Planning a Software Project

Agenda

- Background
- Process planning
- Effort estimation
- Schedule and resource estimation
- Quality Planning
- Risk management
- Configuration Management
- Project monitoring plans

Software Project

- Goal: Build a software system to meet commitments on cost, schedule, quality
- Worldwide - many projects fail
  - one-third are runaways with cost or schedule overrun of more than 125%

Project Failures

- Major reasons for project runaways
  - unclear objectives
  - bad planning
  - no project management methodology
  - new technology
  - insufficient staff
- All of these relate to project management
- Effective project management is key to successfully executing a project

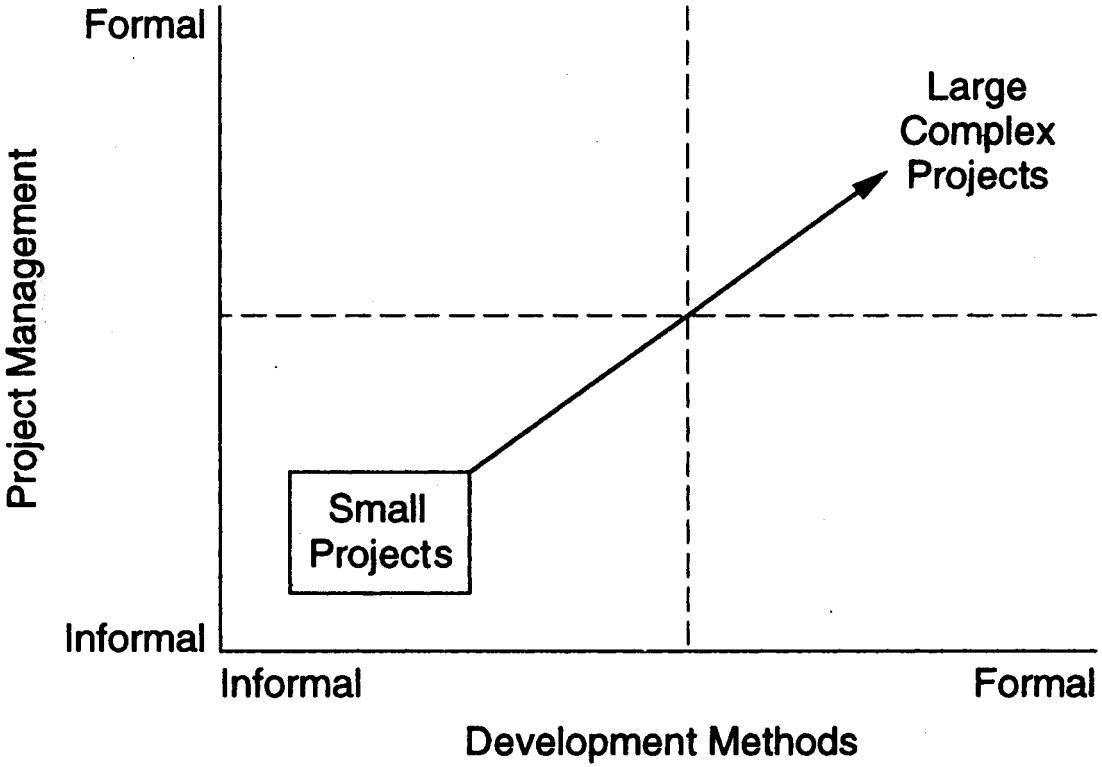
Why improve PM?

- Better predictability leading to commitments that can be met
- Lower cost through reduced rework, better resource mgmt, better planning,..
- Improved quality through proper quality planning and control
- Better control through change control, CM, monitoring etc.

Why improve PM ....

- Better visibility into project health and state leading to timely intervention
- Better handling of risks reducing the chances of failure
- All this leads to higher customer satisfaction
- And self and organization improvement

Two Dimensions in project execution



Process-based Project Execution

- Small project both engg and PM can be done informally
- Large projects require formality
- Formality: well defined processes used for each task; measurements used to control
- Here we focus on processes for PM only

The Project Mgmt Process

- Has three phases - planning, monitoring and control, and closure
- Planning is done before the main engineering LC and closure after the LC
- Monitoring phase is in parallel with LC

Project Planning

- Basic objective: To create a plan to meet the commitments of the project, I.e. create a path that, if followed, will lead to a successful project
- Planning involves defining the LC process to be followed, estimates, detailed schedule, plan for quality, etc.
- Main output - a project management plan and the project schedule

Key Planning Tasks

- Define suitable processes for executing the project
- Estimate effort
- Define project milestones and create a schedule
- Define quality objectives and a quality plan
- Identify risks and make plans to mitigate them
- Define measurement plan, project-tracking procedures, training plan, team organization, etc.

