



第 11 章 软件工程管理



本章内容

11.1 软件工程管理概述

11.2 软件开发成本估算

11.3 软件工程人员组织

11.4 软件配置管理

11.5 软件质量保证

11.6 软件开发风险管理

11.7 软件工程标准与软件工程文档

01

OPTION

软件产品的特点

- 软件是逻辑产品，具有高度的抽象性。
- 具有同一功能的软件可以有多样性。
- 软件生产过程复杂，具有易错性。
- 软件开发和维护主要是根据用户需求“定制”的，其过程具有复杂性和易变性。
- 软件的开发和运行经常受到计算机系统环境的限制，因而软件需要有安全性和可移植性等。
- 软件生产有许多新技术需要软件工程师进一步研究和实践。



02

OPTION

软件工程管理的重要性

基于软件本身的复杂性，软件工程将软件开发划分为若干个阶段，每个阶段完成不同的任务、采取不同的方法。

如果软件开发管理不善，造成的后果会很严重。因此软件工程管理非常重要。



软件工程管理的内容

软件工程管理的内容包括对软件开发成本、软件开发控制、开发人员、组织机构、用户、软件开发文档、软件质量等方面的管理。软件开发成本的管理主要是对软件规模进行估算，从而估算软件开发所需要的时间、人员和经费。

软件开发控制包括进度控制、人员控制、经费控制和质量控制。由于软件产品的特殊性和软件工程技术的不成熟，制定软件工程进度计划比较困难。



本章内容

11.1

软件工程管理概述

11.2

软件开发成本估算

11.3

软件工程人员组织

11.4

软件配置管理

11.5

软件质量保证

11.6

软件开发风险管理

11.7

软件工程标准与软件工程文档

(1) 自顶向下估算法



(2) 自底向上估算法



(3) 差别估算法



(4) 专家估算法

(5) 类推估算法

(6) 算式估算法

01

OPTION

代码行技术

代码行技术是一个相对简单的定量估算软件规模的方法。该方法先根据以往经验及历史数据估算出将要编写的软件的源代码行数，然后以每行的平均成本乘估计的总行数，估算出总的成本。

优点



代码行是所有软件开发项目都有的“产品”，很容易计算。



已有大量的基于代码行的文献资料和数据。

缺点



用不同语言实现同一软件产品时所需要的代码行数并不相同。



代码行技术不适用于非过程性语言。

02

OPTION

任务估算技术

先把开发任务分解成许多子任务，子任务又分解成下一层次子任务，直到每一任务单元的内容都足够明确为止。然后把每个任务单元的成本估算出来，汇总即得项目的总成本。

这种估算方法的优点是，由于各个任务单元的成本可交给该任务的开发人员去估计，因此估计结果比较准确。缺点在于，由于具体工作人员往往只注意到自己职责范围内的工作，而对涉及全局的成本。



COCOMO2 模型分为如下3 个模型，在估算软件开发工作量时，对软件细节问题考虑的详尽程度逐渐增加。

1 应用系统组成模型

用于估算构建原型的工作量，这种模型考虑到大量使用已有构件的情况。

2 早期设计模型

用于软件结构设计阶段

3 后期设计模型

用于软件结构设计完成之后的软件开发阶段

COCOMO2 模型把软件开发工作量表示成千行代码数 (KLOC) 的非线性函数, 如式所示。

$$E = \alpha \times \text{KLOC}^b \times \prod_{i=1}^{17} f_i$$

E 是开发工作量 (以人 × 月为单位), α 是模型系数, KLOC 是估算的源代码行数 (以千行为单位), b 是模型指数, f_i ($i = 1 \sim 17$) 是软件成本因素。

