**北京邮电大学计算机学院（国家示范性软件学院）**

**多源灾情数据管理服务系统**

**经济决策书**

**课程名称： 软件工程经济学**

**完成人：**

**组号：**

**姓名：\_**

**学号：**

**指导教师：**

**日 期： 2023年 10月 27日**

目录

[一、 背景与目的 4](#_Toc153802011)

[项目背景 4](#_Toc153802012)

[迭代计划 4](#_Toc153802013)

[经济决策 6](#_Toc153802014)

[二、 项目初始的合同阶段 7](#_Toc153802015)

[1.项目成本计算 7](#_Toc153802016)

[功能点分析法（Function Point Analysis） 7](#_Toc153802017)

[COCOMO模型估算（基于KLOC） 9](#_Toc153802018)

[Walston-Felix模型估算（基于KLOC） 10](#_Toc153802019)

[成本估算 11](#_Toc153802020)

[2.项目报价 12](#_Toc153802021)

[3.付款方案及现金流图 12](#_Toc153802022)

[付款方案 12](#_Toc153802023)

[现金流图 13](#_Toc153802024)

[三、 项目需求分析阶段 13](#_Toc153802025)

[1.任务分解 15](#_Toc153802026)

[2.成本效益分析 16](#_Toc153802027)

[Fast Story Point Estimation(T-shirt) 16](#_Toc153802028)

[成本计算 17](#_Toc153802029)

[项目收益 18](#_Toc153802030)

[3.项目可行性分析 19](#_Toc153802031)

[经济可行性 19](#_Toc153802032)

[技术可行性 19](#_Toc153802033)

[组织可行性 19](#_Toc153802034)

[社会可行性 19](#_Toc153802035)

[资源可行性 19](#_Toc153802036)

[四、项目概要设计阶段 20](#_Toc153802037)

[1.成本估算和资源分配 20](#_Toc153802038)

[Bottom-up估算方法 21](#_Toc153802039)

[2.项目进度计划制定 22](#_Toc153802040)

[3.项目风险评估和管理 24](#_Toc153802041)

[需求变更风险 24](#_Toc153802042)

[进度风险 24](#_Toc153802043)

[技术风险 25](#_Toc153802044)

[人力资源风险 25](#_Toc153802045)

[工具风险 25](#_Toc153802046)

[质量风险 25](#_Toc153802047)

[安全风险 25](#_Toc153802048)

[成本风险 26](#_Toc153802049)

[沟通风险 26](#_Toc153802050)

[五、项目编码阶段 27](#_Toc153802051)

[1.项目进度监控 28](#_Toc153802052)

[赚取值管理 28](#_Toc153802053)

[第一次迭代 29](#_Toc153802054)

[第二次迭代 31](#_Toc153802055)

[第三次迭代 33](#_Toc153802056)

[六、项目测试阶段 35](#_Toc153802057)

[1.软件测试 36](#_Toc153802058)

[2.修复成本 37](#_Toc153802059)

[3.系统建议 37](#_Toc153802060)

[性能提升 37](#_Toc153802061)

[安全性提升 38](#_Toc153802062)

**项目经济决策书**

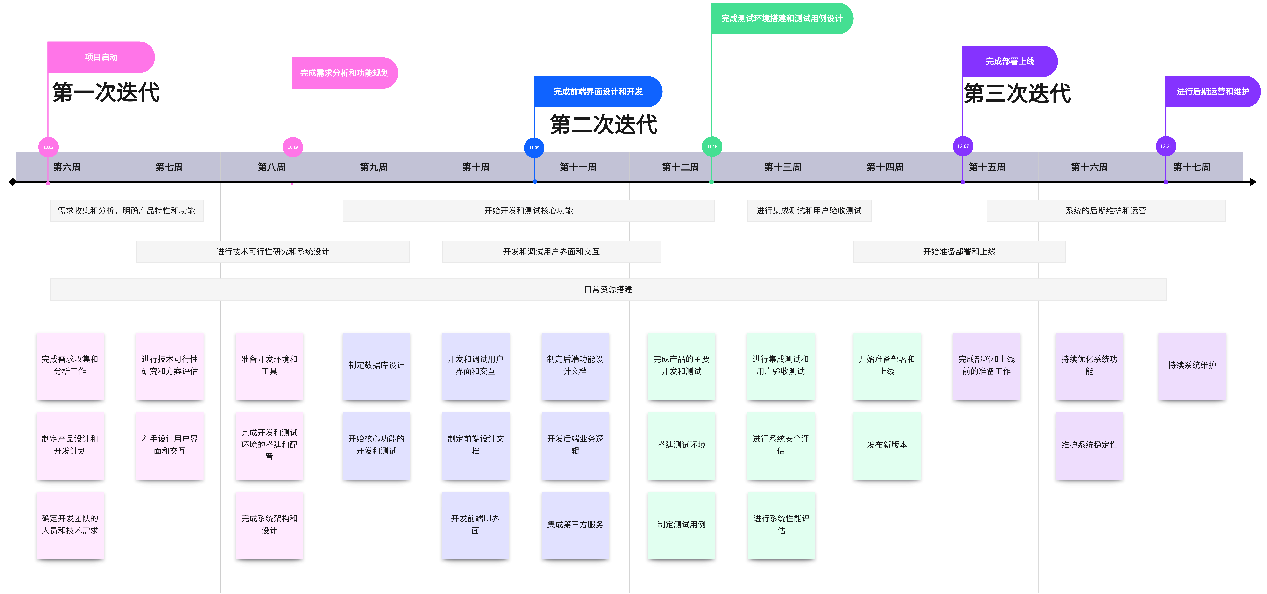
# 背景与目的

## 项目背景

随着自然灾害和突发事件频发，对于快速、准确、多源、全面的灾情信息获取与管理需求日益迫切。本项目是国家震情应急管理系统的一部分，针对震后灾情获取缓慢且碎片化，针对公众涉灾信息数据异构、多维、数据格式差异大、部分数据维度缺失导致的数据无法得到综合利用的现状，研究基于异构公众涉灾信息，进行一体化编码，利用开放接口技术和实时动态管理技术，对灾情信息进行管理，实现必要的可视化，实现灾情数据统一管理和高效合理利用。

本文将深入探讨MSHD-2.0项目，并充分运用软件工程经济学的理论框架和工具，以经济利益最大化为导向，对项目在不同阶段的执行情况进行全面分析。在这个过程中，将对预算分配、资源分配和项目计划制定等方面展开细致考量。通过对项目各个阶段的经济实施情况进行深入评估，旨在为项目管理人员提供重要的决策支持，帮助其更有效地管理项目，并在经济层面做出明智的决策，以最大化项目的经济效益。

## 迭代计划

****

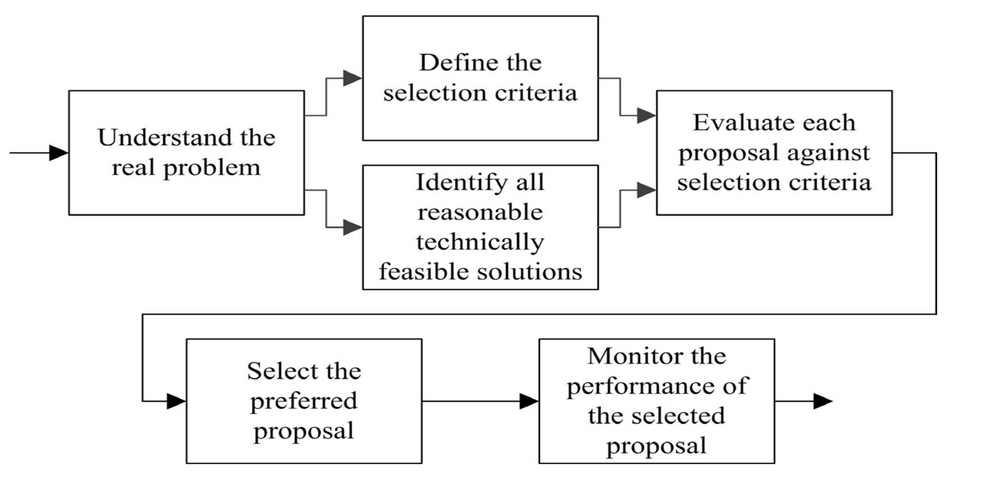
本项目计划于2023年10月2日正式启动，预计2023年12月21日完成所有迭代周期并交付最终版本，总共耗时约11周。具体的项目进度计划如下表所示：

|  |  |  |
| --- | --- | --- |
| **迭代周期** | **起止日期** | **主要内容** |
| 第一次迭代 | 10月2日-11月9日 | 完成系统的需求收集和分析工作，完成开发和测试环境的搭建和配置 ，完成系统架构和设计，制定数据库设计，开始核心功能的开发和测试，开发和调试用户界面和交互。此阶段的成果有需求分析文档、概要设计文档、详细设计文档、代码说明文档，以及可运行的前端界面和部分系统核心功能。 |
| 第二次迭代 | 11月9日-12月7日 | 开发后端业务逻辑，集成第三方服务，完成产品的主要开发和测试，进行集成测试和用户验收测试，进行系统安全评估，开始准备部署和上线。此阶段得到最终可运行并通过多项测试后的系统，并开始准备线上部署发布。 |
| 第三次迭代 | 12月7日-12月21日 | 完成部署和上线前的准备工作，持续优化系统功能，维护系统稳定性。此阶段主要是完成系统的部署和上线发布，以及后期系统的维护和运营。 |

本项目迭代计划的目标是逐步完善系统，确保其具备基本功能、性能和安全性，然后逐渐引入高级功能和增强扩展性。通过迭代增量式的开发模式，每次迭代都应与项目干系人保持密切合作。通过三次迭代，逐步构建出高质量、符合实际需求的国家震情应急管理系统，提高其可用性和用户满意度。

## 经济决策

在抽象的层面上，无论是财务决策还是其他类型的决策，都是关于最大化价值。总价值最大化的替代方案应始终被选择。价值比较的财务基础是比较两个或更多现金流。



*经济决策流程图*

在本项目的各个阶段中，将使用以下几种分析方法和工具来辅助项目管理者和团队对软件工程项目进行合理评估和决策，确保项目在经济上是可行的，并且能够为组织带来良好的投资回报。

