, the client may decide to examine fewer suppliers at any time. Typically, this is the best course of action.

Negotiated procedure :

- The restricted tendering procedure might not, however, be the best option in some specific situations for a number of valid reasons. Imagine, for instance, that some ICT equipment is destroyed in a fire. It might be more important than everything else in this situation to get the new equipment up and operating as soon as possible and there might not be enough time to start a protracted procurement process. Another scenario would be where a new software program has been effectively developed by a third party supplier, but additional system extensions are needed.
- It could be awkward to contact other possible suppliers through a complete tendering process because the original supplier has staff that is familiar with the current method. In these circumstances, approaching a single provider might be appropriate. Approaching a single supplier, however, could lead to accusations of bias, thus it should only be done with a compelling argument.

Review Question

1. What are different types of contract in software process management ?

❑❑❑

Unit V

5 Managing People and Organizing Teams

Syllabus

Understanding Behavior - Organizational Behavior - Selecting the Right Person for the Job - Instruction in the Best Methods - Motivation - The Oldham - Hackman Job Characteristics Model - Stress - Health and Safety - Ethical and Professional Concerns - Becoming a team - Decision Making - Organization and Team Structures - Coordination Dependencies - Dispersed and Virtual Teams - Communication Genres and plans - Leadership.

Case study : Team Building in Project Management with reference to academic project work.

Contents

5.1 Understanding Behavior
5.2 Organizational Behavior
5.3 Selecting the Right Person for the Job
5.4 Instruction in the Best Methods
5.5 The Oldham - Hackman Job Characteristics Model
5.6 Stress
5.7 Health and Safety
5.8 Ethical and Professional Concerns
5.9 Working in Team
5.10 Decision Making
5.11 Organization and Team Structures
5.12 Co-ordination Dependencies
5.13 Dispersed and Virtual Teams
5.14 Communication Genres
5.15 Communication Plans
5.16 Leadership
5.17 Case Study : Team Building in Project Management

5.1 Understanding Behavior

Introduction :

- The managing of people is consistently cited as a crucial component of project management by those with actual experience in the field. People like Brigette and Amanda would like to know whether or not professional counsel may assist in the successful and compassionate administration of workers. If the advice is supported by data that has been acquired through some form of research, it may be more persuasive.
- Social science research methodologies must be used in this study of individual and group behavior in software and ICT development environments. The perspective needed for this kind of research is distinct from what software developers often need.
- Although user requirements that can be interpreted in several ways are typically the basis for system development, the final result is a system that operates in a flawlessly consistent manner. The designers who create these systems will unavoidably have a predisposition to think of things in terms of deterministic systems, where the results can be predicted quite well once the sequence of inputs is known.
- As in the physical sciences like chemistry, such systems are thought to be guided by mechanistic laws. This attitude, which is also known as a positivist approach, frequently favors experimentation as a way to determine the links between inputs and outputs.
- This approach has been attempted to be applied to social systems. However, due to the complexity of social systems, including commercial organizations, it is impossible to forecast their results with any degree of precision. Finding statistical correlations inside such systems that can be stated as generalized models or theories is what can be done.
- Theories that attempt to explain human behavior have developed in the field of organizational behavior. These theories are sometimes formulated as "If A is the scenario, then B is probably going to happen." A statistical relationship between the two variables is sought while attempting to monitor behavior where variables for A and B are measured. Contrary to physical science, it is almost never true that B must always follow A.
- In the real world, a scenario will be impacted by a variety of factors, many of which are imperceptible to the spectator. Determining which set of research findings is pertinent is so challenging. We run the risk of developing a collection of maxims that are essentially superstitions. People can, at the very least, become more sensitive and considerate about these concerns by studying them.

5.2 Organizational Behavior

Background :

- Studies in OB have its roots in Frederick Taylor's work from the late 19th and early 20th centuries. Taylor made an effort to analyze the most effective manner to carry out manual chores. The personnel were then instructed on how to complete the task.
- Taylor has three main goals :
  - To choose the best candidates for the position;
  - To teach them the most effective techniques;
  - To reward top performers with increased compensation as incentives.
- Taylorism is frequently portrayed as being simplistic and mechanistic. The need for defining best practices is real, nevertheless. The development of both structured and agile methodologies in the less glamorous area of software development is an illustration of the focus on best practice. Brigette and Amanda will be anxious that duties are completed correctly. Although Amanda and Brigette will encounter many coworkers who share Taylor's opinion on the significance of "performance-related pay," his focus on the solely financial basis of staff motivation is more problematic.
- Unfortunately, It's likely that Amanda and Brigette have very little influence over the financial incentives provided to their employees. They should take heart from research showing that motivation is not only dependent on such benefits.
- While conducting a now-famous series of studies on the working circumstances of staff in the 1920s, OB researchers found that both a control group and a group of workers who had their working conditions left unaltered increased their work rates. The act of merely caring about what employees accomplished enhanced output. This demonstrated how employees' attitudes affected their productivity.
- Thus, a more nuanced picture of people at work can be contrasted with the cash-oriented or instrumental, view of labor held by certain managers. Donald McGregor gave the two attitudes the names Theory X and Theory Y.
- Based on Theory X,
  - The typical person has a natural aversion to work;
  - Therefore, there is a need for compulsion, direction and control;
  - People frequently avoid their responsibilities.

- On the other hand, Theory Y proposes that :
  - Play and rest are equally natural as is work;
  - External control and coercion are not the sole means of enforcing an organization's objectives;
  - Commitment to goals is a result of the benefits attached to their accomplishment;
  - The average person can learn to take responsibility and pursue it;
  - There are many people with the ability to use their imagination and other qualities.
- A "reward" need not be monetary; it could be something positive, like a feeling of accomplishment.

5.3 Selecting the Right Person for the Job

- Taylor emphasized the necessity for the right person for the job. The productivity of programming is impacted by a variety of elements, including the use of software tools and approaches.
- The biggest variations in software development performance, nevertheless, are between individuals. In 1968, experienced professional programmers working on the same assignment were compared.
- In one case, the ratio between the quickest and longest times to code the programme was 1:25 and probably more importantly, the time required to debug it was 1:28. Therefore, it would be understandable for Amanda and Brigette to want to hire only the best candidates.
- Staff members who can interact successfully with customers and with one another are what Amanda and Brigette will seek.
- Information Systems (IS) professionals appeared to have substantially less "social needs" than persons in other occupations, according to American academics Couger and Zawacki. Gerald Weinberg is quoted as saying, "If questioned, most programmers probably say they want to work alone where they wouldn't be bothered by other people."
- Many software developers are talented and drawn to the field, but many do not make effective managers later in their careers.

The recruitment process :

- It must be stressed that project managers frequently have little control on the individuals who will make up their team - they are forced to make due with "resources that are available." Recruiting is frequently the responsibility of the organization and the hired individual may eventually work in a variety of different departments.

- Meredith Belbin makes a helpful distinction between qualified and qualified applicants. Candidates that are qualified have a Curriculum Vitae (CV) that includes information like the "correct" number of years spent in a previous position and the "proper" academic credentials. Effective applicants are capable of performing the task well.
- Making the error of choosing an eligible candidate who is not actually appropriate. On the other side, qualified candidates who are not formally eligible may be the best choices since, once appointed, they are more loyal.
- Belbin advises that we aim to evaluate actual talents rather than prior experience and offer training to fill up any slight gaps in expertise.
- We believe this proves that avoiding discrimination on the basis of race, gender, age or irrelevant disability may be both a socially decent and effective hiring strategy.
- **The following might be a broad strategy :**
  - **Create a job specification :** Advice is frequently required because a formal document could have legal ramifications. The needs of the job, including the kinds of tasks to be performed, should be written and agreed upon, whether formally or informally.
  - **Create a profile for the job holder :** The individual required to perform the job is profiled using the job specification. The necessary characteristics, credentials, training and experience would be listed.
  - **Recruit applicants :** Typically, an advertisement would be published, either internally within the company or outside in the local or trade press. To determine the media most likely to reach the most number of possible applicants at the lowest cost, the job holder profile would be thoroughly evaluated. For instance, it would make sense to place an ad in the appropriate specialized publication if a specialist was required. The second rule is to include enough details in the advertisement to permit some self-elimination. The applicant pool will be reduced to those who have more realistic qualifications by providing the income, location, job description and any prerequisites.
  - **Examine resumes/CV :** Nothing is more unpleasant for everyone involved than when persons have CVs that make it plain they are ineligible for the job but are still invited for an interview despite this. These should be carefully examined and compared to the job holder profile.
  - **Interviews :** Aptitude exams, personality tests and the scrutiny of work samples are all examples of selection approaches. Any technique must put to the test certain traits that are listed in the profile of the job holder.

- The method that is most frequently utilized is an interview. In order to maximize the chance of follow-up questions and discussion, it is preferable to conduct more than one interview session with each applicant.
- Additionally, there should only be two interviewers each session. For the attributes being evaluated, a formal scoring system should be developed and interviewers should then decide on individual scores that are then compared.
- An interview could be of a more general or technical nature, when the candidate's practical expertise is evaluated. In the latter scenario, a significant portion of the interview may include assessing and correlating information from the CV. For instance, gaps in schooling and career history would be looked into and the specifics of former jobs would need to be probed.
- **Other techniques :** A medical checkup can be required and references will need to be contacted if applicable.

Review Questions

1. Explain the recruitment process in detail.
2. How to select a right person for the job ?
3. Write a note on behavior. Explain organizational behavior in detail.

5.4 Instruction in the Best Methods

- This is the second issue that Taylor raised that we are concerned about. The team leader will need to carefully arrange the on-boarding of any new team members when they are hired.
- This might be challenging in cases where a project is already well underway. However, the effort should be made because the new hire will join the company more rapidly and be able to contribute completely.
- The team leader should be aware of the necessity of regularly evaluating the team members' training requirements. Just as we should create a user requirement before contemplating a new system and a job holder profile before hiring an employee, we should also create a training needs profile for each employee before evaluating any particular courses.
- Commercial training organizations may offer some training. Alternative training options should be taken into consideration when money is limited, but training should not be ceased.

- It may just be a team member learning about a new software application and showing it to coworkers. Attending meetings of our local branch of a computer-related professional body, such as the British Computer Society (BCS) in the United Kingdom, might serve the similar purpose. Of course, the good thing about external courses is talking to colleagues from different organizations.
- Obviously, the techniques learned need to be put into practice. This is helped to guarantee via reviews and inspections.
- It will be necessary to make choices regarding whether a newcomer may acquire technical skills more successfully through on-the-job training or formal training courses.

Motivation :

- Taylor's third concern was how to get people to work harder. We'll examine a few motivational model examples.

1. The Taylorist model :

- The usage of piece rates in manufacturing and sales bonuses among sales staff are examples of how Taylor's philosophy is represented in these practices. Problems with piece rates may arise if a new system alters how people work.
- If new technology increases production, it will be difficult to change piece rates to reflect this. A shift from piece rates to day rates is typically required prior to implementing major changes in labor patterns.
- Workers who are paid a fixed amount per item they create are said to be paid on a piece-rate basis. When discussing pay for time worked, day rates are used.
- People who are paid according to how much they do will not necessarily optimize their productivity in order to maximize their pay, even in stable work environments where output and reward are easily correlated. "Group norms" informal, sometimes unspoken agreements among co-workers about the amount to be produced often serve as a constraint on the output.
- Rewards based on piece rates must be directly related to the task completed. Since system development and support are typically team efforts, it is challenging to isolate and quantify work performed by one individual when a computer application is being developed.
- This support department succeeds because we are a team, not because we are all individuals, as one employee in a study of software support work put it. This is the only way the support team can function effectively.

- A reward system that unfairly discriminates between coworkers could harm morale and productivity in this type of workplace.
- Giving project team members bonuses at the conclusion of a successful project is one way that organizations occasionally get around this issue. This is especially true if staff workers "volunteered" a significant amount of unpaid overtime to complete the project.

2. Maslow's hierarchy of needs :

- People are motivated for different reasons. When we're poor, money can be a powerful drive. However, when the fundamental need for money is met, new drivers are likely to emerge.
- American psychologist Abraham Maslow proposed a hierarchy of needs. A greater level of needs gradually appears as a lower level of demands is met. If requirements are met, another level will then emerge.
- Food, shelter and personal safety are considered basic needs. Maslow asserts that the need for "self-actualization," or the sensation of fully realizing one's potential, is the highest level needed.
- In practice, people are probably inspired by many factors depending on their stage of life. For instance, salary raises, while always appreciated, probably have less of an effect on a more experienced worker who is already quite well paid than they do on a trainee who is paid little.
- Older team members might place higher significance on aspects of the job that respect their judgment and sense of responsibility, such as being provided autonomy.

3. Herzberg's two-factor theory :

- We might not always be happy with a job. The job may not become more enjoyable even if the reasons for this discontent are eliminated.
- Herzberg and his colleagues conducted research on characteristics that affect job satisfaction and discovered two sets of factors :
  - Aspects relating to hygiene or maintenance that, if not met, might lead to dissatisfaction, such as wages or working conditions;
  - Motivators, such as a sense of accomplishment or the difficulty of the work itself, that make us feel like the job is worthwhile.
- At Brightmouth College, Brigette may find it challenging to compete with the high level of maintenance factors that may be offered by a major organization like IOE, but the smaller organization may be able to offer greater motivators due to its closer ties to the users.

4. The expectancy theory of motivation :

- The ups and downs of system growth must be considered by Amanda and Brigette in order to understand how motivation is impacted. This is demonstrated through a motivational model created by Vroom and his associates.
- It lists three factors that affect motivation :
  - **Expectancy :** The conviction that exerting more effort will result in improved results;
  - **Instrumentality :** The conviction that superior work will be rewarded;
  - **Perceivable value** of the subsequent prize.
- All three conditions must be favorable for motivation to be high. Motivation can be eliminated if any one of the elements is at zero.
- Imagine attempting to make a software package from a third party work. We give up after realizing that there is a bug that will prevent us from ever getting it to function. No matter how hard we try, we will never be successful((zero expectancy).
- We are working on a package for a user and even though we think we can make it work, we find out that the user has moved on to another package and no longer requires the one we are working on. It's likely that we'll feel like we're wasting our time and give up(zero instrumentality).
- Given that the users genuinely desire the product, our incentive may be nothing more than the nice glow of helping our coworkers and their appreciation. We might decide not to become involved if users later ask for assistance implementing an alternative package, if all they do when using the package is complain and blame us for flaws. (low perceived value of reward).

5.5 The Oldham - Hackman Job Characteristics Model

- The elements of tasks that need to be completed should be grouped together by managers so that they constitute relevant and fulfilling assignments. According to Oldham and Hackman, there are five characteristics that contribute to job satisfaction.
- The first three elements below help a person's employment feel "meaningful" to them :
  - **Skill variety :** The range of abilities that the job holder has the chance to use;
  - **Task identity :** Refers to how clearly our labor and results can be attributed to us;
  - **Task significance :** The extent to which our work affects other people.

- The remaining two elements are :
  - **Autonomy :** The freedom to choose how we will carry out our duties;
  - **Feedback** is information we receive regarding the outcomes of our job.
- The requirement for personal development on the part of job holders as well as their workplace environment had an impact on how they perceived their jobs, according to Oldham and Hackman.
- Some authors have argued that people are more inclined to rank their work higher on the Oldham-Hackman dimensions if they are happy with it for other reasons. As a result, cause and effect could be inverted.
- Practical terms, activities should be planned such that, whenever feasible, personnel can track the development of a certain product and experience a sense of personal connection to it.

Methods of improving motivation

- The manager could thus take the following actions to increase motivation.
  - **Set specific goals :** These objectives must be challenging while still being supported by personnel. Goals are more likely to be accepted if staff members are involved in their creation.
  - **Provide feedback :** Goals must be established, but employees also require regular feedback on their development.
  - **Consider about job design :** Jobs can be changed to increase staff members' sense of responsibility and make them more interesting.
- Job enlargement and job enrichment are two strategies that are frequently used to improve job design.
  - **Job enlargement :** The person performing the job does a larger range of tasks. It goes against the trend of specialization growth. For instance, a software developer in a maintenance team might be assigned responsible for both specifying and implementing minor changes to the code. Programmers / analysts had a greater level of job satisfaction than programmers, according to Couger and Zawacki.
  - **Job enrichment :** The position holder performs duties that would often be performed by managers or supervisors. In a maintenance team, programmers could be given the permission to accept requests for improvements that need less than five days' worth of work without the manager's approval.

Review Questions

1. Explain motivational models with examples.
2. What are the different methods for improving motivation ?
3. Write a short note on : The Oldham - Hackman job characteristics model.

5.6 Stress

- Projects are about getting things done and overcoming challenges. Both the project manager and the team members will almost certainly be under pressure.
- 'Once a project gets going, we should expect participants to be putting in at least 60 hours a week,' an American project manager is reported as stating. The project manager should plan to work as many hours as possible.
- In fact, some pressure is healthy. Many careers might become soul-destroying due to boredom. However, at a certain point, the quality of the labor declines and health issues may arise. There is solid evidence indicating that working more than 40 hours per week reduces output quality and productivity.
- In a US research conducted in 1960, it was shown that those under 45 who worked more than 48 hours per week had a twofold increased chance of dying from coronary heart disease.
- Many software developers are required to work additional hours without getting paid for them. In these situations, the fact that the work is essentially free to the employer more than makes up for any loss in productivity.
- Clearly, there are occasions when extra effort is required to get overcome a momentary roadblock or handle an emergency, but if working overtime becomes the norm, there will be long-term issues.
- By making a more accurate evaluation of the effort and time required, based on comprehensive documentation and study of the results of past projects, good project management can reduce the need for overtime. Effective planning and management will also lessen 'unanticipated' issues that cause unneeded crises.
- **Role ambiguity** can lead to stress when employees are unsure about the goals that their job is designed to achieve, what is expected of them by others and the precise extent of their obligations. There is no doubt that the project manager is at fault in these situations.

- **Role conflict** can also cause stress to increase. The individual is now conflicted by the demands of two distinct roles. A parent of a young child could feel conflicted about attending a crucial meeting to land new business versus caring for a sick child.
- Some managers assert that they achieve success by using essentially bullying tactics to force projects through.
- To support the employment of such strategies, they must manufacture crises. However, this is the exact opposite of professional project management, which tries to create complicated goods in an organized, methodical and sensible manner.

5.7 Health and Safety

- Construction and other heavy engineering projects tend to have greater health and safety concerns than ICT development. However, there are occasions when setting up office systems calls for building physical infrastructure, which might come with its own set of risks. ICT infrastructure might be implemented in a building that is still under construction, for instance.
- A project manager needs to be familiar with the information in the document that pertains to the setting where the project will be carried out.
- As far as the project manager is concerned, safety goals should be handled the same way as other goals, such as the degree of dependability of the finished product or the project's overall cost, when applicable. Therefore, the general management of the project should include the management of safety.
- The specifics of the many laws that control safety policy can be found in the relevant literature. A documented safety policy statement is required by law in the United Kingdom for companies with more than five employees.
- All levels must have a clear understanding of who is responsible for safety. There are a few things to take into account, such as :
  1. The safety policy must have the support of top management;
  2. Duties for safety must be clearly delegated;
  3. Job descriptions should outline the responsibilities for maintaining safety;
  4. The people to whom responsibilities are assigned must comprehend and accept those obligations;
  5. The dispatch of a safety officer and the assistance of professionals in specific technical fields;

6. Safety consultation;
7. A budget that takes safety expenses into account.

- Employees need to be made aware of safety procedures and receive the proper instruction as needed.

Review Questions

1. How to deal with stress in project management ?
2. Who is responsible for safety ? Explain in detail.
3. What is health and safety in the construction and engineering projects ?

5.8 Ethical and Professional Concerns

- Regardless of their position, all community members have some ethical responsibilities, like notifying the emergency services when a serious motor vehicle accident occurs.
- Other ethical responsibilities have an impact on certain organizations and the members of them. Additional responsibilities are related to a person's field of expertise, such as a software engineer or an IT professional.
- Given their larger ability to cause harm than people, organizations may be regarded to have more ethical obligations, especially when they carry out significant development projects of all types.
- However, there is a claim that people employed by for-profit businesses have a duty to protect and improve the assets of the company's investors. This claim is notably linked to the economist Milton Friedman.
- These company's stockholders, which could include regular people who invested their retirement funds in the company, are those who have contributed money to the business and are therefore legally its owners.
- It was suggested that the company's employees would act dishonestly if they pursued alternative objectives that would benefit the community as a whole at the expense of the stockholders.
- Commercial organizations' reduced - or at least unusual - ethical responsibilities are supported by the fact that they compete with other companies.
- Our competitors must lose if our company succeeds in any way in this game investors might lose money and employees might lose their jobs. However, it is countered that this is simply how the market functions and as a result, consumers benefit from lower prices.

- However, in the long run, eliminating competitors through competition results in monopolies taking control and driving up prices.
- However, the majority of corporations will acknowledge that they do have ethical obligations. This might only be done for personal gain. Potential customers like we might be leery of giving their business to companies that are plainly driven by greed.
- Organizations frequently articulate their goals and aspirations, perhaps in the form of a mission statement and these goals frequently include concerns for the environment and other issues of general public interest.
- Despite flattening reporting systems and eliminating levels of management, hierarchies in large organizations will never completely disappear.
- Next-level managers will adopt the strategy and create work plans to carry out the strategic objectives in their spheres of accountability. They are making decisions in accordance with their assigned responsibilities when they do this.
- Until we reach the individuals who really implement the decisions, this process will be repeated at progressively lower levels inside the organization.
- Any choice must satisfy a number of organizational constraints, some of which may appear to clash. To fulfill a legal obligation with a certain time, for instance, a new ICT application can be required.
- It would take a big team to create a high-quality system with guaranteed correctness and reliability. This would be extremely expensive and require a reduction in customer service levels.
- The need for reliability in the new system and the standard of customer service would need to be balanced in some way. There would be some risk associated with the results regardless of the ultimate choice.
- Risk-related choices will include those that are delegated to technical professionals like engineers and ICT specialists. Since they possess information and expertise that others may not completely understand but on which they rely, they will have particular ethical obligations. These specialists will probably be trusted with making decisions on the introduction of new technologies.
- Knowing a person's area of competence is essential because ICT practitioners are unlikely to be knowledgeable in every facet of ICT and its evolution.
- An ICT practitioner would clearly be acting unethically if they pretended to be an expert in a field they are not. It follows that it would be unethical for an ICT practitioner to keep quiet if their knowledge could stop a colleague from acting in a damaging way.

- The decisions made by these specialists would need to be fair as well as technically justifiable. It is obvious that accepting payments that resemble bribes is unacceptable.
- However, suggesting a certain technology just because the practitioner is an expert in it and its adoption would further his or her career might not seem unethical to an individual right away.
- All decisions include some level of risk, which real professionals would need to recognize and highlight.
- Organizational actions tend to implement a top-down approach, with major strategic decisions made first, followed by an examination of the various components of the overall plan and more specific decisions. These high-level decisions sometimes contain technical errors and it is the job of the software engineer or ICT professional to draw attention to these faults.
- This responsibility to manage new technical risks does not fall exclusively on the shoulders of the practitioner. A method for communicating such concerns to a responsible manager qualified to assess the situation and take appropriate action is required in the organization. This can entail escalating the problem to a higher management level.

Review Question

1. Write a short note on ethical and professional concerns.

5.9 Working in Team

Introduction :

- Software development is a work that requires a lot of human mental effort, despite the fact that we identify it with advanced technologies.
- Software-based systems can be very complex; for example, the software that controls a telephone switching system may have five million lines of code. As a result, this human effort must be split among several software engineers working in teams and across companies.
- Teams are typically defined as groups of individuals working together. The individuals typically work together or co-locate, although we will see that this is not always the case. However, all the individuals working on a project may occasionally be referred to as the "project team." These people might be separated from one another by a distance in distinct work groups. These groups are subject to evolution. Thus, as projects begin and end, specific software developers are likely to switch between teams.

- We'll begin by examining the context of small groups, where the term "team" may be most appropriate. We'll examine the formation of real teams. We'll see how team members take on social duties that enhance team effectiveness in addition to their technical roles.
- To complete a joint assignment, a team is formed. We'll learn how some tasks that contribute to the project's goals are best carried out by a single person. Other tasks, typically those requiring judgments or decision-making, might be carried out more effectively by teams.
- We'll examine team coordination strategies. The distribution of employees among tasks within a company must be under control. This is one sort of coordination between teams and individuals that is required for a project; other types will be described.
- Communication methods are often referred to as **communication genres**. This encompasses the organizational conventions involved in the communication in addition to the technologies used. It is possible to choose and construct specific communication genres to address a specific need for project coordination. We'll look at how a **communication plan** can be used to formalize agreements for communication among project stakeholders.
- In addition to coordination that responds to daily issues, there also has to be proactive central guidance. This introduces leadership-related topics.
- Nearly all phases of the stepwise project planning framework will be influenced by the collaborative nature of project work (Fig. 5.9.1).

1. Identify the goals and scope of the project. Here, communication routes are developed and project stakeholders are identified.
2. Name the infrastructure for the project. The project team will operate inside a specific organizational structure, which has been specified.
3. Examine the attributes of the project. The team structure required will depend on choices made for how the project will be carried out, such as whether to purchase or develop software capabilities.
4. Calculate the effort for every activity. Developer productivity will be significantly influenced by individual and group experience.
5. Determine activity dangers. There will be risks related to the staff, such as their availability going forward.
6. Distribute resources.
7. Review and promote the plan. At this time, a communication plan might be created.

