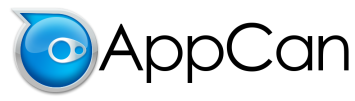
**开发流程分享流程**

**Dev / share**



**正益移动互联科技股份有限公司**

|  |  |  |
| --- | --- | --- |
| 文件状态：  [√] 草稿  [ ] 正式发布  [ ] 正在修改 | 文件标识： | 无 |
| 当前版本： | 无 |
| 作者部门： | 行业应用研发部 |
| 作 者： | 刘树友 |
| 职 位： | 架构师 |
| 完成日期： | 无 |

**修订历史**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 版本 | 编写/修订说明 | 修订人 | 修订日期 | 审核人 |
| 1 | V1.0.0 | 创建 |  |  |  |
| 2 |  |  |  |  |  |
| 3 |  |  |  |  |  |

目录

[目的 - 4 -](#_Toc436317086)

[现状 - 4 -](#_Toc436317087)

[解决方案 - 5 -](#_Toc436317088)

[统一规范 - 5 -](#_Toc436317089)

[统一把控 - 5 -](#_Toc436317090)

[统一架构 - 5 -](#_Toc436317091)

[协同 - 6 -](#_Toc436317092)

[知识库 - 6 -](#_Toc436317093)

[------------------------------------------------------------一下是资料------------------------------------- - 8 -](#_Toc436317094)

[实施细节 - 8 -](#_Toc436317095)

[测试用例 - 8 -](#_Toc436317096)

[2、部署规划 - 8 -](#_Toc436317097)

[3、运维手册 - 8 -](#_Toc436317098)

[4、项目上线（内部发行） - 9 -](#_Toc436317099)

[5、对外推广 - 9 -](#_Toc436317100)

[四、开发规范/组件分享 - 9 -](#_Toc436317101)

[4、实施保障 - 9 -](#_Toc436317102)

[5、文档需要： - 9 -](#_Toc436317103)

[6、项目推送 - 10 -](#_Toc436317104)

[7、项目粉丝 - 10 -](#_Toc436317105)

[8、服务请求 - 10 -](#_Toc436317106)

[9、使用组件 - 10 -](#_Toc436317107)

[10、粉丝评价 - 10 -](#_Toc436317108)

[11、积分 - 10 -](#_Toc436317109)

[12、积分兑换奖励kpi等 - 10 -](#_Toc436317110)

[13、平台运营客户平台/论坛 - 11 -](#_Toc436317111)

[14、风险及办法 - 11 -](#_Toc436317112)

[15、风暴 - 11 -](#_Toc436317113)

[2.2无 - 11 -](#_Toc436317114)

[2.3无 - 11 -](#_Toc436317115)

[2.4 无 - 12 -](#_Toc436317116)

[2.5 无 - 12 -](#_Toc436317117)

[2.7 - 12 -](#_Toc436317118)

