



# Software Project Management

Durga Prasad Mohapatra

Professor

CSE Deptt.

NIT Rourkela



# **A step wise approach to planning software projects**

## ‘Step Wise’ - aspirations

- Students, and others, were often at a loss as to where to start when new to project planning. A structured approach was seen as catering for the needs of such people.
- The approach described here is designed to be applicable to a range of different types of project.

# ‘Step Wise’ - aspirations

- It might be asked why a standard approach such as PRINCE2 has not been adopted. There has been caution about using PRINCE2 more centrally because:
  - ❖ PRINCE2 tend to be used mainly in the UK
  - ❖ PRINCE2 tends to focus more on procedural and bureaucratic matters at the expense of techniques – the few planning techniques that are associated with PRINCE, for example, the development of product flow diagrams, are used in the Step Wise approach as well.

```
graph TD; A[0. Select project] --> B[1. Identify project objectives]; A --> C[2. Identify project infrastructure]; B --> D[3. Analyse project characteristics]; C --> D; D --> E[4. Identify products and activities]; E --> F[5. Estimate effort for activity]; F --> G[6. Identify activity risks]; G --> H[7. Allocate resources]; H --> I[8. Review/ publicize plan]; I --> J[9. Execute plan]; J --> K[10. Lower level planning]; K -- "Review" --> E; K -- "Lower level detail" --> F; L["For each activity"] --- F; L --- G;
```

The flowchart illustrates the project planning process, starting with selecting a project and identifying its objectives and infrastructure. It then moves through analyzing project characteristics, identifying products and activities, estimating effort, identifying risks, allocating resources, reviewing/publicizing the plan, executing the plan, and finally lower-level planning. The process includes feedback loops for review and lower-level detail, and a loop for each activity.

# 'Step Wise' - an overview

**0. Select project** There must be some process by which the project to be executed was selected.

**1. Identify project objectives** It is important that at the outset the main stakeholders are all aware of the precise objectives of the project.

# 'Step Wise' - an overview

## **2. Identify project infrastructure**

- This may not be a significant step where you are working on an in-house project in a very familiar environment.
- However, where the project is being carried out for external clients then you may need to investigate the characteristics of the environment in which the project is to be carried out.

# 'Step Wise' - an overview

## **3. Analyse project characteristics**

- Different types of projects will need different technical and management approaches.
- For example,
  - ❖ A project to implement control software embedded in industrial equipment will need a different set of methods than a project to implement a business information system.
  - ❖ A multimedia application would again need a different set of activities.



# 'Step Wise' - an overview

## **4. Identify products and activities**

- With software projects, it is best to start by listing the products, both deliverable and intermediate, to be created.
- The activities needed to create the products can then be identified

## **5. Estimate effort for activity.**

- After identifying the activities, estimate the effort that will be required for carrying the activities.

# 'Step Wise' - an overview

## **6. Identify activity risks**

- Having assessed the amount of effort and the elapsed time for a project, the reasons why these might be vary during the actual execution of the project need to be considered.
- Where there is a very high risk of additional effort/time being needed, then actions to reduce this risk may be formulated.

## **7. Allocate resources**

- With software projects, these resources will mainly be staff, but could be equipment etc.

# 'Step Wise' - an overview

## **8. Review/publicize plan**

- It is no good having a plan if no one knows about it.
- So review and publicize the proposed plan.

## **9. Execute Plan**

- Execute the proposed plan.

# 'Step Wise' - an overview

## **10. Lower level planning**

- Not all of a project, especially when it is large, can be planned in detail at the outset.
- Not all the information needed to plan the later stages will be available at the beginning: for example software development cannot be broken down into precise sub-tasks with realistic target times until more is known about what the overall design of the system is known.
- So, plan later at the lower level / in detail.



# A project scenario: Brightmouth College Payroll

- College currently has payroll processing carried out by a services company
- This is very expensive and does not allow detailed analysis of personnel data to be carried out
- Decision made to bring payroll 'in-house' by acquiring an 'off-the-shelf' application

# Project scenario - continued

- The use of the off-the-shelf system will require a new, internal, payroll office to be set up
- There will be a need to develop some software 'add-ons': one will take payroll data and combine it with time-table data to calculate the staff costs for each course run in the college
- The project manager is Brigitte.



# Step 1 establish 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?’
- 1.3 Identify all stakeholders in the project and their interests
  - ‘who will be affected/involved in the project?’

