Learning Resource
On
Software Project Management

Chapter-9
Monitoring and Control

Prepared By:

Kunal Anand

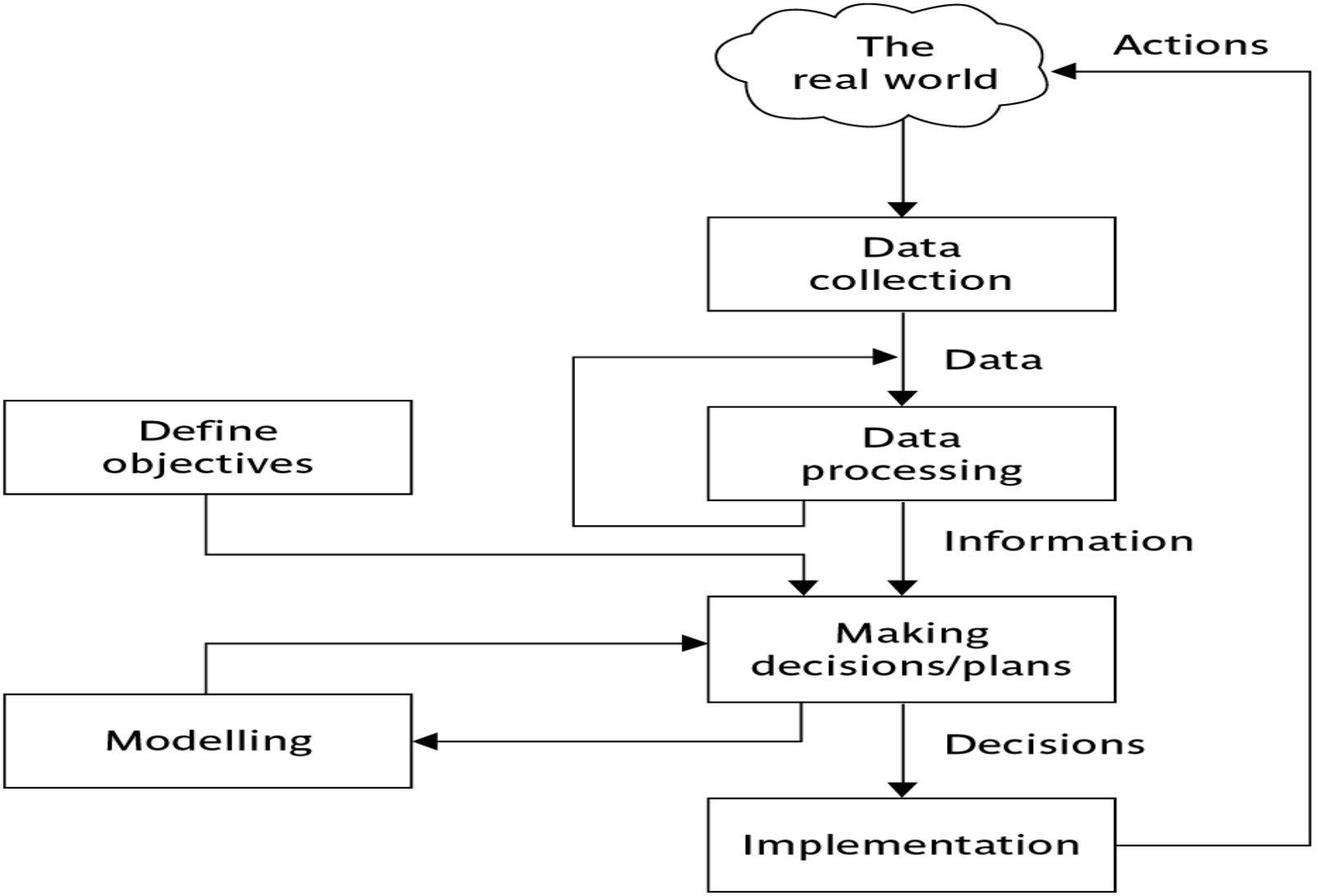
Assistant Professor, SCE

KIIT, DU, Bhubaneswar-24

Talkflow

- The Control Cycle
- Responsibilities
- Assessing Progress
  - Collecting progress reporting
  - Red/Green/Amber reporting
- Review
- Project Termination Review
- Gantt Chart and Slip Chart
- Earned Value Analysis
- Prioritizing Monitoring
- Exception Planning
- Change Control
- Software Configuration Management

