目录

[一、 APP开发 2](#_Toc497303744)

[1. 简介 2](#_Toc497303745)

[2. 开发流程 2](#_Toc497303746)

[3. App系统 3](#_Toc497303747)

[4. APP开发工具 3](#_Toc497303748)

[5. 发展背景 4](#_Toc497303749)

[6. app数据 4](#_Toc497303750)

[7. 意义 4](#_Toc497303751)

[8. 市场现状 5](#_Toc497303752)

[9. 监管政策 5](#_Toc497303753)

[10. 解决方案 6](#_Toc497303754)

[11. 现金流与品牌 6](#_Toc497303755)

[12. 开发技术分类 6](#_Toc497303756)

[二、 Native App（原生开发） 7](#_Toc497303757)

[1. 基本简介 8](#_Toc497303758)

[2. 概述 8](#_Toc497303759)

[3. 发展现状 9](#_Toc497303760)

[三、 Web App 9](#_Toc497303761)

[1. 基本简介 9](#_Toc497303762)

[2. 区分 10](#_Toc497303763)

[3. 界面 10](#_Toc497303764)

[4. 优点 12](#_Toc497303765)

[5. 制作 12](#_Toc497303766)

[6. 开发 13](#_Toc497303767)

[四、 Hybrid App（混合开发） 13](#_Toc497303768)

[1. 基本简介 14](#_Toc497303769)

[2. 兴起原因 15](#_Toc497303770)

[3. 分类 15](#_Toc497303771)

[4. 未来发展 17](#_Toc497303772)

[5. hybrid app开发工具 19](#_Toc497303773)

[五、 Web App、Hybrid App与Native App的设计差异 21](#_Toc497303774)

[1. 引言 21](#_Toc497303775)

[2. Web App、Hybrid App、Native App 纵向对比 21](#_Toc497303776)

[3. Web App、Hybrid App、Native App 技术特性 23](#_Toc497303777)

[4. Web App受限制因素及设计要点 23](#_Toc497303778)

[5. 小结 31](#_Toc497303779)

1. APP开发

# 简介

App开发，是指专注于手机应用软件开发与服务。 App是application的缩写，通常专指手机上的应用软件，或称手机客户端。另外目前有很多在线app开发平台。

当然移动互联网时代是全民的移动互联网时代，是每个人的时代，也是每个企业的时代。APP便捷了每个人的生活，APP开发让每个企业都开始了移动信息化进程。

# 开发流程

首先，制作一款APP，必须要有相关的idea，也就是说，第一步是APP的idea形成。

其次，就是通过那些idea来进行APP的主要功能设计以及大概界面构思和设计。

接着是大功能模块代码编写以及大概的界面模块编写。在界面模块编写之前，开发者可以在模拟器做大的功能开发。但事实上，对于iNotes开发来说，模拟器是不够用，simulator的多触点（multi-touch）支持是非常弱的，很多touch的测试是无法在simulator做的。特别值得注意的是，在功能开发的过程中要注意内存的使用，这也是在iOS开发上最最重要的思维。

然后把大概的界面和功能连接后，app的大致demo就出来了。值得一提的是，如果有界面设计师，就能节省大量时间。比如界面设计上，可以编写功能模块和设计师同步进行。这样app的demo出来后，基本上可以有界面可以用了。

在demo出来之后要自己试用和体验几遍，然后根据情况修改。

APP的0.8左右版本完成后可以加入production的图标和部分UI图片，如果没有大错误后，0.9版本可以尝试寻找beta用户。能够在产品设计和开发过程中，找一部分测试用户参与是非常有必要的。对他们的回馈的方法，可以在app发布后对这些用户发放免费产品。这个做法即可以提高产品质量，又可令到测试用户拿到免费的软件产品。

根据测试用户的反馈，重复之前demo出来后的一系列步骤。

最后在app完成后，加入app icon, iTunesArtwork等等UI元素。反复测试无错误后上传iTunes。之后大概要花7-14天来等候审批。

# App系统

主流的四大App系统：

1、苹果ios系统版本，开发语言是Objective-C；

2、微软Windows phone系统版本，开发语言是C#；

3、安卓Android系统版本，开发语言是Java；

4、塞班symbian系统版本，开发语言是C++。

# APP开发工具

**MOTODEV Studio for Android**

MOTODEV Studio for Android，这是基于Android的开发环境，为开发者们提供新的MOTODEV App Accelerator Program使他们可以开发出更适合摩托罗拉Android手机的应用程序。

**J2ME开发插件 Mobile Tools for Java**

Mobile Tools for Java (MTJ) 是Nokia公司开发的一款 Eclipse插件，用于支持 Java 手机应用程序开发。其前身就是大名鼎鼎的 EclipseME。

**NOKIA手机开发包 gnokii**

gnokii 是一个NOKIA手机开发包，可支持大多数NOKIA手机的型号。功能无比强大可以修改LOGO,收发短信，拨打/接听电话，编辑铃声。甚至还可以取到对方手机的蜂窝号（Cell ID），从而起到定位的作用。

**apk文件修改工具 Root Tools**

RootTools是一个新的工具软件，Android开发者可以在这一工具软件的支持下，对.apk格式的文件进行再次修改，让程序表现更加出色，满足用户的需求。Root Tools里面自带有很多工具，比如BusyBox，它里面集成压缩了很多Linux的工具和命令，这样软件开发者在对....

