

课程目标：

了解软件项目管理的过程。

了解软件生命周期的每个过程。

# 软件项目管理定义

## 【项目与软件项目】

### 项目的定义

下列哪些是项目？

* 吃饭
* 聚餐
* 集体婚礼
* 举办奥运会
* 开发OA系统
* 生产线产品制造

按项目与日常动作分类，如下：

吃饭

聚餐

集体婚礼

举办奥运会

开发OA系统

生产线产品制造

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，而日常运作则基本保持连贯性的

项目的特征：

* 有明确的目标性
* 明确的时限性
* 资源成本的约束性
* 项目的不确定性
* 唯一性(一次性)

项目的定义：

**项目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活动。**



### 软件项目

软件项目除了具有项目的特征外还具有软件项目的特殊性：

* 抽象性：软件是逻辑实体，不是具体的物理实体，具有抽象性，软件由程序员编写计算机能够识别的代码，由计算机硬件去执行。
* 复杂性：软件的开发受计算机系统的限制，开发语言众多，软件对硬件有不同程度的依赖，软件系统之间存在交互。
* 成本高：软件开发成本高，软件主要依靠人的脑力劳动，考虑人力资源成本；软件的运行依赖计算机硬件，考虑硬件资源成本，软件的运行依赖第三方中间件，考虑软件成本；软件的运行依赖后期维护，考虑维护成本。
* 变更大：软件项目变更大，前期需求变更，后期需求变更，需求变更会引发计划、实施、维护等变更。
* 渐近明细：软件项目开发过程通常包括需求分析、概要设计 、详细设计、编码、测试几个过程，开发一个符合用户需求的功能都需要经过这几个过程。

软件项目实例：

人力资源管理系统：

目标：实现公司人员招聘、公司人员离职、公司人员档案管理、公司人才库四大功能

时间：半年

资金：50万

资源：6人、设备、资料等等

B to C商城系统：

目标：实现在前台在线销售平台（门户、商品搜索、商品展示、购物车、订单管理、会员中心、客户服务）、后台管理平台（商品管理、订单管理、物流管理、财务管理、统计分析）两个功能

时间：半年

资金：500万

资源：20人、设备、资料等等

## 【项目管理与软件项目管理】

### 项目管理的定义

项目管理的重要性：

* 据有关统计，全球的项目投资高达十万亿美元，其中在美国有2.5万亿美元，占GDP25%，全球从事项目管理人员大约有11650多万人，其中美国有450多万人。中国未来5年内在信息化建设的投资有上万亿人民币。
* 软件项目越来越复杂（需求、技术、用户关系、后期维护等），很多项目都失败了，导致失败的根本原因 是缺乏有效的项目管理；缺乏合格的软件项目经理。
* 企业成功在于有效的推行项目管理。

项目管理的通俗理解：

假设我们要做一件事情，有一定的约束和目标要求，诸如时间、资金、人力等条件限制，那么如何在这些约束条件下有效地达到我们预想的目标，通过相关的理念、技术方法和工具进行管理的过程就是项目管理。

项目管理的定义：

**使项目能够按照预定的成本、进度、质量顺利完成并让所有干系人得到满意，而对成本、人员、进度、质量、风险等进行分析和管理的活动。**

### 软件项目管理

软件项目管理是为了使软件项目能够按照预定的成本、进度、质量顺利完成，而对成本、人员、进度、质量、风险等进行分析和管理的活动。

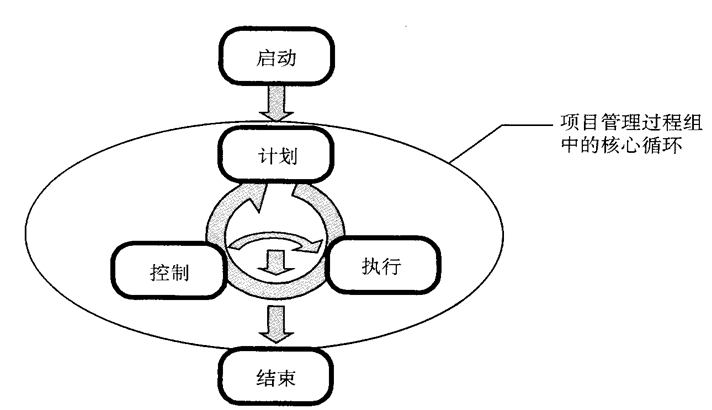
软件项目管理的根本目的是为了让软件项目尤其是大型项目的整个软件生命周期（从分析、设计、编码到测试、维护全过程）都能在管理者的控制之下，以预定成本按期，按质的完成软件交付用户使用。而研究软件项目管理为了从已有的成功或失败的案例中总结出能够指导今后开发的通用原则，方法，同时避免前人的失误。

软件项目管理和其他的项目管理相比有相当的特殊性：

* 首先，软件是纯知识产品，其开发进度和质量很难估计和度量，生产效率也难以预测和保证。
* 其次，软件系统的复杂性也导致了开发过程中各种风险的难以预见和控制。Windows这样的操作系统有1500万行以上的代码，同时有数千个程序员在进行开发，项目经理都有上百个。这样庞大的系统如果没有很好的管理，其软件质量是难以想象的。

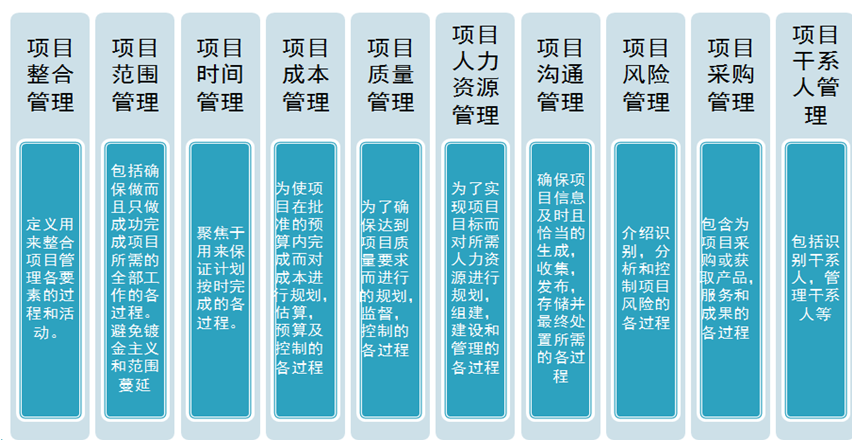
## 【项目管理过程】

### 五大标准化过程



* 启动阶段  
  项目的可行性分析、立项、招投标、合同签署。
* 计划阶段  
  范围定义、进度安排、资源计划、成本估计、质量保证计划、风险计划、实施计划等。
* 实施及控制阶段  
  项目实施、进度控制、费用控制、质量控制、变更控制等。
* 结束阶段  
  范围确认、质量验收、费用结算与审计、项目资料验收、项目交接与清算、项目审计与评估、项目总结等。

