

《软件项目管理》期末论文

**2020-2021(2)**



项目名称 基于大数据的城市智慧  
停车平台系统软件项目

项目经理： 章节：2、10

项目成员： 章节：3、 6

项目成员： 章节：5、 7

项目成员： 章节：4、 8

项目成员： 章节：1、 9

计算机科学与技术学院、软件学院

2020年6月10日

目录

[1.绪论 4](#_Toc76071866)

[1.1项目概述 4](#_Toc76071867)

[1.2开发目的与意义 4](#_Toc76071868)

[2.项目需求分析 5](#_Toc76071869)

[2.1可行性分析 5](#_Toc76071870)

[2.1.1经济可行性 5](#_Toc76071871)

[2.1.2技术可行性 5](#_Toc76071872)

[2.1.3社会可行性 6](#_Toc76071873)

[2.1.4风险因素及对策 6](#_Toc76071874)

[2.1.5可行性分析结论 6](#_Toc76071875)

[2.2同类产品比较分析 6](#_Toc76071876)

[2.2.1易管理性 6](#_Toc76071877)

[2.2.2便利性 7](#_Toc76071878)

[2.2.3精确性 7](#_Toc76071879)

[2.3项目开发模式与方法 7](#_Toc76071880)

[2.3.1需求合理性 8](#_Toc76071881)

[2.3.2需求完整性 8](#_Toc76071882)

[2.3.3需求优先级 8](#_Toc76071883)

[2.4预期交付成果 9](#_Toc76071884)

[3. 项目团队组织 9](#_Toc76071885)

[3.1 项目经理及其职责 9](#_Toc76071886)

[3.2 团队成员介绍及其职责 9](#_Toc76071887)

[3.3 项目团队价值与原则 10](#_Toc76071888)

[3.4 团队绩效管理 11](#_Toc76071889)

[4 项目进度管理计划 11](#_Toc76071890)

[4.1 项目关键路径确定 11](#_Toc76071891)

[4.2 项目工作分解 13](#_Toc76071892)

[4.3 项目时间甘特图分析 15](#_Toc76071893)

[4.4 优化进度控制过程 17](#_Toc76071894)

[5 项目成本管理计划 17](#_Toc76071895)

[5.1项目资源计划 17](#_Toc76071896)

[5.2项目成本估算 18](#_Toc76071897)

[5.2.1 软件开发成本估算 18](#_Toc76071898)

[5.2.2 项目成本估算 18](#_Toc76071899)

[5.3项目成本预算 19](#_Toc76071900)

[5.3.1分摊总预算成本 19](#_Toc76071901)

[5.3.2制定累计预算成本 19](#_Toc76071902)

[5.3.3优化基准计划 20](#_Toc76071903)

[5.4项目成本控制 20](#_Toc76071904)

[5.4.1 实际成本与盈余累计 20](#_Toc76071905)

[5.4.2 绩效分析 21](#_Toc76071906)

[6 项目质量管理计划 23](#_Toc76071907)

[6.1 软件质量的要素与度量 24](#_Toc76071908)

[6.2 技术标准与软件质量规划 25](#_Toc76071909)

[6.3 需求阶段质量控制 26](#_Toc76071910)

[6.4 设计阶段质量控制 27](#_Toc76071911)

[6.5 编码阶段质量控制 29](#_Toc76071912)

[6.6 测试阶段质量控制 31](#_Toc76071913)

[7 项目风险管理计划 33](#_Toc76071914)

[7.1 风险识别 33](#_Toc76071915)

[7.2 风险分析 33](#_Toc76071916)

[7.3 风险跟踪 34](#_Toc76071917)

[7.4 风险应对 34](#_Toc76071918)

[8 项目采购管理计划 34](#_Toc76071919)

[9 项目沟通和冲突管理 35](#_Toc76071920)

[9.1 沟通管理 35](#_Toc76071921)

[9.2 冲突管理 35](#_Toc76071922)

[10 项目管理总结 36](#_Toc76071923)

[11 参考文献 36](#_Toc76071924)

# 1.绪论

## 1.1项目概述

随着国民经济的不断提升，汽车越来越多的进入了寻常百姓人家。加上我国近年来大力发张汽车产业，频繁出台各种鼓励家庭拥有汽车的消费政策，使得汽车保有量逐年提高。根据中国汽车工业协会数据统计，2014年中国汽车产销量达2400万辆，有望连续第六年稳居世界头号汽车产销大国的地位。

伴随着汽车保有量的迅速增加，停车场规模的不断扩大，城市汽车和停车场之间的矛盾也日益突出。停车场停车难，找车难，车场管理难等问题频频浮现。

停车场智能化管理的需求，以实现停车场智能化管理为目标，以购买月卡的固定停车用户为服务主要服务对象，兼顾外来临时车辆停车的服务，以达到停车用户进出方便、快捷、安全，管理处管理科学、高效、服务优质文明的目的。对提高管理处的管理层次和综合服务水平方面将起重要的作用。我公司经反复调研，并针对未来发展需求，精心设计出全新智能停车场管理系统方案，包括基于非接触式IC卡实现出入口管理的方案和基于车牌识别实现出入口管理的方案。

## 1.2开发目的与意义

在基于非接触式IC卡的停车场管理系统中，持有月租卡和贵宾卡的车主在出入停车场时，经车辆检测器检测到车辆后，将IC卡在出入口控制机的读卡区掠过，读卡器读卡并判断卡的有效性。对于有效的IC卡，自动道闸的闸杆升起放行并将相应的数据存入数据库中。若为无效的IC卡或进出场的车辆，则不给予放行。对临时停车的车主，在车辆检测器检测到车辆后，按入口控制机上的按键取出一张IC卡，并完成读卡放行。在出场时，车主将临时卡交给保安读卡并交纳停车费用，无异常情况时道闸升起放行。

在基于车牌识别的停车场管理系统中，针对内部车辆开放的场所，则固定用户或贵宾用户的车辆在出入停车场时，出入口摄像机自动抓拍车辆的车牌。对于有效车牌的车辆，自动道闸的闸杆升起放行并将相应的数据存入数据库中。若为无效车牌的车辆，则不给予放行。针对所有车辆开放的场所，则车辆在入停车场时，入口摄像机自动抓拍车辆的车牌，保存车辆图片后，自动开闸放行；出停车场时，出口摄像机自动抓拍车辆的车牌，保存车辆图片后，与进入时的抓拍图片进行比对，比对通过后缴费，道闸自动开闸放行。

同时，针对大型地下停车场内，车主停车难，返回时又容易在停车场内迷失方向，寻找不到自己的车辆。该智能停车诱导方案，通过全视频引导方式，结合场内诱导屏、反向寻车机，帮助车主尽快找到停车位，及找到自己车辆停放的位置。极大降低管理人员成本，消除寻找车位的烦恼，提高顾客对停车场乃至其所属物业公司的满意度；加快停车场的车辆周转，提高停车场的使用率和经济效益；提升停车场管理水平，提高停车场所属物业公司的对外形象。

**（1.绪论由XXX执笔）**

# 2.项目需求分析

## 2.1可行性分析

2.1.1经济可行性

硬件方面我们需要识别一体机和车辆检测器，其中识别一体机每个停车场都会配备，并且我们要求实现的功能现在大多数的一体机都能具备，所以一体机的要求能满足经济可行性。针对车辆检测器，我们要求对每一个车位进行一个红外检测装置的配置用来检测每个车位是否用车，这是基础要求，并且每个装置的费用大约在几十元人民币，相对来说比较经济可行，如果有条件可以增加识别车牌的功能，完善车位识别系统。针对软件方面，我们使用Apache的免费开源软件Tomcat作为服务器、以及开源数据库MySQL作为我们数据存储的方式，这能够最大限度的节省软件费用的开支，也满足经济可行性。

2.1.2技术可行性

本平台采用B/S架构，方便客户的浏览、使用。我们根据J2EE三层结构，分为表现层、业务逻辑层和数据服务层，该分层结构，提高了软件的后期可维护性。三层结构以Spring+Struts+Hibernate作为支持，确保网站平台的高效、稳定、安全。

Struts是Apache 基金会Jakarta 项目组的一个Open Source 项目，它采用MVC模式，能够很好地帮助java 开发者利用J2EE开发Web应用。Hibernate是一个开放源代码的对象关系映射框架，它对JDBC进行了非常轻量级的对象封装，使得Java程序员可以随心所欲的使用对象编程思维来操纵数据库。本平台使用MySQL作为数据库，MySQL与java的交互性较好，并且能够方便地进行数据的转储和处理。

2.1.3社会可行性

智慧停车系统的普及，既方便了用户停车，也能够大大缓解交通压力；交通部门也可以减少工作量，缓解交通压力，实现两者双赢。并且停车场方也能够更加方便地对本停车场进行管理，实现停车场车位的全把控，以及更加智能人性化的查询。

2.1.4风险因素及对策

本项目的软件外包管理平台提供了识别和管理风险的流程，为甲乙双方建立透明的风险应对机制，明确风险发生后的解决方式并建立对后续工作的追踪，在风险管理活动中双方项目经理可随时对风险和应对方案进行评议，最大程度上确保项目的顺利实施。

2.1.5可行性分析结论

本项目的平台能够满足停车场方和用户方两种客户开展收费管理与停车业务的需求；本项目的管理平台将实现社会效益与经济效益并重，优先考虑社会效益。因此，本项目无论从经济、技术、社会还是风险因素等方面，本项目的设计方案是可行的。

