

第1章 软件项目管理概述

- 项目就是为了创造一个唯一的产品或提供一个唯一的服务而进行的临时性的努力
- 区分项目与日常运行

项目	日常运作
一次性的	重复性的
以目标为导向的	通过效率和有效性体现
通过项目经理及其团队工作完成的	职能式的线性管理
存在大量的变更管理	基本保持持续的连贯性

- 项目所具有的特征：目标性、相关性、临时性、独特性、资源约束性、不确定性
- 项目群可能不是项目
- 项目目标的实现受4个因素制约：项目范围、成本、进度计划和客户满意度
- 项目管理体系的10个知识领域：
 - 项目集成管理
 - 项目范围管理
 - 项目时间管理
 - 项目成本管理
 - 项目质量管理
 - 项目人力资源管理
 - 项目沟通管理
 - 项目风险管理
 - 项目采购管理
 - 项目干系人管理
- 项目管理的5个标准化过程组
 - 启动过程组
 - 计划过程组
 - 执行过程组
 - 控制过程组
 - 收尾过程组

第2章 项目确立

- 在立项阶段，产品负责人进行自造-购买(make or buy)决策，确定待开发产品的哪些部分应当采购、外包开发或自主研发
- 当项目外包的时候，就存在甲乙双方之间的责任和义务的关系，甲方即需方或买方，乙方即供方或卖方
- 甲（需）方在招投标阶段的主要任务是招标书定义、供方选择、合同签署
- 乙（供）方在招投标阶段的主要任务是进行项目选择；主要包含三个过程：项目分析、竞标、合同签署
- 招标书包括三部分内容：技术说明、商务说明和投标说明
- 项目建议书是乙方编写的投标书
- 项目章程是项目执行组织高层批准的一份以书面签署的确认项目存在的文件，包括对项目的确认、对项目经理的授权和项目目标的概述等。

- 项目经理要对项目实行全面管理，包括制定计划，报告项目进展，控制反馈，组建团队，进行决策和解决冲突等
- 项目经理的职责有开发计划、组织实施和项目控制

第3章 生存期模型

- 在软件开发中可能用到的主要生存期模型有瀑布模型、V模型、快速原型模型、增量式模型、渐进式阶段模型及敏捷模型等。
- 瀑布模型：它要求项目所有的活动都严格按照顺序自上而下执行，一个阶段的输出是下一阶段的输入，如同瀑布流水，逐级下落。
- 瀑布模型的适用范围：
 - 在项目开始前，项目的需求已经被很好地理解，也很明确，而且项目经理很熟悉为实现这已模型所需要的过程。
 - 解决方案在项目开始前也很明确。
 - 短期项目可以采用瀑布模型。
- V模型：强调测试的重要性，将开发活动与测试活动紧密地练习在一起。每一步都将比前一阶段进行更加完善的测试。
- V模型的适用范围：
 - 项目的需要在项目开始前很明确
 - 解决方案在项目开始前也很明确
 - 项目对系统的性能安全很严格
- 快速原型模型：以逐步增加的方式进行开发，以便于随时根据客户或最终用户的反馈来修正系统
- 快速原型模型的适用范围：
 - 项目的需求在项目开始前不明确
 - 需要减少项目的不确定性的时候
- 增量模型：假设需求可以分段，成为一系列增量产品，每一增量可以分别开发
- 增量模型的适用范围：
 - 进行已有产品升级或新版本升级，增量式模型是非常适合的
 - 对于完成期限要求严格的产品，可以使用增量式模型
 - 对于所开发的领域比较属性而且已有原型系统，增量式模型是非常适合的
 - 对于市场和用户把握不是很准，需要逐步了解的项目，增量式模型是非常适合的。
- 渐进式阶段模型：将大的项目分成几个小项目来做
- 渐进式阶段模型的适用范围：
 - 主要适用于中型或大型项目
- 敏捷生存期模型：用于在一个动态的环境中向干系人快速交付产品，其主要特点是关注持续的交付价值，通过迭代和快速的用户反馈管理不确定性和应对变更
- 敏捷开发宣言：
 - 个体和交互胜过过程和工具
 - 可以工作的软件胜过面面俱到的文档
 - 客户合作胜过合同谈判
 - 响应变化胜过遵循计划