**IDEA的Android开发插件 idea-android**

idea-android 是在 IDEA 集成开发环境中开发 Android 应用程序的插件。

**Android开发工具 MOTODEV Studio**

MOTODEV Studio是摩托罗拉公司开发的 Android 应用开发工具。这是一个 Eclipse 的插件。 该插件同时也提供了 JavaME 应用的开发和 WebUI 的开发功能。

# 发展背景

2008年3月6日，苹果对外发布了针对iPhone的应用开发包（SDK），供免费下载，以便第三方应用开发人员开发针对iPhone及Touch的应用软件。这使得App开发者们从此有了直接面对用户的机会，同时也催生了国内众多APP开发商的出现。2010年，Android平台在国内手机上呈井喷态势发展，虽说Android平台的应用开发还不那么友好，但许多人仍然坚信APP开发的广阔前景。苹果公司的App store开创了手机软件业发展的新篇章，使得第三方软件的提供者参与其中的积极性空前高涨。随着智能手机越发普及、用户越发依赖手机软件商店，App开发的市场需求与发展前景也逐渐蓬勃，截止到2012年12月，App开发已变为红海市场。

# app数据

1. 真实用户数：但是获取用户是推广的第一步，然后统计用户数，要注意的是，真正获取到了新的用户，另一个非常重要的数据。

　　2、每周、每月活跃度：因为获得的用户数中有一部分以广告、并非主动进入的用户，并及时记录用户转化率、新手引导过程流失情况，而且活动度应该记录好周活跃、月活跃度。

　　3. 日留存率、周留存率：这是以保住老客户优先，先统计，日留存率、周留存率、然后在这些用户流失之前想办法提高他们的积极性。

# 意义

企业APP的意义很广泛，可以有以下几种：

**1、建立社会化销售平台**

利用网站、微博、微信、移动客户端的特点，打通社会化营销渠道，提高品牌宣传的渗透度。

**2、二维码应用**

通过二维码应用，实现从线下到线上的无缝连接。

**3、建立强大的用户数据库**

通过会员制度，实现用户行为记录分析，建立用户数据库。

**4、增强数据互通，构建通讯供应链**

实现各系统的数据互通，完善通讯供应链。

**5、建立社交平台**

例如：微信的朋友圈，陌陌的群组。

# 市场现状

APP创新性开发，始终是用户的关注焦点，而商用APP客户端的开发，更得到诸多网络大亨们的一致关注与赞许。“在传统广告、传统互联网与移动互联网融为一个整体的时候，企业和用户之间将可以非常方便地建立一个良性的闭合环：看到你了解你记住你，而这正是企业营销中最为理想的状态，也是互联网最大的价值，或将成为未来的一种新趋势，影响着越来越多的用户和企业主。”

一开始APP只是作为一种第三方应用的合作形式参与到互联网商业活动中去的，随着互联网越来越开放化，APP作为一种萌生与iphone的盈利模式开始被更多的互联网商业大亨看重，如淘宝开放平台(参考买家应用中心优秀APP:开心赚宝),腾讯的微博开发平台，百度的百度应用平台都是 APP思想的具体表现，一方面可以积聚各种不同类型的网络受众，另一方面借助APP平台获取流量，其中包括大众流量和定向流量。

　　随着智能手机和iPad等移动终端设备的普及，人们逐渐习惯了使用APP客户端上网的方式，而目前国内各大电商，均拥有了自己的APP客户端，这标志着，APP客户端的商业使用，已经逐渐普及。

# 监管政策

2012年12月10日，工信部电信经济专家委员会秘书长陈金桥在接受媒体采访时表示："工信部正在建立一个长效的评估体系，对智能手机应用程序、内置软件进行评估和抽查，而且相关的国家实验室和研究院都参与到其中。其次是要将第三方平台纳入管理，成立要备案，运行要监管。而且平台本身的运营也要有所要求，尤其对个人应用开发者要纳入管理体系，如做实名认证等。

# 解决方案

**制作交付**

APP客户端（iOS、Android）APP管理后台

　　APP下载页面（Web/Wap）APP下载对应的二维码

　　APP统计平台（VIP专享）APP宣传物料（VIP专享）

**下载服务**

　　二维码下载（名片、产品包装、官网等）

　　线上推广下载（APP Store、APP Market、百科网站）

　　其他辅助下载手段（短信链接下载、微博/微信分享下载、制定APP下载激励政策）

**运营服务**

APP运营网络环境（6个机房、8G带宽、29组集群服务器）

　　APP运营系统环境（Unix系统、MySQL数据库、SAAS及容灾备份）

　　APP升级服务（APP功能/性能升级、APP后台升级）

　　APP版本升级（适配主流的操作系统、分辨率、机型）

　　APP运营数据分析

# 现金流与品牌

对于一个新创的APP开发团队，现金流尤其重要，因此不少的开发团队都以接外包案子为生，这的确是不错的收入，同时也增加了练兵的机会。但这是快钱，就是代工产业，拼到最后就是接案数以及价格，对于团队的品牌没有多大的加分，更没有留下一个属于自己的产品。开发团队应该在APP数量以及APP品牌上面下点功夫，两条线的策略，一条是以尽可能多数量的APP来开创现金流(不一定是接外包)，另一条是开创自有品牌的APP，在这两者之间保持平衡，才有机会找到突破点。如何吸引用户经常性的愿意转发分享也必须纳入考虑，功能与创意都是一时的，用户活跃度才是关键。

# 开发技术分类

## 12.1 Native App（原生开发）

Native App（原生开发）:目前较为成熟，各大公司均采用此方式。但是其人工成本较高，同一个项目，至少需要Android端、iOS端、Web端三个开发团队。

nativeapp是一个原生程序，一般运行在机器操作系统上，有很强的交互，一般静态资源都是在本地的。浏览使用方便，体验度高。在实现上要么使用Objecttive-c和cocoaTouch Framework撰写IOS程序，要么选择java+Android Framework撰写android应用程序。

## 12.2 Web App

Web App是指基于Web的系统和应用，运行在高端手机的网络和浏览器上，用网页技术开发实现特定功能的应用，对手机性能要求比较高。

webapp是生存在浏览器里的应用，所以只能运行在浏览器里，宿主是浏览器，不再是操作系统。资源一般都在网络上。说的根本点就是一个触屏版的网站。

## 12.3 Hybrid App（混合开发）

Hybrid App（混合开发），基于第三方跨平台移动应用引擎框架进行开发。使用HTML5和JS作为开发，调用引擎封装的底层功能如照相机、传感器、通讯录等。

hybridapp是一个半原生程序，伪造了一个浏览器的apk/ipa原生程序，把地址写死了，然后里面运行了一个webapp。里面是WebView UI 。但是还是运行在机器的操作系统上，交互较弱，资源一般在本地或者网络都可以。浏览体验度次之。

拿酒来形象化这三种模式的话：Web App就像白酒那样浑厚甘醇，度数高，但不适合任何人的口味，制造成本低，Native App就是葡萄酒，分红白类型，对葡萄品质要求高，口味大众化，比较成熟，Hybrid App就是鸡尾酒，混合了多种酒的优势，非常适合跨平台，是以后的一个App开发趋势。

