

分层自动化&持续集成

- Alibaba B2B探索之路

目录

- 1 踌躇满志 — 自动化测试
- 2 发展思考 — 分层自动化
- 3 必然之路 — 持续集成
- 4 远景规划
- 5 总结 & QA

自动化背景

发布频繁，无法及时验证

- 回归量大,成本很高
- 工作枯燥，容易出错
- 重复劳动很多

测试效率低下

- 大量工程师从事手工测试工作
- 工程师技术成就感不高
- 工作重复，效率低

开发自动化测试框架

成立专门的团队，开发包括UI, 后台等的自动化测试框架。

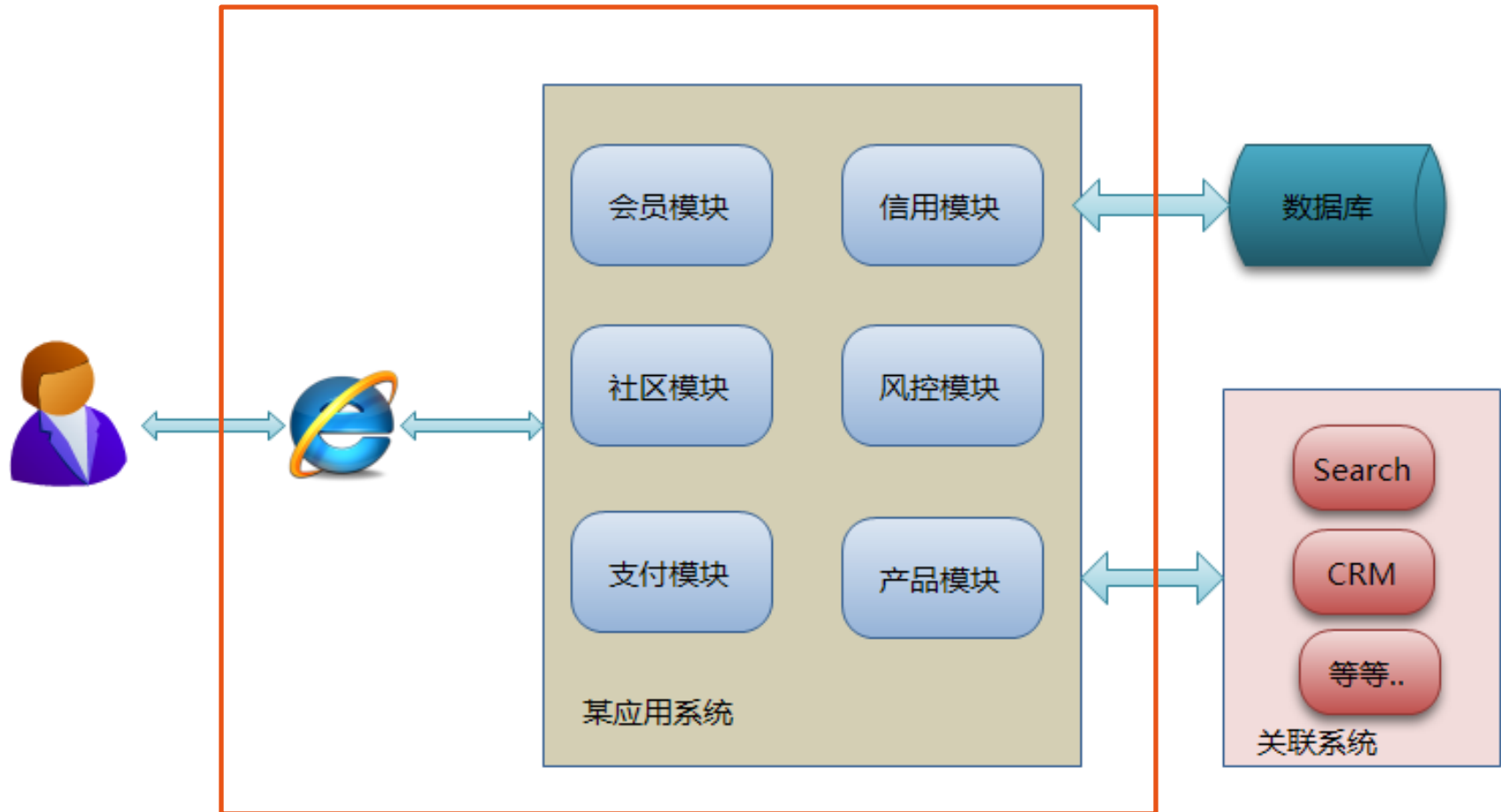
推广自动化测试

在各个团队成立自动化虚拟团队，推进自动化测试的建设，招聘专门的测试开发工程师。

梳理自动化用例

与测试同学一起，将部分适合自动化的用例整理出来，进行开发。

应用模型分析



系统级别自动化测试

自动化思路

■ 从用户的角度出发

- ◆ 能够真实模拟用户的操作

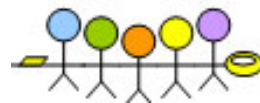
■ 实现现有手工测试用例

- ◆ 能够替代现有的手工测试用例
- ◆ 能够考虑各种异常情况

■ 能够替代繁琐的重复工作

- ◆ 每次重复的工作用自动化来实现
- ◆ 下次我能用自动化来代替我手工测试

真实， 全面， 便捷！



测试框架



自动化平台

框架类库

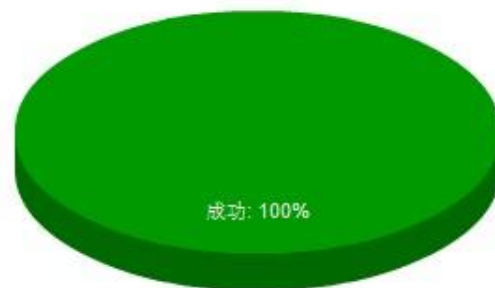
操作对象库

自动化平台

运行结果概述

运行项目	BPM
运行人	汪毅杰
通过率	100.0%
模块数	129
总场景数	11
成功场景数	11
失败场景数	0
运行编号	247032
开始时间	2011-07-04 18:19:31
结束时间	2011-07-04 18:31:13
运行时长	11分钟42秒

总体运行成功率



模块运行结果

该表显示了所有模块执行的总体结果信息，要查看具体信息请点击链接查看。

模块名字	脚本负责人	业务负责人	Total	Pass	Fail	通过率
Nir_AcceptOrder	fangxia.lifx		1	1	0	100.0%
Av_Check	fangxia.lifx		1	1	0	100.0%
Data_AuditOrder	fangxia.lifx		1	1	0	100.0%
Nir_AddOrder	zuo.xuz		1	1	0	100.0%
Data_CheckOrderPublish	fangxia.lifx		1	1	0	100.0%
Data_FinUp	wb_yejiyiing		1	1	0	100.0%
Nir_AuditAv	fangxia.lifx		1	1	0	100.0%
Data_nasdaq	wb_wangyijie		1	1	0	100.0%
Send_Service	zuo.xuz		1	1	0	100.0%
Nir_AddOpportunity	zuo.xuz		1	1	0	100.0%
Av_Send	zuo.xuz		1	1	0	100.0%

