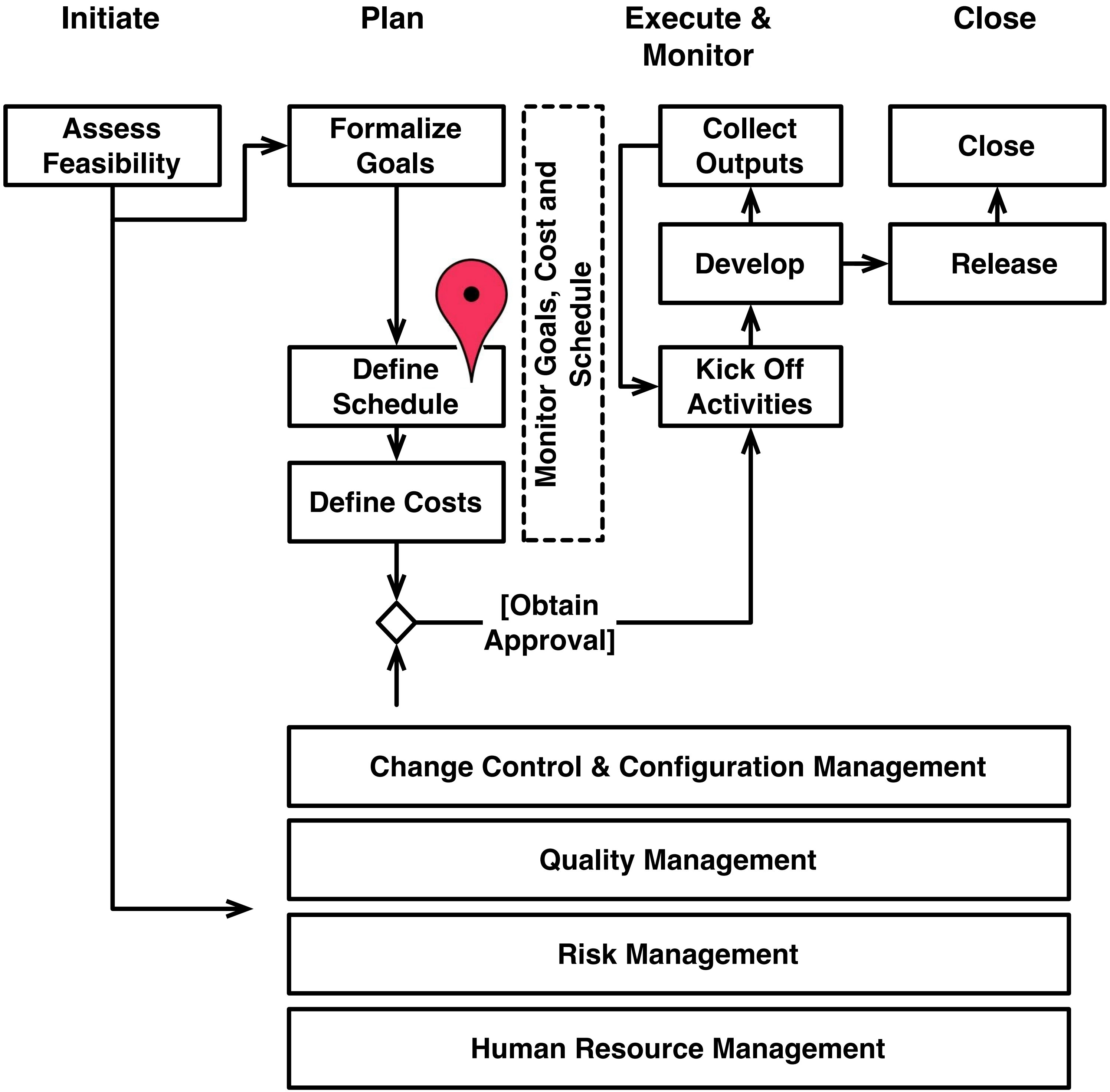
Optimizing the Plan

Goals of the Unit

- Learning the techniques to shorten a plan
- Understanding the risks of shortening a plan