# Step I continued

- 1.4 Modify objectives in the light of stakeholder analysis
  - ‘do we need to do things to win over stakeholders?’
- 1.5 Establish methods of communication with all parties
  - ‘how do we keep in contact?’



# Back to the scenario

- Project authority
  - Brigitte finds she has two different clients for the new system: the finance department and the personnel office. A vice principal agrees to be official client, and monthly meetings are chaired by the VP and attended by Brigitte and the heads of finance and personnel
  - These meetings would also help overcome communication barriers

# Back to the scenario

- Stakeholders/revision to objectives
  - The application will not ultimately be a success if project team members are not happy to use the system.
  - They might be happier to use the testing system if the results of their own tests were automatically notified to them personally by the software application, so that this might have to be added as a requirement for the project.

# Back to the scenario - continued

- Stakeholders
  - For example, personnel office would supply details of new staff, leavers and changes (e.g. promotions)
  - To motivate co-operation Brigitte might ensure new payroll system produces reports that are useful to personnel staff



## Step 2 Establish project infrastructure

- 2.1 Establish link between project and any strategic plan
  - ‘why did they want the project?’
- 2.2 Identify installation standards and procedures
  - ‘what standards do we have to follow?’
- 2.3. Identify project team organization
  - ‘where do I fit in?’

## Step 2 Establish project infrastructure

- At the same time as establishing exactly what the project objectives are, the person responsible may know little about the organizational environment in which the application is to be developed and implemented. The actions in Step 2 address this problem.

## Step 3 Analysis of project characteristics

- 3.1 Distinguish the project as either objective or product-based.
  - Is there more than one way of achieving success?
- 3.2 Analyse other project characteristics (including quality based ones)
  - what is different about this project?

# Step 3 Analysis of project characteristics

- 3.1 Objective-based versus product-based projects.
  - ❖ With a product-based project the developers have to create a product, the specification of which is often (but not always) clearly defined.
  - ❖ In an objective-based project, a problem is defined that needs to be solved but there could be more than one solution. For example, if an organization needed a payroll application they might consider (a) writing the system themselves (b) using a service company to do the payroll for them (c) acquire an off-the-shelf package.

## Step 3 Analysis of project characteristics

- 3.2 Analyse other project characteristics – such as is it an information system or an embedded real time or a multimedia application? Is it safety-critical? etc.
  - The payroll application is clearly an information system.



# Step 3 continued

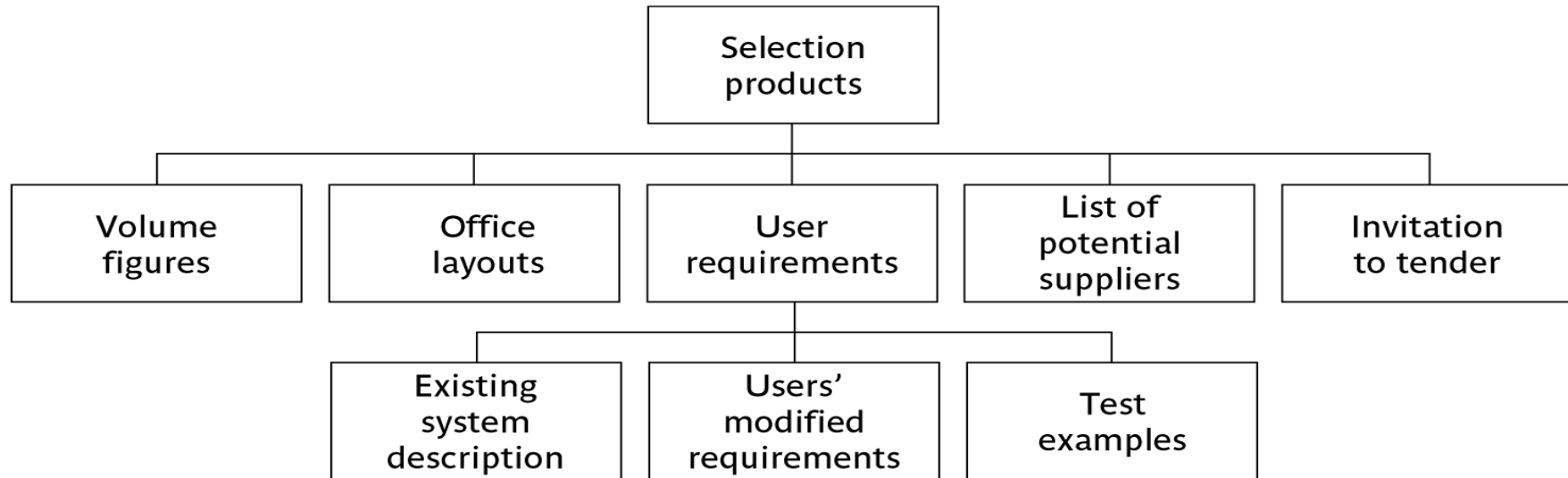
- 3.3 Identify high level project risks
  - ‘what could go wrong?’
  - ‘what can we do to stop it?’
- 3.4 Take into account user requirements concerning implementation
- 3.5 Select general life cycle approach
  - waterfall? Increments? Prototypes?
- 3.6 Review overall resource estimates
  - ‘does all this increase the cost?’

