CSSE 372 Software Project Management:
Critical Chain Method

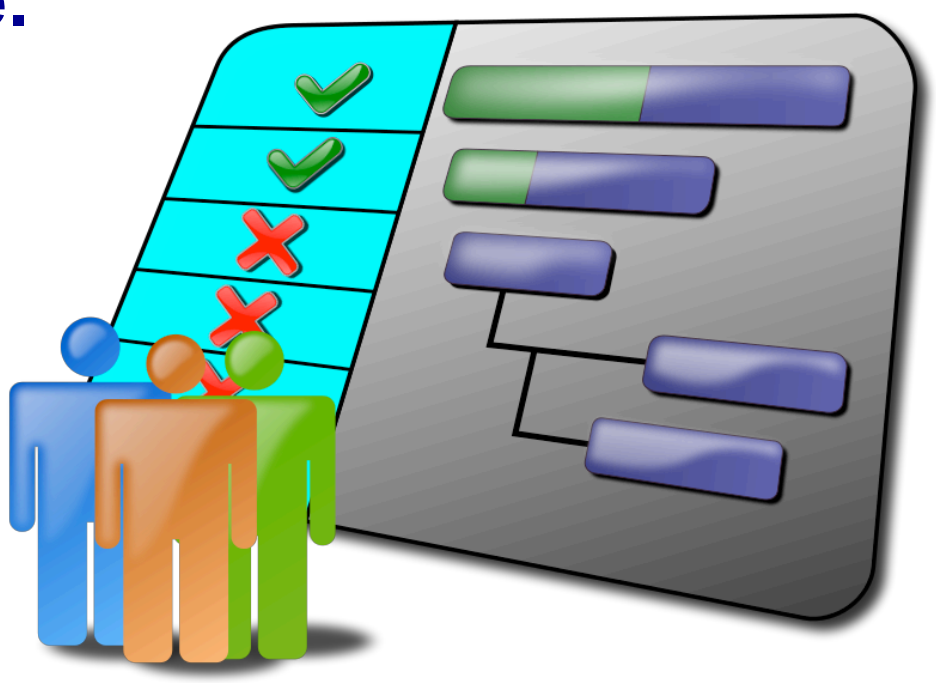
Shawn Bohner
Office: Moench Room F212
Phone: (812) 877-8685
Email: bohner@rose-hulman.edu

ROSE-HULMAN
INSTITUTE OF TECHNOLOGY

Learning Outcomes: Schedule

Create and maintain a software project schedule.

- Introduce Critical Chain
- Examine Critical Chain Project Management
- Do Examples of Critical Chain problems
- Short discussion from Justin Hutchings of Microsoft



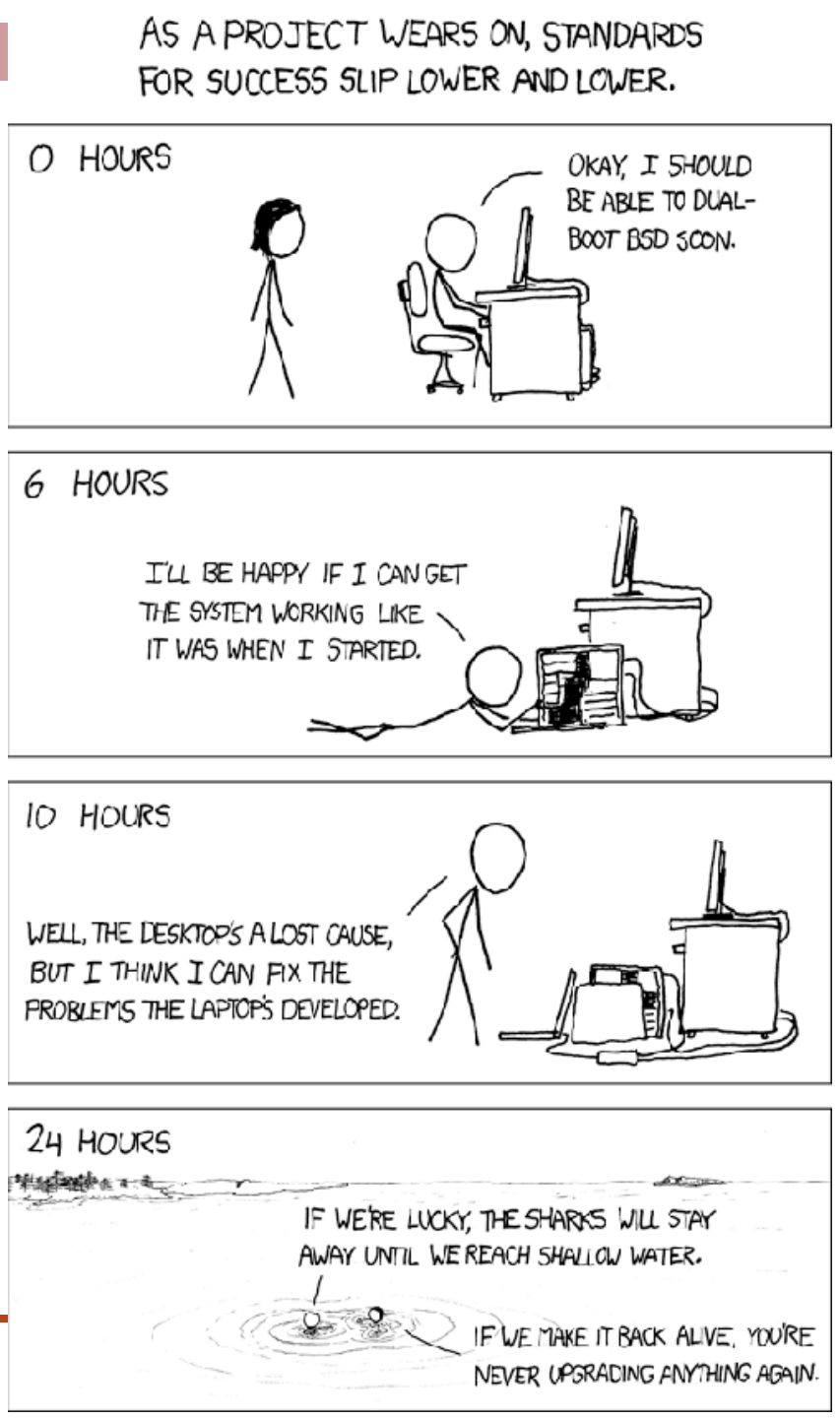
Do you think that Critical Path Method is an optimistic or pessimistic determiner of schedule? Why?

