

测试移动 App

要点浏览

概念：移动 App 测试相关活动的唯一目标是揭示移动 App 在内容、功能、易用性、导航性、性能、容量和安全性等方面的错误。为了做到这一点，就要采用评审和执行测试的测试策略。

人员：软件工程师和其他的项目利益相关者（经理、客户和最终用户）都应参加移动 App 的测试。

重要性：如果移动 App 的用户遭遇到错误或困难，他们便会到别处寻求所需的个性化的内容和功能。因此，在将移动 App 作为产品投放到 App 销售商或存入仓库之前，必须努力寻找和纠正尽可能多的错误。

步骤：移动 App 的测试过程要从关注移动

App 用户的可见方面开始，一直进行到有关所用技术和基础架构的测试。所执行的测试步骤包括：内容测试、界面测试、导航测试、构件测试、配置测试、性能测试和安全性测试。

工作产品：往往首先要制定移动 App 的测试计划。为每个测试步骤开发一套测试用例，并且保存测试结果以供将来使用。

质量保证措施：虽然无法确定已经完成了每一个所要做的测试，但可以肯定测试已发现了错误（并且已纠正了那些错误）。此外，如果已经制定了测试计划，便可以进行检查，以确保所有计划的测试均已完成。

推动着 WebApp 项目的紧迫感也波及到移动 App 项目之中。利益相关者都担心会错过营销机会，纷纷催促上架移动 App。然而诸如性能测试和安全性测试这样的一些技术活动往往被安排在开发过程的后阶段，因而常常是匆忙地结束。本应在测试阶段实施的可用性测试却被推迟到交付之前进行。这样就会导致灾难性错误。为避免出现这种情况，开发团队成员必须确保每一项工作产品都具有高质量，正如 Soasta 在他发表的白皮书 [Soa11] 中所表述的：

移动技术的发展简直是直上云霄，其速度超过以往的任何技术，这是史无前例的最快创新采用曲线。而且它对于业务模式具有重要意义。你的产品会快速进入市场，而且要准备被迅速采用，如果你的应用系统性能不佳，或是承载时发生故障，那么正准备取代你的市场地位的很多竞争对手就会使你陷入困境。

关键概念

- 自动化
- 检查单
- 准则
- 国际化
- 实时测试
- 策略
- 压力测试
- 自然环境测试
- 测试矩阵
- 工具与环境
- 可用性测试

使用可执行的测试用例无法单独测试移动 App 的需求和设计模型。你和你的团队应该进行技术评审（第 20 章），并测试可用性（第 15 章）及性能。作为验证移动 App 能力的一种机制，必须考虑环境的情况，此时配置测试显得尤为重要。在将移动 App 交付给最终用

户之前，这样做的目的是发现和纠正错误。

在制定移动 App 的测试策略时，要问的几个重要问题是 [Scho09]：

- 在你和用户一起测试之前，是否必须建立功能齐全的原型？
- 你应该利用用户设备进行测试还是提供测试设备？
- 在测试中你应该拥有什么设备和用户组？
- 实验室测试和远程测试相比各有什么优缺点？

在本章的整个讲述中，我们会涉及以上每个问题。

26.1 测试准则

完全在移动设备上运行的移动 App 既可以使用传统的软件测试方法（第 23 章）进行测试，也可以在个人计算机上使用模拟运行的方式测试。另一方面，利用基于服务器资源的瘦客户端移动 App 对测试特别具有挑战性。除了 WebApp 提出的许多测试挑战（第 25 章）以外，瘦客户端移动 App 的测试还必须考虑互联网网关和电话网络进行的数据传输 [Was10]。用户期望了解移动 App 运行环境，并提交基于设备的物理位置，以及与可用的网络功能相关的个性化用户体验。但是，使用每个可能的设备和网络配置在一个动态的特定网络环境中测试移动 App，这如果不是不可能也是十分困难的。

要了解移动 App 测试的目标，就应该首先考虑 App 设计师所面临的许多独特的挑战。人们期望所得到的移动 App 具有复杂的功能，并且还具有基于个人计算机的 App 所具有的可靠性。但是移动 App 常驻于具有相对有限资源的移动平台。下列准则为移动 App 的测试提供了基础 [Kea07]：