自动化的转变

追求覆盖率

1. 自动化用例的覆盖率
2. 分支等情况考虑周全

追求正确性

1. 追求正确的用例的数量
2. 关注主干的流程
3. 关注运行时间

自动流转

1. 关注流程的自动化
2. 提高脚本的重用性和使用场景

自动化的困惑

成本高,效果差：

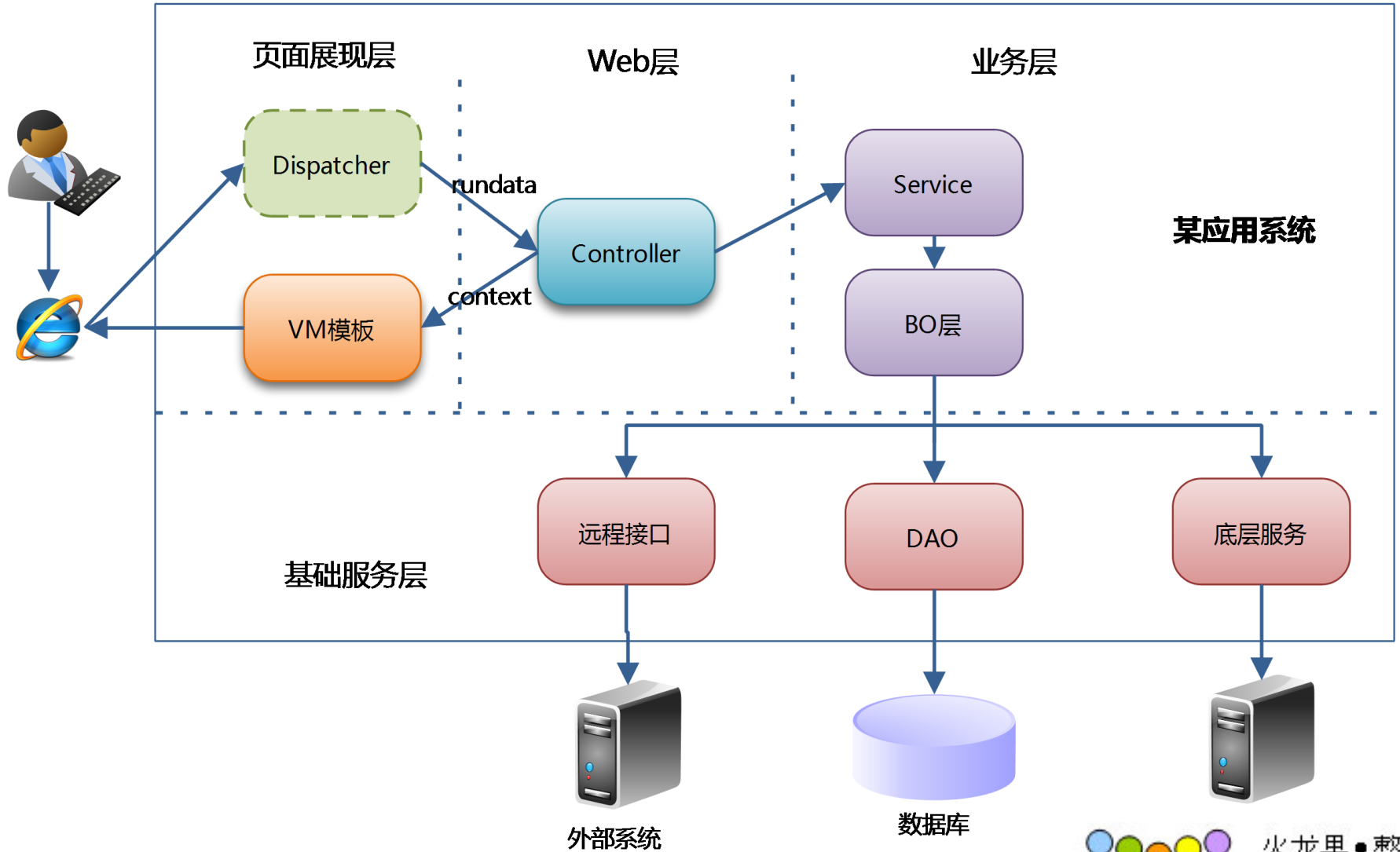
- ◆ 太多的原因导致case的失败:界面, 关联系统,数据等等..
- ◆ 代码的变更，不能及时的对应相关的用例，不能及时反馈
- ◆ 不能完整的覆盖测试点，不是所有用例都能自动化
- ◆ 系统级自动化，测试的代码覆盖率有限



目录

- 1 踌躇满志 — 自动化测试
- 2 发展思考 — 分层自动化
- 3 必然之路 — 持续集成