第4章 软件项目范围计划——需求管理

- 软件需求是指用户对软件的功能和性能的要求，就是用户希望软件能做什么事情，完成什么样的功能，达到什么样的性能
- 需求管理包括需求获取、需求分析、需求规格编写、需求验证、需求变更5个过程
- 软件的需求具有模糊性、不确定性、变化性和主观性的特点
- 需求分析的任务是借助于当前系统的逻辑模型导出目标系统的逻辑模型，解决目标系统“做什么”的问题
- 需求分析完成的标志是提交一份完整的软件需求规格说明书（SRS）
- 需求建模的方法有：原型分析方法、结构化分析法、面向对象的用例分析法、功能列表法以及其他方法等
- 需求变更是软件项目一个突出的特点
- 结构化分析方法将现实世界描绘为数据在信息系统中的流动，以及在数据流动过程中数据向信息的转化，帮助开发人员定义系统需要做什么，系统需要存储和使用哪些数据，需要什么样的输入和输出，以及如何将这些功能结合在一起完成任务
- 数据流图（DFD）、数据字典（DD）、实体联系图（ERD）、系统流程图等都是结构化分析技术
 - 数据流图（DFD）作为结构化系统分析与设计的主要方法，是一种描述软件系统逻辑模型的图形符号，使用四种基本元素来描述系统的行为，即过程、实体、数据流和数据存储
 - 数据字典（DD）描述系统中设计的每个数据，是数据描述的集合，通常配合DFD使用，用来描述数据流图中出现的各种数据和加工，包括数据项、数据流、数据文件等
 - 实体联系图（ERD）用来描述系统需求要存储的数据信息
 - 系统流程图是一种表示操作顺序和信息流动过程的图表

第5章 软件项目范围计划——任务分解

- 任务分解是将一个项目分解为更多的工作细目或者子项目，使项目变得更小、更易管理、更易操作，这样可以提高估算成本、时间和
- 资源的准确性，使工作变得更易操作，责任分工更加明确
- WBS是最终得出项目的任务分解结构（Work Breakdown Structure）
- 任务分解的结果是WBS，WBS是面向可交付成果的对项目元素的分组，不包括在WBS中得工作不是该项目的工作，只有在WBS中得工作才是该项目的工作范围，它是一个分级的树形结构，是对项目由粗到细的分析过程
- WBS中的每一个具体细目通常指定唯一的编码（code of accounts）
- 工作包是WBS的最低层次的可交付成果，是WBS的最小元素，工作包应当由唯一主体负责，可以分配给另外一位项目经理通过子项目的方法完成
- 一般说，进行任务分解时，可以采用清单或者图表的形式来表达任务分解的结果
- 任务分解有很多具体方法，如模板参照、类比、自顶向下和自底向上等方法
- 如果WBS开发人员对项目比较熟悉或者对项目大局有把握，可以使用自顶向下方法
- 自顶向下方法从一般到特殊的方向进行，而自底向上是从特殊向一般方向进行的
- 如果对于项目人员来说，这个项目是一个崭新的项目，可以采用自底向上方法开发WBS
- WBS提供了项目范围极限，是范围变更的重要输入

第6章 软件项目成本计划

- 代码行（Lines Of Code, LOC）、功能点、人天、人月、人年等都是规模的单位；成本一般采用货币单位，如人民币和美元等
- 成本包括直接成本和间接成本。
 - 直接成本是与开发的具体项目直接相关的成本（人员的工资，材料费，外包外购成本）；
 - 间接成本不能归属于一个具体的项目，是企业的运营成本，可以分摊到各个项目中（房租、水电、员工福利、税收）
- 常用的成本估算方法：代码行估算法、功能点估算法、用力点估算法、类比（自顶向下）估算法，自下而上估算法、参数模型估算法、专家估算法、猜测估算法等
- 功能点分析把系统分为5类组件分别是
 - 外部输入（External Input, EI）
 - 外部输出（External Output, EO）
 - 外部查询（External Inquiry, EQ）
 - 内部逻辑文件（Internal Logical File, ILF）
 - 外部接口文件（External Interface File, EIF）
- 功能点计算公式是：FP = UFC × TCF，其中，UFC（Unadjusted Function Component）表示未调整功能点计数；TCF（Technical Complexity Factor）表示计数复杂度因子（即调整系数）
- 类比估算法是从项目的整体出发进行类推，通常在项目的初期或信息不足时采用此方法，如在合同期和市场招标时等。它的特点是简单易行，花费少，但是具有一定的局限性，准确性差，可能导致项目出现困难。
- 自上而下估算法是利用 WBS，对各个具体工作进行详细的成本估算，然后将结果累加起来得出项目总成本。
- 参数模型估算法，也成为算法模型或者经验导出模型，是一种使用项目特性参数建立数学模型来估算成本的方法，是通过大量的项目数据进行数学分析导出的模型，是一种统计技术。
- 专家估算法计算专家的平均值

