**实验六 工作量估算****，风险管理，软件需求规格说明SRS（1）**

**夏思晓 202100031020**

**一、实验目的：**

1. 工作量估算

2. 风险管理

3. 学习软件需求规格说明SRS文档的要求和结构

**二、实验内容：**

1. **工作量估算：**

**ch3 习题12（小组讨论）。**

习题12. 很多项目经理根据过去项目中程序员的生产率来计划项目的进度，生产率通常根据单位时间的单位规模来测量。例如，一个组织机构可能每天生产300行代码或每月生产1200个应用点。用这种方法测量生产率合适吗？根据下列事项讨论生产率的测度：

● 用不同的语言实现同样的设计，可能产生的代码行数不同。

● 在实现开始之前不能用基于代码行的生产率进行测量。

● 程序员可能为了达到生产率的目标而堆积代码。

使用单位时间内的代码行数或应用点数量的方法来测量程序员的生产率是一种比较常见的做法，但其存在一些问题，并不总是合适的。具体分析如下：

（1）用不同的语言实现同样的设计，可能产生的代码行数不同。不同的编程语言有不同的语法、结构特点，这可能会导致实现相同功能所需的代码函数差异较大。例如，如果使用C语言、C++或者是Java等来进行设计，其代码量往往很大，但如果使用python等倾向于使用更少的代码行来完成任务的语言来说，其代码量相对较少。因此不能简单地按照代码行数来判断生产率。

（2）在实现开始之前不能用基于代码行的生产率进行测量。在实现开始之前，虽然已经对工程量进行了估计，但项目的难度和具体的难点可能还未知。比如开发一个OS内核的速度要慢于开发一个简单的web项目，以程序员过去写代码的速度估计本项目的速度不一定合适。此外，项目的早期阶段通常涉及更多的设计、规划和测试工作，这些工作很难用代码行数来衡量。

（3）程序员可能为了达到生产率的目标而堆积代码。当程序员没办法按时写出所要求数量的代码时，可能为了完成任务而堆积很多无用或冗余的代码，比如堆积大量无用的if else语句，这并不利于系统的代码架构和性能的提高，即会滋生为了追求生产率而牺牲项目的长期利益行为的出现。

总结来看，生产率应该是一个综合评估的结果，包括代码质量、功能实现、测试覆盖率、文档完整性等多方面的内容，仅仅依靠代码行数或应用点数量作为生产率的唯一指标是片面且不合理的，应考虑更加全面的度量方法来评估生产率，更加注重代码质量和可维护性，避免出现为了追求生产率而牺牲项目长期利益的行为出现。

**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。**

人才招聘系统主要包含个人用户子系统、企业用户子系统、客服管理子系统三大部分，根据小组的项目活动图得出预估初始工作量，再结合COCOMOII工作量模型进行调整，小组项目的初始工作量如下表所示。

**表 1 个人用户子系统**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 比例因子 | 非常低（5） | 低（4） | 一般（3） | 高（2） | 非常高（1） | 极高（0） |
| 先例 |  |  | √ |  |  |  |
| 灵活性 |  | √ |  |  |  |  |
| 去除的重大风险 |  |  |  | √ |  |  |
| 团队交互过程 |  |  | √ |  |  |  |
| 过程成熟度 |  | √ |  |  |  |  |

初始预计工作量为20人日，约1人月，计算的比例因子为3+4+2+3+4=16，计算的比例指数为1.01+0.01（16）=1.17，调整后新的估算为11.17，即1个人月。

**表 2 企业用户子系统**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 比例因子 | 非常低（5） | 低（4） | 一般（3） | 高（2） | 非常高（1） | 极高（0） |
| 先例 |  |  |  | √ |  |  |
| 灵活性 | √ |  |  |  |  |  |
| 去除的重大风险 |  |  |  | √ |  |  |
| 团队交互过程 |  |  |  | √ |  |  |
| 过程成熟度 |  | √ |  |  |  |  |

初始预计工作量为30人日，约1.5人月，计算的比例因子为2+5+2+2+4=15，计算的比例指数为1.01+0.01（15）=1.16，调整后新的估算为1.51.16，即1.6个人月。

**表 3 客服管理子系统**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 比例因子 | 非常低（5） | 低（4） | 一般（3） | 高（2） | 非常高（1） | 极高（0） |
| 先例 |  | √ |  |  |  |  |
| 灵活性 | √ |  |  |  |  |  |
| 去除的重大风险 |  |  | √ |  |  |  |
| 团队交互过程 |  |  | √ |  |  |  |
| 过程成熟度 | √ |  |  |  |  |  |

初始预计工作量为40人日，约2人月，计算的比例因子为4+5+3+3+5=20，计算的比例指数为1.01+0.01（20）=1.21，调整后新的估算为21.21，即2.31个人月。