- 4 远景规划
- 5 总结 & QA

应用模型分析



分层思想

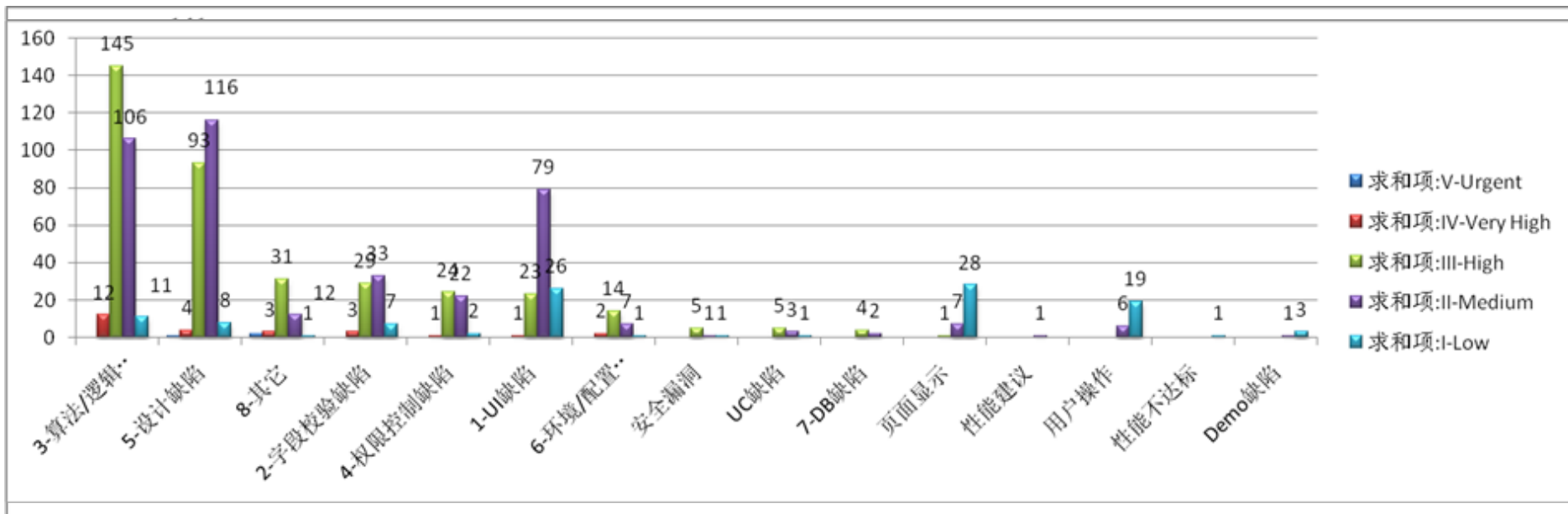
	基础服务层		业务层	Web层	页面展现层	集成测试
	DAO	接口				
颗粒度	DAO	对外暴露的远程接口	Biz层的Service	除了页面渲染层外的集成级别业务逻辑	独立的View展现功能的测试	模拟用户操作
实施者	开发	QA,开发	开发	QA	QA	QA
数据准备	真实数据准备	真实数据准备	Mock数据	真实数据准备	Mock数据	真实数据准备
执行速度	快	一般	快	一般	一般	慢
维护成本	低	一般	低	一般	较高	高
测试方法	自动化	自动化	自动化	自动化	自动/手工结合	自动/手工结合

