Module - 1

SPM

What is a project?

Some dictionary definitions:

"A specific plan or design"

"A planned undertaking"

"A large undertaking e.g. a public works scheme"

Longmans dictionary

Key points above are ***planning*** and ***size*** of task

Jobs versus projects

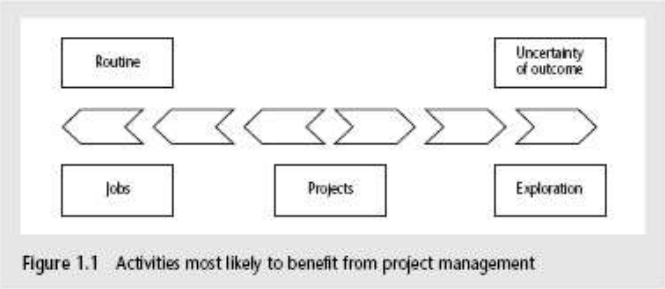


Figure 1.1 Activities most likely to benefit from project management

'Jobs' – repetition of very well-defined and well understood tasks with very little uncertainty

'Exploration' – e.g. finding a cure for cancer: the outcome is very uncertain

'Projects' – in the middle!

Characteristics of projects

A task is more 'project-like' if it is:

- Non-routine
- Planned
- Aiming at a specific target
- Work carried out for a customer
- Involving several specialisms
- Made up of several different phases
- Constrained by time and resources
- Large and/or complex

Are *software* projects really different from other projects?

Not really! …but…

- Invisibility
- Complexity
- Conformity
- Flexibility

make software more problematic to build than other engineered artefacts.

Activities covered by project management

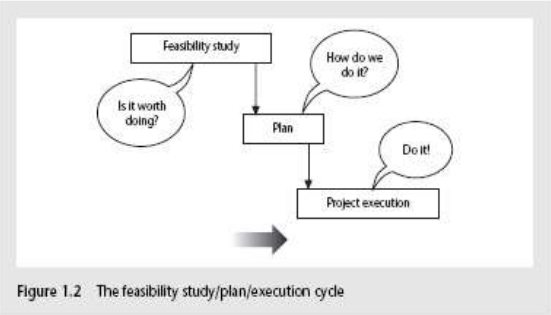


Figure 1.2 The feasibility study/plan/execution cycle

Feasibility study

Is project technically feasible and worthwhile from a business point of view?

Planning

Only done if project is feasible

Execution

Implement plan, but plan may be changed as we go along

The software development life-cycle (ISO 12207)

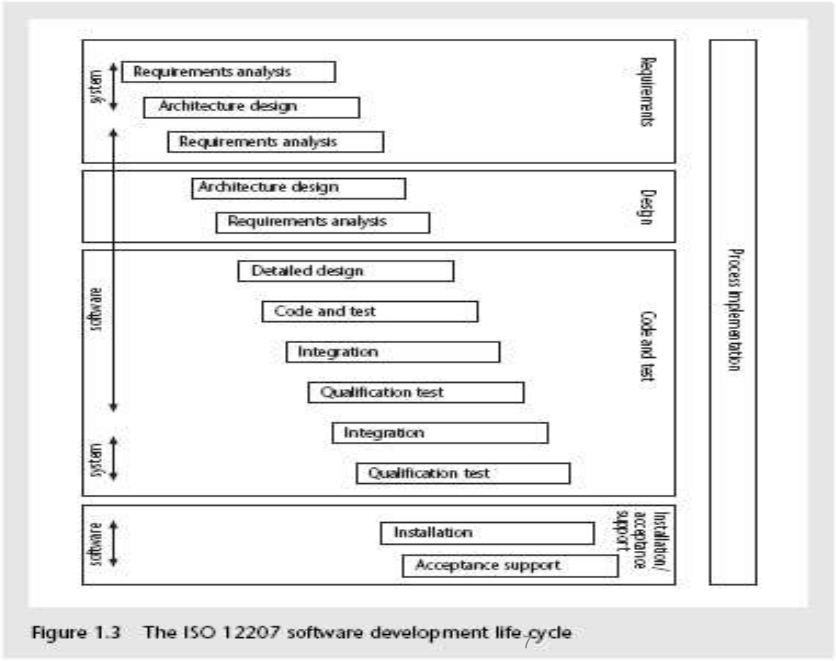


Figure 1.3 The ISO 12207 software development life-cycle

ISO 12207 life-cycle

Requirements analysis

- Requirements elicitation: what does the client need?
- Analysis: converting 'customer-facing' requirements into equivalents that developers can understand
- Requirements will cover
  - Functions
  - Quality
  - Resource constraints i.e. costs

ISO 12207 life-cycle

- Architecture design
  - Based on *system requirements*
  - Defines components of system: hardware, software, organizational
  - *Software requirements* will come out of this
- Code and test
  - Of individual components
- Integration
  - Putting the components together

ISO12207 continued

- Qualification testing
  - Testing the *system* (not just the *software*)
- Installation
  - The process of making the system operational
  - Includes setting up standing data, setting system parameters, installing on operational hardware platforms, user training etc
- Acceptance support
  - Including maintenance and enhancement

Some ways of categorizing projects

Distinguishing different types of project is important as different types of task need different project approaches e.g.

