CHAPTER 21 PROJECT MANAGEMENT CONCEPTS

Overview

- Project management involves the planning, monitoring, and control of people, process, and events that occur during software development.
- Everyone manages, but the scope of each person's management activities varies according his or her role in the project.
- Software needs to be managed because it is a complex, long duration undertaking.
- Managers must focus on the four P's to be successful (people, product, process, and project).
- A project plan is a document that defines the four P's in such a way as to ensure a cost effective, high quality software product.
- The only way to be sure that a project plan worked correctly is by observing that a high quality product was delivered on time and under budget.

Management Spectrum

- People (recruiting, selection, performance management, training, compensation, career development, organization, work design, team/culture development)
- Product (product objectives, scope, alternative solutions, constraint tradeoffs)
- Process (framework activities populated with tasks, milestones, work products, and QA points)
- Project (planning, monitoring, controlling)

People

- Players (senior managers, project managers, practitioners, customers, end-users), also called "stakeholders"
- Team leadership model (motivation, organization, skills)
- Characteristics of effective project managers (problem solving, managerial identity, achievement, influence and team building)

Factors Affecting Team Organization

- Difficulty of problem to be solved
- Size of resulting program
- Team lifetime
- Degree to which problem can be modularized
- Required quality and reliability of the system to be built
- Rigidity of the delivery date
- Degree of communication required for the project
Team Organizational Paradigms

- Closed paradigm (top level problem solving and internal coordination managed by team leader, good for projects that repeat past efforts)
- Random paradigm (team loosely structured success depends on initiative of individual team members, paradigm excels when innovation and technical breakthroughs required)
- Open paradigm (rotating task coordinators and group consensus, good for solving complex problems - not always efficient as other paradigms)
- Synchronous paradigm (relies on natural problem compartmentalization and team organized to require little active communication with each other)

Toxic Team Environment Characteristics

1. Frenzied work atmosphere where team members waste energy and lose focus on work objectives
2. High frustration and group friction caused by personal, business, or technological problems
3. Fragmented or poorly coordinated procedures or improperly chosen process model blocks accomplishments
4. Unclear role definition that results in lack of accountability or finger pointing
5. Repeated exposure to failure that leads to loss of confidence and lower morale

Agile Teams

- Teams have significant autonomy to make their own project management and technical decisions
- Planning kept to minimum and is constrained only by business requirements and organizational standards
- Team self-organizes as project proceeds to maximize contributions of each individual's talents
- May conduct daily (10 - 20 minute) meeting to synchronize and coordinate each day's work
  - What has been accomplished since the last meeting?
  - What needs to be accomplished by the next meeting?
  - How will each team member contribute to accomplishing what needs to be done?
  - What roadblocks exist that have to be overcome?

Coordination and Communication Issues

- Formal, impersonal approaches (e.g., documents, milestones, memos)
- Formal interpersonal approaches (e.g., review meetings, inspections)
- Informal interpersonal approaches (e.g., information meetings, problem solving)
• Electronic communication (e.g., e-mail, bulletin boards, video conferencing)
• Interpersonal networking (e.g., informal discussion with people other than project team members)

The Product

• Software scope (context, information objectives, function, and performance)
• Problem decomposition (partitioning or problem elaboration - focus is on functionality to be delivered and the process used to deliver it)

The Process

• Process model chosen must be appropriate for the: customers and developers, characteristics of the product, and project development environment
• Project planning begins with melding the product and the process
• Each function to be engineered must pass through the set of framework activities defined for a software organization
• Work tasks may vary but the common process framework (CPF) is invariant (project size does not change the CPF)
• The job of the software engineer is to estimate the resources required to move each function through the framework activities to produce each work product
• Project decomposition begins when the project manager tries to determine how to accomplish each CPF activity

Signs of Potential Project Failure

1. Developers do not understand customer's needs
2. Product scope poorly defined
3. Changes poorly managed
4. Chosen technology changes
5. Business needs change or ill-defined
6. Deadlines unrealistic
7. Users are resistant
8. Sponsorship lost or never obtained
9. Project team members lack appropriate skills
10. Managers and practitioners avoid best practices and lessons learned

Avoiding Project Failure

1. Start on the right foot
2. Maintain momentum
3. Track progress
4. Make smart decisions
5. Conduct a postmortem analysis

W5HH Principle
• Why is the system being developed?
• What will be done?
• When will it be accomplished?
• Who is responsible for a function?
• Where are they organizationally located?
• How will the job be done technically and managerially?
• How much of each resource is needed?

Critical Practices

• Formal risk management
• Empirical cost and schedule estimation
• Metric-based project management
• Earned value tracking
• Defect tracking against quality targets
• People-aware program management