The Control Cycle

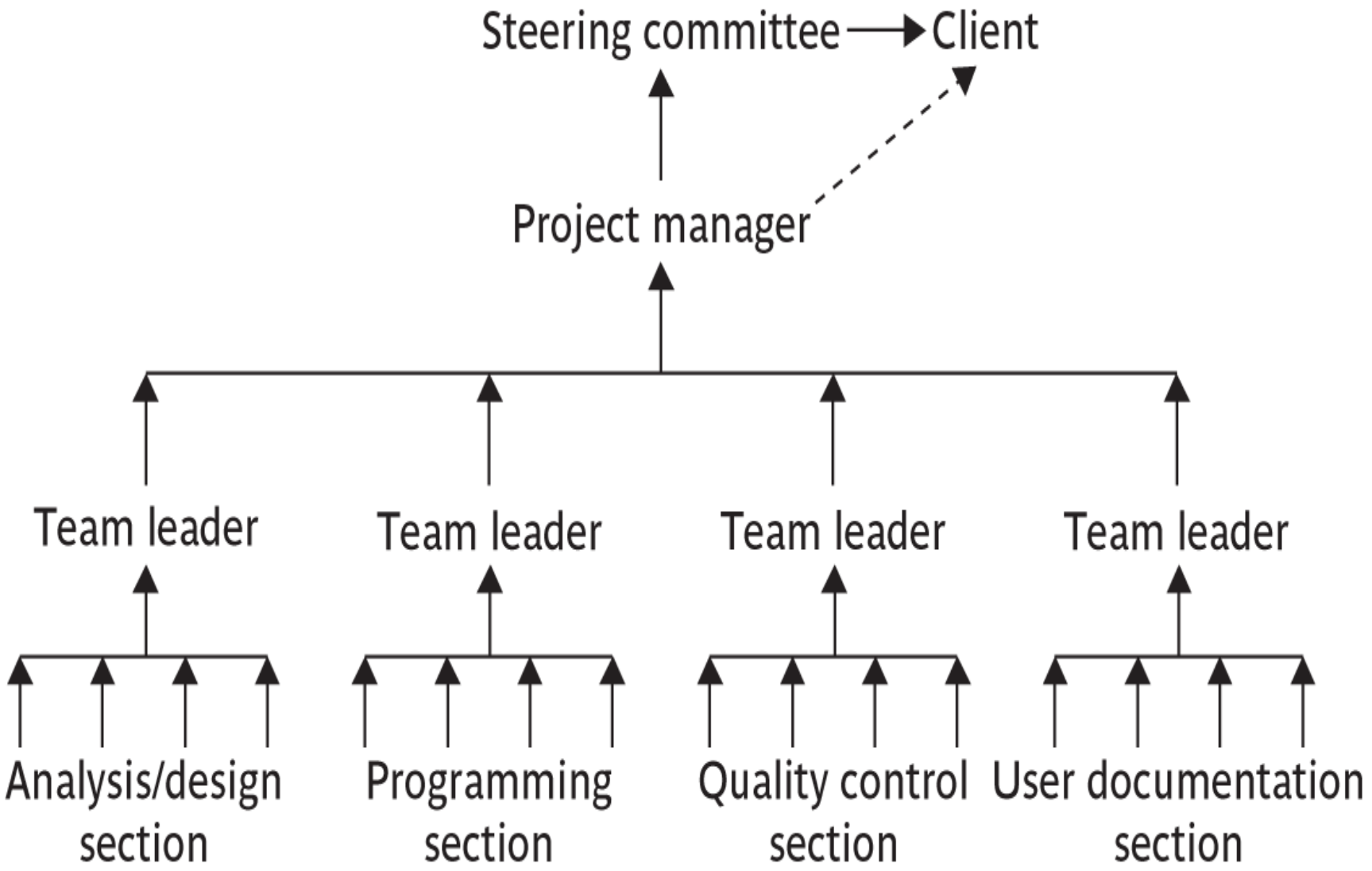


contd..