**成本效益分析：**这涉及评估项目所需的成本与预期获得的收益之间的关系。这种分析可以帮助确定投资回报率、成本效益比等关键指标，从而评估项目的经济可行性。

**净现值分析（NPV）：**使用净现值方法，将项目未来现金流的折现值与初始投资进行比较。如果净现值为正，则项目通常被视为可行的投资。

**投资回收期（IRR）：**投资回收期指的是项目产生的现金流量可以收回初始投资的时间。通常，较短的投资回收期意味着项目更具吸引力。

**成本估算和预算控制：**在项目早期，进行成本估算以预测和规划项目的开销，并在项目执行过程中进行预算控制以确保在预算范围内。

**风险评估：**对项目风险进行评估，识别可能的风险并采取相应的措施来管理和减少风险。风险对项目的经济效益有重要影响，因此必须进行合理评估和处理。

**ROI（投资回报率）分析：** 投资回报率是衡量投资效益的指标，计算方法是将投资的收益与投资的成本进行比较。

**边际效益分析：**考虑到项目增量变化所带来的收益和成本。边际效益分析有助于确定是否值得继续进行某项投资。

# 项目初始的合同阶段

## 1.项目成本计算

### 功能点分析法（Function Point Analysis）

功能点估算法是软件项目管理众多方法中比较有技术含量的一个,也是最实用的一个。基于中国软件行业基准数据库的预算成本估算，首先对规模进行估算，规模采用功能点进行计数，方法采用快速功能点方法。考虑到原始需求书写比较粗略，因此预算小组决定采用快速功能点中的预估功能点的方法进行规模估算，即只需要对数据文件进行计数即可。

功能点估算法将功能点分为以下 5 类：

1. ILF:Internal Logical File 内部逻辑文件

内部逻辑文件是指一组以用户角度识别的、在应用程序边界内且被维护的逻辑相关数据或控制信息。ILF 的主要目的是通过应用程序的一个或多个基本处理过程来维护数据。

1. EIF: External Interface File 外部接口文件

外部接口文件是指一组在应用程序边界内被查询,但在其他应用程序中被维护的、以用户角度来识别的、逻辑上相关的数据。因此，一个应用程序中的 EIF必然是其他应用程序中的ILF。EIF的主要目的是为边界内的应用程序提供一个或多个通过基础操作过程来引用的一组数据或信息。

1. EI: External Input 外部输入

EI是处理来自应用程序边界外部的一组数据输入,它的主要目的是维护个或多个ILF,以及/或者更改系统的行为。

1. EO: External Output 外部输出

EO是输送数据到应用程序边界外部的过程。它的主要目的是通过逻辑处理过程向用户呈现信息。该处理过程必须包含至少一个数学公式或计算方法，或生成派生数据。一个EO也可以维护一个或多个ILF，并/或改变系统行为。

1. EQ: External Inquiry 外部查询

EQ是向应用程序边界外发送数据基本处理的过程。其主要目的是从ILF或EIF中通过恢复数据信息来向用户呈现。该处理逻辑不包括任何数学公式或计算方法,也不会生成任何派生数据。EQ不会维护任何一个ILF，也不会改变应用程序的系统行为。

**内部逻辑文件与外部接口文件**

因为本系统的主要功能是多源社会灾情数据通过接口输入到多源灾情数据管理服务系统平台，进行一体化编码，然后将编码后的数据输入到虚拟化管理系统。所以系统中维护的逻辑相关数据主要是用户信息和一体化编码后的基本灾情信息，系统还需要从FTP云服务器中读取多源异构数据并将整理后的数据通过GIS服务可视化展示。

下图是本系统的系统周境图，通过此图可以确定出本系统的ILF为基本震情信息和用户；EIF为FTP服务器和GIS服务。

多源灾情数据

管理服务系统

系统用户

系统管理员

数据管理员

云服务器

系统边界

数据来源平台

GIS服务

**外部输入、外部输出和外部查询**

本系统从多个数据来源获取灾情数据，这个数据是没有标准的，需要与数据源商量确定数据内容和数据格式，这些数据放在指定的云服务器上（FTP服务器）。本系统作为多源灾情数据的管理服务系统的外部输入主要是从云服务器中读取的36位一体化震情编码以及震情信息的相关载体。

为了实现数据的可视化展示，系统还需要从外部GIS服务系统中读取详细的地理位置信息，来结合系统内的震情信息呈现给用户直观的可视化数据以及统计分析图。还要根据用户请求，抽取合适的灾情数据的数据编码，实现用户的个性化数据展示。

**IFPUG估算方法估算规模总和**

IFPUG（International Function Point Users Group）功能点估算方法是一种常用的软件工作量估算方法，用于衡量软件系统的功能规模。它基于对软件系统功能的分析和评估，以功能点（Function Points，FP）为单位来度量和估算软件的大小和复杂性。

|  |  |  |  |
| --- | --- | --- | --- |
|  | 复杂度权重因素 | | |
| 项 | 简单(低) | 一般 (中) | 复杂 (高) |
| 外部输入 | 3 | 4 | 6 |
| 外部输出 | 4 | 5 | 7 |
| 外部查询 | 3 | 4 | 6 |
| 外部接口文件 | 5 | 7 | 10 |
| 内部逻辑文件 | 7 | 10 | 15 |

*功能计数项的复杂度等级*

外部输入：3项

外部输出：6项

外部查询：6项

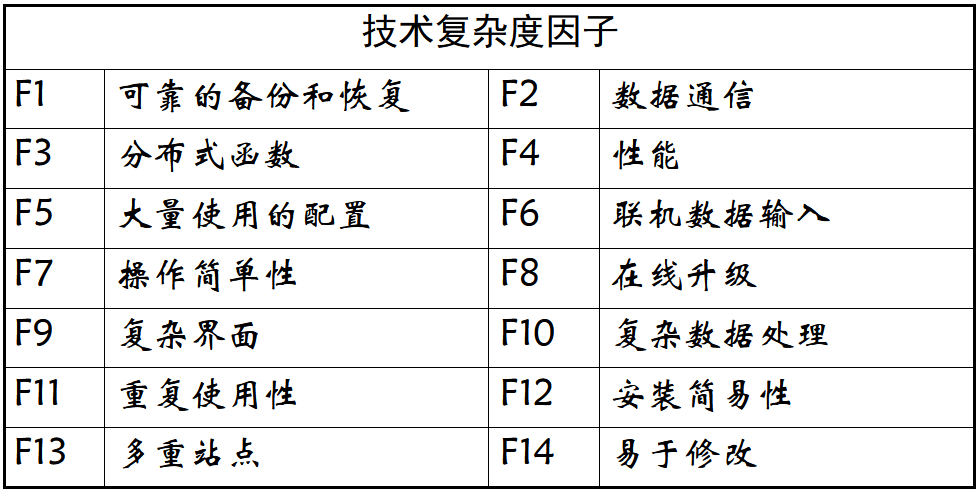
外部接口文件：2项

内部逻辑文件：6项

|  |  |  |  |
| --- | --- | --- | --- |
|  | 功能点 | | |
| 项 | 简单 | 一般 | 复杂 |
| 外部输入 | 1\*3 | 1\*4 | 1\*6 |
| 外部输出 | 2\*4 | 4\*5 | 0\*7 |
| 外部查询 | 6\*3 | 0\*4 | 1\*6 |
| 外部接口文件 | 1\*5 | 1\*7 | 0\*10 |
| 内部逻辑文件 | 3\*7 | 2\*10 | 1\*15 |
| 总计 |  |  |  |
| UFC | 133 |  |  |

用功能点估算法计算软件项目功能点时会用到调整因子(或称调整系数)功能点的调整系数是通过通用系统特性及其影响程度来评定的,对每个常规系统特性的评估由其影响程度 (DI)而定，分为 0-5 级。根据下表分析各技术复杂度因子的影响程度，再据此求出系统的VAF。

VAF = 0.65 + 0.01 \*(sum (Fi)) = 0.65+0.51 = 1.16



最后得出项目的FP = UFC \* VAF = 154.28

### COCOMO模型估算（基于KLOC）

COCOMO（Constructive Cost Model）是一种软件成本估算模型，用于估计软件开发项目的成本、工作量和时间。它是根据代码行数（KLOC，即千行源代码）来进行估算的一种模型。

基本COCOMO经验估算模型（COCOMO I）：这个阶段基于代码行数（KLOC）来估算软件开发的工作量和成本。它分为有机型、半混合型和嵌入式型三种子模型，根据项目的特征和复杂度选择相应的模型。

|  |  |  |  |
| --- | --- | --- | --- |
| 软件类型 | a | b | 适用范围 |
| 有机 | 2.4 | 1.05 | 各类应⽤程序 |
| 半有机 | 3.0 | 1.12 | 各类实⽤程序、编译程序等 |
| 嵌入式 | 3.6 | 1.2 | 实时处理、控制程序、操作系统 |

*Effort* 表示工作量（通常以人月或人年为单位）。

*KLOC*是以千行代码为单位的软件规模。

*A*和*B*是与项目类型和复杂性相关的常数，不同级别的COCOMO模型会有不同的取值。

*EAF*是调整因子，称为“成本驱动因子”（Effort Adjustment Factor），用于考虑项目特定的环境、工具、团队能力和其他影响因素。计算EAF的过程中，易知EAF与以下因素有关：1.软件可靠性、2.软件复杂性、3.数据库的规模。4.程

序执行时间、5.程序占用内存的大小、6.软件开发环境的变化、7.软件开发环境的响应速度。8.分析员的能力、9.程序员的能力、10.有关应用领域的经验、11.开发环境的经验、12.程序设计语言的经验13.软件开发方法的能力，14.软件工具的质量和数量、15.软件开发的进度要求。这里乘法因子取值为1.09。

考虑到本项目的工作量较大、规模较大且涉及到新平台的搭建，这里A、B的值估算为2.4和1.05。又考虑到开发团队的技能水平和经验有限并且开发工具和支持系统的可用性和质量不高，鉴于项目开发有较大的风险和挑战，这里的EAF选择为1.05。

最终根据公式计算出*Effort*的值约等于23.2211人月。

### Walston-Felix模型估算（基于KLOC）

，KLOC是源代码行数，E是工作量（以PM计）

，S是人员需要量(以人计)

，DOC是文档数量(以页计)

