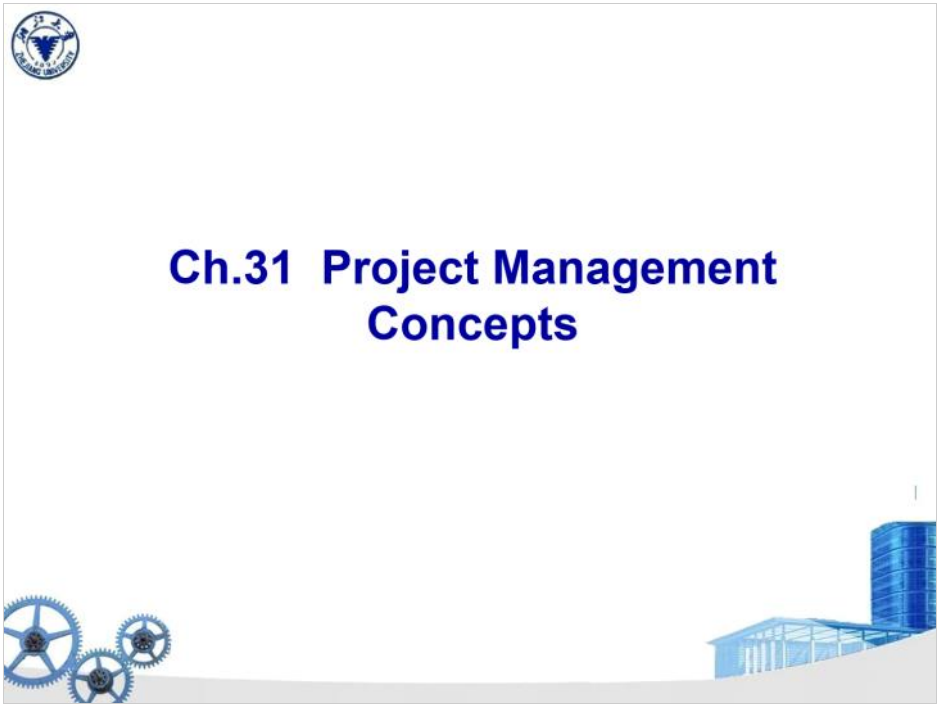
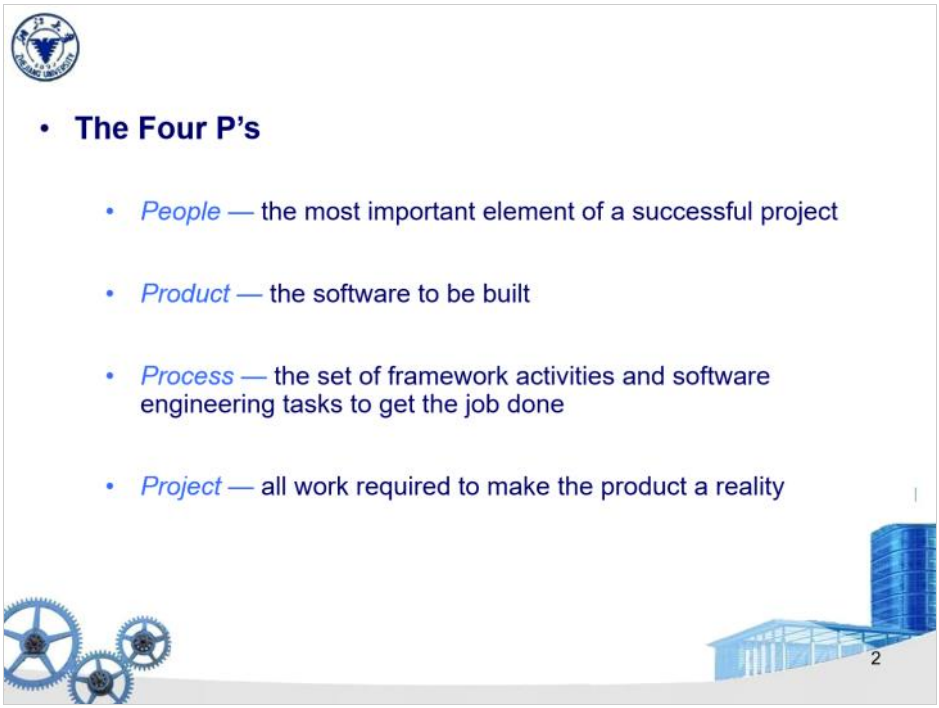
Lecture Note 31

