Why Are Projects Late?

- an unrealistic deadline established by someone outside the software development group
- changing customer requirements that are not reflected in schedule changes;
- an honest underestimate of the amount of effort and/or the number of resources that will be required to do the job;
- predictable and/or unpredictable risks that were not considered when the project commenced;
- technical difficulties that could not have been foreseen in advance;
- human difficulties that could not have been foreseen in advance;
- miscommunication among project staff that results in delays;
- a failure by project management to recognize that the project is falling behind schedule and a lack of action to correct the problem
Scheduling Principles

- compartmentalization—define distinct tasks
- interdependency—indicate task interrelationship
- effort validation—be sure resources are available
- defined responsibilities—people must be assigned
- defined outcomes—each task must have an output
- defined milestones—review for quality

Effort and Delivery Time

\[ E_d = m \left( t_d^{-0.75} t_o^4 \right) \]

- \( E_d \): effort in person-months
- \( t_d \): nominal delivery time for schedule
- \( t_o \): optimal development time (in terms of cost)
- \( t_d \): actual delivery time desired

These courseware materials are to be used in conjunction with Software Engineering: A Practitioner’s Approach, 6/e and are provided with permission by R.S. Pressman & Associates, Inc., copyright © 1996, 2001, 2005.
**Effort Allocation**

- 40-50% “front end” activities
  - customer communication
  - analysis
  - design
  - review and modification

- 15-20% construction activities
  - coding or code generation

- 30-40% testing and installation
  - unit, integration
  - white-box, black box
  - regression

**Defining Task Sets**

- determine type of project
- assess the degree of rigor required
- identify adaptation criteria
- select appropriate software engineering tasks

These courseware materials are to be used in conjunction with *Software Engineering: A Practitioner’s Approach, 6/e* and are provided with permission by R.S. Pressman & Associates, Inc., copyright © 1996, 2001, 2006.
Task Set Refinement

1.1 Concept scoping determines the overall scope of the project.

- Task definition: Task 1.1 Concept Scoping
  - 1.1.1 Identify need, benefits and potential customers;
  - 1.1.2 Define desired output/control and input events that drive the application;
    - Begin Task 1.1.2 FTR: Review written description of need
      - 1.1.2.1 Derive a list of customer visible outputs/inputs
    - Begin Task 1.1.3 Define the functionality/behavior for each major function;
      - 1.1.3.1 Derive a model of functions/behaviors;
      - 1.1.3.2 Begin Task 1.1.4 Isolate those elements of the technology to be implemented in software;
      - 1.1.5 Research availability of existing software;
      - 1.1.6 Define technical feasibility;
      - 1.1.7 Make quick estimate of size;
      - 1.1.8 Create a Scope Definition;
      - end task definition: Task 1.1

Define a Task Network

Three I.3 tasks are applied in parallel to 3 different concept functions.

Three I.3 tasks are applied in parallel to 3 different concept functions.
### Timeline Charts

<table>
<thead>
<tr>
<th>Tasks</th>
<th>Week 1</th>
<th>Week 2</th>
<th>Week 3</th>
<th>Week 4</th>
<th>Week n</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Task 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Task 3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Task 4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Task 5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Task 6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Task 7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Task 8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Task 9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Task 10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Task 11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Task 12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

These courseware materials are to be used in conjunction with Software Engineering: A Practitioner’s Approach, 6/e and are provided with permission by R.S. Pressman & Associates, Inc., copyright © 1998, 2001, 2006.

---

### Use Automated Tools to Derive a Timeline Chart

#### Use Automated Tools to Derive a Timeline Chart

- **I.1.1 Identify need and benefits**
  - Meet with customers
  - Identify needs and project constraints

- **I.1.2 Define desired output/control/input (OCI)**
  - Scope keyboard functions
  - Scope voice input functions
  - Scope modes of interaction
  - Scope document diagnostics
  - Scope other WP functions

- **I.1.3 Define the functionality/behavior**
  - Define keyboard functions
  - Define voice input functions
  - Define modes of interaction
  - Define spell/grammar check
  - Define other WP functions

- **I.1.4 Isolate software elements**

- **I.1.5 Research availability of existing software**
  - Research text editioning components
  - Research voice input components
  - Research file management components
  - Research Spell/Grammar check components