预估最后的源代码行数约为8000行代码，最终的工作量E约等于34.4997人月。人员需要量大概为5人，文档数量约为400页。

### 成本估算

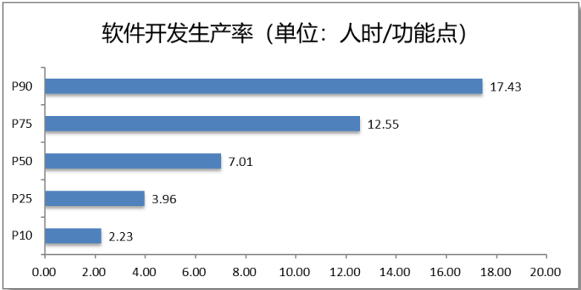
本文主要参照“中国软件行业基准数据报告（SSM-BK-202309）”关于北京地区的基准数据来进行成本估算，具体使用到的基准数据见下表。

|  |  |
| --- | --- |
| 基准人月费率（软件开发） | 3.20万元 |
| 基准人月费率（软件运维） | 2.60万元 |
| 功能点单价 | 1313.10元/功能点 |
| 全行业生产率 | 7.14人时/功能点 |
| 维护型开发生产率 | 3.81人时/功能点 |

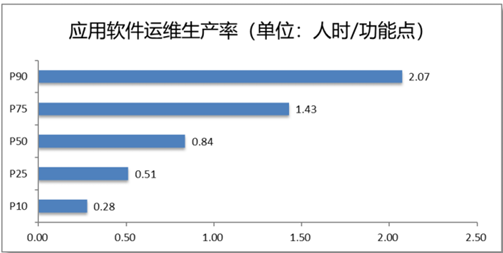
**功能点单价：**

北京地区功能点单价基准为1303.01元/功能点，费用包含软件开发的直接人力成本、间接人力成本、间接非人力成本及合理利润（含税），但不包括直接非人力成本。其他地区功能点单价基准可参照该地区与北京地区软件开发人月费率对应关系折算。

**全行业软件开发生产率基准数据**：



**应用软件生产率基准数据**：



**软件开发费用估算**：

*P*:软件开发费用，单位为万元；

*AE*:**调整后工作量**，单位为人时,取值为**1,101.5592人时**；

*HM2*:人月折算系数，单位为人时/人月，取值为174(21.75天\*8人时/天=174人时);

经过计算后可以得出本项目至少**需要6人全职开发一个月**。

*F2*:人月基准单价，单位为万元/人月，取值为3.20万元；

*DNC*:直接非人力成本，单位为元;直接非人力成本由委托方根据实际情况进行分析估算；

综合各方面因素计算出本项目的**软件开发费用**大约为**20.258万元**。

除此之外，还要计算后续的**运营维护的费用**：

P:软件开发费用，单位为万元；

AE:调整后**工作量**，单位为人时,取值为**587.8068人时**；

HM2:人月折算系数，单位为人时/人月，取值为174(21.75天\*8人时/天=174人时);

F2:人月基准单价，单位为万元/人月，取值为2.60万元；

DNC:直接非人力成本，单位为元;直接非人力成本由委托方根据实际情况进行分析估算；

结合上面的软件开发费用，综合各方面因素计算出本项目的**软件维护费用**为**8.78332万元**，**整个项目**的成本至少为**29.04132万元**。

## 2.项目报价

计算出软件开发费用之后，再按照50%的投资回报率计算项目报价。

**计算预期利润**：利润等于投资回报率乘以成本。在这种情况下，预期利润 = 20.258 万元 × 50% = 10.129 万元。

**计算报价**：报价等于成本加上预期利润。在这里，项目报价 = 20.258 万元 + 10.129 万元 = 30.387 万元。

## 3.付款方案及现金流图

### 付款方案

现在以项目的合同金额为报价金额，按照年利率8%，项目开发周期为1年，共设计出两种付款方案：

1）年初40%，年中20%，年底40%；

2）年初50%，年末50%。现采用净现值方法进行目标决策。

***净现值****=未来报酬总现值-建设投资总额*

**第一种付款方案：**年初40%，年中20%，年底40%

计算每个时间点的现金流量（将合同金额拆分成不同时间段的现金流量）。

年初40%：30.387万元 × 40% = 12.1548 万元

年中20%：30.387万元 × 20% = 6.0774 万元

年底40%：30.387万元 × 40% = 12.1548 万元

**第二种付款方案：**年初50%，年末50%

计算每个时间点的现金流量（将合同金额拆分成不同时间段的现金流量）。

年初50%：30.387万元 × 50% = 15.1935 万元

年末50%：30.387万元 × 50% = 15.1935 万元

使用年利率8%计算每个现金流量的净现值（NPV）。公式为：

其中，是每年的现金流量，r是年利率，t是时间。

计算两种方案的净现值（NPV）：

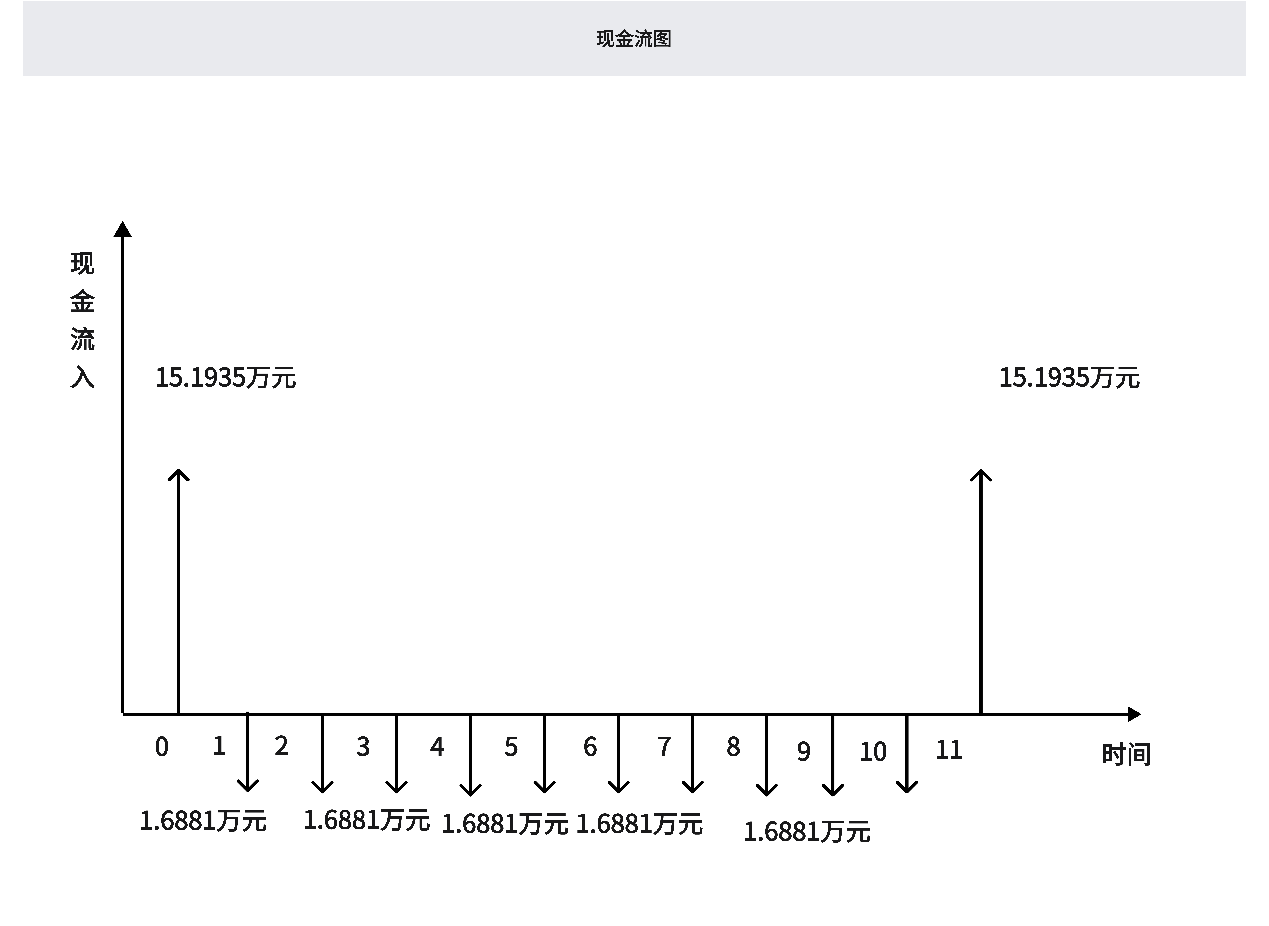
**第一种付款方案**：

**第二种付款方案：**

两种方案的NPV值都为正值，表明未来现金流量的现值总和大于投资成本，这意味着项目有能力提供超过预期的回报。其中方案2的NPV值更大一些，因此，方案2相对于方案1而言更有利一些，其净现值较大，具有较高的盈利潜力，其现金流量超过了预期成本，为投资者带来了正面的经济效益。