- Information systems versus embedded systems
- Objective-based versus product-based

What is management?

This involves the following eight management responsibility activities:

1. Planning – deciding what is to be done
2. Organizing – making arrangements
3. Staffing – selecting the right people for the job
4. Directing – giving instructions

continued…

What is management? (continued)

5. Monitoring – checking on progress
6. Controlling – taking action to remedy hold-ups
7. Innovating – coming up with solutions when problems emerge
8. Representing – liaising with clients, users, developers and other stakeholders

Setting objectives

- Answering the question '*What do we have to do to have a success?'*
- Need for a *project authority*
  - Sets the project scope
  - Allocates/approves costs
- Could be one person - or a group
  - Project Board
  - Project Management Board
  - Steering committee

Objectives

Informally, the objective of a project can be defined by completing the statement:

The project will be regarded as a success if……..

Rather like *post-conditions* for the project

Focus on *what* will be put in place, rather than *how* activities will be carried out

Objectives should be SMART

S – specific, that is, concrete and well-defined

M – measurable, that is, satisfaction of the objective can be objectively judged

A – achievable, that is, it is within the power of the individual or group concerned to meet the target

R – relevant, the objective must relevant to the true purpose of the project

T – time constrained: there is defined point in time by which the objective should be achieved

Goals/sub-objectives

These are steps along the way to achieving the objective. Informally, these can be defined by completing the sentence…

Objective X will be achieved
IF the following goals are all achieved
A……
B……
C…… etc

Goals/sub-objectives continued

Often a goal can be allocated to an individual.

Individual may have the capability of achieving goal, but not the objective on their own e.g.

Objective – user satisfaction with software product

Analyst goal – accurate requirements

Developer goal – software that is reliable

Measures of effectiveness

How do we know that the goal or objective has been achieved?

By a practical test, that can be objectively assessed.

e.g. for user satisfaction with software product:

- Repeat business – they buy further products from us
- Number of complaints – if low etc etc

Stakeholders

These are people who have a stake or interest in the project

In general, they could be *users/clients* or *developers/implementers*

They could be:

- Within the project team
- Outside the project team, but within the same organization
- Outside both the project team and the organization

The business case

Benefits of delivered project must outweigh costs

Costs include:

- Development
- Operation

Benefits

- Quantifiable
- Non-quantifiable

Management control

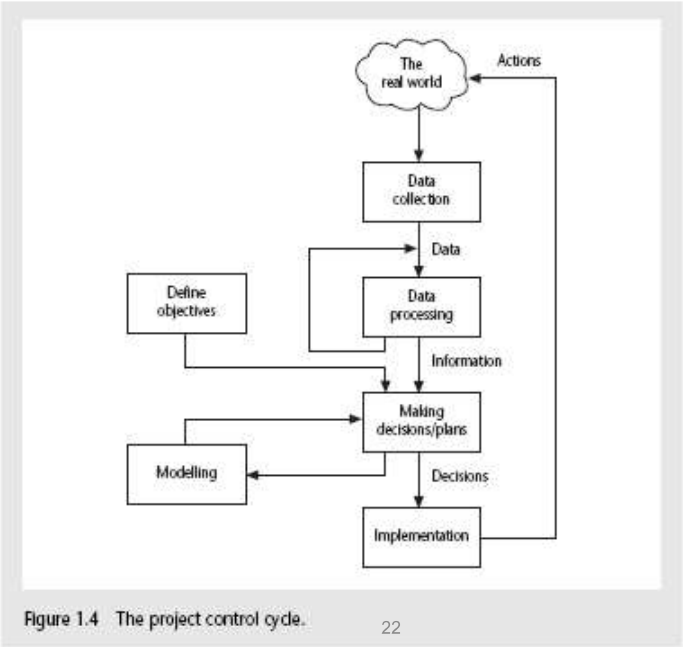


Figure 1.4 The project control cycle.

Management control

Data – the raw details

e.g. *'6,000 documents processed at location X'*

Information – the data is processed to produce something that is meaningful and useful

e.g. 'productivity is 100 documents a day'

Comparison with objectives/goals

e.g. we will not meet target of processing all documents by 31st March

continued.....

Management control - continued

Modelling – working out the probable outcomes of various decisions

e.g. if we employ two more staff at location X how quickly can we get the documents processed?

