

OSC 原创会

年终盛典 2016

应用升级进阶之路

Bugly 一种愉悦的开发方式



移动互联网时代，一切都更快捷
但应用升级却遇到了这些问题.....



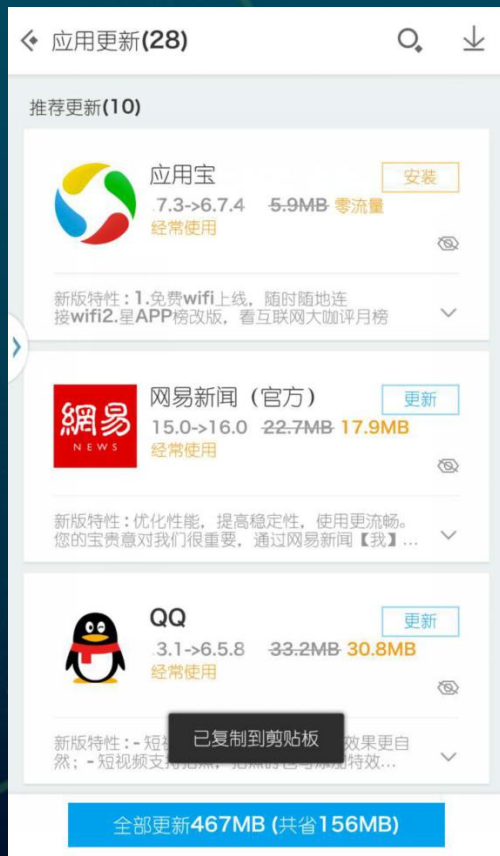
- 应用的平均上线时间需**10小时以上**
- 平均迭代周期在**一个月至一个半月内**
- 一年平均发布**8—10个版本**

iOS平台		Android平台
695	发布版本总量 (个)	739
73	平均上线节奏 (小时)	10
44	平均发布周期 (天)	33
8	平均发布版本 (个/年)	10

应用市场 转化少

OSC 原创会
年终盛典 2016

- 应用市场中的版本升级需依赖应用市场主动推送，用户主动点击，**版本转化率小。**



应用市场 效率低

OSC 原创会
年终盛典 2016

- 基于应用市场版本升级转化率小，如果希望用户能升级最新版本获得最佳体验，往往需要开发；运营；产品；客服等多角色参与，版本兼容成了大问题。



弹窗升级 & 热更新

OSC原创会
年终盛典 2016

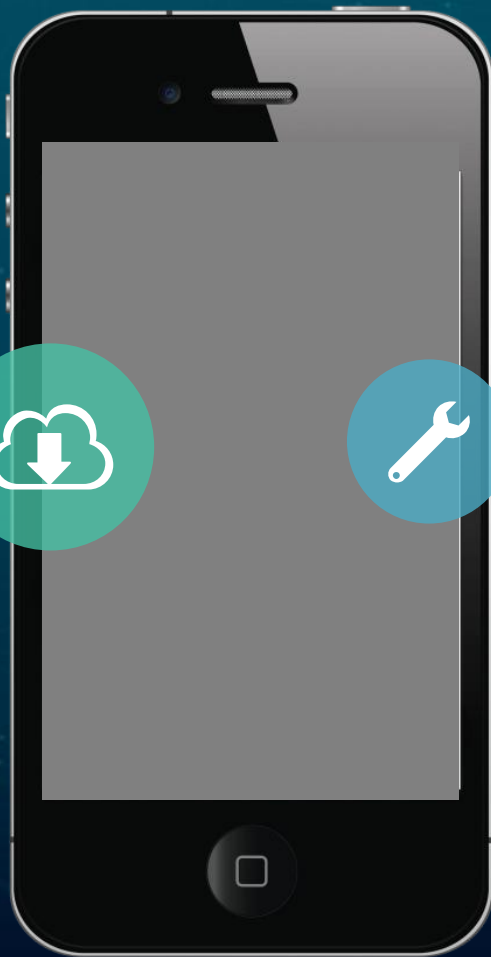
为了让用户能更便捷的体验应用最新版本，弹窗升级&热更新技术应运而生

弹窗升级

在App内弹框提醒用户升级，并可对升级用户范围及数量进行精准控制。

适用场景

新功能上线；页面改版；新版本灰度



热更新

在用户无感知的状态下完成Bug修复，并可多纬度设定更新范围，精准命中问题版本。

适用场景

线上Bug修复



iOS

- WaxPatch (Lua)
- JSPatch (Javascript)

