



移动应用开发技术手册

移动应用开发技术手册

移动应用开发是为小型、无线计算设备编写软件的流程和程序的集合，像智能手机或者平板电脑。移动应用开发类似于 Web 应用开发，起源于更为传统的软件开发。但关键的不同在于移动应用通常利用一个具体移动设备提供的独特性能编写软件。例如，利用 iPhone 的加速器编写游戏应用。在这本技术手册中，我们将着重为您介绍企业级移动应用开发的基本内容以及云端移动应用如何开发。

资源和战略

移动设备开发主要通过运行应用的硬件和设备的其他系统约束区别于桌面和 Web 应用开发，这些约束从平台到平台有微小的变化。同样的，下面的这些内容有助于应用开发设计者和开发者学习一些最主流的平台的相关资源。

- ❖ [移动开发资源指南](#)
- ❖ [移动应用服务战略制定：本地型移动开发](#)
- ❖ [移动应用服务战略制定：Web 型 混合型和中间件移动开发](#)

一步一步构建移动应用

你一定听过平台即服务（PaaS）或者是基于云的服务，如 Google Apps Engine、CloudBees、Heruko、Engine Yard 和 Cloud Foundry。这些优秀的平台都可以轻松地对你部署的应用进行流程管理、维护和平台的管理。大多数倾向于托管主机和应用程序的管理端。但是，云服务如何构建一个应用程序，还有移动应用程序呢？

- ❖ 手把手教你用云服务构建移动应用
- ❖ 移动应用界面：搞定用户三板斧
- ❖ 移动开发：敏捷方法更胜一筹

移动开发作用

现在来评估移动设备开发对于总和开发实践产生的最终影响还为时过早。但是应用大的观念转型正在根据输入和地理性能而发生变化，其他的改变也可能随之而来。

- ❖ 移动应用促使程序设计改变
- ❖ 追踪 Android：移动平台发展记实录

移动开发资源指南

移动设备开发主要通过运行应用的硬件和设备的其他系统约束区别于桌面和 Web 应用开发，这些约束从平台到平台有微小的变化。同样的，这本指南关注有助于应用开发设计者和开发者学习一些最主流的平台的相关资源。

苹果 iOS

苹果移动设备是消费性电子产品市场的巨人，包括 iPhone、iPod Touch 和 iPad。尽管第一代设备达不到很多人的需求，苹果产品的易用性和乔布斯的个人魅力仍旧吸引了很多狂热追随者。iPad 尤其展示了企业的承诺，其革新的用户界面已经影响了新应用设计。

[苹果官方 iOS 开发者网站](#)允许开发者下载 SDK 并开始开发和测试新应用。对于[应用商店](#)分配新应用也有相关信息。免费的 iOS SDK 包括 Xcode IDE、Open GL ES 支持的 iPhone 模拟器、Interface Builder、Instruments、框架、编译器和 Shark 分析工具。

查看一下致力于[苹果开发](#)的博客和论坛也许是个不错的想法。他们在 Ajaxian.com 对于 iPhone 开发有很多不错的点子。

Android 操作系统

Android 是基于 Linux 的移动操作系统，由谷歌和开源手机联盟拥有。[Android 开发者网站](#)提供了 Android SDK 以及 Android 平台上开发移动应用的各种工具。SDK 旨在为使用 Eclipse 的人设计，但是这个 SDK 包括调试、封装以及在模拟器安装应用的其它一些工具。

过去一年 William Brogden 给予我们很多这方面的建议，包括[Android 架构](#)，[Android 移动平台开发](#)和[Android 以及 Kindle 开发的比较](#)。

Windows Mobile 和 Windows Phone 7

当谈论到移动设备的时候，来自微软的两个主要操作系统是 [Windows Mobile 6](#) 和更新一点的 Windows Phone 7。已经习惯在 Windows 桌面工作的开发者将会发现它们在年过渡到手机上的时候有很多类似的地方。根据 [Windows Mobile 开发者中心](#)，Windows 移动应用的大多数开发协同使用 Visual Studio 和 Windows Mobile SDK。

正如其他的主要移动开发平台一样，Windows 为开发者提供集中发布解决方案。[Windows 的应用中心](#) 为 Windows Phone 操作系统上创建和分配应用提供必要资源。此外，[Windows Phone 博客](#) 提供来自 Windows 移动开发专家的新闻和评论。

其他移动设备平台

其他移动平台包括黑莓应用平台，以及塞班开源平台。黑莓平台支持 Java 和基于 Web 的应用。[黑莓开发者地带](#) 提供应用设计者感兴趣的黑莓系列设备的搜索资源，包括新的 [黑莓平板电脑](#)。

塞班开源平台将自己吹捧为全球使用最广泛的智能手机平台。该平台出自诺基亚的 s40 和 s60 平台。现在提供的大多数代码位于 Eclipse Public License (EPL) 之下，尽管其他一些代码尚未公开，仍旧受制于私有 Symbian Foundation License (SFL)。[塞班开发者网站](#) 提供了更多的源码信息、工具和套件、文档以及其他塞班平台的信息。如果你对于塞班开发感兴趣，可能也想看看 [诺基亚的 Qt SDK](#)。

