**实验六 工作量估算，风险管理，软件需求规格说明SRS（1）**

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

实验内容：

1. 工作量估算：

ch3 习题12（小组讨论）。

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。

1. 风险管理

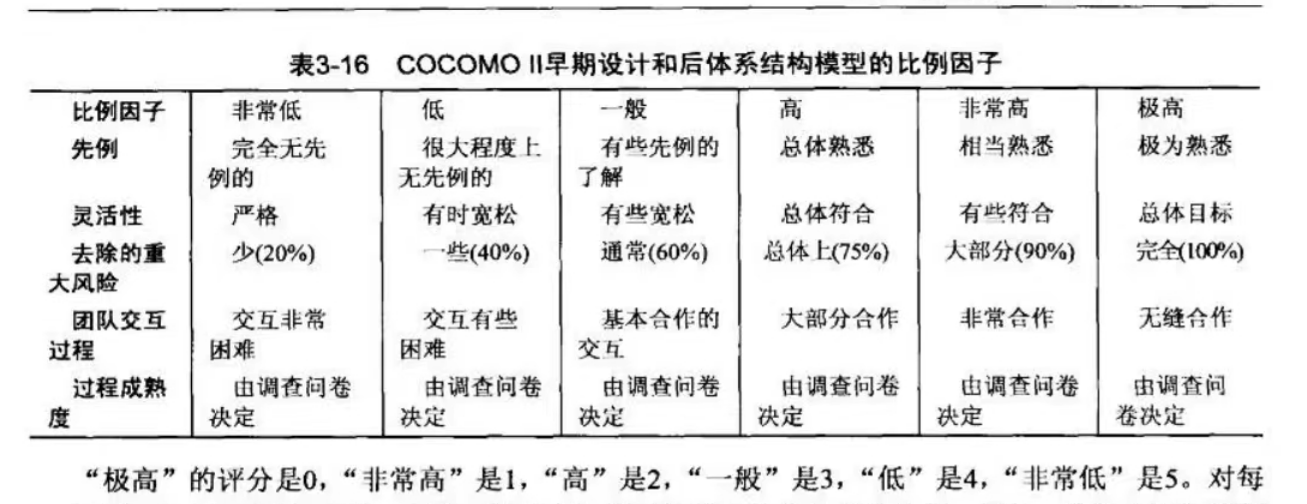
ch3 习题11 （小组讨论）

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）。

1. 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）

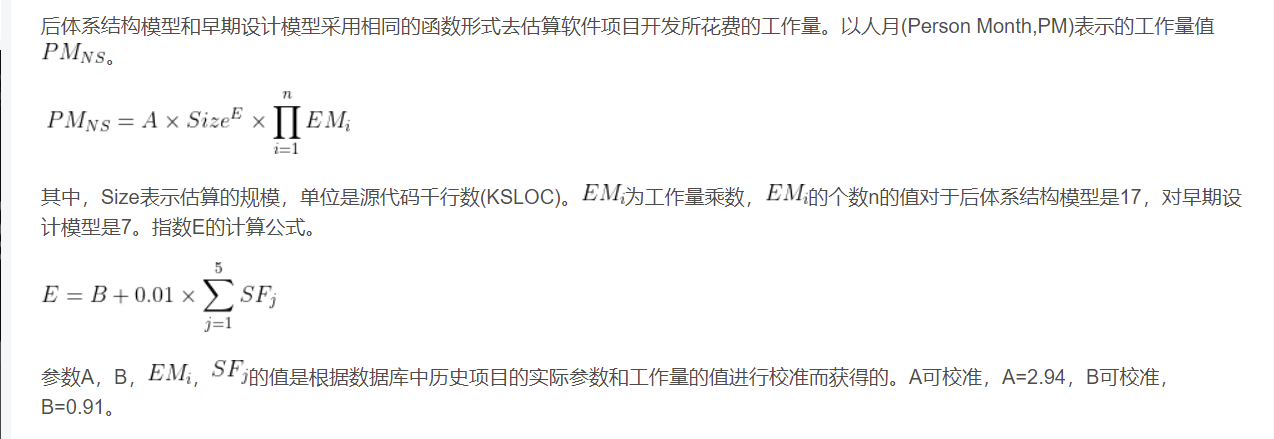
项目跟踪，建立能反映项目及小组每个人工作的进度、里程碑、工作量的跟踪图或表，每周更新。

