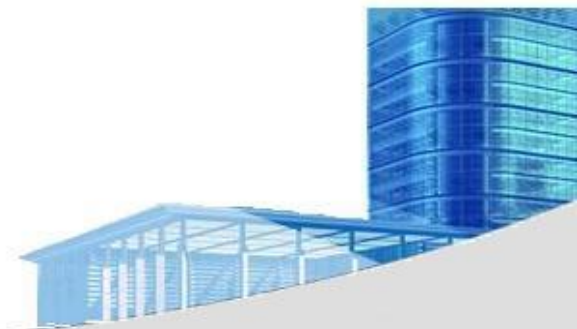




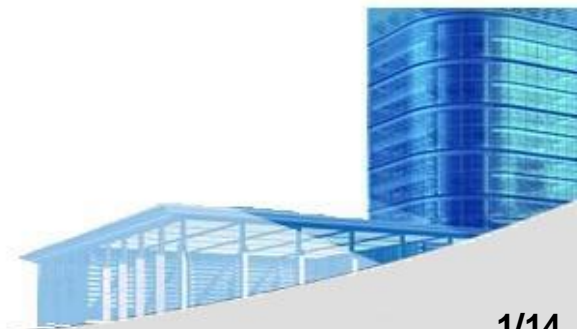
第21章项目管理概念





21.1 4p

- 人——一个成功的项目最重要的因素
- 产品——要开发的软件
- 过程——完成工作的框架活动和软件工程任务的集合
- 项目——使产品成为现实所需的所有工作

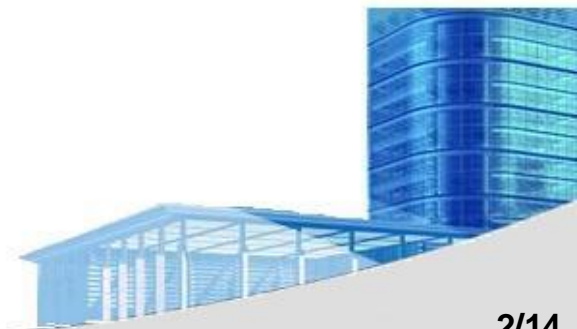




21.2人

- 利益相关者

1. 高级经理:他们定义了通常对项目有重大影响的业务问题。
2. 项目(技术)经理:他们必须计划、激励、组织和控制从事软件工作的从业人员。
3. 从业者:交付设计产品或应用所必需的技术技能的人。
4. 客户:指定要设计的软件的需求的人, 以及对结果有外围兴趣的其他利益相关者。
5. 最终用户:一旦软件发布用于生产使用, 他们就会与软件进行交互。





21.2人

如何引导？

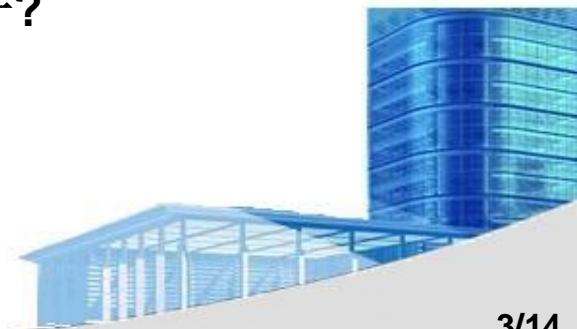
如何组织？

如何合作？



如何激励？

如何创造好点子？



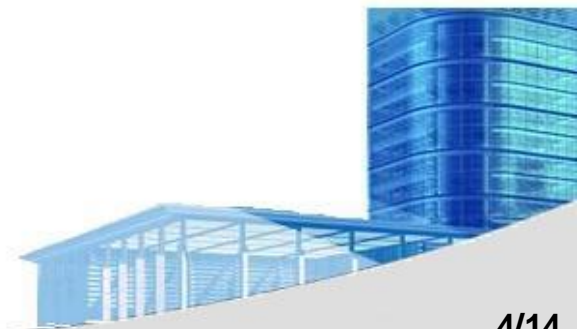


21.2人

- 团队领导

- 我的模型

- 动机。鼓励(通过“推或拉”)技术人员发挥其最大能力的能力。
- 组织。塑造现有流程(或发明新流程)的能力，使初始概念能够转化为最终产品。
- 观点或创新。鼓励人们创造和感到有创造力的能力，即使他们必须在为特定软件产品或应用程序确定的范围内工作。



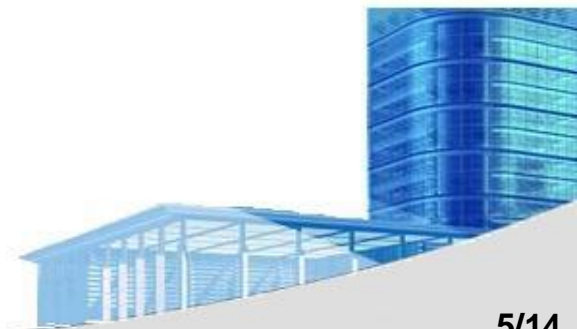


21.2人

• 软件团队

选择a时必须考虑以下因素
软件项目团队结构...

