

# PM 2.5 项目管理书

学生：毛晓雯

学号：1252982

老师：杜庆峰

2015-06-28

# 目录

1. 项目总述.....	1
1.1 项目概要.....	1
1. 外部接口描述.....	2
1.1 用户接口.....	2
1.2 硬件接口.....	2
1.3 软件接口.....	2
1.4 通讯接口.....	2
1.4.1 记录用户每日吸入量.....	2
1.4.2 播报 PM2.5 空气质量指数.....	3
1.4.3 管理城市列表.....	4
1.4.4 开放个体吸入量数据.....	4
1.5 其他非功能性需求.....	5
1.5.1 安全性需求.....	5
1.5.2 软件质量属性.....	5
1.2. 项目成果.....	5
1.3. 项目假设.....	6
2. 项目计划.....	6
2.1 项目过程.....	6
2.1.1. 过程模型.....	6
2.1.2. 裁剪要点.....	7
2.1.3 需求变更管理过程.....	8
2.1.4 需求变更生命周期.....	8
2.1.5. 需求变更可跟踪性.....	8
2.2 估算.....	9
2.2.1. 系统抽象.....	9
2.2.2. 功能点估算.....	9
2.2.3. 用例估算.....	12
2.2.4. 综合估算结果.....	14
2.3 时刻表.....	15
2.4 人员.....	15
2.5 开发环境.....	15
2.6 硬件与软件资源需求.....	15
2.7 工具.....	16
2.8 训练.....	16
2.9 质量计划.....	16
2.9.1 质量目标.....	16
2.9.2 检测缺陷评估.....	17
2.9.3 达到质量目标的策略.....	17
2.9.4 审核.....	17
2.10 风险管理.....	18
3. 项目跟踪.....	19

3.1	测量计划.....	19
3.2	任务跟踪.....	19
3.3	问题追踪.....	19
3.4	用户反馈.....	20
3.5	质量追踪.....	20
3.6	需要高级管理员审核的内容.....	20
3.7	状态报告.....	20
3.8	里程统计偏移.....	21
3.9	客户报告.....	21
3.10	业务经理报告.....	21
4.	项目团队.....	22
4.1	项目组织.....	22
4.2	项目团队.....	22
4.3	角色与责任.....	22
5.	参考.....	23
6.	缩写说明.....	23

# 1. 项目总述

## 1.1 项目概要

本项目产出 PM2.5 个体水平暴露的数据收集和平台系统，该系统旨在借助普及的智能手机上来估算累计用户的 PM2.5 吸入量，并将用户的吸入量数据记录入平台数据库，为将来的个体暴露水平与环境流行病的关系科学研究做数据集准备，以此来建立一个基于科学采集方法的 PM2.5 个体水平暴露记录系统和数据开放平台。

该系统中包括 iOS 端移动应用，安装在用户 iPhone 实行实时监测和记录计算的手机应用程序；数据获取和收集的服务端后台，一方面获得开放平台提供的国家监测局公布的实时空气质量数据，为手机应用客户端提供实时的 PM2.5 空气质量数据。另一方面，收集用户吸入量数据并存储入数据库，建立个人吸入量记录数据集和数据开放平台。

项目代码	项目名称			客户
20150313	PM2.5 个人监测应用			纽约西奈山医学院
项目经理	配置经理	业务经理	项目副经理	配置副经理
Xiaowen Mao	Xiaowen Mao	Xiaowen Mao	Xiaowen Mao	Xiaowen Mao
项目类型	平台			项目阶段数
新开发	iOS 8 以上，Mac OS X, Ubuntu 14.04(Apache+MySQL+PHP)			5
目开始时间		项目截止时间		总预算预计
2015 年 3 月 13 日		2015 年 9 月		20 万人民币
项目与客户联系人				
姓名	手机号		传真	邮件
Xiaowen Mao	152-2142-0965			mxw_699@163.com
项目边界				

## 1. 外部接口描述

### 1.1 用户接口

这一部分用户接口主要就是 iOS 手机端用户界面部分。用户可查看任意城市的空气质量指数，添加、删除关注城市。

### 1.2 硬件接口

PM2.5 云测仪 SDM805 是一款综合性的 24 小时空气质量监测仪，基本款搭载高精度激光 PM2.5 传感器 SDS011，还可以扩展监测温度、湿度、CO<sub>2</sub>、TVOC 等空气质量参数。SDM805 内置 Wi-Fi 模块，可以把数据上传到云服务器，可以跟净化器、新风控制系统进行通讯控制，从而实现净化设备的手动或智能化控制；SDM805 支持二次开发，开放串口通讯协议、Wi-Fi 通讯协议和云服务器的数据接口 API，方便将数据嵌入到系统中，为智能家居、物联网、云数据和大数据分析提供了强有力的监测终端。

