

## 第三章 软件项目管理

### 3.1 软件项目管理概述

软件项目管理具有相当的复杂性和特殊性。

#### 3.1.1 软件项目的特征

- 1) 软件产品的不可见性。
- 2) 项目的高度不确定性。
- 3) 软件过程的多变化性。
- 4) 软件人员的高流通性。

#### 3.1.2 软件项目管理的 4P

- 1) 人员 People

2) 产品 Product。软件产品的问题往往发生在软件需求阶段，其根本原因在于软件需求说明不明确。

3) 过程 Process。软件过程将软件开发和维护所用到的技术、方法、活动和工具有机地结合在一起，确保项目的成功经验和最佳实践得以有效的总结和重用，并在以后的项目实践中不断地完善和优化。

4) 项目 Project。软件项目的核心在于项目的规划和跟踪控制。(软件项目管理与过程管理的关系)

#### 3.1.3 软件项目管理活动

- 1) 项目启动阶段。确定项目范围；建立项目团队；建立项目环境。
- 2) 项目规划阶段。确定项目活动，预算项目成本，制定进度计划。
- 3) 项目实施阶段。监控项目执行，管理项目风险，控制项目变更。
- 4) 项目收尾阶段。客户验收项目，安装培训软件，总结项目经验。

### 3.2 人员组织与管理

在各种资源中，人员是最重要的资源。

#### 3.2.1 软件项目组织

- 1) 民主式组织结构。小组成员完全平等，所有工作商讨完成。

适用于规模小、能力强、习惯于共同工作的软件开发组。

缺点是缺乏明确的权威领导，很难解决意见分歧，无法适用于大规模的开发。

2) 主程序员式的组织结构。主程序员负责软件体系结构的设计和关键部分的详细设计，并指导程序员完成其他部分的详细设计和编程工作。

优点是视线里项目人员的专业化分工，确保了工作概念的完整性；同时降低了管理的复杂性，简化了成员之间的沟通和协调提高了效率。

缺点是在现实社会中，这种全才的人凤毛麟角。

3) 技术管理式的组织结构。将技术开发管理工作与行政管理工作分离，分别由两人承担。结合了以上两种方式的优点。但是也会出现职责不清的矛盾。

对于大型的项目，往往采用层次式组织结构，项目经理下设组长，组长下再设…

#### 3.2.2 微软公司的软件开发组织

- 1) 微软开发团队的特点。采用小型的多元化的项目组织进行开发工作。

优点在于交流管理成本低，决策和执行速度快，产品易于控制。

对于规模大的项目，将一个大型团队拆分成若干个结构清晰、目标明确、可灵活管理的小型项目组织。

2) 项目团队角色的划分。

六种类型的重要角色：产品管理、程序管理、软件开发、软件测试、用户体验、测试管理。

3) 大型项目的团队规模。将一个大型团队拆分成若干个结构清晰、目标明确、可灵活管理的小型项目组织。

### 3.2.3 软件团队的建设

1.一个高效的软件开发团队的特征有

- 1) 明确的目标与共同分享的项目前景。
- 2) 清晰的觉得分工与紧密的团队合作。
- 3) 融洽的管理员通畅的沟通。
- 4) 高昂的士气与高效的生产力。

2.团队建设的基本原则

- 1) 项目成员集中办公。
- 2) 人人参与产品设计。
- 3) 关注最终产品发布。
- 4) 注重项目人员培训。
- 5) 建立良好的企业文化。

## 3.3 项目沟通管理

项目沟通管理式软件工程中最关键而且最耗时的活动,大多数项目失败的原因都是项目内部或者外部沟通不畅造成的。

### 3.3.1 项目沟通的复杂性

团队规模式影响沟通有效性的重要因素之一。

最佳的团队规模是 3-7 人。

### 3.3.2 项目沟通方式（表 3-1 不同沟通方式使用的场合）

- 1.直接交谈。
- 2.电话交谈。
- 3.电子邮件。
- 4.会议。
- 5.项目网站。
- 6.书面报告。

### 3.3.3 项目沟通活动

1.规划项目沟通。

项目组内部交流信息活动：状态检查，组内讨论，问题讨论，需求阐明，项目变更。

项目组之间交流信息活动：客户评审，项目评审，版本发布，需求阐明，项目变更，问题讨论。

2.建立基础设施