建议 尽早让用户参与测试活动，这样可以尽早得到有助于纠正设计的反馈信息。

- 在测试以确定瓶颈之前了解网络和设备的运行环境。跨边界测试在 26.4 节讨论。
- 在不受控的现实环境的测试条件下进行测试（基于现场的测试），特别是针对多端网络（multitier）移动 App 进行测试。生产环境中的测试在 26.2.5 节讨论。
- 选择适当的自动测试工具。理想情况下，这些工具应该支持所有要使用的平台，允许测试各种各样类型的屏幕、分辨率和输入机制，如触控式和小键盘等，并实现与外部系统的联通以进行终端对终端的测试。有关移动 App 测试工具更详细的讨论见 26.6 节。
- 利用加载设备平台矩阵法确定最为关键的硬件 / 平台测试组合。特别是在硬件 / 平台组合数较多而测试时间又很短时，这种方法是非常有效的。采用这一方法的详细情况可参看 26.2.3 节。
- 至少检查一次在所有可能的平台上的端对端功能流。当涉及网页服务时，如果没有性能测试工具，跟踪所需的实际网络路径以交付移动 App 的功能是很难的。关于工具的使用在 26.6 节讨论。
- 使用实际设备进行性能测试、图形用户界面（GUI）测试和兼容性测试。虽然可以利用模拟器完成这些测试，但在此仍然建议使用实际设备。用户交互测试在 26.3 节讨论，而性能测试问题在 26.2 节讨论。
- 测量性能只在无线通信和用户负载的实际条件下进行。移动 App 的实时测试问题在 26.5 节讨论。

26.2 测试策略

仅使用技术手段不能保证移动 App 在商业上的成功。如果这些 App 在运行中失灵,或是未能达到预期的使用效果,那么用户将会很快遗弃这些 App。回顾测试工作所具有的两个重要目标是很有意义的,那就是:(1)为在开发活动早期阶段发现缺陷而制定测试用例;(2)验证其具有重要的质量属性。移动 App 应具有的质量属性是基于国际标准 ISO 9126[Spr04]提出的那套软件产品的质量属性,包括功能性、可靠性、易用性、效率、维护性和可移植性(第 19 章)。

引述 如果我还
没有死去,那么
我宁愿葬身于手
机之中。

Amanda Holden

制定移动 App 测试策略既要了解软件测试知识,又需理解移动设备及其网络基础架构特性所面临的挑战[Kho12a]。除了具有常规软件测试方法的全面知识(第 22、23 章)以外,移动 App 的测试人员还应该对电信原理有很好的理解,并且要认识到移动操作系统平台的差异和功能。对于这些基础知识,必须要有另外的知识对其加以补充,包括:对不同类型移动测试的深入理解(例如,移动 App 测试、移动手持终端设备的测试、移动网站测试),模拟器的使用,测试自动化工具以及远程数据存取服务(RDA)。上述专题都将在本章后面的小节中陆续讨论。

信息栏 移动 App 测试——检查单

移动 App 咨询有限公司 Appsparq 的首席执行官 Nari Kannan 对测试移动 App 发表了以下建议[Kan11]:

- **概念测试**——在开始开发之前,从移动 App 的未来用户处获取输入信息,有助于确保 App 已囊括必要的系统特征,而不致遗漏最为重要的特征(收集需求和确认技术在第 9 章中讨论)。
- **单元测试和系统测试**——实施正规的单元测试和系统测试针对的是含有多个构件的任何软件应用,并与网络交互作用(第 22、26 章)。
- **用户体验测试**——如在开发过程早期就吸收实际用户参与,便能保证所提供的体验达到利益相关者对易用性和可访问性的期望。移动 App 的易用性测试在 26.3 节讨论。
- **稳定性测试**——确保移动 App 不会由于与

网络或网站服务不兼容而崩溃。在生产环境中测试移动 App 在 26.2.5 节讨论。

- **连接性测试**——测试直接进入所有重要构件和网络资源,这对于确保移动 App 在运行环境中适当地使用是非常重要的。跨界测试在本章 26.6 节讨论。
- **性能测试**——测试移动 App 的能力以满足其非功能性需求(下载时间、处理器速度、存储容量等)。性能测试在 26.2.4 节和 26.5 节讨论。
- **设备兼容性测试**——验证移动 App 是否在所有目标设备上正常运行。该项任务在 26.2.3 节讨论。
- **安全性测试**——确保移动 App 满足其利益相关者设定的安全性需求。安全性保证测试在第 27 章讨论。
- **认证测试**——检查移动 App 是否满足由分发机构制定的标准。