## 12.4 Web App、Hybrid App、Native APP对比



1. Native App（原生开发）

# 基本简介

Native App是一种基于智能手机本地操作系统如iOS、Android、WP并使用原生程式编写运行的第三方应用程序,也叫本地app。一般使用的开发语言为JAVA、C++、Objective-C。

# 概述

自iOS和Android这两个的手机操作系统发布以来，在互联网界从此就多了一个新的名词：App(意为运行在智能的移动终端设备第三方应用程序)。

开发者们都知道在高端智能手机系统中有两种应用程序：一种是基于本地（操作系统）运行的APP；一种是基于高端机的浏览器运行的Web App。

Native App因为位于平台层上方，向下访问和兼容的能力会比较好一些，可以支持在线或离线，消息推送或本地资源访问，摄像拨号功能的调取。但是由于设备碎片化，App的开发成本要高很多，维持多个版本的更新升级比较麻烦，用户的安装门槛也比较高。但是比较乐观的是，AppStore培养了一种比较好的用户付费模式，所以在Apple的生态圈里，开发者的盈利模式是一种明朗状态，其他market也在往这条路上靠拢。

## 2.1优势

1、提供最佳的用户体验，最优质的用户界面，最华丽的交互

2、针对不同平台提供不同体验

3、可节省带宽成本

4、可访问本地资源

5、盈利模式明朗

6、打开速度更快

## 2.2劣势

1、移植到不同平台上比较麻烦

2、维持多个版本的成本比较高

3、需要通过store或market的确认

4、盈利需要与第三方分成

5、开发的成本比较大

# 发展现状

根据百度2013年Q1发布的《移动互联网发展趋势报告》显示，在全行业的移动浪潮中，Native App开发者遇到前所未有的挑战。Native APP遇到的三大困境:

1、虽然用户手机里的Native APP数量在增多，但日均启动个数却在减少；

2、用户的使用时长不断向高频Native APP集中，加剧了头部效应；

3、对于低频和不知名的Native App，则面临着更严峻的“分发”和“使用”长尾困境。

这三大困境对开发者形成了较大的挑战。其中，大部分低频和不知名Native App在应用商店少人问津，无法到达用户手机。数据显示，应用商店下载量最大的前1000个（数量占比不足0.1%）App，占据了总下载次数的一半以上。这说明用户在下载应用时“马太效应”明显。困境同样也体现在应用使用率上，用户下载之后极少使用，要用时却想不起或找不到它，甚至是六成的Native APP被下载安装后的一周内并未被使用，其中又有约1/3在一个月内都未被使用，逐渐成为“僵尸应用”。

1. Web App

# 基本简介

WebApp是指基于Web的系统和应用，其作用是向广大的最终用户发布一组复杂的内容和功能。

从一个简单的帮助消费者计算汽车租借费用的网页，到为商业人员和度假者提供全套旅游服务的大型复杂的WEB站点，都是WebApp。它包括一些完整的WEB站点，WEB站点的专门功能以及在Internet、Intranet或ExtraNet上的信息处理应用。

webapp 框架是一种简单的与WSGI兼容的网络应用程序框架，可以与 App Engine 配合使用。不必为了使用 App Engine 而使用 webapp：网络服务器支持任何使用 CGI 的 Python应用程序。webapp 提供一种简单的方式来开始为 App Engine 开发应用程序。

响应式网页设计的大部分技术，是可用在WebApp开发中的。

移动端Web App和WAP有什么不同？最直接的区别就是功能层面。WAP更侧重使用网页技术在移动端做展示，包括文字、媒体文件等。而Web App更侧重“功能”，是使用网页技术实现的App。总的来说，Web App就是运行于网络和标准浏览器上，基于网页技术开发实现特定功能的应用。

# 区分

撇开各种不同的Web技术不谈，弄清楚Web Apps定义最困难的是如何从各种纷繁复杂的Web App中提取出他们的共同特征。当用户登录一个网站（如Pixlr），大家很容易理解这是在访问一个Web App。但是对那些仅仅提供基础服务（如电话查询或是信息查询）的网站，区分用户是否在访问Web App就变得相当困难了。

其实这些服务大多都是Web App。我常常这样问自己“这个程序是否完成了某个任务？”。即便它只完成了某个非常小的任务，那么它也是一个Web App。Google的搜索引擎就是一个Web App，它本质上和电话查询服务没有什么区别。

这样说来，也并非所有的网站都是Web App。如果这个网站并没有执行任何任务，那么它就并不是Web App。

# 界面

## 3.1元素变化

力求简洁明了是用户界面设计的重要原则。在同一时间给用户展示的功能越多，用户需要寻找和思考的时间也就越多。同样，界面中存在的选项越少，可用功能就越明显、越容易浏览。不过简化界面并非轻而易举，尤其是你不想减少应用程序功能的情况下。

将高级功能隐藏起来是一种有效的简化方法。搞清楚在界面中用户最经常用的是哪些功能，然后把其他功能隐藏处理。这些可由下拉式菜单和控件完成。例如，搜索栏中的高级过滤器可以做成尾部的特殊下拉菜单样式。当用户需要这些过滤器的时候只需要几次点击就可以使用。决定哪些功能保留展示哪些需要隐藏起来，并不是一个简单的工作，需要取决于功能控件的重要程度和被使用的频繁程度。

## 3.2增加阴影

弹出式菜单和窗口周边的阴影不仅仅是为了视觉美观。阴影一方面增大了菜单或窗口的尺寸，有助于将菜单或窗口从背景中区别开来；另一方面通过灰度化的边缘阴影可以屏蔽背景内容的噪音干扰。

这个技巧根植于传统桌面程序，帮助用户将注意力集中在弹出的窗口。由于很多模态窗口不容易从桌面程序内容页面中凸显出来，阴影可以使它们看起来具有立体效果、仿佛悬浮于其他内容之上，于是拉近了模态窗口与用户的距离。

为实现这样的效果，设计师往往将透明的PNG背景图片作为容器，再把内容填充到容器中，同时等距离填充弹出框各边缘。或者使用具有透明边框的背景图片，并将内容框绝对定位在其中。另外，也可以使用基于JavaScript的lightboxes命令或者CSS3中的drop shadows命令，但需要注意浏览器是否支持。