## 2.2同类产品比较分析

与行业龙头公司诸如Jieshun捷顺，HongMen红门，KEYTOP科拓，FUJICA富士智能的同类产品进行比较分析，最后总结出了我们产品相较于这些同类产品的优势在于三点。

2.2.1易管理性

相较于市面上的同类产品单纯从停车场与客户的角度出发，提供服务，我们的产品还兼顾到了停车场管理员的使用，单独开发了管理员模块，集成了相关接口，并提供了更易于使用的操作界面。降低了管理门槛，有利于没有相关软件经验的人使用我们的系统。

2.2.2便利性

市面上的同类产品，大多采用“入计时，出缴费”的缴费方式，这种缴费方式的最大缺点就是在出停车场的高峰期，常常因为“出缴费”需要车辆在闸口经历“停车—缴费—放行”这一标准流程的原因，而造成堵车，大大降低了出停车场的效率和用户的体验。而我们的系统采用会员制度，会员在进入停车场后可以随时缴费，如若没有及时缴费，也不会影响出停车场，只是需要在事后“未缴清单”中进行补交。这一措施，极大的减少了停车场闸口拥堵现象，提升了用户体验。

2.2.3精确性

市面上的同类产品，在客户寻找车位时往往只能提供“剩余车位”的信息，但是“剩余车位”在哪呢？缺乏一个清晰的指示。而我们的产品，可以提供整个停车场的平面图，根据红外遥感提供的车位信息，精准的映射到平面图上供客户挑选车位，实现了车位与客户的直接联系，减少了客户寻找车位的时间，提高了效率。

## 2.3项目开发模式与方法

项目采用敏捷开发与极限编程。

* 在更短的周期内，更早地提供具体、持续的反馈信息。
* 在迭代的进行计划编制，首先在最开始迅速生成一个总体计划，然后在整个项目开发过程中不断的发展它。
* 依赖于自动测试程序来监控开发进度，并及早地捕获缺陷。
* 依赖于口头交流、测试和源程序进行沟通。
* 倡导持续的演化式设计。
* 依赖于开发团队内部的紧密协作。
* 尽可能达到程序员短期利益和项目长期利益的平衡。

再此基础上我们进行了需求分析的进一步优化，从以下几个角度进行迭代，

2.3.1需求合理性

我们对此项目的各项需求进行了简单整理，分析了停车场系统各个系统模块的需求之间是否存在矛盾，各项需求的功能实现是否存在流程逻辑上的互斥。整体上优化了停车场项目功能的内在逻辑，使后续开发更为顺利。

2.3.2需求完整性

我们将客户要求的各项功能，与我们团队预期的各项功能进行了对比，查看是否完成了功能上的覆盖，是否有遗漏。并对一些描述笼统的功能进行了工作上的细化，确保不会因文本上的笼统而造成工作上的功能缺失。

2.3.3需求优先级

我们根据客户的要求和工作上的经验，在与客户的频繁交流中对客户提出的大量需求进行了优先级的评定，优先级的评定，有助于后续开发中的各项工作安排，能有效提高工作效率。

其中优先级的制定过程相当复杂，最终我们在团队的合作下制定了停车场项目的优先级。

1. 用户体验：停车场系统是一个面向用户的产品，它要经历市场的考验，在市场中用户的体验十分重要，市面上同类产品也不少，良好的用户体验是用户选择我们产品的理由，恶劣的用户体验则会使用户放弃我们的产品而转投其他同类产品。用户体验是我们产品在市场中的核心竞争力。其中我们发现，市面中的很多产品在停车场与车位信息精度的实时交互上做的不好，很多车位已经停车了，但是没有在管理系统上体现，导致用户白跑一场，一旦发生用户体验十分差。
2. 管理员体验：停车场系统不仅要面向停车场的用户，还要面向停车场的管理者，在市面上很多产品在这一块做的不好，忽视了停车场管理者的体验，加大了停车场管理的难度，也增加了停车场管理员的学习成本。很可能会导致停车场的管理混乱，停车场管理混乱最终的结果就是用户体验很差。
3. 开发成本：成本又分为时间成本与经济成本，停车场系统这个项目是一个小型的项目，且需要较为快速的交付。这使得我们在开发过程中需要注重开发效率，减少时间成本，提高代码复用性。并且在框架，数据库的选择上尽量选择开源的以减少经济成本。

## 2.4预期交付成果

一个完整的智慧停车系统包含：

管理系统可实现停车场的后台管理，进行车位的调整，时间的调整等。

客户端：可实现缴费，可进行停车场、车位导航，实时查询等功能。

停车系统：可实现车辆进出的控制，缴费等功能

**（2.项目需求分析由XXX执笔）**

# 3. 项目团队组织

## 3.1 项目经理及其职责

邵佳乐：编写项目质量管理计划和项目实施总结，给成员分配具体的工作。确定产品功能，决定发布的日期和发布内容。根据市场价值确定功能优先级；每个sprint中，根据需要调整功能和优先级；接受或拒绝开发团队的工作成果；参与Sprint计划会议，Sprint评审会和Sprint回顾会。

## 3.2 团队成员介绍及其职责

朱可心：对需求调研获得的结果进行提取，画出数据流图。在模块设计方面负责车牌识别模块，同时负责对软件的持续测试。

章志怡：在项目初期进行需求调研，在模块设计方面负责用户信息模块，负责后端代码。

周纹慧：进行用例分析并作出用例图，在模块设计方面负责车牌识别模块，负责前端代码。

申潇添：编写规格说明文档，在模块设计方面负责管理员模块，负责UI设计。

## 3.3 项目团队价值与原则

制定良好的规章制度：由项目经理分配、调整工作，成员积极配合，尽最大的力量去完成任务，如果无法按时完成任务需和项目经理沟通，无故失职有惩罚，做的好的给予表扬。

共同建立明确的目标：由于处于的工作位置不同，对相关项目的期望就会产生不同，成员们各执己见会导致整个团队不和谐并且效率低。项目经理需要及时发现并协调各个成员，在大方向上达成一致，在细节上做出妥协，协调每一个人的要求，达到平衡。

主张简单：当从事开发工作时，你应当主张最简单的解决方案就是最好的解决方案。不要过分构建（overbuild）你的软件。用AM的说法就是，如果你现在并不需要这项额外功能，那就不要在模型中增加它。要有这样的勇气：你现在不必要对这个系统进行过分的建模（over-model），只要基于现有的需求进行建模，日后需求有变更时，再来重构这个系统。尽可能的保持模型的简单。

递增的变化：和建模相关的一个重要概念是你不用在一开始就准备好一切。实际上，你就算想这么做也不太可能。而且，你不用在模型中包容所有的细节，你只要足够的细节就够了。没有必要试图在一开始就建立一个囊括一切的模型，你只要开发一个小的模型，或是概要模型，打下一个基础，然后慢慢的改进模型，或是在不再需要的时候丢弃这个模型。这就是递增的思想。

增强沟通和交流：团队中的每一个人都要积极沟通，出了问题要交流解决，不能一个人埋头苦干导致所有人的延期。当客户对项目需求有所更改时，项目经理也要有和客户积极沟通的能力，没有沟通好可能会导致团队更改不够及时，做很多无用功，最后超出原定的截止日期，导致超出预算且完成不了任务，在这种时候，项目主管与公司高层，与客户之间的沟通能力就极其重要，良好的沟通能力将有助于解决这类复杂问题。

## 3.4 团队绩效管理

组建团队之前，团队就有了明确的目标，打造简单且智慧的智能停车系统。实现停车场与车位信息精度与实时交互，为用户提供多种便捷并且符合用户实际需求。

实现迭代目标：作为Scrum原则，每一个迭代都会有一个目标，团队应该在合理的限度内进行工作，尽最大努力实现目标，以完成效率和效果进行评估。

交付价值：对每一次迭代给出的成本和最后交付的价值比较，达到标准或超出标准就是好的表现，如果交付价值不如当初的成本就说明这次迭代的负责人是失职的。

可持续的工作节奏：成员每天坚持工作和将工作时间压缩在短时间内，两种工作节奏，第一种是可取的，良性的工作节奏，能让效率更高，短时间高强度的工作会造成细节上失误，对此进行评估。

快速频繁的学习能力：可以认真学习且学习能力强的是优秀的，对于重要信息的学习拖拉，会拉断整体的工作效率，这也是重要的指标。

**（3.项目团队组织由XXX执笔）**

# 4 项目进度管理计划

## 4.1 项目关键路径确定

项目的关键路径确定，首先是规划分工，即对项目各部分进行拆解，并根据人员情况进行相应的工作安排。其次是工作进度管理，预估各个工作分解部分所需要的完成时间，进行统筹，避免因为进度不齐而导致工作时间浪费。之后是需求调研，形式是向目标人群分发调查问卷，得到调查结果。得到结果后，对结果进行分析，从中提取到项目的需求。进行用例分析，得到项目具体要实现的用例，完成项目的基础建模。之后根据用例分析的结果画出用例图，具象化用例，进行数据流图的绘制。此时项目初具雏形，与客户进行交流，在功能，性能，UI方面进行进一步确认，完成规格说明文档的建立。组长再次确认需求后，进行到正式的编码阶段，将整个系统根据功能实现划分为5个模块，由5名成员进行并行开发提高效率，全部完成后由组长在相应模块负责人的帮助下进行系统的耦合，完成功能上的初步实现。之后是测试环节，根据前后端划分测试模块，其中前端测试主要是功能上的测试，测试完成后进行UI的设计及相应代码的编写，对前端进美化。期间需要持续测试来确保项目不会因变动而出错。测试完成后项目进入到可交付阶段，由客户进行验收，验收通过后，项目团队将为项目提供长久的基本维护。

