CH03 Planning and Managing the Project

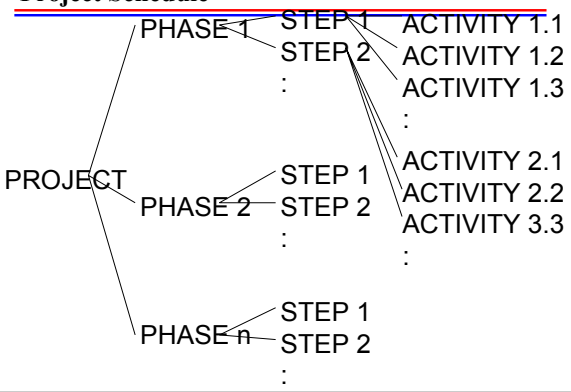
- * Tracking Progress
- * Project Personnel
- * Effort Estimation
- * Risk Management
- * The Project Plan
- * Process Models and Project Management

TECH
Computer Science

Tracking Progress

- Questions from our customers
  - **Do you understand my problem and my needs?**
  - **Can you design a system that will solve my problem or satisfy my needs?**
  - **How** long will it take **you to develop such a system?**
  - **How** much will it cost **to have you develop such a system?**
- Require a well-thought-out project schedule

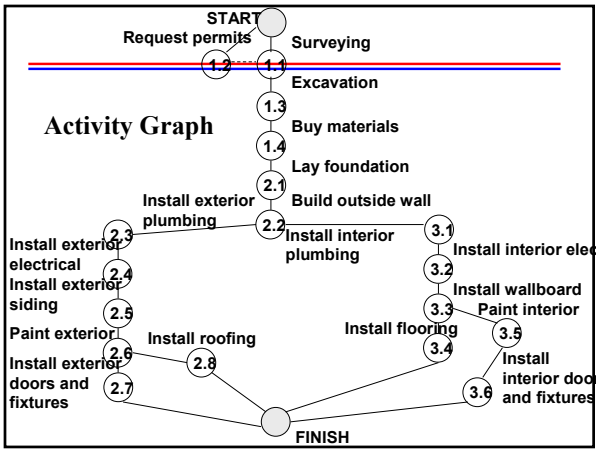
Project Schedule



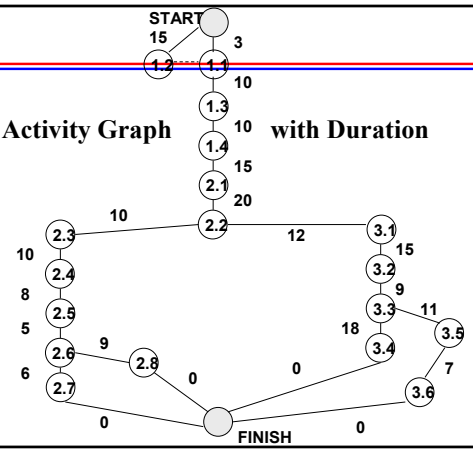
Activity and Milestone

- Activity is a part of the project that takes place over a period of time.
- Milestone is the completion of an activity -- a particular point in time.
- Describe each activity
  - **precursor**
  - **duration**
  - **due date**
  - **endpoint**

Activity Graph



Activity Graph with Duration



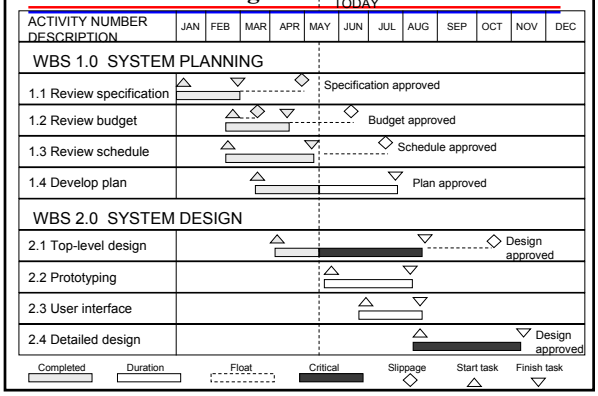
Estimating Completion

- Critical Path Method (CPM) analyzes the paths among the milestones.
  - **Shows the minimum amount of time it will take to complete the project.**
  - **Reveals those activities that are most critical to completing the project on time.**
- Trace through each path from start to finish, adding up the time.
- Critical path is the longest path.

