



# NPU BUS

## 软件开发计划

编写人员： 刘龙涛、李云飞、王琪、刘峥、李宗明

校对人员： 刘嘉林、康浩然、苏畅、王泽堃、史明明

编写日期： 2019 年 7 月 2 日

文档版本： 1.0 版

# 目录

1 引言.....	6
1.1 标识.....	6
1.2 系统概述.....	6
1.3 文档概述.....	6
1.4 基线.....	7
2 引用文件.....	7
3 交付产品.....	7
3.1 程序.....	7
3.2 文档.....	7
3.3 服务.....	8
3.4 非移交产品.....	9
3.5 验收标准.....	9
3.5.1 代码验收.....	9
3.5.2 文档验收.....	9
3.5.3 服务验收.....	9
3.6 最后交付期限.....	9
4 所需工作概述.....	10
4.1 文档编号规则: .....	10
4.2 角色分工: .....	10
5 实施整个软件开发活动的计划.....	11
5.1 软件开发过程.....	11
5.2 软件开发总体计划.....	11
5.2.1 软件开发方法.....	11
5.2.2 软件产品标准.....	12
5.2.3 可重用的软件产品.....	13
5.2.4 处理关键性需求.....	13
5.2.5 计算机硬件资源利用.....	14
5.2.6 记录原理.....	14
5.2.7 需方评审途径.....	15
库函数, 模板, 面向对象、设计模式、组件、框架、构架。 .....	13
6 实施详细软件开发活动的计划.....	15
6.1 项目计划和监督.....	15
6.1.1 软件开发计划(包括对该计划的更新) .....	15
6.1.2 CSCI 测试计划.....	16
6.1.3 系统测试计划.....	17
6.1.4 软件安装计划.....	17
6.1.5 软件移交计划.....	17
6.1.6 跟踪和更新计划, 包括评审管理的时间间隔.....	18
6.2 建立软件开发环境.....	18
6.2.1 软件工程环境.....	18
6.2.2 软件测试环境.....	18
6.2.3 软件开发库.....	19

6.2.4 软件开发文档.....	19
6.2.5 非交付软件.....	19
6.3 系统需求分析.....	19
6.3.1 用户输入分析.....	19
6.3.2 运行概念.....	19
6.3.3 系统需求.....	19
6.4 系统设计.....	20
6.4.1 系统级设计决策.....	20
6.4.2 系统体系结构设计.....	21
6.5 软件需求分析.....	21
6.6 软件设计.....	23
6.6.1 CSCI 级设计决策.....	23
6.6.2 CSCI 体系结构设计.....	24
6.6.3 CSCI 详细设计.....	24
6.7 软件实现和配置项测试.....	25
6.7.1 软件实现.....	25
6.7.2 配置项测试准备.....	25
6.7.3 配置项测试执行.....	26
6.7.4 修改和再测试.....	26
6.7.5 配置项测试结果分析与记录.....	26
6.8 配置项集成和测试.....	26
6.8.1 配置项集成和测试准备.....	26
6.8.2 配置项集成和测试执行.....	26
6.8.3 修改和再测试.....	26
6.8.4 配置项集成和测试结果分析与记录.....	27
6.9 CSCI 合格性测试.....	27
6.9.1 CSCI 合格性测试的独立性.....	27
测试分为多块进行, 每块只对一个 CSCI 设计模块进行测试.....	27
6.9.2 在目标计算机系统(或模拟的环境)上测试.....	27
6.9.3 CSCI 合格性测试准备.....	27
6.9.4 CSCI 合格性测试演练.....	27
6.9.5 CSCI 合格性测试执行.....	27
6.9.6 修改和再测试.....	27
6.9.7 CSCI 合格性测试结果分析与记录.....	28
6.10 CSCI/HWCI 集成和测试.....	28
6.10.1 CSCI/HWCI 集成和测试准备.....	28
6.10.2 CSCI/HWCI 集成和测试执行.....	28
将各测试块按准备的类型进行集成、并各自测试.....	28
6.10.3 修改和再测试.....	28
6.10.4 CSCI/HWCI 集成和测试结果分析与记录.....	28
6.11 系统合格性测试.....	28
6.11.1 系统合格性测试的独立性.....	28
6.11.2 在目标计算机系统(或模拟的环境)上测试.....	29
6.11.3 系统合格性测试准备.....	29

6.11.4 系统合格性测试演练.....	29
6.11.5 系统合格性测试执行.....	29
6.11.6 修改和再测试.....	29
6.11.7 系统合格性测试结果分析与记录.....	29
6.12 软件使用准备.....	30
6.12.1 用户现场的版本说明的准备.....	30
6.12.2 用户手册的准备.....	30
6.12.3 在用户现场安装.....	30
6.13 软件移交准备.....	30
6.13.1 可执行软件的准备.....	30
6.13.2 支持手册准备.....	30
6.14 软件配置管理.....	31
6.14.1 配置标识.....	31
6.14.2 配置控制.....	31
6.14.3 配置状态统计.....	31
6.14.4 配置审核.....	32
6.14.5 发行管理和交付.....	32
6.15 软件产品评估.....	32
6.15.1 软件产品评估的独立性.....	32
6.16 软件质量保证.....	32
6.16.1 软件质量保证评估.....	32
6.16.2 软件质量保证记录、包括所记录的具体条目.....	33
6.16.3 软件质量保证的独立性.....	33
6.17 问题解决过程(更正活动).....	34
6.17.1 问题/变更报告.....	34
6.17.2 更正活动系统.....	35
6.18 联合评审(联合技术评审和联合管理评审).....	35
6.18.1 联合技术评审包括---组建议的评审.....	35
6.18.2 联合管理评审包括---组建议的评审.....	35
6.19 文档编制.....	35
6.20 其他软件开发活动.....	35
6.20.1 风险管理, 包括已知的风险和相应的对策.....	35
6.20.2 软件管理指标, 包括要使用的指标.....	38
6.20.3 保密性和私密性.....	40
6.20.4 分承包方管理.....	40
6.20.5 与软件独立验证与确认(IV&V)机构的接口.....	40
6.20.6 和有关开发方的协调.....	40
6.20.7 项目过程的改进.....	40
6.20.8 计划中未提及的其他活动.....	40
7 进度表和活动网络图.....	40
8 项目组织和资源.....	41
8.1 项目组织.....	41
8.2 项目资源.....	41
9 培训.....	42

9.1 项目的技术要求.....	42
9.2 培训计划.....	42
10 项目估算.....	42
10.1 规模估算.....	42
10.2 工作量估算.....	42
10.3 成本估算.....	43
10.4 关键计算机资源估算.....	43
10.5 管理预留.....	43
11 风险管理.....	43
12 支持条件.....	44
12.1 计算机系统支持。.....	44
12.2 需要需方承担的工作和提供的条件。.....	44
12.3 需要分包商承担的工作和提供的条件。.....	45
13 注解.....	45
附录.....	46

# 1 引言

## 1.1 标识

中文名称：《软件开发计划》。

英文名称：“Software Development Plan (SDP)”。

文档版本：“1.0”。

文档编号：“NPU-BUS-SDP-1.0(E)”。

## 1.2 系统概述

本文档适用于“西北工业大学校车管理开发”项目（以下简称“NPUBUS 项目”）的开发过程。NPUBUS 项目，由本小组负责实施，该项目标识号为“NPU-BUS”，其软件产品版本号为“1.0”，包括三个内部版本，分别是 0.1 版、0.2 版和 0.3 版。

项目内容为：

本产品是一款校车服务平台，分为管理层面和用户层面。目的是为了解决近年来，随着

新老校区间的课程安排调度以及每年的学生数量不断增加，乘坐校车在新老校区来往的人数也相应有所增多，对校车工作方的校车调度和发车数量控制有了更高的要求。鉴于校车工作方的数据来源仅仅是根据个人的经验去判断，尚未实现数字化规范化程序化，导致经常有学生等待数量及校车数量之间出现了矛盾，资源未能充分利用好、学生和老师权利不得已正常落实的情况等现实问题。这个平台可以实现给校车工作方提供准确的人数大数据，校车工作方能根据实时情况智能调度学校班车；能够让学生免去苦苦等待校车却上不了校车的情况。