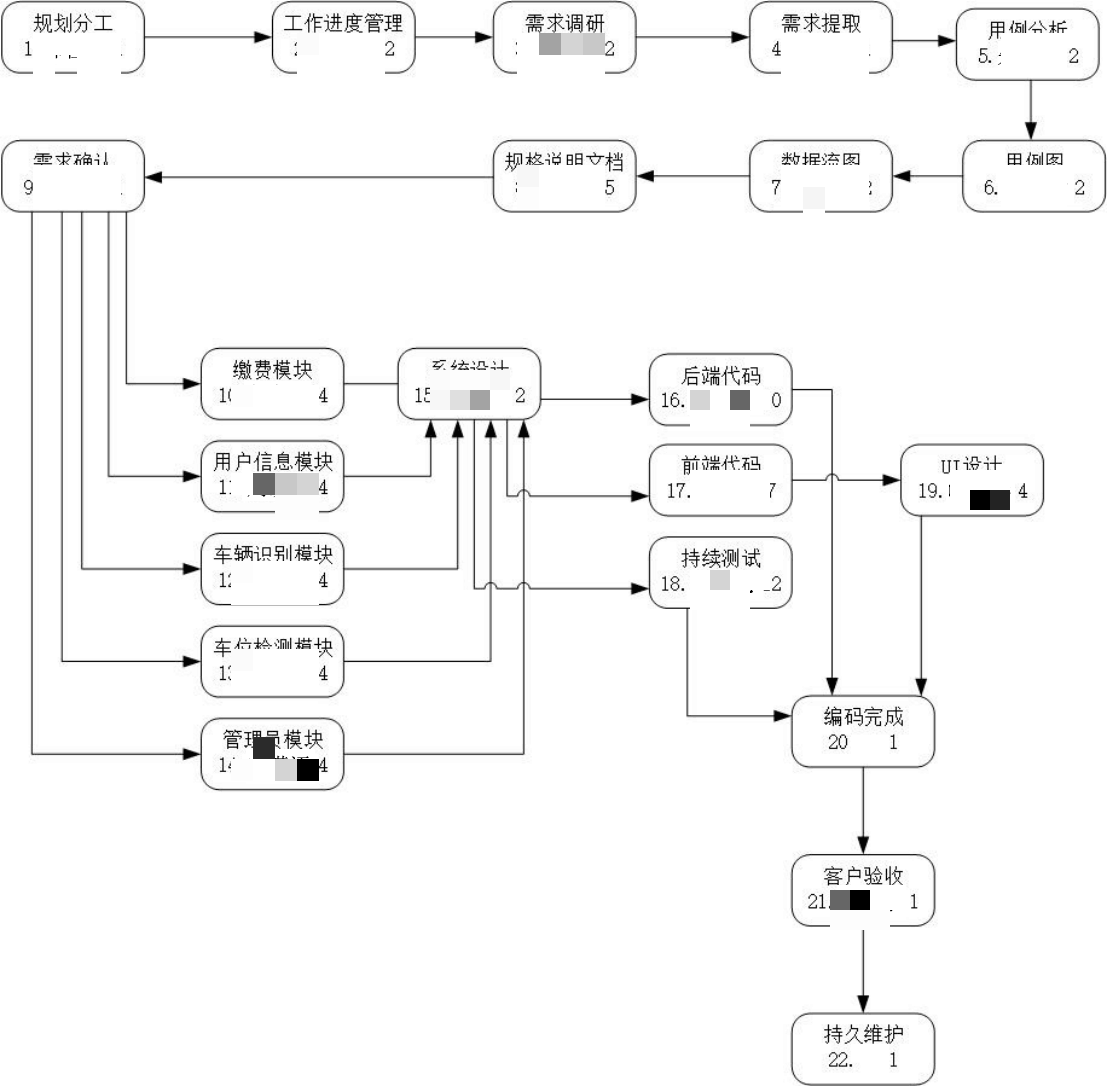


图4-1 关键路径图

## 4.2 项目工作分解

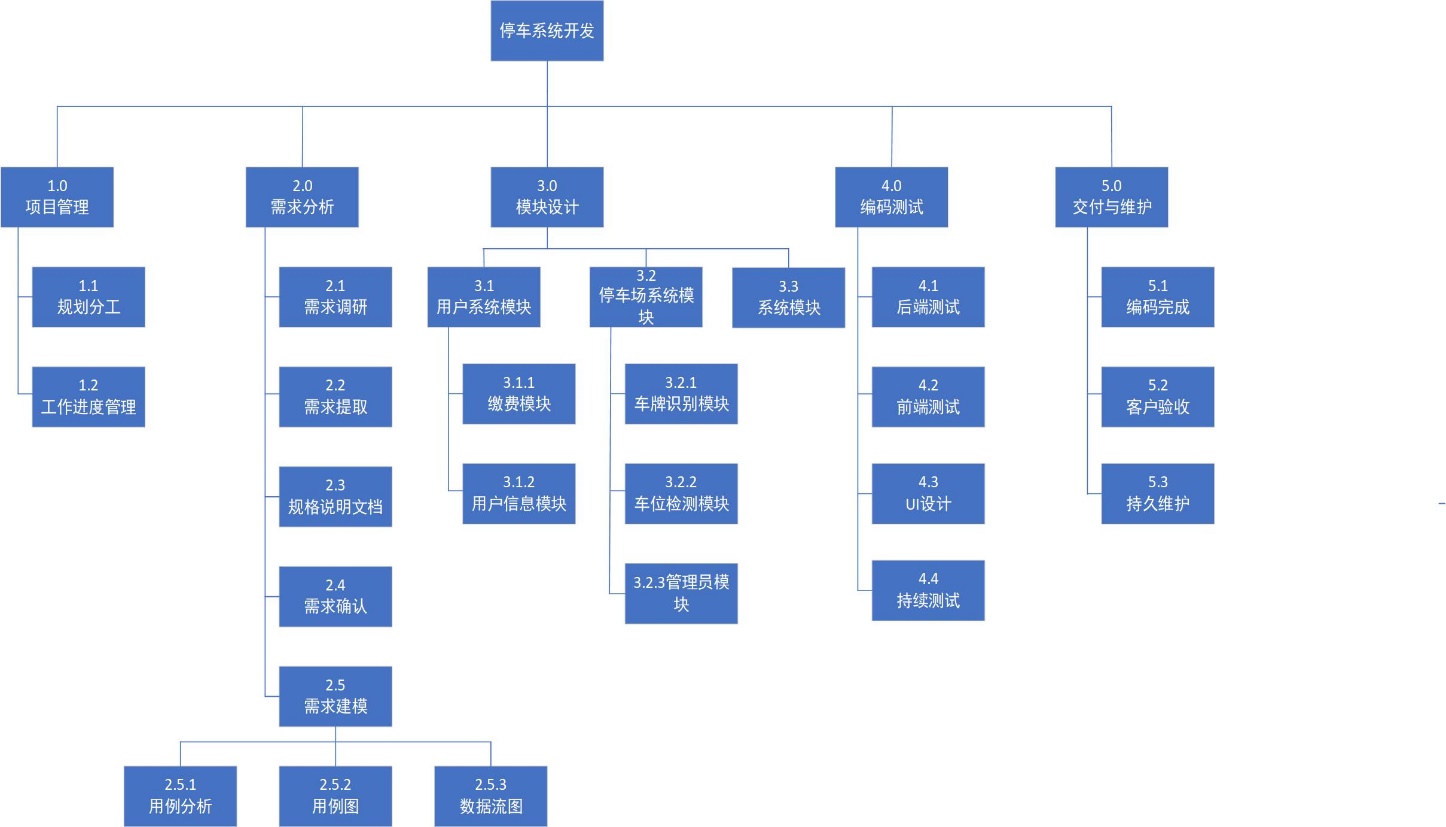


图4-1 项目工作分解图（WBS）

本项目的工作分解结构（WBS）主要分为3层，第一层根据项目的生命周期划分，分成项目管理，需求分析，模块设计，编码测试，交付与维护5个大块，第二层根据系统内容进一步划分17个块，最后部分块再细分。

表4-1 WBS词典

|  |  |
| --- | --- |
| 工作包 | 工作包描述 |
| 1项目管理 | 整个项目的管理 |
| 1.1规划分工 | 对项目工作的分配，落实到人 |
| 1.2工作进度管理 | 对项目工作各个分块完成日期的统筹安排 |
| 2需求分析 | 对项目的需求分析 |
| 2.1需求调研 | 对项目的需求进行小范围的随机调研 |
| 2.2需求提取 | 对项目的需求进行进一步的提取分离 |
| 2.3规格说明文档 | 完成项目的规格说明文档 |
| 2.4需求确认 | 对前面提取出来的项目需求进行进一步的确认，总结 |
| 2.5需求建模 | 对项目需求建模，系统化的展示需求 |
| 2.5.1用例分析 | 将需求转化为用例 |
| 2.5.2用例图 | 画用例图，更直观的展示用例需求 |
| 2.5.3数据流图 | 画数据流图，展示数据的在系统中的逻辑走向 |
| 3模块设计 | 将系统划分为模块 |
| 3.1用户系统模块 | 客户端的用户系统模块，有前端内容 |
| 3.1.1缴费模块 | 缴费模块的代码编写 |
| 3.1.2用户信息模块 | 用户信息模块的代码编写 |
| 3.2停车场系统模块 | 停车场系统模块 |
| 3.2.1车牌识别模块 | 车牌识别模块的代码编写 |
| 3.2.2车位检测模块 | 车牌识别模块的代码编写 |
| 3.2.3管理员模块 | 管理员端的系统 |
| 3.3系统设计 | 整个系统的整合 |
| 4.编码测试 | 整个系统的代码测试 |
| 4.1后端测试 | 后端方面的测试 |
| 4.2前端测试 | 前端方面的测试 |
| 4.3 UI设计 | 美术上UI的设计 |
| 4.4 持续测试 | 后续随系统迭代进行的测试 |
| 5 交付与维护 | 系统完成后 与客户的交付和后续系统的维护 |
| 5.1 编码完成 | 对编码进行完成度确认 |
| 5.2 客户验收 | 系统交付客户 客户进行验收 |
| 5.3 持久维护 | 系统的后续维护 |