移动应用服务战略制定：本地型移动开发

受到极具人气的苹果 iPhone 的影响，[移动应用开发](#)已经进入新阶段。设备和应用的能力已经远远超越了第一代的移动开发杀手应用——电子邮件。好戏还在后头。在销售方面，智能移动设备已经赶超传统的计算机客户端。此外，据 Forrester Research 计算，2010 年全球智能手机和平板电脑上的付费应用所得收入为 22 亿美元，预期至 2015 年的复合年增长率（CAGR）为 82%。

尽管移动应用开发仍然主要是做消费应用，但有迹象表明企业开发团队需要做好准备，以便让他们的应用运行在新的、更为智能移动设备上。各类应用开发经理及软件架构师都注意到了。

[RESTful SOA 服务](#)经常倾向于成为[移动应用集成](#)中服务器端架构的一部分。在细节上，移动应用的服务器架构与已有的三层架构不一样，但是并没有像移动客户端架构的差异那样显著。有大量的设备，其行为跟典型的桌面或便携式 PC 不同。取决于你的选择，它们得益于或者说是受累于匆忙的创新步伐。

为了驯服这头猛虎，开发领导需要迅速定义出一个既实际又灵活的策略。IT 必须确保移动应用快速投入市场，同时还要考虑到移动工具和平台那令人炫目的变化节奏，Forrester Research 的分析师 Jeffrey Hammond 如是说道。

Hammond 强调了目前出现的若干移动应用的主要范畴。理解这些东西是迈向实现移动应用开发策略的第一步。

他列出的基本类别是：本地类型，直接在移动设备上运行；基于 Web 型，利用了设备的 Web 浏览器；本地和 Web 混合型；以及第三方移动中间件服务。

每一种类型的移动开发方案 IT 决策者都面临着多种选择，每一项决定都会受到移动客户端异于企业传统的桌面或笔记本客户端这一事实的影响。

总的来看，Forrester 的 Hammond 把成本问题、应用创建和升级的敏捷性、性能以及用户体验视为决定移动应用开发方式的关键因素。正如我们将会看到的，这些选择并不互斥。如 Hammond 所述，移动应用开发“是一门平衡的艺术”。

本地型[移动开发](#)

在 Web 浏览器登录智能手机之前，编写设备的本地操作系统是应用的开发方式。这意味着开发人员不得不针对每一种制造商的设备学会一组独特的工具和技巧。

本地型开发采用低级软件语言，相对于主流的企业开发而言，这仍然类似于嵌入式系统的开发方式。但是移动设备已经发生了演变。突破性的 iPhone 允许团队里有高级语言—Objective C 进行开发，尽管这一语言尚未得到主流开发者队伍的广泛支持。Google 则坚定地将其 Android 移动开发平台置于流行的 Java 语言基础上。

但是，对于每一种采用 [Android](#) 的移动设备，开发团队仍然面临着特定的工作。这是因为屏幕大小、内存补充以及其他元素在不同的 Android 之间均各不相同。

经理的困境进一步恶化，因为 Android 和 iPhone 加起来也仅占移动设备市场的一小部分。为了支持大量的终端用户，移植到许多设备类型上也许是必要的。除了 iPhone 和 Android 以外，还有 Windows Phone、苹果的 iPad、Rim BlackBerry 以及 Symbian 等。

“如果你关注的重点在于提供一个很棒的体验，想要进行优化，那么你有可能会选择本地化，” Hammond 在 Forrester 最近于波士顿举办的 ADD 论坛上说。但是，这不是老生常谈。Hammond 补充说某些移动中间件提供商能够提供实现高度优化的用户体验的应用。

提示：在考虑本地性移动应用作为战略路径时，你尤其应当权衡一下性能和用户体验的重要性—要问一下这些特性对于你的应用集成的成功来说究竟有多重要。

移动应用服务战略制定：Web 型 混合型和中间件移动开发

Web 型移动开发

Web 浏览器在移动设备上的出现受到了欢迎，尤其是当 [HTML5](#) 面世时。HTML5 拓展了用于前端开发的 JavaScript 的使用增长，为了更好地迎合移动需求，相关的框架，如 JQuery 和 Dojo 也得到了增强。许多开发经理有可能被 Web 型移动开发吸引，因为基于 PC 的 Web 浏览器有助于规范客户端。但是，正如移动开发困境的大部分情况一样，这并没有那么的直截了当。

“如果你主要是对尽可能低地保持成本或尽可能快地升级应用感兴趣……那么你需要看一看基于 Web 的方案，” Hammond 说。

本地型移动开发好的一面是这个：这是与智能手机的神奇品质接轨的最好方式。本地型应用能够开发 GPS 等其他设备能力，且工作迅速。

据 Hammond 说，尽管 HTML 5 存在优势，但是 Web 型的方案也有不利的一面。如果你的应用需要离线支持，以及先进的富媒体及高性能图形渲染的支持，或者如果它需要发挥尖端平台特性的优势的话。

混合型和中间件[移动开发](#)

Web 型和本地型方案不是互斥的一对。开发经理可以两边下注避免损失：根据情况，创建基于组合集成、利用了每一种方案的优势的混合型的应用。最终的应用可使用 Web 方案进行，比如说内容交付，然后用本地型方案实现途中的地理定位能力。

据 Hammond 说，混合型方案尤其适合于高并发或性能敏感型应用的 B2E、B2C 解决方案。

此外，这家 IT 商店不需要单干。[中间件服务](#)提供商能够接过全部或部分的工作量，提供一个以云为基础的中间件集成层，这样团队可以针对各种设备类型进行编程，也能够把应用部署到各种设备上。