本项目中通过 Wi-Fi 通讯协议和诺方电子云服务器的数据接口 API 连接到硬件设备来获取数据。

### 1.3 软件接口

本项目使用到的外部软件接口是连接 <http://pm25.in> 数据平台所提供的数据 API，使用 HTTP 传输协议。具体接口说明文档见平台网站。

### 1.4 通讯接口

在 iOS 客户端和服务端之间，使用 HTTP 传输协议，考虑到用户隐私数据的安全问题，必须防止中途窃听数据盗取用户隐私数据，改良为使用 HTTPS，在 HTTP 层之下建设 Secure Sockets Layer 来建立安全的传输连接。iOS 端擅长解析 JSON 格式数据，但考虑到别的平台客户端开发，后台提供多种数据格式的接口，包括 XML 和 JSON 等。

在本小节中按照系统功能来组织和描述了各个功能性需求，功能性需求的按照 REQ-N-M 的方式标签，N 区分客户端、服务端，M 作连续编号。

#### 1.4.1 记录用户每日吸入量

##### a. 功能描述和优先级

手机应用通过判断用户所处区域，与服务器通讯获得 PM2.5 空气质量指数作为运算基数，再判断用户所处环境类型，如室内室外、车内，再进一步结合用户的运动情况，如跑步、静止、走路、骑车，将这些不同的环境类型和运动状态对应到各自的系数因子，作为比例来调整环境 PM2.5 浓度，将浓度换算到单位时间吸入量，最后在时间上做一个积分。

一天的吸入量累积过程结束后，将最终的吸入量存入本地 Plist 文件，同时传回到服务器，服务器加密后保存在数据库。

这一功能是本系统的核心功能，优先级为高。

#### b. 执行序列

用户下载安装手机应用后第一次打开应用，应用会询问用户是否允许查看用户的位置，用户选择是否授予应用获取用户位置的权限，应用进一步询问用户是否允许应用在后台保持运行和持续读取用户位置，用户选择是否授予权限。

接下来在用户只有查看自己 PM2.5 吸入量数据的行为，不再在应用内对应用是否获取用户位置进行设置，但是 iOS 系统支持在系统设置中统一设置各个应用读取用户位置等隐私信息的功能。

第一次打开完成这些操作之后，应用从此保持读取用户位置、状态，计算并累计用户的 PM2.5 吸入量，并在每天定点时间将一天的吸入总量传回到服务器加密保存。

#### c. 功能性需求

REQ-1-1: 询问并在获得允许之后保持对用户位置、状态的读取

REQ-1-2: 获取 PM2.5 实时数据

REQ-1-3: 获取城市列表和区域监测点坐标

REQ-1-4: 累计计算用户 PM2.5 吸入量

REQ-1-5: 保存每日吸入量到手机本地

REQ-1-6: 通过安全连接传输吸入量数据到服务器

REQ-1-13: 连接监测设备

REQ-2-1: 获取 PM2.5 实时数据并保存的数据库

REQ-2-2: 提供 PM2.5 数据 RESTful 接口

REQ-2-3: 更新城市列表和监测点坐标

REQ-2-4: 提供城市列表接口

REQ-2-6: 加密存储用户吸入量数据

### 1.4.2 播报 PM2.5 空气质量指数

#### a. 功能描述和优先级

用户可以查看当地空气质量，所添加城市（区域）的空气质量，过去 24 小时的空气质量，过去 1 周的每日平均空气质量指数。作为一款完整的产品，向用户播报空气质量指数也是核心功能之一，此功能优先级为高。

#### b. 执行序列

用户可以查看到城市（区域）列表，见下方所示的用户接口，点击某一城市，应用会跳转到城市详情界面，见下方所示的用户接口，在这个界面中，用户可以通过上下滚动查看更多历史信息，也包括吸入量数据和更具体的天气情况如主要颗粒物。点击城市详情界面右下角的列表按钮，应用将返回到城市列表。

#### c. 功能性需求

REQ-1-7: 查看城市列表

REQ-1-8: 查看城市空气质量详细情况，包括历史数据和吸入量、空气污染详细情况

REQ-1-9: 直接在不同城市详情页之间切换

REQ-2-2: 提供 PM2.5 数据 RESTful 接口

### 1.4.3 管理城市列表

#### a. 功能描述和优先级

用户可以添加城市、删除城市和调整城市排列位置。

#### b. 执行序列

在城市列表页面，点击加号按钮，应用会进入搜索城市（区域）界面，在搜索栏中输入城市名称，或者区域名称，点击键盘上的搜索按钮，应用会更新搜索栏下方的城市列表，点击某一栏，应用则会添加该栏所对应的城市区域，并返回城市列表界面，城市列表此时就会被更新，新添加的城市会出现在最低部。

REQ-1-10: 添加城市（区域）

REQ-1-11: 删除城市（区域）

REQ-1-12: 调整城市的排列位置

REQ-2-4: 提供城市列表接口

### 1.4.4 开放个体吸入量数据

#### a. 功能描述和优先级

数据库保存了个体吸入量，系统将这部分数据对第三方开放。

#### b. 执行序列

这部分的用户实际是开发者，开发者调用数据的 HTTP 接口，系统返回对应数据。

#### c. 功能性需求

<p>REQ-2-5: 提供个人吸入量的 API</p> <p>1.5 其他非功能性需求</p> <p>1.5.1 安全性需求</p> <p>在项目难点和技术要求小节中强调过用户个人吸入量属于隐私数据，系统应保证在传输过程中使用安全连接，加密后存储入数据库。</p> <p>1.5.2 软件质量属性</p> <p>项目应保证应用在美国地区的适应性，即完成国际化和本地化。在应用动画质量方面，保持和原生应用基础动画质量一致，保持用户界面的易用性。</p>
项目增值服务
开发配套宣传与文档网站
公司目标
建立首个记录个体吸入量的 APP，搭建未来环境流行病与污染物吸入量关系研究的数据平台。