**1.工作量估算**



COCOMOII模型估算方法：

在利用COCOMOII模型进行软件成本估算过程中，首先采用软件规模估算方法对项目的规模进行估算。再应用五个比例因子，通过相关计算，将规模转化为工作量，并通过十七个成本驱动因子对工作量进行调整。最后，采用进度计算公式，计算出开发该项目所需要的进度以及人数。



SF代表指数比例因子：

先例性(PREC)：表示以前是否开发过类似项目。

开发灵活性(FLEX)：表示软件性能与已经建立的需求和外部接口规范的一致程度。

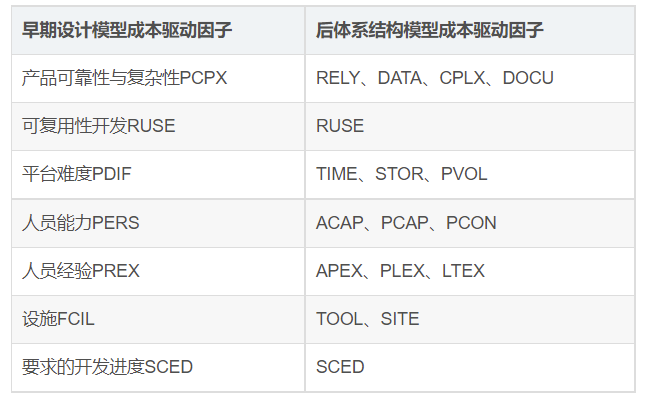
体系结构/风险化解(RESL)：通过风险管理衡量项目的风险及建立体系结构的工作量。

团队凝聚力(RERM)：衡量项目相关人员的管理状况。

过程成熟度(PMAT)：衡量项目过程的规范程度，主要围绕SEI的CMM而进行



早期设计模型主要用在项目早期阶段中，由于此时对待开发产品的规模、目标平台性质、项目相关人员状态以及采用过程的想象说明等还不够了解，所以将后体系结构模型中的17个成本驱动因子综合成7个成本驱动因子。



计算自己项目的初始工作量：

PREC ：低，分数为4.96

FLEX：非常低的，分数为5.07

RESL ：高，分数为2.83

TEAM：高，分数为2.19

PMAT：低，分数为6.24

计算SF=0.91 + 0.01 × (1.24 + 5.07 + 2.83 + 2.19 + 6.24) = 0.91 + 0.01 × 21.28

= 0.91 + 0.2128

=1.1228

系统包括1KSLOC

成本驱动因子如下：

PCPX (产品的可靠性和复杂度) 非常高 1

PDIF (平台难度) 高 2

PERS (人员的能力) 一般 3

PREX (人员的经验) 一般 3

PM=2.94\*1^1.1228=2.94

比例因子：1+2+3+3=9 比例指数：1.10

PM=2.94^1.10（人月）

**2.风险管理**

存在的风险

1、未经权威部门确认的功能标准、开发规范以及质量技术标准，均可能导致软件无法达到预期标准，从而引起质量风险。

2、在理解项目标准及范围等问题上，企业管理层、项目组以及技术性人员的接不一致，导致计划与资金安排有所改变，因而极易引发风险。

3、潜在的维护、验证、接口、实现以及设计等环节出现的问题，存在技术空白及未知领域，为软件开发工作带来较大的风险。系统性能风险，项目属于多用户并发的应用系统，系统对性能要求很高，需要关注项目的性能风险。

4、来自于外包项目组、客户、国家政策以及市场等方面的变化及压力，这类风险具有明显的不可控特点，一旦遭遇，应谨慎对待，及时制定解决策略。

如何应对风险

1、出台合理的软件开发模式与相关规程，确保开发工作合理、有序进行，并符合国家出台的相关标准及要求。

2、对于项目组全体成员的开发行为进行严格规范，加强小组成员之间的交流与互动，以免由于沟通与交流不当，引发软件开发风险。项目建设之初就和项目各干系方约定好沟通的渠道和方式、项目建设过程中多和项目各干系方交流和沟通、注意培养和锻炼自身的沟通技巧。