专家 Hammond 指出，类似于混合型移动开发方案，移动中间件也可包含有本地和 Web 两种类型。通常是以软件即服务（SaaS）或平台即服务（PaaS）的形式出现，移动中间件平台一般由致力于移动应用交付的供应商提供支持。同样地，他们也提供客户可用于开发的中间件和 API。其中间件曾意在“抽象”掉独立设备类型的细节。

预期消息中间件服务供应商会跟进设备及设备软件的发布，以及新的设备能力（语音识别、地理定位等等），也会跟进最新的 HTML 5 库及浏览器改进。在分析服务的时候，从总体上确定供应商所掌握的资源范围是很重要的。

提示：寻找架构中那些能够分化为 Web 或本地部署的点。如果你要选择第三方消息中间件，需仔细打量其所提供的 API，这些 API 应当以最小的调整就能应用到已有的服务架构上。你需要考虑尝试将自己的架构接入供应商[消息中间件平台](#)的后果。

手把手教你用云服务构建移动应用

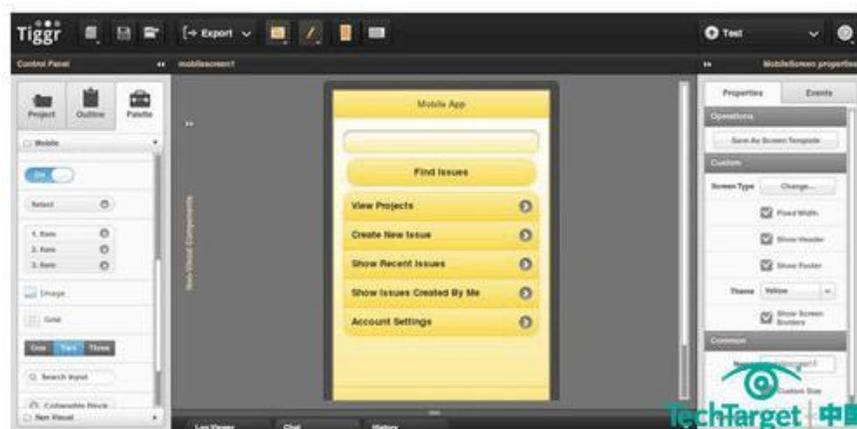
你一定听过平台即服务 ([PaaS](#)) 或者是基于云的服务，如 Google Apps Engine、CloudBees、Heruko、Engine Yard 和 Cloud Foundry。这些优秀的平台都可以轻松地对你部署的应用进行流程管理、维护和管理。大多数倾向于托管主机和应用程序的管理端。但是，[云服务](#)如何构建一个应用程序，还有移动应用程序呢？如一个已经存在的服务。基于云的服务，我们称它 Tigr 移动应用程序构造器。

[Tigr](#) 是什么？

Tigr 是基于移动应用程序的构建器。它使你能够构建移动 web 和本地应用程序。Tigr 使用 PhoneGap 库可以得到一个本地应用程序。Tigr 移动应用程序构造器的优点是不用下载、安装、配置或者设置。在 Tigr 中建立一个新的移动应用程序使用的时间比传统方法少。Tigr 在配置项目的各种库时为开发者节省了不少的时间。

用 [jQuery Mobile](#) 构建移动界面

Tigr 使用 jQuery Mobile 组件：这是一个跨平台和跨设备的移动组件。因为你已经从屏幕截图上看到，左侧移动调色板含有所有组件，你可以非常容易的把它们拖到手机上。

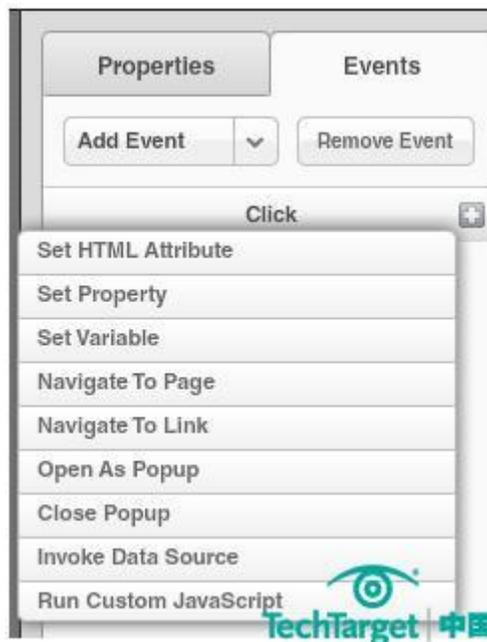


你可以重新排列组件或者是设置属性。强大的 Grid 组件可以简化构建复杂应用程序的布局。一个应用程序肯定会有很多界面。

添加 HTML 事件和行为

你可以在 Tigr 中构建真实的移动应用。为了它能够构建真实的移动应用，我们需要能够定义 HTML 事件，如点击和值的变化，为事件添加一个调用行为。这个行为可以导航到另一个界面、打开一个弹出窗口、设置 JavaScript 变量、或者是引用客户的 JavaScript。

看一下定义 HTML 事件：



引用客户的 JavaScript 行为是一个强大的特性，因为它允许你在 Tigr 初始化参数之外编写你的应用程序。你可以编写并引用客户的 JavaScript 代码，引用 PhoneGap API 访问设备的特性，或者是导入其他第三方类库，并在你的应用程序中使用。例如，这里有一个使用 PhoneGap API 的例子，使手机震动 2 秒：