Real time, Available time, Slack time

- **Real time** for an activity is the estimated amount of time required for the activity to be completed.
- **Available time** is the amount of time available in the schedule for the activity's completion.
- **Slack time** = available time - real time
- Slack time = latest start time - earliest start time
- e.g. surveying

Tools to Track Progress: Gantt chart



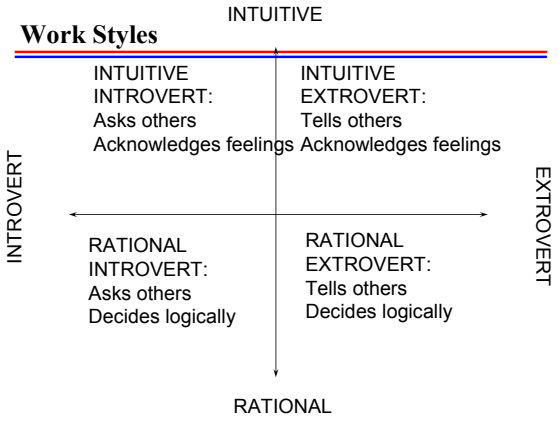
Project Personnel

- To determine the project schedule and estimate the associated effort and costs, we need to know:
  - **home many people will be working on the project,**
  - **what tasks they will perform, and**
  - **what abilities and experience they must have.**
- Who does what, and how the staff can be organized.

Staff Roles and Characteristics

- ability to perform the work
- interest in the work
- experience with similar applications, tools, languages, techniques, and development environment
- training
- ability to communicate and to share responsibility with others
- management skills

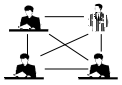
Work Styles

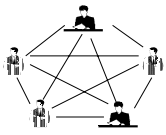


Communications

Two people  1 line of communicatio

Three people  3 lines of communicatio

Four people  6 lines of communicatio

Five people  10 lines of communicatio

:

n people

:

n(n-1)/2 lines of communication

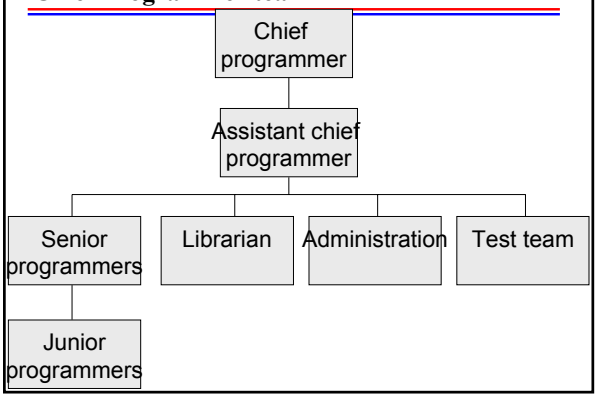
Meetings (complaints) //

- The purpose of the meeting is unclear.
- The attendees are unprepared.
- Essential people are absent or late.
- The conversation veers away from its purpose.
- Participants argue, dominate the conversation, or do not participate.
- Decisions made at the meeting are never enacted afterward.

Project Team Organization

- Team members are organized in ways that enhance the swift completion of quality products.
- Choice of an appropriate structure depends:
  - **the backgrounds and work styles of the team members**
  - **the number of people on the team**
  - **the management styles of the customers and developers**

Chief Programmer team



Egoless Approach

- holds everyone equally responsible.
- Criticism is made of the product or the result, not the people
- all team members vote on a decision.

Comparison of Organizational Structures

- Highly Structured
  - **High certainty**
  - **Repetition**
  - **Large projects**
- Loosely Structured
  - **Uncertainty**
  - **New Techniques or technology**
  - **Small projects**
  - **Creative**

Effort Estimation: How much the project COST

- Several types of costs: facilities, staff, and methods and tools.
- Facilities: 100 square feet of dedicated floor space, 30 square feet of table space, floor-to-ceiling enclosure, free from phone calls and uninvited visitors --- for each person to work effectively.
- Staff: the biggest component of cost is effort, how many staff-days
- Tools: tools and training cost.