Process Planning

- Plan how the project will be executed, I.e. the process to be followed
- Process will decide the tasks, their ordering, milestones
- Hence process planning is an important project planning task
- Should plan for LC and PM processes as well as supporting processes

Life Cycle Process

- Various LC models - waterfall, iterative, prototyping; diff models suit different projects
- During planning can select the model that is best for the project
- This gives the overall process which has to be fine-tuned to suit the project needs
- Usually done by process tailoring - changing the process to suit the project
- Tailoring finally results in adding, deleting, modifying some process steps

Effort Estimation

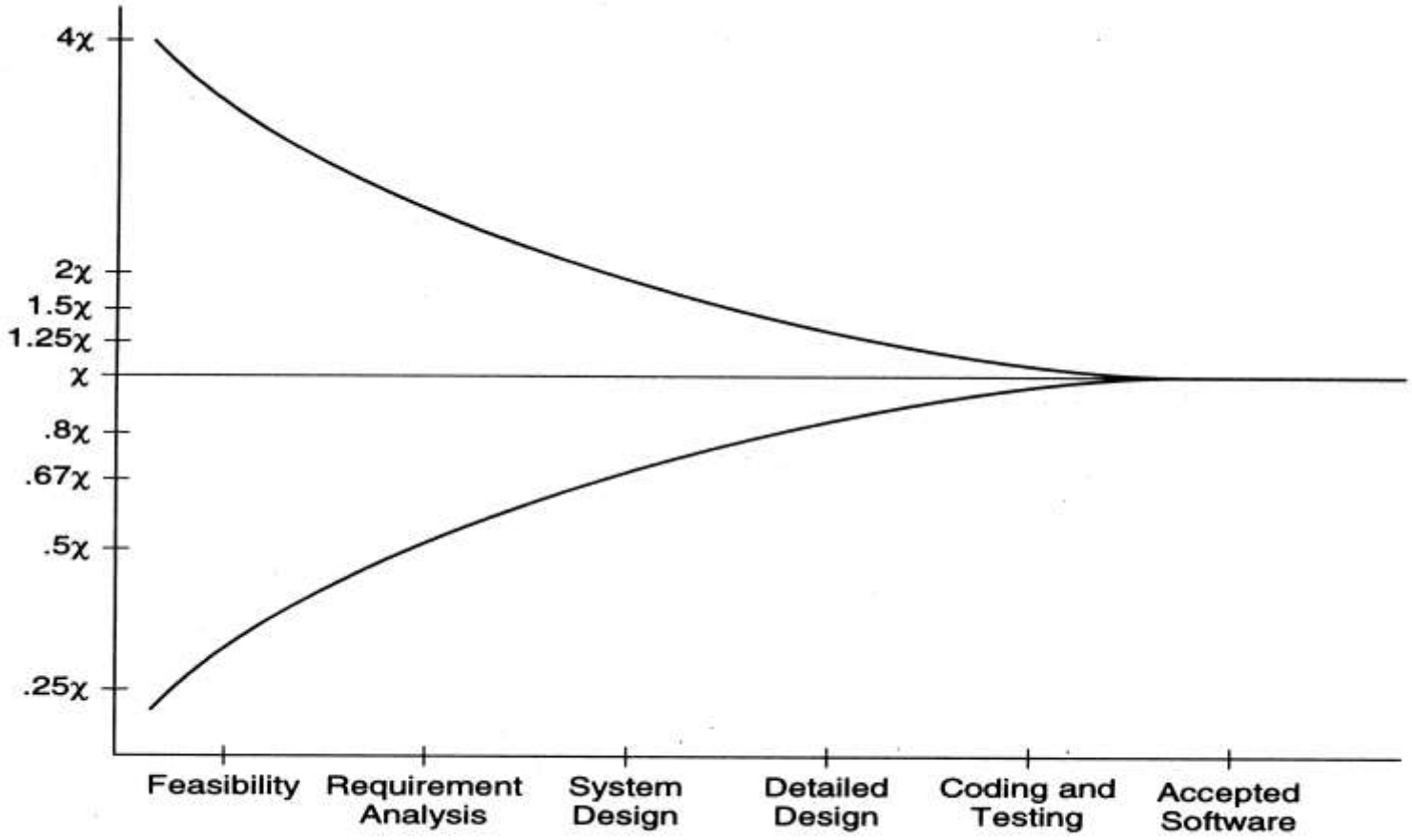
Effort Estimation

- For a project total cost and duration has to be committed in start
- Requires effort estimation, often in terms of person-months
- Effort estimate is key to planning - schedule, cost, resources depend on it
- Many problems in project execution stem from improper estimation

Estimation..

- No easy way, no silver bullet
- Estimation accuracy can improve with more information about the project
- Early estimates are more likely to be inaccurate than later
  - More uncertainties in the start
  - With more info, estimation becomes easier

