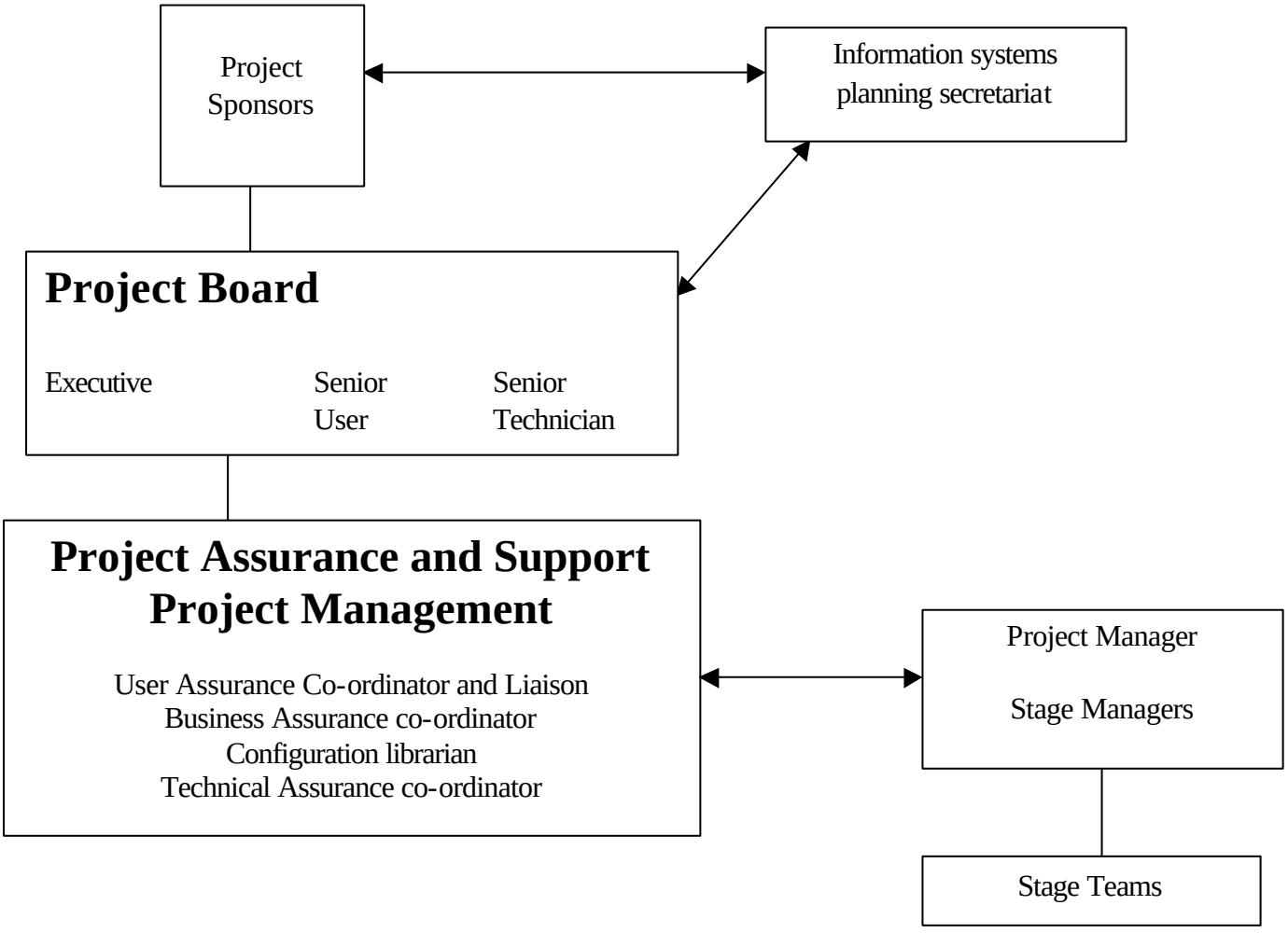
# Project planning and control

PRINCE – Projects IN a Controlled Environment, is an open method for managing and controlling IT projects. PRINCE consists of several components

- Organisation
- Products
- Plans
- Controls
- Activities

# Typical project organisation




**Project Sponsors**
Individuals who have responsibility for paying for the development. They will usually have ultimate responsibility for the parts of the organisation that will be affected by the system. By their nature these tend to be very senior people and must be kept informed of progress.

**Information Systems and planning secretariat**
Monitors the links between the individual project and the overall information systems plan or strategy. They also have responsibility for coordinating the links between projects in the strategic plan.

**Project board**
The project board will be appointed by the project sponsors to have overall control of the project. Members will be managers themselves and are not usually involved in the running of the project on a day-to-day basis. It comprises three senior management roles – an *executive*, a *senior user*, and a *senior technician*.

The *executive* provides overall guidance for the project. S/he normally acts as the chairperson of the project board. The *senior user* represents the users of the system (current and future) and/or the major products of the system. This will most often be the manager of the user most likely to be affected by the system under development. The *senior technician* represents the technical side of the development of the system and of its future operations. S/he will be an IT manager from the IS department. For large projects this role will be taken by the IS director or most senior person.

The project board has overall authority to commit resources to and take resources from the project. They review all major plans and exceptions, authorising each stage of the project. It is their responsibility to provide overall guidance and direction for the project ensuring it meets agreed standards of quality, time and cost and that it meets the business objectives set for the project. Quite often they will oversee several projects.

