NOTES

UNIT I

CHAPTER 1

INTRODUCTION TO SOFTWARE PROJECT MANAGEMENT

Software Project Management is an important area of study since most non-trivial software development efforts will be make use of some type of project management approach in an aim to manage the development process in such a way that the software meets its requirements and is on-time and within budget.

1.1 OVERVIEW OF PROJECT MANAGEMENT:

- Project management involves the planning, monitoring, and control of people, process, and events that occur during software development.
- Everyone manages, but the scope of each person's management activities varies according his or her role in the project.
- Software needs to be managed because it is a complex undertaking with a long duration time.
- Managers must focus on the fours P's to be successful (people, product, process, and project).
- A project plan is a document that defines the four P's in such a way as to ensure a cost effective, high quality software product.
- The only way to be sure that a project plan worked correctly is by observing that a high quality product was delivered on time and under budget.

1.1.1 Management Spectrum:

- People (recruiting, selection, performance management, training, compensation, career development, organization, work design, team/culture development)
- Product (product objectives, scope, alternative solutions, constraint tradeoffs)
- Process (framework activities populated with tasks, milestones, work products, and Quality Assurance (QA) points)
- Project (planning, monitoring, controlling)

NOTES

1.1.2 People:

- Players (senior managers, technical managers, practitioners, customers, end-users)
- Team leadership model (motivation, organization, skills)
- Characteristics of effective project managers (problem solving, managerial identity, achievement, influence and team building)

1.1.3 Software Team Organization:

- Democratic decentralized (rotating task coordinators and group consensus)
- Controlled decentralized (permanent leader, group problem solving, subgroup implementation of solutions)
- Controlled centralized (top level problem solving and internal coordination managed by team leader)

1.1.4 Factors Affecting Team Organization:

- Difficulty of problem to be solved
- Size of resulting program
- Team lifetime
- Degree to which problem can be modularized
- Required quality and reliability of the system to be built
- Rigidity of the delivery date
- Degree of communication required for the project

1.1.5 Coordination and Communication Issues:

- Formal, impersonal approaches (e.g. documents, milestones, memos)
- Formal interpersonal approaches (e.g. review meetings, inspections)
- Informal interpersonal approaches (e.g. information meetings, problem solving)
- Electronic communication (e.g. e-mail, bulletin boards, video conferencing)
- Interpersonal network (e.g. informal discussion with people other than project team members)

1.1.6 The Product:

- Software scope (context, information objectives, function, performance)
- Problem decomposition (partitioning or problem elaboration - focus is on functionality to be delivered and the process used to deliver it)

1.1.7 The Process:

- Process model chosen must be appropriate for the: customers and developers, characteristics of the product, and project development environment

NOTES

- Project planning begins with melding the product and the process
- Each function to be engineered must pass though the set of framework activities defined for a software organization
- Work tasks may vary but the Common Process Framework (CPF) is invariant (project size does not change the CPF)
- The job of the software engineer is to estimate the resources required to move each function though the framework activities to produce each work product
- Project decomposition begins when the project manager tries to determine how to accomplish each CPF activity

1.1.8 Critical Practices:

- Formal risk management
- Empirical cost and schedule estimation
- Metric-based project management
- Earned value tracking
- Defect tracking against quality targets
- People-aware program management

1.2 PROJECT:

The definition of a project as being planned assume that to a large extent we can determine how we are going to carry out a task before we start. There may be some projects of an exploratory nature where this might be quite hard. Planning is in essence thinking carefully about something before you do it and even in the case of uncertain projects this is worth doing as long as it is accepted that the resulting plans will have provisional and speculative elements. Other activities, concerning, for example, to routine maintenance, might have been performed so many times that everyone involved knows exactly what needs to be done. In these cases, planning hardly seems necessary, although procedures might need to be documented to ensure consistency and to help newcomers to the job.

Dictionary definitions of 'project' include:

- A specific plan or design
- A planned undertaking
- A large undertaking

1.2.1 Characteristics of Project:

- Non-routine tasks are involved
- Planning is required
- Specific objectives are to be met or a specified product is to be created

NOTES

- The project has a pre-determined time span
- Work is carried out for someone other than yourself
- Work involves several specialism's
- Work is carried out in several phases
- The resources that are available for use on the project are constrained
- The project is large or complex

1.2.2 Software Projects versus Other Types of Project:

Many of the techniques of general project management are applicable to software project management. One way of perceiving software project management is as the process of making visible that which is invisible.

Invisibility: When a physical artifact such as a bridge or road is being constructed the progress being made can actually be seen. With software, progress is not immediately visible.

Complexity: **S**oftware products contain more complexity than other engineered artifacts.

Conformity: The 'traditional' engineer is usually working with physical systems and physical materials like cement and steel. These physical systems can have some complexity, but are governed by physical laws that are consistent. Software developers have to conform to the requirements of human clients. It is not just that individuals can be inconsistent.

Flexibility: The ease with which software can be changed is usually seen as one of its strengths. However, this means that where the software system interfaces with a physical or organizational system, it is expected that, where necessary, the software will change to accommodate the other components rather than vice versa. This means the software systems are likely to be subject to a high degree of change.

An example for infrastructure project is construction of a flyover. An example for a software project is development of a payroll management system for an organization using Oracle l0g and Oracle Forms 10G.

1.3 ACTIVITIES COVERED BY SOFTWARE PROJECT MANAGEMENT:

The activities covered by Software Project management are diagrammatically illustrated as follows:

NOTES

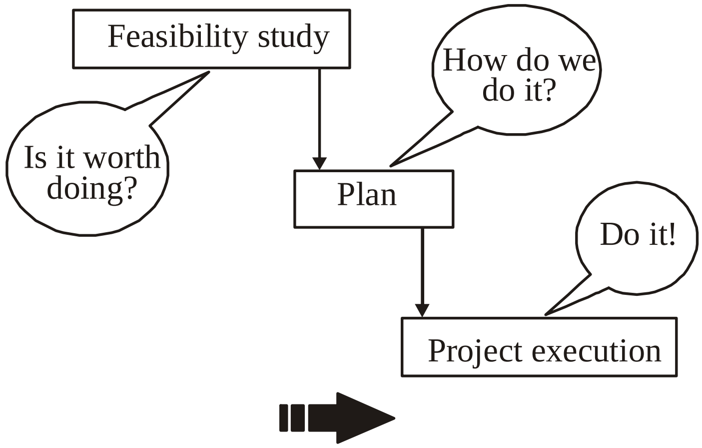

Figure 1.1: The Feasibility Study / Plan / Execution Cycle

1.3.1 The Feasibility Study:

This is an investigation into whether a prospective project is worth starting. Information is gathered about the requirements of the proposed application. The probable developmental and operational costs, along with the value of the benefits of the new system, are estimated. The study could be part of a strategic planning exercise examining and prioritizing a range of potential software developments.

1.3.2 Planning:

If the feasibility study produces results which indicate that the prospective project appears viable, planning of the project can take place. However, for a large project, we would not do all our detailed planning right at the beginning. We would formulate an outline plan for the whole project and a detailed one for the first stage. More detailed planning of the later stages would be done as they approached. This is because we would have more detailed and accurate information upon which to base our plans nearer to the start of the later stages.

1.3.3 Project Execution:

The project can now be executed. The execution of a project often contains design and implementation subphases. The same is illustrated in Figure 1.2.

NOTES

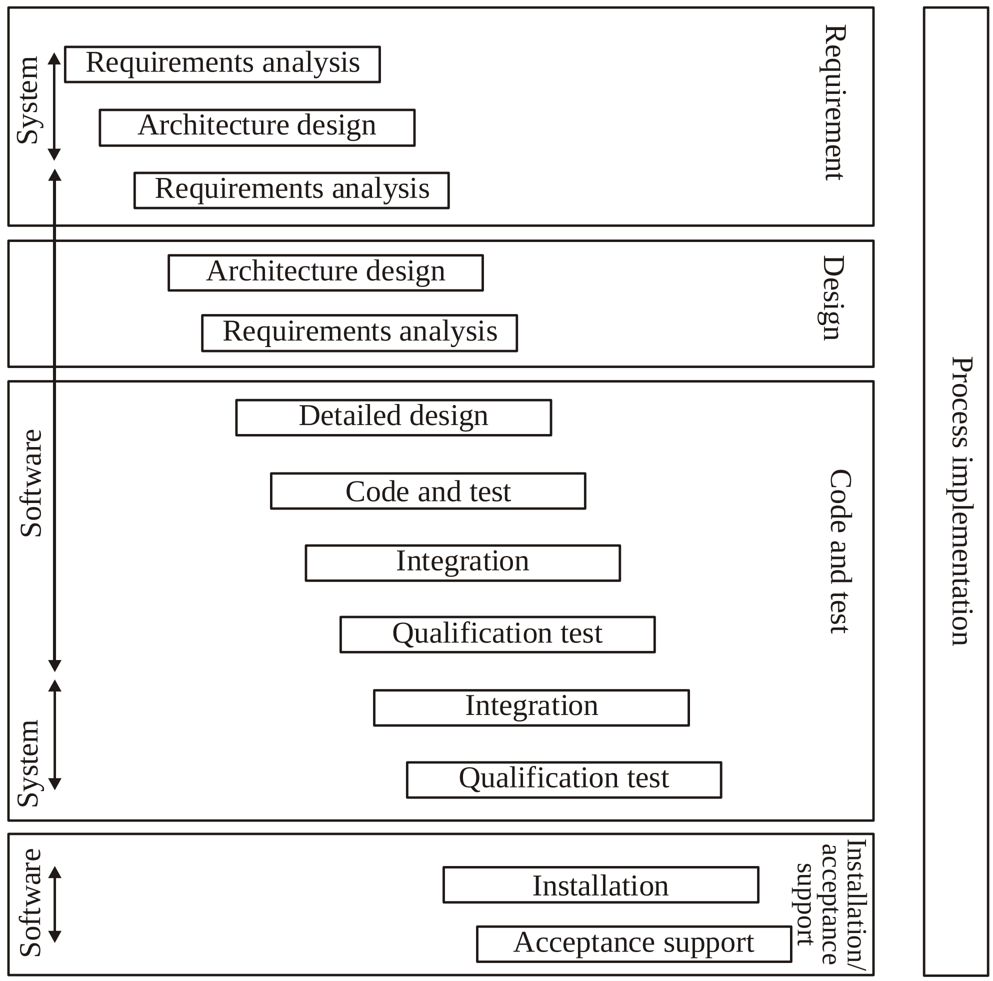

Figure 1.2: Sub Phases in Project Execution

1.3.3.1 Requirements Analysis:

This is finding out in detail what the users require of the system that the project is to implement. Some work along these lines will almost certainly have been carried out when the project was evaluated, but now the original information obtained needs to be updated and supplemented.

1.3.3.2 Specification:

Detailed documentation of what the proposed system is to do.

1.3.3.3 Design:

A design has to be drawn up which meets the specification. This design will be in two stages. One will be the external or user design concerned with the external appearance of the application. The other produces the physical design which tackles the way that the data and software procedures are to be structured internally.

NOTES

Architecture Design:

This maps the requirements to the components of the system that is to be built. At the system level, decisions will need to be made about which processes in the new system will be carried out by the user and which can be computerized. This design of the system architecture thus forms an input to the development of the software requirements. A second architecture design process then takes place which maps the software requirements to software components.

Detailed Design:

Each software component is made up of a number of software units that can be separately coded and tested. The detailed design of these units is carried out separately.

1.3.3.4 Coding:

This may refer to writing code in a procedural language or an object oriented language or could refer to the use of an application-builder. Even where software is not being built from scratch, some modification to the base package could be required to meet the needs of the new application.

1.3.3.5 Verification and Validation:

Whether software is developed specially for the current application or not, careful testing will be needed to check that the proposed system meets its requirements.

Integration:

The individual components are collected together and tested to see if they meet the overall requirements. Integration could be at the level of software where different software components are combined, or at the level of the system as a whole where the software and other components of the system such as the hardware platforms and networks and the user procedures are brought together.

Qualification Testing:

The system, including the software components, has to be tested carefully to ensure that all the requirements have been fulfilled.

1.3.3.6 Implementation/ Installation:

Some system development practitioners refer to the whole of the project after design as ‘implementation’ (that is, the implementation of the design) while others insist that the term refers to the installation of the system after the software has been developed.

1.3.3.7 Maintenance and Support:

Once the system has been implemented there is a continuing need for the correction of any errors that may have crept into the system and for extensions and improvements to the system. Maintenance and support activities may be seen as a series of minor software projects

NOTES

1.4 ACTIVITIES INVOLVED IN MANAGEMENT:

Management involves the following activities:

- Planning – deciding what is to be done
- Organizing – making arrangements
- Staffing – selecting the right people for the job, etc
- Directing – giving instructions
- Monitoring – checking on progress
- Controlling – taking action to remedy hold-ups
- Innovating – coming up with new solutions
- Representing – liaising with clients, users, developer, suppliers and other stake-holders

1.5 PROBLEMS WITH SOFTWARE PROJECTS:

Commonly experienced problems from the manager's point of view include:

- Poor estimates and plans
- Lack of quality standards and measures
- Lack of guidance about making organizational decisions
- Lack of techniques to make progress visible
- Poor role definition – who does what?
- Incorrect success criteria

Other Problems include:

- Inadequate specification of work
- Management ignorance of it
- Lack of knowledge of application area
- Lack of standards
- Lack of up-to-date documentation
- Preceding activities not completed on time – including late delivery of equipment
- Lack of communication between users and technicians
- Lack of communication leading to duplication of work
- Lack of commitment – especially when a project is tied to one person who then moves
- Narrow scope of technical expertise
- Changing statutory requirements
- Changing software environment
- Deadline pressure

NOTES

- Lack of quality control
- Remote management
- Lack of training

1.6 CONTRACT MANAGEMENT AND TECHNICAL PROJECT MANAGEMENT:

Many organizations contract out development to outside specialist developers. In such cases, the client organization will often appoint a 'project manager' to supervise the contract. This project manager will be able to delegate many technically oriented decisions to the contractors. For instance, the project manager will not be concerned about estimating the effort needed to write individual software components as long as the overall project is fulfilled within budget and on time. On the supplier side, there will need to be project managers who are concerned with the more technical management issues.

1.7 PLANS, METHODS AND METHODOLOGIES

A plan for an activity must be based on some idea of a method of work. To take a simple example, if you were asked to test some software, even though you do not know anything about the software to be tested, you could assume that you would need to:

- Analyze the requirements for the software
- Devise and write test cases that will check that each requirement has been satisfied
- Create test scripts and expected results for each test case
- Compare the actual results and the expected results and identify discrepancies
- While a method relates to a type of activity in general, a plan takes that method (and perhaps others) and converts it to real activities, identifying for each activity:
- Its start and end dates
- Who will carry it out?
- What tools and materials will be used?

'Materials' in this context could include information, for example a requirements document. With complex procedures, several methods may be deployed, in sequence or in parallel. The output from one method might be the input to another. Groups of methods or techniques are often referred to as methodologies.

1.8 OBJECTIVES VERSUS PRODUCTS:

Projects may be distinguished by whether their aim is to produce a product or to meet certain objectives.

A project might be to create a product, the details of which have been specified by the client. The client has the responsibility for justifying the product. On the other hand, the project may be required to meet certain objectives. There could be several ways of achieving

NOTES

these objectives. A new information system might be implemented to improve some service to users inside or outside an organization. The level of service that is the target would be the subject of an agreement rather than the internal characteristics of a particular information system.

Many software projects have two stages. The first stage is an objectives-driven project which results in a recommended course of action and may even specify a new software system to meet identified requirements. The next stage is a project actually to create the software product.

1.8.1 Setting Objectives

To have a successful software project, the manager and the project team members must know what will constitute success. This will make them concentrate on what is essential to project success.

There may be several sets of users of a system and there may be several different groups of specialists involved its development. There is a need for well-defined objectives that are accepted by all these people. Where there is more than one user group, a project authority needs to be identified which has overall authority over what the project is to achieve.

This authority is often held by a project steering committee (or project board or project management board) which has overall responsibility for setting, monitoring and modifying objectives. The project manager still has responsibility for running the project on a day-to-day basis, but has to report to the steering committee at regular intervals. Only the steering committee can authorize changes to the project objectives and resources.

Sub-objectives and Goals:

Setting objectives can guide and motivate individuals and groups of staff. An effective objective for an individual must be something that is within the control of that individual. An objective might be that the software application to be produced must pay for itself by reducing staff costs over two years. As an overall business objective this might be reasonable. For software developers it would be unreasonable as, though they can control development costs, any reduction in operational staff costs depends not just on them but on the operational management after the application has ‘gone live’. What would be appropriate would be to set a goal or sub-objective for the software developers to keep development costs within a certain budget.

Thus, objectives will need be broken down into goals or sub-objectives. Here we say that in order to achieve the objective we must achieve certain goals first. These goals are steps on the way to achieving an objective, just as goals scored in a football match are steps towards the objective of winning the match.

NOTES

The mnemonic SMART is sometimes used to describe well defined objectives:

- Specific: Effective objectives are concrete and well defined. Vague aspirations such as ‘to improve customer relations’ are unsatisfactory. Objectives should be defined in such a way that it is obvious to all whether the project has been successful or not.
- Measurable: Ideally there should be measures of effectiveness which tell us how successful the project has been. For example, ‘to reduce customer complaints’ would be more satisfactory as an objective than ‘to improve customer relations’. The measure can, in some cases, be an answer to simple yes/no questions, e.g. ‘Can we install the new software by 1 November 2011?’
- Achievable: It must be within the power of the individual or group to achieve the objective.
- Relevant: The objective must be relevant to the true purpose of the project.
- Time constrained: There should be a defined point in time by which the objective should have been achieved.

Measures of effectiveness provide practical methods of ascertaining whether an objective has been met. A measure of effectiveness will usually be related to the installed operational system.

1.9 STAKEHOLDERS:

These are people who have a stake or interest in the project. It is important that they be identified as early as possible, because you need to set up adequate communication channels with them right from the start. The project leader also has to be aware that not everybody who is involved with a project has the same motivation and objectives. The end-users might, for instance, be concerned about the ease of use of the system while their managers might be interested in the staff savings the new system will allow.

Stakeholders might be internal to the project team, external to the project team but in the same organization, or totally external to the organization.

- Internal to the project team: This means that they will be under the direct managerial control of the project leader.
- External to the project team but within the same organization: For example, the project leader might need the assistance of the information management group in order to add some additional data types to a database or the assistance of the users to carry out systems testing. Here the commitment of the people involved has to be negotiated.
- External to both the project team and the organization: External stakeholders may be customers (or users) who will benefit from the system that the project implements or contractors who will carry out work for the project. One feature of the relationship with these people is that it is likely to be based on a legally binding contract.

NOTES

Different types of stakeholder may have different objectives and one of the jobs of the successful project leader is to recognize these different interests and to be able to reconcile them. It should therefore come as no surprise that the project leader needs to be a good communicator and negotiator.

1.10 THE BUSINESS CASE:

Any project plan must ensure that the business case is kept intact. For example:

- The development costs are not allowed to rise to a level which threatens to exceed the value of benefits
- The features of the system are not reduced to a level where the expected benefits cannot be realized
- The delivery date is not delayed so that there is an unacceptable loss benefits

1.11 REQUIREMENT SPECIFICATION

Very often, especially in the case of product-driven projects, the objectives of the project are carefully defined in terms of functional requirements, quality requirements, and resource requirements.

- Functional requirements: These define what the end-product of the project is to do.
- Quality requirements: There will be other attributes of the application to be implemented that do not relate so much to what the system is to do but how it is to do it. These are still things that the user will be able to experience. They include, for example, response time, the ease of using the system and its reliability.
- Resource requirements: A record of how much the organization is willing to spend on the system. There may be a trade-off between this and the time it takes to implement the system. In general it costs disproportionately more to implement a system by an earlier date than a later one. There may also be a trade-off between the functional and quality requirements and cost. We would all like exceptionally reliable and user-friendly systems which do exactly what we want but we may not be able to afford them.

All these requirements must be consistent with the business case.

Summary:

Software project management is the art and science of planning and leading software projects. It is a sub-discipline of project management in which software projects are planned, monitored and controlled.

Project management is the discipline of planning, organizing, securing and managing resources to bring about the successful completion of specific project goals and objectives. It is sometimes conflated with program management, however technically that is actually a higher level construction: a group of related and somehow interdependent engineering projects.

Questions:

1. What is a project? List the characteristics of project.
2. List and discuss the activities covered by software project management.
3. List and discuss the activities involved in management.
4. List the problems with software projects.
5. What does the mnemonic SMART describe?
6. Distinguish between functional and non-functional requirements.

NOTES

NOTES

CHAPTER 2

AN OVERVIEW OF PROJECT PLANNING

Planning in organizations and public policy is both the organizational process of creating and maintaining a plan; and the psychological process of thinking about the activities required to create a desired goal on some scale. Process is an ongoing collection of activities, with inputs, outputs and the energy required to transform inputs to outputs. Process Architecture is the structural design of general process systems and applies to fields such as computers (software, hardware, networks, etc.), business processes (enterprise architecture, policy and procedures, logistics, project management, etc.), and any other process system of varying degrees of complexity. Process Management is the ensemble of activities of planning and monitoring the performance of a process, especially in the sense of business process.

2.1 STEPWISE PLANNING ACTIVITIES:

The Step wise project planning activities is tabulated in Table 2.1.

Table 2.1: An Outline of Step Wise Planning Activities

Step	Activity
0	Select Project
1	Identify Project Scope and Objectives: 1.1 Establish a project authority 1.2 Identify stakeholders 1.3 Modify objectives in the light of stakeholder analysis 1.4 Establish methods of communications with all parties
2	Identify project infrastructure: 2.1 Establish relationship between project and strategic planning 2.2 Identify installation standards and procedures 2.3 Identify project team organization
3	Analyze project characteristics: 3.1 Distinguish the project as either objective-or product-driven 3.2 Analyse other project characteristics 3.3 Identify high-level project risks 3.4 Take into account user requirements concerning implementation 3.5 Select general life-cycle approach 3.6 Review overall resource estimates

NOTES

Step	Activity
4	Identify project products and activities: 4.1 Identify and describe project products(including quality criteria) 4.2 Document generic product flows 4.3 Recognize product instances 4.4 Produce ideal activity network 4.5 Modify ideal to take into account need for stages and checkpoints
5	Estimate effort for each activity: 5.1 Carry out bottom-up estimates 5.2 Revise plan to create controllable activities
6	Identify activity risks: 6.1 Identify and qualify activity-based risks 6.2 Plan risk reduction and contingency measures where appropriate 6.3 Adjust plans and estimates to take account of risks
7	Allocate resources: 7.1 Identify and allocate resources 7.2 Revise plans and estimates to take account of resource constraints
8	Review/publicize plan: 8.1 Review quality aspects of project plan 8.2 Document plans and obtain agreement
9/10	Execute plan/lower levels of planning : This may require the reiteration of the planning process at a lower level

The Step wise project planning activities is diagrammatically illustrated in Figure 2.1.

NOTES

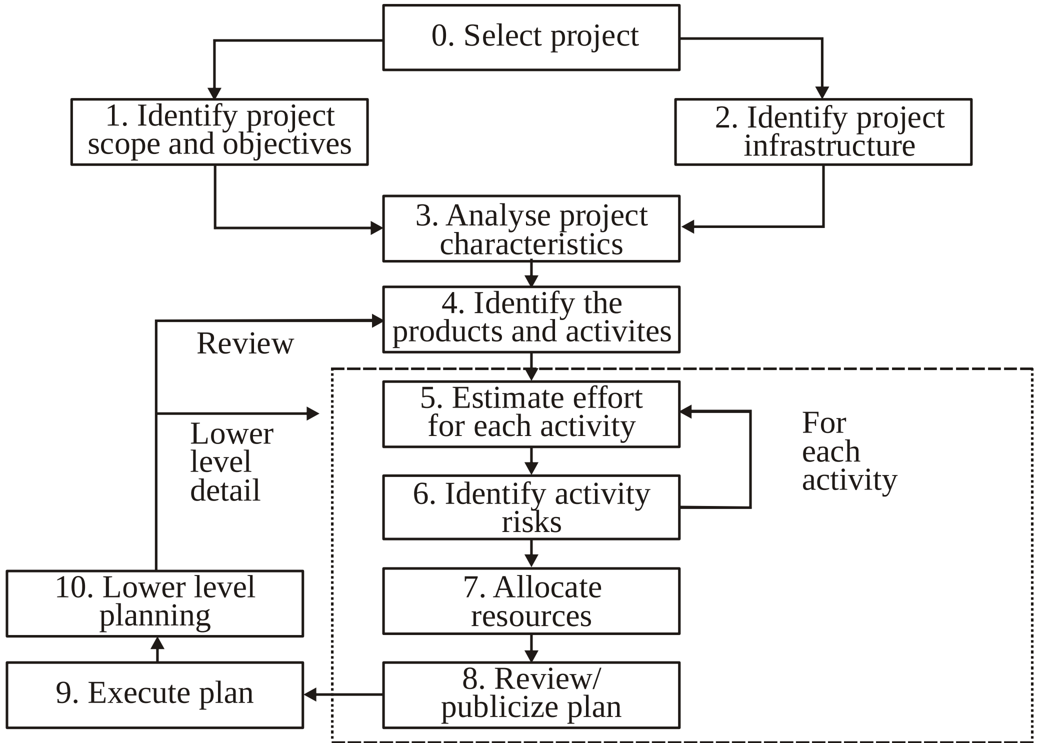

Figure 2.1: Step Wise Project Planning Activities

2.2 STEP 0: SELECT PROJECT:

This is called Step 0 because in a way it is outside the main project planning process. Proposed projects do not appear out of thin air, some process must decide to initiate this project rather than some other. While a feasibility study might suggest that the project is worthwhile, it would still need to be established that it should have priority over other projects found to be worthwhile. This evaluation of the merits of projects could be part of strategic planning.

2.3 STEP 1: IDENTIFY PROJECT SCOPE AND OBJECTIVES:

The activities in this step ensure that all the parties to the project agree on the objectives and are committed to the success of the project.

Step 1.1: **Identify objectives and practical measures of the effectiveness in meeting those objectives**

Step 1.2: **Establish a project authority**

Step 1.3: **Stakeholder analysis — identify all stakeholders in the project and their interests**

Step 1.4: **Modify objectives in the light of stakeholder analysis**

NOTES

In order to gain the full cooperation of all concerned, it might be necessary to modify the project objectives. This could mean adding new features to the system which give a benefit to some stakeholders as a means of assuring their commitment to the project.

Step 1.5: Establish methods of communication with all parties

2.4 STEP 2: IDENTIFY PROJECT INFRASTRUCTURE

Step 2.1: Identify relationship between the project and strategic planning

As well as identifying projects to be carried out, an organization needs to give priorities to the projects to be carried out. It also needs to establish the framework within which the proposed new systems are to fit. Hardware and software standards, for example, are needed so that various systems can communicate with each other. These strategic decisions must be documented in a strategic business plan or in an information technology plan that is developed from the business plan.

Step 2.2: Identify installation standards and procedures

Any organization that develops software should define their development procedures. As a minimum, the normal stages in the software life cycle to be carried out should be documented along with the products created at each stage.

Change control and configuration management standards should be in place to ensure that changes to requirements are implemented in a safe and orderly way.

The procedural standards may lay down the quality checks that need to be done at each point of the project life cycle or these may be documented in a separate quality standards and procedures manual.

The organization as part of its monitoring and control policy may have a measurement programme in place which dictates that certain statistics have to be collected at various stages of a project.

Finally the project manager should be aware of any project planning and control standards. These will relate to how the project is controlled: for example, the way that the hours spent by team members on individual tasks are recorded on timesheets.

Step 2.3: Identify project team organization

Project leaders, especially in the case of large projects, might have some control over the way that their project team is to be organized. Often, though, the organizational structure will be dictated to them. For example, a high-level managerial decision might have been taken that software developers and business analysts will be in different groups, or that the development of business-to-customer web applications will be done within a separate group from that responsible for ‘traditional’ database applications.

NOTES

2.5 STEP 3: ANALYSE PROJECT CHARACTERISTICS

The general purpose of this part of the planning operation is to ensure that the appropriate methods are used for the project.

Step 3.1: Distinguish the project as either objective- or product-driven

As development of a system advances it tends to become more product-driven, although the underlying objectives always remain and must be respected.

Step 3.2: Analyse other project characteristics (including quality-based ones)

For example, is an information system to be developed or a process control system, or will there be elements of both? Will the system be safety critical, where human life could be threatened by a malfunction?

Step 3.3: Identify high-level project risks

Consideration must be given to the risks that threaten the successful outcome of the project. Generally speaking, most risks can be attributed to the operational or development environment, the technical nature of the project or the type of product being created.

Step 3.4: Take into account user requirements concerning implementation

The clients may have their own procedural requirements. For example, an organization might mandate the use of a particular development method.

Step 3.5: Select development methodology and life-cycle approach

The development methodology and project life cycle to be used for the project will be influenced by the issues raised above. For many software developers, the choice of methods will seem obvious because they will use the ones that they have always used in the past.

While the setting of objectives involves identifying the problems to be solved, this part of planning is working out the ways in which these problems are to be solved. For a project that is novel to the planner, some research into the methods typically used in the problem domain is worthwhile. For example, sometimes, as part of a project, a questionnaire survey has to be conducted.

Step 3.6: Review overall resource estimates

Once the major risks have been identified and the broad project approach has been decided upon, this would be a good point at which to reestimate the effort and other resources required to implement the project. Where enough information is available a function point-based estimate might be appropriate.

NOTES

2.6 STEP 4: IDENTIFY PROJECT PRODUCTS AND ACTIVITIES

The more detailed planning of the individual activities now takes place. The longer-term planning is broad and in outline, while the more immediate tasks are planned in some detail.

Step 4.1: Identify and Describe Project Products (or Deliverables)

In general there can be no project products that do not have activities that create them. Wherever possible, we ought also to ensure the reverse: that there are no activities that do not produce a tangible product. Identifying all the things the project is to create helps to ensure that all the activities we need to carry out are accounted for. Some of these products will be handed over to the client at the end of the project — these are deliverables. Other products might not be in the final configuration, but are needed as intermediate products used in the process of creating the deliverables.

These products will include a large number of technical products, including training material and operating instructions, but also products to do with the management and the quality of the project. Planning documents would, for example, be management products.

The products will form a hierarchy. The main products will have sets of component products which in turn may have sub-component products and so on. These relationships can be documented in a Product Breakdown Structure (PBS). The same is illustrated in Figure 2.2. In this example the products have been grouped into those relating to the system as a whole, and those related to individual modules. A third group', which happens to have only one product, is called management products' and consists of progress reports. The asterisk in the progress reports indicates that these will have new instances created repeatedly throughout the project.

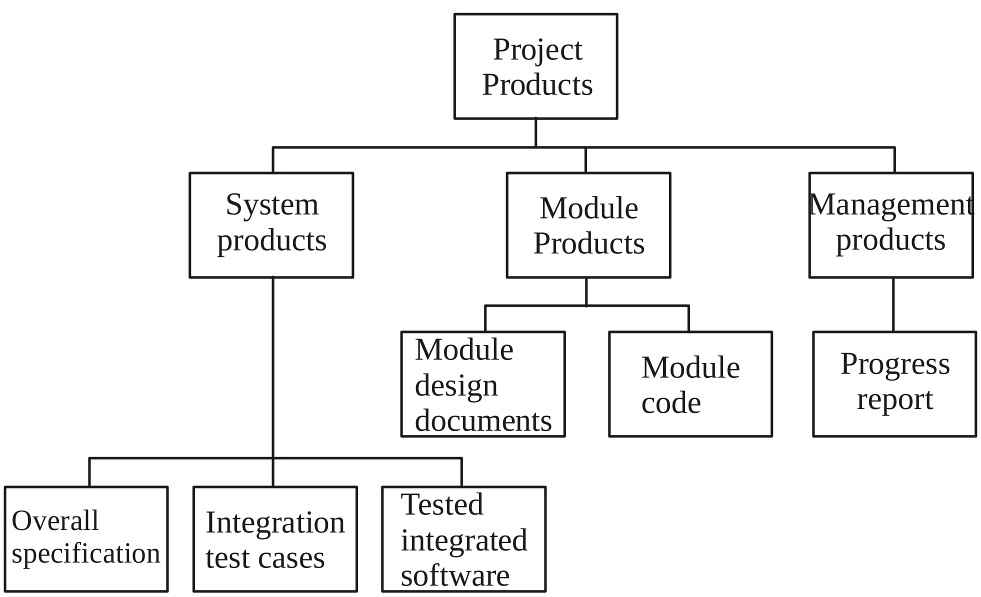

Figure 2.2: A Fragment of a Product Breakdown Structure for a System Development Task

NOTES

Some products are created from scratch, for example new software components. A product could quite easily be a document, like a software design document. It might be a modified version of something that already exists, such as an amended piece of code. A product could even be a person, such as a 'trained user', a product of the process of training. A product is the result of an activity. A common error is to identify as products things that are really activities, such as 'training', 'design' and 'testing'. Specifying 'documentation' as a product should also be avoided. These specify that products at the bottom of the PBS should be documented by Product Descriptions which contain:

- The name / identity of the product;
- The purpose of the product;
- The derivation of the product (that is, the other products from which it is derived);
- The composition of the product;
- The form of the product;
- The relevant standards;
- The quality criteria that should apply to it.

