

前端敏捷开发

我是谁？

许湛 (Justin)

阿里巴巴国际网站前端开发主管

Justin.xuzh@alibaba-inc.com

<http://www.alldone.cn>

提纲

1. 国际站工作流程
2. 过程中的问题分析及解决
3. 国际站前端开发成长史

案例分享

项目：异步上传

[从我的电脑选取](#) | [从图片库选取](#) | [全部删除](#)

所选图片必须是JPEG格式，图片的大小在200K以内，图片上无邮箱地址。

H_836.jpg | [取消](#) 已上传:3 / 待上传: 0 剩余 2

 100%

					
TV.jpg 删除	DVD.jpg 删除	dvd...er.jpg 删除	H_836.jpg 删除		

图片加水印以防盗用 [查看水印效果](#)

同时将上传的图片保存到图片银行

同时将上传的图片保存到图片银行

图片加水印以防盗用 [查看水印效果](#)

异步上传

- 背景：
 - 同步图片上传，影响表单提交成功率
 - 文件上传需要统一接口管理
- 项目目标：
 - 提高表单提交成功率5%
 - 降低表单提交等待时间 300%

项目流程

需求&UE

- 需求文档
- 线框图
- 需求评审

分析

- UC
- DEMO
- UC评审

设计

- 设计文档
- HTML
- 设计评审

开发

- 功能程序

测试&发布

- 上线

项目开始了

混乱的开场

BOSS：卖萝卜不赚钱

开发：红烧肉我不会做

前端：我还有个BUG没修呢。

RA：本季度没资源了

PD：做一碗红烧肉

用研：用户要吃萝卜



避免混乱 统一目标

设定项目目标

明确的目的

量化的目标



- 例如

- 目的：提高表单提交成功率

- 量化的衡量标准：

- 提高表单提交成功率 5%

- 降低表单提交等待时间 500 ms

先定目标，再讨论需求

需求 & UE

- 产物：
 - 商业需求文档
 - 线框图

- 前端做什么？
 - 评估各种方案的实现成本&风险
 - 给出方案和建议
 - 清晰地在足够范围内传达以上两点

- 问题：
 - 没有资源
 - 用户体验 VS 商业指标
 - 用户体验，谁都来说两句。
 - N多意见，N多不可实现的想法
 - 前端评估、开会花去N多资源

需求评审



分析阶段

- 产物：
 - UC
 - DEMO
- 问题：
 - 需求变更
 - UC颗粒度难以把握

UC评审



系统设计开始



需求

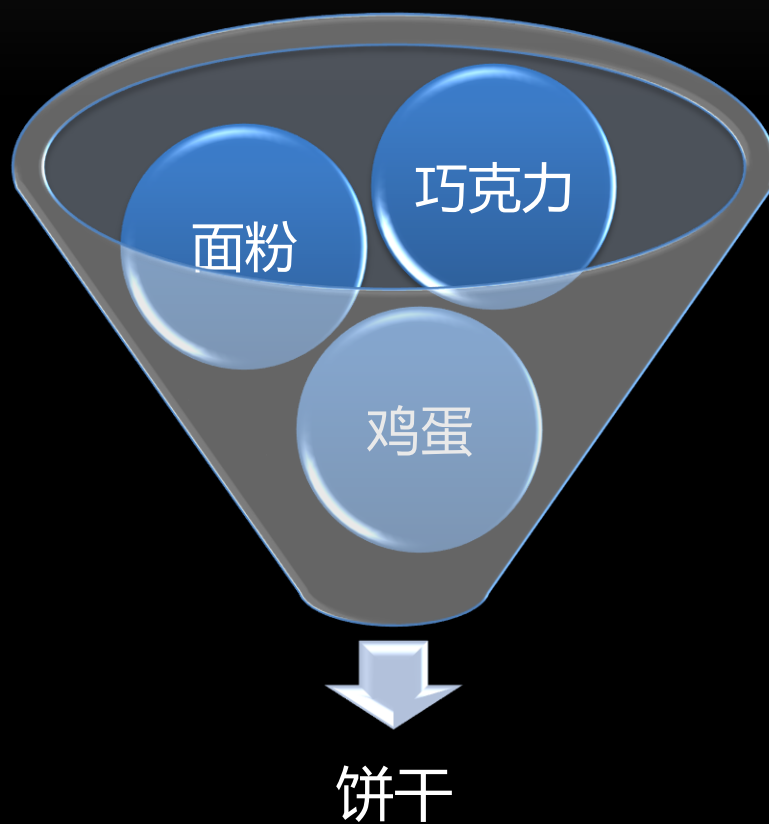
 **Alibaba.com**[®]
Global trade starts here.™

但是往往 被设计成了.....