# Back to the scenario

- Objectives vs. products
  - An objective-based approach has been adopted
- Some risks
  - There may not be an off-the-shelf package that caters for the way payroll is processed at Brightmouth College
- Answer?
  - Brigitte decides to obtain details of how main candidate packages work as soon as possible; also agreement that if necessary processes will be changed to fit in with new system.

# Step 4 Identify project products and activities

- 4.1 Identify and describe project products - 'what do we have to produce?'



# Products

- The result of an activity: intermediate / final product to be delivered
- 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*

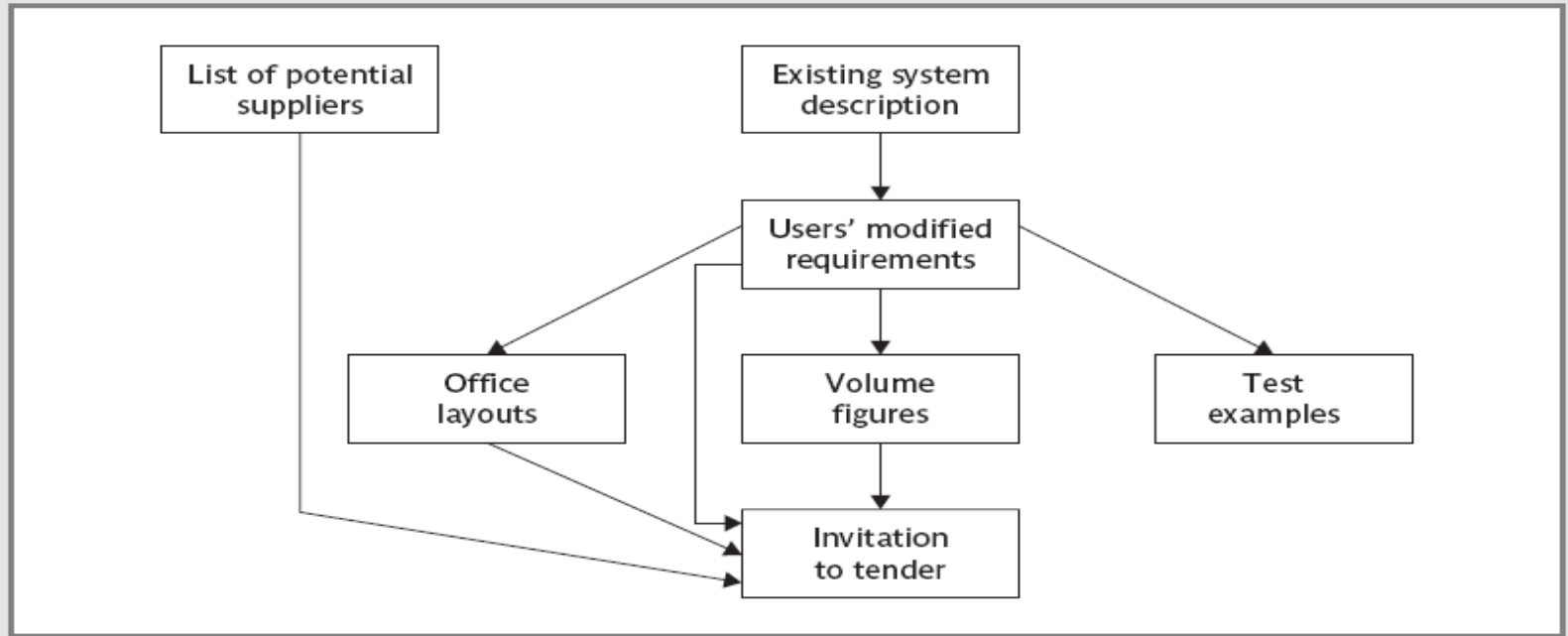
# Product description (PD)