### 项目管理的知识领域



### 项目管理47个过程



## 练习

* 项目：同学聚会，比如十一聚会
* 目标：利用放假时间沟通同学感情
* 人数：?? 比如：30人至40人
* 成本：?? 比如：100元\*(30至40)
* 时间：?? 比如：十月一日至十月七日
* 地点：?? 比如：北京
* 如果你是本次活动的项目经理，你如何安排？

# 项目启动

项目启动阶段又称为项目立项阶段，在启动阶段需要进行项目可行性分析、任命项目经理、项目目标等。

## 【项目经理】

项目启动阶段首先要确定一名项目经理，由项目经理负责项目管理工作。在项目启动阶段所需要准备的一些分析报告等就需要项目经理组织、分析、编写文档。

### 项目经理的定位

项目经理( Project Manager ) ，从职业角度，是指企业建立以项目经理责任制为核心，对项目实行质量、安全、进度、成本管理的责任保证体系和全面提高项目管理水平设立的重要管理岗位。

**项目经理是为项目的成功策划和执行负总责的人**。**项目经理是项目团队的领导者**，项目经理首要职责是在预算范围内按时优质地领导项目小组完成全部项目工作内容，并使客户满意。为此项目经理必须 在一系列的项目计划、组织和控制活动中做好领导工作，从而实现项目目标。

### 项目经理能力要求

* 计划能力：表现在对工作的策划上，要能分辩工作的主次，项目WBS本身是一项需要计划性很强的工作，需要对工作量，工作人员分配工作时间评估做一个相对准确的计划。有了这些计划作为基标你才能完成预期的工作目标统筹。
* 执行力：最关键的能力，除了项目经理个人,整个团队执行力也必须强，否则再好的战略没有执行也是零
* 沟通能力： 沟通能力对于项目经理很关键, 软件项目需求和变更很多，需要项目经理和客户建立良好的沟通，控制范围，否则很容易失败
* 组织能力：项目的干系人很多，项目经理要有很强的组织能力迅速将相关干系人组织起来完成工作。
* 协调能力：任何公司资源都是有限的，要有较强的协调强力能获取到项目所需的资源和支持;
* 谈判能力: 管理客户期望，必须通过谈判. 软件项目销售时往往销售人员给客户很高的期望，实施中要通过管理期望,否则很难成功实施项目，令客户满意。
* 风险处理能力：如果需求变了，如果出现人员离职等，这些变化都需要考虑进来，我们把这些变化称为项目的风险，为了保证项目的顺利进行，需要对项目的风险进行识别，对风险进行有效控制、处理。

### 项目经理岗位职责

项目经理在项目启动、计划、实施、控制、结束都有相应的职责。

参考“软件项目经理岗位职责.doc”

## 【初始项目分析】

### 项目类型

* 合同项目   
  招投标、合同谈判、合同签署，甲乙双方有合同约束。   
  参考项目文档“XXXXX合同书”
* 内部项目   
  确定任务范围和相关各人员进行有效地配合，无合同约束。

### 可行性分析

针对项目的市场前景、项目中所用到的技术分析项目的可行性，对分析结果进行认证讨论。  
成果：项目可行性分析报告

### 项目范围分析

分析项目的范围、确定项目的目标，注意启动阶段范围分析其目的是确定项目目标，比项目计划阶段要粗，比如：一个电商网站项目其目标就是完成电商网站的开发、上线，至于完成开发、上线工作的细化是项目计划阶段要做 的。

### 项目干系人分析

分析确定项目相关人员，包括：项目发起人、项目投资人、项目开发人员、测试人员、维护人员、客户等。

## 【项目生存期模型】

项目启动阶段要确定项目生存期模型，常用生存期模型包括 ：瀑布模型、原型模型、增量模型等。

### 软件生命周期

软件生命周期又称为软件生存周期或系统开发生命周期，是指从软件的产生直到报废的整个过程，它包括问题定义、可行性分析、总体描述、系统设计、编码、调试和测试、验收与运行、维护升级到废弃等阶段。每一个阶段都有确定的任务，并产生一定规格的文档（资料），提交给下一个周期作为继续工作的依据。

软件生命周期的各各阶段：

1、问题定义：用户需要计算机解决什么问题

2、可行性研究：解决这些问题是否可行

3、需求分析：问题细化

4、设计阶段：确定解决这些细化的问题的方法

5、编码阶段：去解决这些细化的问题

6、测试阶段：验证问题是否解决

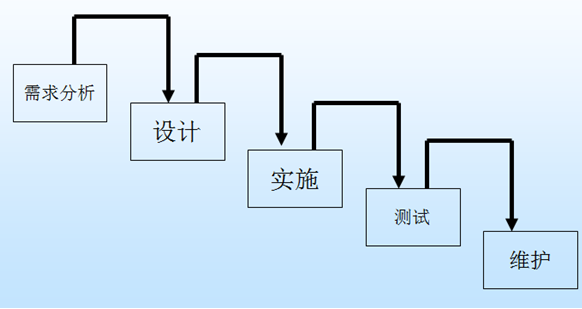
7、验收阶段：开发测试完成由甲方验收、并完成上线

8、维护阶段：包括改正性维护、完善性维护

预防性维护。

### 瀑布模型

瀑布模型（Waterfall Model） 是软件开发的各各阶段是顺序进行，从系统需求分析开始直到产品发布和维护，每个阶段都会产生循环反馈，因此，如果有信息未被覆盖或者发现了问题，那么最好 “返回”上一个阶段并进行适当的修改，项目开发进程从一个阶段“流动”到下一个阶段，这也是瀑布模型名称的由来。