- Think for 15.678 seconds…
- Turn to a neighbor and discuss it for a minute

Address Avoidance

Parkinson Law (Avoidance)

- Build schedule with tight duration estimates that discourage diversion of attention/focus
- Get rid of task due dates!?!
- Get management to protect resources from disruptions



Address Protection

Murphy Law (Protection)

- Avoid distractions and interruptions
- Single focus when possible
- Account for working on more than one project at a time

What is Critical Chain?

Critical chain is the longest duration path through the project, if you consider both task dependencies (precedence) and resource constraints

- **Based on Theory of Constraints**

Eli Goldratt

An Insurance Analogy

Consider fire insurance for five homes…

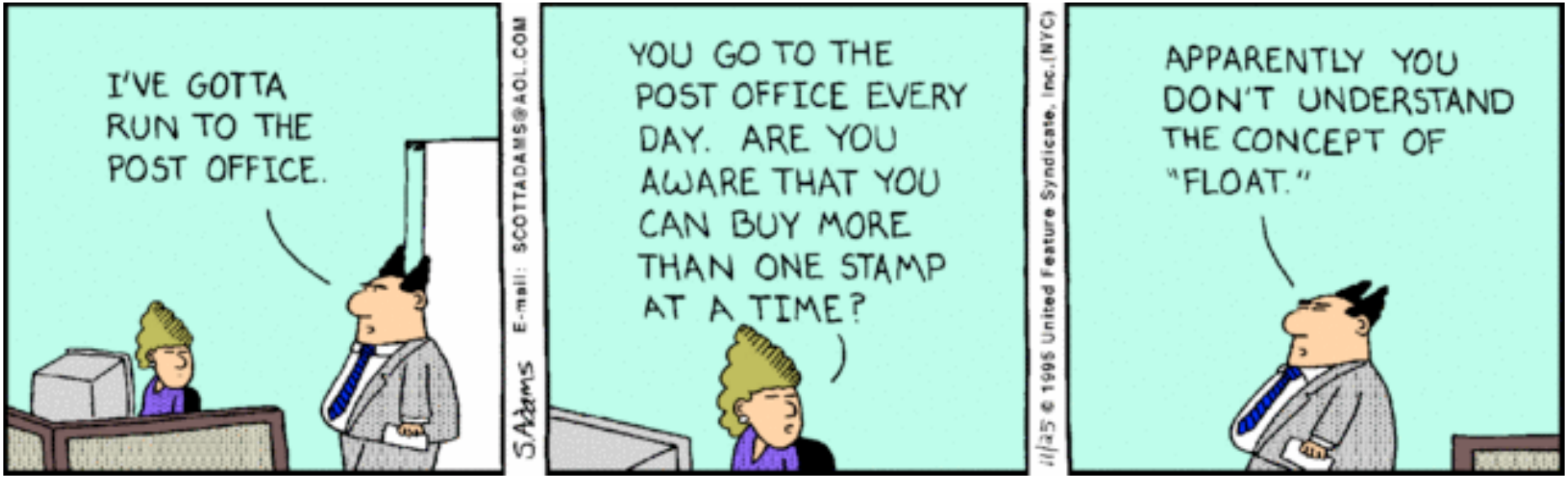
- Critical Path Method way
  - Assume each home *will* suffer a serious problem
  - So, each home self-insures
  - May not be enough $ to cover a serious problem

- Critical Chain Way
  - Assume that not every home will get hit with a serious problem
  - Pool the contingency resources
  - Enough to cover a serious problem, possibly more

What's Wrong with CPM?