Figure 2.3 illustrates a fragment of a PFD for a Software Development Task.

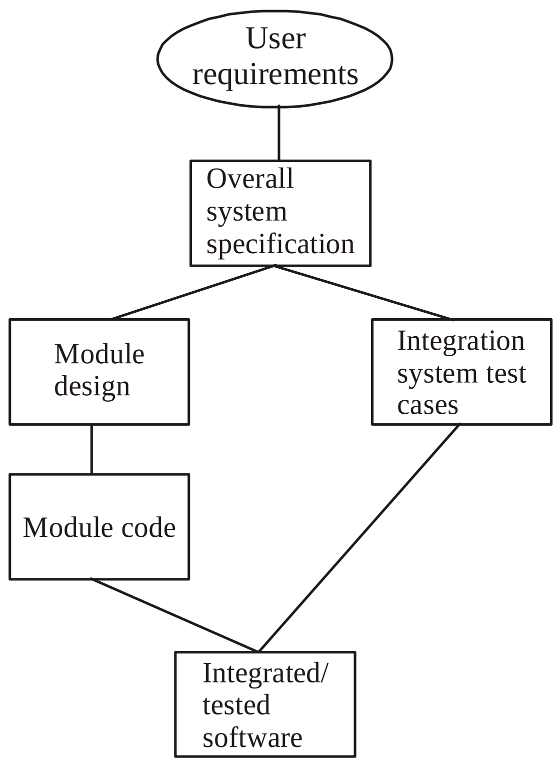

Figure 2.3: A Fragment of a PFD for a Software Development Task

Step 4.2: Document generic product flows

Some products will need one or more other products to exist first before they can be created. For example, a program design must be created before the program can be written and the program specification must exist before the design can be commenced. These relationships can be portrayed in a Product Flow Diagram (PFD). Figure 2.3 gives an example. Note that the 'flow' in the diagram is assumed to be from top to bottom and left to right. In the example in Figure 2.3, 'user requirements' is in an oval which means that it is used by the project but is not created by it. It is often convenient to identify an overall product at the bottom of the diagram, in this case 'integrated/tested software', into which all the other products feed.

Step 4.3: Recognize product instances

Where the same generic PFD fragment relates to more than one instance of a particular type of product, an attempt should be made to identify each of those instances. In the example in Figure 2.2, it could be that in fact there are just two component software modules in the software to be built.

Step 4.4: Produce ideal activity network

In order to generate one product from another there must be one or more activities which carry out the transformation. By identifying these activities we can create an activity network which shows the tasks that have to be carried out and the order in which they have to be executed. An example of an activity network is illustrated in Figure 2.4.

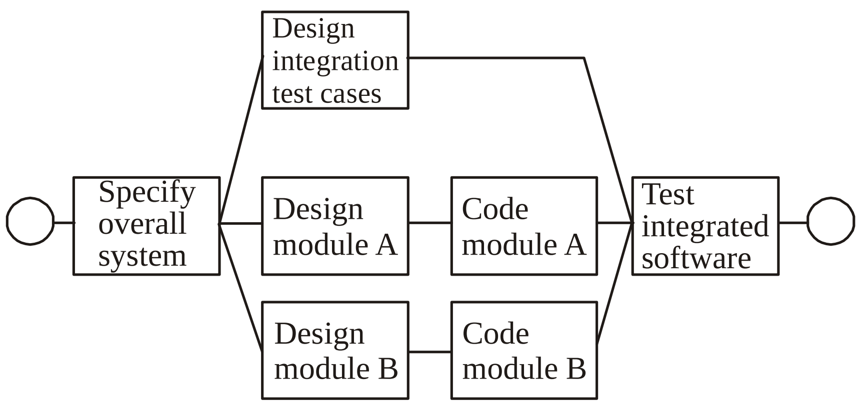

Figure 2.4: An Example of an Activity Network.

The activity networks are 'ideal' in the sense that no account has been taken of resource constraints. For example, in Figure 2.4, it is assumed that resources are available for both the software modules to be developed in parallel. The rule is that activity networks are never amended to take account of resource constraints.

Step 4.5: Modify the ideal to take into account need for stages and checkpoints

The approach to sequencing activities described above encourages the formulation of a plan which will minimize the overall duration, or 'elapsed time', for the project. It

NOTES

assumes that an activity will start as soon as the preceding ones upon which it depends have been completed.

There might, however, be a need to modify this by dividing the project into stages and introducing checkpoint activities. These are activities which draw together the products of preceding activities to check that they are compatible. This could potentially delay work on some elements of the project and there has to be a trade-off between efficiency and quality.

The people to whom the project manager reports could decide to leave the routine monitoring of activities to the project manager. However, there could be some key activities, or milestones, which represent the completion of important stages of the project of which they would want to take particular note. Checkpoint activities are often useful milestones.

2.7 STEP 5: ESTIMATE EFFORT FOR EACH ACTIVITY

Step 5.1: Carry out bottom-up estimates

At this point, estimates of the staff effort required, the probable elapsed time and the non-staff resources needed for each activity will need to be produced. The method of arriving at each of these estimates will vary depending on the type of activity.

The difference between elapsed time and effort should be noted. Effort is the amount of work that needs to be done. If a task requires three members of staff to work for two full days each, the effort expended is six days. Elapsed time is the time between the start and end of a task. In our example above, if the three members of staff start and finish at the same time then the elapsed time for the activity would be two days.

The individual activity estimates of effort should be summed to get an overall bottom-up estimate which can be reconciled with the previous top-down estimate.

The activities on the activity network can be annotated with their elapsed times so that the overall duration of the project can be calculated.

Step 5.2: Revise plan to create controllable activities

The estimates for individual activities could reveal that some are going to take quite a long time. Long activities make a project difficult to control. If an activity involving system testing is to take 12 weeks, it would be difficult after six weeks to judge accurately whether 50% of the work is completed. It would be better to break this down into a series of smaller subtasks.

There might be a number of activities that are important, but individually take up very little time.

In general, try to make activities about the length of the reporting period used for monitoring and controlling the project. If you have a progress meeting every two weeks,

NOTES

then it would convenient to have activities of two weeks' duration on average, so that progress meetings would normally be made aware of completed tasks each time they are held.

2.8 STEP 6: IDENTIFY ACTIVITY RISKS

Step 6.1: Identify and quantify activity-based risks

Risks inherent in the overall nature of the project have already been considered in Step 3. We now want to look at each activity in turn and assess the risks to its successful outcome. Any plan is always based on certain assumptions. Say the design of a component is planned to take five days. This is based on the assumption that the client's requirement is clear and unambiguous. If it is not then additional effort to clarify the requirement would be needed. The possibility that an assumption upon which a plan is based is incorrect constitutes a risk. In this example, one way of expressing the uncertainty would be to express the estimate of effort as a range of values.

A project plan will be based on a huge number of assumptions, and so some way of picking out the risks that are most important is needed. The damage that each risk could cause and the likelihood of it occurring have to be gauged. This assessment can draw attention to the most serious risks. The usual effect if a problem materializes is to make the task longer or more costly.

Step 6.2: Man risk reduction and contingency measures where ever appropriate

It may be possible to avoid or at least reduce some of the identified risks. On the other hand, contingency plans specify action that is to be taken if a risk materializes. For example, a contingency plan could be to use contract staff if a member of the project team is unavailable at a key time because of serious illness.

Step 6.3: Adjust overall plans and estimates to take account of risks

We may change our plans, perhaps by adding new activities which reduce risks. For example, a new programming language might mean we schedule training courses and time for the programmers to practice their new programming skills on some non-essential work.

2.9 STEP 7: ALLOCATE RESOURCES

Step 7.1: Identify and allocate resources

The type of staff needed for each activity is recorded. The staff available for the project is identified and is provisionally allocated to tasks.

NOTES

Step 7.2: Revise plans and estimates to take into account resource constraints

Some staff may be needed for more than one task at the same time and, in this case, an order of priority is established. The decisions made here may have an effect on the overall duration of the project when some tasks are delayed while waiting for staff to become free.

Figure 2.5 illustrates a Gantt chart showing when staff will be carrying out tasks

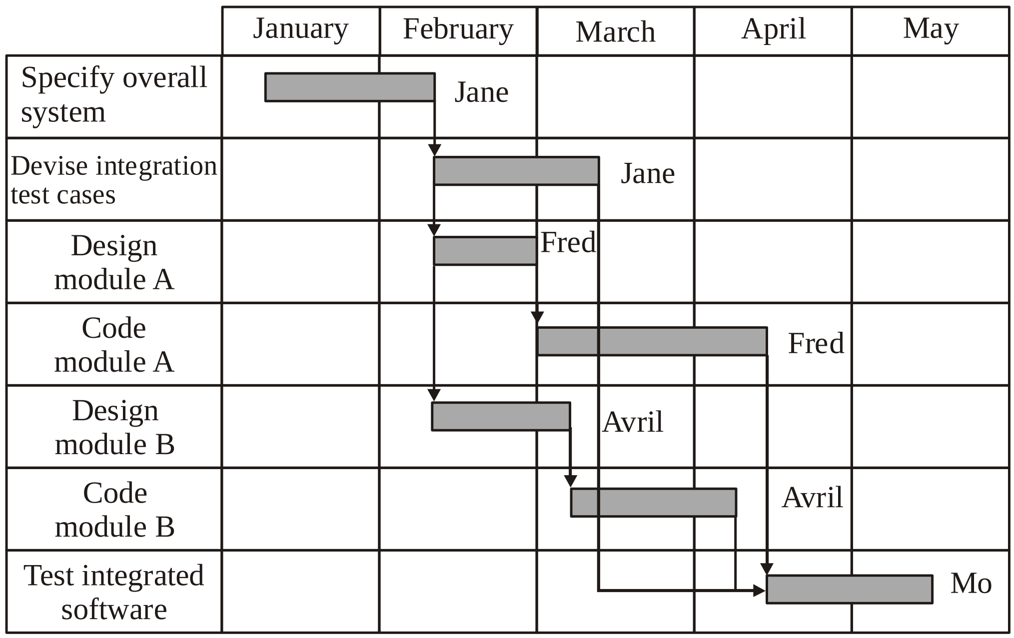

Figure 2.5: Gantt Chart Showing when Staff will be Carrying out Tasks

Ensuring someone is available to start work on an activity as soon as the preceding activities have been completed might mean that they are idle while waiting for the job to start and are therefore used inefficiently.

The product of steps 7.1 and 7.2 would typically be a Gantt chart. The Gantt chart gives a clear picture of when activities will actually take place and highlights which ones will be executed at the same time. Activity networks can be misleading in this respect

2.10 STEP 8: REVIEW / PUBLICIZE PLAN

Step 8.1: Review quality aspects of the project plan

A danger when controlling any project is that an activity can reveal that an earlier activity was not properly completed and needs to be reworked. This, at a stroke, can transform a project that appears to be progressing satisfactorily into one that is badly out of control. It is important to know that when a task is reported as completed, it really is — hence the importance of quality reviews. Each task should have quality criteria. These are quality checks that have to be passed before the activity can be signed off' as completed.

Step 8.2: Document plans and obtain agreement

NOTES

It is important that the plans be carefully documented and that all the parties to the project understand and agree to the commitments required of them in the plan. This may sound obvious, but it is amazing how often this is not done.

2.11 STEPS 9 AND 10: EXECUTE PLAN/ LOWER LEVELS OF PLANNING

Once the project is under way, plans will need to be drawn up in greater detail for each activity as it becomes due. Detailed planning of the later stages will need to be delayed because more information will be available nearer the start of the stage. It is necessary to make provisional plans for the more distant tasks, because thinking about what needs to be done can help unearth potential problems, but sight should not be lost of the fact that these plans are provisional.

In the Step Wise framework that was introduced in this chapter, Step 0 was 'Select project'. It must be confessed that this is Step 0 rather than Step 1 because when the framework was originally drawn up, the process of deciding whether or not a project should be initiated was ignored. In one way this was justifiable as project teams are brought together after the decision to embark on the project has been made, and thus they are not a party to this decision. However, the success of a project, both technically and in business terms, is clearly built upon a sound initial evaluation of the technical, organizational and financial feasibility of the project.

As will be seen, in some cases managers will be able to justify a commitment to a single project as the financial benefits will clearly exceed the costs of the implementation and operation of the new application. In other cases, managers would not be able to approve a project on its own, but can see that it enables the fulfillment of strategic objectives when combined with the outcomes of other project.

Summary:

The Project Planning Process can be summarized as follows:

- Establish project scope
- Determine feasibility
- Analyze risks
- Determine required resources
    - o Determine required human resources
    - o Define reusable software resources
    - o Identify environmental resources
- Estimate cost and effort
    - o Decompose the problem

NOTES

    - Develop two or more estimates
    - Reconcile the estimates
- Develop project schedule
    - Establish a meaningful task set
    - Define task network
    - Use scheduling tools to develop timeline chart
    - Define schedule tracking mechanisms

Questions:

1. Diagrammatically illustrate and discuss the step wise project planning activities.
2. With an example explain a product breakdown structure.
3. With an example explain an activity network.
4. What is a Product Flow Diagram (PDF) used for? Discuss with an example.
5. What is Gantt chart used for? Discuss.

NOTES

UNIT II

CHAPTER 3

PROGRAMME MANAGEMENT AND PROJECT EVALUATION

Projects can be grouped into programmes. The benefits of projects and programmes should be identified and managed. Finally, some methods of evaluating the costs and benefits of individual projects must be explored.

3.1 PROGRAMME MANAGEMENT

A programme can be defined as 'a group of projects that are managed in a coordinated way to gain benefits that would not be possible were the projects to be managed independently'. Programmes can exist in different forms. They are listed below:

Strategic Programmes

Several projects together can implement a single strategy. For example, the merging of two organizations could involve the creation of unified payroll and accounting applications, the physical reorganization of offices, training, new, organizational procedures, re-creating a corporate image through advertising and so on. Each activity could be treated as a distinct project, but would be coordinated as a programme.

Business Cycle Programmes

The collection of projects that an organization undertakes within a particular planning cycle is sometimes referred to as a portfolio. Many organizations have a fixed budget for development. Decisions have to be made about which projects to implement within that budget within the accounting period, which often coincides with the financial year. If expenditure on one project is delayed to a later year, then this could release resources for some other project. On the other hand, if a project absorbs more resources than expected, then this could be at the expense of other projects. Planners need to assess the comparative value and urgency of projects within a portfolio.

NOTES

Research and Development Programmes

Some development projects will be relatively safe, and result in the final planned product, but that product might not be radically different from existing ones on the market. Other projects might be extremely risky, but the end result, if successful, could be a revolutionary technological breakthrough that meets some pressing but previously unsatisfied need. The risks associated with an innovative project will fluctuate. Research work may lead to technological breakthroughs or to the discovery of insurmountable problems, so the portfolio of projects needs to be reviewed on a regular basis.

Innovative Partnerships:

Some technological developments, if handled properly, benefit whole industries. The development of the internet and the World Wide Web is an example. Often, technological products can only be exploited if they interoperate with other products. Companies therefore often come together to work collaboratively on new technologies in a 'pre-competitive' phase. Separate projects in different organizations need to be coordinated and this might be done as a programme.

3.2 MANAGING ALLOCATION OF RESOURCES WITHIN PROGRAMMES:

Resources have to be shared between concurrent projects. In these circumstances, programme managers will have concerns about the optimal use of specialist staff. These concerns can be contrasted with those of project managers. The same is tabulated in Table 3.1.

Table 3.1: Programme Managers versus Project Managers

Programme Manager	Project Manager
Many simultaneous projects	One project at a time
Personal relationship with skilled resources	Impersonal relationship with resource type
Need to maximize utilization of resources	Need to minimize demand for resources
Projects tend to be similar	Projects tend to be dissimilar

When a project is planned, at the stage of allocating resources, programme management will be involved. The same is diagrammatically illustrated in Figure 3.1. Some activities in the project might have to be delayed until the requisite technical staffs are freed from work on other projects. Where expensive technical staff is employed full time, having short periods of intense activity interspersed with long period of idleness must be avoided.

NOTES

When a project is executed activities can take longer (or sometimes even less time) than planned. Delays can mean that specialist staffs are prevented from moving on to their next project. Hence it can be seen that programme management needs to be concerned with the continual monitoring of the progress of projects and the use of resources.

Project managers

Programme management	Project A	Project B	Project C	Project D
Resource W	X	X		
Resource X		X		
Resource Y			X	X
Resource Z	X	X		X

Figure 3.1: Projects Sharing Resources

3.3 STRATEGIC PROGRAMME MANAGEMENT

Strategic programme management is a form of programme management is where a portfolio of projects all contributes to a common objective. For example a business which carries out maintenance work for clients. A customer's experience of the organization might be found to be very variable and inconsistent. The employee who records the customer's requirements is different from the people who actually carry out the work. Often a customer has to explain to one company employee a problem that has already been discussed at length with some other employee. A business objective might be to present a consistent and uniform front to the client. This objective might require changes to a number of different systems which until now have been largely self-contained. The work to reorganize each individual area could be treated as a separate project, coordinated at a higher level as a programme.

3.3.1 Creating a Programme

Programme Mandate

The planning of a programme will be triggered by the cre-ation of an agreed programme mandate. Ideally this should be a formal document describing the following:

- The new services or capabilities the programme should deliver
- How the organization will be improved by use of the new services or capability?
- How the programme fits with corporate goals and any other initiatives?

NOTES

At this point, a programme director ought to be appointed to provide initial leadership for the programme. To be successful, the programme needs a champion who is in a prominent position within the organization. This will signal the seriousness with which the organization takes the programme. The programme director is likely to come from the sponsoring group who have identified the need for a programme and have initiated its establishment.

Program Brief

A Program brief is now produced which would be the equivalent of a feasibility study for the programme. Ideally this should be a formal document describing the following:

- A preliminary vision statement which describes the new capacity that the organ-ization seeks.
- The benefits that the programme should create - including when they are likely to be generated and how they might be measured
- Risks and issues
- Estimated costs, timescales and effort

Vision Statement

The programme brief should have given the sponsoring group enough information to decide whether it is worth moving to a more detailed definition of the pro-gramme. This stage will involve a lot of detailed planning work and would justify the setting up of a small team. At this point a programme manager who would have day-to-day responsibility for running the programme could well be appointed. The programme manager is likely to be someone with considerable project management experience.

This group can now take the vision statement outlined in the project brief and refine and expand it. It should describe in detail the new capability that the pro-gramme will give the organization. If estimates for costs, performance, and service levels cannot be provided, then there should at least be an indication of how they might be measured. For example, the headings under which costs will be incurred can be recorded. Similarly, for performance, one might be able to say that repeat business will be increased, even if the precise size of the increase cannot be provided.

The Blueprint

The achievement of the improved capability described in the vision statement can only come about when changes have been made to the structure and oper-ations of the organization. These are detailed in the blueprint. This should contain the following:

- Business models outlining the new processes required
- Organizational structure which includes the numbers of staff required in the new systems and the skills they will need

NOTES

- The information systems, equipment and other, non-staff, resources that will be needed
- Data and information requirements
- Costs, performance and service level requirements

An organization which wants to present a consis-tent interface to its customers, while this aspiration might be stated in the vision statement, the way that it is to be achieved would have to be stated in the blue-print. This might, for example, suggest:

- The appointment of 'account managers' who could act as a point of contact for the client throughout their business transactions with the company
- A common computer interface allowing the account manager to have access to all the information relating to a particular client or job

The blueprint will need to be supported by benefit profiles which estimate when the expected benefits will start to be realized following the implementation of the enhanced capability. One principle in programme management is that the programme should deliver tangible benefits. Because an organization is given the capability to do something, this does not guarantee that the capability will be used to obtain the benefits originally envisaged. For example, the marketing department of a company might be provided with a database of sales and demographic information which would allow them to target potential customers more accurately. This could be used to focus sales and promotions more effectively and improve the ratio of sales revenue to advertising costs. However, just because this database is made available does not mean that the marketing staff will necessarily make effective use of it. Evidence of the business benefits actually experienced needs to be collected. However, the timing of the benefits needs to be carefully considered. For instance, marketing campaigns that target particular cus-tomers might take some time to plan and organize.

The management structure needed to drive this programme forward would also need to be planned and organized.

A preliminary list of the projects that the programme will need in order to achieve its objective will be created as the programme portfolio. This will contain an outline schedule to be presented to the sponsoring group with the preliminary estimated timescales for the projects.

Programmes will affect many different groups and individuals in an organiz-ation. A major risk is that some of those whose work will be affected by the programme or whose -cooperation is needed will not be drawn into the programme effectively. A stake holder map identifying the groups of people with an interest in the project and its outcomes and their particular interests could be drawn up. This can be used to write a communications strategy and plan showing how the appropriate information flows between stakeholders can be set up and maintained.

NOTES

With conventional project planning, it is not usually possible to plan all the phases of a project at the outset, as much of the information needed to produce the detailed plans will not be available until later in the project. It can easily be imagined that this is even more so with programmes. However at the initial programme planning stage, a preliminary plan can be produced containing:

- The project portfolio
- Cost estimates for each project
- The benefits expected
- Risks identified
- The resources needed to manage, support and monitor the programme

With this information a financial plan can be created. This allows higher management to put in place the budget arrangements to meet the expected costs at the times when they are due.

3.4 AIDS TO PROGRAM MANAGEMENT

Dependency Diagrams

There will often be physical and technical dependencies between projects. For example, a project to relocate staff from one building to another might not be able to start until a project to construct a new building has been completed. Dependency diagrams can be used to show these dependencies. Where projects run concurrently in a programme and pass products between one another, the dependency diagrams could become complicated.

Figure 3.2 shows a dependency diagram, the constituent parts of which are explained below.

A. System Study/ Design A project is carried out which examines the various existing IT applications in the two old organizations, analyses their functionality, and makes recommendations about how they are to be combined.

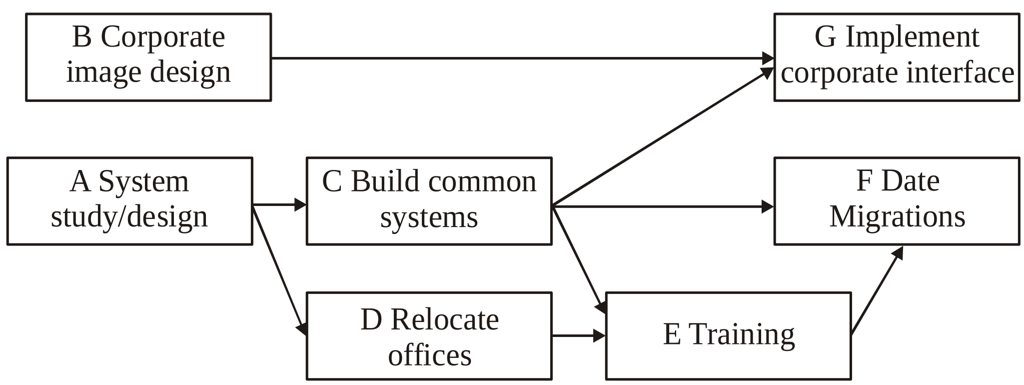

Figure 3.2: Example for Dependency Diagram

NOTES

B. Corporate Image Design: Independently of Project A, this project is designing the corporate image for the new organization. This would include design of the new logo to be put on company documents.

C. Build Common Systems: Once Project A has been completed, work can be triggered on the construction of the new common Information and Communication Technologies applications.

D. Relocate Offices: This is the project that plans and carries out the physical collo-cation of the staff in the two former organizations. In this scenario, this has to wait until the completion of Project A because that project has examined how the two sets of applications for the previous organizations could be brought together, and this has repercussions on the departmental structure of the new merged organization.

E. Training: Once staffs have been brought together, perhaps with some staff being made redundant, training in the use of the new systems can begin.

F. Data Migration: When the new, joint, applications have been developed and staff have been trained in their use, data can be migrated from existing data-bases to the new consolidated database.

G. Implement Corporate Interface: Before the new applications can 'go live', the interfaces, including the documentation generated for external customers, must be modified to conform to the new company image.

There will be interdependencies between C and D that will need to be managed.

Delivery Planning

The creation of a delivery dependency diagram would typically be a pioneer to more detailed programme planning. As part of this planning, tranches of projects could be defined. A tranche is a group of projects that will deliver their products as one step in the programme. The main criterion for grouping projects into tranches is that the deliverables of each of the projects combine to provide a coherent new capability or set of benefits for the client. An equally pressing consideration will be the need to avoid contention for scarce resources. Figure 3.3 illustrates how the programme's portfolio of projects can be organized into tranches, each of which delivers some tangible benefits to the user. At this point, the planning of individual projects can be considered. This could be initiated by the writing of project briefs, defining the scope and objectives of each project.

NOTES

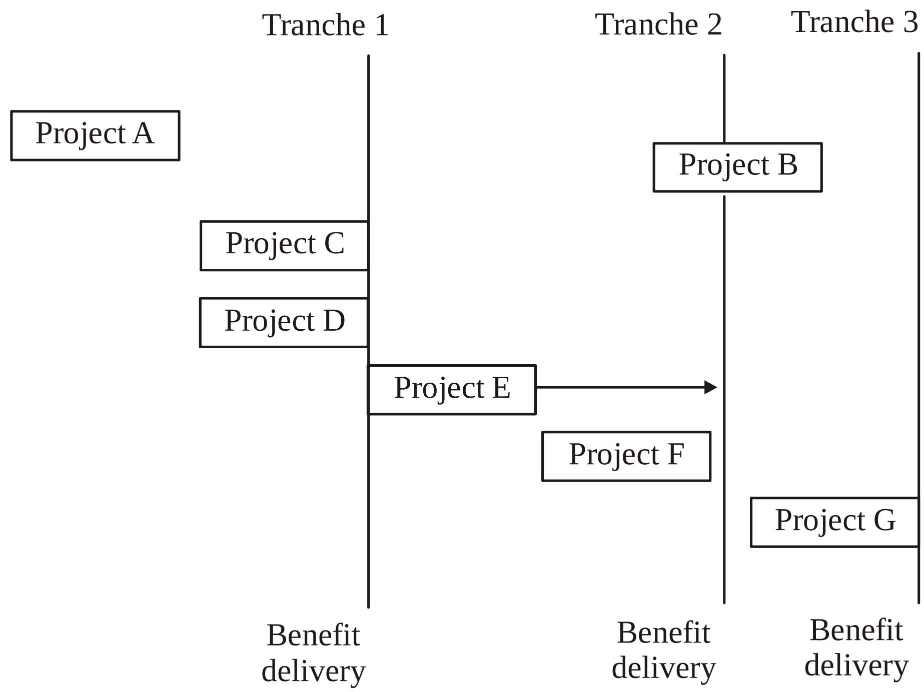

Figure 3.3: Delivering Tranches of Project Deliverables

3.5 BENEFITS MANAGEMENT

Providing an organization with a capability does not guarantee that the capability will be used to deliver the benefits originally envisaged. This is very important, because businesses have become aware of the lack of evidence of the some investments in Information and Communication Technologies increasing the productivity of organizations. Even with Business Process Reengineering (BPR), the radical reorganization of businesses to deliver improvements in efficiency and effectiveness, there are many reported cases where the expected benefits have not materialized. Benefits management encompasses the identification, optimization and tracking of the expected benefits from a business change in order to ensure that they are actually achieved.

To do this, the following points must be taken into consideration:

- Define the expected benefits from the programme
- Analyze the balance between costs and benefits
- Plan how the benefits will be achieved and measured
- Allocate responsibilities for the successful delivery of the benefits
- Monitor the realization of the benefits

Benefits can be of different types which includes the following:

- Mandatory Compliance: Governmental or legislation might make certain changes mandatory

NOTES

- Quality of Service: An insurance company, for example, might want to settle claims by customers more quickly
- Productivity: The same, or even more, work can be done at less cost in staff time
- More Motivated Work Force: This might be because of an improved rewards system, or through job enlargement or job enrichment
- Internal Management Benefits: To take an insurance example, again, better analysis of insurance claims could pinpoint those categories of business which are most risky and allow an insurance company to adjust premiums to cover this
- Risk Reduction: The insurance example might also be applicable here, but measures to protect an organization's networks and databases from intrusion and external malicious attack would be even more pertinent
- Economy: The reduction of costs, other than those related to staff procurement policies might be put in place which encourage the consolidation of purchasing in order to take advantage of bulkbuying at discount
- Revenue Enhancement / Acceleration: The sooner bills reach customers, the sooner they can pay them
- Strategic Fit: A change might not directly benefit a particular group within the organization but has to be made in order to obtain some strategic advantage for the organization as a whole

A change could have more than one of these types of benefits. In fact, benefits are often interlinked. An example of this is an insurance company which intro-duced a facility whereby when settling claims for damage to property, they directly arranged for contractors to carry out the remedial work. This improved quality of service for customers as it saved them the trouble of locating a reputable con-tractor, reduced costs to the insurance company because they could take advantage of the bulk purchase of services and improved staff morale because of the goodwill generated between the insurance company's front-line staff and the customer.

Quantifying Benefits

Benefits can be:

- Quantified and Valued: A direct financial benefit is experienced
- Quantified but Not Valued: For example, a decrease in the number of customer complaints
- Identified but Not Easily Quantified: For example, public approval of the organization in the locality where it is based

A particular activity might also have disbenefits. For example, increased sales might mean that more money has to be spent on expensive overtime working. There can be controversy over a whether a business change will lead to the particular benefits claimed for it, for example that a new company logo will improve staff morale. Some key tests have been suggested in order to sound out whether a legal / formal benefit is likely to be genuine:

NOTES

- Can you explain in precise terms why this benefit should result from this busi-ness change?
- Can you identify the ways in which we will be able to see the consequences of this benefit?
- If the required outcomes do occur, can they be attributed directly to the change, or could other factors explain them?
- Is there any way in which the benefits can be measured?

Benefit profiles that estimate when and how benefits will be experienced must be maintained. Specific staff has to be allocated responsibility for ensuring that the planned benefits actually materialize. These will often be business change managers. Benefits cannot normally be monitored in a purely project environment because the project will almost certainly have been officially closed before the benefits start to filter through. Benefits management brings the thought that developers and users are jointly responsible for ensuring the delivery of the ben-efits of projects.

3.6 PROJECT EVALUATION

Three major factors will need to be considered in the evaluation of potential projects: technical feasibility, the balance of costs and benefits, and finally the degree of risk associated with the project.

Technical Assessment

Technical assessment of a proposed system consists of evaluating the required functionality against the hardware and software available. Organizational policy, aimed at the provision of a uniform and consistent hardware/software infrastructure, is likely to place limitations on the nature of technical solutions that might be considered. The constraints will influence the cost of the solution and this must be taken into account in the cost-benefit analysis.

Cost-Benefit Analysis

The most common way of carrying out an economic assessment of a proposed information system or software product, is by comparing the expected costs of development and operation of the system with the benefits of having it in place. Assessment focuses on whether the estimated income and other benefits exceed the estimated costs. Additionally, it is usually necessary to ask whether the project under consideration is the best of a number of options. There might be more candidate projects than can be undertaken at anyone time and, in any case, projects will need to be prioritized so that resources are allocated effectively.

The standard way of evaluating the economic benefits of any project is cost-benefit analysis, comprising of two steps. The steps are listed below:

NOTES

Step 1: Identifying and estimating all of the costs and benefits of carrying out the project and operating the delivered application:

These include the development costs, the operating costs and the benefits that are expected to accrue from the new system. Where the proposed system is replacing an existing one, these estimates should reflect the change in costs and benefits due to the new system. A new sales order processing system, for example, could not claim to benefit an organ-ization by the total value of sales only by the increase due to the use of the new system.

Step 2: Expressing these costs and benefits in common units:

The net benefit hat is the difference between the total benefit and the total cost of creating and operating the system need to be evaluated. To do this, we must express each cost and each benefit in some common unit.

Most direct costs are relatively easy to identify and quantify in approximate monetary terms. It is helpful to categorize costs according to where they originate in the life of the project.

