CAS CS 411
Lec 9

Project Management

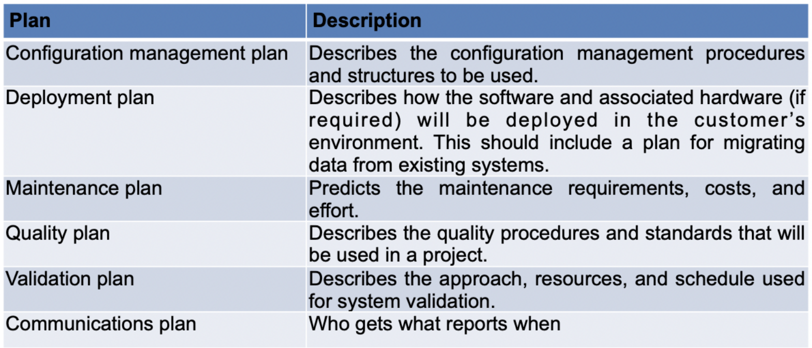
1. Project Planning
    a. Project planning involves breaking down the work into parts and assign these to project team members, anticipate problems that might arise and prepare tentative solutions to those problems.
    b. The project plan, which is created at the start of a project, is used to communicate how the work will be done to the project team and customers, and to help assess progress on the project.
    c. Software engineers get involved at the planning stage to help establish the time it will take to complete the project
2. Planning Stages
    a. At the proposal stage, when you are bidding for a contract to develop or provide a software system.
    b. During the project startup phase, when you have to plan who will work on the project, how the project will be broken down into increments, how resources will be allocated across your company, etc.
    c. Periodically throughout the project, when you modify your plan in the light of experience gained and information from monitoring the progress of the work.
3. Where do the number come from?
    a. It's fine to have a large ballpark number in your head, however contracts are not written using ballpark numbers
    b. We need to be able to accurately predict what the cost of a project will be, plus any contingencies
    c. For software projects we typically use hours as the unit of measure
    d. So, we come up with the total number of hours for the project and multiply by either a specific rate for each role (SE = $120, QA = $100 etc) or a blended rate
    e. This sounds easy, but it turns out to be astoundingly difficult!
4. Plan-driven development
    a. Plan-driven or plan-based development is an approach to software engineering where the development process is planned in detail.
        i. Plan-driven development is based on engineering project management techniques and is the 'traditional' way of managing large software development projects.
    b. A project plan is created that records the work to be done, who will do it, the development schedule and the work products.

    c. Managers use the plan to support project decision making and as a way of measuring progress.
    d. We say that we ‘plan the work, then work the plan’

5. Pros and Cons
    a. The arguments in favor of a plan-driven approach are that early planning allows organizational issues (availability of staff, other projects, etc.) to be closely taken into account, and that potential problems and dependencies are discovered before the project starts, rather than once the project is underway.
    b. The principal argument against plan-driven development is that many early decisions have to be revised because of changes to the environment in which the software is to be developed and used.

6. Project Plans
    a. In a plan-driven development project, a project plan sets out the resources available to the project, the work breakdown and a schedule for carrying out the work.
    b. Plan sections
        i. Introduction
        ii. Project organization
        iii. Risk analysis
        iv. Hardware and software resource requirements
        v. Work breakdown
        vi. Project schedule
        vii. Monitoring and reporting mechanisms