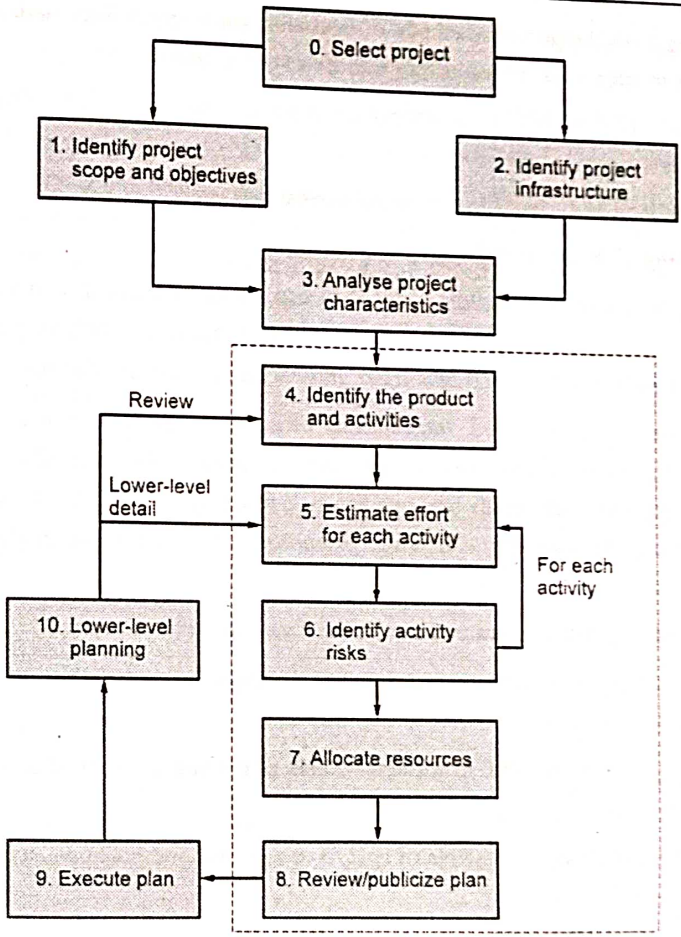


Fig. 5.9.1 : The stepwise structure includes some places where collaboration is influenced

5.9.1 Becoming a Team

- We start by examining the formation of small work groups, where the term "team" is likely most appropriate.
- Just putting people together won't make them instantly able to work as a team. The five fundamental stages of development that teams should go through are as follows :
  - **Forming :** The group members strive to establish some ground rules for behavior as they get to know one another.

- **Storming :** As the group's methods of operation are being defined and various group members attempt to assert leadership, conflicts start to occur.
- **Norming :** The majority of problems are resolved and a sense of collective identity develops.
- **Performing :** The current focus is on the current tasks.
- **Adjourning :** The group splits up.

- Sometimes, particular team-building activities can be carried out. For instance, several companies take their management teams on outdoor activities without going to these extremes, Amanda and Brigette might create some training exercises that foster teamwork.
- Valuable research has examined the best mix of personalities for a project team. Belbin looked at cooperative teams playing management games. He originally attempted to combine the smartest individuals into one group. Surprisingly, these elite teams frequently performed poorly because they frequently disagreed, which led to the neglect of crucial responsibilities.
- Belbin concluded that teams required a balance of various types of people.
  - **The chair :** They need to be adept at running meetings, be cool and firm yet tolerant not necessarily smart leaders.
  - **The plant :** A person who, in general, excels at coming up with ideas and potential solutions to issues.
  - **The monitor evaluator :** Capable of analyzing concepts and potential answers to aid in choosing the optimal one.
  - **The shaper :** Rather a worrier who aids in focusing the team on the crucial issues.
  - **The team worker :** Adept in fostering a positive work environment, such as by "jollying people along."
  - **The resource investigator :** Skilled at finding resources, both in terms of data and material resources.
  - **The completer finisher :** Focused on getting work done.
  - **The company worker :** A strong team player who is willing to carry out less appealing activities when necessary for the success of the group.
- A person can exhibit traits from multiple types. On the other hand, only roughly 30 % of the individuals Belbin investigated could be categorized in any way.

- When there is an imbalance in the role types of the individuals in a group, issues can arise. For instance, a group is likely to have a stormy atmosphere if there are two or more shapers present and no one volunteers to serve as the group's chair and mediate disagreements.
- On the other hand, if a group is primarily made up of plants and experts without any shapers or completer-finishers, the team is more likely to have stimulating conversations but may not end up putting anything into practice. Belbin advises choosing the crucial technical specialists first when assembling a team. These people's group responsibilities can then be evaluated and any remaining team members can be assigned with the goal of balancing out the group roles.

5.9.2 Group Performance

- Decisions on which tasks are best completed collectively and which are best allocated to individuals are necessary for many initiatives. In the words of one IBM manager, "Some work provides better results if carried out as a team whereas some things are slowed down if the work is not compartmentalized on an individual basis."
- The following are some categories for group tasks :
  - Additive tasks;
  - Compensatory tasks;
  - Disjunctive tasks;
  - Conjunctive tasks.
- When performing an **additive task**, each participant's efforts are combined to produce the final outcome, such as when a group of people clean snow. There is no fixed cast of characters.
- With **compensatory tasks**, group members' judgments are combined so that the faults of some are made up for by the contributions of others. For instance, the work required to build a piece of software may be estimated individually by group members, with the results then being averaged. In these situations, group efforts typically outperform those of an individual.
- There is only one right response when dealing with disjunctive jobs. The group's effectiveness is dependent on :
  - Someone providing the proper answer;
  - The others accepting it as accurate.
- The group's performance here is only as excellent as its best member and it may be worse!

- **Conjunctive tasks** are those where the slowest performer's rate determines how quickly things move along. A excellent illustration of this is the manufacturing of software where different staff members are in charge of various components. The assignment as a whole is not finished until each person has finished their portion of the labor. In this situation, having a cooperative attitude is advantageous since the team members who are ahead can contribute to the achievement of group goals by helping the members who are behind. This demonstrates collective heedfulness.
- Social loafing, in which some people fail to contribute as they should, is a risk with all forms of group tasks, but additive ones in particular. This can happen with student group activities without a doubt, but it also happens in "real" job contexts.

Review Questions

1. What is a group task ? Explain different types of group tasks in detail.
2. Define group performance. List out group tasks and explain each task in detail.
3. Explain five fundamental stages of development.
4. Discuss different types of people required for team forming.
5. How to build a team and how to work in a team. Justify the answer.

5.10 Decision Making

- We must first take a broad view of the decision-making process before we can examine the effectiveness of group decision-making in more detail.
- The following categories of decisions exist :
  - Structured judgments are usually simple, routine ones where rules may be applied reasonably simply or
  - Unstructured : More difficult and frequently requiring some inventiveness.
- Decisions can also be grouped according to how much risk and uncertainty are there.

5.10.1 Some Mental Obstacles to Good Decision Making

- We have correctly stressed a structured, rational approach to decision-making up to this point.
- However, in the real world, many management decisions are made under duress and on the basis of inaccurate information. In certain situations, intuition must be accepted, however there are several mental barriers to efficient intuitive thinking. For example :
  - **Faulty heuristics :** Despite the risks, heuristics or "rules of thumb" might be helpful.

  - They are solely reliant on readily available information, which may be deceptive.
  - These are founded on stereotypes, such as admitting a Welshman without an audition to a male voice chorus due to the "well-known" fact that the Welsh are a talented singing people.
  - **Escalation of commitment :** This is a reference to the fact that it gets harder and harder to change our mind once we've made a decision, even if there is proof that it was incorrect.
  - **Information overload :** There is a risk of having too much knowledge and failing to "see the forest for the trees."

5.10.2 Group Decision Making

- "When end users are consulted regarding the functionality of a projected computer system, a different sort of participatory decision making may take place".
- There may be times when Amanda at IOE, for example, wants to speak with the entire project team. A project team might bring together various specialists and points of view. Rather than being imposed, decisions made by the team as a whole are more likely to be accepted.
- Research seems to indicate that groups are more effective at solving complicated problems when the members of the group have complementary talents and competence, assuming that the meetings are genuinely collectively responsible and have been appropriately prepared. They can express themselves openly and get their views accepted at the meeting.
- When faced with poorly organized issues that call for original solutions, groups do less well. Groups can benefit from brainstorming approaches in this situation, although research indicates that individuals frequently generate more ideas on their own than in a group. When involving end users in the development of a computer system is the goal, participatory methods like JAD (Joint Application Development) and prototyping may be used.

5.10.3 Obstacles to Good Group Decision Making

- Group decision-making has drawbacks, which Amanda would discover: it takes time; it can cause disagreements within the group; and decisions can be excessively influenced by dominating personalities.

5.10.8 Extreme Programming (XP)

- Some of these principles have been carried over into the new Extreme Programming (XP) approaches. The majority of XP techniques can be viewed as means of encouraging a "collective consciousness." Adding extra documentation is a standard strategy used in traditional software development projects to enhance collaboration and communication. This, according to XP proponents, is counterproductive. They propose different, less formal strategies for coordination and communication. The essential software products software code and test data are improved rather than separate documents being created. For instance, to ensure that the code accurately reflects how the system operates, it is regularly refactored (i.e., rewritten). Before the code is written, test cases and expected outcomes are created and they serve as a kind of specification.
- To clarify user wants, there should be a user representative present. Continuous integration testing ensures the fit of software components. It is recommended that developers work in pairs to create software; this appears to be a modernized version of the chief programmer/co-pilot arrangement.

5.10.9 Scrum

- It would be counterproductive if the agile approaches' recommended practices ended up being rigidly codified, regimented and codified themselves. The first to emphasize that various project types will require different approaches are proponents of agile methods like Kent Beck.
- Some of these ideas are shown by the Scrum software development process, which combines parts of both the chief programmer mentality and agile methodologies. The word "Scrum" is derived from rugby scrums and conjures up images of a group of people working together to complete a task. Instead of as a commission for a single client, the procedure was initially created for the development of new software products for a market that is competitive.
- In this case, getting something to market before our rivals might be more crucial than having a wide selection of optional options. While having a product that is appealing to a variety of clients is crucial, there is no way to precisely specify the needs of a certain client. As concepts are tested out during development, feature proposals are likely to change.
- The systems architecture and planning phase are the first steps in the Scrum process. This resembles the chief programmer technique in that the chief architect establishes the entire product architecture. At this stage, the product's needed release date and a list of its desirable features, each with a priority, would be established.

- After this phase, there are several sprints that usually last between one and four weeks each.
- Selected are the features that are hoped to be created during a sprint. There is a list of the steps required to implement the features. Groups, ideally with seven developers and no more than ten, carry out sprints. Scrum teams may work on many sprints concurrently, but each team's sprint must be completed on the same day.
- There are daily brief meetings (usually lasting 15 minutes) to track the sprint's progress. Members of the team update the group on how they are doing with their current work and discuss any difficulties they are running into. Any coworkers who can help with a problem are welcome to speak up during the discussion. This can be the case since the coworker previously encountered a similar issue and resolved it. Discussions about potential solutions take place after the meeting. The team should be motivated throughout these Scrum meetings because everyone can see each other's progress, which should assist the group develop a shared understanding.
- Sprints are time-boxed and some incomplete, lower-priority features may be left over at the end of the sprint period. It will be recalled that with XP, adjustments can be requested at any time. In contrast, external needs are frozen throughout the sprint. To examine the finished products, all sprint teams, together with the other project stakeholders, gather at the end of the sprint. At this time, new features may be introduced and existing ones may be eliminated or changed. Desired features' order of importance could be changed. The tasks required to deploy those features are then selected for the upcoming sprint and planned. The sprint procedure is then carried out once more. After each sprint is finished, there is a final closure phase where duties like regression and integration testing, as well as the creation of user and training guides, are carried out to produce a finished product for market release.

Review Questions

1. What is decision making ? Explain mental obstacles to good decision making.
2. What is group decision making ? Explain obstacles to good group decision making.
3. What are the measures to reduce the disadvantages of group decision making ?
4. Write a note on following terms :

   i) Team heedfulness ii) Egoless programming

   iii) Chief programmer teams

   iv) Extreme Programming (XP) v) Scrum.

5.11 Organization and Team Structures

- In large software development businesses, departments are typically set up. A company's departmentalization may be based on a variety of factors, including employee specialty, product lines, client demographics or geographic location. A certain business, for instance, might have departments for banking, embedded applications and telecom software development. Verticals are another name for these divisions. Small company do not typically have extensive departmentalization. Consequently, we can think of a small business as having just one department.
- Each department often manages a number of projects at once, with a different team of developers being allocated to each one. The way a department is divided into teams and how each team is set up has a big impact on how effective the developers are at completing the project's goals.
- In this context, two crucial challenges that are essential to every organization's ability to function effectively are :
  - **Department Structure :** How are teams arranged inside a department ?
  - **Team Structure :** How are project teams structured ?

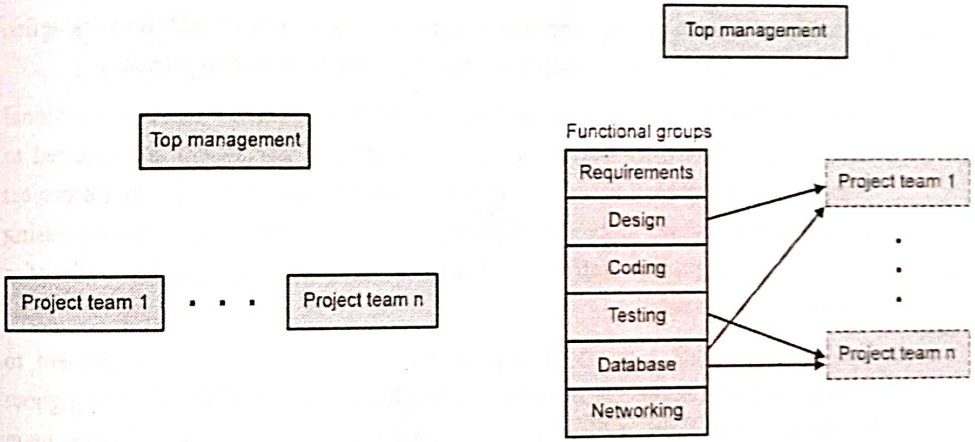
A. Department structure

- A software development department can be organized in one of three general ways : Functional, project-wise or matrix-wise.

1. Functional format

- According to their areas of expertise or experience, the developers are organized into functional groups in the functional format. To put it another way, each functional group is made up of developers who have experience with a particular task or functional area. For instance, an organization's many functional groups may have areas of expertise in areas like database, networking, requirements analysis, design, testing and so forth. Fig. 5.11.1 depicts the functional organizational structure.
- Depending on their area of expertise, each developer in an organization would belong to a specific functional group. Different projects hire developers from the respective functional groupings to complete specific tasks (shown using arrows in Fig. 5.11.1). The developers return to their respective functional groups after finishing their tasks.

- The work done on the partially finished product by many teams allows it to evolve as it is passed from one team to another. The members of other functional teams who worked on different aspects of the project are not physically present when one functional team is working on it.
- As a result, a team can only fully comprehend the work done by the other functional teams by carefully reading the documentation they create. Teams working on the project afterwards will struggle to comprehend the work already done unless each team produces high-quality papers. So, we may conclude that producing high-quality documentation is required for a functional organization.



(a) Project organization (b) Functional organization

Fig. 5.11.1 : Project and functional organization structures

2. Project format

- The project format is intended to help task-oriented teams function. Each project in the project format has a group of developers assigned to it at the beginning (See Fig. 5.11.1 (a)). It is expected that the designated personnel can do the various tasks necessary for the project's completion. Until the project is finished, the developers are still working on it. Thus, all project operations are completed by the same team.
- This is different from a functional organization, where each developer is a member of a functional group and members of the related functional area are temporarily allocated to the project and returned to their own functional area after the activity is completed.

Functional versus project formats

- The team members working on the various project activities do not cross paths in a functional structure. This format's flaw is the resulting communication gap. Additionally, customers frequently prefer the project team method because they will have a team that is solely focused on meeting their needs and because they won't have to interact with many other functional group members. Additionally, the members of the project team become proficient with the software over a considerable amount of time and can complete maintenance tasks quickly. The project team model is therefore appropriate for doing the software maintenance tasks.
- The functional format, on the other hand, comes with a variety of additional benefits. The following are the primary benefits that the functional team format provides.
  - **Ease of staffing :** The staffing issue is effectively solved by the functional organizational structure. Typically, various numbers of developers are required to complete certain project tasks. Any relevant personnel can be brought into a project as needed and returned to the functional group after their work is done, greatly easing the problem of project staffing. This may be the most significant benefit of a functional organization.
  - On the other hand, a project organization structure requires the management to accept a specific number of developers at the beginning of the project. Throughout the course of the project, these developers are employed. As a result, many team members waste time during the early stages of software development and the team as a whole is under extreme pressure during the final stages of development.
  - **Production of good quality documents :** Since the team members working on one aspect of a project do not interact with the developers who have finished another aspect of the project and returned to their functional teams, a functional organization necessitates the production of high-quality papers.
  - **Job specialization :** Developers can become specialists in specialized tasks like database, networking, compilers, requirements analysis, design, coding, testing, maintenance, etc. thanks to the functional organizational structure. They carry out these tasks repeatedly for many projects, gaining experience and understanding in their individual fields of specialization.
  - **Efficient handling of the problems associated with manpower turnover :** Compared to a democratic organization, functional organizations help to successfully manage the issue of employee turnover. When necessary, developers are initially

pulled from the functional pool. Additionally, the effective documentation created in this organizational structure makes it easier for any new member to immediately become familiar with the job already done.

- **Career planning :** Compared to a democratic organization, functional organizations help to successfully manage the issue of employee turnover. When necessary, developers are initially pulled from the functional pool. Additionally, the effective documentation created in this organizational structure makes it easier for any new member to immediately become familiar with the job already done.

3. Matrix format

- In order to offer the benefits of both functional and project structures, matrix format is an extension of functional format. The pool of functional specialists in a matrix organization is apportioned to various projects as required. Thus, a matrix can be used to depict the placement of various functional specialists in various projects (See Fig. 5.11.2).

	Project			
Functional group	#1	#2	#3	
#1	2	0	3	Functional manager 1
#2	0	5	3	Functional manager 2
#3	0	4	2	Functional manager 3
#4	1	4	0	Functional manager 4
#5	0	4	6	Functional manager 5
	Project manager 1	Project manager 2	Project manager 3	

Fig. 5.11.2 : Matrix organization

- Be aware that a team member who is assigned to a project reports to both the manager of his functional group and the manager of the project. As a result, in a matrix organization, the project manager must collaborate with several individual functional managers to complete the project.
- Depending on the balance of power between the functional managers and the project managers, matrix organizations can be categorized as strong or weak. A powerful

functional matrix gives functional managers the power to allocate employees to projects and project managers are required to accept the individuals they are given. In a weak matrix, the project manager has total control over the project budget, the choice of whether to hire outside workers and the ability to reject applicants from functional groups.

- Despite the fact that a matrix team organization has many benefits, it may also experience the following two serious issues.
  - Conflicts over staffing can arise between functional and project managers as a result of the multiplicity of authority.
  - In a strong matrix organization, personnel may be often shifted around when the functional managers go into firefighting mode to address the problems in various projects.

B. Team structure

- The reporting, responsibility and communication arrangements within specific project teams are referred to as team structure. We only take into account the democratic, chief programmer and mixed team organizations, while there are many other types that might be used. Please be aware that not every project team inside an organization must be constituted in the same way. In fact, due to variations in their complexities and sizes, various projects typically adopt diverse team structures within the same business.

1. Chief programmer team

- Fig. 5.11.3 shows a schematic illustration of the chief programmer team structure. In this team organization, the head programmer is a senior member who offers technical leadership. The chief programmer team's mentality and organizational structure are in line with Brooks' recommendations that a small team should handle all design work in order to preserve uniformity in the look and feel of the final product. He stated that the other team members should help the designers be as productive as they can be. The chief programmer creates the high-level design and defines the specification. After that, he or she divides the remaining detailed design activities such as coding, testing, documentation, etc. into several smaller jobs and distributes them to the team members. Additionally, he or she combines and verifies the work that other team members have performed.

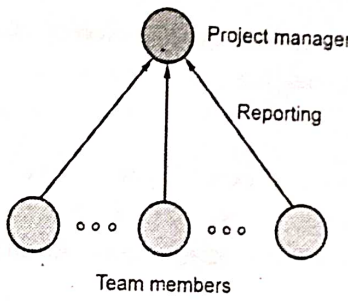


Fig. 5.11.3 : Chief programmer team structure

Advantages of the chief programmer team

- For basic and short projects, a chief programmer team is more effective than a democratic team because the chief programmer can quickly come up with an acceptable design and delegate work to the team members to code and test various modules of his design solution.

Disadvantages of the chief programmer team

- The chief programmer is given the power to delegate work to team members and oversee their performance. However, because the team members work under the main programmer's constant supervision, this lowers team morale.
- The chief programmer team structure prevents the team members from thinking creatively and independently and the chief programmer typically makes all significant decisions by themselves.
- Since the chief programmer is given an excessive amount of authority and responsibility, the chief programmer team is vulnerable to single point failure. In other words, if the chief programmer leaves the company or is otherwise unable to work on a project, it could be seriously hampered.
- Finding a truly great programmer to fill the position of chief programmer is a significant issue with the chief programmer system. The primary programmer runs the risk of information overload because they perform numerous responsibilities independently. Large projects are a case in point.
- Let's say an organization has successfully completed a number of straight forward MIS initiatives. After that, a chief programmer team structure can be used for a comparable MIS project. When a single person has the necessary intellectual capacity to complete the assignment, the chief programmer team structure performs well. The chief programmer team structure should not be used, even for short projects, unless the

urgency of completing the project quickly exceeds other considerations like team morale, individual growth, etc.

2. Democratic team

- The democratic team structure does not impose a formal team hierarchy, as the name suggests. As indicated in Fig. 5.11.4, talks constitute the basis for decisions and any member is free to discuss with any other member. An administrator typically provides the leadership in this area.
- Different team members take on different roles as technical leads at various points. Since there is a lot of discussion and disagreement among team members, there is a high overhead cost for large teams.
- It is well acknowledged that a democratic structure gives team members increased morale and job satisfaction. As a result, a democratic team typically experiences lower staff turnover than a chief programmer team. A group of developers can come up with better answers than a single developer as in a chief programmer team, hence democratic teams are suited for less known problems even though they are less productive than chief programmer teams for short and basic tasks.
- For research-oriented projects requiring fewer than five or six developers, a democratic team structure is appropriate. Purely democratic organizations often devolve into chaos when dealing with major projects. Programmers can exchange and critique one another's work in a democratic team environment, which promotes egoless programming.

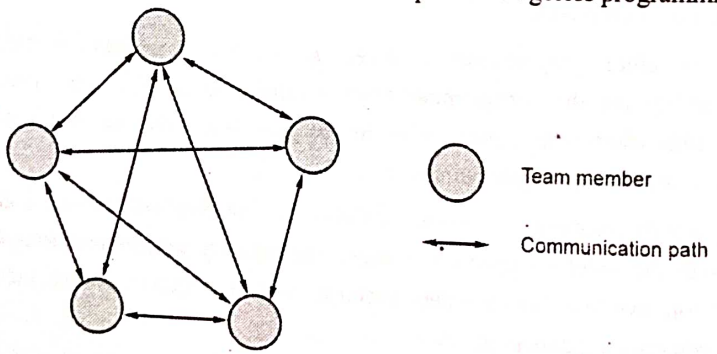


Fig. 5.11.4 : Democratic team structure

3. Mixed control team structure

- As its name suggests, the mixed team structure incorporates elements of both the democratic and chief-programmer team models. Fig. 5.11.5 depicts the mixed control

team structure in visual form. Fig. 5.11.5 depicts the reporting structure using solid arrows and the communication pathways using dashed lines. Be aware that this team structure combines democratic setting with hierarchical reporting. The problem is divided into manageable pieces using the democratic system at the level of senior developers.

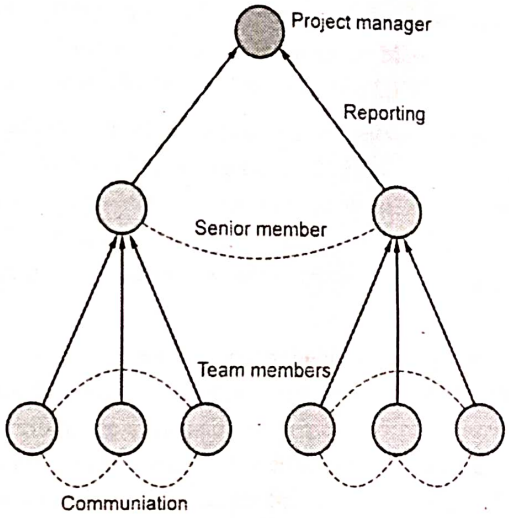


Fig. 5.11.5 : Mixed control team structure

Review Questions

1. What is organization and team structures ? Discuss different types of software development departments.
2. Discuss benefits of functional format in detail.
3. Differentiate between functional and project formats.
4. What is matrix format ? Explain with a suitable diagram.
5. What is team structure ? Discuss advantages and disadvantages of the chief programming team.
6. Explain democratic team and mixed control team structure.