Initiate
Plan
Execute & Monitor
Close
Assess Feasibility
Formalize Goals
Define Schedule
Define Costs
Monitor Goals, Cost and Schedule
Collect Outputs
Develop
Kick Off Activities
Close
Release
[Obtain Approval]
Change Control & Configuration Management
Quality Management
Risk Management
Human Resource Management

Preliminary Consideration

- The first version of your plan will most likely show that you will not be able to deliver according to the deadline set by the sponsor
- On the hypothesis that the plan is accurate, there are two options:
  - The project is not started (since it is not feasible, given the constraints)
  - The project is shortened, using one of the techniques described in this unit
- The “third” option (the estimations are revised) is a bad idea

Making your plan feasible

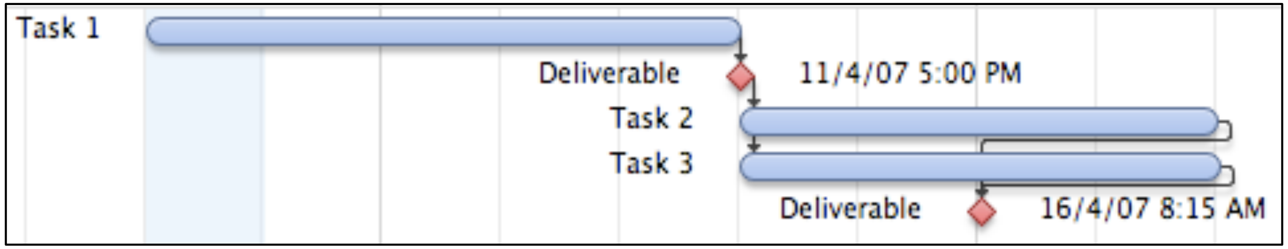
- Change some of the hypotheses on which your plan is based. For instance:
  1. Reduce scope (it makes some activities shorter or useless)
  2. Reduce quality (it makes some activities shorter or useless)
  3. Outsource some activities (it increases risk and, possibly, costs)
- Act on the logic of the plan:
  4. Increase resources (it makes the project more costly) [**PROJECT CRASHING**]
  5. Evaluate alternative courses of actions (e.g. change development process). However: it might be difficult, if your team is accustomed to a specific development process.
  6. Substitute activities
  7. Break the rules: eliminate hard constraints on your plan (it increases risk, and, possibly costs, due to re-work) [**FAST TRACKING**]
  8. Work on probability of delivering on time [**CRITICAL CHAIN MANAGEMENT**]

Project Crashing

Project Crashing

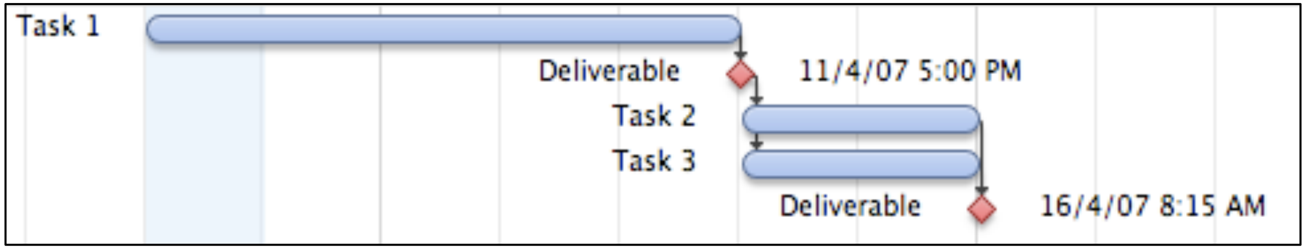
- Project duration can (often) be reduced by assigning more labor to project activities, in the form of over time, and by assigning more resources, such as material, equipment, etc.
- However, the additional labor and resources increase the project cost.
- So, the decision to reduce the project duration must based on an analysis of the trade-off between time and cost.
- Project crashing is a method for shortening the project duration by reducing the time of one or more of the critical project activities to less than its normal activity time. The object crashing is to reduce project duration while minimizing the cost of crashing.