### 现金流图

**现金流图：**现金流图是一种可视化工具，用于展示在一段时间内（通常是项目周期）内现金流量的情况。它显示了现金流入和流出的情况，帮助人们更直观地了解资金的流动情况和变化。



# 项目需求分析阶段

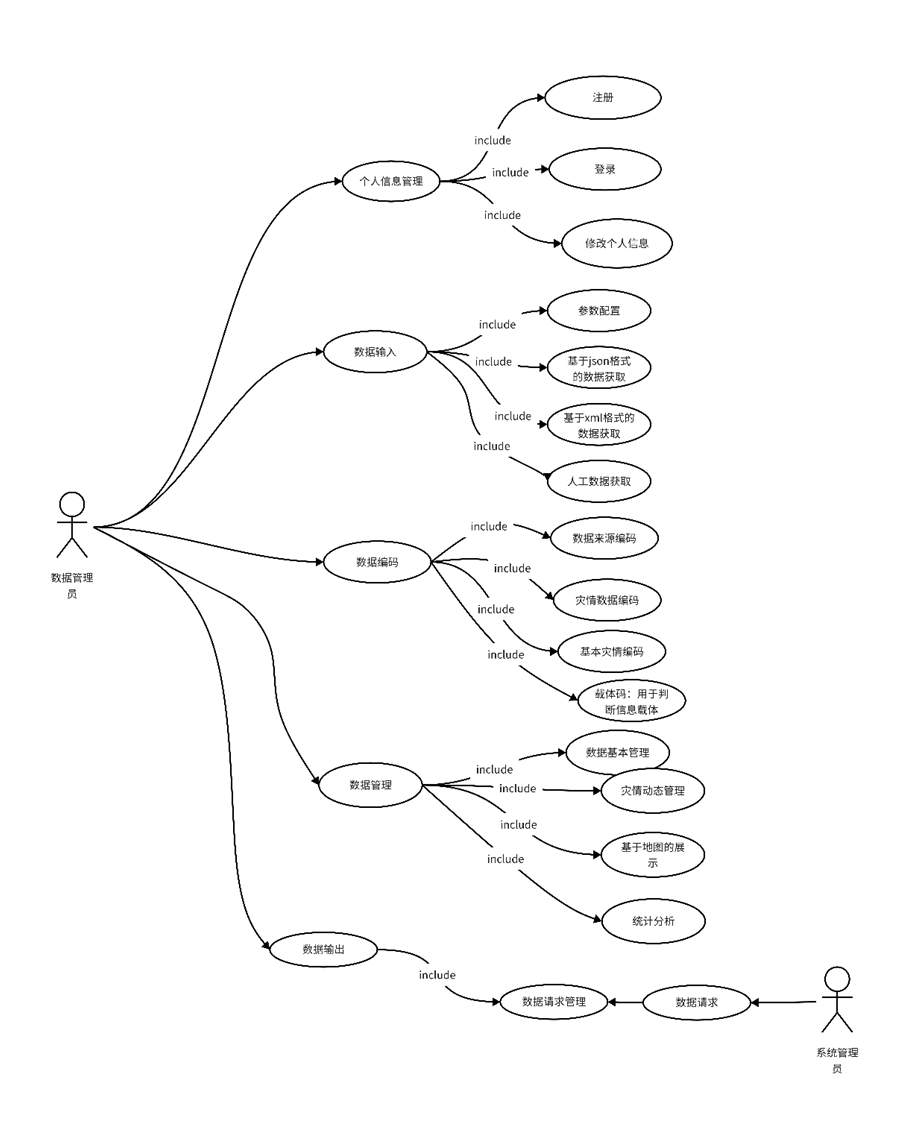
在项目初始合同阶段，我运用功能点分析和COCOMO模型等方法估算了本项目的成本，并基于50%的回报率计算出项目的合同金额。这个合同金额被用来选择最适合的付款方案，并且我们绘制了相应的现金流图。

然而，在项目的实际开发进入需求分析阶段时，我意识到具体的需求将会逐步完善、细致。这个阶段将会为项目提供更详尽、更精确的需求信息。因此，这里需要重新进行项目的成本效益分析，以确保我们在后续开发阶段制定出更加可靠的经济决策方案。

在需求分析阶段，我们对项目的需求进行了更深入的挖掘和理解。确定和澄清了用户最终的期望和需求，以便项目团队能够精确地把握需求细节；与利益相关者和用户持续沟通，以便捕捉所有可能的需求变化和补充，确保成本估算的准确性和全面性；进一步细化项目的功能和特性，明确项目交付物的范围和规格。

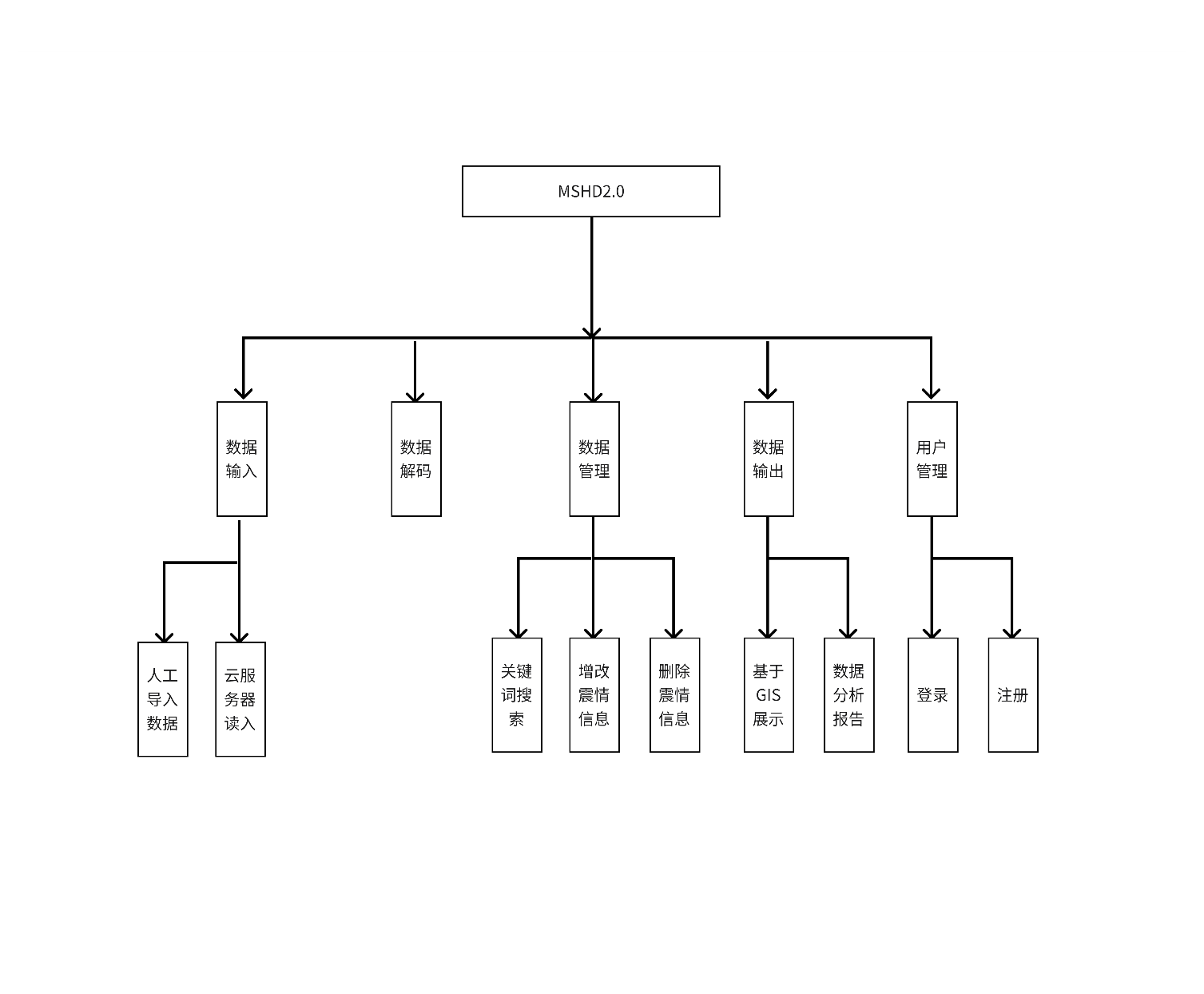
这些详细的需求信息有助于我们重新评估项目的经济可行性，确保我们的成本估算和经济决策基于更加准确和全面的信息。为我们提供了更有利的基础，以便在后续的开发阶段制定出更有效的经济决策，并最大程度地实现项目的经济目标。

## 1.任务分解



在项目的需求分析阶段，我们小组根据项目介绍绘制了项目用例图，明确了项目的基本功能需求。通过该图，我将本项目划分出以下五个主要任务：

1. 建立系统中基本的用户管理系统。
2. 通过外部接口实现一体化震情码的读取功能。
3. 确保系统能够快速、准确地从36位震情码中解构出震情信息。
4. 在系统中实现对震情信息的管理，包括基本的增删改查操作。
5. 要求系统能够将存储在其中的震情信息进行整合、分析，并生成可视化的数据报告。



*任务分解图*

这些任务在需求分析阶段体现了经济决策的内容，构成了系统功能的基本支柱，并为随后的开发阶段提供了基础。

## 2.成本效益分析

### Fast Story Point Estimation(T-shirt)

**"Fast Story Point Estimation"** 是敏捷开发中用于快速估算用户故事（User Stories）复杂度或工作量的方法。这种方法旨在以快速、直观的方式对用户故事进行估算，通常在规划会议或团队会议中使用。

这种估算方法通常使用相对单位（比如故事点）而不是具体的时间单位（比如小时）来评估任务的难度或复杂度。这有助于避免固定在具体时间上，而是关注任务的相对大小和难度。

接下来我将根据本项目的任务分解图和基准估算参考图来快速估算本项目的成本，并制定成本效益分析计划，比较项目成本和收益。



*基准估算参考*

这里我将任务分成不同的大小类别，比如 S、M、B、L、XL 等，以代表不同的相对大小和复杂度。这种方法更加直观，有助于快速估算任务的相对大小。

|  |  |
| --- | --- |
| 任务 | 点数 |
| 数据输入（数据接口）总和：10 | |
| Excel (cvs) 格式 | 1 |
| Sql格式 | 1 |
| Txt格式 | 1 |
| Json格式 | 1 |
| XML格式 | 1 |
| 图片信息 | 2 |
| 音频、视频信息 | 3 |
| 数据解码总和：4 | |
| 文字信息 | 1 |
| 图片、音频、视频信息 | 3 |
| 数据管理总和：6 | |
| 增加震情信息 | 2 |
| 修改震情信息 | 1 |
| 删除震情信息 | 1 |
| 根据关键词搜索 | 2 |
| 数据输出总和：7 | |
| 灾情信息列表可视化 | 2 |
| GIS可视化 | 5 |
| 用户管理总和：5 | |
| 登录 | 2 |
| 注册 | 3 |
| 总任务点数：32 | |

### 成本计算

软件研发成本通常分为直接成本和间接成本，同时考虑到软件行业的特性，直接成本和间接成本又可分为人力成本和非人力成本。接下来我将根据《软件开发成本度量规范》来估算本项目的开发成本。

**工作量 = 软件规模 \* 软件因素调整因子 \* 开发因素调整因子 \* 生产率**

软件因素包括规模、质量要求、应用类型、业务领域等。

开发因素包括采用技术、过程、团队经验等。

|  |  |
| --- | --- |
|  | 预算规模变更因子 |
| GSC调警因子 | 1.00 |
| 需求变更调整因子值 | 1.50 |
| 应用类型调整因子 | 1.00 |
| 质量要求调整因子 | 0.975 |
| 业务领域调整因子 | 0.93 |
| 开发语言调整因子 | 1.00 |
| 团队经验调整因子 | 1.00 |
| 工作量调整因子 | 0.91 |
| 人月费率 | 1.95 |

**软件研发成本 = 工作量 \* 人月费率 + 直接非人力成本 = 12.872223万元**

### 项目收益

因为本项目属于国家震情应急管理系统的一部分，项目本身并不具有商业盈利目的。虽然本项目的收益不以商业盈利为目标，但其社会、政府和经济效益对于国家和公众的利益都是至关重要的。因此，在评估本项目收益时综合考虑了这些非财务性的方面。

