Project Scheduling and Tracking

SWEN 256 – Software Process & Project Management
Plan: Identify activities. No specific start and end dates.

Estimating: Determining the size & duration of activities.

Schedule: Adds specific start and end dates, relationships, and resources.

Track: Uses monitoring and tools to determine if plans, estimates, and schedules are accurate.
1. Identify “what” needs to be done
   - Work Breakdown Structure (WBS)
2. Identify “how much” (the size)
   - Size estimation techniques
3. Identify the dependency between tasks
   - Dependency graph, network diagram
4. Estimate total duration of the work to be done
   - The actual schedule
You need to decompose your project into manageable chunks

ALL projects need this step

Divide & Conquer

Two main causes of project failure
  - Forgetting something critical
  - Ballpark estimates become targets

How does partitioning help this?
Project Elements

A Project: functions, activities, tasks

- Function
  - Management activity
  - Often spanning the life of the project, such as:
    - Project Management
    - Risk Management
    - Change Management

- Activity
  - An element of work with expected duration, cost, resources
  - Can be subdivided into other activities or tasks

- Task
  - Lowest level of work on the project
  - Typically not shown on preliminary WBS (too granular)
  - Smallest unit of work in the real schedule
Work Breakdown Structures

a.k.a. WBS
Work Break Down Structure (WBS): a check list of the work that must be accomplished to meet the project objectives.

The WBS lists the major project outputs and those departments or individuals primarily responsible for their completion.
0.0 Retail Web Site
1.0 Project Management
2.0 Requirements Gathering
3.0 Analysis & Design
4.0 Site Software Development
   4.1 HTML Design and Creation
   4.2 Backend Software
      4.2.1 Database Implementation
      4.2.2 Middleware Development
      4.2.3 Security Subsystems
      4.2.4 Catalog Engine
      4.2.5 Transaction Processing
   4.3 Graphics and Interface
   4.4 Content Creation
5.0 Testing and Production
Process WBS
- a.k.a Activity-oriented
- Ex: Requirements, Analysis, Design, Testing
- Typically used by PM

Product WBS
- a.k.a. Entity-oriented
- Ex: Financial engine, Interface system, DB
- Typically used by engineering manager

Hybrid WBS: both above
- This is not unusual
- Ex: Lifecycle phases at high level with component or feature-specifics within phases
- Rationale: processes produce products
Product WBS

Intranet

Web Site Design
  - Site Map
  - Graphic Design
  - Programs

Home Page Design
  - Text
  - Images
  - Hyperlinks

Marketing Pages
  - Text
  - Images
  - Hyperlinks

Sales Pages
  - Text
  - Images
  - Hyperlinks
List of Activities, not Things

List of items can come from many sources
  o SOW, Proposal, brainstorming, stakeholders, team

Describe activities using “bullet language”
  o Meaningful but terse labels

All WBS paths do not have to go to the same level

Do not plan more detail than you can manage
Work Packages (Tasks)

- Generic term for discrete tasks with definable end results
- The “one-to-two” rule
  - Often at: 1 or 2 persons for 1 or 2 weeks
- Basis for monitoring and reporting progress
  - Can be tied to budget items (charge numbers)
  - Resources (personnel) assigned
- Ideally shorter rather than longer
  - Longer makes in-progress estimates needed
  - These are more subjective than “done”
  - “4/40” or “8/80” rule (shortest/longest duration)
  - Not so small as to micro-manage
PM must map activities to chosen lifecycle
Each lifecycle has different sets of activities
Integral process activities occur for all
  - Planning, configuration, testing
Operations and maintenance phases are not normally in plan (considered post-project)
Some models are “straightened” for WBS
  - Spiral and other iterative models
  - Linear sequence several times
Deliverables of tasks vary by methodology
WBS Techniques

- Top-Down
- Bottom-Up
- Analogy
- Rolling Wave
  - 1st pass: go 1-3 levels deep
  - Gather more requirements or data
  - Add more detail later
- Post-its on a wall

All WBS Techniques rely upon Expert Judgment!
Top-down
- Start at highest level
- Systematically develop increasing level of detail
- Best if
  - The problem is well understood
  - Technology and methodology are not new
  - This is similar to an earlier project or problem
- But is also applied in majority of situations
Bottom-up
- Start at lowest level tasks
- Aggregate into summaries and higher levels
- Cons
  - Time consuming
  - Needs more requirements complete
- Pros
  - Detailed
Analogy

- Base WBS upon that of a “similar” project
- Use a template
- Analogy also can be estimation basis

Pros
- Based on past actual experience

Cons
- Needs comparable project
Brainstorming

- Generate all activities you can think of that need to be done
- Group them into categories

Both Top-down and Brainstorming can be used on the same WBS

Remember to get the people who will be doing the work involved (buy-in matters!)
Network scheduling
Costing
Risk analysis
Organizational structure
Control
Measurement
Should be easy to understand

Some companies have corporate standards for these schemes

Some top-level items, like Project Mgmt. are in WBS for each project
  - Others vary by project

What often hurts most is what’s missing

Break down until you can generate accurate time & cost estimates

Ensure each element corresponds to a deliverable
Sequencing Tasks & Activities

Tools and Techniques
Sequence the Work Activities

- Milestone Chart
- Gantt chart
- Network Techniques
  - CPM (Critical Path Method)
  - PERT (Program Evaluation and Review Technique)
Gantt chart is a means of displaying simple activities or events plotted against time or dollars.