适用场景：

项目的需求很明确

解决方案也很明确

类似的项目如：

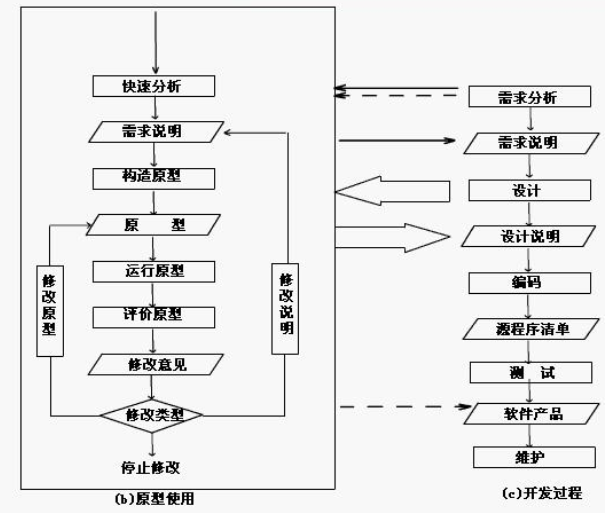
库存管理系统

短期项目

### 原型模型

原型模型即样品模型，先借用已有系统作为原型模型，通过“样品”不断改进，使得最后的产品就是用户所需要的。

原型模型采用逐步求精的方法完善原型，使得原型能够“快速”开发，避免了像瀑布模型一样在冗长的开发过程中难以对用户的反馈作出快速的响应。



适用场景：

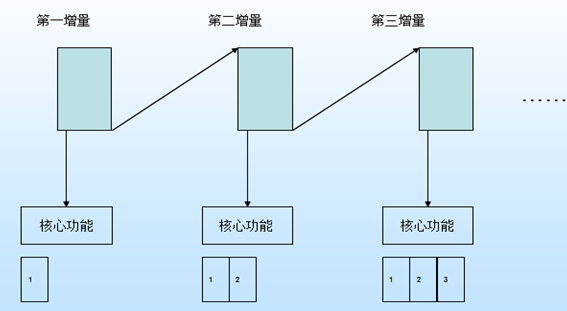
在项目开始前,项目的需求不明确，需要减少项目需求的不确定性

类似的项目如：

互联网项目、新领域的项目

### 增量模型

增量模型融合了瀑布模型的基本成分（重复应用）和原型实现的迭代特征，该模型采用随着日程时间的进展而交错的线性序列，每一个线性序列产生软件的一个可发布的“增量”。当使用增量模型时，第1个增量往往是核心的产品，即第1个增量实现了基本的需求，但很多补充的特征还没有发布。客户对每一个增量的使用和评估都作为下一个增量发布的新特征和功能，这个过程在每一个增量发布后不断重复，直到产生了最终的完善产品。



适用场景：

项目开始，明确了需求的一部分，但是需求可能会发生变化

对于市场和用户把握不是很准，需要逐步了解

对于有庞大和复杂功能的系统进行功能改进，就需要一步一步实施的。

## 【项目立项】

### 项目章程

将项目启动阶段的工作内容形成统一的文档，文档没有统一的格式根据公司实际管理水平制定。

参考：“软件项目章程模版.doc”

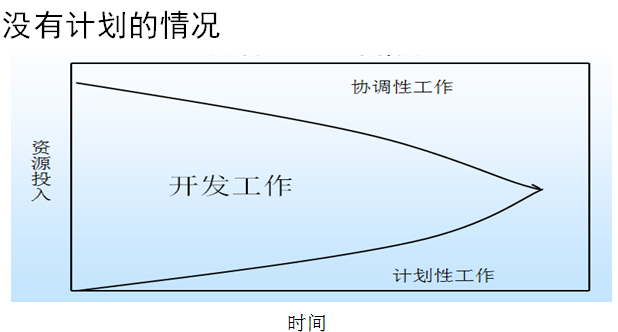
### 立项会

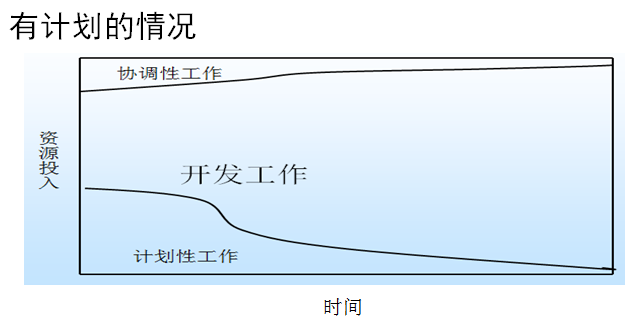
通常由公司PMO(项目管理办公室)组织立项会，对项目的可行性分析、范围目标、项目经理认命等进行评审，确定。

# 项目计划

## 计划的重要性

* 项目成功的三大要素（法宝）：计划、计划、计划
* 计划是通向项目成功的路线图
* 进度计划是最重要的计划





## 【范围计划】

### 什么是范围计划

项目范围计划也就是对项目应该包括什么和不应该包括什么进行相应的定义，这需要经过问题定义、需求分析、任务分解确定项目的范围。

### 【WBS任务分解】

#### 什么是WBS

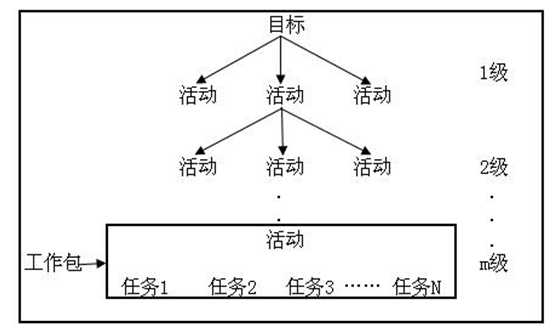
工作分解结构（Work Breakdown Structure，简称WBS）就是把一个项目按一定的原则分解，项目分解成任务，任务再分解成一项项工作，再把一项项工作分配到每个人的日常活动中，直到分解不下去为止。