26.2.1 传统方法适用吗

综合性移动 App 测试程序包括第 22 章讨论的通用螺旋方式,以及第 23 ~ 25 章中提到

的客户-服务器架构、实时计算、图形用户界面、WebApp 以及面向对象系统。移动 App 测试也面临着独特的挑战,它要确保 App 可满足其功能需求和非功能需求。^①

Vinson[Vin11] 建议,移动 App 的测试人员可适当调整测试 WebApp 所用的策略(第 25 章)。当然,测试内容时应考虑到移动设备和特定网络的局限性。由于设备特性和用户环境的多样化,兼容性测试和部署测试在移动领域中更加具有挑战性。性能测试需要确定在移动设备的有限存储、处理、连通性以及可用能力上,是否会对特性或功能性产生负面影响。移动 App 的性能测试经常是在某些细节层次上进行的,这只见于实时系统的开发。安全性测试必须考虑到物理设备的损耗,并且注意到一个事实,移动 App 经常是直接在移动设备硬件上运行,这就为窃取个人数据带来一定的显露度风险。移动 App 的设计往往考虑使用者是那些缺乏技术知识的人员,这与 Web 用户相比是不同的。这就需要进行更为广泛的用户体验测试(26.3 节)。敏捷开发过程模型(第 5 章)和测试驱动开发模型(第 24 章)均可采用。

关键点: 为测试 Web 应用所采取的所有策略都是测试移动 App 的人员必须要采纳的。

[570]

26.2.2 对自动化的要求

移动 App 的测试人员常会遇到许多不同的配置(多种设备、多种操作系统和各种移动网络),还要保证移动系统的功能和运行环境是已知的。由于高效和全面地测试移动 App 十分重要,所以自动化地完成配置测试(25.7 节)或回归测试(22.3 节)就很有意义。不过还应该注意,要做到全面实现移动 App 的自动化测试(例如,在掌上视频游戏中的用户交互查找)是不可能的。

当测试人员不得不机械地处理大量重复性的测试用例时,自动测试工具便能提高团队成员的士气。越早使用自动测试工具,测试频度越高,就能越早地发现移动 App 的缺陷。敏捷开发过程(第 5 章)要求使用按日的构建周期,需用回归测试保证变更不会产生意想不到的负面影响。

移动通信公司(Mobile Labs, Inc.)提出了一种移动 App 的自动测试方法[Mob 11],内容包括:

可行性分析。若采用自动化测试,就应确知哪些测试和哪些测试用例具有最大的投资回报率(ROI),侧重点放在自动重复和经常使用的测试用例上,目标是争取做到手工测试用例的 50% ~ 60% 实现自动化。

概念证明。确认测试自动化的价值。为做到这一点,要使有限数量的手工测试脚本自动化,以确定工作量的投资回报率(ROI)。测试团队必须确定其他脚本的可测量性以及后续周期中的复用度。

测试框架的最佳实践。针对移动 App 提出了一种方法,以此作为其测试过程的基础。框架决定了移动 App 的实现规则和测试规则,框架是针对每个移动平台开发的,并且适合企业中的所有 App 的情况。

定制的测试工具。借助于先进的脚本技术,对每个移动平台(和被测的 App)定制测试工具。

引述: 如今有几十个移动设备,但作为一名开发人员,为了和它打交道,你必须对每种设备能做什么有一定的理解。

John Fowler

提问: 自动化移动测试最重要的是哪些方面?

① 对移动 App 测试特有问题的概述见 <http://www.utest.com/landing-interior/crowdsource-your-mobile-app-testing>。

[571]

在实际条件下测试。确知 App 如何在测试实验室之外的真实设备上运行。在真实设备上而不是在模拟器上进行测试能够降低假缺陷报告的发生率，并且更有可能发现用户级的错误（26.2.5 节）。

快速处理缺陷。通过自动提交缺陷信息和生成差异报告的加速实施，减少了开发人员处理缺陷的时间。

复用测试脚本。在考虑增强 App 的能力时，不必从头开始创建测试用例，复用可以帮助我们节省成本。要使功能界面和测试逻辑分离，测试工具的体系结构是很重要的。这就使界面包括在可复用的功能中，如同让工具作为新平台和新设备一样。

