**Dex2jar的测试及优化**

**综合实验总结**

Version 2.0

小组成员：

卢兴海

王文茹

蒋 波

**版本变更历史**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本 | 变更时间 | 修改人 | 审核人 | 备注 |
| 1.0 | 2017/06/23 | 蒋波、卢兴海、王文茹 | 蒋波、卢兴海、王文茹 | 初稿 |
| 2.0 | 2017/06/23 | 蒋波、卢兴海、王文茹 | 蒋波、卢兴海、王文茹 | 二稿 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

目录

[一、 概述 5](#_Toc485997937)

[二、 实验工作量 6](#_Toc485997938)

[2.1需求分析 6](#_Toc485997939)

[2.2需求评审 6](#_Toc485997940)

[2.3产品改进与展示 7](#_Toc485997941)

[2.4软件测试 7](#_Toc485997942)

[2.5软件测试评审 8](#_Toc485997943)

[2.6进度计划与控制 8](#_Toc485997944)

[2.7配置管理 9](#_Toc485997945)

[2.8工作量估计与统计分析 9](#_Toc485997946)

[三、 数据分析与说明 10](#_Toc485997947)

[3.1项目数据分析 10](#_Toc485997948)

[3.2项目产生的数据分析 11](#_Toc485997949)

[四、 质量水平 11](#_Toc485997950)

[4.1需求分析 11](#_Toc485997951)

[4.2软件测试 12](#_Toc485997952)

[4.3产品改进与扩展及其测试 12](#_Toc485997953)

[4.4进度计划与控制 12](#_Toc485997954)

[4.4.1实验过程及特点 12](#_Toc485997955)

[4.4.2计划变更及影响因素 13](#_Toc485997956)

[4.4.3工作分配分析与说明 14](#_Toc485997957)

[五、 有效方法 16](#_Toc485997958)

[5.1软件需求分析 16](#_Toc485997959)

[5.1.1如何进行需求分析 16](#_Toc485997960)

[5.1.2如何描述需求用例 16](#_Toc485997961)

[5.2软件需求评审 17](#_Toc485997962)

[5.2.1如何提出有效的评审问题 17](#_Toc485997963)

[5.2.2如何处理收到的问题 17](#_Toc485997964)

[5.3改进与展示 17](#_Toc485997965)

[5.3.1如何确定要改进的内容 17](#_Toc485997966)

[5.4测试需求分析与评审 18](#_Toc485997967)

[5.4.1需要注意的问题 18](#_Toc485997968)

[5.5进度计划与控制 18](#_Toc485997969)

[5.5.1一个有效的工作模式 18](#_Toc485997970)

[5.6配置管理 18](#_Toc485997971)

[5.6.1提交人与编写人一致性的问题 18](#_Toc485997972)

[5.7工作量估计与统计分析 19](#_Toc485997973)

[5.7.1如何进行统计 19](#_Toc485997974)

[六、 结论与建议 20](#_Toc485997975)

# 概述

本次软件工程综合实验遵循了软件工程的开发过程，过程中包含了需求分析、软件测试等方面，还补充了评审相关部分，具体过程包括：软件项目计划阶段、软件需求分析阶段、软件需求评审阶段、软件产品改进与展示阶段、软件测试阶段、软件测试评审阶段。除了以上软件开发过程应用的步骤，还对软件开发的过程进行了管理与分析，具体包括：软件进度计划与控制、配置管理、工作量估计与统计分析，这些管理工作贯穿整个项目始终。

在实验过程中，既完成了软件工程开发所要求的开发过程，又在开发过程中通过软件进度计划与控制、工作量估计与统计分析、配置管理这些工作对整个项目开发进行管理和控制，在保证项目按照计划执行的同时，还保留了项目执行和开发过程中的各项证据，进一步形成了证据链，使得项目的开发与执行更具有说服力。