分层思想

- ◆ DAO/业务层 (优先级1)
 - ✓ 通过开发的单元测试来保证
 - ✓ 保证代码的覆盖率
- ◆ 远程接口 (优先级2)
 - ✓ 用代码的方式来实现，争取100%覆盖
 - ✓ 完善的数据准备
- ◆ Web层 (优先级3)
 - ✓ 如果可以剥离UI的，一定要剥离UI
- ◆ UI层 (优先级3，不一定要追求完全自动化)
 - ✓ 只关注UI层的问题
 - ✓ 减少对后台数据的依赖
- ◆ 集成自动化 (优先级4)
 - ✓ 不要太复杂
 - ✓ 尽量只保证主干功能

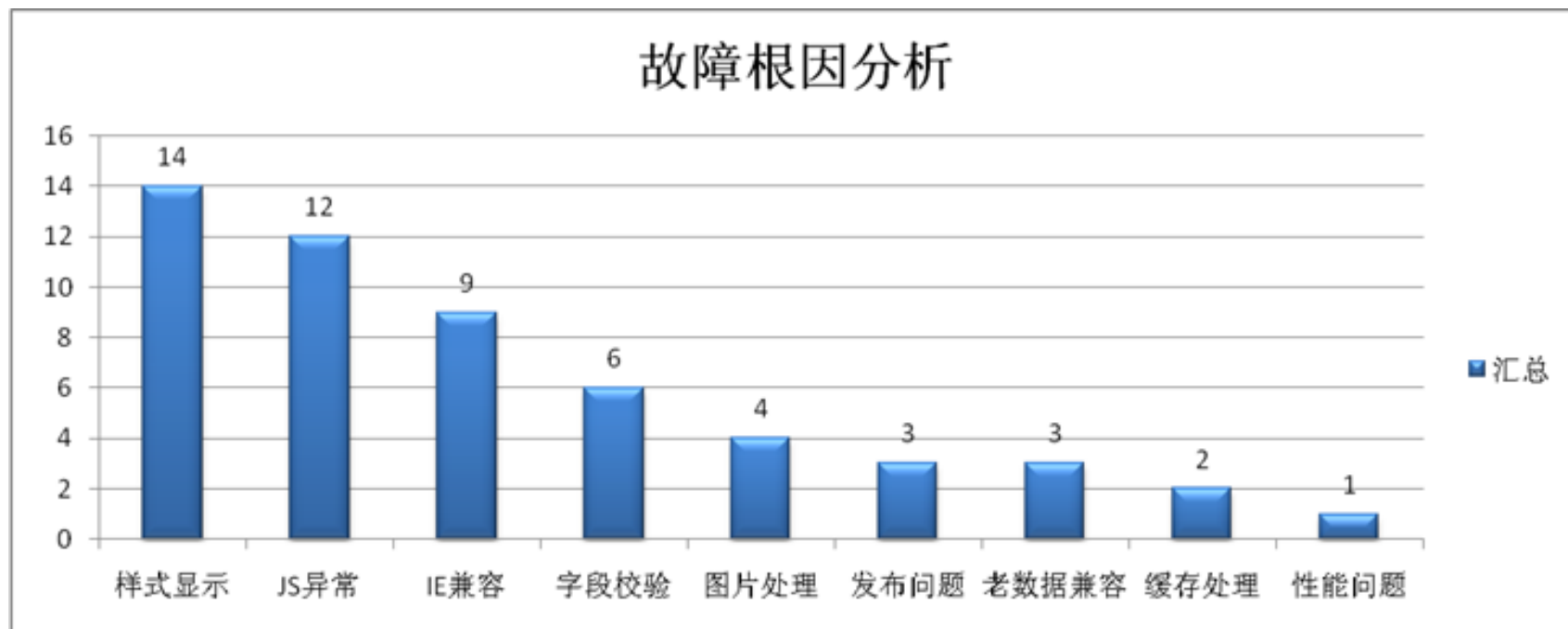
为什么UI和集成优先级低?

