1. 软件项目管理概述
2. 什么是项目

* 项目是为完成某一独特的产品或服务所做的临时性的努力
* 项目有一个独特的目的,Unique
* 项目是临时性的,Temporary
* 项目需要使用资源，资源的类型和来源一般会有很多种
* 项目有一个主要发起人或客户
* 项目一般要涉及一些人员，由这些人员完成一些相关联活动
* 项目含有不确定性

1. 项目与日常运作的区别
   * 项目是一次性的，日常运作是重复进行的
   * 项目是以目标为导向的，日常运作是通过效率和有效性体现的
   * 项目是通过与项目经理及其团队工作完成的，而日常运作是职能式的线性管理
   * 项目存在大量的变更管理，而日常运作基本保持持续的连贯性的
2. 项目管理三约束
   * 范围
     + 项目的任务？
     + 顾客或发起人要通过项目获取产品或服务是什么？
   * 时间
     + 完成该项目需要多长时间？
     + 项目进度如何安排？
   * 成本
     + 完成该项目需要的花费？
3. 软件项目目标的制约因素

* 衡量项目是否成功，应该看该项目是否在工程允许范围内按照成本预算和进度计划，生产出客户满意的产品
* **四因素：项目范围，成本，进度计划，客户满意度**
  + 项目范围是为使客户满意，必须做的所有工作
  + 项目成本就是完成项目所需要的费用。
  + 项目进度是安排每项任务的起止时间以及所需的资源等，是为项目描绘的一个过程蓝图
  + 项目目标就是在一定时间、预算内完成工作范围，以使客户满意。
  + 客户能否满意要看交付的成果质量，只有客户满意才能意味着可以更快地结束项目，否则会导致项目的拖延，从而增加额外的费用

1. 项目管理

* **项目管理**
  + 指在项目活动中运用*专门的知识*、技能、工具和方法，使项目能够实现或超过*项目干系人*(stakeholder)的需要和期望
  + 一定主体，为实现其目标，利用各种有效的手段，对执行中的项目周期的各阶段工作进行计划，组织，协调，指挥，控制，以取得良好经济效益的各项活动总和
* **项目干系人**
  + 指参与项目和受项目活动影响的人，包括项目发起人、项目组、协助人员、客户、使用者、供应商，甚至是项目的反对者

6、软件项目管理

* + 软件项目管理定义：为使软件项目能按预定成本，进度，质量要求顺利完成，而对成本，人员，进度，质量，风险等进行分析和管理的活动
  + 独特性
    - * 更多的是设计过程（非制造）
      * 需要使用人力资源
      * 产品是代码和技术文件，无其他物质结果
  + 有利于将开发人员的个人开发能力转化成企业的开发能力

7、项目管理知识体系

* + PMP(project management professional)：是美国项目管理学会(PMI)开发并负责组织实施的一种专业资格认证
  + 项目管理知识体系(Project Management Body Of Knowledge， PMBOK):
    - * 项目管理的知识框架
      * 9个知识领域，5个标准化过程组及39个模块

8、PMBOK九个知识领域

* **知识领域**
  + 指项目经理必须具备的一些重要的知识和能力
  + **四大核心知识领域**
    - 范围、时间、成本、质量
  + **四大辅助知识领域**
    - 人力资源、沟通、风险、采购
  + **项目整体管理**
* 项目核心知识领域
  + 项目范围管理
    - 确定和管理为完成项目所要做的全部工作
  + 项目时间管理
    - 包括项目所需时间的估算，制定可以接受的项目进度计划，并确保项目的及时完工
  + 项目成本管理
    - 包括项目预算的准备和管理工作
  + 项目质量管理
    - 是要确保项目满足明确约定的或各方默认的需要
* 项目管理辅助知识领域
  + 项目人力资源管理
    - 关心的是如何有效利用参与项目的人
  + 项目沟通管理
    - 包括产生、采集、发布和保存项目信息
  + 项目风险管理
    - 包括对项目相关的风险进行识别、分析和应对
  + 项目采购管理
    - 根据项目的需要从项目执行组织外部获取和购进产品和服务
* 项目整体管理
  + 要求发挥项目管理整体上的支撑作用，它与其他项目管理知识领域相互影响