本次实验基本实现了预期的工作目标，通过小组合作的方式实现了dex2jar的完整开发过程，对开发过程进行了管理，对产生的数据进行了统计分析。

# 实验工作量

## 2.1需求分析

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 实验1：需求分析 | | | | | |
| 需求文档规模（字数） | 需求项数（RUCM用例描述数量） | 其他模型（类图、顺序图、状态图等，不含非规范的示意图） | 其他模型中包含的元素累积数（节点和边的数量） | 版本更新次数 | 累积工时（含需求修改工时） |
| 4231 | 14 | 类图1 |  | 6 | 56.4 |
| 顺序图1 |  |
| 流程图 |  |
| 状态图 |  |
| … |  |

表2.1需求分析工作量

## 2.2需求评审

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 实验2：需求评审（评审+复评审） | | | | | | |
| 检查单中的检查项数量 | 互评审中给被评审组提出的问题数（评审+复评审） | 接收到的问题数（评审+复评审） | 老师的问题数（含各组存在的共性问题） | 接受并修改的问题数 | 评审报告字数 | 累积工时（不含需求修改工时） |
|  | 51 | 64 | 17 | 81 |  | 35.35 |
|
|
|
|

表2.2需求评审工作量

## 2.3产品改进与展示

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 实验3：改进与扩展 | | | | | | | |
| 设计实现规模（字数） | 其他模型（类图、顺序图、状态图等，不含非规范的示意图） | 其他模型中包含的元素累积数（节点和边的数量） | 代码行数 | 问题数 | 修改的问题数 | 版本更新（最大）次数 | 累积工时数 |
|  | 类图 |  | 168不含测试 | 3 | 2 | 2 | 56.81 |
| 顺序图 |  |
| 流程图 |  |
| 状态图 |  |
| … |  |

表2.3产品改进与展示工作量

## 2.4软件测试

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 实验4：测试需求 | | | | | | |
| 测试需求和测试报告等文档规模（字数） | 测试需求用例数 | 测试数据（实例）数 | 测试类型 | 覆盖率 | 测试需求文档更新次数 | 累积工时（含需求修改工时） |
| 4714 | 14 | 100 | 业务需求 | 100% | 3 | 50.97 |
| 功能需求 | 100% |
| 非功能需求 | 100% |
|  |  |
|  |  |

表2.4软件测试工作量

## 2.5软件测试评审

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 实验5：软件测试评审 | | | | | | |
| 检查单中的检查项数量 | 互评审中给被评审组提出的问题数（评审+复评审） | 接收到的问题数（评审+复评审） | 老师的问题数（含各组存在的共性问题） | 接受并修改的问题数 | 评审报告字数 | 累积工时（不含测试需求修改工时） |
| 13 | 9 | 18 | 4 | 22 | 440 | 9.35 |
|
|
|
|

表2.5软件测试评审工作量

## 2.6进度计划与控制

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 实验6：项目计划 | | | | |
| 项目计划书、小组会议记录和周记等累积字数 | 项目计划表（MPP）中分解的任务项数（含组合任务） | 项目计划耗费的实际工时数 | 项目计划（MPP）更新次数 | 项目的累积实际工时数 |
| 8787 | 116 | 332 | 13 | 391.88 |
|
|
|
|

表2.6进度计划与控制工作量

## 2.7配置管理

|  |  |  |
| --- | --- | --- |
| 实验7：配置管理 | | |
| 版本更新此数（提交次数） | 配置管理报告等累积字数 | 累积工时 |
| 3 | 2986 | 16.97 |
|
|
|
|

表2.7配置管理工作量

## 2.8工作量估计与统计分析

|  |  |  |
| --- | --- | --- |
| 实验8：统计分析 | | |
| 统计分析项数 | 统计分析报告字数 | 累积工时 |
| 5 | 4617 | 40.57 |
|
|
|
|

表2.8工作量估计与统计分析工作量

# 数据分析与说明

## 3.1项目数据分析

首先，实验所用apk文件来自华为应用市场各个时段的top100，为了获取这些apk，我们用自动化测试工具selenium实现了自动下载，这些apk全部可以在真实的安卓手机上安装运行，这部分所获得的数据完全准确。