**Project manager**
This person has overall responsibility for the day to day running of the project. They report to the project board and have responsibility for planning the project, defining controls, liasing with other projects and defining configuration of management. If the project is large the day-to-day management role will sometimes be split between an overall project manager and several stage managers. More typically project and stage management roles are combined. If the roles are distinguished then the project manager takes a more strategic view and the stage managers will take a more staff management view managing the project teams.

**Project assurance team**
This group of roles is involved in quality assurance of all products and all plans. They are divides into business, technical and project management issues. The team splits into three roles

- Business assurance co-ordinator.
- Technical assurance co-ordinator
- User assurance co-ordinator

A further role is the configuration librarian who has overall responsibility for configuration management of the project. They act as the custodian for all master copies of products produced by the project and are the focus for change control.

*Business assurance co-ordinator* is usually someone with financial and administrative knowledge including experience of cost/benefit analysis. They are primarily responsible for assessing the business aspects of the project.

*Technical assurance co-ordinator* is usually someone with considerable technical experience who is not directly associated with the project. Their primary responsibility is to monitor and report on all technical assurance aspects of the project and to ensure that the technical standards defined by the project are met.

*User assurance co-ordinator* is someone with user knowledge of the business environment and of the individuals affected by the project. They are responsible for representing the users of the project on a day-to-day basis and co-ordinating their activities. Typically this involves selecting which users should receive which products for review, determining the training needs for users, deciding which users should be involved in prototype evaluations. They need to be good communicators and demand respect in the user community.

# Products

A project is judged on what it delivers so the emphasis is on products rather than on the activities that produce them. For each stage in the project a hierarchy of products to be produced is defines and shown in a product breakdown structure and for each product there should be a product description. A product description should have the following components

- Purpose
- Composition (what is included)
- Derivation (what products is it derived from)
- Quality criteria

# Planning

Product breakdown structures are used to produce product flow diagrams showing the derivation of each product from the others and the interdependencies between products. This is then used to produce an activity network showing which activities are needed to produce the products.

Several types of plans are required

- Technical plan schedules activities needed to produce a product
- Resource plan shows the amount and cost of each resource that is needed.
- Quality plan defines quality assurance standards to be followed.
- Configuration management plan describes how configuration management should be applied to the project.
- Exception plans will need to be raised when problem arise with original plans.

# Project control

Control is necessary to ensure that the project stays on schedule, within budget and delivers a quality product. Project control operates at 2 levels, technical and management. Technical controls are generally associated with the quality of products and how well they meet the users requirements. Management control is mainly concerned with schedules and resources. The senior level involves the project board, the project manager, the stage manager and the project assurance team. Junior level involves the stage manager, their teams and team leaders in both meetings and written report.

*Control Mechanisms*
At the senior level meetings and assessments are held at the major milestones of the project, project initiation, project closure and end stage. In each case plans and authority for the next stage are reviewed and performance during the previous stage assessed.

Project initiation involves a meeting and a review of the project initiation document to ensure that the project starts on a firm business footing with clear objectives. At the end of each stage and end stage assessment occurs. This reviews the performance of the stage that has been finished against budget and schedule and the status and quality of the end products at this stage.

Frequent *highlight reports* will be produced by the project manager and circulated to the project board and the project assurance team. These summarise the progress on the stage thus far highlighting any problems real or potential and forecasting progress.

At a more junior level *checkpoint meetings* are held by the stage manager and involve the team members. They review progress against work plans, discuss any technical problems and identify targets for the next checkpoint meeting.

*Exception reporting*
A *project issue report* can be raised at any point during the project by any person associated with the project and can cover anything to do with the project. The stage manager and the project assurance team consider project issue reports. The project manager has final responsibility on what should happen to a project issue report.

An *off specification report* documents any technical situation where the system fails to meet its specification. These are authorised by the project manager based on recommendations by the project assurance team after their analysis of project issue reports. Some assessment is required of the work needed to correct the off specification report and its impacts. Ultimately a decision will be authorised by the project board as to how the problem should be corrected.

# Project Initiation Document

This is a short document – often called the terms of reference –, which is drawn up and agreed between the senior users -usually the people who requested the system and who have the financial authority to pay for it –, and the development team. If a

strategy or feasibility study has been done then this could be a very detailed document giving clear boundaries for the project and precise budgets and timescales. If no studies have been performed then the project initiation document might contain rather vague statements about scope, constraints, budgets and timescales.

It is important that some contract be agreed between the user and developer. This is often the first formal statement by the users of what they want and thus forms the basis for subsequent work. It is important that the project initiation document is sufficiently detailed and clear for the project team to be confident that they can proceed.

# Configuration management

A software project involved development of a variety of complex inter linked products, each of which is subject to change. The process of controlling these products ensuring the right version is used and managing changes to products is known as configuration management. It covers the management tracking of all technical products.

It is particularly significant when it comes to the delivery of a released system. This will comprise hardware, software, data and documentation. Unless this complex configuration is managed and controlled chaos can occur with different team members using different versions of documents, plans, software etc.

**Configuration Items**
Those products critical to the workings of the system, which must be subject to formal configuration control procedures. The project team must identify which products will become Configuration items. The default set of Configuration items is defined by the product breakdown structures. Typically these comprise any products that are carried forward into the next stage of the project. Ultimately the Configuration items will be those that comprise the final system.

A good example of such a product is the required system logical data model – changes to this may necessitate major changes to other products such as the entity life histories, update and enquiry process models. This would become a configuration item when it passes through quality control at the end of the requirements specification module.

**Baselines**
A released set of configuration items forms a system baseline defining a fixed reference point in the development, usually at the end of a stage. Baselines can be established at any time in the project and will be an important part of the configuration management plan.