**社会效益：**

公共安全提升： 减少灾害损失、保障公众生命财产安全。

救灾效率提高： 快速获取和分析灾情信息，有助于快速响应和救援。

社会稳定维护： 促进社会稳定，减少灾后混乱和恐慌。

**政府效益：**

政府治理能力提升： 提高政府在应急管理领域的能力和效率。

舆情管理： 及时有效地管理灾后舆情，维护政府形象和信任度。

**经济效益：**

减少灾害损失： 通过快速、准确的灾情管理，减少灾害带来的经济损失。

资源合理利用： 灾情信息的统一管理和利用，有助于资源合理配置和利用。

**其他方面：**

技术积累： 项目实施过程中的技术积累和创新。

国家形象提升： 提升国家在灾害管理领域的形象和声誉。

## 3.项目可行性分析

在项目的可行性方面，我着重考虑以下了几个关键因素。

### 经济可行性

我们采用了多种方法来估算项目成本，包括功能点估算、COCOMO模型、Walston-Felix模型等。根据我们的估算，预计投入成本大约为20万元，项目报价约为30万元；根据合同金额和年利率8%，采用年初50%、年末50%的付款方案计算，最终净现值为9.0035。这意味着我们在一年内可以收回成本，并且项目后续还能持续创造价值。因此，从经济角度来看，这是一个可行的项目。

### 技术可行性

从技术可行性考虑，经过多年发展，相关技术已相当成熟并稳定。在本项目中，我们将应用开放接口技术和动态管理技术，以实现对灾情信息的有效管理和必要功能的实现。我们团队成员具备类似项目操作经验，而其他成员也可以通过简短的培训快速上手。因此，从技术角度看，这是可行的。

### 组织可行性

项目组织架构已明确描述了需求、设计、编码和测试人员的职责和所需人数，以及每个里程碑的工作内容、时间和输出内容。甲方建立了完善的管理制度和流程，为项目实施提供了制度上的保障。这些措施将有效促进项目的顺利推进，因此从组织角度来看，项目也是可行的。

### 社会可行性

项目是国家震情应急管理系统的一部分，针对震后灾情获取缓慢且碎片化的问题，旨在实现异构公众涉灾信息的一体化编码，以实现数据统一管理和高效合理利用。项目并不触犯法律和道德，所有使用的软件和中间件均为正版，不会引发法律纠纷或造成社会负面影响。从社会层面来看，项目也是可行的。

### 资源可行性

我们能够满足所需软件开发人员的数量和技术要求。数据来源丰富多样，各地政府工作人员可以完成资源上传工作，解决了数据来源的问题。因此，从资源角度看，项目同样是可行的。

这些评估考虑了项目在经济、技术、组织、社会和资源等方面的可行性，表明该项目在多个方面具备良好的可行性。

# 四、项目概要设计阶段

在项目的概要设计阶段，优化资源配置是确保经济利益最大化的关键之一。所以，此阶段的重点工作是建立起高效的资源利用和风险管理机制。通过精准的资源配置和风险控制，我们可以最大化经济利益，确保项目顺利执行并在预算范围内取得成功。

作为项目的管理和开发人员，在此阶段需要注重以下几个方面：

**成本控制和优化**：追踪成本，并确保在预算范围内合理分配。定期审查预算执行情况，寻找成本节约的可能途径，如优化流程、资源再分配等。

**技术资源规划**：保证所需软硬件设备和工具的有效利用。选择合适的技术栈和工具能够提高开发效率和系统性能，但也要考虑成本效益，避免过度采购或不必要的投入。

**风险管理策略**：在概要设计阶段考虑风险并制定应对策略。识别可能的风险点，实时监测潜在风险并及时调整策略。通过持续的风险评估，可以迅速做出反应，采取措施以减少或规避可能的损失、降低风险影响。

## 1.成本估算和资源分配

在概要设计阶段，系统被细分为四大功能模块，每个模块都承担着特定的职责，以便更好地满足系统的整体需求：

账户管理模块： 这个模块负责用户身份验证、权限控制以及账户信息管理。它可能包括子模块如注册、登录、权限分配等，以确保用户信息安全可靠。

数据管理模块：这个模块专注于数据的收集、存储和处理。它包含的子模块如数据采集、数据存储、数据清洗和数据备份等，以确保系统能够高效地管理大量灾情信息。

数据分析模块：这个模块用于对收集到的数据进行分析和挖掘，以提供有用的洞察和预测。可能包括子模块如数据可视化、统计分析等，以支持决策制定和应急响应。

安全管理模块：这个模块专注于系统的安全性，包括防止数据泄露、恶意攻击和系统故障。它包括的子模块如身份验证、加密、安全审计等，以确保系统运行的稳定和安全。

每个功能模块又进一步划分为多个功能子模块，这种模块化设计使开发团队更好地分工合作，提高了代码的可维护性和系统的扩展性。

### Bottom-up估算方法

综合考虑到各模块的复杂度、开发难度和所需资源，下面将使用Bottom-up估算方法通过对项目中各个组成部分、子任务的详细估算来得出模块的整体成本，并制定合理的成本预算。

首先，我将整个系统划分成不同的功能模块，每个模块中包含不同的子任务，然后再为每个子任务分配不同的人力资源。因为不同类型人力资源的成本是不同的，可以根据每种人力资源单位的成本来计算总成本。

M（管理人员）单位成本为1000元/单位

D（开发人员）单位成本为800元/单位

Q（质量保证）单位成本为1200元/单位

S（技术支持）单位成本为900元/单位

|  |  |  |  |
| --- | --- | --- | --- |
| 系统模块 | 人力 | 时间（时） | 成本 |
| 账户管理模块 | 总和：12000 | | |
| 登录 | M:1/D:1/Q:1/S:1 | 5 | 4000 |
| 注册 | M:1/D:1/Q:1/S:1 | 5 | 4000 |
| 个人信息管理 | M:1/D:1/Q:1/S:1 | 5 | 4000 |
| 数据管理模块 | 总和：22400 | | |
| 数据输入 | M:1/D:1/Q:1/S:1 | 5 | 4000 |
| 数据解码 | M:1/D:2/Q:1/S:1 | 10 | 4800 |
| 数据信息管理 | M:1/D:3/Q:1/S:1 | 15 | 5600 |
| 数据审计和追踪 | M:1/D:1/Q:1/S:1 | 5 | 4000 |
| 数据查询 | M:1/D:1/Q:1/S:1 | 5 | 4000 |
| 数据分析模块 | 总和：14400 | | |
| 数据分析和挖掘 | M:1/D:2/Q:1/S:1 | 10 | 4800 |
| GIS（地理信息系统）集成 | M:1/D:2/Q:1/S:1 | 10 | 4800 |
| 通知和警报 | M:1/D:2/Q:1/S:1 | 10 | 4800 |
| 安全管理模块 | 总和：12000 | | |
| 权限和角色管理 | M:1/D:1/Q:1/S:1 | 5 | 4000 |
| 数据备份和灾难恢复 | M:1/D:1/Q:1/S:1 | 5 | 4000 |
| 日志和监控 | M:1/D:1/Q:1/S:1 | 5 | 4000 |
| 成本总和：60800元，工时总和：100小时 | | | |

通过bottom-up估算方法，通过将各功能模块分解成可管理的任务单元，根据团队成员的技能和工作负载，确定了较合适的人力资源分配方案。然后对每个任务进行了单独的成本估算，再将每个任务的估算成本进行累加，并逐步汇总到更高层级的任务或项目整体，以得出整体项目的成本估算。最终在预算范围内合理分配并优化了项目成本。

## 2.项目进度计划制定

看板（Kanban）是一种敏捷项目管理方法，通过视觉化的方式来跟踪任务、工作流程和项目进度。在看板中，任务通常以卡片的形式呈现，每个卡片代表一个特定的工作项或任务。这些卡片被放置在一个共享的面板上，面板被分成不同的列或区域，代表工作流程的不同阶段，比如“待办”、“进行中”和“已完成”。

计算完各模块子功能的成本之后，再为各个子功能划分优先级并使用看板来管理系统开发的进度计划，确保在紧急情况下能够快速响应，同时降低时间成本。







## 3.项目风险评估和管理

对于本项目的风险评估和管理，我发现存在几个关键的不确定性和潜在风险并提出了相应的可能应对方案：

### 需求变更风险

需求变更风险可能随着软件项目的进行逐渐显现。这意味着我们可能会在项目开展的过程中发现需要新增或修改的功能。为了预防此类风险，我们需要团队成员高度配合，并在需求分析阶段精确分配团队成员的工作任务。

一种应对方式是安排定期的小组会议以加强沟通，尽可能避免方向错误的发生。

### 进度风险

进度风险是制约软件项目的一个主要因素。在软件项目招标阶段，为争取项目，开发方往往会承诺超出实际可行的项目进度，导致项目在初期就面临严重的时间问题。

为了应对这种风险，我们需要进行市场调研和数据调查，预测用户需求压力可能发生的变化，然后对项目的时间安排进行调整。

### 技术风险

技术风险也是需要关注的一点。我们必须根据项目实际需求，选择适合并成熟的技术，而非盲目追求先进性，特别是对于团队未熟悉或并非项目所必需的技术。对于技术要求超出团队掌握能力的情况，我们需要重点关注和管理该风险因素。

### 人力资源风险

人力资源风险也是软件开发项目中的重要因素。软件开发是智力密集型和劳动密集型的项目，人员资源对项目的影响巨大。在不同阶段，需要不同的人员配备，同时也需要团队协作。人员流失或人员不适应项目要求都可能造成人力资源上的风险。

