Project Management –
Project Scheduling

Benjamin D. Sweet
Lawrence Technological University
Overview

• Task Sequencing & Task/Activity Network Diagrams
• Critical Path Method (CPM)
• Project Scheduling
Project Scheduling - Rationale

• Project Scheduling involves:
  – Identifying interdependencies between project tasks (from the WBS)
  – Identifying the availability of resources (both human and material resources) that are required to complete each task
  – Arranging the project tasks in an optimal time sequence such that the available resources can be applied to each task, and all of the tasks can be completed within the allotted time

• As with other preliminary steps in Project Planning, Scheduling does not appear to be moving the project deliverables toward completion, and the project team may face pressure to abbreviate or skip this phase.

• As with the other preliminary planning steps, Scheduling helps to identify potential problems (overall project time duration, task sequencing, and resource availability and conflicts in the case of Scheduling) before they become critical “show stoppers.”
Falling Behind Schedule

Roger S. Pressman[5] gives several fundamental root causes for projects being delivered late:

- An unrealistic deadline is established by someone outside of the development group and forced on the project managers and practitioners
- Changes in customer requirements are not accommodated by changes in the schedule
- An honest underestimate of necessary effort and resources
- Predictable and/or unpredictable risks not considered
- Technical difficulties that were not seen in advance
- Human difficulties that were not seen in advance
- Miscommunication between project staff
- Failure to recognize that the project is falling behind schedule, and lack of action to correct the problem
Project Lateness

• Projects don’t suddenly become late
• Fred Brooks, *The Mythical Man Month*: Projects fall behind schedule “one day at a time”
• Project Manager’s objectives\[^5\]
  – Define all project tasks
  – Build a network that depicts task interdependencies
  – Identify the tasks that are critical within the network
  – Track critical task progress to recognize delays *early*
Task/Activity Network Diagrams

• The Work Breakdown Structure (WBS – developed in Project Planning) listed only the necessary tasks and sub-tasks, and did NOT address the sequence and interdependencies of the tasks.
• The Task (or Activity) Network diagram provides a graphical representation of the project plan, and indicates the interrelationships and dependencies between the project activities.
• The Network diagram begins to arrange the activities in the optimal sequence for completing the project in the minimum possible time.
• When developing a Network diagram, assume that there are sufficient resources to perform all project activities, such that it would be possible to have all activities begin at the same time.
Task/Activity Network Diagrams (2)

- In almost all cases, some activities must be completed before others can begin.
  - For example, when building a house, the basement concrete cannot be poured before the ground is excavated, the second-floor cannot be constructed before the basement is poured, the roof cannot be built and shingled before the second-floor is constructed, etc.
- Activities, or tasks, that can be worked on at the same time are drawn “in parallel” on the Network diagram.
- Activities that must follow other activities are drawn “in series” with the activates that they follow.
- Activities consume time in the schedule.
- Two different Network representations are used:
  - Activity-on-Node
  - Activity-on-Arrow
Activity Network Representations[1]

- Activity-on-Node networks represent activity as blocks connected by arrows. The time is consumed in the block that represents the activity.
Activity Network Representations\cite{1} (2)

- Activity-on-Arrow networks represent starting and ending times (events) as nodes, and the activity that consumes time is represented by arrows connecting the nodes.
Typical Task/Activity Network Diagrams

• PERT Charts:
  – Program Evaluation and Review Technique (PERT) – Developed by U.S. Navy in ???
  – PERT Charts show Task Sequencing, Estimated Time Duration, and Task/Path Dependencies

• Gantt Charts:
  – Named after Henry Gantt who developed the notational system in ??? for showing progress on “bar charts”\[1\]
  – Gantt Charts show Task Sequencing, Estimated Time Duration, and Task/Path Dependencies
  – Gantt Charts, unlike PERT Charts, show start-time decisions for NON-CRITICAL PATHS
Task Sequencing in Network Diagrams