- ***Define objectives*** – at the beginning of the project we decide on what we want to achieve
- ***Making decisions/plans*** – we decide how we are going to achieve the objectives i.e., we create a plan
- ***Modelling*** – as part of the process of creating a plan we will consider different approaches and attempt to assess the consequences of each of these approaches in terms of how much it will cost and how long it will take, and so on.
- ***Implementation*** – the plan is now carried out
- ***Data collection*** – we gather information at regular intervals about how the project is progressing. These raw details could be quite numerous and complex on a large project
- ***Data processing*** – we process the progress data and convert it into 'information' which makes it easier for the project managers and others to understand the overall condition of the project
- ***Making decisions/plans*** – in the light of the comparison of actual project progress with that planned, the plans are modified. This may require the modelling of the outcomes of different possible courses of action

…and so the cycle goes on.

Responsibilities



Assessing progress

Checkpoints – predetermined times when progress is checked

- **Event driven** check takes place when a particular event has been **achieved**
- **Time driven:** date of the check is **pre-determined**

- **Frequency of reporting:** The higher the management level then generally the longer the gaps between checkpoints.

Collecting progress details

- Need to collect data about:
  - Achievements
  - Costs

- A big problem: how to deal with *partial completions*
  - ***99% completion syndrome***

- Possible solutions:
  - Control of products, not activities
  - Subdivide into lots of sub-activities

Red/Amber/Green reporting

- Identify key tasks
- Break down into sub-tasks
- Assess subtasks as:

  Green – 'on target'

  Amber – 'not on target but recoverable'

  Red – 'not on target and recoverable only with difficulty'

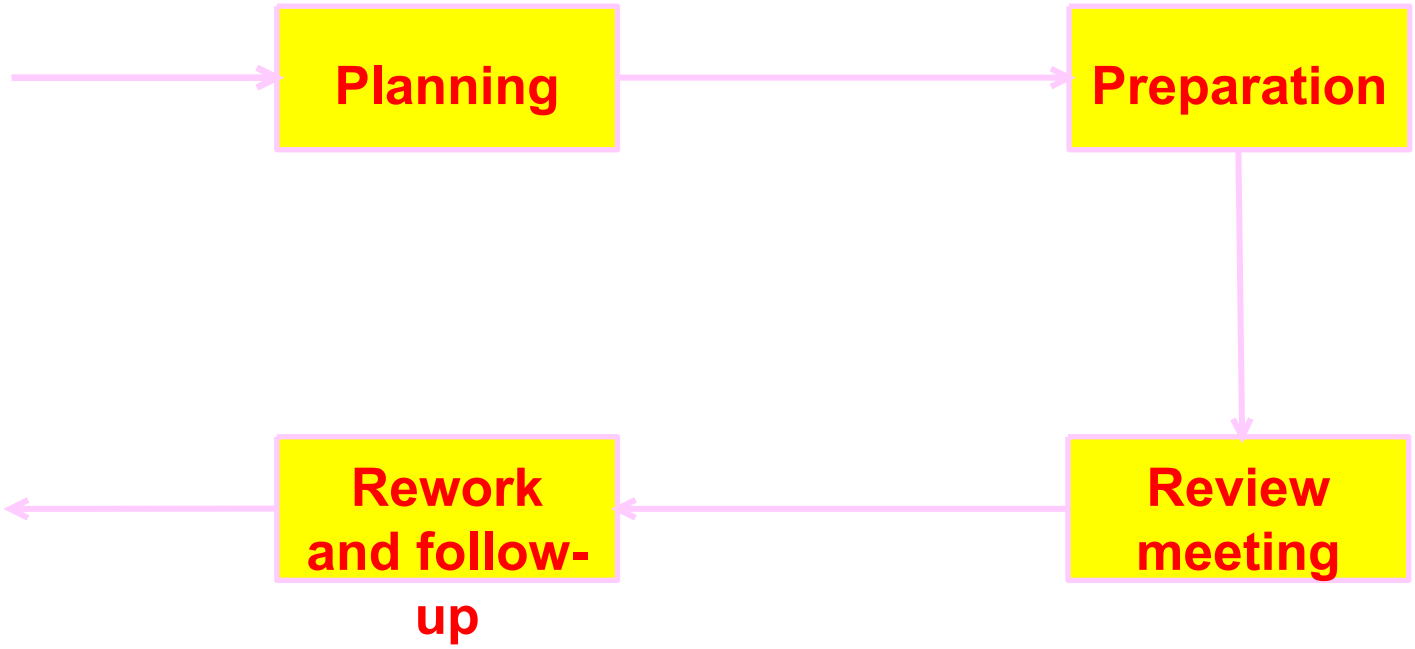
- Status of **'critical'** tasks is particularly important.