Project Crashing Example



cost = $ 3000
cost = $ 4000

Project Crashing: we allocate more people to Task 2 an 3



cost = $ 6000
cost = $ 8000

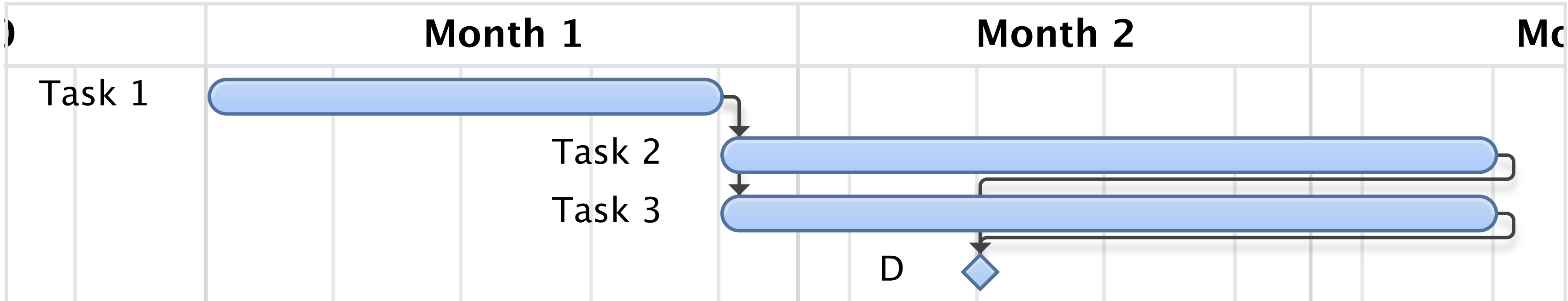
CRASH TIME = 2 calendar units (e.g. months)

CRASH COST = (6000 + 8000) - (3000 + 4000) =
= 14000 - 7000 = 7000

Crashing convenient according to costs of delivering late

Project Crashing (simple) Example

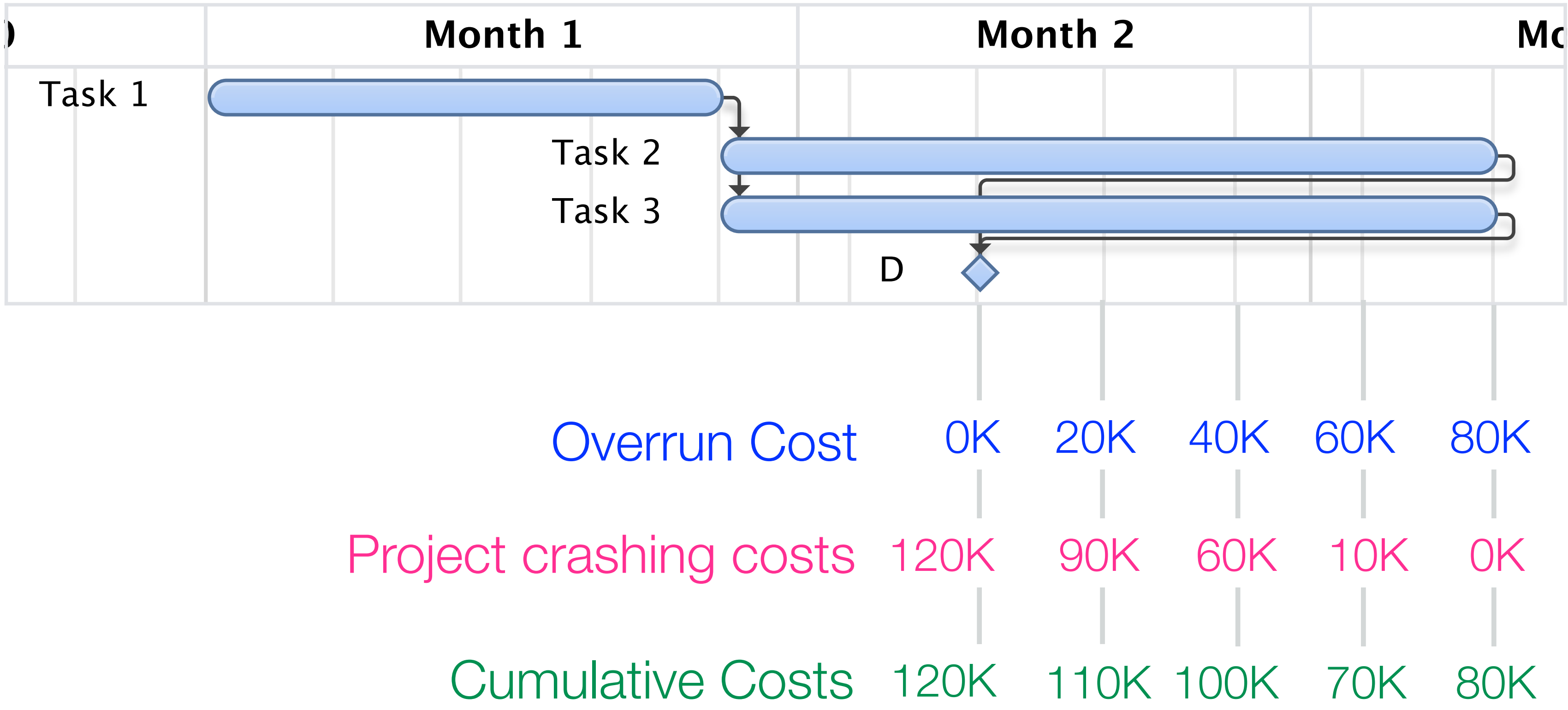
- Project is late: we are supposed to pay 20K euro per week of delay