- Product identity
- Description - what is it?
- Derivation - what is it based on?
- Composition - what does it contain?
- Format
- Relevant standards
- Quality criteria

*Create a PD for 'test data'*

# Step 4 continued

- 4.2 document generic product flows



**FIGURE B.1** Product Flow Diagram for the creation of an 'invitation to tender'

## Step 4 continued

- The product flow diagram shows the order in which the products have to be completed. Effectively it defines a method of working. The example above is a possible solution to Exercise 3.3 in the textbook.
- The flow of the PFD is generally from top to bottom and left to right. We do not put in lines which loop back. This is not because iterative and back-tracking is not accepted. Rather it is that you can in theory jump back to any preceding product.



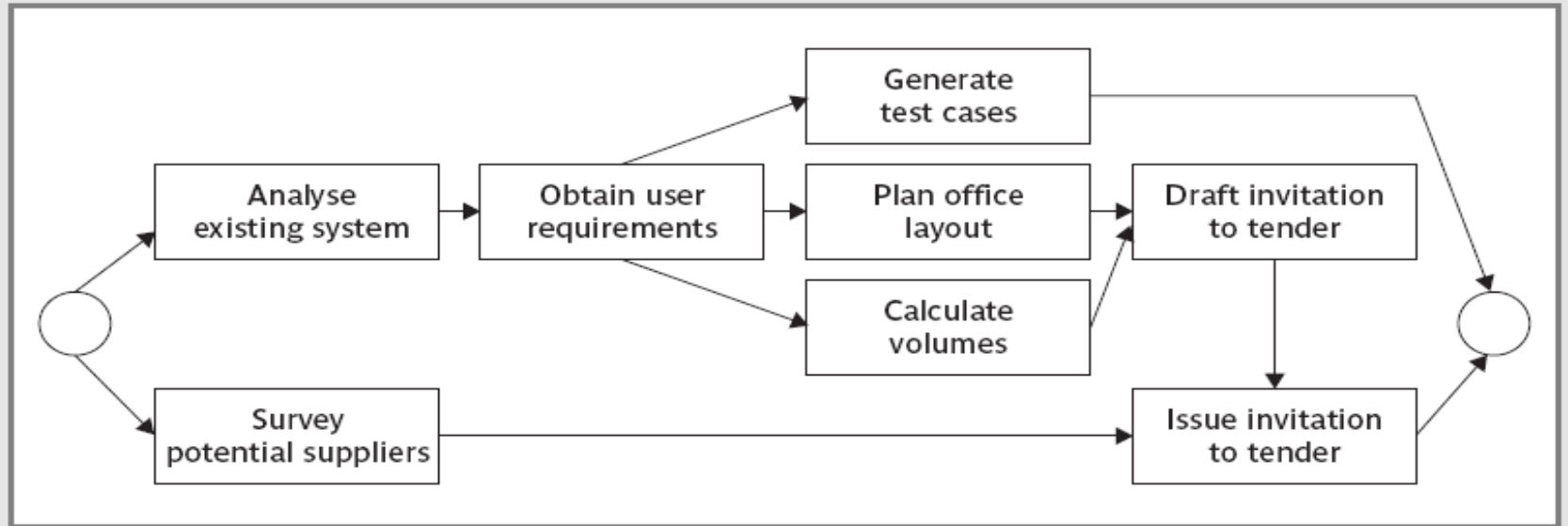