2024年3月17日 18:58





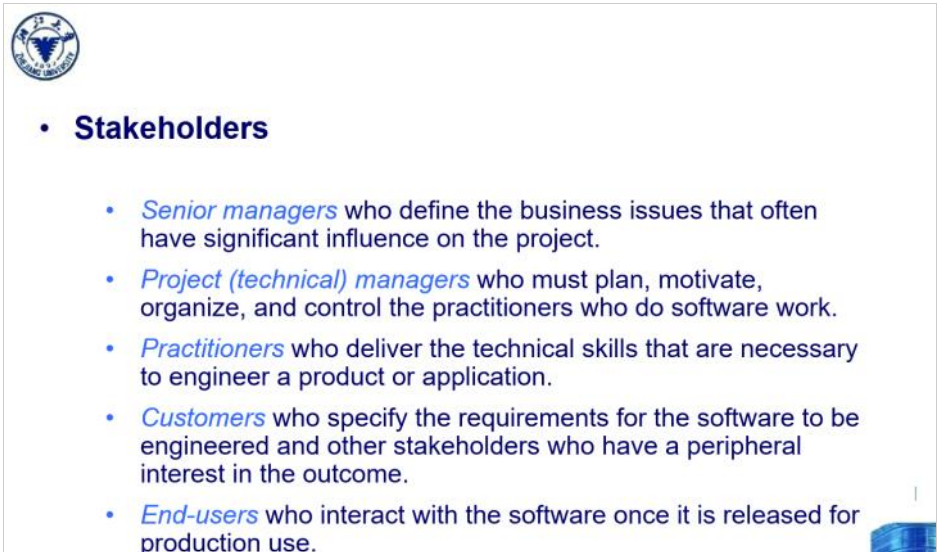
项目管理：

1、管人

2、管产品

3、管过程：大致分为哪些活动，哪些是framework，哪些是支撑性活动

4、eg. 资源、协调、宣传



利益相关者

1、高级管理者：给资源，决定商业方向

2、项目管理者：技术头和总头（eg. 总设计师、总指挥）

3、实践者：技术人员

4、客户：把需求告诉你

5、终端用户：操作员

interest in the outcome.

- *End-users* who interact with the software once it is released for production use.

3

Software Teams

4

Team Leader

- *The MOI Model*
  - *Motivation.* The ability to encourage (by "push or pull") technical people to produce to their best ability.
  - *Organization.* The ability to mold existing processes (or invent new ones) that will enable the initial concept to be translated into a final product.
  - *Ideas or innovation.* The ability to encourage people to create and feel creative even when they must work within bounds established for a particular software product or application.

5

团队领导者

1、需要能够激励人，激发人的潜质

2、组织能力

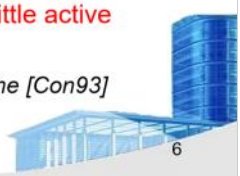
3、创新能力

Organizational Paradigms

- *closed paradigm*—structures a team along a traditional hierarchy of authority
- *random paradigm*—structures a team loosely and depends on individual initiative of the team members
- *open paradigm*—attempts to structure a team in a manner that

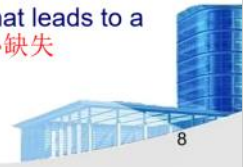
- *random paradigm*—structures a team loosely and depends on individual initiative of the team members
- *open paradigm*—attempts to structure a team in a manner that achieves some of the controls associated with the closed paradigm but also much of the innovation that occurs when using the random paradigm
- *synchronous paradigm*—relies on the natural compartmentalization of a problem and organizes team members to work on pieces of the problem with little active communication among themselves

suggested by Constantine [Con93]



- **Software Teams**
  - *The following factors must be considered when selecting a software project team structure ...*
    - the *difficulty of the problem* to be solved
    - the *size of the resultant program(s)* in lines of code or function points
    - the *time that the team will stay together* (team lifetime)
    - the *degree to which the problem can be modularized*
    - the *required quality and reliability* of the system to be built
    - the *rigidity of the delivery date*
    - the *degree of sociability (communication)* required for the project