Once a product has been baselined then control of it passes to the configuration librarian and a version number is allocated. Requests to change baselined products must follow formal configuration control procedures through requests for change or off specification reports. In this way changes to baselined products must be sanctioned by project management.

# Estimating

One of the most difficult tasks in project management is estimating the time and resources required to perform each of the activities. Usually a combination of approaches are used, ranging through the highly scientific, the completely intuitive to the pragmatic.

As a project is being developed to meet a business need then the business must know when its need can be satisfied and how much it will cost. The business can then make decisions about whether additional resources should be applied and whether the costs justify the end result.

Some form of estimating is necessary in the very early stages of a project in order to decide whether it is worthwhile to continue. Of course estimating early in the life cycle is inherently inaccurate, the scope of the project is not completely known, technology and skills available may change and unforeseen problems may arise. As the project progresses then estimates should become progressively more accurate.

**Initial estimates**
Early in the project life estimating must be done top down. We rely upon the experience of the estimator and the organisation. He/she must look at the whole project, including the size of the application, the technology to be employed and the development team. With reference to past history and previous experience they develop an estimate based on an analogy or more scientifically by calculation.

The pragmatic project manager will look at the required delivery date and work backwards using the resources that he or she has. This presumes that some resources and time have already been allocated to the project, which then presumes that some estimation of the effort required has been made. These estimates may have been made by waving a finger in the air and coming up with the magical number of person months. Hopefully the estimator has some experience of similar projects on which to base an estimate.

**Refined estimates**
The other approach to estimation – more applicable later in the development – is a bottom up approach. In this the project is broken down into low level activities and the effort for each activity is estimated. This form of estimation is necessary for planning the next phase of the development or for the whole project when a full set of requirements are known.

**Function point analysis**
A more scientific approach to estimation is to use techniques such as function point analysis mark 2. In FPA a metric is developed known as a function point – systems can be compared in terms of their function point counts. A 1000 function point system should have twice as much functionality as a 500-function point system.

The mark 2 method calculates the size of the system by first estimating the number of unadjusted function points, and then multiplying these by an adjustment factor for technical complexity to give a total size of the system.

Once the system size has been estimated we can estimate the effort required by dividing the size by the productivity – measured in person hours per function point. Other factors need to be taken into account that may be unpredictable such as relations with users, staff expertise and the time taken to adjust to new methods and tools. You can only estimate effectively if you have a history of past projects. It is thus most important to collect metrics about your organisation's performance on previous projects.

# Quality Assurance

It is very important to ensure that the products from each stage are technically correct and that they meet the objectives of the users. Each stage builds on the work done in the previous stage. With poor foundations there is a high risk that all subsequent work will be poor.

At the end of each stage a formal quality assurance review of its end products is suggested. This involves identifying and educating reviewers, sending them documentation and responding to any queries, a formal meeting and follow up actions and eventual sign – off of products to the next stage of the project. This should be done on a formal basis to force the correction of errors identified by the reviewers before work is allowed to proceed to the subsequent stages.

A sign off by a group consisting principally of users emphasises the joint responsibility fro the project of both the users and the project team. This ensures the continuing active interest of the users in the project and helps avoid the situation commonly met in systems development when poor communication between the project team and the users leads to an implemented system that does not meet the users requirements.

Products from each stage should be reviewed by a team of responsible and representative users who can authorise changes in the working environment. The review team should include at least one technical reviewer who has a good understanding of systems development.

There is a danger that users may feel rail roaded or blackmailed into signing off large specification documents which they haven't had time to read and barely understand. Project and user management has a clear responsibility to ensure that this does not happen. This means that users are trained to review products, that users are informally walked through all significant products before the formal review, that users are given sufficient time to read and digest the documentation and that users views are listened to, understood and acted upon by the project team.

**Review Procedures**

*Before the review*

The project team and the project assurance team select the reviewers and fix a time and a place. All recipients receive a Quality review invitation one week in advance of the meeting, together with a neat copy of all documents they will be required to review. If any of the reviewers is unfamiliar with the convention of the diagrams, then the analysts might arrange to explain the aspects of the diagram that are relevant

to a reviewer. This can be done more efficiently by organising a presentation. The reviewers study the product and note errors.

*The review meeting*
This should not be more than 2 hours long. The chairman is either a user who has been closely involved with the project or the project team leader. The meeting begins with the circulation of an agenda, possibly some introductions and a clear statement of the objectives of the meeting. The meeting should not attempt to solve errors but simply highlight them for subsequent resolution away from the meeting. An analyst from the project team talks through the documentation being reviewed and invites comments from the reviewers. A list of errors is compiled by the chairman on the follow up action list and agreed by the meeting.

The reviewer may decide that the documentation contains no errors and meets its objectives in which case they will sign the stage off at this meeting. More commonly there will be a number of non critical errors detected in which the documentation may be signed off provided that certain follow up action is taken and subsequently agreed by the reviewers out of the meeting. If there are many severe errors and the reviewers are not confident that the project team have met the objectives of the stage, then a date for another quality assurance review is set and the documentation failed.

*After the review*
Any necessary corrections are made to the documents within a week of the review and circulated to the members of the review team. If the errors are only minor the reviewers may sign it off individually. If the errors are more serious the documentation is reviewed again.

At least 3 elapsed weeks should be allowed for each formal review and 1 to 2 weeks for informal reviews.

# Risk Analysis

Attempts to determine the risks to a project. Risk management minimises the chance of risks occurring and contains them if they do occur.

