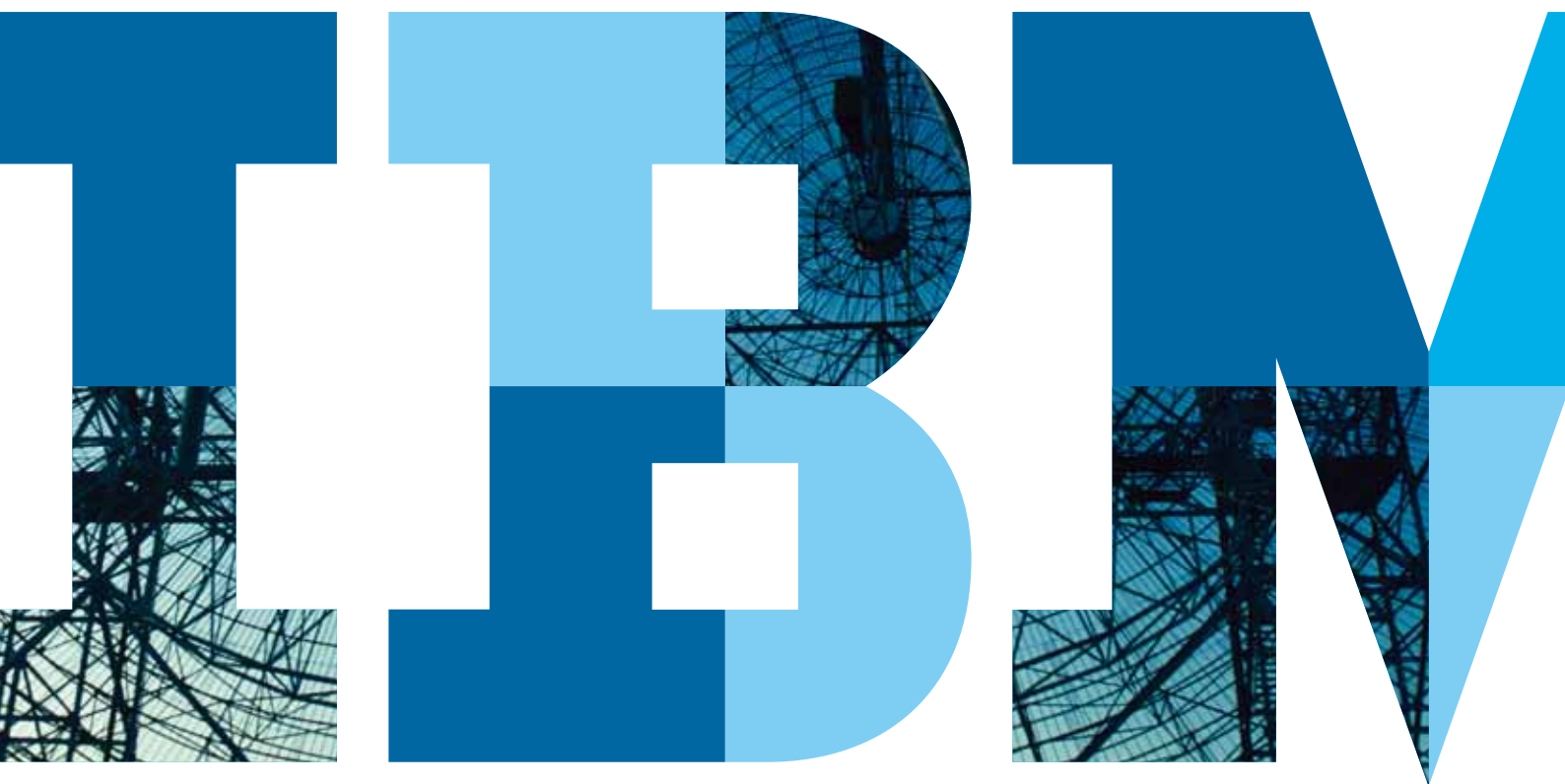


# 移动应用开发入门

企业团队开发移动应用项目指南



## 执行摘要

各行各业纷纷开始认识到,其业务应用的目标受众已经从使用传统的个人计算机(如台式机和笔记本电脑)大规模转向使用移动设备(如智能手机和平板电脑)来访问 Internet 并获取所需的信息。适用于应用的预期受众是企业(企业对消费者应用,简称为 B2C)的直接客户,以及目标用户是员工或业务合作伙伴(企业对员工和企业对企业应用,分别简称为 B2E 和 B2B)。纵观全球,现在越来越多的用户使用移动设备作为通过 Internet 获取信息和请求服务的主要途径,因为无论去哪里都能随身携带移动设备,并且移动设备更加直观易用。