需要设计评审

统一实现方案

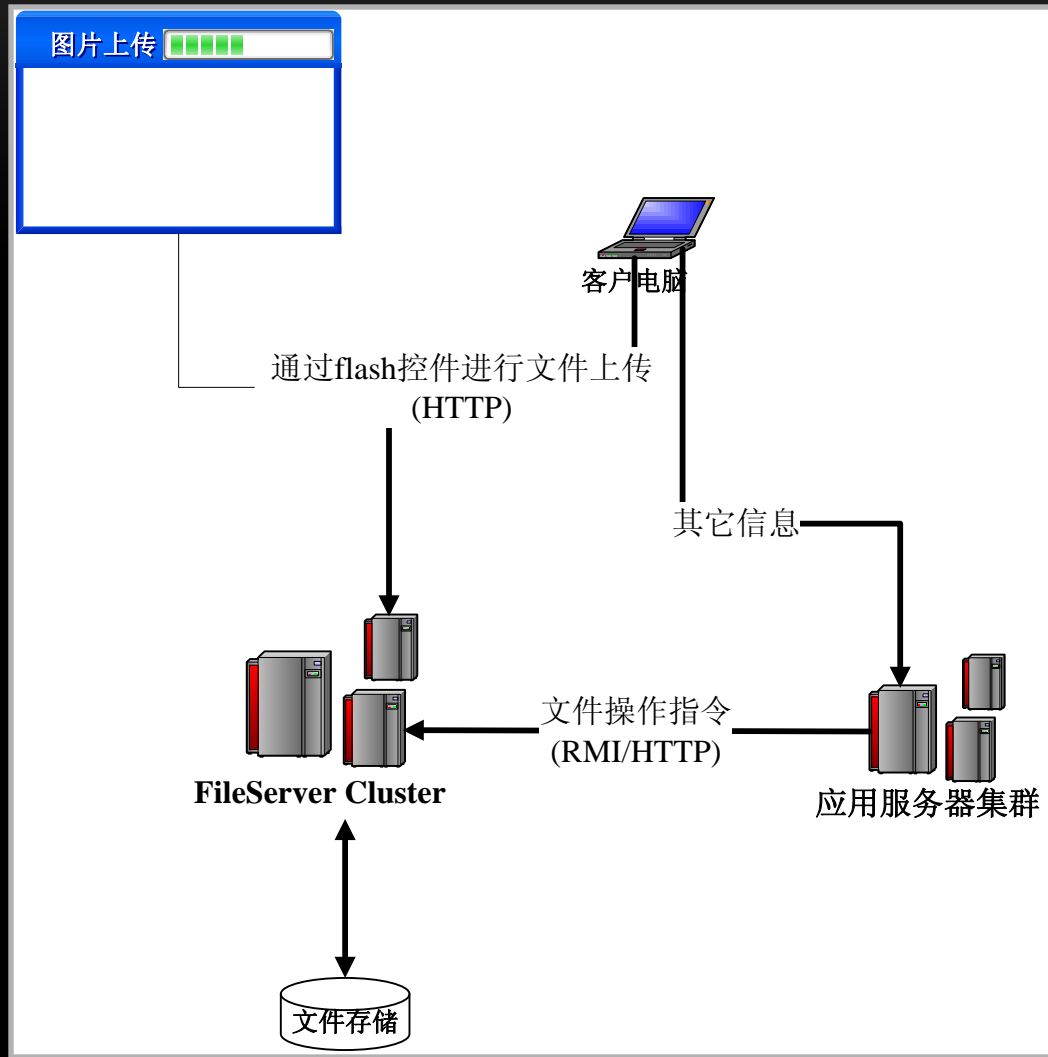


系统设计

- 产物：
 - 整体方案描述
 - 模块关系图 & 类图 & 接口& 数据结构
 - 文件目录结构
- 问题：
 - 需求变更
 - 技术人才的商业意识不够强

例如

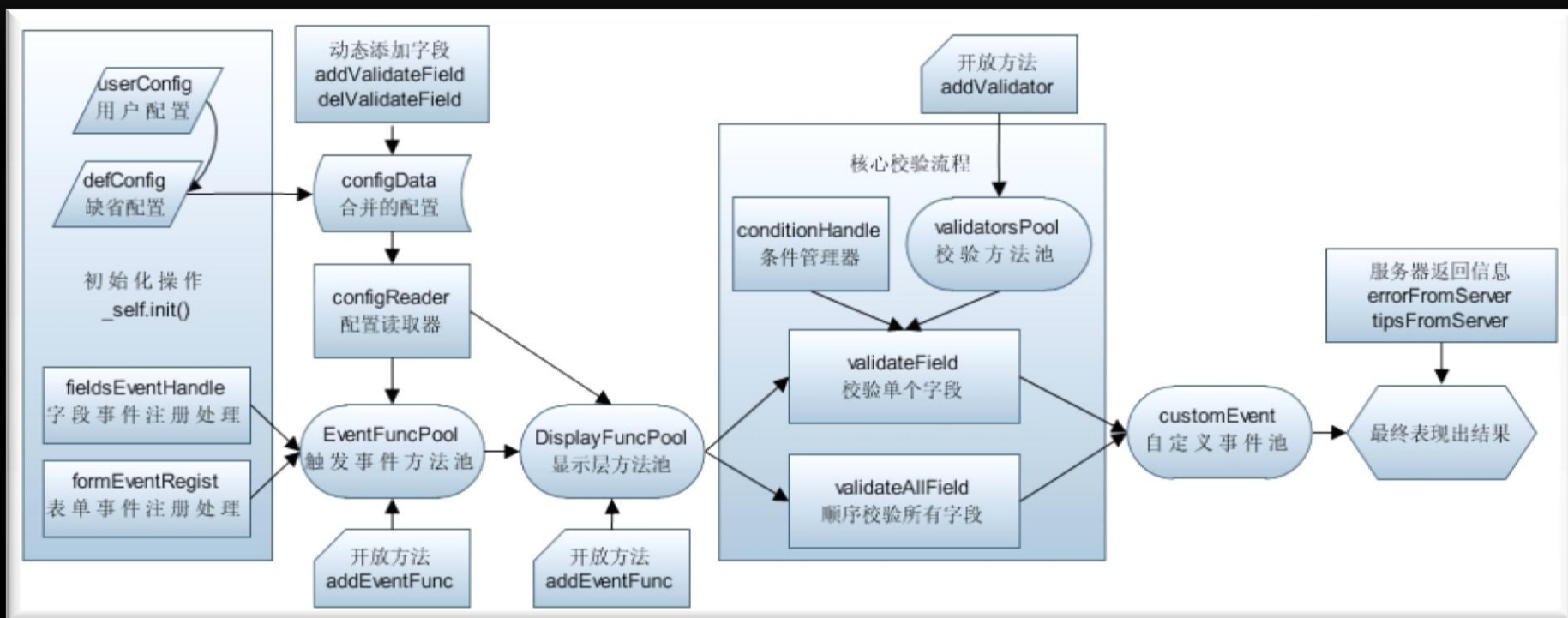
总体方案图一示例



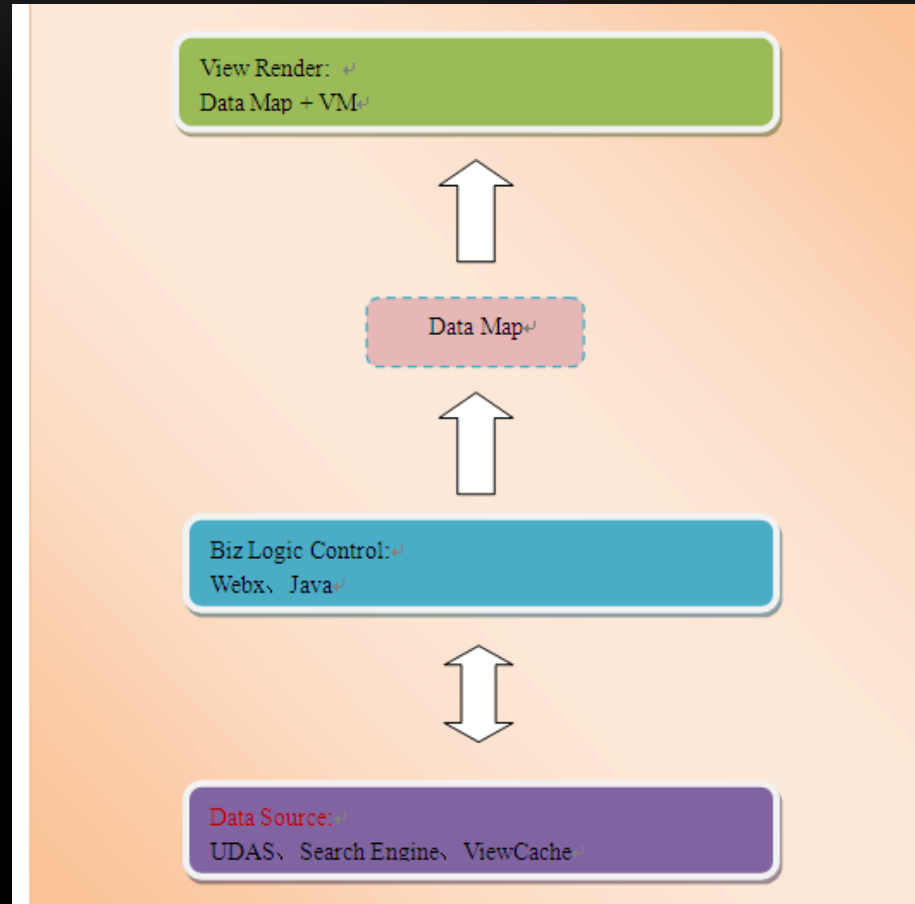
设计文档一示例



模块关系图一示例



分层图一示例



接口说明一示例

DataMap 定义:

使用方法	返回值	使用说明
<code>SdataMap.get("isddp")</code>	Y 或 N	是否为属性直达条件