## 1.3 文档概述

本文档依据国家标准《GB/T 8567-2006 计算机软件文档编制规范》制定，属于技术文档，仅限于联合实验室和中兴通讯的项目相关人员阅读。

本文档从项目计划交付产品、软件开发过程、项目计划投入资源，以及项目实施计划这四个方面，详细说明 AndroidUI 项目的软件开发计划。

## 1.4 基线

《软件需求规格说明 (SRS)》

# 2 引用文件

《GB/T 8567-2006 计算机软件文档编制规范》

《NPUBUS 活动网络图》

《NPUBUS 进度表》

# 3 交付产品

## 3.1 程序

NPUBUS 项目计划提交的程序文件包括：

1. 软件 v0.1 版本源代码
2. 软件 v0.2 版本源代码
3. 软件 v0.3 版本源代码
4. 软件 v1.0 版本源代码

## 3.2 文档

AndroidUI 项目计划提交的软件文档包括：

1. 《NPU-BUS-SDP-1.0(E) 软件开发计划》
2. 《NPU-BUS-SUM-1.0 软件用户手册》

3. 《NPU-BUS-SFT-1.0 软件功能列表》
4. 《NPU-BUS-SADD-1.0 软件结构设计说明》
5. 《NPU-BUS-DBDD-1.0 数据库顶层设计设计说明》
6. 《NPU-BUS-SCMP-1.0 软件配置管理计划》
7. 《NPU-BUS-SRS-1.0 软件需求规格说明》
8. 《NPU-BUS -STD-1.0 软件测试说明》
9. 《NPU-BUS -STR-1.0 软件测试报告》
10. 《NPU-BUS -SVD-1.0 软件版本说明》
11. 《NPU-BUS -IDD-1.0 接口设计说明》
12. 《NPU-BUS-SCS-1.0 软件编码规范》
13. 《NPU-BUS-PMP-1.0 项目管理计划》
14. 《NPU-BUS-SDS-1.0 软件文档规范》
15. 《NPU-BUS-DNR-1.0 文档编号规则》

### 3.3 服务

计划提供以下服务：

软件宣传：通过在微信、网页等方式使用户了解到这款产品的便捷性和易用性。

技术支持：对于部分客户，如系统管理员、司机等采取面对面指导。

软件维护：在软件的使用过程中，若遇到合适用户反馈，进行及时的修复。

升级通知：在产品新版本发布以后，及时通知已注册的客户，新版本的改动和特性。



## 3.4 非移交产品

1. 《NPU-BUS-DRN-1.0 文档编号规则》
2. 《NPU-BUS-SDS-1.0 软件文档规范》
3. 《NPU-BUS-SCS-1.0 软件编码规范》
4. 《NPU-BUS-DEC-1.0 开发环境配置》
5. 《会议纪要》

## 3.5 验收标准

### 3.5.1 代码验收

最后交付产品之前进行小组内评审，代码编写符合统一的标准，与文档说明保持一致，代码书写风格统一，采用标准规范，没有下列错误：由于软件缺陷造成数据丢失，不符合设计要求，响应时间太长无法接受等问题。

### 3.5.2 文档验收

最后交付产品之前进行小组内评审，代码编写符合《GB/T 8567-2006 计算机软件文档编制规范》的标准，产品实现的功能与文档所描述的功能列表清单一致，文档清晰易懂，没有语病与歧义。

### 3.5.3 服务验收

服务硬件达到文档说明的要求，人员技术考核合格，定期进行产品的维护与升级。

## 3.6 最后交付期限

从 2019 年 7 月 1 日开始至 2019 年 7 月 11 日，完成对整个系统的可行性分

析、软件开发计划、软件配置管理计划、软件需求规格说明书、软件设计说明（功能设计、流程设计、数据字典、界面设计、数据库设计、接口设计、算法设计等）、软件测试说明书、软件用户手册、软件测试报告等文档对系统各个功能进行详细的设计与实现、数据库结构的设计与构建。最终所有的代码、文档符合统一的标准，交付使用。最后交付期限为 2017 年 7 月 12 日。

## 4 所需工作概述

### 4.1 文档编号规则：

本项目技术文档的编号规则如下：

NPU-BUS-ZZZZ-X.Y(E)

其中：

NPU 是“Northwestern Polytechnical University”的首字母缩写，表示西北工业大学

BUS 是“NPUBUS 软件设计与开发”项目的英文代号；

ZZZZ 为 2~4 个英文字母，是文档名称的英文缩写；

X.Y 用两位数字表示文档的版本号，目前可选的版本号有“0.1”、“0.2”、“0.3”和“1.0”；

(E)为可选项，不填写表示该文档属于保密文档，不得向外泄露，填写则表示该文档可以对外发布。

### 4.2 角色分工：

项目团队共设立如下十个角色：

1. 项目管理
2. 架构设计
3. 需求分析
4. 界面设计

5. 物理设计
6. 程序开发
7. 美术设计
8. 代码评审
9. 软件测试
10. 项目评审

## 5 实施整个软件开发活动的计划

### 5.1 软件开发过程

NPUBUS 项目具有这样的特点：功能明确，功能间相互关系简单，并且需求容易稳定在特定的软件版本之上，逻辑清晰，比较容易划分成独立的模块。因此，其项目开发过程可采用敏捷方法，根据目前流行的 SCRUM 过程规范，进行裁剪定制生成。

### 5.2 软件开发总体计划

#### 5.2.1 软件开发方法

采用敏捷式开发方法，软件项目在构建初期被切分成多个子项目，各个子项目的成果都经过测试，具备可视、可集成和可运行使用的特征。换言之，就是把一个大项目分为多个相互联系，但也可独立运行的小项目，并分别完成，在此过程中软件一直处于可使用状态。

1. 我们首先需要确定一个 Product Backlog（产品需求列表），这个是由 PO 负责的（

2、有了 Product Backlog 列表，我们需要通过 Sprint Planning Meeting（Sprint 计划会议）来从中挑选出一个 Story 作为本次迭代完成的目标，这个

目标的时间周期是 1 个星期，然后把这个 Story 进行细化，形成一个 Sprint Backlog；

3、Sprint Backlog 是由 Scrum Team 去完成的，每个成员根据 Sprint Backlog 再细化成更小的任务（细到每个任务的工作量在 2 天内能完成）；

4、在 Scrum Team 完成计划会议上选出的 Sprint Backlog 过程中，需要进行 Daily Scrum Meeting（每日站立会议），每次会议控制在 15 分钟左右，每个人都必须发言，并且要向所有成员当面汇报你昨天完成了什么，并且向所有成员承诺你今天要完成什么，同时遇到不能解决的问题也可以提出，每个人回答完成后，要走到黑板前更新自己的 Sprint burn down

5、做到每日集成，也就是每天都要有一个可以成功编译、并且可以演示的版本。

6、当一个 Story 完成，也就是 Sprint Backlog 被完成，也就表示一次 Sprint 完成，这时，我们要进行 Sprint Review Meeting（演示会议），也称为评审会议，产品负责人和客户都要参加，每一个 Scrum Team 的成员都要向他们演示自己完成的软件产品。

7、最后就是 Sprint Retrospective Meeting（回顾会议），也称为总结会议，以轮流发言方式进行，每个人都要发言，总结并讨论改进的地方，放入下一轮 Sprint 的产品需求中；

## 5.2.2 软件产品标准

软件编码规范：

《NPU-BUS-SCS-1.0 软件编码规范》 见附录

软件文档规范

《NPU-BUS-SDS-1.0 软件文档规范》 见附录

文档编号规则

《NPU-BUS-DRN-1.0 文档编号规则》 见附录

## 5.2.3 可重用的软件产品

### 5.2.3.1 吸纳可重用的软件产品

标识、评估和吸纳可重用软件产品要遵循的方法：

1. 系统地开发可重用的软部品。这些软部品可以是代码，但不应该仅仅局限在代码。我们必须从更广泛和更高层次来理解，这样才会带来更大的重用收益。比如软部品还可以是：分析，设计，测试数据，原型，计划，文档，模板，框架等等。

