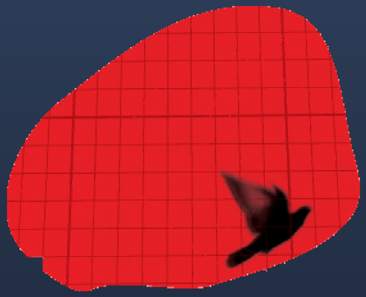
Applied Software Project Management

Project Schedules



What is a project schedule?

- The *project schedule* is a calendar that links the tasks to be done with the resources that will do them.
  - ▷Before a project schedule can be created, the project manager must have a work breakdown structure (WBS) and estimates.
  - ▷The schedule is part of the project plan.

Scheduling concepts: Effort vs. Duration

- *Effort* represents the work required to perform a task.
  - Effort is measured in person-hours (or person-days, person-weeks, etc.)
  - It represents the total number of hours that each person spent working on the task.
- *Duration* is amount of time that elapses between the time the task is started and the time it is completed.
  - Duration is measured in hours (or days, weeks, etc.)
  - It does not take into account the number of people performing the task

Scheduling concepts: Slack and Overhead

- *Slack* is the amount of time which any of the tasks can be delayed without causing the due date of the final task in the sequence to be delayed as well.
  - ▷ A tight schedule has very little slack; a delay in any task will cause a delay in the due date
  - ▷ Parkinson's Law: "Work expands so as to fill the time available for its completion."
- *Overhead* is any effort that does not go to the core activities of the task but is still required in order for the people to perform it—a sort of "real world" cost of actually doing the work.
  - ▷ Two people performing a task will require more effort than one person doing the same task
  - ▷ Assigning two people to the task requires more effort, but the task has a shorter duration

Building the project schedule

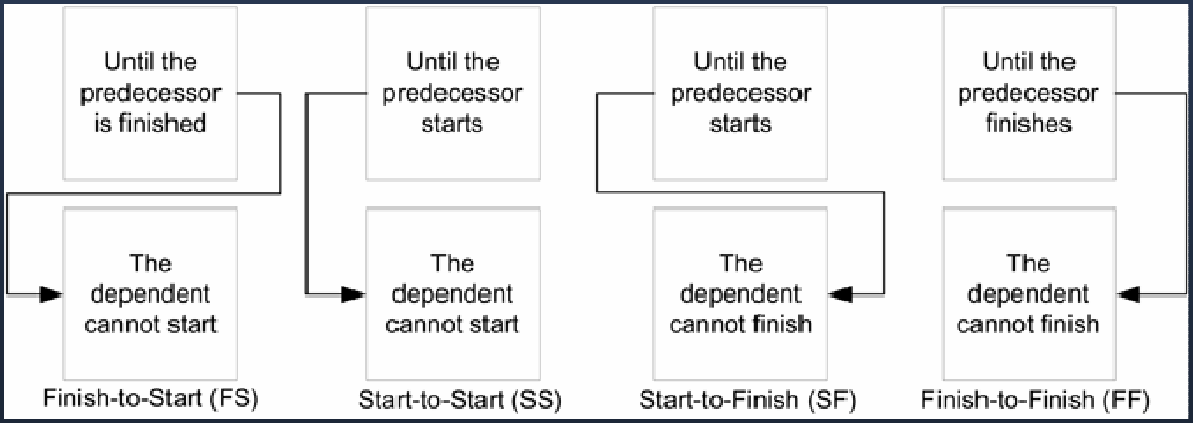
- Allocate resources
  - For each task in the WBS, one or more resources must be assigned
  - Choose person or people for each task based on qualifications, familiarity and availability
  - Take overhead into account when calculating the duration of each task

Building the project schedule

- Identify dependencies
  - A task has a dependency if it involves an activity, resource or work product which is subsequently required by another task
  - Tasks may have dependencies because they require the same resource