其次，我们进行的是分析dex2jar工具本身的可靠性，主要是在类及其方法这个层面。为此，我们用Java语言编写了测试工具，获取类和方法的途径就是按照指定文件的格式进行读取。影响因素主要是安卓平台版本的不一致（工具本身的兼容性），而经过读取资料可以发现，我们要分析的文件（Dex、class）的格式均固定，所以只要按照官方给出的格式进行读取就不会出错。经过测试后可发现，dex2jar的工具是可靠的。至少在我们所测试的apk文件里面，还没有发现偏差。

然后，我们开始进行工具的优化。分为两部分，增加解析资源文件的功能，增加图形界面。后一个功能较容易实现，并且通过最后的运行可以看出，这个图形界面可以完成指定的功能。至于解析资源文件，我们的思路是利用已有的可以解析资源文件的工具—apktool，将这个工具与dex2jar结合。为此，我们写了相应的bat脚本，所以有一个影响因素就是平台限制，目前只能在Windows平台上实现此项优化。我们测试了这项优化，发现大部分应用均可正常解析出资源文件，只有一两个apk文件出现异常，问题的原因在于apktool这个工具与有的apk文件不兼容。我们的保证措施就是使用最新版的apktool，偏差估计是3%，（依据公式：成功次数/测试总次数）。

## 3.2项目产生的数据分析

实验6所涉及的文档为项目计划与控制分析报告，数据包括文档的版本号以及文档中提到的工时数据。关于版本号，我们小组严格以最终提交到github的文档为依据，确定版本号，在此方面没有任何误差。至于工时，由于我们采用的是组员自己在微信群里报工时的方式，肯定会有误差，控制方法就是根据分配给每个组员任务的难度、任务多少，在主观得出的工时的基础上，最终确定一个较为合理的工时数。

实验7所涉及的文档为配置管理，数据包括版本号，每个人的提交内容、提交概要、提交时间，这部分内容又卢兴海同学负责维护。至于版本号，如上所述，依然是没有误差。因为小组在项目初期或者每个实验的初期有很多疑问，所以有时会导致重复提交，初次以外，有些与项目关系并不大的提交内容或者修改在最终的报告中就没有呈现。所以，github上面显示的提交次数与配置管理中显示的提交次数有偏差。根据两者次数差异，可以得出偏差度在10%以内。其他方面，如提交内容，提交时间均如实填写，没有任何偏差。

实验8所涉及的文档为工作量统计与分析报告。里面涉及的数据主要与代码量、文档书写数量有关，这两部分都有客观文件作为依据，统计方法也很简单，就是计算代码行数和文档行数。可以说，这部分没有误差，是准确的。

# 质量水平

## 4.1需求分析

软件需求分析阶段的主要制品为软件需求说明书。本项目需求说明书共有6个版本，最终版本为《需求规格说明书v2.1》，包含中文字符数为4231，需求用例为13项。在需求规格说明书的编写阶段，共收到别组评审64项问题，老师评审17项问题，均已在编制过程中对其进行了修改。收到的问题数较多，原因为最开始编写的比较粗糙，在之后的修订中有较大的改动，各个版本间均有不小的变化，文档质量的提升较为明显，最终版本的需求规格说明书质量应为合格。

## 4.2软件测试

软件测试阶段的主要制品为测试需求说明书。编写阶段有3个版本，最终版本为《测试需求说明书v1.2》，包含中文字符数为4714，测试用例数为14。共收到同学评审建议18个，老师评审建议4个，在最终版本中都已经进行了修改。测试需求说明书的第一个版本格式不规范，不易于对测试工作内容进行阅读与理解，第二个版本中采用规范格式，之后修改了评审问题，最终版本需求规格说明书质量应为合格。

## 4.3产品改进与扩展及其测试

产品改进与展示阶段的主要制品为实现扩展功能的程序。此阶段我们小组实现的功能有：dex2jar反编译资源文件，图形界面。测试阶段的主要内容为测试dex2jar本身的反编译正确性与扩展功能的正确性。测试阶段采用了对100个apk文件进行反编译测试，dex2jar的本身功能较为完善，均可成功反编译，扩展功能也成功实现。