2. 系统地使用这些软部品作为构筑模块，来建立新的系统。

搜寻这些产品的范围：已开发并开源的支持复用的软件源码。

进行评估的准则，以及同时应给出与使用有关的优点、缺陷和限制。重用的级别一般可分为代码重用、设计重用、规范重用和概念重用。设计重用指使用在不同环境下构造的设计。此时应考虑环境中硬件或性能约束带来的影响，从相同的设计可以得到不同的代码。规范重用有同设计重用相似的特征。但从相同的规范可得出不同的设计和不同的代码。代码重用有非常明显的效益，它能大量节省程序设计人员的时间，因此为一般人所接受。但代码重用的问题只能发生在编码阶段，在软件开发过程的早期阶段无法确定是否要重用以及重用带来的影响。在对设计重用的基本理论进行研究的基础上，建立了能够记录概念设计历史的基于功能-行为-结构框架的概念设计信息模型，实现了功能、行为和结构的形式化表示，建立了多粒度层次的概念设计重用框架，才系统成为概念设计重用的系统平台。

### 5.2.3.2 开发可重用的软件产品

库函数，模板，面向对象、设计模式、组件、框架、构架。

## 5.2.4 处理关键性需求

1. 提高软件生成率。

2. 缩短开发周期 。
3. 降低软件开发和维护费用。
4. 生产更加标准化的软件。
5. 提高软件开发质量。
6. 增强软件系统的互操作性。
7. 减少软件开发人员数量。
8. 使开发人员能比较容易的适应不同性质的项目开发。

#### 5.2.4.1 安全性保证

暂无

#### 5.2.4.2 保密性保证

遵循本团队保密协议。

#### 5.2.4.3 私密性保证

本团队承诺该保护用户隐私,并做好安全系统防御,不会泄露用户个人信息。

#### 5.2.4.4 其他关键性需求保证

暂无

### 5.2.5 计算机硬件资源利用

本团队笔记本均可满足资源利用。

### 5.2.6 记录原理

1. 源代码模块或者类一级的重用,这是最基本的软件重用形式记录。
2. 二进制形式的重用。如组件重用记录。
3. 组装式重用。比如:把好几个应用程序的功能集成在一起。例如,要建

立校车门户站点应用，登陆用户既可以查询天气情况，又可以查看校车情况，还可以在线预约。这些功能由不同网络应用服务供应商提供，通过这种组装式重用，就可以非常容易地把上述功能都集成到新的门户站点中。

4. 分析级别重用记录。
5. 设计级别重用记录。
6. 软件文档重用记录。

## 5.2.7 需方评审途径

1. 首先进行域评审。根据应用领域的特征及相似性预测软部件的可重用性。
2. 然后进行软部件的开发。一旦确认了软部件的重用价值，即可进行软部件的开发并对具有重用价值的软部件进行一般化，以便它们能够适应新的类似的应用领域。
3. 最后，软部件及其文档即可进入软部品库，成为可供后续项目使用的可重用资源。

# 6 实施详细软件开发活动的计划

## 6.1 项目计划和监督

### 6.1.1 软件开发计划(包括对该计划的更新)

#### (1) 编写目的

为了便利学校教师及同学的校车乘坐情况，方便学校对校车司机和学生志愿者的管理和调控，决定编写此校车管理项目。适用人群为学生、教师、校车司机及学校管理部门。

#### (2) 背景

待开发的软件系统为校车管理系统。

本项目的任务提出者和开发者均为第五组成员。

用户为本校学生、教师、司机及校管理部门。

在 eclipse 上创建 Dynamic Web Project 项目，采用 MVC 设计模式编写该软件。连接本地 MySQL 数据库，使用 JDBC 保持数据持久化。

(3) 定义:

MVC:MVC 全名是 Model View Controller，是模型(model)－视图(view)－控制器(controller)的缩写，一种软件设计典范，用一种业务逻辑、数据、界面显示分离的方法组织代码，将业务逻辑聚集到一个部件里面，在改进和个性化定制界面及用户交互的同时，不需要重新编写业务逻辑。

CSCI:CSCI 是计算机软件配置项(Computer Software Configuration Item)简称

(4) 参考资料

《深入解析 Spring MVC 与 Web Flow》Seth Ladd

Spring MVC 学习指南/（美）戴克著，林仪明，崔毅

(5) 工作内容

校车管理系统功能的全面实现与相关文档编写

(6) 主要参与人员:

第五组成员

(7) 时间期限

完成项目的时间应该在 7 月 12 日之前。

## 6.1.2CSCI 测试计划

CSCI 合格性测试从以下几个方面入手进行测试:

- (1) 文档审查
- (2) 静态分析
- (3) 内存使用缺陷测试
- (4) 功能测试
- (5) 性能测试
- (6) 余量测试



### (7) 人机界面测试

在校车管理系统测试之前，先检查系统的基础功能有没有实现，再进行编写测试用例。在测试完成修改后进行回归测试，并记录 CSCI 测试的结果。

## 6.1.3 系统测试计划

(1) 被测对象的版本：

v1.0

(2) 测试的任务划分：

计划、设计、实现、执行

(3) 被测对象：

校车管理系统所用功能与非功能需求（如登录、注册、登出）

(4) 测试通过失败标准：

用例的执行情况要达到 1、2 级百分百执行，3、4 级百分之五十执行  
覆盖率要达到所有的功能需求、性能需求都被覆盖

(5) 测试挂起恢复标准

挂起：测试无法进行下去或失去了继续测试的意义时，将测试活动挂起

恢复：挂起的条件满足时，将测试活动恢复。

## 6.1.4 软件安装计划

由保密人员在指定地点进行软件交付，并对安装方式、详细过程予以说明

## 6.1.5 软件移交计划

本软件经由本组人员便已编写完毕

### 6.1.6 跟踪和更新计划，包括评审管理的时间间隔

(1) 为每一份功能/非功能需求提供一个需求追踪表的表栏，并将记录版本信息便于查看更新记录。

(2) 评审管理间隔为每日一次小评，五日一次大评

## 6.2 建立软件开发环境

### 6.2.1 软件工程环境

数据库服务器端

(1) 操作系统: Microsoft Windows 2003

(2) 数据库管理系统: SQL Server 2005, 配置 TCP/IP 协议

Web 服务器端

(1) 操作系统: Microsoft Windows 2003

(2) Internet 信息服务 (IIS) 6.0 管理器

(3) Visual Studio.NET 2005, 配置 TCP/IP 协议

客户端

(1) 操作系统: Windows 7/8/10

(2) Web 浏览器: Internet Explorer 6.0 以上或 Netscape 4.0

以上, 配置 TCP/IP 协议

### 6.2.2 软件测试环境

使用集成开发工具 Eclipse5.5.1, 项目运行环境为 JDK6.0.

其他开发工具包括: Dreamweaver, Microsoft Visio, Rational Rose, Power Designer Trial 11, TomCat6.0 CVSNT2.5.03

### 6.2.3 软件开发库

使用集成开发工具 Eclipse5.5.1，项目运行环境为 JDK6.0.

### 6.2.4 软件开发文档

word、wps 均可

### 6.2.5 非交付软件

本软件全部交付给客户，并提供后续维护服务

## 6.3 系统需求分析

### 6.3.1 用户输入分析

用户输入有以下类型：

用户名、密码、反馈信息（遇到的问题方便维护）

### 6.3.2 运行概念

系统运行主要有学生/老师、志愿者、司机、管理员四个板块，各个模块之间可以进行信息交互，用户登录时选择进入的模块即可实现对应的功能

### 6.3.3 系统需求

系统功能：

0. 用户登陆注册（记住密码）（忘记密码）（不同用户进入不同界面）
1. 用户反馈（遇到的问题方便维护）

2. 天气、日历（点击日历上的日期查看当天的发车信息）；

用户分类：

老师|学生： 3. 预约取消（查看自己已预约得车辆的发车信息） 4. 查询发车时间、及车上剩余位置；（老师预约比学生有优先级）

志愿者： 5. 查询预约情况； 6. 更新车上剩余数量（有人排队上车） 7. 向管理员发出增发车辆信息

司机： 8. 接受调度信息、 9. 请假

管理员： 10. 更新校车发车信息（安排校车时间、数量、接受司机请假信息安排新的校车）

## 6.4 系统设计

### 6.4.1 系统级设计决策

(1) 功能列表：

0. 用户登陆注册（记住密码）（忘记密码）（不同用户进入不同界面）

1. 用户反馈（遇到的问题方便维护）

2. 天气、日历（点击日历上的日期查看当天的发车信息）；

3. 预约取消（查看自己已预约得车辆的发车信息）（老师学生）

4. 查询发车时间、及车上剩余位置；（老师预约比学生有优先级）（志愿者）

5. 查询预约情况；（志愿者）

6. 更新车上剩余数量（有人排队上车）（志愿者）

7. 向管理员发出增发车辆信息(志愿者)