## Step 4.3 Recognize product instances

- The PBS 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

## 4.4. Produce ideal activity network

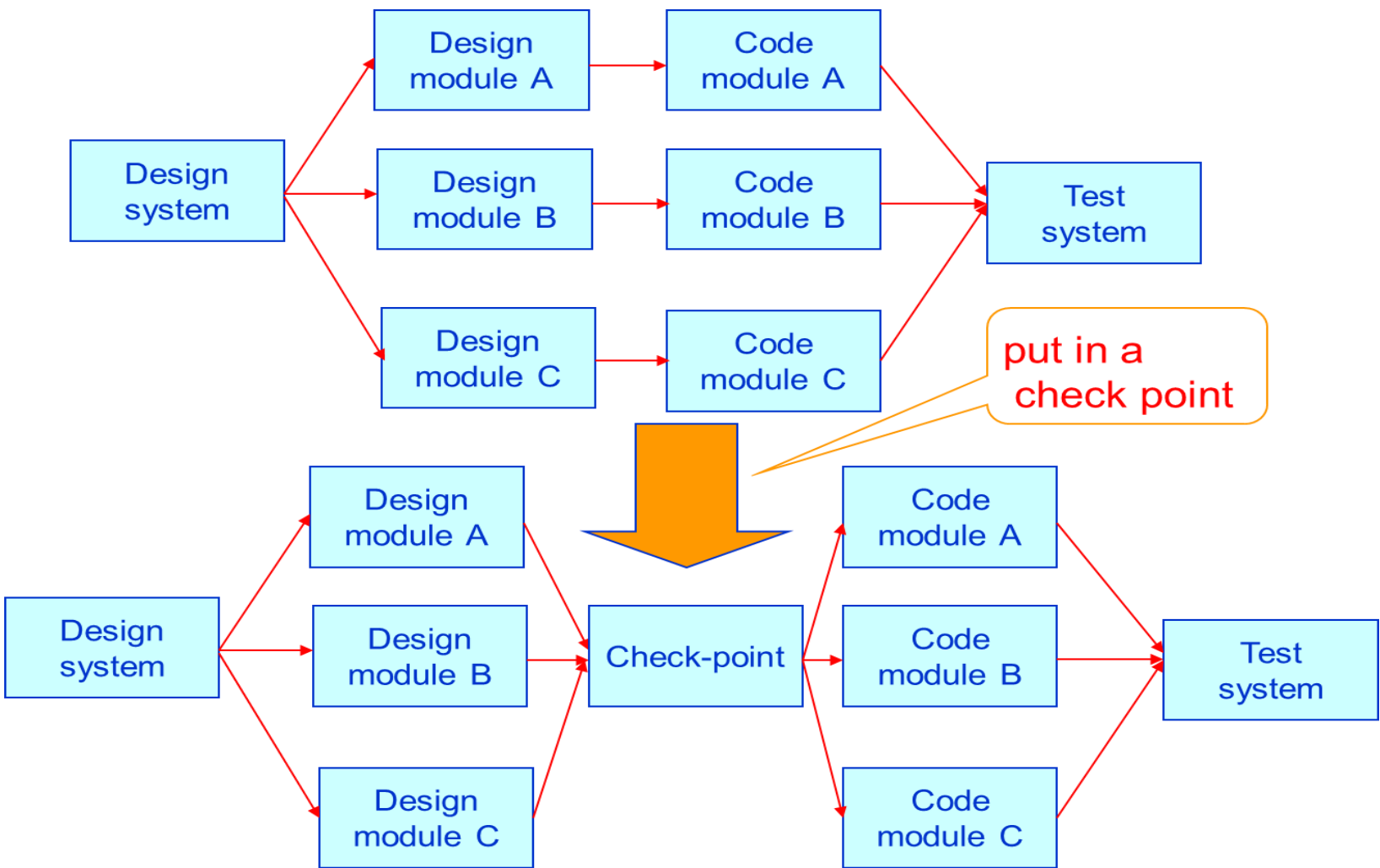
- Identify the activities needed to create each product in the PFD
- More than one activity might be needed to create a single product
- Hint: Identify activities by verb + noun but avoid 'produce...' (too vague)
- Draw up activity network

# An 'ideal' activity



**FIGURE B.2** Brightmouth College payroll project activity network fragment

# 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
- 5.2. Revise plan to create controllable activities
  - 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
  - contingency: action if risk does occur
- 6.3 Adjust overall plans & estimates to take account of risks
  - For example, add new activities which reduce risks associated with other activities e.g. training, pilot trials, information gathering