## 1.2. 项目成果

序号	里程碑日期	里程碑	交付物
1	2015-03-13	立项	
2	2015-03-25	可行性研究完成	可行性报告
3	2015-04-07	需求分析完成	需求规约
4	2015-04-22	计划阶段完成	项目企划书（采购计划、开发计划、测试计划、人力计划）
5	2015-05-11	概要设计完成	概要设计书（初步客户端设计、初步服务端设计、初步数据库设计、初步测试计划设计）
6	2015-05-29	详细设计完成	详细设计书（客户端设计、服务端设计、数据库设计、测试计划）
7	2015-06-03	设计阶段完成	
8	2015-08-07	Construction 1: 代码实现与测试实现	客户端 demo（达到展示要求）
9	2015-08-22	Construction 2: 代码审核与重构	通过测试样例,达到覆盖率要求的项目
10	2015-08-25	Transition: 代码提交与合并	审核结果与通过审核的项目



11	2015-08-26	Deployment 1: 部署项目	部署项目进行用户接受测试
12	2015-09-03	Deployment 2: 收集用户反馈	用户反馈报告
13	2015-09-07	Construction 3: 项目重构	修复已知 bug 的项目
14	2015-09-08	Deployment 3: 项目上架更新	项目上架销售
<b>其他成果</b>			
<b>序号</b>	<b>成果</b>		
1	向第三方开发者开放的数据平台		
2	展示项目的网页		

### 1.3. 项目假设

项目采用的数据源是由 BestApp 工作室提供的 <http://pm25.in> 数据平台，因此在项目进行过程以及产品投放后都假设该数据源保持长期稳定的数据提供。

## 2. 项目计划

### 2.1 项目过程

#### 2.1.1. 过程模型

本项目采用了结合标准模型、原型模型与增量模型的开发模型。

PM2.5 个人监测应用开发项目结合使用了原型模型和增量模型。由于项目是要从零开始开发一个新的手机 APP，并且市面上没有类似功能的 APP 在架销售，因此应用所能够达到的效果和市场的真正需求都是未知。在这种对需求了解不成熟的情形下，具有迭代开发特征的增量模型则值得被尝试。又因为，在技术方面存在难度和风险，核心功能可能由于缺乏技术可行性而无法实现，所以，在第一轮迭代，也就是核心产品看法过程中，用到了原型开发的思想，先将核心功能实现，并在第一次交付后验证实现效果，然后再进一步决定下一步迭代计划。