Implementation – carrying out the remedial actions that have been decided upon

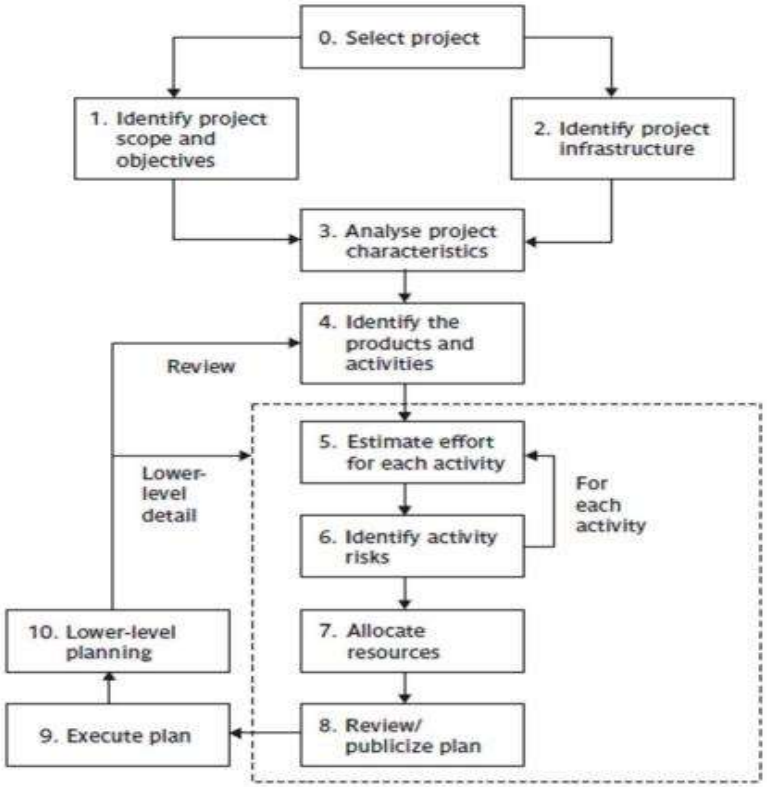
Key points in lecture

- Projects are non-routine - thus uncertain
- The particular problems of projects e.g. lack of visibility
- Clear objectives are essential which can be objectively assessed
- Stuff happens. Not usually possible to keep precisely plan – need for control
- Communicate, communicate, communicate!

Module 1.2

SPM

Step Wise Project Planning



'Step Wise' - an overview

A project scenario

- Hardware/software engineering company (C++ language of choice)
- teams are selected for individual projects - some friction has been found between team members
- HR manager suggests psychometric testing to select team

Project scenario - continued

- Software package to be used to test staff
- Visual basic suggested as a vehicle for implementation
- usability is important - decision to carry out usability tests

Step 1 - Identify project scope and objectives

- 1.1 Identify objectives and measures of effectiveness
  - 'how do we know if we have succeeded?'
- 1.2 Establish a project authority
  - 'who is the boss?'
  - A single overall project authority needs to be established so that there is unity of purpose among all those concerned.
- 1.3 Identify all stakeholders in the project and their interests
  - 'who will be affected/involved in the project?'

Step 1 continued

- 1.4 Modify objectives in the light of stakeholder analysis
  - 'do we need to do things to win over stakeholders?'
  - potentially dangerous because the system size might be increased and the original objectives obscured, so be done consciously and in a controlled manner.
- 1.5 Establish methods of communication with all parties
  - 'how do we keep in contact?'
  - With internal staff and with the client (point of contact)

Back to the scenario

- Project authority
  - should be a project manager rather than HR manager?
- Stakeholders
  - project team members to complete on-line questionnaires: concern about results?
- Revision to objectives
  - provide feedback to team members on results

Step 2 Identify project infrastructure

- 2.1 Establish link between project and any strategic plan
  - 'why did they want the project?'
  - Order in which the projects are to be carried out
  - Establish framework within which the proposed new system are to fit.
  - H/W and S/W standards needed

Step 2 Identify project infrastructure

- 2.2 Identify installation standards and procedures
  - 'what standards do we have to follow?'
  - 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.
  - Monitoring and control policy of organization must have measurement programme that dictates that certain statistics have to be collected at various stage of project.

Step 2 Identify project infrastructure

- 2.3. Identify project team organization
  - 'where do I fit in?'
  - Project leaders should often have some control over the organizational structure of the project team.

Step 3 Analysis of project characteristics

- 3.1 Distinguish the project as either objective or product-based.
- 3.2 Analyse other project characteristics (including quality based ones)
  - what is different about this project?
- 3.3 Identify high level project risks
  - 'what could go wrong?'
  - 'what can we do to stop it?'