- Development costs: Include the salaries and other employment costs of the staff involved in the development project and all associated costs.
- Setup Costs: Include the costs of putting the system into place. These consist mainly of the costs of any new hardware and ancillary equipment but will also include costs of file conversion, recruitment and staff training.
- Operational Costs: Consist of the costs of operating the system once it has been installed.

Benefits, on the other hand, are often quite difficult to quantify in monetary terms even once they have been identified. In cost benefit analysis the focus has to be on the benefits that can be financially valued in some way.

3.7 CASH FLOW FORECASTING

As important as estimating the overall costs and benefits of a project is the forecasting of the cash flows that will take place and their timing. A cash flow forecast will indicate when expenditure and income will take place.

Money has to be spend for expenses such as staff wages, during the development stages of a project. Such expenditure cannot be deferred until income is received either from using the software if it is being developed for inhouse use or from selling it. It is important that we know that we can fund the development expenditure either from the company's own resources or by borrowing from the bank. Thus some forecast is needed of when expenditure such as the payment of salaries and bank interest will take place and when any income is to be expected.

NOTES

Accurate cash flow forecasting is not easy, as it is done early in the project's life cycle at least before any significant expenditure is committed and many items to be estimated particularly the benefits of using software or decommissioning-costs might be some years in the future.

When estimating future cash flows, it is usual to ignore the effects of inflation. Forecasts of inflation rates tend to be uncertain. Moreover, if expenditure is increased due to inflation it is likely that income will increase proportionately. However, measures to deal with increase in costs where work is being done for an external customer must be in place - for example index linked prices where work involves use of raw materials.

The cash flow forecast for four projects is illustrated in Table 3.2. Negative values represent expenditure and positive values represent income. In each case it is assumed that the cash flows take place at the end of each year. For short-term projects or where there are significant seasonal cash flow patterns, quarterly, or even monthly, cash flow forecasts could be appropriate.

Typically products generate a negative cash flow during their development followed by a positive cash flow over their operating life. There might be decommissioning costs at the end of a product's life. The difficulty and importance of cash flow forecasting is evidenced by the number of companies that suffer bankruptcy because, although they are developing profitable products or services, they cannot sustain an unplanned negative cash flow.

Table 3.2: Four Project Cash Flow Projections – Figures and End of Year Totals (Rs.)

Year	Project 1	Project 2	Project 3	Project 4
0	-100,000	-1,000,000	-100,000	-120,000
1	10,000	200,000	30,000	30,000
2	10,000	200,000	30,000	30,000
3	10,000	200,000	30,000	30,000
4	20,000	200,000	30,000	30,000
5	100,000	300,000	30,000	75,000
Net profit	50,000	100,000	50,000	75,000

Cash flows take place at the end of each year. The year 0 figure represents the initial investment made at the start of the project.

3.8 COST-BENEFIT EVALUATION TECHNIQUES

Some common methods for comparing projects on the basis of their cash flow forecasts are discussed below.

NOTES

Net Profit

The net profit of a project is the difference between the total costs and the total income over the life of the project. Project 2 in Table 3.2 shows the greatest net profit but this is at the expense of a large investment.

Payback Period:

The payback period is the time taken to break even or pay back the initial invest-ment. Normally, the project with the shortest payback period will be chosen on the basis that an organization will wish to minimize the time that a project is 'in debt'.

The advantage of the payback period is that it is simple to calculate and is not particularly sensitive to small forecasting errors. Its disadvantage as a selection technique is that it ignores the overall profitability of the project infact, it totally ignores any income (or expenditure) once the project has broken even.

Return on Investment:

The return on investment (ROI), also known as the accounting rate of return (ARR), provides a way of comparing the net profitability to the investment required. There are some variations on the mathematical model used to calculate the return on invest-ment but a straightforward common version is:

ROI = (Average Annual Profit/ Total Investment) x 100

Example:

Calculating the ROI for project 1 (Table 3.2), the net profit is Rs. 50,000 and the total investment is Rs. 100,000.

The return on investment is therefore calculated as follows:

ROI = ((50,000 /5) /100,000) x 100 = 10%

The return on investment provides a simple, easy to calculate measure of return on capital but it suffers from two major disadvantages. Like the net profitability, it takes no account of the timing of the cash flows. More importantly, this rate of return bears no relationship to the interest rates offered or charged by banks (or any other normal interest rate) since it takes no account of the timing of the cash flows or of the compounding of interest.

Net Present Value

The computation of net present value (NPV) is a project evaluation technique that takes into account the profitability of a project and the timing of the cash flows that are pro-duced. This is based on the view that receiving Rs.100 today is better than having to

NOTES

wait until next year to receive it, because the Rs.100 next year is worth less than Rs.100 now. We could, for example, invest the Rs.100 in a bank today and have Rs.100 plus the interest in a year's time. If we say that the present value of Rs.100 in a year's time is Rs.91, we mean that Rs.100 in a year's time is the equivalent of Rs.91 now.

The equivalence of Rs.91 now and Rs.100 in a year's time means we are dis-counting the future income by approximately 10% - that is, we would need an extra 10% to make it worthwhile waiting for a year. An alternative way of con-sidering the equivalence of the two is to consider that, if we received Rs.91 now and invested for a year at an annual interest rate of 10%, it would be worth Rs.100 in a year's time. The annual rate by which we discount future earnings is known as the discount rate - 10% in the above example.

Similarly, Rs.100 received in two years' time would have a present value of approximately Rs.83 - in other words, Rs.83 invested at an interest rate of 10% would yield approximately Rs.100 in two years' time. The present value of any future cash flow may be obtained by applying the following mathematical model:

Present Value= Value in Year t / $(1 + r)^t$

In the above mathematical model r is the discount rate, expressed as a decimal value, and t is the number of years into the future that the cash flow occurs.

Alternatively, the present value of a cash flow may be calculated by multiplying the cash flow by the appropriate discount factor. More extensive or detailed tables may be constructed using the mathematical model:

Discount Factor = $1 / (1 + r)^t$

for various values of r the discount rate and t the number of years from now.

The net present value for a project is obtained by discounting each cash flow (both negative and positive) and summing the discounted values. It is normally assumed that any initial investment takes place immediately (indicated as year 0) and is not dis-counted. Later cash flows are normally assumed to take place at the end of each year and are discounted by the appropriate amount.

The major difficulty with net present value for deciding between projects is selecting an appropriate discount rate. Some organizations have a standard rate but, where this is not the case, then the discount rate should be chosen to reflect available interest rates (borrowing costs where the project must be funded from loans) plus some premium to reflect the fact that software projects are inherently more risky than lending money to a bank. The exact discount rate is normally less important than ensuring that the same discount rate is used for all projects being compared. However, it is important to check that the ranking of projects is not sensitive to small changes in the discount rate.

NOTES

Internal Rate of Return

The major disadvantage of net present value as a measure of profitability is that, although it may be used to compare projects, it might not be directly comparable with earnings from other investments or the costs of borrowing capital. Such costs are usually quoted as a percentage interest rate. The internal rate of return (IRR) attempts to provide a profitability measure as a percentage return that is directly comparable with interest rates. Thus, a project that showed an estimated internal rate of return of 10% would be worthwhile if the capital could be borrowed for less than 10% or if the capital could not be invested else where for a return greater than 10%.

If internal rate return is calculated as that percentage discount rate that would produce a net present value of zero it is most easily calculated using a spreadsheet or other computer program that provides functions for calculating the internal rate of return.

NPV and IRR are not a complete answer to economic project evaluation. The following points may be taken into consideration:

- A total evaluation must also take into account the problems of funding the cash flows. For example: Can the interest on borrowed money and development staff salaries be paid at the appropriate time?
- While a project’s IRR might indicate a profitable project, future earnings from a project might be far less reliable than earnings from, say, investing with a bank. To take account of the risk inherent in investing in a project, we might require that a project earn a ‘risk premium’ or we might undertake a more detailed risk analysis.
- We must also consider any one project within the financial and economic framework of the organization as a whole and answer the question if we fund this one, will we also be able to fund other worthy projects?

Risk Evaluation

Every project involves risk of some form. When assessing and planning a project, we are concerned with the risk that the project might not meet its objectives. Risk must be taken into account when deciding whether to proceed with a proposed project.

Risk Identification and Ranking

In any project evaluation we should attempt to identify the risk and quantify their potential effects. One common approach to risk analysis is to construct a project risk matrix utilizing a checklist of possible risks and to classify each risk according to its relative importance and likelihood. Note that the importance and likelihood need to be separately assessed - we might be less concerned with something that, although serious, is very unlikely to occur than with something less serious that is almost certain. Table 3.3 illustrates a basic project risk matrix listing some of the risks that might be considered for a project, with their importance and likelihood classified as high (H), medium (M), low (L) or exceedingly

NOTES

unlikely (-). So that projects may be compared the list of risks must be the same for each project being assessed. It is likely, in reality, that it would be somewhat longer than shown and more precisely defined.

Table 3.3: A Fragment of a Basic Project Risk Matrix

Risk	Importance	Likelihood
Software never completed or delivered	H	-
Project cancelled after design stage	H	-
Software delivered late	M	M
Development budget exceeded < 20%	L	M
Development budget exceeded > 20 %	M	L
Maintenance costs higher than estimated	L	L
Response time targets not met	L	H

The project risk matrix may be used as a way of evaluating projects those with high risks being less favored or as a means of identifying and ranking the risks for a specific project.

Risk and Net Present Value

Where a project is relatively risky it is common practice to use a higher discount rate to calculate net present value. This risk premium, might, for example, be an additional 2% for a reasonably safe project or 5% for a fairly risky one. Projects may be categorized as high, medium or low risk using a scoring method and risk premiums designated for each category. The premiums, even if subjective, provide a consistent method of taking risk into account.

Cost-Benefit Analysis

An approach to the evaluation of risk is to consider each possible outcome and estimate the probability of its occurring and the correspon-ding value of the outcome. Rather than a single cash flow forecast for a project, we will then have a set of cash flow forecasts, each with an associated probability of occurring. The value of the project is then obtained by summing the cost or benefit for each possible outcome weighted by its corresponding probability.

This approach is normally used in the evaluation of large projects such as the building of new motorways, where variables such as future traffic volumes, and hence the total benefit of shorter journey times, are subject to uncertainty. The method does, of course, rely on our being able to assign probabilities of occur-rence to each scenario and, without extensive study, this can be difficult.

NOTES

When used to evaluate a single major project, the cost benefit approach, by averaging out the negative and positive outcomes of the different scenarios, does not take account of the full force of damaging outcomes. Because of this, where overall profitability is the primary concern, it is often more appropriate for the evaluation of a portfolio of projects where more successful projects can offset the impact of less successful ones.

Risk Profile Analysis

An approach which attempts to overcome some of the objections to cost-benefit averaging is the construction of risk profiles using sensitivity analysis. This involves varying each of the parameters that affect the project's cost or benefits to ascertain how sensitive the project's profitability is to each factor. We might, for example, vary one of our original estimates by plus or minus 5% and recalculate the expected costs and benefits for the project. By repeating this exer-cise for each of our estimates in turn we can evaluate the sensitivity of the project to each factor.

By studying the results of a sensitivity analysis we can identify those factors that are most important to the success of the project. We then need to decide whether we can exercise greater control over them or otherwise mitigate their effects. If neither is the case, then we must live with the risk or abandon the project. Sensitivity analysis demands that we vary each factor one at a time. It does not easily allow us to consider the effects of combinations of circumstances, neither does it evaluate the chances of a particular outcome occurring. In order to do this we need to use a more sophisticated tool such as Monte Carlo simulation. There are a number of risk analysis applications available that use Monte Carlo simulation and produce risk profiles.

Using Decision Trees

An invoicing system in a company is operational for a decade and has to be extended. Prior to contract the job of extending their invoicing system, the company must consider the alternative of completely replacing the existing system, which they will have to do at some point in the future. The decision largely rests upon the rate at which their equipment maintenance business expands, if their market share significantly increases the existing system might need to be replaced within a short period. Not replacing the system in time could be an expensive option as it could lead to lost revenue if they cannot cope with the increase in invoicing demand. Replacing it immediately will, however, be expensive as it will mean deferring other projects that have already been scheduled.

They have calculated that extending the existing system will have an NPV of Rs. 75,000, although if the market expands significantly, this will be turned into a loss with an NPV of –Rs. 100,000 due to lost revenue. If the market does expand, replacing the system now has an NPV of Rs. 250,000 due to the benefits of being able to handle increased sales and other benefits such as improved management information. If sales do not increase, however, the benefits will be severely reduced and the project will suffer a

NOTES

loss with an NPV of Rs.50,000. The company estimate the likelihood of the market increasing significantly at 20% and, hence, the probability that it will not increase as 80%. This scenario can be represented as a tree structure as shown in Figure 3.4.

The analysis of a decision tree consists of evaluating the expected benefit of taking each path from a decision point (denoted by D in Figure 3.4). The expected value of each path is the sum of the value of each possible outcome multiplied by its probability of occurrence. The expected value of extending the system is therefore Rs. 40,000 (75,000 X 0.8 - 100,000 X 0.2) and the expected value of replacing the system Rs.10,000 (250,000 X 0.2 - 50,000 X 0.8). The company should therefore choose the option of extending the existing system.

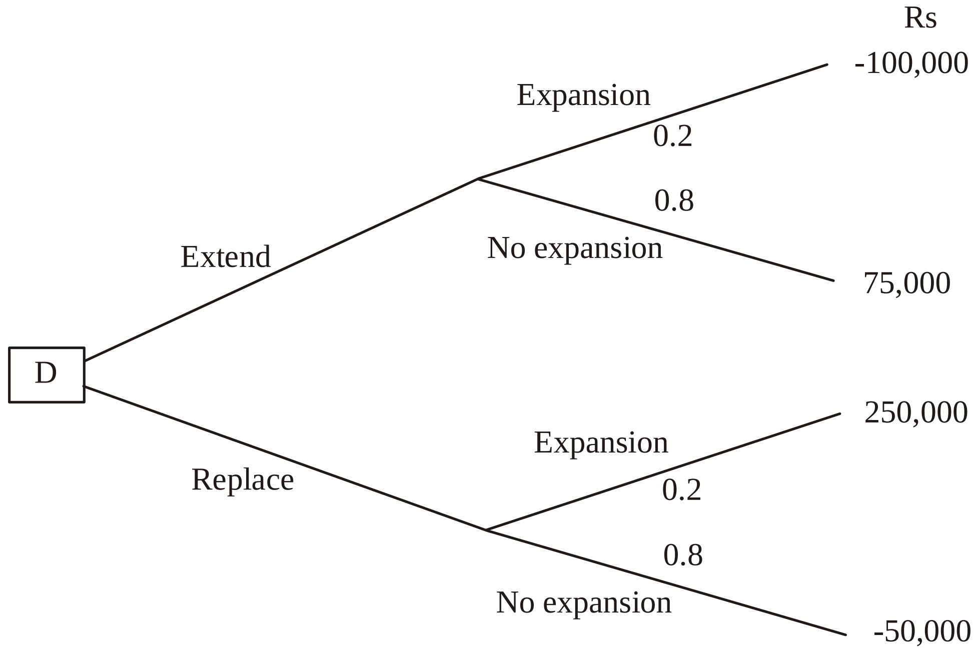

Figure 3.4: Decision Tree

This example illustrates the use of a decision tree to evaluate a simple decision at the start of a project. One of the great advantages of using decision trees to model and analyze problems is the ease with which they can be extended.

Summary:

Projects must be evaluated on strategic, technical and economic grounds Projects are not justifiable on their own, but are as part of a broader programme of projects that put into practice an organization’s strategy. Not all benefits can be accurately quantified in financial values. Economic assessment involves the identification of all costs and income over the lifetime of the system, including its development and operation and checking that the total value of benefits exceeds total expenditure. Money received in the future is worth less than the same amount of money in hand now, which may be invested to earn interest.

NOTES

The uncertainty surrounding estimates of future returns lowers their real value measured now. Discounted cash flow techniques may be used to evaluate the present value of future cash flows taking account of interest rates and uncertainty. Cost-benefit analysis techniques and decision trees provide tools for evaluating anticipated outcomes and choosing between alternative strategies.

Questions:

1. List and discuss the different forms of programmes.
2. Discuss the steps involved in creating a programme.
3. What is a dependency diagram used for? Discuss with an example.
4. What is delivery planning? Discuss.
5. Present a tutorial on benefits management.
6. Discuss the techniques for evaluation of individual projects.
7. What is risk profile analysis? Discuss.

NOTES

NOTES

UNIT III

CHAPTER - 4

ACTIVITY PLANNING

Techniques for forecasting the effort required for a project both for the project as a whole and for individual activities are needed. A detailed plan for the project, however, must also include a schedule indicating the start and completion times for each activity. This will enable us to:

- Ensure that the appropriate resources will be available precisely when required
- Avoid different activities competing for the same resources at the same time
- Produce a detailed schedule showing which staffs carry out each activity
- Produce a detailed plan against which actual achievement may be measured
- Produce a timed cash flow forecast
- Replan the project during its life to correct drift from the target

To be effective, a plan must be stated as a set of targets, the achievement or non-achievement of which can be unambiguously measured. The activity plan does this by providing a target start and completion date for each activity. The starts and completions of activities must be clearly visible and this is one of the reasons why it is advisable to ensure that each and every project activity produces some tangible product or 'deliverable'. Monitoring the project's progress is then, a case of ensuring that the products of each activity are delivered on time.

As a project progresses it is unlikely that everything will go according to plan. Much of the job of project management concerns recognizing when something has gone wrong, identifying its causes and revising the plan to mitigate its effects. The activity plan should provide a means of evaluating the consequences of not meeting any of the activity target dates and guidance as to how the plan might most effectively be modified to bring the project back to target.

NOTES

4.1 OBJECTIVES OF ACTIVITY PLANNING

In addition to providing project and resource schedules, activity planning aims to achieve a number of other objectives. The objectives are summarized as follows:

- Feasibility Assessment: Is the project possible within required timescales and resource constraints? It is not until we have constructed a detailed plan that we can forecast a completion date the any reasonable knowledge of its achievability.
- Resource Allocation: What are the most effective ways of allocating resources to the project? When should the resources be available? The project plan allows us to investigate the relationship between timescales and resource availability (in general, allocating additional resources to a project shortens its duration) and the efficacy of additional spending on resource procurement.
- Detailed Costing: How much will the project cost and when is that expenditure likely to take place? After producing an activity plan and allocating specific resources, we can obtain more detailed estimates of costs and their timing.
- Motivation: Providing targets and being seen to monitor achievement against targets is an effective way of motivating staff, particularly where they have been involved in setting those targets in the first place.
- Co-ordination: When do the staffs in different departments need to be available to work on a particular project and when does staff need to be transferred between projects? The project plan, particularly with large projects involving more than a single project team, provides an effective vehicle for communication and coordination among teams. In situations where staff may need to be transferred between project teams (or work concurrently on more than one project), a set of integrated project schedules should ensure that such staff are available when required and do not suffer periods of enforced idleness.

Activity planning and scheduling techniques place an emphasis on completing the project in a minimum time at an acceptable cost or, alternatively, meeting a set target date at minimum cost. These are not, in themselves, concerned with meeting quality targets, which generally impose constraints on the scheduling process.

One effective way of shortening project durations is to carry out activities in parallel. We cannot undertake all the activities at the same time, some require the completion of others before they can start and there are likely to be resource constraints limiting how much may be done simultaneously. Activity scheduling will, however, give us an indication of the cost of these constraints in terms of lengthening timescales and provide us with an indication of how timescales may be shortened by relaxing those constraints. It is up to us, if we try relaxing precedence constraints by, for example, allowing a program coding task to commence before the design has been completed, to ensure that we are clear about the potential effects on product quality.

NOTES

4.2 PROJECT SCHEDULES

Before work commences on a project or, possibly, a stage of a larger project, the project plan must be developed to the level of showing dates when each activity should start and finish and when and how much of each resource will be required. Once the plan has been refined to this level of detail we call it a project schedule. Creating a project schedule comprises four main stages.

- The first step in producing the plan is to decide what activities need to be carried out and in what order they are to be done. From this we can construct an ideal activity plan that is, a plan of when each activity would ideally be undertaken were resources not, a constraint. It is the creation of the ideal activity plan.
- The ideal activity plan will then be the subject of an activity risk analysis, aimed at identifying potential problems. This might suggest alterations to the ideal activity plan and will almost certainly have implications for resource allocation.
- The third step is resource allocation. The expected availability of resources might place constraints on when certain activities can be carried out, and our ideal plan might need to be adapted to take account of this.
- The final step is schedule production. Once resources have been allocated to each activity, we will be in a position to draw up and publish a project schedule, which indicates planned start and completion dates and a resource requirements statement for each activity.

4.3 PROJECTS AND ACTIVITIES

Defining Activities

Before we try to identify the activities that make up a project it is worth reviewing what we mean by a project and its activities and adding some assumptions that will be relevant when we start to produce an activity plan.

- A project is composed of a number of interrelated activities
- A project may start when at least one of its activities is ready to start
- A project will be completed when all of the activities it encompasses have been completed
- If an activity must have a clearly defined start and a clearly defined end-point, normally marked by the production of a tangible deliverable
- An activity requires a resource then that resource requirement must be forecastable and is assumed to be required at a constant level throughout the duration of the activity
- The duration of an activity must be forecastable — assuming normal circumstances, and the reasonable availability of resources
- Some activities might require that others are completed before they can begin (these are known as precedence requirements)

NOTES

Identifying Activities

Essentially there are three approaches to identifying the activities or tasks that make up a project. They are

- The activity-based approach
- The product-based approach and
- The hybrid approach

The Activity-based Approach

The activity-based approach consists of creating a list of all the activities that the project is thought to involve. This might involve a brainstorming session involving the whole project team or it might stem from an analysis of similar past projects. When listing activities, particularly for a large project, it might be helpful to subdivide the project into the main life-style stages and consider each of these separately.

Rather than doing this in an ad hoc manner, a much favoured way of generating a task list is to create a Work Breakdown Structure (WBS). This involves identifying the main (or high- level) tasks required to complete a project and then breaking each of these down into a set of lower-level tasks. Figure 4.1 illustrates a fragment of a WBS where the design task has been broken down into three tasks and one of these has been further decomposed into two tasks.

Activities are added to a branch in the structure if they directly contribute to the task immediately above if they do not contribute to the parent task, then they should not be added to that branch. The tasks at each level in any branch should include everything that is required to complete the task at the higher level if they are not a comprehensive definition of the parent task, and then something is missing.

When preparing a WBS, consideration must be given to the final level of detail or depth of the structure. Too great a depth will result in a large number of small tasks that will be difficult to manage, whereas a too shallow structure will provide insufficient detail for project control. Each branch should, however, be broken down at least to a level where each leaf may be assigned to an individual or responsible section within the organization.

NOTES

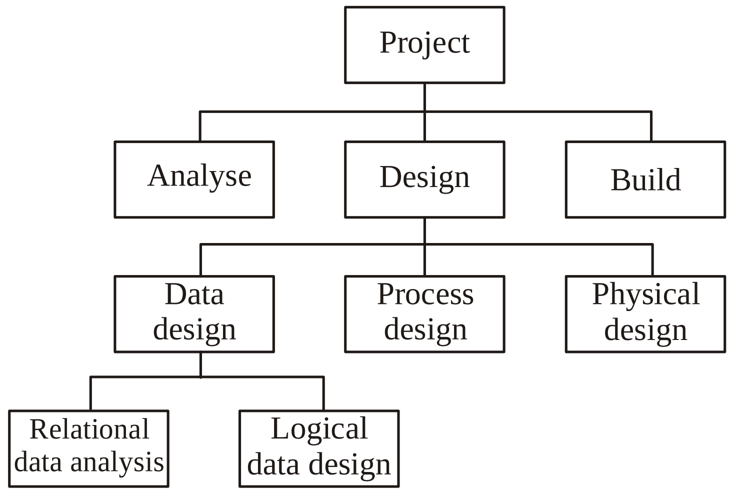

Figure 4.1: A Fragment of an Activity-based Work Breakdown Structure

The advantages of the WBS approach include the belief that it is much more likely to result in a task catalogue that is complete and is composed of non- overlapping activities. Note that it is only the leaves of the structure that comprise the list of activities in the project — higher-level nodes merely represent collections of activities.

The WBS also represents a structure that may be refined as the project proceeds. In the early part of a project we might use a relatively high-level or shallow WBS, which can be developed as information becomes available, typically during the project's analysis and specification phases.

Once the project's activities have been identified (whether or not by using a WBS), they need to be sequenced in the sense of deciding which activities need to be completed before others can start.

The Product-based Approach

The product-based approach consists of producing a Product Breakdown Structure and a Product Flow Diagram. The PFD indicates, for each product, which other products are required as inputs. The PFD can therefore be easily transformed into an ordered list of activities by identifying the transformations that turn some products into others. Proponents of this approach claim that it is less likely that a product will be left out of a PBS than that an activity might be omitted from an unstructured activity list.

This approach is particularly appropriate if using a methodology such as Structured Systems Analysis and Design Methodology (SSADM), which clearly specifies, for each step or task, each of the products required and the activities required to produce it. The SSADM Reference Manual provides a set of generic PBSs for each stage in SSADM, which can be used as a basis for generating a project- specific PBS. The same is illustrated in Figure 4.2

NOTES

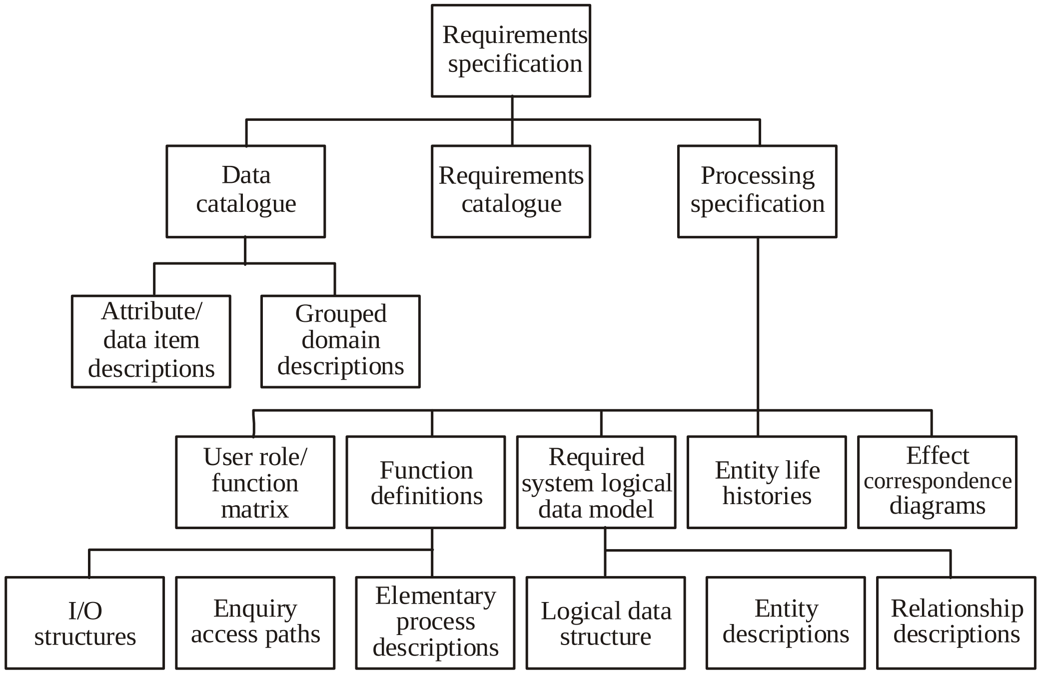

Figure 4.2: SSADM Product Breakdown Structure for Requirements Specification

The SSADM Reference Manual also supplies generic activity networks and, using the project-specific PBS and derived PFD, these may be used as a basis for developing a project-specific activity network. Figure 4.3 illustrates an activity network for the activities required to create the products in Figure 4.2.

Notice how the development of a PFD leads directly to an activity network that indicates the sequencing of activities in Figure 4.3, activity 340 (Enhance required data model) requires products from activity 330 and activity 360 needs products from both activities 330 and 340.

The Hybrid Approach

The WBS illustrated in Figure 4.1 is based entirely on a structuring of activities. Alternatively, and perhaps more commonly, a WBS may be based upon the project's products as illustrated in Figure 4.4, which is in turn based on a simple list of final deliverables and, for each deliverable, a set of activities required to produce that product.

The degree to which the structuring is product-based or activity-based might be influenced by the nature of the project and the particular development method adopted. As with a purely activity-based WBS, having identified the activities we are then left with the task of sequencing them.

NOTES

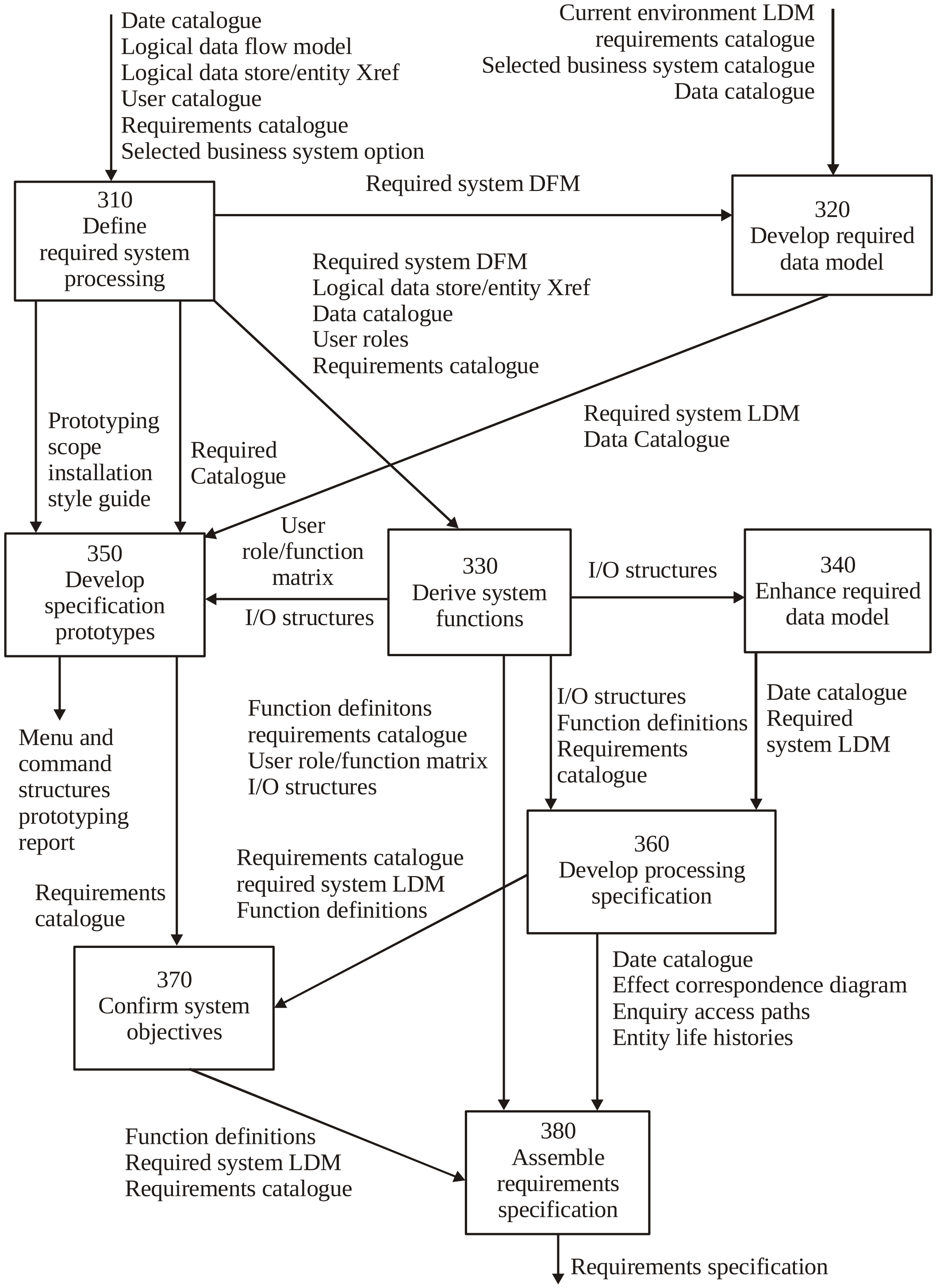

Figure 4.3: A Structuring of Activities for the SSADM Requirements Specification Stage

A framework dictating the number of levels and the nature of each level in the structure may be imposed on a WBS. The following five levels should be used in a WBS:

- Level 1: Project
- Level 2: Deliverables such as software, manuals and training courses

NOTES

- Level 3: Components which are the key work items needed to produce deliverables, such as the modules and tests required to produce the system software
- Level 4: Work-packages which are major work items, or collections of related tasks, required to produce a component
- Level 5: Tasks which are tasks that will normally be the responsibility of a single person

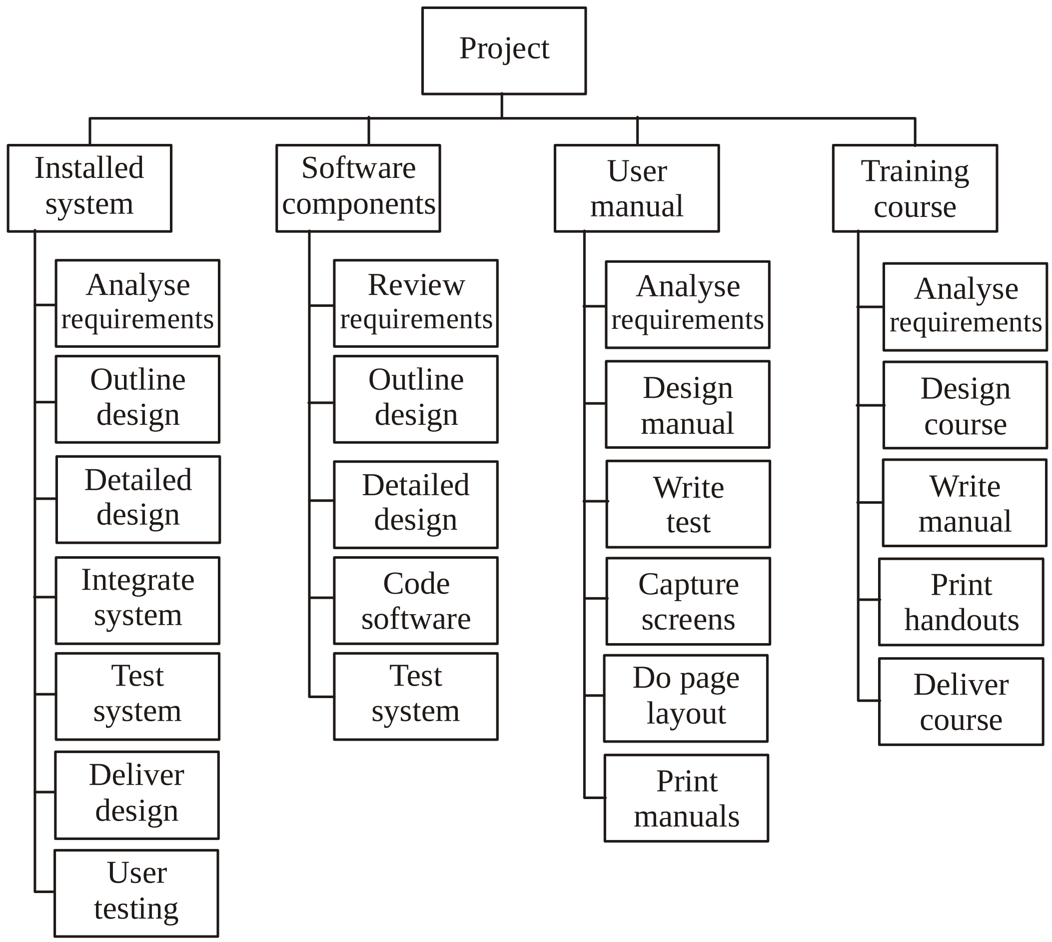

Figure 4.4: A Work Breakdown Structure based on Deliverables

4.4 SEQUENCING AND SCHEDULING ACTIVITIES

Throughout a project, we will require a schedule that clearly indicates when each of the project's activities is planned to occur and what resources it will need. One way of presenting such a plan is to use a bar chart as shown in Figure 4.5.