<code>SdataMap.get("historyList")</code>	<code>List<NarrowPropertyItem></code>	获取用户浏览历史纪录
<code>SdataMap.get("historyMap")</code>	<code>Map<Key,NarrowPropertyItem></code>	获取用户浏览历史纪录
<code>SdataMap.get("catePropList")</code>	<code>List<NarrowPropertyGroup></code>	获取类目属性筛选组
<code>SdataMap.get("catePropMap")</code>	<code>Map<Key, NarrowPropertyGroup></code>	获取类目属性筛选组
<code>SdataMap.get("relatedCategory")</code>	<code>NarrowPropertyGroup</code>	获取关联类目筛选组
<code>SdataMap.get("mainExportMarkets")</code>	<code>NarrowPropertyGroup</code>	获取 MEM 筛选组
其他属性同理		

设计 & 文档

- 目的：
 - 为开发做指导
 - 确保实现方案的质量
 - 帮助后人理解程序，降低维护成本

- 所以：
 - 只出必须的文档
 - 尽量简单
 - 尽量图形化
 - 让核心成员参与设计评审

进入开发阶段

开发者 == 消防员



好重的罐头啊!

谁设计的头盔?
又重又丑

这个大水龙头
怎么用?

救人如救火
到底先救哪个?

5分钟搞定?

什么?
边上茅房也着火了?

开发



开发

- 问题：
 - 需求变更
 - 未预料到的风险发生
 - 历史代码不敢动
 - 外界干扰
 - 联调，各方面资源难以同步
 - 要兼容N个浏览器
 - 多人合作的开发环境配置管理