- Buffer built into each individual estimate can be overkill and inefficient
- Tracking individual task due dates to indicate project health encourages:
  - Estimates → commitments (Protection)
  - Work expands to fill time available (Avoidance)
- Encourages individual ownership of individual commitments
  …rather than team ownership of the team commitment

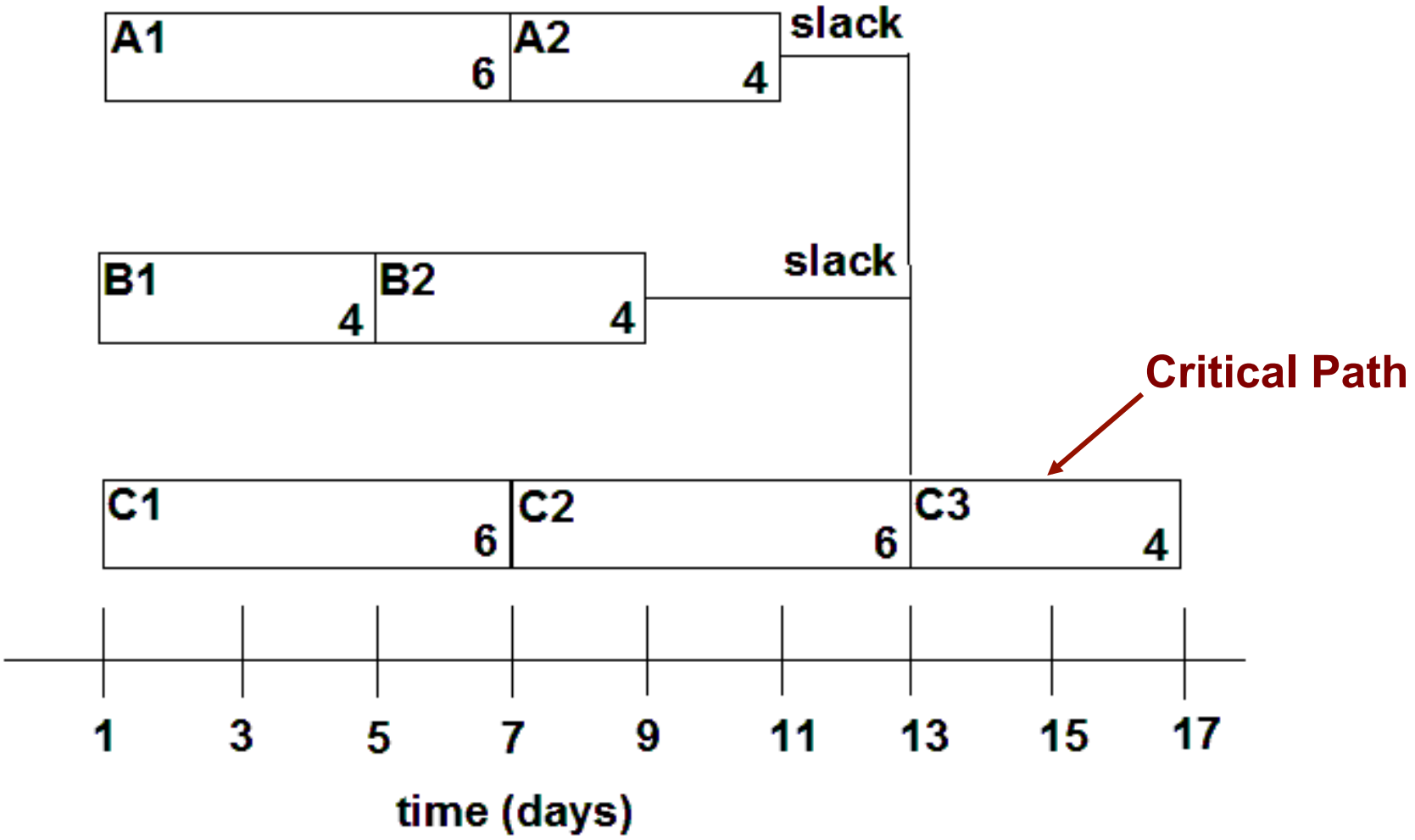
Dilbert on Critical Path Method



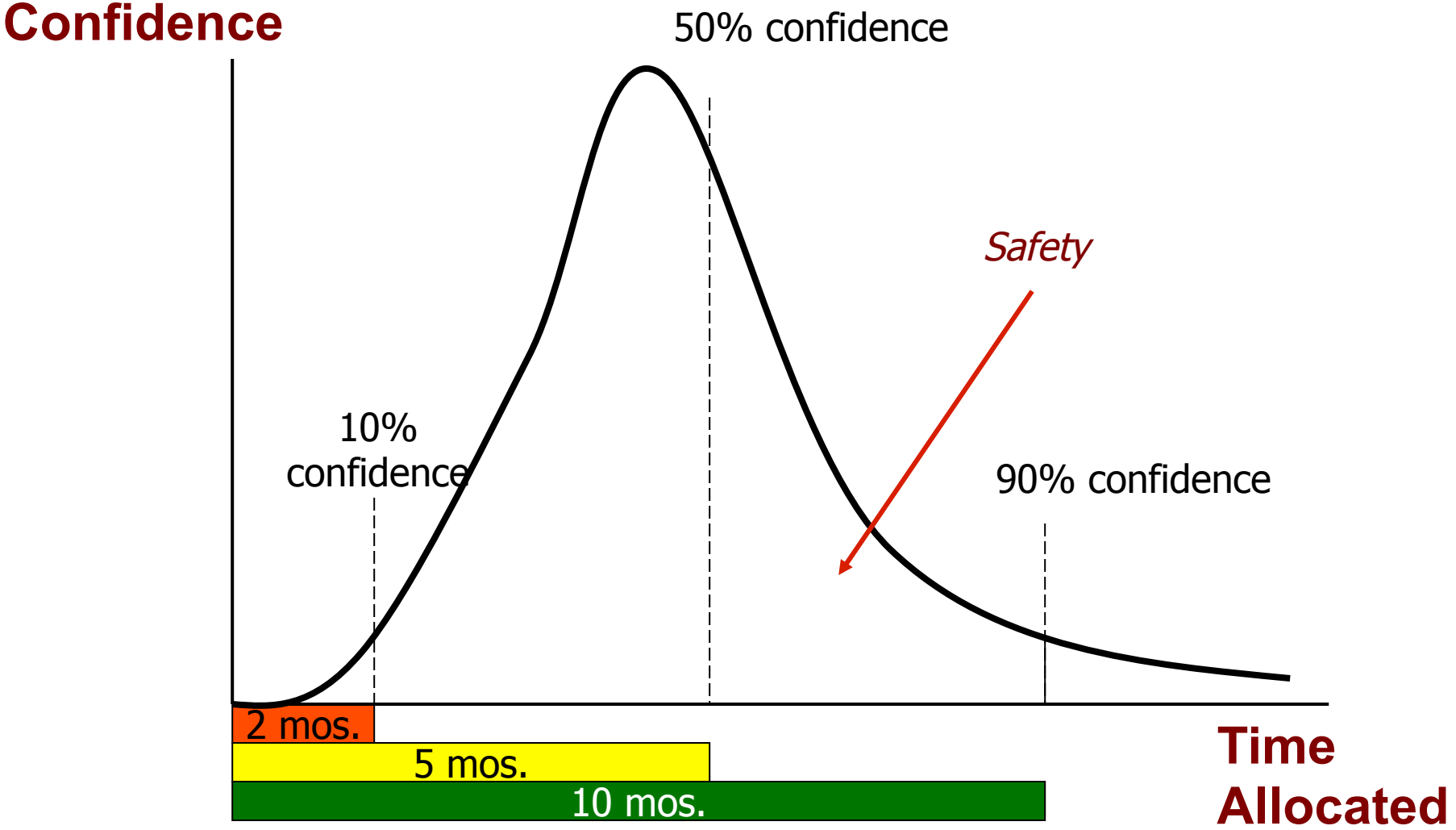
What's different about Critical Chain?

- Reorganizes risk into manageable safety buffers
- Focus on buffer utilization for project health
- Focus on start dates
- Exploit early finishes
- Focus on aggressive, targeted completion of tasks

