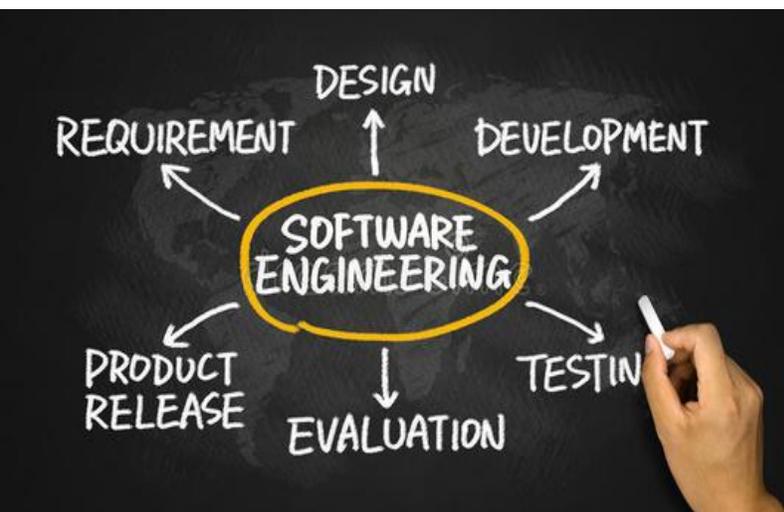




软件工程基础

—— 第22章 项目管理概念



计算机学院 孟宇龙

22.1 管理涉及的范围

22.2 人员

22.3 产品

22.4 过程

22.5 项目

22.6 W⁵HH原则

22.7 关键实践

关键概念

- 敏捷团队
- 协调和沟通
- 关键实践
- 人员
- 问题分解
- 产品
- 项目
- 软件范围
- 软件团队
- 利益相关者
- 团队负责人
- W⁵HH原则

有效的软件项目管理-4个P

- **人员** —— 一个成功项目最重要的元素
- **产品** —— 建立的软件
- **过程** —— 框架活动集和完成工作的软件工程任务
- **项目** —— 所有需要做的工作，以使一个产品变成现实

- 人员配备
- 沟通与协调
- 工作环境
- 业绩管理
- 培训
- 报酬
- 能力素质分析与开发
- 个人事业发展
- 工作组发展
- 团队精神或企业文化培养

People-CMM成熟度

22.1.2 产品

- 确定产品的目标和范围
 - 识别出产品的总体目标（从利益相关者的角度）不考虑如何实现
 - 标识出产品的主要数据、功能和行为特性并以量化的方式界定这些特性
- 考虑可选的解决方案
- 识别技术和管理上的限制
 - 产品的交付日期、预算限制、可用人员、技术接口等

22.1.3 过程

- **框架**

在框架下制定软件开发的综合计划

- **多种不同的任务集合**

每种集合都由任务、里程碑、工作产品及质量保证点组成

- **普适性活动**

软件质量保证、软件配置管理、测量等

22.1.4 项目

- 避免一些常见的警告信号
- 了解实施成功的项目管理的关键因素
- 确定计划和监控项目的一目了然的方法

- **高级管理者**：负责定义业务问题，这些问题往往对项目产生很大影响。
- **项目 (技术) 管理者**：必须计划、激励、组织和控制软件开发人员。
- **开发人员**：拥有开发产品或应用软件所需技能的人员。
- **客户**：阐明待开发软件需求的人员以及关心项目成败的其他利益相关者。
- **最终用户**：一旦软件发布成为产品，最终用户就是直接与软件进行交互的人。

22.2.2 团队负责人

如何领导？

如何组织？

如何合作？



如何激励？

如何产生好的创意？

- MOI 模型 (Motivation, Organization, Ideas or Innovation)
 - **激励**。(通过“推”或“拉”)鼓励技术人员发挥其最大才能的一种能力。
 - **组织**。形成能够将最初概念转换成最终产品的现有过程 (或创造新的过程) 的能力。
 - **思想或创新**。即使必须在特定软件产品或应用系统的约束下工作，也能鼓励人们去创造并让人感到有创造性的一种能力。