　　即：项目→任务→工作→日常活动

　　工作分解结构以可交付成果为导向对项目要素进行的分组，它归纳和定义了项目的整个工作范围，每下降一层代表对项目工作的更详细定义。

　　WBS总是处于计划过程的中心，也是制定进度计划、资源需求、成本预算、风险管理计划和采购计划等的重要基础。

#### WBS分解结构



WBS的最低层次的项目可交付成果称为工作包(Work Package)，具有以下特点：

•工作包可以分配给另一位项目经理进行计划和执行。

•工作包可以通过子项目的方式进一步分解为子项目的WBS。

•工作包可以在制定项目进度计划时，进一步分解为活动。

•工作包可以由惟一的一个部门或承包商负责。用于在组织之外分包时，称为委托包(Commitment Package)。

　　工作包的定义应考虑80小时法则(80-HourRule)或两周法则(Two Week Rule)，即任何工作包的完成时间应当不超过80小时。在每个80小时或少于80小时结束时，只报告该工作包是否完成。通过这种定期检查的方法，可以控制项目的变化。

#### 分解标准

WBS任务分解可以按照软件生命周期分解、系统功能分解，但是分解的标准要采用一种，如下：

b/s电商系统：

按照生命期分解

规划

需求

前台购买平台

后台管理平台

....

设计

编码

测试

提交

按照产品组成分解

 前台购买平台

需求

设计

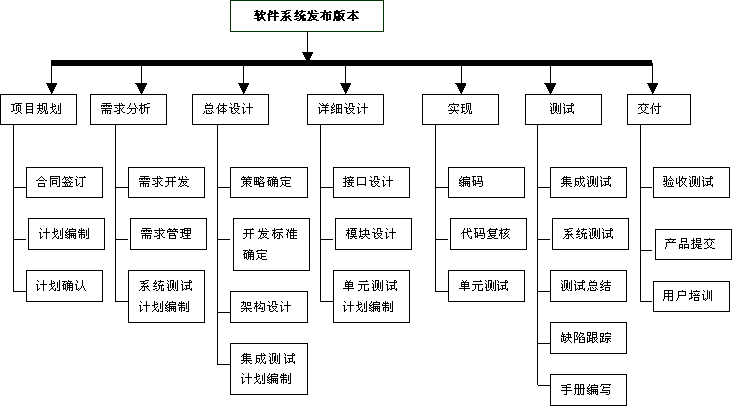
编码

。。。

后台管理平台

企业实际开发通常是按照按照生命期分解。

参考模板如下：

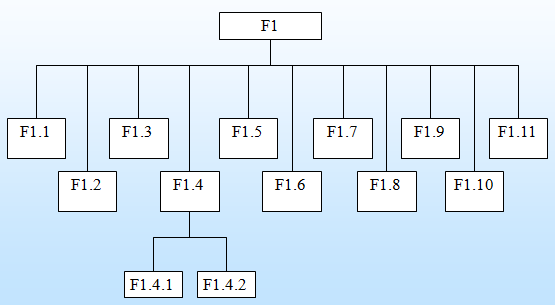


#### 检验分解结果的标准

* 最底层的要素是否是实现目标的充分必要条件
* 最底层要素是否有重复的
* 每个要素是否清晰完整定义
* 最底层要素是否有定义清晰的责任人,是否可以进行成本估算和进度安排

#### WBS字典内容

#### 案例1 XXXX系统功能分解



#### 案例2野餐准备活动任务分解

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 任务 | 持续时间 | 工作人员 |
| 1 | 开始 |  |  |
| 2 | 做冰茶 |  |  |
| 3 | 准备三明治 |  |  |
| 4 | 准备水果 |  |  |
| 5 | 准备篮子 |  |  |
| 6 | 收拾毛毯 |  |  |
| 7 | 收拾运动服 |  |  |
| 8 | 装车 |  |  |
| 9 | 加油 |  |  |
| 10 | 开车去野餐营地 |  |  |
| 11 | 结束 |  |  |

#### 练习

你是某个项目的项目经理,这个项目是为某高校创建一个学籍管理系统。

请提交任务分解结果WBS。

## 【进度计划】

### 什么是进度计划

进度计划是对执行的活动和里程碑制定的工作计划日期表。

进度计划的重要性：

* 为了确保项目按期完成所需要的过程
* 按时完成项目是项目经理最大的挑战之一
* 时间是项目规划中灵活性最小的因素
* 进度问题是项目冲突的主要原因，尤其在项目的后期。