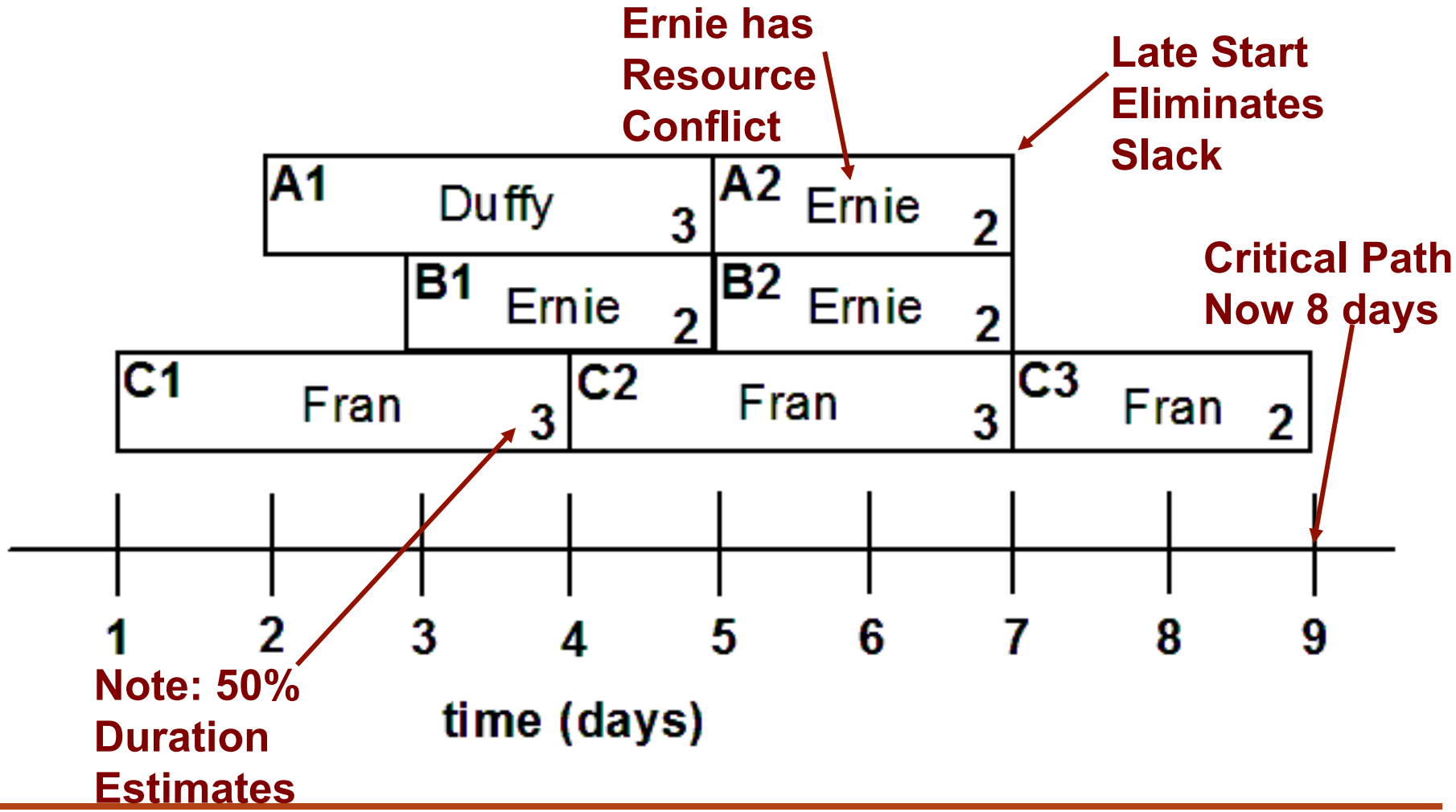
Step 1: Create the Early Schedule



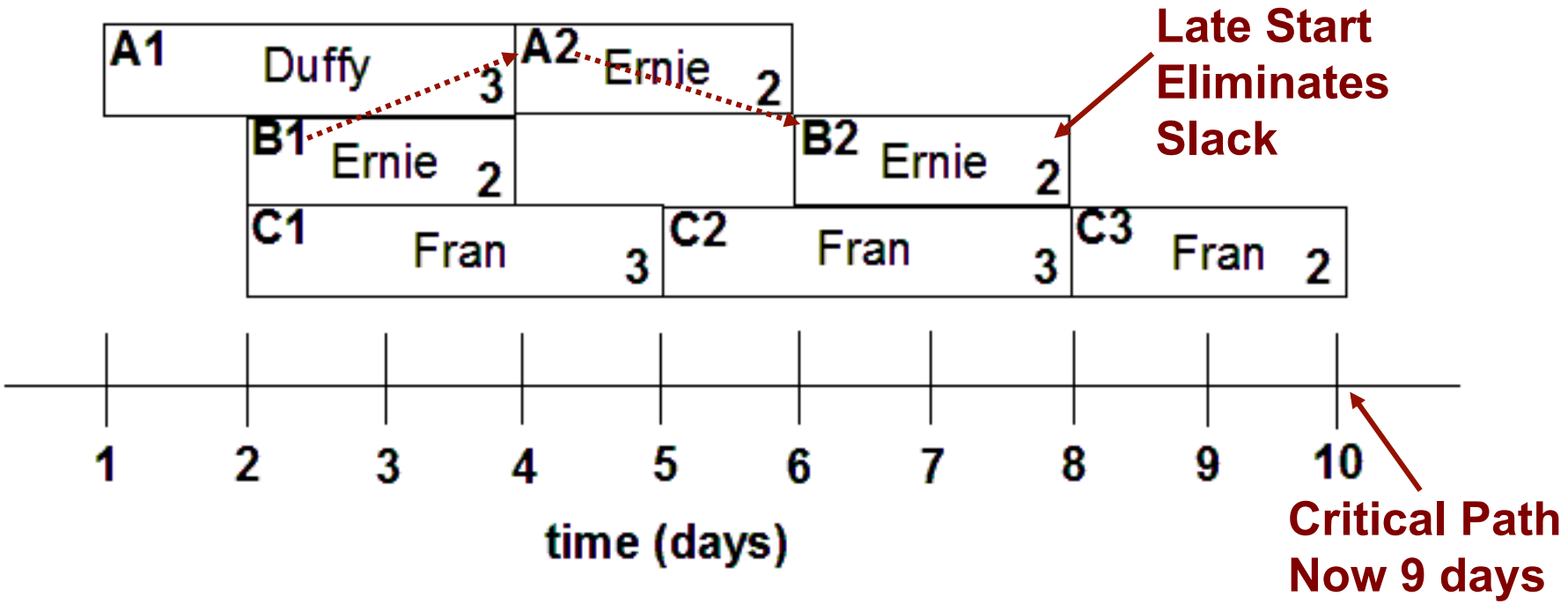