In large projects risk analysis should be a significant exercise. As with estimating the level of risk can be assessed more accurately as the project progresses. There are 2 major methods employed,

- Qualitative approach using questionnaires
- Quantative approach using statistical analysis

In practice it is often sensible to use a combination of the two approaches.

The most significant risks are always difficult to spot until after they have happened but a good project manager should be aware of all risks and the factors that could cause them. Most importantly they must communicate these risks to their managers and their project team to help avoid or contain them if they do occur.

All this extra effort will cost - a management decision needs to be made as to whether the risks should be taken or contained but it is better to make this decision in knowledge rather than ignorance.

# Team Building

More and more businesses are turning to teams as a way of boosting creativity and performance in the workplace. As organizations become flatter and responsibility works its way throughout the organization, managers have discovered that they need to be skilled in team facilitation.

Management experts have identified five stages that a work team goes through. Understanding these stages and what to expect during them helps managers to choose appropriate techniques to lead employees to greater productivity and job satisfaction. The Tuchman Model of group development gives these names to the stages: Forming, storming, norming, performing, and adjourning (sometimes called transforming if the team is an ongoing one).

**Stage 1: Forming**
A team will go through the forming stage at two different times: When the team is new and first put together and whenever a new member joins the team.

During the forming stage, members of the work team try to determine whether they should join the team and how well they fit in. If it is an entirely new team, this stage is spent with the individuals determining whether they want to be a part of it and whether it will be useful. Managers should strive to create a relaxed and friendly environment for the first team meeting. That meeting can be used to introduce people, explain what the team is to achieve, who will be affected by their work, and what the purpose of the group is. It is also a time to clarify team member roles, communicate timelines, and set up meeting dates. Team members must also determine how they are going to achieve their goals.

This is also a stage teams go through whenever a new person joins them. The new person is asking:
- How will I benefit from membership in this group? What can I contribute to be accepted?
- What will the other team members expect from me? What do I expect from them?
- How will I learn what is really important to the team? What are their goals? Are my goals compatible with them?
- Will I find that being a member of the team is boring or exciting? Threatening or rewarding?

The existing members will also have questions. The questions or concerns that they might have will have to be resolved before the new member becomes accepted. These questions can include:
- Can I trust this person?
- How will this person change the dynamics of the group?
- Can I work effectively with this person?
- Will this person become a cooperative team player?

There are many things a manager can do to help both a new employee or new team and experienced employees through this stage. A few techniques include:
- Tell new employees information about their co-workers that will make the new employees want to be part of the team.
- Tell experienced employees a little bit about the new person.
- Explain how the new person will help the team effort to both the new person and the experienced employees.
- Have a solid orientation and training program in place.
- Recognize the challenges to joining a new team.
- Communicate clearly and be available to answer questions.
- Tell experienced employees what they have in common with the new person.
- Ask employees to help make the new person feel welcome.

**Stage 2: Storming**
The second stage of team development involves either the new member or the new team determining who is in charge. Team members will spend time finding out whom they can turn to for leadership and guidance. They might ask questions such as:
- Who has the most power to influence people on this team? Who are the strong formal and informal leaders?
- How are others in the work group influencing me? Who am I learning from?
- How are others in the work group influencing each other? Is there really a team effort?
- What opportunities are there for me to influence the team? Can I become an informal leader?
- The storming stage is also when the group begins to see conflict or experience tension. Teams will have to resolve such issues as company politics, lack of commitment, miscommunication, hidden agendas, missed deadlines, competing priorities, etc. The team must learn how to appreciate their differences and move beyond self-interest. The team leader must not ignore the conflict, but rather bring it into the open and coach the team members on how to resolve it.

The storming stage is the time to focus members on how to solve problems and move forward.

**Stage 3: Norming**
The norming stage occurs when the team identity begins to crystallize. Team members are growing comfortable with each other and the feelings of their co-workers become more important. Team members may have the following concerns:
- Can I freely express my feelings in this team? Will other members accept constructive criticism? Is the work atmosphere open and honest?
- When I'm under stress, frustrated, or angry, can I work things out with the team?
- When members agree with others or express positive feelings, do others see it as honest feedback or do they see it as insincere patronizing?
- When members criticize ideas or express negative feelings, do others see it as honest feedback or do they see it as a clash of personalities?

It is during this stage that patterns for teamwork develop and the team is able to successfully build consensus. It is during this time that the team leader should focus on the issues or challenges that the team was formed to address. Other things the leader should do include:

- Provide time for team members to share their experiences and attitudes with each other.
- Let team members know that their opinions and comments are appreciated.
- Use the feedback provided by group members whenever possible.
- Compliment employees for their contributions to the team.
- Recognize when the team has gone beyond expectations.

**Stage 4: Performing**
During this stage, the team values the individual differences among its members and produces results at its highest level. Team members trust each other, enjoy working together, are highly committed to the team and have high energy levels. It is at this stage when the team leader needs to participate the least.

When team members are in the performing stage, they are more likely to:
- Spend time discovering the causes of problems
- Analyse causes of problems
- Seek ideas about the best way to resolve problems
- Decide whether the new way is working or if different procedures are still necessary

**Stage 5: Adjourning or Transforming**
If the team was meeting for the performance of a single task that is, it was an ad hoc team, or a project team, then the final stage is called adjourning. It is in this stage where the team's work is brought to a close and the members can celebrate what they have accomplished. It is a time to evaluate what was done and how it was done. It's a good time to determine what mistakes can be avoided in the future and what things should be repeated.

