Project Planning

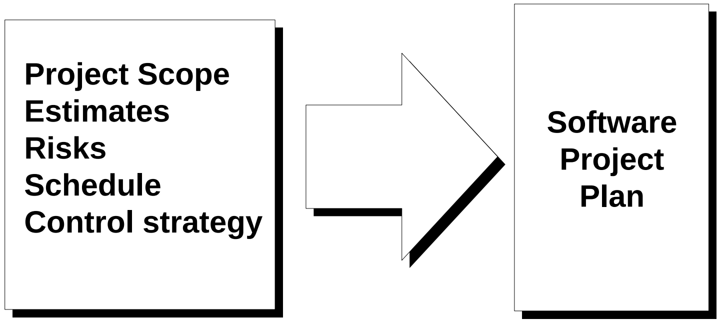
Purpose of Planning

- The overall objective of management is to deliver a product of adequate quality, on schedule, and within budget.
- To achieve control, a plan is needed.
- To know progress of project, measurement is needed and what is to be measured must be defined in the plan.
- In order to plan, it is necessary to estimate in advance the cost and schedule of the project, and to establish quality goals.
- Estimation is therefore an essential part of planning.

The Steps

- **Scoping** — Understand the problem and the work that must be done
- **Estimation** — How much effort? how much time?
- **Risk** — What can go wrong? how can we avoid it? what can we do about it?
- **Schedule** — How do we allocate resources along the timeline? what are the milestones?
- **Control Strategy** — How do we control quality? how do we control change?

Software Project Plan



Contents of Project Plan

- The relationship between the development of the software and the development of the total system within which it will be executed.
- The organization of the development team, the roles (project manager, software manager, designer, programmer, tester, secretary and technical clerk) that must be filled, and the amount of effort required for each.
- The life-cycle model to be used and the expenditure of resources within each phase, and an estimate of the total cost of the project.

Contents of Project Plan

- “Milestones” in development: times at which defined intermediate products should have been delivered (so defining the end of phase), and an estimate of the total time to completion.
- Quality goals, which can be assessed at each milestone.
- Identified risks: things (loss of key personal, requirements may change mid-project, non-availability of essential hardware), which could go wrong to prevent successful completion, and a defined course of action to be taken if the risk materializes.

To Understand Scope ...

- Understand the customers needs
- Understand the business context
- Understand the project boundaries
- Understand the customer’s motivation
- Understand the likely paths for change
- Understand that ...

Even when you understand, nothing is guaranteed!

System Development

- Some software development projects will be “purely software” that will run on commercially available platform. Other projects will involve both software and specialized hardware.
- In order to understand a complex system, we perform hierarchical decomposition. Each step in the decomposition includes more detail, and requires the input of new information.
- At the top level the system usually consists of both hardware and software, but at a certain depth, components consisting purely of software will be identified. Larger system cannot be ignored since software plan must fit in the system plans.
- Software development plan must be based on initial decomposition of the total system which identify the main functions to be performed by each software module.

Software Scope

- Software scope describes function, performance, constraints, interfaces and reliability.
- Through communication, the software engineer can collect all information regarding scope.

Team Organization

- The best team structure depends on the management style of an organization, the number of people who will be in the team & their skill levels, and the overall problem difficulty.

Project Roles

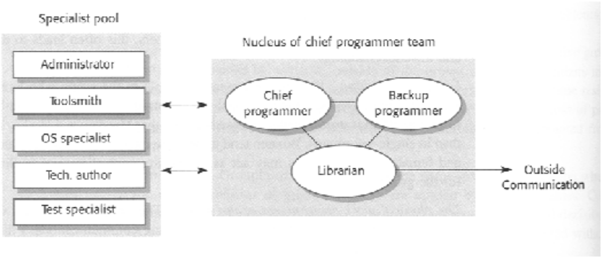
- It is necessary to decide the roles and personnel to fulfill them.
  - E.g. Project manager, Software Manager, Designer, Programmer, Tester, Secretary, Technical Clerk, etc.

Team Organization Approach

Democratic Decentralized	No permanent leader. Decisions are made by group consensus
Controlled Decentralized	Has a defined leader who coordinate specific tasks and secondary leaders responsible for subtasks. Problem solving remains a group activity, but implementation is partitioned among subgroups
Controlled Centralized	Top-level problem solving and internal team coordination are merge by a team leader.

Chief Programmer Teams

- Each module is developed by a structured team led by a "chief programmer", supported by an assistant, several junior programmers, a secretary, a technical writer and a clerk.
- The chief programmer is responsible for all design decisions.



Clean Room

- Divide into "Programmers" and "Testers".
- Based on ideas
  - Defect Prevention
  - Use of Formal Specification
  - Incremental Development
  - Statistical-based testing.

Independent Verification and Validation

- Setting up two contracts.
  - The development contractor develop the system.
  - The checking contractor monitor the activities of the developer. Similar to Clean Room but independent.
- Big cost overhead.

Verification and Validation

- **Verification -** Set of activities that ensure that software correctly implements a specific function
  - i.e. Are we building the project right?
- **Validation -** Set of activities that ensure that the software that has been built is traceable to customer requirements
  - i.e. Are we building the right product?