- **Avoid Team "Toxicity"**
  - A frenzied work atmosphere in which team members waste energy and lose focus on the objectives of the work to be performed. 目标混乱
  - High frustration caused by personal, business, or technological factors that cause friction among team members.矛盾重重
  - "Fragmented or poorly coordinated procedures" or a poorly defined or improperly chosen process model that becomes a roadblock to accomplishment.管理不善
  - Unclear definition of roles resulting in a lack of accountability and resultant finger-pointing.分工不明
  - "Continuous and repeated exposure to failure" that leads to a loss of confidence and a lowering of morale.信心缺失

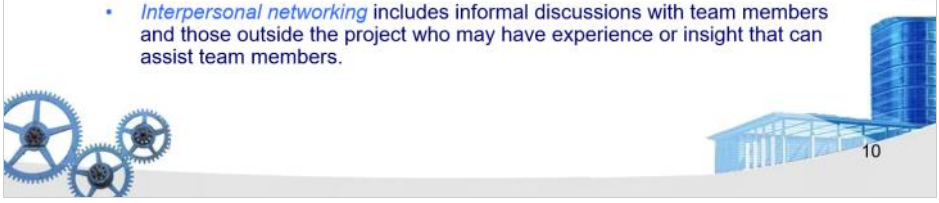


- **Agile Teams**

- **Agile Teams**
  - Team members must have trust in one another. 相互信任
  - The distribution of skills must be appropriate to the problem. 技能匹配
  - Mavericks may have to be excluded from the team, if team cohesiveness is to be maintained.去除刺头
  - Team is "self-organizing"
  - An adaptive team structure
  - Uses elements of Constantine's random, open, and synchronous paradigms
  - Significant autonomy

9

- **Team Coordination & Communication**
  - *Formal, impersonal approaches* include software engineering documents and work products (including source code), technical memos, project milestones, schedules, and project control tools (Chapter 23), change requests and related documentation, error tracking reports, and repository data (see Chapter 26).
  - *Formal, interpersonal procedures* focus on quality assurance activities (Chapter 25) applied to software engineering work products. These include status review meetings and design and code inspections.
  - *Informal, interpersonal procedures* include group meetings for information dissemination and problem solving and "collocation of requirements and development staff."
  - *Electronic communication* encompasses electronic mail, electronic bulletin boards, and by extension, video-based conferencing systems.
  - *Interpersonal networking* includes informal discussions with team members and those outside the project who may have experience or insight that can assist team members.



10

沟通协调

1、正式的，和人无关的：文档、资料

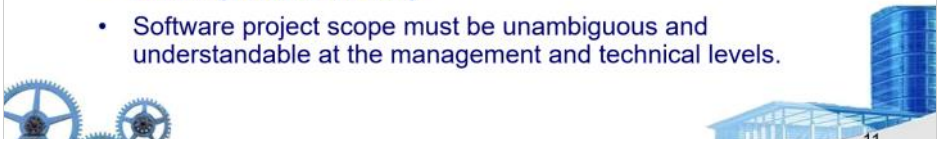
2、正式、人际之间：代码走查、技术复审

3、非正式的，人际：开会

4、电子化通信方式

5、人之间的网络：技术讲座

- **The Product Scope**
  - Scope
  - *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 (Chapter 8) 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?
  - Reliability,Interface,Security
  - Software project scope must be unambiguous and understandable at the management and technical levels.



产品范围

1、软件上下文及环境

2、定义信息目标：输入？输出？对

3、功能和性能：如何把input转换成output？能支撑多少用户？

4、软件可靠性、接口、安全：24小时不宕机？面向人，面向机器的接口？认证、授权、加密、解密、DOS攻击

understandable at the management and technical levels.