代码检查 & 安全检查

- 检查是否符合规范
- 产出检查报告
- 前端0.5天左右
- 后端1天左右
- 安全检查专人负责

测试

- 产物：
 - TC
 - TC评审
 - 功能测试报告
 - 压力测试报告
 - 用户测试报告

- 问题：
 - 需求变更
 - BUG无法及时解决
 - 测试环境不稳定
 - 线下环境与线上环境存在差异
 - 多浏览器测试成本高

发布

- 产物：
 - 发布计划
 - 上线
- 问题：
 - 浏览器缓存
 - 服务器缓存
 - 网络运营商屏蔽

解决问题



解决具体问题--需求变更

- 原因：
 - 产品规划不清晰
 - 情况发生变化
 - 测试发现问题
- 应对之策：
 - 流程保证，各种评审，需求变更流程
 - 能力提升。
 - 拥抱变化

解决具体问题—外界干扰

- 问题描述：
 - “有个紧急BUG，来帮忙修一下吧”
 - “有个方案，来帮忙评估一下吧”
 - “这个会议很好重要，来参加吧”
- 原因
 - 分工不清晰
 - 人员不足
 - 配合部门能力不全

解决具体问题—外界干扰

- 应对之策
 - 细化分工，抽调资源做维护。
 - 闭关，强制隔离
 - 资源计划
 - 培训

解决具体问题—配置管理

- 问题描述：
 - 多人开发一个环境
 - 一人开发多个环境
 - 一个项目涉及多个环境
 - 多个项目涉及一个环境
 - 发布缓存等问题

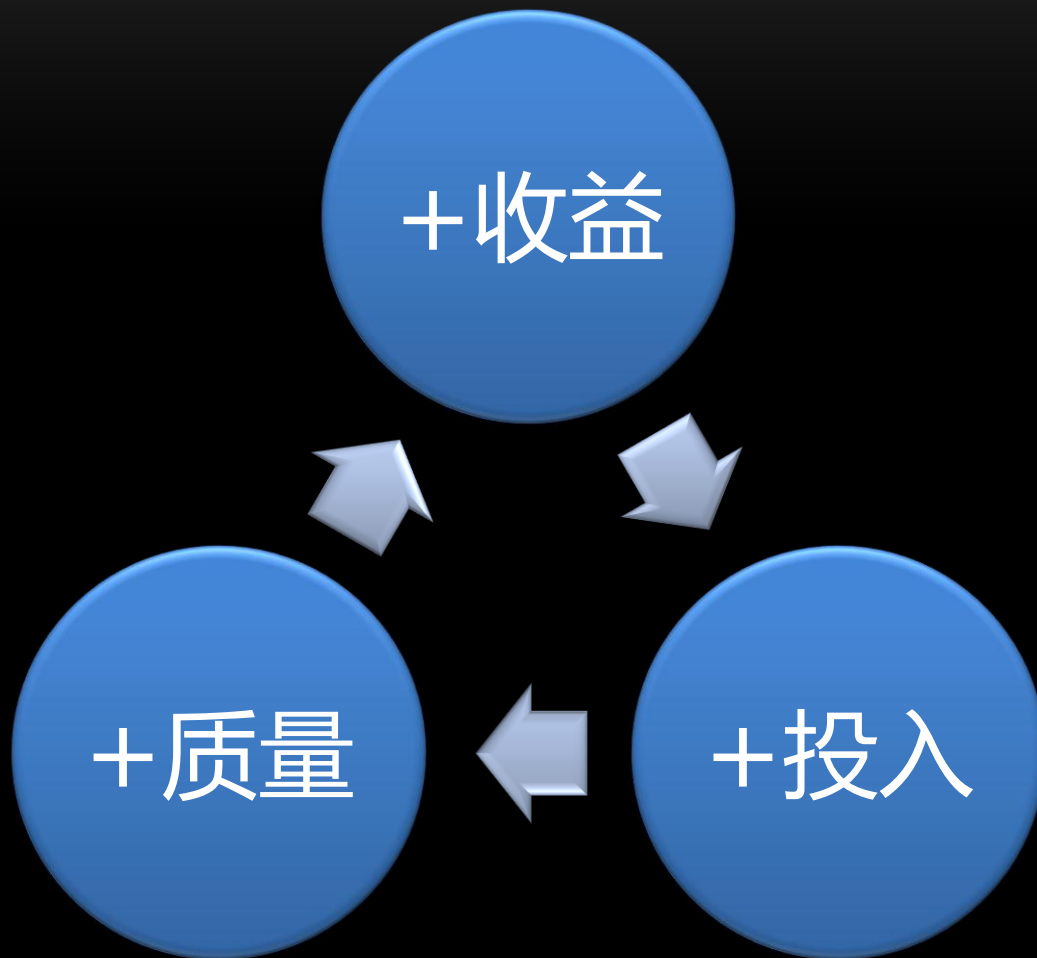
解决具体问题—配置管理

- 原因：人员&网站逐渐壮大
- 应对之策：
 - SVN
 - 重复劳动自动化，环境配置脚本、发布脚本、编译脚本、压缩工具。
 - 模块划分
 - 开发、测试、联调，环境分开