If the team is an ongoing one, the transforming stage is where it passes from one function to the next. When it accomplishes a set of goals, it must determine its next set of goals, whether the vision is still valid and what they need to do differently. Like the adjourning stage, they must evaluate what they have accomplished and what was most effective

**Moving Through Stages**
It is important to note that teams will not necessarily progress in a straight line through these stages. They may cycle through several different stages multiple times. They may stagnate at one stage for a while before moving on. As long as the manager and/or team leader recognize the stages, they are able to respond appropriately to keep the team focused on its goals and moving toward the performing stage.

# Definitions

Data flow model,

A set of data flow diagrams and their associated documentation.

Data flow diagram,

Shows how services are organised and processing is undertaken. It should be a simple diagram that is readily understood so that it can act as an effective means of communication between analysts and users,

Context diagram,

This may be drawn to illustrate the initial scope of the proposed system. The diagram concentrates on the major inputs and outputs of the system and shows the external sources and recipients of the system data.

Logical data structure,

A diagrammatic representation of the information needs of an organisation in the form of entities and the important business relationships between them.

Entity,

Something whether concrete or abstract which is of importance to the area of business being investigated.

Business systems options,

The means by which users agree the new applications desired functionality with developers. Used to define the functionality needs and the boundary for the system with reference to the business needs.

Technical systems options,

The means by which users agree the new applications implementation strategy incorporating the desired functionality. It gives the technical direction for future development.

Outline current environment description,

A brief description of the current services and any existing problems.

Outline required environment description,

A brief description of the requirements to be included within the proposed system.

Problem definition statement,

A statement of the user requirement for the system which may include charts and diagrams to supplement the detail.

Requirements catalogue,

Is the central repository for information covering all identified requirements both functional and non-functional. Each entry is textual and describes a required facility or feature of the proposed system.

User catalogue,

Provides a description of the on-line users of the proposed system. It includes details of job titles and the tasks undertaken by each of the identified users.

# Feasibility Study

A short assessment of a proposed information study to decide whether the system can effectively meet the specified business requirements of the organization and whether a business case exists for developing such a system.

Feasibility studies are recommended for all projects except those of very low risk. In practice feasibility is often carried out by means of a strategy study which identifies several potential systems (projects) and performs a short feasibility study on each of them. Feasibility for a large project could require as much as one-person year of effort. Outline models or documents are produced sufficiently detailed to assess the feasibility of the project.

The end of the module is a decision making point at which the senior users and the project board must decide the future of the project. A range of options both business and technical will be presented to them and they will make a decision to select one of these, a combination of these or to terminate the project.

**Structure of Feasibility Study module.**

Step 010 – Prepare for the study.
This reviews any documentation already produced for the project such as the Project initiation document which should include the terms of reference. An initial requirements catalogue is produced. The outline boundaries of the project are developed using data flow models to produce context diagrams, a level 1 current physical data flow diagram and an overview logical data structure. Plans are developed for the feasibility study which includes roles for both analysts and users, activities and products to be produced.

Step 020 – Define the problem.
The current and required environments are described textually. A more detailed current physical data flow diagram and a more detailed logical data structure are produced. The users of the future system are identified and defined in the user catalogue. New functions, data and non-functional requirements are recorded in the requirements catalogue. A problem definition statement is produced and agreed with the project board and senior user management. This summarises the requirements and ranks their priority in relation to the business objectives.

Step 030 – Select feasibility option,
This step combines the Business systems options and the Technical systems options techniques to produce a minimum of two feasibility options for recommendation to the project board – who will choose an option (or combination). The suggested approach is to first identify the minimal requirements, then to produce some outline Business System options, then to produce outline technical system options each satisfying one business system option. Combined options are produced and reduced to a manageable level (ideally 3). Descriptions are prepared for each option containing an outline Development plan. These are presented to the project board and senior users who will make a decision and finalise an action plan for the development.

Step 040 – Assemble the feasibility report.
Like all other assemble steps this checks the completeness and consistency of the feasibility study module products. These consist of an action plan, feasibility options, outline current environment description, outline required environment description, problem definition statement, requirements catalogue and user catalogue. These are brought together to produce the feasibility report.

A feasibility report can be a detailed document that records all decisions made and forms the basis for a full study of the project. It should be produced in accordance with the organizations standards for internal reports. It contains the following sections,

- Introduction
- Management/executive summary
- Existing business and IS support to the business
- Future IS support required by the business
- Options considered but rejected
- Financial assessment
- Project plan
- Conclusions and recommendations
- Appendices and supporting documents.

Feasibility analysis is used to aid the decision of whether or not to proceed with a project.

Main areas

- Technical feasibility – Can we build it?
- Economic feasibility – Should we build it?
- Ecological feasibility – Should we build it? Socially!
- Organisational feasibility – If we build it will they come?

Feasibility analysis should kill some projects but for others it should highlight the risks

Technical feasibility

- Familiarity with application – knowledge of business domain
- Familiarity with technology – extension of existing firm technologies
- Project size – number of people, time and features

Economic feasibility

- Benefits
- Development costs
- Operational costs

Financial evaluation of IS projects is problematic and unfortunately favours pure automation. Typically cash flows are quantified and capital budgeting evaluation techniques are applied

Organisational feasibility – most difficult to evaluate

- Stakeholder analysis – project champion, organisational management, system users

Classic project management mistakes

- Overly optimistic schedule
- Failure to monitor schedule
- Failure to update schedule
- Adding people late to a project

# Typical costs

Hardware
Software
Telecommunications
Services
Personnel

# Tangible Benefits (cost savings)

A tangible benefit is one that can be quantified and assigned a monetary benefit.