- We can reduce Task 2 and Task 3 by allocating more resources: 10K for a week, 60K for two weeks, 90K for three weeks, 120K for four weeks)

Project Crashing (simple) Example

- Project is late: we are supposed to pay 10K euro per week of delay



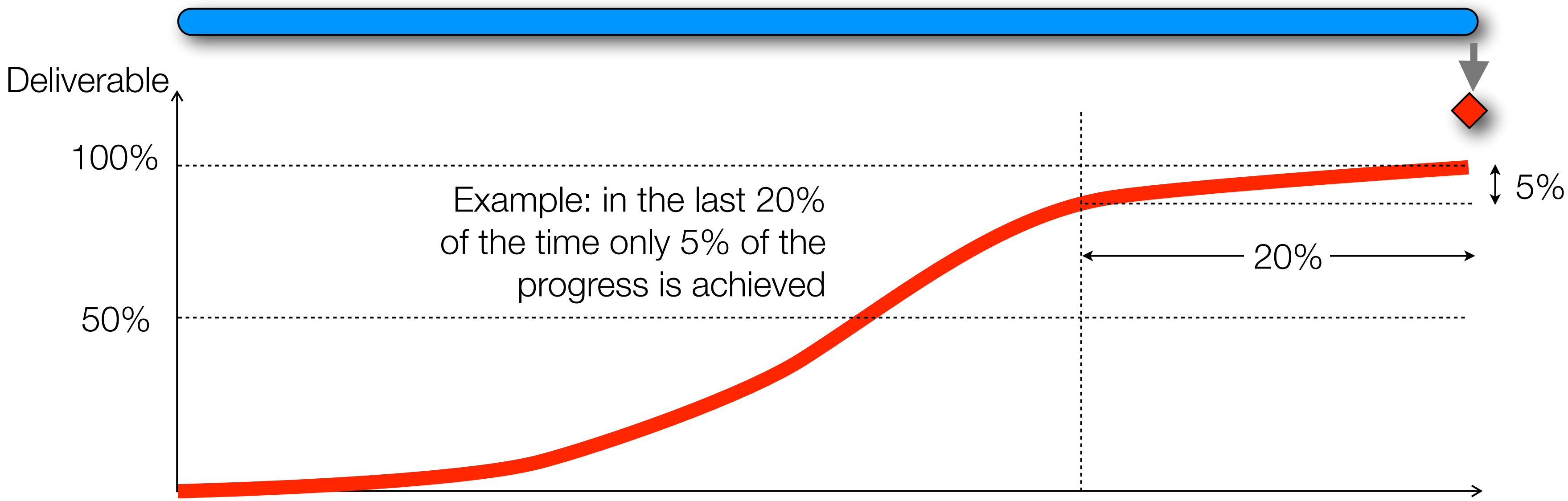
Project Crashing (simple) Example

- Best solution (only if costs matter): shorten by a week (cumulative costs is the minimum value)
- Other considerations might make other choices more appealing. For instance:
  - Delivering on time might be essential to keep a client. In such a case we might decide to incur in increased expenditures