The chart shown has been drawn up taking account of the nature of the development process (that is, certain tasks must be completed before others may start) and the resources that are available (for example, activity C follows activity B because Ani cannot work on both tasks at the same time). In drawing up the chart, we have therefore done two things, we have sequenced the tasks (that is, identified the dependencies among activities dictated by the development process) and scheduled them (that is, specified when they should take place). The scheduling has had to take account of the availability of staff and the ways in which the activities have been allocated to them. The schedule might look quite different were there a different number of staff or were we to allocate the activities differently.

In the case of small projects, this combined sequencing, scheduling approach might be quite suitable, particularly where we wish to allocate individuals to particular tasks at an early planning stage. However, on larger projects it is better to separate out these two activities: to sequence the tasks according to their logical relationships and then to schedule them taking into account resources and other factors.

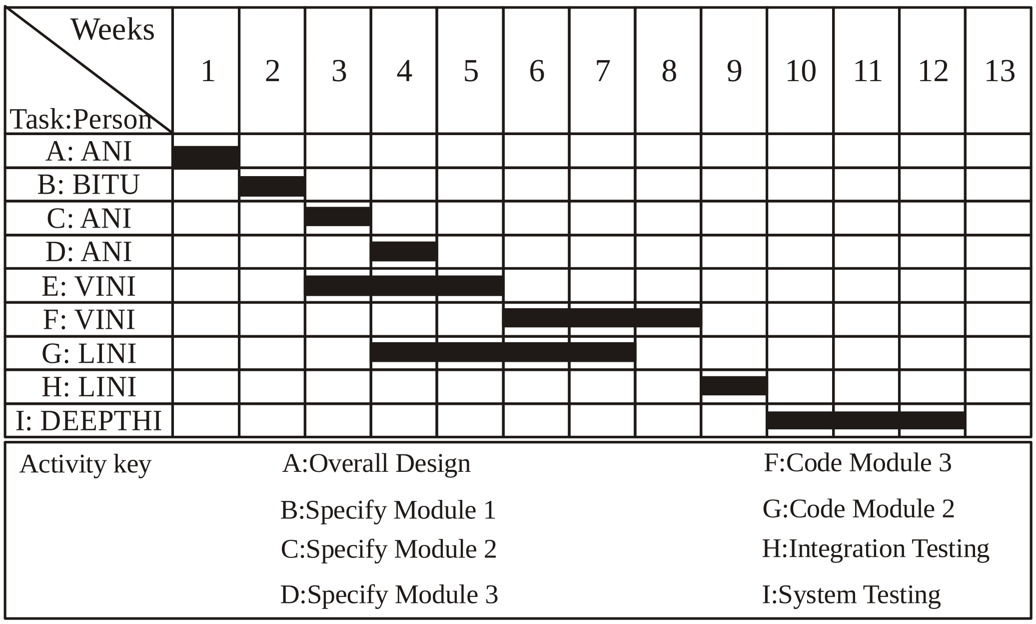

Figure 4.5: A Project Plan as a Bar Chart

Approaches to scheduling that achieve this separation between the logical and the physical use networks to model the project.

4.5 NETWORK PLANNING MODELS

These project scheduling techniques model the project's activities and their relationships as a network. In the network, time flows from left to right. The best known techniques are CPM (Critical Path Method) and PERT (Program Evaluation Review Technique).

Both of these techniques used an activity-on-arrow approach to visualizing the project as a network where activities are drawn as arrows joining circles, or nodes, which represent the possible start and/or completion of an activity or set of activities. More recently a variation on these techniques, called precedence networks, has become popular. This method uses activity-on-node networks where activities are represented as nodes and the links between nodes represent precedence (or sequencing) requirements. This latter approach avoids some of the problems inherent in the activity-on-arrow representation

NOTES

and provides more scope for easily representing certain situations. It is this method that is adopted in the majority of computer applications currently available. These three methods are very similar and it must be admitted that many people use the same name (particularly CPM) indiscriminately to refer to any or all of the methods.

4.6 FORMULATING A NETWORK MODEL

The first stage in creating a network model is to represent the activities and their interrelationships as a graph. In activity-on-node we do this by representing activities as links (arrowed lines) in the graph — the nodes (circles) representing the events of activities starting and finishing.

Constructing Precedence Networks

Rules for constructing network models:

- A project network should have only one start node: Although it is logically possible to draw a network with more than one starting node, it is undesirable to do so as it is a potential source of confusion. In such cases (for example, where more than one activity can start immediately the project starts) it is normal to invent a 'start' activity which has zero duration but may have an actual start date.
- A project network should have only one end node: The end node designates the completion of the project and a project may only finish once. Although it is possible to draw a network with more than one end node it will almost certainly lead to confusion if this is done. Where the completion of a project depends upon more than one final' activity it is normal to invent a 'finish' activity.
- A node has duration: A node represents an activity and, in general, activities take time to execute. Notice, however, that the network need not contain any reference to durations. This network drawing merely represents the logic of the project, the rules governing the order in which activities are to be carried out.
- Links normally have no duration: Links represent the relationships between activities. For example program testing cannot start until both coding and data take-on has been completed.
- Precedents are the immediate preceding activities: The activity 'Program test' cannot start until both 'Code' and 'Data take-on' have been completed and activity Install' cannot start until 'Program test' has finished. 'Code' and 'Data take-on' can therefore be said to be precedents of 'Program test', and 'Program test' is a precedent of 'Install'. Note that we do not speak of 'Code' and 'Data take-on' as precedents of 'Install', that relationship is implicit in the previous statement. The same is illustrated in Figure 4.6.
- Time moves from left to right: If at all possible, networks are drawn so that time moves from left to right. It is rare that this convention needs to be flouted but some people add arrows to the lines to give a stronger visual indication of the time flow of the project.

NOTES

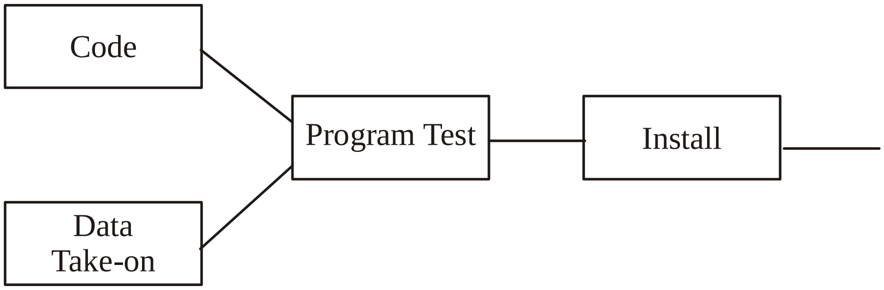

Figure 4.6: Fragment of a Precedence Network

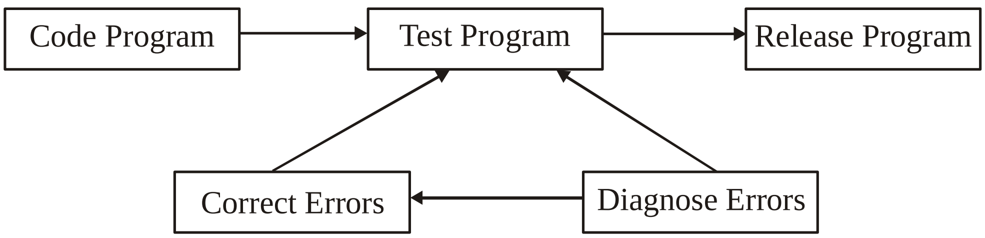

Figure 4.7: A Loop Represents an Impossible Sequence

- A network may not contain loops Figure 4.7 demonstrates a loop in a network. A loop is an error in that it represents a situation that cannot occur in practice. While loops, in the sense of iteration, may occur in practice, they cannot be directly represented in a project network. Note that the logic of Figure 4.7 suggests that program testing cannot start until the errors have been corrected.
- If we know the number of times we expect to repeat a set of activities, a test—diagnose—correct sequence, for example, then we can draw that set of activities as a straight sequence, repeating it the appropriate number of times. If we do not know how many times a sequence is going to be repeated then we cannot calculate the duration of the project unless we adopt an alternative strategy such as redefining the complete sequence as a single activity and estimating how long it will take to complete it.
- Although it is easy to see the loop in this simple network fragment, very large networks can easily contain complex loops which are difficult to spot when they are initially constructed. Fortunately, all network planning applications will detect loops and generate error messages when they are found.
- A network should not contain dangles A dangling activity such as Write user manual' in Figure 4.8 should not exist as it is likely to lead to errors in subsequent analysis. Indeed, in many cases dangling activities indicate errors in logic when activities are added as an afterthought. If, in Figure 4.8, we mean to indicate that the project is complete once the software has been installed and the user manual written then we should redraw the network with a final completion activity which, at least in this case, is probably a more accurate representation of what should happen. The redrawn network is shown in Figure 4.9.

NOTES

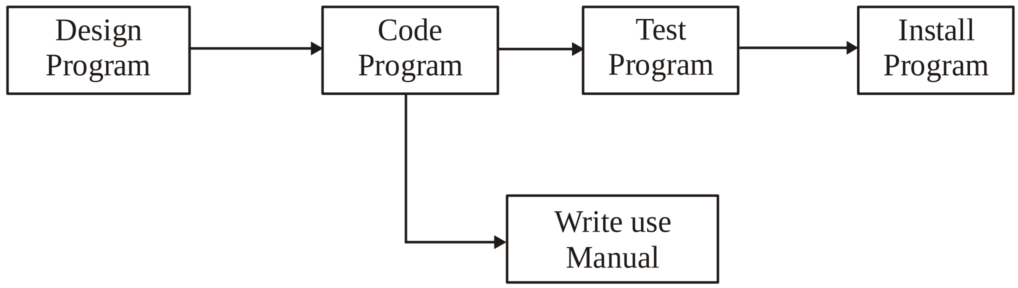

Figure 4.8: A Dangle

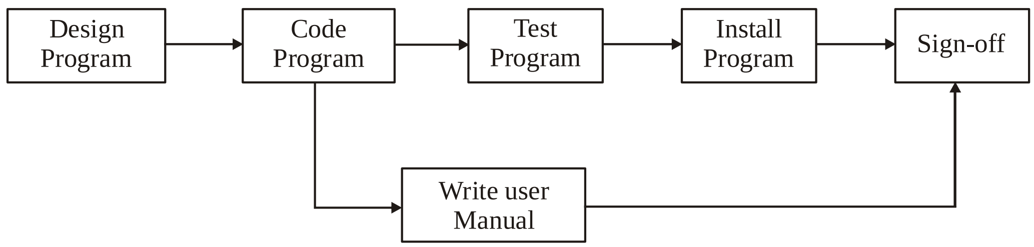

Figure 4.9: Resolving the Dangle

Representing Lagged Activities

There may be situations where we wish to undertake two activities in parallel as there is a lag between the two. We might wish to document amendments to a program as it was being tested. In such a case we could designate an activity test and document amendments'. This would, however, make it impossible to show that amendment recording could start, say, one day after testing had begun and finish a little after the completion of testing.

Where activities can occur in parallel with a time lag between them we represent the lag with duration on the linking arrow as shown in Figure 4.10. This indicates that documenting amendments can start one day after the start of prototype testing and will be completed two days after prototype testing is completed.

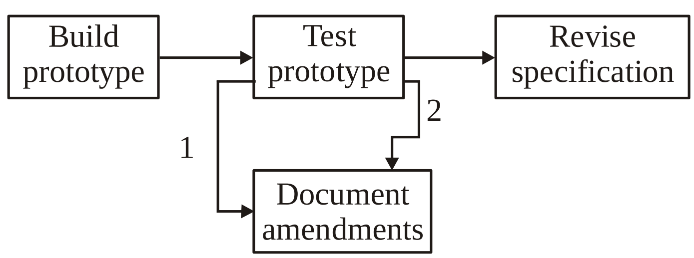

Figure 4.10: Indicating Lags

Hammock Activities

Hammock activities are activities which, in themselves, have zero duration but are assumed to start at the same time as the first 'hammocked' activity and to end at the same time as the last one. They are normally used for representing overhead costs or other

NOTES

resources that will be incurred or used at a constant rate over the duration of a set of activities.

Labeling Conventions:

There are a number of differing conventions that have been adopted for entering information on an activity-on-node network. One of the more common conventions for labeling nodes is illustrated in Figure 4.11.

<table>
<tr><td colspan="2">Activity Label</td><td colspan="2">Duration</td></tr>
<tr><td>Earliest Start</td><td colspan="2" rowspan="2">Activity Description</td><td>Earliest Finish</td></tr>
<tr><td>Latest Start</td><td>Latest Finish</td></tr>
<tr><td colspan="2">Activity Span</td><td colspan="2">Float</td></tr>
</table>

Figure 4.11: Commonly Used Labeling Convention

4.7 ADDING THE TIME DIMENSION

Having created the logical network model indicating what needs to be done and the interrelationships between those activities, we are now ready to start thinking about when each activity should be undertaken.

The critical path approach is concerned with two primary objectives: planning the project in such a way that it is completed as quickly as possible; and identifying those activities where a delay in their execution is likely to affect the overall end date of the project or later activities' start dates.

The method requires that for each activity we have an estimate of its duration. The network is then analyzed by carrying out a forward pass, to calculate the earliest dates at which activities may commence and the project are completed, and a backward pass, to calculate the latest start dates for activities and the critical path.

In practice we would use a software application to carry out these calculations for anything but the smallest of projects. It is important, though, that we understand how the calculations are carried out in order to interpret the results correctly and understand the limitations of the method.

Table 4.1 illustrates a project composed of eight activities whose durations have been estimated as shown in the table.

NOTES

Table 4.1: An Example Project Specification with Estimated Activity Durations and Precedence Requirements

Activity		Duration (weeks)	Precedents
A	Hardware Selection	6	
B	Software Design	4	
C	Install Hardware	3	A
D	Code & Test Software	4	B
E	File Take-on	3	B
F	Write User Manuals	10	
G	User Training	3	E,F
H	Install & Test System	2	C,D

The precedence network for the example project is illustrated in Table 4.12.a

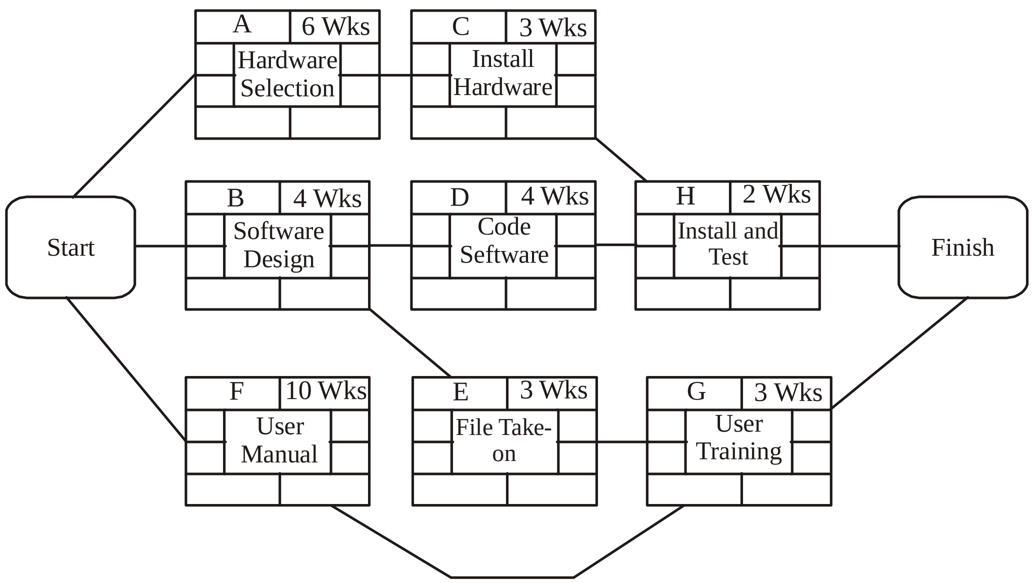

Figure 4.12.a: The Precedence Network for the Example Project

4.8 THE FORWARD PASS

The forward pass is carried out to calculate the earliest dates on which each activity may be started and completed.

Where an actual start date is known, the calculations may be carried out using actual dates. Alternatively we can use day or week numbers and that is the approach we shall adopt here. By convention, dates indicate the end of a period and the project is therefore shown as starting at the end of week zero (or the beginning of week 1).

NOTES

The forward pass and the calculation of earliest start dates is calculated according to the following reasoning:

- Activities A, B and F may start immediately, so the earliest date for their start is zero.
- Activity A will take 6 weeks, so the earliest it can finish is week 6.
- Activity B will take 4 weeks, so the earliest it can finish is week 4.
- Activity F will take 10 weeks, so the earliest it can finish is week 10.
- Activity C can start as soon as A has finished so its earliest start date is week 6. It will take 3 weeks so the earliest it can finish is week 9.
- Activities D and E can start as soon as B is complete so the earliest they can each start is week 4. Activity D, which will take 4 weeks, can therefore finish by week 8 and activity E, which will take 3 weeks, can therefore finish by week 7.
- Activity G cannot start until both E and F have been completed. It cannot therefore start until week 10 — the later of weeks 7 (for activity E) and 10 (for activity F). It takes 3 weeks and finishes in week 13.
- Similarly, Activity H cannot start until week 9 — the later of the two earliest finished dates for the preceding activities C and D.
- The project will be complete when both activities H and G have been completed. Thus the earliest project completion date will be the later of weeks 11 and 13 — that is, week 13.

Figure 4.12.b illustrates the network after the forward pass.

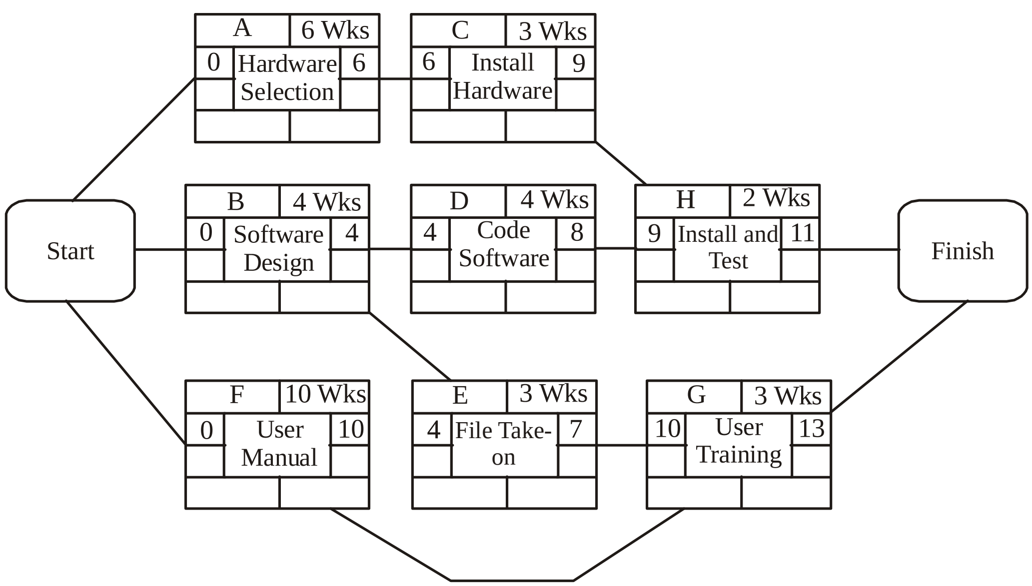

Figure 4.12.b: The Network after the Forward Pass.

NOTES

4.9 THE BACKWARD PASS

The second stage in the analysis of a critical path network is to carry out a backward pass to calculate the latest date at which each activity may be started and finished without delaying the end date of the project. In calculating the latest dates, we assume that the latest finish date for the project is the same as the earliest finish date that is, we wish to complete the project as early as possible.

Figure 4.13 illustrates our network after carrying out the backward pass. The latest activity dates are calculated as follows.

- The latest completion date for activities G and H is assumed to be week 13.
- Activity H must therefore start at week 11 at the latest (13—2) and the latest start date for activity G is week 10 (13—3).
- The latest completion date for activities C and D is the latest date at which activity H must start - that is, week 11. They therefore have latest start dates of week 8 (11—3) and week 7 (11—4) respectively.
- Activities E and F must be completed by week 10 so their earliest start dates are weeks 7 (10—3) and 0 (10—10) respectively.
- Activity B must be completed by week 7 (the latest start date for both activities D and E) so its latest start is week 3 (7—4).
- Activity A must be completed by week 8 (the latest start date for activity C) so its latest start is week 2 (8—6).
- The latest start date for the project start is the earliest of the latest start dates for activities A, B and F. This is week zero.

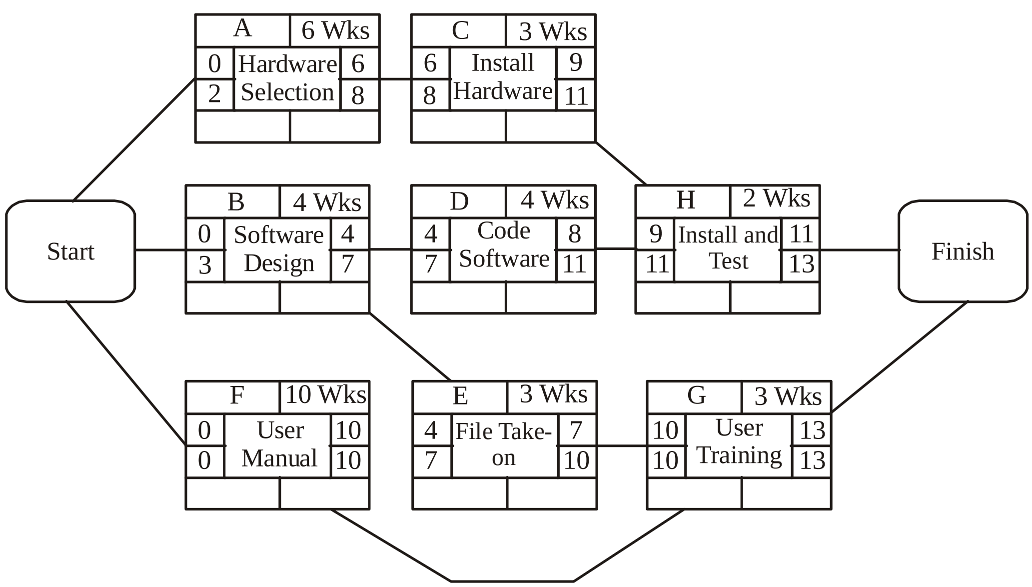

Figure 4.13: The Network after the Backward Pass

NOTES

4.10 IDENTIFYING THE CRITICAL PATH

There will be at least one path through the network (that is, one set of successive activities) that defines the duration of the project. This is known as the critical path. Any delay to any activity on this critical path will delay the completion of the project.

The difference between an activity's earliest start date and its latest start date (or, equally, the difference between its earliest and latest finish dates) is known as the activity's float. Float is a measure of how much the start or completion of an activity may be delayed without affecting the end date of the project. Any activity with a float of zero is critical in the sense that any delay in carrying out the activity will delay the completion date of the project as a whole. There will always be at least one path through the network joining those critical activities. This path is known as the critical path and is illustrated as bold in Figure 4.14.

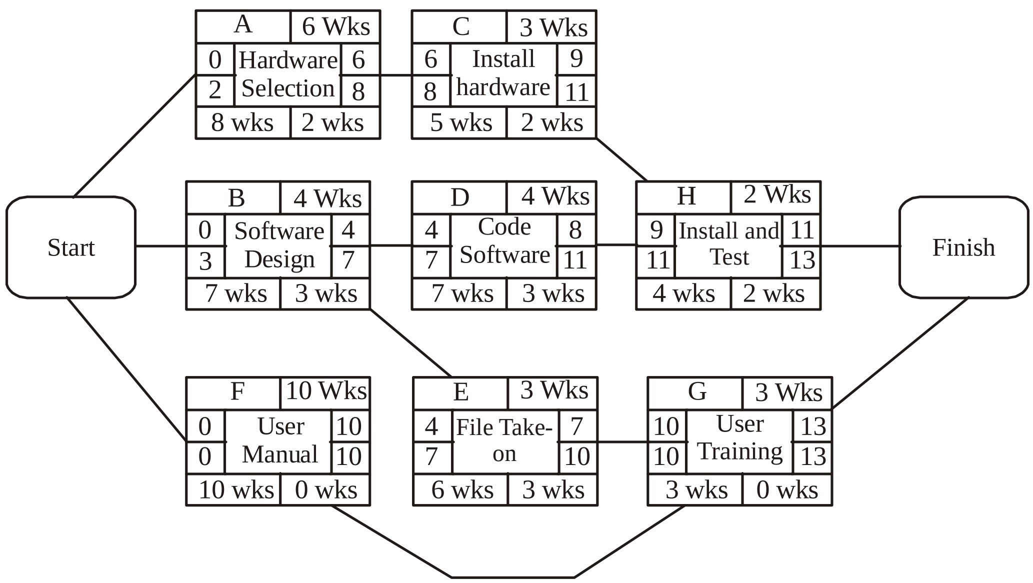

Figure 4.14: The Critical Path

The significance of the critical path is two-fold:

- In managing the project, we must pay particular attention to monitoring activities on the critical path so that the effects of any delay or resource unavailability are detected and corrected at the earliest opportunity.
- In planning the project, it is the critical path that we must shorten if we are to reduce the overall duration of the project.

Figure 4.14 also shows the activity span. This is the difference between the earliest start date and the latest finish date and is a measure of the maximum time allowable for the activity.

NOTES

4.11 ACTIVITY FLOAT

Although the total float is shown for each activity, it really 'belongs' to a path through the network. Activities A and C in Figure 4.13 each have 2 weeks' total float. If, however, activity A uses up its float (that is, it is not completed until week 8) then activity B will have zero float (it will have become critical). In such circumstances it may be misleading and detrimental to the project's success to publicize total float.

There are a number of other measures of activity float, including the following:

- Free Float: the time by which an activity may be delayed without affecting any subsequent activity. It is calculated as the difference between the earliest completion date for the activity and the earliest start date of the succeeding activity. This might be considered a more satisfactory measure of float for publicizing to the staff involved in undertaking the activities.
- Interfering Float: the difference between total float and free float. This is quite commonly used, particularly in association with the free float. Once the free float has been used (or if it is zero), the interfering float tells us by how much the activity may be delayed without delaying the project end date, even though it will delay the start of subsequent activities.

4.12 SHORTENING THE PROJECT DURATION

If we wish to shorten the overall duration of a project we would normally consider attempting to reduce activity durations. In many cases this can be done by applying more resources to the task, working overtime or procuring additional staff.

4.13 ACTIVITY-ON-ARROW NETWORKS

In activity-on-arrow networks activities are represented by links (or arrows) and the nodes represent events of activities (or groups of activities) starting or finishing. An example project specification with estimated activity durations and precedence requirements shown in Table 4.1 is used here for illustrating how to draw activity-on-arrow networks. Figure 4.15 illustrates an activity-on-arrow network for the example project.

Rules for Developing Activity-on-Arrow Networks

- A project network may have only one start node: This is a requirement of activity-on-arrow networks rather than merely desirable as is the case with activity-on-node networks.
- A project network may have only one end node: This is a requirement for activity-on-arrow networks.
- A link has duration: A link represents an activity and, in general, activities take time to execute. The network in Figure 4.15 does not contain any reference to durations. The links are not drawn in any way to represent the activity durations.

NOTES

- Nodes have no duration: Nodes are events and, as such, are instantaneous points in time. The source node is the event of the project becoming ready to start and the sink node is the event of the project becoming completed. Intermediate nodes represent two simultaneous events:
  - o The event of all activities leading in to a node having been completed and
  - o The event of all activities leading out of that node being in a position to be started

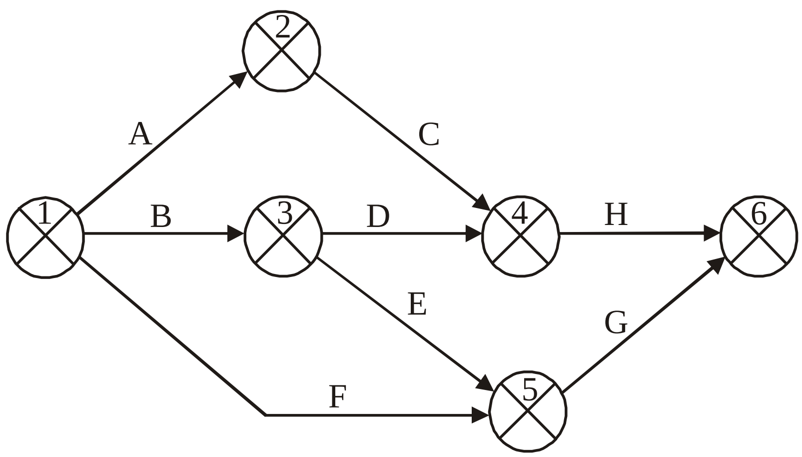

Figure 4.15: An Activity-on-Arrow Network

In Figure 4.16 node 3 is the event that both coding' and data take-on' have been completed and activity program test' is free to start. Instal1ation' may be started only when event 4 has been achieved, that is, as soon as program test' has been completed.