- A Task/Activity Network Diagram is created by:
  - Arranging Dependent Tasks (tasks that require one or more tasks to be completed before they begin) Sequentially in Paths after completion of all “predecessor” tasks
  - Arranging Independent Tasks (tasks that do not depend on the completion of other tasks) and independent paths in Parallel
  - Merging Parallel Paths as needed based on Path Dependencies
- Time Estimates are assigned to each Task.
  - The estimated time through each path is the sum of the time estimates for each of the associated Tasks
- Critical Path
  - The longest (in time) chain of dependent tasks
  - Determines the earliest possible completion of the project
  - Any delay in the Critical Path will delay completion of the project
Task Start-Time Decisions

- Task start-time decisions for **NON-CRITICAL PATHS**:
  - **Early Start (ES)** – Starting a Task as Early as Possible
    - The “**Gap**” or “**Slack Time**” in the (non-critical) path is placed at the **end** of the path
    - Benefit – Helps avoid risks from unexpected delays that make the task take longer than estimated
    - Risk – Requires investing/expending of resources before they are absolutely needed
    - Risk – Late changes in the project can require ES tasks to be reworked or abandoned
    - Risk – Starting ALL Non-Critical tasks at the earliest possible time may cause the project leader to lose focus
  - **Late Start (LS)** – Starting a Task as Late as Possible
    - The “**Gap**” or “**Slack Time**” in the (non-critical) path is placed at the **beginning** of the path
    - Benefit – Helps avoid the cost of investing/expending resources before they are absolutely needed
    - Benefit – Helps avoid the cost of reworking or abandoning a task due to other changes in the project
    - Risk – Any delays in a LS task will delay the Critical Path
    - Risk – Starting ALL Non-Critical tasks at the latest possible time may cause the project leader to lose focus
Definitions[1]

- **Activity/Task** – Consumes time on the project schedule, and possibly resources as well.
- **Critical** – An Activity, Task or Event that must be achieved by a specific time with no “slack” or “float” time.
- **Critical Path** – The longest duration path through an activity network; determines the earliest possible completion of the project.
- **Event** – A specific point in time. Often represented on schedule timelines as circles or diamonds.
- **Milestone** – An Event that represents a significant point in the project. Often mark the beginning and/or completion of key Activities in the project, key deliveries, and meetings with key stakeholders. Smaller and more frequent Milestones are sometimes called “Inch-Pebbles”. This term may be used by the project team to express the feeling that the project is being micro-managed.
Total Effort vs. Elapsed Time

• In project task/activity estimation & WBS, time estimates on Tasks, Sub-Tasks, etc. give an indication of expected cost from a pure effort (person-hour) standpoint, but does NOT give an overall elapsed-time estimate.

• Once Scheduling identifies task/activity interdependencies (parallel efforts, activity predecessors, etc.), the critical path provides the estimate for overall elapsed-time.
Network Diagram Scheduling Computations\textsuperscript{[1]}

- James P. Lewis describes basic scheduling computations for Task/Activity Network Diagrams\textsuperscript{[1]}:
  - “Forward-Pass” Computations – determines the Earliest Start (ES) and Earliest Finish (EF) times for tasks in a task/activity network diagram
  - “Backward-Pass” Computations – determine the Latest Start (LS) and Latest Finish (LF) times for tasks in a task/activity network diagram

- The following rules of thumb apply:
  - ALWAYS use the same units of time for each task
  - Don’t use units of time that are finer in detail than you can manage
  - If a task takes longer than four to six weeks, consider breaking it up into smaller tasks
“Forward-Pass” Computations[1]

- Computations proceed from the first task to the last task (usually from the left side to the right side of the network diagram).
- The first task in each independent path begins at time Zero
  - Early Start (ES) = 0.
- The Early Finish (EF) for each task is its Early Start time plus its Task Duration
  - EF = ES + Task Duration
- Dependent Tasks in a Path:
  - For each successive dependent task in a path, the Early Start is the Early Finish of the previous task.
- Merging Parallel Paths:
  - When a task depends on completion of two or more parallel Tasks or Paths before it can begin (ie: parallel paths merge), its Early Start is the latest (ie: largest) Early Finish of the preceding parallel tasks.
“Backward-Pass” Computations[1]