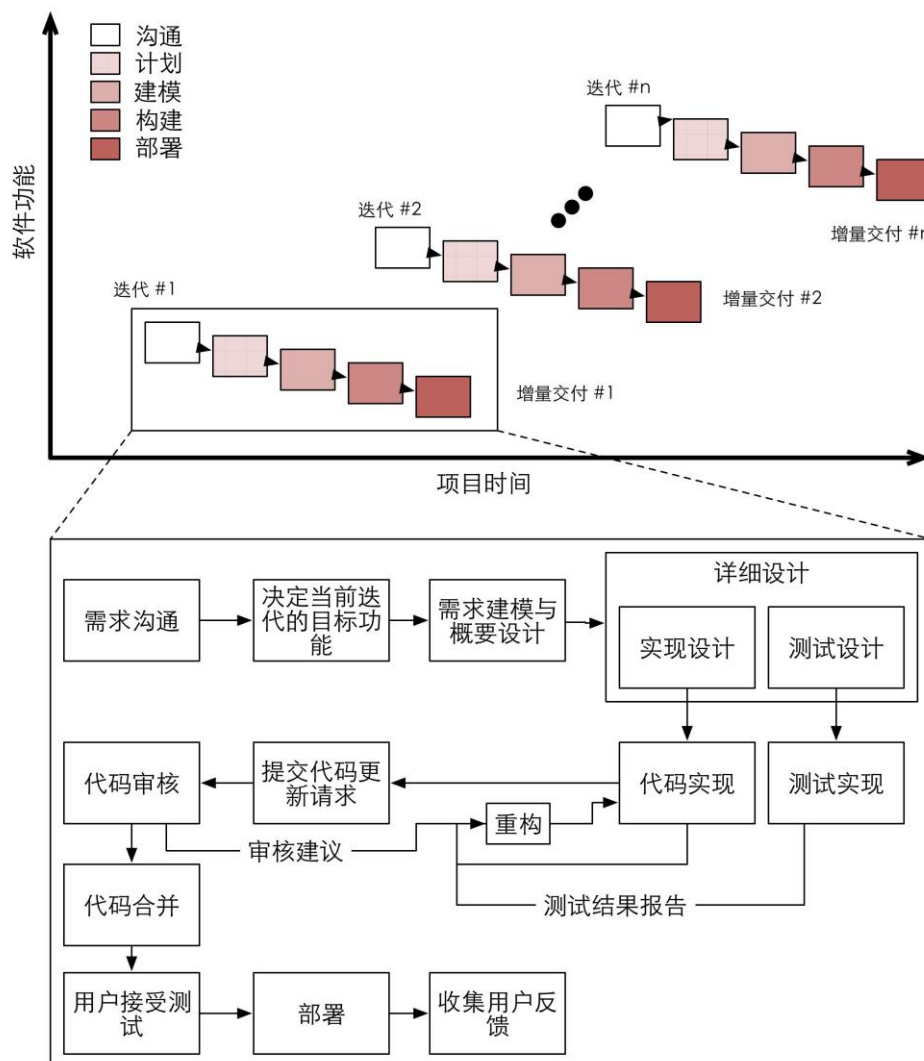


图 2-1 过程模型图

### 2.1.2. 裁剪要点

与标准模型的差异	增加/修改/删除	理由
相对于迭代模型，将第一次增量开发过程置换为原型开发。	修改	因为项目在技术方面存在难点和风险，在第一轮增量开发中，应该考虑到技术可行性，因此第一次迭代主要是针对技术的尝试。
相对于标准模型，更加注重开发过程和代码质量。	增加、修改	开发人员的平台开发经验不足，项目管理者对项目经验丰富，因此通过强化开发过程的质量审核过程来提高项目质量。
增加了代码审核与合并	增加	理由同上

细化了详细设计	修改	理由同上
细化了测试步骤	修改	理由同上
增加了收集用户反馈	增加	理由同上

### 2.1.3 需求变更管理过程

需求变更跟踪
1. 客户要求的需求变更必须被登记记录在 <b>ChangeRequests.xls</b> 中。登记条目有：变更编号、变更简述、变更影响、变更状态和关键日期。其中变更状态参照需求变更生命周期。
2. 被登记的需求变更都需要进行影响分析。主要考虑原则有：变更所涉及的工作量、对工作产品交付承诺的影响、对成本和交付质量的影响、对整体日程和工作量的潜在引申影响等。
3. 估算修改变量所需的工作量
4. 重新估算交付日程表
5. 计算累积变更并分析影响
6. 检查是否超过阈值。若工作量、日程、累积变更等超过可容阈值，需上报高级经理
7. 获得客户批准
8. 进入返工

### 2.1.4 需求变更生命周期

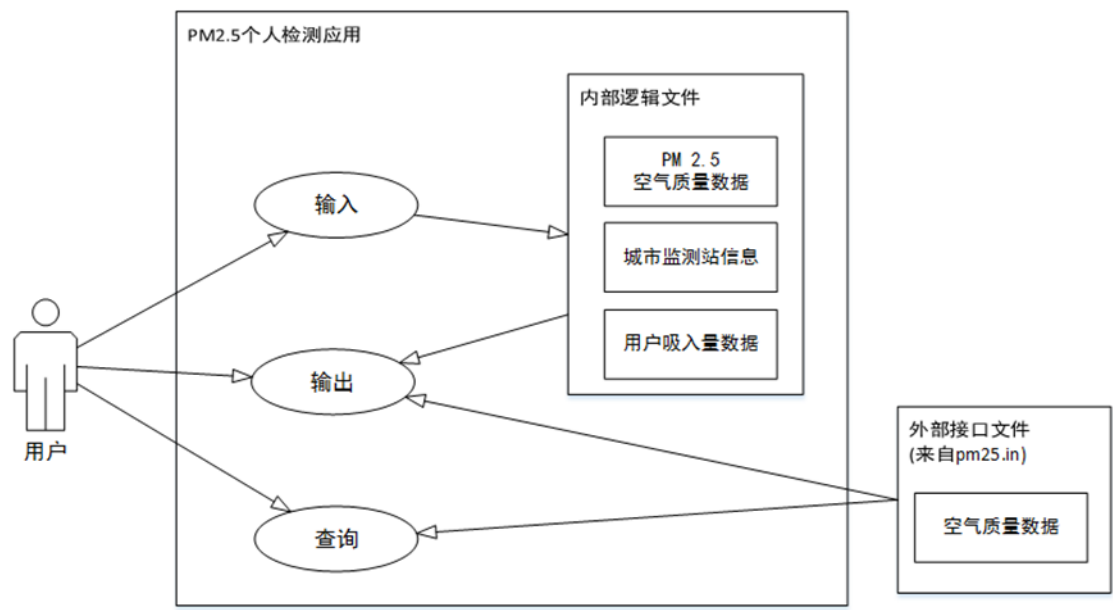
状态	说明
已登记	该需求变更已登记在 <b>ChangeRequests.xls</b>
已批准	需求变更分析结果得到高级经理和客户的批准
返工中	得到批准后，按照项目经理制定的计划返工
验证中	返工完成后，需要验证各类规约是否修改一致
测试中	验证完成后，需要额外对代码进行测试
完结	测试通过
挂起	测试不通过，需进一步检查
<b>Note:</b> 状态之间可以流水跳转，也可以随机跳转。	