这种最终用户行为的重大转变不断激励企业为现有的企业应用开发移动渠道,同时计划推出各种新型应用,从而充分利用市场上现有移动设备的各种独有特性。与信息技术行业的所有主要演进一样,在这次转变的前几年,人们疯狂地采取各种手段来满足需求,在不计战略后果(如应用开发成本、可维护性、质量和安全性)的前提下建立市场地位。随着移动应用市场的成熟,最初的市场潮流趋于平稳,负责制定长期规划和经济发展的企业员工将这些更为全面的软件开发问题呈现到人们的视野之中。

IBM 在众多行业企业中因其产品而享有盛誉,是严谨负责的软件开发合作伙伴。IBM 全球服务事业部近期发表了一篇文章,题为“为您的移动企业制定有效的应用战略”,<sup>1</sup>其中概括介绍了 IBM 提供的移动应用规划、开发、部署和管理建议。本文仅对这篇范围宽

泛的文章中一个主题进行了说明,重点介绍一种开发移动业务应用的全面方法。其中描述的这项技术完美融合了协作软件生命周期管理最佳实践与创建移动应用的最新独特要求。本文的内容旨在为移动企业业务应用开发项目涉及的所有角色创造价值,包括规划移动项目的架构师、制定各种实施决策的开发团队、制定项目活动细节的项目经理、处理这些全新应用的测试组织,以及需要了解这些全新移动应用与现有企业应用和开发流程的匹配度的主管人员。

## 移动应用开发所面临的独特挑战

创建用于在智能手机和平板电脑等最新移动设备上运行的应用要面临各种独特的要求和挑战。

### 形状和用户输入技术

移动应用首要也是最明显的特征就是显示屏形状,用户交互方式也与先前的软件形式大相径庭。智能手机通常只能提供 4 英寸的区域来显示应用内容,并且屏幕分辨率像素密度比个人计算机(PC)的显示屏要小得多,个人计算机的发展趋势是显示屏更大且屏幕像素数更高。即使是平板电脑,显示屏通常也小于 PC,与客户使用的最新桌面 PC 的大平板显示器相比显得尤为小巧。

外形较小意味着向最终用户显示的数据量较少,因此这些应用的数据布局需要不同于在 PC 设备上运行的应用。每次可显示的数据量

会大大减少,所以必须显示“适当”的数据,即用户使用该应用时所需的最相关数据。

移动应用面临的另一个明显的实质性差别是:用户输入机制的不同。移动设备已率先采用了无键盘的“手势”作为一种常用且有效的用户输入方法。必须计划并支持触摸、轻扫和捏合手势,这样才能满足移动应用用户体验的要求。经证实,这些触觉型的最终用户输入机制非常受欢迎,并且现已被广泛改造为在传统的桌面 PC 系统上使用,如 Apple “Lion” OS X 版本和 Windows 8 “Metro” 操作系统。除了触感用户输入机制,移动设备还是语音用户输入的自然目标。事实上,传统的键盘用户输入形式可能是效率最低、也是最不受移动应用用户欢迎的输入交付机制。

除了直接从最终用户接收输入内容,移动设备还能够接受其他来源的输入内容,如设备 GPS 组件的地理位置输入信息,以及通常内置于设备的照相机中的图像信息。移动应用设计和开发期间,必须将这些独特的输入形式考虑在内。相较于输入形式有限的应用而言,它们提供了全新且宝贵的机制,让移动应用更加强大和高效。

## 可用性和用户交互设计

有多个因素促使企业需要更加关注移动应用的可用性和用户交互设计。一个是设备大小和用户输入方法的差异。计划如何仅显示必要的信息比简单显示所有可能的数据,并让最终用户直观地筛选自