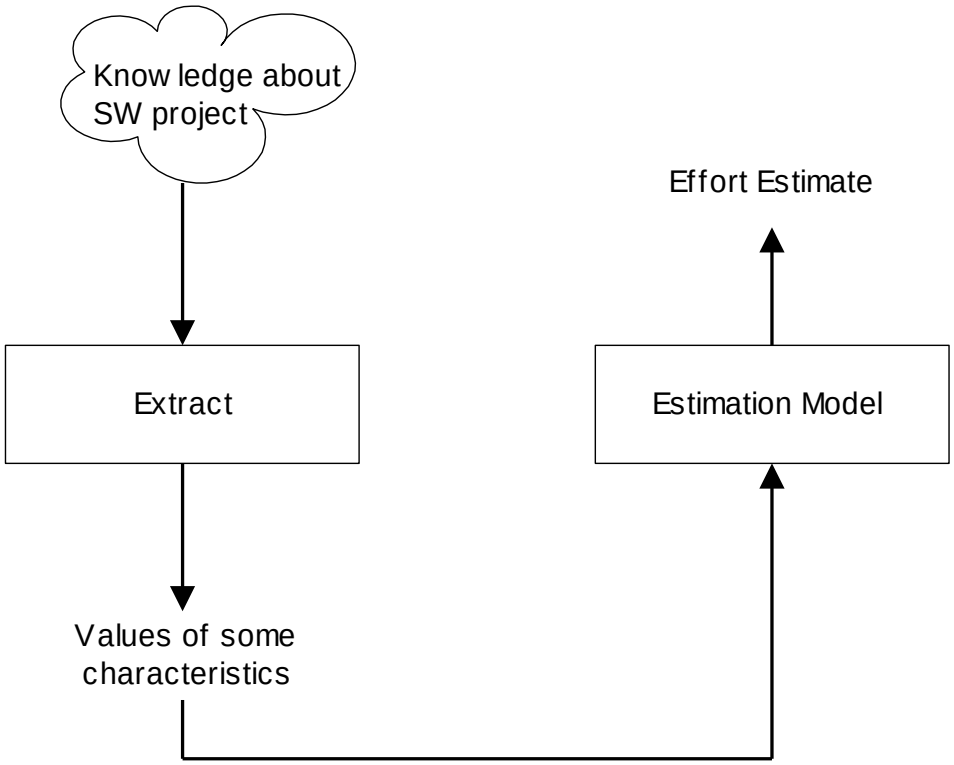
Estimation accuracy



Effort Estimation Models..

- A model tries to determine the effort estimate from some parameter values
- A model also requires input about the project, and cannot work in vacuum
- So to apply a model, we should be able to extract properties about the system
- Two types of models - top-down and bottom-up

Effort Estimation Models



Top-down Estimation

- First determines the total effort, then effort for components
- Usually works with overall size
- One method is to see estimate as a function of effort; the common function used is

$$\text{Effort} = a * \text{size}^{b}$$

- E is in person-months, size in KLOC
- Constants a and b determined through regression analysis of past project data

Top down estimation

- Can also estimate from size and productivity
  - Get the estimate of the total size of the software
  - Estimate project productivity using past data and project characteristics
  - Obtain the overall effort estimate from productivity and size estimates
- Effort distribution data from similar project are used to estimate effort for different phases

Bottom-up Estimation

- Effort for components and phases first estimated, then the total
- Can use activity based costing - all activities enumerated and then each activity estimated separately
- Can group activities into classes - their effort estimate from past data

An Estimation Procedure

- Identify programs in the system and classify them as simple, medium, or complex (S/M/C)
- Define the average coding effort for S/M/C
- Get the total coding effort.
- Use the effort distribution in similar projects to estimate effort for other tasks and total
- Refine the estimates based on project specific factors

COCOMO Model for Estimation

- Is a top-down approach
- Uses size, but adjusts using some factors
- Basic procedure
  - Obtain initial estimate using size
  - Determine a set of 15 multiplying factors from different project attributes
  - Adjust the effort estimate by scaling it with the final multiplying factor

COCOMO..

- Initial estimate: a * size ^{b} ; some standard values for a, b given for diff project types
- There are 15 cost driver attributes line reliability, complexity, application experience, capability, ...
- Each factor is rated, and for the rating a multiplication factor is given
- Final effort adjustment factor is the product of the factors for all 15 attributes

COCOMO – Some cost drivers

Cost Driver	Very low	Low	Nominal	High	Very High
Required reliability	.75	.88	1.0	1.15	1.4
Database size		.94	1.0	1.08	1.16
Product complexity	.7	.85	1.0	1.15	1.3
Execution time constraint			1.0	1.11	1.3
Memory constraint			1.0	1.06	1.21
Analyst capability	1.46	1.19	1.0	.86	.71
Application experience	1.29	1.13	1.0	.91	.82
Programmer capability	1.42	1.17	1.0	.86	.70
Use of software tools	1.24	1.10	1.0	.91	.83
Development schedule	1.23	1.08	1.0	1.04	1.1

COCOMO – effort distribution

- Effort distribution among different phases is given as a percent of effort
- Eg. For medium size product it is
  - Product design – 16%
  - Detailed design – 24%
  - Coding and UT – 38%
  - Integration and test – 22%

Scheduling and Staffing

Project Schedule

- A project Schedule is at two levels - overall schedule and detailed schedule
- Overall schedule comprises of major milestones and final date
- Detailed schedule is the assignment of lowest level tasks to resources