## 4.4进度计划与控制

### 4.4.1实验过程及特点

本次软件工程综合实验遵循了软件工程的开发过程，过程中包含了需求分析、软件测试等方面，还补充了评审相关部分，具体过程包括：软件项目计划阶段、软件需求分析阶段、软件需求评审阶段、软件产品改进与展示阶段、软件测试阶段、软件测试评审阶段。除了以上软件开发过程应用的步骤，还对软件开发的过程进行了管理与分析，具体包括：软件进度计划与控制、配置管理、工作量估计与统计分析，这些管理工作贯穿整个项目始终。

本次实验基本实现了预期的工作目标，通过小组合作的方式实现了dex2jar的完整开发过程，对开发过程进行了管理，对产生的数据进行了统计分析。

### 4.4.2计划变更及影响因素

本项目计划变更的具体情况如下表4.1所示。

**表4.1 项目计划变更情况**

|  |  |  |  |
| --- | --- | --- | --- |
| 项目计划版本号 | 变更情况 | 变更原因 | 时间 |
| V1 | 大致计划本学期dex2jar各周工作安排 | 无 | 20170310 |
| V2 | 增加了第二周工作内容，修改规范列名 | 需细化第二周工作，根据老师规范变更列名 | 20170331 |
| V3 | 补全实验6-8工作内容，新增了需求评审部分工作内容 | 根据老师课堂上指正的缺少内容进行补充 | 20170413 |
| V4 | 填写本周实际工作进度 | 定期更新 | 20170421 |
| V5 | 变更列名，补全本学期所有工作计划，增加基线 | 根据老师要求规范列名，补全计划，增加基线 | 20170427 |
| V6 | 填写本周实际工作进度 | 定期更新 | 20170505 |
| V7 | 填写本周实际工作进度 | 定期更新 | 20170512 |
| V8 | 填写本周实际工作进度 | 定期更新 | 20170519 |
| V9 | 填写本周实际工作进度 | 定期更新 | 20170526 |
| V10 | 填写本周实际工作进度 | 定期更新 | 20170602 |
| V11 | 填写本周实际工作进度 | 定期更新 | 20170609 |

本项目的项目计划，在V5版本补全了整个计划的所有内容，之后每周一更，将实际执行情况和多出的工作加入到项目计划中。V1版本是在项目最开始时制定，仅仅列出了本学期大致工作划分，V2版本按要求规范了列名，细化了第二周工作，V3版本开始更新了实验6-8的工作内容，并且新增了需求评审部分工作，V5版本才补充了本学期所有工作计划，并制定了基线，之后每周正常更新，将实际的工作情况写入工作计划中。

目前项目计划一共11个版本，在第三周后就是在每周的周五进行更新，主要记录本周实际的工作情况，版本V5才列全了整个项目的工作计划，因为之前对项目计划与控制文档理解有误，所以修改较多，之前的问题也比较多，V5之后就每周更新实际情况到项目计划中。

### 4.4.3工作分配分析与说明

工作分配的具体情况如表2所示。

**表4.2 小组工作分配表**

|  |  |  |  |
| --- | --- | --- | --- |
| 姓名 | 性别 | 分工 | 职位 |
| 蒋波 | 男 | 需求分析阶段的文档工作、需求评审文档工作、改进与展示阶段文档和少量代码工作、测试阶段文档和测试工作、项目计划、会议记录 | 组长 |
| 卢兴海 | 男 | 需求分析阶段的文档工作、需求评审文档工作、改进与展示阶段代码工作、测试阶段文档和测试工作、配置管理 | 组员 |
| 王文茹 | 女 | 需求分析阶段的文档工作、需求评审文档工作、改进与展示阶段文档、测试阶段文档和测试工作、工作量统计与分析、项目统计分析 | 组员 |