己所需的信息要困难得多,也要耗时得多。一种书面比喻方法是,编写简明精炼的摘要比创作整篇文章更加困难。移动应用设计人员必须考虑屏幕的实际大小。当应用需要显示更大范围的数据且数据包含多层详细的信息时,最好使用循序渐进的发现方法,让用户逐步“深入”专门介绍少量特定项目的细节信息。

移动设备上提供的丰富多样的输入方法对早期设计工作而言是一种动力,让人们认识到并使用更加高效的方法提供输入数据,而不是采用传统 Web 和 PC 应用的默认“在表单中键入数据”的设计模式。必须避免在移动应用中使用大量的键盘键入形式,这样可以降低最终用户的挫折感,在触摸键盘极小且缺乏传统键入反馈功能的情况下尤其需要注意。确定使用无键盘模式后,在这种情况下,收集信息并将其提供给移动应用是一个重大的设计挑战。

另一个需要格外注意但很微妙的推动因素是移动应用设计工作。最终用户与移动设备以及移动设备上运行的应用之间的交互方式,同他们与固定 PC 乃至平板电脑之间的交互方式截然不同。移动设备的最终用户通常将设备握在手中,同时还要与周围的物理环境进行交互。这些应用的用户往往不能长时间集中精力来操作移动应用,因为他们还需要留意周围的物理环境。移动应用用户的交互模式是短暂的,常被打断且具“突发性”,也就是说,他们需要在注意力转移到他处之前快速完成应用任务。

所有这些因素促使企业在开发项目的初期就加大对以用户为中心的移动应用的设计投资。理想情况下,应该将这些可用性考虑因素和

设计方面编入移动应用需求,然后链接到后一阶段的开发交付成果,以及用于验证用户交互和应用的“可使用性”是否能尽量让人满意的测试阶段。

### 实现技术的选择

市场中的移动应用实施方案选择有很多。关于如何选择移动应用的实施方法,目前还没有完美的答案,该领域内的所有方案都有各自的优缺点。因此,移动开发团队面临一项艰难的挑战:权衡各项技术,根据特定的应用需求做出选择。先前提及的 IBM 全球服务事业部文章对实施方案的选择进行了简要描述,并提供了比较表。本文将就此提供几点补充。

移动项目实施技术的选择将对其他应用开发相关决策造成一定的影响。它可能会限定开发工具的选择范围。实施方案的选择很可能会影响团队的各种角色和结构,可能会影响应用的测试和验证方法,以及向最终用户分发和交付应用的方法。因此,为移动应用选择实施方法至关重要,制定早期阶段的决策时应十分谨慎。

### 原生应用实现

“原生”实现意味着使用特定类型设备上的移动操作系统所提供的编程语言和编程接口来编写应用。

例如,iPhone 原生实现将使用 Apple 提供且支持的 Objective-C 语言和 iOS 操作系统应用编程接口 (API) 来编写应用。

原生应用实现的优点在于,可让移动设备达到最高保真度。由于 API 在较低的层级被调用且特定于应用的专用设备,所以应用可以充分利用该设备提供的各项功能和服务。

原生的移动应用实现完全不可移植到任何其他移动操作系统上。如果要让本机 Apple iOS 应用在 Android 设备上运行,则必须全部重写应用。因此这种移动业务应用实施方案异常昂贵。

### Web 应用

最新的智能手机和平板电脑往往预先安装了高级 Web 浏览器,这十分适于实施标准 Web 应用形式的移动业务应用,并且通过提供特殊的样式表,以适应移动设备的大小以及近似的移动设备“外观”。采用这种方法实施的移动应用支持的移动设备类型最广,因为支持 JavaScript 和 HTML5 的 Web 浏览器相当一致。所提供的几种 Web 2.0 小组件商业开源库有助于实现这种方法。对于已经为 web 应用开发进行了开发人员语言和技术培训的企业而言,针对移动应用实施的 Web 编程模型具有一定的优势。

纯粹的 Web 应用实现方法的缺点在于,此类应用无法访问直接在移动设备上运行的功能和特性(如照相机、联系人列表等)。

但是,如果您的移动应用不依赖于设备上运行的本地服务,单纯采用 Web 应用方法或许就足够了。随着 HTML5 规范逐渐成熟并日益获得移动 Web 浏览器的广泛支持,可通过该 W3C 编程标准在纯粹的 Web 应用上提供许多移动设备本地服务。