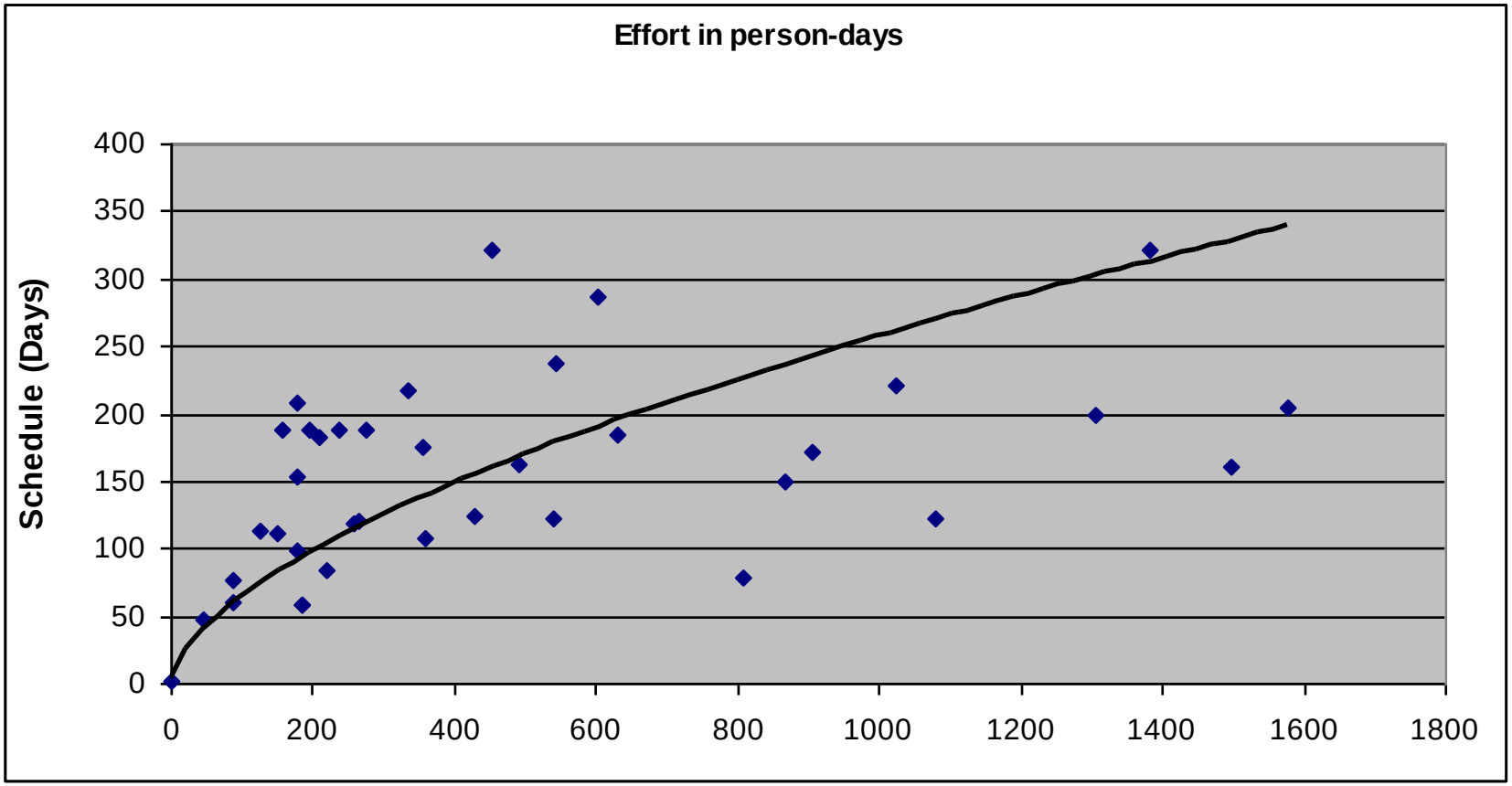
Overall Schedule

- Depends heavily on the effort estimate
- For an effort estimate, *some* flexibility exists depending on resources assigned
- Eg a 56 PM project can be done in 8 months (7 people) or 7 months (8 people)
- Stretching a schedule is easy; compressing is hard and expensive

Overall Scheduling…

- One method is to estimate schedule S (in months) as a function of effort in PMs
- Can determine the fn through analysis of past data; the function is non linear
- COCOMO: $S = 2.5\ E^{3.8}$
- Often this schedule is checked and corrected for the specific project
- One checking method – square root check