## 4.3 项目时间甘特图分析

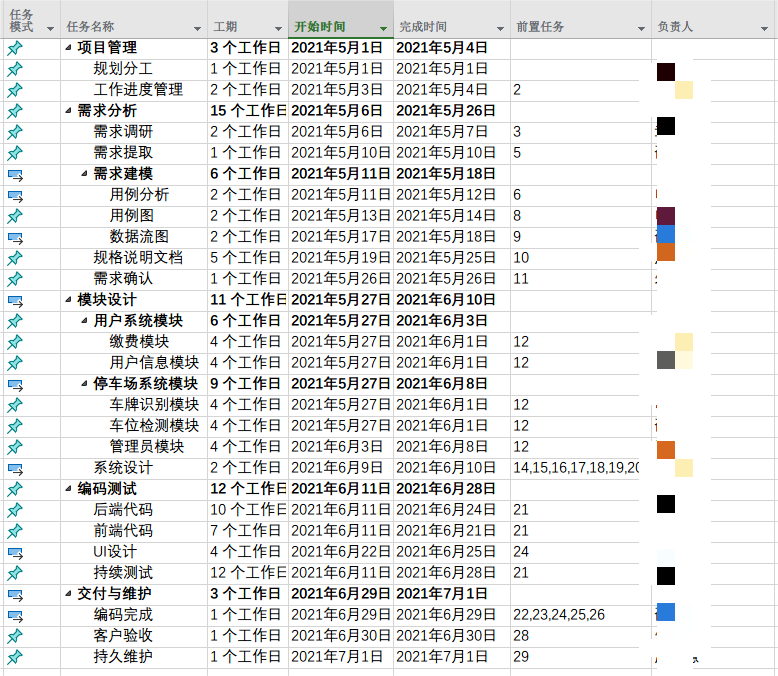


图4-2 时间甘特图（1）

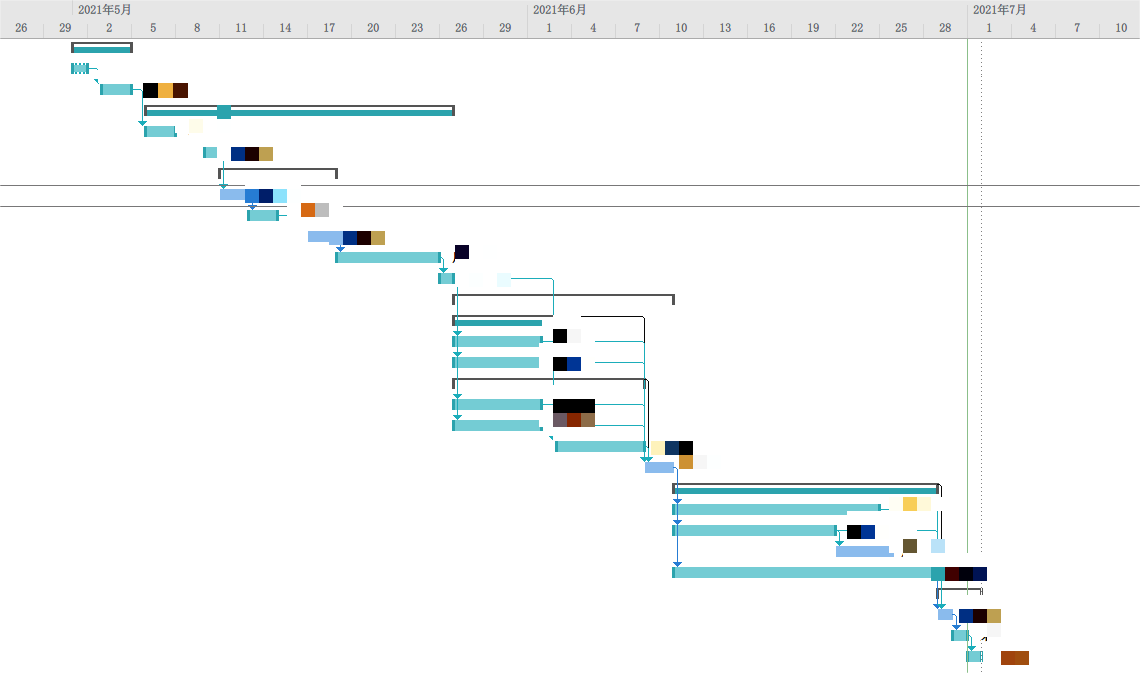


图4-3 时间甘特图（2）

从甘特图可以直观的，迅速的获得项目的信息

* 项目持续了约2个月，是一个短期项目
* 项目主要成员有5个，是一个轻量级项目
* 需求分析阶段占了整个项目约1/3的时间，可以体现出需求分析在整个项目中的地位很高，花费较多时间。
* 需求建模又是需求分析里占比较大的一部分
* 第二个较大的部分是模块设计
* 模块设计中的三个模块采用并行开发，大大提高了效率，节省了项目发开的时间长度
* 测试是第三个花费时间较多的模块，在整个项目中也占有相当重要的地位
* 测试前后端分开测试，提高了测试效率

## 4.4 优化进度控制过程

对于项目而言，精确估计产品的开发与交付进度是很困难的，一个有效的方法是将进度安排和工作管理有机的结合起来，通过创建阶段性报告来正确地反映项目如何按计划进行，得出如何纠正与项目的偏差等信息。

本团队为避免项目失控，造成更多的质量问题，在各个时间点处，进行系统开发的回顾与测试检查，指定更加详细的内部开发进度表，当进度任务出现问题时则先解决当前问题再进入下一进度，将每一进度的问题和解决方案汇总起来交给项目经理，由项目经理进行汇合整理，总结出设计思路，有助于后期工作的顺利高效开展。

**（4.项目进度管理计划由XXX执笔）**

# 5 项目成本管理计划

## 5.1项目资源计划

制定项目资源计划，首先要进行资源需求分析。而资源需求分析的第一步是工作量计算。

通过资料统计法，结合项目进度管理计划，工作量共计76个工作日。由团队工作人员的特点，实施方案为每人依次轮流负责每个阶段的任务，便于减轻工作压力，提高效率。

工作量为三人月，人员需求量定为5人，人均实际工作量为两周，设备、材料需求量为5台电脑，由于人员间工作存在交接，所以资源使用时间为所有工作日内。一体机单价600元，车辆检测器单价30元为例。资金需求计算：600\*2+30\*200=7200（元）。

表 5-1 详细资源需求表

|  |  |
| --- | --- |
| 人员需求量 | 5人 |
| 设备、材料需求量 | 5台电脑、两个一体机、200个车辆检测器 |
| 工作量 | 3人月 |
| 资源使用时间 | 所有工作日内 |
| 资金需求 | 7200元 |

该项目资源需求较少，暂时不需要进行资源供给分析、资源成本比较与资源组合、资源分配与计划编制。

## 5.2项目成本估算

5.2.1 软件开发成本估算

通过代码行（LOC）来估算软件开发成本，由往期的得100KB的代码约为1860行，投入工作量为200人天。每人每日费用为100元，1LOC的价值为：（100\*100）/3260=6.2元/L，经过软件规模测算，本项目代码大小约为520KB，则估计代码行数约为9672行。

所以该项目软件开发成本：9672\*0.6=60000元。

5.2.2 项目成本估算

由经验成本估算模型 **SLIM模型** (Putnam) ,其中L表示源代码行数，K指整个开发过程所花费的工作量，td指开发持续时间，Ck指技术状态常数。

由于本项目的开发时间不足一年，因此提高开发持续时间的影响因素到1次，公式修正为 。

由此估算工作量K===2.91（人年）。

开发成本：2.91\*25000=72750元。

使用基本**COCOMO**模型，ED为总的开发工作量，S为源指令数。常数r和c为校正因子。本项目采用结构体的开发类型。所以参数r为2.4，参数c为1.05。计算得ED=2.4\*=25.9（人月）。

