LECTURE 11

Project execution & closure

Content

Project Control Techniques (II)

- **Resource Leveling**:
  - ➢ Resource leveling is one technique that is employed to resolve resource conflicts during project execution.
  - ➢ Sometimes, it so happens that a resource is to do more than one task.
  - ➢ Now it is found that one task will get delayed due to the delay in the other task.
  - ➢ If there is a slack found in the schedule, the other task that has not started yet can be taken to some other time frame so that it will not be affected due to delay in the first task.
  - ➢ Or if this is not possible, then adding some more resources to the task can resolve this issue.
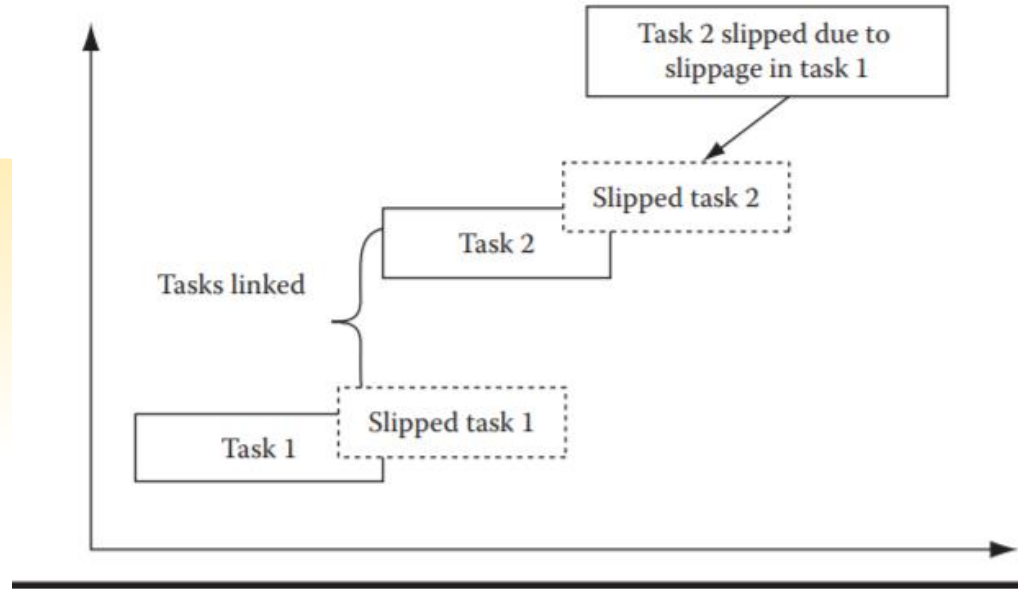
Project Control Techniques

- **Resource Leveling**:

  - When using software such as Microsoft Project for making the project schedule, the software has tasks that conflict with other tasks in the schedule.

  - These conflicts could be due to impractical start or finish dates for tasks, resource overallocation, or dependency of tasks on each other (so that if the first task gets delayed, the other will also get delayed).

