Agenda

- Project Scheduling
  - work breakdown and task definition
  - task dependency (PERT) analysis
  - milestones and Earned Value Analysis
  - resource and staffing dependencies
  - time lines (Gantt Charts)
  - staffing levels (PNR curves)
- Project Status Tracking
  - Using EVA to measure progress
- Automated Scheduling tools

Project Scheduling

- decompose project into distinct tasks and subtasks
- establish dependency relationships among sub-tasks
- identify or define project milestones
- estimate the size of, and work to accomplish each sub-task
- characterize the required resources for each sub-task
- identify or define project milestones
- assign dates to initial tasks
- assign resources to tasks

Work Breakdown

- hierarchical decomposition of work
  - independent sub-tasks that sum to the whole
  - can be based on either tasks or goals
  - both process- and problem-centric
- sub-tasks are fundamental unit of work
  - the granularity of work estimation
  - the granularity of assignment & scheduling
  - the granularity of task inter-dependencies
  - the basis for progress tracking
- granularity will evolve with the project

Task Dependencies

- input/output relationships between tasks
  - may be strict
  - e.g. system test starts after component integration
  - may allow overlap
  - e.g. design can start one month after architecture
  - process-mandated pre-requisites
  - support training must precede beta shipment
- these dictate the order of scheduling
- they also determine the (longest) critical path

Dependencies (PERT charts)

Establish Project Milestones

- Specific and Measurable
  - an objectively ascertainable moment
  - avoid subjective assessments
- Relevant measures of project progress
  - goals achieved, work completed
  - not merely hours of work done
- Timely (relatively closely spaced)
  - enable fine-grained progress tracking
  - accurate assessment of work state and rate
Staffing and Resources

- task resource needs must be described
- projects require people
  - classes of people with specific skills
  - specific people with unique skills
  - these people tend to have many commitments
- other non-sharable resources
  - special laboratories (e.g., usability)
  - special equipment (e.g., a system emulator)
- some of these may be critical resources

The Scheduling Process

- order the tasks based on dependencies
- define the pool of available resources
- assign appropriate resources to each task
  - there will probably be resource conflicts
  - these will create additional dependencies
- start each task as soon as possible
  - as soon as all dependencies are satisfied
  - completion time based on estimate and staffing
  - can be done backwards (from due date)
- display results as a Gantt chart

Rayleigh Staffing Profile Curve

PNR Effort/Time Curve

(there is an optimal size and time)

time to complete project

Timelines (Gantt charts)

Project Management 1A

management

evaluate plan

Mark project per current methodology

process improvement

execute plan

monitor project per current methodology

project post-mortem

update methodology

define and understand the problem

plan solution per current methodology

plan still valid

revise plan

design test stand

execute the plan

monitor project per current methodology

project post-mortem

update methodology

define and understand the problem

plan solution per current methodology

plan still valid

revise plan

design test stand

execute the plan

monitor project per current methodology

project post-mortem

update methodology
Project Status Tracking

- what is each person doing
  - what tasks are they currently working on
  - when will each of these tasks be completed
  - have they encountered any problems
- how is project progressing
  - are resources allocated according to plan
  - is progress proceeding according to plan
  - does someone need a little help
  - does the plan need to be revised
    - changes in problem, resources, approach

A Typical Status Discussion

mgr: Where are we on the app-server?
engr: I’m about 80% done.
mgr: You’ve been 80% done for six weeks!
engr: The first ¼ was easy.
   All the hard stuff is in the last ¼!

Q: How does a project get to be a year late?
A: One day at a time.

Fred Brooks, The Mythical Man-Month

Quantifying Progress

- task completions are obvious milestones
  - they are specific, measurable, relevant
- they may be poor measures of progress
  - not usually evenly spaced measures of work
  - may be too large for fine grained tracking
- we need a different kind of measure
  - to enable fine grained (e.g. daily) tracking
  - to enable meaningful schedule tracking
  - to enable meaningful budget tracking

Earned Value Analysis

- construction size and effort estimates
  - yield an expected cost for each sub-task
  - this is the budgeted value of that sub-task
- the Earned Value of an effort
  - is the value of all the tasks completed so far or ... ¼ earned at start, ¾ earned at completion or ... partial value for progress (e.g. tests passed)
- Tracking Earned Value enables us to
  - assess project completion and speed
  - meaningfully assess cost-performance

The Need for Automation

- schedules are revised continuously
  - tweaking to get a schedule that works
  - changes to tasks and estimates
  - changes in available resources
  - updates to reflect actual progress
- automation is essential
  - task descriptions maintained in a database
  - automatic schedule generation
  - planned vs. actual comparisons

Tracking with Earned Value

Comparing Earned Value with Planned value enables us to determine...
(a) what fraction of the project is complete
(b) whether or not the work is on schedule

Comparing Earned Value with Actual Cost enables us to...
(a) determine whether or not we are within budget (on the work completed so far)
(b) estimate the cost to completion, based on the performance so far.
Scheduling Tools

• define project
  – describe all tasks, sub-tasks and milestones
  – describe all resources
  – describe all dependencies and constraints
• automatically produce schedules
  – which meet all specified constraints
  – this is very difficult to do by hand
• enter task and resource status information
  – generate reports on progress and problems

For Next Lecture

• McConnell, chapter 6, section 34.4
  – class design, programming into your language
• Spolsky: What are specs
  – the starting point for your design
• Gabriel/Steele: Objects have(n’t) failed
  – promises and disappointments of the OO revolution
• Selected class patterns
  – a few front-ending and instantiation tricks
• UML class, package and object diagrams