  - We might decide to deliver late (and keep activities as they are) to avoid having our team work overtime or because risks associated to decreased quality might increase with shortening activities

Fast Tracking

Fast Tracking

- Fast tracking is based on the fact that deliverables of activities are incrementally produced (and refined) during the execution of the activity
- Example: a requirement document is not written on the last day of the "requirement writing" activity. Rather it gets written a bit at a time, when the activity is executed

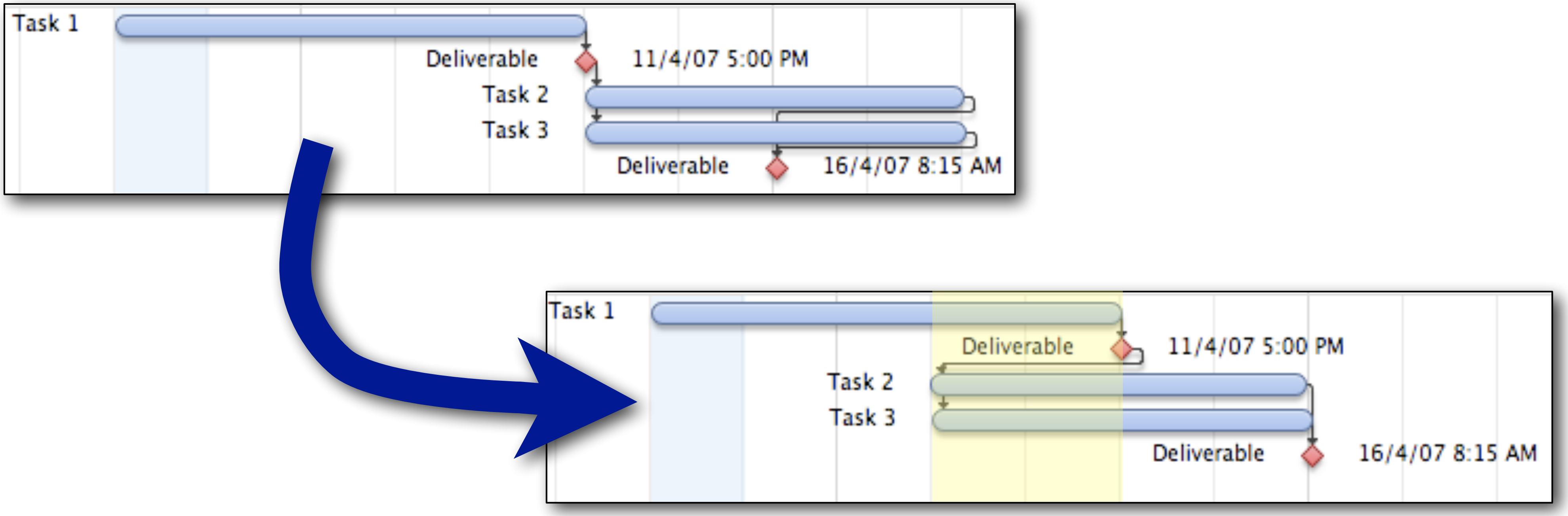


Fast Tracking

- Consider the example of the previous slide: if the last 5% is not the critical part of your deliverable, then we could start the activities depending on the requirement document earlier.