Review

- A **cost-effective** defect removal mechanism.
- **Review of work products** is an important mechanism for monitoring the progress of a project and ensuring the quality of the work products.
- Testing is an effective defect removal mechanism.
  - However, testing is applicable to only executable code.
  - Review is applicable to all work products.
- Review usually helps to **identify any deviation** from standards.
- Reviewers suggest ways to **improve** the work product.
- A review meeting often provides learning opportunities to not only the author of a work product, but also the other participants of the review meeting.
- The review participants gain a good understanding of the work product under review, making it easier for them to interface or use the work product in their work.

Review Roles

- **Moderator:** Schedules and convenes meetings, distributes review materials, leads and moderates review sessions.
- **Reviewers:** Performs the review of the assigned task.
- **Recorder:** Records the defects found and the time and effort data.
- **Review Process**



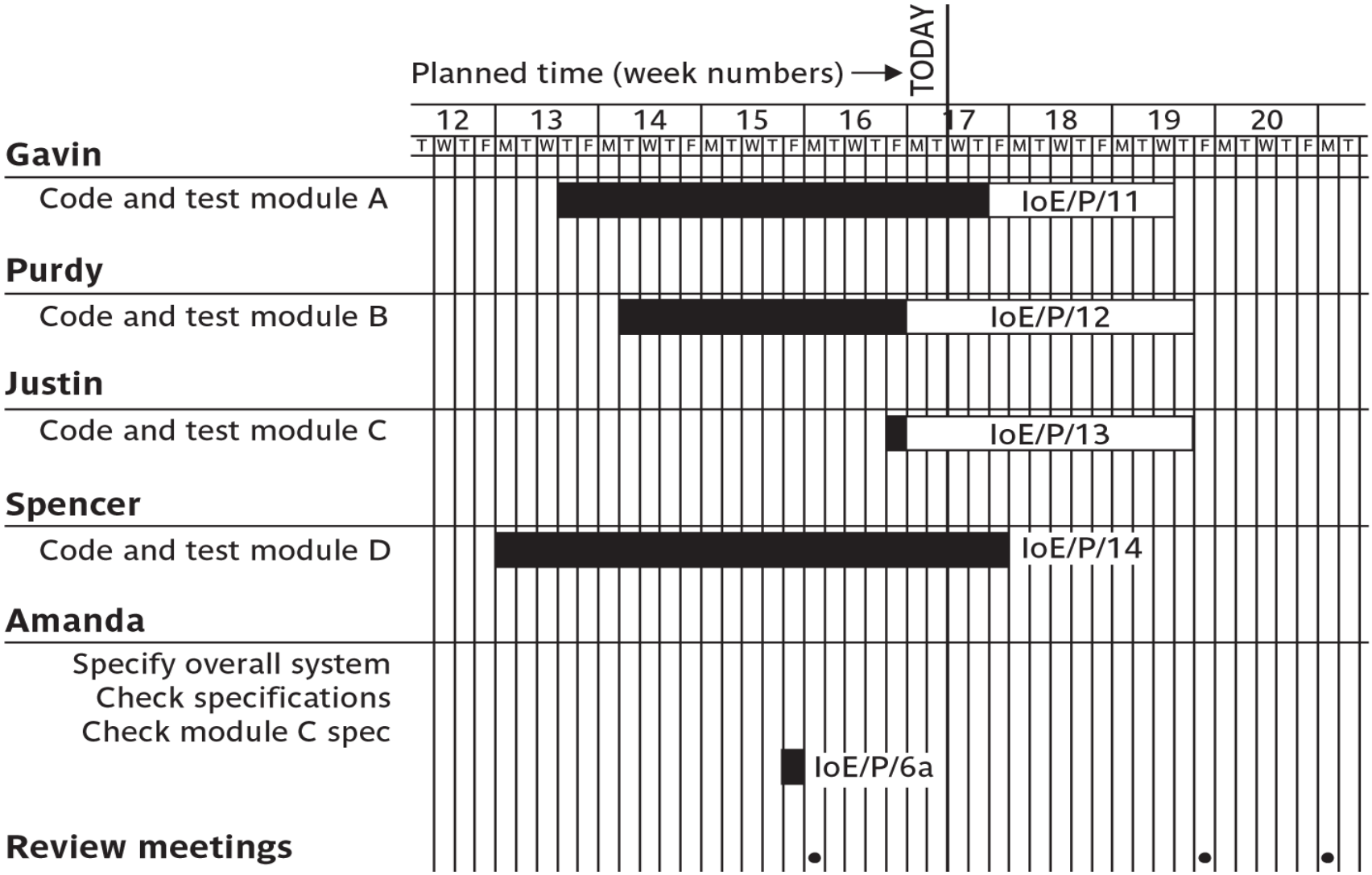
Project Termination Review

- **Project termination reviews** provide important opportunities to learn from past mistakes as well as successes.
- Reasons for project Termination
  - Project is **completed successfully** handed over to the customer.
  - **Incomplete** requirements
  - **Lack** of resources
  - Some key technologies used in the project have become **obsolete** during project execution
  - **Economics** of the project has changed, for example because many competing product may have become available in the market.

Project Termination Process

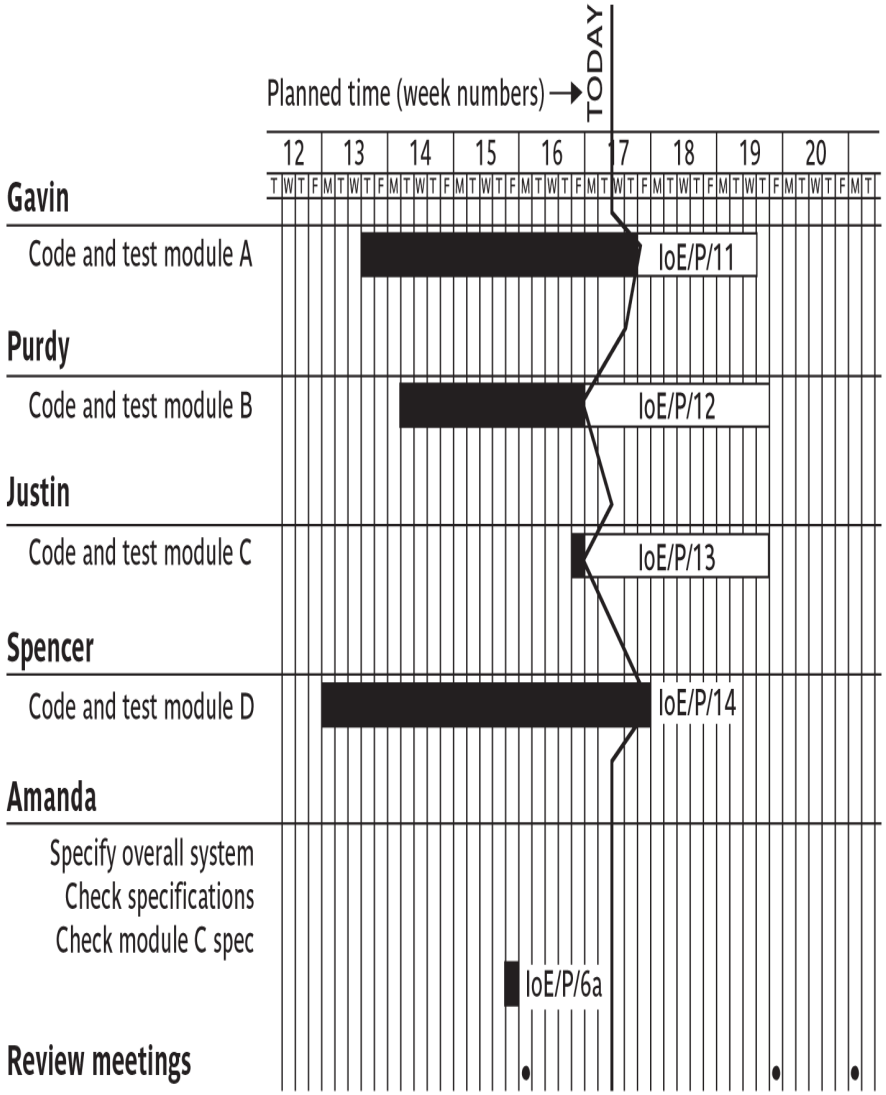
- Project survey
- Collection of objective information
- Debriefing meeting
- Final project review
- Result publication