- 要解决问题的难度
- 结果程序的代码行或功能点的大小
- 团队待在一起的时间(团队生命周期)
- 问题可以模块化的程度
- 需要建立的系统的质量和可靠性
- 交货期的刚性
- 项目所需的社交(沟通)程度

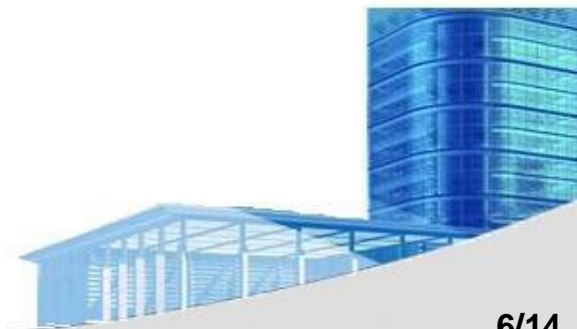




21.2人

- 组织模式

- 封闭式范式——按照传统的权威等级来组织团队
- 随机范式——松散地组建团队，并依赖于团队成员的个人主动性
- 开放范式——试图以一种方式来组织团队，这种方式既能实现与封闭范式相关的一些控制，也能实现在使用随机范式时发生的大部分创新
- 同步范式——依赖于问题的自然划分，并组织团队成员在彼此之间很少主动沟通的情况下处理问题的碎片

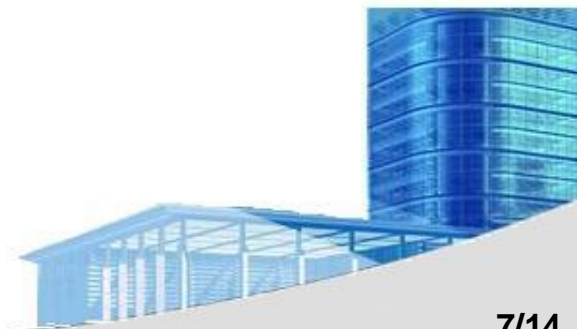




21.2人

• 避免团队“中毒”

- 一种疯狂的工作氛围，在这种氛围中，团队成员浪费精力，对要完成的工作目标失去关注。
- 由个人、业务或技术因素引起的高度沮丧，导致团队成员之间产生摩擦。
- “碎片化或协调不良的程序”或定义不佳或选择不当的过程模型成为成就的障碍。
- 角色定义不清，导致责任缺失，导致相互指责。
- “持续、反复地暴露在失败面前”，导致信心丧失和士气低落。



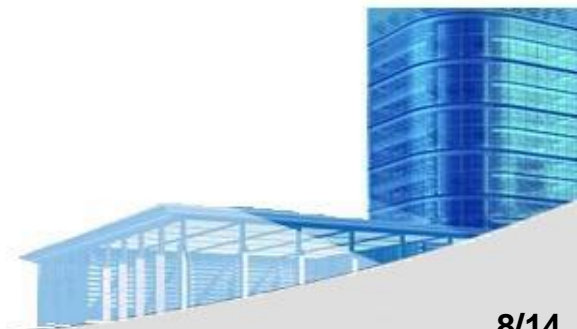


21.3 产品

• 软件范围

- 上下文。要构建的软件如何适应更大的系统、产品或业务环境，以及由于环境而施加的约束是什么？
- 信息的目标。软件输出的是什么客户可见的数据对象？输入需要哪些数据对象？
- 功能和性能。软件执行什么功能将输入数据转换为输出？有什么特殊的性能特征需要处理吗？