Android

- QZone 解决方案 (第三方基于该方案实现的有 Nuwa 等)
- 微信 Tinker
- AndFix
- Dexposed

各框架差异可参阅:

<http://www.jianshu.com/p/e9c6d4f0d073>

<http://albert43.net/2015/07/12/JSPatch总结/#jspatch-wax>

iOS 主流热更新框架差异

OSC 原创会
年终盛典 2016

WaxPatch (Lua)

利用 Objective-C 语言的动态性及消息传递机制实现方法替换，新增方法。WaxPatch 基于 Wax 进行了扩展，主要是增加了替换原始方法实现的能力

优势

支持 iOS 6，理论上支持所有 iOS 系统版本

劣势

需要引入解析引擎，包体积会增大，并存在一定的审核风险

iOS 主流热更新框架差异

OSC 原创会
年终盛典 2016

JSPatch (JavaScript)

通过 iOS 7.0 新增的 JavaScriptCore 及开放的接口与 Objective-C 互传消息，利用 Objective-C 语言的动态性及消息传递机制实现方法替换，新增方法等

优势

基于系统内置 JavaScriptCore，无需额外引入解析引擎

JavaScript 语言相比 Lua 有更广泛的群众基础，并且不存在审核风险

劣势

不支持 iOS 7 以下系统

QZone 解决方案

Android 中 ClassLoader 可以包含多个 dex 文件，如果在不同的 dex 中有相同的类存在，那么会优先选择排在前面的 dex 文件内的类，所以把需要替换的类打包到一个 dex 文件中，然后把这个 dex 插到最前面就可以满足热更新的需求

优势

开发透明，简单，是所有热更新方案中可靠性最高的，Java 层兼容性问题会比较少

劣势

需要类进行插桩来跳过 CLASS_ISPREVERIFIED 的校验，有一定的性能影响，主要体现在启动速度上，补丁只有在下次启动才生效

Android 主流热更新框架差异

OSC 原创会
年终盛典 2016

微信 Tinke

自研dxdiff算法，在编译时通过比较新旧两个Dex生成差异patch.dex。在运行时，将差异patch.dex重新跟原始安装包的旧dex还原成新的Dex

优势

功能完善同时支持代码、so库以及资源的修复，全平台支持，性能损耗较小

劣势

占用Rom体积，有一个额外的合成过程内存占用过大



如何选型

OSC 原创会
年终盛典 2016

开发人员选择服务时应该主要关注以下几点



安全性

防止补丁被篡改

兼容性

支持 iOS, Android 及各自的主流系统版本



灰度控制

能有效的控制补丁影响范围



Bugly 弹窗升级 & 热更新



Bugly 选用的技术框架

OSC 原创会
年终盛典 2016

iOS 基于 JSPatch; Android 基于 Tinker 方案

Why JSPatch

- JavaScript 比 Lua 在应用开发领域有更广泛的应用
- 使用系统内置的 JavaScriptCore.framework, 无需内嵌脚本引擎, 体积小
- 更符合 Apple 的审核规则

Why Tinker

- 性能耗损低
- 应用成功率高
- 支持资源, OS 更新
- 成熟, 稳定, 大产品验证 (微信)

微信Tinker 的设计目标

OSC 原创会
年终盛典 2016

- **开发透明：** 开发者无需关心是否在补丁版本，他可以随意修改，不由框架限制
- **性能无影响：** 补丁框架不能对应用带来性能损耗
- **完整支持：** 支持代码，so 库以及资源的修复，可以发布功能
- **补丁大小较小：** 补丁大小应该尽量的小，提高升级率
- **稳定，兼容性好：** 保证微信的数亿用户的使用，尽量减少反射



Dexdiff 算法目的

OSC 原创会
年终盛典 2016

The DEX file format

Magic	
checksum	
signature	
File size	Header size
Endian tag	Link size
Link offset	Map offset
String IDs Size	String IDs offset
Type IDs Size	Type IDs offset
Proto IDs Size	Proto IDs offset
Field IDs Size	Field IDs offset
Method IDs Size	Method IDs offset
Classdef IDs Size	Classdef IDs offset
Data Size	Data offset