### 2.1.5. 需求变更可跟踪性

利用需求管理工具 RequisitePro 进行跟踪。

2.2 估算

2.2.1. 系统抽象



2.2.2. 功能点估算

相关数据如下：

PM2.5 空气质量数据	城市监测站信息	用户吸入量数据
1. 空气质量数据 id 2. PM2.5 数值 3. 城市名称 4. 城市监测站名称 5. 空气质量评级 6. AQI 数值 7. 时间点	1. 城市监测站 id 2. 城市名称 3. 城市监测站名称 4. 经度 5. 纬度 6. 别名	1. 用户吸入量 id 2. 用户设备号 3. 用户吸入量 4. 时间点
空气质量数据（来自 pm25.in）		百度地图信息
1. aqi 2. area 3. pm2_5 4. pm2_5_24h 5. position_name 6. primary_pollutant 7. quality 8. station_code 9. time_point		1. 城市名称 2. 城市监测站名称 3. 经度 4. 纬度

识别出的 ILF 和 EIF 功能点个数如下表所示。

ILF 内部逻辑文件	RET	DET	复杂度	未调整 FP 个数
PM2.5 空气质量数据	1 个	7 个	低	7
城市监测站信息	1 个	6 个	低	7
用户吸入量数据	1 个	4 个	低	7
EIF 外部接口文件	RET	DET	复杂度	未调整 FP 个数
空气质量数据	1 个	9 个	低	7
百度地图信息	1 个	4 个	低	7
共计：35 个				

由此，数据功能点  $DFP = ILF + EIF = 35$ 。

识别出来的 EI、EQ 和 EO 个数如下表所示。

EI	FTR	DET 个数	复杂度	未调整 FP 个数
添加 PM2.5 空气质量数据	PM2.5 空气质量数据	空气质量数据 id 为自动生成，共 6 个	低	3
添加城市监测站信息	城市监测站信息	城市监测站 id 为自动生成，共 5 个	低	3
修改城市监测站信息	城市监测站信息	共 6 个	低	3
删除城市监测站信息	城市监测站信息	城市监测站 id，共 1 个	低	3
添加用户吸入量数据	用户吸入量数据	用户吸入量数据 id 为自动生成，共 3 个	低	3
修改用户吸入量数据	用户吸入量数据	共 4 个	低	3
共计：18 个				
EQ	FTR	DET 个数	复杂度	未调整 FP 个数
查询 PM2.5 空气质量数据	PM2.5 空气质量数据	城市名称和城市监测站名称，共 2 个	低	3
查询城市监测站信息	城市监测站信息	城市名称和城市监测站名称，共 2 个	低	3
查询用户吸入量数据	用户吸入量数据	用户设备号，共 1 个	低	3
共计：9 个				
EO	FTR	DET 个数	复杂度	未调整 FP 个数
查询 PM2.5 空气质	PM2.5 空气质量	7 个	低	4

量数据	数据			
查询城市监测站信息	城市监测站信息	6 个	低	4
查询用户吸入量数据	用户吸入量数据	4 个	低	4
<b>共计：16 个</b>				

由此，事务功能点  $TFP = EI + EQ + EO = 18 + 9 + 16 = 43$ 。

所以，未调整功能点  $UFP = TFP + DFP = 35 + 43 = 78$ 。

进一步考虑，本系统的通用系统特性及其影响程度如下表所示。

系统特性	分数
数据通讯	4
分布式数据处理	0
性能	0
大业务量配置	0
事务处理率	3
在线数据输入	1
最终用户效率	2
在线更新	1
复杂处理	2
可复用性	3
易安装性	1
易操作性	2
多场地	2
支持变更	3
<b>合计：24</b>	
<b>调整因子=24*0.01+0.65=0.89</b>	

由此，调整后功能点  $AFP = UFP * 0.89 = 69.42$

根据日本 IPA SEC 组织 (<http://sec.ipa.go.jp/>) 所提供的工作量估算公式：

$Effort = e^{0.542} * FP^{1.154} = 229.33$  人-天。

假设日标准工作时间为 8 小时，工作量估计为 1832 人-小时。

2.2.3. 用例估算

用例分类依据如下表。

用例类型	描述	因子
简单	0-3 个事件	5
中等	4-7 个事件	10
复杂	> 7 个事件	15

考虑本项目中的用例。

用例编号	描述	复杂度
1	查看历史吸入量数据	简单
2	查看今日、本周、本月吸入量	简单
3	查看最近监测站空气质量	中等
4	查看城市空气质量	简单
5	添加关注的城市	中等
6	删除关注的城市	简单
7	排列城市顺序	简单
8	查看监测设备数据	复杂
9	定位最近监测站	简单
10	获取空气质量数据	简单
11	保存用户吸入量到本地	简单
12	传输用户吸入量数据	简单
13	获取城市数据	简单
14	申请开发 token	简单
15	发放开发 token	简单
16	增加开发会员	中等
17	获取用户吸入量数据	简单