7. Project Plan Supplements
    a.

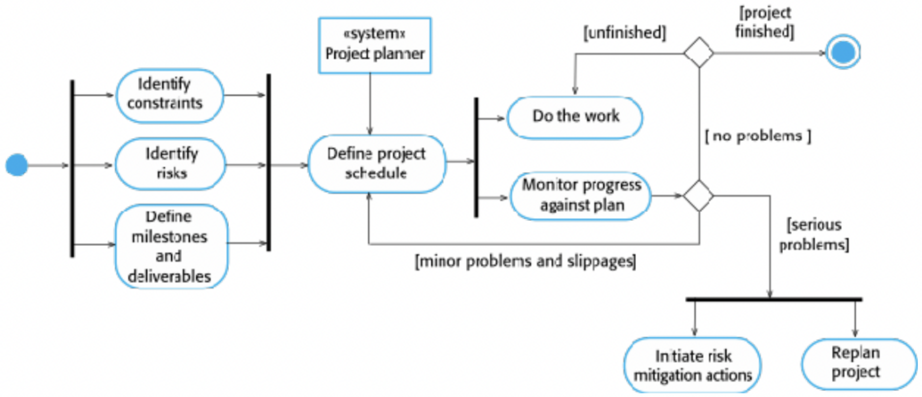


Plan	Description
Configuration management plan	Describes the configuration management procedures and structures to be used.
Deployment plan	Describes how the software and associated hardware (if required) will be deployed in the customer’s environment. This should include a plan for migrating data from existing systems.
Maintenance plan	Predicts the maintenance requirements, costs, and effort.
Quality plan	Describes the quality procedures and standards that will be used in a project.
Validation plan	Describes the approach, resources, and schedule used for system validation.
Communications plan	Who gets what reports when

8. The Planning Process
    a. Project planning is an iterative process that starts when you create an initial project plan during the project startup phase.
    b. Plan changes are inevitable.
        i. As more information about the system and the project team becomes available during the project, you should regularly revise the plan to reflect requirements, schedule and risk changes.

ii. Changing business goals also leads to changes in project plans. As business goals change, this could affect all projects, which may then have to be re-planned.

9. The Project Process

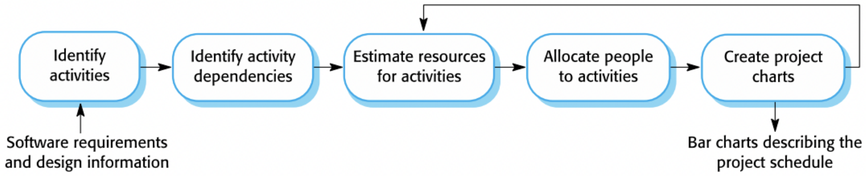


a.

10. Planning Assumptions
    a. You should make realistic rather than optimistic assumptions when you are defining a project plan.
    b. Problems of some description always arise during a project, and these lead to project delays.
    c. Your initial assumptions and scheduling should therefore take unexpected problems into account.
    d. You should include contingency in your plan so that if things go wrong, then your delivery schedule is not seriously disrupted.

11. Risk Mitigation
    a. If there are serious problems with the development work that are likely to lead to significant delays, you need to initiate risk mitigation actions to reduce the risks of project failure.
    b. In conjunction with these actions, you also have to re-plan the project.
    c. This may involve renegotiating the project constraints and deliverables with the customer. A new schedule of when work should be completed also has to be established and agreed with the customer

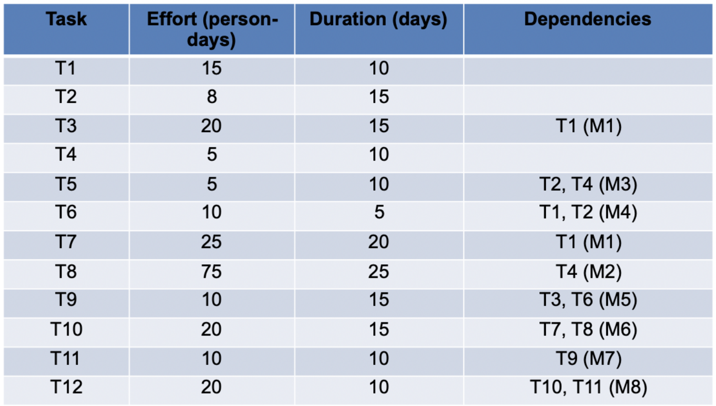
12. Project Scheduling
    a. Project scheduling is the process of deciding how the work in a project will be organized as separate tasks, and when and how these tasks will be executed.
    b. You estimate the calendar time, usually in hours, needed to complete each task, the effort required and who will work on the tasks that have been identified.
    c. You also have to estimate the resources needed to complete each task, such as the disk space required on a server, the time required on specialized hardware, such as a simulator, and what the travel budget will be.