缺陷分析



Alibaba 国际站2010年缺陷类型分析 - 团队2

缺陷分析 – UI缺陷



2010年11月 UI缺陷故障分类

分层实践 - Web层

所有类目 共找到**391318**条符合"Mp3"查询结果

类目：	MP3	车载mp3	MP4	音箱	迷你音箱	更多
屏幕色彩：	26万色	1600万色	4096色	单色屏幕	无屏幕	更多
接口：	USB 2.0	mini USB				
FM功能：	有FM功能	无FM功能				

[显示更多](#)

您还可以找： MP3耳机 MP3手表 MP3播放器 运动MP3 插卡MP3 礼品MP3 索尼MP3 车载MP3播放器

单价 至 元 起订量 以下 高级筛选

所有地区 默认排序 支付宝 诚信保障 在线

 **供应南美地区热销款式MP3/MP4 支持支付宝操作** 07/05 标王

品牌:OEM 型号:KMC380 内存容量:1GB-8GB 屏幕色彩:无屏幕 接口:USB,2.0 支持音频格...

深圳市科美创数码有限公司

广东 深圳市福田区 ¥13.00
20台起
 和我联系

 **供应礼品MP3、加工MP3、批发mp3、生产MP3、米奇MP3，** 07/06 推广

品牌:礼品mp3 型号:0 内存容量:2GB 屏幕尺寸:3 屏幕色彩:液晶屏幕彩色 接口:USB,2.0 支持...

广东 深圳市 ¥50.00
100台起
 和我联系

火龙果·整理
uml.org.cn

分层实践 – Web层

测试方法

1. 在自己的Spring容器内测试，而不是Web容器内
2. Mock HttpRequest和RunData值(页面参数的封装类)
3. 直接调用Controller类，取得页面渲染前的Context
4. 直接校验Context

关联应用

- ◆ Mock相关的Search，缓存等外部系统

执行效果

- ◆ 执行时间大大缩短，提高了反馈时间
- ◆ 剥离了不同类目页面展现的差异性
- ◆ 专注于业务逻辑，测试的投入产出比高

UI自动化?

- ◆ 是否继续需要做UI自动化？
 - 做，但是如果有更底层的方法，用底层的方法
 - 只做主要的功能验证，脚本尽量简单
- ◆ 如何降低UI自动化成本？
 - 剥离数据对脚本的影响
 - 完善的数据准备
 - 环境的自动构建，对依赖环境的mock
 - 流程的自动化
 - 及时反馈变更
- ◆ UI的其他方法？
 - 图片对比

图片对比

主

支

Spotlight

Cut your sourcing time in half with instant access to FOB Pricing, MOQ and other sourcing details!

Novelty Products

Gifts & Crafts >>



Electronic >>



Home & Office >>



Cut your sourcing time in half with instant access to FOB Pricing, MOQ and other sourcing details!

Novelty Products

Gifts & Crafts >>



Electronics >>



Home & Office >>



LED Umbrella



From US\$1.0
MOQ: from 100pcs

Car Wireless Mouse



From US\$7.0
MOQ: from 200pcs

Wooden Telephone



From US\$7.0
MOQ: from 100pcs

Cut your sourcing time in half with instant access to FOB Pricing, MOQ and other sourcing details!

LED Umbrella



From US\$1.0
MOQ: from 100pcs

Car Wireless Mouse



From US\$7.0
MOQ: from 200pcs

Wooden Telephone



From US\$7.0
MOQ: from 100pcs

对比结果



火龙果 • 整理
uml.org.cn

效益最大化?