Fast Tracking

- Fast tracking works by overlapping activities which would otherwise be sequential



Fast Tracking: Issues and Rules of the Thumb

- Fast tracking is risky and it might cause rework
- When deciding what dependencies are better to break, consider the following:
  - How the deliverable production during the activity will progress (will it produce intermediate outputs?) and consolidate (will the intermediate output be stable?)
  - The risk involved in changes to the output (what if the consequence of re-work in the subsequent activity; how will it affect the rest of the plan?)

Critical Chain Management

Critical Chain Management

- Critical Chain Management starts from the assumption that estimations are averages guesses
- Saying that an activity lasts X (or it takes an effort of Y) means that **most of the times** the activity will take X to complete. **There could be cases,** however, in which the activity will take shorter or longer to complete
- To manage contingencies (and contrast their optimism), project managers pad their estimations, moving the estimations to the "pessimistic" side
- However, only some of the buffers will actually be needed

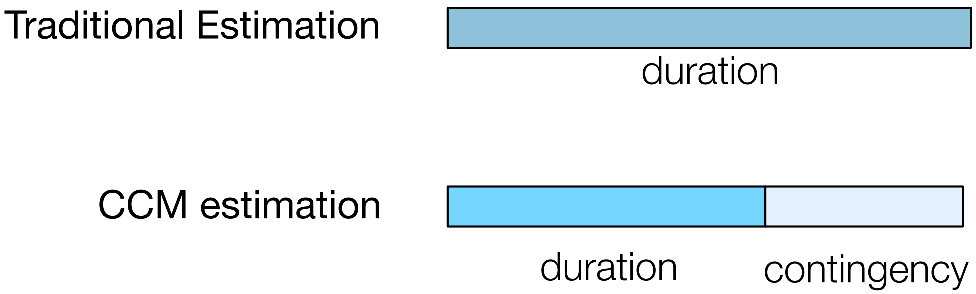
Some statistical considerations

- Critical Chain Management is interesting and effective for two reasons:
  - **Item 1. We reason on most probable estimates rather than pessimistic estimates:** in particular we call contingency the difference in duration between a 50% probable estimate and a 90% probable estimate
  - **Item 2.** We reason on **chains of activities**. The **standard deviation** of a chain of activities is less than the sum of the standard deviations of the activities in the chain

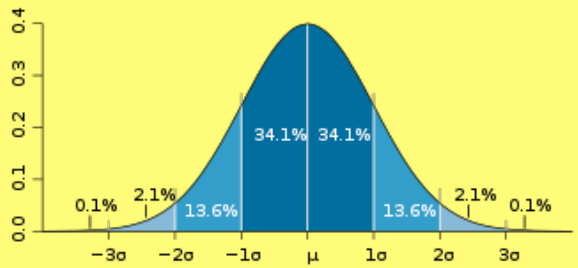
$$\sigma_{a+b} = \sqrt{\sigma_a^2 + \sigma_b^2}$$