- Computations proceed from the last task to the first task (usually from the right side to the left side of the network diagram).
- The Late Finish (LF) of the LAST Task is either:
  - The latest time (duration) that the overall project can finish, or
  - The Early Finish (EF) time from the “Forward-Pass” computation
- The Late State (LS) for each task is its Late Finish time minus its Task Duration
  - LS = LF – Task Duration
- Dependent Tasks in a Path:
  - For each predecessor task in a path, the Late Finish is the Late Start of the subsequent task.
- Merging Parallel Paths:
  - When two or more parallel Tasks or Paths branch from a common predecessor task, the Late Finish for the predecessor task is the earliest (ie: smallest) Late Start of the subsequent parallel tasks
More Definitions\[1\]

- **Float or Slack Time** – The difference between Early Start and Late Start, and difference between Early Finish and Late Finish:
  - Float Time = LS – ES = LF – EF
  - Tasks with Float Time have *some* latitude as to when they can start

- **Critical** – Tasks that have no Float Time:
  - LS = ES; LF = EF; Float Time = 0
  - Tasks with NO Float Time have NO latitude as to when they can start
  - Delays in starting Critical Tasks will delay the completion of the overall project

<table>
<thead>
<tr>
<th>Early Start</th>
<th>Duration</th>
<th>Early Finish</th>
</tr>
</thead>
<tbody>
<tr>
<td>Late Start</td>
<td>Slack</td>
<td>Late Finish</td>
</tr>
</tbody>
</table>

Note: MS Visio’s “PERT Chart Shapes” template under “Project Schedule” Drawing Type.
Example – Scheduling Computations

Given the tasks with the following durations:
• Task 1 – 3 days
• Task 2 – 4 days
• Task 3 – 2 days
• Task 4 – 5 days
• Task 5 – 5 days
• Task 6 – 2 days

Note:
• These durations do NOT account for weekends and other non-working days (Holidays, etc.)
Example – PERT Chart

<table>
<thead>
<tr>
<th>Early Start</th>
<th>Duration</th>
<th>Early Finish</th>
</tr>
</thead>
<tbody>
<tr>
<td>Late Start</td>
<td>Slack</td>
<td>Late Finish</td>
</tr>
</tbody>
</table>

Key

- **Early Start**
- **Duration**
- **Early Finish**
- **Late Start**
- **Slack**
- **Late Finish**

Task 1

- Early Start: 0
- Duration: 3
- Early Finish: 3

Task 2

- Early Start: 3
- Duration: 4
- Early Finish: 7

Task 3

- Early Start: 7
- Duration: 2
- Early Finish: 9

Task 4

- Early Start: 3
- Duration: 5
- Early Finish: 8

Task 5

- Early Start: 0
- Duration: 4
- Early Finish: 9

Task 6

- Early Start: 9
- Duration: 2
- Early Finish: 11
Example – Gantt Chart

<table>
<thead>
<tr>
<th>ID</th>
<th>Task Name</th>
<th>Start</th>
<th>Finish</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Task 1</td>
<td>1/3/2005</td>
<td>1/5/2005</td>
<td>3d</td>
</tr>
<tr>
<td>2</td>
<td>Task 2</td>
<td>1/6/2005</td>
<td>1/11/2005</td>
<td>4d</td>
</tr>
<tr>
<td>3</td>
<td>Task 3</td>
<td>1/12/2005</td>
<td>1/13/2005</td>
<td>2d</td>
</tr>
<tr>
<td>4</td>
<td>Task 4</td>
<td>1/6/2005</td>
<td>1/12/2005</td>
<td>5d</td>
</tr>
<tr>
<td>5</td>
<td>Task 5</td>
<td>1/3/2005</td>
<td>1/7/2005</td>
<td>5d</td>
</tr>
<tr>
<td>6</td>
<td>Task 6</td>
<td>1/14/2005</td>
<td>1/17/2005</td>
<td>2d</td>
</tr>
</tbody>
</table>

Note: MS Visio’s “Gantt Chart Shapes” template under “Project Schedule” Drawing Type.
Activity Network Exercise – Planning a Party

• ...