13. Project Scheduling Activities
    a. We split project into tasks and estimate time and resources required to complete each task.
    b. Organize tasks concurrently to make optimal use of workforce.
    c. Minimize task dependencies to avoid delays caused by one task waiting for another to complete.
    d. Dependent on project managers intuition and experience.
14. The Project Scheduling Process
    a. 
Identify activities
Identify activity dependencies
Estimate resources for activities
Allocate people to activities
Create project charts
Software requirements and design information
Bar charts describing the project schedule
15. Scheduling Problems
    a. Estimating the difficulty of problems and hence the cost of developing a solution is hard.
    b. Productivity is not proportional to the number of people working on a task.
    c. Adding people to a late project makes it later because of communication overheads (Brooks' Law - see his The Mythical ManMonth)
    d. The unexpected always happens. Always allow contingency in planning.
16. Types of Budgeting
    a. Macro vs Micro
        i. Top-down
        ii. Bottom up
    b. Negotiated
17. Top-Down Budgeting
    a. Top managers estimate/decide on the overall budget for the project
    b. These trickle down through the organization where the estimates are broken down into greater detail at each lower level
    c. The process continues to the bottom level
    d. Numbers are usually based on prior experience, similar projects, other budget constraints
18. Advantages
    a. Overall project budgets can be set/controlled very accurately
        i. A few elements may have significant error
    b. Management has more control over budgets
    c. Small tasks need not be identified individually
19. Disadvantages
    a. More difficult to get buy in
    b. Leads to low level competition for larger shares of budget

20. Bottom-Up Budgeting
    a. Project is broken down into work packages
    b. Low level managers price out each work package
    c. Overhead and profits are added to develop the budget
    d. We get finer-grained estimates
    e. However, the final number tends to be much larger than the top-down estimate
21. Estimating Time / Budgets is Difficult!
    a. There may not be as much historical data or none at all
    b. Even with similar projects, there may be significant differences
    c. Many people have input to the budget
22. Parkinson's Law
    a. "Work expands so as to fill the time available for its completion"
    b. A corollary: If you wait until the last minute, the task only takes a minute to do
23. Student Syndrome
    a. Focus on task doesn't happen until the task is due for completion
    b. How's that homework coming along?
24. Hofstadter's Law
    a. "It always takes longer than you expect, even when you take into account Hofstadter's Law."
    b. From his classic book Escher, Goedel Bach; An Eternal Golden Braid
25. Planning Fallacy
    a. Proposed by Kahneman and Tversky
    b. People tend to underestimate the amount of time necessary to complete a task
    c. But not if the estimates are submitted anonymously
    d. What can you infer from this?
26. Project Activities
    a. Project activities (tasks) are the basic planning element. Each activity has:
        i. a duration in calendar hours, days, or months,
        ii. an effort estimate, which shows the number of person-days or person-months to complete the work,
        iii. a deadline by which the activity should be complete,
        iv. a defined end-point, which might be a document, the holding of a review meeting, the successful execution of all tests, etc.
27. Milestones and deliverables
    a. Milestones are points in the schedule against which you can assess progress, for example, the handover of the system for testing.
    b. Deliverables are work products that are delivered to the customer, e.g. a requirements document for the system.
    c. The reality is that we often start with the end date and work backwards

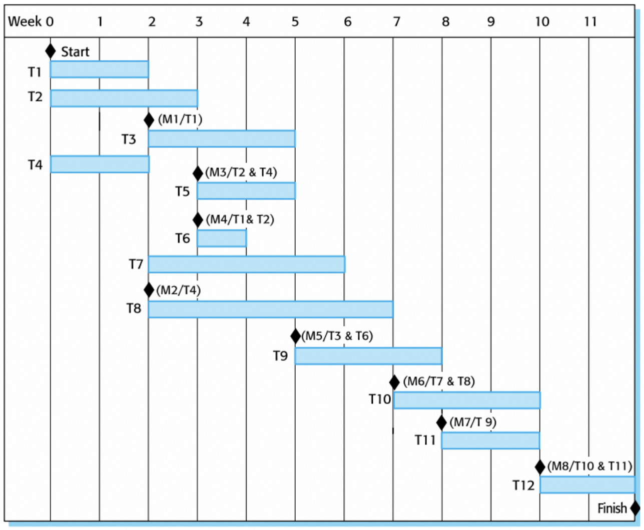
        i. Projects are often driven by marketing, and they have specific rollout dates to hit
        ii. This becomes a challenge to come up with the resources to complete the project in a reasonable budget

28. How Many Hours Does a Contributor Work?
    a. For planning purposes we assume that an individual contributor will work 2000 hour per year
    b. Everyone does this, however it is wildly optimistic
        i. There are roughly 2100 available hours, so 2000 hours assumes around 95% efficiency
        ii. Reality is that most ICs are around 80-85% efficient (they'll take days off, be sick, slack off at work, and so on)
        iii. 2000 however is a nice round number
    c. My approach is to use the 2000 number when estimating budget, but use 1800 (85% efficiency) for schedule
29. Tasks, Durations, and Dependencies
    a.

