

# Project Management Concepts Based on Chapter 24 of the textbook



- 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.



- Senior managers define the business issues that often have significant influence on the project.
- Project (technical) managers who must plan, motivate, organize, and control the practitioner.
- Practitioners who deliver the technical skills that are necessary to engineer a product or application.
- Customers specify requirements for the software to be engineered and other interested product stakeholders.
- End-users interact with the software once it is released for production use.

### Team Leaders

Kouzes exemplary practices for technology leaders:

- Model the way. Leaders must practice what they preach. They
  demonstrate commitment to team and project by shared
  sacrifice.
- Inspire and shared vision. Motivate team members to tie their personal aspirations to team goals. Involve stakeholders early.
- Challenge the process. Encourage team members to experiment and take risks by helping them generate frequent small successes while learning from their failures.
- Enable others to act. Increase the team's sense of competence through sharing decision making and goal setting.
- Encourage the heart. Build community (team) spirit by celebrating shared goals and victories (individual and team).



- Difficulty of the problem to be solved.
- Size of the resultant program(s) in lines of code or function points.
- Team lifetimes time that the team will stay together.
- Degree to which the problem can be modularized.
- Required quality and reliability of the system to be built.
- Rigidity of the delivery date.
- Communication (degree of sociability) required.



- Frenzied work atmosphere team members waste energy and lose focus on work objectives.
- High frustration caused by personal, business, or technological factors causing team member friction.
- Fragmented or poorly coordinated procedures poorly defined or improperly chosen process model.
- Unclear definition of roles resulting in lack of accountability and resultant finger-pointing.
- Continuous and repeated exposure to failure leads to a loss of confidence and a lowering of morale.



- Team members must have trust in one another.
- The distribution of skills must be appropriate to the problem.
- Mavericks may have to be excluded from the team, if team cohesiveness is to be maintained.
- Team is "self-organizing."
  - An adaptive team structure.
  - Planning is kept to a minimum.
  - Team select its own approach constrained by business requirements and organizational standards.



- Scale of many development efforts is large, leading to complexity, confusion, and significant difficulties in coordinating team members.
- Uncertainty is common, resulting in a continuing stream of changes that ratchets the project team.
- Interoperability new software must communicate with existing software and conform to constraints imposed by existing systems or products.



- To accomplish this, mechanisms for formal and informal communication among team members and between multiple teams must be established.
- Formal communication is accomplished through writing, structured meetings, and other relatively non-interactive and impersonal communication channels.
- Informal communication is more personal and allow team members to interact with one another on a daily basis - share ideas on an ad hoc basis and ask for help as problems arise.



- Software project scope must be unambiguous and understandable at management and technical levels.
- Context. How does the software to be built fit into a larger system, product, or business context and what constraints are imposed as a result of the context?
- Information objectives. What customer-visible data objects are produced as output from the software? What data objects are required for input?
- Function and performance. What function does the software perform to transform input data into output? Are any special performance characteristics to be addressed?



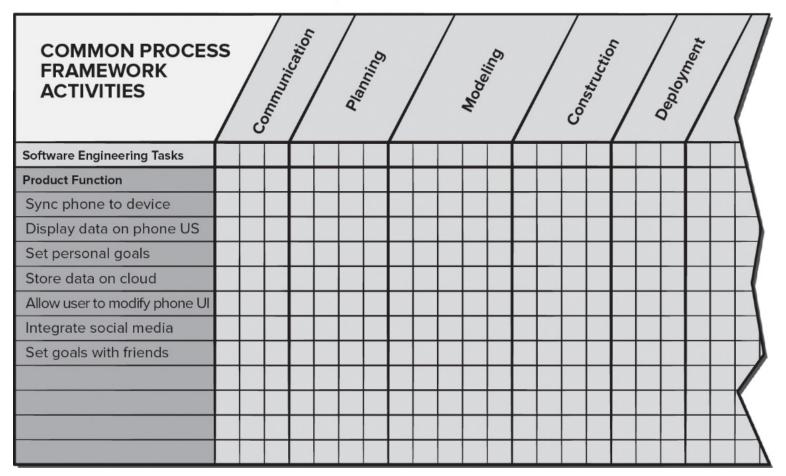
- Sometimes called partitioning or problem elaboration.
- Once scope is defined ...
  - It is decomposed into constituent functions.
  - It can be decomposed into user-visible data objects. or
  - It can be decomposed into a set of problem classes.
- Decomposition process continues until all functions or problem classes have been defined.

## **Process**

- Your team must decide which process model is most appropriate for.
  - the customers who have requested the product.
  - the people who will do the work.
  - the characteristics of the product itself.
  - the project environment in which the software team works.
- Team selects process model and defines a preliminary project plan (based set of process framework activities).
- Once the preliminary plan is established, process decomposition begins.



Copyright © McGraw-Hill Education. All rights reserved. No reproduction or distribution without the prior written consent of McGraw-Hill Education.



# Successful Project Characteristics

- Clear and well-understood requirements accepted by all stakeholders.
- Active and continuous participation of users throughout the development process.
- A project manager with required leadership skills who is able to share project vision with the team.
- A project plan and schedule developed with stakeholder participation to achieve user goals.
- Skilled and engaged team members.



- Development team members with compatible personalities who enjoy working in a collaborative environment.
- Realistic schedule and budget estimates which are monitored and maintained.
- Customer needs that are understood and satisfied.
- Team members who experience a high degree of job satisfaction.
- A working product that reflects desired scope and quality.



- Why is the system being developed?
- What will be done?
- When will it be accomplished?
- Who is responsible?
- Where are they organizationally located?
- How will the job be done technically and managerially?
- How much of each resource (for example, people, software, tools, database) will be needed?



#### **Critical Practices**

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