22.2.3 软件团队

选择软件项目团队结构时应该考虑以下7个因素：

- 待解决问题的难度
- 开发程序的规模，以代码行或者功能点来度量
- 团队成员需要共同工作的时间 (团队生存期)
- 能够对问题做模块化划分的程度
- 待开发系统的质量要求和可靠性要求
- 交付日期的严格程度
- 项目所需要的友好交流的程度

- **封闭式范型** ——按照传统的权利层次来组织团队
1个高级工程师（主程序员），2-5个技术人员，1个后备工程师
- **随机式范型** ——松散地组织团队，团队工作依赖于团队成员个人的主动性
- **开放式范型** ——试图以一种既具有封闭式范型的控制性，又包含随机式范型的创新性的方式来组织团队
- **同步式范型** ——依赖于问题的自然划分，组织团队成员各自解决问题的一部分，他们之间没有什么主动的交流

- 狂乱的工作氛围；
使团队成员浪费精力，同时也使他们在工作中表现出毫无目的性。
- 引起团队成员间产生摩擦的重大挫折
由个人、商业和技术因素引起的重大挫折导致团队成员间产生摩擦。
- “碎片式的或协调很差”的软件过程
缺乏定义的或选择不合适的过程模型都会成为成功路上的路障。
- 在软件团队中没有清晰的角色定义
不清晰的角色定义导致缺乏责任，并相互指责。
- “接连不断地重蹈覆辙”
使团队成员失去信心并降低斗志。

- 项目经理应确保团队可获取完成工作所需的所有信息；主要目标一旦确定（除非绝对必要），就不应该修改
- 给予软件团队尽可能多的决策权。
- 理解将要开发的产品和完成工作的人员，以及允许团队选择过程模型。
- 团队本身应该建立自己的责任机制（技术评审），并规定一系列当团队未能完成任务时的纠正方法。
- 建立基于团队的信息反馈方法和解决问题的技术。

22.2.4 敏捷团队

- 团队成员必须相互信任。
- 团队成员的技能分布必须适合于要解决的问题。
- 如果要保持团队的凝聚力，必须将坚持个人己见的人员排除于团队之外。
- 团队是 “自组织的”
 - 自适应团队结构
 - 使用Constantine 提出的随机、开放和同步式的范型
 - 相当大的自主权

22.2.5 团队协调与沟通

- 正式的，客观的方法 包括软件工程文档和工作产品（包括源代码），技术日志，项目里程碑，日程安排表和项目控制工具，变更要求和相关的文档，错误跟踪报告和存储库数据。
- 正式的交流机制 将目光集中在应用于软件工程项目产品的质量活动。这些活动包括状态审查会、设计和代码检查。
- 非正式的交流机制 包括信息传递的团队会议、问题解决和“需求的搭配和开发人员”。
- 电子交流 包括电子邮箱，电子公告板，相关地，还有视频会议系统。
- 人际网络 包括与团队成员进行非正式讨论，那些没有参与工程的有经验或见解的人可以协助团队成员。

- 范围
 - **项目环境。** 要开发的软件如何适应于大型的系统、产品或业务环境，该环境下要施加什么约束？
 - **信息目标。** 软件要产生哪些客户可见的数据对象（第8章）作为输出？需要什么数据对象作为输入？
 - **功能和性能。** 软件要执行什么功能才能将输入数据变换成输出数据？软件需要满足什么特殊的性能要求？
- 软件项目范围在管理层和技术层都必须是无歧义的和可理解的。

22.3.2 问题分解

- 有时称为 问题划分 或 问题细化
- 一旦范围被定义 ……
 - 它被分解成一系列功能
 - 它被分解成用户可见的数据对象
或
 - 它被分解成一系列问题类
- 分解过程一直进行直到所有功能或问题类都定义