3、定期开展业务和技术交流大会，引导技术人员摒除过于落后、陈旧的工作思想，通过引进先进的技术、设备与验证方式，明确技术人员的预期发展目标，令其不断的改进自我、完善自我，提升技术及设备的质量及效果。

4、对开发所用的方法及技术进行客观、合理的评价，避免由于无法把握技术而引发风险。在进行项目开发之前先设计和搭建出系统的基础架构并进行性能测试，确保架构符合性能指标后再进行后续工作。

5、建立完善的风险应对程序与管理计划，如此一来，才能确保在发生风险的时候，能够快速、合理、技术的作出反映，并通过制定适宜的策略，对风险进行专业性处理。

**3.软件需求规格说明SRS**

1.《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。涉及该CSCI外部接口的需求可在本SRS中给出：或在本SRS引用的一个或多个《接口需求规格说明》(IRS)中给出。

2.这个SRS，可能还要用IRS加以补充，是CSCI设计与合格性测试的基础。

SRS(Software Requirements Specification)

软件需求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。包含硬件、功能、性能、输入输出、接口需求、警示信息、保密安全、数据与数据库、文档和法规的要求等等。

《软件需求规格说明书》的目的是作为用户和软件开发人员达成的技术协议书，作为着手进行设计工作的基础和依据，系统开发完成以后，为产品的验收提供了依据。由于用户要能看得懂，并且还能发现和指出其中的错误，这对于保证软件系统的质量有很大的作用。

需求说明书主要具备以下几个功能：

1、便于与用户、开发人员达成共识：界定清楚软件的边界、功能、性能要求等，并据此与用户、开发人员达成共识；

2、软件设计的依据：是后续设计、开发工作的依据；

3、软件验收测试的依据：是最终测试、验收的依据；

4、需求管控：可以用于鉴别用户需求是否超出边界或是否变更、新增需求。

对于软件设计而言，有两个域：问题域，和设计域。需求规格和后期的设计文档很不一样，它是工作在问题域的。问题域不应当出现诸如模块分割这种明显地设计的痕迹。但是如果用户的项目要求要多地部署，那么他仍然要进行分系统描述，进而描述分系统的接口。

问题域描述可以以用例图的方式为主体，也可以用原型描述的方式进行。推荐用原型描述的方法，因为它的内容可以与后期的设计文档（比如测试大纲，一份好的需求规格说明书可以直接导出一份全面的测试大纲）更好的衔接。写软件需求规格说明书要非常注意和后期设计工作的衔接，否则它的应用价值就不大。

软件需求规格说明(SRS)

1范围

1.1标识

1.2系统概述

1.3文档概述

1.4基线

2引用文件

3需求

3.1所需的状态和方式

3.2需求概述

3.2.1目标

3.2.2运行环境

3.2.3用户的特点

3.2.4关键点

3.2.5约束条件

3.3需求规格

3.3.1软件系统总体功能/对象结构

3.3.2软件子系统功能/对象结构

3.3.3描述约定

3.4CSCI能力需求

3.5CSCI外部接口需求

3.5.1接口标识和接口图

3.6CSCI内部接口需求

3.7CSCI内部数据需求

3.8适应性需求

3.9保密性需求

3.10保密性和私密性需求

3.11CSCI环境需求

3.12计算机资源需求

3.12.1计算机硬件需求

3.12.2计算机硬件资源利用需求

3.12.3计算机软件需求

3.12.4计算机通信需求

3.13软件质量因素

3.14设计和实现的约束

3.15数据

3.16操作

3.17故障处理

3.18算法说明

3.19有关人员需求

3.20有关培训需求

3.21有关后勤需求

3.22其他需求

3.23包装需求

3.24需求的优先次序和关键程度

4合格性规定

5需求可追踪性

6尚未解决的问题

7注解

附录