每个软件成本因素都根据其重要程度和对工作量影响的大小被赋予一定的数值, 称为工作量系数。这些软件成本因素对任何一个项目的开发工作都有影响, 应该重视这些因素。

McCabe 方法根据程序控制流的复杂度定量度量程序复杂度，首先画出程序图，然后计算程序的环行复杂度。

1

画出程序图

McCabe 方法首先画出程序图（也称流图），这是一种简化的流程图，把流程图中的每个框都画成一个圆圈，流程图中连接不同框的箭头变成程序图中的弧即得到程序图。

2

计算程序的环行复杂度

用McCabe 方法度量得出的结果称为程序的环行复杂度



本章内容

11.1

软件工程管理概述

11.2

软件开发成本估算

11.3

软件工程人员组织

11.4

软件配置管理

11.5

软件质量保证

11.6

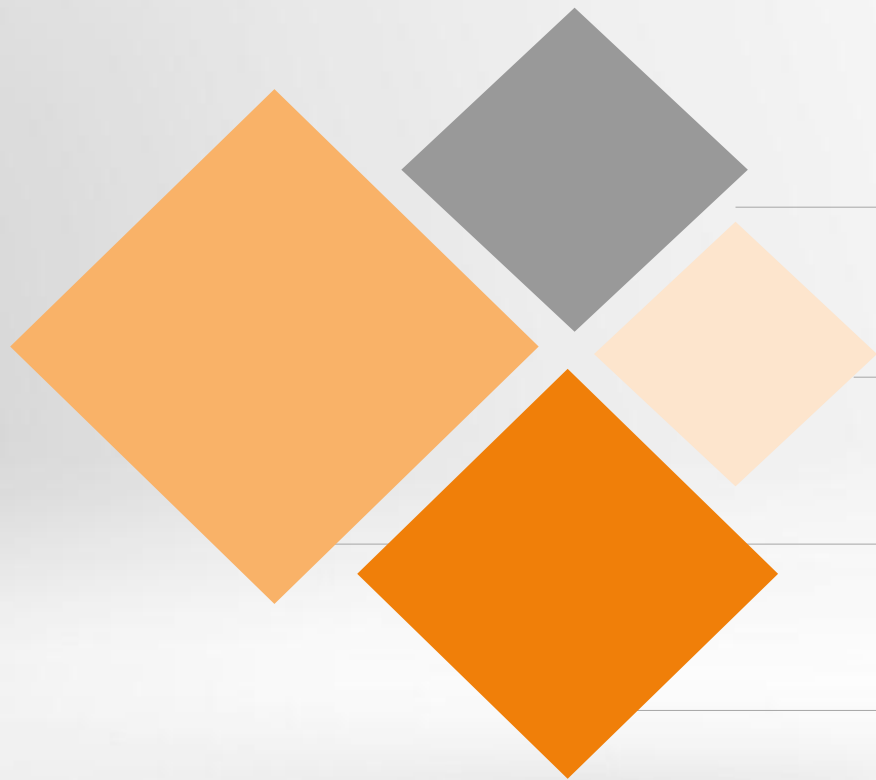
软件开发风险管理

11.7

软件工程标准与软件工程文档

软件开发人员

软件开发人员一般分为项目负责人、系统分析员、高级程序员、程序员、初级程序员、资料员和其他辅助人员。



项目负责人需要对项目的需求和团队人员有全面的了解

系统分析员需要有概括能力、分析能力和社交活动能力

程序员需要有熟练的编程能力等

资料员和其他辅助人员负责及时登记软件工程每个阶段的文档等资料

02

OPTION

组织机构

软件开发团队不能只是一个简单的集合，要求具有良好的组织机构，要具有合理的人员分工和有效的通信，共同高效率地完成任务。



按项目划分的模式



按职能划分的模式



矩阵型模式

软件工程团队人员应遵循如下职业道德。



诚实可信、恪尽职守、敬重法律、遵守道德

服从项目领导，严守国家机密，重视合同和协议

遵从知识产权的各项规则，不剽窃他人的研究成果

每位团队成员都为团队的目标、具体目标积极承担各自的责任

遇到问题，擅于沟通、协调，共同完成所担负的工作

03

OPTION

用户

软件是为用户开发的，在开发过程中必须自始至终得到用户的密切合作和支持。



不积极配合



求快求全



要求的不断变化



本章内容

11.1

软件工程管理概述

11.2

软件开发成本估算

11.3

软件工程人员组织

11.4

软件配置管理

11.5

软件质量保证

11.6

软件开发风险管理

11.7

软件工程标准与软件工程文档

软件配置 (Software Configuration) 是软件产品在开发和运行过程中产生的全部信息，这些信息随着软件开发运行工作的进展而不断变更。

软件配置管理 (Software Configuration Management, SCM) 是在软件的整个生命周期内管理变更的一组活动。软件配置管理与软件维护不同，软件维护是在软件交付给用户使用以后，根据用户的需要修改软件；软件配置管理是在软件项目定义时就开始，一直到软件“退役”后才停止的控制活动。软件配置管理的目的是使软件变更所产生的错误达到最少，从而有效地提高软件生产率。