26.2.3 建立测试矩阵

移动 App 往往是为多种设备开发的，并且可用于不同的环境和不同的位置。加权设备平台矩阵（Weighted Device Platform Matrix, WDPM）有助于确保测试范围覆盖移动设备的各种组合和各种不同的环境^①。加权设备平台矩阵也可协助进行设备 / 环境组合的优先排序，以便于优先做最重要的测试。

为多个设备和操作系统建立加权设备平台矩阵（图 26-1）的步骤是：（1）矩阵的纵列给出各种重要的操作系统；（2）矩阵的横行是各种目标设备；（3）矩阵的第二行和第二列分配了排名次序（如从 0 到 10），这是为了表明每个操作系统和每种设备的相对重要性；（4）计算每对排名次序数的乘积，并填入矩阵（矩阵中有的排名次序对可能是无效组合）。

		操作系统 1	操作系统 2	操作系统 3
	排名	3	4	7
设备 1	7	无效组合	28	49
设备 2	3	9	无效组合	无效组合
设备 3	4	12	无效组合	无效组合
设备 4	9	无效组合	36	63

图 26-1 设备平台加权矩阵

要调整测试工作，使对于正在考虑的每个环境变量而言，具有最高评级的设备 / 平台组合得到最多的关注。在图 26-1 中设备 4 和操作系统 3 就具有最高评级，因此，在测试过程中将会获得最优先的关注。

[572]

26.2.4 压力测试

对移动 App 进行压力测试是要在极限运行条件下力图查找错误。此外，压力测试还提供了一种机制，在不损害安全性的情况下观察移动 App 的运行水平是否会降低。极限活动包括：（1）在同一设备平台上运行几个移动 App；（2）感染带病毒的或是恶意的系统软件；（3）尝试接管设备并使其传播垃圾邮件；（4）使移动 App 处理数量极大的事务；（5）在设备上存储异常大量的数据。遇到这些情况时，检查移动 App 以确保资源集中服务（例如流媒体）得以适当实施。

有效的压力测试 [Soa11] 应该“在自然环境下”进行，测试中会涉及

关键点 故障弱

化是容错系统的重要特性。在出现错误时，如果关闭之前系统不能修复损害并允许继续执行，那么系统将使故障弱化，以期获得已知的安全状态。

^① 环境变量是与当前连接或当前事务相关的变量，移动 App 将用来指导其可见的用户行为。

各种设备和各种操作系统，用两到三倍的额定量进行测试。测试应该在现实生活环境反映用户的各种使用状况，包括本地和其他国家的不同网络配置和不同标准。

在以下几节中将更为详细地叙述这些原则。

26.2.5 生产环境中的测试

许多移动 App 的开发商主张进行自然环境测试，或是在用户的本地环境中使用移动 App 资源的生产发布版本进行测试。伴随着移动 App 的演变，自然环境测试是要敏捷地响应变更 [Ute12]。

自然环境测试的特征包括不利的和不可预测的环境、过时的浏览器和插件、独特的硬件以及不完善的联通性（无线网络和动态载流）。为了反映真实情况，测试人员的人口统计学特征应该与目标用户的特征及他们的设备特征相匹配。此外，应该包括涉及少量用户的用例、不太流行的浏览器以及各式各样的移动设备。自然环境测试总是会带有不可预测性，并且测试计划必须适应测试的进展。为了解更多的信息，可参看 Rooksby 和他的同事在他们论文中关于自然环境测试的成功策略 [Roo 09]。

引述 每个人口袋里都有手机这种微型计算机，这是重要的移动设备。

Ben Horowitz

创建自用测试环境的过程是昂贵且易于出错的。基于云计算的测试可以提供标准化的基础架构和预配置的软件映像，使得移动 App 团队不必担心找不到服务器或是无处购买软件和测试工具的许可证。云服务提供商为测试人员的访问提供可扩展性，并为用户准备好虚拟实验室，其中有操作系统库、测试管理工具和执行管理工具，以及为生成能够准确反映现实生活的测试环境所需的存储器 [Myl 11]。

基于云的测试并不是没有潜在的问题，例如在使用云方法时，存在缺少标准、安全隐患问题、数据布局和完整性问题、不完备的基础架构支持、服务的不当应用以及性能问题等，但这也仅仅是开发团队所面临的一些共性挑战。