Estimation Variation



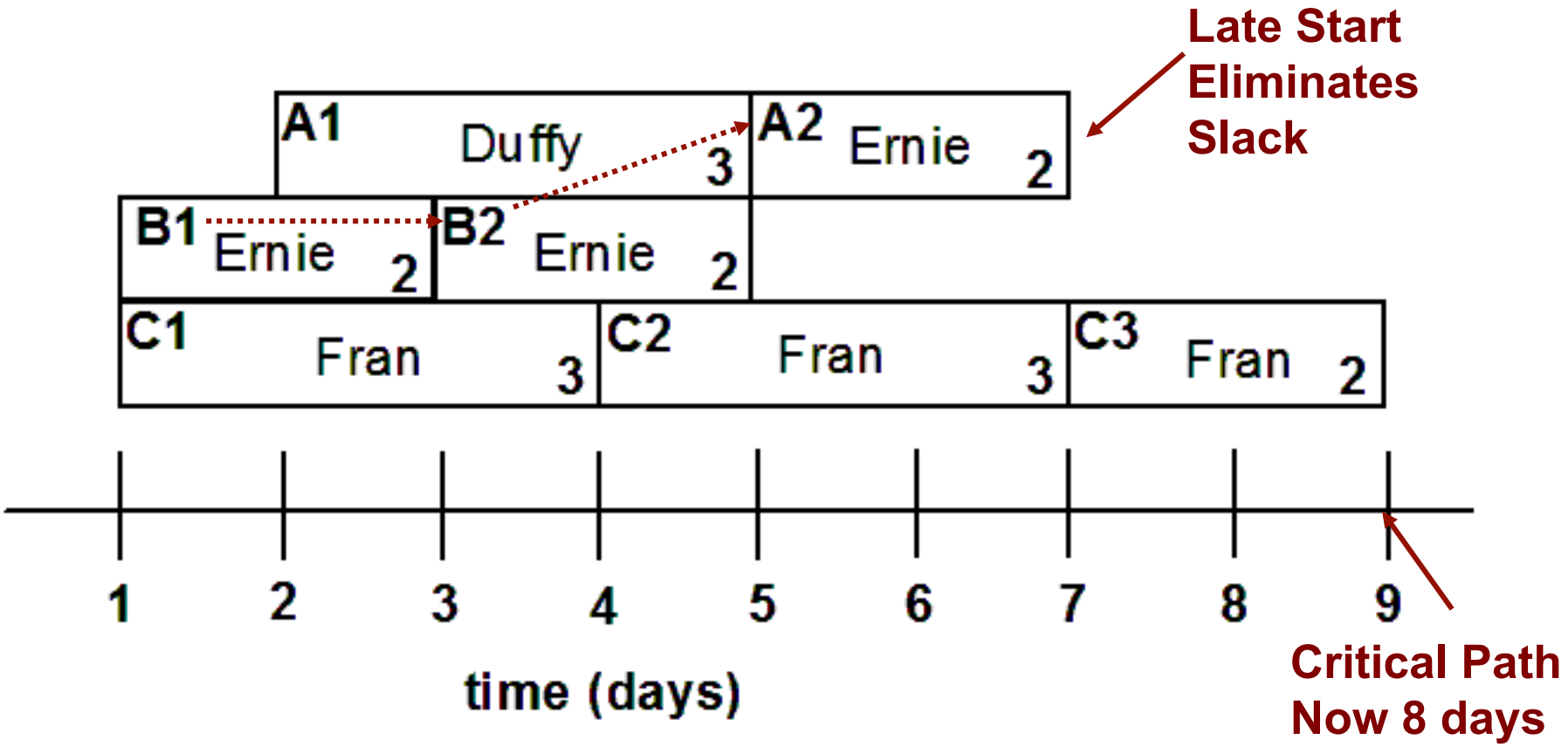
Step 2: Convert to Late Schedule with 50% Durations & Resources Added