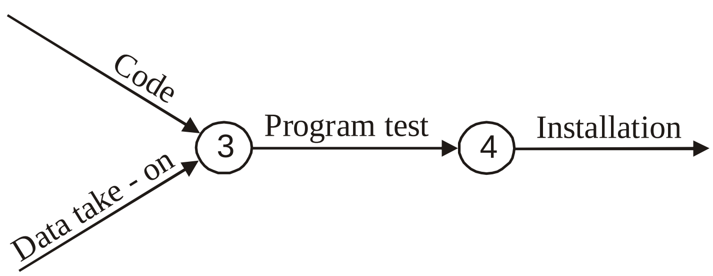

Figure 4.16: Fragment of a CPM network

- Time moves from left to right: As with activity-on-node networks, activity-on-arrow networks are drawn, if at all possible, so that time moves from left to right.
- Nodes are numbered sequentially: There are no precise rules about node numbering but nodes should be numbered so that head nodes always have a higher number than tail events
- A network may not contain loops. Figure 4.17 demonstrates a loop in an activity-on-arrow network. Loops are either an error of logic or a situation that must be resolved by itemizing iterations of activity groups.

NOTES

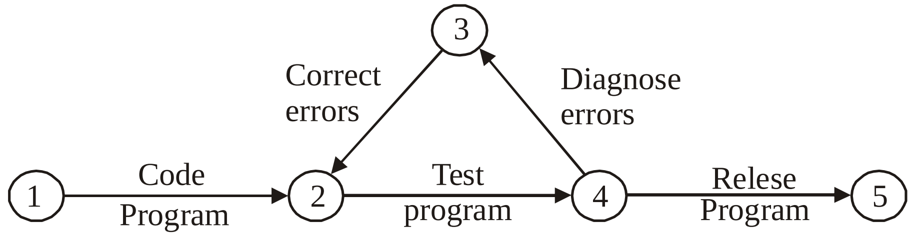

Figure 4.17: A Loop Represents an Impossible Sequence

- A network may not contain dangles: A dangling activity such as 'Write user manual' in Figure 4.18 cannot exist, as it would suggest there are two completion points for the project. If, in Figure 4.18 node 5 represents the true project completion point and there are no activities dependent on activity 'Write user manual', then the network should be redrawn so that activity 'Write user manual' starts at node 2 and terminates at node 5. In practice, a dummy activity has to be inserted between nodes 3 and 5. In other words, all events, except the first and the last, must have at least one activity entering them and at least one activity leaving them and all activities must start and end with an event.

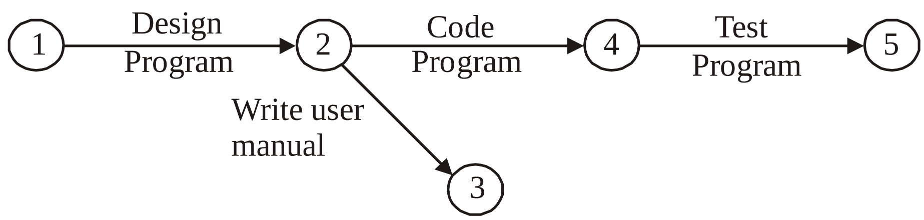

Figure 4.18: A Dangle

Using Dummy Activities

When two paths within a network have a common event although they are, in other respects, independent, a logical error such as that illustrated in figure 4.19 might occur.

Suppose that, in a particular project, it is necessary to specify a certain piece of hardware before placing an order for it and before coding the software. Before coding the software it is also necessary to specify the appropriate data structures, although clearly we do not need to wait for this 'to be done before the hardware is ordered.

Figure 4.19 is an attempt to model the situation described above, although it is incorrect in that it requires both hardware specification and data structure design to be completed before either an order may be placed or software coding may commence.

This problem can be resolved by separating the two (more or less) independent paths and introducing a dummy activity to link the completion of 'specify hardware' to the start of the activity 'code software'. This effectively breaks the link between data structure design and placing the order and is shown in Figure 4.20.

Dummy activities, shown as dotted lines on the network diagram, have zero duration and use no resources.

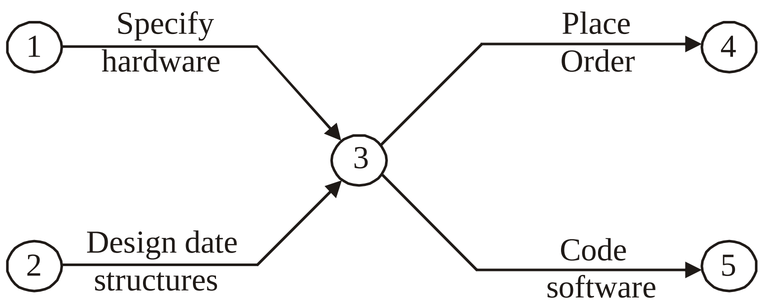

Figure 4.19 Two Paths with a Common Node

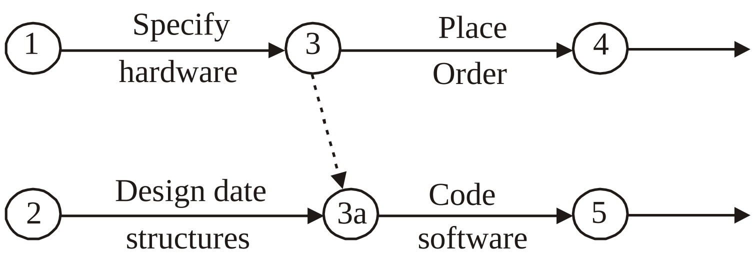

Figure 4.20: Two Paths Linked by a Dummy Activity

Representing Lagged Activities

Activity-on-arrow networks are less elegant when it comes to representing lagged parallel activities. This can be represented with pairs of dummy activities as shown in Figure 4.21. Where the activities are lagged because a stage in one activity must be completed before the other may proceed, it is likely to be better to show each stage as a separate activity.

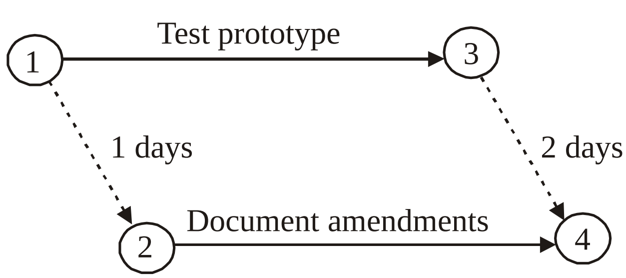

Figure 4.21: Using the Ladder Technique to Indicate Lags

Activity Labeling

One of the more common conventions for labeling nodes is to divide the node circle into quadrants and use those quadrants to show the event number, the latest and earliest dates by which the event should occur, and the event slack. The same is illustrated in Figure 4.22.

NOTES

NOTES

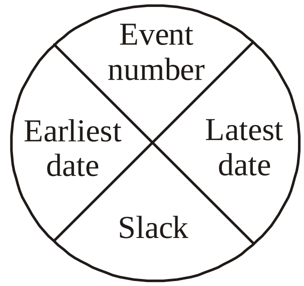

Figure 4.22: Convention for Activity labeling

Network Analysis

Analysis proceeds in the same way as with activity-on-node networks.

The Forward Pass: The forward pass is carried out to calculate the earliest date on which each event may be achieved and the earliest dates on which each activity may be started and completed. The earliest date for an event is the earliest date by which all activities upon which it depends can be completed. Using Figure 4.15 and an example project specification with estimated activity durations and precedence requirements shown in Table 4.1, the calculation proceeds according to the following reasoning.

- Activities A, B and F may start immediately, so the earliest date for event 1 is zero and the earliest start date for these three activities is also zero.
- Activity A will take 6 weeks, so the earliest it can finish is week 6 (recorded in the activity table). Therefore the earliest we can achieve event 2 is week 6.
- Activity B will take 4 weeks, so the earliest it can finish and the earliest we can achieve event 3 is week 4.
- Activity F will take 10 weeks, so the earliest it can finish is week 10. However, we cannot tell whether or not this is also the earliest date that we can achieve event 5 since we have not, as yet, calculated when activity E will finish
- Activity E can start as early as week 4 (the earliest date for event 3) and, since it is forecasted to take 3 weeks, will be completed, at the earliest, at the end of week 7.
- Event 5 may be achieved when both E and F have been completed, that is, week 10 (the later of 7 and 10).
- Similarly, we can reason that event 4 will have an earliest date of week 9. This is the later of the earliest finish for activity D (week 8) and the earliest finish for activity C (week 9).
- The earliest date for the completion of the project, event 6, is therefore the end of week 13 — the later of 11 (the earliest finish for H) and 13 (the earliest finish for G).

NOTES

The results of the forward pass are shown in Figure 4.23 and Table 4.2.

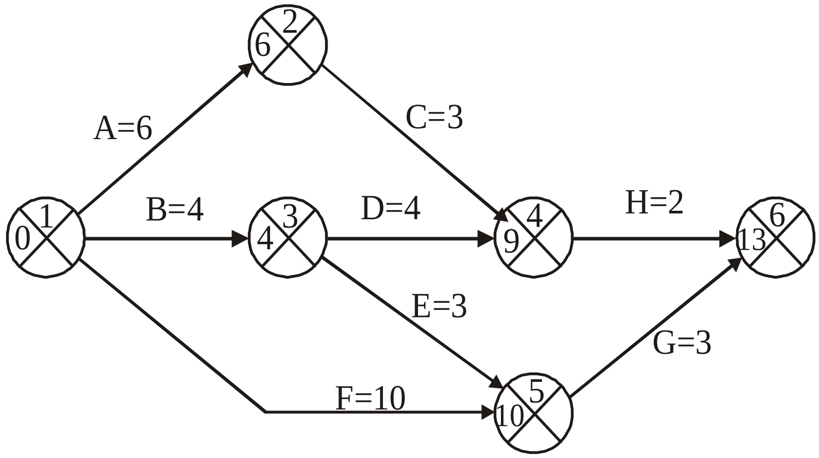

Figure 4.23 A CPM Network after the Forward Pass.

Table 4.2: The Activity Table after the Forward Pass

Activity	Duration (Weeks)	Earliest Start Date	Latest Start Date	Earliest Finish Date	Latest Finish Date	Total Float
A	6	0		6		
B	4	0		4		
C	3	6		9		
D	4	4		8		
E	3	4		7		
F	10	0		10		
G	3	10		13		
H	2	9		11		

The Backward Pass: The second stage is to carry out a backward pass to calculate the latest date at which each event may be achieved, and each activity started and finished, without delaying the end date of the project. The latest date for an event is the latest date by which all immediately following activities must be started for the project to be completed on time. As with activity-on-node networks, we assume that the latest finish date for the project is the same as the earliest finish date that is, we wish to complete the project as early as possible. Figure 4.24 illustrates our network and Table 4.3 the activity table after carrying out the backward pass as with the forward pass, event dates are recorded on the diagram and activity dates on the activity table.

NOTES

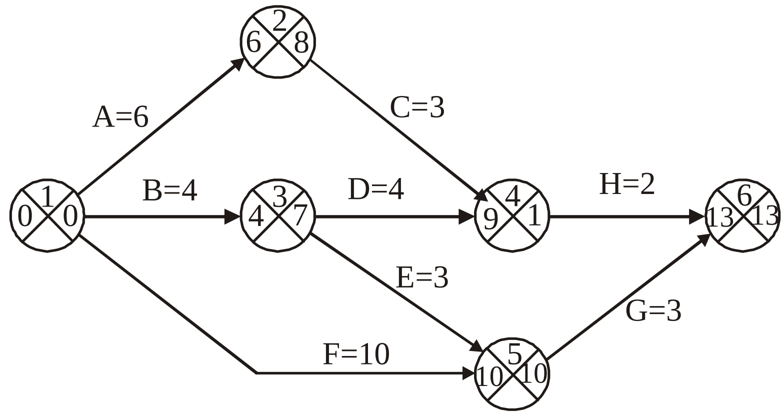

Figure 4.24: The CPM Network after the Backward Pass

Table 4.2: The Activity Table after the Backward Pass

Activity	Duration (Weeks)	Earliest Start Date	Latest Start Date	Earliest Finish Date	Latest Finish Date	Total Float
A	6	0	2	6	8	
B	4	0	3	4	7	
C	3	6	8	9	11	
D	4	4	7	8	11	
E	3	4	7	7	10	
F	10	0	0	10	10	
G	3	10	10	13	13	
H	2	9	11	11	13	

Identifying the Critical Path: The critical path is identified in similar manner used in activity-on-node networks. However, a different concept, of slack is used, in identifying the path. Slack is the difference between the earliest date and the latest date for an event. It is a measure of how late an event may be without affecting the end date of the project. The critical path is the path joining all nodes with a zero slack (Figure 4.25).

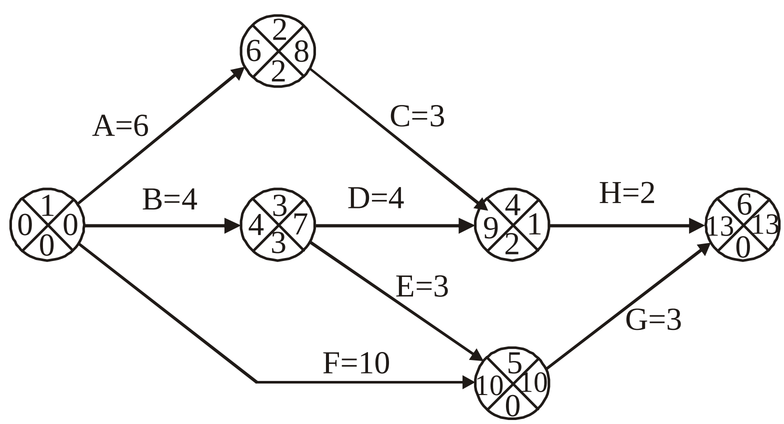

Figure 4.25: The Critical Path

NOTES

Summary:

Planning is an ongoing process of refinement, each iteration becoming more detailed and more accurate than the last. Over successive iterations, the emphasis and purpose of planning will shift. During the feasibility study and project startup, the main purpose of planning will be to estimate timescales and the risks of not achieving target completion dates or keeping within budget. As the project proceeds beyond the feasibility study, the emphasis will be placed upon the production of activity plans for ensuring resource availability and cash flow control.

Throughout the project, until the final deliverable has .reached the customer, monitoring and replanning must continue to correct any drift that might prevent meeting time or cost targets.

Questions:

1. List the objectives of activity planning.
2. What is a project schedule? Discuss.
3. With an example explain a work break down structure.
4. List the steps involved in constructing precedence networks.
5. List the rules for constructing activity-on-arrow networks. Give relevant examples.

NOTES

CHAPTER - 5

RISK MANAGEMENT

Projects can fail for a number of reasons and the risks are always high. While a project manager cannot eliminate risk, he/she can prevent or mitigate its impacts by using risk management. Project Risk Management, describes the processes concerned with conducting risk management on a project. It consists of the Risk Management Planning, Risk Identification, Qualitative Risk Analysis, Quantitative Risk Analysis, Risk Response Planning, and Risk Monitoring and Control project management processes.

Risk is an uncertain event that could have a positive or negative outcome. One of the toughest jobs for any project manager is managing project risks. In addition to risks that affect your ability to complete your assignments, project risks include the following:

- Inadequate time for completing the project
- Inadequate budget for completing the project
- Unrealistic scope expectations
- A project team that needs additional time to ramp up development language
- Stakeholders that do not or cannot provide clear project requirements

5.1 RISK

"Risk is the potential future harm that may arise from some present action" ("A possible future event that, if it occurs, will lead to an undesirable outcome"). Management is a process that is used to minimize or eradicate risk before it can harm the productivity of software. There are two risk strategies namely reactive strategies and proactive strategies. A reactive software engineer corrects a problem as it occurs, while a proactive software engineer starts thinking about possible risks in a project before they occur.

Risks are potential problems that might affect the successful completion of a software project. Risks involve uncertainty and potential losses. Risk analysis and management are intended to help a software team understand and manage uncertainty during the development process. The important thing is to remember that things can go wrong and to make plans to minimize their impact when they do. The work product is called a Risk Mitigation, Monitoring, and Management Plan (RMMM).

There are several types of risk that can occur during a software development project. The same is illustrated in Table 5.1.

NOTES

Table 5.1: Types of Risk and their Description

Risk Type	Description
Generic Risks	Generic threats across all projects. For example, requirements change, loss of team members, loss of funding
Product-Specific Risks	High level risks associated with the type of product being developed. For example: availability of testing resources
Project Risks	Affect project schedule or resources
Product Risks	Affect quality or performance of software
Business Risks	Affect the viability of the software

There are also specific risks associated with team members, customers, tools, technology, time estimation, and team size. Many of these risks can be minimized by the development methodology used for the project. There are many different tools that can be used to analyze the risk apparent in a project and that can help choose the best way to minimize or eliminate that risk.

Business risks are a big concern for any project manager with an eye towards reality. Business risks are the more common risks you encounter in your project management activities:

- Employees quit
- Mistakes are made in the requirements gathering process
- The software is full of bugs, errors, and failures
- The scope of the project grows, but the budget (or the timeline) doesn't
- The expectations of the project time, cost, and scope are not realistic to begin with
- The project is larger than the capacity of the project team
- The project manager, sponsor, or other stakeholders are not as knowledgeable as you would hope

5.1.1 Risk Management:

Risk management is a management specialism aiming to reduce different risks related to a preselected domain to the level accepted by society. It may refer to numerous types of threats caused by environment, technology, humans, organizations and politics. Risk management is a way to manage risks. In other words, it concerns all activities that are performed to reduce the uncertainties associated with certain tasks, or events. In the context of projects, risk management reduces the impacts of undesirable events on a project. Risk management in any project requires undertaking decision making activities. Software engineering risk management activities are conducted at the project level, process level, and product level. Risk Management is a practice with processes, methods, and tools for

NOTES

managing risks in a project. It provides a disciplined environment for proactive decision making to:

- Assess continuously what could go wrong (risks)
- Determine which risks are important to deal with
- Implement strategies to deal with those risks

Risk management in software projects has different uses. It helps to save projects from failing due to different factors such as non-completion of projects within the specified schedule, and budget constraints, and not meeting customer expectations. Risk management looks at projects from different perspectives to ensure that the threats to the projects are identified, and analyzed, and appropriate strategies are undertaken to mitigate, and control risks. The mitigation strategies may not necessarily mean the cancellation of tasks that involve risks. Many tasks are undertaken in the software industries even after knowing that undertaking them involves taking high risks. The high-risk tasks are sometimes important to provide the industries a leading edge over their competitors.

The main purpose of risk management is to know all possible risks to a project, assess their severity, and consequence, and then determine resolution steps depending on the nature of the risks. The idea is to minimize any unforeseen and unexpected issues arising during the course of the project by properly planning for eventualities. Proper planning leads to minimizing uncertainties, which might lead to a "turbulent" completion, or a complete cancellation of the projects. Software engineering risk management takes a preventative approach leading to completion of projects within predictable time, and money. In fact, risk-managed projects have the ability to reduce project costs, and time of completion, and increase the overall quality of the project deliverables. Without these, projects could risk loss of revenue, and customer trust in an average case, or a complete bankruptcy of the participating organizations in the worst.

Before applying any risk management process, the project team members should be clear about the following dimensions of risks in their projects:

- The nature of uncertainty involved, and the likelihood with which the risk will occur
- The loss that will be incurred if the risk occurs. Loss in software projects can take many forms including loss of revenue, loss of market share, and loss of customer goodwill.
- The severity of the loss
- The duration of the risks

5.1.2 Risk Identification:

- Product-specific risks - the project plan and software statement of scope are examined to identify any special characteristics of the product that may threaten the project plan

- Generic risks - are potential threats to every software product (product size, business impact, customer characteristics, process definition, development environment, technology to be built, staff size and experience)

NOTES

5.1.3 Risk Impact:

- Risk components - performance, cost, support, schedule
- Risk impact - negligible, marginal, critical, catastrophic
- The risk drivers affecting each risk component are classified according to their impact category and the potential consequences of each undetected software fault or unachieved project outcome are described

5.1.4 Risk Projection (Estimation):

- Establish a scale that reflects the perceived likelihood of each risk
- Delineate the consequences of the risk
- Estimate the impact of the risk on the project and product
- Note the overall accuracy of the risk projection to avoid misunderstandings

5.1.5 Risk Table Construction:

- List all risks in the first column of the table
- Classify each risk and enter the category label in column two
- Determine a probability for each risk and enter it into column three
- Enter the severity of each risk (negligible, marginal, critical, catastrophic) in column four
- Sort the table by probability and impact value
- Determine the criteria for deciding where the sorted table will be divided into the first priority concerns and the second priority concerns
- First priority concerns must be managed (a fifth column can be added to contain a pointer into the RMMM)

5.1.6 Assessing Risk Impact:

- Factors affecting risk consequences - nature (types of problems arising), scope (combines severity with extent of project affected), timing (when and how long impact is felt)
- If costs are associated with each risk table entry Halstead's risk exposure metric can be computed (RE = Probability * Cost) and added to the risk table.

5.1.7 Risk Assessment:

- Define referent levels for each project risk that can cause project termination (performance degradation, cost overrun, support difficulty, schedule slippage).

NOTES

- Attempt to develop a relationship between each risk triple (risk, probability, impact) and each of the reference levels.
- Predict the set of referent points that define a region of termination, bounded by a curve or areas of uncertainty.
- Try to predict how combinations of risks will affect a referent level.

5.1.8 Risk Refinement:

- Process of restating the risks as a set of more detailed risks that will be easier to mitigate, monitor, and manage.
- CTC (condition-transition-consequence) format may be a good representation for the detailed risks (e.g. given that <condition> then there is a concern that (possibly) <consequence>).

5.1.9 Risk Mitigation, Monitoring, and Management:

- Risk mitigation (proactive planning for risk avoidance)
- Risk monitoring (assessing whether predicted risks occur or not, ensuring risk aversion steps are being properly applied, collect information for future risk analysis, attempt to determine which risks caused which problems)
- Risk management and contingency planning (actions to be taken in the event that mitigation steps have failed and the risk has become a live problem)

5.1.10 Safety Risks and Hazards:

- Risks are also associated with software failures that occur in the field after the development project has ended.
- Computers control many mission critical applications in modern times (weapons systems, flight control, industrial processes, etc.).
- Software safety and hazard analysis are quality assurance activities that are of particular concern for these types of applications.

5.1.11 Risk Information Sheets:

- Alternative to RMMM in which each risk is documented individually.
- Often risk information sheets (RIS) are maintained using a database system.
- RIS components - risk id, date, probability, impact, description, refinement, mitigation/monitoring, management/contingency/trigger, status, originator, assigned staff member.

5.2 STRATEGIES FOR RISK REDUCTION

There are five strategies for risk reduction:

- **Hazard Prevention:** Some hazards can be prevented from occurring or their likelihood reduced to insignificant levels. The risk of key staff being unavailable for meetings can be minimized by early scheduling, for example.

- **Likelihood Reduction:** Some risks, while they cannot be prevented, can have their likelihoods reduced by prior planning. The risk of late changes to a requirements specification can, for example, be reduced by prototyping. Prototyping will not eliminate the risk of late changes and will need to be supplemented by contingency planning.
- **Risk Avoidance:** A project can, for example, be protected from the risk of overrunning the schedule by increasing duration estimates or reducing functionality.
- **Risk Transfer**: The impact of some risks can be transferred away from the project by, for example, contracting out or taking out insurance.
- **Contingency Planning:** Some risks are not preventable and contingency plans will need to be drawn up to reduce the impact should the hazard occur. A project manager should draw up contingency plans for using agency programmers to minimize the impact of any unplanned absence of programming staff.

5.3 APPROACHES TO RISK IDENTIFICATION

There are two main approaches for indentifying risks. They are listed below:

- Use of Checklists
- Brainstorming

Software projects risks and strategies for risk reduction are tabulated in Table 5.1.

Table 5.1: Software Projects Risks and Strategies for Risk Reduction

RISK	RISK REDUCTION TECHNIQUES
Personnel Shortfalls	Staffing with Top Talent, ob matching, Team Building, Training and Career Development, and Early Scheduling of Key Personnel
Unrealistic Time and Cost Estimates	Multiple Estimation Techniques, Design to Cost, Incremental Development, Recording and Analysis of Past Projects, and Standardization of Methods.
Developing the Wrong Software Functions	Improved Project Evaluation, Formal Specification Methods, User Surveys, Prototyping, and Early Users Manuals
Developing the Wrong User Interface	Prototyping, Task Analysis, and User Involvement
Gold Plating	Requirements Scrubbing, Prototyping, Cost-benefit Analysis and Design to Cost.
Late Changes to Requirements	Stringent Change Control Procedures, High Change Threshold, Incremental Prototyping, and Incremental Development(Defer Changes)

NOTES

Table 5.1: Software Projects Risks and Strategies for Risk Reduction (Continued)

RISK	RISK REDUCTION TECHNIQUES
Shortfalls in External Supplied Components	Benchmarking, Inspections, Formal Specifications, Contractual Agreements, and Quality Assurance Procedures and Certification.
Shortfalls in Externally Performed Tasks	Quality Assurance Procedures, Competitive Design or Prototyping, Team Building, and Contract Incentives.
Real-time Performance Shortfalls	Simulation, Bench marking, Prototyping, Tuning, and Technical Analysis
Development Technically too Difficult	Technical Analysis, Cost-benefit Analysis, Prototyping, Staff Training and Development.

5.4 **SWOT** ANALYSIS

SWOT analysis (strengths, weaknesses, opportunities, and threats) can also be used during risk identification. SWOT analysis helps identify the broad negative and positive risks that apply to a project.

5.5 USE OF PROGRAM EVALUATION REVIEW TECHNIQUE (PERT) TO EVALUATE THE EFFECTS OF UNCERTAINTY

By identifying and categorizing those risks, and in particular, their likely effects on the duration of planned activities, we can assess what impact they are likely to have on the activity plan. PERT can be used for assessing the effects of these uncertainties on the project schedule.

PERT was developed to take account of the uncertainty surrounding estimates of task durations.

PERT requires three estimates:

Most Likely Time: The time we would expect the task to take under normal circumstances. This may be denoted by the letter ***m***.

Optimistic Time: The shortest time, in which we could expect to complete the activity, barring outright miracles. This may be denoted by the letter ***a.***

Pessimistic Time: The worst possible time allowing for all reasonable eventualities. This may be denoted by the letter ***b.***

PERT then combines these three estimates to form a single expected duration, t_e using the mathematical model

$$t_e = (a + 4m + 6) / 6$$

The major advantage of PERT is, PERT focuses attention on the uncertainty of forecasting.

Summary

Risk management is now recognized as one of the most important project management tasks. Risk management involves identifying and assessing project risks to establish the probability that they will occur and the consequences for the project if that risk does arise. You should make plans to avoid, manage or deal with likely risks if or when they arise

Questions

1. List and discuss the different types of risks.
2. Present a tutorial on risk management.
3. Explain risk mitigation, monitoring, and management.
4. What is SWOT analysis?

NOTES

NOTES

UNIT IV

CHAPTER 6
MONITORING AND CONTROL

After the work schedules have been finalized and the project is under way, concentration must be focused on ensuring progress. This requires monitoring of what is happening, comparison of actual achievement against the schedule and, when necessary, revision of plans and schedules to bring the project as far as possible back on target.

6.1 CREATING THE FRAMEWORK

Exercising control over a project and ensuring that targets are met is a matter of regular monitoring, finding out what is happening, and comparing it with current targets. If there is a mismatch between the planned outcomes and the actual one then either replanning is needed to bring the project back on target or the target will have to be revised. A model of the project control cycle is illustrated in Figure 6.1.

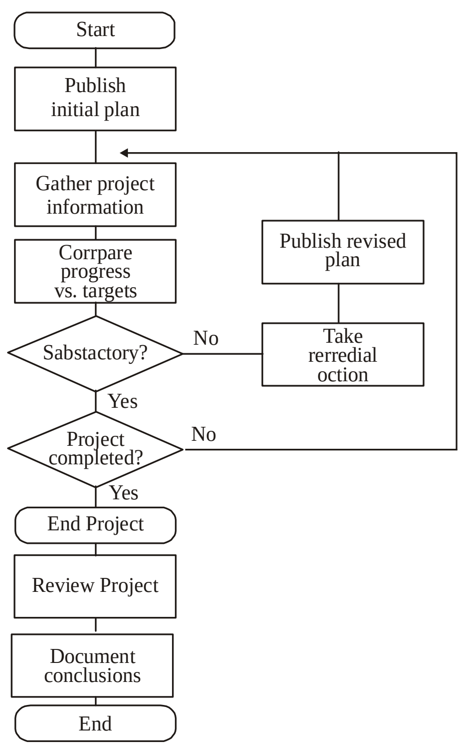

Figure 6.1: The Project Control Cycle

NOTES

It shows how, once the initial project plan has been published, project control a continual process of monitoring progress against that plan and, where necessary revising the plan to take account of deviations. It also illustrates the important steps that must be taken after completion of the project so that the experience gained in anyone project can feed into the planning stages of future projects, thus allowing us to learn from past mistakes.

In practice we are normally concerned with departures from the plan in four dimensions delays in meeting target dates, shortfalls in quality, inadequate func-tionality, and costs going over target. In this chapter we are mainly concerned with the first and last of these.

Responsibility:

The overall responsibility for ensuring adequate progress on a project is often the role of the project-steering committee or Project Board. Day-to-day responsibility will be with the project manager and, in all but the smallest of projects; aspects of this can be delegated to team leaders. Figure 6.2 illustrates the typical reporting structure found with medium and large projects.

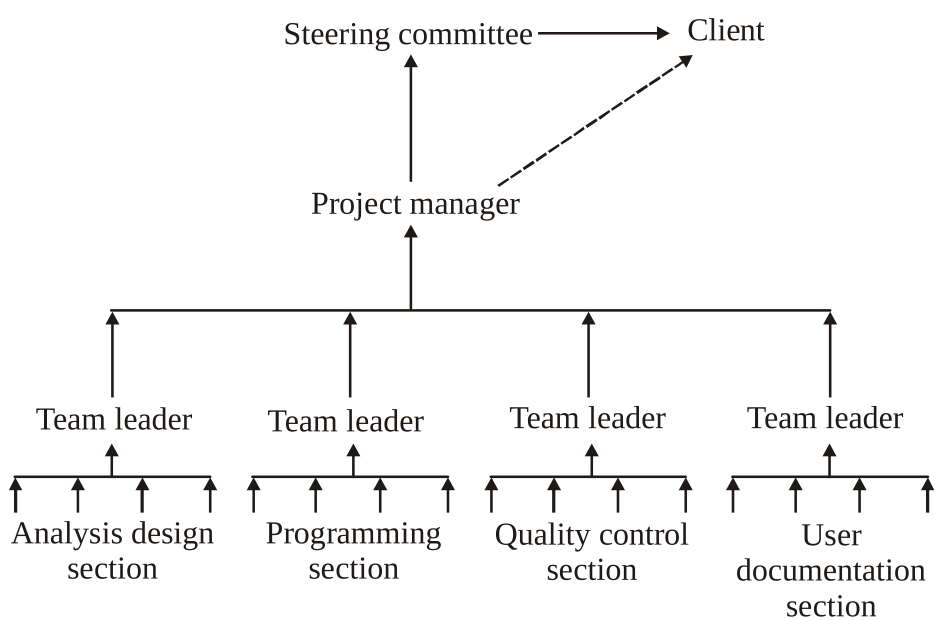

Figure 6.2: Project Reporting Structure

With small projects employing less number of staff individual team members usually report directly to the project manager. But in most cases team leaders will collate reports on their section's progress and forward summaries to the project manager. These, in turn, will be incorporated into project-level reports for the steering committee and, via them or directly, progress reports for the client.

Reporting may be oral or written, formal or informal, or regular or ad hoc. *S*ome examples of each type are tabulated in Table 6.1.

Any effective team leader or project manager will be in touch with team members and available to discuss problems, any such informal reporting of project progress must be comple-mented by formal reporting procedures.

Assessing the Progress

Progress assessment will be made on the basis of information collected and collated at regular intervals or when specific events occur. Wherever possible, this information will be objective and tangible - whether or not a particular report has been delivered. Progress assessment will have to rely on the judgment of the team members who are carrying out the project activities.