5.12 Co-ordination Dependencies

- A useful classification of the coordination dependencies that are likely to occur in any significant organizational project has been constructed using a coordination theory. Here is a list of them.

- **Shared resources :** As an example, consider software development projects where various projects require the skills of specific sorts of technical professionals who are in high demand for specific project components. A project can be delayed if these professionals are unavailable due to commitments elsewhere.
- **Producer - customer ('right time') relationships :** A project activity can be reliant on a product delivery coming first. Before developing software components, a business analyst, for instance, could be required to create a document containing agreed-upon requirements. The PRINCE2 methodology's proposed Product Flow Diagram (PFD).
- **Task - subtask dependencies :** A series of related jobs must be completed in order for the main work to be finished. This might be reflected in the PFD, similar to the producer - customer interactions previously mentioned. Contrary to the producer-customer relationship, this sequencing is not determined by who gets to do what but rather by the technical requirements of the product being generated or the method being used.
- **Accessibility ('right place') dependencies :** This kind of dependence is more pertinent to occupations that call for travel over a wide area. A good illustration of this would be the challenges involved in dispatching an ambulance as rapidly as possible to the scene of a medical emergency. The instances in ICT and software development are less clear, although coordinating the delivery and installation of ICT equipment could be used to support the claim.
- **Usability ('right thing') dependencies :** This is a wider issue than the design of user interfaces, which is more typically linked to the term "usability," in this context. It has to do with the general issue of suitability for purpose, which also includes meeting organizational standards. This might result in activities like prototyping in software development that make sure the system being built will satisfy the operational requirements of the users. Additionally, it could involve change management procedures when users need to alter their requirements as a result of, say, occurrences in the corporate environment like legislation changes.
- **Fit requirements :** This guarantees that various system parts communicate and cooperate efficiently. One method of ensuring that these requirements are met is integration testing. Identifying whether modifications to one component will have an impact on other components is one aspect of configuration management.

- Information systems tools can frequently assist with these coordinating duties. For instance, project planning tools like Microsoft Project can be used to assist in decision-making regarding the allocation of resources across a portfolio of active projects as well as inside individual projects. Such tools can aid in the control of producer-customer and task-subtask

relationships by supporting the creation, analysis and manipulation of activity networks. Use of change management and configuration management databases to track changes to the system under development and hence support usability and fit dependencies is another form of software tool support.

Review Questions

1. Write a note on coordination dependencies.
2. Explain organizational projects using a coordination theory.

5.13 Dispersed and Virtual Teams

- As we have seen, projects need a team of individuals to complete them and each member of this team may be an expert in a specific sector. Collaboration in problem resolution is therefore at the core of many projects. The second world war emphasized the necessity of individual and group collaboration to carry out significant international activities and it promoted post-conflict research on efficient teamwork. At the time, group working essentially required that the team members be physically close to one another while they worked. However, the idea and practice of using scattered or "virtual teams" have gained popularity recently.
- We also need to understand how projects function. Large software projects, in particular, require coordination, which necessitates team communication. Being situated in the same physical area undoubtedly helps with this. Although software development requires communication, it also requires periods of focused, alone work. Offices can be noisy places. DeMarco and Lister refer to the state of intense focus required for productive creative work as "flow," and they contend that it takes around 15 minutes of continuous effort to reach this state. Every pause ruins the flow and it takes another 15 minutes for it to resume.
- As was said before, project managers frequently have limited influence on who is assigned to their team.
- They have even less chance of having any influence over the setting in which the team will work. According to studies conducted by IBM many years ago, a software developer should ideally possess :
  - 100 square feet of specific area;
  - 30 square feet or less of work space;
  - Partitions or covered offices that are at least six feet tall for noise protection.
- DeMarco and Lister discovered definite correlations between reported noise levels at the workplace and the amount of defects found in the produced software, despite the fact that it

is frequently challenging to provide this kind of accommodation for software engineers for a variety of reasons.

- Modern communication technology also make it easier for organizations to create temporary teams from inside their workforces to complete certain tasks without having to transfer their workforce. Because of the nature of some projects' work, there is sporadic demand for specific specialized talents.
- The project manager would want to have temporary access to these abilities that he or she could then release, saving money. A brief requirement for a graphic designer to create aesthetically acceptable graphics for a web application project can serve as an illustration of this. Because of the need for flexible labor, contract employees are frequently employed. These contractors can complete clearly defined duties at their own location using the internet, eliminating the need to travel to the client's location.
- The usage of "off-shore" staff who reside and work in another region of the world is then merely a matter of time. Thus, the dispersed or virtual team is reached.
- The following are potential benefits of this arrangement :
  - A reduction in staff costs due to the use of labor from undeveloped countries where salaries are lower.
  - A reduction in the overhead expenditures associated with having our own personnel on-site, such as lodging, social security contributions and training expenses.
  - Employees are used flexibly; they are not hired when they are not required.
  - Productivity might be greater.
  - Using specialized staff for particular tasks rather than more broad project workers could increase quality.
  - People working in various time zones can be utilized to reduce task duration. For instance, software engineers can send new code versions to testers in a different time zone so that they can test them and return the results at the beginning of the following working day.

- The following are a few difficulties with distributed working :
  - Carefully defining the requirements for work that is allocated to contractors.
  - Previously, practices might have been picked up through observation and imitation of coworkers on the job site. Now, procedures must be explicitly stated.
  - It might be challenging to coordinate a workforce that is distributed.
  - It could be necessary to change payment systems to fixed-price or piece-rate ones.
  - Employees that work remotely and are never seen might not be trusted.
  - Thorough evaluation of the provided products' quality is required.
  - Different time zones might interfere with coordination and communication.

Review Question

1. What are the benefits of the dispersed or virtual team ?

5.14 Communication Genres

- Research has been done to examine the properties of various communication methods. One strategy has been to look for communication genres. These are not simply talking about the technical ways of communication. These are the kinds of communications that individuals are used to having and where there are "ground rules" for when and how they should be conducted. These guidelines could be rather formal for some forms of communication, including formal meetings. There may be several different meeting styles, each with their own customs, that fall under the umbrella term "management meetings." These can be regarded as distinct genres.
- Advanced email-based apps can be created within the general heading of the email communication genre in which the email's content is organized according to predetermined proformas created to satisfy a standard process, such as asking for a change to a software specification. These applications might be thought of as distinct communication genres.
- The limitations of time and place have a significant impact on the character of communication genres. Same time/different time and same place/different place are two opposites that can be combined to form multiple modes of communication (see Table 5.14.1).

	Same place	Different place
Same time	Meeting	Telephone
	Interviews etc	Instant messaging
Different times	Notice boards	E-mail
	Pigeon-holes	Voicemail
		Documents

Table 5.14.1 : Time / place constraints on some communication methods

- It's also important to take into account the type of information being conveyed.
  - How extensive and intricate is the information that needs to be communicated ? A phone call can be a rapid and efficient means to communicate short messages, but it is not the best choice for exchanging detailed information.
  - Is it simple to comprehend ? For instance, are both the sender and the recipient aware of the context ? A two-way communication method would be preferable if it is expected that the recipient may seek clarification of certain informational elements.
  - Face-to-face contact is the most effective mode of communication when it comes to matters that are particularly sensitive, even though it may be uncomfortable or inconvenient for those involved.
- Different communication styles will probably be preferred at various stages of a project.

1. Early stages of a project

- Team members will need to gain each other's trust and confidence early on in the project. It is definitely ideal to really meet for this conversation at the same time and location. According to Charles Handy, "paradoxically, the more virtual an organization becomes, the more its personnel need to interact in person." It is asserted that this is especially true for distributed initiatives.
- Nevertheless, not everyone might agree with this. In his account of co-authoring a philosophy textbook with a person he had never met, Julian Baggini makes the following claim : "It is simply not required to know the full person in order to have a good relationship with them.
- At least a portion of the team will be involved in making decisions during the early phases of a project regarding the general design of the product to be delivered and the procedure to be followed in its creation. Same time/same place meetings are the most efficient way to advance, at least in the early phases.

2. Intermediate design stages of the project

- Once the product's broad architecture has been determined, it is possible to work concurrently on the detailed design of various components in various places. However, some details will need to be addressed and for this, same-time, different-location contact, like teleconferencing, may be best.

3. Implementation stages of the project

- Work can move forward after the design is clear and everyone is aware of their responsibilities. At this point, communication tools like email that allow for information interchange across time and space are likely to be enough.
- Even at this point, some advise holding regular face-to-face meetings with at least a portion of the workforce since it gives the project a rhythm, aids in coordination and keeps employees motivated. The workings of a distributed project were examined by Martha Maznevski and Katherine Chudoba, who reported that "interaction between coordinating sessions was mostly in response to the previous meeting or in anticipation of the next one." The team's procedures received a rhythmic infusion of new energy at the coordination meeting.

Review Questions

1. What are communication genres ?
2. What are different communication styles for project ?

5.15 Communication Plans

- While communication is necessary for all projects, it becomes especially crucial for distributed initiatives. Because of this, the project planning process should take into account how project stakeholders will communicate. Some people have even gone so far as to say that a specific planning document should address communication challenges that affect the project, not just for dispersed projects but for every project with a sizable number of significant players.
- The identification of stakeholders and their concerns was underlined earlier, therefore the first step in creating such a plan is to make a list of the major stakeholders, paying specific attention to those involved in the project's creation and implementation.
- Each of the key tasks and milestones is examined once the overall project plan has been developed using a procedure to determine which communication channels and techniques will be most effective with these stakeholders. In our examination of the communication

requirements of the various phases of a scattered project, we have already gone through this process. A crucial step in this approach would be consultation with the representatives of the various stakeholder groups.

- These column titles could be used in a table to list the results of this method.
  - **What :** This contains the name of a specific communication channel or event, such as "project intranet site," for example, "kick-off meeting."
  - **Who/target :** The communication's intended audience. 'Target audience' might not completely express the intended meaning because it suggests that there are informational passive recipients. In fact, the communication channel or event may be used to gather data from the "audience."
  - The goal, the **purpose** of the communication.
  - **When/frequency :** If the message refers to a single event, a precise date can be provided. The frequency should be stated if the event is a recurrent one, such as a progress meeting.
  - **Type/method :** The type of communication, such as a meeting or a document that was distributed.
  - **Responsibility :** The one who starts the conversation.

Review Question

1. Explain communication plans ?

5.16 Leadership

- One of Amanda and Brigette's private concerns may have been that personnel wouldn't take them seriously when they first took on project management duties.
- Leadership is typically understood to entail the capacity to persuade members of a group to act in a certain way in order to accomplish shared objectives. Since managers also have other responsibilities like organizing, planning and controlling, they are not necessarily effective leaders or vice versa.
- It has proven challenging for experts in the field to come up with a list of the traits that all effective leaders share.
- However, it would be safe to conclude that they appear to have a larger need for success and power, as well as more self-control and self-confidence than other people.

- Although leaders don't always have a lot of official authority, leadership is founded on the concept of power or authority. A person's position (position power), personal traits (personal power) or a combination of the two may provide them influence. Additional analysis of position power has revealed :
  - **Coercive power :** The power to force someone to act by threatening punishment;
  - **Connection power :** Which relies on having access to those in positions of power;
  - **Legitimate power :** Which is founded on the idea that a person's title bestows a unique status;
  - **Reward power :** Where the holder can reward those who do duties to their pleasure.
- On the other hand, personal power can be further broken down into :
  - **Expert power**, which results from being capable of performing a specific task;
  - **Power of information**, which grants the holder exclusive access to information;
  - **Referent power**, which is based on the leader's allure personally.

5.16.1 Leadership Styles

- It's possible that Amanda and Brigette are initially worried about establishing their own power. The necessity of involving the staff in decision-making in order to make the best use of expertise and to garner commitment balances this. When they need to be dominant and insist on something, Amanda and Brigette will need to know when to be more tolerant and flexible. When creating plans, Amanda, for instance, might opt to be extremely democratic. However, once the plans have been agreed upon, Amanda might decide to insist on a very disciplined implementation of the plan.
- On the other side, Brigette might discover at Brightmouth College that she is the only person with the technical knowledge to make some decisions, yet after she has briefed the staff, they anticipate being left alone to do the task as they see fit. Leadership styles have been evaluated along two axes : Autocratic vs. democratic and directive vs. permissive.
  - **A controlling autocrat :** Makes decisions alone and closely monitors their execution;
  - **Permissive autocrat :** Makes judgments alone and gives discretion to subordinates to carry them out;
  - **Directive democrat :** Participates in decision-making and closely monitors execution;
  - **Permissive democracy :** Takes decisions collaboratively, gives discretion to subordinates in carrying them out.

- The degree to which a manager is task-oriented or the extent to which the accomplishment of the current work is of the utmost importance and the degree to which the manager is concerned about the people around them, has been another axis used to measure management styles (people orientation). Perhaps it is not surprising that employees seem to perform at their highest levels under supervisors who rank well in both categories.
- Workplaces differ depending on how much control employees have over their jobs. Some professions are predictable and routine (e.g. dealing with batched computer output). Others are influenced by outside forces (such as a help desk) or include uncertain future directions (e.g. at the early stages of a feasibility study).
- When there is a lot of uncertainty, subordinates will look to the top for direction and appreciate task-oriented management. As the amount of uncertainty decreases, the task-oriented manager is likely to unwind and become more people-oriented, which will produce positive outcomes. Where employees may handle their own work without consulting their line managers, people-oriented managers perform better. The argument is then made that if maintaining control becomes even simpler, the people-focused manager will be enticed to become more involved with task-focused issues, leading to unfavorable outcomes.
- Research has also shown that a task-oriented strategy works best when the team members are relatively inexperienced. Consideration for each individual's needs and goals becomes in importance as the group's members develop. There is no need to place a heavy emphasis on either of these strategies in situations where maturity is very high.

Review Question

1. What is leadership ? Explain leadership styles in detail.

5.17 Case Study : Team Building in Project Management

Why team building is important ?

- It is our duty as the project manager to make sure that everyone contributes to achieving the objectives of our project. Teams that have trouble cooperating find it difficult to achieve this, either as a result of member discord or a lack of direction. As a result, projects aren't finished on time, the quality of the job starts to slip and management and team members frequently fail to communicate effectively. The most cutting-edge project management tools and enough resources hardly ever prevent a project from failing due to inadequate team development.

- Typical obstacles to team development include :
  - Team members' varying interests, priorities and professional objectives.
  - Disputes brought on by unclear team roles and responsibilities.
  - Uncertain project goals.
  - Instead of cooperating, team members compete for leadership roles.
- As we might imagine, these obstacles can result in a variety of long-term issues that hinder team productivity and lower the standard of work. As a result, it's crucial for project managers to build a solid team that gels well and supports one another.

Team members' varying perspectives, priorities, interests and judgments

- The fact that team members frequently have varied professional goals and interests is a significant impediment. However, team members frequently need to put the interests of the project ahead of their own in order for the project to be successful. When team members are unwilling to do this, serious issues arise when forming an effective team. When the team depends on support organizations with diverse aims and interests, the issue is exacerbated.

Role conflicts

- Role conflicts between team members can sometimes obstruct efforts to develop the team. When there is confusion about who does what within the project team and between the team and external team support organizations, role conflicts are more likely to arise. Role conflicts are also greatly influenced by overlapping and unclear role obligations.

Project objectives / Outcomes not clear

- Project objectives that are not clear are one of the most commonly mentioned obstacles to team building.

Environments for dynamic projects

- The environment in which many projects function are always changing, which is one of their common characteristics. Senior management, for instance, might frequently alter the project's goals, scope and resource allocation. In other cases, regulatory adjustments or customer requests for new and different standards might have a significant impact on a project team's internal processes. Project teams usually operate in disruptive situations. Finally, there may be impediments to team formation depending on how quickly a team "builds up" to its full manpower base.

Teams compete for leadership

- The number of project leaders who cited competition for a leadership position initially caught us off guard. They stated that this barrier was most likely to appear in the beginning of a project or if the project had significant difficulties and the caliber of team leadership was called into doubt. Undoubtedly, both types of leadership challenges can create impediments to team building, albeit temporary ones. These difficulties were frequently subtle tests of the project manager's skills.

❑❑❑

Unit VI

6 Applications of Software Project Management in Industry

Syllabus

Agile Project Management with Azure DevOps : An Overview of Application Lifecycle Management and Azure DevOps, Traceability, Visibility, Collaboration and Extensibility. Difference between Microsoft TFS and Azure DevOps.

Metrics in Agile Practice : Introduction to Metrics inAgile Practice, Metrics for Project Management, Agile Project Management in Azure DevOps and TFS.

Case study : Online Shopping System.

Contents

6.1 An Overview of Application Lifecycle Management and Azure DevOps

6.2 Overview of Azure DevOps

6.3 Traceability

6.4 Visibility

6.5 Collaboration

6.6 Extensibility

6.7 Difference between TFS and Azure DevOps

6.8 Metrics in Agile Projects

6.9 Agile Project Management in Azure DevOps and TFS

6.1 An Overview of Application Lifecycle Management and Azure DevOps

6.1.1 Introduction to Application Life Cycle

What is Application Lifecycle Management ?

- Application Lifecycle Management (ALM) is an integrated system of people, tools and processes that oversee a software application from its initial planning and development, through testing and maintenance, to decommissioning. By combining and organizing lifecycle elements, ALM improves product quality, optimizes productivity and facilitates the management and maintenance of related products and services.
- ALM tools automate software development and deployment processes, help ensure compliance is achieved and maintained and create a standardized environment where all teams involved in the application lifecycle can communicate and collaborate.

6.1.2 Why is ALM Important ?

- ALM helps companies set and meet appropriate project requirements. ALM also improves the development process by incorporating frequent and thorough testing. It also helps developers adjust development processes and goals throughout the software lifecycle. ALM helps ensure that all teams including development, operations and security can work together effectively to produce the best software possible.
- In addition, leading software companies deploy product updates daily. Application lifecycle management helps businesses achieve high efficiency and gain competitive advantage by accelerating workflows and ensuring the highest quality products are deployed.

6.1.3 What is ALM used for ?

- ALM provides a framework for defining requirements and establishing processes, controls and methodologies prior to software deployment. ALM thus provides the scaffolding within which software is developed, tested and maintained. By design, it also includes safeguards and controls to ensure that software meets compliance, governance, efficiency, usability, performance and other benchmarks before it goes into production. ALM provides ongoing opportunities to review and adjust costs to meet changing budget requirements and productivity metrics to ensure companies realize their software development ROI goals.

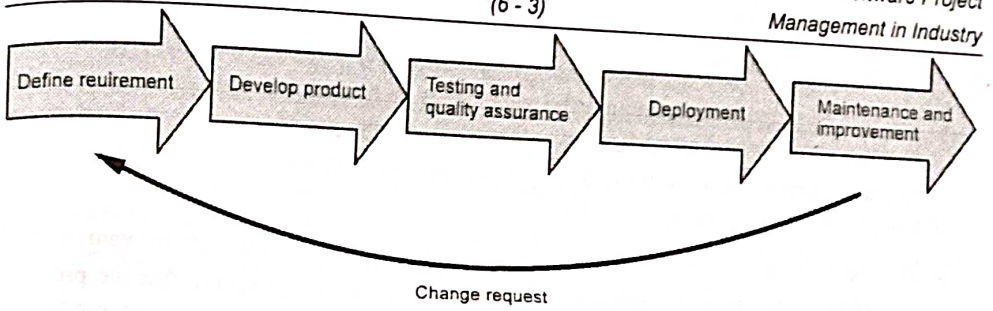


Fig. 6.1.1 : ALM process

6.1.4 What is the ALM Process ?

- The ALM process provides DevOps teams and executives with ongoing opportunities to :
  - Establish and manage requirements and establish processes and procedures;
  - Set up and manage lifecycle governance and compliance needs;
  - Establish methodologies for managing and controlling development, testing and maintenance activities; and
  - Ensure testing meets functionality, performance, usability and security needs.

6.1.5 ALM Phase

- Application lifecycle management consists of five phases :
  - Defining requirements
  - Product development
  - Testing and quality assurance
  - Development
  - Continuous product maintenance and improvement

1. Defining requirements

- When defining requirements, all stakeholders come together to declare what they need from the application to support their business cases. The application design is created based on their expressed needs. Requirements can include a number of factors, from the business needs of stakeholders to compliance and governance requirements.
- Requirements specification usually proceeds from the top down, meaning that the needs start with the most general and move to the more specific and detailed. As a result, case requirements are often in a hierarchical tree structure, with each node representing a more specific sub-requirement for a more general parent node. However, other development approaches, such as the iterative agile development process, use a less hierarchical structure for the requirements list, with defined needs identified as use cases.

2. Product development

- Product development begins once the team agrees on the requirements. In this phase, the product moves from an idea and design to a real, working application. At the beginning, the development team must divide the application requirements into parts and phases to create a development plan.
- During this time, it is beneficial to involve representatives from all relevant teams, including sales, product marketing, IT and testing. This helps ensure that the product created meets all defined needs and is easy to use, test and deploy.
- A wide variety of development methodologies can occur during this phase. The most popular are sequential - such as the waterfall model - or iterative - such as agile development.

3. Testing and quality assurance

- Testing and quality assurance often overlap with the development phase. Testers should start preparing their test cases and test environment before the official release of the product. Testers should also be available to provide feedback on the application during development. In addition, integration and unit tests should be incorporated into programming activities. Development teams often use continuous integration systems.
- During the formal testing and quality assurance phase, testers must verify that the application meets the requirements defined in the first phase of the process. Testers should also review any other stakeholder expectations that the application will need to support during its lifecycle. This phase also includes full integration testing and any issues or bugs found and reported can be addressed by the development team.
- The development and testing phase ends when the product reaches quality and stability good enough for release. This level is defined by the product marketing team.

4. Deployment

- The deployment phase involves releasing the product to users. This process varies depending on the type of application, as each type of product requires different properties and specifications. For example, software-as-a-service (SaaS) applications must be deployed on a company's internal servers, while users can access web applications over the Internet.

5. Continuous maintenance and improvement of the product

- After deployment, the product is continuously maintained and improved to monitor and manage the performance of the released application. During this phase, the team resolves any remaining bugs while planning and prioritizing new updates.

- Maintenance is often the longest phase of application lifecycle management but may also require the least involvement of the development team if the previous steps have been effective.
- An important element of the maintenance phase is defining system decommissioning. In other words, teams must decide when work should be stopped and moved to a newer version of the product or migrated to another product entirely.

6.1.6 Benefits of Application Lifecycle Management

- Some key benefits of ALM include :
  - Better overview of the workflow
  - Better compliance
  - Faster deployment
  - Higher quality products
- An integrated system created by ALM is more efficient than a collection of unrelated tools and processes spread across different teams. This integration also benefits organizations by improving communication and collaboration and aligning software goals with any business value or corporate goal.
- The ability of teams to work together ensures that every worker understands the project and its phase. ALM tools allow workers to track strategies, changes, requirements and project status in real time, regardless of their location. ALM tools also prioritize different team goals and help define different skill sets needed for different processes.
- In addition, application lifecycle management provides an organization with clear direction for its workflow before developers begin building the application. The first phase - defining requirements - allows companies to develop a business case, determine the application's lifecycle and plan for the necessary resources before embarking on development. This saves the organization time and money by avoiding unnecessary work and costly mistakes.
- ALM also improves teams' decision-making abilities when dealing with aging software. Most ALM tools include version control and real-time planning, allowing team leaders to easily map out the future of the application. This capability can also eliminate confusion for companies dealing with multiple applications.

6.1.7 ALM Tools

- A set of project management tools that integrate people and processes, called application lifecycle management tools. A number of ALM tools are available to track application changes.

- These tools range from specialized ALM products that monitor an application from start to finish, automatically sorting files into logical segments, to simple wikis that require team members to record changes manually.
- Organizations should look for a few key features when choosing an ALM tool :
  - Version control;
  - Real-time team communication and planning;
  - Estimating and planning;
  - Requirements management;
  - Test management and quality assurance;
  - Source code management;
  - Automated deployment;
  - Application portfolio management; and
  - Maintenance and support.
- The ALM tool allows users to define project requirements and develop user stories, which can then be prioritized, scheduled and broken down into detailed tasks used to track resources. Resource monitoring analyzes how well an organization uses its resources throughout the application lifecycle. ALM users can also attach documents, screenshots and URLs to all artifacts and customize all charts and reports in a variety of formats - including Adobe Acrobat and HTML.
- ALM tools allow users to create, edit and execute test cases; manage automated and manual tests; track issues, bugs, risks and improvements related to the source code repository; and access to a full audit history of all changes made to the application. ALM toolbars can be personalized and the messages that appear can be tailored to best benefit a particular user.
- Popular examples of ALM tools include the following :
  - Jama software
  - MeisterTask
  - Codebeamer
  - Vision
  - Jira
  - Microsoft Azure DevOps
  - Tuleap

6.1.8 ALM vs. SDLC

- The software development life cycle (SDLC) refers to the processes or procedures involved in creating a high-quality software product. Application lifecycle management is similar to SDLC, but includes a larger variety of processes.
- While ALM covers the entire application life cycle, SDLC focuses only on the software development process. In other words, ALM includes all five phases of the application lifecycle requirements, development, testing, deployment and maintenance but SDLC includes only one phase development.
- As a result, application lifecycle management includes the software development lifecycle, but SDLC only focuses on the ALM fraction.