Most commonly used for exhibiting program progress or for defining specific work required to reach an objective.

Gantt charts may include listing of activities, activity duration, scheduled dates, and progress-to-date.
<table>
<thead>
<tr>
<th>WBS</th>
<th>Task Name</th>
<th>Cost</th>
<th>January</th>
<th>February</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Project Summary</td>
<td>$38,000.00</td>
<td>12/31</td>
<td>01/07</td>
</tr>
<tr>
<td>2</td>
<td>Design Phase</td>
<td>$18,400.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>First Design Phase</td>
<td>$4,000.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Start Milestone</td>
<td>$0.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Design Task 1</td>
<td>$4,000.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Second Design Phase</td>
<td>$14,400.00</td>
<td>01/01</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Design Task 2</td>
<td>$6,000.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Design Task 3</td>
<td>$4,000.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Design Task 4</td>
<td>$4,400.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>End Design Milestone</td>
<td>$0.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Programming Phase</td>
<td>$10,000.00</td>
<td>02/09</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Program Task 1</td>
<td>$6,000.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>Program Task 2</td>
<td>$4,000.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>End Program Milestone</td>
<td>$0.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>Testing Phase</td>
<td>$9,600.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>Test Task 1</td>
<td>$2,400.00</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Advantages:
- Easy to understand
- Easy to change

Disadvantages:
- Only a vague description of the project
- Does not always show interdependency of activities
- May not show results of an early or late start of an activity
A precedence network diagram is a graphic model portraying the sequential relationship between key events in a project.

Initial development of the network requires that the project be defined and thought out.

The network diagram clearly and precisely communicates the plan of action to the project team and the client.
<table>
<thead>
<tr>
<th>Task</th>
<th>Duration</th>
<th>Dependencies</th>
</tr>
</thead>
<tbody>
<tr>
<td>A - Architecture &amp; design strategy</td>
<td>9</td>
<td>start</td>
</tr>
<tr>
<td>B - Decide on number of releases</td>
<td>5</td>
<td>A</td>
</tr>
<tr>
<td>C - Develop acceptance test plan</td>
<td>7</td>
<td>A</td>
</tr>
<tr>
<td>D - Develop customer support plan</td>
<td>11</td>
<td>B,C</td>
</tr>
<tr>
<td>E - Final sizing &amp; costing</td>
<td>8</td>
<td>D</td>
</tr>
</tbody>
</table>
Critical Path Method (CPM) tries to answer the following questions:

☞ What is the duration of the project?

☞ By how much (if at all) will the project be delayed if any one of the activities takes N days longer?

☞ How long can certain activities be postponed without increasing the total project duration?
Sequence of activities that have to be executed one after another

Duration times of these activities will determine the overall project time, because there is no slack/float time for these activities

If any of the activities on the critical path takes longer than projected, the entire project will be delayed by that same amount

Critical path = Longest path in the precedence network (generally, the longest in time)
Critical Path Example

- Critical Path = A – C – D – E (35 time units)
- Critical Tasks = A, C, D, E
- Non-Critical Path = A-B-D-E
- Non-Critical Tasks = B (only)
### Critical Path Example (Continued)

<table>
<thead>
<tr>
<th>Task</th>
<th>Duration</th>
<th>Depend</th>
<th>Earliest Start</th>
<th>Earliest Finish</th>
<th>Latest Start</th>
<th>Latest Finish</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>9</td>
<td>none</td>
<td>0</td>
<td>9</td>
<td>0</td>
<td>9</td>
</tr>
<tr>
<td>B</td>
<td>5</td>
<td>A</td>
<td>9</td>
<td>14</td>
<td>11</td>
<td>16</td>
</tr>
<tr>
<td>C</td>
<td>7</td>
<td>A</td>
<td>9</td>
<td>16</td>
<td>9</td>
<td>16</td>
</tr>
<tr>
<td>D</td>
<td>11</td>
<td>B,C</td>
<td>16</td>
<td>27</td>
<td>16</td>
<td>27</td>
</tr>
<tr>
<td>E</td>
<td>8</td>
<td>D</td>
<td>27</td>
<td>35</td>
<td>27</td>
<td>35</td>
</tr>
</tbody>
</table>