573

SafeHome 生产环境中的移动 App 测试

[场景] Doug Miller 办公室。

[人物] Doug Miller (SafeHome 软件团队经理)；Vinod Raman (SafeHome 产品软件团队工程师)

[对话]

Doug：你认为我们的 SafeHome 移动 App V0.0 版的电子商务部分怎么样？

Vinod：外包商调整了 SafeHomeAssured.com 上的移动 App 来适应移动 App 环境，做得很好。Sharon (供应商的开发经理)告诉我，现在他们正在测试原型。

Doug：我听说他们在用设备模拟器为电子商务网站做测试。我认为我们应该在实际设备上做些测试。

Vinod (扮了个鬼脸)：我以为应该请一个第三方测试公司来验证移动 App，好像我们还在对自己的产品赶尽杀绝！

Doug：我们将请测试供应商来做性能测试、安全性测试以及配置测试。我们的外包商已经做了一些测试。我只是想换个角度，看来会是有益的，而且我们希望能控制成本，所以……

Vinod (叹了一口气)：你期望怎样呢？

Doug：我想要确保用户有可靠的体验。

Vinod：我认为可以从每一个主要的界面功能的用例开始。

Doug：好的。只是跟踪它们的逻辑路径，从头一直到尾。检查加权设备平台矩阵。

我希望你能在顶层的六个最重要的设备上检查其性能，同时你还要检查出现在每个导航节点处的内容。当每个屏幕显示出内容时，确保设备的特性都考虑在内。

Vinod：当然要这么做。那么还有功能元素也要考虑。现在谁在测试可用性呢？

Doug：哦，测试厂商会调整可用性测试。我们已经聘请了一家市场调研公司，他们为可用性研究列出了 20 个典型用户。不过如果你发现了任何可用性问题……

Vinod：我知道，那就一起传给他们。

Doug（笑着说）：多谢了，Vinod。

26.3 与用户交互的各种情况

在功能相同的多种产品充满市场的情况下，用户自然会挑选易于使用的移动 App，其中用户界面及其交互机制是移动 App 用户的可见部分。移动 App 提供的用户体验质量测试能满足用户的期望，这是非常重要的。

在第 15 章中讨论的许多评估软件用户界面的可用性规程可以用来评估移动 App。同样许多用于评估移动 App 质量的策略（第 25 章）也可用来测试移动 App 的用户界面部分。人们还构建了更多良好的移动 App 的用户界面，这些并不是简单地从现有的个人计算机应用系统中缩放用户界面的尺寸而得到的。

提问 哪些移动 App 的可用性特征成为了测试的重点？这会涉及哪些特定目标？

574

信息栏 移动 App 可用性测试构件

在移动 App 测试博客中提供了许多有关移动 App 关键构件可用性测试的建议。^①

- **功能性**——确保用户故事支持核心功能性，确保利益相关者的目标和期望也在考虑之中。
- **信息架构**——确保内容和链接已结构化，并以逻辑形式呈现，而且数据块^②及其联系也在考虑之中。
- **内容**——使用文本、视频、图像和多媒体，只有当其在移动环境下支持用户任务时，才帮助用户控制是否启动媒体，以保证内容用移动设备的格式呈现。
- **设计**——设计屏幕的快速可视化扫描，同时考虑纵向和横向显示，重新考虑屏幕的布局，不只是缩放。
- **用户输入**——让用户易于输入数据，提

供自动完成和拼写检查，显示默认值，提供基于单个设备能力的交替输入机制。

- **移动环境**——说明环境的变更（一天中的某一时间、地点、网络）并利用设备特性和能力进行预测，并支持用户使用环境。
- **可用性**——确保交互设备（触摸屏、键盘、音频）和小部件（按键、链接、滚动条）都共同地向着目标设备协调运行，遵循约定并适应学习曲线。
- **可信性**——对保密性和安全性要敏感，未取得明确的用户许可之前，不收集个人信息，允许用户控制如何共享个人信息以及如何叙述商业活动。
- **反馈**——向用户提供重要信息，使报警

① 博客可见于 <http://www.mobileapptesting.com/10-key-components-of-successful-mobile-app-usability/2012/09/>。