Review Questions

1. What is application lifecycle management ? Why it is important ? Why it is used ?
2. Explain application lifecycle management process with its phases.
3. Explain ALM with suitable diagram.
4. What are the benefits of application lifecycle management ?
5. Define ALM tools. What features should be considered while choosing an ALM tools ? List some examples of ALM tools.
6. Compare ALM and SDLC.

6.2 Overview of Azure DevOps

- Azure DevOps is a software-as-a-service (SaaS) platform that provides DevOps practices and tools for the complete software lifecycle. Azure DevOps is not limited to internal tools, but can be integrated with most other top DevOps tools. Below are the various services provided by Azure DevOps :
  - **Azure Boards** - Agile planning, work item tracking, dashboard visualization and reporting.
  - **Azure Pipelines** - Language and platform agnostic CI/CD with inclusion of containers and Kubernetes.
  - **Azure Repos** - Cloud hosted GIT and TFVC repositories.

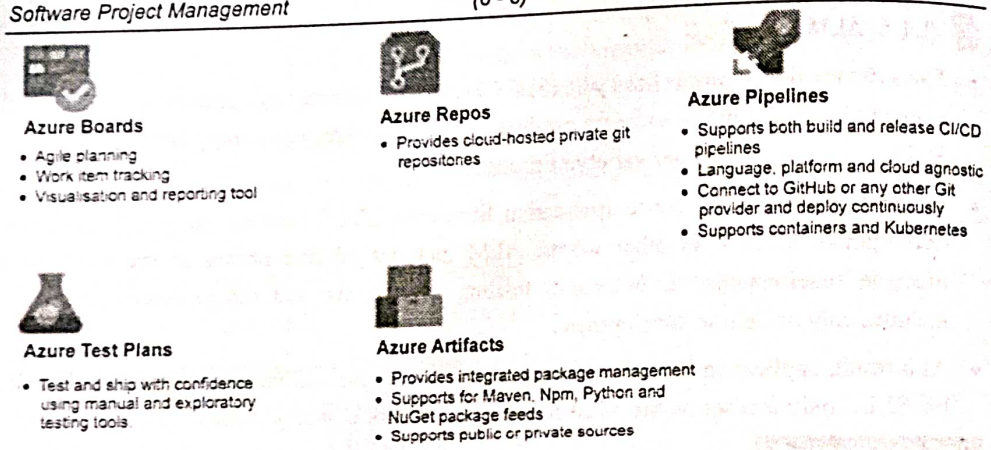


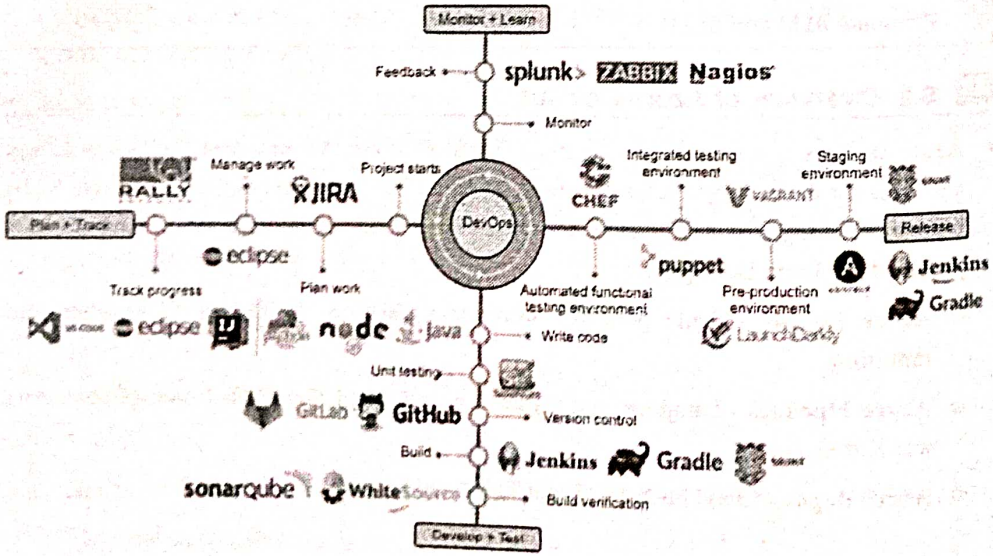
Fig. 6.2.1 : Azure DevOps services

- **Azure Artifacts** - Integrated package management with support for Maven, NPM, Python and NuGet packages.
- **Azure Test Plans** - Providing an integrated testing solution including manual and exploratory.

Migrating or integrating existing work management tools

- Azure DevOps can be integrated with a wide variety of existing work management tools :

Integrations and Extensions



Azure DevOps Server :

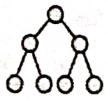
- TFS and Azure DevOps have come a long way to fulfill ALM vision, but it does not cover everything. As mentioned, VSTS now has renamed Azure DevOps; TFS 2018 will be renamed Azure DevOps Server. The former VSTS functions are now separate services in Azure DevOps.

1. **Azure Repos -** Source control and code storage
   - Azure Repos is a set of version control tools that you can use to manage your code. Azure Repos provides two types of version control :
     - **Git :** Distributed version control
     - **Team Foundation Version Control (TFVC) :** Centralized version control

Note : Git is the default version control provider for new projects. You should use Git for version control in your projects and start moving your existing TFVC projects to Git. TFVC is considered a completed function. Azure DevOps will retain compatibility with TFVC, but Git will acquire all future investments.

Types of source control systems - Centralized vs Distributed


Centralized


Distributed

- There is a single central copy of your project and programmers commit their changes to this central copy
- **Examples** : TFVC, Visual Source Safe. Assembla Subversion (or SVN) and Perforce

- Every developer clones a copy of a repository and has the full history of the project
- Commits are extremely fast
- **Examples** : Mercurial, Git and Bazaar

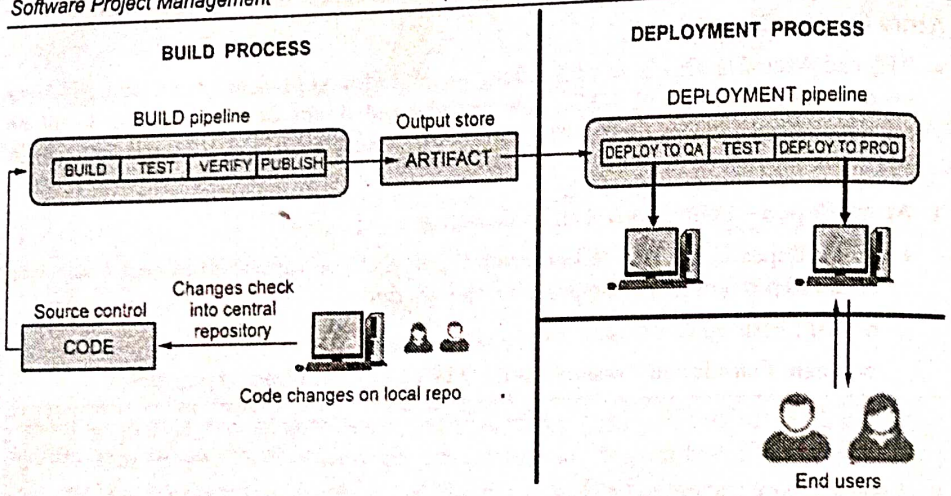
2. **Azure Pipelines -** Continuous integration and continuous delivery

Continuous integration :

- Automatically make sure you're not sending broken code
- Run tests continuously
- Increase your code coverage
- Build faster by splitting test and build runs

Continuous delivery :

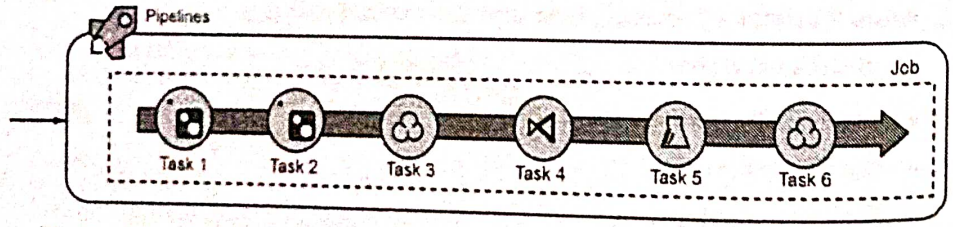
- Automatically deploy code to production
- Make sure your deployment targets have the latest code
- Use tested code from CI process



Azure Pipeline

- Build Azure pipeline with two options
  - YAML pipeline (yaml file)
  - Classic pipeline with interface editor
- Azure pipeline is a workflow for automated tasks
- Azure pipeline is a collection of jobs. A task is a set of tasks
- Azure pipeline consists of steps called tasks, so we map scripts to tasks

Azure DevOps Pipeline



- Pipeline jobs will need an agent (server) for computing resources to run guest agents.
- There are two types of agents : Microsoft-hosted agents and self-hosted agents.

1. Microsoft agents

- Maintenance and upgrades are taken care of for you
- Fresh VM every time you run a channel

- Windows Server with Visual Studio, Ubuntu, MAC
- Not all build software is available in the Microsoft-hosted agent. For those that don't exist, in Azure Pipeline you will first need to install the software and run the jobs.

2. Self-service agents

- As a customer, you have provisioning and management - download the Azure pipelines agent and install.
- It gives you more control when installing any dependent software that is needed upfront. Azure pipelines can only run code without reducing runtime by not running additional software installation tasks for the build.
- Working without agents.
- It does not require a target calculation to run.

Supported Tasks :

- Postpone the task
- Invoke an Azure function task
- Invoke a REST API task
- Manual verification task
- Publish task to Azure service bus
- Query the Azure monitor alerts task
- Query the work items of a task

Azure Boards

- Azure Boards is an interactive and customizable project planning and management tool. It supports agile, scrum and Kanban processes, as well as dashboards and reporting.

Why use Azure Boards ?

- Predefined work item types to track features, user stories, bugs and tasks
- Highly interactive and visual tools
- Easy customization
- Built-in discussions and communication
- Generous cloud storage
- Notification of changes

- Built-in dashboards and analytics for health monitoring
- Office M365 integration
- Third party extensions
- In addition to all of the above, you can start using Azure Boards for free with up to five users and unlimited stakeholders.

Azure Artifacts

- Azure Artifacts enable developers and teams to share packages of code from different sources and public registries. Azure Artifacts support several package types :
  - NuGet
  - E.g
  - Python
  - Maven
  - Universal packages

Note : Azure Artifact storage is pay-as-you-go and free up to 2 GB.

Azure test plans

- Azure test plans is a browser-based test management solution that provides all the functionality needed for planned manual testing, user acceptance testing, exploratory testing and stakeholder feedback.
- You can :
  - Create test plans and test suites
  - Manage test plan execution settings and configurations
  - Run tests on any platform (Windows, Linux, Mac) with Test Runner
  - Create charts with different pivots such as priority and configuration to track test progress
  - Browse test results
  - Export test plans and test suites for review
  - Azure DevOps security and compliance

- Safety is everyone's responsibility and is an ongoing process that should include :
  - Infrastructure
  - Architecture design with high security priority
  - Continuous security checks
  - Continuous monitoring of foreign threats

Azure DevOps storage and pipeline security :

What needs to be scanned ?

- Application code
- Infrared code
- Images of containers
- Configuration code

What to evaluate ?

- Hard-coded secrets
- Application package security vulnerabilities
- Code quality issues
- Embedded malware

What languages and platforms are supported ?

- Application code - C#, Java, Node, Python
- Infrastructure as code - ARM, Terraform, Cloud Formation, Dockerfile, Kubernetes
- Container images - Linux or Windows, Files or Applications
- Below are a few third-party tools integrated with Azure DevOps to help with overall application and infrastructure security compliance :

Application code analysis

- SonarCloud
- SonarQube
- WhiteSource Bolt
- Checkmarx
- Kiuwan

- Veracode
- Security scan Snyk
- ShiftLeft Inc
- Infrastructure code analysis
- Secure DevOps Kit
- ARM Template Test Kit
- Checkov/Bridgecrew

Container security tools

- Trivvy
- Clair
- Twistlock (Prisma Cloud)
- Aqua Sec

Conclusion

- Azure DevOps has paved the way for faster and more agile software development processes by unifying teams, processes and technologies to create a continuously evolving software development lifecycle (SDLC).

Review Quesitons

1. Define Azure DevOps. What different services Azure DevOps provides ?
2. Write a short note on: Azure pipelines
3. Explain different types of agents in Azure DevOps pipeline.
4. What is Azure board ? Why we use Azure boards ?
5. What are artifacts in Azure ?
6. Explain Azure test plans.
7. Explain the Azure DevOps in detail.

6.3 Traceability

- Having traceability in your DevOps processes is key to successfully delivering and maintaining your applications and systems. I once visited a company that stopped making changes to their systems simply because no one ever knew where the change (or bug fix) might have an impact. You don't have to deal with such a situation.

- Azure DevOps features can help you with traceability, so you can avoid such problems :
  - Work item tracking
  - TDD/unit testing
  - Azure pipelines
  - Check-in policy
  - Version control system
- Let's look at some of the specifics associated with these features, starting with how the work item tracking system implements traceability.

6.3.1 Azure DevOps Work Item Tracking System

- Sometimes you may have a lot of post-it on your monitors and desk, each with at least one task that you should tackle. You'll most likely want to track these tasks with a tool that can help you, but often that's just not possible. It is possible that some tasks are associated with one project and others with another. You may have tried recording them in an Excel spreadsheet and saving them to your computer. However, you soon realize that the table is located on your laptop, your customer's computer, your desktop, another customer's computer, etc. on. And you have no idea which table is the current version. The same often happens with projects. PMs have their to-do lists for the project and they all have their own way of keeping them updated. Let's say a PM uses Excel to track tasks - the status of tasks, who they're assigned to and so on. How can PMs keep the team updated on the latest to-do list ? If the PM chooses to send it via email, it's likely that some team members won't save the new version to disk or will miss it in the endless stream of incoming emails. Soon there are different versions floating around and things are generally a mess.

6.3.2 Work Items

- With Azure DevOps, you have a task tracking system. At the heart of this system are the tasks themselves, which, as we said earlier, are called work items. The work item can be pretty much whatever we want it to be. It can be a bug, a request of some kind, a general task and so on. Each work item has a unique ID that helps you keep track of the locations it references (Fig. 6.3.1). An ID allows you to track a work item say, a request from its creation to its implementation as executable software (a component). Work items are a great way to simplify project task management while allowing for traceability. No more confusion about which version of the to-do list is current; manual work is no longer needed to collect work status reports that are only used in steering group meetings. Now you have a solution that makes it easier to collaborate with your teams and allows all members and

stakeholders to view status messages whenever they want. You can also more easily collaborate with people outside the project team by adding work items through the web client.

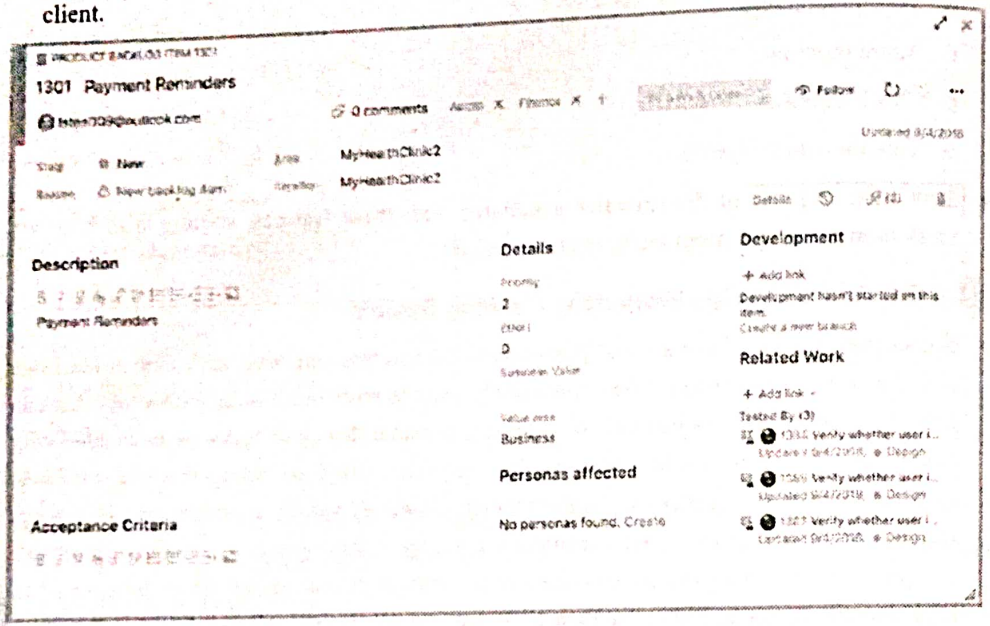


Fig. 6.3.1 : Each work item has a unique ID in this case, 1301

- Azure DevOps is so flexible in this regard that it allows us to customize the work items as we want them. The work item tracking system is one of the core components of Azure DevOps. This system allows you to create work items or units of work and can be used to enable traceability. You can use the work items included in Azure DevOps from the start, customize them to suit your needs or even create your own work item types. Each work item instance has a unique ID that you can attach to the things you do in Azure DevOps. This allows you to track a single work item say, a request from its creation to its implementation as an executable (component). You can also associate one work item with others to create a hierarchy of work items. Work items can contain information in various fields that define the data to be stored in the work item. This means that each field has a name and a data type.
- All work items can have different information attached to them. You can have information about who the work item is assigned to and the current status of the work (for example, the bug can be open, closed, investigated, resolved, etc.). The status field can be modified so that each work item type has its own status mechanism. This is logical because the error is

likely to go through states different from those that the general task goes through. You can also attach documents to a work item and link one work item to other work items. You can even create a hierarchy of work items if you want. Let's say you're implementing a request as a work item and that request contains many smaller tasks. You can then have the request itself at the top and nest other requests below it so you know which work items belong to which request.

- For example, when a bug is discovered, you can quickly trace the original request by its work item ID and see where in the code you need to make some corrections. You can also visualize the associated work items to evaluate whether other parts of the code need to be changed as a result of the bug fix.
- Azure DevOps stores work item information at the data layer, which helps you track the change history of a work item. You can see who created it, who solved it, who closed it and so on. Information in databases can be used to display in reports, allowing you to customize them to suit your needs. For example, a single message can display the status of all errors. Stakeholders can see how many open bugs there are, how many are resolved and much, much more. How you use the work items is entirely up to you.
- If you implement a request as a work item, you can use the work item ID to track that request through the source code to the final executable build. You can enable this traceability by requiring all developers to add one or more work item IDs to a commit using a commit policy.

6.3.3 Configuration Management with Azure DevOps

- In every (development) organization there must be version control of systems in production. If you don't have this, the overall DevOps process suffers because you suddenly lose traceability. This makes it difficult to implement changes and bug fixes because you won't know which versions need to be updated.
- Without the help of a proper tool, you will soon get lost in the variety of applications you have. Azure DevOps can help you with version control in a number of ways. After a brief description of software configuration management, I cover some of the most important concepts that are heavily supported in Azure DevOps and Visual Studio :
  - Version control
  - Azure pipelines

- In software engineering, the task of software configuration management (SCM) is to track and control changes in software. Configuration management practices include revision control and establishing baselines and are very important. There are several objectives of SCM, including the following :
  - Configuration identification : Making sure you know the code you are working with
  - Configuration management : Release management a product and its changes (versions)
  - Build management : Process and tool management used for construction
  - Defect tracking : Make sure every defect has traceability back to the source
- If your DevOps process doesn't cover these issues, you could find yourself in trouble very soon. It is crucial for development teams to have full control over which versions of applications exist, which are in production and where they are in production.

6.3.4 Version Control and Release Management in Azure DevOps

- Using the version control system in Azure DevOps, you can manage and control multiple revisions of the same information in your projects. This information can be source code, documents, work items and other important information that you want to submit to version control. When you want to work on an item under source control, you send it to your local machine to start working on it. When the work is done and tested, you commit the changes to update the version on the server.
- Azure DevOps version control features are powerful. Microsoft supports Git and native Visual Studio version control. They are fully integrated into the GUI, which is also something that ALM prescribes. If you want, you can also access some features from the project portal. Many people want to use the command line for their work and Azure DevOps allows them to use the command line for version control work as well.
- However, if you want to use Visual Studio to access the Azure DevOps version control system, you can do so. Azure DevOps extensibility makes this possible. One example is a set of TEE client applications that have access to Azure DevOps, including a version control system. TEE allows you to access Azure DevOps from Eclipse on Mac OS X and also via command lines. This way, you can more easily integrate different development platforms into an Azure DevOps project. You will still use the Azure DevOps repository and be able to get reports and other information directly from Azure DevOps.

6.3.5 Build Management with Azure Pipelines

- Building is basically the process of taking the source code and all the other items necessary in the application and building them into executable software. Azure pipelines is a cloud service that you can use to automatically build and test your code and make it available to

other users. Works with almost any language or project type. Pipelines combines CI and CD to continuously and consistently test your code builds and deliver them to any target.

- CI is about test and build automation for your project. It helps catch bugs or issues early in the development cycle, making them easier and faster to fix. Items known as artifacts are produced from CI systems and used by CD release channels to control automated deployment.
- CD is about automatically deploying and testing code in several stages to increase quality. CI systems produce deployable artifacts, including infrastructure and applications and then automated release pipelines consume those artifacts to release new versions and fixes to a target of your choice.
- There are many reasons to use Azure pipelines for your CI/CD solution :
  - Works with any language or platform.
  - On different types of targets simultaneously it is deployed.
  - It is the best integration experience to deploy to Azure.
  - You can build on Windows, Linux or Mac computers.
  - Provides great integration for GitHub.
  - It's a great deal for open-source projects.

6.3.6 Process Automation at a High Level

- Without one or more templates, Azure DevOps is not being used to its full potential, as mentioned earlier. You can still use its version control system and some other tools, but the real value comes from using Azure DevOps to automate your DevOps process. Using a process template, your entire DevOps process is defined.
- The template defines the following :
  - **Work item types :** Refer to these types of work items which are necessary and the information which should be attached to them. You can also define a workflow for a work item. In the event of an error, you can have different states that the item goes through, such as active, resolved, closed and so on.
  - **Project phase :** Using areas and iterations you can define the initial setup of the project phase of your projects. If you use RUP, you can define process phases in this model or you can create the first sprints of a Scrum project. Areas and iterations are flexible and you can create your own way of working with these concepts.

- **Document structure and templates :** The number of documents that should be created during a project varies depending on your process model. In the process template, you define the document structure you need and the templates you should use. For example, you can add templates for requirements specifications or acceptance tests.
- **Messages and queries :** In the process template you can specify which work item messages and queries you need to have by default in your projects. You'll probably want messages and queries that show the progress of your project, such as bug status or remaining work. You can create your own reports using Power BI, SQL Server Reporting Services (TFS 2018) or Excel and then add them to all projects by editing the process template.
- **Security :** The template also adds information about which users or groups of users have access to which information. For example, you can connect Azure DevOps groups to your Active Directory accounts.

- The process template is the overall process of our ALM implementation. Many of our customers create different templates for different kinds of projects. They also create templates for operations so that after a development project is complete and deployed, operations staff can use the template to run the system until the system dies. For example, several customers have started to create an Information Technology Library (ITIL) template and we look forward to the outcome of this work.
- It's important to note that you can customize the process to suit your needs. You should consider changing the default templates or even replacing them, rather than adapting your own way of working to the templates that come with Azure DevOps out of the box. Microsoft enables this flexibility by giving you easy access to process templates to modify them or add new templates.

Review Questions

1. What is traceability in DevOps ? What problems can be avoided using traceability feature of Azure DevOps.
2. What contents are define in template of DevOps process ?

6.4 Visibility

- Project status information is important to all project participants – and we don't just mean team members, but also stakeholders and decision makers. As project managers, we spent

too much time searching for information to answer questions about the status of projects, how much work is left and what the latest bug status is.

- Azure DevOps provides three main ways to enable visibility :
  1. **Widgets and dashboards :** Customizable, highly configurable dashboards give us and our teams the flexibility to share information, track progress and trends and improve our workflows.
  2. **Queries :** Queries are used to query the work item tracking service. Some questions might be : How many work items do we have with errors ? How many and which ones are dedicated to me ? How many errors are there ? And so on. You can create new queries if needed.
  3. **Power BI :** Integrating analytics with power BI makes it easy to get data into power BI so you can focus on creating power BI reports.
- These components make it easier to gather the information you need for status reports for a steering group meeting or project meeting. You will no longer have to look around in several places and in several applications for this information; instead, you can use automated reports and queries from within Azure DevOps.
- Project owners, PMs and Scrum masters certainly benefit from Azure DevOps. Because Azure DevOps has all the data in the same repository, you can more easily retrieve the right information when you want it. The flexibility of a SQL server database that stores all the information is great. You can work with data warehouse information just like any other database.
- You can publish using the Azure DevOps site (Fig. 6.4.1). Information (in the form of custom controls that users cannot change at this time) so that anyone with permission can see it. It's an easy way to ensure that information is always available. It is this small, relatively non-technical improvement that eases the PM's job, freeing up the PM's or Product Owner's time for better things.