以下是引用片段：

```
navigator.notification.vibrate(2000);
```

定义 [REST 服务](#) 并在屏幕中使用

什么是没有数据的应用？你可以在 Tigr 中定义并使用任何可用的 REST 服务。建立一个 REST 服务通常要设置服务的 URL，然后定义服务的输入和输出。例如，点击 URL，<http://search.twitter.com/search.json?q=html5>，可以看到从 Twitter 的搜索 REST 服务的结果。在这个例子中，输入是 q，输出是你在 Web 浏览器上看到的内容。

指定 REST 服务的 URL：

The screenshot shows a 'REST Service Properties' dialog box with the following fields:

- Name: REST Service
- URL: https://mongolab.com/api/1/databases/tiggrdemo/collections/sysl
- Method: get
- Data Type: json
- Security Context: Select Security Context
- Use Tigr Proxy:

指定 REST 服务的输入：

REST Service Properties

Request Parameters

[Empty text box]

Name	Default value
name	
api	

TechTarget | 中国

指定 REST 服务的输出:

REST Service Properties

Request Parameters

Response Parameters

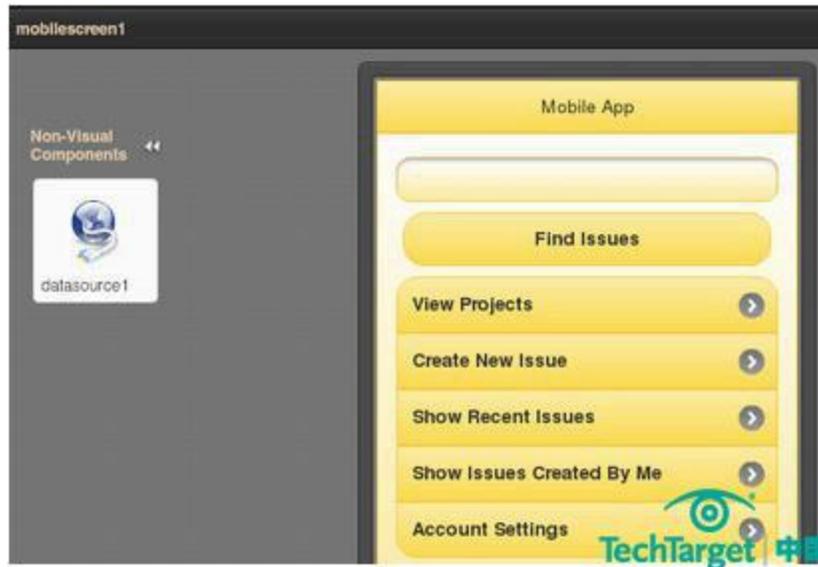
[Empty text box]

Name	Array	Add level
key	<input type="checkbox"/>	+
ns	<input type="checkbox"/>	+
v	<input type="checkbox"/>	+
name	<input type="checkbox"/>	

TechTarget | 中国

服务定义完之后，可以在屏幕中使用它，特殊的数据资源组件必须关联屏幕。数据资源组件关联完后（通过拖、拽到屏幕上），它用来指定界面输入组件的映射，为服务提供

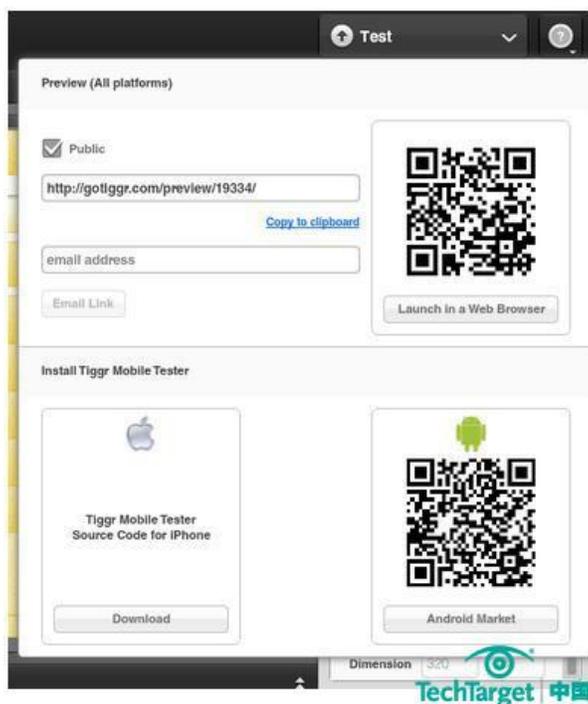
输入数据。当服务执行完后，数据资源用来指定结果的映射，并作为输出返回的截面组件上。



测试应用程序

这是 [Tiggr](#) 最具创新特性之一。在应用程序的开发期间，可以在浏览器和移动设备上测试它。启动后，第一个组件出现在屏幕上，点击测试按钮，就会在默认的浏览器上打开应用程序。你就可以看到应用程序的样子了。