小组内的分工，主要是以组员擅长的方向来分配，卢兴海对dex2jar比较熟悉，主要负责dex2jar技术方向的工作，包括设计扩展功能与代码的编写，蒋波和王文茹主要负责文档书写的方面，可以较为熟练的完成RUCM图绘制、分析需求等工作。除此之外的影响因素包括个人的时间和精力、工作的积极性以及态度等。

# 有效方法

## 5.1软件需求分析

### 5.1.1如何进行需求分析

1. 从开源项目的文档中提取相关内容

一般来说，项目的文档和使用手册中都会先对项目的架构、原理、设计思想进行描述，这给我们提供了很多有用的信息，对我们初步了解项目有很大帮助。

而且，一般来说，使用手册会提供很多样例，这也侧面反映了项目的功能有哪些。

1. 从项目源代码中提取相关内容

由于实验是利用已经开发完成的项目进行的，所以可以通过对源代码的类的分析，得出项目的需求。通过类图的建立和分析，可以得到比较详尽的需求分析结果。

1. 从用户的角度出发进行需求分析

如果仅从代码角度分析，很容易忽略用户角色不同和场景不同，对功能的需求不同。为了得到具体详细的需求用例，最好从用户的角度出发再次进行分析。

### 5.1.2如何描述需求用例

由于我们组的三位组员都没有软件工程的工作经验和知识背景，所以参考的多本软件工程的文献和书籍。并通过和老师同学的讨论，最后选用RUCM来进行用例图的描述。

## 5.2软件需求评审

### 5.2.1如何提出有效的评审问题

由于对其他组项目的不了解，很难提出有价值的问题。针对这一问题，我们总结出一下两个有效的方法。

1. 从软件工程需求分析的标准出发

通过查阅软件工程的书籍文献，以对需求分析的详细要求为标准，评审各组的需求分析做得是否详尽到位。

1. 从项目本身出发

通过模拟自己是该组项目的用户，看自己的需求有没有被表述清楚。同时，学习一些项目背景，更好的对项目本身进行理解。

### 5.2.2如何处理收到的问题

最开始采取的是全部接受，但是随着实验的进行，我们采取了先讨论后接受的方法。也就是，先进行组内讨论，如果对问题有疑问，再与评审组进行讨论或解释，最后在达成一致之后，再对文档进行修改。

## 5.3改进与展示

### 5.3.1如何确定要改进的内容

（1）首先，这项工作应该在实验最初就开始进行考虑，随着实验的进行以及对项目理解的加深不断进行调整

（2）从用户的角度出发，分析在使用软件的过程中，期望软件能提供的功能。

## 5.4测试需求分析与评审

### 5.4.1需要注意的问题

测试需求分析要与需求分析对应，测试需求评审时也需要对照需求分析文档一起审阅。否则，不仅无法完成较好的测试需求文档，在对其他组进行评审的时候也很难找到有效的问题。

## 5.5进度计划与控制

### 5.5.1一个有效的工作模式

每周的开始时，对本周工作进行细化的计划，并对下周工作进行粗化的计划。在本周的最后，对本周的实际工作时间进行统计和记录，更新mpp文件。

## 5.6配置管理

### 5.6.1提交人与编写人一致性的问题

由于网络等其他原因，常常发生提交人可能并不是文档的编写人，对于这一问题我们有两个办法。

1. 通过在文档中添加编写人的记录表格
2. 通过在工作日志中记录编写的文档及次数

## 5.7工作量估计与统计分析

### 5.7.1如何进行统计

1. 在实验最初就要树立统计的意识，由于最初统计比较粗，导致后期分析的时候数据不全。
2. 统计要统一标准，由于我们组是采取充分相信队友的模式，所以对于工作量的统计需要统一的数量级，比如小时或半小时的整数倍。

# 结论与建议

课程设计比较新颖，理论和实践相互结合，而且除了编程之外，还有一些评审和讨论的环节，让学生更加地了解软件工程过程中的各个阶段，并且亲身的体验，非常有意义。因为刚开始做的时候感觉很困难，不知道怎样做，所以建议老师可以在课堂上多讲一些内容，把需要课下做的内容讲清楚，这样应该会在课程最开始能更好的进入状态。