  - Adjusting those tasks manually or automatically will resolve the conflict.
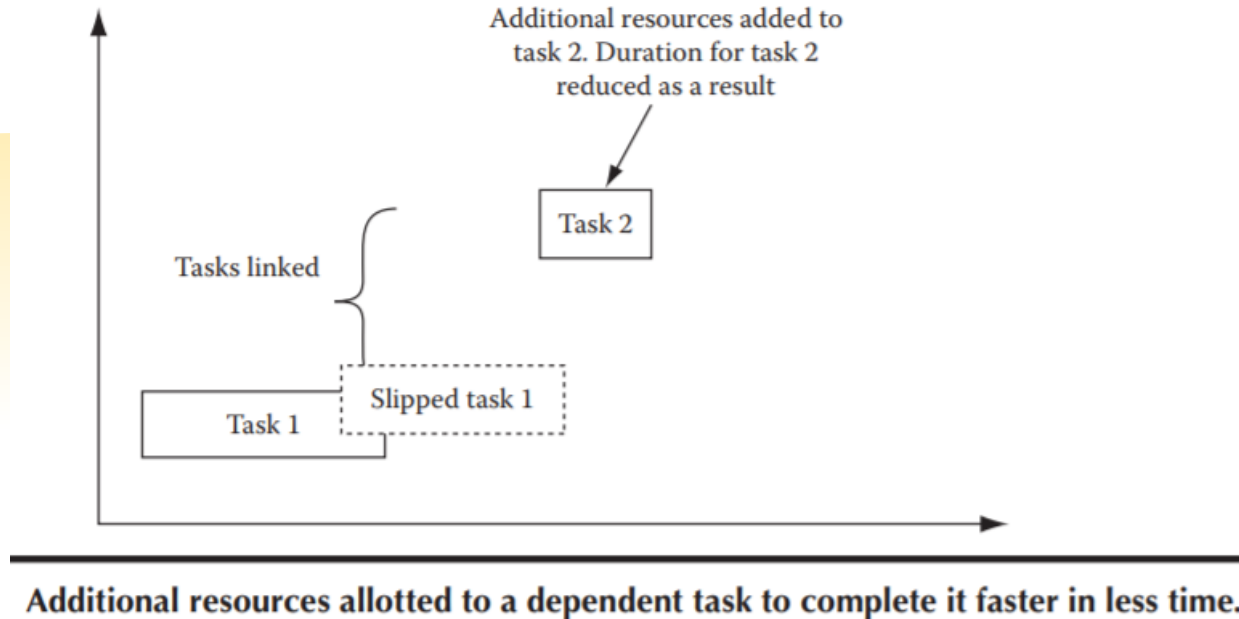
Project Control Techniques

- **Resource Leveling**:

Slippage in a task leads to slippage in the dependent task.

Project Control Techniques

- **Resource Leveling**:

Additional resources added to task 2. Duration for task 2 reduced as a result

Task 2

Tasks linked

Slipped task 1

Task 1

Additional resources allotted to a dependent task to complete it faster in less time.

Project Control Techniques

- **Schedule Optimization**:

  - Using PERT/CPM methods, we can determine the critical path of the project. But before drawing the critical path, the project manager should ascertain that there is no unnecessary slack in the project plan.

  - If there is any slack anywhere on the critical path, it should be removed to make the project plan optimized.

  - Similarly, as there could be many critical paths for the same project plan, unnecessary slack on all paths should be identified and removed.

  - Now the longest path out of these will be the critical path for the project.

Project Control Techniques

- **Schedule Optimization**:

  - Schedule optimization can also be done during execution.

  - If during execution, any task on the critical path is found to be done earlier than planned, then the critical path can be shortened.

  - This way schedule for the project can be collapsed or the extra time available can be used for starting dependent tasks earlier than planned schedule.
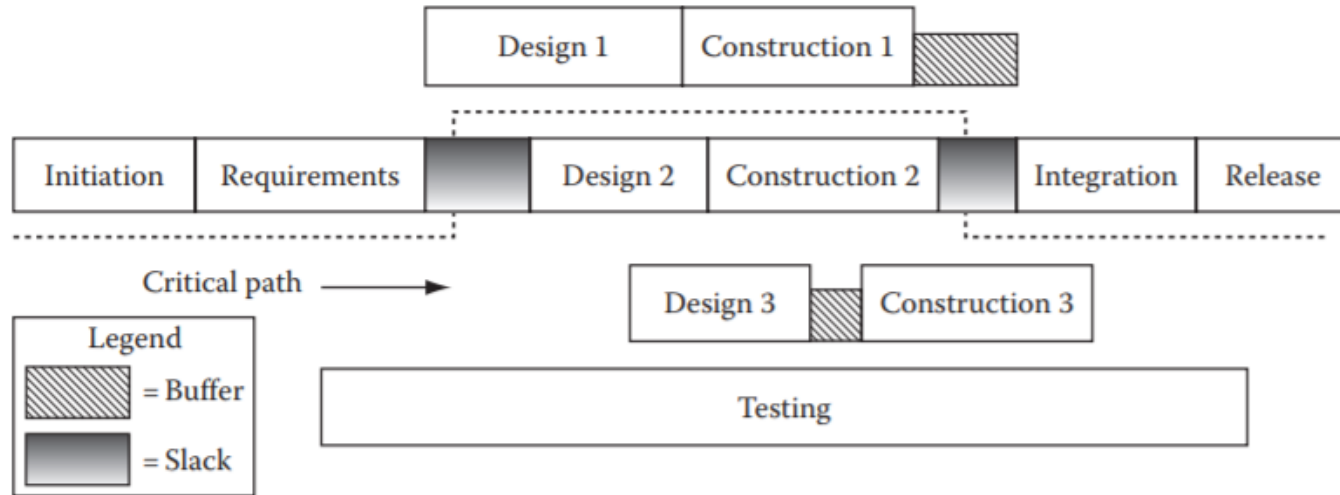
Project Control Techniques

- **Schedule Optimization**:

    ➤ One more technique of schedule optimization is to find if any tasks can be put in parallel that are currently in sequence.