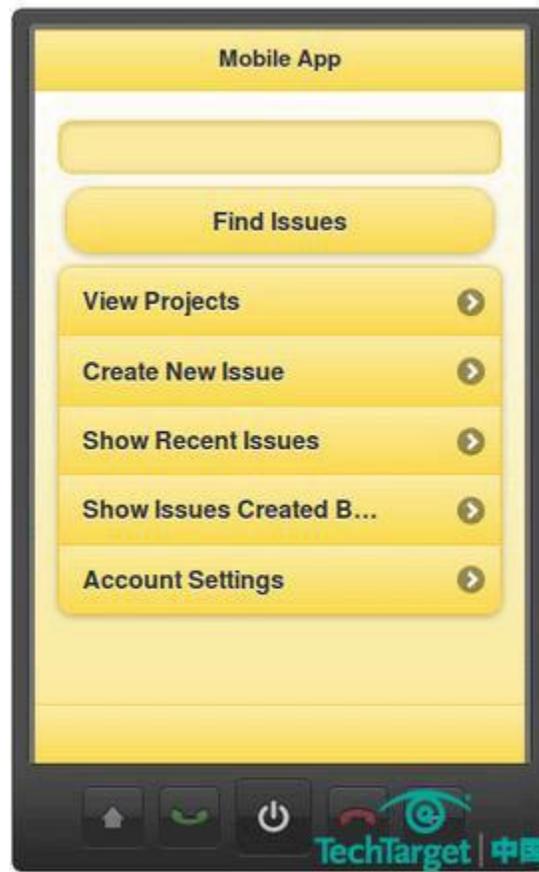
这是 Tiggr 中的测试窗口：



在桌面浏览器中测试肯定是很伟大的事情，但是，那不是应用程序运行的地方，对吗？你想在移动设备的浏览器上测试它。为了在移动设备上得到应用程序，使应用程序是公开的。这时，你会有一些选项：

- 发送应用程序的 URL 到移动设备
- 扫描 QA 码（用 QA 扫描仪）
- 在移动浏览器中输入 URL

下面是在桌面移动浏览器测试的结果：



测试本地应用

你会认为在 web 浏览器（桌面或移动）上测试一切都很好，但是，仅仅是构建移动 web 应用。我们怎样能看到，使用 PhoneGap API 访问设备时发生了什么呢？在 web 浏览器测试是不会显示的。

Tigr 移动测试仪应用（Android 和 iOS）旨在移动设备上简化本地应用的测试。在设备上安装后，输入 Tigr 帐号，你的移动应用程序会显示出来。只需点击一个应用，本地的应用程序就会启动。

下面是 Tigr Mobile 测试仪的结果：

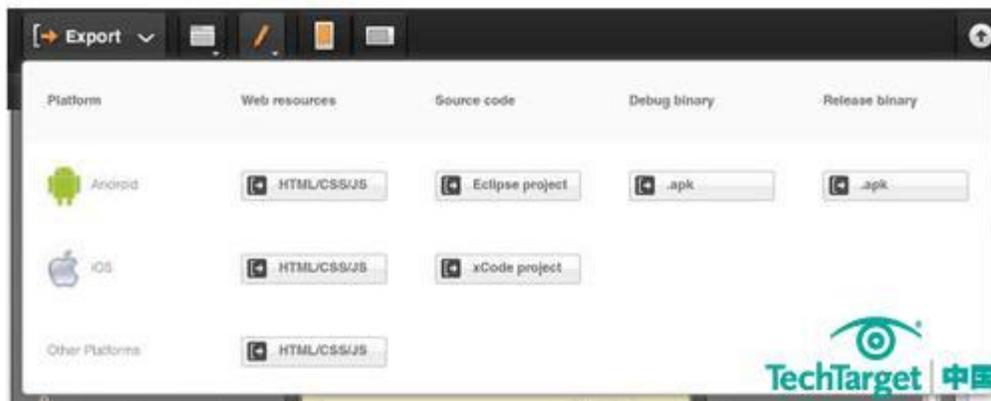


这是一个很棒的测试本地应用程序的方法。你不需要每次都安装应用程序，你需轻按，就可以得到最新的本地应用程序的版本。

导出应用程序

完成应用后，你可以导出应用为移动 Web (HTML、JavaScript、CSS) 或者是本地。本地选项，你可以把每一个 Android 或 iOS 源码和 PhoneGap 的配置导入到 Eclipse 项目。这可以让你在 Tigr 外继续工作。Android 选项，你可以生成发布包 (.apk 文件)，并可以发布到 Android Market。

需要一个 BlackBerry 应用？导出为 HTML/JavaScript/CSS，使用 PhoneGap Build，为 BlackBerry 或者其他 PhoneGap 支持的平台，构建一个本地应用。



[社交应用程序](#)的开发

在 Tigr 中有内置的社交的特性。不仅可以共享测试应用的链接，还可以邀请与你工作和协作的人。用户可以同一时间或不同时间工作在应用程序上。也有内置的聊天特性。基本上，你可以得到 Google Docs 的移动原型。现在看来，这是很酷的事情。

接下来做什么？

如果你今天已经构建了一个[移动应用](#)程序，注册 Tigr，并构建了你的应用程序。这是一个超级简化的构建移动应用的方法。我确信你会喜欢它，甚至还有一些乐趣。

移动应用界面：搞定用户三板斧

根据 [Gomez 2009 年和 2011 年所进行的调查](#)显示，用户对移动应用的期待正变得越来越挑剔。用户现在期待移动软件服务交付的性能水平不断增长，包括页面的快速加载和更高的可靠性。除了对移动应用的需求更多，终端用户转投竞争对手的可能性也越来越大。如果你将要为你的公司开发一个移动应用，它必须不辜负用户的期望，否则他们可能不会持续太久。