9、PMBOK五个标准化过程组

10、项目管理的范围

* 五要素：技术，方法，团队建设，信息，沟通
* 项目经理主要工作是沟通
* 战术上，关注：产品规格(满足质量要求)，成本，进度
* 战略上，关注3P：人员，问题，过程

11、过程管理与项目管理的关系

* 软件过程的定义
  + 软件开发的 最佳实践
  + 包括：流程、技术、产品、活动间关系、角色、工具等，是软件开发过程的各个方面因素的有机结合
  + 需要进行过程定义，使过程成为企业内部可以被重用的共享资源
  + 要不断地进行改进、以不断地改善和规范过程，帮助企业提高生产力

12、软件项目管理过程

* + 项目启动
    - 确定项目的目标范围
    - 包括：开发合同、主要功能及这些功能的量化范围、项目开发的阶段周期等
  + 项目规划
    - 对软件项目的估算
    - 风险分析
    - 进度规划
    - 人员的选择与配备
    - 产品质量规划等
  + 项目跟踪控制
    - 测量世界的进程，并于计划进程相比较
  + 项目结束
    - 做出项目终止的决策
    - 确认项目实施的各项成果
    - 进行项目的交接和清算
    - 最后评审
    - 项目总结

1. 项目确立

1、项目立项

明确项目的目标、时间表、项目使用的资源和经费，而且得到执行该项目的项目经理和项目发起人的认可 .

2、项目章程

**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。**

第四章 软件项目范围计划——需求管理

1、软件项目需求管理

（1）、需求的定义

* + **软件需求是指用户对软件功能和性能的要求**
* **业务需求：组织机构或客户对系统，产品高层次的目标要求，由管理人员或市场分析人员确定**
* **用户需求：用户通过使用本软件产品必须完成的任务，用户协助提供**
* **功能需求：开发人员必须实现的软件功能**
* **软件需求规格：描述了软件系统应具有的外部行为，即系统展现给用户的行为和执行的操作**
* **约束：对开发人员在软件产品设计和构造上的限制**

（2）、需求的类型

* + **功能需求：系统必须执行的功能**
  + **非功能需求：对实际使用环境所做的要求，如性能要求，可靠性，安全性**
  + **非功能需求比功能需求要求更严格，更不易满足**

1. 需求管理过程

* **5个独立过程：获取，分析，规格，验证，变更**

（1）、需求开发的四个阶段、各阶段任务

①、需求获取

* + **就是将用户要求变成项目需求**
  + **主要任务：和用户方的领导层，业务层人员的访谈式沟通**
  + **目的：从宏观上把握用户的具体需求方向和趋势，了解现有的组织架构，业务流程，硬件环境，软件环境，现有运行系统**

②、需求分析

* + **也称需求建模，为最终用户所看到的系统建立一个概念模型**
  + **与获取的区别：使用模型描述，以获取用户更明确的需求**
    - **图形描述整体结构**
    - **原型，页面流或其他方式提供可视化界面**
    - **模型描述系统功能项，数据实体，外部实体，实体关系及状态转换**
  + **基本策略：头脑风暴，专家评审，焦点会议组等方式进行具体的流程细化，数据项的确认**
  + **需求分析难点：**
* **问题本身复杂**
* **交流障碍：不同背景，角度**
* **不完备和不一致性：陈述不完备；需求的矛盾**
* **需求易变性**

③、SRS

* + **目的：使用户和软件开发者双方对软件的初始规定有一个共同理解，使之成为整个开发工作的基础**
  + **需求规格说明书(SRS)**

④、验证和确认

* + **以需求规格说明为输入，通过符号执行、模拟或快速原型等途径**
  + **正确性**
  + **一致性**
  + **完整性**
  + **可行性：必须在已知系统和环境的权能和限制范围内可以实施**
  + **必要性**
  + **可检验性：能写出测试用例满足需求**
  + **可跟踪性：在每项软件需求与它的根源和设计元素、源代码、测试用例之间建立起链接**
  + **最后的签字**