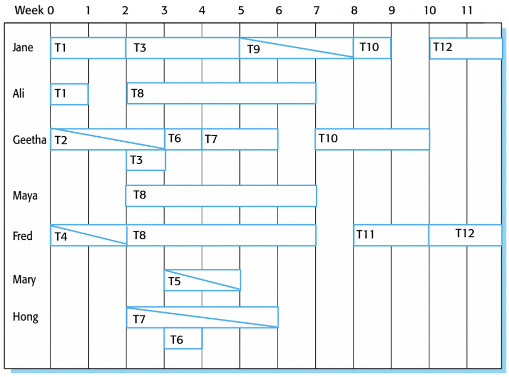


Task	Effort (person-days)	Duration (days)	Dependencies
T1	15	10	
T2	8	15	
T3	20	15	T1 (M1)
T4	5	10	
T5	5	10	T2, T4 (M3)
T6	10	5	T1, T2 (M4)
T7	25	20	T1 (M1)
T8	75	25	T4 (M2)
T9	10	15	T3, T6 (M5)
T10	20	15	T7, T8 (M6)
T11	10	10	T9 (M7)
T12	20	10	T10, T11 (M8)

30. Activity Bar Chart
    a.



31. Staff Allocation Chart

a. 

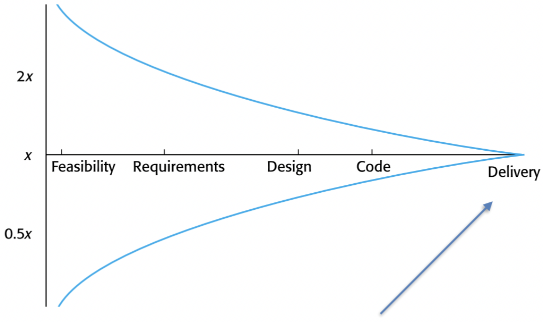
32. Estimation Techniques

a. Organizations need to make software effort and cost estimates. There are two types of technique that can be used to do this:

    i. Experience-based techniques The estimate of future effort requirements is based on the manager's experience of past projects and the application domain. Essentially, the manager makes an informed judgment of what the effort requirements are likely to be.

    ii. Algorithmic cost modeling In this approach, a formulaic approach is used to compute the project effort based on estimates of product attributes, such as size, and process characteristics, such as experience of staff involved.

33. Estimate Uncertainty

a. 

This should not be a huge surprise...

34. Experience-based Approaches

a. Experience-based techniques rely on judgments based on experience of past projects and the effort expended in these projects on software development activities.

b. Typically, you identify the deliverables to be produced in a project and the different software components or systems that are to be developed.

c. You document these in a spreadsheet, estimate them individually and compute the total effort required.

d. It usually helps to get a group of people involved in the effort estimation and to ask each member of the group to explain their estimate.

35. Problem with Experience-based Approaches
    a. The difficulty with experience-based techniques is that a new software project may not have much in common with previous projects.
    b. Software development changes very quickly and a project will often use unfamiliar techniques such as web services, application system configuration or HTML5.
    c. If you have not worked with these techniques, your previous experience may not help you to estimate the effort required, making it more difficult to produce accurate costs and schedule estimates.
36. Algorithmic Cost Modeling
    a. Some organizations try to use algorithms to come up with cost estimates
    b. The algorithms are tuned over time and reflect historical project information
    c. Many of these use touch-point or level-of-complexity counts
        i. i.e. The app has two pages of medium complexity, one high, eight SQL views of low complexity, etc
        ii. Each metric has a cost associated, and we just add them up
    d. Note that we're almost always using 'cost' as a synonym for 'time' here…projects are discussed interns of the number of hours
37. Estimation Accuracy
    a. The size of a software system can only be known accurately when it is finished*
    b. Several factors influence the final size
        i. Use of reused systems and components;
        ii. Programming language;
        iii. Distribution of system.
    c. As the development process progresses then the size estimate becomes more accurate.
    d. The estimates of the factors contributing to B and M are subjective and vary according to the judgment of the estimator.
38. Scheduling
    a. No matter how you come up with the numbers, we typically use three for each task:
        i. 90% confidence: Absolutely sure you can do it in this amount of time (called the Optimistic estimate)
        ii. 50% confidence: If everything goes perfectly, you can pull it off in this amount of time (called the Pessimistic estimate)
        iii. Most likely time: Somewhere in between the other two… sometimes it is the same as the 90% value
    b. Which will be the larger number?
    c. And which do we use?