    ➤ This way the schedule can be collapsed by a big margin. Then we can also optimize the schedule by checking if any task can be split and then putting these split tasks in parallel so that the schedule can be collapsed.

Project Control Techniques

- **Schedule Optimization**:

Slack in the critical path of a project plan.

Project Control Techniques

- **Corrective Actions against Deviations**:

  - From the project monitoring status reports, if it is observed that the project is deviating from plan, then corrective actions are to be taken by the project manager.

  - For taking corrective action, the situation is to be analyzed and root causes are to be identified.

  - Once root causes are found, solutions to fix them can be thought of and then action can be taken accordingly.

  - It is also advisable to have a good measurement of all process- and product-related attributes that are relevant to the project. Good measurements will help in decision-making process.

Project Control Techniques

- **Corrective Actions against Deviations**:
  - ➢ Some of the reasons for increased project cost include increase in overhead (higher cost of procuring tools, infrastructure, etc.) or salary.
  - ➢ It could also be due to schedule overrun.
  - ➢ So cost increases could either be schedule dependent or schedule independent.
  - ➢ If the cost increase is due to schedule overrun, then immediate action should be taken to correct the schedule deviation.

Project Control Techniques

- **Corrective Actions against Deviations**:

  ➤ Schedule deviation (almost always overrun) can happen due to faulty effort estimate, faulty scheduling, resource unavailability, loss of critical resources midway in the project, requirement creep, etc.

  ➤ Requirement creep is the most cited problem attributed to schedule overruns. The best policy regarding requirement creep is to bargain with the customer whenever any requirement change request comes.

Project Control Techniques

- **Corrective Actions against Deviations**:

  - The customer should be made aware of the consequences of the change request in project schedule.

  - Accept a change request only after the customer understands and agrees on the consequences in the project schedule.

  - Risks of resource unavailability or loss of resources pose a serious threat for the project

Project Control Techniques

- **Corrective Actions against Issues**:

  ➢ Issues should be classified into many categories and top-priority issues should be tackled first.

  ➢ Issues are also time-sensitive, and if they are not tackled in time, they will impact the project.

  ➢ How severe the impact will be depended on the kind of the issue itself.

  ➢ When many issues are in hand at a given time, it is difficult to identify their priority. All of them seem important.

Project Control Techniques

- **Corrective Actions against Issues**:

  - In such cases, it will be best to list them and put a weight against each of them.

  - Time sensitivity should also be considered (e.g., in how many days the issue should be sorted out). Now sort out your list with these two values against each issue.

  - If an issue with more weight has a bigger time window and if an issue with lower weight has a smaller time window, then if time permits, both should be tackled in parallel so that both can be resolved within their time windows

Project Control Techniques

- **Corrective Actions against Issues**:
  - ➤ However, if the project manager does not have much time to tackle both simultaneously, then it will be best to tackle the issue with the higher weight. So if a lower-priority issue cannot be resolved, it will not have much impact on the project, and at the same time, a bigger impact on the project can be avoided by resolving a higher-priority issue.

Project Control Techniques

- **Resource Optimization**:

  - The project manager from the outsourcing company may have to think about benefits to his organization from the project.

  - During project execution, however, there are many factors that threaten to eat into the profit margin.

  - The project manager has to keep an eye on the expenses so that profit margin could be kept intact.

  - In this regard, one known source of threat is an increasing wage of employees.

Project Control Techniques

- **Resource Optimization**:

  - To handle this issue, the project manager may have to make sure that productivity of the employees gets increased commensurate with the hike in salaries.

  - There are many practical ways of optimizing your resources in projects.