## 3.3提醒用户

当设计web app的时候，不仅需要关心一般情况下的信息展示，还要确保界面在空白状态时表现良好、具有指引作用。页面中还没有产生任何信息的时候，可以在空白区域放置一条帮助信息告诉用户如何开始。

例如，一个项目管理的应用程序主页会列出用户的项目，假如还没有什么项目信息，可以为用户提供一个项目创建页面的链接。即使这个页面上已经存在了这样一个功能按钮，一个额外的帮助并不会有什么妨碍。这个技巧可以有效地鼓励用户试用该服务，并在注册后立即进行使用。通过应用程序的单一操作步骤可以帮助用户理解这个应用的优势以及对他们是否有用。

此外，只为用户展示最重要的功能选项也很关键。一股脑的将众多功能倾泻给用户并没有什么实际意义。需要牢记的是，用户通常想从应用中获得或多或少的信息，但却不想跳进细节中，用户没有时间也没有兴趣。在空白状态中激励用户，可以显著地降低用户的流失率，并帮助潜在的用户更好的理解程序系统是如何工作的。

## 3.4 Button

许多web app拥有自定义样式的按钮。默认的输入按钮可能不适合某些情景，文字链接有时候看起来又太含蓄。需要注意的是，把链接做成Button样式的时候，它们就应该有button的表现形式。

比如，在点击button的时候它们应该会出现被“压”过的样子。这不仅仅是纯粹的视觉变化。及时反馈给用户，可以使web app感觉起来更灵敏，与桌面应用程序的用户体验更接近。可以使用CSS添加按钮的“pressed”等状态，实现处于不同状态下显示不同背景图片的功能。

## 3.5情境导航

在既定的情境下考虑用户希望看什么、需要什么是非常重要的。不需要在每一个地方都放置相同的导航控件，因为用户不是在任何情况下都需要它们。

上下文情境导航最好的一个例子就是Office 2007中，原先默认的工具栏集合被换成了带状控件形式。每一项tab控制着一组相关联的功能，如编辑图形、校对或者简单书写。Web app可以从这种上下文情境导航中获益，仅展示用户需要的、而不是所有可用的功能，从而保持用户界面的整洁清爽。

## 3.6关键功能

并不是所有的控件都拥有相同的重要性。例如创建一个新的条目，页面中会有“创建”“取消”两个button. 这里的“创建”就要更加重要些，因为这是大多数情况下用户即将要做的事情。极少的情况下用户才会去点击取消。虽然这两个控件并排放置，但是不要给予相同的重视程度。

为了将注意力引导到“创建”上，我们可以尝试使用不用的风格或样式。一种方式是将“创建”设计成button样式，“取消”设计成文字链接样式。另一种方式是在视觉上使用使用不同的颜色，并使button略有凸起的效果。这样便于抓住用户的目光。

# 优点

第一、使用W3C标准的HTML（标准通用标记语言下的一个应用）语言开发，能够轻松实现跨平台，移动应用开发者不再需要考虑复杂的底层适配和跨平台开发语言的问题。与此同时，使用HTML来开发的Web App在投入上会大大的低于传统的Native App。

第二、基于当下开始普及流行的HTML5，Web App可以实现很多原本Native App才可以实现的功能，比如LBS的功能、本地数据存储、音视频播放的功能，甚至还有调用照相机和结合GPU的硬件加速功能。

第三、移动应用的迭代周期平均不到1个月，用户需要频繁的重新下载与升级。而Web App则无需用户下载，并且和传统网站一样可以动态升级。

第四、Web App有App的特性，更有Web的特性。每一个Native App在当前的用户使用场景下是相对孤立的，而Web App则可以像传统互联网网页那样相互链接，从一个Web App直接跳转到另外一个Web App。这无论是从用户的使用体验层面还是从应用之间的数据传输来看都是非常不错的选择。

# 制作

## 5.1常用工具

ASP，PHP，HTML，JAVA等。使用这些常见的网络开发工具，可以制作绝大部分的webapp程序，使其在网站页面上实现传统的C/S架构软件的功能，也就是我们常说的SaaS模式。

## 5.2其他工具

CBX，AthTekWebAPPKit等。CBX和AthTek WebAPP Kit都是国产软件，它们的主要功能就是以将使用传统编程语言如C，C++，Delphi等开发的C/S架构客户端软件工程，以所见即所得的方式，快速部署到网络服务器上，让传统的编程语言也能够实现基于web的互联网应用程序开发。

## 5.3一键生成webapp工具

百度SiteApp是致力于为开发者、站长提供从生成WebApp到流量、用户引入再到变现的综合服务平台，亦是国内首家的WebApp在线生成服务平台。

# 开发

在iOS上开发APP，需要通过Objective-C那样精细复杂的语言去开发，这对广大的开发者而言是个不小的难题。值得庆幸的是，开发者们也可以通过开发Web APP来达到曲线救国的目的。也就是说，可以通过HTML、 CSS或者JavaScript来进行Web APP的开发。

其实Web APP说白了就是一个针对Iphone、Android优化后的web站点，它使用的技术无非就是HTML或HTML5、CSS3、JavaScript，服务端技术JAVA、PHP、ASP。

需要注意的是web app开发还是比较有限的。因为Web APP开发不能整合设备的核心功能，比如发文本信息，也不能充分使用APP Store进行销售和更新。但是Web APP开发也有其优势所在。

首先它解决了iphone APP的可扩展性问题，因为它是可以跨平台使用的。比如你开发了一款Web App，那么它既可以在手机iphone上使用，也可以在平板ipad上使用，而不像iphone APP那样只针对某个平台。

其次web APP也绕开了APP store严格的提交和更新审查规则。众所周知，随着APP store中的APP逐渐增多，APP store也推出了一系列的提交和审查规则，可谓相当之严格。而web APP则绕开了这些提交和更新审查规则，从而使得web APP的升级和维护变得更容易。因为它是一个独立的站点，而不是依附于app store上的，不管是升级还是维护在客户端进行即可，无需提交审核。

最后则是开发语言了。相比较Objective-C而言，HTML、CSS以及JavaScript学起来要容易多了。且对于传统开发者来说，HTML、CSS以及JavaScript基本都是可以直接上手的，省去了前期的培训工作。