软件项目范围在管理和技术层面必须是明确的和可理解的。

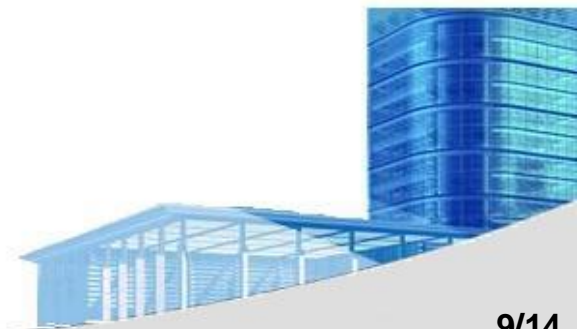




21.3 产品

- 问题分解

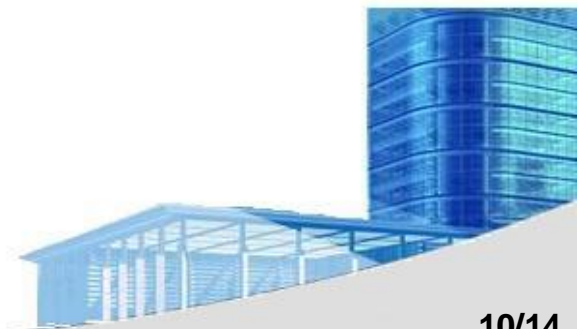
- 有时被称为划分或问题细化
- 一旦定义了范围.....
 - ✓ 它被分解为组成函数
 - ✓ 它被分解为用户可见的数据对象
或
 - ✓ 它被分解成一组问题类
- 分解过程持续进行，直到所有的函数或问题类都被定义





21.4过程

- 一旦建立了过程框架，选择适当的过程模型
 - 考虑项目特点
 - 确定所需的严格程度
 - 为每个软件工程活动定义一个任务集
 - 任务设置=
 - 软件工程任务
 - 工作产品
 - 质量保证分
 - 里程碑

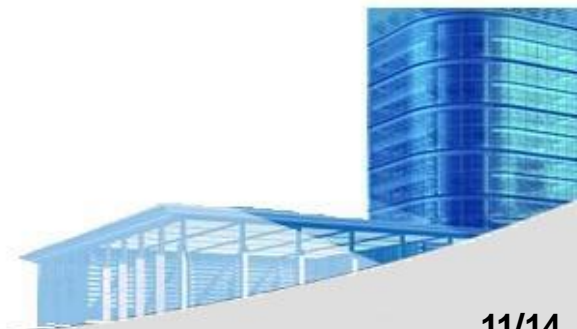




21.5项目

- 项目陷入困境时.....

- 软件开发人员不了解客户的需求。
- 产品范围定义不清。
- 变更管理不善。
- 所选择的技术发生了变化。
- 业务需求改变(或者定义不清)。
- 最后期限是不现实的。
- 用户是耐药。
- 失去了赞助(或者从未获得过)。
- 项目团队缺乏具有适当技能的人员。
- 管理者(和从业者)回避最佳实践和经验教训。

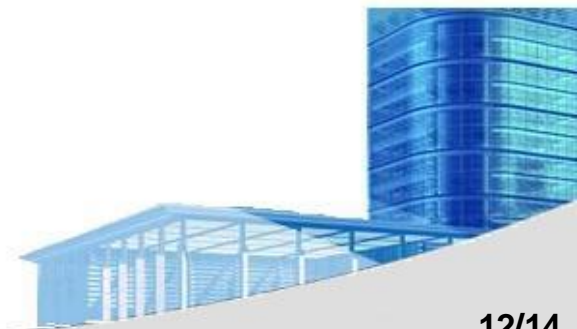




21.5项目

- 项目的常识性方法

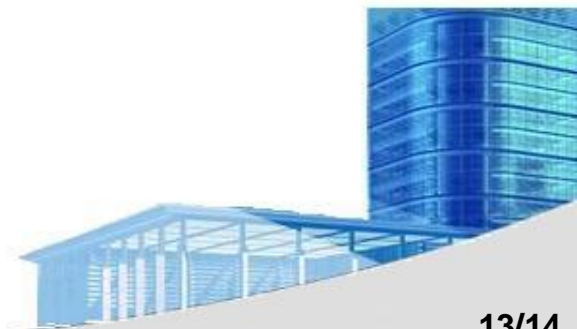
- 从正确的起点开始:设定现实的目标和期望。
- 保持势头:强调完成每一项任务的质量, 高级管理人员应该尽可能地远离团队。
- 跟踪进度:作为质量保证活动的一部分。
- 做出明智的决定:“保持简单。”
- 进行事后分析:建立一个一致的机制来提取每个项目的经验教训。





21.6 w5hh原则

- 为什么要开发这个系统？
- 将要做什么？
- 什么时候能完成？
- 负责的是谁？
- 他们的组织定位在哪里？
- 技术和管理方面的工作将如何完成？
- 每种资源(例如，人、软件、工具、数据库)需要多少？





21.7 关键实践

1. 经验成本和进度估计
2. 参数标准项目管理
3. 挣值跟踪
4. 正式的风险管理
5. 针对质量目标的缺陷跟踪
6. 人们意识到管理