  - The best option is to use project portfolio management to utilize your available resources to the best possible way.

Project Control Techniques

▪ **Resource Optimization**:

  ➢ When you have a pool of resources and a list of projects, you can staff the projects in such a way that your pool of resources are utilized in such a way that no or least resources are sitting idle.

  ➢ Even within the pool of resources, some are costlier than others. It definitely makes sense that time of these higher-paid staff should be utilized to the maximum.

Project Monitoring and Control Artifacts

- Project monitoring provides project process and work product data that we can use to make decision and control the project so that later on it can be kept on track despite derailings in the past.

- The cost could have gone up from what was budgeted, the schedule could have overrun, or the work product quality could have gone down from what was expected.

- So basically we have three attributes of a project that should be monitored and controlled: schedule, cost and quality. The artifacts belonging to the schedule include PERT/CPM charts, network diagrams, resource charts, EVM, etc.

Project Monitoring and Control Artifacts

- The artifacts belonging to cost include budget analysis, resource optimization, EVM, etc.

- The artifacts for quality include requirement document review, design document review, source code review, test cycle logs, etc.

- The most important artifacts of project monitoring and control are actual project cost, product quality and schedule data.

- The overall project cost and schedule data in relation to project size and quality level determines productivity on the project

Project Monitoring and Control in Iterative Model

- Most of the project risks are tackled by dividing the entire project into small iterations.

- Thus, for iterative projects, the risks are manageable because their sizes are reduced, and they are distributed throughout the project by means of breaking the project into small iterations.

- However, sometimes it may happen that the customer demands some drastic change in his requirements, which may force an iteration to undergo a large change from the planned activities.

Project Monitoring and Control in Iterative Model

- In such cases, the project monitoring and control will be out of control, and thus, the project plan (iteration plan) will become invalid. A new project plan will have to be made.

- But in general, a project plan (or iteration plan) can be controlled using typical controlling techniques. A good technique to control an iteration is using a priority system for requirements or features.

- All the high-priority features will be completed in the iteration, and the low-priority features can be kept as options if time permits in the iteration.

Project Monitoring and Control in Iterative Model
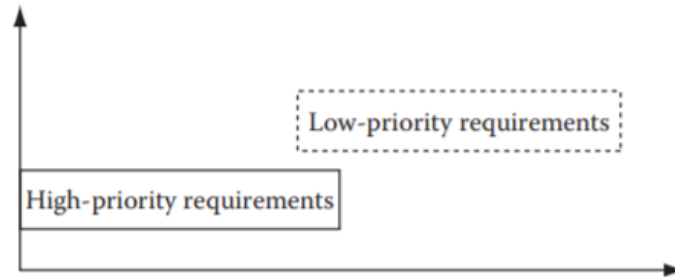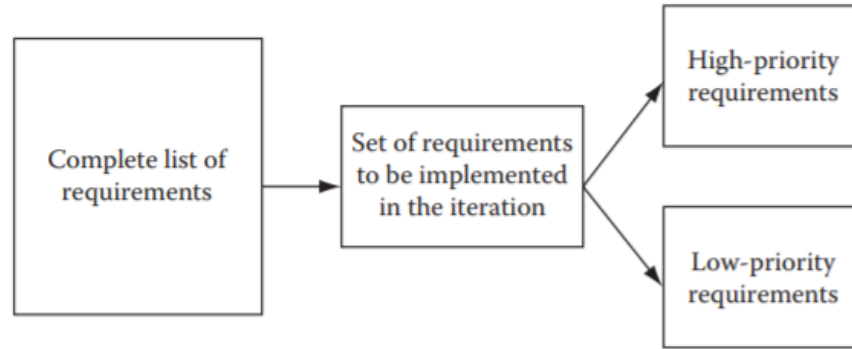
High- and low-priority requirements and keeping their schedule accordingly.

Introduction to Project Closure

- A software project could be a software development, software customization, software integration, software maintenance, or just one phase of the software development lifecycle (requirements, design, construction, or testing a software product).

- As per the contract, the final deliverables have to be handed over to the customer before the project deadline.

- Before the closure of the project, you need to check if all deliverables are going to be achieved before the set deadline.

