COURSE NAME

SOFTWARE ENGINEERING

CSC 3114

(UNDERGRADUATE)

# CHAPTER 15

# PROJECT SCHEDULING

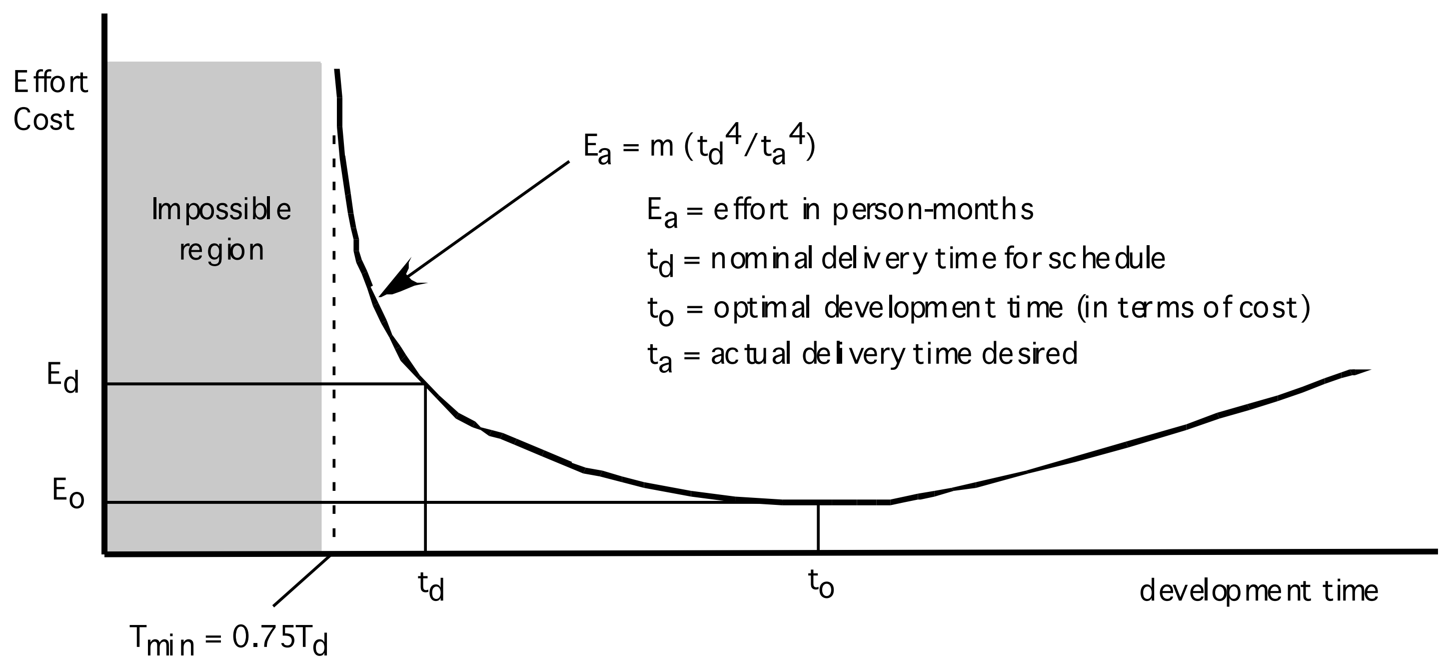
# WHY ARE PROJECTS LATE?