8. 接受调度信息（司机）

9. 请假（司机）

10. 更新校车发车信息（安排校车时间、数量、接受司机请假信息安排新的校车）（管理员）

(2) 界面设计决策

采用 jsp、servlet 结合的方式，实现基于动态 web 页面的各项功能

## 6.4.2 系统体系结构设计

系统结构运行分为学生/老师、志愿者、司机、管理员四个模块，各个模块之间可以进行信息交互，用户登录时选择进入的模块即可实现对应的功能

## 6.5 软件需求分析

- 1、在开发初期，要想得到一个完整准确的规格说明不是一件容易的事
- 2、用户往往对系统只有一个模糊的想法，很难完全准确的表达对系统的全面要求

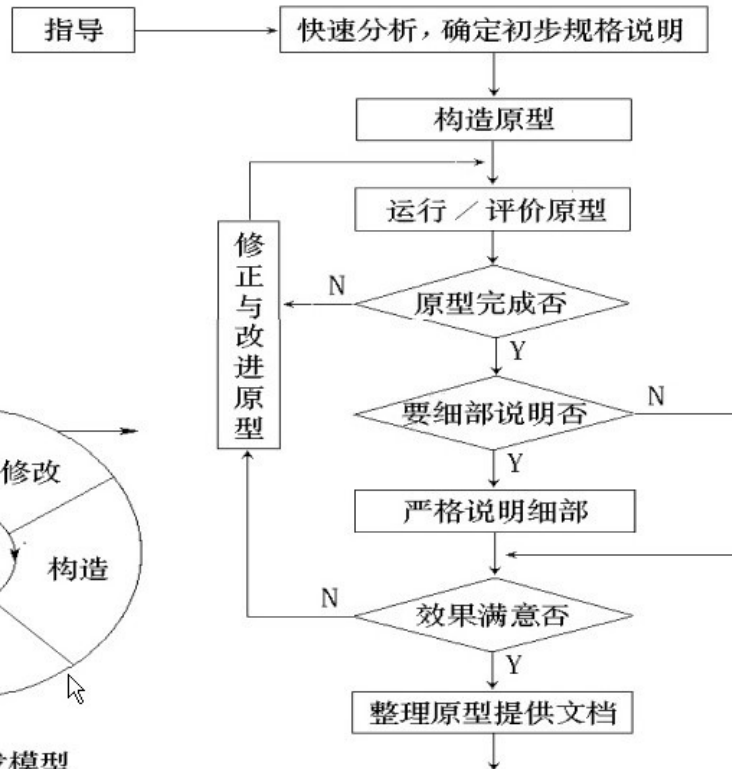
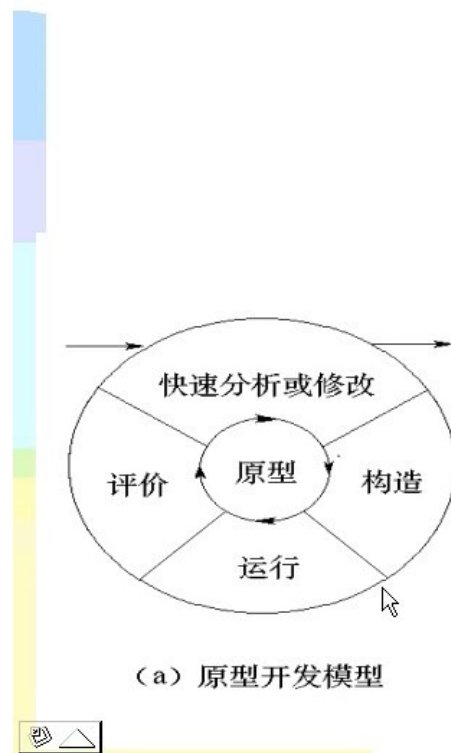
两类：

- 1、探索型
- 2、实验型
- 3、进化型

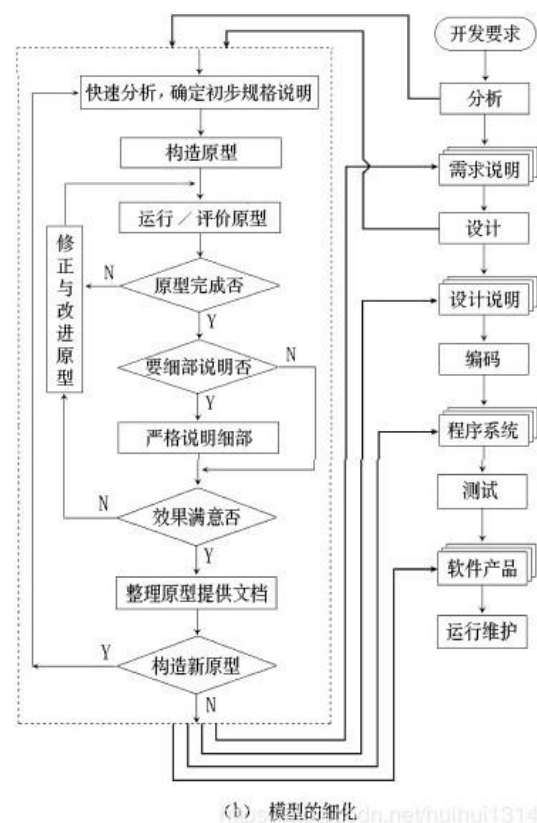
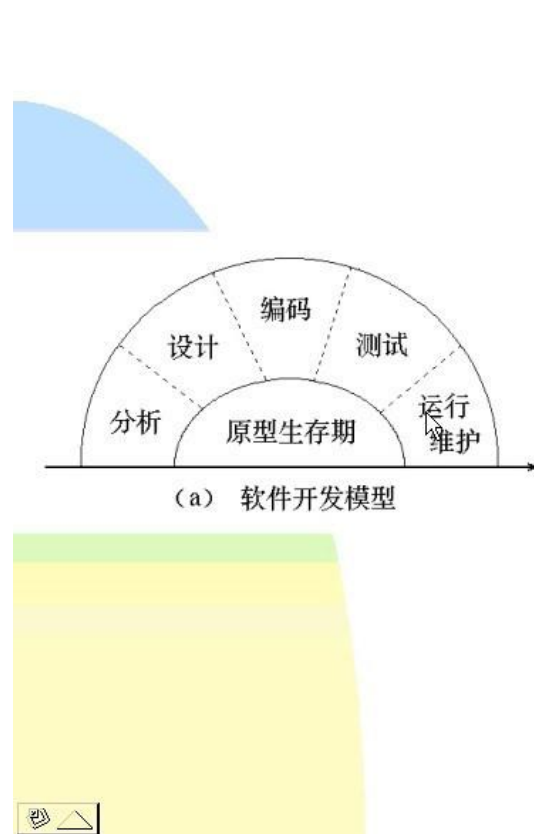
使用策略：

- 1、废弃策略
- 2、追加策略（常用）

开发模型：



软件:



# 6.6 软件设计

本条应分成若干分条描述软件设计中所遵循的方法。各分条的计划应覆盖合同中论及它的所有条款。

## 6.6.1 CSCI 级设计决策

- a. 关于 CSCI 将接收的输入、将产生的输出、要执行的动作的设计决策，
- | 输入         | 输出     | 执行动作                        |
|------------|--------|-----------------------------|
| (登录)用户名+密码 | 无      | 查看数据与数据库中信息是否匹配，<br>匹配则进入页面 |
| (注册)用户名+密码 | 无      | 将信息存入数据库                    |
| 反馈信息       | 无      | 将信息存入数据库                    |
| 中英文切换      | 中/英文页面 | 切换至中文/英文                    |
| 预约/取消预约    | 提示成功   | 将信息存入数据库                    |
| 查询校车信息     | 校车信息   | 将信息从数据库取出输出到界面              |
| 查询预约情况     | 预约情况   | 将信息从数据库取出输出到界面              |
| 更新座位剩余信息   | 更新好的信息 | 将信息存入数据库                    |
| 接受调度信息/请假  |        | 将信息存入数据库                    |
| 安排发车信息     | 安排好的信息 | 将信息存入数据库                    |
- b. 有关响应每个输入或条件的 CSCI 行为的设计决策，包括 CSCI 要执行的动作、响应时间和其他性能特性，模型化的物理系统的说明，选定的方程式/算法/规则，以及对不允许的输入或条件进行的处理。如 a 所示
- c. 有关数据库/数据文件如何呈现给用户的设计决策：  
通过查询或输出到页面的方式呈现
- d. 为满足安全性和保密性需求所选择的方法。
- e. 为满足需求所做的其他 CSCI 级设计决策：  
为提供所需的灵活性、可用性和可维护性所选择的 MVC 开发模式

