

- ... project status reports
- ... grades due
- ... swiki submissions (add team name & members to group list, then create and link to your team's page)

Project Planning

- team & project
- project plan
- scheduling, cost estimates, and risks

Team

- common goal/shared vision
 - project topic, and project success
- diverse backgrounds
 - exploit strengths, domain knowledge, technical skills...
- communicate
 - good & bad news; arrange meeting times
- work together
 - to resolve problems/bugs, issues, revise plan...

Member Roles

- Project Manager
- Principal Engineer
- Programmer
- Technical Lead
- Designer
- Marketing
- Analyst
- Documentation Coordinator/Technical Writer
- Tester
- Customer Support
 - installation, training, maintenance,

(small teams – not always separated roles)

Role distribution

- Project Manager
 - motivate;
 - run & organize meetings;
 - status reports;
 - produce project plan & updates to plan;
- Development Manager
 - development strategy;
 - time/size estimates of product;
 - lead designer;
 - lead in preparing SRS (interaction with customer);
 - SDS

- Quality Manager
 - producing and tracking the quality plan; alert to quality problems;
 - lead in developing test plan for development, integration and system testing; test materials and running tests
 - review proposed changes to baseline plan
 - moderator for reviews / recorder at team meetings
- Support Manager
 - support needs – resources, facilities, etc.
 - prepare user documentation
 - monitor risk issues
 - manage source control system
 - maintain project files
- Everyone also acts as Development Engineer

Project Planning

- goals:
 - on-time delivery – within schedule
 - within budget (using available resources)
 - product satisfies requirements
- main asset – manager's experience
 - but, product is intangible, no standard SE processes, changes in requirements, technologies, tools, personnel capabilities...

Management activities

- Proposal writing.
- Project planning and scheduling.
- Project costing.
- Project monitoring and reviews.
- Personnel selection and evaluation.
- Report writing and presentations.

Project planning process

```
Establish the project constraints
Make initial assessments of the project parameters
Define project milestones and deliverables
while project has not been completed or cancelled loop
  Draw up project schedule
  Initiate activities according to schedule
  Wait ( for a while )
  Review project progress
  Revise estimates of project parameters
  Update the project schedule
  Re-negotiate project constraints and deliverables
  if ( problems arise ) then
    Initiate technical review and possible revision
  end if
end loop
```

Project plan structure

- Use online templates.
- Plan components include:
 - Introduction.
 - Project organisation.
 - Risk analysis.
 - Hardware and software resource requirements.
 - Work breakdown.
 - Project schedule.
 - Monitoring and reporting mechanisms.

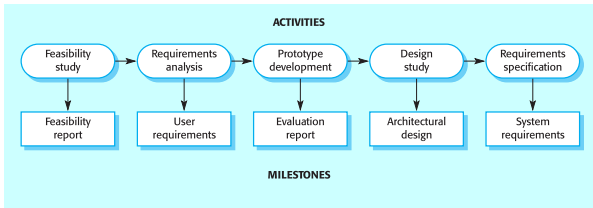
So How Do we Do this?

- Spend time understanding the problem
- Estimate amount of effort required
 - Number of major functions
 - Difficulty of each function
- Develop schedule with built in safety nets
 - Increase estimates by some factor
 - Have a backup plan for worst case
 - Make sure schedule is realistic
- Revise schedule as project understanding increases

Estimation

- hard to estimate well during the first iteration
 - need constant improvements
- estimate difficulty of each task
 - size of problem (LOC) and
 - resource requirements -> person-hours, person-months
 - you have 50-60 hours per credit per person
 - provision for error
- understand tasks dependencies and ability for concurrent progress
 - objective: minimize dependencies, optimize concurrency for workforce
- identify milestones
 - 1 per week, milestone != deliverable
- calendar estimate

Milestones in the RE process



deliverables – user requirements and requirements specification

Beware

- “Fuzzy Front End” – minimize time here...
- things may go wrong – have contingency plans
 - have to do risk analysis and monitoring
- adding people to a late project may cost you more – overhead of extra communication
- productivity not always scalable in terms of number of people
- you and team members have other classes/jobs/lives...