22.4 过程

- 团队必须决定哪种过程模型最适合（人员、产品、环境）
- 一旦过程框架被建立
 - 考虑产品本身的特性
 - 确定产品严格要求的程度
 - 为每一个软件工程活动定义一个任务集
 - 任务集 =
 - 软件工程任务
 - 工作产品
 - 质量保证点
 - 里程碑

22.4.1 合并产品和过程

沟通 策划 建模 构造

公共过程
框架活动

- 软件工程任务
- 产品功能
- 文本输入
- 编辑与格式设计
- 自动复制编辑
- 页面布局能力
- 自动生成索引和目录
- 文件管理
- 文档生成

COMMON PROCESS FRAMEWORK ACTIVITIES	communication	planning	modeling	construction
Software Engineering Tools				
Product Functions				
Text input				
Editing and formatting				
Automatic copy edit				
Page layout capability				
Automatic indexing and TOC				
File management				
Document production				



22.4.2 过程分解

- 选定过程模型：项目团队可根据需要灵活地确定过程模型应包含的软件工程任务。

可根据项目规模，采用**线性顺序方法**（规模小且做过类似项目），**增量策略**（时间紧）或其他过程模型。

- 根据过程模型对过程框架做适应性修改。

虽然过程框架不变，但实际任务在完成规模和复杂性不同的框架活动时是不同的。同样的框架活动可分解为不同的任务序列。

22.5 项目

- 项目正处于危险状态的信号，当……
 - 软件人员不了解客户的需求。
 - 产品范围定义得很糟糕。
 - 没有很好地管理变更。
 - 所选的技术发生了变化。
 - 业务需求发生了变化 [或未能很好地定义]。
 - 截止日期是不切实际的。
 - 客户抵制。
 - 失去赞助 [或从来没有真正得到过赞助]。
 - 项目团队缺乏具有合适技能的人员。
 - 管理者 [和开发人员] 没有很好地利用已学到的最佳实践和经验。

90-90规则

- 系统前面90%的任务会花费所分配总工作量和时间的90%，系统后面10%的任务也会花费所分配总工作量和时间的90% [Zah94]
- 其根源就在于上述“信号”

- **在正确的基础上开始工作。** 首先努力(非常努力) 地正确理解要解决的问题，然后为每个参与项目的人员设置现实的目标和期望。
- **保持动力。** 为了维持动力，项目经理必须采取激励措施使人员变动量保持绝对最小，团队应该重视它完成的每项任务的质量，而高层管理应该尽可能不干涉团队的工作。
- **跟踪进展。** 对于软件项目而言，当工作产品（如模型、源代码、测试用例集）正在产生或被认可（通过技术评审）时，跟踪项目进展要作为质量保证活动的一部分。
- **做出英明的决策。** 总体上，项目经理和软件团队的决策应该 “保持项目的简单性”。
- **进行事后分析。** 建立统一的机制，从每个项目中获取可学习的经验。

22.6 W⁵HH原则

- **为什么 (Why)** 要开发这个系统?
商业目的是否有效? 为达到此目的是否值得花费这些人、时间和金钱?
- **将要做什么(What)?**
定义项目所需的任务集
- **什么时候(When)做?**
制定项目进度, 标识出何时开展项目任务以及何时到达里程碑。
- 某功能由**谁(Who)**负责?
规定软件团队每个成员的角色和责任。
- 他们的机构组织位于**何处(Where)?**
并非所有角色和责任均属于软件团队, 客户、用户和其他利益相关者也有责任。
- **如何(How)**完成技术工作和管理工作?
一旦确定了产品范围, 就必须定义项目的管理策略和技术策略。
- 每种资源需要**多少(how much)**
(例如, 人、软件、工具、数据库)? 通过估算。

22.7 关键实践

- 正规的风险管理
- 成本及进度的经验估算
- 基于度量的项目管理
- 获得价值跟踪
- 根据质量目标跟踪缺陷
- 人员计划管理