② 数据块是指打破超媒体文件，使之成为更小的相关信息组，以利于读写器更快地进行吸收。

次数最小化,使报警信息简明,为用户操作提供确认而无需中断用户的工作流。

- 帮助——让用户易于访问帮助和支持选项,提供环境帮助。

26.3.1 手语测试

由于当前移动设备中非常流行触摸屏,因此,开发人员已添加了多种触摸手语(例如,刷新、缩放、滚动、选择等)作为扩展用户交互的可能性,这些手语不会造成屏幕损耗。然而不幸的是,手语密集界面带来了大量的测试挑战。

使用自动化工具测试触摸或手语界面操作是很难的。手语是难以准确记录以用于重现的。屏幕大小和分辨率以及之前的用户操作都会影响到屏幕上对象的位置。有时纸上原型会作为设计的一部分,但这并不能满足手语测试的要求。相反,测试人员需要开发测试框架程序,使其完成模拟手语测试的功能。但是这些做法都是既昂贵又费时的。

[575]

视障用户的可访问性测试是具有挑战性的,因为手语界面通常不提供任何触觉反馈和听觉反馈。对于智能手机之类的无处不在的设备,手语的可用性和可访问性测试是非常重要的。当手语操作无效时,测试设备的操作就更加重要。

用户故事或用例在理想上可以写得足够详细,使其可作为测试脚本的基础。补充有代表性的用户是非常重要的,包括所有目标设备在内都要进行补充,当使用移动 App 测试手语时,要考虑屏幕差异。最后,测试人员应确保手语符合为移动设备或平台设定的标准和环境。

26.3.2 语音输入和识别

目前智能移动设备可以使用语音输入,也允许同时进行设备的手忙操作和眼忙操作。语音输入可以采用几种形式,其中每个过程都带有所需的不同级别的编程复杂性。当消息记录只用于后期回放时,电子语音信箱输入便可以起作用了。离散字识别可以让用户从具有数量不多选项的菜单中,以口语提出选项。连续语音识别的目的是将口述语音直接转化为有意义的文本串。每一种口音的输入都将对其自身的测试构成挑战。

根据 Shneiderman[Shn09] 的理论,来自噪音环境的干扰会妨害各种形式语音的输入和处理。与指向屏幕对象或按键相比,使用语音命令来控制设备会给用户带来更大的认知负担。用户必须想出正确的字和词,以便移动 App 执行所需的动作。当屏幕上显示出一个对象时,用户只是要辨认出适合的屏幕对象,并将其选中即可。然而语音识别系统的广度和准确性还在迅速发展。在很多移动 App 中,语音识别很可能成为通信的主要形式。

建议 语音输入测试应该考虑到环境状况以及个人声音的变化。

即使对于最好的测试机构来说,测试语音输入和识别的质量和可靠性都是技术上的挑战。不正确的语音输入(由于用户错误而发出错音的字、词或是环境的干扰)都必须通过测试分辨出来,以确保不正确的输入不致造成移动 App 或设备的失灵。由于每个用户/设备组合都会有所不同,因此用户的广大人群和环境都应该考虑在内,以保证将差错率限制在可接受的范围之内。最后,记录错误也是重要的,这可以帮助开发人员提高移动 App 的语音输入能力。

[576]

26.3.3 虚拟键盘输入

由于激活虚拟键盘时可能会遮挡部分显示屏,因而应测试移动 App 以确保当用户键入时重要的屏幕信息不会被隐藏,这是十分重要的。如果必须隐藏屏幕信息以测试移动 App

的能力,则要让用户轻触页面,但并不丢失输入的信息。[Sch09]。

通常虚拟键盘比个人计算机的键盘小,因此很难用十个手指按键。由于键本身较小,难以准确敲击,而且并不提供触觉反馈,因此必须测试移动 App 以确保它易于纠错,并且在键入错误词语时不导致崩溃。

预测技术(即自动完成部分词语的输入)往往使用虚拟键盘来帮助用户加快输入。如果考虑要使移动 App 面向全球市场,针对用户选择的自然语言,测试输入词语完成后的正确性是十分重要的。同样重要的是测试任何机制的可用性,以允许用户可以不顾建议的做法。

通常虚拟键盘测试是在可用性实验室中进行的,但有些则应该在自然环境下进行。如果虚拟键盘测试发现了重要的问题,那么唯一的选择是确保移动 App 可以接受设备的输入,而不用虚拟键盘输入(如人工输入或语音输入)。