另一个值得思考的 Web 应用和原生应用差别在于, 应用分发方式以及在设备上提供应用的方式。原生应用必须从某种“应用商店”(如允许公开访问的 Apple iTunes 商店或 Google Android Marketplace) 进行下载和安装。这种应用商店分发机制的优点在于, 允许使用搜索算法轻松查找移动应用。企业还可以不时关注移动应用商店产生的市场知名度信息和最终用户反馈。

应用商店(特别是公共应用商店)的缺点在于, 它们直接处于企业及其预期目标受众之间。所有移动应用和更新必须全部由应用商店分发, 因而难以远程控制和管理已通过应用商店机制提供的移动应用。Web 应用分发不经过应用商店。最终用户只需将应用的 Web 地址输入移动 Web 浏览器, 即可通过 Internet 提供该应用。更新移动 Web 应用就像更新服务器或应用托管服务器一样轻松。任何用户下次访问该网站时, 都会将新版的移动应用下载到设备中。

### 混合移动应用实施

混合移动应用实施方法是纯粹的原生实施和纯粹的 Web 实施之间的一种妥协。

运用行业标准 Web 编程语言和技术(如 HTML5 和 JavaScript) 编写移动应用。但是, 将应用打包成原生安装格式, 并通过应用商店机制进行分发。

混合应用链接到其他本机库, 这些本机库允许该应用通过单个应用代码基访问本机设备功能。由于混合应用的大部分结构采用的技术实施方法并非特定于任何单一设备, 所以该应用的绝大部分代码是可移植的, 并且能在多个不同的移动操作系统之间重用。不过, 小型本机代码段也可以同混合应用相集成。因此, 开发人员可决定有多少应用实施是通用的共享代码基, 有多少是设备特定的自定义内容。

您还可以选择将多少代码打包为“原生”可安装应用并通过应用商店来交付, 以及多少代码可通过网络下载。您可以打包要显示的首批应用元素, 以便在设备上直接进行安装, 这样, 用户启动该应用时能够迅速加载内容。其他更为动态的元素则可构建为网页, 在服务器上进行管理, 这样用户访问时始终可提供最新版本的应用。

对于普通的移动业务应用, 不少行业分析师坚信, 从长远的角度来看, 代码重用的经济性和灵活的应用开发必将青睐折中的混合方法。

## 移动应用构建与交付

由于受到将移动应用快速投放到市场的强劲业务动力的驱使, 移动开发项目的时限往往极为紧迫。从项目伊始到交付的整个周期仅有数月时间, 这种做法十分常见。移动应用交付压力很快迫使人们对大部分移动项目采用敏捷开发方法。

敏捷开发实践的重要元素是持续集成和构建。开发人员做出的应用变更需要即时处理,并且必须对要运行该应用的所有移动操作系统进行一一处理。如果该移动应用采用混合或原生实现,每次开发人员交付应用变更集时,都需要构建该应用。每种受支持的移动环境的构建设置和配置都不相同,很像是一个小型的构建服务器“场”,需要人们对其进行配备并且可用,从而处理针对多个操作系统的移动应用构建工作。

## 测试

促使移动应用开发面临巨大挑战的另一个领域是测试。移动应用测试代表着在复杂度和成本方面比传统应用有了重大的飞跃。与传统PC和Web应用不同,可能支持的移动设备和版本级别范围非常庞大。移动项目测试表动辄包含数百乃至数千种设备、移动操作系统级别、网络运营商、区域设置和设备方向组合的情况相当普遍。

人们将越来越多的变量抛入与其他类型的软件毫不相关的移动测试方程。连接不同的运营商网络时,同一型号的设备功能可能会略有不同。此外,网络连接的质量对于移动应用的表现也会产生深远的影响。即便是移动设备本身的移动也可能会成为影响应用表现的重要因素,因为某些应用专为设备移动而设计。

绝大部分移动应用具有多层的架构,代码在设备上运行,该设备本身作为传统中间层和数据中心“后端”提供的数据和服务的“前端”客

户端。全面且有效的移动应用测试要求各应用层面面面俱到,而不只是测试移动设备上的代码。中间层和后端服务测试版的安装以及可用性,可能会为移动应用测试带来极高的成本和复杂性挑战。