Compuware 公司负责 Gomez APM 移动项目策略的移动项目经理 Amir Rosenberg 给出了三个妙招，可以用于企业应用的[移动应用](#)和[移动界面开发](#)。这些建议特别适合于企业移动应用，这和个人移动应用略有区别：

1. *准确辨识关键用例，确保应用程序能够完整，快速，可靠地完成这一任务。企业用户不需要花哨的功能，他们只想要一个工具能够帮他们更好或更快的做完他们的工作。*
2. *调研一下你的用户将会使用什么样的硬件，他们公司会给他们提供多少培训，以及这些用户是否能够和多大程度上访问网络。然后利用这些信息，提供一个在上述情景中性能较好并且可靠的用户界面。*
3. *根据实际终端用户的反馈来了解实际的运行性能，并据此进行持续改进。使用自动化的性能测量工具来监视这些应用的负载情况，同时也要将用户的反应考虑在内。*

抓住关键用例

根据 Rosenberg 所述，与普通个人用户用的 WEB 应用相比，为企业用户设计的移动应用，其用例通常相对较少。个人用户 WEB 应用设计人员会通过令人印象深刻的动画，动听的声音以及众多的新奇体验来让用户发出“哇”的惊叹声。而企业应用则相反，它需要关注在一点：把工作做好。确保你的移动服务能够迅速，简单和可靠地把工作做完。

Rosenberg 说，你应该尽量让你的用户在 3 次（或更少）鼠标操作之内就能找到他需要找的东西。

举例来说，他提到了通用汽车公司的销售人员使用一个特别设计的移动应用在停车场上来给潜在的客户提供报价，而不需要将客户带到库房里去。他们的移动应用使用一个条形码读码器来为某种车提供所需的报价，而不是一个能够访问包括车、制造商、车型、特性、可选项和价格等等大量信息数据库的移动 WEB 应用。

销售人员只需要开启扫描器（点击一次鼠标），扫描停车场上汽车的条形码（第二次点击鼠标），报价信息很快就显示在屏幕上。根据 Rosenberg 的说法，通用汽车在这一应用上显示除了极高的判断力，因为他们集中关注在这一个销售人员实际在使用的用例，并且努力将其实现地快速、可靠，没有牵涉进任何多余的特性或可能影响性能的数据集。

根据硬件进行剪裁

Rosenberg 解释道，企业应用实际上在很多方面要比个人用户应用的设计要简单得多。企业应用的设计人员的一个优势是他们通常可以准确地定位谁是他们的用户，他们使用什么样的硬件，他们受培训的情况如何，以及他们可以在何种程度上访问网络。而商业应用，往往在开发时对上述因素一知半解。

商业应用供应商可能会需要适应 iPhone，采用不同设计的 Android 手机、黑莓、平板电脑及其他移动设备，而企业应用的架构师可以节省大量的精力，只为企业提供了（或在他们的 IT 政策的规定）的特定型号的设备开发，其屏幕大小、硬件接口、内存限制和其他可用的硬件都是已知的。

并且，[企业架构师](#)也能够知道他们的用户有多少经验，以及这些用户会受到多少培训。如果用户未来每天都在使用这一应用，并且会得到足够的培训去掌握如何使用，那么你就不必过于担心图形界面的直观性和自解释特性。在这种情况下，也没有很大必要去在屏幕上包含如何使用每项特性的操作指南。另一方面，如果你的用户通常都在办公室里工

作，一年之中只有偶尔机会才会使用这个移动应用，那么在开发中最好多花一些时间，确保用户能够拿起来就用，而无需每次都要琢磨半天。

Rosenberg 说需要考虑的第三个因素是用户的上网类型。如果用户只是使用地板上的设备、扩展坞、展览室或其他公司内的建筑物，那么你可能会确定他们将使用无线 WiFi。在这种情况下，你可以并且可能应该在设备和服务器之间经常性地传输数据以减少加载时间。Lowe's 有限公司采用这种方式，他们用 iPhone 应用取代了他们零售店里面的条形码扫描器。

相反，如果用户有可能会在旅行中使用这个应用，那么你可能会希望将所有的数据一次性下载下来。这样，即使在用户连接信号不好的时候，用户也可以使用已经下载的数据进行工作，然后在他们再次到达一个具有可靠的网络连接的地点时再去上载新的数据。Dow Jones 在他们的行政计划应用中采用的就是这种方式。

持续改进性能

非常重要的一点是，要经常去检查你的应用，以确保他们的性能达到预期的要求，并能持续改进。Rosenberg 指出，这对于移动应用尤为重要，因为用户的期望在不断发生变化，并且硬件和网络联接也在不断改进。一个好的移动应用需要做到与时俱进。

Rosenberg 解释说，性能更新应该是一个不断迭代的过程，并由用户输入的信息所驱动。很重要的一点是使用自动化测量来主动的发现和解决服务故障和服务性能缓慢，但 Rosenberg 说，这还不够。这些客观数据只能告诉你这个移动应用的工作状况如何，但这并非全部。你应该从真实的用户那里搜集数据并据此来进行改进，这样进行的改进才会更有效地改善用户的日常使用。

移动开发：敏捷方法更胜一筹

[移动应用](#)（app）可以下载、使用，甚至是被用户丢弃的形式都不同于用户同传统应用交互的方式。敏捷开发方法恰好适用于移动应用开发。下面将提供敏捷开发实践可以加强移动应用开发的七种方法：