（2）、需求管理

* **必要性：**
  + **需求供求双方固有的矛盾**
  + **需求具有易变性和难以表述性**
  + **需求错误出现的高频性和修复的高昂成本**
* **目标**
  + **使软件需求受控，并建立供软件工程和管理使用的需求基线**
  + **使软件计划、产品和活动与软件需求保持一致**
* **原则**
  + **需求一定要分类管理**
    - **分出层次，不同层次需求的侧重点、描述方式不同。领导、管理人员、业务人员**
  + **需求必须分优先级**
  + **需求必须文档化**
  + **需求一旦变化，就必须对需求变更的影响进行评估**
  + **需求管理必须与需求工程的其他活动紧密整合**

（3）、需求管理过程各活动及各自任务

|  |  |
| --- | --- |
| **需求管理活动** | **活动的任务** |
| **变更控制** | **建议需求变更并分析其影响，做出是否变更的决策** |
| **版本控制** | **确定单个需求和SRS版本** |
| **需求跟踪** | **定义对于其他需求及系统元素的联系链** |
| **需求状态** | **定义并跟踪需求的状态** |

（3）、需求变更控制过程

第五章 软件项目范围计划——任务分解

1、任务分解

（1）、定义

* **将项目分解为更多的工作细目和子项目，使项目更小，更易管理，易操作**

（2）、目的

* **提高估算成本，时间和资源的准确性**
* **是对需求的进一步细化，是最后确定项目所有任务范围的过程**
* **结果是任务分解结构(Work breakdown structure, WBS)**

（3）、意义

* **WBS是一个分级的树型结果，是对项目由粗到细的分解过程**
* **WBS提供了项目范围基线，是范围变更的重要输入**
* **WBS为开发项目提供了一个实施框架，为评估和分配任务提供具体的工作包，是进行估算和编制项目进度的基础**

2、WBS

（1）、定义

* WBS(Work Breakdown Structure)是将一个项目分解成易于管理的几个部分或几个细目，以便确保找出完成项目工作范围所需的所有工作要素
* WBS按照项目发展的规律，依据一定的原则和规定，进行系统化的、相互关联和协调的层次分解
* 结构层次越往下层则项目组成部分的定义越详细，WBS最后构成一份层次清晰，可以具体作为组织项目实施的工作依据
* ＷＢＳ是管理项目范围的基础，详细描述了项目所要完成的工作

（2）、如何理解WBS

* **WBS是用来确定项目范围的，必须包括项目的所有工作**
* **WBS是把所有项目工作逐层分解到较小的、便于管理的要素（可交付成果）** 
  1. **ＷＢＳ的最低层元素是能够被评估的，可以安排进度的和被追踪的**
* **没有包含在WBS中的工作就不是项目范围内的工作，都不要做。如果要做，必须通过范围控制过程。**
* **WBS的编制需要“全员参与”，项目经理主要发挥“整合者”的作用**

（2）、表现形式

树状图、气泡图、列表法

3、如何理解工作包

* **工作包（Work Package），是WBS中最底层的项目可交付成果。工作包可以再细分为工作活动。**
* **项目经理通常只负责到工作包的层次，负责该工作包的团队成员负责把工作包细分为作活动。**
* **一个工作包必须由一个人或者一个部门来负责，而不能由两个或两个以上的人或部门来负责。**
* **工作包单元的周期应该是最短周期**
* **应明确本工作包与其他工作包之间的关系**

4、如何理解WBS字典

* **WBS词典，是在制定WBS过程中生成，并与WBS配合使用的说明性文件。包括：**
  1. **每个WBS要素，列出帐户编号、工作说明书、负责的组织、以及进度里程碑清单。**
  2. **WBS要素的其他相关信息，如合同要求、质量要求、以及技术参考文献。**
  3. **控制帐户的其他相关信息，如收费编号。**
  4. **工作包的其他相关信息，如计划活动安排表、所需资源、以及费用估算清单。**