Table 6.1: Categories of Reporting

Report Type	Examples	Comment
Oral formal regular	Weekly or monthly progress meetings	While reports may be oral, formal written minutes should be kept
Oral formal ad hoc	End-of-stage review meetings	While largely oral, likely to receive and generate written reports
Written formal regular	Job sheets, Progress reports	Normally weekly using forms
Written formal ad hoc	Exception reports, Change reports	
Oral informal ad hoc	Canteen discussion, Social interaction	Often provides early warning; must be backend up by formal reporting

Setting Checkpoints

A series of checkpoints in the initial activity plan need to be set. Checkpoints maybe:

- Regular (Daily, for example)
- Tied to specific events such as the production of a report or other deliverable

Taking Snapshots

The frequency with which a manager needs to receive information about progress will depend upon the size and degree of risk of the project or that part of the project under their control. Team leaders, for example, need to assess progress daily whereas project managers may find weekly or monthly reporting appropriate. In general, the higher the level, the less frequent and less detailed the reporting needs to be. A formal weekly collection of information from staff carrying out activities is favored.

NOTES

Collecting data at the end of each week ensures that information is provided while memories are still relatively fresh and provides a mechanism for individuals to review and reflect upon their progress during the past few days. Major, or project-level, progress reviews will generally take place at particular points during the life of a project - commonly known as review points or control points.

6.2 DATA COLLECTION

Managers will try to break down long activities into more controllable tasks of one or two weeks' duration. However, it will still be necessary to gather information about partially completed activities and, in particular, forecasts of how much work is left to be completed. It may be difficult to make such forecasts accurately.

Where there is a series of products, partial completion of activities is easier to estimate. Counting the number of record specifications or screen layouts produced, for example, can provide a reasonable measure of progress.

In some cases, intermediate products can be used as in-activity milestones. The first successful compilation of a program, for example, might be considered a milestone even though it is not the final product of the activity code and test.

Partial Completion Reporting

All organizations use standard accounting systems with weekly timesheets to charge staff time to individual jobs. The staff time booked to a project indicates the work carried out and the charges to the project. However, it does not, tell the project manager what has been produced or whether tasks are on schedule. It is therefore common to adapt or enhance existing accounting data collection systems to meet the needs of project control. Weekly time sheets, for example, are frequently adapted by breaking jobs down to activity level and requiring information about work done in addition to time spent. Figure 6.3 illustrates a typical example of such a report form, in this case requesting information about likely slippage of completion dates as well as estimates of completeness.

Asking for estimated completion times frequently should be avoided as this may affect the importance of the originally scheduled targets.

NOTES

TIME SHEET

Staff LOKESH S Week Ending 30/3/11

Rechargeable hours

Project	Activity Code	Description	Hours This Week	% Complete	Scheduled Completion	Estimated Completion
P21	A243	Code Module 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	S	Authorized by John Prakash

Total Non-Rechargeable Hours	8

Figure 6.3: A Weekly Timesheet and Progress Review Form

Risk Reporting

One method overcoming the objections to partial completion reporting is to avoid asking for estimated completion dates, but to ask instead for the team members' estimates of the likelihood of meeting the planned target date. One way of doing this is the traffic-light method. This consists of the following steps:

- Identify the key elements for assessment in a piece of work (first level)
- Break these key elements into constituent elements (second level)
- Assess each of the second-level elements on the scale green for 'on target', amber for 'not on target but recoverable', and red for 'not on target and recov-erable only with difficulty'
- Review all the second level assessments to arrive at first level assessments
- Review first and second level assessments to produce an overall assessment

NOTES

Each activity is broken into a number of component parts and deciding whether a further breakdown is needed and get the team members to complete a return at the end of each week.

Traffic-light assessment highlights only risk of non-achievement; it is not an attempt to estimate work done or to quantify expected delays. Following completion of assessment forms for all activities, the project manager uses these as a basis for evaluating the overall status of the project. Any critical activity classified as amber or red will require further consideration and often leads to a revision of the project schedule. Non-critical activities are likely to be con-sidered as a problem if they are classified as red, especially if their entire float is likely to be consumed. The same is illustrated in Figure 6.4.

ACTIVITY ASSESSMENT SHEET

Staff LINI

Ref: KoE/P/12 Activity: Code and test module C

Week Number	13	14	15	16	17	18	
Activity Summary	G	A	A	R			
Component							Comments
Screening Handling	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			

Figure 6.4: Example for a Traffic Light Assessment

6.3 VISUALIZING PROGRESS

Once data has been collected about project progress, a manager needs some way of pre-senting that data to greatest effect. Some of these methods such as Gantt charts provide a static picture, a single snapshot, whereas others such as time line charts try to show how the project has progressed and changed over time.

The Gantt Chart

Gantt chart is a technique used for tracking project progress. This is essentially an activity bar chart indicating scheduled activity dates and durations frequently augmented with activity floats. Reported progress is recorded on the chart by shading activity bars and a 'today cursor' pro-vides an immediate visual indication of which activities are ahead or behind schedule. Figure 6.5 shows an example of Gantt chart as at the end of Tuesday

NOTES

of week 17. 'Code & test module D' has been completed ahead of schedule and 'Code & test module A' appears also to be ahead of schedule. The coding and testing of the other two modules are behind schedule.

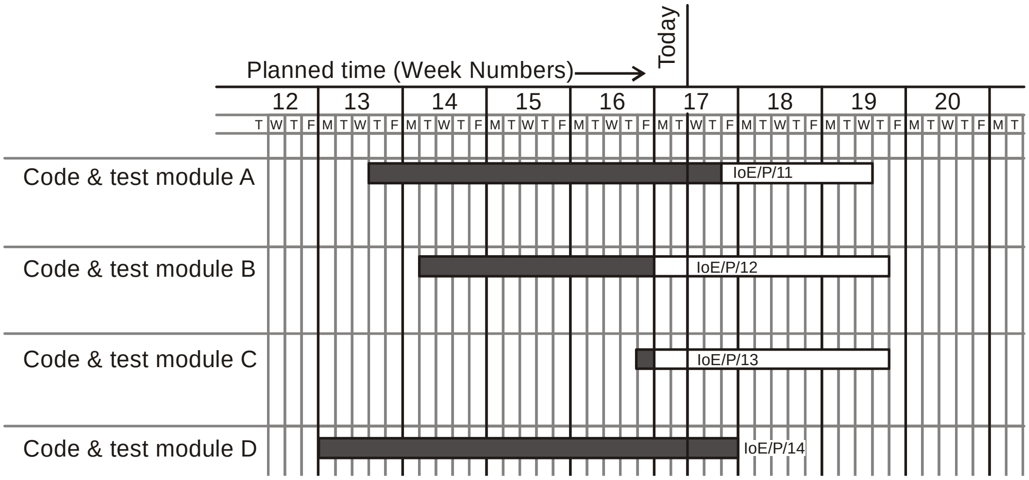

Figure 6.5: Gantt Chart with the 'Today Cursor' in Week 17

The Slip Chart

A slip chart illustrated in Figure 6.6 is an alternative favored by some project managers who believe it provides a more striking visual indication of those activi-ties that are not progressing to schedule. The more the slip line bends, the greater the variation from the plan. Additional slip lines are added at intervals and, as they build up, the project manager will gain an idea as to whether the project is improving subsequent slip lines bend less or not. A very jagged slip line indicates a need for rescheduling.

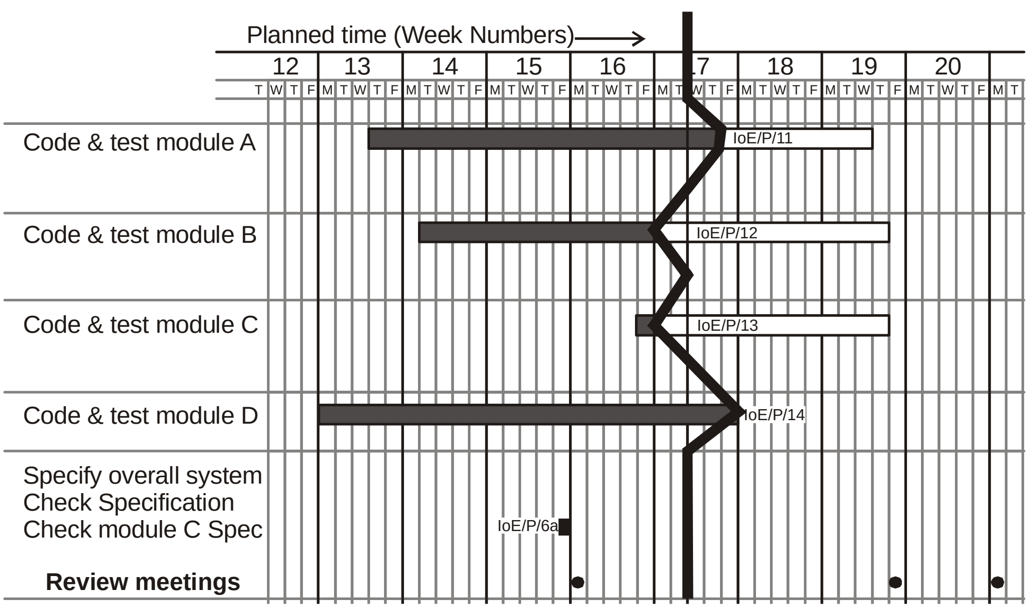

Figure 6.6: Slip Chart illustrating the Relative Position of Each Activity

NOTES

Ball Charts

A more prominent way of showing whether or not targets have been met is to use a ball chart. The same is illustrated in Figure 6.7. In this version of the ball chart, the circles indicate start and completion points for activities. The circles initially contain the original scheduled dates. Whenever revisions are produced, these are added as second dates in the appropriate circle until an activity is actually started or com-pleted, when the relevant date replaces the revised estimate (in bold italic in Figure 6.7).

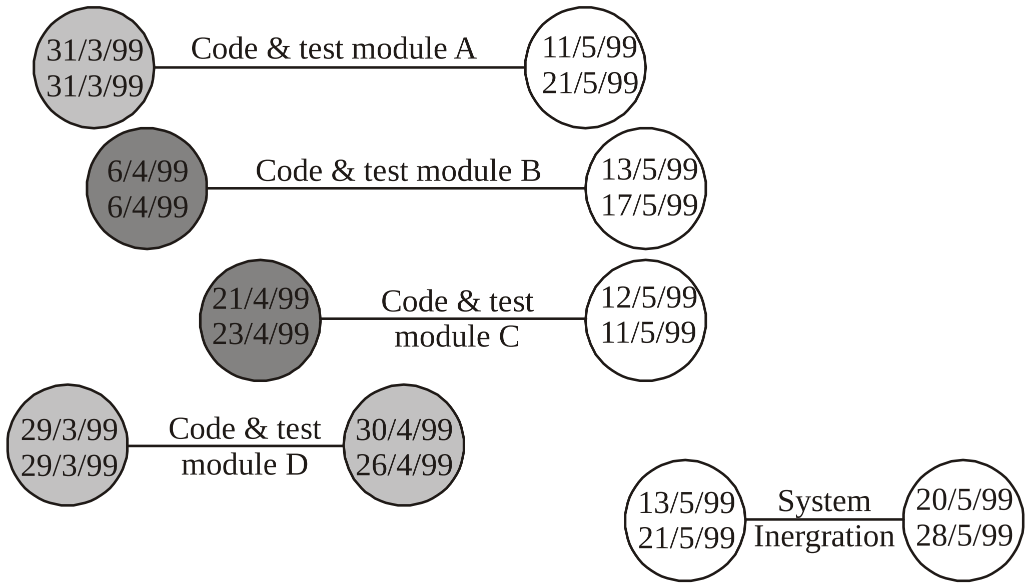

Figure 6.7: Example for Ball Chart

Circles will contain only two dates, the original and most recent target dates, or the original and actual dates. Where the actual start or finish date for an activity is later than the target date, the circle is colored red (dark grey in Figure 6.7) - where an actual date is on time or earlier than the target then the circle is colored green (light grey in Figure 6.7).

Such charts are frequently placed in a prominent position and the color-coded balls provide a constant reminder to the project team. Where more than one team is working in close proximity, such a highly visible record of achievement can encourage competitiveness between teams.

Another advantage of ball charts over Gantt and slip charts is that they are rela-tively easy to keep up to date. Only the dates and possibly colors need to be changed, whereas the others need to be redrawn each time target dates are revised.

The Timeline

One major disadvantage of Gantt chart, Slip chart and Ball chart is that they do not slow clearly the slippage of the project completion date through the life of the project. Knowing the current state of a project helps in revising plans to bring it back on target, but analyzing and understanding trends helps to avoid slippage in future projects.

NOTES

The timeline chart is a method of recording and displaying the way in which targets have changed throughout the duration of the project. Figure 6.8 illustrates a timeline chart for a project at the end of the sixth week. Planned time is plotted along the horizontal axis and elapsed time down the vertical axis. The lines meandering down the chart represent scheduled activity completion date.

During the start of the project 'analyze existing system' is scheduled to be completed by the Tuesday of week 3, 'obtain user requirements' by Thursday of week 5, 'issue tender', the final activity, by Tuesday of week 9, and so on.

At the end of the first week the project manager reviews these target dates and leaves them as they are - lines are therefore drawn vertically downwards from the target dates to the end of week one on the actual time axis. At the end of week 2, the project manager decides that 'obtain user requirements' will not be completed until Tuesday of week 6. So the project manager therefore extends that activity line diagonally to reflect this. The other activity completion targets are also delayed correspondingly. By the Tuesday of week 3, 'analyze existing system' is completed and the project manager puts a blob on the diagonal timeline to indicate that this has happened. At the end of week 3 the project manager decides to keep to the existing targets.

At the end of week 4 the project manager adds another three days to 'draft tender' and 'issue tender'

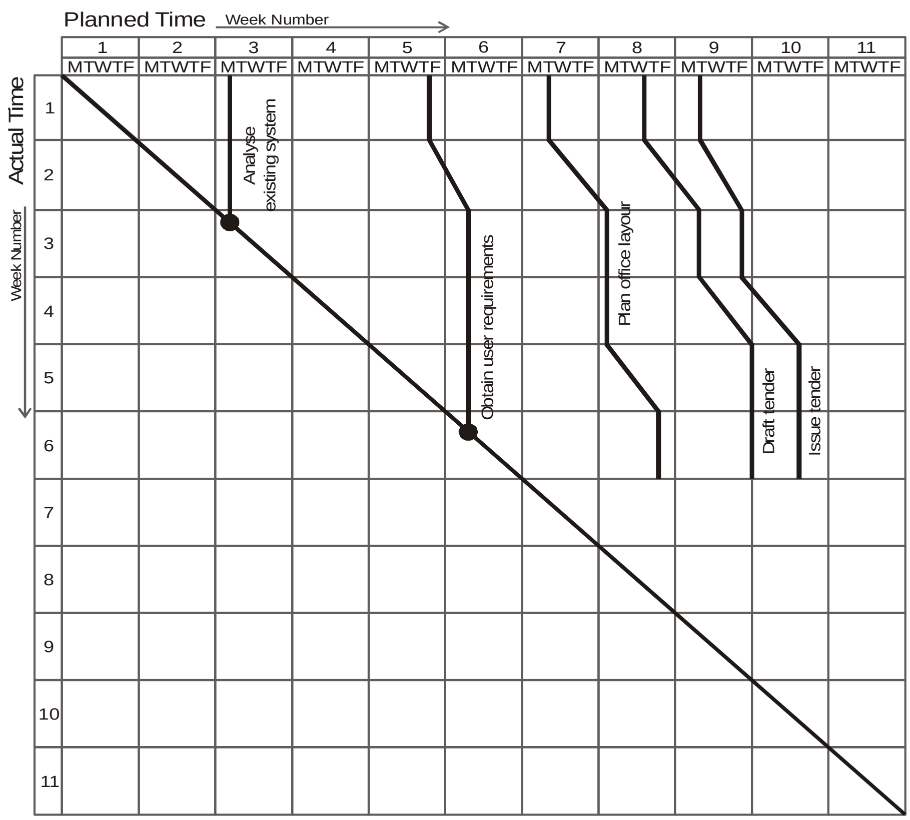

Figure 6.8: Example for Time Line Chart

NOTES

6.4 COST MONITORING

Expenditure monitoring is a vital component of project control because it provides an indication of the effort that has gone into a project. A project might be on time but only because more money has been spent on activities than originally budgeted. A cumulative expenditure chart such as that shown in Figure 6.9 provides a simple method of comparing actual and planned expenditure. Figure 6.9 illustrates a project that is running late or one that is on time but has shown substantial costs savings. The current status of the project activities has to be taken into account before attempting to interpret the meaning of recorded expenditure.

Cost charts become useful if we add projected future costs calcu-lated by adding the estimated costs of uncompleted work to the costs already incurred. Where a computer-based planning tool is used, revision of cost schedules is generally provided automatically once actual expenditure has been recorded.

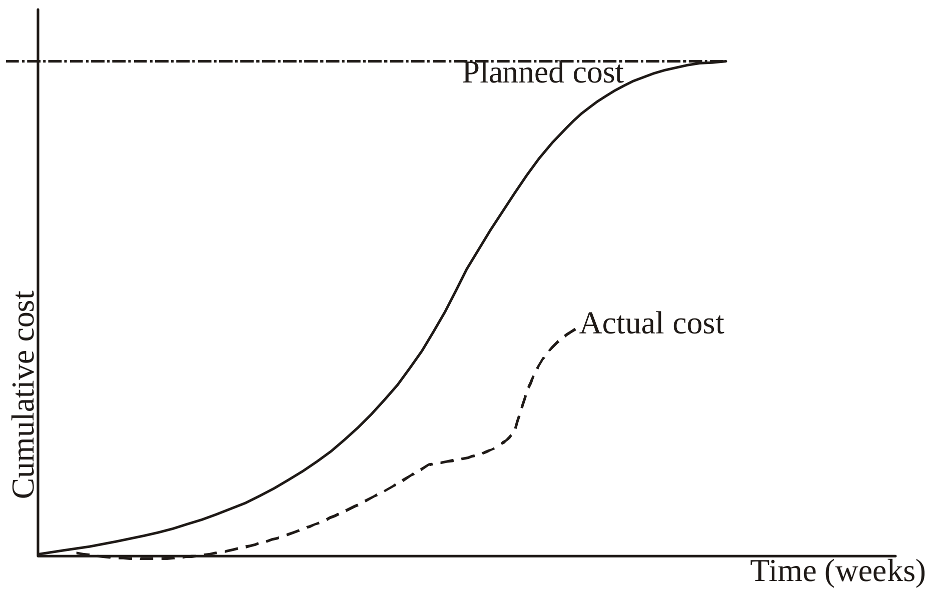

Figure 6.9: Tracking Cumulative Expenditure

Figure 6.10 illustrates the additional information available once the revised cost schedule is included. In this case it is apparent that the project is behind schedule and over budget.

NOTES

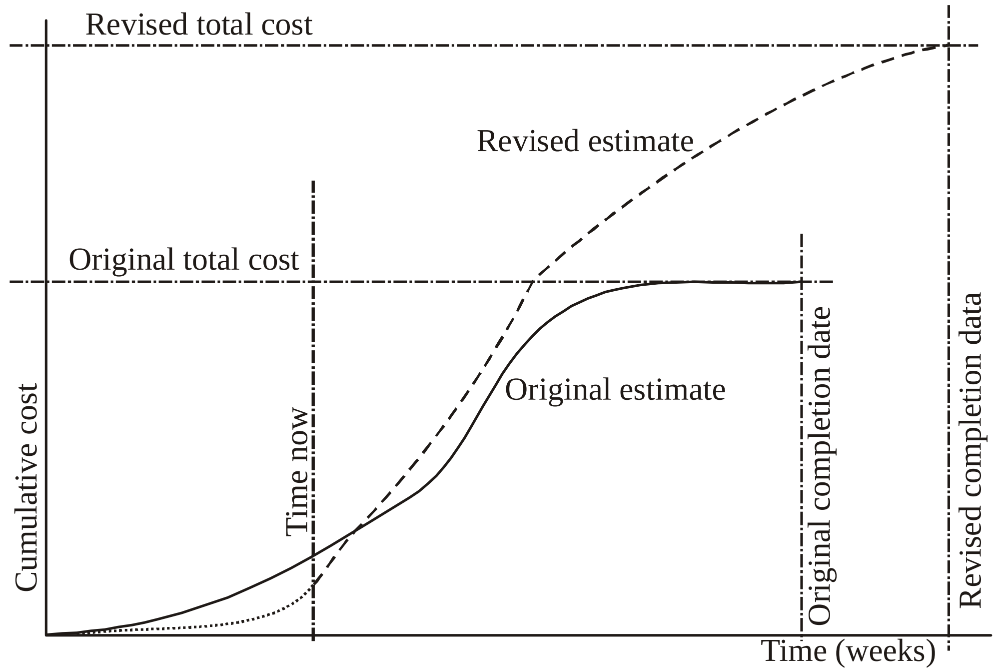

Figure 6.10: Cumulative Expenditure Chart Illustrating Revised Estimates of Cost and Completion Date

6.5 EARNED VALUE ANALYSIS

Earned value is as a refinement of the cost monitoring technique discussed in section 6.4. Earned value analysis is based on assigning a 'value' to each task or work package as identified in the work breakdown structure (WBS) based on the original expenditure forecasts. The assigned value is the original budgeted cost for the item and is known as the planned value (PV) or budgeted cost of work scheduled (BCWS). A task that has not started is assigned the value zero and when it has been completed, it, and hence the project, is credited with the value of the task. The total value credited to a project at any point is known as the earned value (EV) or budgeted cost of work performed (BCWP) and this can be represented as a value or as a percentage of the PV. Where tasks have been started but are not yet complete, some consistent method of assigning an earned value must be applied. Common methods in soft-ware projects are:

- The 0/100 Technique: A task is assigned a value of zero until such time that it is completed when it is given a value of 100% of the budgeted value
- The 50/50 Technique: A task is assigned a value of 50% of its value as soon as it is started and then given a value of 100% once it is complete
- The Milestone Technique: A task is given a value based on the achievement of milestones that have been assigned values as part of the original budget plan

The 0/100 technique is the preferred technique. The 50/50 technique can give a false sense of security by over-valuing the reporting of activity starts. The milestone technique

NOTES

might be appropriate for activities with a long duration esti-mate but, in such cases, it is better to break that activity into a number of smaller ones.

The Baseline Budget

The first stage in setting up an earned value analysis is to create the baseline budget. The baseline budget is based on the project plan and shows the forecast growth in earned value through time. Earned value may be measured in monetary values but in the case of staff-intensive projects such as software development it is common to measure earned value in person-hours or work-days. For example, the baseline budget based on the schedule shown in Figure 6.11 is shown in Table 6.2 and the same is diagrammatically in Figure 6.12. It can be inferred that the baseline budget is based on workdays and is using the 0/100 technique for crediting earned value to the project.

NOTES

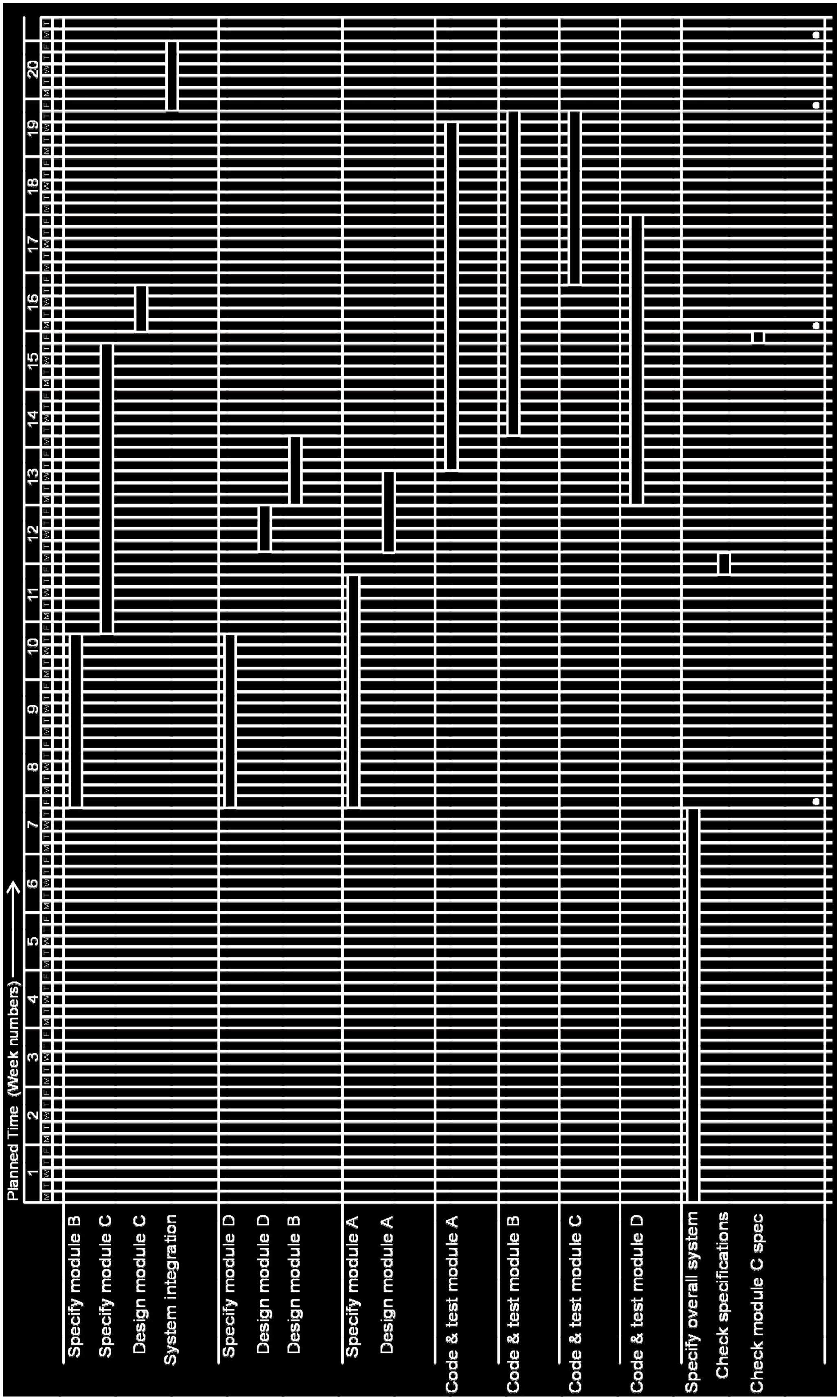

Figure 6.11: Work Schedule

NOTES

The project is not expected to be credited with any earned value until day 34, when the activity 'specify overall system' is to be completed. This activity was forecast to consume 34 person-days and it will therefore be credited with 34 person-days of earned value when it has been completed. The other steps in the baseline budget chart coincide with the scheduled completion dates of other activities.

Table 6.2: Baseline Budget Calculation

Task	Budgeted Workdays	Scheduled completion	Cumulative workdays	%Cumulative earned value
Specify overall system	34	34	34	14.35
Specify module B	15	49	64	27.00
Specify Module D	15	49		
Specify Module A	20	54	84	35.44
Check specification	2	56	86	36.28
Design module D	4	60	90	37.97
Design module A	7	63	97	40.93
Design module B	6	66	103	43.46
Check module C spec	1	70	104	43.88
Specific Module C	25	74	129	54.43
Design module C	4	79	133	56.12
Code & test module D	25	85	158	66.67
Code & test module A	30	93	188	79.32
Code & test module B	28	94	231	97.47
Code & test module C	15	94		
System integration	6	100	237	100.00

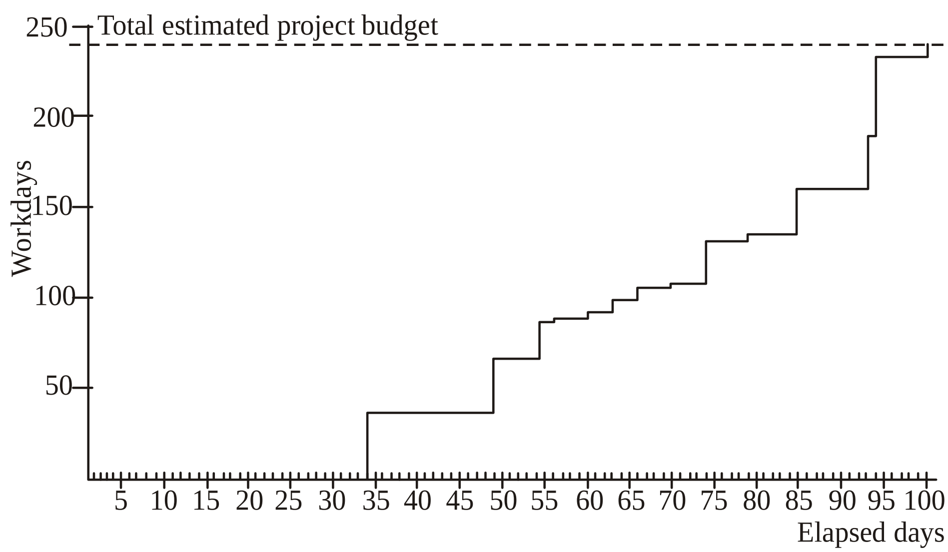

Figure 6.12: Diagrammatic Illustration of Baseline Budget

NOTES

Monitoring Earned Value

Once the baseline budget has been created the next task is to monitor earned value as the project progresses. This is done by monitoring the completion of tasks or activity starts and milestone achievements. Figure 6.13 illustrates earned value analysis at the start of week 12 of the project. It can be inferred that the earned value (EV) is clearly lagging behind the baseline budget, indicating that the project is behind schedule. By analyzing Figure 6.13 can it can easily be told what has gone wrong with her project and what the consequences might be.

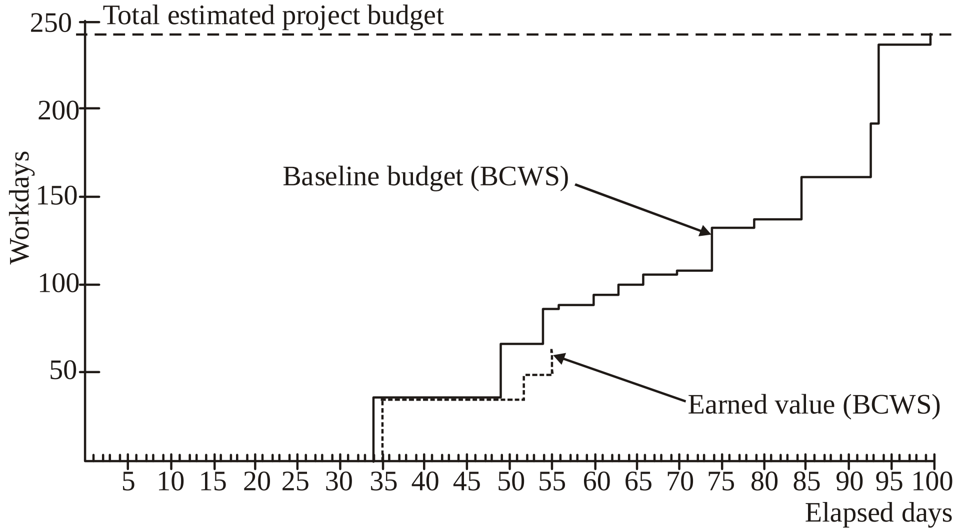

Figure 6.13: Diagrammatic Illustration of Earned Value analysis at Week 12

As well as recording EV, the actual cost of each task can be collected as actual cost (AC). This is also known as the actual cost of work performed (ACWP). The same is illustrated in Figure 6.14, which, in this case, records the values as percentages of the total budgeted cost.

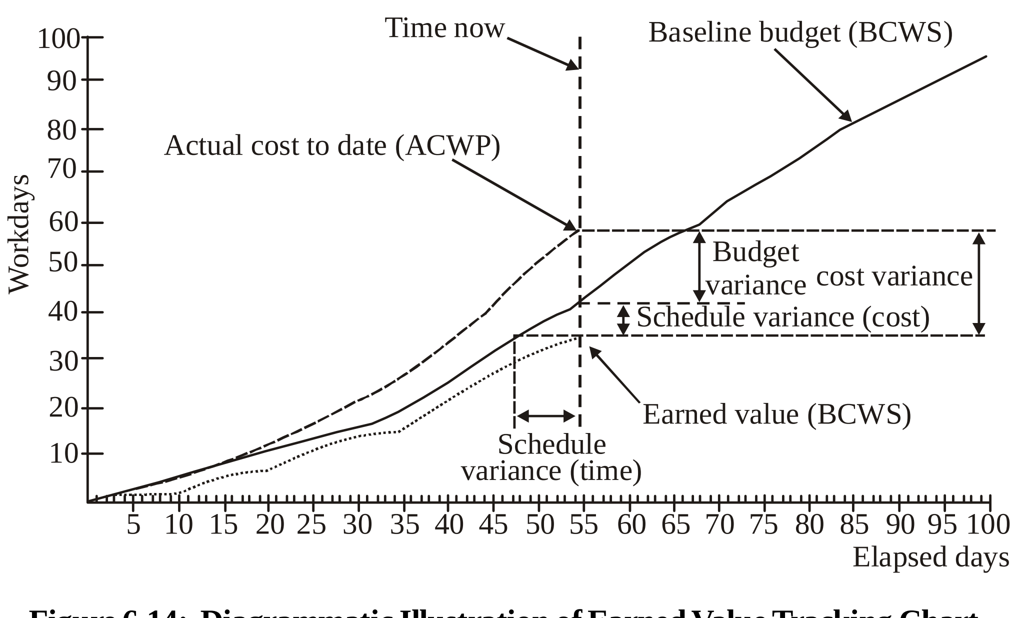

Figure 6.14: Diagrammatic Illustration of Earned Value Tracking Chart

NOTES

Figure 6.14 illustrates the following performance statistics, which can be shown directly or derived from the earned value chart.