39. PERT

a. The Program Evaluation Review Technique (PERT) was developed for large-scale defense contract budgeting

b. It uses a simple formula to calculate an estimated time to complete a task

T = (Optimistic + 4*Likely + Pessimistic) / 6

c. For example, if O = 24, P = 40, L = 30:

(24 + 4*30 + 40) / 6

184 / 6

31

d. Once we have these numbers for each task, we lay the tasks out in a way that shows their dependencies

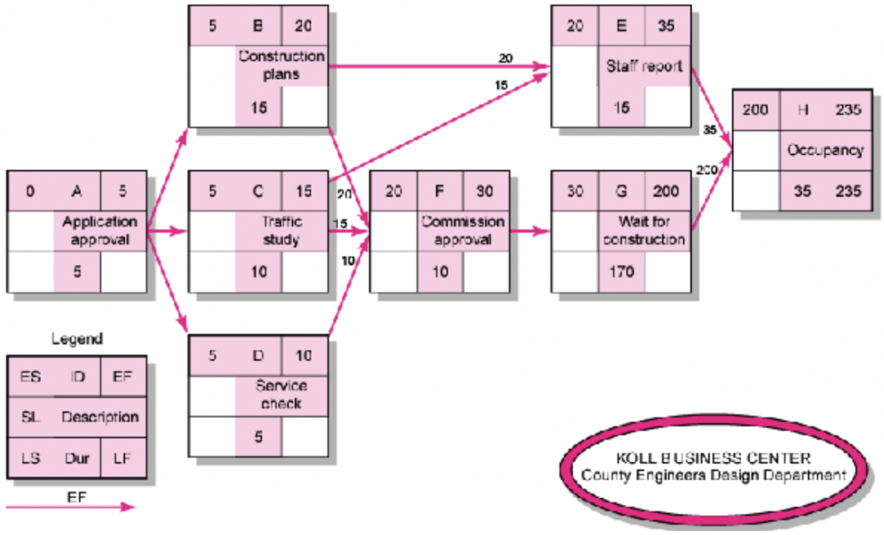
e. For example, Task B can’t start until Task A finishes

i. This might be due to resource constraints, or maybe A produces something that B needs

f. Here’s a diagram with eight tasks that uses the calculated times to find the earliest end date of the each task (and thus the entire project)

g. Calculation is done left to right (forward)

40. Activity Network: Early Finish

a. 

b. Once we have that value in hand, we want to see if any tasks could be delayed without affecting the end date

c. These are the late start / late finish numbers

d. Calculation is done right to left (backward)

41. Slack

a. Finally, with the early and late start/finish numbers for all of the tasks, we can figure out how late a particular task can start without affecting those that follow

b. This is called slack and is used by PMs to provide some flexibility in scheduling resources

42. Critical Path
    a. The path through the project that has the least (typically 0) slack is called the critical path
    b. Any task on the critical path that starts late will delay the entire project
    c. If you are working on a task that is on the CP, expect your PM to drop by frequently to see how you are doing!
43. Managing the Numbers
    a. PMs do many things, but one task is to compare the actual hours worked to the budgeted hours and schedule to monitor for variations in either
    b. There are tools to do this, and a good PM will not focus only on the numbers…they are also a people manager and the project's success or failure rests with the PM
    c. As a software engineer your contributions will usually be
        i. Estimates at the start of the project
        ii. Periodic (weekly, etc) project meetings and status reports
        iii. Reporting hours
        iv. Raising concerns / red flags as early as possible
        v. Strategizing with the team to move the project forward on budget and schedule
44. Btoom Line
    a. Plan-driven development is organized around a complete project plan that defines the project activities, the planned effort, the activity schedule and who is responsible for each activity.
    b. Project scheduling involves the creation of various graphical representations of part of the project plan. Bar charts, which show the activity duration and staffing timelines, are the most commonly used schedule representations.
    c. A project milestone is a predictable outcome of an activity or set of activities. At each milestone, a formal report of progress should be presented to management. A deliverable is a work product that is delivered to the project customer.
    d. Estimation techniques for software may be experience-based, where managers judge the effort required, or algorithmic, where the effort required is computed from other estimated project parameters.