第六章 软件项目成本计划

1、估算的概念

* **在项目初期要对项目规模，成本和进度进行估算，且是同步进行的**
* **进度估算是从时间角度；成本估算是从费用角度**
* **项目进行过程中，会有更多新的信息，需不断重复进行估算**
* **在项目不同阶段可采用不同估算方法，开始可能很粗糙，随着项目进展，会逐步精细**
* **对于估算，没有特效的方法和通用模型，项目经理可根据以前的经验和验证过的指南来提高精度**

2、成本估算方法

（1）、代码行估算法

（2）、功能点估算法

（3）、用例点估算法

（4）、类比估算法

（5）、自下而上估算法

（6）、参数模型估算法

第七章 软件项目进度计划

1、制定项目计划

* **软件项目计划的要素**
  + **目标**
  + **合理的概念设计**
  + **工作分解结构**
  + **规模估计**
  + **工作量估计**
  + **项目进度安排**
* **项目计划还为管理者提供了根据计划定期评审和跟踪项目进展的基础**

2、项目进度定义

* **按时完成项目是对项目经理最大的挑战**
* **进度**
  + **对执行的活动和里程碑制定的工作计划日期表**
  + **进度的作用**
  + **决定是否达到预期目的，是跟踪和沟通项目进展状态的依据**
  + **是跟踪变更对项目影响的依据**
* **进度安排有两种前提:**
  + **1 交付日期确定后安排计划**
  + **2 使用资源确定后安排计划**

1. 进度管理图示

（1）、甘特图

* **可显示任务的基本信息,工期,开始,结束时间以及资源信息**
* **棒状图:表示任务的起止时间**
* **三角形:三角向上表示开始;向下表示结束**

（2）、网络图

PDM网络图

* 构成PDM网络图的基本特点是节点(Box)
* 节点(Box)表示活动(工序,工作)
* 用箭线表示各活动(工序,工作)之间的逻辑关系.
* 可以方便的表示活动之间的各种逻辑关系。
* 在软件项目中PDM比ADM更通用

（3）、里程碑图

* **显示项目进展中的重大工作完成**
* **不同于活动:活动需要消耗资源且需要时间完成；里程碑表示事件的标志，不消耗资源和时间**
* **对里程碑阶段点的设置必须符合实际，具有挑战性和可达性**
* **进度管理与控制其实就是确保项目里程碑的达到**

（4）、资源图

* **显示项目进展过程中资源的分配情况**
* **资源包括人力，设备等**
* **安排人力时一定合理，管理复杂度和效率协调**

4、项目进度估算

* **估计任务的持续时间，是项目计划的基础工作**
* **首先对项目中的活动进行时间估计**
* **常用估算方法：**
  + **基于规模的进度估算**
  + **工程评价技术**
  + **关键路径法**
  + **专家估算方法**
  + **类推估算**
  + **进度表估算**
  + **基于承诺的进度估算**

关键路径法

* **基本概念**
  + **最早开始时间(ES:early start)：最早可开始时间**
  + **最晚开始时间(LS)：最晚可开始时间**
  + **最早完成时间(EF)**
  + **最晚完成时间(LF)**
  + **超前(Lead)：两个任务的逻辑关系所允许的提前后置任务的时间，它是网络图中活动的固定可提前时间**
  + **滞后(Lag)：两个任务的逻辑关系所允许的推迟后置任务的时间**
  + **浮动时间(Float): 一个活动在不影响项目完成的情况下可以延迟的时间**
    - **自由浮动（Free Float）:是在不影响后置任务最早开始时间的情况下本任务可以推迟的时间**
    - **总浮动（Total Float）:是在不影响项目最早完成时间的情况下本任务可以推迟的时间。**
  + **关键路径：浮动为0且是网络图中的最长路径，其上的任何活动延迟都会导致整个项目的延迟；是完成项目的最短时间量**