软件配置（Software Configuration）是软件产品在开发和运行过程中产生的全部信息，这些信息随着软件开发运行工作的进展而不断变更。软件过程产生的全部信息可分为3类。



供技术人员或用户使用的
软件工程文档

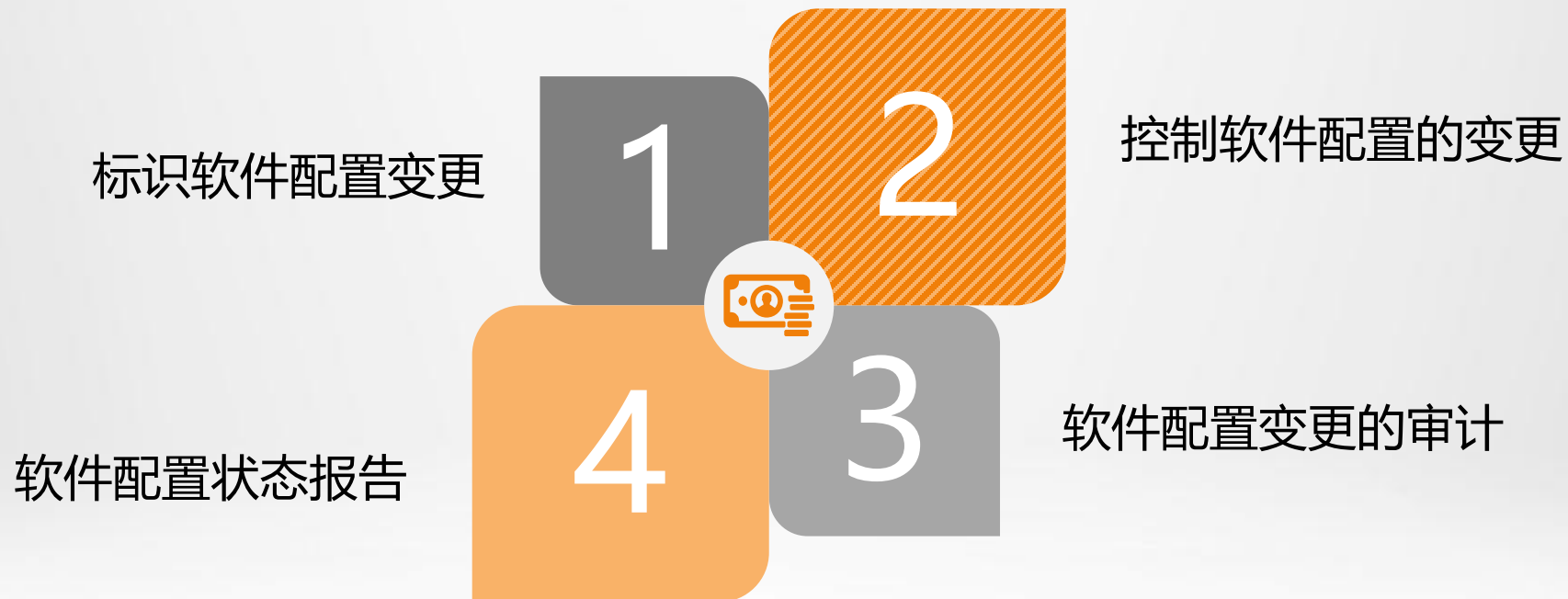


计算机程序源代码、可执行程序及存储在计算机内的
数据库



数据（程序内包含的
数据或程序外的数据

软件配置管理有4 项任务。





本章内容

- 11.1 软件工程管理概述
- 11.2 软件开发成本估算
- 11.3 软件工程人员组织
- 11.4 软件配置管理
- 11.5 软件质量保证
- 11.6 软件开发风险管理
- 11.7 软件工程标准与软件工程文档

软件质量是指软件满足明确规定或隐含定义的需求的程度。软件质量的要点如下。



软件功能必须满足用
户规定的需求



软件应遵守规定标准所定
义的一系列开发准则



软件应满足某些隐含的需
求

01
OPTION

质量特性

(1) 功能性



(2) 可靠性



(3) 易使用性



(4) 效率

(5) 可维护性

(6) 可移植性

02
OPTION

质量子特性

功能性

适合性、准确性、互用性、
安全性



效率

资源效率、时间效率

可靠性

成熟性、容错性、易恢复性



可维护性

易分析性、易改变性、稳
定性、易测试性

易使用性

易理解性、易学性、易
操作性



可移植性

适应性、易安装性、一致性、
易替换性

软件质量保证是一个复杂的系统，它采用一定的技术、方法和工具，以确保软件产品达到或超过在该产品的开发过程中所规定的标准。

- 应用好的技术、方法和工具
- 软件测试
- 进行正式的技术评审
- 软件质量标准的实施
- 控制软件配置变更
- 程序正确性证明
- 记录、保存和报告软件过程信息