- ◆ 这么多测试手段如何结合起来？
- ◆ 怎么样能够最快的反馈？
- ◆ 自动化脚本的效益，价值能最高？



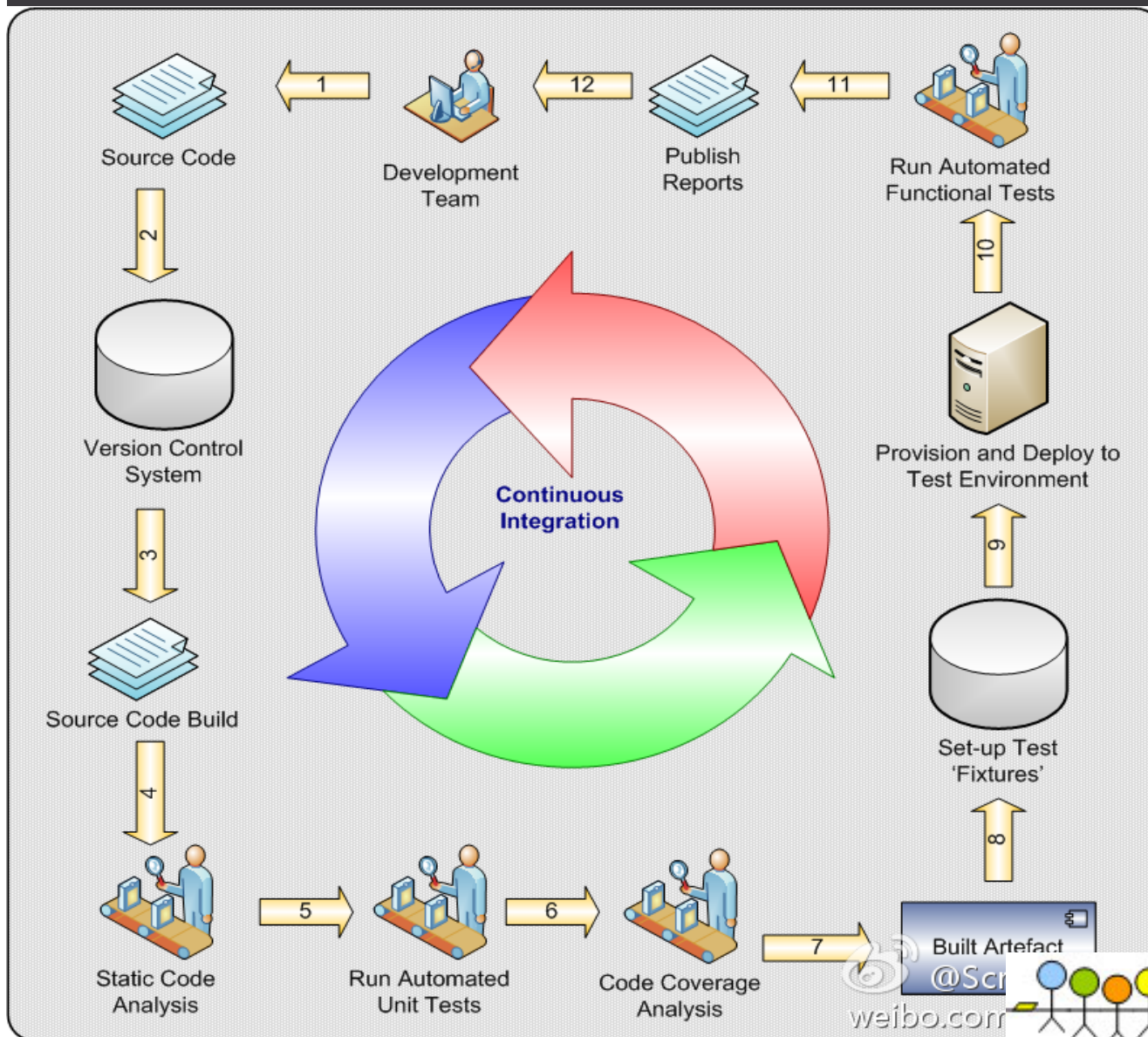
目录

- 1 踌躇满志 — 自动化测试
- 2 发展思考 — 分层自动化
- 3 必然之路 — 持续集成
- 4 远景规划
- 5 总结 & QA

持续集成的目的

- ◆ 及时的反馈
- ◆ 自动化脚本价值的最大化
- ◆ 减少定位问题发生范围
- ◆ 流程的自动化，提高工作效率

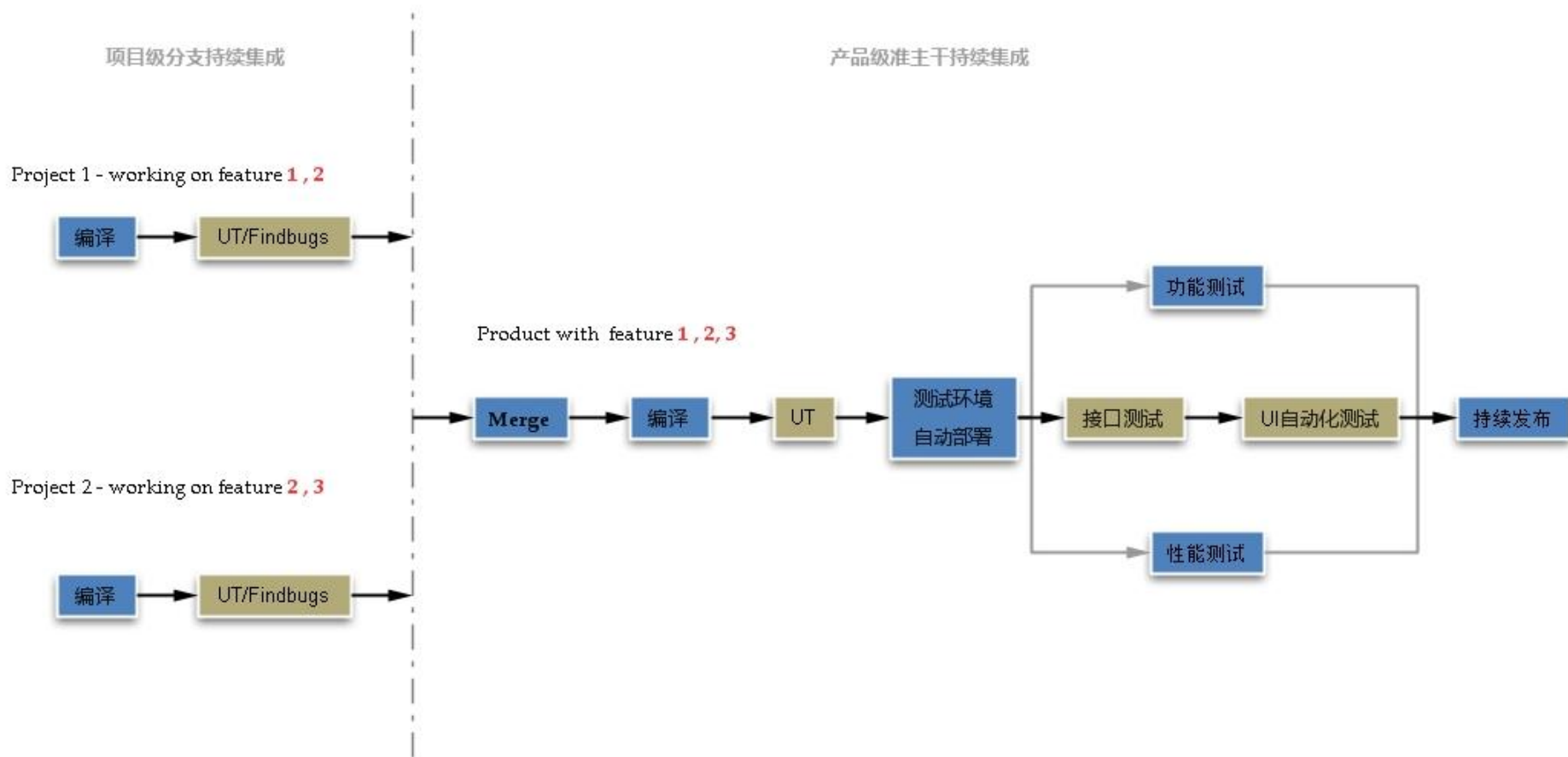
持续集成 - 项目级



weibo.com



持续集成 - 产品级



持续集成思路

- 检测代码提交，提测，发布的动作，自动执行代码检测，单元测试
- 代码检测，单元测试通过后，自动部署测试环境
- 自动执行对应的集成级别自动化测试(包括Web层测试)
- 在发布，提测阶段，如果自动化测试通过，直接通知QA接受
- 在各个环节中，如果有失败情况，自动反馈给相应的人员

持续集成支撑

- 代码覆盖率的改进
 - 提供增量代码覆盖率
 - 支持集成测试代码覆盖率自动收集
 - ✓ 分布式
 - ✓ 自动合并
 - ✓ 定时获取
- 测试环境自动部署
 - 自动选择合适的机器
 - 主动拿包，部署相关环境和测试环境
 - 测试环境代码覆盖率插桩
 - 发送测试host文件
- Hudson平台与其他平台的交互