1. Hybrid App（混合开发）

# 基本简介

Hybrid App（混合模式移动应用）是指介于web-app、native-app这两者之间的app，兼具“Native App良好用户交互体验的优势”和“Web App跨平台开发的优势”。

“云”时代的来临正在改变App和运营团队之间的关系，一场不能避免的变革正在进行。鉴于移动终端的局限性，移动终端上的APP由本地化应用(Native App)，到基于WEB的应用Web App，再到混合型应用（Hybrid APP），这一连串的变化都源于技术的更新和市场的需要[1] 。

Hybrid App是指介于web-app、native-app这两者之间的app,它虽然看上去是一个Native App，但只有一个UI WebView，里面访问的是一个Web App，比如街旁网最开始的应用就是包了个客户端的壳，其实里面是HTML5的网页，后来才推出真正的原生应用。再彻底一点的，如掌上百度和淘宝客户端Android版，走的也是Hybrid App的路线，不过掌上百度里面封装的不是WebView，而是自己的浏览内核，所以体验上更像客户端，更高效。

汽车有混合动力Hybrid，移动应用同样也有混合模式。Hybrid App（混合模式移动应用）兼具“Native App良好用户交互体验的优势”和“Web App跨平台开发的优势”。很多人不知道市场上一些主流移动应用都是基于Hybrid App的方式开发，比如国外有Facebook、国内有百度搜索等。但究竟什么是Hybrid App？如何定义？

Hybrid App：Hybrid App is a mobile application that is coded in both browser-supported language and computer language. They are available through application distribution platforms such as the Apple App Store, Google Play etc. Usually, they are downloaded from the platform to a target device, such as iPhone, Android phone or Windows Phone. The subscribers need to install to run them.

我们来拆解一下里面的含义：

1、mobile application：Hybrid App就是一个移动应用

2、both browser-supported language and computer language：同时使用网页语言与程序语言编写

3、available through application distribution platforms：通过应用商店进行分发

4、a target device：区分目标平台

5、install to run：用户需要安装使用

综合一下就是：“Hybrid App同时使用网页语言与程序语言开发，通过应用商店区分移动操作系统分发，用户需要安装使用的移动应用”。总体特性更接近Native App但是和Web App区别较大。只是因为同时使用了网页语言编码，所以开发成本和难度比Native App要小很多。因此说，Hybrid App兼具了Native App的所有优势，也兼具了Web App使用HTML5跨平台开发低成本的优势

# 兴起原因

Hybrid App的兴起是现阶段移动互联网产业的一种偶然。移动互联网的热潮刮起后，众多公司前赴后继的进入。但是很快发现移动应用的开发人员太少，所以导致疯狂的人才争夺。市场机制下移动应用开发人才的待遇扶摇直上，最终变成众多企业无法负担养一个具备跨平台开发能力的专业移动应用开发团队。而HTML5的出现让Web App露出曙光，HTML5开发移动应用的跨平台和廉价优势让众多想进入移动互联网领域的公司开始心动。可是当下基于HTML5的Web App更是雾里看花，在用户入口习惯、分发渠道和应用体验这三个核心问题没解决之前，Web App也很难得以爆发。

正是在这样是机缘巧合下，基于HTML5低成本跨平台开发优势又兼具Native App特质的Hybrid App技术杀入混战，并且很快吸引了众人的目光。大幅的降低了移动应用的开发成本，可以通过现有应用商店模式发行，在用户桌面形成独立入口等等这些，让Hybrid App成为解决移动应用开发困境不错的选择，也成为现阶段Web App的代言人。Hybrid App像刺客一样，在Native App和Web App混战之时，偶然间的在移动应用开发领域占有了一席之地。

# 分类

Hybrid App按网页语言与程序语言的混合，通常分为三种类型：**多View混合型，单View混合型，Web主体型。**

## 3.1多View混合型

即Native View和Web View独立展示，交替出现。2012年常见的Hybrid App是Native View与WebView交替的场景出现。这种应用混合逻辑相对简单。即在需要的时候，将WebView当成一个独立的View（Activity）运行起来，在WebView内完成相关的展示操作。这种移动应用主体通常是Native App，Web技术只是起到补充作用。开发难度和Native App基本相当。

## 3.2单View混合型

即在同一个View内，同时包括Native View和Web View。互相之间是覆盖（层叠）的关系。这种Hybrid App的开发成本较高，开发难度较大，但是体验较好。如百度搜索为代表的单View混合型移动应用，既可以实现充分的灵活性，又能实现较好的用户体验。

## 3.3 Web主体型

即移动应用的主体是Web View，主要以网页语言编写，穿插Native功能的Hybrid App开发类型。这种类型开发的移动应用体验相对而言存在缺陷，但整体开发难度大幅降低，并且基本可以实现跨平台。

Web主体型的移动应用用户体验的好坏，主要取决于底层中间件的交互与跨平台的能力。国外的appMobi、PhoneGap和国内的WeX5、AppCan和Rexsee都属于Web主体型移动应用中间件。其中Rexsee不支持跨平台开发。appMobi和PhoneGap除基础的底层能力更多是通过插件（Plugins）扩展的机制实现Hybrid。AppCan除了插件机制，还提供了大量的单View混合型的接口来完善和弥补Web主体型Hybrid App体验差的问题，接近Native App的体验。而WeX5则在揉合PhoneGap和Bootstrap等主流技术的基础上，对性能进一步做了深度优化，不但完全具备Native App对本地资源的调用能力，性能体验也不输原生；WeX5所开发出来的app具备完全的跨端运行能力，可以无需任何修改直接运行在各种前端环境上。

从分析可见，Hybrid App中的Web主体型只要能够解决用户体验差的问题，就可以变成最佳Hybrid App解决方案类型。



## 3.4多主体共存型（灵活型）

这是一种新型的开发模式，即支持Web主体型的应用，又支持以Native主体的应用，也支持两者混合的开发模式。比如kerkee框架，它具有跨平台、用户体验好、性能高、扩展性好、灵活性强、易维护、规范化、具有Debug环境、彻底解决跨域问题等特点。用户体验与Native App媲美。功能方面，开发者可随意扩展接口。