**2. 风险管理**

**ch3 习题11 （小组讨论）**

习题11. 即使你在做学生项目，在按时完成项目方面也有极大的风险。分析一个学生软件开发项目并列出其中的风险。风险暴露是什么？你可以使用什么技术来减轻各种风险？

学生软件开发项目可能面临的风险及应对技术：

1. **技术风险**：学生可能不熟悉所使用的编程语言、框架或工具，缺乏经验导致无法熟练运用导致开发进度受阻。可以进行技术预调研，提前学习并掌握所需技术，尽可能选取已经上过专业课程或者进行相关运用的编程语言、工具等进行项目开发，此外，也可以通过组建技术互助小组，加入一些论坛来共享学习资源和经验，必要时及时咨询专业老师的指导。
2. **时间管理风险**：学生可能因为拖延症、课业压力导致的时间紧张灯导致项目难以按时完成，往往是在截止日期前仓促收尾。可以使用的技术为制定详细的项目时间表，通过阶段性目标来督促完成项目任务，此外，也可以通过使用项目管理工具来进行时间跟踪和进度监控等。
3. **需求明确风险**：学生可能对项目需求理解不足，导致开发过程中频繁变更需求，影响项目进度。可以通过在项目开始阶段，与客户进行充分的沟通，确保对需求有清晰、准确的理解。在原型设计方面，通过原型设计或草图来展示软件的基本功能和界面，以便客户能够更直观地理解需求。此外，制定详细的需求文档，记录所有的功能点、界面设计和业务流程，确保开发团队和客户之间对需求有共同的理解。
4. **团队协作风险**：由于学生可能缺乏团队合作经验，或者团队成员之间的性格、技能差异较大，导致团队协作出现问题。团队成员之间沟通不畅或分工不明确，可能导致工作重复或遗漏，从而降低工作效率，还可能引发团队成员之间的矛盾和不满，影响团队凝聚力。可以在项目开始阶段，明确每个团队成员的职责和任务，确保工作分配合理、明确。

建立沟通机制，定期召开团队会议，分享工作进展、讨论问题和交流经验，确保团队成员之间信息畅通。

1. **资源限制风险**：学生可能面临资金、设备或数据资源的限制，无法实现一些复杂的功能或优化性能，影响软件的质量和用户体验。可以在项目开始阶段，对所需资源进行充分评估，制定合理的预算和采购计划。如果资源有限，可以寻求学校、老师或外部机构的支持和帮助，例如申请项目资助、借用设备等。此外，还可以充分利用开源工具和资源，减少成本开支。

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）。

项目风险管理内容详见小组《可行性分析（研究）报告FAR》的第十部分：风险分析、分类及应对措施及实验六负责人（姜世慧）的实验报告对应内容。

3. 学习国标中的文档11《软件需求规格说明SRS》，了解文档的要求和结构及与其他相关文档 （07,08,12,17等）的关系。（文档见实验1的压缩文件）

1. **文档11《软件需求规格说明SRS》的要求和结构**

《软件需求规格说明》（SRS）描述对计算机软件配置项CSCI的需求，及确保每个要求得以满足的所使用的方法。《软件需求规格说明》（SRS）的主要包含范围、引用文件、需求合法性规定、需求可追踪性、尚未解决的问题、注解、附录八个部分。文档11《软件需求规格说明SRS》是软件项目的核心文档，它详细描述了软件的功能性需求、非功能性需求、业务规则、用户接口以及其他与软件相关的需求。这份文档为软件开发团队提供了明确的目标和方向，确保软件的开发能够满足用户的需求和期望。

1. **文档11《软件需求规格说明SRS》与其他相关文档（07,08,12,17等）的关系**

SRS可能要用IRS加以补充，为CSCI设计与合格性测试提供基础。CSCI需求是为了满足分配给该CSCI的系统需求所形成的软件需求。

**①文档07《系统/子系统需求规格说明》（SSS）**

《系统/子系统需求规格说明》（SSS）为一个系统或子系统指定需求和指定保证每个需求得到满足所使用的方法。与系统或子系统外部接口相关的需求可在SSS中或在该SSS引用到的一个或多个《接口需求规格说明》（IRS）中给出。这个SSS，可能还要用《接口需求规格说明》（IRS）加以补充，是构成系统或子系统设计与合格性测试的基础。贯穿本文的术语“系统”，如果适用的话，也可解释为“子系统”。所形成的文档应冠名为“系统需求规格说明”或“子系统需求规格说明”。

这份文档主要关注于系统和子系统的整体需求，包括系统的总体功能、业务结构、硬件和软件的需求等。它与SRS的关系在于，SRS中的软件需求是基于SSS中的系统和子系统需求来细化和展开的。因此，SSS为SRS提供了更高层次的需求背景和框架。

**②文档08《接口需求规格说明》（IRS）**

《接口需求规格说明》（IRS）描述为实现一个或多个系统、子系统、硬件配置项HWCI，计算机软件配置项CSCI、手工操作、其他系统部件之间的一个或多个接口，而强加在这些实体上的需求。这个IRS，还可以被用来补充《系统/子系统需求规格说明》（SSS）及《软件需求规格说明》（SRS），作为系统和CSCI设计与合格性测试的基础。

因此，可以说IRS为SRS提供了关于接口方面的详细补充和说明，确保软件在与其他组件交互时能够满足预定的需求。

**③文档12《数据需求说明》（DRD）**

《数据需求说明》（DRD）的编制目的是为了向整个开发时期提供关于被处理数据的描述和数据采集要求的技术信息。

在SRS中，可能需要引用DRD来描述软件处理的数据类型、数据结构、数据流程等需求。因此，DRD为SRS提供了关于数据方面的详细信息和要求，确保软件能够正确地处理和管理数据。

**④文档17《软件配置管理计划》（SCMP）**

《软件配置管理计划》（SCMP）说明在项目中如何实现配置管理，它描述了如何对软件项目中的各种配置项（如代码、文档、测试数据等）进行标识、控制、记录和报告。虽然SCMP并不直接描述软件需求，但它为SRS和其他需求文档提供了配置管理的支持和保障。通过有效的配置管理，可以确保SRS中的需求在软件开发过程中得到一致和准确的实现。

项目跟踪，建立能反映项目及小组每个人工作的进度、里程碑、工作量的跟踪图或表，每周更新。