### 进度计划管理过程

* 活动定义

确定为完成项目的各个交付成果所必须进行的诸项具体活动

* 活动排序

确定任务依赖、前置任务、里程碑（里程碑显示项目进展中的重大工作完成）

* 活动历时估计

每个任务的历时估计、项目总历时估计，可采用定额算法、经验算法。

* 任务资源估计

每个任务需要的资源类型和数量有一定的考虑，这些资源包括，人力资源，设备资源，以及其它资料资源

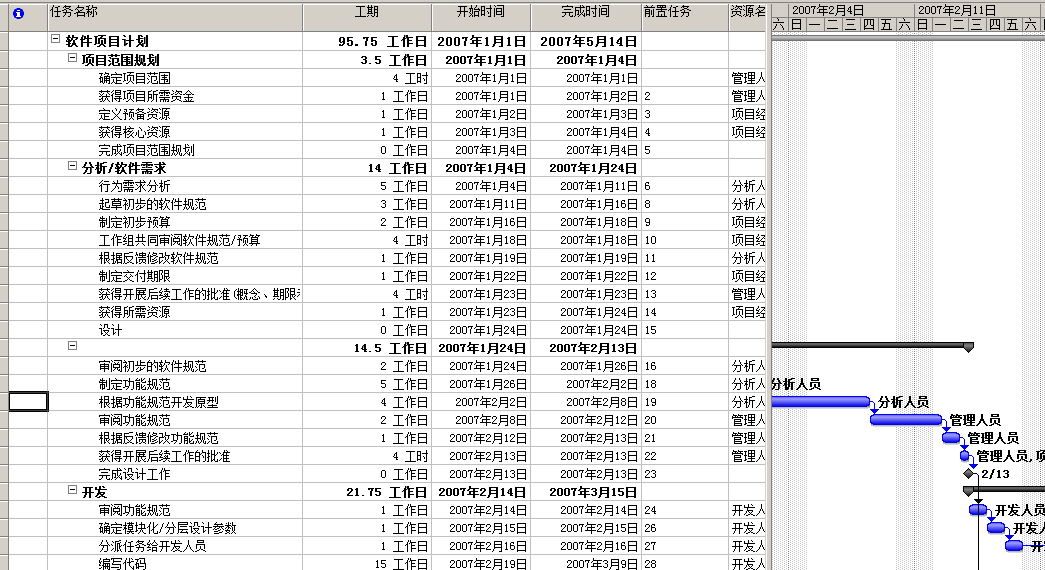
### 甘特图

显示基本的任务信息

可以查看任务的工期、开始时间和结束时间以及资源的信息。

只有时标，没有活动的逻辑关系

参考：软件项目计划.mpp



### 关键路径与里程碑

* 关键路径

关键路径是项目计划中最长的路线。它决定了项目的总实耗时间。项目经理必须把注意力集中于那些优先等级最高的任务，确保它们准时完成，关键路径上的任何活动的推迟将使整个项目推迟。向关键路径要时间，向非关键路径要资源。所以在进行项目操作的时候确定关键路径并进行有效的管理是至关重要的。

* 里程碑

在进度时间表上设立一些重要的时间检查点，这样一来，就可以在项目执行过程中利用这些重要的时间检查点来对项目的进度进行检查和控制。这些重要的时间检查点被称作项目的里程碑

## 【成本计划】

### 资源管理计划编制

确定项目需要的资源种类和数量。

参考项目计划书

### 成本估算

编制一个为完成项目各活动所需要的资源成本的近似估算。

* 估算不是很准确的，有误差的
* 经验（历史）数据非常重要
* 不要太迷信数学模型

#### 软件项目规模

软件项目规模即工作量，是从软件项目范围中抽出的软件功能，然后确定每个软件功能所必须执行的一系列软件工程任务包括：软件规划，软件管理，需求，设计，编码，测试，以及后期的维护等任务。

#### 规模的单位

* LOC(Loc of Code)

源代码程序长度的测量，单位K代码行

* FP(Function Point)

用系统的功能数量来测量

* 人月
* 人天
* 人年

#### 软件项目成本

* 项目资源所需要付出的资金
* 完成软件规模相应付出的代价。
* 搞清楚直接成本和间接成本
* 直接成本

与具体项目相关的成本

* 间接成本

不能具体到某个项目中的成本，可以分摊到各个具体项目中的成本，例如：培训、房租水电、员工福利、市场费用、管理费、其他等等

#### 成本的单位

* 货币单位

人民币元

美元

## 【质量计划】

### 什么是质量计划

**质量的多种定义**

* 符合目的或者用途
* 用户的感觉就是质量
* 符合顾客在其合理价格下对产品的要求
* 产品或者服务满足明确和隐含需要能力的性能特性的总体

质量的定义：

质量是满足要求的程度,包括符合规定的要求和满足顾客的需求.

软件质量：是软件满足明确说明或者隐含的需求的程度

明确说明：查询功能

隐含说明：查询速度

质量计划就是：

* 确定项目应达到的质量标准
* 决定如何满足质量标准的计划安排和方法

### 质量保证(QA)

* 确定项目应达到的质量标准
* 决定如何满足质量标准的计划安排和方法
* 通过评价项目整体绩效,建立对质量要求的信任
* 提供项目和产品可视化的管理报告
* 例如：《总体设计规格》质量审计
* 这个任务本身并不能提高产品的质量
* 一般由质量保证部门人员编写质量保证计划，根据计划实施

### 质量活动

* 静态分析

不实际运行程序，而是通过检查和阅读等手段来发现错误并评估代码质量的软件测试技术。也称为静态测试技术。

静态分析对象：

* 软件需求规格
* 软件设计规格
* 测试计划
* 用户手册
* 源代码

。。。。

* 动态测试(Test)

单元测试

集成测试

系统测试

性能测试

* 缺陷追踪

测试人员测试出bug由开发人员修改，修改后测试人员进行验证

## 【人力资源计划】

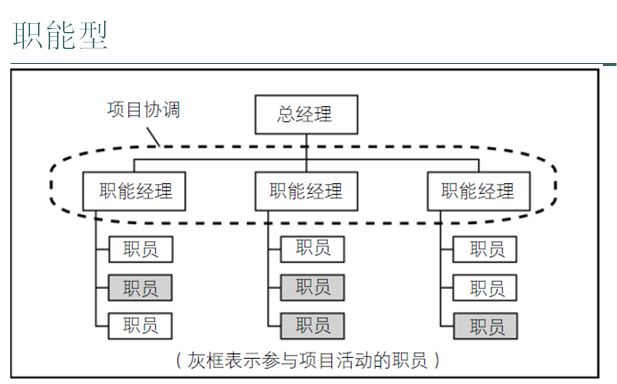
### 项目团队

* 团队是一定数量的个体成员组织的集合
* 团体包括自己组织的人、供应商、分包商、客户等
* 团体为一个共同的目标工作，协调一致，愉快合作最终开发出来高质量的产品

### 组织结构

由于组织结构的不同，团队成员的构成也不同：

#### 职能型



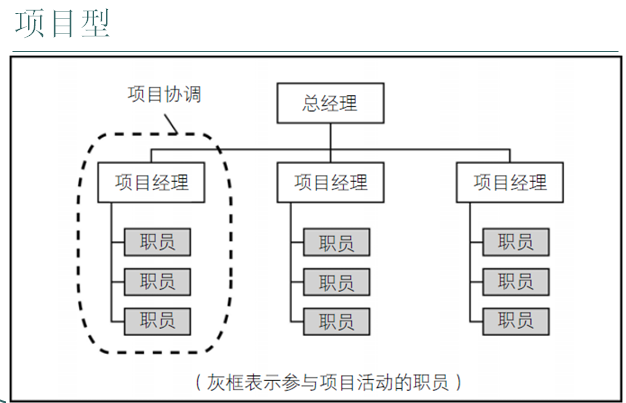
优点：

1. 可以充分发挥职能部门的资源集中优势
2. 部门的专家可以同时为部门内不同项目使用
3. 便于相互交流 , 相互支援
4. 可以随时增派人员
5. 可以将项目和本部门的职能工作融为一体