Introduction to Project Closure

- The deliverables include the tested software product, user/training manuals, user training, and installation/implementation of the software product at client site. It may also include product release in formation if the project is to develop a software product with many iterations and is built incrementally.

- Do not forget that you need to keep a record of what happened during the execution of the project.

- If your company has a software engineering group and data from all projects that need to be kept in a central repository for statistical process-control purposes, then you also need to make sure that all relevant project data available before the closure of the project are fed into this repository.

Introduction to Project Closure
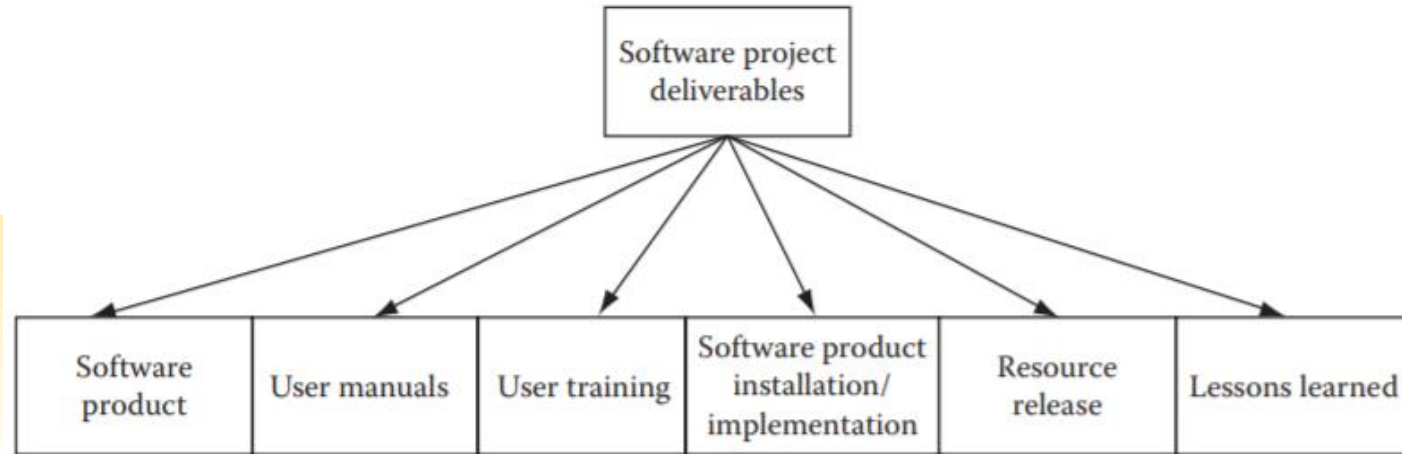
Software project deliverables before project closure.

Source Code Management

- Many versions of the source code get generated as requirements, and designs get changed during the software development life cycle.

- During testing, many bugs are discovered and they are fixed.

- The final source code thus has seen a lot of change, and which version will be shipped to customer needs to be identified.

- The configuration management system should be kept up to date with all source code changes.

Source Code Management
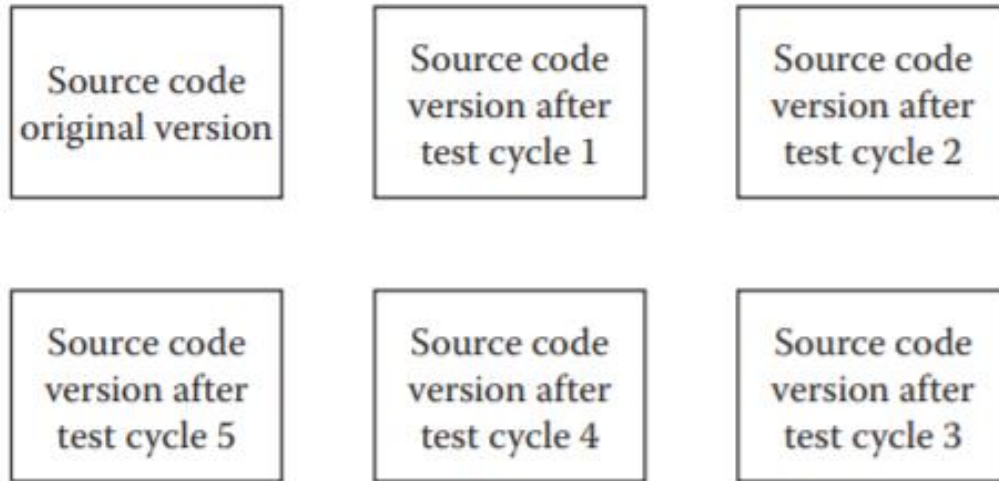
Many versions of source code.