Start → A 9 → B 5 → C 9 → D 11 → E 8 → End

**Slack time** – maximum allowable delay for a non-critical activity.

Task slack time = LS – ES 
- or - 
Task slack time = LF - EF 

Task B has 2 time units of *slack time*
What is the Critical Path?
What is the Near Critical Path?
What is the float of “Choose Soap”?
What is the float of “Mop Floor”?
Fill Bucket with Water 2 Min

Pour Soap into Bucket 1 Min

Find Mop & Bucket 3 Min

Find Broom & Dustpan 3 Min

Empty Dustpan in Garbage 1 Min

Choose Soap 4 Min

Move Obstacles 4 Min

Sweep Floor 8 Min

Sweep Debris into Dustpan 2 Min

Mop Floor 15 Min

Sweep Debris into Dustpan 1 Min

Clean Used Mop & Bucket 4 Min

Put Broom & Dustpan Away 2 Min

Calculate Float Late Start – Early Start or Late Finish – Early Finish

Time
Start-Early-Finish
Start-Late-Finish

Start

End
Fast Tracking – Work critical path activities in parallel, instead of sequentially.

- **Crashing** – Cost and schedule tradeoffs (Time $\approx$ Money) with the end result of reducing overall time to completion

- **Other Ways** (Less preferred but sometimes needed)
  - Reduce Scope/Quality of product
  - Increase resources, Reduce risks
  - Say no: Sometimes schedule compression just isn’t an option
Expected Time = (a + 4m + b)/6
Expected Time = 3.8
Project Tracking
Tracking and Visibility

- Ideal project's visibility
- Efficient project's visibility
- Well-run waterfall-lifecycle-model project’s visibility
- Typical project’s visibility

Project progress
<table>
<thead>
<tr>
<th>Task</th>
<th>Complete?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conceptual Design</td>
<td>Complete</td>
</tr>
<tr>
<td>Program Specification</td>
<td>Complete</td>
</tr>
<tr>
<td>Coding</td>
<td>In Progress</td>
</tr>
<tr>
<td>Documentation</td>
<td>In Progress</td>
</tr>
<tr>
<td>User Manual Production</td>
<td>Not Started</td>
</tr>
<tr>
<td>Testing</td>
<td>Not Started</td>
</tr>
</tbody>
</table>
### Percent Complete

<table>
<thead>
<tr>
<th>Task</th>
<th>How Complete?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conceptual Design</td>
<td>200/200</td>
</tr>
<tr>
<td>Program Specification</td>
<td>300/300</td>
</tr>
<tr>
<td>Coding</td>
<td>150/600</td>
</tr>
<tr>
<td>Documentation</td>
<td>10/100</td>
</tr>
<tr>
<td>User Manual Production</td>
<td>0/400</td>
</tr>
<tr>
<td>Testing</td>
<td>0/500</td>
</tr>
</tbody>
</table>

\[
\frac{660}{2100} \times 100 = 31.4\% \text{ complete}
\]
Earned Value (EV) is a methodology used to control a project.

- It provides a uniform measure for project progress for the entire project or any sub-element.
- Provides a consistent method of project progress and performance.
- Provides a basis for cost performance analysis of a project.
1. Establish a WBS to divide the project into manageable parts
2. Identify the activities required for the current project
3. Allocate the effort required for each activity
4. Schedule the activities over time and resources
5. Analyze/review the schedule
   (continued)
6. Update the schedule by reporting activity progress
7. Enter the actual cost on the activities
8. Execute the Earned Value calculations
9. Analyze the data and make course corrections as necessary
Earned Value

- Establish a common value scale for every task, regardless of the type of work involved (software projects use effort)
- Total effort for the entire project is estimated
- Every task is given a planned value based on its estimated percentage of the total project effort
- Completion of a task results in a credit, or an earned value, of the value allocated to the task
Total Project Effort: 1000 person hours
Task A Estimate: 15 person hours
Planned Value: 1.5

Completing task A contributes 1.5 to the cumulative earned value total for the project
Earned value credit is given only when the task is 100% complete

- Partially completed tasks are NOT given partial credit (in most software projects)
- Large tasks can/must be broken into subtasks
- Size tasks up to 80 person-hours; aim for 2 to 4 task completions per week
Planning, Estimating, Scheduling, and Tracking are a continuum

Projects need to be partitioned for manageability
  - Work Breakdown Structures are a great way to do this

Sequencing Tasks & Activities is vital
  - Gantt Charts allow quick reference
  - Network Techniques such as Precedence Network Diagrams, the Critical Path Method, and PERT Charts are useful tools

Project Tracking is important for project visibility
  - The Earned Value Technique is a key tool in this
Questions/Discussion