Increased Productivity
Lower Operational costs
Reduced workforce
Lower computer costs
Lower outside vendor costs
Reduced rate of growth in expenses
Reduced Facility costs

# Intangible Benefits

These cannot be immediately quantified but may lead to quantifiable gains in the long run.

Improved asset utilization
Improved resource control
Improved organizational planning
Increased organizational flexibility
More timely information
More information
Increased organisational learning
Legal requirements attained
Enhanced employee goodwill
Increased job satisfaction
Improved operations
Higher client satisfaction
Better corporate image

# Financial models for measuring Capital Investment

**Payback method**

A measure of the time required to pay back the initial investment of a project. It is computed as,

$$\frac{\text{Original Investment}}{\text{Annual net cashflow}} = \text{number of years to payback}$$

The weakness of this measure is that it ignores the present value of money, the amount of cash flow after the payback period, the disposal value and the profitability of the investment.

**Accounting rate of Return on Investment (ROI)**

Determining a satisfactory rate of return of investment depends on the cost of borrowing money and other factors. In the long run the desired rate of return must be equal to or exceed the cost of capital in the marketplace. Otherwise no-one will lend the firm money.

The ROI calculates the rate of return from an investment by adjusting the cash inflows produced by the investment for depreciation. It gives an approximation of the accounting income earned by the project.

First calculate the average net benefit,

$$\frac{(\text{Total Benefits} - \text{Total Cost} - \text{Depreciation})}{\text{Useful Life}} = \text{Net Benefit}$$

Then divide this by the total investment,

$$\frac{\text{Net Benefit}}{\text{Total Initial Investment}} = \text{ROI}$$

The weakness of this method is that it ignores the time value of money – future savings are not worth as much in today's dollars as are current savings. ROI can be modified so that future benefits and costs are calculated in today's dollars.

**Net Present Value (NPV)**

Evaluating a capital investment requires that the cost of an investment be compared with the net cash inflows that occur many years later. These two kinds of inflows are not comparable due to the time value of money. Money received in the future must be discounted by some appropriate percentage rate – usually the prevailing interest rate. Present value is the value in current dollars of a payment or stream of payments to be received in the future,

$$\text{Payment} \times \frac{1 - (1 + \text{interest})^{-n}}{\text{interest}} = \text{Present value}$$

The net present value is the amount of money an investment is worth taking into account its cost, earnings and time value of money,

$$\text{Present value} - \text{Initial Investment} = \text{Net Present Value}$$

**Cost-Benefit Ratio**

A simple method for calculating the returns from a capital expenditure,

$$\frac{\text{Total Benefits}}{\text{Total Cost}} = \text{Cost Benefit ratio}$$

**Profitability Index**

Calculated by dividing the present value of the cash inflow from a capital investment by the initial cost. The result can be used to compare the profitability of alternative investments,

$$\frac{\text{Present va les of cash inflows}}{\text{Investment}} = \text{PI}$$

**Internal Rate of Return (IRR)**

Defined as the rate of return or profit that an investment is expected to earn. IRR is the discount rate that will equate the present value of the projects future cash flows to the initial cost of the project. The value of R (discount rate) is such that Present value – Initial cost = 0.

# What is Work Breakdown Structure (WBS)?

A hierarchical structure that is used to organize tasks for reporting schedules and tracking costs. With Microsoft Project you can represent the WBS using task Ids or by assigning your own WBS code to each task.

# What are WBS codes?

Work breakdown structure (WBS) codes are alphanumeric codes that identify each task's unique place in the outline structure of your project. WBS codes can be used for reporting schedules and tracking costs. There are two types of WBS codes in Microsoft Project.

- Outline numbers are the simplest type of WBS code; Microsoft Project calculates outline numbers for each task based on the outline structure of the task list. Outline numbers consist of numbers only, and you can't edit them, but they change automatically when you move a task up or down in the task list or indent or outdent tasks. Learn how to display outline numbers.
- The second type of WBS code is a custom code that you define. You can define one set of custom WBS codes per project and display it in the WBS field. Each level of the WBS code represents an outline level in the task list. But unlike outline numbers, the levels of the code can be represented as uppercase or lowercase letters, numbers, or characters (a combination of uppercase and lowercase letters and numbers), depending on which you

specify for each level in the code mask when you create the WBS code. You can choose whether to automatically calculate custom WBS codes for new tasks and whether to allow duplicate WBS codes for different tasks.

# How do you create a WBS code?

If your organization or client requires a particular WBS code format, you can define sequence for each level of the WBS code using a custom code mask. A code mask is a format that you define for a WBS code or a custom outline code. The mask specifies the sequence and number of the letters or numbers required for each level and the symbol separating the level. Microsoft Project uses the code mask to assign WBS codes to tasks depending on their places in the hierarchy of the project's outline. You can define a WBS code mask using:

- A project code prefix to identify the project at the highest level of the WBS code. This prefix can be useful to identify subprojects consolidated within a master project. You can enter any combination of numbers, uppercase and lowercase letters, and symbols for the project code prefix.
- Numbers (ordered) to display a numerical WBS code.
- Uppercase Letters (ordered) to display uppercase alphabetical WBS codes, for example A, B, and C for the first three summary tasks in the project.
- Lowercase Letters (ordered) to display lowercase alphabetical WBS codes, for example a, b, and c for the first three summary tasks in the project.
- Characters (unordered) to display any combination of numbers and uppercase and lowercase letters you enter, for example, Arch1, Const1, and Insp1 and for the first three summary tasks in the project. Choosing unordered characters gives you the most flexibility for entering customized WBS codes. Microsoft Project displays an asterisk * in the WBS field until you type or enter a string of characters for this level of the WBS code.