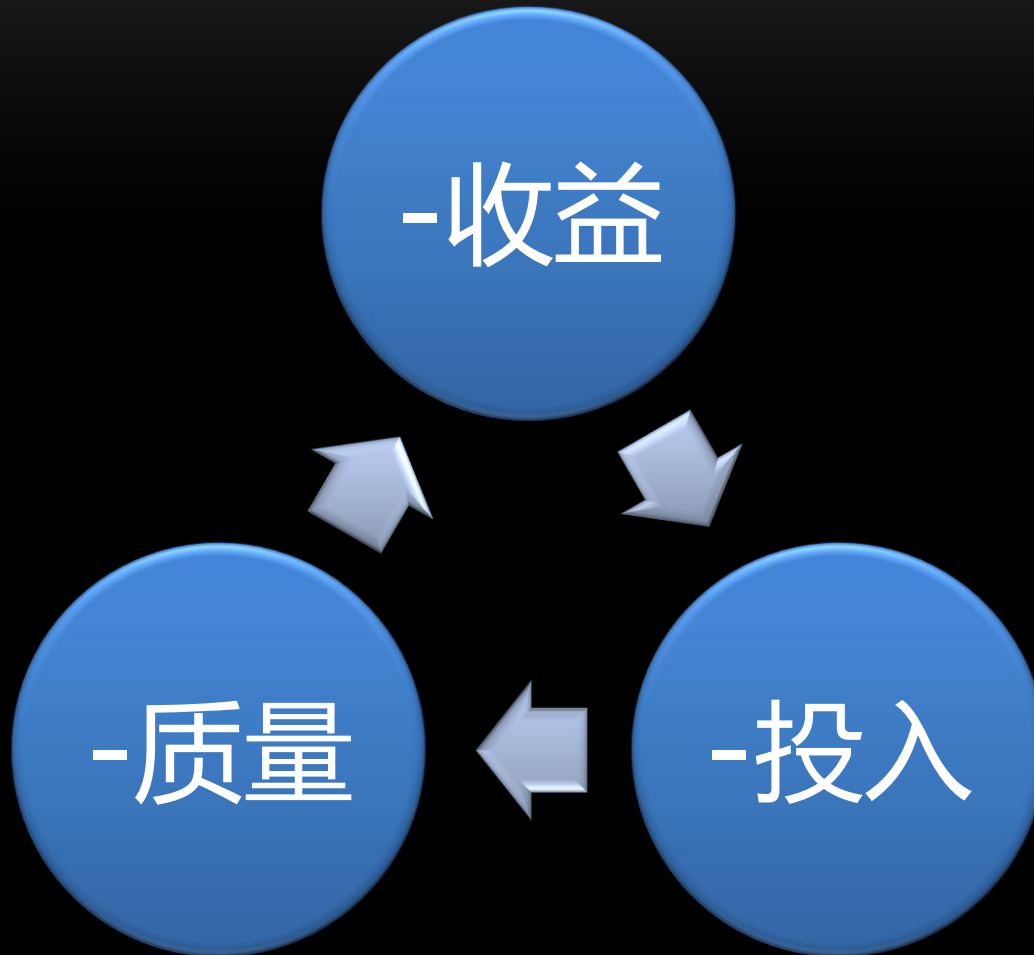
各种评审值得我们花时间吗？

利润 = 收益 - 投入

良性循环



恶性循环



先看看BUG的情况

修复功能故障

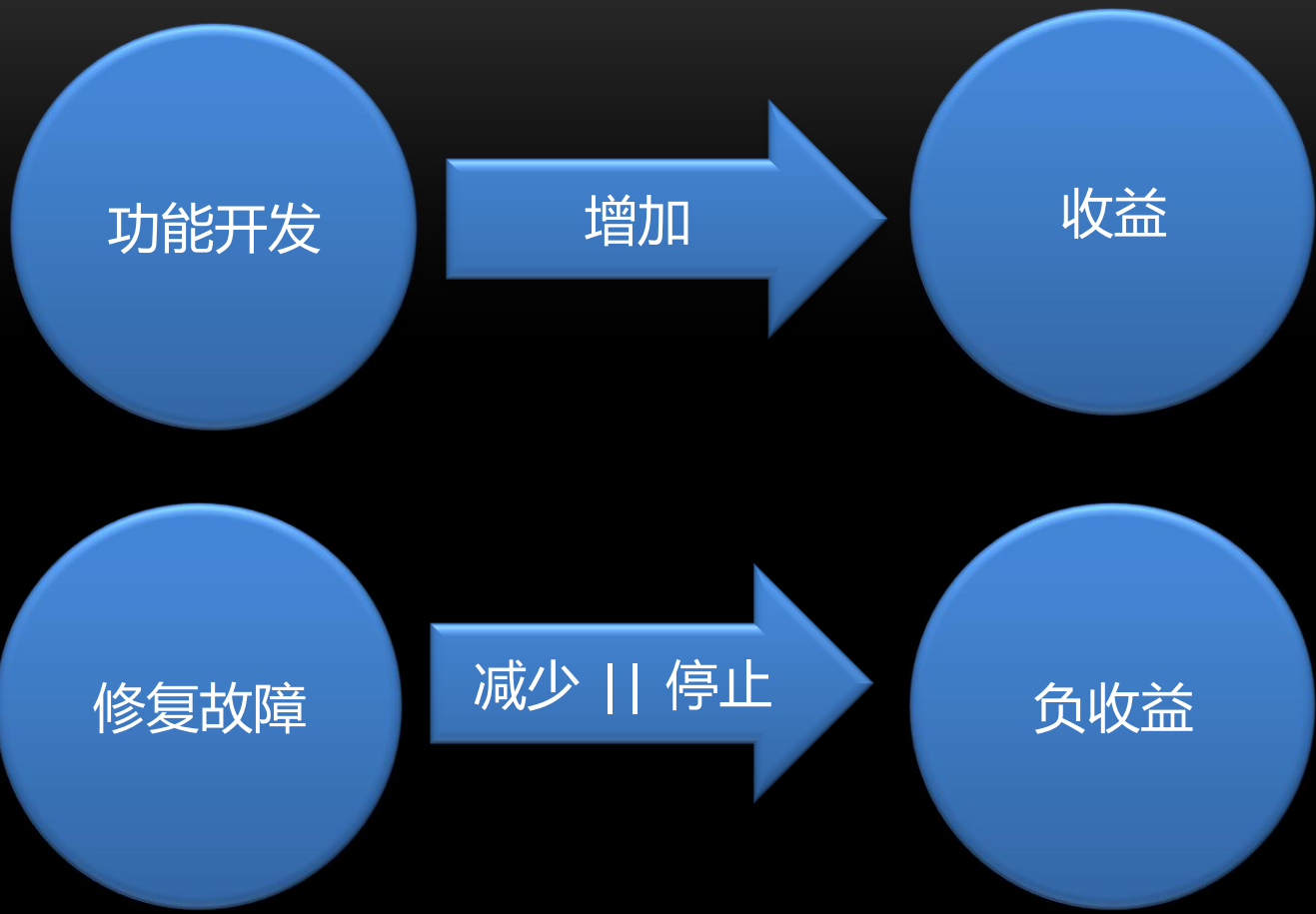
修复
4天

开发新功能
12天

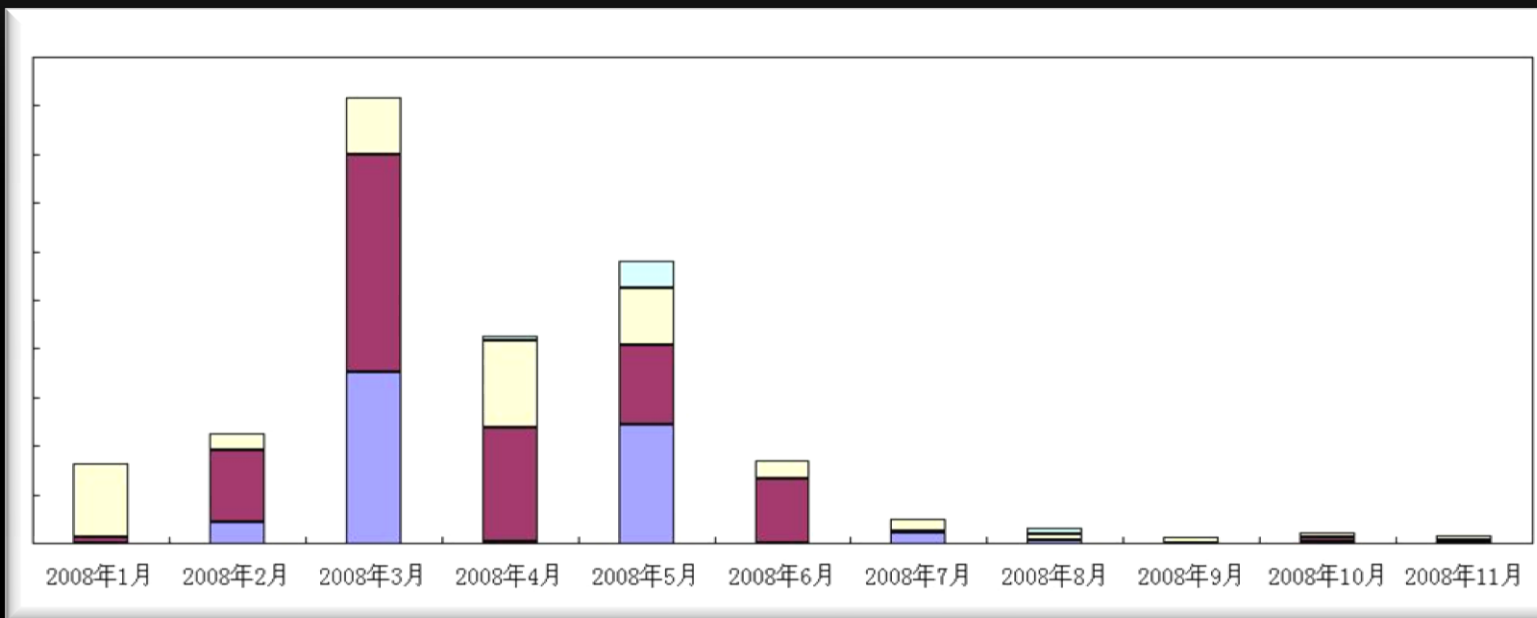
修复界面故障

修复
4小时

新界面
12小时



繁琐流程的收益-BUG减少





阿里巴巴国际站前端发展史

0.5个人--先锋

- 0.5前端