26.3.4 警报和异常条件

当移动 App 在实时环境中运行时,有许多因素在影响着它的行为。例如,当用户在使用移动 App 时,可能丢失无线网络信号,或是传入文本消息、电话呼叫,还可能接收的日历警报。

这些因素可能破坏移动 App 用户的工作流,然而大多数用户会允许弹出警报或是中断,因此,移动 App 测试环境必须能够模拟这些警报和条件。此外,在实际设备的工作环境中,应该测试移动 App 处理警报和条件的能力(26.5 节)。

移动 App 测试应该注重与警报和弹出消息相关的可用性问题。测试应该检查警报的清晰度和环境,检查这些事件在设备显示屏上出现位置的适当性,并且当涉及外语时,要验证一种语言翻译成另一种语言的正确性。

在各种移动设备上,由于网络或环境的变化,可能会引发许多不同的警报和条件,虽然许多异常处理过程可以用软件测试工具进行模拟,但在开发环境中,不能仅仅依靠模拟测试。这里再次强调实际设备在自然条件下测试移动 App 的重要性。

[577]

26.4 跨界测试

国际化是一个创建软件产品的过程,它使得在多个国家、操着各种语言来使用产品成为可能,而不需做任何工程的变化。本地化是调整软件应用以适应全球各地区使用情况的过程,通过添加各地的特定需求和把产品文本部件翻译为适用的语言来实现。除了语言差异以外,本地化还应考虑到不同国家的货币、不同的文化、税收和标准(包括技术标准和法律等)[Sla12]。因此,若要在世界许多地区投放和使用移动 App,则在这些方面进行测试显然是非常有必要的。

每个国家为实施本地化计划构建内部测试设施是非常昂贵的,相比之下,对每个国家本地供应商的外包测试则是较为划算的[Reu12]。然而,采用外包方式时,在移动 App 开发团队和实施本地化测试的供应商之间会有通信水平下降的风险。

众包(crowdsourcing)^①在很多在线社区都很流行。Reuveni[Reu12]

引述 通过移动设备、社交、云计算、大数据、社区以及其他强大力量的融合,整个世界正在重塑。这些技术的结合开启了令人难以置信的机会,以一种全新的方式把一切都连接在一起,使我们的生活方式和工作方式发生着惊人的转变。

Marc Benioff

① 众包是一种分布式问题解决模型,其中社区成员影响着指派给小组的问题解决方案。

建议在开发环境之外，众包可以供分布于全球的本地化测试人员使用。要做到这一点，发现声誉高且有成功业绩的社区是非常重要的。易于使用的实时平台使社区成员可与项目决策者沟通。为了保护知识产权，只有愿意签署保密协议且可信的社区成员才允许参加测试。

26.5 实时测试问题

由于在设备中实现的硬件和固件的组合，使得实际移动设备具有固有的局限性。如果潜在的设备平台范围很大，那么执行移动 App 测试将是昂贵的，而且还是耗时的。

设计移动设备时并没有考虑到测试。有限的处理能力和存储容量可能不允许装载所需的诊断软件以记录测试用例性能。而模拟设备则往往更易于管理，也更易于获取测试数据。

每个移动网络（全球有数百个）均采用自己独特的基本架构。移动网络运营商实现的网络代理指导用户如何连接到特定网络资源，以使用其网络。这可能会在服务器和测试客户端之间限制信息流传输。有些代理可能会在超文本传输协议（http）的头部信息中清除重要信息，其中包括你的 App 需要的功能或设备适配的信息。网络信号强度可能会成为其中的问题。模拟器经常无法模拟网络服务的效果和时序，当移动 App 在实际设备上运行时，可能观察不到用户发现的问题。

578

远程移动设备是一种有用的测试工具，使用它可以克服使用模拟器的一些局限性。远程移动设备是一个实际的移动手持机，安装在带有遥控器和遥控天线的盒子里。遥控器与设备屏幕和键盘控制电路相连。当连接到互联网时，这种组合方式允许本地个人计算机上的用户或是网络客户端的用户按下按钮，收集在远程设备上发生的数据。此外，为了随后可以重放，一些远程设备具有记录测试用例的能力，以此协助自动实施回归测试的过程。