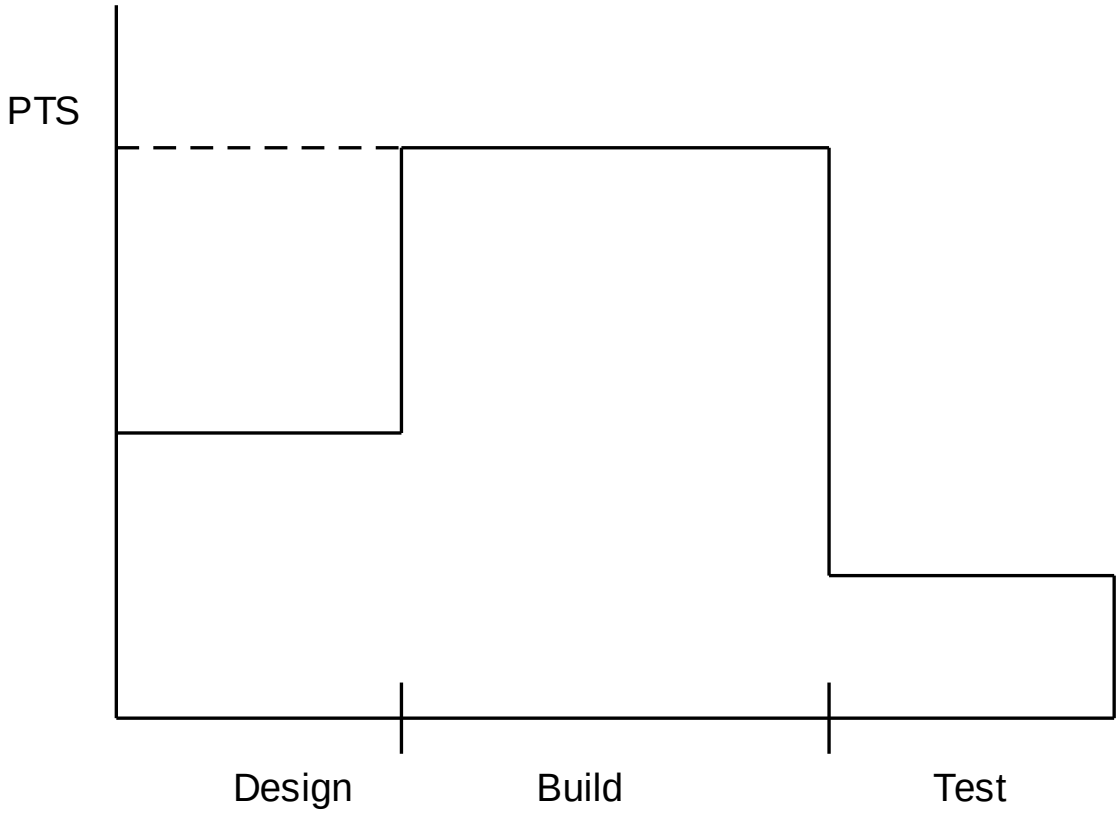
Determining Overall Schedule from past data



Determining Milestones

- With effort and overall schedule decided, avg project resources are fixed
- Manpower ramp-up in a project decides the milestones
- Manpower ramp-up in a project follows a Rayleigh curve - like a normal curve
- In reality manpower build-up is a step function

Manpower Ramp-up



Milestones …

- With manpower ramp-up and effort distribution, milestones can be decided
- Effort distribution and schedule distribution in phases are different
- Generally, the build has larger effort but not correspondingly large schedule
- COCOMO specifies distr of overall sched. Design – 19%, programming – 62%, integration – 18%

An Example Schedule

#	Task	Dur. (days)	Work (p-days)	Start Date	End Date
2	Project Init tasks	33	24	5/4	6/23
74	Training	95	49	5/8	9/29
104	Knowledge sharing	78	20	6/2	9/30
114	Elaboration iteration I	55	55	5/15	6/23
198	Construction iteration I	9	35	7/10	7/21

Detailed Scheduling

- To reach a milestone, many tasks have to be performed
- Lowest level tasks - those that can be done by a person (in less than 2-3 days)
- Scheduling - decide the tasks, assign them while preserving high-level schedule
- Is an iterative task - if cannot "fit" all tasks, must revisit high level schedule

Detailed Scheduling

- Detailed schedule not done completely in the start - it evolves
- Can use MS Project for keeping it
- Detailed Schedule is the most live document for managing the project
- Any activity to be done must get reflected in the detailed schedule

An example task in detail schedule

Module	Act Code	Task	Duration	Effort
History	PUT	Unit test # 17	1 day	7 hrs
St. date	**End date**	**%comp**	**Depend.**	**Resource**
7/18	7/18	0%	Nil	SB

Detail schedule

- Each task has name, date, duration, resource etc assigned
- % done is for tracking (tools use it)
- The detailed schedule has to be consistent with milestones
  - Tasks are sub-activities of milestone level activities, so effort should add up, total schedule should be preserved

Team Structure

- To assign tasks in detailed schedule, need to have a clear team structure
- Hierarchic team org is most common
  - Project manager has overall responsibility; also does design etc.
  - Has programmers and testers for executing detailed tasks
  - May have config controller, db manager, etc

Team structure..

- An alternative – democratic teams
  - Can work for small teams; leadership rotates
- Another one used for products
  - A dev team led by a dev mgr, a test team led by test mgr, and a prog. Mgmt team
  - All three report to a product mgr
  - Allows specialization of tasks and separate career ladders for devs, tests, PMs

SCM process and planning

- Have discussed SCM process earlier
- During planning, the SCM activities are planned along with who will perform them
- Have discussed planning also earlier
  - Includes defining CM items, naming scheme, directory structure, access restrictions, change control, versioning, release procedure etc