0.5个人--先锋

- 配合：
 - 完成所有的CSS&HTML
 - 组件调用
 - 后端工程师开发JS

0.5个人--先锋

- 喜悦：
 - HTML重构完成
 - 一切尽在掌握之中
- 悲伤：
 - 望着绚烂的效果叹息

2个人--进化

- 2前端

2个人--进化

- 配合：
 - 简单实现
 - 挑最重要的功能做
 - HTML基本由设计师完成
 - 功能控件化
 - 布道

2个人--进化

- 喜悦：
 - 页面交互生动活泼
 - 有了自己的前端架构
- 悲伤：
 - 功能&效果简单
 - 只有少数设计能够被实现
 - 架构规划不清晰，问题多多

3个人--成长

- 3前端

3个人--成长

- 配合：
 - 负责整站表现层的实现
 - 开发各种中型控件
 - 人肉文档
 - 布道

3个人--成长

- 喜悦：
 - 看上去像前端开发了
- 悲伤：
 - 工作压力大

10个人--规范

- 10前端

10个人--规范

- 配合：
 - 规范
 - 流程
 - 文档

10个人--规范

喜悦：

- 民兵向正规军发展
- 为用户体验提升供给能量



悲伤：

- 有时候大炮打蚊子
- 规范限制创新？



壮大

- 20前端--2009年

Q&A



Thanks