Type	Implies	Size	Offset
0x0	DEX Header	1 (Implies Header Size)	0x0
0x1	String ID Pool	Same as String IDs size	Same as String IDs offset
0x2	Type ID Pool	Same as Type IDs size	Same as String IDs offset
0x3	Prototype ID Pool	Same as Proto IDs size	Same as ProtoIDs offset
0x4	Field ID Pool	Same as Field IDs size	Same as Field IDs offset
0x5	Method ID Pool	Same as Method IDs size	Same as Method IDs offset
0x6	Class Defs	Same as ClassDef IDs size	Same as ClassDef IDs offset
0x1000	Map List	1	Same as Map offset
0x1001	Type List	List of type indexes (from Type ID Pool)	
0x1002	Annotation set	Used by Class, method and field annotations	
0x1003	Annotation Ref		
0x2000	Class Data Item	For each class def, class/instance methods and fields	
0x2001	Code	DexCodeItems -- contains the actual byte code	
0x2002	String Data	Pointers to actual string data	
0x2003	Debug Information	Debug_info_items containing line no and variable data)	
0x2004	Annotation	Field and Method annotations	
0x2005	Encoded Array	Used by static values	
0x2006	Annotations Directory	Annotations referenced from individual classdefs	

- diff 结果小
- 合成过程占用内存小
- 支持删除、新增、修改 dex 中的 class
- 通过DexDiff算法的实现，既解决Dalvik平台的性能损耗问题，又解决了Art平台补丁包过大的问题

用例ID	原始Dex大小(KB)	新Dex大小(KB)	BsDiff(KB)	DexDiff(KB)	时间(ms)	峰值内存(M)
1	228.72	624.76	48.00	1.64	95	3
2	10617.16	10617.7	260.00	7.64	7917	26
3	12723.15	12723.2	173.00	1.56	8448	39
4	12744.78	12744.78	10570.00	5881		

Tinker 的框架设计

OSC 原创会
年终盛典 2016



Bugly 热更新平台特色

OSC 原创会
年终盛典 2016

可靠的安全性

- 策略加密传输 (iOS HTTPS, Android RSA)
- 文件校验, 防篡改

丰富的维度控制

- 可按操作系统, 下发量, 开发环境/生产环境控制热补丁下发范围

异常解决方案

- 丰富的异常解决方案供参考

问题无缝修复

- 发现 Crash, 下发 Hotfix 一气呵成, 无需跳转多个平台.

服务免费

腾讯 CDN技术支持

iOS / Android 双平台支持



Bugly 热更新平台特色

OSC 原创会
年终盛典 2016

发布新补丁 ✕

补丁文件 请选择补丁文件 选择文件

目标版本

下发范围 开发设备 全量设备 自定义

下发数量 0 即不限制下发数量

系统版本

备注

0/500

立即下载

Bugly 弹窗升级平台特色

OSC 原创会
年终盛典 2016

弹窗样式多选

- 三类基础弹窗样式
- 根据产品UI风格自定义弹窗样式

丰富的维度控制

- 可按版本，渠道，网络环境对升级用户进行筛选，精准控制版本灰度范围

灵活控制启动&停止条件

- 可根据预定时间或实际升级用户数，精准控制版本灰度范围

数据统计实时看

- 提供分发、下载、安装、激活等多维度数据监控，让您实时了解用户的升级转化率

升级方式

- 可根据产品需求选择推荐升级或强制升级



Bugly 弹窗升级平台特色

OSC 原创会
年终盛典 2016

创建升级策略

BuglySDKDemo2
包名: xxx.bugly.demo 版本: 2 包大小: 435.09KB [重新选择](#)

基础配置

策略名称:

升级方式: 推荐升级 强制升级

高级配置

策略下发条件:

指定源版本:

升级渠道:

网络环境:

策略启动条件:

策略停止条件:

自动弹窗规则:

弹窗样式

自定义

Bugly SDKDemo2

版本: 2
包大小: 435.09KB
更新时间: 2016-04-28 12:18:32

更新说明:
Bugly DEMO

Bugly 一种愉悦的开发方式

OSC 原创会
年终盛典 2016



<https://bugly.qq.com>

