

# 工作量估算与风险管理

## 1 工作量估算

### 1.1 ch3 习题 12

单位时间单位规模来测量生产率合适吗？

使用单位时间单位规模来测量生产率并不是一个完全准确的方法，因为这种方法并没有考虑到以下因素：

1. 代码行数可能会因为使用不同的编程语言而产生差异。例如，使用高级语言比如 Python 编写的代码通常比使用低级语言比如 C++ 编写的代码行数更少，但是功能却可能相同。
2. 生产率的测量应该在实现开始之前进行。如果在实现过程中测量生产率，它很可能会受到一些无法控制的因素的影响，例如需求变更、技术难题、人员变动等等。
3. 程序员可能会为了达到生产率的目标而堆积代码，从而导致代码的质量下降。因此，生产率的测量应该同时考虑代码质量，例如代码的可维护性、可读性、健壮性等等。

当评估生产率时，仅仅使用代码行数或应用点数等指标并不足够准确和全面。因为这些指标只能反映开发过程中的数量，而并不能说明开发过程的质量和效率。为了更全面地评估生产率，可以考虑以下几个指标：

1. 代码质量：代码质量是衡量软件开发效率的重要指标之一。代码质量好的代码更容易被维护和修改，从而可以减少开发人员的工作量。常见的代码质量指标包括代码可读性、代码规范性、代码复杂度等。
2. 测试覆盖率：测试覆盖率是衡量软件测试效果的指标之一。测试覆盖率越高，表示测试用例对软件的覆盖率越高，缺陷也就越容易被发现和修复。因此，测试覆盖率可以反映软件开发效率和质量。
3. 缺陷密度：缺陷密度是衡量软件质量的指标之一。缺陷密度越低，表示软件质量越高，软件的稳定性和可靠性也就越高。因此，缺陷密度可以反映软件开发的效率和质量。
4. 项目进度：项目进度是衡量软件开发效率的指标之一。项目进度的快慢可以反映开发团队的效率和工作状态，也可以帮助项目经理进行合理的时间安排和资源分配。

综上所述，评估生产率需要综合考虑多个指标，从不同角度反映软件开发的质量和效率。只有通过全面的评估才能更好地指导软件开发过程，并最终实现高质量、高效率的软件产品。

## 1.2 估算自己项目的初始工作量

参考书 3.7 (P94)皮卡地里电视广告销售系统按 COCOMOII 的工作量模型进行工作量估算的例子 (结合 P79-80 表)，估算自己项目的初始工作量。

对于小型博客平台，假定初始估算包含 7 个屏幕和 4 个报表，汇总如下表所示

名字	屏幕或报表	复杂性	权值
登录/注册	屏幕	简单	1
用户信息	屏幕	简单	2
日志	屏幕	适中	5
日历	屏幕	简单	1
广告	屏幕	适中	4
推荐	屏幕	适中	5
公告	屏幕	简单	1
文章	屏幕	难	10
用户管理	报表	难	10
文章管理	报表	难	15
图片管理	报表	难	10
统计信息	报表	难	15

可得应用点计数 (NOPS) 为 79。

假定开发人员具有低的经验和低的 CASE 熟练度，可知生产率评分为 7。

可以得到小型博客平台的估算工作量为 **NOPS/生产率评分=11.29 人月**。

由于本小组已经事先对小型博客平台的需求有一定的理解，故进一步采用早期设计模型

先例：小组并无独立开发博客平台的先例，但由于网上可以找到许多类似开源项目作为参考，所以有先例的了解（一般，评分 3），相对一般的软件工程文档要求（一般，评分 3），要求几乎不改变（非常高，1），大部分时间要求团队合作（高，评分 2），过程非正式且成熟度较低（低，评分 4）。

对评分求和： $3+3+1+2+4=13$ 。计算比例指数为  $1.01+0.01(13)=1.14$ 。

新的估算为  $11.29^{1.14}=15.85$  人月。

## 2 风险管理

### 2.1 ch3 习题 11

学生软件开发项目的风险如下：

1. 项目规模和复杂度：学生团队可能没有足够的经验和技能来开发一个完整的社交媒体平台。此外，需要实现的功能比较多，如用户认证、实时聊天、数据加密等，可能会增加开发难度和风险。
2. 团队成员和沟通：学生团队成员可能来自不同的专业和背景，需要花费更多的时间来协调和沟通。此外，由于社交媒体平台的特殊性质，需要注意用户体验和用户隐私问题，需要团队成员之间建立良好的沟通和理解。并且学生较为稚嫩，在需求分析过程中，软件设计负责人带领的需求分析小组和用户在进行交流的过程中可能会发生矛盾，出现争吵，会对软件项目的实施产生影响，影响软件项目的进度，甚至会导致项目失败。
3. 时间和进度安排：学生缺乏经验，很可能造成原先计划制定的不科学和不准确，因此会导致实施过程中进度难以控制，如果强行按照计划来执行显然是不可行的，为此，必须对计划重新进行分析和调整。
4. 技术和工具：学生团队的技术水平和使用的工具可能比较有限，这可能会导致技术难题并且找不到解决途径，或者使用不当的工具，从而影响开发进度和软件质量。此外，由于社交媒体平台需要处理大量的用户数据和隐私问题，需要使用适当的技术和工具来确保数据的安全性和可靠性。
5. 个人安排的不确定性：学生并没有一个固定的职位，可能在软件开发的过程中出现各种突发情况，例如某项目组成员突然告诉项目负责人，他已办理好了去德国的签证，2 周后他会辞职离开公司前往德国留学，该成员的离开显然将会影响项目