缺点:

1. 项目和部门利益发生冲突，职能部门更重视本部门的目标,会忽视项目目标
2. 资源平衡会出现问题
3. 权利分割不利于各个职能部门的交流和团结协作
4. 行政隶属关系使得项目经理没有充分的权利

#### 项目型



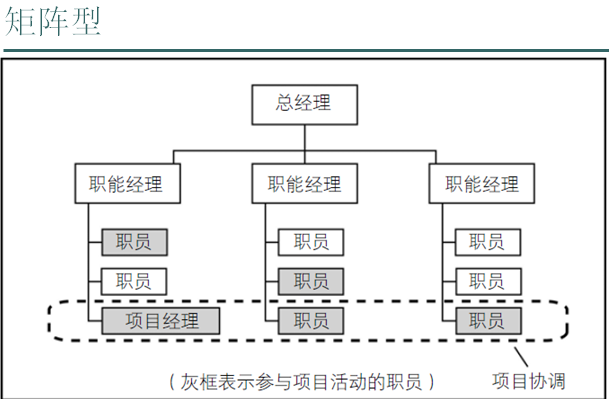
优点：

1. 项目经理对项目可以负全责
2. 项目目标单一，可以以项目为中心,有利于项目顺利进行
3. 避免多重领导
4. 组织结构简单，交流简单,快速

缺点：

1. 资源不能共享
2. 各个独立的项目处于相对封闭状态,不利于公司政策的贯彻
3. 对项目组织的成员缺少一种事业上的连续性和安全感
4. 项目组织之间处于分割状态,缺少信息交流

#### 矩阵型



优点：

1. 专职的项目经理负责整个项目 , 以项目为中心,
2. 公司的多个项目可以共享各个职能部门的资源
3. 即利于项目目标的实现,又利于公司目标方针的贯彻
4. 项目成员的顾虑减少了