[敏捷开发](#)适用于实验和改版类型的移动应用

移动应用在经历过很多次的实验和改版之后才能成功。改善流程以及接触用户可能每天都在发生，因此应用就出自那些在应用商店以及应用上发布评论以及评级的客户。敏捷完美适应移动应用开发固有的这种迭代元素需求。

敏捷增加可靠性并导致应用持续使用

移动应用用户对于移动应用的错误和崩溃的忍耐度要比非移动应用世界的用户少很多。一个应用首次较短时间的崩溃，足以导致用户删除这个应用。在应用商店有成百上千的其他替代应用，他们有太多选择了。敏捷开发可以实现测试和质量多重周期保障，应用开发者可以通过对每一次冲刺进行重复周期，从而测试构建更好的质量和可靠性。

敏捷冲刺自然地扩展了移动应用的更新模型

如果应用有新版本，大部分情况下，会在应用商店显示更新。在开发周期的最初阶段，只有少数测试用户可以下载，安装和使用这个应用，但是下载和安装应用类似于其未来运行模式，一旦其测试完成，将发布给所有用户。正式版本发布给最终用户之前，将会形成一种很好的冲刺和发布节奏。

敏捷促进[移动应用开发](#)更快响应技术变革

移动技术，尤其是移动操作系统技术，改变、经历功能增强以及比非移动操作系统接受更为快的新性能更新。敏捷虫子以及发布频率让企业可以快速响应这些技术改变，新的增强可以在没有什么影响的情况下推出更新，甚至是每一天，如果需要的话。唯一的约束是这些用户可能会讨厌过于频繁的更新。

敏捷移动开发快速适应客户反馈

当敏捷冲刺和大多数移动应用软件更新模型联合时，客户反馈的新性能和功能增强可以迅速开展。在非敏捷、非移动应用中，更新版本很痛苦，要确保应用用老数据来运作。在敏捷移动开发中，具有持续、平稳的应用更新流程，确保应用更新向下可以同旧数据运作。如果不能，将会在新的更新开始时调试。

[敏捷移动开发](#)更关注用户体验

移动应用运行在有限的环境中，应用大小有限制，要边下载边安装，或者稍后更新。如果应用花费多余两分钟的时间来下载，用户可能会删除正在进行的下载。此外，对于应用数据可以下载多少也有限制。敏捷开发允许开发者在随后的冲刺中用这些选择实验，调整应用设计和性能，让用户体验更快、更平稳和无缝。

敏捷移动开发有助于推出特性集

独立应用就像游戏中的单一玩家需要简洁，通常不会和任何后端服务器往复沟通。但是，很多严肃的客户应用需要这个后端服务器可以定期访问（天气相关的应用、航班应用显示飞行位置、旅行网站等）。他们受到很多有意义的性能的约束，任何应用都不能超载过多的性能。聪明的应用开发者在一个应用已经拥有太多性能太大的时，生成附加应用。敏捷移动开发协助企业调整其设计，如果需要，在随后的冲刺中可以缩减性能。

总结

移动应用不同于其他应用。他们需要下载或者更新的更快更流畅。他们需要在需要的时候同后端服务器无缝交互。而这种情况只有通过数次改变以及调整才能实现。敏捷开发加强了质量保证以及多重测试循环，提供了移动应用这些必需的内容。

移动应用促使程序设计改变

现在来评估移动设备开发对于总和开发实践产生的最终影响还为时过早。但是应用大的观念转型正在根据输入和地理性能而发生变化，其他的改变也可能随之而来。这些改变之一可能是搜索技术的普遍影响的减少，“问答系统”技术的增加。数学软件拓荒者 Stephen Wolfram 如是说道。

由于手机容量有限，输入是很难处理的问题。用户界面设计者尽力而为，但是问题仍旧很明显，没有开发者会无视它。一些权威人士建议新的语音识别输入可能是小型设备难题的一种答案，但是这种解决方案在其令人失望的结果之前就已经提出过了。

无线和基于 Web 的服务仍旧是小型设备生命线，但是开发变化也潜伏于此。新设备的内存、带宽和连通性要求服务开发者关注其消息决策的大小和时间。最近也我们参加“Xconomy Mobile Madness 2011 论坛”会议也证明了领域更深层的变化迹象。

会上，演讲者 Stephen Wolfram 认为搜索技术很少影响“无线网络”。相反，Wolfram 看到移动开发成为他所支持的“问答系统”技术生长的温床。最近的 IBM Watson 系统 Jeopardy 电视机智问答节目的获胜者表示这种技术和软件有直接的相似点。

“移动是一个你必须尽快切入正题的媒介，” Wolfram 讲到。

“你不必滚动并点击[多重]链接。”因此，他对于其他也看到搜索技术使用的减少，新电话拥有更多用户的人表示赞同。

Wolfram Research 创始人兼 CEO Stephen Wolfram 正是有影响的数学软件计算软件平台的缔造者。科学哲学上将他描述为影响深远且具有争议的未来学者。

他和他的公司近年来创造了 Wolfram|Alpha，这是一个类搜索界面的 Web 网站，通过访问专门的计算知识引擎回答问题。

这个软件和一个大型的、定期更新的、十万亿以上级别（10-trillion-piece-plus）的数据库、语言分析和动态算法以及方程式相结合。Wolfram|Alpha 应用已经可以在 iPad、iPhone、Android 以及其他设备上使用。