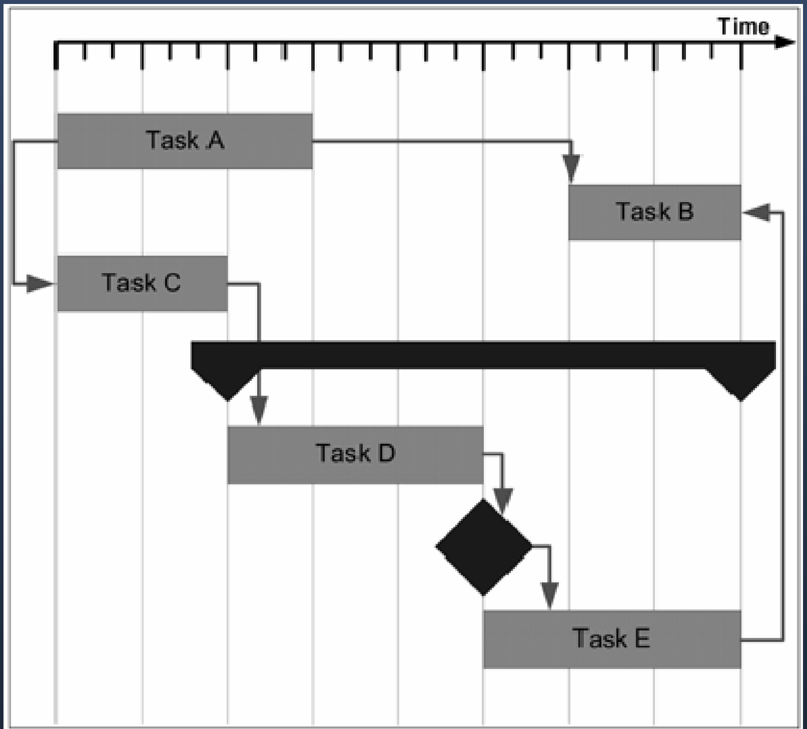
Building the project schedule

- Identify dependencies (continued)
  - Every dependency has a *predecessor*, or a task that must be begun, in progress, or completed, for another task to begin
  - Identify the type of predecessor for each dependency


Until the predecessor is finished
The dependent cannot start
Finish-to-Start (FS)
Until the predecessor starts
The dependent cannot start
Start-to-Start (SS)
Until the predecessor starts
The dependent cannot finish
Start-to-Finish (SF)
Until the predecessor finishes
The dependent cannot finish
Finish-to-Finish (FF)

Building the project schedule

- Create the schedule
  - Most project schedules are represented using a Gantt chart
  - The Gantt chart shows tasks, dependencies and milestones using different shapes



Building the project schedule

- Reconcile the schedule with the organization's needs
  - ▷ Once resources are allocated to each task, a final date can be calculated
  - ▷ If this date is unacceptable, the project plan must change
  - ▷ Either additional resources must be allocated to the project or the scope must be cut down
  - ▷ Brooks' Law: "Nine women cannot have a baby in one month."
    - In other words, some tasks can only be done by one person, no matter how critical they are.

Building the project schedule

- Add review meetings to the schedule
  - Progress reviews are meetings held regularly to check the progress of a project versus it's scheduled progress.
  - Milestone reviews are meetings which the project manager schedules in advance to coincide with project events.
    - The most common way for project managers to handle milestone reviews is to schedule them to occur after the last task in a project phase (such as the end of design or programming).

Building the project schedule

- Step 4: Optimize the schedule
  - The *critical path* is the sequence of tasks that represent the minimum time required to complete the project.
    - If a task is only on the critical path when delaying that task will delay the project.
    - Allocating resources to tasks on the critical path will reduce the project schedule; allocating them to other tasks will have less effect.
  - A resource is *over-allocated* if more than 100% allocated to multiple tasks simultaneously
    - If any resource is over-allocated, it means that there is a dependency between two tasks which was not discovered.
    - When this happens, the schedule is guaranteed to be inaccurate. Find and fix over-allocated resources.

Don’t abuse buffers

- A *buffer* is a task added to the schedule with no specific purpose except to account for unexpected delays.
  - This practice involves either adding extra tasks or padding existing tasks at strategic points in the schedule where overruns are “expected”.
  - Buffers can be useful:
    - On a year-long project, every programmer will take two weeks of vacation
    - Buffers can be used to account for this known delay
  - Buffers are often abused
    - The idea that overruns are expected means that there is an implicit assumption that the estimate is incorrect.
    - Buffers should not be used to add time to compensate for an inaccurate estimate.

Project metrics

- The *baseline* is the version of the schedule that has been approved
  - The schedule will change based on the actual work done by the project team.
  - When the deadline of the revised schedule is later than that of the baseline, the project has *slipped*.
- *Variance* is the difference between the estimated effort in the baseline and the actual effort performed by the team.

Project metrics

- *Earned value management* tracks the project by considering effort "earned" against a budget only after it has actually been performed
  - The *budgeted cost for work scheduled* (BCWS) is the estimated effort of the actual tasks that appear on the schedule to date.
  - The *actual cost of work performed* (ACWP) is the effort spent on the tasks in the schedule that have actually been completed by the development team members.
  - Variance = BCWS – ACWP

Project metrics

- The *cost performance index* is used to compare projects with each other or to compare phases within a project
  - CPI is calculated by dividing BCWS / ACWP (budgeted cost for work scheduled/actual cost for work performed) and multiplying by 100 to express it as a percentage.
  - A CPI of 100% means that the estimated cost was exactly right and the project came in exactly on budget.
  - A CPI under 100%, the work cost less effort than planned; a CPI greater than 100% means that the estimate was not adequate for the work involved.
    - For example, if the programming tasks took twice as long as estimated but every other type of task in the project took less time than estimated, the total variance for the project might still be low. However, the problem can still be pinpointed by calculating the CPI for each phase of development.