- 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

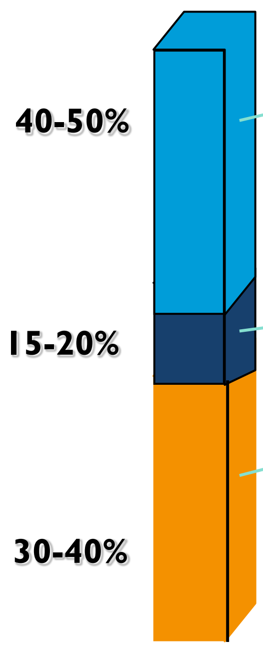
# SCHEDULING PRINCIPLES

- 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

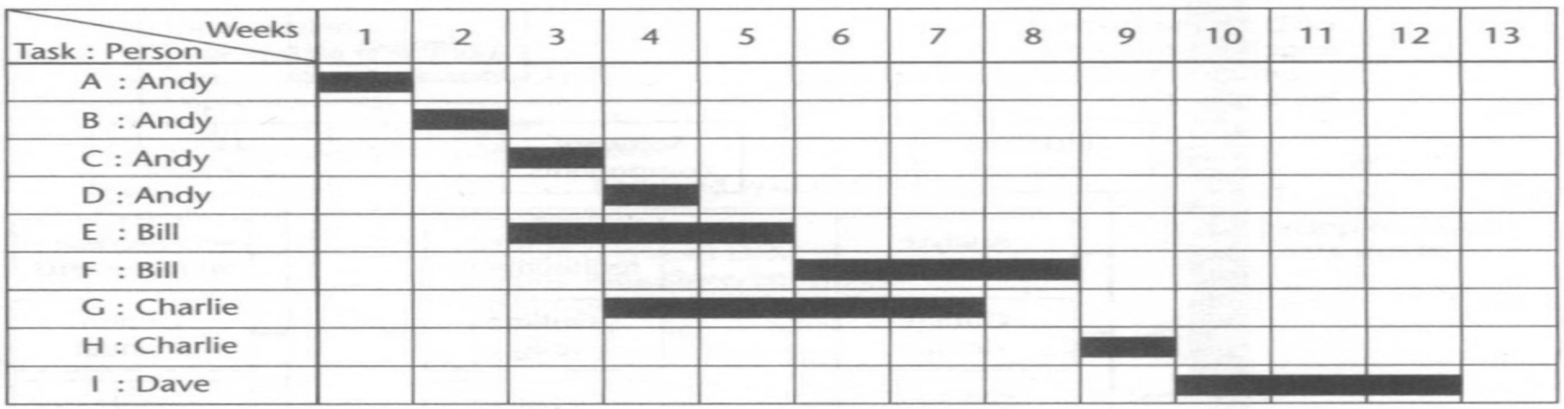


# EFFORT ALLOCATION (40-20-40 RULE)




- "front end" activities
  - customer communication
  - analysis
  - design
  - review and modification
- construction activities
  - coding or code generation
- testing and installation
  - unit, integration
  - white-box, black box
  - regression