To set a prescribed number of characters or numbers for each level users must enter the exact length of the level you specify, or you can specify Any to allow any length of characters or numbers for the code level.

Code separators to distinguish each level of the WBS code by displaying a different symbol instead of the default separator, a period.

After you create a WBS code mask, you can display custom WBS codes in a task sheet.

# The Network Diagram view

The Network Diagram view displays tasks and task dependencies in a network or flowchart format. A box (sometimes called a node) represents each task, and a line connecting two boxes represents the dependency between the two tasks. By default,

the Network Diagram view displays one diagonal line through a task that is in progress and crossed diagonal lines through a completed task.

You can use the Network Diagram view to:

- Create and fine-tune your schedule.
- Link tasks to specify the task sequence, as well as determine start and finish dates.
- Graphically show completed, in-progress, and not-yet-started tasks.
- Assign personnel and other resources (such as equipment) to specific tasks.

You can apply any of the task filters to the Network Diagram view to display only the tasks you want to see. To make the Network Diagram view fit your needs exactly, you can customize its appearance or create new, customized versions. When you save your project, the customized view is saved with the project. When you customize the Network Diagram view, you can:

- Adjust Network Diagram boxes to contain the task and resource information that is most important to you. For example, instead of displaying the scheduled start and finish dates, you can display the work and the cost.
- Apply a different Network Diagram shape or border style to a category of tasks, such as critical summary tasks
- Change the appearance of the lines that connect Network Diagram boxes or label them, and prevent Network Diagram boxes from crossing page breaks.
- Format specific information to distinguish it from all other information. For example, you can italicize all summary tasks and format all milestone tasks as bold text.
- Filter the Network Diagram view like any other view to display the information you want to focus on.
- Choose from different layout patterns and align Network Diagram boxes to quickly give your project an orderly appearance.
- Expand or collapse the Network Diagram to the level you want to see, by showing and hiding the subtasks of summary tasks.
- Create a combination view for the Network Diagram to display additional information about the tasks shown in the Network Diagram view or about the resources assigned to those tasks.

# What is the critical path?

The critical path is the series of tasks (or even a single task) that dictates the calculated finish date of the project. That is, when the last task in the critical path is completed, the project is completed. If it's important for your project to finish on schedule, pay close attention to the tasks on the critical path and the resources assigned to them. These elements determine whether your project will finish on time. The series of tasks are generally interrelated by task dependencies. Although there are likely to be many such networks of tasks throughout your project plan, the network finishing the latest is the project's critical path.

Note that the critical path can change from one series of tasks to another as you progress through the schedule. The critical path can change as critical tasks are completed, or tasks in another series of tasks are delayed. There is always one overall

critical path for any project schedule. The new critical path then becomes the series of tasks you track more closely to ensure the finish date you want.

# What is a critical task?

Those tasks that cannot be delayed without affecting the project finish date are the critical tasks. In a typical project, many tasks have some slack and can therefore be delayed a little without affecting the project finish date. Slack is the amount of time a task can be delayed without delaying other tasks or affecting the project finish date.

As you modify tasks to resolve over allocations, adjust costs, or revise scope, be aware of the critical tasks and that changes to them will affect your project finish date.
Critical tasks make up the schedule's critical path.

A task becomes critical when it meets any one of the following conditions:

- There is 0 slack on the task.
- It has a Must Start On or Must Finish On date constraint.
- It has an As Late As Possible constraint in a project scheduled from a start date.
- It has an As Soon As Possible constraint in a project scheduled from a finish date.
- It has a finish date that is the same or beyond its deadline date.

A task stops being critical when it's marked as completed because it can no longer affect the completion of successor tasks or the project finish date.

# How does Microsoft Project calculate the critical path?

Microsoft Project defines critical tasks as those that have no slack. Slack is the amount of time a task can be delayed without delaying other tasks or affecting the project finish date. Those tasks that cannot be delayed without affecting the project finish date are the critical tasks.

Although Microsoft Project defines a task as critical if it has zero days of slack, you can change the definition of a critical task. For example, you can make a task critical if it has one or two days of slack. This can be helpful if you want to be alerted to tasks becoming critical when you still have a day or two of buffer.

Slack is determined by the early finish and late finish dates of the tasks in your schedule. A task's early finish date is the earliest date that the task could finish, based on its start date and scheduled duration. A task's late finish date is the latest date that the task can finish without delaying the project finish. The difference between early finish and late finish dates determines the amount of slack. For critical path tasks (tasks that have no slack), the early finish and late finish dates are identical.

# Why is it important for me to know the critical path?

By knowing and tracking the critical path for your project, as well as the resources assigned to critical tasks, you can determine which tasks can affect your project's finish date and whether your project will finish on time. You can see the critical path using the Detail Gantt, which highlights the critical path, along with any slack the noncritical tasks have.

# How do I shorten the critical path?

If you want to bring in the project finish date, you need to bring in the dates of your critical path tasks. This is also known as "crashing." To do this, you can:

- Shorten the duration or work on a task on the critical path.
- Change a task constraint to allow for more scheduling flexibility.
- Break a critical task into smaller tasks that can be worked on simultaneously by different resources.
- Revise task dependencies to allow more scheduling flexibility.
- Set lead-time between dependent tasks where applicable.
- Schedule overtime.
- Assign additional resources to work on critical path tasks.

Be aware that if you bring in the dates of your critical path, a different series of tasks could become the new critical path. There is always one overall critical path for any project schedule. The new critical path would then become the series of tasks you track more closely to ensure the finish date you want. However, if you bring in the finish dates of the critical path, and another series of tasks does not overtake it, then you can successfully bring in the finish date of the project as a whole.