Sample averages for small project

- Architecture/Design 10%
- Detailed Design 20%
- Code/Debug 25%
- Unit Testing 20%
- Integration 15%
- System Test 10%

Scheduling

- understand dependencies
 - another activity cannot proceed
- build dependency graphs, charts...
- build in safety nets and backup plans
- identify critical path

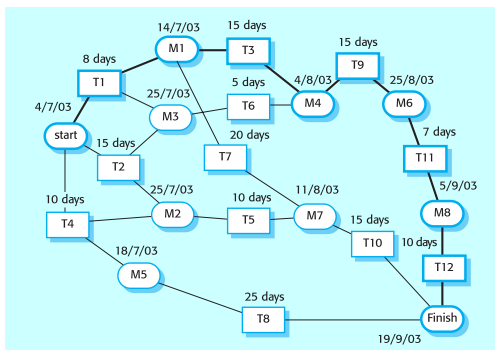
Scheduling mechanisms

- Process model
- Work breakdowns
 - Tasks and Milestones
- PERT (Program Evaluation and Review Technique) charts
- Gantt charts

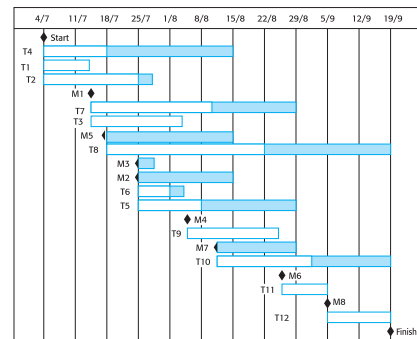
Task durations and dependencies

Activity	Duration (days)	Dependencies
T1	8	
T2	15	
T3	15	T1 (M1)
T4	10	
T5	10	T2, T4 (M2)
T6	5	T1, T2 (M3)
T7	20	T1 (M1)
T8	25	T4 (M5)
T9	15	T3, T6 (M4)
T10	15	T5, T7 (M7)
T11	7	T9 (M6)
T12	10	T11 (M8)

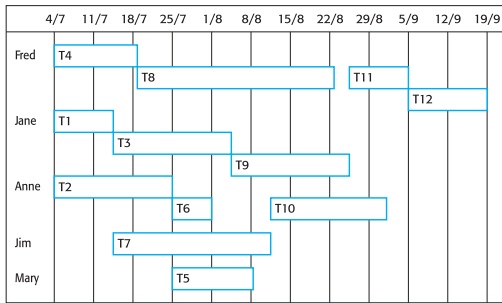
Activity network – PERT graph



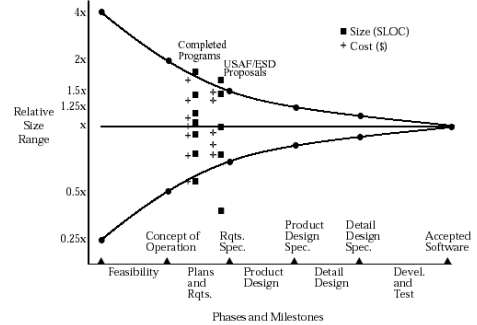
Activity timeline – Gantt chart



Staff allocation



Cost Estimation – improves with time...



Metrics

- In estimation, planning, monitoring...
 - cost (dollars)
 - duration (months)
 - effort (person-months)
 - size (LOC...)
 - quality (# of faults)

Size of Product

- Lines of Code (LOC) or K delivered Source Instructions (KDSI)
 - different languages; while lines?; hard to estimate size of code
 - source code only portion of effort
 - weighted av. optimistic, pessimistic and most likely case
- Function Points
 - functionality delivered by software
 - components – input, output, user inquiries, #files, interfaces
 - weight for simple, average, complex
 - technical complexity – performance, transaction rates, distributed, communication...
 - more in advanced SE classes

- Files, Flows, Processes
- 3D Function Points
 - for real time s/w, considers #state transitions in control process
- Feature Points

COCOMO

- Constructive Cost Model
 - originally Barry Boehm '81
 - revised in 90's COCOMO II
 - for OO, component-based, tool-based, PC, real-time turnaround software development, COTS, instructions/lines-of-code vs. function points vs. objects, ...
 - objective -> estimate costs within 20%, 70% of the time
- Database of product histories, calibrated annually
 - calibration -> adjustment of weights to product components, circumstances, language level, tools, development stage...
- Original COCOMO
 - Simple, Intermediate and Detailed Forms
 - Organic, Semidetached and Embedded

- Input
 - estimate of project size
 - type of development
- Output
 - effort and phase distribution

Inputs

Development	
Delivered Source Instructions (thousands) (KDSI)	5
Development Mode	Organic <input checked="" type="checkbox"/>
Average Cost Rate (\$/PM)	10000
Maintenance	
KDSI added (annual)	0
KDSI modified (annual)	0
Average Cost Rate (\$/PM)	10000

(<http://www1.jsc.nasa.gov/bu2/COCOMO.html>)