许多移动项目首先会使用手动测试方法。这是一种快速开始测试的最显著方式。您必须购买该应用支持的各种移动设备,付费雇人(更可能是雇佣团队)不厌其烦地按照已编写好的脚本说明(其中介绍了针对这些设备上的每个应用构建版执行的测试)执行测试。这种手动测试十分昂贵,但效率极低。但是,通过提供一种机制来获取应用关键可用性的反馈信息,手动测试确实能发挥巨大的作用。

除了实际购买移动设备,您还可以依靠移动设备模拟器和仿真器来执行测试。借助这种方法,在桌面工作站上运行的软件程序能够代替实际的设备来完成测试。运用模拟器和仿真器开展移动应用测试对于开发人员单元测试等任务十分有效。一些设备仿真器十分出色,但也有一些仿真器不太尽如人意,无法有效地模拟实际的设备。因此,无论是手动测试还是自动测试,在实际的移动设备上执行某种形式的测试往往十分必要。

有些移动应用测试解决方案依赖于在设备上运行的代理程序,可用自动执行的方式与测试脚本进行交互。

这种方法能够灵活地运用实际物理设备或仿真器开展测试,并可通



过自动化技术提高效率。但是，测试组织需要承担设置被测设备以及在這些设备上安装测试代理的费用。

应对这项移动应用测试挑战的另一个途径是运用所谓的“设备云”。云可以提供实际的移动设备资源，而不是通用计算机资源。您无需“租用” Linux 虚拟机执行长达数小时乃至数天的测试，而是可以租用特定型号和版本的移动设备。这种方法可为企业节省购买数百台设备以及使用这些设备执行测试的成本。

## 移动开发与其他软件开发有哪些相似之处？

尽管移动应用的开发具有很多独到之处，但整个开发生命周期涉及的许多角色和任务与其他类型软件的企业级开发完全相同。在“敏捷性和架构完整性评估”一文中，2 Walker Royce 介绍了使用敏捷和测试优先原则有效交付软件的主要技术和实践。Walker 在文中介绍的软件交付实践堪称移动开发项目的最佳选择。

## 完整的项目生命周期

软件开发项目的生命周期模式基本类似，无论用户创建哪类软件都是如此。首先需要根据某些分析做出业务决策，决定投资开发应用。捕获并描述应用要求。这些应用要求将被进一步分解为多个用户案例和特色工作项，然后将其组合成可迭代使用和发布的工作计划，在交付应用时要完成该计划。

为担任不同角色的团队成员分配工作项，并使用各种工具完成工作，将工作成果交付到项目之中。测试最终应用并进行验证，确保它能满足各项要求。特定公司软件项目所遵循的确切流程和生命周期通常进行了量身定制保证能够达到特定的企业目标并符合相关策略的要求。

移动应用也遵循同样的生命周期。移动应用开发一般具有以下特色：由小型团队实施、采用现有基础架构，交付用户交互性非常出色的应用。敏捷方法和测试优先原则非常适合此类项目。尽管移动应用开发的某些特定要求很可能与某些其他软件开发项目有所区别，但上述要求的收集、描述和沟通工具与流程完全一致。两者都需要将要求与代码变更紧密关联，从而实现这些要求，这一点在移动应用和其他软件之间并无二致。换言之，移动软件开发项目与其他软件开发项目的项目流程以及项目的集成性和可追溯性需求完全相同。

## 集成多种工具

即便有，也只有极少的软件开发项目只通过一种开发工具就能成功交付。绝大多数项目均涉及不同供应商提供的大量工具，旨在满足整个项目生命周期的特定角色或任务需求。

例如，移动应用代码开发人员或许会发现某款简单的代码构建工具能够满足自己的需求，而这款工具恰好与即将执行该应用的移动平台完全匹配。但是，这款工具漏掉了可促进协作和协调的功能，无法在敏捷开发团队参与应用创建时推动实现团队协作和协调。

通过集成每个开发人员的代码构建工具和兼容的团队协作开发平台，敏捷团队能够提高工作效率和产品质量。

### 需要整个团队协作合作

移动应用往往由具备不同技能和专业知识的小型团队创建。一个典型的团队可能包含几名基本业务逻辑和 Web 服务开发人员、几名用户界面 (UI) 开发人员、一名用户体验设计人员、几名测试人员，以及一名团队领导者或管理者。