Quality Planning

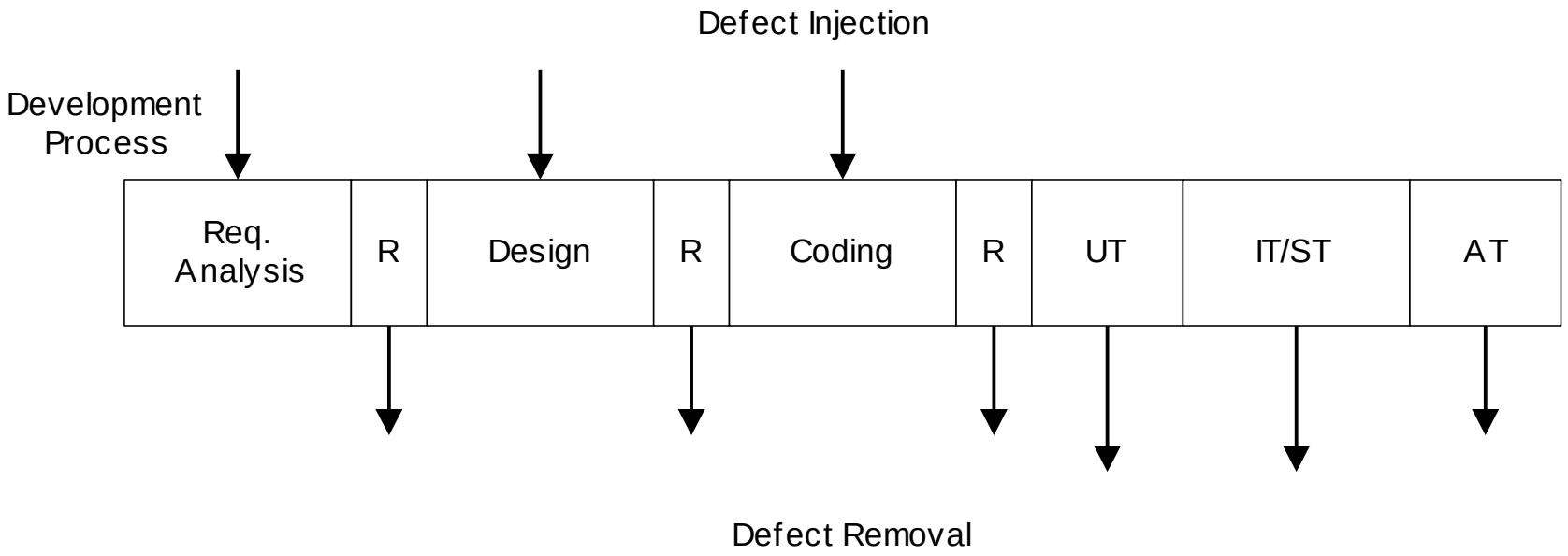
Quality Planning

- Delivering high quality is a basic goal
- Quality can be defined in many ways
- Current industry standard - *delivered defect density* (e.g. #defects/KLOC)
- Defect - something that causes software to behave in an inconsistent manner
- Aim of a project - deliver software with low delivered defect density

Defect Injection and Removal

- Software development is labor intensive
- Defects are injected at any stage
- As quality goal is low delivered defect density, these defects have to be removed
- Done primarily by quality control (QC) activities of reviews and testing

Defect Injection and Removal



Approaches to Quality Management

- Ad hoc - some testing, some reviews done as and when needed
- Procedural - defined procedures are followed in a project
- Quantitative - defect data analysis done to manage the quality process

Procedural Approach

- A quality plan defines what QC tasks will be undertaken and when
- Main QC tasks - reviews and testing
- Guidelines and procedures for reviews and testing are provided
- During project execution, adherence to the plan and procedures ensured

Quantitative Approach

- Goes beyond asking "has the procedure been executed"
- Analyzes defect data to make judgements about quality
- Past data is very important
- Key parameters - defect injection and removal rates, defect removal efficiency (DRE)

Quality Plan

- The quality plan drives the quality activities in the project
- Level of plan depends on models available
- Must define QC tasks that have to be performed in the project
- Can specify defect levels for each QC tasks (if models and data available)