# 未来发展

国内外Hybrid App的开发框架众多。如何选择又成为一个难题。下面对开发者比较关心的几种知名跨平台开发移动应用中间件进行列表和对比，以便选择最适合您的移动应用中间件。

PhoneGap是相对比较早进入公众视线的一种选择。但是，开发者简单的基于PhoneGap来开发移动应用肯定会发现结果和Web App比较差的用户体验类似。这也是为什么基于PhoneGap有实用性的移动应用主要集中在iOS上。可是PhoneGap这种现状弱化了HTML5的跨平台价值。

Titanium是一种基于翻译机制的跨平台中间件，能够开发出具有Native体验的移动应用，但是因为翻译机制的限制导致移动应用开发不能像真正的HTML5开发一样灵活。哪怕一个按钮也不能像普通HTML一样来编写，而必须按照Titanium约定的特定格式。

Salama是全新研发的一套Hybrid APP和云端服务整合的开发套件。在终端，一共有三种开发模式：高度混合模式、JS模式和低度混合模式。在低度混合模式下，需要追求终端速度和显示效果的开发者，可以在不改变原有的构架思路的前提下进行开发。在JS模式下，所有的画面显示及业务逻辑均由JS程序实现，对于原来的基于WEB的开发者，只要熟悉JS、CSS、HTML的前端工程师就可以轻松构建自己的应用。在高度混合模式下，原生和HTML5可以随意组合，可以为开发者在进行大型商业软件开发的过程中提供最高的性价比。不仅如此，在云端Salama还提供了一整套云服务，涵盖了业务处理、数据库操作、文件等资源的存储分发等服务。Salama已经为多家公司提供了方案，知名客户有TOSHIBA-AIRCON、GEDORE等品牌商，同时也在Salama平台上构造了Ishow企业电子目录产品服务。

WeX5具备单View的性能特点，相比PhoneGap、Bootstrap有大幅度性能提升，同时也是以Html5、Javascript、Css3为技术基础的Web型App框架；WeX5框架按照Apache协议将所有源代码进行了开源，技术和模式上显得很开放。

Kerkee是一个多主体共存型Hybrid框架，具有跨平台、用户体验好、性能高、扩展性好、灵活性强、易维护、规范化、集成云服务、具有Debug环境、彻底解决跨域问题。

AppCan在技术架构上和PhoneGap类似是Web主体型中间件，但是通过结合了一些原生交互效果能够达到iOS、Android平台都比较一致的用户体验。与Phonegap支持单一webview使用div为单位开发移动应用不同。AppCan支持多窗口机制，让开发者可以像最传统的网页开发一样，通过页面链接的方式灵活的开发移动应用。基于这种机制，开发者可以开发出大型的移动应用，而不是只能开发简易类型的移动应用。

Hybrid App这个领域虽然还处于比较初期的阶段，但是已经有很多优秀的公司和技术团队在致力于跨平台开发移动应用中间件技术的研究，给了开发者众多选择。开发者可以根据实际的项目需求来选择中间件。Web App虽被浏览器厂商和搜索引擎公司所推崇，但存在用户体验差、盈利模式不明确等现阶段无法解决的问题，或最终夭折。Hybrid App正在被越来越多的公司和开发者所认同，势必会成为新世界的王。



# hybrid app开发工具

## 5.1 AppCan

AppCan是国内Hybrid App混合模式开发的倡导者，AppCan应用引擎支持Hybrid App的开发和运行。并且着重解决了基于HTML5的移动应用"不流畅"和"体验差"的问题。使用AppCan应用引擎提供的Native交互能力，可以让HTML5开发的移动应用基本接近Native App的体验。

AppCan作为中国Hybrid混合应用开发、移动平台、移动云平台的倡导者和领导者，以“免费+开源+开放”的互联网模式，为广大开发者提供一站式的移动应用开发支持服务。[4] 与此同时，从移动应用开发、管理、运营、安全四个方面，为各级政府和企事业单位，构建运营一体化的企业移动平台，企业通过个性化的移动运营门户，增强客户服务品质，提升整体经营管理水平。

现在，正益移动AppCan行业解决方案已成功应用于金融、航空、政府、石化、传媒等领域，客户包括东方航空、国家电网、中化集团、泰康人寿、新华社等众多大型企业，赢得了市场广泛认可，是国内企业移动信息化领域的龙头企业。

## 5.2 Appcelerator

Appcelerator的Titanium开发平台使开发者可以通过HTML、PHP、JavaScript、Ruby、Python等Web编程语言开发手机、平板和桌面的原生App。其优势在于它可以让用户轻松地访问超过300个API以及定位信息。

此外，Appcelerator提供针对特定行为或事件定制的统计。App的数据既可储存在云端，也可储存在设备上。

## 5.3 Kerkee

Kerkee是一个多主体共存型Hybrid框架，具有跨平台、用户体验好、性能高、扩展性好、灵活性强、易维护、规范化、集成云服务、具有Debug环境、彻底解决跨域问题。[2]

从开发者角度来说，它支持三种的团队开发模式：

**针对Web开发者：**

这种模式其中的一个场景是：只会Web开发，却不会Native开发的开发者提供了一系列的平台型接口。这种方式具有开发周期短，跨平台等优点。

**针对Native开发者** ：

这种开发模式的其中一个场景是：Native开发者想要截获Web页面的数据或者对数据进行自己的处理，或者Web页面中的行为进行修改。在这个时候，Kerkee框架将会为他们带来便利。

**针对Web开发者和Native团队共同合作的开发团队**

对于这种模式的团队，kerkee框架具体更开放更透明的协作，并且严格地隔离各自职责。各得Web团队和Native团队把主要精力定位到各自的模块上，有利于各自的模块优化到极致。

## 5.4 WeX5

WeX5采用混合应用（hybrid app）开发模式, UI体系完全基于w3c的html5+css3+js；引入jquery和bootstrap并对移动做了底层优化，效率和性能接近原生应用。WeX5本机API Framework采用phonegap(cordova)框架。

## 5.5 APICloud

APICloud是一款“云端一体”的移动开发平台，信仰“云端一体”的理念，重新定义了移动应用开发。APICloud为开发者从“云”和“端”两个方向提供API，简化移动应用开发技术，让移动应用的开发周期从一个月缩短到7天。APICloud由“云API”和“端API”两部分组成，可以帮助开发者快速实现移动应用的开发、测试、发布、管理和运营的全生命周期管理。