由于移动应用的交付时间通常十分紧迫，即使是小型团队也必须以最高的效率运转。任何由团队成员之间的误解和误传造成的延迟均可能会延误整个交付计划。

支持不同移动操作系统的移动应用项目需要代码共享和重用。某位开发人员的专长可能是 Android 和另一项 iOS 技能。明确了解团队成员预期要开展的工作以及需要开展工作的时机十分必要。在这种情况下，项目要求、时间期限和计划等是共享的，只有源代码、测试和构建工作可能有所不同。

### 集成的变更管理和软件版本控制

需要将特定工作项有关的所有代码变更组合成一个特定的“变更集”，或者是一个变更源代码文件列表，并一次性提供，从而将所有代码变更作为一个整体进行跟踪。理想情况下，组装一个变更集的流程应当尽量做到自然无缝，不会打断开发人员的逻辑创建思路。

版本控制以及合并 / 回滚流程还需要自动化且直观。每当开发人员不得不切换工作上下文以便执行某项任务时，在开发流程中都代表一次中断点和潜在的“减速带”。

### 需要整个项目的可追溯性

典型敏捷团队的软件开发方法是在要实施和验证的一小部分应用功能增强中定义多次较短的迭代。典型的敏捷迭代周期一般为 2 到 4 周。团队领导者可以带领该团队将工作项从积压工作列表映射到特定的迭代，并将这些工作项分配给每位开发人员。

随着开发人员承接工作项并开始执行项目，需要自动记录他们的工作状况，供团队领导者进行跟踪和审查。这样团队领导者就能在仪表盘演示中轻松跟踪和审查已经完成的工作、此刻正在处理的工作以及未来需要处理的工作信息。每位团队成员都需要能够查看迭代工作进展，以及在该迭代周期中计划完成的工作项的状态。

团队中的测试人员开始移动应用功能测试时，需要在共享开发项目中打开工作项，在测试期间查找未发现的缺陷。如果失败的测试案例链接到项目计划内的特定变更集或功能项，那么有关代码的信息已发生改变，很可能与该测试案例的失败存在联系，并将自动输入缺陷数据。此外，如果变更集链接到激发代码变更的原始要求，可利用从原始要求到测试案例的整个项目生命周期可追溯性，验证应用是否满足了这些要求。

这种“完整的项目视图”和端到端可追溯性对于任何类型的软件开



发项目都是至关重要的，但与时间紧迫且采用敏捷开发方法的移动应用开发团队的关系尤为密切。这些精益开发团队没有时间深入跟踪某项特定要求是否经过验证并得到满足以及何时执行相应操作等细节问题。

## 选择移动应用开发解决方案

IBM 移动企业应用开发方法融合了我们在常规企业软件开发流程方面累积的多年经验，以及特定于移动设备及其底层软件基础的新型工具和技术。

凭借为各行各业设计和部署企业软件的丰富专业知识，IBM 可以针对移动应用项目的开发需求提供量身定制的解决方案。IBM 本身也一直在实施转型，在数千个项目中采用敏捷方法，涉及的开发人员多达数万名。

## 协作生命周期管理(CLM)实现全面的项目可见性

如果未能对整个移动应用交付生命周期进行集成，那么开发团队仍以各种孤岛形式独立运作。形成各种孤岛时，产品交付的有效性就会受到影响。为了提供富有竞争力的移动企业应用解决方案，适应不断变化的市场需求和标准，软件工程团队必须行之有效地开展工作，协同管理所有生命周期工作产品。

IBM Rational 移动业务应用开发解决方案提供了一个全面的生命周期管理平台，支持实现任务协作并帮助衔接整个产品生命周期中

开发的各个产品组成部分。

该解决方案还制定了交付工作流程，并提供了多项任务管理功能，从而保证有效开展移动应用开发项目。我们还通过在需要移动特定功能的项目生命周期阶段中集成这些功能来充实项目。

敏捷团队协作开发工具（如 IBM® Rational® Team Concert™ 软件）支持在整个项目中定义多个短迭代。这样就能轻松地将工作项从项目积压列表中移到特定的迭代计划中。