“这么多年我们的业务一直是桌面业务，然后是服务器端业务，再后来是云计算业务。之后就会是 Wolfram|Alpha，移动应用，” Wolfram 讲到。

Wolfram|Alpha 移动应用的直接目的是为在校学生量身定做主要课程，Wolfram 讲到。例如该公司提供代数学、微积分音乐理论课程助教，学生轻敲（以手指等）该公司的知识引擎进入即可。

“应用世界的战略是针对不同的工作有不同的应用，” Wolfram 表示，他说未来的应用将会为医生、系统管理员和很多其他类型的工作者服务。

移动计算和传统平台之间也出现了微妙的不同。传统开发你必须提前指定功能。

“现在你可以通过搜索引擎或者数据库获取知识，为你所处理的功能类型创建软件，” 他最后讲到。

追踪 Android: 移动平台发展纪实录

目前，移动设备上的 Android 操作系统能够迅速闯入当前操作系统领域是个很好的机会，该领域一直由像台式机上的 Windows 这样的操作系统占有。一方面，它是面向服务的。而且，随着人们花费大量时间用 Android 联机，他们将更加快速的推动发布周期，并不断的增加功能。Android 给开发者提供工具来满足他们的需要。

Android 是由 Google 发起的，移动设备上的开源操作系统/平台。它的特点是，用 Java 来编写应用程序，并可以设计安全预防措施，允许多个不同的应用程序，在不暴露用户信息的情况下，运行在相同的设备上。同样值得注意的是，Android 旨在支持越来越多的硬件功能。这个操作系统提出一个对硬件和应用开发者非常有利的环境，因为它是开源的，Android 的低级内核是基于 Linux 的。

一年前，当我第一次注意 Android 平台的开发，我下载的版本是 1.5r2。新版本的更新比我能回想起来的其他任何技术都要频繁，也许你想问，如何能让开发者跟得上呢？

下面的表格显示的是 API 的版本和与之相关的数字。此处显示的版本号是 2010 年 7 月 SDK 版本号。API 的版本是必不可少的，因为在 Android 应用程序包里，开发者必须指定所设计的 API 版本。API 5 和 API 6 不在更新，因为使用版本 2 的人都将移至到 2.1 版本（使用 API 7），但是 API 3 和 4 会继续更新，用来介绍软件开发包（SDK）工具中各项改善措施和漏洞的修正。

版本	第一次发布	更新	最近市场使用率
1.5 API 3 rev 4	2009 年 4 月	2010 年 5 月	18.9%
1.6 API 4 rev 3	2009 年 9 月	2010 年 5 月	22.1%
2.0 API 5	2009 年 10 月	到 2.01	微量
2.0.1 API 6	2009 年 12 月	到 2.1	微量

2.1 API 7 rev 2	2010 年 1 月	2010 年 5 月	55.5%
2.2 API 8 rev 2	2010 年 5 月	2010 年 7 月	3.3%

Android 市场提供的统计数据，人们使用每个版本设备的指示器。下面的表格包括到 2010 年 7 月前两周的数据。显然，旧版本手机仍然构成巨大的市场。几乎所有 2.0 和 2.01 版本都迅速升级到 2.1。我非常惊讶看到 2.1 已经有这么大的市场。

API 版本和特性

幸运的是，它并不是基础语言，它正在改变，并一个接着一个的发布新版本。相反，Java 类的增加以适应对新的硬件和版本的支持以及支持用新的功能在应用程序中进行创建。只有少数与用户界面相关的 API 被改变了，几乎没有类被废除。一般来说，用 1.5 版本写的应用程序能被 2.2 版本兼容。

下面是从 1.1 版本开始的重大变化（2009 年 2 月）

1.5 API 3 用户界面设计工具、音频和视频录制和回放的媒介处理的多重改进。支持语音识别。SDK 模拟器的屏幕分辨率提高到 320x480。

1.6 API 4 支持多种屏幕分辨率。图形和动画的改进，支持手势识别。从文本到语音的合成。短消息服务（SMS）的支持。SDK 模拟器的屏幕分辨率提高到 480x854。

2.0 API 5 此次重大升级添加许多新标准的应用和特性。能同时追踪多达 3 个屏幕的触摸。支持 Microsoft Exchange 邮件，与 API 联系，共享应用程序间的交往信息。图形的性能和虚拟键盘的支持得到了改良。支持蓝牙连接。

2.01 API 6 修复 2.0 版本的主要漏洞。

2.1 API 7 修复 2.0.1 版本的主要漏洞。

2.2 API 8 主要是对 Java 引擎速度的提升，提高了 2 到 5 倍。浏览器中 JavaScript 的性能也得到了极大的改良。浏览器改良了 HTML5，支持视频流，可以安装 Flash 播放器插件。与 Microsoft Exchange 的融合也得到了改良。

版本 3—— 我们看到了对下一个预计将在第四季度的主要版本的各种猜测，但是一直没有正式公布。

开发者的生态体系

你从开发者网站下载的 Android SDK 包仅包含一部分 SDK 的环境。最重要是部分是开始下载的 SDK Manager，它可以让你更新你需要的正确组件的工具。例如，如果你只希望开发 2.1 版本（API 7），这就是你需要下载的。支持每个 API 的原文件储存在自己的目录结构，所以，如果你稍后决定安装 2.2 版本，它不会干扰 2.1 版本的工具和文档。

鉴于发布数字的迅速变化，保持文档更新是至关重要的。Google 似乎做的很好，包括文档下载的更新，网上相互参照的网页和开发者的博客。