Step 3 continued

- 3.4 Take into account user requirements concerning implementation
  - The clients will usually have their own procedural requirements
- 3.5 Select general life cycle approach
  - waterfall? Increments? Prototypes?
- 3.6 Review overall resource estimates
  - 'does all this increase the cost?'
  - once the major risk have been identified and the broad project approach has been decided upon, this would be a good point at which to re-estimate the effort and other resources required to implement the project.

Back to the scenario

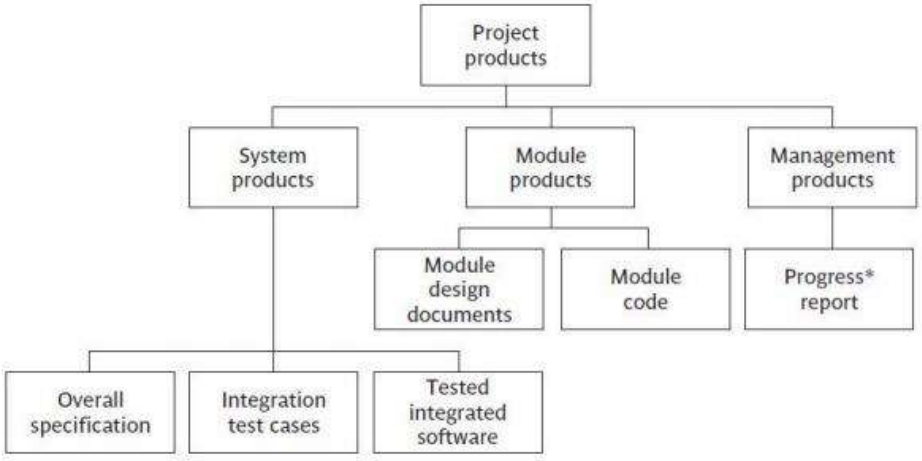
- Objectives vs. products
- Some risks
  - team members worried about implications and do no co-operate
  - project managers unwilling to try out application
  - Developer not familiar with features of VB
- Answer? - evolutionary prototype?

Step 4 - Identify project products and activities

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

Step 4

4.1 Identify and describe project products (or deliverables) - 'what do we have to produce?'



Product Breakdown Structure (PBS) for a system development task

Products

- The result of an activity
- Could be (among other things)
  - physical thing ('installed pc'),
  - a document ('logical data structure')
  - a person ('trained user')
  - a new version of an old product ('updated software')

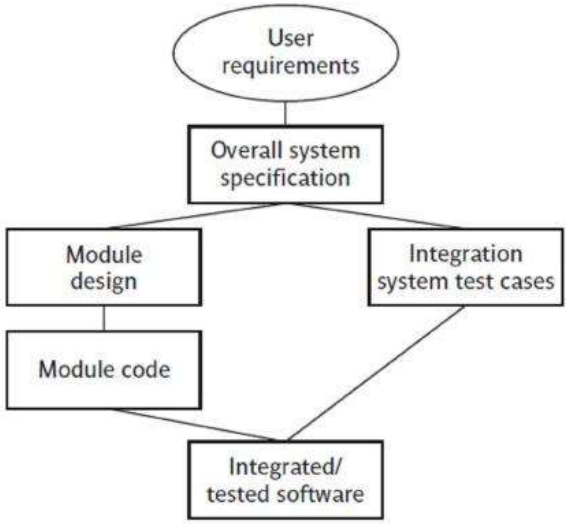
Products

- The following are **NOT** normally products:
  - activities (e.g. 'training')
  - events (e.g. 'interviews completed')
  - resources and actors (e.g. 'software developer') - may be exceptions to this
- Products CAN BE *deliverable* or *intermediate*

Product Description (PD)