## 6.6.2 CSCI 体系结构设计

系统结构运行分为学生/老师、志愿者（调度员）、司机、管理员四个模块，各个模块之间可以进行信息交互，用户登录时选择进入的模块即可实现对应的功能

## 6.6.3 CSCI 详细设计

a. 构成该 CSCI 的所有软件单元。

登录页面、注册页面、学生/老师页面、管理员页面、司机页面、调度员页面

b. 软件单元的静态关系。

登陆页面由用户名、密码输入框、跳转至注册页面的按钮、登录按钮、中英文切换按钮、组成

注册页面由信息输入框、注册按钮组成

学生/老师页面由预约相关按钮、查询校车信息按钮、退出按钮组成

调度员页面由查询预约情况模块、更新座位信息模块、退出按钮组成

司机页面由接受调度信息/请假模块、退出按钮组成

管理员页面由校车安排模块、退出按钮组成

c. 每个软件单元的用途

登陆页面由用户名、密码输入框、跳转至注册页面的按钮、登录按钮、中英文切换按钮、组成

注册页面： 用户登陆注册（记住密码）（忘记密码）（不同用户进入不同界面）天气、日历（点击日历上的日期查看当天的发车信息）；用户反馈（遇到的问题方便维护）；

学生/老师页面： 预约取消（查看自己已预约得车辆的发车信息）查询发车时间、及车上剩余位置；（老师预约比学生有优先级）

调度员页面： 查询预约情况；更新车上剩余数量（有人排队上车）；

司机页面： 接受调度信息、请假



管理员页面 安排校车发车信息（安排校车时间、数量、接受司机请假信息安排新的校车）根据实际情况（排队人数、剩余的车辆信息）做出判断，自动进行发车信息的更新（如增发车辆）

## 6.7 软件实现和配置项测试

### 6.7.1 软件实现

在 eclipse 上创建 Dynamic Web Project 项目，采用 MVC 设计模式编写该软件。连接本地 MySQL 数据库，使用 JDBC 保持数据持久化。根据不同的任务分配，项目团队的每个成员先完成自己的代码，然后进行各自的白盒测试，编写《测试用例文档》。最后拼接并进行系统测试。

应当在每个函数的编写之后执行单元测试，因此需要提前完成与数据库的连接，并在数据库中录入测试数据，这是单元测试的准备。然后编写简单的测试代码开始执行单元测试。如果对数据库的修改符合预期结果，则单元测试通过并做出相应记录；如果不符合预期结果则表明测试失败，也要做出相应记录。测试失败后要进行白盒测试分析，修改源代码并重新测试，即回归测试。每个回归测试的步骤与第一次单元测试的步骤相同，直至测试通过为止。

### 6.7.2 配置项测试准备

准备好待测试项：外部交付的软件产品和数据、指定的内部软件工作产品和数据、指定的用于创建或支持软件产品的支持工具（集成开发工具 Eclipse，项目运行环境为 JDK6.0.）、典型配置项包括项目计划书、需求文档、设计文档、源代码、可执行代码、测试用例、运行软件所需的各种数据，它们经评审和检查通过后进入配置管理。

### 6.7.3 配置项测试执行

测试配置项的各个单元的运行情况，并记录结果。

### 6.7.4 修改和再测试

对测试中本应正常工作却出错的情况进行记录

在对错误情况进行再测试明确错误。

### 6.7.5 配置项测试结果分析与记录

将测试结果列表保存记录，分析暂无法预知

## 6.8 配置项集成和测试

### 6.8.1 配置项集成和测试准备

将已经测试的各个配置项分类准备集成到一起准备进行测试

测试块：管理文档（测试为查验）、代码和数据（通过编写的测试用例测试）

### 6.8.2 配置项集成和测试执行

首先按测试块分块对每个测试块进行测试即可

### 6.8.3 修改和再测试

对测试中本应正常工作却出错的情况进行记录

在对错误情况进行再测试明确错误。

## **6.8.4 配置项集成和测试结果分析与记录**

将测试结果列表保存记录，分析暂无法预知

## **6.9 CSCI 合格性测试**

### **6.9.1 CSCI 合格性测试的独立性**

测试分为多块进行，每块只对一个 CSCI 设计模块进行测试

### **6.9.2 在目标计算机系统(或模拟的环境)上测试**

对每个支持的 windows、主流浏览器进行测试

### **6.9.3 CSCI 合格性测试准备**

各个支持版本的 windows 系统、主流浏览器，准备好各个 csci 设计模块的相关功能部分。

### **6.9.4 CSCI 合格性测试演练**

初步测试查看整体是否合格

### **6.9.5 CSCI 合格性测试执行**

对每个 csci 设计模块进行测试

### **6.9.6 修改和再测试**

对测试中本应正常工作却出错的情况进行记录

在对错误情况进行再测试明确错误。

## **6.9.7 CSCI 合格性测试结果分析与记录**

将测试结果列表保存记录，分析暂无法预知

## **6.10 CSCI/HWCI 集成和测试**

### **6.10.1 CSCI/HWCI 集成和测试准备**

将已经测试的各个配置项按功能分块准备集成到一起准备进行测试

测试块：教师学生模块、管理员模块、司机模块、调度员模块

### **6.10.2 CSCI/HWCI 集成和测试执行**

将各测试块按准备的类型进行集成、并各自测试

### **6.10.3 修改和再测试**

对测试中本应正常工作却出错的情况进行记录

在对错误情况进行再测试明确错误。

### **6.10.4 CSCI/HWCI 集成和测试结果分析与记录**

将测试结果列表保存记录，分析暂无法预知

## **6.11 系统合格性测试**

### **6.11.1 系统合格性测试的独立性**

作为软件测试的最后一步，系统测试基于之前的各项测试，进行整合测试

### **6.11.2 在目标计算机系统(或模拟的环境)上测试**

对每个支持的 windows、主流浏览器进行测试

### **6.11.3 系统合格性测试准备**

已经经过测试的软件系统以及文档系统

### **6.11.4 系统合格性测试演练**

初步测试主要功能，查看是否与预期相符

### **6.11.5 系统合格性测试执行**

对每个功能需求逐个进行测试

### **6.11.6 修改和再测试**

对测试中本应正常工作却出错的情况进行记录  
在对错误情况进行再测试明确错误。

### **6.11.7 系统合格性测试结果分析与记录**

将测试结果列表保存记录，分析暂无法预知

## 6.12 软件使用准备

### 6.12.1 用户现场的版本说明的准备

本文档适用于“校车管理系统”的开发过程。该项目的标识号为“”，其软件产品版本号为“1.0”，包括三个内部版本，分别是 0.1 版、0.2 版和 0.3 版。

项目内容为：

基于 windows10 系统版本，针对当前学生教师乘坐校车存在不方便情况而提出的。从而方便用户使用及管理员调控的校车管理系统。

### 6.12.2 用户手册的准备

首先提供用户注册账户，及登录使用的教程介绍。其次，需要对用户明确其作为软件使用者的身份，即学生、教师、司机、调度员或管理者。

对不同身份的用户，在用户手册的不同部分进行分别介绍，着重介绍不同身份在使用上的差别。

### 6.12.3 在用户现场安装

先对 mysql 数据库的数据表进行设置，接着可以让现场用户从注册开始，进行使用网站的尝试。进而得到用户反馈。

## 6.13 软件移交准备

### 6.13.1 可执行软件的准备

登录网页即可使用。

### 6.13.2 支持手册准备

对用户使用的全过程（从注册开始）进行详细必要的解说，保证不同用户可

以最大限度的上手新软件。

## 6.14 软件配置管理

### 6.14.1 配置标识

项目组人员将要标识或已标识的配置项提交给软件配置管理负责人，由软件配置管理负责人统一管理，并填写“配置状态报告”。

开发部门在开发过程中要向软件配置管理负责人提交基准配置项，由软件配置管理负责人管理基准配置项，并及时填写“配置状态报告”。

### 6.14.2 配置控制

#### (1) 更改请求的提出及审批

如果需要对配置项进行修改，客户或开发部门按照“配置更改单”的格式填写更改请求说明和更改评估。一般情况下，更改评估要考虑更改对其它配置项的影响及更改的效果；如果更改较大，还要评估更改对时间和成本的影响。如果更改是为了增加需求，还要将报价单交给客户审批。填好后，提交给软件配置管理负责人，由软件配置管理负责人组织相关部门评审。