可以建立项目信息系统（包括项目公告、问题讨论、项目文档等）

3.实施阶段性评审

一是由客户参与的评审，发生在需求文档发布和系统验收交付时。

二是项目组进行的评审，包括总体设计评审，详细设计评审，代码审查等。

4.每周组织小组会议

### 3.4 软件项目规划

#### 3.4.1 软件规模估算

##### 1.代码行技术

1) 代码估算可以由多名有经验的开发人员分别给出，最后计算出所有估算的可能性。

2) 开发人员可以提出一个有代表性的估算值范围，分别按最佳的 a、可能的 m、悲观的 b 三种情况给出估算值，再用公式计算出期望值。

$$L = (a + 4m + b) / 6$$

可以使用 L 进一步计算生产率、单位成本、代码出错率。(分别对应三个公式)

简单方便，但是需要依赖比较详细的功能分解结果，难以再开发初期进行估计，估算结果与所用的语言紧密相关，而且无法适用于非过程语言。

##### 2.功能点技术

计算未调整功能点->计算调整因子->计算调整功能点。

#### 3.4.2 软件成本估算

##### 1.专家判断。Delphi 方法。

2.类比估算。适用于评估一些与历史项目再应用领域、环境和复杂度上相似的项目，通过新项目与历史项目的比较得到规模估算。

##### 3.COCOMO 模型。

1) 基本 COCOMO 模型。利用一个估算出来的代码行数以自变量的函数来计算软件开发的工作量，函数的参数根据不同的项目有不同的取值。

2) 中间 COCOMO 模型。在基本模型上，根据不同的因素来对因素进行调节。新公式。

#### 3.4.3 软件项目计划

软件项目计划的形式多种多样，一般采用 IEEE 标准 1058-1998。

### 3.5 软件风险管理

#### 3.5.1 风险识别

试图采用系统化的方法，识别特定项目一致的和可预测的风险。

主要风险有：软件规模风险，商业影响风险，客户相关风险，软件过程风险，开发技术风险，开发环境风险，开发人员风险。

#### 3.5.2 风险分析

对已经识别的风险进行评估和评价，确定风险发生的概率和后果。

软件开发风险通常包括性能、成本、支持和进度等因素。

这些因素对项目目标可能产生的影响通常可以划分为忽略的、轻微的、严重的、灾难性的四个等级。

#### 3.5.3 风险规划

常见的风险的对应策略包括风险规避（设法降低风险出现的可能性）、风险缓解（设法减少风险产生的影响）、风险转移（将风险移交给第三方）、风险接受（采取应急方案应对风险的发生）等。

#### 3.5.4 风险监控

### 3.6 软件配置管理

#### 3.6.1 基本概念

##### 1.软件配置项（SCI）

是为了配置管理而作为单独实体处理的一个工作产品或软件。

可以是配置控制下的一组相关程序、文档和数据的集合。

##### 2.基线

是已经通过了正式复审的规格说明或中间产品，它可以作为进一步开发的基础，并且只有通过正式的变化控制过程才能改变。

简单地说，基线是软件配置项通过正式复审而进入正式受控的一种状态。

基线标示着软件开发过程的各个里程碑。

通常的软件基线有计划基线、需求基线、设计基线、实现基线、测试基线。

##### 3.版本

是在明确定义的实践上的某个配置项的状态。

##### 4.软件配置库

软件配置库用于记录整个软件声明周期内与配置相关的所有信息。

#### 3.6.2 配置管理活动

##### 1.配置项标识

将配置项采用合适的方式进行命名组织。

如 PCL-TOOLS 项目的配置层次结构。

##### 2.版本管理

版本的演变可以是串行的也可以是并行的。

不同的分支可能代表着不同的软件开发过程。

版本管理是对版本的各种操作进行控制，包括检出控制、分支与合并、版本历史记录和版本发布。

版本同步存取控制流程。

##### 3.系统构建

把系统组件编译和链接在一个特定的目标配置上的运行程序的过程。

##### 4. 变更控制

变更申请->变更许可->变更实施。

#### 3.6.3 配置管理工具

CVS