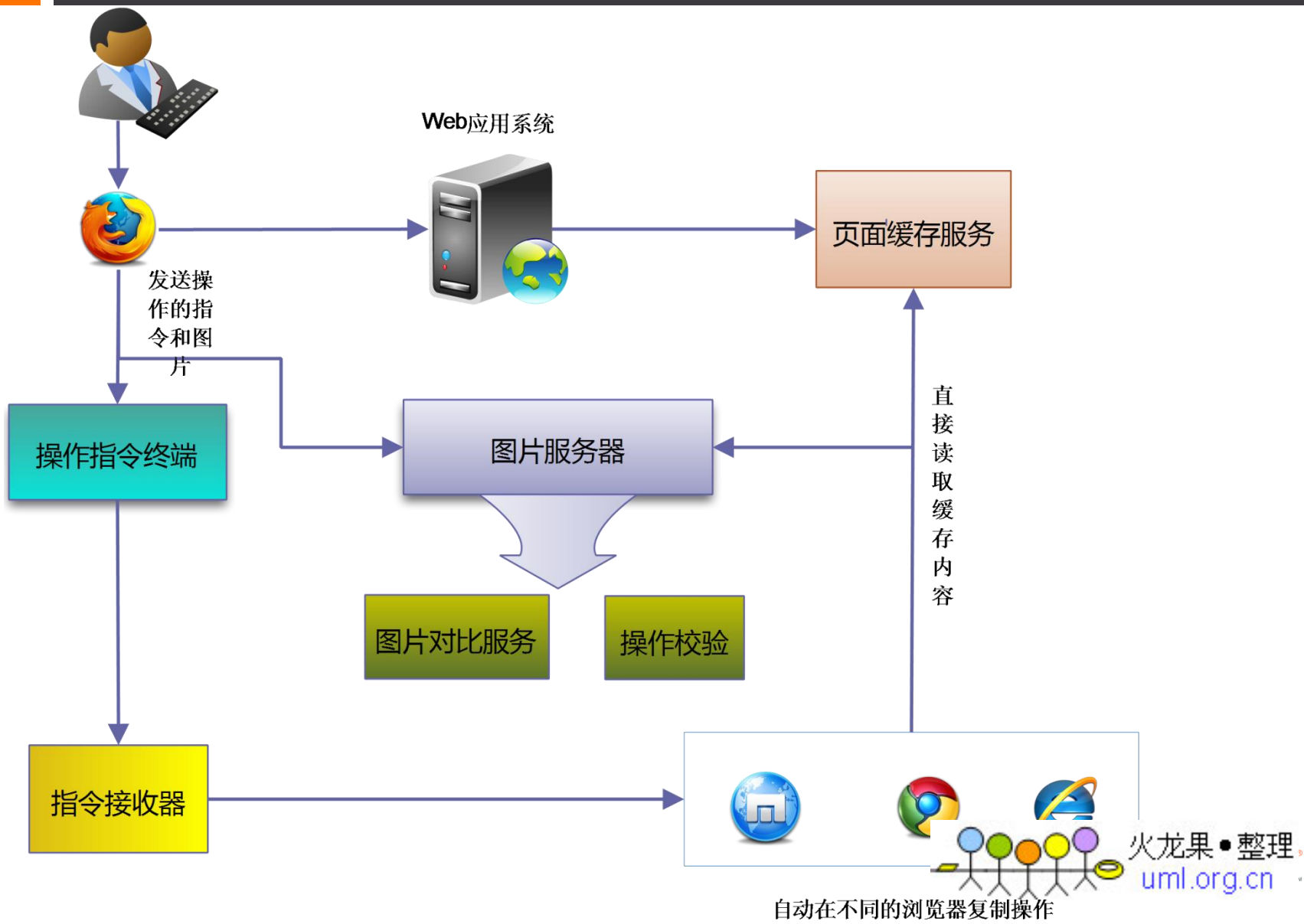
目录

- 1 踌躇满志 — 自动化测试
- 2 发展思考 — 分层自动化
- 3 必然之路 — 持续集成
- 4 远景规划
- 5 总结 & QA

分层之路

- 测试用例的分层，各种测试方法的有效搭配
 1. 单元测试,接口测试，Web层测试的分工
 2. 自动化测试与手工测试的搭配
 3. 分层用例如何彼此不重复，最高效?
- UI问题的解决
 1. 解决多浏览器兼容，样式问题
 2. 前端业务逻辑(前端单元测试?)

多浏览器测试设想



“处处” 自动化

- 测试数据，跨平台数据
 1. 解决手工测试中数据准备问题
 2. 充分利用自动化中的成果
- 环境依赖
 1. 减少对重量级系统的依赖
 2. 功能数据Mock

目录

- 1 踌躇满志 — 自动化测试
- 2 发展思考 — 分层自动化
- 3 必然之路 — 持续集成
- 4 远景规划
- 5 总结 & QA

总结

1. 自动化测试，越底层收益越大
2. 做分层，需要结合缺陷等数据，做最有价值的区域
3. 不要为了持续集成而持续集成，注重基础设施建设
4. 在分层测试中，不是QA一个人的事情
5. 将能够自动的都整合到CI，能更加有价值

QA环节