第九章 软件配置管理计划

1、配置管理定义

* **软件配置管理SCM，通过一套工程规范，在整个软件生命周期中跟踪、记录软件，保证全部变更都记录在案，并保证软件的当前状态是已知的和可重复的。**
* **配置项（Configuration Item）**

**配置项是处于配置管理之下的软件或硬件的集合体。这个集合体在配置管理过程中作为一个实体出现。**

* + **例如：项目计划，软件配置管理计划，设计文档，源代码，测试数据，项目数据，用户手册，等等。**
* **基线（Baseline）**

**已经通过正式复审和批准的某规约或产品，它因此可以作为进一步开发的基础，并且只能通过正式的变更控制过程来改变**

* **配置数据库（软件制品基线库）**
  + **项目建立和访问软件制品库，这个制品库主要用来对保存配置项和一些与软件配置管理相关的记录。**
  + **目前比较好的配置管理工具：Clearcase (Rational), Notes/Domino (Lotus), PVCS (Merant) and VSS (Microsoft).**
* **配置数据库的内容**
  + **包括软件产品及其开发过程中的所有的制品**

2、配置管理作用

4、配置管理的目标

（1）、版本控制

* **版本控制是所有配置管理系统的核心功能，是实现团队并行开发、提高开发效率的基础。**
* **版本控制的对象是软件开发过程中涉及的所有文件系统对象，包括文件、目录和链接。**
* **版本控制的目的在于对软件开发进程中文件或目录的发展过程提供有效的追踪手段，保证在需要时可回到旧的版本，避免文件的丢失、修改的丢失和相互覆盖。**

（2）、工作空间管理

* **所谓工作空间，就是为了完成特定的开发任务（如开发新功能、进行软件测试、或修复BUG，等等），从版本库中选择一组正确的文件/目录的正确版本拷贝到开发人员的开发环境。**
* **存在两类工作空间，一类是开发人员的私有空间，在私有空间中，开发人员可以相对独立地编写和测试自己的代码。**
* **另一类工作空间是团队共享的集成空间，该空间用于集成所有开发人员的开发成果。**
* **工作空间管理包括工作空间的创建、维护与更新、删除等**
* **工作空间应具备以下特点：稳定、一致、透明**

（3）、并行开发支持

* **为实现并行开发，配置管理系统需要提供灵活的分支机制和工作空间管理。**
* **创建分支的过程实际上就是一个建立副本的过程**
  + **针对每个发布分别建立相应的分支，分支之间具备相对的独立性，这样不同的发布就可以在各自的分支上并行进行开发**
  + **在适当的时候，分支之间可以进行合并，从而实现将Release 1 中后期开发的功能合并到Release 2 中**

（4）、过程控制

* **配置管理系统应该能够支持灵活的配置管理策略和配置管理流程，并实现过程自动化以提高配置管理效率**

（5）、异地开发支持

* **经济全球化趋势、企业兼并以及软件规模和复杂性的增加等因素使地理分布的多个软件开发团队进行协作开发的开发模式日益普遍。**
* **配置管理系统应该能够有效支持地理上分布的团队之间的远程协作。**
* **由于广域网带宽的限制，单一的版本库模式被证明是低效和不实用的**
* **典型的异地开发模式在每个地点都有版本库的副本，日常开发针对本地版本库进行，版本库之间有复制和同步机制。**

（6）、变更请求管理

* **变更请求管理记录、跟踪和报告软件系统的任何变更**
* **其核心是一个适合软件开发组织的变更处理流程**
* **典型的变更处理流程涉及：如何提交变更请求，如何对变更请求进行复审以便决定是否实施，由谁实施，如何实施，如何确定变更请求准确实施完成等方面**
* **典型的变更请求有：新的功能需求、对已有功能的优化和改进、针对发现的缺陷的修复等。**

（后续见课本）