## 5.6 PhoneGap

PhoneGap是一个免费且开源的开发环境，使开发者可以开发出在Android、Palm、黑莓、iPhone、iTouch及iPad等设备上运行的App。其使用的是HTML和JavaScript等标准的Web开发语言。开发者使用PhoneGap进行开发，可调用加速计、GPS/定位、照相机、声音等功能。

　　PhoneGap还提供Adobe AIR App以及在线的培训课程，帮助开发者了解原生API并在他们自己的平台上开发移动App。

## 5.7 NativeScript

NativeScript是使用移动平台的 JavaScript 引擎来进行跨平台开发。逻辑部分自然无需多说，关键在于如何使用平台特性。NativeScript是通过反射得到所有平台 API，预编译它们，然后将这些 API 注入到 JavaScript 运行环境，接下来在 Javascript 调用后拦截这个调用，并运行 native 代码。NativeScript是使用大量 web 开发的技巧来进行 app 开发，因为工具链和语言都非常熟悉受到了很多前端开发者的欢迎。

## 5.8 Kinvey

Kinvey同样是一个为移动应用开发者提供后台创建服务的平台。Kinvey强调加速移动应用开发与销售的“即取即用”理念。Kinvey的中间层与数据层均托管在多个云服务提供商处，包括 Rackspace、Amazon与Microsoft。所有通过Kinvey存储的数据都会有四种方式备份：Amazon EC2、Windows Azure、Rackspace以及Kinvey自己的服务器，假如其中一两个出现了故障，用户的数据依然安然无恙。

## 5.9 ExMobi

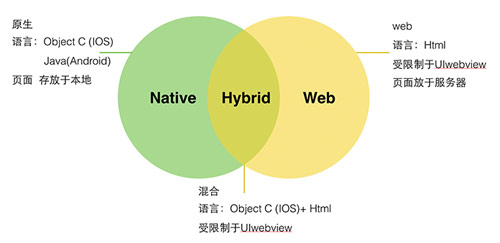
ExMobi通过全面的数据集成技术和丰富的跨平台客户端展现能力，将业务系统快速、安全、高效的移植于移动终端。ExMobi从开发（IDE环境）、集成（IT系统对接、云服务）、打包（各个操作系统的应用打包）、发布（应用的运行）、管理（日志管理，更新管理）上提供了一套完整的解决方案。并通过专业的培训和支撑渠道为开发者提供可持续的学习和交流空间，扫除开发障碍。

1. Web App、Hybrid App与Native App的设计差异

# 引言

这3类主流应用你都了解吗？设计师除了要有视觉功夫，对不同形式的APP也应当了然于胸，百度的同学写了一篇非常全面的总结，帮你迅速搞定3类主流APP的设计方法，附带一大波避雷针，带你巧妙跳过APP设计的雷区，涨姿势是分分钟刻不容缓的事咯！

目前主流应用程序大体分为三类：Web App、Hybrid App、 Native App。



# Web App、Hybrid App、Native App 纵向对比

首先，我们来看看什么是 Web App、Hybrid App、 Native App。

**1. Web APP**

Web App 指采用Html5语言写出的App，不需要下载安装。类似于现在所说的轻应用。生存在浏览器中的应用，基本上可以说是触屏版的网页应用。

优点：

（1）开发成本低，

（2）更新快，

（3）更新无需通知用户，不需要手动升级，

（4）能够跨多个平台和终端。

缺点：

（1）临时性的入口

（2）无法获取系统级别的通知，提醒，动效等等

（3）用户留存率低

（4）设计受限制诸多

（5）体验较差

**2. Hybrid App**

Hybrid APP指的是半原生半Web的混合类App。需要下载安装，看上去类似Native App，但只有很少的UI Web View，访问的内容是 Web 。

例如Store里的新闻类APP，视频类APP普遍采取的是Native的框架，Web的内容。

Hybrid App 极力去打造类似于Native App 的体验，但仍受限于技术，网速，等等很多因素。尚不完美。

**3. Native App**

Native APP 指的是原生程序，一般依托于操作系统，有很强的交互，是一个完整的App，可拓展性强。需要用户下载安装使用。

优点：

（1）打造完美的用户体验

（2）性能稳定

（3）操作速度快，上手流畅

（4）访问本地资源（通讯录，相册）

（5）设计出色的动效，转场，

（6）拥有系统级别的贴心通知或提醒

（7）用户留存率高

缺点：

（1）分发成本高（不同平台有不同的开发语言和界面适配）

（2）维护成本高（例如一款App已更新至V5版本，但仍有用户在使用V2， V3， V4版本，需要更多的开发人员维护之前的版本）

（3）更新缓慢，根据不同平台，提交–审核–上线 等等不同的流程，需要经过的流程较复杂

# Web App、Hybrid App、Native App 技术特性



由上图可见，Web APP 的开发基于Html5语言。而Html5语言本身又有着不可避免的局限性。正是这些局限性的存在，使得Web App在体验中要逊于Native App。

# Web App受限制因素及设计要点



相比Native App，Web App体验中受限于以上5个因素：网络环境，渲染性能，平台特性，受限于浏览器，系统限制。

**1. 网络环境，渲染性能**

Web APP对网络环境的依赖性较大，因为Web APP中的H5页面，当用户使用时，去服务器请求显示页面。如果此时用户恰巧遇到网速慢，网络不稳定等其他环境时，用户请求页面的效率大打折扣，在用户使用中会出现不流畅，断断续续的不良感受。同时，H5技术自身渲染性能较弱：对复杂的图形样式，多样的动效，自定义字体等的支持性不强。