建议的更改被批准，进行下一步的实施更改工作；否则，终止更改。由软件配置管理负责人将“配置更改单”归档。

#### (2) 更改的实施跟踪与记录

更改被批准后，“开发计划”、“配置管理计划”等文档也要进行相应的更改，项目管理部门要对更改的实施进行跟踪。项目组人员要按照更改过的开发计划表、配置管理计划表提交配置项，软件配置管理负责人管理这些配置项，重新标识所有被影响的配置项及版本。

### 6.14.3 配置状态统计

每个项目已完成的配置项应在“配置状态报告”中登记，以便及时跟踪各项目的配置情况。尤其要注意对更改的基准配置项及其受影响的配置项的标识，明

确基准配置项的状态。

#### **6.14.4 配置审核**

《配置管理计划》由软件配置管理负责人审批。

#### **6.14.5 发行管理和交付**

发行的是完成版本，后续会对用户反馈及软件使用中的问题进行修复和更新。

### **6.15 软件产品评估**

#### **6.15.1 软件产品评估的独立性**

各功能模块之间，并无功能的嵌套，可单独编写、测试。各个产品新旧之间运行独立。

### **6.16 软件质量保证**

#### **6.16.1 软件质量保证评估**

##### **1、需求复杂度+代码量**

这里的需求复杂度主要是指需求实现的难易程度，需要综合考虑开发对需求实现难易的程度、测试对需求可测试性评估程度、是否有成熟的方案、代码是否复用等。通常来说需求实现越复杂，代码量大的版本风险较高。

##### **2、人力成熟度**

人力成熟度，需要综合考虑 PM、开发、测试人力的成熟度。人力越成熟，开发、测试的风险就会降低。如果是新手开发的模块，就需要特别重视了。当然，在开发、测试分工时，开发经理和测试经理通常是回根据需求复杂度和人力成熟度进行相应的分工。降低版本质量风险。

##### **3、版本开发测试流程**

版本开发测试流程，包括开发团队和测试团队工作的测试流程，从需求进



入研发团队到版本发布上线。包括各个阶段的出、入口准则和工作内容、版本发布的准则等。这里重点描述几个环节的出、入口条件和注意事项。

#### 4、需求变更

需求变更在软件开发测试过程都是难以避免，在需求变更管控流程规范下积极面对。做好需求变更管理，分析引入变更对现有交付范围、影响，刷新测试策略和计划。确保风险可控。不能盲目的、一味的接受，避免到最后风险不可控。

### 6.16.2 软件质量保证记录、包括所记录的具体条目

### 6.16.3 软件质量保证的独立性

(1) 需求准入和传递。需求已签字和所有参与该版本的测试、开发及其他相关人员均参与了需求传递。可以开展开发需求反串讲和测试需求反串讲活动保证开发、测试人员均熟悉理解需求内容。

(2) 代码检视。首先确保检视人员，通常是项目组的核心骨干，熟悉业务特性和代码框架。安排检视活动时间，有些项目组会忽略这个活动。同时，输出检视结果，会起到一定的督促效果，也便于回顾。

(3) 代码工具扫描。确保项目组规定的代码扫描工具完成，并提供扫描报告，要求扫描结果通过。可以减少代码低级 bug、降低圈复杂度，保证代码质量。同时也可以降低后续测试人员用例设计代码覆盖的难度。

(4) 测试分层出入口条件。严守各个环节的出入口条件。测试分层包括 UT/IT/MST/BBIT/SDV/SIT/SVT 等，通常不会分的这么细，根据自己项目情况进行变更。确保各个环节的工作内容、测试对象都已达到出口准则。分析各个环节的执行情况和交付件，调整下一环节的测试策略，确保已知风险尽早解决。

(5) 测试用例评审。确保核心人员参与，特别是 SE、开发 Leader、测试 Leader 以及核心人员、版本维护人员等。评审形式采用尽量采用会议形式。对于电话会议或者邮件评审，酱油居多，不建议采用，其他特殊情况特殊考虑。评审闭环，确保意见全部修复。

(6) 测试策略和测试计划评审。核心人员必须参与评审，包括 pm、开发 Leader、

测试 Leader 及相关人员等。确保测试任务分工、测试时间合理。不出现明显的时间压缩，因为版本计划通常都是根据发布计划倒排的，所以测试 Leader 必须有主见，对于测试时间有风险必须及时提出。

(7) 测试报告评审。测试报告评审通常都有规定的人员参与。希望到这一步骤时在任何人提出质疑时都可以准确答复质疑。对于版本的质量能有一个相对清晰的描述。虽然很多描述还是依靠主观的。

## 6.17 问题解决过程(更正活动)

### 6.17.1 问题/变更报告

它包括要记录的具体条目(可选的条目包括：项目名称，提出者，问题编号，问题名称，受影响的软件元素或文档，发生日期，类别和优先级，描述，指派的该问题的分析者，指派日期，完成日期，分析时间，推荐的解决方案，影响，问题状态，解决方案的批准，随后的动作，更正者，更正日期，被更正的版本. 更正时间，已实现的解决方案的描述)。

## **6.17.2 更正活动系统**

## **6.18 联合评审(联合技术评审和联合管理评审)**

### **6.18.1 联合技术评审包括----组建议的评审**

### **6.18.2 联合管理评审包括----组建议的评审**

## **6.19 文档编制**

## **6.20 其他软件开发活动**

### **6.20.1 风险管理，包括已知的风险和相应的对策**

#### **(1) 合同风险**

签订的合同不科学、不严谨，项目边界和各方面责任界定不清等是影响项目成败的重大因素之一。

预防这种风险的办法是项目建设之初项目经理就需要全面准确地了解合同各条款的内容、尽早和合同各方就模糊或不明确的条款签订补充协议。

#### **(2) 需求变更风险**

需求变更是软件项目经常发生的事情。一个看似很有“钱途”的软件项目，往往由于无限度的需求变更而让项目承建方苦不堪言，甚至最终亏损(实际上项目建设方也面临巨大的风险)。

预防这种风险的办法是项目建设之初就和用户书面约定好需求变更控制流程、记录并归档用户的需求变更申请。

#### **(3) 沟通不良风险**

项目组与项目各干系方沟通不良是影响项目顺利进展的一个非常重要的因素。

预防这种风险的办法是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。

#### (4) 进度风险

有些项目对进度要求非常苛刻(进度要求不高的项目，我们同样要考虑该风险)，项目进度的延迟意味着违约或市场机会的错失。

预防这种风险的办法一般是分阶段交付产品、增加项目监控的频度和力度、多运用可行的办法保证工作质量避免返工。

#### (5) 质量风险

有些项目，用户对软件质量有很高的要求，如果项目组成员同类型项目的开发经验不足，则需要密切关注项目的质量风险。

预防这种风险的办法一般是经常和用户交流工作成果、采用符合要求的开发流程、认真组织对产出物的检查和评审、计划和组织严格的独立测试等。

#### (6) 系统性能风险

有些软件项目属于多用户并发的应用系统，系统对性能要求很高，这时项目组就需要关注项目的性能风险。

预防这种风险的办法一般是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。

#### (7) 工具风险

软件项目开发和实施过程，所必须用到的管理工具、开发工具、测试工具等是否能及时到位、到位的工具版本是否符合项目要求等，是项目组需要考虑的风险因素。

预防这种风险的办法一般是在项目的启动阶段就落实好各项工具的来源或可能的替代工具，在这些工具需要使用之前(一般需要提前一个月左右)跟踪并落实工具的到位事宜。

#### (8) 技术风险

在软件项目开发和建设的过程中，技术因素是一个非常重要的因素。项目组一定要本着项目的实际要求，选用合适、成熟的技术，千万不要无视项目的实际情况而选用一些虽然先进但并非项目所必须且自己又不熟悉的技术。如果项目所要求的技术项目成员不具备或掌握不够，则需要重点关注该风险因素。