Results

Effort	13	person-months (PM)
Schedule	7	months
Development Cost	130000	
Productivity	385	instructions per person-month
Average Staffing	1.9	full-time-equivalent software personnel
Annual Maintenance Effort	0	person-months
Annual Maintenance Cost	0	

Phase Distribution

	Effort (PM)	Schedule (mo.)	Staff (avg.)	Cost
Plans and requirements *	0.8	0.8	1	8000
Product Design	2.1	1.3	1.6	21000
Programming	8.5	4.1	2.1	85000
Detailed Design	3.3			33000
Code and unit test	5.2			52000
Integration and test	2.5	1.5	1.7	25000

Activity Distribution (Staff) by Phase

Activity	Phase				
	Plans and Requirements	Product Design	Programming	Integration and Test	Maintenan
Requirements Analysis	0.5	0.2	0.1	0.1	0
Product Design	0.2	0.6	0.2	0.1	0
Programming	0	0.2	1.2	0.6	0
Test Planning	0	0.1	0.1	0	0
Verification and Validation	0.1	0.1	0.1	0.6	0
Project Office	0.2	0.2	0.1	0.1	0
CM/QA	0	0	0.1	0.1	0
Manuals	0.1	0.1	0.1	0.1	0
TOTAL	1.1	1.500000000	2.000000000	1.700000000	0

Basic COCOMO

- SM = staff month = 152 hours = $2.4 * KDSI^{1.05}$
 - KDSI = 1000 delivered lines of source code
- TDEV = development time in calendar months = $2.5 * SM^{0.38}$

Risk Management

- Risk Something that can go wrong
 - Often a result of inadequate information
- Assessment
 - Identify, Analyze, Prioritize
- Control
 - Planning, Resolution, Monitoring

Risk identification

- Technology risks.
- People risks.
- Organisational risks.
- Requirements risks.
- Estimation risks.

Risk analysis (i)

Risk	Probability	Effects
Organisational financial problems force reductions in the project budget.	Low	Catastrophic
It is impossible to recruit staff with the skills required for the project.	High	Catastrophic
Key staff are ill at critical times in the project.	Moderate	Serious
Software components that should be reused contain defects which limit their functionality.	Moderate	Serious
Changes to requirements that require major design rework are proposed.	Moderate	Serious
The organisation is restructured so that different management are responsible for the project.	High	Serious

Risk analysis (ii)

Risk	Probability	Effects
The database used in the system cannot process as many transactions per second as expected.	Moderate	Serious
The time required to develop the software is underestimated.	High	Serious
CASE tools cannot be integrated.	High	Tolerable
Customers fail to understand the impact of requirements changes.	Moderate	Tolerable
Required training for staff is not available.	Moderate	Tolerable
The rate of defect repair is underestimated.	Moderate	Tolerable
The size of the software is underestimated.	High	Tolerable
The code generated by CASE tools is inefficient.	Moderate	Insignificant

Risk Planning

- Consider each risk and develop a strategy to manage that risk.
- Avoidance strategies
 - The probability that the risk will arise is reduced;
- Minimisation strategies
 - The impact of the risk on the project or product will be reduced;
- Contingency plans
 - If the risk arises, contingency plans are plans to deal with that risk;

Risk management strategies (i)

Risk	Strategy
Organisational financial problems	Prepare a briefing document for senior management showing how the project is making a very important contribution to the goals of the business.
Recruitment problems	Alert customer of potential difficulties and the possibility of delays, investigate buying-in components.
Staff illness	Reorganise team so that there is more overlap of work and people therefore understand each other's jobs.
Defective components	Replace potentially defective components with bought-in components of known reliability.

Risk management strategies (ii)

Risk	Strategy
Requirements changes	Derive traceability information to assess requirements change impact, maximise information hiding in the design.
Organisational restructuring	Prepare a briefing document for senior management showing how the project is making a very important contribution to the goals of the business.
Database performance	Investigate the possibility of buying a higher-performance database.
Underestimated development time	Investigate buying in components, investigate use of a program generator

Risk monitoring

- Assess each identified risks regularly to decide whether or not it is becoming less or more probable.
- Also assess whether the effects of the risk have changed.
- Each key risk should be discussed at management progress meetings.

Risk indicators

Risk type	Potential indicators
Technology	Late delivery of hardware or support software, many reported technology problems
People	Poor staff morale, poor relationships amongst team member, job availability
Organisational	Organisational gossip, lack of action by senior management
Tools	Reluctance by team members to use tools, complaints about CASE tools, demands for higher-powered workstations
Requirements	Many requirements change requests, customer complaints
Estimation	Failure to meet agreed schedule, failure to clear reported defects