$$E_i = \frac{(a_i + 4m_i + b_i)}{6}$$

- 其中 a_i 是最小值， m_i 是最可能值， b_i 是最大值
- 成本预算是将项目的总成本按照项目的进度分摊到各个工作单元中去。成本预算的目的是产生成本极限，他可以作为度量项目成本性能的基础。

第7章 软件项目进度计划

- 时间是一种特殊的资源，以其单向性，不可重复性、不可重复性、不可替代性而有别于其他资源。时间是项目规划中灵活性最小的因素
- 项目各项任务（活动）之间存在相互联系和相互依赖关系，可以根据这些关系安排各项活动的先后顺序
- 任务之间的关系主要有4种，如图。
 - 结束到开始：表示A任务（活动）在B任务（活动）开始前结束。
 - 开始到开始：表示A任务（活动）开始，B任务（活动）也可以开始，即A、B任务有相同的前置任务
 - 结束到结束：表示A任务（活动）结束，B任务（活动）也可以结束，即A、B任务有相同的后置任务

开始到结束：表示A任务（活动）结束，B任务（活动）也应该结束

- 任务间关联关系的依据主要有：强制性依赖关系、软逻辑关系、外部依赖关系
 - 强制性依赖关系：工作任务中固有的依赖关系，是一种不可违背的逻辑关系，又称硬逻辑关系或内在的相关性，它是由客观规律和物质条件的限制造成的。例如，需求分析一定要在软件设计之前完成，测试活动一定要在编码任务之后执行
 - 软逻辑关系：有项目管理人员确定的项目活动之间的关系，是人为的、主观的，是一种根据主观意志去调整和确定的项目活动的关系，也可称指定性相关，或者偏好相关或软相关。例如，安排计划的时候，哪个模块先做，哪个模块后做，哪个任务先做好一些，哪些任务同时做好一些，都可以由项目经理确定
 - 外部依赖关系：项目活动与非项目活动之间的依赖关系，例如，环境测试依赖于外部提供的环境设备等。
- 甘特图（Gantt）可以很容易看出一个任务的开始时间和结束时间，但是甘特图不能反映某一项任务的进度变化对整个项目的影响，不能明显地表示各项任务彼此间的依赖关系，也不能明显地表示关键路径和关键任务
- 网络图（network diagramming）是活动排序的一个输出，用于展示项目中的各个活动以及活动之间的逻辑关系，表明项目任务将如何和以什么顺序进行。常用的网络图有PDM网络图、ADM网络图、CDM网络图
 - PDM（Precedence Diagramming Method）网络图称优先图或者节点法或者单代号网络图。这个网络图中节点表示任务（活动），箭线表示各任务（活动）之间的逻辑关系
 - ADM（Arrow Diagramming Method）网络图也成为箭线法或者双代号网络图，箭线表示活动（任务），节点表示前一个任务的结果，同时表示后一个任务的开始
 - CDM（Conditional Diagramming Method）网络图也称为箭线图法，允许活动序列相互循环与反馈
- 里程碑图显示项目进展中的重大工作的完成情况，里程碑仅仅表示事件的标记，不消耗资源和时间
- 资源图可以用来显示项目进展过程中资源的分配情况，资源包括人力资源、设备资源等。
- 浮动为0且是网络图中的最长路径叫关键路径