# TIMELINE CHARTS

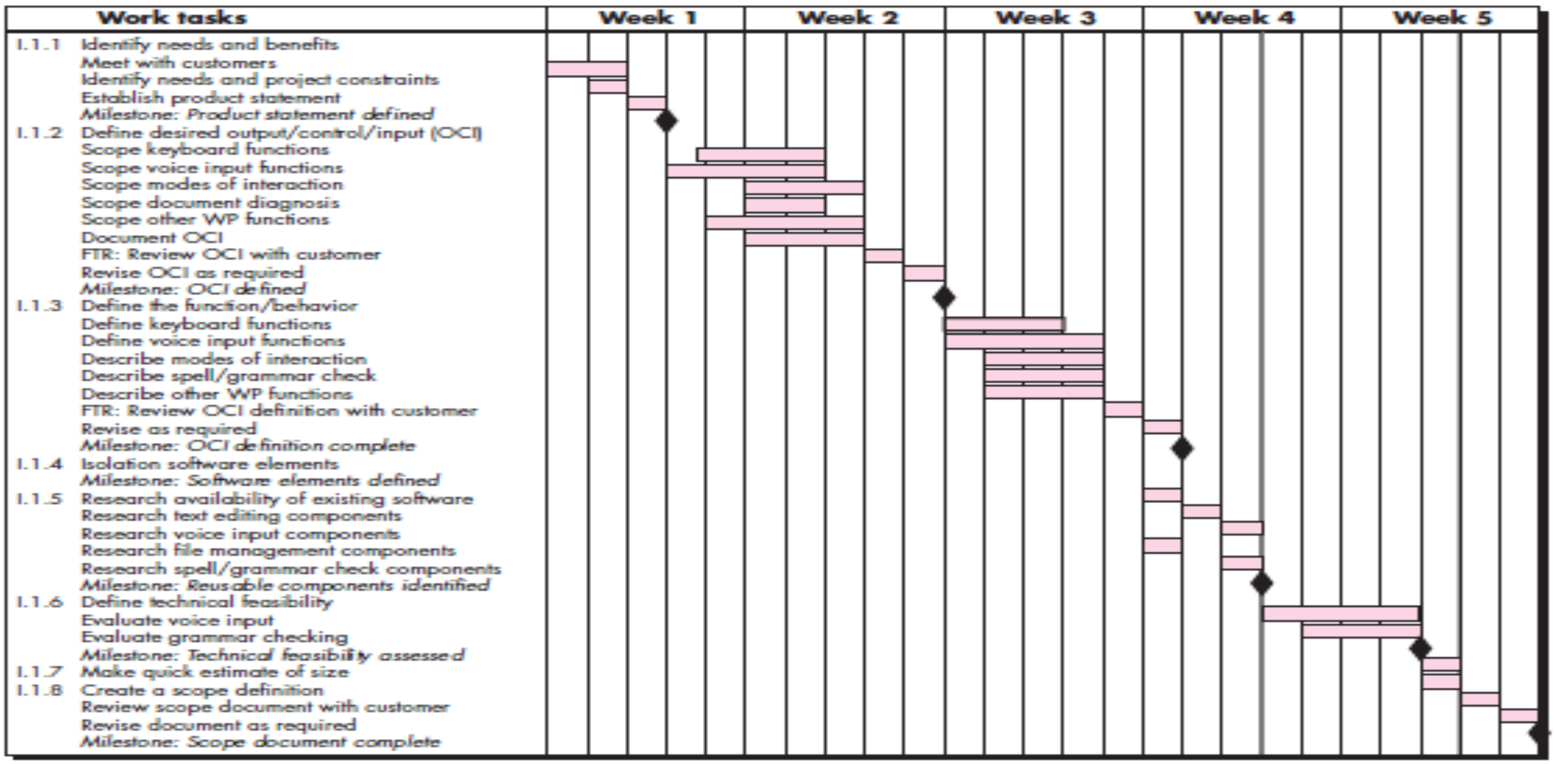



Activity key:

A : Overall design
B : Specify module 1
C : Specify module 2
D : Specify module 3
E : Code module 1
F : Code module 3
G : Code module 2
H : Integration testing
I : System testing

**Figure 6.6** A project plan as a bar chart

# TIMELINE CHARTS



# 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 27.3) have been accomplished by the scheduled date.
- Compare actual start-date to planned start-date for each project task listed in the resource table.
- Use **earned value analysis** to assess progress quantitatively.

# 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."

# COMPUTING EARNED VALUE

- The *budgeted cost of work scheduled* (BCWS) is determined for each work task represented in the schedule.
- $BCWS_i$ 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 $BCWS_i$ 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,

$$BAC = \sum (BCWS_k) \text{ for all tasks } k$$

# COMPUTING EARNED VALUE

- *Budgeted cost of work performed* (BCWP)
- 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."
- Given values for BCWS, BAC, and BCWP, important progress indicators can be computed:
- Schedule performance index, SPI = BCWP/BCWS
- SPI tells you how efficiently you are actually progressing compared to the planned progress.
- SPI is an indication of the efficiency with which the project is utilizing scheduled resources.
- Schedule variance, SV = BCWP – BCWS

# 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

  The Cost Performance Index helps you analyze the efficiency of the cost utilized by the project.

- Cost variance, CV = BCWP – ACWP

# EVA EXERCISE

- Assume you are a software project manager and you've been asked to compute earned value statistics for a small software project.
- The project has 56 planned work tasks that are estimated to require 582 person-days to complete.
- At the time that you've been asked to do the earned value analysis, 12 tasks have been completed.
- However the project schedule indicates that 15 tasks should have been completed.
- The following scheduling data (in person-days) are avalable:

# EVA EXERCISE

| Task | Planned Effort | Actual Effort |
|---|---|---|
| 1 | 12.0 | 12.5 |
| 2 | 15.0 | 11.0 |
| 3 | 13.0 | 17.0 |
| 4 | 8.0 | 9.5 |
| 5 | 9.5 | 9.0 |
| 6 | 18.0 | 19.0 |
| 7 | 10.0 | 10.0 |
| 8 | 4.0 | 4.5 |
| 9 | 12.0 | 10.0 |
| 10 | 6.0 | 6.5 |
| 11 | 5.0 | 4.0 |
| 12 | 14.0 | 14.5 |
| 13 | 16.0 | — |
| 14 | 6.0 | — |
| 15 | 8.0 | — |

BCWP= 126.50

BCWS= 156.50

ACWP= 127.50

Given Total Task = 56; Effort Estimated= 582 Person Day

- BAC = 582.00
- SPI = BCWP/ BCWS = 126.5/ 156.5 = 0.808307
- SV = BCWP - BCWS = 126.5 - 156.5 = -30 person-day
- CPI = BCWP/ ACWP = 0.99
- CV = BCWP – ACWP = -1 person-day
- % schedule for completion = BCWS/ BAC = 156.5/ 582.00 = 26.89%

  [% of work scheduled to be done at this time]
- % complete = BCWP/ BAC = 126.5/ 582.00 = 21.74%
  [% of work completed at this time]

# REFERENCES


- R.S. Pressman & Associates, Inc. (2010). *Software Engineering: A Practitioner's Approach.*
- Kelly, J. C., Sherif, J. S., & Hops, J. (1992). An analysis of defect densities found during software inspections. *Journal of Systems and Software*, *17*(2), 111-117.
- Bhandari, I., Halliday, M. J., Chaar, J., Chillarege, R., Jones, K., Atkinson, J. S., & Yonezawa, M. (1994). In-process improvement through defect data interpretation. *IBM Systems Journal*, *33*(1), 182-214.