开发成本：25.9\*2083=53958元。

折中考虑项目成本为（72750+53958）/2=63354元。

## 5.3项目成本预算

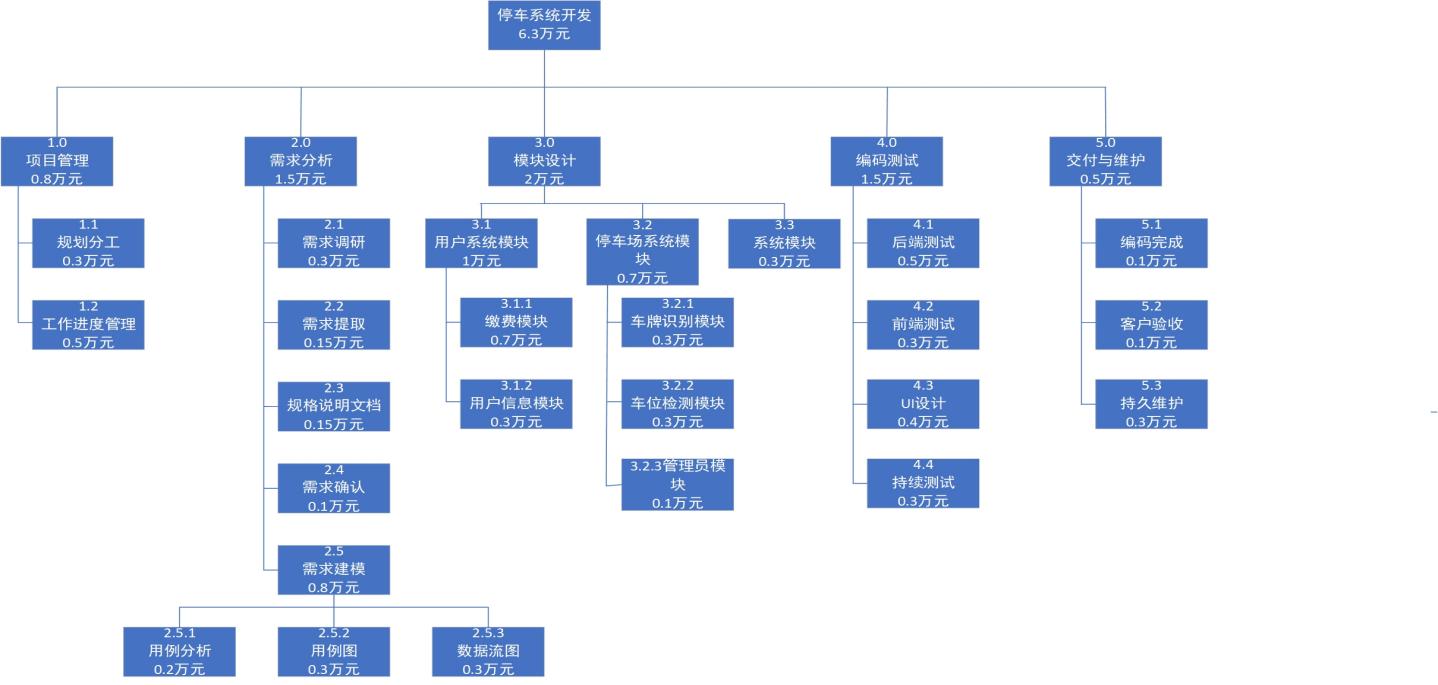
5.3.1分摊总预算成本

图 5-1 总预算成本分配图 单位:万元

5.3.2制定累计预算成本

表 5-2 项目每五天分摊预算与预算累计表 单位：万元

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 活动 | 天 | | | | | | | | | | | | 小计 |
| 1-5 | 6-10 | 11-15 | 16-20 | 21-25 | 26-30 | 31-35 | 36-40 | 41-45 | 46-50 | 51-55 | 56-60 |
| 项目管理 | 0.8 |  |  |  |  |  |  |  |  |  |  |  | 0.8 |
| 需求分析 |  | 0.45 | 0.8 | 0.0375 | 0.1125 | 0.1 |  |  |  |  |  |  | 1.5 |
| 模块设计 |  |  |  |  |  | 1.6 | 0.1 | 0.3 |  |  |  |  | 2 |
| 编码测试 |  |  |  |  |  |  |  |  | 0.5 | 0.5 | 0.5 |  | 1.5 |
| 交付与维护 |  |  |  |  |  |  |  |  |  |  |  | 0.5 | 0.5 |
| 预算累计 | 0.8 | 1.25 | 2.05 | 2.0875 | 2.2 | 3.9 | 4 | 4.3 | 4.8 | 5.3 | 5.8 | 6.3 |  |

从表可以看出，本项目60天的累计量是6.3元人民币。

5.3.3优化基准计划

本项目是智慧停车管理系统，所以重在需求分析、模块设计，在交付与维护阶段也需要工作人员进行调试，因此在这三个活动板块设置了0.5万元的弹性预算。而且项目资金也相对集中于这三个板块，为了使得交付的长期稳定性，可以考虑增加交付与维护的预算。从以往项目的实际工作中可以对项目进度进行优化，例如，加快需求分析与模块设计，将项目的完成时间提前，通过增加后期的编码测试与维护来加强系统的可靠性。

## 5.4项目成本控制

5.4.1 实际成本与盈余累计

表 5-3 项目每五天实际成本累计表 单位：万元

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 活动 | 天 | | | | | | | | | | | | 活动小计 |
| 5 | 10 | 15 | 20 | 25 | 30 | 35 | 40 | 45 | 50 | 55 | 60 |
| 项目管理 | 0.8 |  |  |  |  |  |  |  |  |  |  |  | 0.8 |
| 需求分析 |  | 0.4 | 0.7 | 0.03 | 0.1 | 0.1 |  |  |  |  |  |  | 1.24 |
| 模块设计 |  |  |  |  |  | 1.5 | 0.1 | 0.2 |  |  |  |  | 1.8 |
| 编码测试 |  |  |  |  |  |  |  |  | 0.4 | 0.4 | 0.4 |  | 1.2 |
| 交付与维护 |  |  |  |  |  |  |  |  |  |  |  | 0.5 | 0.5 |
| 每五日成本小计 | 0.8 | 0.4 | 0.7 | 0.03 | 0.1 | 1.6 | 0.1 | 0.2 | 0.4 | 0.4 | 0.4 | 0.5 |  |
| 从项目开始累计成本 | 0.8 | 1.2 | 1.9 | 1.93 | 2.03 | 3.63 | 3.73 | 3.93 | 4.33 | 4.73 | 5.13 | 5.63 |  |

表 5-4 项目每五天实际盈余累计表 单位：万元

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 活动 | 天 | | | | | | | | | | | | 活动小计 |
| 5 | 10 | 15 | 20 | 25 | 30 | 35 | 40 | 45 | 50 | 55 | 60 |
| 项目管理 | 0.8 | 0.8 | 0.8 | 0.8 | 0.8 | 0.8 | 0.8 | 0.8 | 0.8 | 0.8 | 0.8 | 0.8 | 0.8 |
| 需求分析 |  | 0.55 | 1.3 | 1.4 | 1.5 | 1.5 | 1.5 | 1.5 | 1.5 | 1.5 | 1.5 | 1.5 | 1.5 |
| 模块设计 |  |  |  |  |  | 1.6 | 1.8 | 2 | 2 | 2 | 2 | 2 | 2 |
| 编码测试 |  |  |  |  |  |  |  |  | 0.4 | 0.9 | 1.4 | 1.5 | 1.5 |
| 交付与维护 |  |  |  |  |  |  |  |  |  |  |  | 0.5 | 0.5 |
| 累计盈余 | 0.8 | 1.35 | 2.1 | 2.2 | 2.3 | 3.9 | 4.1 | 4.3 | 4.7 | 5.2 | 5.7 | 6.3 |  |

表 5-5 项目3个累计量比较表 单位：万元

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 项目 | 天 | | | | | | | | | | | |
| 5 | 10 | 15 | 20 | 25 | 30 | 35 | 40 | 45 | 50 | 55 | 60 |
| 预算累计 | 0.8 | 1.25 | 2.05 | 2.0875 | 2.2 | 3.9 | 4 | 4.3 | 4.8 | 5.3 | 5.8 | 6.3 |
| 从项目开始累计成本 | 0.8 | 1.2 | 1.9 | 1.93 | 2.03 | 3.63 | 3.73 | 3.93 | 4.33 | 4.73 | 5.13 | 5.63 |
| 累计盈余 | 0.8 | 1.35 | 2.1 | 2.2 | 2.3 | 3.9 | 4.1 | 4.3 | 4.7 | 5.2 | 5.7 | 6.3 |