- 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

4.2. Document generic product flows

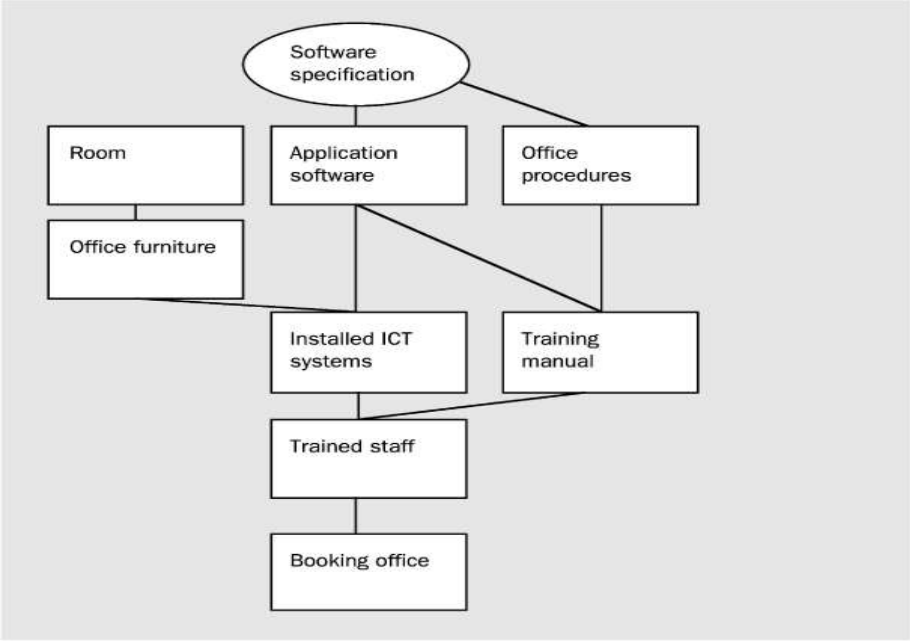


A fragment of a Product Flow Diagram (PFD) for a software development task

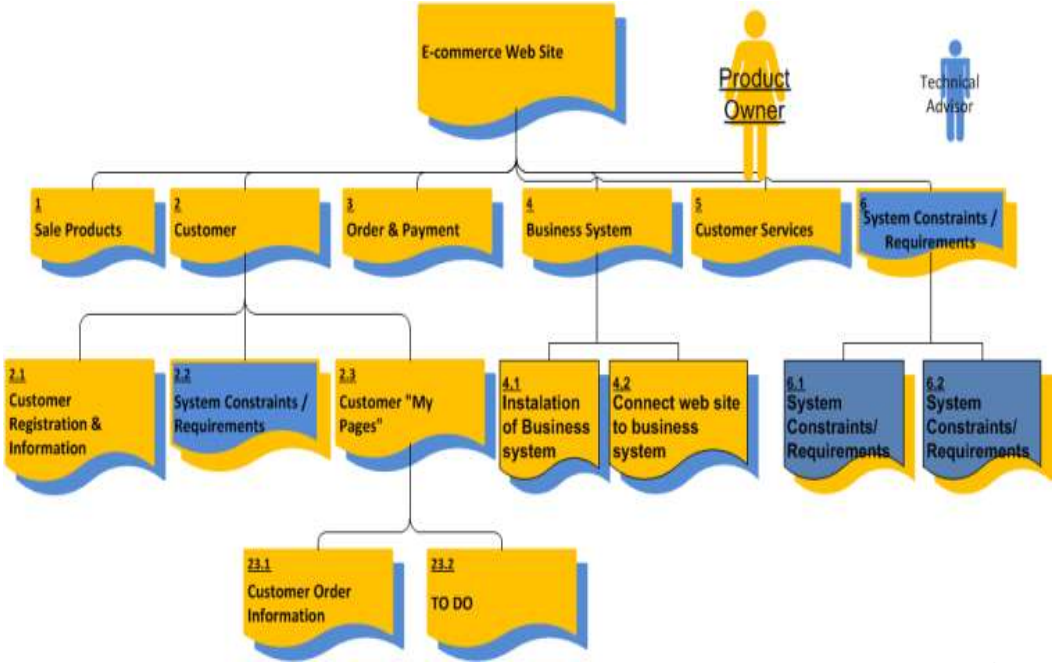
PFD..

- The Product flow diagram (PFD) is a graphical representation of the order by which a sequence of products is created according to Product based planning principles.
- It is related to the Product breakdown structure (PBS).
- A product flow diagram (PFD) showing the order in which products have to be created.
- This should be relatively easy to draft if you have already produced product descriptions that specify from which other products each product is derived.

PFD More Ex..



PFD More Ex..



Step 4.3 Recognize product instances

- The PBS (**Product Breakdown Structure)** and PFD will probably have identified generic products e.g. ‘software modules’
- It might be possible to identify specific instances e.g. ‘module A’, ‘module B’ …
- But in many cases this will have to be left to later, more detailed, planning

Cont..

