

Project Management....



Work Smart !!!

Project...

A collection of linked activities, carried out in an organized manner, with a clearly defined START POINT and END POINT to achieve some specific results desired to satisfy the needs of the organization at the current time.

Project Management

- A dynamic process that utilizes the appropriate resources of the organization in a controlled and structured manner, to achieve some clearly defined objectives identified as needs.
- It is always conducted within a defined set of constraints

What does Project Management Entail?

- **Planning:** is the most critical and gets the least amount of our time

Beginning with the End in mind-Stephen Covey

- **Organizing:** Orderly fashion
(Contingent/Prerequisites)
- **Controlling:** is critical if we are to use our limited resources wisely
- **Measuring:** To determine if we accomplished the goal or met the target?



Why is Project Management Important?

- Enables us to map out a course of action or work plan
- Helps us to think systematically and thoroughly
- Unique Task
- Specific Objective
- Variety of Resources
- Time bound

The Management Spectrum

Effective software project management focuses on the four P's:

- **People** — the most important element of a successful project
- **Product** — the software to be built
- **Process** — the set of framework activities and software engineering tasks to get the job done
- **Project** — all work required to make the product a reality

The People

SEI has developed a *people management capability maturity model (PM-CMM)*:

- to enhance the readiness of software organizations
- to undertake increasingly complex applications by helping
- to attract, grow, motivate, deploy, and retain the talented needed
- to improve their software development capability

The People

The people management maturity model defines:

recruiting, selection, performance management, training, compensation, career development, organization and work design, and team/culture development etc.

The Product

Before a project can be planned:

- Product objectives and scope should be established
- Alternative solutions should be considered
- Technical and management constraints should be identified

Estimates of cost, effective assessment of risk, realistic breakdown of project tasks, or manageable project schedule

The Process

A software process provides the framework for which a comprehensive plan for software development can be established.

- Task sets – tasks, milestones, work products, and quality assurance points
- Umbrella activities – software quality assurance, software configuration management, and measurement

The Project

- To manage complexity
- To avoid failure
- To develop a common sense approach for planning, monitoring, and controlling the project.

People

- Stakeholders
- The Players
- Team Leaders
- The Software Team
- Coordination and Communication Issues

The Stakeholders/Players

Five categories:

1. Senior Managers: defines business issues
2. Project (technical) managers: plan, motivate, organize, and control the practitioners
3. Practitioners: deliver the technical skills
4. Customers: specify the requirements for the software
5. End-Users: interact with the software once it is released

Team Leaders

- Project management is a people-intensive activity □ need “people skill”
- MOI model for Leadership:
 - Motivation
 - Organization
 - Ideas for innovation
- Characteristics of Effective Project Manager:
Problem Solving, Managerial Identity, Achievement, Influence and Team Building

The Software Team

- *N individuals vs. m tasks*
- Team organizations
 - Democratic decentralized (DD): no permanent leader, rather “task coordinator”, decision made by group consensus.
 - Controlled decentralized (CD): has defined leader, decision remains group activity, works partitioned
 - Controlled centralized (CC): Top-level problem solving, internal coordination

Organizational Paradigms for Team

- A Closed Paradigm
- A Random Paradigm
- An Open Paradigm
- A Synchronous Paradigm

The Software Team

Seven project factors when planning the structure of software engineering team:

- The difficulty of the problem
- The size of the resultant program
- The time
- The degree of problem to be modularized
- The required quality and reliability
- The rigidity of the delivery date
- Degree of sociability (communication)

Coordination and Communication Issues

Many reasons that software projects get into trouble:

- Scale
- Uncertainty
- Interoperability

Therefore, must establish methods for coordinating the people.

Coordination and Communication Issues

Hence, establish formal and informal communication among team members:

- Formal, impersonal approaches: SE docs and deliverables, tech memo.
- Formal, interpersonal procedures: QA activities, status review meetings and design
- Informal, interpersonal procedures: group meeting
- Electronic communication: email, forums
- Interpersonal networking: interpersonal discussion with outsiders.

The Product

- Dilemma: Quantitative estimates, but no solid information
 1. Software scope:
 2. Problem decomposition

Software Scope

- Context:
- Information objectives
- Function and performance

Software scope must be unambiguous and understandable.

Problem Decomposition

- Sometimes call partitioning or problem elaboration
- The core of software requirement analysis
 1. Functionality
 2. Process

The Process

- The generic phases that characterize the software process – definition, development, and support – are applicable to all software.
- The problem is to select the process that is appropriate for the software to be engineered by a project team.

The Process

Must decide which model is most appropriate for

1. The customers
2. The characteristics of the product
3. The project environment

The Project

Must understand what can go wrong (so that problems can be avoided)

Ten signs that indicate that an information systems project is in jeopardy:

1. Software people don't understand their customer's needs
2. The product scope is poorly defined
3. Changes are managed poorly

The Project

Ten signs (cont..)

4. The chosen technology changes
5. Business needs change (or ill-defined)
6. Deadlines are unrealistic
7. Users are resistant
8. Sponsorship is lost (or was never properly obtained)
9. The project team lacks people with appropriate skills
10. Managers (and practitioners) avoid best practices and lessons learned

The Project

Five-part commonsense approach to software project:

1. **Start on the right foot:** working hard to understand the problem
2. **Maintain momentum:** provide incentives
3. **Track progress:** track work products
4. **Make smart decisions:** decisions should be “keep it simple”
5. **Conduct a postmortem analysis:** lessons learned and evaluation of project

The W5HH Principle

Barry Boehm suggests an approach that addresses project objectives, milestones and schedules, responsibilities, management and technical approaches, and requires resources:

- Why is the system being developed?
- What will be done, by when?
- Who is responsible for a function?
- Where are they organizationally located?
- How will the job be done technically and managerially?
- How much of each resources is needed?