为了应对这种风险，我们可以制定详细的招募合同，明确对项目的要求，并在项目开始时组织培训以确保员工能够完成项目要求。

### 工具风险

工具风险是软件项目开发和实施过程中的另一个重要因素。必要的管理工具、开发工具、测试工具等是否能够及时到位，以及其版本是否符合项目需求，都需要仔细考虑。

为了应对这种风险，我们需要在项目开发之前进行充分的项目调研，比较和选择适合我们项目的稳定、广泛应用的开发和测试工具。

### 质量风险

软件质量问题可能导致项目延迟交付或功能异常。不完善的测试策略和质量控制可能造成软件质量不达标。

应对策略可能包括加强测试流程和质量控制机制，确保交付的软件达到高质量标准。

### 安全风险

软件安全漏洞可能导致数据泄露、系统瘫痪或未经授权的访问。项目需要重视安全开发标准和安全审计，采取措施确保软件安全性。

### 成本风险

预算超支可能导致项目停滞或无法按时完成。

应对这种风险，需要严格控制项目成本，制定有效的成本控制策略。

### 沟通风险

不清晰的沟通可能导致项目团队合作不畅、理解偏差或需求误解。建立良好的沟通机制和团队协作平台，确保信息畅通和需求明确。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 类别 | 概率 | 影响程度 | 风险描述 | 应对措施 |
| 需求变更风险 | 50% | 8 | 需求变更风险可能随着软件项目的进行逐渐显现。这意味着我们可能会在项目开展的过程中发现需要新增或修改的功能。 | 强调安全审查和代码审查的重要性。建议运行软件前进行代码审查，同时监测可能的偏见和安全问题。 |
| 进度风险 | 40% | 7 | 进度风险是制约软件项目的一个主要因素。在软件项目招标阶段，为争取项目，开发方往往会承诺超出实际可行的项目进度，导致项目在初期就面临严重的时间问题。 | 我们需要进行市场调研和数据调查，预测用户需求压力可能发生的变化，然后对项目的时间安排进行调整。 |
| 技术风险 | 40% | 8 | 技术风险是我们不根据项目实际出发，盲目追求先进性，特别是对于团队未熟悉或并非项目所必需的技术。 | 对于技术要求超出团队掌握能力的情况，我们需要重点关注和管理该风险因素。 |
| 人力资源风险 | 50% | 9 | 软件开发是智力密集型和劳动密集型的项目，人员资源对项目的影响巨大。在不同阶段，需要不同的人员配备，同时也需要团队协作。人员流失或人员不适应项目要求都可能造成人力资源上的风险。 | 我们可以制定详细的招募合同，明确对项目的要求，并在项目开始时组织培训以确保员工能够完成项目要求。 |
| 工具风险 | 30% | 7 | 工具风险是软件项目开发和实施过程中的另一个重要因素。必要的管理工具、开发工具、测试工具等是否能够及时到位，以及其版本是否符合项目需求，都需要仔细考虑。 | 我们需要在项目开发之前进行充分的项目调研，比较和选择适合我们项目的稳定、广泛应用的开发和测试工具。 |
| 质量风险 | 20% | 8 | 软件质量问题可能导致项目延迟交付或功能异常。不完善的测试策略和质量控制可能造成软件质量不达标。 | 应对策略可能包括加强测试流程和质量控制机制，确保交付的软件达到高质量标准。 |
| 安全风险 | 40% | 9 | 软件安全漏洞可能导致数据泄露、系统瘫痪或未经授权的访问。 | 项目需要重视安全开发标准和安全审计，采取措施确保软件安全性。 |
| 沟通风险 | 30% | 6 | 不清晰的沟通可能导致项目团队合作不畅、理解偏差或需求误解。。 | 建立良好的沟通机制和团队协作平台，确保信息畅通和需求明确 |
| 成本风险 | 10% | 7 | 预算超支可能导致项目停滞或无法按时完成。 | 应对这种风险，需要严格控制项目成本，制定有效的成本控制策略。 |

# 五、项目编码阶段

在项目的需求分析和概要设计阶段，我们成功地完成了对项目整体需求和开发流程的规划设计。随着我们进入项目的编码阶段，我们意识到项目的成功实施不仅仅依赖于技术实现，还需要有效的进度监控和经济管理。

在编码阶段，我们的主要焦点将是实施严格的进度监控，以确保项目按计划进行，同时在控制成本和时间的基础上保证质量和功能需求的充分实现。我们将采用赚取值管理（Earned Value Management，EVM）等有效的项目管理工具和技术，以全面了解项目的绩效情况。通过监控计划价值（Planned Value，PV）、实际成本（Actual Cost，AC）和赚取值（Earned Value，EV），我们能够及时识别潜在的问题和风险，从而迅速采取纠正措施，确保项目顺利推进。

同时，我们深知实施阶段的经验教训对项目的成功至关重要。因此，我们将在实践中不断收集、分析和总结经验，以便及时提出改进建议，优化项目的经济效益。这可能涉及到工作流程的调整、技术实现的优化或团队协作的改进等方面。通过持续改进，我们追求最佳实践，确保项目在经济层面达到最大的效益。

## 1.项目进度监控

### 赚取值管理

赚取值管理（Earned Value Management，EVM）是一种项目管理方法，它结合了项目进度、成本和质量等方面的指标，用于评估项目的实际进展情况并预测最终成果。它主要通过比较计划值、实际值和赚取值来评估项目的绩效和进展。

赚取值管理的输入：

BAC (Budget at Completion)：项目完工预算，即整个项目完成时预期的总成本。

TAC (Total Actual Cost)：项目当前的实际总成本，即到目前为止已经花费在项目上的总成本。

BCWS (Budgeted Cost of Work Scheduled)：也称为PV（Planned Value），指的是在特定时间点或阶段计划完成的工作的预算成本。

ACWP (Actual Cost of Work Performed)：也称为AC（Actual Cost），指的是实际上已经花费在项目上的成本。

BCWP (Budgeted Cost of Work Performed)：也称为EV（Earned Value），指的是在特定时间点或阶段实际完成的工作的预算成本。

赚取值管理的输出：

SV( Schedule Var iance ) = BCWP - BCWS

=0: 按照计划进度进行

<0: 落后于进度

>0: 超前于进度

CV (Cost Variance)=BCWP-ACWP

=0: 按照计划预算进行

<0: 比预算差

>0: 比预算好

进度效能指标 SPI( Schedule Performance Index)= BCWP / BCWS

=1: 按照计划进度进行

>1: 超前于进度

<1: 落后于进度

成本效能指标: CPI(Cost Performance Index)= BCWP / ACWP

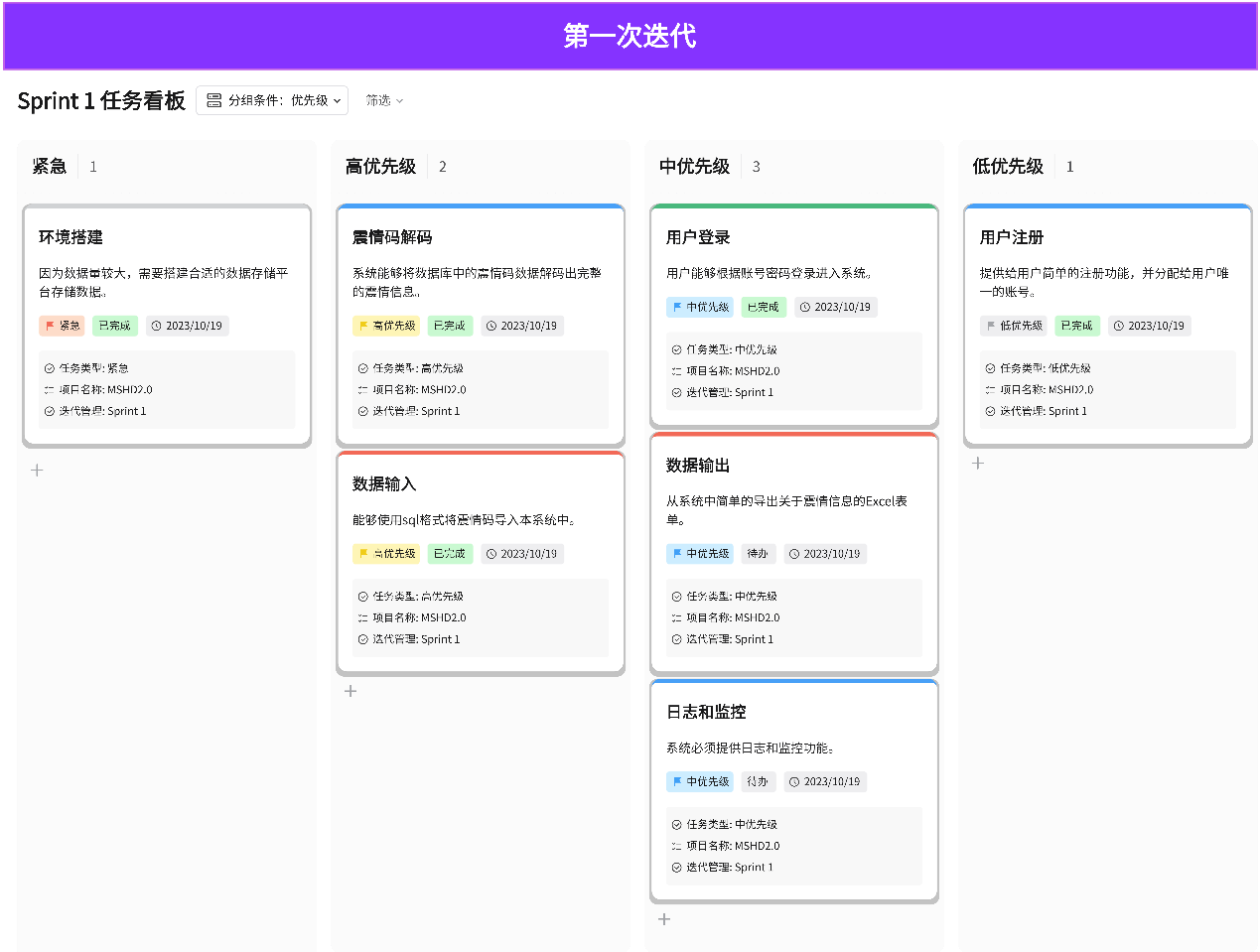
=1: 按照计划预算进行

>1: 低于预算

<1: 超出预算

这里我将分别通过每次迭代过程中的BAC，TAC，BCWC，ACWP，BCWP计算出SV、CV、SPI、CPI等值，进行赚取值分析以评估项目进度。

### 第一次迭代



*任务看板*

上图是项目具体开发过程中的任务看板，通过任务看板首先计算出这次迭代过程中的人力资源分配、工作量和成本投入。

|  |  |  |  |
| --- | --- | --- | --- |
| 系统模块 | 人力 | 时间（工时） | 成本 |
| 账户管理模块 | 总和：8000 | | |
| 登录 | M:1/D:1/Q:1/S:1 | 5 | 4000 |
| 注册 | M:1/D:1/Q:1/S:1 | 5 | 4000 |
| 数据管理模块 | 总和：8800 | | |
| 数据输入 | M:1/D:1/Q:1/S:1 | 5 | 4000 |
| 数据输出（简单的表格页面） | M:1/D:2/Q:1/S:1 | 10 | 4800 |
| 数据分析模块 | 总和：5600 | | |
| 震情码解码 | M:1/D:3/Q:1/S:1 | 15 | 5600 |
| 安全管理模块 | 总和：8000 | | |
| 环境搭建 | M:1/D:1/Q:1/S:1 | 5 | 4000 |
| 日志和监控 | M:1/D:1/Q:1/S:1 | 5 | 4000 |
| 成本总和：30400，工时总和：50 | | | |