# Step 7: Allocate resources

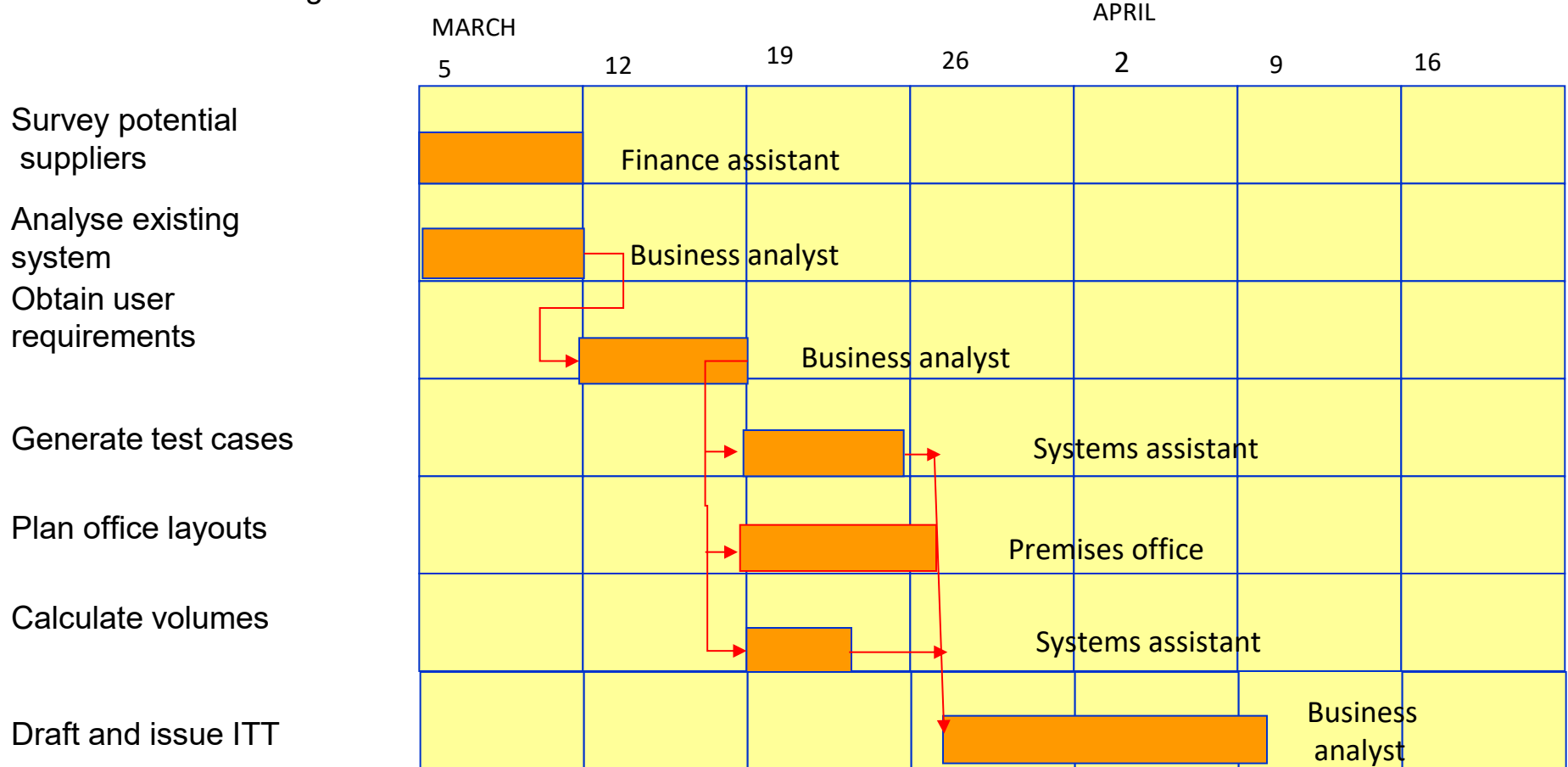
- 7.1 Identify and allocate resources to activities
- 7.2 Revise plans and estimates to take into account resource constraints
  - e.g. staff not being available until a later date
  - non-project activities

# Gantt charts

LT = lead tester

TA = testing assistant

Week commencing





# Step 8: Review/publicize plan

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

## Step 8: Review/publicise plan

- We have noted already that it is not feasible to produce a detailed plan for all stages of the project right at the beginning of the project planning process and not all the information needed for the detailed planning of the later stages is available at the outset. Initially an outline plan for the whole project would be produced, plus a detailed plan for the first stage.



**Step 9: Execute plan**

**Step 10: Create lower level plans**

# Summary

- Establish your objectives
- Think about the characteristics of the project
- Discover/set up the infrastructure to support the project (including standards)
- Identify **products** to be created and the **activities** that will create them
- Allocate resources
- Set up quality processes



# References :

1. B. Hughes, M. Cotterell, R. Mall, *Software Project Management*, Sixth Edition, McGraw Hill Education (India) Pvt. Ltd., 2018.



Thank you