$$\sigma_{a+b} < \sigma_a + \sigma_b$$

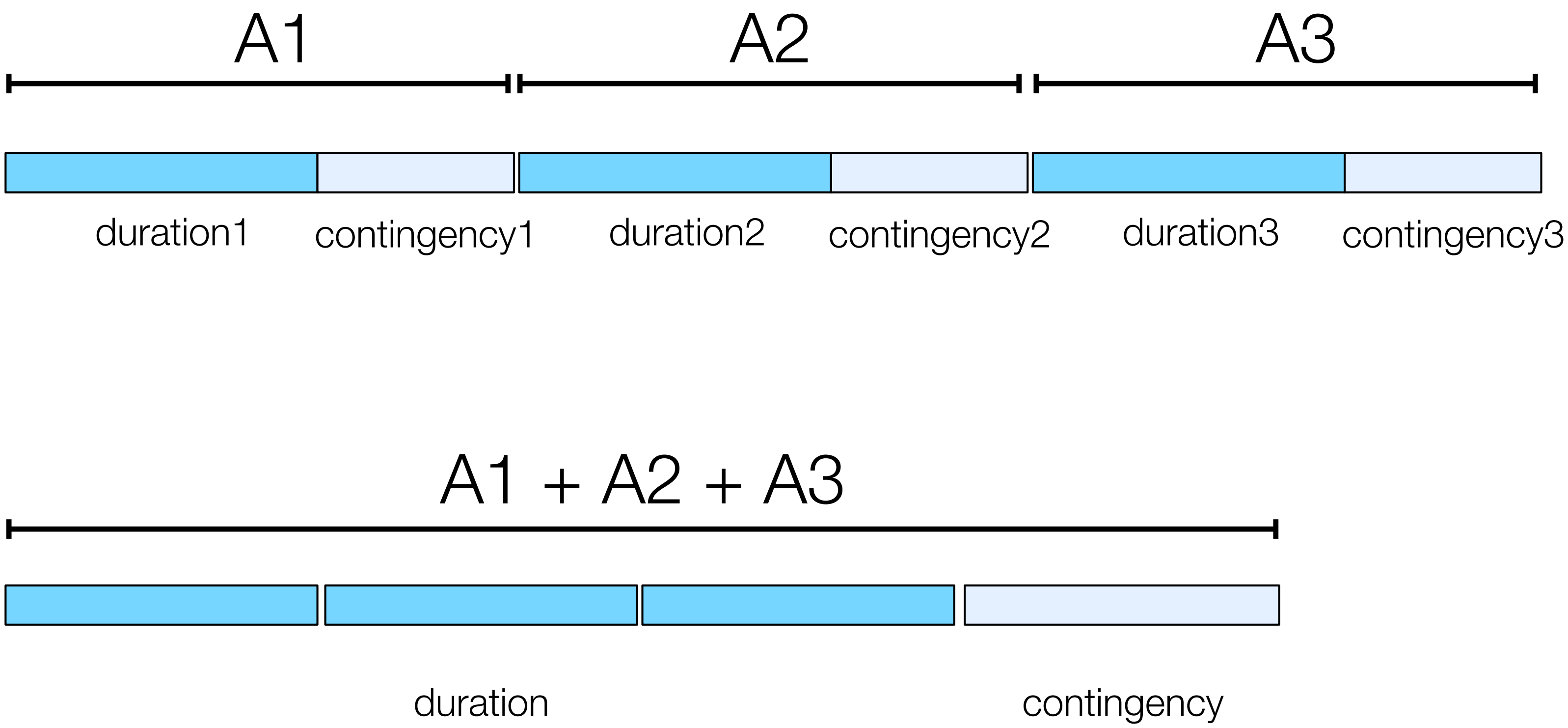
CCM: Item 1 (Estimations)



Notice that the contingency depends upon the standard deviation!



CCM: Item 2 (sum of variances)