Estimation techniques:

- Expert Judgement
- Algorithmic Methods
- Machine-learning Methods
- Finding the Model for Your Situation

Expert Judgement

- Ask an Expert
- Analogies and educated guess
- Asking three predictions:
  - **pessimistic one (x)**
  - **optimistic one (y)**
  - **Most likely guess (z)**
- Normalized by (x + 4y + z)/6

Expert Judgement: Delphi technique

- Ask each individual predictions secretly
- Calculate average estimate
- Present the estimate to the group
- Allow them to revise

Algorithmic Methods

- Express the relationship between the effort and the factors that influence it.
- $E = 5.25\ S^{0.91}$
  - **S is size in lines of code**
- $E = 5.5 + 0.73\ S^{1.16}$ (by Bailey)
- Size Estimation problem
  - **Estimates required before size information is available.**

COCOMO 2.0

- Estimates size in terms of high-level objects, such as,
- number of server data tables,
- number of client data tables,
- number of screens,
- number of reports
- percentage of screens and reports reused from previous projects

Productivity Factors

- Customer interface complexity
- User participation in requirements definition
- Customer-originated program design changes
- Customer experience with the application area
- Overall personnel experience

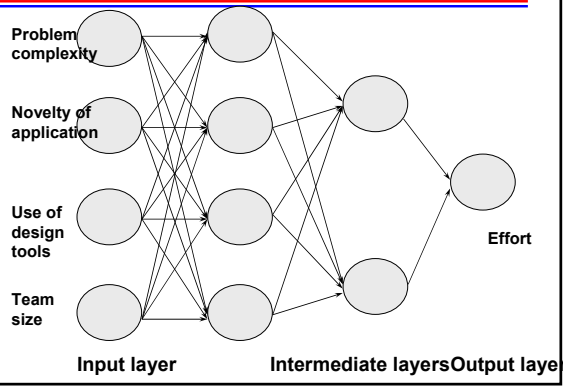
Productivity Factors (continue)

- Use of Structured Programming
- Use of design and code inspections
- Use of top-down development
- Overall complexity of code
- Complexity of program flow
- Overall constraints on program's design

Machine-learning Methods

- Learn from the past projects
- predict the future cost
- Neural Network approach
  - **training the network with data from past projects**
  - **network values are adjusted to reflect past experience**
  - **training methods, such as back-propagation**
  - **use the network to produce an effort estimate**

Neural Network e.g.



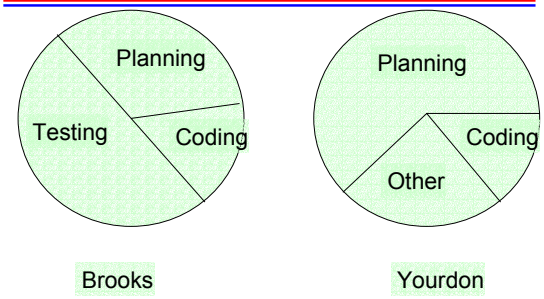
Case-based Reasoning (CBR)

- build case history of past projects
- Using CBR system
  - **User identifies a new problem as a case**
  - **it retrieves similar cases from a repository of historical information**
  - **it reuses knowledge from previous cases**
  - **it suggests a solution for the new case**
- How to characterizing cases and determining similarity

Finding the Model for Your Situation

- Which model is the best?
- Depending on your situation
- In general based on comparison data, Bailey-Basili model performs batter than others.

Different reports of effort distribution //



Risk Management

- A risk is an unwanted event that has negative consequences.
- Risk management involves understanding and controlling the risks.
- Generic Risk: common to all software projects
- Project specific Risk: particular vulnerabilities of the given project.

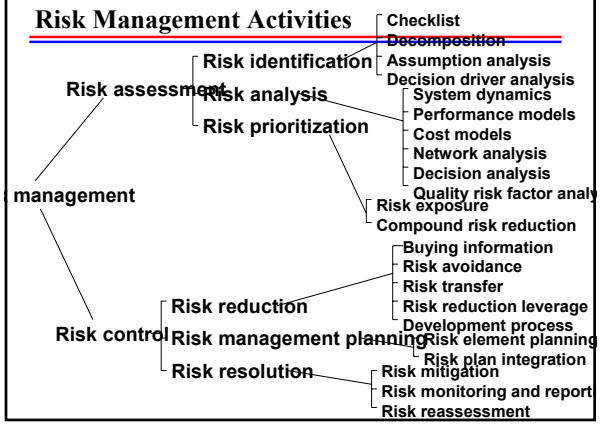
Defining and Quantifying the Risk

- Risk impart: the loss associated with a risk
- Risk probability: the likelihood for the risk to occur
- Risk exposure = risk impart * risk probability

Risk Reduction

- Risk control: a set of actions taken to reduce or eliminate a risk
- Justify the action:
- Risk leverage = (Risk exposure before reduction - risk exposure after reduction) / (cost of risk reduction)
- Record your decisions in a risk management plan.