本章内容

- 11.1 软件工程管理概述
- 11.2 软件开发成本估算
- 11.3 软件工程人员组织
- 11.4 软件配置管理
- 11.5 软件质量保证
- 11.6 软件开发风险管理
- 11.7 软件工程标准与软件工程文档

01
OPTION

按照风险的影响范围分类

1

项目风险

项目风险是指预算、进度、人力、资源、用户及需求等方面的潜在问题和它们对软件项目的影响

2

技术风险

技术风险是指设计、实现、接口、验证和维护等方面的潜在问题。

3

商业风险

- 市场风险
- 策略风险
- 销售风险
- 管理风险
- 预算风险

02

OPTION

按照风险的可预测性分类

1

已知风险

通过仔细评估项目计划、开发项目的商业和技术环境及其他可靠的信息来源，可以发现的那些风险

2

可预测的风险

这类风险可以从过去项目的经验中推测出来

3

不可预测的风险

这类风险可能真的会出现，但是很难事先加以识别。

采用建立风险条目检查表（简称风险表）的方法，可以帮助人们有效地识别风险，该表主要用来识别下列已知的和可预测的风险。

- 产品规模：与要开发或要修改的软件总体规模相关的风险。
- 商业影响：与管理或市场所施加的约束相关的风险。
- 用户特性：与用户素质以及开发人员和用户定期沟通的能力相关的风险。
- 过程定义：与软件过程定义的程度以及软件开发组织遵守软件过程的程度相关的风险。
- 开发环境：与用来开发产品的工具的可用性以及质量相关的风险。
- 开发技术：与待开发系统的复杂性及系统所包含的技术的“新奇性”相关的风险。
- 人员能力与经验：与软件工程师的总体技术水平及项目经验相关的风险。

01

OPTION

建立风险表

建立风险表是一种简单的风险预测技术。

一旦填好了风险表的风险列、风险类型列、概率列和影响等级列，就应该根据概率和影响等级来排序。高概率、影响大的风险放在表的上方，低概率、影响小的风险放在表的下方，这样就完成了第一次风险排序。

影响等级：

- 1—灾难性的
- 2—严重的
- 3—轻微的
- 4—可忽略的

风险类型：

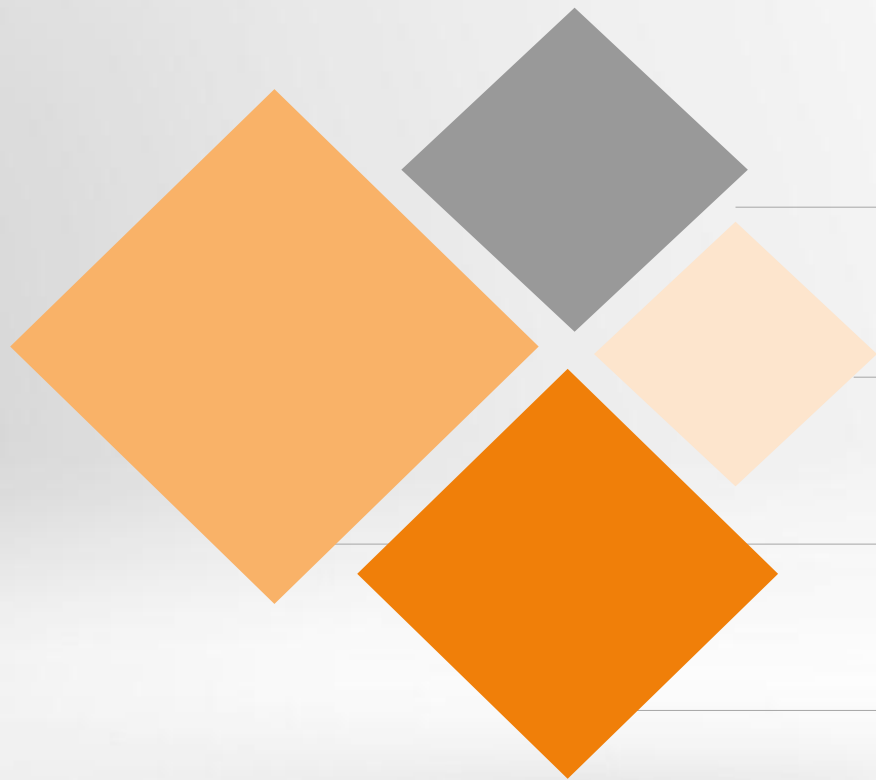
- PS—产品规模
- BU—商业影响
- CU—用户特性
- TE—开发技术
- DE—开发环境
- ST—人员能力与经验