估算工作量			
用例	工作量（基于以往项目）	个数	总工作量（人-天）
简单用例	1 人-天	13	13
中等用例	5 人-天	3	15
复杂用例	8 人-天	1	8
共计			36

计算未调整用例点  $UUCP = 13 * 5 + 3 * 10 + 1 * 15 = 110$

考虑技术复杂度因子 TCP 如下表，得分 5 表示对项目很重要，0 表示非项目要求。

序号	技术复杂度因子	权重	得分
1	Distributed system	2	2
2	Response or throughput performance objectives	1	4
3	End-user efficiency (online)	1	5
4	Complex internal processing	1	4
5	Code must be reusable	1	3
6	Easy to install	0.5	5
7	Easy to use	0.5	5
8	Portable	2	5
9	Easy to change	1	5
10	Concurrent	1	3
11	Includes special security features	1	4
12	Provides direct access for third parties	1	4
13	Special user training facilities required	1	3

根据权重计算：

$$TFactor = 2*2+1*4+1*5+1*4+1*3+0.5*5+0.5*5+2*5+1*5+1*3+1*4+1*4+1*3 = 54$$

$$TCP = 0.6 + (0.01 * TFactor) = 1.4$$

进一步考虑环境调整因子 EF 如下表。

序号	环境调整因子	权重	得分
1	Familiar with Internet process	1.5	5
2	Application experience	0.5	5
3	Object-oriented experience	1	5
4	Lead analyst capability	0.5	4
5	Motivation	1	4
6	Stable requirements	2	3
7	Part-time workers	-1	0
8	Difficult programming language	-1	4

$$EFactor = 1.5*5+0.5*5+1*5+0.5*4+1*4+2*3-1*0-1*4 = 23$$

$$EF = 1.4 + (-0.03 * EFactor) = 0.71$$

最后，根据技术复杂度和环境因素：



计算  $UCP = UUCP * TCF * EF = 110 * 1.4 * 0.71 = 109.34$

根据标准工作量 1 个 UCP 对应 20 人-小时,得到工作量估算为  $109 * 20 = 2180$  人-小时。

假设日工作时间为 8 小时,即需要 272.5 人-天。

#### 2.2.4. 综合估算结果

综合 2.2.3 和 2.2.4 两种估算方法得到的工作量估计,求其平均得到工作量为 2006 人-小时。

假设日工作时间为 8 小时,即需要 250.75 人-天。

Phase-wise Effort Estimate		
Activity/Phase	人-天	占总工作量比例
Requirements	25	10
Design	30	12
Build	73	29
Integration testing	18	7
Regression testing	5	2
Acceptance testing	15	6
Project management	38	15
Configuration management	8	3
Training	25	10
Others	15	6
Estimated effort	251	100%

Effort Estimate by Iterations	Person-days	% of Total Effort
Project initiation	13	5
Inception phase	13	5
Elaboration phase: Iteration 1	23	9
Elaboration phase: Iteration 2	18	7
Construction phase: Iteration 1	13	5
Construction phase: Iteration 2	13	5
Construction phase: Iteration 3	10	4
Transition phase	55	22
Project closure	5	2
Project management	38	15
Configuration management	8	3
Training	25	10
Others	20	8
Total estimated effort	251	100%

### 2.3 时刻表

详见附件：PM 2.5 Project.mpp

### 2.4 人员

按角色			
角色		要求人数	日期
项目经理		1	2015-03-13
配置管理		1	2015-03-13
开发者		3	2015-04-23
数据库管理员		1	2015-06-04
测试经理		1	2015-03-13
测试人员		2	2015-06-04
业务经理		1	2015-03-13
配置经理		1	2015-03-13
总计		11	
按技能和经验			
领域	总数	经验（1-12 月）人数	经验（>12 月）人数
iOS	2	1	1
PHP	1	1	0
MySQL	3	1	2
总数	6	3	3
人员需求计划			
月份	Offshore	Onsite	总数
2015-03 - 2015-09	0	4	4

### 2.5 开发环境

硬件：Macbook，高配置服务器，iPhone 6 Plus

软件：Mac OS X 10.10 Yosemite，iOS 8.2，Ubuntu 14.04，Apache，MySQL

中间件：Yii PHP 开发框架

IDE：Xcode 6.2

### 2.6 硬件与软件资源需求

描述	所需数量	日期
Workstation with 16GB RAM	1	2015-4-15
iPhone with iOS 8	1	2015-3-13

Macbook with Mac OS X	1	2015-3-13
SDL307	1	2015-5-1
SDM805	1	2015-5-15

## 2.7 工具

UML 建模工具：StarUML

配置管理工具：Git, GitHub

需求管理工具：Requisite Pro

缺陷跟踪工具：Bugzilla

项目管理工具：Microsoft Project 2013

集成测试服务工具：Xcode Server, GitHub

测试结果分析工具：XcodeCoverage

测试工具：XCTest 测试套件

## 2.8 训练

训练领域	时长	免去条件
<b>技术领域</b>		
iOS 开发	10 天	有上架作品
Objective-C 开发	10 天	有 2 万行以上的编程经验或 GitHub 简历精彩
PHP 开发	3 天	有 PHP 开发经验
Yii 框架	5 天	有 Yii 开发经验
MySQL	2 小时	有 MySQL 使用经验
<b>业务领域</b>		
<b>过程相关</b>		
质量管理体系	3 小时	有项目管理工作经验
配置管理	2 小时	有项目配置管理经验
代码审核	3 小时	有代码审核经验