Risk Management

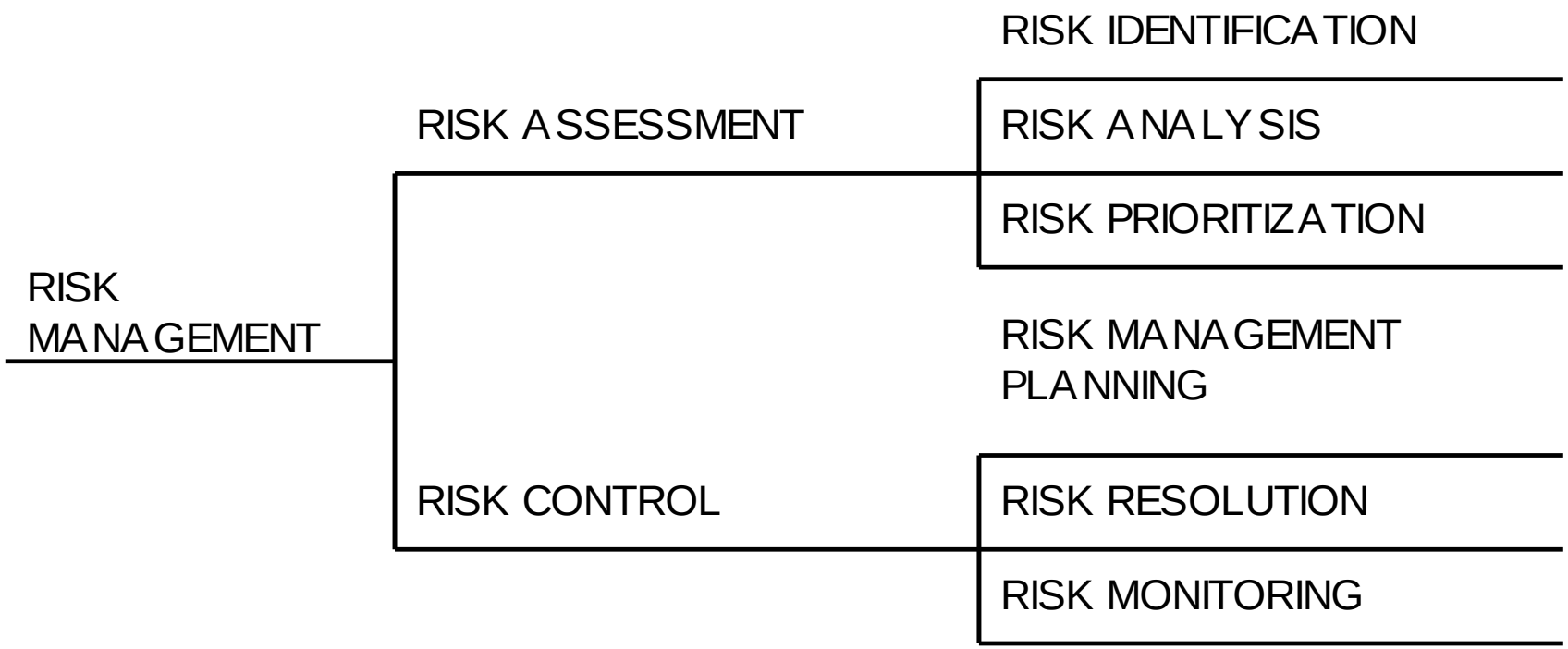
Risk Management

- Any project can fail - reasons can be technical, managerial, etc.
- Project management aims to tackle the project management aspect
- Engineering life cycles aim to tackle the engineering issues
- A project may fail due to unforeseen events - risk management aims to tackle this

Risk Management

- Risk: any condition or event whose occurrence is not certain but which can cause the project to fail
- Aim of risk management: minimize the effect of risks on a project

Risk Management Tasks



Risk Identification

- To identify possible risks to a project, i.e. to those events that might occur and which might cause the project to fail
- No "algorithm" possible, done by "what ifs", checklists, past experience
- Can have a list of "top 10" risks that projects have seen in past

Top Risk Examples

- Shortage of technically trained manpower
- Too many requirement changes
- Unclear requirements
- Not meeting performance requirements
- Unrealistic schedules
- Insufficient business knowledge
- Working on new technology

Risk Prioritization

- The number of risks might be large
- Must prioritize them to focus attention on the "high risk" areas
- For prioritization, impact of each risk must be understood
- In addition, probability of the risk occurring should also be understood

Risk Prioritization ...

- Risk exposure (RE) = probability of risk occurring * risk impact
- RE is the expected value of loss for a risk
- Prioritization can be done based on risk exposure value
- Plans can be made to handle high RE risks

A Simple approach to Risk Prioritization

- Classify risk occurrence probabilities as: Low, Medium, High
- Classify risk impact as: Low, Medium, High
- Identify those that are HH, or HM/MH
- Focus on these for risk mitigation
- Will work for most small and medium sized projects

Risk Control

- Can the risk be avoided?
  - E.g. if new hardware is a risk, it can be avoided by working with proven hardware
- For others, risk mitigation steps need to be planned and executed
  - Actions taken in the project such that if the risk materializes, its impact is minimal
  - Involves extra cost

Risk Mitigation Examples

- Too many requirement changes
  - Convince client that changes in requirements will have an impact on the schedule
  - Define a procedure for requirement changes
  - Maintain cumulative impact of changes and make it visible to client
  - Negotiate payment on actual effort.

Examples ...

- Manpower attrition
  - Ensure that multiple resources are assigned on key project areas
  - Have team building sessions
  - Rotate jobs among team members
  - Keep backup resources in the project
  - Maintain documentation of individual's work
  - Follow the CM process and guidelines strictly

Examples ...

- Unrealistic schedules
  - Negotiate for better schedule
  - Identify parallel tasks
  - Have resources ready early
  - Identify areas that can be automated
  - If the critical path is not within the schedule, negotiate with the client
  - Negotiate payment on actual effort

Risk Mitigation Plan

- Risk mitigation involves steps that are to be performed (hence has extra cost)
- It is not a paper plan - these steps should be scheduled and executed
- These are different from the steps one would take if the risk materializes - they are performed only if needed
- Risks must be revisited periodically

Project Monitoring Plans

Background

- A plan is a mere document that can guide
- It must be executed
- To ensure execution goes as per plan, it must be monitored and controlled
- Monitoring requires measurements
- And methods for interpreting them
- Monitoring plan has to plan for all the tasks related to monitoring

Measurements

- Must plan for measurements in a project
- Without planning, measurements will not be done
- Main measurements – effort, size, schedule, and defects
  - Effort – as this is the main resource; often tracked through effort reporting tools
  - Defects – as they determine quality; often defect logging and tracking systems used
- During planning – what will be measured, how, tool support, and data management

Project Tracking

- Goal: To get visibility in project execution so corrective actions can be taken when needed to ensure project succeeds
- Diff types of monitoring done at projects; measurements provide data for it