5.4.2 绩效分析

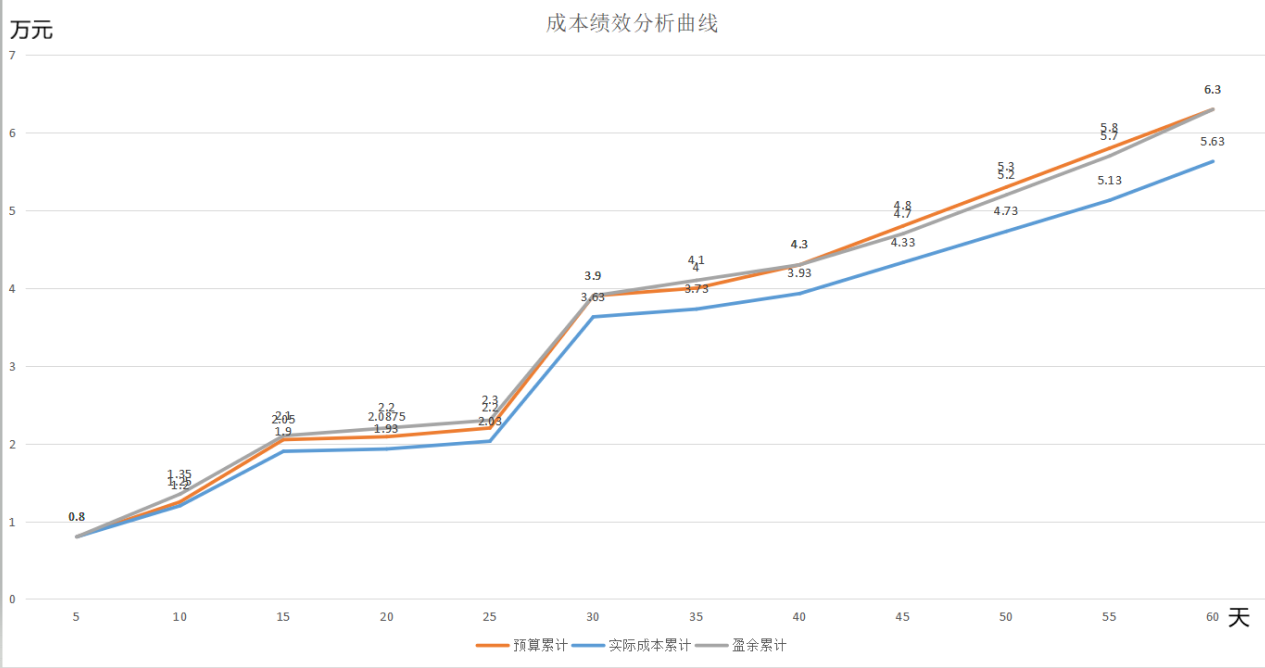


图 5-2 成本绩效分析曲线

PV:计划价值 EV：挣值 AC：实际成本 BAC：完成预算

* 完成预算（BAC）均高于实际成本，说明项目完成效率高于预期值。
* 成本绩效指数CPI=挣值(EV)/实际成本(AC)均大于1，最高为1.125，说明成本计划和进度计划都得到较好的控制。
* 成本差异（CV）=累计盈余量（CEV）－累计实际成本（CAC）,最高为0.57万元，在第60天达成。
* 进度绩效指数（SPI）=挣值(EV)/计划价值(PV)在第四十五天到第五十五天低于1。其余时间大于1.
* CPI和CV随着时间波动上升，说明工效值随工期推进持续增长，预算有盈余，进度超前。
* SPI指数变动说明项目进度在中后期稍差，稍稍未达到预期结果。
* 根据剩余工作的CPI与当前一致。以项目中期（30天）为准。完工尚需成本估算（EAC）=BAC/CPI=3.9/(3.9/3.63)=3.63（万元）。
* 按照完成计划，必须维持的绩效计算。项目中后期（45天）为准，完工尚需绩效指数（TCPI）=（BAC-EV）/(BAC-AC)=（4.8-4.7）/(4.8-4.33)=0.212，表示按照此绩效，很容易完成项目。

基于 **PMBOK7** 12个项目管理原则中的根据环境裁剪以及为实现预期的未来状态而驱动变革的思想，根据一系列数值得出以后类似项目需要做出的调整是保持目前的绩效，在中后期加快进度，保证中后期的进度正常推进。

基于上述绩效分析产生的一些问题，提出以下改进措施。

1. 强化预算过程管控

项目的挣值管理是一种把资源计划编制和进度安排、技术成本以及进度要求相关联的管理技术。它是利用进度计划、成本计划、实际成本这三个开发过程常用且相对独立的因素去进行项目绩效评估。挣值管理也可以预测到项目可能发生的工期滞后量和费用超支量，从而判断项目的进度和成本是否与所作计划保持一致，然后及时采取纠正措施，这样就为项目的成本管理与控制提供了较为有效的手段。[1]

1. 完善动态的敏捷开发流程

在敏捷开发过程中加强需求调研、意见交流、建模、部署等等过程，在保证项目最终成果的同时尽可能缩减项目则的规模。同时，注重团队建设，还要按照需求建立文档，重视测试和营造适宜的开发环境。[2]

1. 建立工作效果考核制度

对员工负责的模块完成的准确性和稳定性采取奖励方案。后续模块包括集成测试、后期维护等将继续采用挣值管理法对项目的进度和成本进行预测和及时调控，确保项目成本和进度处于合理的范围内。

最后，考虑到停车场管理项目存在的特殊性，重点是停车场与车位信息精度与实时交互，为用户提供多种便捷并且符合用户实际需求的方案，可以在编码测试阶段对设备如车辆检测器和一体机、软件和场地的结合提前进行测验，可以减少后期维护的开支。并且在需求分析阶段尽可能地通过与客户交流、实地考察等方法完善方案，避免增加模块设计和编码测试地开支。

**（5.项目成本管理计划由朱可心执笔）**

# 6 项目质量管理计划

软件质量是“反映实体满足明确的和隐含的需求的能力的特性的总和”。具体而言，软件质量是软件符合明确叙述的功能和性能需求、文档中明确描述的开发标准、以及所有专业开发的软件都应具有的和隐含特征相一致的程度[1]。

在PMBOK第6版中，质量管理计划是项目管理计划的组成部分，描述如何实施适用的政策、程序和指南以实现质量目标。它描述了项目管理团队为实现一系列项目质量目标所需的活动和资源。在本次智慧停车系统中，应该在项目早期就对质量管理计划进行评审，以确保决策是基于准确信息的。这样做的好处是，能够尽可能准确的获取客户需求，降低因返工而造成的成本超支金额和进度延误次数。

此次质量管理首先从软件质量的要素与度量说起，就是给定需要管理的内容和方式，再结合技术标准给出具体的质量管理规划。然后从软件项目开发所要进行的四个阶段分别针对不同阶段，采取相应的措施，来保证停车管理系统开发的质量。

## 6.1 软件质量的要素与度量

对项目进行质量管理首先需要分清从哪些不同角度评定软件质量，从管理角度衡量软件质量可以将影响软件质量的主要因素分为三组，这反映了用户在使用软件产品时的三种观点：正确性，健壮性，效率，完整性，可用性，有效性（产品操作）;可理解性，可维护性，灵活性，可测试性（产品修改）；可移植性，可重用性，可操作性（产品转移）。

此次软件质量的需要控制和评估的要素从以上三个大的方向提取，由于本项目是一个以敏捷开发为控制方法，历经多次迭代的需求分析，最终树立“简单且智慧”的开发理念。并且根据成本预算和进度管理，结合此次项目的实际情况，最重要的功能点就是停车场与车位信息精度与实时交互，为用户提供多种便捷并且符合用户实际需求的方案，而在其他方面可以通过简单朴素的界面设计来表现。

“度量是指在现实的世界中，把数字或符号指定给实体的某一属性，以便以这种方式来根据已明确的规则来描述它们”。软件质量的度量就是选择合适的指标来描述软件质量的高与低。

在确定要素以后，需要在项目执行的各个方面对上述要素进行准确的度量。软件度量是对软件开发项目、过程及其产品进行数据定义、收集以及分析的持续性定量化过程，目的在于对此加以理解、预测、评估、控制和改善。在整个软件项目的生命周期，度量活动始终贯穿其中。所以在停车场项目开发的四个主要阶段需求分析阶段、设计阶段、编码阶段和测试阶段，都需要通过度量来判断是否符合质量要求。具体的度量方法会在不同阶段具体分析。

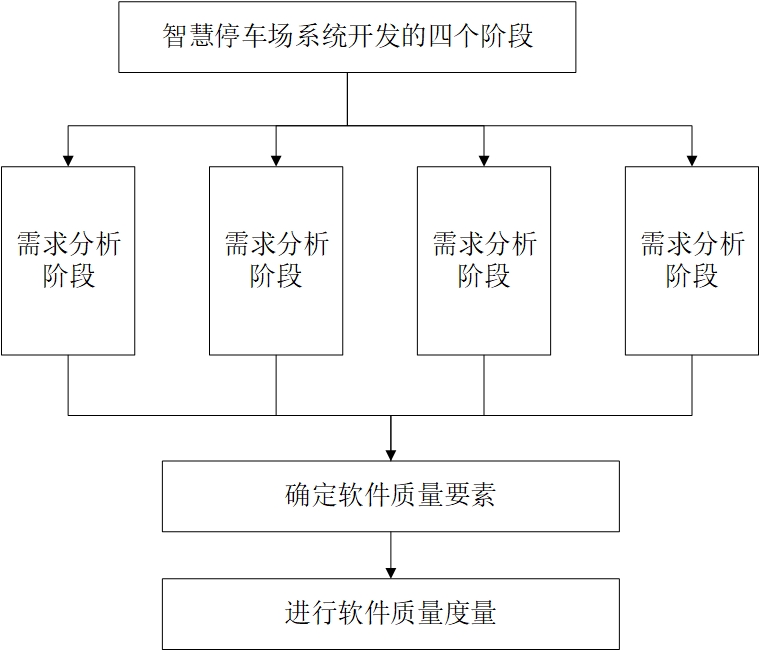


图6-1软件质量控制

## 6.2 技术标准与软件质量规划

智慧停车系统项目在软件开发中遵循的ISO-9001质量指南与软件能力成熟度标准（CMM），为软件开发项目可续试质量管理进行了保障。ISO-9001是迄今为止全世界最成熟的质量框架，而CMM指一个软件过程被明确定义、管理、度量和控制的有效程度，它分为5个过程能力等级分别为：初始级，可重复级，已定义级已定量管理级，优化级。而我们项目过程所要求的达到的是可重复级。