## 2.9 质量计划

### 2.9.1 质量目标

项目质量目标		
目标	值	目标设置基础
产生的缺陷总数	180	0.04 缺陷每人时。比类似的项目 TiCheck 要好 5%， TiCheck 的缺陷率是 0.042
质量（接受缺陷密度）	3	预估检测到的缺陷数的 2%或更少
生产力	70	5%优于 TiCheck 项目
时刻表	按时交付	
质量成本	35%	TiCheck 质量成本为 33%

### 2.9.2 检测缺陷评估

审核/测试阶段	被检测到的缺陷预估	百分比	预估基础
需求和设计审核	34	20%	类似项目 TiCheck
单元测试	68	40%	类似项目 TiCheck
代码审核	34	20%	类似项目 TiCheck
连续集成测试	31	18%	类似项目 TiCheck
用户接受测试	3	2%	类似项目 TiCheck
总数	170	100%	

### 2.9.3 达到质量目标的策略

策略	期望
按照缺陷预防纲领和过程来做缺陷预防；按照 TiCheck 缺陷预防标准来编码。	降低 15%缺陷产生，提高 2%生产力
结合 GitHub 做代码提交申请和代码审核，直至审核通过才合并代码。	有规范的代码提交记录，保证每一次代码合并都产生完整的可执行的项目。
按照敏捷开发的思路在每一个开发迭代结束后进行总结和审核。	高效执行每一次迭代，有效计划下一次迭代

### 2.9.4 审核

审核点	审核对象	审核类型
项目计划结束	项目计划 项目时刻表	团队审核 SQA 审核

原型需求讨论结束	需求规约 用例条目 界面框架	团队审核
概要设计结束	概要规约（模块说明、部署方案）	团队审核
详细设计结束	详细设计规约（类图、序列图、测试套件）	团队审核
功能实现结束	代码	单人审核
单元测试结束	测试结果和代码	单人审核
代码审核结束	审核结果和代码	单人审核
用户反馈报告生成结束	用户反馈报告	团队审核

## 2.10 风险管理

本项目的设计与完成完全由我独立完成，在能够完成全栈开发和软件工程时间的同时，项目中所设计的部分技术和平台建立的经验欠缺是本项目的主要难点：

### （1）用户产生数据的安全传输和加密存储：

用户开始使用手机应用客户端之后，应用会实时记录累计用户的 PM2.5 吸入量，根据项目需求和目的，这类数据需要传输到服务器后端并保存，以供作后期病理与环境关系研究的数据集。这类数据属于用户的隐私数据，首先应当避免传输过程中的窃听和数据库直接被明文读取，其次在用户未授权相关科研工作展开使用其本人个人吸入量数据之前，系统应维持用户隐私数据处在完全保密状态。数据的安全传输涉及到服务器开通 SSL 模块即对 HTTPS 访问的支持，另外，在存储进入数据库之前应当对数据进行对称加密，同时，对私钥妥善存储。

### （2）用户使用手机时的精确定位：

在监测个体暴露水平一节中提到了用户处在不同城市区域，以及室内室外甚至车内这些不同情况时的当前暴露水平变化很大，因此判断当前环境的空气 PM2.5 含量则成为了记录用户吸入量监测中的至关重要的环节。国家监测局公布的 PM2.5 含量是由每个城市的各个监测站收集得到的，因此我们能够获得的实时 PM2.5 含量数据只能精确到区域范围。因此，本项目对不同环境下的空气 PM2.5 含量的变化作出了一定的假设，主要以区域数据作为基数，对不同环境做比例换算。从而，判断用户所处的具体环境区域和环境类型成为了计算过程中的重点和难点。

### （3）数据集建立和数据平台建设：

因为本项目所产出的数据收集系统主要为后期科研工作做数据收集平台建设和数据集准备，在完成数据收集功能需求的同时还需要合理化数据平台的 API 开放使用，建设完备的数据开放机制，保证仅在获得科研对象用户授权的前提下开放用户隐私数据以供科研分析使用。合理完整的数据集准备和数据平台的开放形式设计则成为了项目的难点之一。

序列号	风险	概率	影响因子	风险值	预防计划
1	数据被窃听	0.2	6	1.2	建立 SSL 安全传输协议； 先将数据加密再传输
2	数据库被违法读取	0.4	10	4	关键数据加密存储； 建立完整防火墙机制
3	手机定位失败	0.5	8	4	建立复杂的定位算法，多 方面综合定位，如结合 Wi-Fi 定位
4	第三方开发者访问频率过高	0.8	6	4.8	做访问频率限制
5	数据源数据中断	0.2	10	5	建立其他数据源，降低单 方数据依赖