# 目的

1、统一开发规范

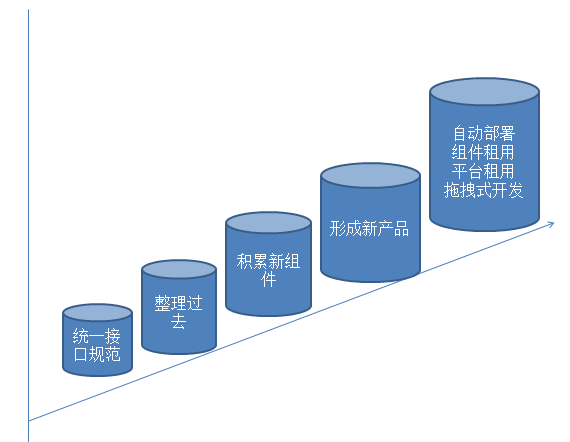
2、积累组件

3、形成产品

4、提高组件和产品使用、维护、服务的效率

5、完善项目质量把控体系

6、形成正益工场规范



# 现状

1、产品使用率不高、技术支持不及时、反复给不同人解决相同问题、升级后影响一些项目

2、架构不统一、开发结构不统一、产品插件实例不全面、没有统一媒体

2、代码没有统一管理

2、共享思想没有普及

3、沟通成本太大、不愿意使用别人的成品

4、沟通只在遇到难题层面

5、解决问题在CTO层面，和现场支持层面

6、样式与实现没有分开、共享难度偏高

7、插件化需要扩展到非原生态层次

# 解决方案

## 统一规范

文件目录结构

Appcan、Mas、java、node.js

数据库统一公共字段

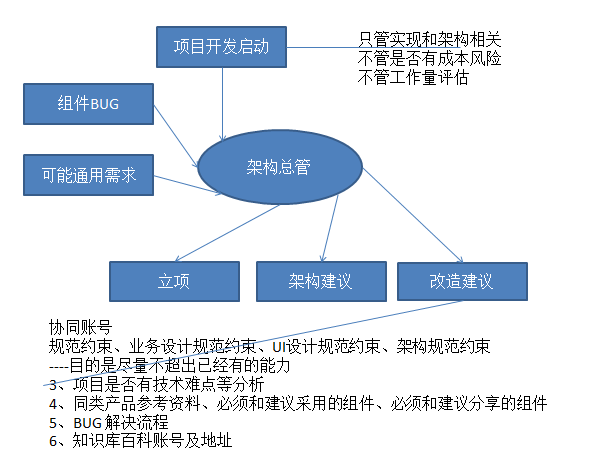
CREATE\_TIME CREATE\_ID 等

部署脚本

等。

## 统一把控

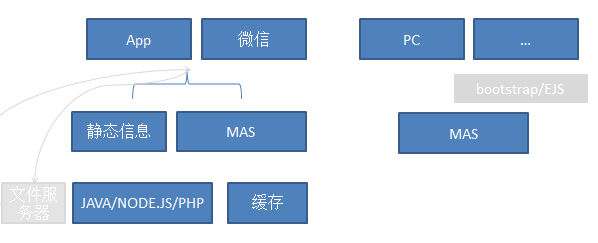
架构总管的概念



## 统一架构

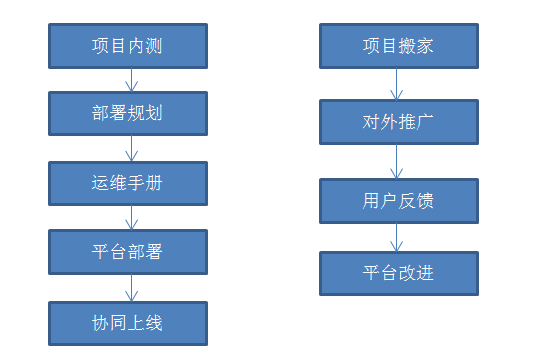
结构上的统一（）、不是使用技术上的统一、未来会逐渐技术上统一。

已经统一的appcan

mas 待测试、mum带测试 

## 协同

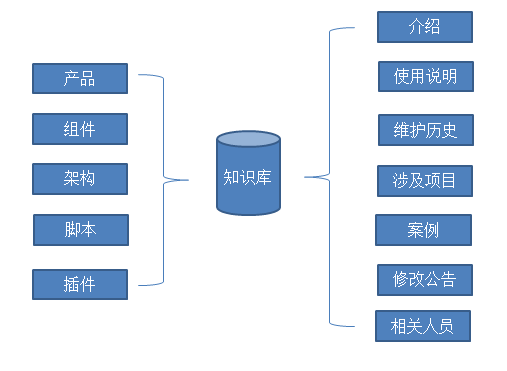
项目统一建立在系统上进行管理



## 知识库

-维科（**MediaWiki.org**）

组件和产品一切



# ------------------------------------------------------------一下是资料-------------------------------------

# 实施细节

开发中积累

立项、人员管理、代码隔离、内容备份机制等

组件项目测试，下载运行组件，租用搜索引擎、文件上传服务等

### 测试用例

团队展示、成员信息、创建项目、emm调用等

## 2、部署规划

硬件、预估用户访问量及存储配置

## 3、运维手册

部署手册、健康监控/检查手册、异常处理手册、信息备份手册、最好有个手机版健康监测app

## 4、项目上线（内部发行）

OA项目、中盟、中民项目、搜索引擎组件项目、

## 5、对外推广

Wordpress 移动实现

CMS 一键安装，应用，建站。

# 四、开发规范/组件分享

流程

创建项目

项目公告

## 4、实施保障

上线必须使用脚本打包

上线必须申请并提交修改内容、增量表、修改原因

上线内容必须在uat测试过的内容，UAT到上线专人管理和控制，只允许调整配置。

## 5、文档需要：

部署手册、

测试用例

压力测试

使用说明

## 6、项目推送

新项目/组件信息推送

## 7、项目粉丝

项目更新时、通知

## 8、服务请求

项目使用者发起

## 9、使用组件

使用了那些组件

## 10、粉丝评价

匿名、好或差评、好评价分（占比10%）， 差评减分（40%）

## 11、积分

有人用了，多个粉丝

对某人服务了，多个服务次数

使用了其他组件，加积分

## 12、积分兑换奖励kpi等

## 13、平台运营客户平台/论坛

观察：

BUG、用户反馈

运维 24小时开机/短信/微信/QQ/内信， 看平台问题紧迫度，给予解决

## 14、风险及办法

## 15、风暴

1、项目开发，需要代码质量管理，sonar/jacoco 有类似功能。

2、前端Appcan , 后端，需要一个或多个方案支撑。

3、如今 redis 等流行，　好多人用不好。

4、分布式，云方案

5、netty 等

## 2.2无

立项申请

项目账号开通

项目技术架构说明（必须使用某组件列表，必须积累某组件例如搜索引擎）

例如：中盟项目、中民项目

中盟，有搜索引擎人才，那么就积累搜索引擎

中民，积累js组件。。。

## 2.3无

搜索引擎、安全、校验、等spring结合使用的引擎，试图独立部署。

AppCan开发，试图。。。

## 2.4 无

分享， 强制办法

## 2.5 无

开发流程

开源项目炒作

开源项目对接

开源/常用引擎样例 springsite

常用应用appcan实现

开源项目孵化

## 2.7

分享

积分