开发人员在与软件版本控制系统相集成的移动应用代码开发工具内编辑各种文件时，系统会自动更新和维护变更集。除编辑自身需要处理的文件外，开发人员不必执行任何操作来生成变更集。

团队成员之间可以共享变更集，然后再将其与主代码流完全集成。因此，可以将 Web 服务开发人员创建的变更集（改变了 Web 服务所提供的格式）与负责处理显示新数据逻辑的 UI 开发人员进行共享，并且不会影响其他团队成员。当 UI 代码变更和 Web 服务代码变更相匹配并已准备就绪后，就可以将它们作为一项同步任务集成到主线代码流中，供其余团队成员选取使用。

## IBM Mobile Enterprise 战略提供一种移动应用运行时

协作生命周期管理平台与专供移动应用使用的代码开发工具的组合，是 IBM 全面移动应用解决方案的一个重要组成部分。但是，

有些企业级移动应用开发挑战却无法通过各种开发工具和实践加以解决。为了向单个通用的移动应用编程模型提供这项功能，我们提供了 IBM Mobile Enterprise 软件运行时。

IBM 移动业务应用开发解决方案融合了强大的团队开发功能（嵌入到 IBM Rational Collaborative Lifecycle Management 中）与 IBM Mobile Enterprise 解决方案中提供的移动工具和运行时功能。

## 全面的测试方法

全面的多层移动应用测试意味着必须使用多项测试执行功能，并通过协调得到单一的应用质量结果。如果要搭配并管理移动测试所需的各种测试执行引擎，IBM Rational Quality Manager 软件是一个出色的选择。我们的测试环境可让您制定集成的测试优先方法，从而帮助在生命周期的早期阶段就消除各种重大问题并改进经济性治理。Walker Royce 创作的“敏捷性和架构完整性评估”一文详细介绍了这项技术，提出了优先测试最难以解决的问题。

正如本文前一节所述，在当今的市场环境下，有多种技术可供测试和验证当前正在使用的移动应用。有效开发项目的经理将对其移动应用应用所有现行的技术，因为每项技术都有其自身的优缺点。对于移动应用测试，没有单个完美的答案，并且各种可用技术并不相互排斥。最有效的测试战略需要平衡使用所有形式的移动测试执行方法，并将各测试的结果编译成全面的整体移动应用“质量衡量指标”。

举例来说，IBM 方法支持的移动应用测试技术包括：

## 手动测试

手动测试是该行业目前应用最普遍的移动测试方法。但是，手动测试也是最耗时、最容易出错、最昂贵的移动测试技术。组织各种手动测试案例、为测试人员提供执行指导并存储测试结果的解决方案可以大幅降低成本。Rational Quality Manager 软件可以提供这些功能。

## 仿真器和模拟器

所有本机移动操作系统开发工具包中都随附了仿真器。模拟器则可以从多种来源获得（包括 IBM），IBM 将移动模拟器作为 IBM Mobile Enterprise 解决方案开发工具的一部分来提供。

## 协议虚拟化，应用层隔离

由于移动应用具有多层架构，所以建立基础架构，以支持在移动设备上执行代码的测试可能很耗时，并且成本也很高。通过 IBM Rational 解决方案，能够最大限度地降低成本和缩短部署延迟。测试团队不必建立复杂的中间件环境就能对移动设备上运行的代码执行测试。Rational 解决方案可以模拟中间层和后端服务与协议，这样就能集中精力对在设备本身上运行的移动应用客户端层展开测试。

## 设备上的工具和代理

最后，需要使用在移动设备上执行的某种形式的自动代码测试来替换手动功能验证。不同的供应商为此创建了不同的方法。一种典型方法是在进行自动测试的设备上额外部署某种类型的代码。此代码

将作为本地“设备上”的代理，自动地将用户输入信息送入应用，并监控这些输入信息所导致的应用行为。这项移动功能测试自动化技术与本文中所述的其他技术具有很好的互补性，并且可以同其他此类技术有效地结合使用。

## 设备云

持有、设置和管理所有这些不同移动测试设备组合的费用令人咋舌，即使资金充沛的项目对此也会望而生畏。解决这一问题的技术在于使用“设备云”测试解决方案。这种方法在降低现有以及移动应用用户有望在生产环节中使用的品类繁多的设备拥有成本方面十分有效。IBM 将整体的移动测试管理解决方案、Rational Quality Manager 软件以及拥有设备云的各种业务合作伙伴集成在一起。