于是，我们将项目质量规划总结如下几点：

1. 正确性：在各种环境下，智慧停车系统必须满足设计要求并向用户及时准确的提供停车场和车位的实时情况，结合如地址等多种因素向用户提供合理的停车方案；
2. 适应性：能够在不同的用户环境下、多变的停车场情况下适应用户的使用需求，并符合软件的质量标准，且可以度量；
3. 易用性：用户界面简洁，操作简单易上手，任何用户都能够直观方便的对程序进行操作并得到结果；
4. 可靠性：停车系统是为了方便用户停车开发，所以如果用户得到的解决方案没有很好的满足到用户，需要对该种情况进行处理，从而确保用户体验；
5. 针对性：在软件设计阶段确立软件质量目标，并在各个实施阶段落实质量目标；
6. 经济性：平衡质量和成本要求，把质量度量和保证所需要的时间成本控制在适当的范围内；

## 6.3 需求阶段质量控制

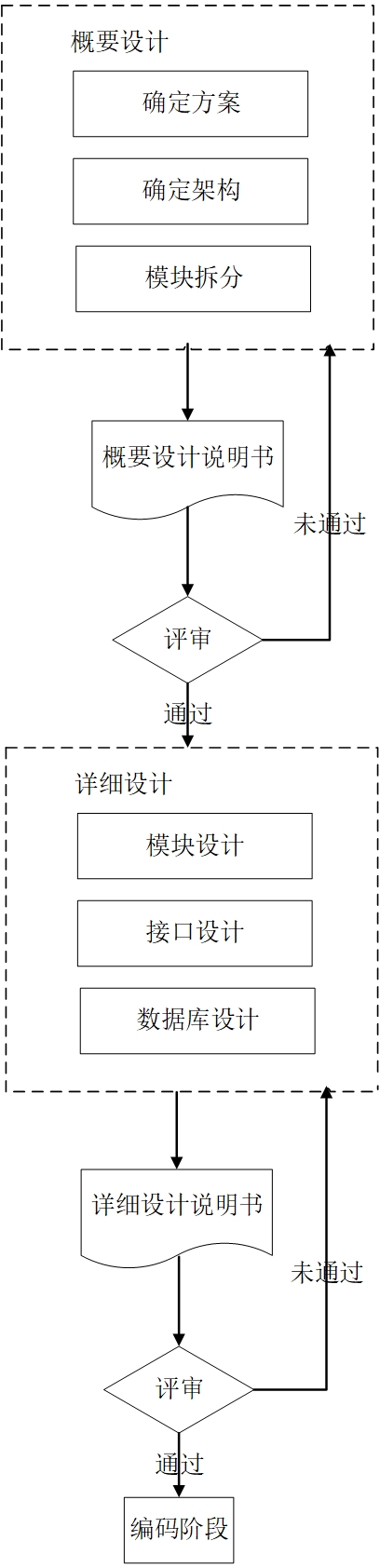
由于需求阶段处于项目活动的早期，在这一阶段出现的问题如果能够及时发现并得到有效解决，对于项目工作的顺利进行有很大的积极作用。否则如果在开发阶段或者产品走查阶段才发现问题，则会对项目的进度造成严重影响，所以对需求阶段的质量控制要做到严格把控。在停车管理系统项目质量管理活动中，主要从文档质量控制和需求过程质量控制来对该阶段的工作产出进行质量控制。

（１）需求文档质量控制。对于需求文档的质量控制主要采用规范化文档结构的思想，在对当前项目需求文档进行检查，查找需求中潜在的漏洞。高质量的需求文档必须能够做到整理产品具体结构、分析核心业务流程、分析及整理非功能性需求等工作。

（２）需求过程质量控制。需求过程质量控制主要包括：需求确认，通过市场调研、竞品分析、用户建议、运营需要等渠道获取需求，并加以汇总分析，得到需求文档。对于停车管理系统而言，对需求的确认和客户要求的具体捕捉是非常重要的，因为这牵扯到多方的利益和广大用户群体的体验，比如在获取停车场实时信息方面，使用什么样的产品（地磁感应装置、摄像机等）能够提高获取信息的灵敏度或者与软件适配性更好。产品经理和开发人员需要严格检查需求说明书，使需求文档中的每一个需求都具有正确性、清晰性、无二义性、一致性、必要性、完整性、可实现性、可验证性、可测性；需求变更控制，随着项目生命周期的推进，产品或开发人员对需求的理解越来越深刻，原来的需求可能有一定的缺陷，导致需要需求变更，对于己经在计划中的需求，需要做到尽量减少需求变更对项目开发进度的影响，否则会降低计划的执行完成度、导致在设计或者开发阶段造成质量下滑。

## 6.4 设计阶段质量控制

软件设计基于软件需求规范。根据需求分析阶段确定的功能设计软件系统的总体结构，划分功能模块，确定每个模块的实现算法，并编写具体代码。形成软件的具体设计方案。软件设计分两步：概要设计和详细设计。概要设计主要关注需求如何转换成数据和软件的整体框架；详细设计关注于将软件整体框架逐步细化为具体的数据结构和软件算法表达。本次智慧停车系统采用敏捷开发模式，设计阶段建立迭代小组和领导人，经过相关人员评审，来保证设计的合理性。项目的设计质量控制流程如图6-2所示：

  
图6-2 智慧停车系统软件设计质量控制流程

## 6.5 编码阶段质量控制

在软件开发项目中，需求分析和系统设计完成之后就进入了软件编码阶段，该阶段是软件从抽象的产品需求到实际的最终实现的重要环节，所以如何对该阶段的质量进行控制，是许多项目管理者的主要关注问题之一。

作为一个实时停车系统的软件，有三个要点需要重点支持：首先，要保证软件的稳定性，由于停车系统的的使用者众多，并且由于停车线路和方案的多变性，如果软件运行不稳定，时常没有跟进数据或出错，会导致糟糕的用户体验，这就要求软件开发过程中的编码的健壮性需要保证；其次，需要保证软件运行高效率，用户一般在停车的时候都会在路况较为紧急的马路上，此时使用软件时，如果没有及时的用户响应，也会对用户造成负面影响，直接导致产品日活的降低，这就要求项目开发中，代码执行效率要进行不断优化；最后，由于停车系统应用的场景越来越多，停车场规模大小不一，因此需求是不断递增的，并且人员在不断变更，所以产品也会不断迭代，这就要求软件开发过程中代码的可读性和可维护行要有保障。

基于以上的软件质量要求，智慧停车系统软件项目编码阶段的质量保障主要通过编码规范和代码走查制度两方面来控制。

（１）编码规范

由于软件开发项目一般由多人协作完成，如果缺乏统一的编码规范，会导致程序后期的可读性、可礼节性变差，直接增加后期维护的工作量，因为即使后期维护过程中，维护人员和开发人员是同一个人，也会由于时间间隔太久，导致原开发思想遗忘等问题，导致人员效率低。设立编码规范还有其他益处：

1.提高代码可读性。编码规范帮助我们编写易于人类理解的代码。您可以获得一些通过名称获取IDE所需的提示，例如可访问性，继承基类等。

2.有助于信息传递，加快工作交接。风格的相似性使开发人员更快，更容易理解一些不熟悉的代码，并更快地了解其他人的代码。因为他和你的代码风格是相同的，所以你不必推测他的一些个人风格。这样做的好处是开发人员可以快速接管项目团队其他成员的工作，并快速完成工作移交。

3.减少命名冲突，降低维护成本。没有采用规范的情况下，可能会产生诸多命名冲突问题，而对于后来的代码维护者而言也可能会令他们感到困惑。

4.减少开发阶段引入缺陷的可能性。

介于以上优点，智慧停车系统项目制定了一系列编码规范，分为前端、服务端、客户端编码规范。主要从编码变量命名、注释、循环控制、函数过程处理、异常处理等对软件编码进行规范。这些规范只是一个全局性质的规范和一种编程约定，开发人员可以参考但是不要求强制遵循。

（２）代码评审

代码评审是的对代码质量的控制活动，虽然现代语言编译器可以检查代码静态编译上的漏洞，但是无法全面的评估代码质量和逻辑问题。代码评审可以在代码提交前发现潜在问题，包括代码设计、命名规范、编程风格、实现错误等多方面问题，并能通过缺陷检测、漏洞检查等工具检查代码问题，降低软件测试成本，减少软件项目风险。

由于智慧停车系统项目在设计服务端、前端、客户端开发，各个端所用语言都不尽相同，所以项目所有开发编写代码推送到代码管理平台前必须经过人工评审且通过代码缺陷检查和代码安全漏洞检查后方可合入。

技术经理在开发进行前明确划分了代码走查的范围和具体责任人，每人在提交代码前，需要发出代码评审请求，其他人在评审其代码后，对其代码进行打分，打分范围为０－２，多人评审打分累积，只有在代码累积评分超过一定分数以后才能合并入远程代码库。

这样就能从入口保证代码质量的控制，代码评审的整体流程如下图6－3所示：

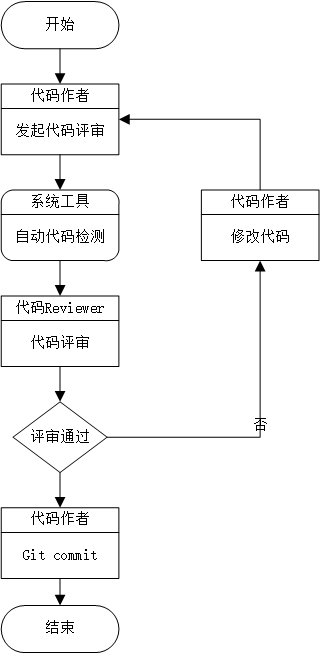


图6-3 评审流程

## 6.6 测试阶段质量控制

软件测试是指在规定的条件下对程序进行操作，以发现程序错误，衡量软件质量，并对其是否能满足设计要求进行评估的过程。在智慧停车系统软件开发项目中，并没有单独的软件测试阶段，假如在完成该项目的编码再准备进行测试时，可能有一个需求阶段的缺陷没有被指正，由此被代入设计阶段后就如同滚雪球一般被放大，再经过编码阶段这一个缺陷可能已经演化成无数的bug需要修改，所以这时候再想测试已经来不及了。