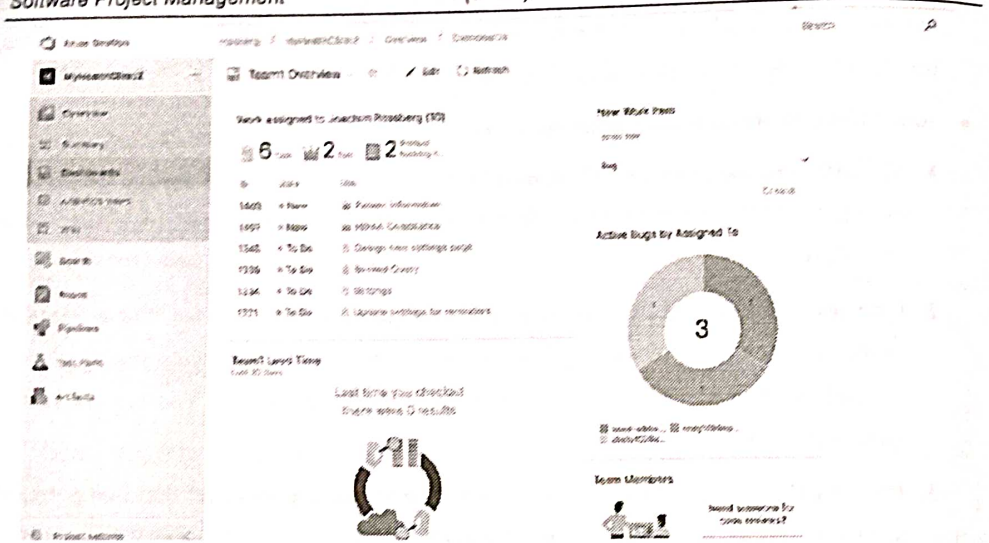


Fig. 6.4.1 : Viewing dashboards from Azure DevOps

Review Questions

1. What is visibility in DevOps ? What are the different ways to enable the visibility in Azure DevOps ?

6.5 Collaboration

- As you know, Azure DevOps comes with Team Explorer, which is an add-on to Visual Studio. Using this tool, developers can access all aspects of an Azure DevOps project. For example, they can view messages and queries, as well as access documents in the project. Developers can access the version control system and also create systems, tests and so on.
- Team Explorer is fully featured, but still a tool for people used to working in Visual Studio. This is no problem for us; but for most PMs and stakeholders, the GUI is confusing. They want an easy-to-use tool to access relevant information.
- Every project created with Azure DevOps also has a project portal created. This portal gives you access to reports, documents, project process instructions and other project-related information through a web interface that makes it easy for people unfamiliar with the Visual Studio interface to get the information they need.
- There is also a wiki (Fig. 6.5.1) that can be used for each Azure DevOps project. It is a collaborative area where team members can share information.

- Collaboration is not just about providing access to information, although that is as important as any other way of collaboration. Collaboration also means that you should be able to work together to accomplish one or more goals. Another way to enable this is that each work item can have a discussion thread (Fig. 6.5.2). In the thread, you can post comments, ask questions, etc., discussing that particular work item with your colleagues and stakeholders.

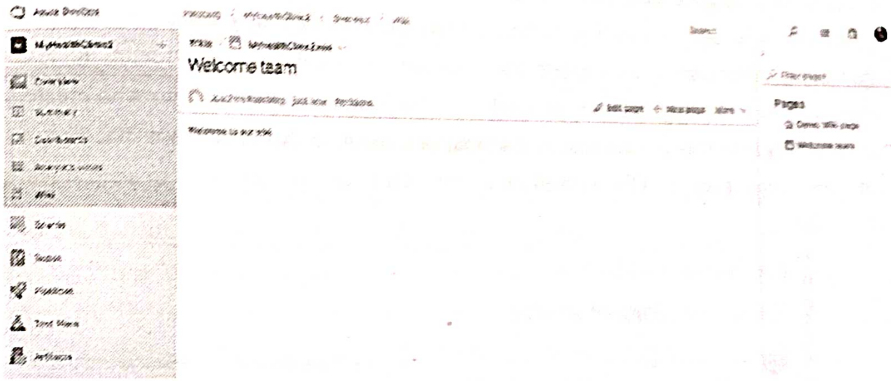


Fig. 6.5.1 : The Azure DevOps wiki

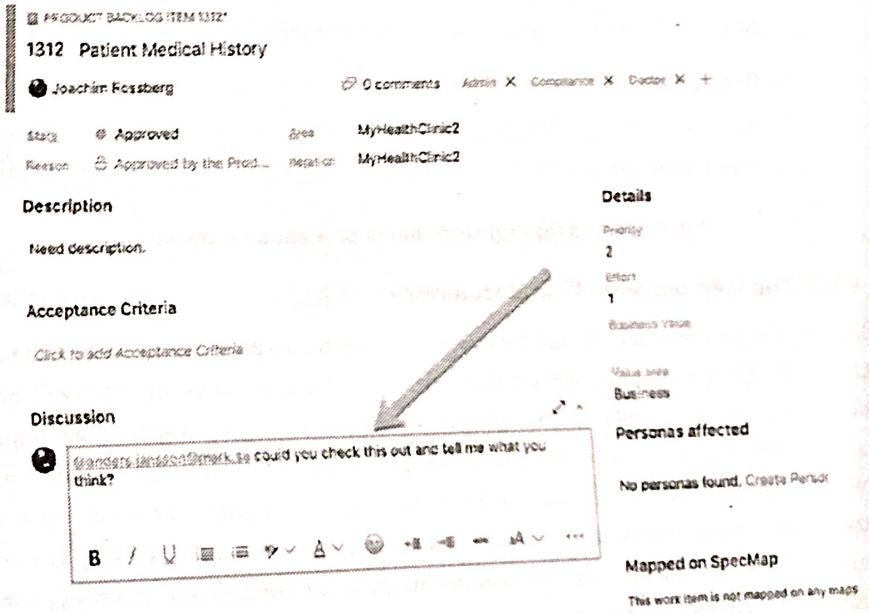


Fig. 6.5.2 : Sample discussion thread in a work item

6.5.1 Collaborative Work Items

- You can use Azure DevOps work item features to enable process workflows. Let's say a PM or someone responsible for entering requirements as work items into Azure DevOps creates a new work item of type Scenario. This scenario should probably be assigned to the developer to implement. The PM can use the work item system to assign (Fig. 6.5.3) a scenario to a specific developer in this case, Joachim. Joachim continues to work on the script until it is ready for testing. It then assigns the work item to a tester who will perform the testing. After testing is complete, the work item is closed. If a bug is found, the tester or whoever found the bug can use the work item tracking system to find out who created the scenario implementation and assign it to that developer, in this case again Joachim. Azure DevOps keeps track of who worked on a work item, so you don't have to keep track of it manually.

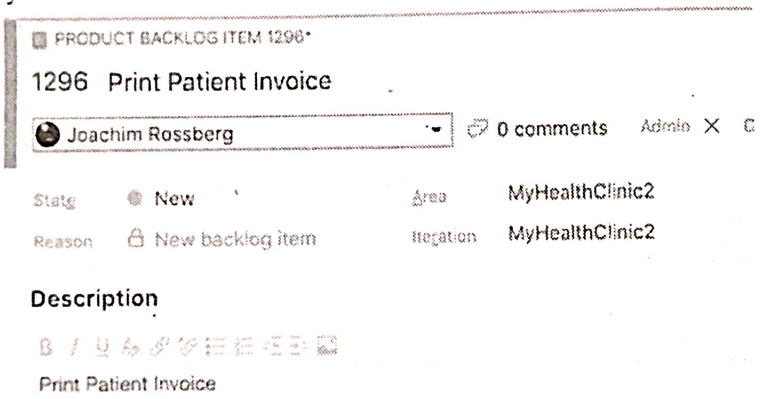


Fig. 6.5.3 : Assigning work items to a specific person

6.5.2 The Gap between IT and Business

- Bridging the gap between IT and business is of course a very difficult problem to solve. Azure DevOps doesn't quite get us; that's for sure. We don't think any tool will ever be because so much depends on the people in the organization, which is an important consideration. However, tools can help us bridge the gap, so you should carefully consider how you can use them to do so. To begin to address this issue, organizations need to improve their ALM process and way of working. Once a new way of working is in place, Azure DevOps can support much of their efforts by using, for example, a process template to implement that new way of working.

- The difference between IT and business is often a matter of workflow. There are many things to consider and when we have a solution - or start working on one - we need to evaluate which parts of that workflow we can automate and use tools to solve. One thing worth noting here is the use of Azure DevOps.
- The Project Server Connector with Azure DevOps allows you to integrate Azure DevOps with Microsoft Office Project Server. Thanks to this integration, you can manage your resources and automate this process more effectively. This way, you can better align your portfolio management process so you can choose which things to work on more efficiently.

6.5.3 Office / Microsoft Project Integration

- When we started projects in the past, we mostly used Microsoft Office Project, especially the Gantt chart, to plan the projects. We suspect that this is also the case with many of our fellow Prime Ministers. In many cases, we've used this product not primarily for the tool itself, but because so many of our customers use Microsoft Office that using Project comes naturally to them. The project has its strengths and weaknesses, like all tools and we can't say we don't like it, but we've never made friends with it. Sometimes it does things we don't expect and even though we know it's because we're not too familiar with its features, we still blame the product from time to time (which is unfair, but that's life sometimes).
- Excel and Project are two tools that most companies use on both the business and IT side of the company. Being able to use these tools makes it easier for business people to be a part of the ALM process because they can use a tool they are already used to working with. A nice feature is that the communication between Office and Azure DevOps is two-way. This means that updates in Azure DevOps will be reflected in Office and vice versa. This enables a dynamic way of working with Azure DevOps information.

6.5.4 Using a Single Tool / Role

- A good ALM tool should allow you to use add-ons that provide new functionality within a single interface. If a developer needs testing features, you should be able to integrate them into the development tool. A developer should not switch tools to perform testing tasks. This is also what Visual Studio offers. There is no context switching because team members can use the same GUI no matter what role they are currently performing. Azure DevOps is also extensible, allowing you to create your own add-ons as well as purchase third-party add-ons that are accessible from within Azure DevOps.

6.6 Extensibility

- When the built-in Azure DevOps features aren't enough, you can use extensibility features to extend and enhance them. Azure DevOps is often seen as a closed black box that Microsoft ships when it is more of an enterprise resource planning (ERP) system for DevOps. Any DevOps environment must be tailored to the organization's processes, existing applications and services.
- Many of our customers have been a bit reluctant to adapt to TFS. Instead, they tried to squeeze their way of working into the templates provided by Microsoft. We think this is not the right way. Our suggestion is that you start in reverse. Start by asking yourself how you want your organization to operate. This process involves all parts of the organization, from the business side to operations. Try to find consensus on how to work in the DevOps process. You'll find that it's also a good start for collaboration within the company. With Azure DevOps, you can customize your process (easier than with TFS), so you shouldn't shy away from customization.
- For example, consider work items and the information they contain. If the fields and information in the templates are not enough, you can expand or modify them. Azure DevOps allows you to do this by changing the process template. You can choose to add the information you need and it will be stored in Azure DevOps databases so you can access it from your reports and queries. Also, don't forget to change reports or queries; otherwise you will not see your information.
- Some of our customers have changed the workflow of a work item by adding additional states to it when the ones supplied were not enough. We've often used TFS Power Tools for TFS to do this (and still do for TFS 2018), but Azure DevOps provides an easy way to do this through a web interface.
- Once you have an initial idea of how you want to implement DevOps, start exploring what Azure DevOps gives you out of the box. Use what can be used, change other things and create your own solution if needed.
- One of the great strengths of Azure DevOps is its extensibility and flexibility. You can customize the entire tool to fit most parts of your DevOps process. If you want, you can develop your own add-ons by supporting roles that are not included from the start. We strongly encourage you to use these extensibility features; but ultimately it's your choice.
- Extensibility is a great way to integrate existing systems and eventually migrate some of them to Azure DevOps to reduce the toolset in the organization.

6.7 Difference between TFS and Azure DevOps

- Azure DevOps is cloud-based. This is also the version of TFS for which Microsoft deploys all new features first. Every three weeks, Microsoft tries to update Azure DevOps. These updates are then packaged into a TFS update that is released approximately every three months. Table presents an overview of the feature differences between Azure DevOps and TFS. Keep in mind that the information in the table changes over time. VSTS is updated every three weeks with new features and the gap between the two is likely to be bridged.

Feature	TFS	Azure DevOps
Work items, version control and build	Yes	Yes
Agile product/project management	Yes	Yes
Test case management	Yes	Yes
Heterogeneous development (Eclipse, Git)	Yes	Yes
Ease of installation and setup	Good	Better
Collaborate with anyone, from anywhere	Good	Better
Data stay inside your network	Yes	No
Process template and work item customization	Yes	Good
Data warehouse and reporting	Yes	No
CodeLens support	Yes	Yes
Cloud load testing	No	Yes
Application insights	No	Yes
Always running the latest version of Azure DevOps	No	Yes

Table 6.7.1 : Comparison between TFS and Azure DevOps

Review Question

1. Explain the concept of Collaboration.

6.8 Metrics in Agile Projects

6.8.1 Introduction to Metrics in Agile Practice

- A key performance indicator (KPI) is a performance measurement used in most organizations to evaluate the success of the organization or the success of a specific activity

within the organization. KPIs are often used to measure the effects of a change project for example, implementing a good DevOps process or to evaluate the progress of a development project.

- You can use DevOps online assessment scores as KPIs and compare assessment scores before and after implementing DevOps process improvements. In this way, you will get information about whether you have improved as a result of implementing the new process.
- You should also be able to use reports from your DevOps toolkit during projects to see if you are continuously improving your work. Continuous improvement is something to strive for in my opinion. For example, when it comes to project management, you can look at team velocity (how quickly the team is able to work) and determine if it is increasing or decreasing. Using reports and metrics from your DevOps tools, you can choose the KPIs you want and learn how to measure them.
- It covers five most aspects of software development. Note that they are not only for agile projects, but can be used in many other projects as well :
  1. Project management
  2. Architecture, analysis and design
  3. Developer practices
  4. Software testing
  5. Release management
- One important thing that we need to keep in mind, in TFS 2018 no metrics are available for Azure DevOps. It is recommended that metrics are available for full information.
- Some reports require that the team project collection that contains your team project be provisioned by SQL Server Reporting Services. A report is not available if reports are not displayed when you open the Team Explorer and expand the Team Project node.

6.8.2 Metrics for Project Management

- To get good metrics about the status of your projects, it's important to measure your progress. You can do this in several ways. If you use Agile as a methodology, you should be familiar with many of these metrics and reports. They may be new to others. Note that not all of these reports are available in Azure DevOps, but only in TFS on-prem.

Agile metrics

- Let's look at some important messages that are commonly used in agile practices :
  - Overview of pending matters

  - Sprint burndown
  - Velocity report
  - Release burndown
  - Remaining work
  - Unplanned work
- The backlog overview lists all user stories, filtered by tags and iterations and order of importance. It's basically a list of user stories filtered by the criteria you need. Many people use Excel (or other spreadsheet) to create this report, but many DevOps tools have built-in support for creating it. Fig. 6.8.1 backlog overview in Azure DevOps shows how this looks in Azure DevOps.

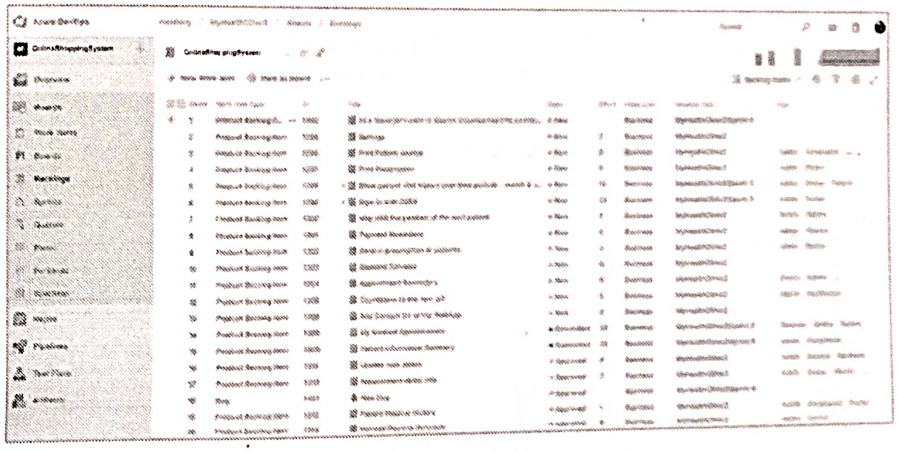


Fig. 6.8.1 : Backlog overview in Azure DevOps

- The sprint burnout report shows how much work remains to be done in the sprint. This report shows how much work remains to be done in the sprint. You can use it to predict when the team will complete the work assigned to this sprint, either during the sprint or after it is completed. Based on this information, the team and product owner can take steps to ensure that they deliver what they promised to deliver.
- The release burn-in report (Fig. 6.8.3) shows the same as the sprint burn-in, but for the work contained in the release.

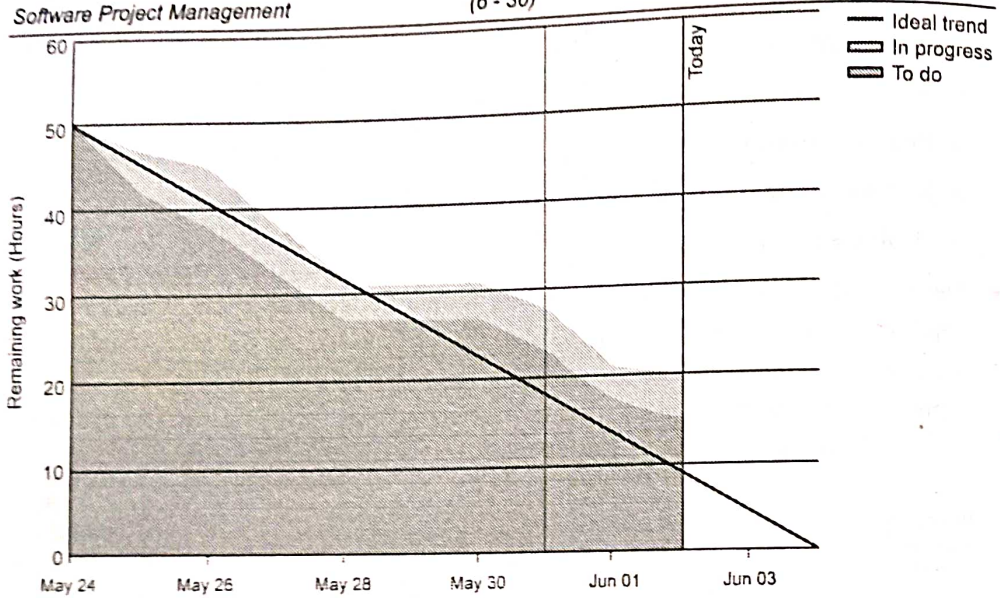


Fig. 6.8.2 : Sprint burndown report

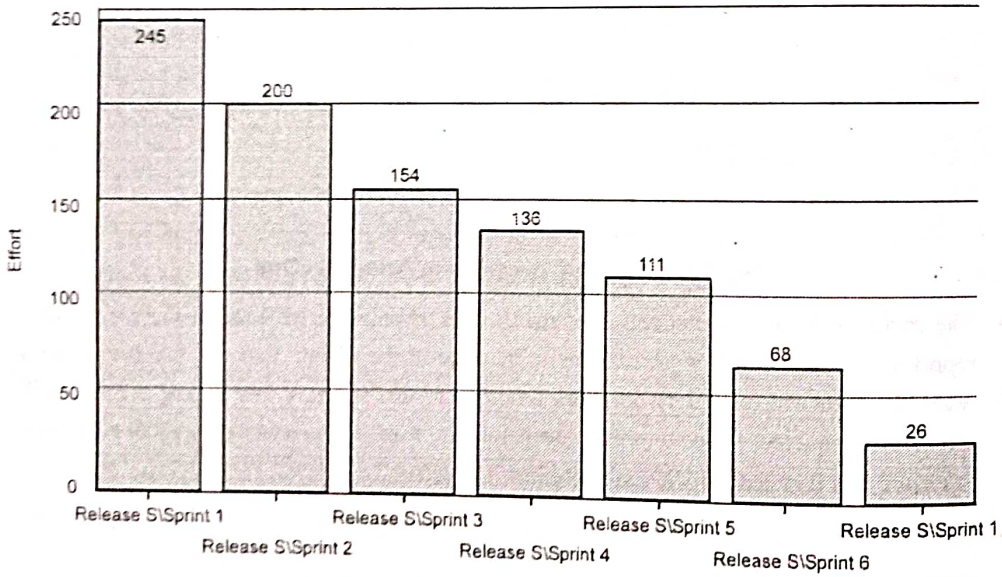


Fig. 6.8.3 : Release burndown report

- The burn-in and burn-rate report (Fig. 6.8.4) is another way to view project burn-in. No surprises : This is the same information as in Fig. 6.8.1. The burn rate provides summaries for the completed and requested rate of work for a specified time period. You can also view information for team members in some tools. Sometimes you can choose to view a report based on hours worked or number of work items.

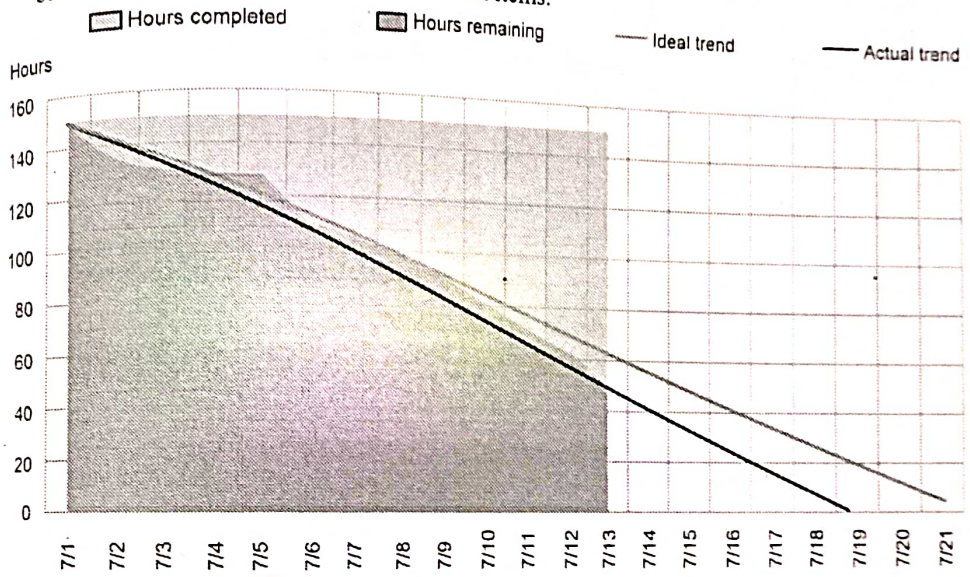


Fig. 6.8.4 : Burndown and burn rate report

- Velocity (how much work the team can take on in a sprint) is important, especially for the product owner planning how much work can be accomplished in upcoming sprints. Speed is usually a measure of the effect per story point a team can achieve.
- Before starting any work, the product owner calculates the theoretical speed to start planning. It updates over time according to the team's actual velocity based on how much work they complete in each sprint. This helps the product owner estimate how much work the team can take on in the upcoming sprints. The velocity report (Fig. 6.8.5) can help you get this information easily. Here you can see how much effort the team put into each sprint.
- Another useful metric is the remaining work report (Fig. 6.8.6). You can use it to track your team's progress and identify any issues in your workflow. With some tools, you can view this report in either the labor hours view or the work item count view.