1st Way to Resolve Resource Conflict



2nd Way to Resolve Resource Conflict

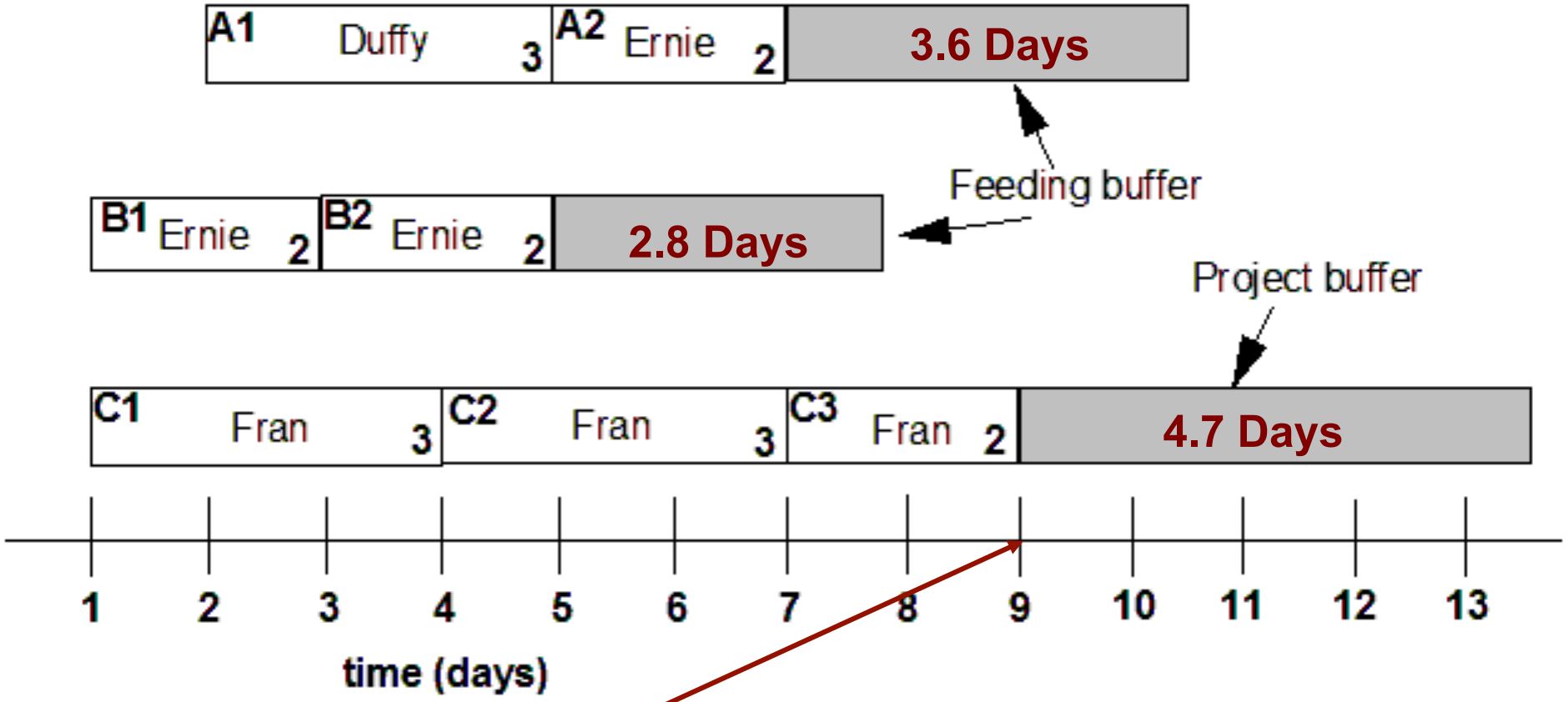


Critical Chain Buffer Management

Buffers are segments of time that are placed at the end of a sequence of tasks for the purpose of protecting the schedule of those tasks

- **Feeding Buffer:** Provides contingency to keep tasks not on the critical path off the critical path
- **Project Buffer:** Provides contingency for the entire project
- **Resource Buffer:** Provides a wakeup call to alert resources to be ready to work on critical tasks
- Others include:
  Capacity, Cost, and Drum Buffers