预防这种风险的办法是选用项目所必须的技术、在技术应用之前，针对

相关人员开展好技术培训工作。

#### (9) 团队成员能力和素质风险

团队成员的能力(包括业务能力和技术能力)和素质,对项目的进展、项目的质量具有很大的影响,项目经理在项目的建设过程需要实时关注该因素。

预防这种风险的办法是在用人之前先选对人、开展有针对性的培训、将合适的人安排到合适的岗位上。

#### (10) 团队成员协作风险

团队成员是否能齐心协力为项目的共同目标服务,是影响进度和质量的关键因素。

预防这种风险的办法是项目在建设之初项目经理就需要将项目目标、工作任务等和项目成员沟通清楚,采用公平、公正、公开的绩效考评制度,倡导团结互助的工作风尚等。

#### (11) 人员流动风险

项目成员特别是核心成员的流动给项目造成的影响是非常可怕的。人员的流动轻则影响项目进度,重则导致项目无法继续甚至被迫夭折。

预防这种风险的办法是尽可能将项目的核心工作分派给多人(而不要集中在个别人身上)、加强同类型人才的培养和储备。

#### (12) 工作环境风险

工作环境(包括办公环境和人文环境)的好坏直接影响项目成员的工作情绪和工作效率。

预防这种风险的办法是在项目建设之前就选择和建设好适合项目特点和满足项目成员期望的办公环境、在项目的建设过程中不断培育和调整出和谐的人文环境。

#### (13) 系统运行环境风险

目前,大部分项目系统集成和软件开发是分开进行的(甚至由不同公司承接)。因此,软件系统赖以运行的硬件环境和网络环境的建设进度对软件系统是否能顺利实施具有相当大的影响。

预防这种风险的办法是和用户签定相关的协议、跟进系统集成部分的实施进度、及时提醒用户等。

软件项目都存在着这样那样的风险，尤其是数据分析项目这就需要在进行软件开发项目时更加注重风险管理，注重风险分析，做好风险管理计划，积极寻求风险应对方法，从而提高项目成功的机会。

6.20.2 软件管理指标，包括要使用的指标

序号	技术要求
1	管理界面：B/S 架构，中英文 WEB 界面： 所有系统可以在任何地方随时通过浏览器在同一界面远程管理；
2	监测方式：集中非代理式监测，无需在被监控对象上安装任何代理软件，对原有系统不产生任何影响；
3	检测顺序：可将网络设备、服务器、负载均衡设备等分组管理，提供方便如资源管理器一样的树型视图；能对被监控对象的添加、删除；
4	网络设备管理：能支持对不同厂商、不同型号的网络设备的接口流量、接口状态、接口丢包率、Cpu 利用率、Memory、配置文件是否修改等进行集中监测管理，能展现网络设备的真实面板图和逻辑面板图。
5	跨平台监测：能跨平台支持 Windows、Linux 等操作系统，对 Cpu、Memory、Disk、Service、Process、日志及文件等进行实时监测管理
6	拓扑管理：拓扑自动搜索功能，同时支持 VISIO 视图导入，网络管理员可以直接用已知设备的 IP 地址，根据同一网络可以用 VISIO 绘制出应用结构拓扑图、机架位置摆放图、逻辑结构图、物理结构图等等。
7	服务器状态：实时监测网管中的服务器的性能与负载，界面上能清晰的反映所监视的服务器的各项指标、状态。指标包括 CPU 利用率，内存利用率，磁盘利用率。
8	目录文件管理：可以监视文件大小、修改时间、生存时间；目录的大小、生存时间、包含文件的个数。
9	进程管理：对服务器的进程信息进行监视。在表格中显示了要监视服务器的所有的进程信息，如进程的序列、名称、状态、类型、CPU 利用率、内存利用率等等。

10	数据库管理:能对 ORACLE、SQLSERVER、DB2、SYBASE 等数据库进行有效的监控和管理, 可对这些服务的可用性、响应时间的趋势变化进行实时、集中的监控。
11	中间件管理: 能对 WEBLOGIC 和 WEBSPHERE、MQ 等中间件进行有效的监控和管理, 可对这些服务的可用性、响应时间的趋势变化进行实时、集中的监控。
12	OA 系统管理: 能对 OA 中的 Domino 服务、WEB 服务、目录服务、DNS、FTP、EMAIL、HTTP 等基础网络服务进行集中监视, 可对这些服务的可用性、响应时间的趋势变化进行实时、集中的监控。
13	事件统计报表功能: 能通过对历史统计数据进行分析, 生成趋势报告, TOPN 报告等, 发现并消除网络瓶颈及安全隐患, 为提高网络资源利用率, 制定网络战略规划提供依据
14	报警方式: 提供声音、远程声音、Email、手机短信息、脚本等多种报警方式
15	故障管理: 具有监测故障、故障确认、故障连锁诊断和及时报警等功能, 以及监控错误日志, 形成故障统计的功能
16	故障自动处理: 能自动对常见故障通过脚本或 Post 数据等方式进行重启服务、重启机器、恢复配置文件等处理
17	权限管理: 可以详细设置不同身份的网管人员登录网管系统所看到和操作权限不同实现权限的平滑过渡
18	系统登录: 从网络安全的角度考虑, 对相关设备采用加密方式登录。
19	二次开发支持: 开放式 API, 能进行功能扩展、二次开发或实现与其它网管系统的无缝集成
20	协议支持: 监测支持 SNMP、RMON、ICMP、SYS LOG、Telnet、SSH、DNS、LDAP 和 TCP/IP 等协议; 能广泛支持不同厂家设备;