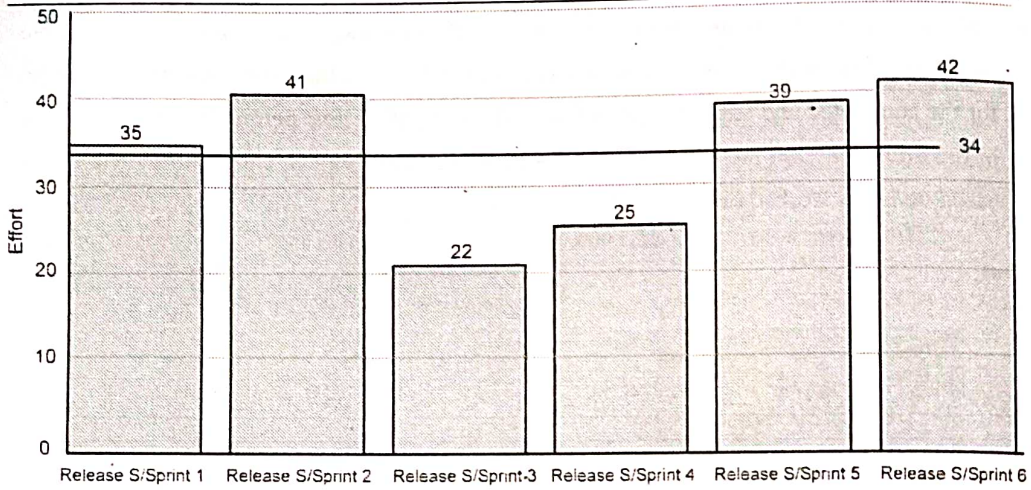


Fig. 6.8.5 : Velocity report

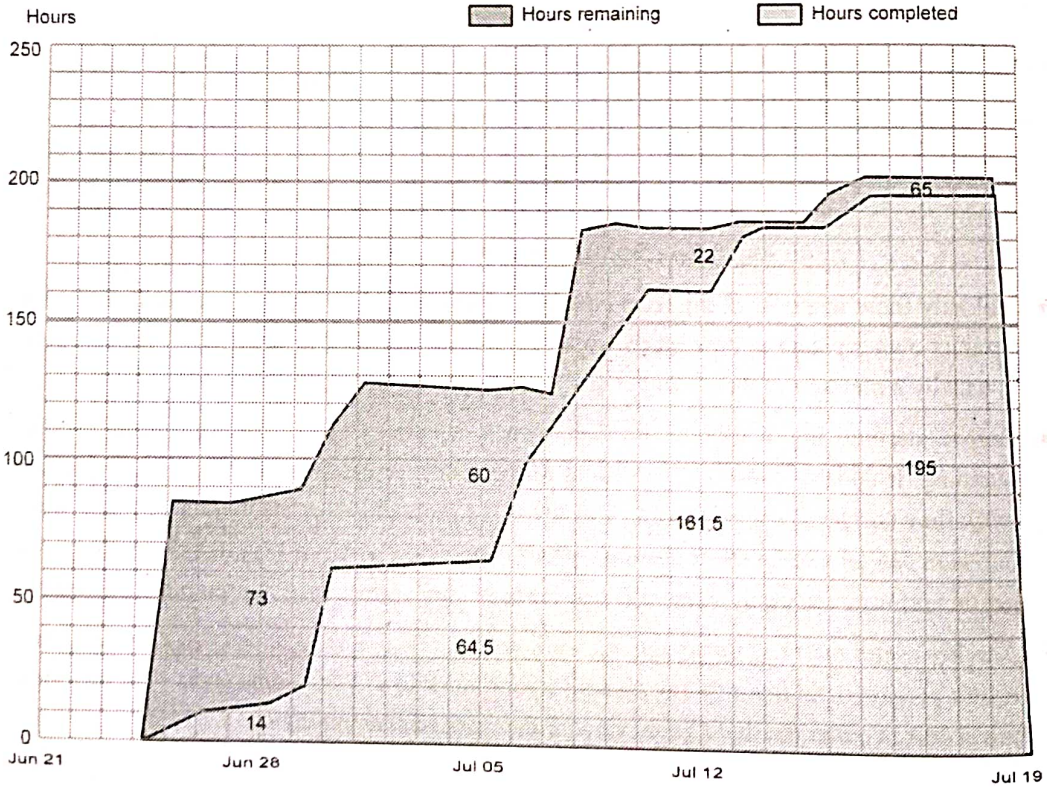


Fig. 6.8.6 : Remaining work report

- The unplanned work report (Fig. 6.8.7) is useful when the team is planning an iteration by identifying all the work items they intend to resolve or close during the iteration. Work items assigned to an iteration based on the build plan completion date are considered planned work. All work items added to the iteration after this date are identified as unplanned work.

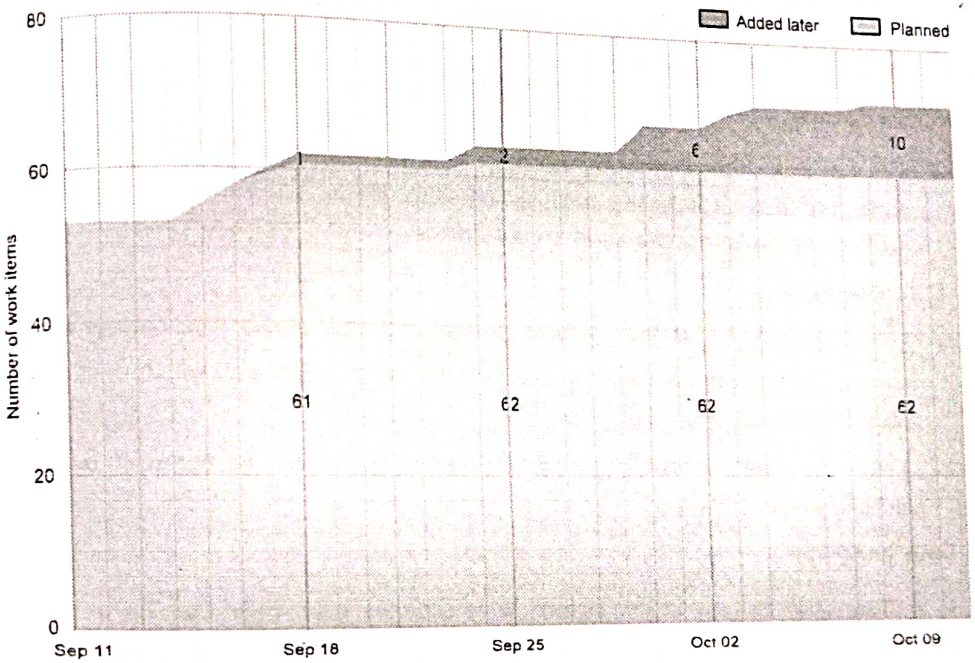


Fig. 6.8.7 : Unplanned work report

6.8.3 Metrics for Developer Practices

- Developer performance metrics are KPIs that can help you understand whether you're successfully working to improve your code. These measures are useful both from an architectural and design point of view, as well as from a developer's point of view. Using them will help you improve the way you design an application or system.
- Several important metrics are automatically available in many tools that can help you better understand the quality of your data development work :

1. Code coverage
2. Code metrics
3. Compiler warnings
4. Code analysis warning

1. Code coverage

- Code coverage shows how much of the code has been covered by automated unit tests. You get the value as a percentage of the entire code base. It is often difficult to decide what percentage is enough. Should you always strive for 100 % ? Or is 80 % enough ? This is something the team needs to discuss with the Product Owner when using Scrum or a similar decision maker when using other processes. This value is specified as the definition of done.

2. Code metrics

- You can look different code metrics like Lines of code, class coupling, inheritance depth, cyclomatic complexity and maintainability index. Metrics for Architecture, Analysis and Design can also apply to code metrics.

3. Compiler warning

- Errors and warnings should be avoided in the project. Allowing more than zero errors or warnings tends to cause the team to accept a lower quality code base, which over time makes the code unmaintainable.
- Monitor this metric to ensure zero errors. Ideally, this should be enforced by the autobuild policy.

4. Code analysis

- Warning code analysis in development tools performs static code analysis to help developers identify potential issues with design, globalization, interoperability, performance and security, to name a few.
- Much of this functionality is currently only available for .NET development; if you're using Java, things may be different.
- Code analysis tools provide warnings that indicate policy violations in managed code libraries. Alerts are organized into policy areas such as design, localization, performance and security. Each warning indicates a code analysis rule violation.
- Code analysis can be used to enforce company policies on the code that developers write. Many DevOps tools offer good support for code analysis and usually include a set of rules. You can often extend the functionality by writing your own set of rules or by suppressing the rules you don't want. Be sure to discuss this analysis with your development team and product owner, as warnings impact the effort required to meet the done definition.

6.8.4 Metrics for Software Testing

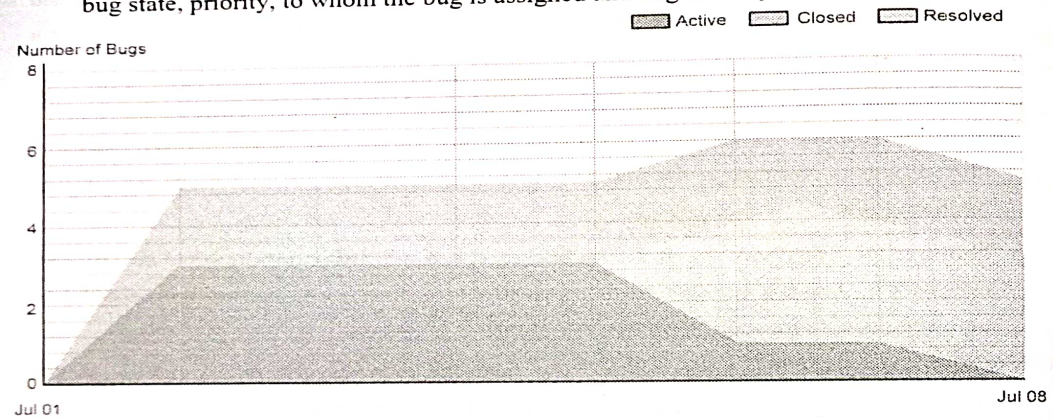
- Software testing is an important area. Testing should be an ongoing part of any development effort, not just an end-of-project phase. There are good metrics you can use throughout your projects to make sure you have high quality testing in place. Below are a number of metrics you can use as a KPI for software testing.
  - **Errors per state :** This metric tells you how many bugs are active, resolved or closed; either the number of active errors increases and whether the number of resolved and closed bugs is constant. If the numbers remain constant, to have to analyse your testing methodology.
  - **Number of bugs sent back from testers for more information (or "reactivated errors") :** A large number of reactivated buses may indicate this communication between developers and testers needs to improve.
  - **Code coverage :** This metric shows how much the code was covered by automated unit tests. You will get value as a percentage of the entire codebase.
  - **Test run results :** This metric shows how your tests perform and whether you have many failed tests. If you do, you need to look at what can be done improve tests.
  - **Percentage of requirements covered by test cases :** As the name suggests, this metric gives a percentage requirements covered by test cases.
  - **Percentage of requirements covered by testing :** You have actually verify the requirements with test cases. If this number is low, then the figure for percentages requirements covered by test cases is high, you can you have a problem to deal with.

6.8.5 Example Reports

- The reports described here are found in many tools :
  1. Bug status reports
  2. Reactivations reports
  3. Bug trend reports

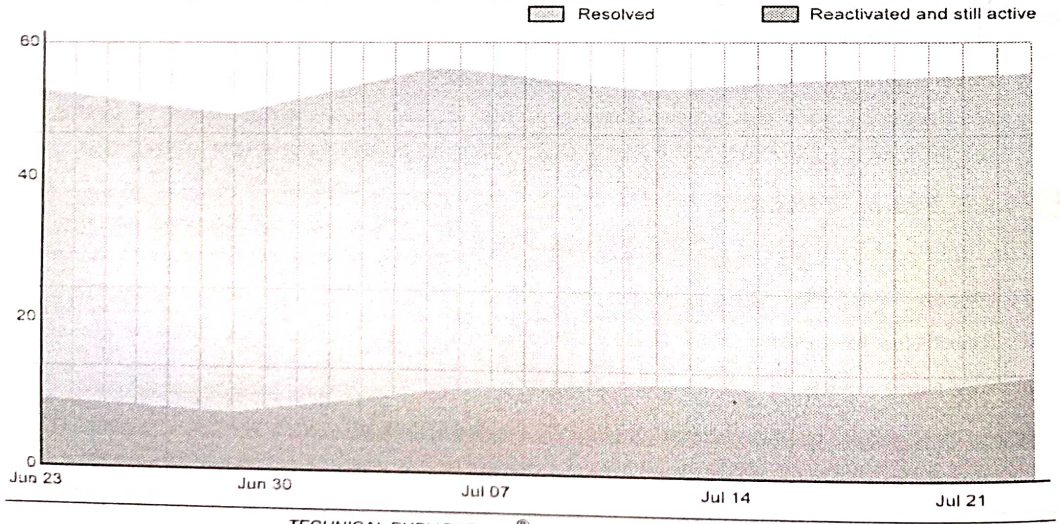
1. Bug status report

- The *bug status report* gives you information about the cumulative bug count based on bug state, priority, to whom the bug is assigned and bug severity.



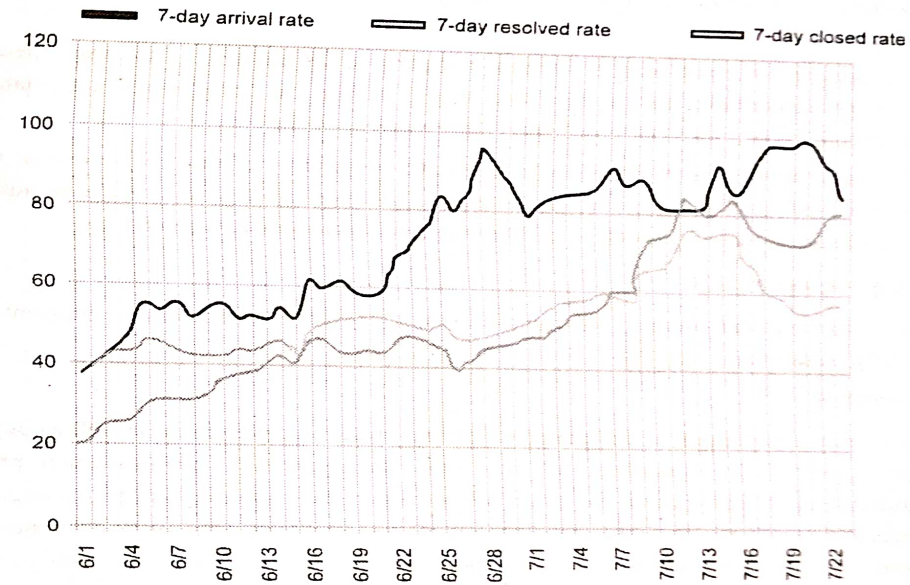
2. Reactivations report

- The *reactivations report* is used to determine how many bugs have been resolved or closed too early.



3. Bug trend report

- *Bug trend report* helps you track the rate at which your team is finding, resolving and closing bugs.



Review Questions

1. Which messages are used in agile practices ?
2. Write short note on : metrics for project management.
3. Explain any four metrics used for developer practices.
4. Which Metrics can be used in software testing ?
5. List any three examples of reports for metrics in Agile Projects
6. Write a short note on metrics in Agile Projects

6.9 Agile Project Management in Azure DevOps and TFS

- In this Section, you will follow the launch of an agile project using Azure DevOps. Many of the concepts discussed earlier in the book are illustrated in this section so you can see

how to go from planning to implementation. We will also look at how Azure DevOps can support the agile project management process during sprints. Note that although Azure DevOps is used as an example, you can do most of the things (and then some) in your on-premises TFS.

- For this purpose, here using a fictitious company in the examples. This way, there is a common denominator in the things presented, so you can more easily understand the process and how Azure DevOps supports development organizations.
- The main part of this chapter is written from the perspective of the product owner, whom we will briefly introduce. Some parts of the text have a personal touch. This is because part of the project focuses so much on collaboration and interaction between people.

6.9.1 Case Study

- Let's start with the company used in the example. Any similarities to real companies are totally unintentional.

Background

- Online shopping system is E-commerce platform created for the sellers as well as buyer to gather and fulfill their requirements. On this platform anyone who want to sell their product globally will list and enter all the details about their product including shipping charges and price of the product with all the specifications of product. While whoever wants to purchase goods will list on this platform and enter their contact details and search for their purchase list and fulfill payment options and purchase item which can be deliver to them within specific period whichever necessary for shipping to their address.

6.9.2 Building the Initial Team

- Product owner knows that it is recommended that product owners start with a small team during the initial project planning. He therefore chose development manager, senior developer and business analyst as the first team members because they have experience in the domain and have experience as senior team members having worked for other companies as well. In addition, they are available for the entire pilot project, which is an important aspect for product owner. He knows how important it is to have consistency among team members during a project. He also selected the Scrum Master for the entire project. The rest of the team will be selected a little later in the project.
- Product owner creates a project in Azure DevOps (Fig. 6.9.1) from the web portal using the Scrum template. Name it OnlineShoppingSystem Pilot, then choose Scrum and Git Version

Control. As an initial team, they debated which version control to use, Git or TFVC and decided on the former.

- Product owner then begins to create the teams needed for the project.

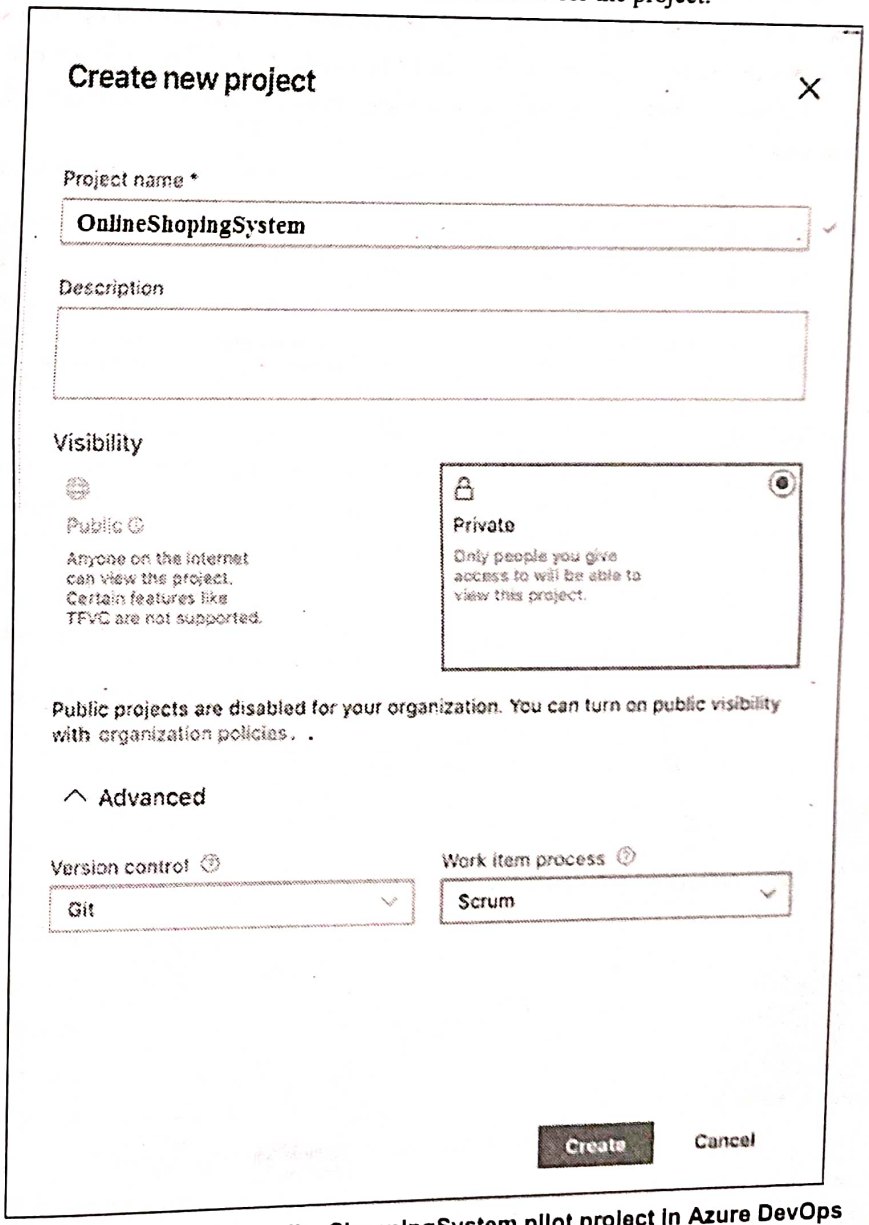


Fig. 6.9.1 : Creating the OnlineShoppingSystem pilot project in Azure DevOps

6.9.3 Creating New Teams

- In Azure DevOps, a team is simply a way to recognize the team or teams you have, whether it's a single team working on a project or multiple teams. One person can be a member of several teams. You can use path areas to mark development work that belongs to a specific team. You can quickly visualize which work items belong to which team. If a work item is assigned to an area path that is assigned to a team, that work item is placed in the backlog for the team.

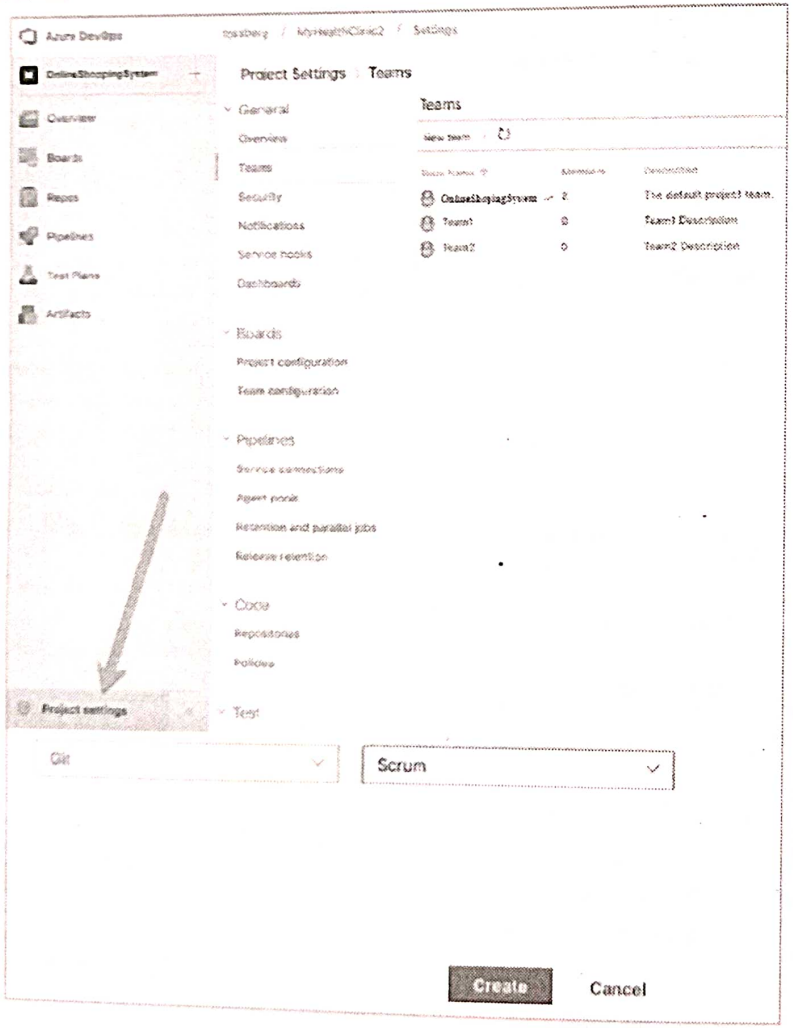


Fig. 6.9.2 : Opening project settings

- A team is a concept used by Microsoft, but it doesn't necessarily correspond to something in your organization. A team can be anything; it can also represent your product. It may be easier for users to think about products instead of teams and then just use the team concept to personally develop the product. One thing you shouldn't do is mimic the company's organizational structure in these teams. Because the company is organized according to different criteria (departments, division into functional teams), the development team for a product can cross these boundaries. This means that the teams to consider here are based on the products you build.
- You can use teams in several ways. Some have a specific team work on a specific part of the solution, while another team works on other parts, all with separate backlogs. In large Scrum projects or when using SAFe, you may also need several teams working in parallel on the same backlog, but you need to differentiate between the teams and the work they do. The best way is to try one team setup in a project and see what works best for you.
- The default team has an area path and iteration path automatically configured when creating a new team project. Once you select areas and iterations for the team, a backlog is automatically generated for the team. If your project uses more than one team, you can easily add new teams by selecting project settings from the left menu (See Fig. 6.9.2 on previous page).
- In the Teams view (Fig. 6.9.3), click New Team to create a new team.

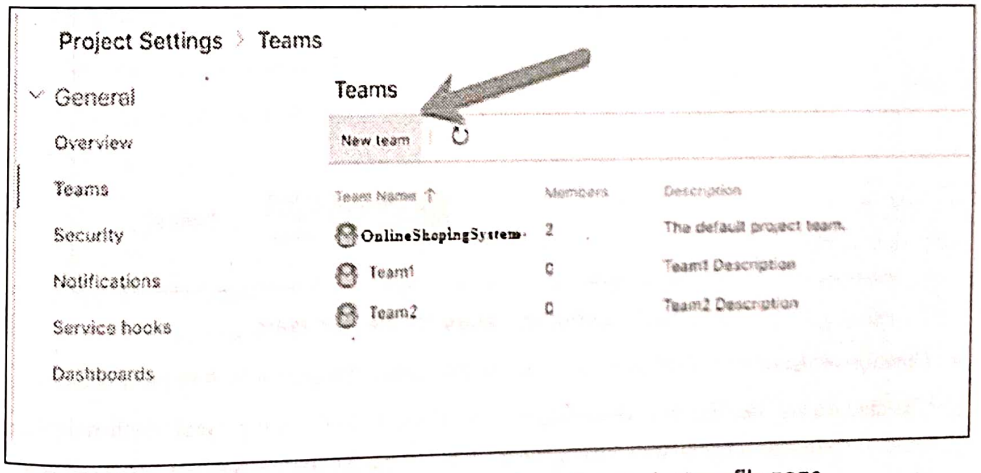


Fig. 6.9.3 : Creating a new team from the Project profile page

- Fill in all team information (Fig. 6.9.4). On this screen, you can also select permissions for the team by adding them to an existing security group. You can also create a team area at

this point. If not, you can assign it to a region later. When you're done, click Create Team and in a few seconds a new team will be created and you can start adding users to it.