缺点：

1. 容易引起职能经理和项目经理权力的冲突
2. 资源共享也能引起项目之间的冲突
3. 项目成员有多头领导

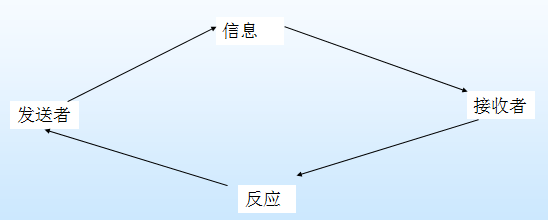
### 人员管理计划

人员管理计划描述了项目团队的组织结构，团队成员及角色、成员加入到团队和离开团队的时间、人员培训计划等。作为项目计划一部分，详细程度因项目而异。

参考项目计划书

## 【沟通计划】

### 项目沟通过程



### 沟通的基本原则

* 及时性
* 准确性
* 完整性
* 可理解性

### 沟通方式

* 书面沟通和口头沟通
* 语言沟通和非语言沟通
* 正式沟通和非正式沟通
* 单向沟通和双向沟通
* 网络沟通

### 沟通活动

项目周例会

### 沟通计划编制

* 沟通需求分类
* 沟通的内容
* 沟通方法
* 沟通职责
* 沟通进度
* 沟通计划维护

参考项目计划书

## 【风险计划】

### 风险定义

风险的定义：

* 损失发生的不确定性；
* 对潜在的，未来可能发生损害的一种度量

风险的三要素：

* 一个事件
* 事件发生的概率
* 事件的影响

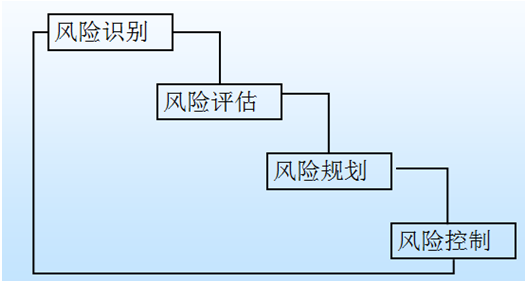
预测角度

* 已知风险－Known known
* 可预测风险-Known unknown
* 不可预测风险-unknown unknown

范围角度

* 项目风险
* 技术风险
* 商业风险

### 风险管理过程



#### 风险识别

风险识别领域：

* 产品规模
* 商业影响
* 客户相关
* 过程定义
* 开发技术
* 开发环境
* 人员数目及经验

风险识别方法：

* 头脑风暴法
* 情景分析法
* 面谈法
* 风险条目检查表

#### 风险评估

确定风险发生概率的估计和评价，项目风险后果严重程度的估计和评价，项目风险影响范围的分析和评价，以及对于项目风险发生时间的估计和评价。

风险概率值：

* >没有可能（0）
* <确定（1）

风险概率度量：

* 高、中、低
* 极高、高、中、低、极低
* 不可能，不一定，可能和极可能
* 等等

风险后果

* 风险影响项目目标的严重程度
* 从无影响到无穷大

风险后果度量

* 高、中、低
* 极高、高、中、低、极低
* 灾难，严重，轻微，可忽略
* 等等

#### 风险规划

针对风险分析的结果，为提高实现项目目标的机会，降低风险的负面影响而制定风险应对策略和应对措施的过程，即制定一定的行动和策略来对付、减少、以至于消灭风险事件

参考风险管理工作表

#### 风险措施

* 回避风险

对所有可能发生的风险尽可能的规避，采取主动放弃或者拒绝使用导致风险的方案。例如放弃采用新技术

* 转移风险

为了避免承担风险损失，有意识将损失或与损失有关的财务后果转嫁出去的方法:例如出售、分包

* 损失控制

损失预防、损失抑制

* 自留风险

由项目组织自己承担风险事故所致损失的措施

# 项目执行

## 【需求分析】

### 什么是需求分析

需求分析是在问题定义及可行性分析完成后细化用户对软件的功能和性能的要求，即用户希望软件能做什么事情，完成什么样的功能，达到什么性能。

需求分析是软件开发的开端，我们设计的软件产品存在不完整性、不正确性大部分原因是需求分析错误所导致的，因此，需求分析是软件生命周期中最重要的过程。

需求分析包括需求调研、需求描述及需求评审三个阶段。

需要注意的地方：需求分析工作早在项目启动阶段就开始了，根据项目的特点，一般情况在项目的计划、实施及控制阶段都存在需求分析。

### 需求调研

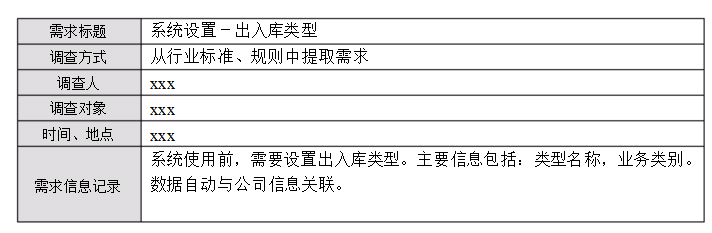
需求调研阶段的目标是挖掘用户的需求。首先确定目标用户，开发人员和目标用户确定一个问题领域，并定义一个描述该问题的系统，用户在这个问题领域和系统下提出需求，需求类型包括：功能需求、质量需求、用户体验需求等。

### 需求调研方法

* 与客户交谈，向用户提问题
* 参观用户工作流程，观察用户操作
* 向用户发调查问卷（通常是以选择题和是非题为主）
* 与同行、专家交谈，听取他们的意见
* 分析已经存在的软件产品，提取需求
* 从行业标准、规划中提取需求
* 上网搜索相关资料

### 需求调研报告

需求调研结束由调研人员编写调研报告，下图是调研报告中的一个需求：



### 需求描述

通过需求调研对收集到的资料进行分析、鉴别、综合和建模，清除用户需求的模糊性、歧义性和不一致性，分析系统的数据要求，为原始问题及目标软件建立逻辑模型。

分析人员需要编写“软件需求说明书”并与用户进行交流确认。

### 软件需求规格说明书重要性

需求分析工作完成的一个基本标志是形成了一份完整的、规范的需求规格说明书

需求规格说明书的编制是为了使用户和软件开发者双方对该软件的初始规定有一个共同的理解，使之成为整个开发工作的基础。

### 软件需求规格说明书编写

软件需求规格说明书的内容也是根据公司的项目管理水平而定。

内容基本包括：系统概述、运行环境（生产环境）、功能模块图、功能用例、性能需求、接口需求、其它需求等。

参考需求规格说明书。

### 辅助工具

一般用户对图形化界面感兴趣，最好是采用图形化的方式把用户的需求展示给用户，常见的有用例图、用户界面、流程图、状态图等。

常用的工具有：powerDesigner、visio

一般使用powerDesigner完成数据模型的描述，

使用visio完成业务流程图的绘制。

用户界面也可以采用visio绘制，用户特别要求的界面需要由美工画出来。

## 【设计】

### 什么是设计

软件设计是从软件需求规格说明书出发，根据需求分析阶段确定的功能设计软件系统的整体结构、划分功能模块、确定每个模块的实现算法以及编写核心代码，形成软件的具体设计方案。软件设计一般分为总体设计（概要设计）和详细设计。

### 概要设计

根据软件需求说明，建立目标系统的总体结构和模块间的关系；系统网络部署结构；核心业务流程； 定义系统的外部接口、及模块间的接口；编写概要文档；概要设计审查。

参考：概要设计文档

### 详细设计

详细设计的主要任务是设计每个模块的类接口、所需的局部数据结构、物理数据模型、页面原型等。

重点：使用powerDesigner设计物理数据模型

## 【编码】

### 统一开发工具

通常企业为了提高团队的生产效率会统一开发过程使用的工具，比如java编码统一使用开源产品eclipse，统一数据库客户端工具、统一文本编辑工具等。

### 编码规范

什么编码规范：

为了提高软件的质量，提高整个团队的工作效率，在编码环节要注重代码的编写规范，团队的高级工程师会制定编码的编写规范，不同的开发语言其它编写规范可能不同，制定好编码规范要求程序员按照规范编写代码。

规范代码的重要性：

* 规范的代码可以促进团队合作
* 规范的代码可以减少bug处理
* 规范的代码可以降低维护成本
* 规范的代码有助于代码审查
* 养成代码规范的习惯，有助于程序员自身的成长

一些建议：

* 求同存异

不要妄图改变组织的编码习惯，除非有绝对合理的理由，否则还是以民主为主，毕竟你没有权利要求所有人都沿用你的编码习惯。

* 定义编码规范越早越好

尽量在编码前定义好规范，团队成员按规范执行，如果已经开发了部分代码再去定义或修改规范那程序员要返工了。

* 编码规范不要太长

太长的文档没人看，所有人都一样，编码规范必须精炼，最好是只有2~3页，让开发人员可以打印出来随时查看。

* 必须是约定俗成的

规范必须是行业中约定俗成的，不要有什么个人个性化的内容。

### 代码审查

什么是代码审查：

代码审查（Code Review）这是一个流程，当开发人员写好代码后，需要让别人来review一下他的代码，这是一种有效发现BUG的方法。由此，我们可以审查代码的风格、逻辑、思路……，找出问题，以及改进代码。因为这是代码刚刚出炉的时候，所以，这也是代码重构，代码调整，代码修改的最佳时候。所以，Code Review是编码实现中最最重要的一个环节。

代码审查的内容举例：

* 代码能够工作么？它有没有实现预期的功能，逻辑是否正确等。
* 所有的代码是否简单易懂？
* 代码符合你所遵循的编程规范么？这通常包括大括号的位置，变量名和函数名，行的长度，缩进，格式和注释。
* 是否存在多余的或是重复的代码？
* 代码是否尽可能的模块化了？
* 是否有可以被替换的全局变量？
* 是否有被注释掉的代码？
* 循环是否设置了长度和正确的终止条件？
* 是否有可以被库函数替代的代码？
* 是否有可以删除的日志或调试代码？
* 是否有注释，并且描述了代码的意图？
* 所有的函数都有注释吗？
* 单元测试是否真正的测试了代码是否可以完成预期的功能？
* 是否检查了数组的“越界“错误？