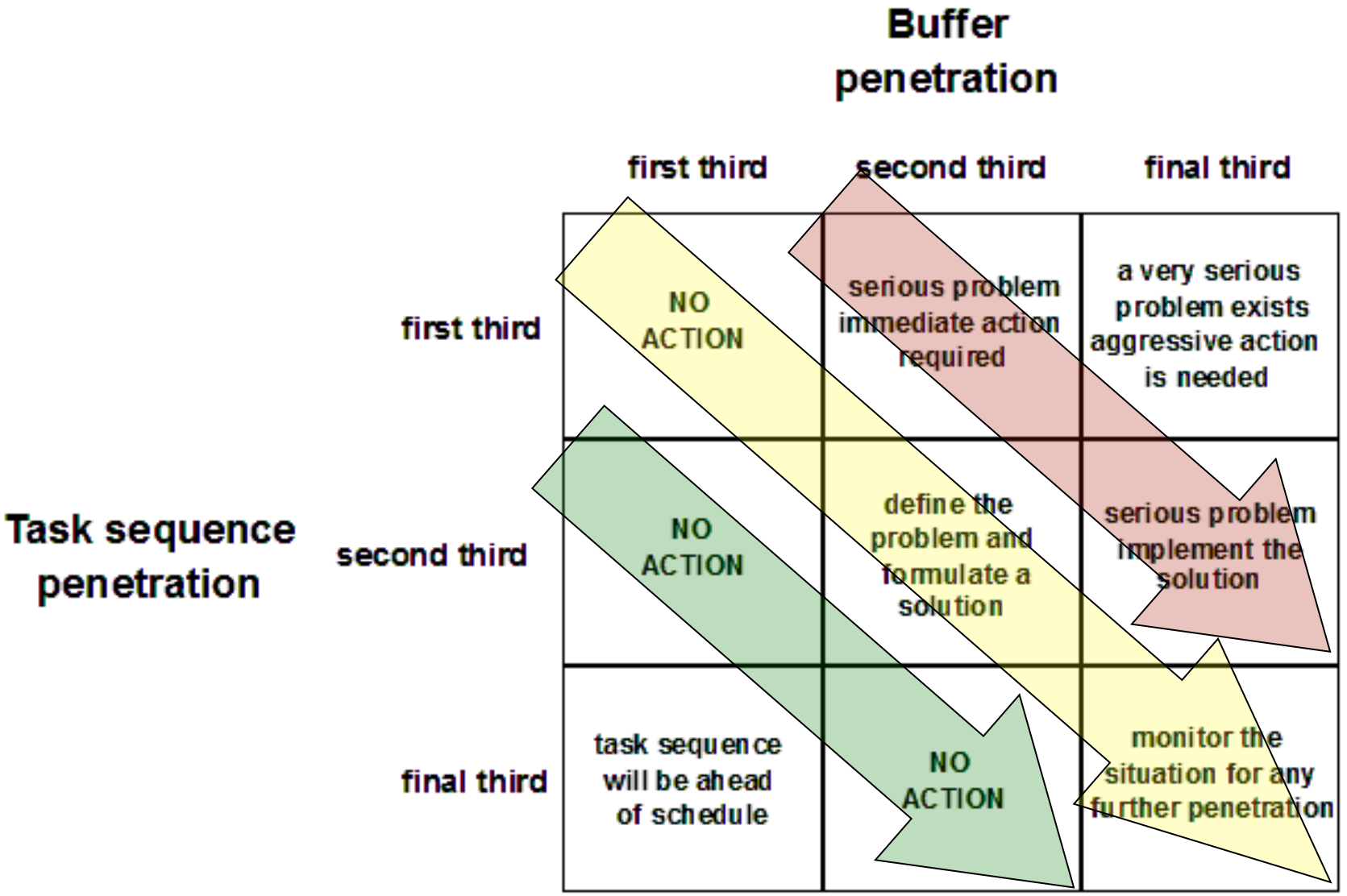
Project Schedule with Buffers Added



Optimistic 8 days (no risk realized), But, even Pessimistic is less than original 16 days

$$\left(\text{Variance}\right)_{path} = \sqrt{\sum \left(\text{Variance}\right)^2_{task}}$$

Buffer Penetration and Action Decisions

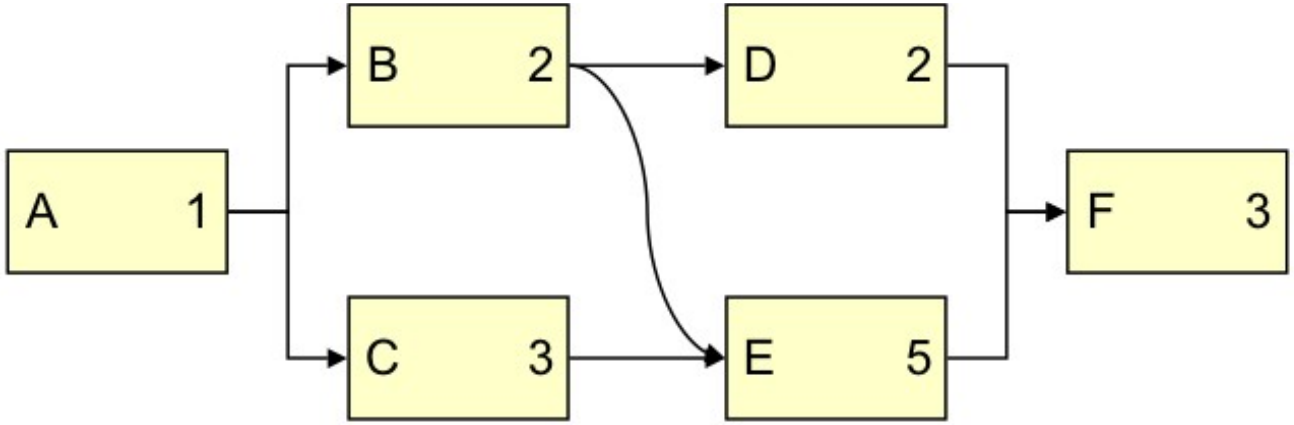


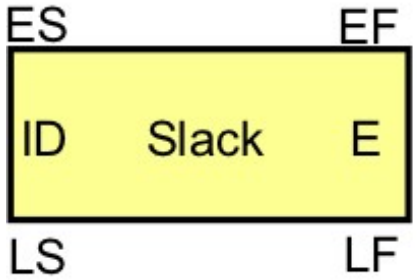
Creating a Critical Chain Schedule

1. Build WBS
2. Identify dependencies
3. Gather estimates with 50% confidence
4. Build your project schedule using Late Finish (rather than “early start”)
5. Resolve resource contentions (leveling)
6. Identify the critical chain
7. Exploit the critical chain to identify any resequencing that can shorten the duration
8. Add the Project Buffer to the end of the critical chain
9. Add Feeding Buffers to all non-critical chains that feed the critical chain
10. Resolve next most relevant resource contention

Critical Chain Exercise (if time)

- WBS done, start with network below
- Based on 50% estimate, determine late-finish schedule
- Allocate resources and deal with resource contention
- Add project and feeding buffers





ES = earliest start, EF = earliest finish time

LS = latest start, LF = latest finish time

E = Effort (duration)

Comparing CPM/PERT and CCPM

CPM/PERT	Critical Chain
Uses worst-case estimates	Uses average-case estimates
Protects individual tasks with safety	Protects project with buffers
Starts and finishes tasks at scheduled start and finish times	Starts tasks as soon as predecessors are done, finishes tasks as quickly as possible.
Individual ownership of task completion	Team ownership of project completion
Project health is based on individual task completion	Project health is based on days used from the project buffer

Homework and Reading Reminders

- Complete Homework 5 – Software Schedule
  - Due by 11:55pm, Tuesday, October 9th, 2012