Schedule Variance

The schedule variance is measured in cost terms as EV - PV and indicates the degree to which the value of completed work differs from that planned. Figure 6.14 also indicates the schedule variance in time which indicates the degree to which the project is behind schedule. A negative SV means the project is behind schedule.

Cost Variance

Cost variance as computed as EV- AC and indicates the difference between the budgeted cost and the actual cost of completed work. It is also an indicator of the accuracy of the original cost estimates. A negative CV means the project is over cost.

Performance Ratios

Two ratios are commonly tracked. They are listed below:

The cost performance index (SPK = EV / AC) the schedule performance index (SPI = EV / PV).

They can be thought of as a 'value -for-money' index. A value greater than one indicates that work is being completed better than planned, whereas value of less than one means that work is costing more than and / or preceding more slowly than planned.

CPI can be used to produce a revised cost estimate for the project or estimate at completion (EAC). EAC is calculated as BAC/CPI where budget at com-pletion BAC- is the current projected budget for the project. Figure 6.15 illustrates an earned value chart with revised forecasts.

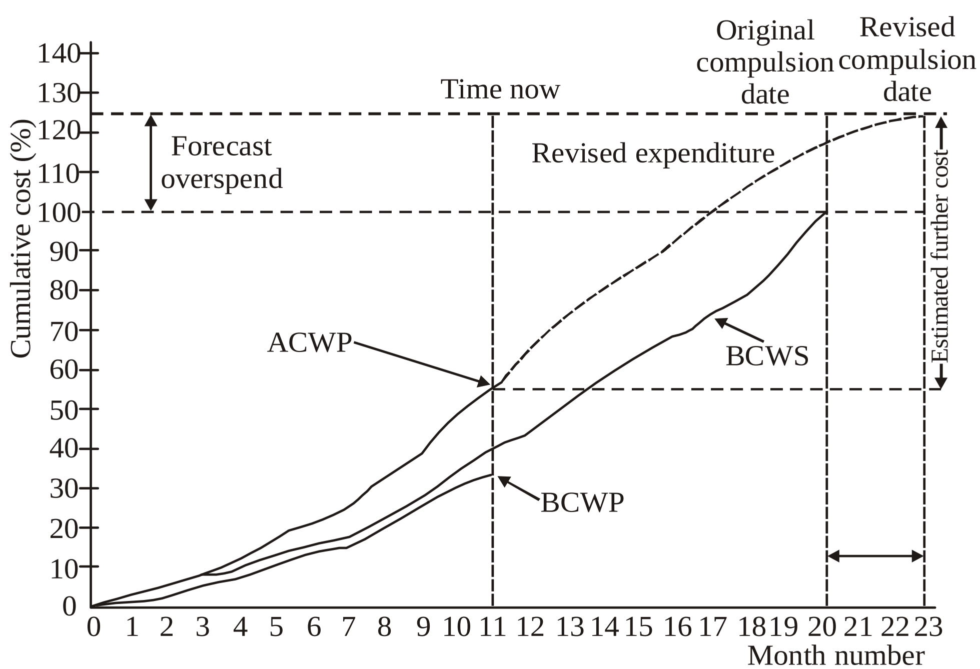

Figure 6.15: Diagrammatic Illustration of Earned Value Chart with Revised Forecasts

NOTES

Earned value analysis has not yet gained universal acceptance for use with soft-ware development projects are partially completed software project has no value at all.

6.6 PRIORITIZING MONITORING

The priority that must be applied in deciding the levels of monitoring is discussed below:

- Critical path activities: Any delay in an activity on the critical path will cause a delay in the completion date for the project. Critical path activities are therefore likely to (have a very high priority) for close monitoring.
- Activities with no free float*:* Free float is the amount of time an activity may be delayed without affecting any subsequent activity. A delay in any activity with no free float will delay at least some subsequent activities even though, if the delay is less than the total float, it might not delay the project completion date. These subsequent delays can have serious effects on our resource schedule as a delay in a subsequent activity could mean that the resources for that activity will become unavailable before that activity is completed because they are committed elsewhere.
- Activities with less than a specified float: If any activity has very little float it might use up this float before the regular activity monitoring brings the problem to the project manager's attention. It is common practice to monitor closely those activities with less than one week free float.
- High risk activities: A set of high-risk activities should have been identified as part of the initial risk profiling exercise. If we are using the PERT three-estimate approach we will designate as high risk those activities that have a high estimated duration variance. These activities will be given close attention because they are most likely to overrun or overspend.
- Activities using critical resources Activities can be critical because they are very expensive (as in the case of specialized contract programmers). Staff or other resources might be available only for a limited period, especially if they are con-trolled outside the project team. In any event, an activity that demands a critical resource requires a high level of monitoring.

6.7 GETTING THE PROJECT BACK TO TARGET

Most of the projects will, at one time or another, be subject to delays and unex-pected events. One of the tasks of the project manager is to recognize when this is happening and with the minimum delay and disruption to the project team, attempt to mitigate the effects of the problem. In most cases, the project manager tries to ensure that the scheduled project end date remains unaffected. This can be done by shortening remaining activity durations or shortening the overall duration of the remaining project.

This might not always be the most appropriate response to disruptions to a plan. There is little point in spending considerable sums in overtime payments in order to speed up a project if the customer is not overly concerned with the delivery date and there is no other valuable work for the team members once this project is completed.

NOTES

There are two main strategies to consider when drawing up plans to bring a project back on target:

- Shortening the critical path
- Altering the activity prece-dence requirements

Shorten the Critical Path

The overall duration of a project is determined by the current critical path, so speeding up non-critical path activities will not bring forward a project completion date.

Encouraging staff to 'work harder' might have some effect, although frequently a more positive form of action is required, such as increasing the resources available for some critical activity. Fact-finding, for example, might be speeded up by allo-cating an additional analyst to interviewing users. It is unlikely; however, that the coding of a small module would be shortened by allocating an additional pro-grammer indeed, it might be counterproductive because of the additional time needed organizing and allocating tasks and communicating.

Resource levels can be increased by making them available for longer. Thus, staff might be asked to work overtime for the duration of an activity and computing resources might be made available at times such as evenings and week-ends when they might otherwise be inaccessible.

Where these do not provide a sufficient solution, the project manager might consider allocating more efficient resources to activities on the critical path or swapping resources between critical and non-critical activities. This will be particularly appropriate with staff - an experienced programmer should be significantly more productive than a more junior member of the team.

By such means we can attempt to shorten the timescale for critical activities until such time as either we have brought the project back to schedule or further efforts prove unproductive or not cost-effective. It should be noted that short-ening a critical path often causes some other path, or paths, to become critical.

Reconsider the Precedence Requirements

The constraints by which some activities have to be deferred pending completion of others have to be considered. The original project network would most probably have been drawn up assuming 'ideal' conditions and 'normal' working practices. In order to avoid the project delivering late, it's better to inquire whether there are many unstarted activities which need to really wait for the completion of other activities. It might, in a particular organization, be 'normal' to complete system testing before commencing user training. In order to avoid late completion of a project it might, however, be considered acceptable to alter 'normal' practice and start training earlier.

One way to overcome precedence constraints is to subdivide an activity into a component that can start immediately and one that is still constrained as before. For example, a user handbook can be drawn up in a draft form from the system specification and then be revised later to take account of subsequent changes.

If the precedence requirements are altered in such a way, it is clearly important to be aware that quality might be compromised and to make a considered decision to compromise quality where needed. It is equally important to assess the degree to which changes in work practices increase risk. It is poss-ible, for example, to start coding a module before its design has been completed. This would increase the risk of having to redo some of the coding once the final design had been completed and thus delay the project even further.

6.8 CHANGE CONTROL

In real world requirements change because of changing circumstances or because the users get a clearer idea of what is really needed. Other, internal, changes will crop up. The project leader might find that there are inconsistencies in the program specifications that become apparent only when the programs are coded, and these would result in amendments to the specifications. Careful control of these changes is needed because an alteration in one document often implies changes to other documents and the system products based on that document.

Change Control Procedures

A simple change control procedure for operational systems might have the following steps:

- One or more users might perceive a need for a modification to a system and ask for a change request to be passed to the development staff.
- The user management considers the change request and, if they approve it, passes it to the development management.
- The development management delegates a member of staff to look at the request and to report on the practicality and cost of carrying out the change. They would, as part of this, assess the products that would be affected by the change.
- The development management report back to the user management on the findings and the user management decide whether, in view of the cost quoted, they wish to go ahead.
- One or more developers are authorized to take copies of the master products that are to be modified.
- The copies are modified. In the case of software components this would involve modifying the code and recompiling and testing it

NOTES

Summary:

- The execution of the plan must be monitored
- Activities that are too long need to be subdivided to make them more control-lable
- Progress should be measured through the delivery of project products
- Progress needs to be shown in a visually striking way, such as through ball charts, in order to communicate information effectively
- Costs need to be monitored as well as elapsed time
- Delayed projects can often be brought back on track by shortening activity times on the critical path or by relaxing some of the precedence constraints

Review Questions

1. Diagrammatically illustrate and discuss the project control cycle.
2. Tabulate and discuss the categories of reporting.
3. What is risk reporting? Discuss.
4. How progress can be visualized? Discuss with examples.
5. How should a schedule slippage be handled? Discuss.
6. What is change control? Discuss.

NOTES

CHAPTER 7

MANAGEMENT OF CONTRACT

Outsourcing is often viewed as involving the contracting out of a business function - commonly one previously performed in-house - to an external provider. In this sense, two organizations may enter into a contractual agreement involving an exchange of services and payments. A contract is a legally enforceable agreement between two or more parties with mutual obligations. The remedy at law for breach of contract is "damages" or monetary compensation.

7.1 INTRODUCTION

The buying in of both goods and services, rather than 'doing it yourself', is attractive when money is available. However, where an organization adopts this policy there are hazards arising from the considerable staff time and attention still needed to manage a contracted-out project successfully. Although the motivation for contracting out may have been to reduce management effort, it is, essential that customer organizations find time to make clear their exact requirements at and also to ensure that the goods and services that result are in fact what are actually required.

Also, potential suppliers are more likely to be flexible and accommodating before any contract has been signed than they will be afterwards, especially if the contract is for a fixed price. Much forethought and planning is needed with an acquisition project as with an internal development project.

7.2 THE ISO 12207 APPROACH TO THE ACQUISITION AND SUPPLY OF SOFTWARE

The ISO 12207 standard identifies five major processes relating to software.

- Acquisition
- Supply
- Operation
- Maintenance
- Development

Maintenance and operation are processes that belong to the post- implementation phase of a development project. A justification for the inclusion of these post-implementation activities is that sometimes a customer will contract a supplier both to develop a system and operate it on their behalf. Similarly, a software supplier could be responsible contractually both for the construction of a software-based system and its maintenance after installation.

NOTES

The acquisition process is the set of procedures that a customer for software should follow in order to obtain that software from an external source. The supply process is the opposite side to the acquisition process and is the set of procedures that the supplier should adopt in order to satisfy the acquirer's needs. The supplier may use an existing piece of software or may develop new software. In the latter case they would need to invoke the development process. The main activities that comprise the acquisition process is illustrated in Figure 7.1.

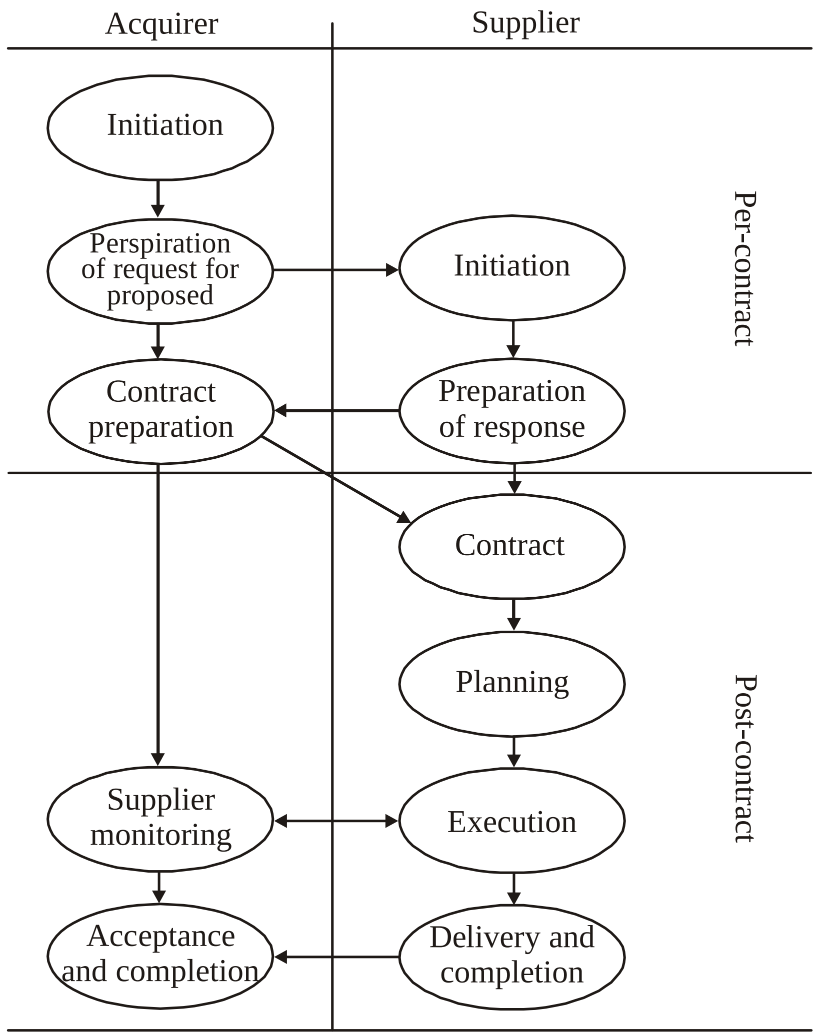

Figure 7.1: ISO12207 Acquisition and Supply Process

Initiation:

Figure 7.1 illustrates the major activities that comprise the acquisition process. The initiation activity starts with the acquirer describing the 'concept' that the acquirer wishes to make real, or the need which they wish to satisfy. The requirements of the system that it is felt would fulfill that need should then be defined by the acquirer organization. An external expert could be employed for this purpose, but the responsibility for the accuracy of these requirements would remain with the acquirer.

NOTES

ISO 12207 distinguishes between system requirements and software requirements. System requirements are broader and relate not just to the software but to the platforms upon which it would run and to changes in the organizational environment needed for the successful operation of the new system.

Software requirements, on the other hand, relate to the distinctly software components within the delivered system and will have to be extracted from the broader system requirements. Once this has been done, a decision needs to be taken about the best way to acquire the software, for example whether to make or buy.

Having made this decision, the acquirer is now in a position to prepare the acquisition plan, detailing the steps needed to take to acquire the software. The precise scope of the work to be undertaken externally needs to be decided. For example, would maintenance and support for the new software be dealt with in- house or by the supplier? Other questions relate to potential risks and the criteria for final system acceptance, including methods by which compliance with requirements is to be evaluated.

Request for proposal:

The request for proposal (RFP) document has to be prepared including the sections on the topics listed in Table 7.1.

Table 7.1: Topics in a Request for Proposal Document

Request for Proposal – Contents
• System Requirements
• Scope Statement
• Instruction for Bidders
• List of Software Products
• Control of Subcontracts
• Technical Constraints (Example: The Target environment)

ISO 12207 makes provision for the tailoring of the standard set of activities expected of the supplier to suit the particular situation. Some activities could be deleted, while the contract for a particular project could add activities if these are needed.

Contract Preparation and Update

Before having a contract, a supplier with whom to have the contract must be identified. The criteria for selecting the supplier and the method of judging the degree of compliance by potential suppliers with the criteria have to be set down. Based on these, the preferred supplier is to be selected. The form of the contract between the supplier and acquirer can

NOTES

now be negotiated and agreed. This may involve some agreed adjustments to the processes that ISO 12207 assumes would be executed during the project.

Monitor Supplier

This will be done using joint reviews, audit, verification and validation.

Accept Completed Contract

When the supplier finally delivers the product, the acquirer will conduct acceptance tests and if the specified acceptance criteria are satisfied, the completed software can be signed off as completed.

7.3 THE SUPPLY PROCESS

This process mirrors the acquisition process, but documents the activities that a supplier would need to undertake in response to the request of a supplier. There may be occasions where the sequence of activities is not that shown and yet other occasions when some of the activities will have to iterated a number of times.

Initiation

The process is started when a potential supplier receives a request for a proposal from an acquirer and the supplier decides to bid for the work.

Preparation of a Response

The supplier, after consulting people with various types of expert knowledge, now prepares a response. This should include any proposals about how ISO 12207 is to be tailored for the project in view.

Contract

If all goes well, the supplier's proposal makes the right impression and leads to acceptance by the acquirer. The details of the contract are then negotiated and signed.

Planning

The supplier can now draw up a detailed plan of how the work is to be done. The starting point for this will be the requirements as laid down in the request for proposal. The life-cycle approach to be applied by the supplier is to be included as this will to some extent dictate the points at which consultation between the supplier and the acquirer is to take place during the development process. The acquirer may have considered, as part of the acquisition process, the options of make versus buy and also whether to use in-house or external sources. ISO12207 now makes provision for the supplier to make a similar choice, and having done this, to develop a plan accordingly.

Execution and Control

The plan can now be executed. If the contract is for the development of software, then at this point the development process is invoked. Depending on whether the contract is for systems development, operation or maintenance, the IS0 12207 processes stipulated in each case should be undertaken.

During execution of the plan, the standard expects the supplier to monitor and control progress and product quality, and to have a mechanism for identifying and recording the problems that occur and to have a mechanism for their analysis and resolution. The supplier will also be responsible for passing on requirements that accurately reflect those of the acquirer to any subcontractors and for ensuring the compliance of subcontractors with those requirements. The supplier also needs to cooperate fully with any independent verification and validation processes that were laid down in the contract.

Review and Evaluation

The provisions here are mainly to ensure that the supplier allows the acquirer access to the information needed to review the progress of the project, although the precise extent to which the acquirer has a reviewing role and access to supplier documentation is to be specified in the contract.

Delivery and Completion

Attention may have to be given in any management plans to the way in which this is to be done and to how any required post-delivery support is to be provided.

7.4 TYPES OF CONTRACT

The external resources required could be in the form of services. A simple example of this could be using temporary staff on short-term contracts to carry out some project tasks. A more far-reaching use of external services would be for the contractor not only to supply the new system but to also operate it on the customer's behalf.

On the other hand, the contract could be, placed for the supply of a completed software package. This could be:

- A bespoke system, that is, a system that is created from scratch specifically for one customer
- An off-the-shelf package which you buy 'as is' — this is sometimes referred to as shrink- wrapped software
- A customized off-the-shelf (COTS) software — this is a basic core system which is, modified to meet the needs of a particular client

Another way of classifying contracts is by the way that the payment to suppliers is calculated. We will look at:

NOTES

- Fixed Price Contracts
- Time and Materials Contracts
- Fixed Price Per Delivered Unit Contracts

Fixed Price Contracts

A price is fixed when the contract is signed. The customer knows that, if there are no changes in the contract terms, this is the price they will have to pay on the completion of the work. In order for this to be effective, the customer's requirement has to be known and fixed at the outset. In other words, when the contract is to construct a software system, the detailed requirements analysis must already have been carried out. Once the development is under way the customer organization will not normally be able to change their requirements without renegotiating the price of the contract.

Advantages:

- Known Customer Expenditure: If the requirements have been defined adequately and there are few subsequent changes, then the customer will have a known outlay.
- Supplier Motivation: The supplier has a motivation to manage the delivery of the system in a cost-effective manner.

Disadvantages:

- Higher Prices to allow for Contingency: The supplier absorbs the risk for any errors in the original estimate of product size. To reduce the impact of this risk, the supplier will add a margin when calculating the price to be quoted in a tender.
- Difficulties in Modifying Requirements: The need to change the scope of the requirements may become apparent as the system is developed. This may cause friction between the supplier and customer.
- Upward Pressure on the Cost of Changes: When competing against other potential suppliers, the supplier will try to quote as low a price as possible. Once the contract is signed, if further requirements are put forward, the supplier is in a strong position to demand a high price for these changes.
- Threat to System Quality: The need to meet a fixed price could mean that the quality of the software suffers.

Time and Materials Contracts

With this type of contract, the customer is charged at a fixed rate per unit of effort, for example per staff-hour. At the start of the project, the supplier may provide an estimate of the overall cost based on their current understanding of the customer's requirements, but this is not the basis for the final payment.

NOTES

Advantages:

- Ease of Changing Requirements: Changes to requirements are dealt with easily. Where a project has a research orientation and the direction of the project may change as options are explored, then this may be an appropriate method of calculating payment.
- Lack of Price Pressure: The lack of price pressure may allow better quality software to be produced.

Disadvantages:

- Customer Liability: The customer absorbs all the risks associated with poorly defined or changing requirements.
- Lack of Incentives for Supplier: The supplier has no incentive to work in a cost-effective manner or to control the scope of the system to be delivered.

Fixed Price per Unit Delivered Contracts

This is often associated with function point (FP) counting. The size of the system to be delivered is calculated or estimated at the outset of the project. The size of the system to be delivered may be estimated in lines of code, but FPs can be more easily and reliably derived from requirements documents.

A price per unit is also quoted. The final price is then the unit price multiplied by the number of units.

Advantages:

- Customer Understanding: The customer can see how the price is calculated and how it will vary with changed requirements.
- Comparability Pricing: Schedules can be compared.
- Emerging Functionality: The supplier does not bear the risk of increasing functionality.
- Supplier Efficiency: The supplier still has an incentive to deliver the required functionality in a cost-effective manner
- Life-cycle Range: The requirements do not have to be definitively specified at the outset. Thus the development contract can cover both the analysis and design stages of the project.

Disadvantages:

- Difficulties with Software Size Measurement: Lines of code can easily he inflated by adopting a verbose coding style. With FPs, there may be disagreements about what the FP count should really be: in some cases, FP counting rules may be seen as unfairly favoring either the supplier or customer. Users, in particular, will almost certainly not be familiar with the concept of FPs and special training may be needed for them. The solution to these problems may be to employ an independent FP counter.

NOTES

- Changing Requirements: Some requested changes may affect existing transactions drastically but not add to the overall FP count. A decision has to be taken about how to deal with these changes. A change made late in the development cycle will almost certainly require more effort to implement than one made earlier.

Another way of categorizing contracts, at least initially, is according to the approach that is used in contractor selection is as follows:

- Open
- Restricted
- Negotiated

Open Tendering Process

In this case, any supplier can bid to supply the goods and services. Furthermore, all bids that are compliant with the original conditions laid down in the invitation to tender must be considered and evaluated in the same way. With a major project with many bids this evaluation process can be time consuming and expensive.

Restricted Tendering Processes

In this case, there are bids only from suppliers who have been invited by the customer. Unlike the open tendering process, the customer may at any point reduce the number of potential suppliers being considered. This is usually the best approach to adopt. However, it is not without risk: where the resulting contract is at a fixed price, the customer assumes responsibility for the correctness and completeness of the requirements specified to the prospective suppliers. Defects in this requirements documentation may allow a successful bidder to subsequently claim for additional payments.

Negotiated Procedure

There may, however, be some good reasons why the restricted tendering process may not be the most suitable some particular sets of circumstances. Say, for example, that there is a fire that destroys part of an office with equipments. The key concern here may be to get replacement equipment up and running as quickly as possible and there may simply not be the time to embark on a lengthy tendering process. Another example might be where a new software application had been successfully built and implemented by a supplier, but the customer realizes that they would like to have some extensions to the system. As the original supplier has staff completely familiar with the existing system, it may once again be inconvenient to approach other potential suppliers via a full tendering process. In these cases, an approach to a single supplier may be justified. However, it takes little imagination to realize that approaching a single supplier could open the customer up to charges of favoritism and should only be done where there is a clear justification.

Stages in Contract Placement

The steps involved in awarding a contract are discussed below:

Requirements Analysis

Before potential suppliers can be approached, a clear set of requirements is needed. The first is that it is easy for this step to be skimped where the user management has many day-to-day pressures and not much time to think about future developments. In this situation, it can be useful to bring in an external consultant to draw up a requirements document. Even here, users and their managers need to look carefully at the resulting requirements document to ensure that it accurately reflects their needs.

The requirements document might typically have sections. The same is tabulated in Table 7.2.

Table 7.2 Main Sections in a Requirements Document

S. No.	Section
1	Introduction
2	A description of any exiting systems and the current environment
3	The customers future strategy or plans
4	System requirements • Mandatory • Desirable
5	Deadlines
6	Additional information required from potential suppliers

The requirements define carefully the functions that need to be carried out by the new application and all the necessary inputs and outputs for these functions. The requirements should also state any standards with which there should be compliance, and the existing systems with which the new system needs to be compatible. As well as these functional requirements, there will also need to be operational and quality requirements concerning such matters as the required response times, reliability, usability and maintainability of the new system.

The requirements document should state needs as accurately as possible and should avoid technical specifications of possible solutions.

NOTES

Each requirement needs to be identified as being either mandatory or desirable:

- Mandatory: If a proposal does not meet this requirement then the proposal is to be immediately rejected. No further evaluation would be required.
- Desirable: A proposal may be deficient in this respect, but other features of the proposal could compensate for this.

Evaluation Plan

Having drawn up a list of requirements, we now need to draw up a plan of how the proposals which arc submitted are to be evaluated. The situation will be slightly different if the contract is for a system that is to be specially written as opposed to an off-the-shelf package. In the latter case, it is the system itself that is being evaluated while in the former situation it is a proposal for a system. First, ways of checking that all the mandatory requirements have been met need to be identified. The next consideration is how the desirable requirements can be evaluated.

The problem here is weighing the value of one quality against another. The ISO 9126 standard, which is on software quality, may be used to decide that one system has more of some quality than another, but if there is a difference in price between the two, we need to be able to estimate if the increase in quality is worth the additional price. Hence 'value for money' is often the key criterion.

It needs to be stressed that the costs to be taken into account are those for the whole of the lifetime of the proposed system, not just the costs of acquiring the system. Also, where the relationship with the supplier is likely to be ongoing, the supplier organization needs to be assessed as well as its products.

Invitation to Tender

Having produced the requirements and the evaluation plan, it is now possible to issue the invitation to tender to prospective suppliers. Essentially, this will be the requirement document with a supporting letter which will have additional information about how responses to the invitation are to be lodged. A deadline will be specified and it is hoped that by then a number of proposals with price quotations will have been received.

In English law, for a contract to exist there must be an offer on one side which must be accepted by the other side. The invitation to tender is not an offer itself but an invitation for prospective suppliers to make an offer.

The requirements that have been laid down could be satisfied in a number of different ways. The customer not only needs to know a potential supplier's price but also the way in which they intend to satisfy the requirements. This will be particularly important where the contract is to build a new system from scratch.

In certain cases, it would be enough to have some post- tender clarification and negotiation to resolve issues in the supplier's proposal. With more complex projects a more elaborate approach may be needed. One way of getting the detail of the suppliers' proposals elaborated is to have a two-stage tendering process.

In the first stage, technical proposals are requested from potential suppliers who do not necessarily quote any prices. Some of these proposals can be rejected as not being able to meet the mandatory requirements. The remaining ones could be discussed with representatives of the suppliers in order to clarify and validate the technical proposals. The suppliers might be asked to demonstrate certain aspects of their proposals. Where shortcomings in the proposal are detected, the supplier could be given the opportunity to remedy these.

The result of these discussions could be a Memorandum of Agreement (M0A) with each prospective supplier. This is an acceptance by the customer that the proposed solution, which might have been modified during discussions offered by the supplier satisfactorily, meets the customer's requirement.

In the second stage tenders are invited from all the suppliers who have signed individual Memoranda of Agreement. The tender would incorporate the MOA and would be concerned with the financial terms of a potential contract.

If a design has to be produced as part of the proposal made by a supplier in response to an invitation to tender then a difficulty would be that the supplier would have to do a considerable amount of detailed design work with only a limited prospect of being paid for it. One way of reducing this burden is for the customer to choose a small number of likely candidates who will be paid a fee to produce design proposals. These can then be compared and the final contract for construction awarded to the most attractive proposal. The ISO 12207 has a rather different approach. Once a contract for software construction is signed, the supplier develops a design which then has to be agreed by the customer.

Evaluation of Proposals

An evaluation plan describing how each proposal will be checked against the selection criteria. This reduces risks of requirements being missed and ensures that all proposals are treated consistently. It would be unfair to favor a proposal because of the presence of a feature that was not requested in the original requirement.

It will be recalled that an application could be either bespoke, off the shelf, or customized. In the case of off-the-shelf packages it would be the software itself that would be evaluated and it might be possible to combine some of the evaluation with acceptance testing. With bespoke development it would be a proposal that is evaluated, while COTS (Commercial off- the-shelf software) may involve elements of both. Thus different planned approaches would be needed in each case.

NOTES

The process of evaluation may include:

- Scrutiny of the proposal documents
- Interviewing suppliers' representatives
- Demonstrations
- Site visits
- Practical tests

The proposal documents provided by the suppliers can be scrutinized to see if they contain features satisfying all the original requirements. Clarification might be sought over certain points. Any factual statements made by a supplier imply a legal commitment on their part if they influence the customer to offer the contract to that supplier. It is therefore important to get a written, agreed, record of these clarifications. The customer might take the initiative here by keeping notes of meetings and then writing afterwards to the suppliers to get them to confirm the accuracy of the notes. A supplier could, in the final contract document, attempt to exclude any commitment to any representations that may have been made in precontract negotiations the terms of the contract need to be scrutinized in this respect. Where the delivered product is to be based on an existing product there could be a demonstration. The customer organization should have their own schedule of what needs to be demonstrated, ensuring that all the important features are seen in operation.

With off-the-shelf software, the customer could actually try out the package. For example, a demonstration version could be made available which closes itself down after 40 days. Once again a test plan is needed to ensure that all the important features are evaluated in a complete and consistent manner. Once a particular package emerges as the most likely candidate, it needs to be carefully investigated to see if there are any previously unforeseen factors that might invalidate this choice.

A frequent problem is that while an existing application works well on one platform with a certain level of transactions, it does not work satisfactorily with the customer's Information and Communication Technologies applications configuration or level of throughput. Demonstrations might not reveal this problem. Visits to operational sites already using the system could be more informative.

The following aspects of a proposal have to be evaluated:

- The usability of an existing software application
- The usability of a software application which is yet to be designed and constructed
- The maintenance costs of hardware to be supplied
- The time taken to respond to requests for software support
- Training

A decision will be made to award the contract to one of the suppliers. One reason for using a structured and, as far as possible, objective approach to evaluation is to be able to demonstrate that the decision has been made impartially and on merit. In most large organizations, placing a contract involves the participation of a second party within the organization, such as a contracts department, who can check that the correct procedures have been carried out. Also, the final legal format of a contract will almost certainly require some legal expertise.

Both the successful candidate and the unsuccessful candidates should be notified of the decision. It makes dealing with unsuccessful bidders easier if they can be given clear and objective reasons why their proposals did not find favor.

7.5 TYPICAL TERMS OF A CONTRACT

The major areas of concern are discussed below:

Definitions

The terminology used in the contract document may need to be defined, e.g. who is meant by the words 'client' and 'supplier'.

Form of Agreement

For example consider the following:

- Is it a contact of sale, a lease, or a license?
- Also, can the subject of the contract, such as a license to use a software package, be transferred to another party?

Goods and Services to be Supplied

Equipment and Software to be Supplied: This should include an actual list of the individual pieces of equipment to be delivered, complete with the specific model numbers.