11

- **Problem Decomposition**
  - Sometimes called *partitioning* or *problem elaboration*
  - Once scope is defined …
    - It is decomposed into constituent functions
    - It is decomposed into user-visible data objects

    or
    - It is decomposed into a set of problem classes
  - Decomposition process continues until all functions or problem classes have been defined

12

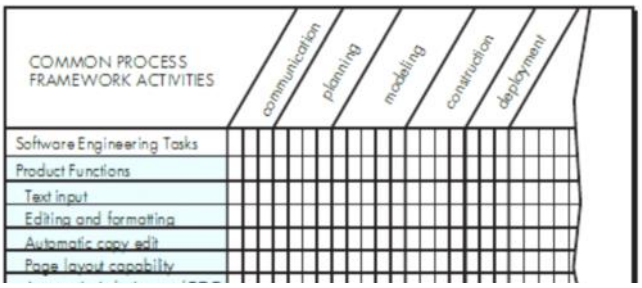
问题分解法
对每个scope细分到一级目录、二级目录
…
分而治之

- **The Process**
  - Once a process framework has been established
    - Consider project characteristics
    - Determine the degree of rigor required
    - Define a task set for each software engineering activity
      - Task set =
        - Software engineering tasks
        - Work products
        - Quality assurance points
        - Milestones

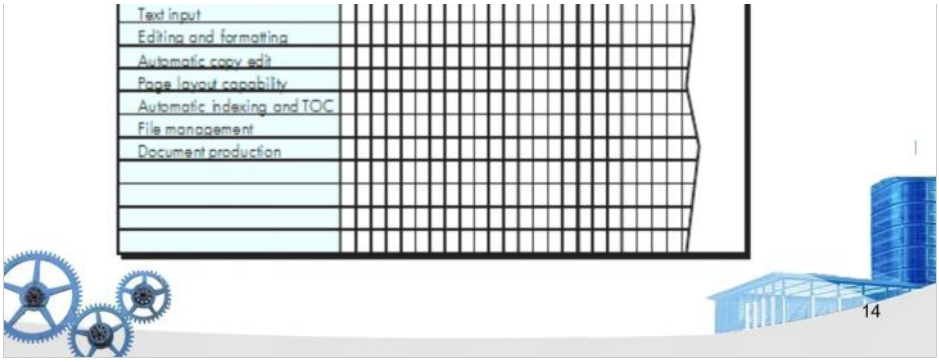
13

将过程任务定义
要做流程适配
1、软件特征
2、严格程度
3、任务集合

- **Melding the Problem and the Process**



横坐标是任务，纵坐标是问题



- **The Project**
  - Projects get into trouble when …
    - Software people don't understand their customer's needs.
    - The product scope is poorly defined.
    - Changes are managed poorly.
    - The chosen technology changes.
    - Business needs change [or are ill-defined].
    - Deadlines are unrealistic.
    - Users are resistant.
    - Sponsorship is lost [or was never properly obtained].
    - The project team lacks people with appropriate skills.
    - Managers [and practitioners] avoid best practices and lessons learned.

15

项目管理

导致困境：

1、团队没理解用户需求
2、产品没有清晰定义
3、更改没有很好的管理
4、技术改变
5、商业需求改变
6、ddl不现实
7、用户抗拒
8、管理者经验不足

- Common-Sense Approach to Projects
  - *Start on the right foot.* This is accomplished by working hard (very hard) to understand the problem that is to be solved and then setting realistic objectives and expectations.
  - *Maintain momentum.* The project manager must provide incentives to keep turnover of personnel to an absolute minimum, the team should emphasize quality in every task it performs, and senior management should do everything possible to stay out of the team's way.
  - *Track progress.* For a software project, progress is tracked as work products (e.g., models, source code, sets of test cases) are produced and approved (using formal technical reviews) as part of a quality assurance activity.
  - *Make smart decisions.* In essence, the decisions of the project manager and the software team should be to "keep it simple."
  - *Conduct a postmortem analysis.* Establish a consistent mechanism for extracting lessons learned for each project.

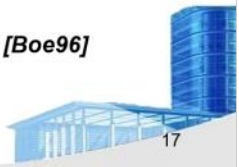
16

1、先定义清楚问题
2、持续保持动力
3、进展要做分析
4、做减法，角色简单
5、总结分析

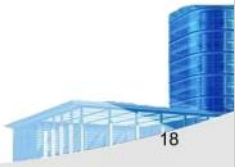
- To Get to the Essence of a Project
  - *W*hy is the system being developed?

- *W*hy is the system being developed?
- *W*hat will be done?
- *W*hen will it be accomplished?
- *W*ho is responsible?
- *W*here are they organizationally located?
- *H*ow will the job be done technically and managerially?
- *H*ow much of each resource (e.g., people, software, tools, database) will be needed?

Barry Boehm [Boe96]



- 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



过程实践