## 结束语

随着越来越多的各类企业意识到需要其业务应用的移动版本，对企业级移动应用开发方法的需求也随之悄然兴起。IBM 已经打造了这种方法。

IBM 移动应用开发方法侧重强调以下五大关键主题：

- 简化移动应用的用户体验。
- 集成优先，实现更好的移动应用项目经济性治理。
- 确保要求可追溯性，通过测试来验证是否满足了这些要求。
- 制定超敏捷方法。
- 改进自动的回归测试套件，实现快速部署并降低变更成本。

这种方法支持各种移动特定的工具和技术，并且其使用效率和严谨程度与其他类型的企业应用开发项目并无二致。

## 关于作者

Leigh Williamson 是一名 IBM 杰出工程师，1988 年以来一直在德克萨斯州奥斯汀实验室工作，曾参与过众多 IBM 主要软件项目，包括 OS/2、DB2<sup>®</sup>、AIX<sup>®</sup>、OpenDoc、Java、Component Broker 和 WebSphere<sup>®</sup> Application Server。目前他是 IBM Rational 软件首席技术官团队成员，负责调整 Rational 品牌产品的战略发展方向，并引导移动应用开发解决方案的定义。Leigh 拥有诺瓦大学计算机科学学士学位及德克萨斯大学计算机工程硕士学位。

## 更多信息

如需进一步了解 IBM Rational 移动应用开发解决方案，请联系 IBM 营销代表或 IBM 业务合作伙伴，或者访问：[ibm.com/software/solutions/mobile-enterprise](http://ibm.com/software/solutions/mobile-enterprise)

此外，IBM Global Financing 可帮助您以最经济高效的战略性的方式获得您的业务所需的 IT 解决方案。我们将与信用合格的客户展开合作，定制一个 IT 财务解决方案来满足您的业务目标，实现有效的现金管理，以及改善您的总体拥有成本。IBM Global Financing 是您进行关键 IT 投资和向前推进您业务的最智慧选择。有关更多信息，请访问：[ibm.com/financing](http://ibm.com/financing)



© 版权所有IBM Corporation 2012  
IBM Corporation Software Group Route 100  
Somers, NY 10589

在中国印刷

2012年12月

IBM、IBM徽标、ibm.com、Rational和Rational Team Concert是国际商业机器公司在全球多个司法管辖区内注册的商标。其他产品和服务名称可能是IBM或其他公司的商标。有关IBM商标的最新列表，请访问[ibm.com/legal/copytrade.shtml](http://ibm.com/legal/copytrade.shtml)的“Copyright and trademark information”部分。

Linux是Linus Torvalds在美国和/或其他国家/地区的注册商标。

Microsoft、Windows、Windows NT和Windows徽标是Microsoft公司在美国和/或其他国家或地区的商标。

Java和所有基于Java的商标和徽标是Oracle和/或其子公司的商标或注册商标。

本文包含截至出版之日的最新信息，IBM可能随时更改这些信息。不是所有产品都可用于IBM运营的每个国家/地区。

本文中的信息“按原样”提供，不含任何明示或暗示的担保，包括但不限于适销性、特定用途的适用性，以及有关非侵权性的任何担保或条件。IBM产品的担保依据的是它们所遵循的协议中的条款和条件。

1 IGS论文“为您的移动企业制定有效的应用战略”，[ibm.com/common/ssi/cgi-bin/ssialias?subtype=WH&infotype=SA&appname=GTSE\\_EN\\_OS\\_USEN\\_C&htmlfid=ENW03007USEN&attachment=ENW03007USEN.PDF](http://ibm.com/common/ssi/cgi-bin/ssialias?subtype=WH&infotype=SA&appname=GTSE_EN_OS_USEN_C&htmlfid=ENW03007USEN&attachment=ENW03007USEN.PDF)

2 Walker Royce, “敏捷性和架构完整性评估”白皮书,  
[www.ijsi.org/ijsi/ch/reader/view\\_abstract.aspx?file\\_no=i92](http://www.ijsi.org/ijsi/ch/reader/view_abstract.aspx?file_no=i92)



请回收利用