组的正常运作，影响项目的进度，为此将会给项目的实施带来损失

### **风险暴露：**

风险暴露 = (风险概率) x (风险影响)，它是指风险在某个特定时刻或某个特定情境下成为现实的可能性，也可以理解为风险事件的发生概率。在风险管理中，风险暴露是一个非常重要的概念，因为它决定了风险对组织或个人造成的实际影响。如果一个风险没有暴露，那么它对组织或个人的影响将是零或极小的，但如果风险已经暴露，那么它就会变成现实，给组织或个人带来一定的损失或影响。

因此，在风险管理中，识别和评估风险暴露是至关重要的。只有了解潜在风险的暴露程度，才能采取适当的风险控制措施，并减轻风险带来的负面影响。通常，风险暴露程度会受到风险发生概率、风险影响范围和风险应对措施等因素的影响。对于不同类型的风险，可以采取不同的应对措施来降低风险暴露，例如，采用防范措施、转移风险、减少风险等方法。

### **减轻各种风险的技术：**

1. 需求变更风险：使用敏捷开发模式，并与用户保持密切沟通，以及及时调整需求。
2. 技术难度风险：进行技术评估，选择适合项目需求的技术，同时开发人员也应该进行必要的技术培训和学习。
3. 安全风险：在开发过程中，实施安全开发和安全测试，以确保系统的安全性。
4. 不现实的进度和预算风险：在制定进度和预算计划时，与团队成员和利益相关者共同确定合理的期望值；对项目进度和预算进行实时跟踪和调整，确保它们始终保持在可控范围内；将项目分解为小的可管理的阶段，以确保每个阶段的进度和预算都是可实现的。
5. 人员管理风险：在团队建设中，注重人才培养和团队建设，制定清晰的团队目标和分工，加强人员沟通和配合。
6. 华丽的计划风险：制定简单、清晰、易于理解和跟踪的计划，以便团队成员可以轻松了解其角色和责任；为每个阶段或任务设置可衡量的目标和里程碑，以确保项目按时交付；使用项目管理工具和技术来跟踪和控制项目进度和预算，及时调整计划和任务。
7. 需求的持续变更风险：在项目开始前与利益相关者沟通，确保所有需求已明确定义和记录；实施敏捷开发流程，强调与利益相关者的协作和及时反馈，以满足他们的需求和期望；制定合理的变更管理计划，确保每个需求变更都得到审批，并考虑其对项目进度和预算的影响。

总之，对于一个学生软件开发项目，识别和管理风险是非常重要的。通过采取适当的风险管理技术，可以帮助开发团队降低风险，保证项目的成功实施。

## 2.2 分析自己项目中可能存在的风险，并进一步细化风险管理

分析自己项目中可能存在的风险，并进一步细化风险管理（做出风险分级及应对预案）。

### 2.2 风险分级及应对预案

#### 风险分级：

##### 1 级风险：

- 该项目规模巨大，涉及多个团队的协作开发。
- 开发和使用新技术很困难。
- 项目太短，无法按时完成任务。
- 需求的频繁变化导致开发进度滞后。
- 该系统要求高可靠性和安全性。

##### 次要风险：

- 开发人员数量不足，导致开发进度缓慢。
- 开发团队缺乏相关技术经验。
- 不当的代码质量管理会导致代码漏洞和缺陷。
- 在发展过程中存在沟通不良的问题。
- 开发工具的不当使用会导致效率低下。

##### 3 级风险：

- 单个模块的实现是复杂的。
- 单个团队成员离开或暂时离开项目，导致开发进度延迟。
- 硬件或网络设备的故障阻碍了开发进度。
- 非关键功能问题的延误对整个项目进度影响很小。
- 在出版之前，会发现一些可以容忍的小问题（如排版问题、拼写错误等）。

#### 应对措施

##### 1 级风险：

- 组织人员和资源，加强项目管理和进度跟踪，确保项目顺利进行。
- 技术团队应加强技术研究和知识培训，确保技术可行性和正确性。
- 进行风险评估和规划，制定详细的项目计划和实施计划。

- 加强沟通协调，明确项目变更范围和优先级，及时调整发展计划。
- 对系统进行全面测试和评估，以确保系统的可靠性和安全性。

次要风险：

- 重新评估团队规模，合理分配资源，调整发展计划。
- 加强技术培训，招聘和培养具有相关技术经验的开发人员。
- 实施质量管理和代码审查，及时修复代码中的错误和缺陷。
- 加强沟通协调，建立有效的沟通渠道和反馈机制。
- 优化开发流程，采用适当的工具和方法，提高效率和质量。

3 级风险：

- 对该模块进行了详细的分析和设计，并采用适当的技术和方法进行了开发。
- 加强人员培训和知识共享，确保团队具备必要的技术能力。
- 实施备份和恢复措施，以防止由于硬件或网络故障导致的数据丢失。
- 根据项目的优先级和紧迫性及时调整开发计划。
- 进行详细的测试和评估，以确保软件的质量和可用性。