Gantt charts



Slip Charts

- A **slip chart** is a version of the Gantt chart where a line is drawn from top to bottom.
- To the **left of the line** are all the **completed activities** and to the **right** those activities (or parts of activities) that **have not been completed.**
- The **more jagged the line**, the more it means that that there are some activities that are lagging to various degrees and some that are ahead of themselves.
- A **very jagged line** means that there is scope for re-planning to move resources from those activities that are ahead to those that are behind.



Cost Monitoring

- A project could be late because the staff originally committed have **not been deployed**.
- In this case the project will be ***behind time*** but ***under budget***.
- A project could be **on time** but only because additional resources have been added and so be ***over budget***.
- Need to monitor both **achievements** and **costs**

Earned value analysis

- ***Planned value (PV)*** **or** ***Budgeted cost of work scheduled (BCWS)***
  - original estimate of the effort/cost to complete a task (compare with idea of a 'price')

- ***Earned value (EV)*** **or** ***Budgeted cost of work performed*** **(BCWP)**
  - total of PVs for the work completed at this time

An Example:

- **Tasks**
  - Specify module: 5 days
  - Code module: 8 days
  - Test module: 6 days

- At the beginning of day 20, **PV = 19 days**
- If everything but testing completed **EV = 13 days**
- **Schedule variance** = **EV-PV** i.e., 13-19 = **-6**
- **Schedule performance indicator (SPI)** = **EV/PV**
  = 13/19 = 0.68
- **SV negative or SPI <1.00**, project **behind schedule**

Accounting conventions

- Work completed allocated on the basis
  - ***50/50:*** half allocated at start, the other half on completion. These proportions can vary e.g., 0/100, 75/25 etc
  - ***Milestone*** current value depends on the milestones achieved.
  - ***Units processed.***
- Can use money values, or staff effort as a surrogate

Earned Value analysis – actual cost

- **Actual cost (AC)** is also known as **Actual cost of work performed (ACWP)**
- In previous example, if
  - '**Specify module**' took 3 days
  - '**Code module**' took 4 days
- **Actual cost** = 7 days
- **Cost variance (CV) = EV-AC**

  i.e., 13-7 = **6 days**
- **Cost performance indicator (CPI)= EV / AC**

  = 13/7 = **1.86**
- **Positive CV** or **CPI > 1.00** means project **within budget.**

contd..

- CPI can be used to produce new cost estimate
- **Budget at completion (BAC)** – current budget allocated to total costs of project
- **Estimate at completion (EAC)** – updated estimate

  = BAC/CPI

  – **e.g.,** say budget at completion is £19,000 and CPI is 1.86
  – **EAC** = BAC/CPI = £10,215 (projected costs reduced because work being completed in less time)
- **Time variance (TV)** – difference between time when specified EV should have been reached and time it was.
- **For example**, say an EV of £19000 was supposed to have been reached on 1st April and it reached on 1st July then TV = **- 3 months**

Earned value chart with revised forecasts

- This shows how the planned value (PV), earned value (EV) and actual cost (AC) can be tracked over the lifetime of a project.
- It also shows how the graph can be used to show adjustments to the final estimated cost and duration.
- A revised assessment of the budget at completion (EAC estimate at completion) can be produced by dividing the original estimated budget at completion (BAC) by the current CPI.
- Similarly, a forecast of the actual duration of the project can be derived by dividing the original estimated duration by the SPI.