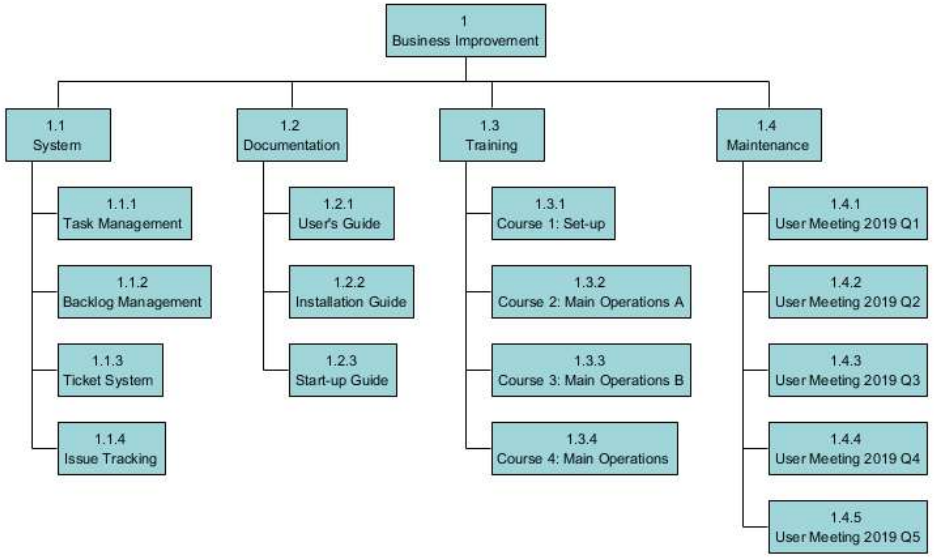
- **Product Breakdown Structure (PBS)**
  - A Product Breakdown Structure (PBS) is a product based planning method used to analyze, document and communicate the outcomes of a project.
  - A product breakdown structure is an effective tool that details the physical components of a particular product, or system, under consideration.
  - A formal PBS comes in the form of a hierarchy.
  - Through this hierarchy, project planners can see a clear relationship between the different levels of a system under consideration.
  - The project tasks are graphically represented in the form of lists or organigram (tree structure) in the product breakdown structure.
  - This plan displays product components and the responsibilities for the delivery of each component.
  - The product components are represented by boxes in the tree structure.

PBS..

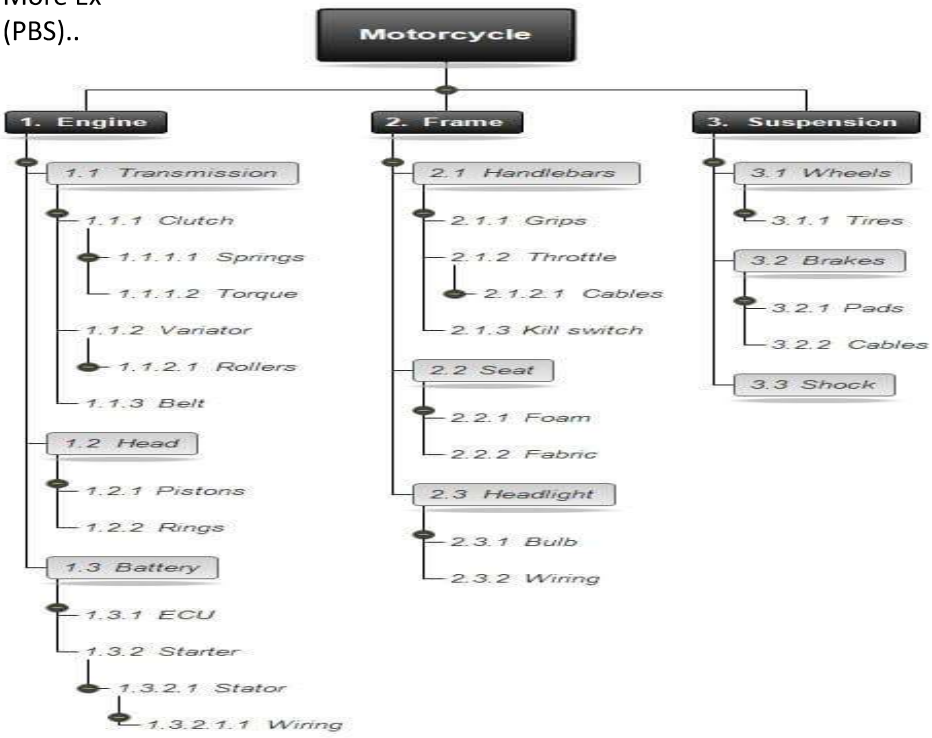
- **Steps to identify products**

  The Product Development Meeting should comprise a series of steps:
  - People think about products and write each product;
  - Consolidate the products into a list from the people in the meeting;
  - Review the list and remove duplicates;
  - Review the remaining products in the list and put them in related groups;
  - Develop the product breakdown structure diagram
  - Repeat the steps as necessary until everyone feels that all the products have been identified.

More Ex (PBS)..



