Official project planning

- Introduction
- Project Organisations
- Risk analysis
- Hardware & Software
- Work breakdown
- Project schedule
- Monitoring and revision plans

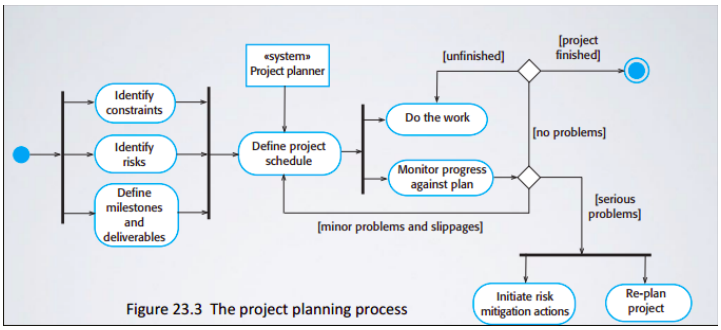


Figure 23.3 The project planning process

Project scheduling

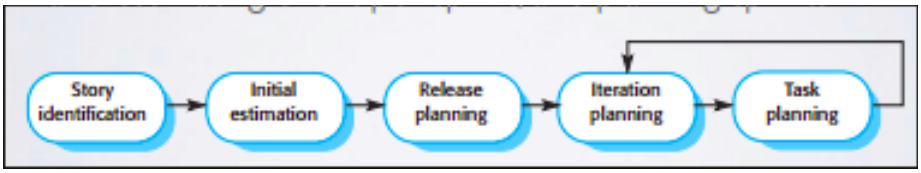
- Activities should produce some **measurable outcome**, so the progress can be assessed.
- A milestone is the end point of an activity
- Deliverables are results delivered to customers

Milestones and outputs

- Every activity/task should produce a tangible output
  - A requirement/specification document, design, code frame, database, etc.
- Tasks should be a few weeks in size
  - Should last any where from 1 to 10 weeks
  - If longer should be broken down into smaller tasks
  - Weeks are **different** to person-days of effort.
- Projects needs certain milestones
  - Which produce project deliverables
  - Might be times when a few activities/tasks all finish

Agile planning

- CPM, PERT, GRANTT charts are all very good for plan-based SE approaches
- As we learnt about Agile, XP , planning is based on % of user stories
- Agile planning involves estimating the effort for each user story, then estimating overall work and estimating effort per sprint, and planning sprints



- You can still plan in Agile
  - It’s a mistake to think that ‘agile’ means : we did what ever we wanted as the process went
- You can do an initial allocation of stories to e.g. Scrum sprints
  - Use that to estimate number of people needed
  - Allocate stories to people
  - Produce milestone versions (expected per each sprint)

Reviewing project progress

- Milestones are times to review progress and the plan
- A plan was an estimate, and so you need to check for slippage
  - A conservative plan includes time for slippage
  - If slippage is small, it might be reduced later in the project
  - If serious, set into actions your risk strategies and re-plan the project
- If the new plan is more expensive
  - Need to change the tasks, or negotiate for more money
  - Decide if the project should continue or be cancelled.

Agile project reviewing

- A project review is built right into e.g. Scrum
- The aim of the start of every sprint: plan ahead
  - Look at the overall backlog
  - Set targets for a sprint
  - Reflect also on overall progress
- Burdown charts help with this

Costing a project

Two ‘overall’ approaches

- By work that needs doing (then estimating developer time)
- By developer time (and defining how much work to do)
  - One reason why scrum is popular
    - You agree a number of sprints for a team
    - Cost revolves around this
    - Expanding a project is ‘buying more sprints’

Estimating the effort needed

- You’ve analysed the main risks and activities
- First you estimate the min/max time the project will take
- One option add a contingency estimate (30-50% extra)
  - To cover what you didn’t include in the plan and isks that do happen
- Or use t = (min + (4xLikelyPath) + max)/6

Costing a project is a complex issue

- Experience-based techniques
  - Experience of prior similar projects
  - Managers familiarity with work involved
- Algorithmic Cost modelling (based on)
  - Estimated amount of effort
  - Experience based constants
  - Still can be as much as 25-400% wrong

Algorithmic cost modelling

Effort = A x $Size^b$ x M

- A – constant base on organisation overheads
- Size – A size estimate on code or amount of functionality
- B – size of the project, likelihood of overrun
- M – a multiplier for product/process/people attributes (project needing more rigorous process for safety)
- Are often complex
- Are typically inaccurate
- Recommend that you produce
  - Optimal cost model
  - Worse case cost model
  - Likely cost model