### 3. 项目跟踪

#### 3.1 测量计划

需要收集的数据	数据单位	所用工具
规模	LOC, FP	XcodeCoverage
人力	人天	MSP
缺陷	个数	BugsBunny
时刻表	时长	MSP

#### 3.2 任务跟踪

活动	过程
任务时间计划	用 MSP 做时间计划； 在必要时调整时间计划以适应最新情况。
任务布置	MSP 软件导出任务提醒文件，团队成员可以自行下载。
任务状态追踪	每天执行一次任务状态更新
项目会议	3 天一次站立会议，一周一次组会
详细分析会议	每次迭代结束进行

#### 3.3 问题追踪

问题类型	日志位置	日志创建可许人	审核人安排，时间	处理时长
线上问题	GitHub issue	任何人	由 PM 指定修改，每日	2 天
用户反馈	UserFeedback.xls	业务人员	由 PM 指定修改，每日	7 天
业务管理问题	周报	业务经理	由 PM 指定修改，每日	5 天
服务支持问题	Request Tracker	任何人	业务经理	2 天

### 3.4 用户反馈

事项	日志和追踪过程
用户反馈	业务经理每周统计用户反馈信息
客户抱怨	业务经理及时答复客户，处理事件

### 3.5 质量追踪

质量活动	行动
缺陷追踪	用 GitHub Issue 记录并跟踪缺陷
审核（需求审核，概要设计审核，详细设计审核，代码审核）	依据质量计划进行检验
代码审核	在提交申请之后安排其他开发者对代码进行审核，并提出改进意见报告
独立单元测试	生成测试报告并交由开发者做代码重构
连续集成测试	测试不通过自动发布报告

### 3.6 需要高级管理员审核的内容

序列号	审核事项	审核频率
1	时刻表	版本更新时审核
2	项目计划	重大计划变动时审核
3	里程碑报告	每个里程碑时间点

### 3.7 状态报告

报告对象	频率
业务经理	每周一邮件报告
项目经理	每日例会报告

### 3.8 里程统计偏移

实际与预估比较	前五个里程	其余里程
人力投入	10%	5%
时刻表	10%	5%
缺陷数	20%	20%

### 3.9 客户报告

项目客户是软件使用者，通过 App Store 的客户开发者联系方式，开发者并不对客户作报告，这部分内容省去。

### 3.10 业务经理报告

客户反馈

里程碑和周报告

问题报告

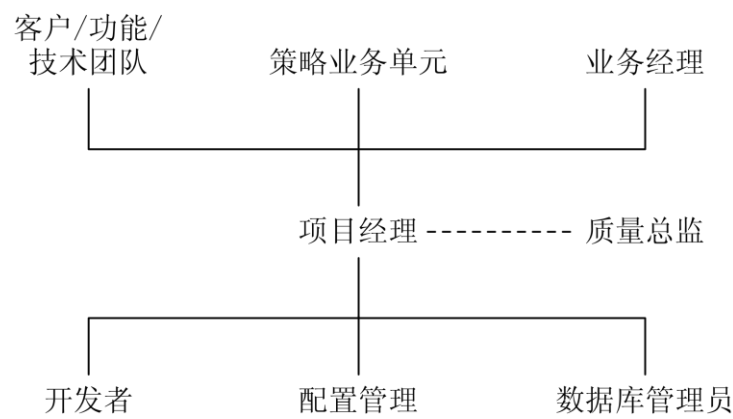
需求改动

计划重大改变



# 4. 项目团队

## 4.1 项目组织



## 4.2 项目团队

序号	职责	开始日期	预计结束日期
1	项目经理	2015-03-13	2015-09-08
2	质量总监	2015-03-13	2015-09-08
3	开发者	2015-04-23	2015-08-25
4	数据库管理员	2015-06-04	2015-08-25
5	测试经理	2015-03-13	2015-09-08
6	测试人员	2015-06-04	2015-07-13
7	业务经理	2015-03-13	2015-09-08
8	配置经理	2015-03-13	2015-09-08

## 4.3 角色与责任

序号	角色	责任
1	项目经理	负责整个项目进度把握； 保证整个项目按计划进行； 审核项目各种重要交付物。
2	质量总监	审核质量计划； 检查质量维护按计划进行。
3	开发者	代码实现和重构； 概要与详细设计。
4	数据库管理员	维护数据库与后台
5	测试经理	整体测试计划设计和监督。

6	测试人员	测试实现。
7	业务经理	用户交流； 用户反馈报告整理； 推广业务执行。
8	配置经理	各软硬件设施准备； 设施维护； 人员配管与管理； 采购与预算计划。

## 5. 参考

Software Project Management in Practice, By Pankaj Jalote

## 6. 缩写说明

缩写	说明
PM2.5	细颗粒物
MSP	Microsoft Project 2013
PM	Project Manager 项目经理
AQI	Air Quality Index 空气质量指数
id	ID 号，标签
FP	Function Point Approach 功能点估算
UCP	Use Case Point Approach 用例点估算

---

<sup>[1]</sup> BestApp 工作室. Pm25.in API[OL]. (2014-01-03) [2015-03-13] . [http://pm25.in/api\\_doc](http://pm25.in/api_doc)