Risk Management Activities



"Top Ten List" of Risk Items

- 1. Personnel shortfalls
- 2. Unrealistic schedules and budgets
- 3. Developing the wrong software functions
- 4. Developing the wrong user interface
- 5. Gold plating (cost overrun)

"Top Ten List" of Risk Items (more)

- 6. Continuing stream of requirements changes
- 7. Shortfalls in externally (other groups or co.) performed tasks
- 8. Shortfalls in externally furnished components
- 9. Real-time performance shortfalls
- 10. Straining computer science capabilities

The Project Plan

- We write a document called a project plan to
  - **communicate risk analysis and management,**
  - **project cost estimates,**
  - **schedule, and**
  - **organization**
- to our customers and our own group

Items for good project plan

- project scope
- project schedule
- project team organization
- technical description of the proposed system
- project standards, procedures, and proposed techniques and tools
- quality assurance plan

More Items for good project plan

- **configuration management plan**
- **documentation plan**
- **data management plan**
- **resource management plan**
- **test plan**
- **training plan**
- **security plan**
- **risk management plan**
- **maintenance plan**

Process Models and Project Management

- managing at building **quality** products on **time** and within **budget**
- tailor the project management techniques to
  - **particular characteristics of the resources needed,**
  - **the chosen process, and**
  - **the people assigned.**

Enrollment Management

- Case study: Digital Alpha AXP project
- Vision enrollment (Business Goals, Project objectives):
  - **establish and shared common goals**
- Commitment delegation (Trust, Accountability)
  - **result was measurable and identified with a particular owner**
  - **who is held accountable for delivery.**

Enrollment Management (more)

- Inspection and support (Review, Encouragement)
  - **inspect to make sure that delivery would be on time, identify risk**
  - **supportive feedback and make changes to help move the project forward**
- Acknowledgement and learning (Personal, Public)
  - **acknowledgement progress both personally and publicly**
  - **recorded what had been learned and how things could be improved**

One-page Master plan

- see the "big picture"
- have a global picture of
  - **what to do,**
  - **and when**
  - **and how to do it**
- regular operation inspections

One-page report: itemizing key points

- schedule
- milestones
- critical-path events in the past month
- activities along the critical path in the next month
- issues and dependencies resolved
- issues and dependencies not resolved (with ownership and due dates)

Recognition of job well done

- engineers are usually motivated more by recognition than by financial gain
- announcing progress in public and managers show appreciation

Flexible and focused management

- resulting in meeting schedule,
- producing exceptional product,
- meeting performance goals.

Accountability Modeling

- U.S. Air Force and Lockheed Martin (F-16 software case study)
- More than **four Million lines of code**, a quarter of which met **real-time** deadlines in flight.
- Integrated product development team:
  - **combining individuals from different functional groups into an interdisciplinary work unit,**
  - **empowered with separate channels of accountability**

Team and Stakeholder

- **Team**: any collection of people responsible for producing a given (desired) result
- **Stakeholder**: anyone affected by that result or the way in which the result is achieved.
- **Accounting**: a report of what you have done, are doing, or plan to do
- **Consequences**: with the goal of doing only what makes sense for both the team and the stakeholders

Accountable

- including a weekly team status review
- each **personal action item** had explicit closure criteria and was tracked to completion
- activity map showing progress on each activity in the overall context of the project
- earned value: a scheme for comparing activities and their contribution to the overall progress of the project

Accounting hand-off from one team to others

- coordination among different teams: use software to track the hand-off from one team to another
- Accountability model coupled with hand-off model
  - **provide mechanism for communication and coordination**
  - **encourage risk management**
  - **integrated progress reporting with problem solving**

Accountability model applied

- to the design of management systems,
- to team operation procedures,
- replacing independent behaviors with interdependence,
- emphasizing:
  "being good rather than looking good" !!!

Anchoring Milestones

- Win-Win Spiral Model
  (US. Department of Defense's STARTS program case study)
  - **reduces from $140 to $57 per delivered line of code,**
  - **quality improved from 3 to 0.035 fault per thousand delivered lines of code.**