通过项目的预算成本估计和在项目开发中的实际成本投入又可以得到以下赚取值管理的各项输入：

|  |  |
| --- | --- |
| 输入 | 成本 |
| BAC | 90400 |
| TAC | 21600 |
| BCWS | 30400 |
| ACWP | 21600 |
| BCWP | 26480 |

再使用赚取值输入根据公式得到具体的项目进度评估指标：

|  |  |
| --- | --- |
| 输出 |  |
| SV | -8800 |
| CV | 4880 |
| SPI | 0.8710 |
| CPI | 1.2259 |

SV表示计划价值（PV）与赚取值（EV）之间的差异。在这种情况下，SV为-8800，表明项目的赚取值比计划价值少8800，表明项目进度落后于计划。

CV表示赚取值（EV）与实际成本（AC）之间的差异。在这种情况下，CV为4880，表示项目的赚取值比实际成本多4880，表明在成本方面表现良好。

SPI表示赚取值（EV）与计划价值（PV）之间的比率。在这种情况下，SPI为0.8710，表明项目进度落后于计划，完成进度为计划的87.10%。

CPI表示赚取值（EV）与实际成本（AC）之间的比率。在这种情况下，CPI为1.2259，表示项目在成本方面表现良好，花费比预期的少。

综合分析这些指标，项目在成本方面表现良好（CV为正值，CPI大于1），但在进度方面落后于计划（SV为负值，SPI小于1）。这意味着尽管在第一次迭代中开发在成本控制方面做得不错，但需要更多努力来赶上或超越原定的进度计划。

### 第二次迭代



上图是项目具体开发过程中的任务看板，通过任务看板首先计算出这次迭代过程中的人力资源分配、工作量和成本投入。

|  |  |  |  |
| --- | --- | --- | --- |
| 系统模块 | 人力 | 时间（工时） | 成本 |
| 账户管理模块 | 总和：9600 | | |
| 个人信息管理 | M:1/D:2/Q:1/S:1 | 10 | 4800 |
| 权限和角色管理 | M:1/D:2/Q:1/S:1 | 10 | 4800 |
| 数据管理模块 | 总和：13600 | | |
| 关键词检索 | M:1/D:1/Q:1/S:1 | 5 | 4000 |
| 数据信息管理 | M:1/D:3/Q:1/S:1 | 15 | 5600 |
| 批量处理 | M:1/D:1/Q:1/S:1 | 5 | 4000 |
| 数据分析模块 | 总和：5600 | | |
| 数据导入和导出（支持各种格式的导入和导出） | M:1/D:3/Q:1/S:1 | 15 | 5600 |
| 安全管理模块 | 总和：4000 | | |
| 性能优化 | M:1/D:1/Q:1/S:1 | 5 | 4000 |
| 成本总和：32800，工时总和：65 | | | |

通过项目的预算成本估计和在项目开发中的实际成本投入又可以得到以下赚取值管理的各项输入：

|  |  |
| --- | --- |
| 输入 | 成本 |
| BAC | 90400 |
| TAC | 63200 |
| BCWS | 32800 |
| ACWP | 32800 |
| BCWP | 32800 |

再使用赚取值输入根据公式得到具体的项目进度评估指标：

|  |  |
| --- | --- |
| 输出 |  |
| SV | 0 |
| CV | 0 |
| SPI | 1 |
| CPI | 1 |

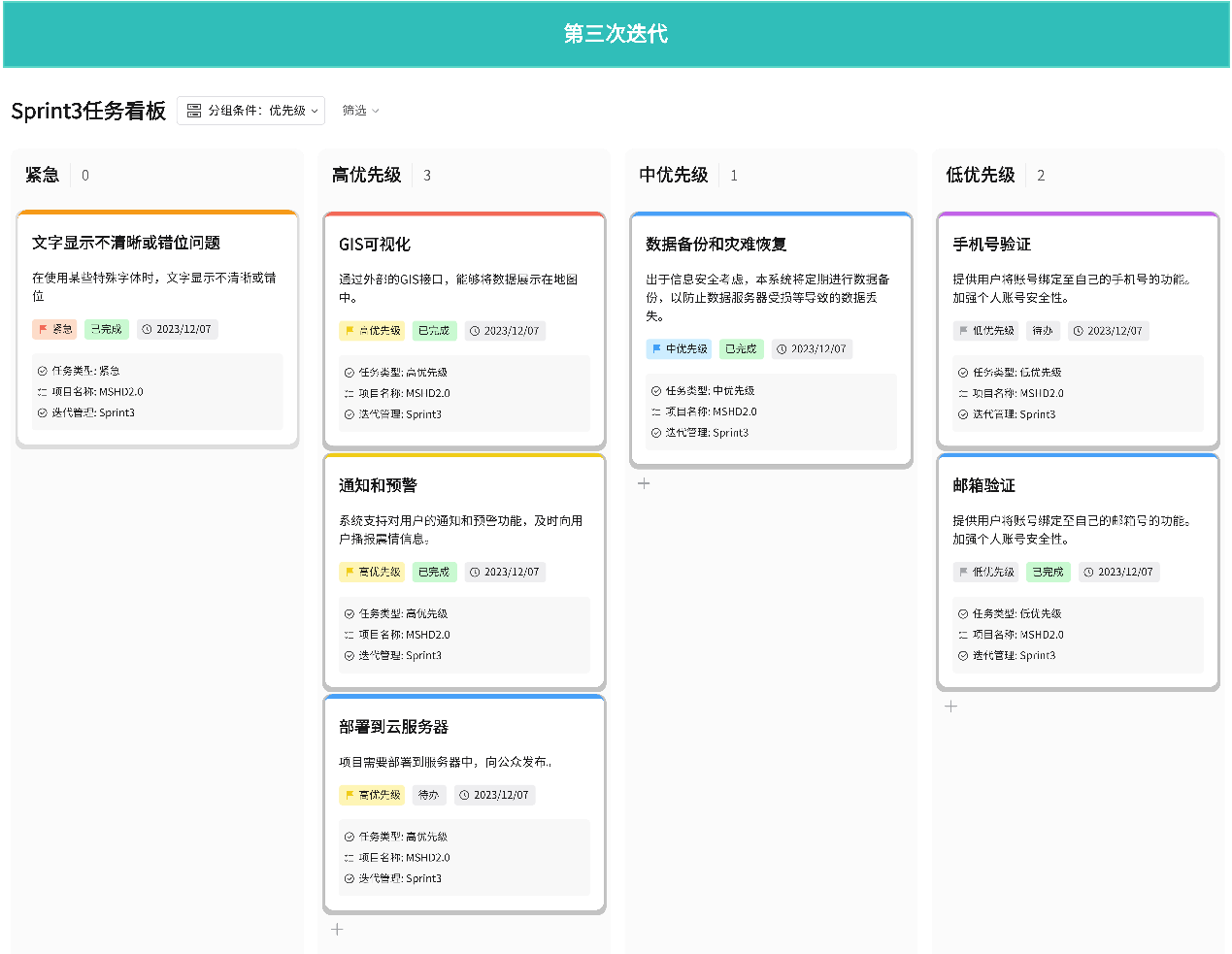
SV和CV为0：表示项目在进度和成本方面都没有偏差，实际完成的工作价值和计划的价值以及实际成本都是一致的。

SPI为1：表明项目进度符合预期，工作完成的速度符合计划。

CPI为1：表示项目的成本效率符合预期，实际花费的成本与完成的工作价值相符合。

这些数据显示第二次迭代中项目处于一个良好的状态，进度和成本方面都与预期一致，没有偏差。

### 第三次迭代



上图是项目具体开发过程中的任务看板，通过任务看板首先计算出这次迭代过程中的人力资源分配、工作量和成本投入。

|  |  |  |  |
| --- | --- | --- | --- |
| 系统模块 | 人力 | 时间（工时） | 成本 |
| 账户管理模块 | 总和：8000 | | |
| 手机号验证 | M:1/D:1/Q:1/S:1 | 5 | 4000 |
| 邮箱验证 | M:1/D:1/Q:1/S:1 | 5 | 4000 |
| 数据分析模块 | 总和：11200 | | |
| GIS可视化 | M:1/D:2/Q:1/S:1 | 10 | 5600 |
| 预警和通知 | M:1/D:2/Q:1/S:1 | 10 | 5600 |
| 安全管理模块 | 总和：8000 | | |
| 数据备份和灾难恢复 | M:1/D:1/Q:1/S:1 | 5 | 4000 |
| 部署到云服务器 | M:1/D:1/Q:1/S:1 | 5 | 4000 |
| 成本总和：27200，工时总和：40 | | | |

通过项目的预算成本估计和在项目开发中的实际成本投入又可以得到以下赚取值管理的各项输入：

|  |  |
| --- | --- |
| 输入 | 成本 |
| BAC | 90400 |
| TAC | 82400 |
| BCWS | 27200 |
| ACWP | 19200 |
| BCWP | 21200 |

再使用赚取值输入根据公式得到具体的项目进度评估指标：

|  |  |
| --- | --- |
| 输出 |  |
| SV | -8000 |
| CV | 2000 |
| SPI | 0.7794 |
| CPI | 1.1041 |

SV是计划价值（PV）与赚取值（EV）之差。在这个情况下，SV为-8000，意味着项目在特定时间点比计划要落后8000的预算成本单位。这表明项目的进度落后于计划。