Project Data Management

- When an existing project comes to an end, it is very important to archive project data.

- The archived data help in estimating effort, schedule, costs, and quality level for new projects.

- This information is very valuable for new projects.

- Providing project data as a performance indicator to the customer not only boosts customer confidence about ability of the project team, but it also helps in increasing productivity, project goal clarity, and reducing schedule and costs when future projects actually get executed.

Project Data Management

- When you need to compare or use data, it should be clean and relevant. So before sending project execution data into the repository, it should be made sure that the data are clean.

- For any project, relevant data are the execution data from similar projects. This similarity is in terms of project size, industry for which the software product was made, programming language used, lifecycle methodology used, etc.

- So the repository should be categorized accordingly. Depending on these variations, there will be many different types of projects.

Project Data Management

- When a new project is to be initiated, the repository should be searched for similar projects.

- Data from these projects can then be used for the new project. This cleaned and filtered data then will be very much relevant for the new project and thus will be extremely useful.

Project Data Management
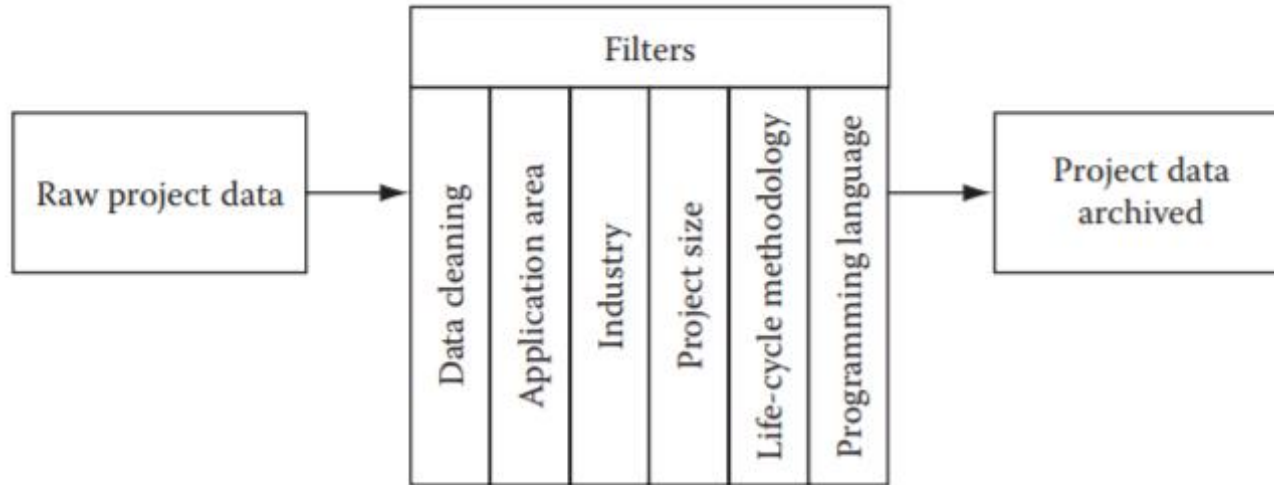
Strategy for project data archiving.

Project Closure in Iterative Model

- The iterative development model is very popular in software product development.

- Software vendors are always keen to launch new versions of their software product in the opportunity time window lest the opportunity is lost.

- This results in some problems on the software development front. Iteration closure is often a messy affair if care and restraint are not exercised.

- Due to market pressure, top management is under pressure to incorporate all the requested features in the release.

Project Closure in Iterative Model

- But it is clearly not feasible to do so.

- It is better to prioritize features based on market demand and effort required to make them. So release planning should be a part of the iteration planning at the beginning of the iteration.

- Features with high demand but requiring lesser effort should ideally be included first in the iteration.

- If time permits, then go for adding another feature. Keep doing it until you do not have any time left for adding any more features.

- Care should also be taken not to compromise on quality.

THANK YOU !