Fig. 6.9.4 : Entering values for the new team

- Creation of the backlog and team structure for the Online Shopping System pilot project.
- 'Product owner' decides to have two development teams work on it project. Both will work from a common accumulation managed by the Online Shopping System program team. However, each team may not see the other's PBI. 'Product owner' wants a subset of the backlog in teams. Fig. 6.9.5 shows an overview of this structure.

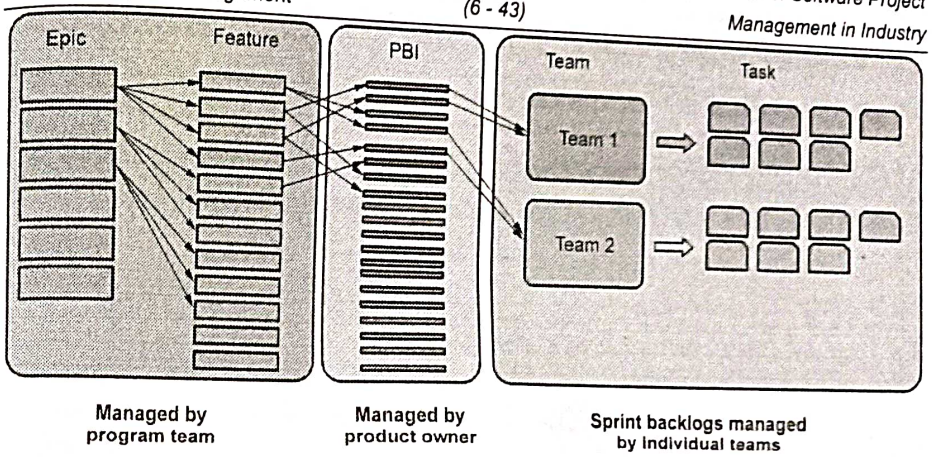


Fig. 6.9.5 : The backlog structure

- To build the structure, 'Product owner' first creates three teams :
  - OnlineShoppingSystem program team
  - OnlineShoppingSystem1 team
  - The OnlineShoppingSystem2 team
- It then configures regions and iterations to fit this backlog structure. 'Product owner' then creates three regions that are not dissimilar to the team structure and each team is associated with their respective region in Azure DevOps :
  - OnlineShoppingSystem program team
  - The OnlineShoppingSystem1 team
  - The OnlineShoppingSystem2 team
- It then proceeds to configure the initial sprint structure. It does this in Project Settings/Project Configuration. 'Product owner' has no estimates yet and cannot complete the entire release and sprint structure. But he knows that there will be several releases and that each release will have its own sprints. 'Product owner' knows he can change this structure, so for now he creates the iteration structure shown in Fig. 6.9.6.
- For each team, 'Product owner' selects the appropriate iterations to see in his backlog. By using ranges and iterations in this way, he is able to implement the structure he wants.

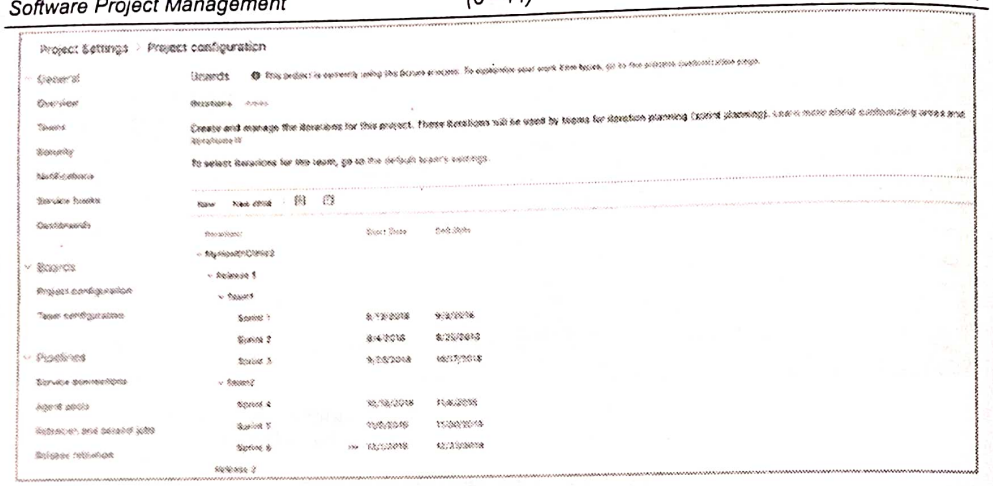


Fig. 6.9.6 : Starting iteration structure

Team building

- Now the team is close to starting. So far, 'Product owner' has development manager, senior developer, business analyst and Scrum master as team members. He talks to them about the project and they decide they need three more people to strengthen development and testing competencies. They also discuss how they need an experienced user experience person on board. After these decisions, 'Product owner' selects the following people to join Team1 :
  - Developer
  - Lead tester
  - User experience
- He will also select four people to work on Team2. 'Product owner' contacts their managers and makes sure these people are available to work on the project. Fortunately, they all are and when they approach a potential team members and explains the project, they are happy to come on board.
- 'Product owner' decides to go for bi-weekly sprints as it has always been a good time box, based on his experience. He once had a team that complained they couldn't complete a PBI during the three-week sprints they were using. This team always seemed to be late or unable to deliver on everything they committed to and complained that team members needed more time in sprints. 'Product owner' then told them, "Okay, then we'll use two-week sprints instead of three." The team was very confused because 'Product owner' had reduced the number of days in the sprints, not increased them. After starting to work in

two-week sprints, they soon found that they delivered more in two weeks than in three weeks. The team was more focused and didn't delay until the end of the sprint; therefore they were more effective.

- One of the product owner's responsibilities is to staff the project, at least initially. After the initial team is formed, it is up to the team members to inform the product owner of what competencies they need to fulfill the project vision.

Adding team members

- Now that 'Product owner' has created a project, he goes to the web access home page. By default, one team is created, so it can start adding team members immediately. 'Product owner' clicks on Team Members, as shown in Fig. 6.9.7 (see the arrow).

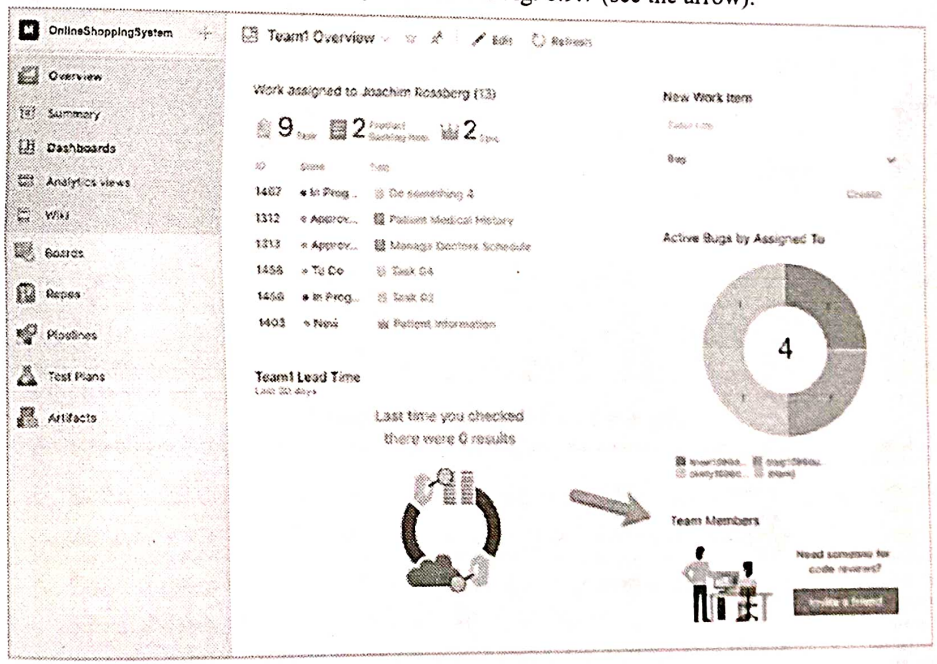


Fig. 6.9.7 : Adding team members

- A dialog box opens (Fig. 6.9.8) and 'Product owner' adds team members. 'Product owner' will then go to Project Settings, then Teams (Fig. 6.9.9) where you can manage team members for all teams.

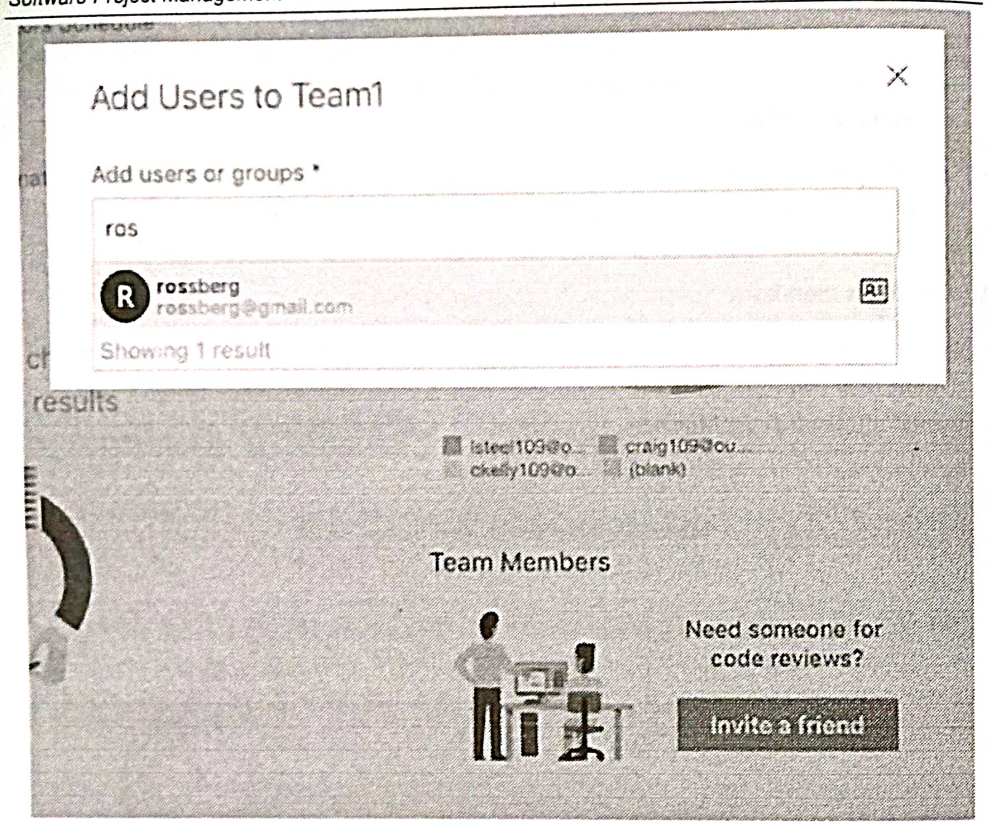


Fig. 6.9.8 : Adding team members2

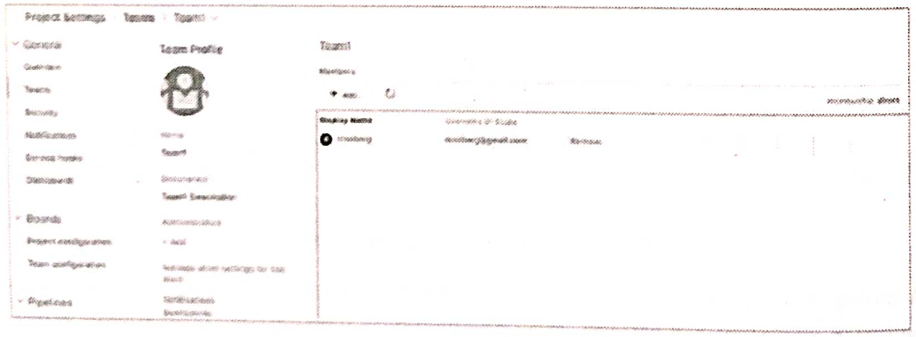


Fig. 6.9.9 : Managing all team members within Project Settings

- Manage Azure DevOps groups, teams and user permissions.

- In the project settings, 'Product owner' changes the permissions for both teams by selecting the Security tab (Fig. 6.9.10) and then selecting the Groups tab.

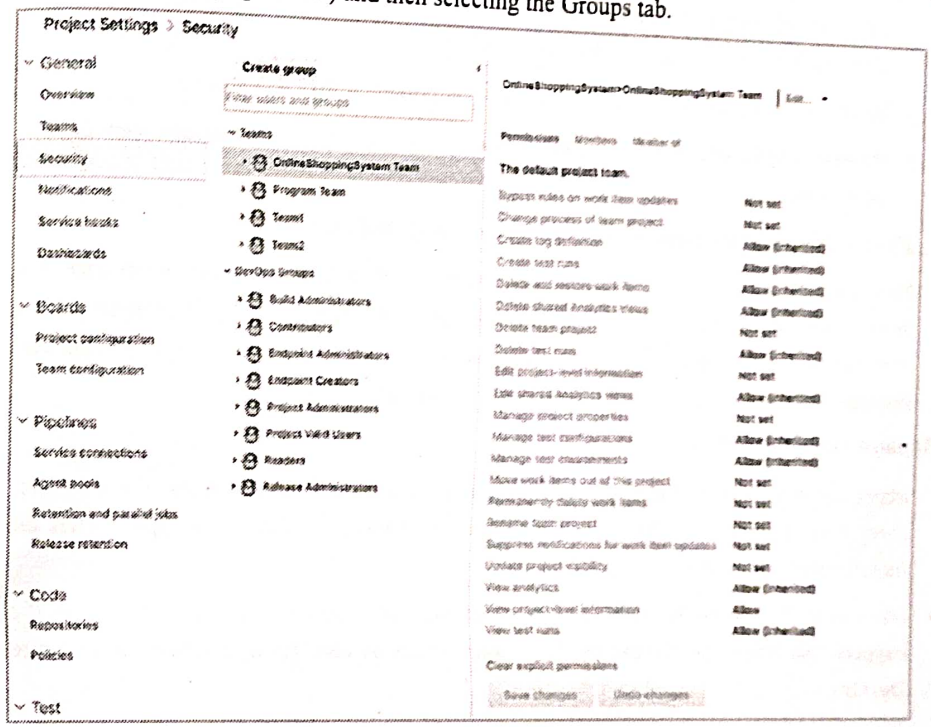


Fig. 6.9.10 : Viewing permissions for DevOps groups and teams in project settings

- As you can see in Fig. 6.9.10, there are eight different Azure DevOps groups by default :
  - **Build managers :** Members of this group have builds project permit. Members can administer the test environment, create test runs and manage builds.
  - **Contributors :** Members of this group can post to the project. This means they can add, edit and delete code; and create and edit work items.
- By default, the team group created when you create the team project is added to this group. Therefore, every user you add to a team will be a member of that group.
  - Endpoint managers
  - Endpoint builders
  - **Project managers :** Members of this group can manage a team project. But they can't create projects.

- **Valid project users :** Members of this group have access to Azure DevOps. This group automatically contains all users and groups that have been added anywhere within Azure DevOps. You cannot edit membership in this group
- **Readers :** Members of this group can view the project. They cannot edit it.
- **Release administrators :** Members of this group can perform all release management operations.

- There are different permissions you can set for groups and team members.
- Note that by default most permissions are inherited. Here you can break inheritance and define your own permissions. One common scenario is when you bring in external people who can only see parts of the work item structure. These specialized permissions can become difficult to track very quickly, so use this feature with caution.

Manage notifications

- Many activities that occur in Azure DevOps are exposed as notifications. For example, every time a build completes or a changeset is checked in, Azure DevOps notifies all stakeholders that an event has occurred.
- As a user, we might be interested in receiving notifications when something specific happens that affects us. We can get these notifications by configuring notifications in Azure DevOps.

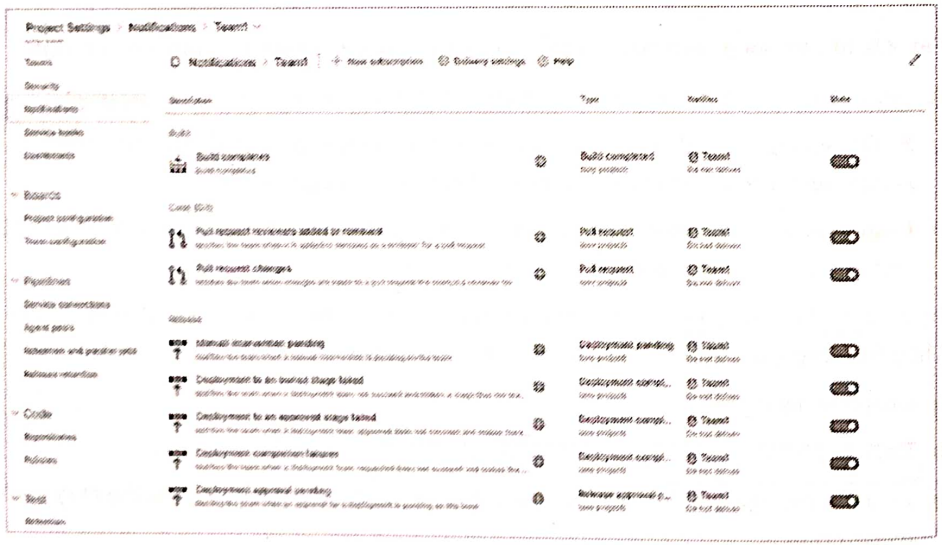


Fig. 6.9.11 : Selecting a notification in project settings

- To configure notifications, use the Notifications tab in Project Settings in the web portal.
- First, the notification editor (Fig. 6.9.11 on previous page) allows you to quickly select from a list of predefined basic alerts that are common, such as "Build Completion".
- To create a notification, select New Subscription (Fig. 6.9.12). This tab allows you to select from a list of templates when creating a new notification.

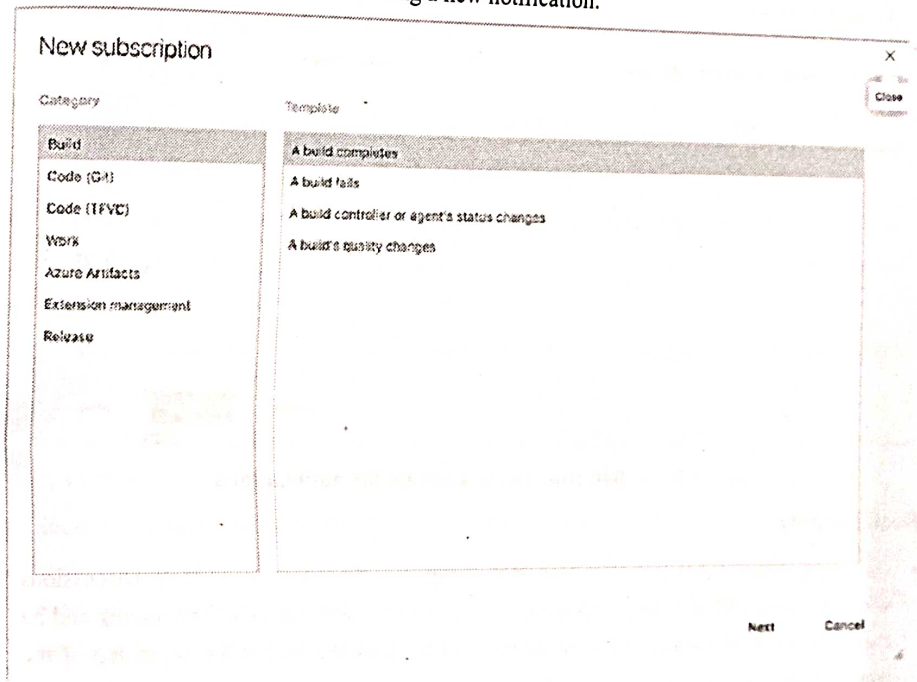


Fig. 6.9.12 : Selecting a custom notification in Project Settings

- There are more settings you can configure for notifications (Fig. 6.9.13), simply by clicking the Next button.
- Now let's get back to watching 'Product owner'. He did everything to build the initial product backlog and build the team. At this time, he has no information on how long the project will take or how much it will cost. In order to get this information in order to show it to the interested parties, he has to take several steps. Let's take a look at them, as they also include the planning of the first sprint.

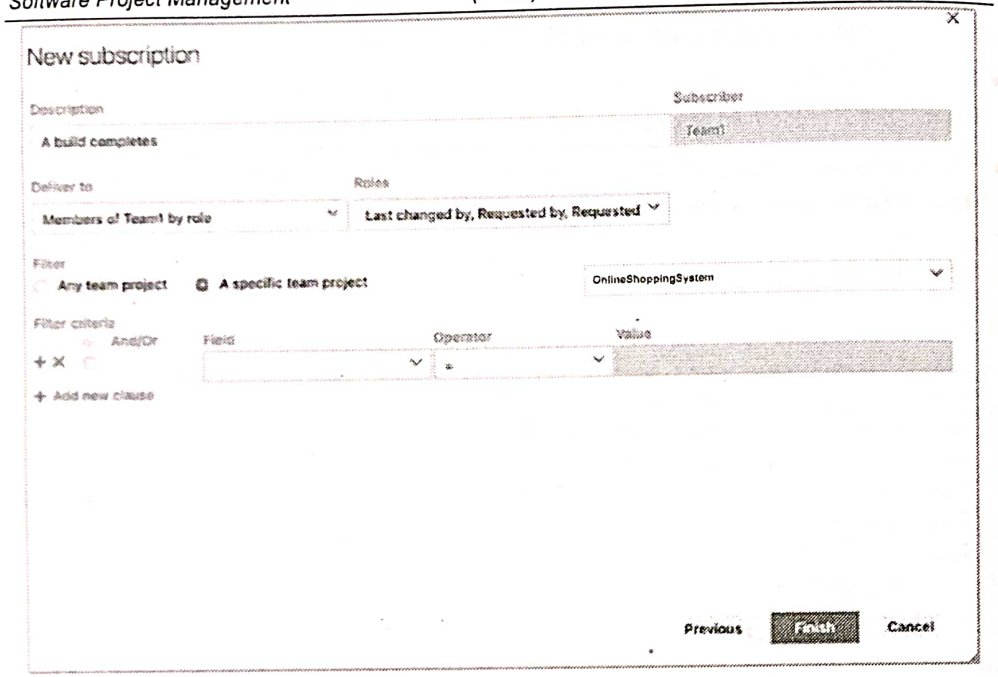


Fig. 6.9.13 : Selecting more settings for notifications

Requirements

- Gathering requirements is the fun part of the project in 'Product owner' eyes. Discussions with traditional PM Managers and stakeholders about requirements are forthcoming and he enjoys them. Traditionally, all requirements had to be established at the beginning of the project and many found it difficult to accept that it was okay to start a project without specifying everything. The fact that so many of these demands were ultimately wrong or unnecessary did not seem to trouble the traditionalists. They kept going headfirst into projects that often failed or were flawed.
- 'Product owner' has done so many successful agile projects that he knows it's okay to catch high-level requirements early on. They can start without all the details as they will be clarified in the upcoming retrospectives and sprint planning meetings as well as during the sprints.
- 'Product owner' convenes the first team for a requirements workshop. Admin manager is joining the workshop as he is one of the main stakeholders from the business side. Since Scrum Master is not present, 'Product owner' explains what they are going to do. He

emphasizes that they should look for higher-level requirements in the sense that they don't need to detail them yet. Solutions or technical details may not be discussed at this point. Development teams will be responsible for these topics to decide when sprints start.

- To avoid confusion, 'Product owner' explains the concept of a user story for the requirements team. He wants all the requests in the form.

  As <user type> I want <some goal> to

  <some reason>.

- 'Product owner' believes the meeting will last three hours, so he books a room with a large whiteboard. It also supplies participants with sticky notes and pens.
- 'Product owner' starts by brainstorming user stories. The participants are a little slow to get ready for the premise, but the meeting gets going when senior developer comes with two user stories :

  1. As a merchant, he want to manage expense reports over the internet so he can be more efficient.
  2. As a manager, he want to search expense reports, so can more easily get an overview of expenses.

- Suddenly all team members start typing. After a little over an hour, the pace slows, so they move on and spend the next hour going through the user stories they've created, clarifying them where necessary. 'Product owner' feels they have done a great job so far and now have a good foundation for the work ahead.

Building Backlog

- After the meeting, 'Product owner' goes to his desk and enters the stories into Azure DevOps. 'Product owner' then starts sorting the list by dragging the PBI in the backlog view. Creates an initial prioritization based on some assumptions :
  - Prioritization is based on the weighted truncated task first (WSJF).
  - After the initial sprint planning and estimation, the list will be updated.
- WSJF is a prioritization model used to rank tasks (such as features, user stories and epics) to achieve maximum economic benefit. Don Reinertsen(1) describes a comprehensive model called weighted shortest order as the first to prioritize orders based on the economics of the product development flow.

- WSJF is estimated as the delay cost divided by the job size (or duration). Jobs with a high WSJF take precedence over jobs with a lower WSJF. This tool is useful for Product Owners who work to prioritize backlogs and take economics into account when working with an order. Fig. 6.9.14 shows that element A has the highest WSJF (10) and therefore should be the element with the highest priority. WSJF became very popular as SAFe appeared in many organizations.