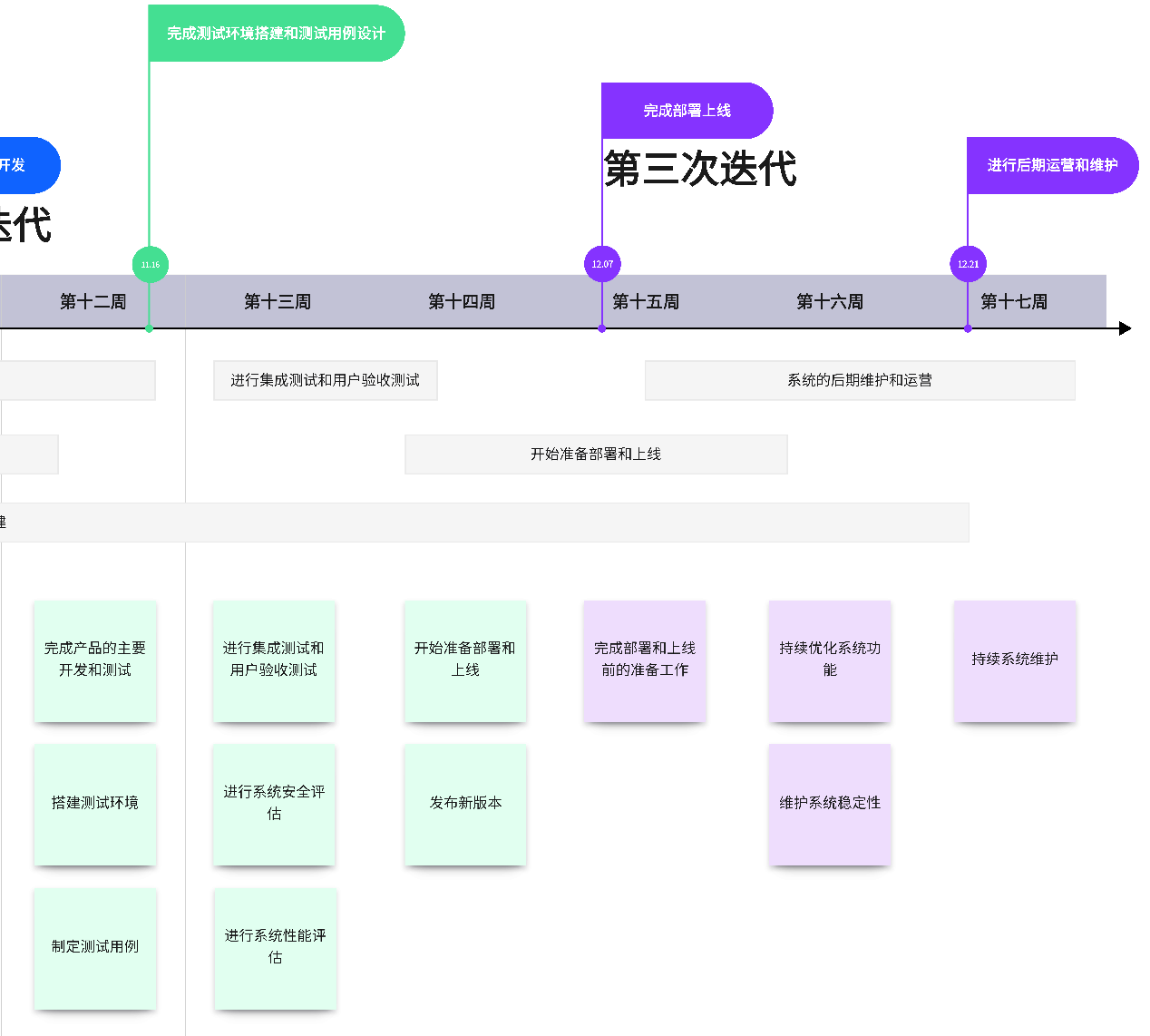
CV是赚取值（EV）与实际成本（AC）之差。在这个情况下，CV为2000，表示实际完成的工作的成本高于预期。这可以被视为成本偏差，但并不提供关于成本高于或低于预算的信息。

SPI是赚取值（EV）与计划价值（PV）的比率。SPI为0.7794，小于1，表明项目进度相对于计划进度较慢。SPI越接近1，表示项目进度越接近或超前于计划。

CPI是赚取值（EV）与实际成本（AC）的比率。CPI为1.1041，大于1，表明项目的成本绩效高于预期。CPI越接近1，表示项目的成本控制较好。

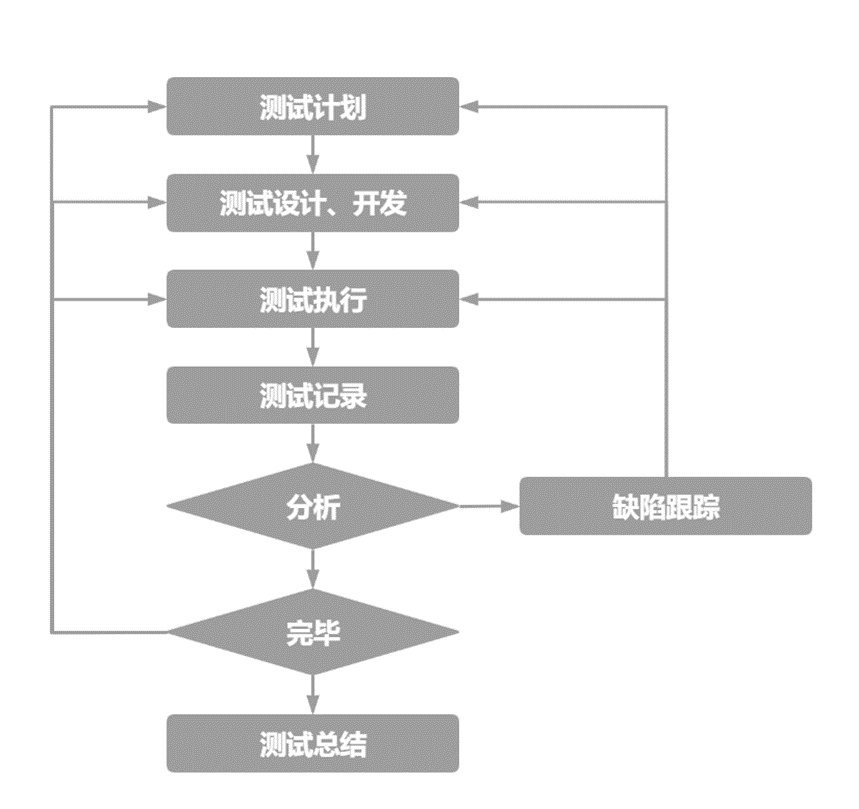
综合来看，这组数据表明在第三次迭代中项目进度落后于计划（SV为负值），而成本控制较好（CV为正值，CPI大于1）。这种情况下，项目需要关注提升项目进度以达到计划，并保持对成本的有效控制。

# 六、项目测试阶段



从上图中本项目的迭代开发计划图可以看出，在经过完整的项目开发阶段之后就进入到软件测试阶段，在这个阶段主要工作是完成产品的集成测试和用户验收测试，并对系统的性能和安全进行评估。这一阶段的任务意味着更广泛的经济考量，测试阶段需要投入额外的人力、时间和技术资源。我们应当关注如何优化这些资源的使用，以最大程度地减少额外成本。

## 1.软件测试



*软件测试的流程图*

Bug等级统计结果如下表。存在严重bug2，一般bug11，轻微bug27和建议（修改）bug29，共计69处，系统基本使用流畅，不存在较为严重的功能影响。其中，严重bug分别为：房屋破坏，土木数据列表中会正常记录为零的填入情况；个人中心，用户名修改功能无法使用。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| BUG统计 | | | | |
| 严重(Urgent) | 一般(High) | 轻微（Medium） | 建议（LOW） | 总计 |
| 2 | 11 | 27 | 29 | 69 |

*等级统计表*

## 2.修复成本

这里我从bug等级的角度来为各个bug的修复工作来分配不同的人力资源，因为不同类型人力资源的成本是不同的，再根据每种人力资源单位的成本来计算总成本。

M（管理人员）单位成本为100元/单位

D（开发人员）单位成本为50元/单位

Q（质量保证）单位成本为100元/单位

S（技术支持）单位成本为50元/单位

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Bug等级 | 数量 | 人力 | 时间（时） | 成本 |
| 严重(Urgent) | 2 | M:1/D:4/Q:1/S:1 | 2 | 900 |
| 一般(High) | 11 | M:1/D:3/Q:1/S:1 | 11 | 4400 |
| 轻微（Medium） | 27 | M:1/D:2/Q:1/S:1 | 13.5 | 9450 |
| 建议（LOW） | 29 | M:1/D:1/Q:1/S:1 | 14.5 | 8700 |
| 总计 | | | 41 | 23450 |

由上表可知，在软件的测试阶段仍需要投入23450元的成本和41工时的时间来进行bug修复。可见在项目迭代后期的投入并不亚于在项目前中期开发期间的投入。

## 3.系统建议

在软件工程经济学的视角下，尽管多源灾情数据管理服务系统已通过技术鉴定测试，且在功能性和性能方面满足了需求，但依然存在潜在的经济优化空间。从经济学的角度来看，以下是一些方面可以考虑优化改进，以降低项目成本并提高经济效益：

### 性能提升

1. 改进关键部分算法，并使用部分资源换取其他资源。多源灾情数据服务管理系统，通常包含大量查询操作，使用正确的算法，如B-/+ Tree存储结构，利用磁盘预读原理，将一个节点大小设为等于一个页，检索操作的渐进复杂度O(h) = O(logdN)，d为出度，h为深度，有效降低针对批量数据检索的复杂度。同时，将得到的中间数据（如部分地区某时段内灾情汇总等）暂时保存在一个储存库中，避免短时间内多次生成造成的事件损耗。

改良核心算法，一方面是提高了系统的运行效率，另一方面也降低系统后期的维护和运营难度，节约了维护开销。

1. 减少冗余功能、控制调用频率，避免过度设计系统或请求丢失。针对多源灾情数据服务管理系统，避免冗余的需求，确保所设计的功能是系统需要的。在这一点上主要避免开发冗余的功能，防止资源重复投入造成的浪费。

### 安全性提升

1. 通过加密存储敏感灾情和灾害等级等信息，并利用认证服务确保通信过程的可信性，可以有效降低潜在的信息泄露风险。这种保密性较高的数据资产可以利用加盐加密或其他编码方式进行保护，同时在系统或数据传输中利用认证服务确保信息的真实性和完整性。
2. 对多源灾情数据服务管理系统进行防火墙设计和部署是一种有效的成本优化策略。建立过滤封锁机制，有效控制内部网络与外部交互网络之间的通信，防止未经授权的数据流进出内部网络，以加强系统安全性。这种方法不仅有助于抵御攻击，还能辅助检测攻击，限制进出系统的数据流必须通过防火墙控制，并只允许经过授权的数据流通过。根据系统需求，可以选择分组过滤防火墙，作用于网络和传输层，通过标志识别允许或阻止数据包传输；或者选择应用代理防火墙（应用网关），监控和控制应用层通信流，以降低安全风险并避免不必要的成本支出。