在智慧停车系统项目中，测试阶段贯穿整个项目，测试流程的质量由测试经理来把控。在项目需求阶段，对要执行测试的需求进行分析，确定测试策略，制定测试计划，只有确定了正确的测试策略和指导方针，才能高效、有序的开展后续工作。接下来在需求确定后需要设计测试用例，确保用例完全覆盖智慧停车系统的所有需求。然后在系统测试阶段根据编写好的测试用例对软件进行功能和性能测试，并记录测试过程和测试结果。对于不满足预期的地方，需要输出缺陷报告，系统测试会根据缺陷的多少反复进行，直到所有缺陷得以修复或备注说明。最后需要撰写测试报告，对测试进行分析，总结本次经验教训，在下次工作中改进。整个测试流程如图6-4所示：

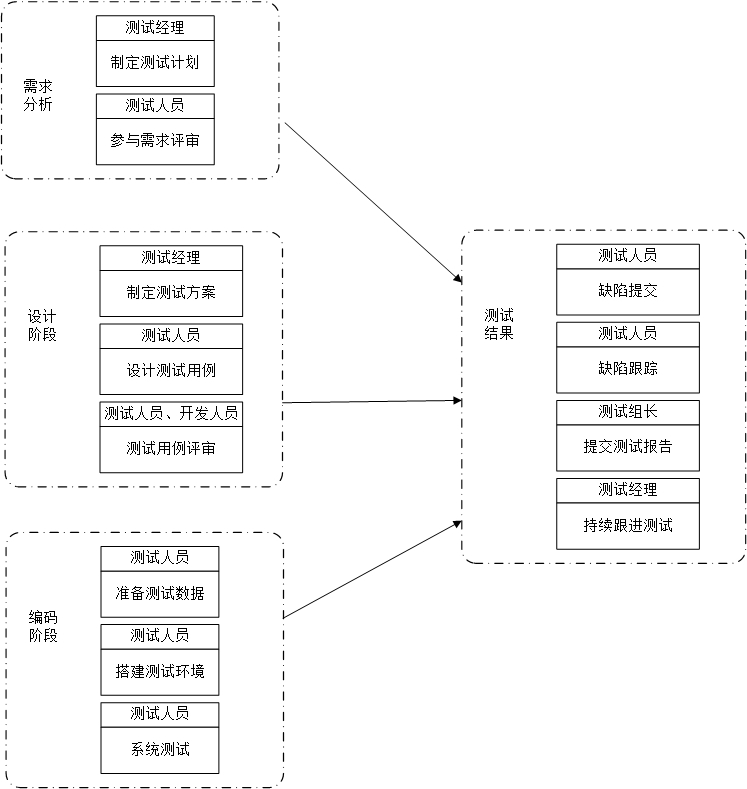


图6-4 测试阶段流程

**（6.项目质量管理计划由XXX执笔）**

# 7 项目风险管理计划

## 7.1 风险识别

自助缴费一般都通过二维码扫码，这里会存在贴在停车场的二维码被人恶意更换的安全风险。这会造成车主缴费了却无法出场。当然这个很少人会去搞，因为只能搞1次，还很容易被抓住。但这确实是一个安全风险。

还有一个就是停车场一般都是地下室，这里手机信号一般比较弱。扫码缴费很容易因为信号不行，车主缴费会出现无法缴费成功，或者缴费延迟，或者重复缴费。这个安全风险需要通过地下停车场的运营商信号覆盖来解决。

## 7.2 风险分析

风险因素分析本项目建设风险因素主要如下：

1. 技术风险，主要指技术路线或设备选型等方面的风险，如关键设备技术成熟度和市场占有率、数据集成及软件开发需要专利技术等。
2. 工程风险，主要指项目管理、工程组织、工程进度方面的风险，如建设单位缺乏工程管理经验、建设单位领导不够重视或协调不力、用户需求变更、项目建设对开发商集成商）工程经验要求较高、项目建设与相关工程间存在较为严格的进度匹配关系、项目建设条件可能发生变化等。
3. 政策风险，主要指政策、管理体制的变化或者调整所带来的风险。
4. 资金风险，主要指资金筹措方面的风险，如配套资金来源不可靠等。
5. 运行风险，主要指系统运行维护保障机制方面的风险，如业务流程重构所需保障机制缺位、信息资源共享存在壁垒、系统安全防护要求较高、建设单位缺乏系统维护专业技术力量或应用人员技术水平较低等。
6. 社会效益风险，在静态交通智能化项目实施过程中，由于涉及到公共服务，可能引起公共服务质量下降的社会效 益风险。

## 7.3 风险跟踪

静态交通智能化项目采取信息系统集成与合作企业购买 服务相结合的运作模式，对公共停车资源进行统一规划、统一标准、统一建设、统一管理和统一收费，为民生提供切实可行的服务保障。

## 7.4 风险应对

本项目建设过程中的风险分配和处理如下。

1. 技术风险对策和管理

一是项目方案采用的先进信息技术，在国内城市有成功案例。二是合作企业购买技术服务。保持系统技术先进性。

1. 工程风险对策和管理

工程风险由通过符合 ISO 9000 系列质量标准控制。

1. 政策风险对策和管理

政策风险由合作企业承担。合作企业方本身具有制定政策的职责和权力。

1. 资金风险对策和管理

资金风险由合作企业财政支持。

1. 运行风险对策和管理

对公共泊车资源进行“公有、公建、公管” 。

1. 投资风险对策和管理

由合作企业财政支持。

**（7.项目风险管理计划由XXX执笔）**

# 8 项目采购管理计划

# 9 项目沟通和冲突管理

## 9.1 沟通管理

书面沟通：主要通过邮件、QQ、微信等工具进行沟通。

口头沟通：每周都有一次口头汇报和讨论。

保持沟通渠道的畅通，尤其是邮件，邮件满了要及时清理，也不要自动回复，导致对方的误解。如果是用QQ、微信沟通，要时不时的留心，不然导致对方无目的的等待。不能屏蔽群或个人，在项目经理发布任务，给出要求的时候，要及时回复，不能假装看不见，或者看见了不回复，及时的反馈可以让对方得知你已经确实收到并确认了。

沟通要求主动沟通和今早沟通，我们要求每一个成员在出现问题或者发现问题的时候能及时和相关人员沟通，避免沟通不及时导致的延期，项目经理也要保持一定的前瞻性，在可能出现问题且能及时解决的时间点去和项目成员进行沟通，避免项目成员埋头苦干，一错再错。项目成员也要主动和交接的人沟通，问题发现的越早，损失越小。尽早的将问题暴露出来，两边都可以解决，避免下一阶段的无用功。

紧急措施，事先做好一个通讯录，在有重要任务且联系不上目标人时，可以通过通讯录打电话，再严重可以直接去家里找。

## 9.2 冲突管理

策略一：在发现冲突时，成员之间尽量回避，以免冲突升级，这时候可以找其他人来调节。

策略二：当决策时或者任务进行时违背了原定的计划，如果结果会严重损害团队的利益，应该及时采取强制措施，掐断冲突，停止任务，按原计划重新开始或者重新规划。

策略三：冲突不是很大且不会严重影响自身的情况下，双方都可以后退半天，团队的利益是最重要的，保持和谐团结是重中之重。

策略四：换位思考，站在对方的角度多想想，反思自己，这是最好的解决方法，希望在项目进行中，每个人都可以达到这个境界，合作共赢。

**（9.项目沟通和冲突管理由XXX执笔）**

# 10 项目管理总结

此次智慧停车系统项目是我们项目组开发的第一个项目，虽然有很多不足的地方，但是对我们组员也是一次崭新的尝试。我们首先通过对市场的分析与调研，通过可行性分析得出了项目可行的结论。然后进入第一个主要阶段需求分析阶段，在该阶段我们充分得了解客户的需求，并且力求创新，最终我们树立了“简单且智慧”的原则，简单意为操作简单、界面干净简单、获取服务简单，智慧则体现在停车是一个动态的过程，并且时刻发生着变化，上一秒没有的车位也许下一秒就成了空位，而我们的项目实时跟进停车场和车位信息，为用户展现最新的信息从而提供最优停车方案，而这也是我们开发这一项目的最初愿景。

与之并行的，我们展开项目团队组织，发扬团队精神，建立绩效考核制度。展开项目进度管理，规划出合理可行的项目实施进度。实施项目成本管理计划，力求在保证质量和完成所有客户需求的前提下，尽可能的降低成本。实施质量管理计划，力求将质量放在第一位，唯有好的质量，软件产品才有更好地立足根基。实施风险管理，力争将风险降到最低。实施采购计划，能够符合软件实施所需的软硬件与场地条件。通过项目沟通和冲突管理，维系开发人员的感情基础，充分的交流对软件项目的开发有十足的好处。

**（10.项目管理总结由XXX执笔）**

# 11 参考文献

吴军刚.软件公司项目成本控制对策研究[J].经营与管理,2019(12):125-127.

张小坚. A公司软件开发项目成本管理研究[D].南京师范大学,2015.

荣鼎慧，程秀才，王蕊，国内软件质量评价服务现状［Ｊ］，软件，