02

OPTION

评估风险后果

如果风险真的发生了，建议从性能、成本、支持和进度等4个方面评估风险的后果。



性能风险——产品能满足需求且符合其使用目的的不确定程度

成本风险——能够维持项目预算的不确定程度

支持风险——软件易于改错、适应和增强的不确定程度

进度风险——能够实现项目进度计划且产品能按时交付的不确定程度

01

OPTION

风险缓解

- 与现有人员一起探讨人员变动的原因，例如工作条件恶劣、报酬低、劳动力市场竞争激烈等。
- 在项目开始之前采取行动，减少不能控制的风险。
- 项目启动之后，假设会发生人员变动，当人员离开时，使用开发技术来保证工作的连续性。
- 组织项目团队，使每一个开发活动的信息能被广泛传播和交流。
- 制定编写文档的标准，并建立相应机制以确保及时创建文档。
- 所有开发工作都经过其他人员的复审，从而让不止一个人熟悉该项工作。
- 为每个关键的技术人员都指定一个后备人员。

02
OPTION

风险监控



团队成员对于项目压力的态度

团队的凝聚力

团队成员彼此之间的关系

与工资和奖金相关的潜在问题

在公司内和公司外工作的可能性

03

OPTION

风险管理和制订意外事件处理计划

大型项目可能产生若干风险，如果为每一个风险都制定风险缓解措施，那么风险管理本身就变成了一个“大项目”，因此，要将Pareto的80/20法则应用于软件风险管理上。经验表明，整个项目风险的80%（可能导致项目失败的80%的潜在因素）能够由20%已经识别的风险来说明。早期风险分析步骤中所做的工作有助于确定哪些风险在这20%中，从而可以让设计人员将精力集中在最高级别的风险上，并为其采取相应的缓解措施。



本章内容

11.1

软件工程管理概述

11.2

软件开发成本估算

11.3

软件工程人员组织

11.4

软件配置管理

11.5

软件质量保证

11.6

软件开发风险管理

11.7

软件工程标准与软件工程文档

01
OPTION

软件工程标准化的定义

软件工程标准化给软件开发工作带来的好处如下。

- 提高软件的可靠性、可维护性和可移植性，进而提高软件产品的质量。
- 提高软件生产率。
- 提高软件工作人员的技术水平。
- 提高软件开发人员之间的通信效率，减少差错。
- 有利于软件工程的管理。
- 有利于降低软件成本并缩短软件开发周期。



软件工程标准化的定义

软件工程标准的类型是多方面的，它包括过程标准（如方法、技术及度量等）、产品标准（如需求、设计、部件、描述及计划报告等）、专业标准（如职别、道德准则、认证、特许及课程等）以及记法标准（如术语、表示法及语言等）。



03
OPTION

软件工程标准的层次

根据软件工程标准的制定机构与适用范围，软件工程标准可分为 5 个等级。



11.7.2

软件工程文档的编写

软件工程文档的作用是提高软件开发过程的能见度，提高开发效率，作为开发人员阶段工作成果和结束标志，记录开发过程的有关信息，便于使用与维护；提供软件运行、维护和指导用户操作的有关资料，便于用户了解软件功能、性能等。

序号	文档	阶段					
		可行性研究	需求分析	设计阶段	系统实现	测试阶段	运行与维护
1	可行性分析研究（报告）	√					
2	软件开发计划	√	√				
3	软件需求规格说明		√				
4	数据需求说明		√				
5	软件测试计划		√				
6	软件（结构）设计说明			√			
7	数据库（顶层）设计说明			√			
8	软件用户手册		√	√	√	√	√
9	计算机操作手册			√	√	√	√
10	软件测试报告					√	
11	开发进度月报	√	√	√	√	√	
12	项目开发总结报告					√	√
13	维护记录						√