因此，基于网络环境和渲染性能的影响，在设计H5页面时，应注意以下几点：

简化不重要的动画/动效

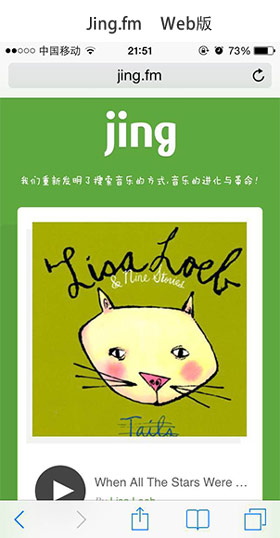
简化复杂的图形文字样式

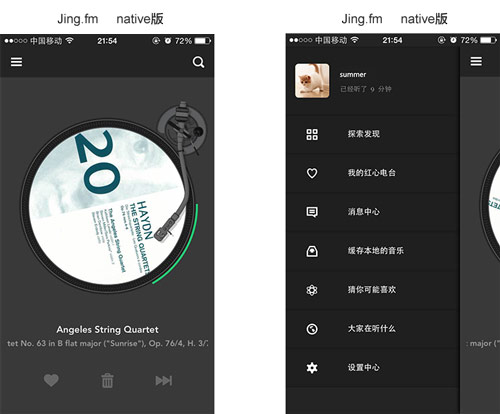
减少页面渲染的频率和次数

从下图移动Web版 jing.fm和Native版jing对比后可以看出：Web APP首页去除冗余的功能，回溯本源，只给用户提供了jing.fm最初的本质需求——电台。既符合H5精简功能又达到了突出核心功能的设计原则。无疑给用户眼前一亮的气息。

正如书中《瞬间之美》的一个核心观点：重要的并不是我们提供的信息量有多大，而是我们能否给他们提供真正需要的信息。

用户是否真的在意准确的信息？这篇文章给你另一个全新的视角！《[重磅推荐！如何设计传单才不会被人随手扔掉？](http://www.uisdc.com/design-fascinating-leaflet)》





再如：百度推出的直达号，以良子健身为例：

从Native App和Web App（百度直达号）的对比中，我们可以看出Native良子以九宫格的形式展现，且属于双重导航，功能入口太多；弊端是用户不知道聚焦在哪里，分散用户的注意力。而Web版良子整合并减少了导航的入口，增强用户的专注度；界面清爽，整洁，很好地传达了良子本身的寓意——轻松、愉悦、休闲、舒适。



**2. 受限于浏览器**

通常Web App生存于浏览器里，宿主是浏览器。不同的浏览器自身的属性不尽相同，如：浏览器自带的手势，页面切换方式，链接跳转方式，版本兼容问题等等。

例如下图：UC 浏览器和百度浏览器自身支持手势切换页面。手指从左侧滑动页面，返回至上一级。百度手机助手H5页面，顶部Banner支持手势左右滑动切换。这一操作与浏览器自身手势是冲突的。



再如，基于浏览器的Web APP在打开新的模块中的页面时，大多会新开窗口来展现。例如用户在使用购物类APP时，浏览每日精选模块时，每当打开新的商品时，默认新开一个窗口。这样的优劣势显而易见：优势是能够记录用户浏览过的痕迹，浏览过的商品，以便后续横向对比；劣势是过多的页面容易使用户迷失在页面中。

正如Google开发手册里描述：当用户打开一个Web App的时候，他们期待这个应用就像是一个单个应用，而不是一系列网页的结合。然而，什么情况下需要跳转页面，什么情况下在当前页面展示则需要设计师细致考量。



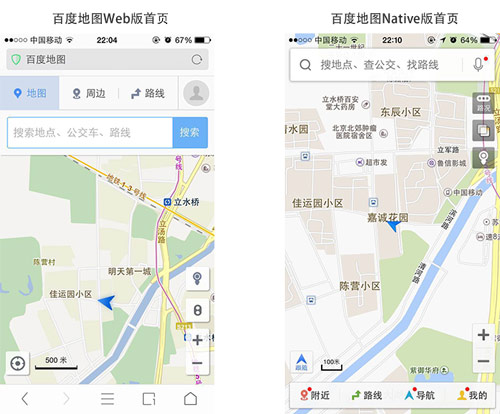
因此，Web App基于浏览器的特性，从设计角度应该遵循以下了两点：

少用手势，避免与浏览器手势冲突。

减少页面跳转次数，尽量在当前页面显示。

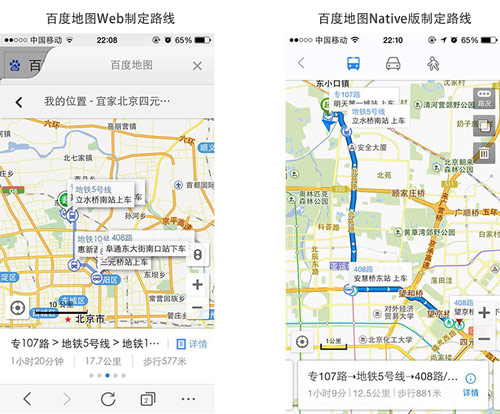
**3. 系统限制，平台特性**

由于Html5语言的技术特性，无法调用系统级别的权限。例如，系统级别的弹窗，系统级别的通知，地理信息，通讯录，语音等等。且与系统的兼容性也会存在一些问题。以上限制通常导致APP的拓展性不强，体验相对较差。例如百度地图：



Web版地图基于浏览器展现，因此，不能全屏显示地图，给用户的眼界带来局限感；相反，Native 版地图以全屏展现的形式，很好的拓展了用户的视野。整个界面干净简洁，首页去除冗余功能。

在制定路线的体验中，如图：



Web 版地图耗费的流量大于Native版，且不能预先缓存离线地图。对于地理位置的判断也是基于宿主浏览器，而非Web地图本身。获取路线后，对于更换到达方式，相对来说是不便利的。

相反，Native 版地图，能够直接访问用户的地理位置，能够很清晰的为用户展现App规划的路线，并能轻松的查看多种路线方案，以便做出符合自己的最佳方案。对于切换公交，走路，自驾等路线方式也是只需一键操作。

Native 版地图相对于 Web版地图增加更多情感化，易用的功能，如：能够记录用户的生活轨迹，记录用户的点滴足迹，能够享受躲避拥堵方案等。而Web版地图基于技术框架，很难实现以上功能，从用户体验角度来看，弱于Native版地图。

# 小结

综述所述，在设计Web APP时，应当遵循以下几点：

**1. 简化**

简化不重要的动画/动效

简化复杂的图形文字样式

**2. 少用**

少用手势，避免与浏览器手势冲突

少用弹窗

**3. 减少**

减少页面内容

减少控件数量

减少页面跳转次数，尽量在当前页面显示

**4. 增强**

增强Loading时的趣味性

增强页面主次关系

增强控件复用性