赞，关注，服务过程日志，记录下被其他人借鉴。

内容

源码、服务、应用案例、升级、分版

捐助

考虑到外部人员、及特殊任务

收入

可以按某比例分配

源码下载流程

使用也分类型和级别

产品

图文手册、引用手册、服务、注意事项

协议

不允许加入暗道、隐晦等内容

统一规划、自主规划

根据项目需求，要求必须有的分享

产品需要有客户

避免形式主义

开发规范

需要评审

项目经理职责

命名规范

编码规范

Static

设计规范

数据库规范

Createt\_time 等， ID

协同应用

环境部署脚本

开发及分享流程

市场->售前->需求概要->工作量评估->里项目

工作量评估形式： 人员等级价格\* 人月 \* 0.4（成本占项目总合同额40%）

->费用分摊->结算->代码检查->测试->上线->

资源列表（git svn 。。。）

架构规范

分布式松散独立， 集群，

编码规范

静态变量统一， 需要申请

相对独立的功能具备独立部署的设计

Google最近发布了一份完整的[Java编码规范](http://google-styleguide.googlecode.com/svn/trunk/javaguide.html)。规范的内容包括一些切实可行的硬性规定。Google内部均遵守此规范。该规范不仅涵盖了代码格式，还包括其他类型的约定和编码标准。

该规范主要分为6大部分：源文件基本规范、源文件结构、代码格式、命名、编程实践和Javadoc。源文件基本规范说明文件名、文件编码、空格和特殊字符等规范要求。源文件结构说明许可信息、package和import声明、类成员顺序等规范要求。代码格式用于说明源代码中大括号、缩进、换行、空格、括号、枚举、数组、switch语句、注解、注释和修饰符等格式要求。命名说明了标识符（包、类、方法、常量、属性、局部变量和类变量）的命名规范和驼峰大小写定义。编程实践说明@Override、异常、静态成员和终结器的用法。Javadoc说明哪些地方需要Javadoc，以及它的格式要求。

以下是从规范中摘录的一些内容：

* import中不能有通配符。
* 重载方法应该放在一起连续出现。
* 即使代码块中没有内容，或者只有一行代码，也必须使用大括号。
* 2空格缩进。
* 列宽是80或100个字符。
* 不能使用类似C语言的数组变量声明。
* switch语句中，必须包含default语句。
* 修饰符的顺序应按照Java语言规范推荐的顺序。
* 常量命名应使用CONSTANT\_CASE格式（译注：所有字母大写，单词使用下划线分隔）。注意，所有常量都必须是static final成员，但并不是所有的static final成员都是常量。

目的：

* 1. 产品积累
  2. 组件积累

分享 开发 规范 ！ 分享可以有下载积分或捐助 ！ 收入可以按某比例分配！ 源码下载 有申请流程， 使用也有类型和级别！

产品，要求有 图手册，引用说明 服务

加入 不允许加暗道等 协议！

前端 图片 有静态服务器

项目 统一规划 组件 统一 问题统一

整体分为 产品 应用

画一个 整体 可升级的架构

问题要越来越少 不能总是一个问题

300 500个没事 3万五万怎么办？

规范细到 项目用啥 js 目录层次

css 可以选择风格后 简化生成

架构

服务

规范

前景

奖励

我们出类似spring 的集成标准

例如mas 给出 aop 方案 pc 方案 给出 压测数据，部署方案

使用者 在mas 上实现 登录 积累

宏观分配 人

上传积累

静态化积累

ejs 积累

使用方式 问题

组件应用时间 报价等 市场价和开发时间就足见明确了

积累过程

知识库/ 

当前项目功能沉淀

创建收集

新项目组件应用

使用手册

应用问题解决重构

BUG提交解决

架构总监

项目启动

项目流程

1. 协同账号
2. 规范约束

业务设计规范约束

UI设计规范约束

架构规范约束

----目的是尽量不超出已经有的能力

3、项目是否有技术难点等分析

4、同类产品参考资料、必须和建议采用的组件、必须和建议分享的组件

5、BUG 解决流程

6、知识库百科账号及地址

7、测试报告

项目简介

汇报架构总监

架构建议

协同办公

定期审查

项目报价公式

协同平台运营

2、应用项目及组件开发互联网化

3、企业项目及组件开发分享

4、促进组件沉淀促进项目开发速度和稳定性

5、代码成果共享，资源共享，开发速度提升，减少沟通成本，实现网上项目跟踪管理。