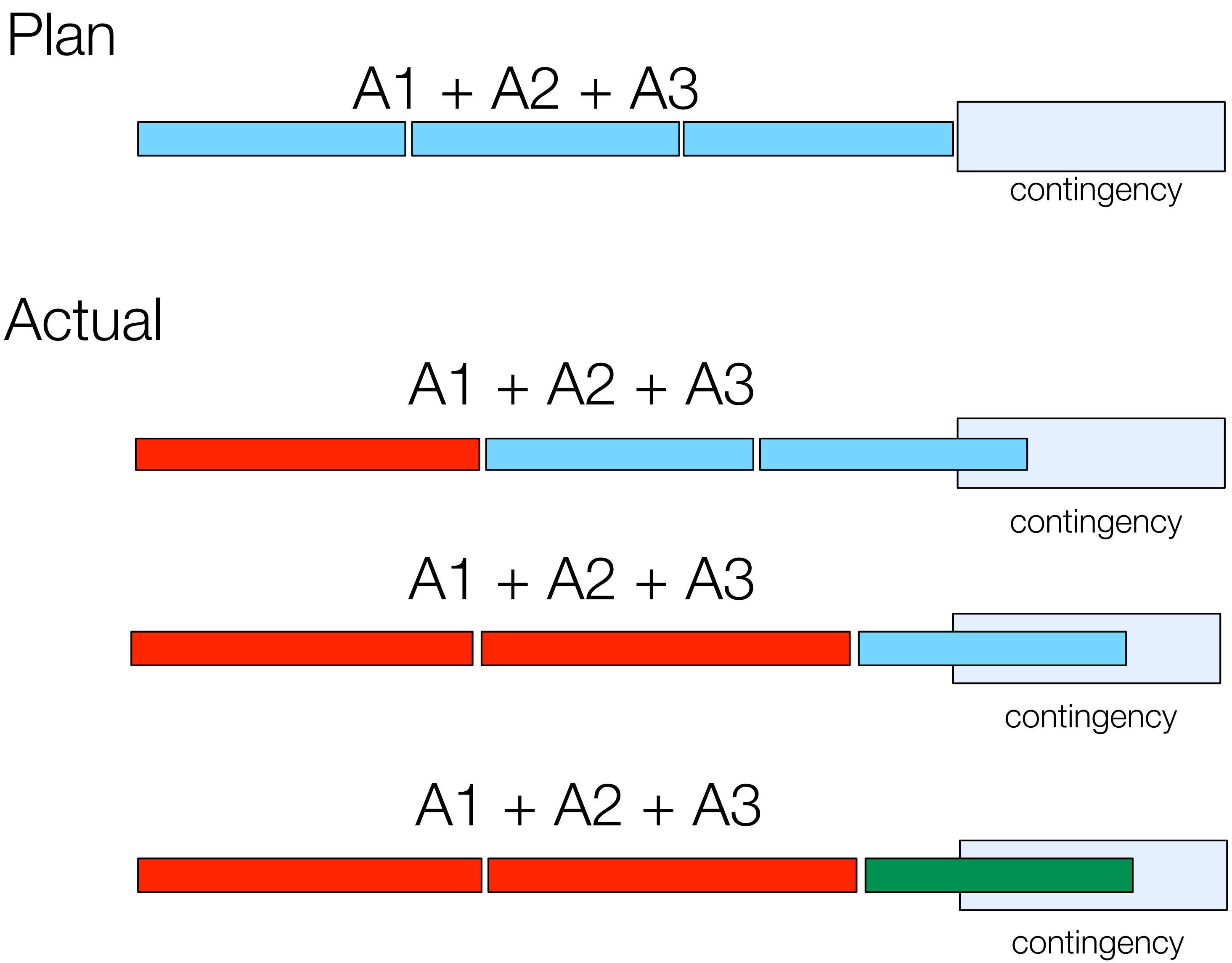
duration = duration1 + duration2 + duration3

contingency < contingency1 + contingency2 + contingency3

CCM Basic Idea

- When planning **we reason and monitor chains** and **we make the contingency buffer explicit**
- Item 1 (duration + contingency) + Item 2 (sum of variances) ensures that the plan is shorter than those obtained from the standard approach
- During plan execution the statistical variation will make some activities last longer than planned. These activities will make the chain overflow the contingency buffer, which is ok up to a point.
- **The manager manages the chains, not activities**

Example



A1 lasts longer than planned and the chain enters the contingency buffer

A2 lasts longer than planned and the chain continues entering the contingency buffer

However A3 lasts as planned and some buffer is spared.

CCM: Managing Chains

- When using CCM two important questions arise:
  - **Buffer definition:** What chains are best to consider or, put it another way, where we put the buffers
  - **Buffer management:** When do we need to start worrying about overflow in the contingency buffer

Buffer Management

Buffer Penetration

Task Execution Status	First Third	Second Third	Final Third
First Third	NO ACTION	Serious problem	**Very serious problem**
Second Third	NO ACTION	Could be a problem: identify and formulate solution	Serious problem
Final Third	**Very good! We are early**	NO ACTION	Monitor the situation