Services to be Provided:

This would cover the following:

- Training
- Documentation
- Installation
- Conversion of existing files
- Maintenance agreements
- Transitional insurance arrangements

NOTES

Ownership of the Software

Two key issues have to be addressed here. Firstly, whether the customer can sell the software to others and, secondly, whether the supplier can sell the software to others. Where an off-the-shelf package is concerned, the supplier often simply grants a license to use the software. Where the software is being written specially for a customer then that customer may wish to ensure that they have exclusive use of the software and may object the supplier to sell the software to their competitors. They could do this by acquiring the copyright to the software outright or by specifying that they should have exclusive use of the software. This would need to be written into the contract. Where a core system has been customized by a supplier then there is less scope for the customer to insist on exclusive use.

Where software is written by an employee as part of their normal job, it is assumed that the copyright belongs to the employer. Where the customer organization have contracted an external supplier to write software for them, the contract needs to make clear who is going to retain the copyright it cannot, in this case, be automatically assumed it is the customer. The customer may have decided to take over responsibility for maintenance and further development once the software is delivered and in this case will need access to the source code. In other cases, where the customer does not have an adequate in-house maintenance function, the supplier may retain the source code, and the customer may have to approach the supplier for any further changes.

Environment

Where software is being supplied, the compatibility of the software with the existing hardware and operating system platforms would need to be confirmed.

Customer Commitments

Even when work is carried out by external contractors, a development project still needs the participation of the customer.

Acceptance Procedures

Good practice would be to accept a delivered system only after it has undergone user acceptance tests. This part of the contract would specify such details as the time that the customer will have to conduct the tests, deliverables upon which the acceptance tests depend and the procedure for signing off the testing as completed.

Standards

This covers the standards with which the goods and services should comply. For example, a customer could require the supplier to conform to the ISO 12207 standard relating to the software life cycle and its documentation.

NOTES

Project and Quality Management

The arrangements for the management of the project must be agreed. Among these would be frequency and nature of progress meetings and the progress information to be supplied to the customer. The contract could require that appropriate standards are followed. The ISO 12207 standard provides for the customer to have access to quality documentation generated internally by the supplier, so that the customer can ensure that there is adherence to standards.

Timetable

This provides a schedule of when the key parts of the project should be completed. This timetable will commit both the supplier and the customer. For example, the supplier may only be able to install the software on the agreed date if the customer makes the hardware platform available to them at that point.

Price and payment method

The schedule of payments must be agreed. The supplier's desire to be able to meet costs as they are incurred needs to be balanced by the customer's requirement to ensure that goods and services are satisfactory before parting with their money.

Miscellaneous legal requirements

This is the legal small print. A contract may have to have clauses which deal with such matters as the definition of terms used in the contract, the legal jurisdiction that will apply to the contract, what conditions would apply to the subcontracting of the work, liability for damage to third parties, and liquidated damages. Liquidated damages are estimates of the financial losses that the customer would suffer if the supplier were to fall short of their obligations. In English law, the penalties laid down in penalty clauses must reflect the actual losses the customer would suffer and cannot be unrealistic and merely punitive. Even this limitation may not be enough in some cases as far as the supplier is concerned. As computer systems assume increasingly critical roles in many organizations and in safety-critical systems can even be life-threatening in the case of malfunction, the possible consequential damage could be astronomical. Suppliers will not unnaturally try to limit this kind of liability. Suppliers will, in the case of major contracts, take out insurance to cover such liabilities.

An alternative is to agree that disputes be settled by arbitration. This requires that any dispute is referred to an expert third party whose decision as to the facts of the case is binding. Even this procedure might not be quick and inexpensive and another option is alternative dispute resolution where a third party acts as a mediator who has only an advisory capacity and attempts to broker an agreement between the two sides.

NOTES

7.6 CONTRACT MANAGEMENT

The communications between the supplier and customer while the contracted work is being carried out must be considered. It would probably suit all concerned if the contractor could be left to get on with the work undisturbed. However, at certain decision points, the customer might wish to examine work already done and make decisions about the future direction of the project. The project could require representatives of the supplier and customer to interact at many points in the development cycle, for example, users may need to be available to provide information needed for the design of interfaces.

This interaction, or other external factors, could lead to changes being needed which effectively vary the terms of the contract and so a careful change control procedure is needed. When the contract is being negotiated, certain key points in the project may be identified where customer approval is needed before the project can proceed. For example, a project to develop a large system could be divided into increments. For each increment there could be an interface design phase, and the customer might need to approve the designs before the increment is built. There could also be decision points between increments.

For each decision point, the deliverables to be presented by the suppliers, the decisions to be made by the customer and their outputs all need to be defined. These decision points have added significance if payments to the supplier are based on them. Not only the supplier but also the customer has responsibilities with respect to these decision points. For example, the supplier should not be unnecessarily delayed while awaiting customer approval of interim deliverables. When work is contracted out there will be concern about the quality of that work. To resolve the above, joint reviews of project processes and products to be agreed when the contract is negotiated.

7.7 ACCEPTANCE

When the work has been completed, the customer needs to take action to carry out acceptance testing. The contract may put a time limit on how long acceptance testing can take, so the customer must be organized to carry out this testing before the time limit for requesting corrections expires.

The customer could find that all their problem reports are being dealt with by relatively junior members of the supplier's staff, who may not be familiar with all aspects of the delivered system.Part or all of the payment to the supplier should depend on this acceptance testing. Sometimes part of the final payment will be retained for a period of operational running and is eventually paid over if the levels of reliability are as contracted for. There may also be a period of warranty during which the supplier should fix any errors found for no charge.

NOTES

Summary

The thriving contracting out of work requires significant amounts of management time. It is easier to gain concessions from a supplier before a contract is signed rather than afterwards. Alternative proposals need to be evaluated as far as possible by comparing costs over the whole lifetime of the system rather than just the acquisition costs. A contract will place obligations on the customer as well as the supplier. A contract negotiation should include reaching agreement on the management of the supplier—customer relationship during the execution of the project.

Questions

1. List the five major processes relating to software identified by ISO 12207.
2. List and discuss the different types of contract.
3. Tabulate and discuss the main sections in a requirements document.
4. Explain the tendering process.

NOTES

NOTES

UNIT V

CHAPTER 8

MANAGING PEOPLE AND ORGANIZING TEAMS

There are three main concerns: staff selection, staff development and staff motivation. A project leader can encourage effective group working and decision making while giving purposeful leadership where needed. As part of the concern for the well-being of team members some attention must be paid to stress and other issues of health and safety.

People with practical experience of projects invariably identify the handling of people as an important aspect of project management. The discipline of organizational behavior has evolved theories that try to explain people's behavior. These theories are often structured if A is the situation then B is likely to result. Attempts are made to observe behavior where variables for A and B are measured and a statistical relationship between the two variables sought. Unlike physical science it is rarely, if ever, that it can be said that B must always follow A.

8.1 ORGANIZATIONAL BEHAVIOR

Organizational Behavior (OB) is the study and application of knowledge about how people, individuals, and groups act in organizations. It does this by taking a system approach. That is, it interprets people-organization relationships in terms of the whole person, whole group, whole organization, and whole social system. Its purpose is to build better relationships by achieving human objectives, organizational objectives, and social objectives.

The organization's base rests on management's philosophy, values, vision and goals. This in turn drives the organizational culture which is composed of the formal organization, informal organization, and the social environment. The culture determines the type of leadership, communication, and group dynamics within the organization. The workers perceive this as the quality of work life which directs their degree of motivation. The final outcomes are performance, individual satisfaction, and personal growth and development. All these elements combine to build the model or framework that the organization operates from.

NOTES

Taylor attempted to analyse the most productive way of doing manual tasks. The workers were then trained to do the work in this way. Taylor had three basic objectives:

- Select the best people for the job;
- Instruct them in the best methods;
- Give incentives in the form of higher wages to the best workers.

8.2 SELECTING THE RIGHT PERSON FOR THE JOB

Taylor stressed the need for the right person for the job. Many factors, such as the use of software tools and methodologies, affect programming productivity. However, one of the biggest differences in software development performance is between individuals. The quality of staff has an influence on productivity.

Information systems (IS) professionals seemed to have much weaker 'social needs' than people in other professions. If asked, most programmers probably say they prefer to work alone where they wouldn't be disturbed by other people. Many who are attracted to writing soft-ware, and are good at it, but do not make good managers later in their careers.

The Recruitment Process

It must be stressed that often project leaders have little choice about the people who will make up their team - they have to make do with the 'materials that are to hand'. Recruitment is often an organizational responsibility: the person recruited might, over a period of time, work in many different parts of the organization. Eligible and Suitable candidates must be distinguished. Eligible candidates have curriculum vitae (CV) which shows, for example, the 'right' number of years in some previous post and the 'right' paper qualifications. Suitable candidates can actually do the job well.

A mistake is to select an eligible candidate who is not in fact suitable. Suitable candidates who are not officially eligible can, on the other hand, be ideal candidates as once in post they are more likely to remain loyal.

We should try to assess actual skills rather than past experience and provide training to make good minor gaps in expertise. It seems to us to show that policies that avoid discrimination on the grounds of race, gender, age or irrelevant disabilities can be not just socially responsible but also a shrewd recruitment policy.

A general approach might be the following:

- Create a job specification: Advice is often needed as there could be legal implications in an official document. However, formally or informally, the requirements of the job, including the types of task to be carried out, should be documented and agreed.
- Create a job holder profile: The job specification is used to construct a profile of the person needed to carry out the job. The qualities, qualifications, education and experience required would be listed.

NOTES

- Obtain applicants: Typically, an advertisement would be placed, either within the organization or outside in the trade or local press. The job holder profile would be examined carefully to identify the medium most likely to reach the largest number of potential applicants at least cost. For example, if a specialist is needed it would make sense to advertise in the' relevant specialist journal. The other principle is to give enough information in the advertisement to allow an element of self elimination. By giving the salary, location, job scope and any essential qualifications, the applicants will be limited to the more realistic candidates.
- Examine curriculum vitas: These should be read carefully and compared to the job holder profile - nothing is more annoying for all concerned than when people have CVs which indicate clearly they are not eligible for the job and yet are called for interview.
- Interviews: Selection techniques include aptitude tests, personality tests and the examination of samples of previous work. Any method must test specific qualities detailed in the job holder profile. Interviews are the most commonly used method. It is better if there is more than one interview session with an applicant and within each session there should not be more than two interviewers as a greater number reduces the possibility of followup questions and discussion. Some formal scoring system for the qualities being judged should be devised and interviewers should then decide scores individually which are then compared. An interview might be of a technical nature where the practical expertise of the candidate is assessed, or of a more general nature. In the latter case, a major part of the interview could be evaluating and confirming statements in the CV - for example, time gaps in the education and employment history would be investigated, and the precise nature of previous jobs would need to be explored.
- Other Procedures: References will need to be taken up where necessary and a medical examination might be needed.

Consider a new analyst/programmer is to be recruited to work. When new members of the team are recruited, the team leader will need to plan their induction into the team very carefully. Where a project is already well under way, this might not be easy. However, the effort should be made - it should pay off as the new recruit will become a fully effective member of the team more quickly. The team leader should be aware of the need to assess continually the training needs of their team members.

8.3 MOTIVATION:

Motivating people to work will yield better performance. Some models of motivation are discussed below:

The Taylorist Model:

Taylor's viewpoint is reflected in the use of piece-rates in manufacturing industries and sales bonuses amongst sales forces. Piece-rates can cause difficulties if a new system will change work practices. If new technology improves productivity, adjusting piece-

NOTES

rates to reflect this will be a sensitive issue. Usually, radical changes in work practices have to be preceded by a move from piece-rates to day-rates. The tendency towards dispersed or 'virtual projects' where staff work on their own premises at some distance from the sponsoring organization's site has seen a movement away from payment based on time worked.

Even where work practices are stable and output can be easily related to reward, people paid by the amount they produce will not automatically maximize their output in order to maximize their income. The amount of output will often be constrained by 'group norms', informal, even unspoken, agreements among colleagues about the amount to be produced.

Rewards based on piece-rates need to relate directly to work produced. Where a computer application is being developed, it is difficult to isolate and quantify work done by an individual, as system development and support is usually a team effort.

In this kind of environment, a reward system that makes excessive distinctions between co-workers could damage morale and productivity. Organizations some-times get around this problem by giving bonuses to project team members at the end of a successful project, especially if staffs have 'volunteered' considerable unpaid overtime to get the project completed.

Maslow's Hierarchy of Needs

The motivation of individuals varies. Money is a strong motivator when you are broke. However, as the basic need for cash is satisfied, other motivators are likely to emerge. Abraham Maslow, an American psychologist, suggested a hierarchy of needs. As a lower level of needs is satisfied then gradually a higher level of needs emerges. If these are then satisfied then yet another level will emerge. Basic needs include food, shelter and personal safety. The highest-level need, according to Maslow, is the need for 'self-actualization', the feeling that you are completely fulfilling your potential.

In practice, people are likely to be motivated by different things at different stages of their life. For example, salary increases, while always welcome, probably have less impact on the more mature employee who is already relatively well paid, than on a lowly paid trainee. Older team-members might place more value on qualities of the job, such as being given autonomy, which show respect for their judgment and sense of responsibility.

Some individual differences in motivation relate simply to personality differ-ences. Some staff have 'growth needs' - they are interested in their work and want to develop their work roles - while others simply see the job as a way of earning a living.

NOTES

Herzberg's Two-Factor Theory:

Some things about a job can make you dissatisfied. If the causes of this dissatisfac-tion are removed, this does not necessarily make the job more exciting. Research into job satisfaction by Herzberg and his associates found two sets of factors about a job.

- Hygiene or Maintenance Factors, which can make you dissatisfied if they are, not right, for example the level of pay or the working conditions
- Motivators, which make you, feel that the job is worthwhile, like a sense of achievement or the challenge of the work itself

The Expectancy Theory of Motivation

It identifies three influences on motivation:

- Expectancy: The belief that working harder will lead to a better performance
- Instrumentality: The belief that better performance will be rewarded; Instrumentality
- Perceived Value of the resulting reward

Motivation will be high when all three factors are high. A zero level for anyone of the factors can remove motivation.

8.4 THE OLDHAM-HACKMAN CHARACTERISTICS MODEL

Managers should group together the elements of tasks to be carried out so that they form meaningful and satisfying assignments. Oldham and Hackman suggest that the satisfaction that a job gives is based on five factors. The first three factors make the job 'meaningful' to the person who is doing it:

- Skill Variety: The number of different skills that the job holder has the opportunity to exercise
- Task Identity: The degree to which your work and its results are identifiable as belonging to you
- Task Significance: The degree to which your job has an influence on others
- The other two factors are:
- Autonomy: The discretion you have about the way that you do the job
- Feedback: The information you get back about the results of your work
- Oldham and Hackman also noted that both the job holders' personal growth needs and their working environment influenced their perception of the job. Some writers have pointed out that if people are happy with their work for other reasons, they are likely to rate it higher on the Oldham-Hackman dimensions anyway. Thus it might be that cause and effect are reversed. In practical terms, activities should be designed so that, where possible, staff follow the progress of a particular product and feel personally associated with it.

NOTES

Motivation:

To improve motivation the manager might therefore do the following:

- Set Specific Goals: These goals need to be demanding and yet acceptable to staff. Involving staff in the setting of goals helps to gain acceptance for them.
- Provide Feedback: Not only do goals have to be set but staff has to have regular feedback about how they are progressing.
- Consider Job Design: Jobs can be altered to make them more interesting and give staff more feeling of responsibility. Two measures are often used to enhance job design. They are discussed below :
    - Job Enlargement: The person doing the job carries out a wider variety of activities. It is the opposite of increasing specialization. For example, a software developer in a maintenance group might be given responsibility for specifying minor amendments as well as carrying out the actual code changes.
    - Job Enrichment: The job holder carries out tasks that are normally done at a managerial or supervisory level. With programmers in a maintenance team, they might be given authority to accept requests for changes that involve less than five days' work without the need for their manager's approval.

8.5 WORKING IN TEAMS

Group feelings develop over time. It is suggested that teams go through five basic stages of development:

- Forming: The members of the group get to know each other and try to set up some ground rules about behavior.
- Storming: Conflicts arise as various members of the group try to exert leadership and the group's methods of operation are being established.
- Norming: Conflicts are largely settled and a feeling of group identity emerges.
- Performing: The emphasis is now on the tasks at hand.
- Adjourning: The group disbands.

Teams needed a balance of different types of people:

- The Chair: Not necessarily brilliant leaders but they must be good at running meetings, being calm, strong but tolerant.
- The Plant: Someone who is essentially very good at generating ideas and poten-tial solutions to problems.
- The Team Worker: Skilled at creating a good working environment, e.g. by 'jollying people along'.

- The Monitor Evaluator: Good at evaluating ideas and potential solutions and helping to select the best one.
- The Sharper: Rather a worrier, who helps to direct the team's attention to the important issues.
- The Resource Investigator: Adept at finding resources in terms of both physical resources and information.
- The Completer-Finisher: Concerned with completing tasks.
- The Company Worker: A good team player who is willing to undertake less attractive tasks if they are needed for team success.

A person can have elements of more than one type. To be a good team member one must be able to:

- Time your interventions, e.g. not overwhelm the others in the team
- Be flexible
- Be restrained
- Keep the common goals of the team in mind all the time

A problem with major software projects is that they always involve working in groups, and many people attracted to software development find this difficult. Formal groups can be either the departmental groupings seen on organization hierarchy diagrams reflecting the formal management, structure, or task groups which carry out specific tasks. Task groups might call on people from different departments and would usually be disbanded once the task was completed.

Group Performance:

In many projects, judgments' need to be made about which tasks are best carried out collectively and which are best delegated to indi-viduals. As one manager at IBM said: 'Some work yields better results if carried out as a team while some things are slowed down if the work is compartmentalized on an indi-vidual basis.' Part of the answer lies in the type of task being undertaken.

One way of categorizing group tasks is into:

- Additive tasks
- Compensatory tasks
- Disjunctive tasks
- Conjunctive tasks

Additive tasks mean the efforts of each participant are added to get the final result. The people involved are interchangeable.

NOTES

With compensatory tasks the judgments of individual group members are pooled so that the errors of some are compensated for by the inputs from others. For example, individual members of a group are asked to provide estimates of the effort needed to produce a piece of software and the results are then averaged. In these circumstances group work is generally more effective than the efforts of individuals.

With disjunctive tasks there is only one correct answer. The effectiveness of the group depends on:

- Someone coming up with the right answer
- The others recognizing it as being correct

Conjunctive tasks are where progress is governed by the rate of the slowest per-former. Software production where different staffs are responsible for different modules is a good example of this. The overall task is not completed until all participants have completed their part of the work. In this case cooperative attitudes are productive. The team members who are ahead can help the meeting of group objectives by assisting those who are behind.

With all types of collective task, but particularly with additive ones, there is a danger of social loafing, where some individuals do not make their proper contribution. This can certainly occur with student group activities, but is not unknown in 'real' work environments.

8.6 DECISION MAKING

Decisions can be categorized as being:

- Structured: Generally relatively simple, routine decisions where rules can be applied in a fairly straightforward way.
- Unstructured: More complex and often requiring a degree of creativity.

Another way of categorizing decisions is by the amount of risk and uncertainty that is involved.

Mental Obstacles to Good Decision Making

Many management decisions in the real world, however, are made under pressure and based on incomplete information. We have to accept the role of intuition in such cases, but be aware of some mental obstacles to effective intuitive thinking, for example:

- Faulty Heuristics: Heuristics or 'rules of thumb' can be useful but there are dangers:
  - o They are based only on information that is to hand, which might be misleading;
  - o They are based on stereotypes.
  - o Escalation of Commitment: This refers to the way that once a decision has been made it is increasingly difficult to alter it even in the face of evidence that it is wrong.
- Information Overload: It is possible to have too much information so that you cannot see the wood of the trees.

NOTES

Group Decision Making:

With a project team different specialists and points of view can be brought together. Decisions made by the team as a whole are more likely to be accepted than those that are imposed. Assuming that the meetings are genuinely collectively responsible and have been properly briefed, research would seem to show that groups are better at solving complex problems when the members of the group have complementary skills and expertise. The meeting allows them to communicate freely and to get ideas accepted.

Groups deal less effectively with poorly structured problems needing creative solutions. Brainstorming techniques can help groups in this situation but research shows that people often come up with more ideas individually than in a group. Where the aim is to get the involvement of endusers of a computer system, then prototyping and participatory approaches such as Joint Application Development might be adopted.

Obstacles to Group Decision making:

- It is time consuming
- It can stir up conflicts within the group
- Decisions can be unduly influenced by dominant personalities

Conflict can, in fact, be less than might be expected. People will modify their personal judgments to conform to group norms, common attitudes developed by a group over time.

In fact, people in groups sometimes make decisions that carry more risk than where they make the decision on their own. This is known as the risky shift.

Measures to Reduce the Disadvantages of Group Decision Making:

One method of making group decision making more efficient and effective is by training members to follow a set procedure. The Delphi technique endeavors to collate the judgments of a number of experts without actually bringing them face to face. Given a problem, the following procedure is carried out:

- The cooperation of a number of experts is enlisted
- The problem is presented to the experts
- The experts record their recommendations
- These recommendations are collated and reproduced
- The collected responses are re-circulated
- The experts comment on the ideas of others and modify their recommendations if so moved
- If the leader detects a consensus then the process is stopped, otherwise the comments are re-circulated to the experts

NOTES

An advantage of this approach is that the experts could be geographically dispersed. However, this means that the process can be time consuming.

8.7 LEADERSHIP

Leadership is generally taken to mean the ability to influence others in a group to act in a particular way to achieve group goals. A leader is not necessarily a good manager or vice versa as managers have other roles such as organizing, planning and controlling.

Leadership is based on the idea of authority or power although leaders do not necessarily have much formal authority. Power may come either from the person's position (position power), from the person's individual qualities (personal power) or may be a mixture of the two. Position power has been further analyzed into:

- Coercive Power: The ability to force someone to do something by threatening punishment
- Connection Power: Based on having access to those who have power
- Legitimate Power: Based on a person's title conferring a special status
- Reward Power: The holder can give rewards to those who carry out tasks to his or her satisfaction

Personal power, on the other hand, can be further analyzed into:

- Expert Power: Comes from being the person who is able to do a special-ized task
- Information Power: The holder has exclusive access to information
- Referent Power: Based on the personal attractiveness of the leader.

Measuring Leadership Styles:

Leadership styles can be measured on two axes: Directive vs. Permissive and Autocratic vs. Democratic:

- Directive Autocrat: Makes decisions alone; close supervision of implementation
- Permissive Autocrat: Makes decision alone, subordinates have latitude in implementation
- Directive Democrat: Makes decisions participatively, close supervision of implementation
- Permissive Democrat: Makes a decision participatively, subordinates have latitude in implementation

Another axis used to measure management styles has been on the degree to which a manager is task-oriented, that is, the extent to which the execution of the task at hand is paramount, and the degree to which the manager is concerned about the people around them (people orientation). Subordinates appear to perform best with managers who are good in both respects.

NOTES

Work environments vary with amount of control exerted over work. Some jobs are routine and predictable (e.g. dealing with batched computer output). Others are driven by outside factors (e.g. a help desk) or are situations where future direction is uncertain (e.g. at the early stages of a feasibility study). With a high degree of uncertainty subordinates will seek guidance from above and welcome a task oriented management style.

As uncertainty is reduced, the task-oriented manager is likely to rest, becoming more people-oriented, and this will have superior results. People-oriented managers are better where staff can control the work they do, without referring matters to their line managers. It is then argued that if control becomes even easier the people-oriented manager will be tempted to get involved in more task-centered questions, with undesirable results.

8.8 ORGANIZATIONAL STRUCTURES

Organizational structures can have an enormous impact on the way a project is conducted.

Formal versus Informal structures:

The formal structure is expressed in the staff hierarchy chart. It is basically concerned with authority, about who has which boss. It is backed by an informal structure of contacts and communication that grows up spontaneously between members of staff during the course of work. When the unexpected happens it is often this system that takes over. Over time the advantages and disadvantages of different organizational structures tend to even out - the informal organization gets built up and staff finds unofficial ways of getting around the obstacles posed by the formal structure.

Hierarchical Approach

The 'traditional' management structure is based on the concept of the hierarchy -each member of staff has only one manager, while a manager will have responsi-bility for several members of staff. Authority flows from the top down through the structure. A traditional concern has been with the span of control - the number of people that a manager can effectively control.

Staff versus Line

Staff in organizations can often be divided into line workers who actually produce the end product and support staff who carry out supporting roles. In some organ-izations that produce software for the market or as a component of a larger product which is sold, the software specialists might be seen as part of the line. In a financial organization, on the other hand, the information systems department would probably be seen as part of the support staff.

NOTES

Departmentalization

Differentiation concerns the departmentalization of organizations. This may be based on staff specialisms, product lines, and categories of customer or geographical location. For example:

Software development is usually organized using either a functional or a task oriented approach. With functional departmentalization, systems analysts may be put in a separate group to the programmers. The programmers would act as a pool from which resources are drawn for particular tasks. With a task-oriented approach the programmers and systems analysts are grouped together in project teams. The project team could be gathered to implement a specific long-term project or could exist on a permanent basis to service the needs of a particular set of users.

The functional approach can lead to a more effective use of staff. Programmers can be allocated to jobs on a need basis and be released for other work when the task is completed. For instance, in a project team there are bound to be periods of greater and lesser coding activity and software developers could experience times when they are under-utilized. The functional organization will also make it easier for the programmer to have a career which is technically oriented - there will probably be a career structure within the software development department which allows programmers to rise without changing their specialism. This type of organization should also encourage the interchange of new technical ideas between technical staff and the promulgation of company-wide standards.

However, having separate departments could lead to communication problems, especially if a developer is unfamiliar with an application area. There will also be problems with software maintenance - here it is helpful to have programmers who have built up a familiarity with particular parts of the application software. Users often prefer the established project team approach because they will have a group dedicated to their needs and will not find themselves in .the position of always having to fight other departments for development resources. The project team structure tends to favor a pattern of career progression where software developers eventually become business analysts.

A third method of departmentalization is based on life cycle. Here there are separate teams for development and maintenance. Some staff can concentrate on developing new systems with few interruptions while other teams, more oriented towards service and support, deal with maintenance.

Some organizations have attempted to get the best of all worlds by having a matrix structure. In this case the developer would have two managers: a project leader who would give them day-to-day direction about the work in hand and a programming manager who would be concerned about such things as career development.

NOTES

Egoless Programming:

Programmers and programming team leaders should read other people's programs. Programs would become in effect the common property of the programming group and programming would become 'egoless'. Peer; code reviews are based on this idea.

Chief Programmer Teams

The larger the development group the slower it will get because of the increased communication. Thus large time-critical projects tend to have a more formalized, centralized structure. Design consistency is important when producing a piece of software.

The chief programmer defines the specification, and designs, codes, tests and documents the software. He or she is assisted by a co-pilot, with whom the chief programmer can discuss problems and who writes some code. They are supported by an editor to write up the documentation drafted by the chief programmer, a program clerk to maintain the actual code, and a tester. The general idea is that this team is under the control of a single uni-fying intellect.

Additional members could be added 'to the team on a temporary basis to deal with particular problems or tasks.

The problem with this kind of organization is getting hold of really outstanding programmers to carry out the chief programmer role. There is also the danger of information overload on the chief programmer. There is in addition the potential for staff dissatisfaction among those who are there simply to minister to the needs of the superstar chief programmers.

8.9 DISPERSED AND VIRTUAL TEAMS

A key feature of projects is that each project is to a greater or lesser extent a unique problem. They also usually required a team of people to carry them out, and the members of this team could each be a specialist in a particular field. Thus, the heart of many projects is collaborative problem solving. Group working meant, almost by definition, that the team members worked in close physical proximity. However, in recent years the concept and practice of having dispersed or virtual teams have emerged.

Development of large software products needs co-ordination which in turn means that team members need to communicate. Being located in the same physical space clearly assists this. However, offices can be noisy places and while software development needs communication it also needs periods of solitary concentrated effort. Deep concentration needed for effective creative work as 'flow'.

Today businesses allowed at least some of their staff to work at home. Most of those working at home reported that they were more productive. The development of cheap

NOTES

internet-based communications, supported by broadband channels, has reduced the coordination problems that were the drawback of home working.

Modern communication technologies also mean that organizations can more easily form temporary teams to carry out specific projects from amongst their employees without having to relocate them. The nature of the work carried out in some projects means that the demand for certain specialist skills is intermittent. Ideally the project manager would like to have access to these skills for a short time but then be able to release them and thus avoid further costs.

An example of this might be the passing need for a graphic designer to produce aesthetically pleasing designs for a web application project. This desire for flexible labor means that contract workers are often used. The internet allows these contractors to carry out well-defined tasks at their own premises without necessarily having to travel to their clients' site. It is then only a short step to use 'off-shore' staff who live and work in a different part of the world. Hence we arrive at the dispersed or virtual team.

The advantages are:

- A reduction in staff costs by using labor from developing countries where salaries are lower
- A reduction in the overheads that arise from having your own employees on site, including costs of accommodation, social security payments, training
- The flexible use of staff - they are not employed when they are not needed
- Productivity might be higher
- Use of specialist staff for specific jobs, rather than more general project workers, might improve quality
- Advantage can be taken of people working in different time zones to reduce task durations - for example, software developers can deliver new versions of code to testers in a different time zone who can test it and deliver the results back at the start of the next working day

Some of the challenges of dispersed working are:

- The requirements for work that is distributed to contractors have to be carefully specified.
- The procedures to be followed will need to be formally expressed, where pre-viously practices might have been picked up through observation and imitation of co-workers on site
- Coordination of dispersed workers can be difficult
- Payment methods may need to be modified to make them fixed price or piece-rate.
- There may be a lack of trust of co-workers who are remote and never seen

NOTES

- Assessment of the quality of delivered products will need to be thorough
- Different time zones can cause communication and coordination problems.

8.10 STRESS

Projects are about overcoming obstacles and achieving objectives. Almost by definition both the project manager and team members will be under pressure.

Beyond a certain level of pressure, however, the quality of work decreases and health can be affected. There is good evidence that productivity and the quality of output go down when an individual works for more than about 40 hours a week.

Many software developers are expected to work overtime on projects for no additional payment. In these cases, a fall in productivity is more than compensated for by the fact that the work is effectively free to the employer.

Clearly, it is sometimes necessary to put in extra effort to overcome some temporary obstacle or to deal with an emergency, but if overtime working becomes a way of life then there will be longer-term problems.

Good project management can reduce the reliance on overtime by the more realistic assessment of effort and· elapsed time needed, based on careful recording and analysis of the performance of previous projects. Good planning and control will also help to reduce 'unexpected' problems generating unneces-sary crises.

Stress can be caused by role ambiguity when staff does not have a clear idea of the objectives that their work is supposed to be fulfilling, what is expected of them by others and the precise scope of their responsibilities. The project manager could clearly be at fault in these instances.

Role conflict can also heighten stress. This is where the person is torn between the demands of two different roles.

Some managers claim to be successful through the use of essentially bullying tactics to push projects through. They need to create crises in order to justify the use of such tactics. This, however, is the antithesis of professional project management which aims at a rational, orderly and careful approach to the creation of complex products.

8.11 HEALTH AND SAFETY

The implementation of office systems requires the creation of physical infrastructure which can have inherent physical dangers. Various pieces of legislation govern safety policy and the details of these can be consulted in the appropriate literature. As an example legislation may require organizations employing more than five employees to have a written safety policy document. A project manager should be aware of the contents of the document that applies to the environment in which the project is to be undertaken.

NOTES

As far as the project manager is concerned, safety objectives, where appropriate, should be treated like any other project objectives, such as the level of reliability of the completed application or the overall cost of the project. The management of safety should therefore be embedded in the general management of the project.

Responsibility for safety must be clearly defined at all levels. Some points that will need to be considered include:

- Top management must be committed to the safety policy
- The delegation of responsibilities for safety must be clear
- Job descriptions should include definitions of duties related to safety
- Those to whom responsibilities are delegated must understand the responsibili-ties and agree to them
- Deployment of a safety officer and the support of experts in particular technical areas
- Consultation on safety
- An adequate budgeting for safety costs

Safety procedures must be brought to the attention of employees and appropriate training be given where needed.

Summary:

Motivation plays a major role in managing people. Both staff selection and the identification of training needs should be done in an orderly, structured, way where requirements are clearly defined first. Thoughtful job design can increase staff motivation. Consideration should be given, when forming a new project team, to getting the right mix of people and to planning activities which will promote team building. Group working is more effective with some types of activity than others. Different styles of leadership are needed in different situations.

Questions:

1. Explain the recruitment process.
2. Explain the Expectancy theory of motivation.
3. Explain Oldham-Hackman characteristics model.
4. Present a tutorial on leadership styles.
5. Explain dispersed and virtual teams.

NOTES

NOTES