Feature	Job Size	CoD	WSJF
A	1	10	10
B	3	3	1
C	10	1	0,1

Fig. 6.9.14 : CoD, cost of delay; WSJF, weighted shorted job first

- It takes 'Product owner' about an hour to complete the initial sorting. Now she really had something to work with.

Adding backlog items in Azure DevOps

- 'Product owner' opens a project in Azure DevOps and feels a bit excited to see an empty project that he will soon fill with activities. 'Product owner' has lots of entries for backlogs. He takes a long look at the results of the initial story writing workshop and begins by going to the Backlog tab on the web page (Fig. 6.9.15).

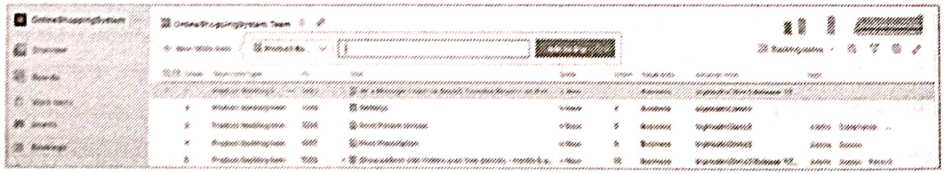


Fig. 6.9.15 : The Backlogs tab

- Quickly creates the first backlog in a new work item. Then double-clicking the created PBI opens the form shown in Fig. 6.9.16. It then uses the PBI in its backlog to start populating the fields. It leaves a lot as is for now and only fills in the name and description of the PBI. The domain is the OnlineShoppingSystem program team and the default is Iteration.
- 'Product owner' then continues to log the remaining high-level use cases until they are all signed into Azure DevOps. He often finds stories that are epic in size and need to be split into features and then PBI. It makes sure that the respective PBIs are linked to their Epic or Properties.

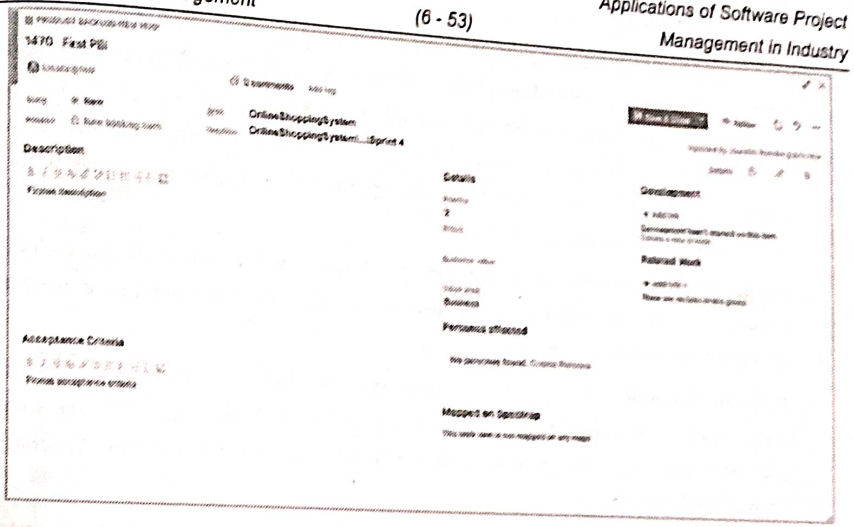


Fig. 6.9.16 : The first PBI

Definition Done

- Before 'Product owner' leaves for the day, 'Product owner' sets a date for another meeting with the team to set the definition of done. He brings in infrastructure specialist for this meeting as there are constraints from the infrastructure team that need to be taken into account when creating and rolling out new projects.
- He wants to discuss the definition of done with the team so that all participants have a common point of view before starting the actual coding. He has encountered problems many times where the definition was not done on the spot for the project, so he knows this meeting and its outcome is important.
- Two days later, they meet to determine the definition of done. 'Product owner' explains the importance of this concept and talks about the problems he encountered when a project did not have a definition of done. During the speech, it records recognition among the participants.
- 'Product owner' then asks all participants to write down the things they want to see done in the definition. After some discussion, they agreed on the following list of approved user stories :
  - **Environments are ready for release :** None not integrated WIP was left in any development or staging Environment. The CI framework has been validated and work, including regression tests and automated code review. The build engine is configured to schedule a build on clearance; runs an hourly or nightly build. All test data used to validate features in a release have been verified.

- **Support handoff is complete** (Note : It may be in a DevOps context or during development the team follows the product to support.) : All design models and specifications, including user specifications stories and tests, were accepted by the support staff to keep the increment moving forward. The support staff is happy to be inside control of the support environment.
- **Preparation for inspection is complete :** All sprint metrics are available, including burn-in or burn-out diagrams. None user stories that were not completed were re-estimated and returned to product backlog.
- **Code is complete :** All annotations have been resolved. The source code has been commented to the satisfaction of the development team. Where necessary, the source code has been refactored to make it understandable, sustainable and more feasible to support future changes.
- Unit test cases have been designed for all functions in development and requirements can be traced to code implementation (for example, using a clear function relevant naming convention). Code grade coverage is known and meets or exceeds the standard required. Unit test cases were executed and the increment has been shown to work as expected. Peer reviews are complete. (Note that if you pair programming, a separate peer review session may not be required.) The source code has been reviewed to the configuration management system, appropriate peer-reviewed comments have been added (when applicable). The source code was merged with main branch and automatic deployment to elevate the environment has been verified.
- **Testing is complete :** Functional testing has been completed completely, including automated testing and manual exploratory testing and a test report was issued created. All outstanding bugs (or incidents such as construction issues) were raised and resolved or accepted by the team as not contraindicated release. Regression testing has been completed and features provided in previous iterations.

Estimate

- With the definition done, the team now has a basis to begin an initial estimate of the work required for the project. 'Product owner' needs to come up with a rough budget to show stakeholders and an initial release plan. He decides to use "planning poker" to do this. He used it before and was happy with the results.

Poker Planning/Story Points

- 'Product owner' will call another meeting and everyone will gather in the same conference room where all the other meetings were held. This time the whole team is present, not just the starting team. 'Product owner' has purchased poker planning packages for everyone and will start the meeting by reading the first user story and then explaining the rules. 'Product owner' reads the story, the team asks any questions for clarification and then decides their value.
- After a short time, participants choose a card from their poker deck, but do not show it to others. After everyone picks a card, 'Product owner' asks them to turn their cards face up. Development Manager and Senior Developer are the furthest apart, so 'Product owner' asks them to explain their opinion on the user story. After this discussion, the team plays again. This time, participants are closer to each other's points (only one step apart) and a larger value is selected for the story. The team continues to play for all user stories until all story points are determined.

PBI update

- After the meeting, 'Product owner' returns to her desk and begins updating the PBI. Can now put story points for each PBI into work items in the Effort field. During sprint planning, the PBIs will be broken down into more manageable chunks and each task will be given a time estimate instead of story points (Fig. 6.9.17).

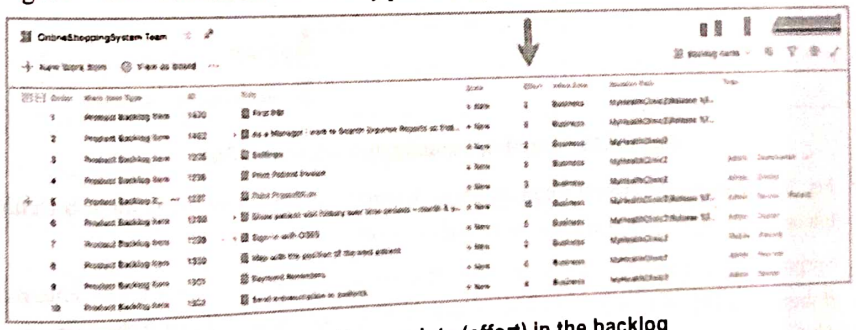


Fig. 6.9.17 : Story points (effort) in the backlog

- Now that the story points are entered, 'Product owner' wants to do an initial risk assessment before moving on to sprint planning and time estimates.

Initial speed

- 'Product owner' needs some additional knowledge before he can derive time estimates for the project. It needs to know the team's initial velocity, which is the team's velocity the number of user stories the team can complete in a given sprint. He also needs to know how many hours he has to work during the sprint.

Capacity planning in Azure DevOps

- Azure DevOps is great for capacity planning. For a sprint, click Backlog and then the Capacity tab (Fig. 6.9.18). You can set capacity, activities and days out of the office for vacations and holidays. When setting capacity, activities and days off, graphical information about hours and capacity is automatically generated in the pane in the right part of the screen.

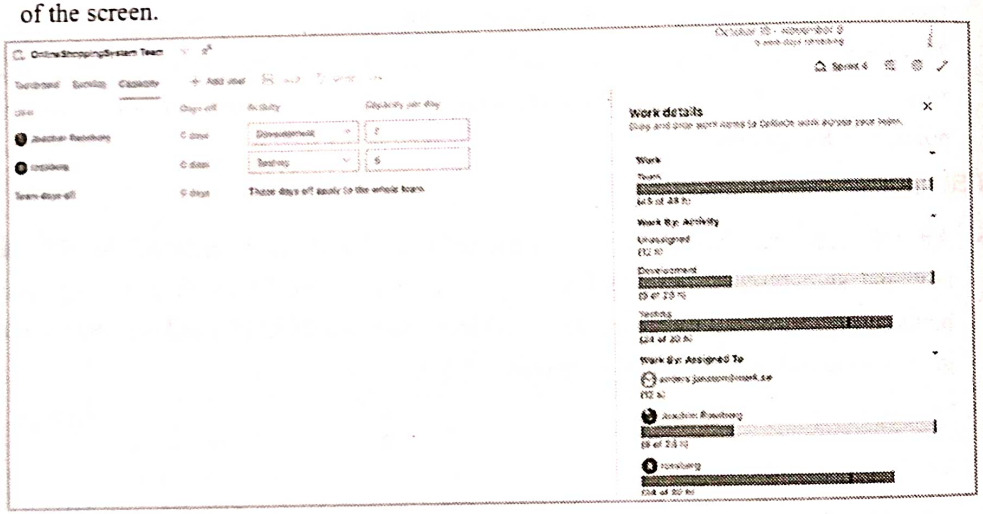


Fig. 6.9.18 : Entering capacity into Azure DevOps

- Fig. 6.9.19 shows the current sprint (sprint 4). Capacity is also displayed on the Job Details tab on the far right. Job details can be turned on or off.
- Sprint planning features in Azure DevOps offer the product owner three ways to determine if there is enough capacity to complete a project: by person, by activity or at the team level. These values are shown in the right panel of Fig. 6.9.19. Above is the team's collective working hours. Below that is Work By : Activity, which shows the work done so far for development and testing activities. You can also see Work By : Assigned To, which shows how much of each team member's capacity is assigned to tasks.

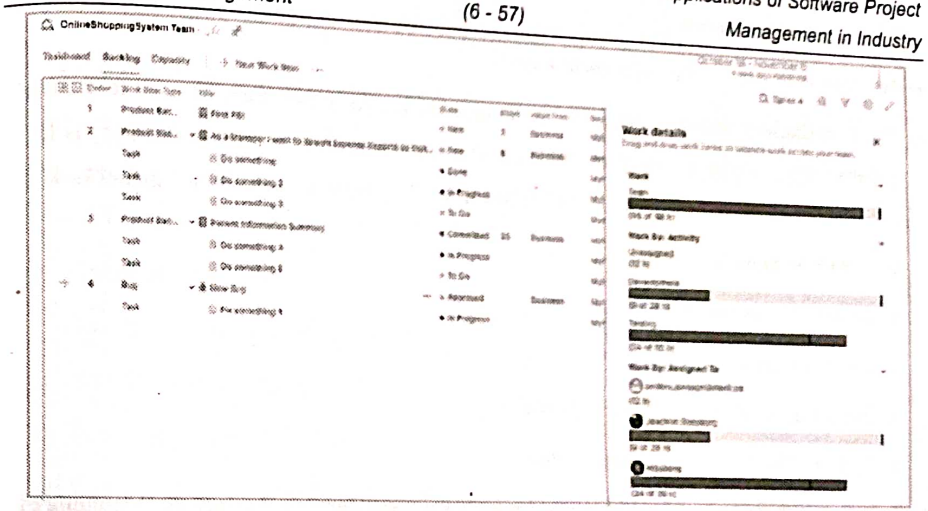


Fig. 6.9.19 : Viewing team capacity in sprint planning

Initial sprint planning

- To calculate the team's initial velocity, 'Product owner' usually conducts the initial sprint planning meeting. This is exactly the same as any sprint planning, except that it is done before the actual sprint starts. 'Product owner' has used this approach on previous projects and found it extremely useful in getting the data she needed to come up with team capacity.
- During this meeting, 'Product owner' and the team estimate the tasks for hours in order to plan the sprint. This value is important in determining how many user stories a team can accept during a sprint. Here's how 'Product owner' and the OnlineShoppingSystem team calculate their speed :
  - Estimate the first user story in detail.
  - Break down what the team needs to do to deliver story.
  - Estimate the number of hours for each activity and summarize.
  - Subtract the summary from the time available to the team has in the sprint.
  - Is there still time left ?
  - Take a new user story and repeat the process until there is no free time left.
  - Sum up the number of story points from the stories which are part of the sprint.

- Now 'Product owner' has theoretical speed.
- The first (highest priority) user story in the backlog is : As a merchant, he want to manage expense reports so he can be more efficient. The number of story points for this story is five (as determined during the planning poker meeting). The story is divided into smaller tasks :
  - Create an expense report.
  - Delete an expense report.
  - Edit the expense report.
  - Submit an expense report for approval.
  - Log into the expense reporting system.
- Together with the team, 'Product owner' prioritizes these items as they are the beginning of the sprint backlog. The sprint backlog looks like after prioritizing the following :
  - Create an expense report.
  - Submit an expense report for approval.
  - Edit the expense report.
  - Delete an expense report.
  - Log into the expense reporting system.
- The team continues to break down each of these tasks into smaller ones pieces and estimate them for hours. To create an expense report, they come up with the following tasks :
  - Create a GUI.
  - Create business logic.
  - Meet the definition of a finished requirement.
  - Write a user manual.
- The estimated number of hours for this user story is 137. With 290 hours available, they still have 153 hours left in the sprint. This means they have room for more work, so they continue with the next user story on the backlog:
- As a controller, I want to be able to manage users in the system so I have full control over the users.

- This user story was worth the points. After dismantling, they have 95.5 hours left, so they continue with the next user story worth two story points. When this planning was done, there were 23.5 hours of available time left.
- The total number of story points for a sprint is now ten, which is the starting speed of the team. The sprint backlog now looks like this :
  - Create an expense report.
  - Submit an expense report for approval.
  - Edit the expense report.
  - Delete an expense report.
  - Log into the expense reporting system.
  - Create a user.
  - Edit user.
  - Delete user.
  - Create a customer.
  - Edit customer.
  - Delete a customer.
- Each of these PBIs is associated with tasks that are part of the complete sprint backlog.

Forecasting in Azure DevOps

- There is a nice feature in Azure DevOps that allows you to create a forecast of how much work you might have in each sprint. It requires you to fill out an effort estimate for each work item. In the example in Fig. 6.9.20, you see estimated story points in the Effort column. Forecasts are based on the velocity of the five story points (on this screen) and Azure DevOps automatically draws sprints and work items to fit into each sprint. The forecast can be turned off if you don't want to see it. Simply turn on or off as shown in the Fig. 6.9.20.

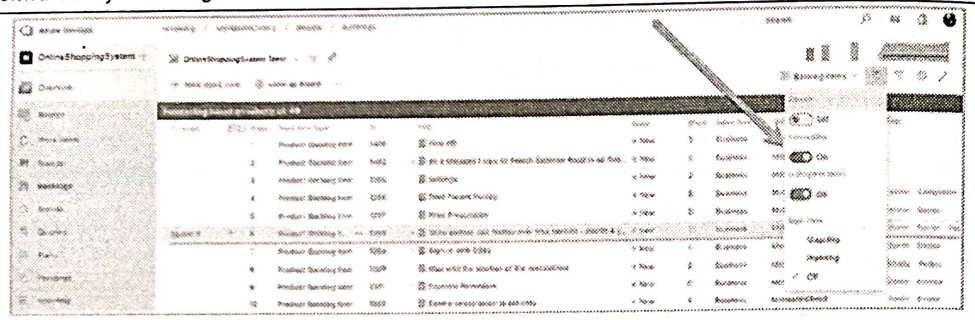


Fig. 6.9.20 : Forecast in Azure DevOps

Sprint planning

- Most of the work in a sprint planning meeting consists of dividing the user stories into tasks and estimating the time for each task. The team starts with the main story from the product backlog, breaks it down and places it on the sprint backlog. They continue to do this until the time available for the sprint is used up.
- In Azure DevOps, you can add tasks to a user story in several different ways. From the PBI, you can go to the Links tab and click the Add Link Work Item icon as shown in the image

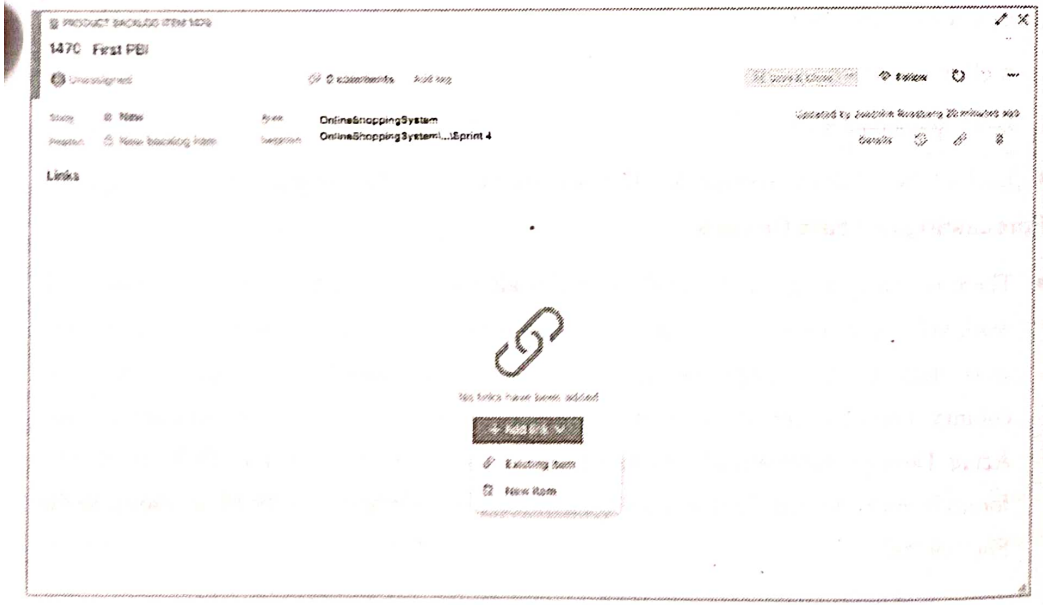


Fig. 6.9.21 : Adding a new task from Inside a user story

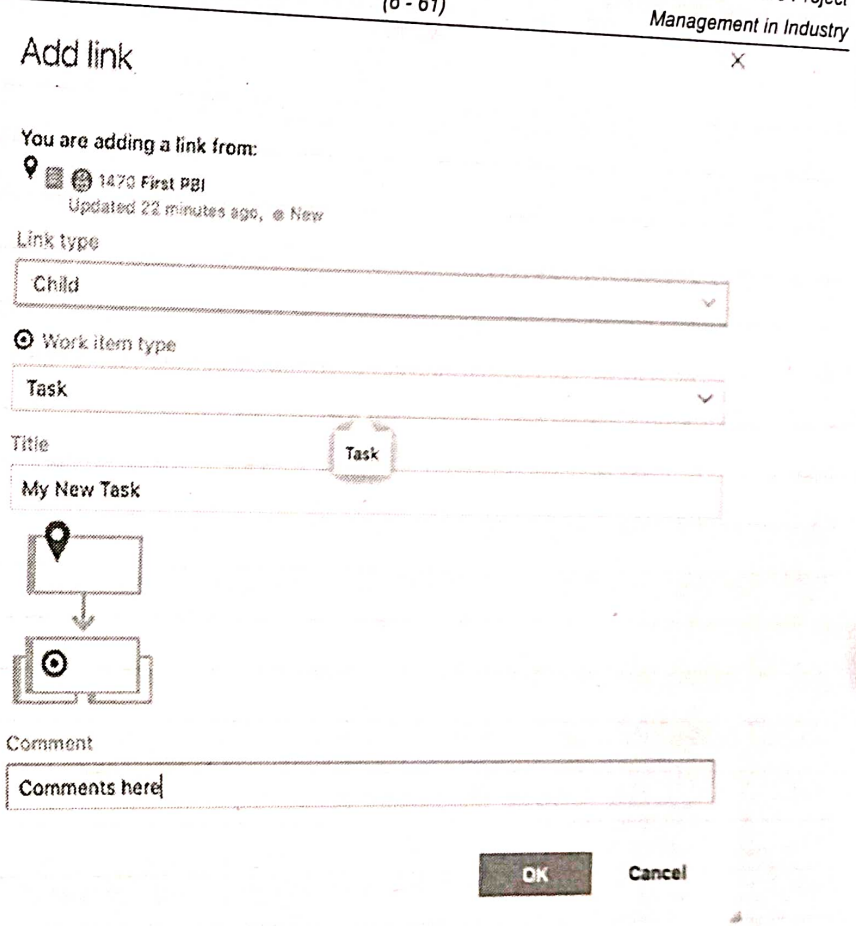


Fig. 6.9.22 : Form for adding a new task

Summary

- **This chapter followed** Product owner' as he starts online shopping project. It explains how the project is initiated started to use Azure DevOps to run it after a Go decision was made. Product owner used Azure DevOps, but he could also have been using on-premise TFS as well.

Review Question

1. How Azure DevOps can support the agile project management process during sprints ? Explain with example.

❑❑❑

SOLVED MODEL QUESTION PAPER (End Sem)

Software Project Management

B.E. (IT) Semester - VII (As Per 2019 Pattern)

Time : $2\frac{1}{2}$ Hours] **[Maximum Marks : 70**

N. B. :

1) *Attempt Q.1 or Q.2, Q.3 or Q.4, Q.5 or Q.6, Q.7 or Q.8.*
2) *Neat diagrams must be drawn wherever necessary.*
3) *Figures to the right side indicate full marks.*
4) *Assume suitable data if necessary.*

Q.1 a) *What are the objectives of activity planning ? Explain in detail.* **(Refer section 3.1)** [9]

b) *Define WBS. What are the advantages of WBS ?* **(Refer section 3.3.1)** [3]

c) *Write a note on Hybrid approach with suitable diagram.* **(Refer section 3.3.1(Point (3))** [6]

OR

Q.2 a) *With the neat sketch explain formulating a network model.* **(Refer section 3.6)** [9]

b) *Explain sequencing and scheduling activities with example.* **(Refer section 3.4)** [5]

c) *Differentiate between forward and backward pass techniques.* **(Refer section 3.7.2)** [2]

d) *Explain various categories for risk.* **(Refer section 3.8.2)** [2]

Q.3 a) *What are the different tools and methods used for monitoring and regulating project operations?* **(Refer section 4.2)** [3]

b) *Explain different tool in detail.* **(Refer section 4.3.1)** [9]

c) *What is Red / Amber / Green (RAG) reporting ?* **(Refer section 4.4.1)** [5]

OR

Q.4 a) *Differentiate Questionnaire vs. Schedule.* **(Refer section 4.3.2)** [3]

b) *Explain visualizing progress in detail.* **(Refer section 4.6)** [9]

c) *Define the following terms : a. Kanban boards b. Project calendars* **(Refer sections 4.8 and 4.9)** [5]

Q.5 a) *How to select a right person for the job ? Explain the recruitment process in detail.* **(Refer section 5.3)** [9]

b) *What are the different methods for improving motivation ?* **(Refer section 5.5)** [3]

c) *What is health and safety in the construction and engineering projects ?* **(Refer section 5.7)** [3]

d) *How to build a team and how to work in a team. Justify the answer.* **(Refer section 5.9.2)** [3]

OR

Q.6 a) *Explain five fundamental stages of development.* **(Refer section 5.9.1)** [5]

b) *What is decision making? Explain mental obstacles to good decision making.* **(Refer section 5.10.1)** [6]

c) *What is team structure ? Discuss advantages and disadvantages of the chief programming team.* **(Refer section 5.11(Only introduction and 5.11 , B part)** [7]

Q.7 a) *Explain application lifecycle management process with its phases.* **(Refer section 6.1.4)** [6]

b) *Explain any four metrics used for developer practices.* **(Refer section 6.8.3)** [6]

c) *What is Azure board? Why we use Azure boards.* **(Refer section 6.2)** [5]

OR

Q.8 a) *What is visibility in DevOps? What are the different ways to enable the visibility in Azure DevOps?* **(Refer section 6.4)** [6]

b) *Define ALM tools. What features should be considered while choosing an ALM Tools? List some examples of ALM tools.* **(Refer section 6.1.7)** [6]

c) *Which messages are used in agile practices?* **(Refer section 6.8.2)** [5]

❑❑❑