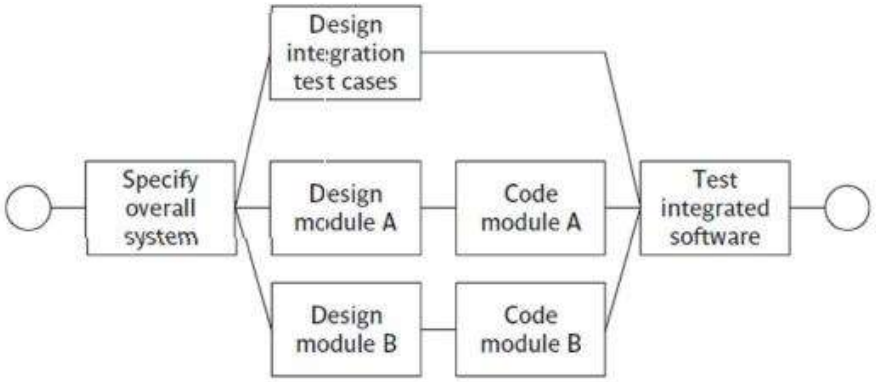
More Ex
(PBS)..



4.4. Produce ideal activity network

- Identify the activities needed to create each product in the PFD
- More than one activity might be needed to create a single product
- Hint: Identify activities by verb + noun but avoid ‘produce…’ (too vague)
- Draw up activity network

An ‘ideal’ activity

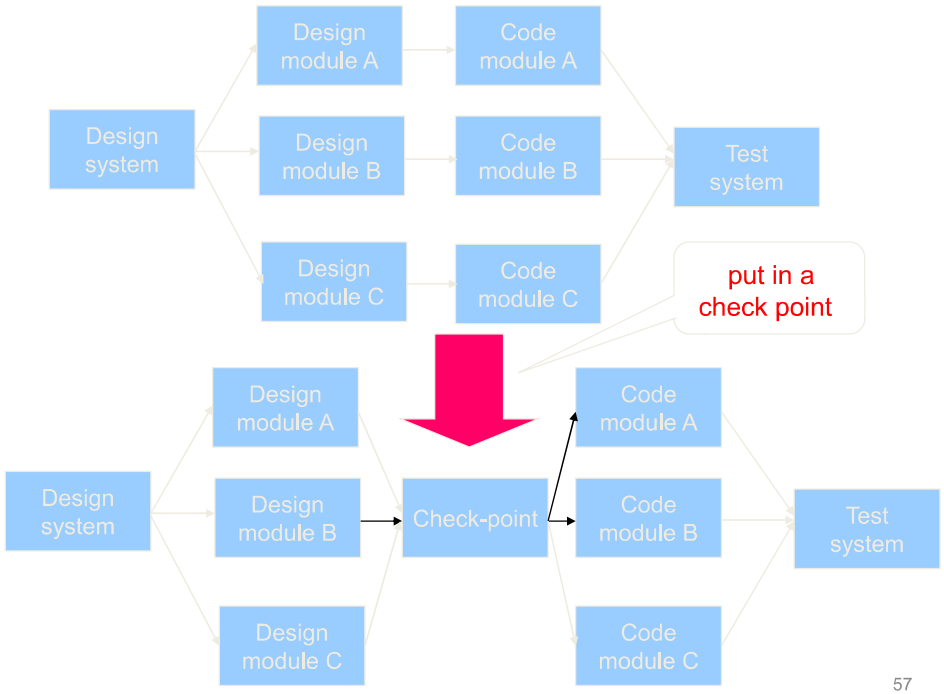


Example

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

- Assumption of ideal activity network:
  - an activity will start as soon as the preceding ones upon which it depends have been completed.
- But we need to divide the project into stages and introducing checkpoint activities
  - to check that products of preceding activities are compatible.
- Milestones represent the completion of important stages of the project of which managers would want to take particular note.
  - Checkpoint activities are often useful milestones.

Step 4.5 Add check-points if needed



Step 5:Estimate effort for each activity

- 5.1 Carry out bottom-up estimates
  - distinguish carefully between *effort* and *elapsed* time
  - The individual activity estimates of effort should be summed to get an overall bottom-up estimate.
- 5.2. Revise plan to create controllable activities
  - Long activities often make a project difficult to control.
  - break up very long activities into a series of smaller ones
  - bundle up very short activities (create check lists?)

Step 6: Identify activity risks

- 6.1.Identify and quantify risks for activities
  - damage if risk occurs (measure in time lost or money)
  - likelihood if risk occurring
- 6.2. Plan risk reduction and contingency measures
  - risk reduction: activity to stop risk occurring
  - It is possible to avoid or at least reduce some of the identified risks.
  - Contingency: action if risk does occur
  - Contingency plan specify action that is to be taken if a risk materializes.

- 6.3 Adjust overall plans and estimates to take account of risks
  - We can change our plans by adding new activities which reduce risks.
  - e.g. add new activities which reduce risks associated with other activities e.g. training, pilot trials, information gathering

The most important risks that can affect the success of a software project:

- Estimation and scheduling.
- Sudden growth in requirements.
- Employee turnover.
- Breakdown of specification.
- Productivity issues.
- Compromising on designs.
- Gold plating.
- Procedural risks.
- Technical risks.
- Unavoidable risks.