代码审查工具：

实际代码审查建议使用工具来代替那些费时费力却收效甚微的人工代码检查工作，比如：PMD、Eclipse Code Review Plugin等。

## 【测试】

### 单元测试

由程序员在编码阶段完成，针对某个功能的代码进行单元测试。

### 集成测试

* 模块集成测试，针对某个模块内部的业务流程进行测试
* 系统集成测试，多个模块之间进行集成测试
* 程序员在开发完成系统的模块功能后进行集成测试
* 程序员对系统的整个业务流程集成测试完成后提交测试人员，测试人员依据开发人员编写的文档进行系统部署，并且再进行一次集成测试

### 系统测试

在系统开发中后期，测试人员参考开发人员编写的需求分析文档、设计 文档等编写测试用例文档，测试用例和开发人员编写的功能代码对应的。

测试方法包括：

黑盒测试：以终端用户身份进行功能测试 (企业开发常用)

白盒测试：针对开发人员编写的功能代码进行测试

测试人员根据测试用例进行详细功能测试，最好编写系统测试报告 。

系统测试时间一般会比开发时间略长。

### 性能测试

测试人员对系统进行压力测试、稳定性测试。

### 缺陷追踪



测试人员进行功能测试

测试出来bug提交到专门 的项目管理平台（测试管理平台是项目管理平台一部分）

提交bug时并没有指定修改bug的程序员，指定给了项目经理

项目经理登陆测试管理平台，将提交的 bug指派给开发人员

开发人员每天定时登陆测试管理平台，可看自己名下的bug，根据测试人员提交 bug信息进行修改bug

提交 bug信息包括：测试场景、测试数据、测试步骤

如果开发人员修改bug完成，提交bug到测试管理平台(bug状态是未修改)

测试人员 天定时登陆测试管理平台，可看自己名下的bug，对开发人员已提交的bug进行回归测试

如果测试通过，将bug关闭，

如果测试 不通过，将bug驳回。

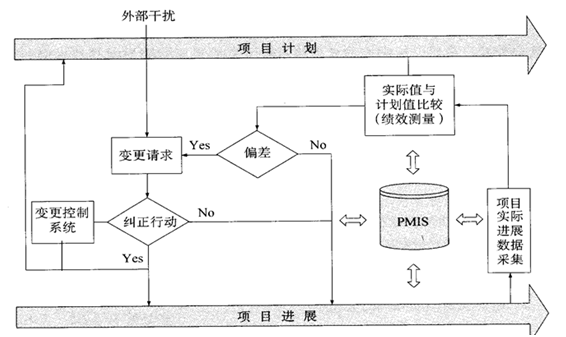
如果开发人员发现bug驳回，详细查看测试场景、测试数据、测试步骤，根据这些进行测试，通过几次测试bug重现不了，找到测试人员进行沟通。

# 【项目控制】

## 项目控制过程

参考项目的47个过程图，项目控制要对项目范围、项目成本、项目进度等进行控制，

项目经理关注的真正问题是项目中的偏差可以接受吗？项目经理要把注意力放在紧急的必须要解决的问题上 。



## 需求变更控制

需求总在变怎么办？

1. 确定需求变更控制过程
2. 建立需求变更控制委员会(SCCB)
3. 进行需求变更影响分析
4. 跟踪所有受需求变更影响的工作产品
5. 建立需求基准版本和需求控制版本文档
6. 维护需求变更的历史记录
7. 跟踪每项需求的状态
8. 衡量需求稳定性

如下图是需求变更的流程图：

## 配置管理

### 配置管理定义

软件项目中是否遇到如下的问题：

* 找不到某个文件的历史版本；
* 开发人员使用错误的版本修改程序
* 开发人员未经授权修改代码或文档；
* 人员流动，交接工作不彻底；
* 已修复的Bug在新版本中出现；
* 无法重新编译某个历史版本；
* 因协同开发中，或者异地开发，版本变更混乱导致整个项目失败；

配置管理定义：

* 记录软件产品的演化过程。
* 确保软件开发者在软件生命周期中的各个阶段都能得到精确的产品配置。
* 最终保证软件产品的完整性、一致性、追溯性、可控性。

配置管理的作用：

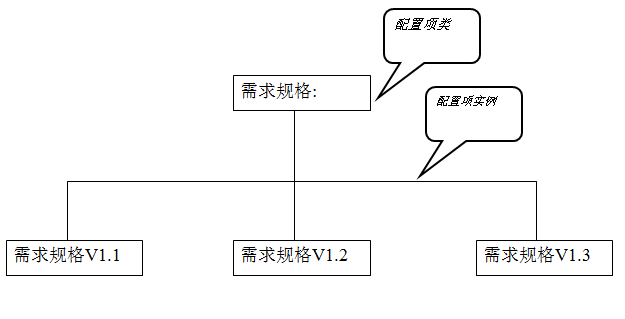
* 版本管理
* 变更管理

### 版本管理

配置项：

* 系统规格说明书
* 软件需求规格说明书
* 设计规格说明书
* 源代码
* 测试规格说明书
* 等

配置项版本：



配置管理工具：

IBM的ClearCase，SVN等

# 【项目结束】

## 项目结束条件

下列条件之一出现，可以终止项目：

* 项目计划中确定的可交付成果已经出现，项目的目标已经成功实现
* 甲方或乙方合同终止
* 项目已经不具备实用价值
* 项目由于各种原因而导致无限期拖长
* 项目出现了环境的变化，它负面影响项目的未来
* 项目所有者的战略发生了变化
* 项目无竞争力，难以生存

## 项目结束工作

制定结束要作的工作

* 制定结束计划

项目计划的一部分

与客户一同评审项目结束计划

细化并实施项目结束计划

* 完成收尾工作

范围确认

项目验收

费用结算

合同终结

* 项目结束评审

是否实现项目目标

是否遵循项目进度

是否在预算成本内完成项目

项目进度过程中出现的突发问题以及解决措施是否合适，问题是否得到解决

从该项目的实践中可以得到哪些经验和教训

* 项目总结

总结成功的经验和失败的教训

将项目中的有用信息进行总结分类，放入信息库，它是软件项目记录的资料。

## 项目验收

项目成功与失败的标准：

* 可交付成果如何
* 是否实现目标
* 是否达到项目业主的期望