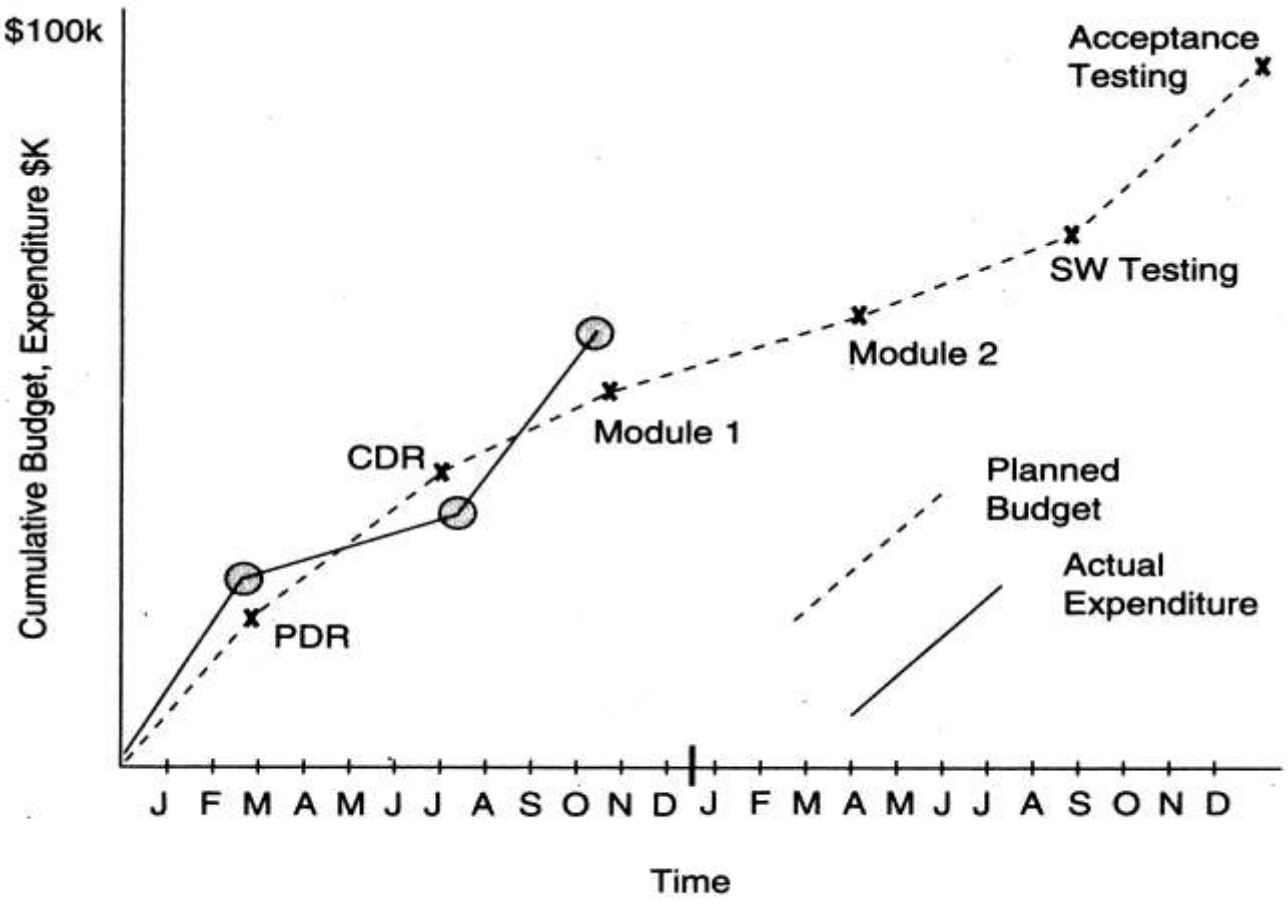
Tracking…

- Activity-level monitoring
  - Each activity in detailed schd is getting done
  - Often done daily by managers
  - A task done marked 100%; tools can determine status of higher level tasks
- Status reports
  - Generally done weekly to take stock
  - Summary of activities completed, pending
  - Issues to be resolved

Tracking...

- Milestone analysis
  - A bigger review at milestones
  - Actual vs estimated for effort and sched is done
  - Risks are revisited
  - Changes to product and their impact may be analyzed
- Cost-schedule milestone graph is another way of doing this

Cost-schedule milestone graph



Project Management Plan

- The project management plan (PMP) contains outcome of all planning activities - focuses on overall project management
- Besides PMP, a project schedule is needed
  - Reflects what activities get done in the project
  - Microsoft project (MSP) can be used for this
  - Based on project planning; is essential for day-to-day management
  - Does not replace PMP !

PMP Structure - Example

- Project overview - customer, start and end date, overall effort, overall value, main contact persons, project milestones, development environment..
- Project planning - process and tailoring, requirements change mgmt, effort estimation, quality goals and plan, risk management plan, ..

PMP Example ...

- Project tracking - data collection, analysis frequency, escalation procedures, status reporting, customer complaints, ...
- Project team, its organization, roles and responsibility, ...

Project Planning - Summary

- Project planning forms the foundation of project management
- Key aspects: effort and schedule estimation, quality planning, risk mgmt., …
- Outputs of all can be documented in a PMP, which carries all relevant info about project
- Besides PMP, a detailed project schedule maintains tasks to be done in the project