6.20.3 保密性和私密性

6.20.4 分承包方管理

6.20.5 与软件独立验证与确认(IV&V)机构的接口

6.20.6 和有关开发方的协调

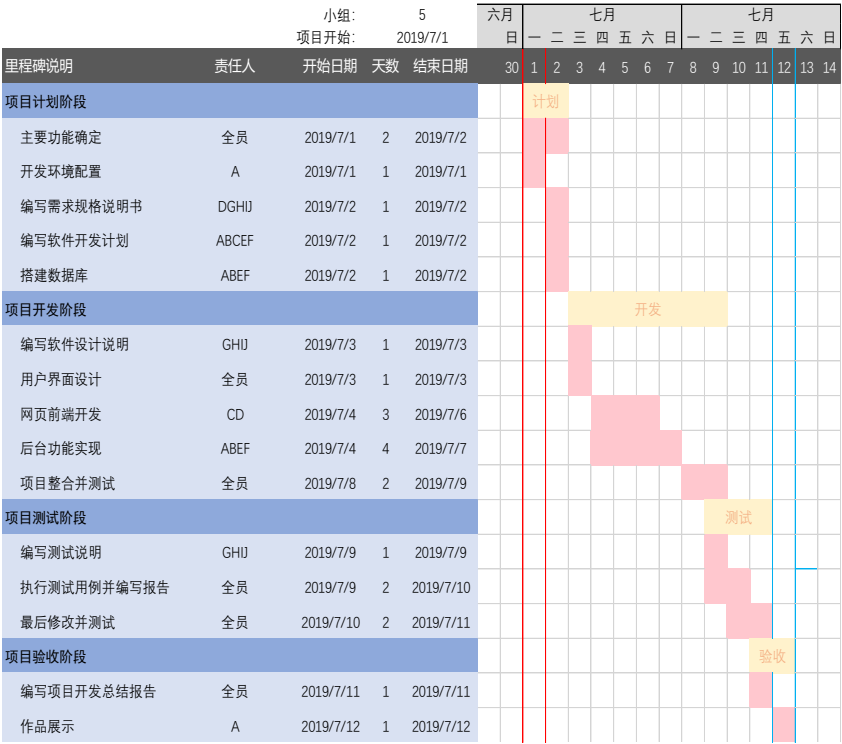
6.20.7 项目过程的改进

6.20.8 计划中未提及的其他活动

7 进度表和活动网络图

进度表：

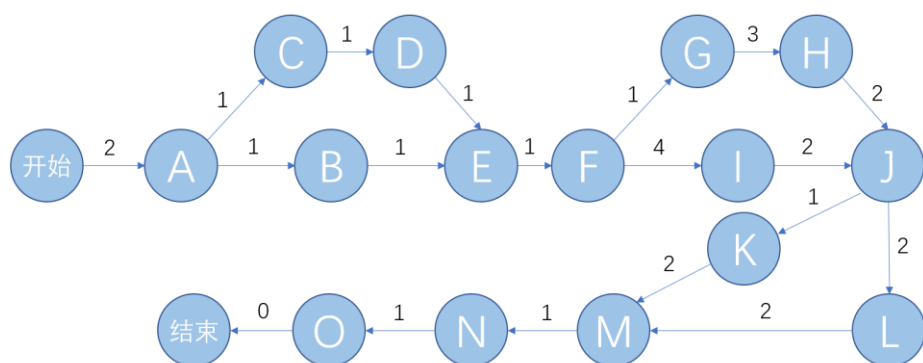
项目名称：NPU BUS



注：组员代号表示：A刘龙涛（组长）B李云飞 C李宗明 D刘嘉林 E王琪 F刘峰 G史明明 H苏畅 I王泽望 J康浩然

网络图：





节点符号说明：

A：主要功能确定  
D：编写软件开发计划  
G：用户界面设计  
J：项目整合并测试  
M：最后修改并测试

B：开发环境配置  
E：搭建数据库  
H：网页前端开发  
K：编写测试说明  
N：编写项目开发总结报告

C：编写需求规格说明书  
F：编写软件设计说明  
I：后台功能实现  
L：执行测试用例并编写报告  
O：作品展示

## 8 项目组织和资源

### 8.1 项目组织

本项目团队成员总共 10 人，组织结构采用简单式组织形式，即一个组长和九个组员。组长即是项目经理（团队经理），团队负责权利最高。

### 8.2 项目资源

#### A 人力资源

1) 该项目投入人员为 10 人，投入时间为 12 天（2019 年 7 月 1 日至 2019 年 7 月 12 日）。

2) 软件计划阶段全员参与；开发阶段将团队分成 4 人和 6 人两个小组，分别实现网页的前端开发和后台的功能实现；测试阶段将团队分成 5 个两人小组，分别对项目设计不同的测试用例，最后将发现的错误分给两个 5 人团队解决；项目验收阶段的项目开发总结报告，由组长做负责人，组织成员根据其开发阶段的工作情况编写。最后由组长负责作品展示。

B 开发人员使用的设施均为个人笔记本电脑，工作地点位于毅字楼实验室和旺园学生公寓，无保密区域。

C 需方需要判定项目团队给定的十个功能需求是否满足条件。

## 9 培训

### 9.1 项目的技术要求

本项目要求团队人员掌握 html、css、javascript、jsp、数据库和服务器等知识。

### 9.2 培训计划

根据项目经理分配的任务，成员需要自学相关技术。

## 10 项目估算

### 10.1 规模估算

本系统根据以下划分标准估算，本项目属于中至大型项目。

微型：只是一个人，甚至是半日工作在几天内完成的软件；

小型：一个人半年内完成的 2000 行以内的程序；

中型：5 个人在 1 年多的时间内完成的 5000-50000 行的程序；

大型：5-10 人在两年内完成的 50000-100000 行的程序；

甚大型：100-1000 人参加用 4-5 年完成的具有 100, 0000 行的软件项目；

极大行：2000-5000 人参加，10 年内完成的 1000 万行以内的程序；

### 10.2 工作量估算

为估算本软件工作量，先介绍一个衡量软件项目规模最常用的概念——LOC(Line of Code)，LOC 所有的可执行的源代码行数，包括可交付的工作控制语言(JCL: Job Control Language)语句、数据定义、数据类型声明、等价声明、输入/输出格式声明等。一代码行(1LOC)的价值和人月均代码行数可以体现一个

软件生产组织的生产能力。组织可以根据对历史项目的审计来核算组织的单行代码价值。有 10,000 行代码，假定 30%需要重新设计，50%需要重新编码，70%需要重新测试，那么其等价的代码行可以计算为：

$$[(30\% + 50\% + 70\%)/3] \times 10,000 = 5,000 \text{ 等价代码行。}$$

意即：重用这 10000 代码相当于编写 5000 代码行的工作量。

## 10.3 成本估算

预计本项目组每一万行语 java 言源代码形成的源文件（.c 和.h 文件）约为 250K。某项目的源文件大小为 4M；

该项目的规模是多少 KLOC（源代码行数）为  $4M/250K=16 \text{ 万行}=160KLOC$ ；

设本团队生产率是 0.625KLOC/人月，人工价是 10000 元/人月，则工作量=规模/生产率= $160KLOC/0.625KLOC=256 \text{ 人月}$ ；

成本=工作量×人工价= $256 \text{ 人月} \times 10000 \text{ 元/人月} = \text{¥}256 \text{ 万}$ 。

## 10.4 关键计算机资源估算

本团队计算机均使用本人笔记本开发，无需考虑计算机资源问题。

## 10.5 管理预留

暂无。

# 11 风险管理

本章应分析可能存在的风险，所采取的对策和风险管理计划。本项目主要存在以下风险：

一、市场风险：市价波动对于校车营运或投资可能产生亏损之风险，如利

率、汇率、股价等变动对学校相关部门损益的影响；

二、信用风险：包括学生存在恶意预约、取消情况，对学生信用度进行记录；调度员的分配出现问题，与实际情况有偏差。

三、作业风险：校车分配制度不良对学校造成之风险，如多发车辆、未完全坐满现象，这与各涉众的良好配合、稳定操作有分不开的关系。

四、法律风险：契约之完备与有效与否对学校可能产生之风险。

五、资讯风险：校车系统之安控、运作、备援失当导致学校承受的风险，如校车系统障碍、当机、资料消灭，安全防护或负责人员电脑病毒预防与处理等。

## 12 支持条件

### 12.1 计算机系统支持。

需要能够运行 Windows 7/8/10 操作系统的计算机，安装并配置软件 Microsoft Office Access 数据库管理系统。

### 12.2 需要需方承担的工作和提供的条件。

1. 内部统一需求。
2. 由固定的一个或几个人来进行沟通。
3. 确定需求实现目的，实现方法，运行环境和开发时间。
4. 根据开发周期，跟进实际开发情况。
5. 定期沟通，时间节点上观看实际进度和 Beta 版演示。
6. 对于相关人员，提前准备培训计划，配合乙方制作用户手册。
7. 试运行前，组织人员，由乙方负责培训。

## 12.3 需要分包商承担的工作和提供的条件。

1. 根据甲方的具体需求进行功能设计方案，并及时与甲方沟通确保设计功能符合实际操作和管理需要；
2. 负责软件的代码编写，确保软件可靠、运行数据准确完成开发工作
3. 乙方收到需求变更表后需要项目组负责人进行需求评估，并及时与甲方沟通。
4. 对甲方提出的修改意见及时修改，修改完毕后并及时通知甲方测试和试运行。
5. 协助甲方进行质量检查，并提供所需的材料和信息。
6. 负责培训甲方人员，提供操作说明文档。
7. 负责合同约定的维护期内的维护。
8. 收到新需求后需要项目组负责人进行需求评估和成本评估，并及时与甲方沟通。

## 13 注解

本文档依据《国标 GB/T 8567-2006 计算机软件文档编制规范》制定，属于技术文档，仅限于实验室相关人员阅读。

本文档给出了本项目中文档的编号规则，所有项目文档都需按照本文档中制订的规则进行编号，从而实现统一规范有效的文档管理。

1. 文件均采用 A4 纸幅面。文件的名称应简明准确，一般不超过 20 个汉字。
2. 文件的内容应表达准确、清楚、简明、严谨。
3. 同一文件中术语、符号、代号应统一。表达同一术语的概念应前后一致。采用的术语尚无标准规定时且容易产生不同理解的，应给出定义或说明。
4. 文件中的缩略词(语)应采用有关标准或专业委员会认定的缩略词(语)，自定缩略词(语)应简明，能反映主题。缩略词(语)在文件中首次出现时应做说明。
5. 文件中引用的标准和文件应是现行有效。
6. 文件中应采用国务院正式公布、实施的简化汉字。

# 附录

1. 《NPU-BUS-DNR-1.0 文档编号规则》
2. 《NPU-BUS-SCS-1.0 软件编码规范》
3. 《NPU-BUS-SDS-1.0 软件文档规范》