Copyright 2005 - Benjamin D. Sweet - Lawrence Technological University 2/11/2005
Critical Path
Reasons for Lateness

• Nobody is 100% Productive – in an 8-hour day, we are only productive 4 to 6 hours due to:
  – “Bio Breaks” – Everybody **needs** to take breaks, stretch and move, drink, eat, and use the restroom regularly throughout the day.
  – Distraction – It can take between ten and twenty minutes to get back into productive/creative “flow” after an interruption, and we are interrupted almost every fifteen to twenty minutes by phone calls, e-mails, and ambient activity.
  – Fatigue – Working overtime on an ongoing basis may be detrimental, as we are less productive and more prone to making errors (that will consume time and other resources to resolve) when we are tired.
• “Student’s Syndrome” – When a due-date is given, people tend not to start work until the latest possible time. (**Why? WHO KNOWS??!!**)  
• Multi-Tasking – Switching between multiple “concurrent” tasks often leads to lateness for several reasons:
  – The “interrupted” tasks take more effort to complete than “uninterrupted” tasks due to the distraction factor (mentioned above): it takes “set-up” time to switch between tasks.
  – Regardless of the additional effort for “interrupted” tasks, each individual multi-tasked task takes more calendar time to complete due to the other intervening tasks.
Avoiding Lateness

Various strategies can help avoid lateness, or give an early indication that lateness is imminent so that adjustments can be made:

• When planning, account for “Human Factors”
  – Plans that assume on 8 productive hours/day, 7 days/week, and/or extended overtime are inherently flawed plans.
• Don’t use Milestones[^8] (ie: completion due dates):
  – Avoids “Student’s Syndrome”
  – Project team members don’t know when project step is “supposed” to be completed, or even if there is enough time to complete it
  – Begin each step as soon as possible (given available resources)
• Measure progress in terms of:[^8]
  – Percent Completion of Critical Path
  – Expected completion times of each task in progress
• Eliminate Multi-Tasking:[^8]
  – For people (or other resources) working on Critical Paths and Feeding Paths:
    • Remind them of Critical Path activity one week before it is scheduled to begin
    • Remind them again three days before
    • Remind them again one day before
    • When the time comes, drop everything to work exclusively on the Critical Path
Strategies for “Aggressive” Projects

- Project teams may use their understanding of the “Triple Constraints” as a basis to negotiate functionality, cost and schedule.

- Consider releasing incremental product functionality in iterative development phases – deliver the highest priority functional needs first.
Risk Management --- Go/No-Go

• Careful evaluation of a project proposal must consider the desired project objectives (outcomes), the available resources, the consequences of failing to achieve the objectives, and the consequences of not attempting the project.
• If this evaluation concludes that it is impossible to achieve the desired objectives within the specified timeframe and with the available resources (people, equipment, budget, etc.), then a *valid* outcome of the P.M. process is to “reject” the project proposal.
• It is far better to decide *not* to proceed with a project that has a high likelihood of failure *before* considerable resources have already been expended in a futile effort.
• Note that the decision to reject a project proposal does NOT mean that the customer’s business is rejected. This is an opportunity to *renegotiate* the desired project outcomes, the schedule, and/or the price (the “PCTS”).
Risk Management --- The Project – What Can Go Wrong

John Reel defines 10 signs indicating problems in an information systems project:
1. Software people don’t understand customer’s needs.
2. Product scope is poorly defined.
3. Changes are managed poorly.
4. The chosen technology changes.
5. Business needs change or are ill-defined.
6. Deadlines are unrealistic.
7. Users are resistant.
8. Sponsorship is lost or was never properly attained.
9. The project team lacks people with appropriate skills.
10. Managers and practitioners avoid best practices and lessons learned.
Summary
Discussion Topics