# Can I see multiple critical paths?

By default, Microsoft Project displays only one critical path, the one overall critical path that affects the plan's finish date. You can set up your project plan to see multiple critical paths for each independent network or series of tasks. You might find this useful if you are working with a master project, and you want to see the critical path for each subproject. This might also be useful if the project is broken into multiple phases, and you want to see the critical path for different phases or milestones. When viewing multiple critical paths, remember that there's still one overall critical path whose finish date affects the project's finish date. Changing the finish date for any of the secondary critical paths will probably not change the project's finish date.

# Can I see the critical path across multiple projects?

If you are working with multiple projects, with cross-project links, or inserted projects, you can see the overall critical path. Inserted projects can be treated as summary tasks for Microsoft Project to calculate the overall critical path.

# Gantt Chart view summary

The Gantt Chart view displays project information in two ways: the left side displays information as a sheet and the right side displays information as a chart. The sheet portion displays information about the project's tasks, such as when they start and end, how long they are, and the resources assigned to them. The chart portion displays each task graphically, most often as a task bar. The bar's position on the timeline, and its length, indicates when that task begins and ends. In addition, the position of one task bar in relation to another indicates whether the tasks follow one after the other or are overlapping.

Use the Gantt Chart view to:

- Create a project by entering tasks and the amount of time each task will take.
- Establish sequential dependencies between tasks by linking them. When you link tasks, you can see how a change in the duration of one task affects the start and finish dates of other tasks and the project finish date.
- Assign personnel and other resources to tasks.
- See how tasks progress across time. You can track progress by comparing planned and actual start and finish dates and by checking the completion percentage of each task.
- View tasks graphically while still having access to detailed information about the tasks.
- Split a task so that the task is interrupted and then resumes later in the schedule.

The sheet portion displays categories of information about the tasks, arranged in tables. The default table for the Gantt Chart view is Entry, but you can select any of the task tables to display information about the project's tasks. The Gantt Chart view can use the following tables:

| | | |
|---|---|---|
| Baseline | Hyperlink | Schedule |
| Constraint Dates | PA_Expected Case | Summary |
| Cost | PA_Optimistic Case | Tracking |
| Delay | PA_PERT Entry | Usage |
| Earned Value | PA_Pessimistic Case | Variance |
| Entry | Rollup Table | Work |
| Export | | |

The Gantt Chart view can use any of the task filters to display only the tasks you want to see. When you select a filter, both the sheet and chart portions of the Gantt Chart display the information as defined by that filter's criteria. The default filter is All Tasks.

The Gantt Chart view can use the following filters:

| | | |
|---|---|---|
| All Tasks | Linked Fields | Tasks With Deadlines. |
| Completed Tasks | Milestones | Tasks With Estimated Duration |
| Confirmed | Resource Group | Tasks With Fixed Dates |
| Cost Greater Than | Should Start By | Tasks With Overtime |

| | | |
|---|---|---|
| Cost Over budget | Should Finish By | Top Level Tasks |
| Created After | Slipped/Late Progress | Unconfirmed |
| Critical Slipping | Tasks | Unstarted Tasks |
| Date Range | Summary Tasks | Update Needed |
| In Progress Tasks | Task Range | Using Resource In Date Range |
| Incomplete Tasks | Tasks With a Task Calendar Assigned | Using Resource |
| Late/Over budget | Tasks Assigned To | Tasks With Attachments |
| Work Over budget | | |

- To see more tasks on the timescale of the Gantt Chart view, click Zoom Out
- To insert a new task in a sheet view, select the task above which you want to insert the new task, and then click New Task on the Insert menu. Tasks are renumbered automatically after you insert a task.
- To link specific tasks using finish-to-start (FS) task links, select the tasks you want to link, and then click Link Tasks

With Microsoft Project, you can display project information in task views or in resource views. Use a task view when you want to enter, change, or display task information. Use a resource view when you want to enter, change, or display resource information.

The following table contains all of the Microsoft Project views, including those traditionally used by project management professionals (such as the Gantt Chart), as well as specialized formats for entering and displaying data. You can work with the predefined views or create your own special-purpose views.

| **View** | **Description** |
|---|---|
| Calendar | A monthly calendar showing tasks and durations. Use this task view to show the tasks scheduled in a specific week or range of weeks. |
| Detail Gantt | A list of tasks and related information and a chart showing slack and slippage. Use this task view to check how far a task can slip without affecting other tasks. |
| Gantt Chart | A list of tasks and related information and a chart showing tasks and durations over time. Use this task view to enter and schedule a list of tasks. |
| Leveling Gantt | A list of tasks, information about task delays and slack and a bar chart showing the before and after effects of leveling. Use this task view to check the amount of task delay. |
| Network Diagram | A network diagram showing all tasks and task dependencies. Use this task view to create and fine-tune your schedule in a flowchart format. |
| PA_Expected Gantt | A Gantt Chart showing your schedule's expected-case scenario. Use this view when performing a PERT analysis on a task's duration or on your project's schedule. |
| PA_Optimistic Gantt | A Gantt Chart showing your schedule's best-case scenario. Use this view when performing a PERT |

| | |
|---|---|
| | analysis on a task's duration or on your project's schedule. |
| PA_Pessimistic Gantt | A Gantt Chart showing your schedule's worse case scenario. Use this view when performing a PERT analysis on a task's duration or on your project's schedule. |
| Tracking Gantt | A list of tasks and related information, and a chart showing baseline and scheduled Gantt bars for each task. Use this task view to compare the baseline schedule with the actual schedule. |