Cont..

- Estimation and scheduling.
  - The unique nature of individual software projects creates problems for developers and managers in estimating and scheduling development time.
- Sudden growth in requirements.
  - As a project progresses, issues that are not identified earlier can create a last-minute hurdle to meeting deadlines.
- Employee turnover.
  - Every project has a number of developers working on it. When a developers leaves, he or she may take critical information with him/her. This can delay, and sometimes derail an entire project.
- Breakdown of specification.
  - During the initial phases of integration and coding, requirements might conflict. Moreover, developers may find that even the specification is unclear or incomplete.

Cont..

- Productivity issues.
  - On projects involving long timelines, developers tend to take things easy to begin with. As a result, sometimes, they lose significant time to complete the project. Set a realistic schedule, and stick to it.
- Compromising on designs.
  - In order to get stuck into the next 'real' tasks, developers tend to rush the design-process. This is a waste of programming hours, as designing is the most critical part of software development.
- Gold plating.
  - Developers sometimes like to show off their skills by adding unnecessary features. For instance, a developer might add Flash to a basic login module to make it look 'stylish'. Again, this is a waste of programming hours.

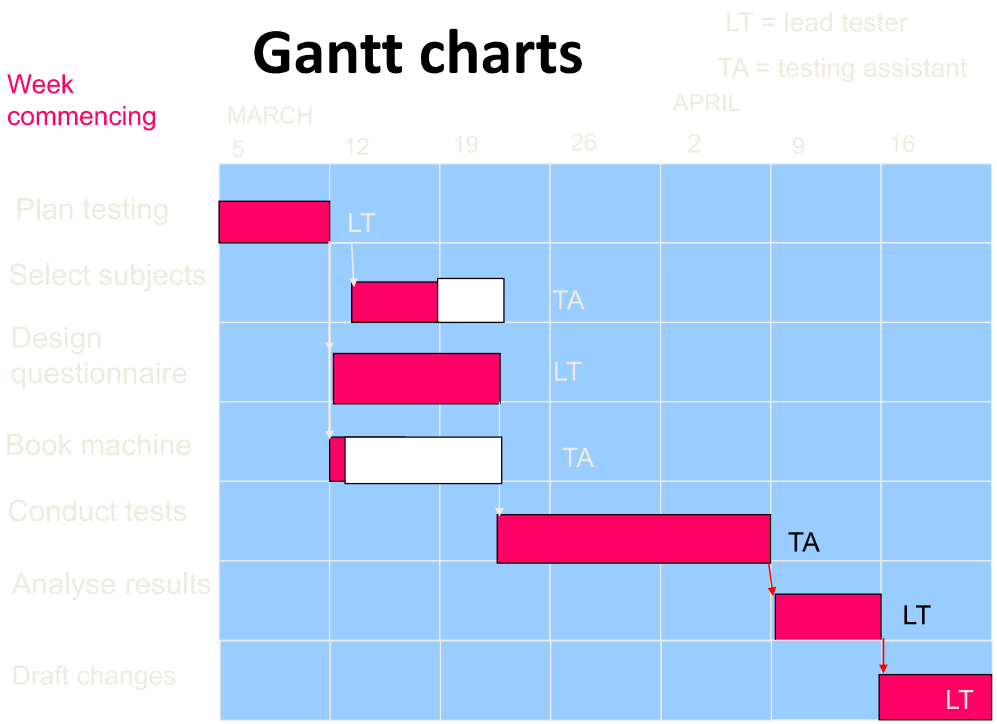
Cont..

- Procedural risks.
  - Day-to-day operational activities might hamper due to improper process implementation, conflicting priorities, or a lack of clarity in responsibilities.
- Technical risks.
  - Sometimes software development firms reduce the functionality of the software to compensate for overruns pertaining to high budgets and scheduling. There is always a conflict between achieving maximum functionality of the software and peak performance. In order to compensate for excessive budget and schedule overruns, companies sometimes reduce the functionality of the software.
- Unavoidable risks.
  - These include changes in government policy, the obsolescence of software or other risks that cannot be controlled or estimated.

Step 7: Allocate resources

- 7.1 Identify and allocate resources to activities
  - The type of staff needed for each activity is recorded.
  - The staff available for the project are identified and are provisionally allocated to tasks.
- 7.2 Revise plans and estimates to take into account resource constraints
  - e.g. staff not being available until a later date
  - non-project activities

Gantt charts



Step 8: Review/publicise plan

- 8.1 Review quality aspects of project plan
- 8.2 Document plan and obtain agreement

Step 9 and 10: Execute plan and create lower level plans