- **I.1.6 Define technical feasibility**

- **I.1.7 Make quick estimate of size**

- **I.1.8 Create a Scope Definition**
  - Review scope document with customer
  - Revise document as required

- **Milestone: Scope document complete**

These courseware materials are to be used in conjunction with Software Engineering: A Practitioner’s Approach, 6/e and are provided with permission by R.S. Pressman & Associates, Inc., copyright © 1998, 2001, 2006.
Schedule Tracking

- conduct periodic project status meetings in which each team member reports progress and problems.
- evaluate the results of all reviews conducted throughout the software engineering process.
- determine whether formal project milestones (the diamonds shown in Figure 24.3) have been accomplished by the scheduled date.
- compare actual start-date to planned start-date for each project task listed in the resource table (Figure 24.4).
- meet informally with practitioners to obtain their subjective assessment of progress to date and problems on the horizon.
- use earned value analysis (Section 24.6) to assess progress quantitatively.

Progress on an OO Project-I

- **Technical milestone: OO analysis completed**
  - All classes and the class hierarchy have been defined and reviewed.
  - Class attributes and operations associated with a class have been defined and reviewed.
  - Class relationships (Chapter 8) have been established and reviewed.
  - A behavioral model (Chapter 8) has been created and reviewed.
  - Reusable classes have been noted.

- **Technical milestone: OO design completed**
  - The set of subsystems (Chapter 9) has been defined and reviewed.
  - Classes are allocated to subsystems and reviewed.
  - Task allocation has been established and reviewed.
  - Responsibilities and collaborations (Chapter 9) have been identified.
  - Attributes and operations have been designed and reviewed.
  - The communication model has been created and reviewed.
Progress on an OO Project-II

**Technical milestone: OO programming completed**
- Each new class has been implemented in code from the design model.
- Extracted classes (from a reuse library) have been implemented.
- Prototype or increment has been built.

**Technical milestone: OO testing**
- The correctness and completeness of OO analysis and design models has been reviewed.
- A class-responsibility-collaboration network (Chapter 8) has been developed and reviewed.
- Test cases are designed and class-level tests (Chapter 14) have been conducted for each class.
- Test cases are designed and cluster testing (Chapter 14) is completed and the classes are integrated.
- System level tests have been completed.

Earned Value Analysis (EVA)

**Earned value**
- is a measure of progress
- enables us to assess the “percent of completeness” of a project using quantitative analysis rather than rely on a gut feeling
- “provides accurate and reliable readings of performance from as early as 15 percent into the project.” [FLE98]
Computing Earned Value-I

- The *budgeted cost of work scheduled* (BCWS) is determined for each work task represented in the schedule.
  - BCWS is the effort planned for work task $i$.
  - To determine progress at a given point along the project schedule, the value of BCWS is the sum of the BCWS values for all work tasks that should have been completed by that point in time on the project schedule.
- The BCWS values for all work tasks are summed to derive the *budget at completion*, BAC. Hence,
  \[ \text{BAC} = \sum (\text{BCWS}_i) \text{ for all tasks } k \]

Computing Earned Value-II

- Next, the value for *budgeted cost of work performed* (BCWP) is computed.
  - The value for BCWP is the sum of the BCWS values for all work tasks that have actually been completed by a point in time on the project schedule.
  - “the distinction between the BCWS and the BCWP is that the former represents the budget of the activities that were planned to be completed and the latter represents the budget of the activities that actually were completed.” [WIL99]
- Given values for BCWS, BAC, and BCWP, important progress indicators can be computed:
  - Schedule performance index, $\text{SPI} = \text{BCWP}/\text{BCWS}$
  - Schedule variance, $\text{SV} = \text{BCWP} − \text{BCWS}$
  - SPI is an indication of the efficiency with which the project is utilizing scheduled resources.
Computing Earned Value

- Percent scheduled for completion = BCWS/BAC
  - provides an indication of the percentage of work that should have been completed by time t.

- Percent complete = BCWP/BAC
  - provides a quantitative indication of the percent of completeness of the project at a given point in time, t.

- Actual cost of work performed, ACWP, is the sum of the effort actually expended on work tasks that have been completed by a point in time on the project schedule. It is then possible to compute
  - Cost performance index, CPI = BCWP/ACWP
  - Cost variance, CV = BCWP – ACWP