Prioritizing Monitoring

We might focus more on monitoring certain types of activity e.g.

- **Critical path** activities
- Activities with **no free float** – if delayed later dependent activities are delayed
- Activities with less than a **specified float**
- **High risk** activities
- Activities using **critical resources**

Getting back on track: options

- **Renegotiate** the deadline – if not possible then

- Try to **shorten critical path** e.g.
  - Work overtime
  - Re-allocate staff from less pressing work
  - Buy in more staff

- Reconsider **activity dependencies**
  - Over-lap the activities so that the start of one activity does not have to wait for completion of another
  - Split activities

Change Control

The role of configuration librarian:

- Identifying items that need to be subject to change control
- Management of a central repository of the master copies of software and documentation
- Administering change procedures
- Maintenance of access records

Typical change control process

1. One or more users might perceive the need for a change
2. User management decide that the change is valid and worthwhile and pass it to development management
3. A developer is assigned to assess the practicality and cost of making the change
4. Development management report back to user management on the cost of the change; user management decide whether to go ahead
5. One or more developers are authorized to make copies of components to be modified
6. Copies modified. After initial testing, a test version might be released to users for acceptance testing
7. When users are satisfied then operational release authorized – master configuration items updated

Software Configuration Management

- The deliverables of a SW product consist of a number of objects, e.g. source code, design document, SRS document, test document, user's manual, etc.
- These objects are modified by many software engineers through out development cycle.
- The state of each object changes as bugs are detected and fixed during development .
- The configuration of the software is the state of all project deliverables at any point of time.
- SCM deals with effectively tracking and controlling the configuration of a software during its life cycle.

Necessity of SCM

- To control access to deliverable objects with a view to avoiding the following problems
  - Inconsistency problem when the objects are replicated.
  - Problems associated with concurrent access.
  - Providing a stable development environment.
  - System accounting and status information.
  - Handling variants. If a bug is found in one of the variants, it has to be fixed in all variants

SCM activities

  - **Configuration identification** : deciding which objects (configuration items) are to be kept track of.
  - **Configuration control :** ensuring that changes to a system happen without ambiguity.
- **Baseline:** When an effective SCM is in place, the manager freezes the objects to form a base line.
  - A baseline is the status of all the objects under configuration control. When any of the objects under configuration control is changed , a new baseline is formed

Configuration item identification

- **Categories of objects:**
  - Controlled objects are under Configuration Control (CC) . Formal procedures followed to change them.
  - Pre-controlled objects are not yet under CC, but will eventually be under CC.
  - Uncontrolled objects are not subjected to CC
  - Controllable objects include both **controlled and precontrolled objects**; examples: SRS document, Design documents, Source code , Test cases

    The SCM plan is written during the project planning phase, and it lists all controlled objects.
- **Configuration Control (CC):** process of managing changes to controlled objects. Allows authorized changes to the controlled objects and prevents unauthorized changes .
  - In order to change a controlled object such as a module, a developer can get a private copy of the module by a reserve operation .
  - Configuration management tools allow only one person to reserve a module at a time. Once an object is reserved, it does not allow any one else to reserve this module until the reserved module is restored .

Changing the Baseline

- When one needs to change an object under configuration control, he is provided with a copy of the base line item.
- The requester makes changes to his private copy.
- After modifications, updated item replaces the old item and a new base line gets formed .

Reserve and restore operation in configuration control

- obtains a private copy of the module through a reserve operation and carries out all changes on this private copy.
- restoring the changed module to the baseline requires the permission of a change control board(CCB).
- Except for very large projects, the functions of the CCB are discharged by the project manager himself.