最后，在移动设备上监测与移动 App 的使用特别相关的功耗是很重要的。与监测网络信号相比，从移动设备传输信息会消耗更多的电能。与加载网页或发送短信相比，处理流媒体会消耗更多的电能。因此，准确评估功耗必须在自然条件下的实际设备上实时监测。

26.6 测试工具和环境

在 26.3.2 节中讨论过，缩短测试时间、提高测试过程的质量和覆盖范围是将移动 App 测试的某些方面实现自动化的理由。同样，我们在 26.2.5 节中讨论了在生产环境中进行测试的重要性。不过，如果时间允许，手工测试还是需要的。但即使是在这种情况下，工具也可以用来监测跨网络设备上的移动 App 和用户的行为。

当选择移动测试自动工具时，Khode[Kho12b] 建议采用以下几条准则。通常情况下这些准则也适用于移动测试工具。

- 对象识别。工具可以使用各种方法识别设备对象，这些方法可能是对象 ID、图像处理、文字识别、HTML5 或 DOM 对象。
- 安全性。工具不得要求使用未受保护的设备与公共互联网相连。
- 设备。工具利用实际用户的设备，无需使用专门的开发商模式。
- 功能性。支持所有设备的功能，包括多点触摸手语、虚拟键盘输入、呼入和手机短信、警报处理及其他。
- 模拟器和插件。在不同的设备和不同的移动操作系统上使用现有的测试环境，可执行同样的测试。

提问 我们应该采用哪些准则为移动测试选择自动化工具？

579

- 连通性。同时使用 USB、Wi-Fi、私有云和电话载波的多设备连接，以测试连接的稳定性和恢复性。

软件工具 为移动 App 测试选择工具

以下列出了可用于移动 App 测试的几种有用的工具。这是一个不断变化的领域。下面的工具较有代表性，是最近由 Brown[Bro 11] 和 Vinson[Vin 11] 推荐的。^①

移动网页工具要确定网页友好的移动设备的友好程度。用户输入网页的网址 (URL)，该工具会给出一个缺陷表。

- WC 3 mobile OKChecker, <http://validator.w3.org/mobile/>。

移动浏览模拟器在移动浏览器上显示网页的外观。用户会输入一个网页的地址 (URL)，工具会在移动浏览器上显示其外观。

- 手机模拟器, <http://www.mobilephonemulator.com/>。
- 苹果 iphoney, <http://www.maketcircle.com/iphoney/>。

设备模拟器通常是在个人计算机上运行的虚拟设备，可以用来开发和测试移动 App，而无需访问实体设备。

- 安卓模拟器, <http://developer.android.com/sdk/index.html>。
- ipad peek, <http://ipadpeek.com/>。
- Adobe Edge Inspect, <http://html.adobe.com/edge/inspect/>。
- BlackberrySimulators, <http://us.blackberry.com/sites/developers/resources/>

simulators.html。

自动化工具在 iOS) 或 Android 上记录交互作用，作为允许重放的测试脚本。通常这些工具在带有设备模拟器的个人计算机上运行。

- MonkeyTalk, [http://www.gori-llallogic.com/testing-tools/monkey talk](http://www.gori-llallogic.com/testing-tools/monkey-talk)。
- Eggplant Mobile, <http://www.testplant.com/>。
- Device Anywhere, <http://www.Keynotedevicewhere.com/>。

网络监测工具添加、修改和过滤发送到网络服务器上的 HTTP 请求信息头。还可作为浏览器插件进行安装。

- 修改信息头, <http://addons.mozilla.org/en-us/firefox/addon/modify-headers/>。

移动分析测试用于收集数据，以分析用户如何与移动 App 进行交互，这对于评估投资回报率 (ROI) 是很重要的。通常需要网络或云服务以协助数据收集和存储。

- Flurry, <http://www.flurry.com/flurry-analytics.html>。
- Google 移动分析, <http://www.google.com/analytics/mobile/>。
- Distimo Monitor, <http://monitor.distimo.com/>。

580

26.7 小结

移动 App 测试的目标在于对多种移动 App 的质量进行检测，以找出错误或是发现可能导致质量事故的问题。测试会集中于若干质量元素，如内容、功能、结构、可用性、使用环

^① 这里提到的工具只是此类工具的例子，并不代表本书支持采用这些工具。在多数情况下，工具的名字由各自的开发者注册为商标。

境、导航性、性能、电能管理、兼容性、互操作性、容量和安全性。在完成移动 App 设计后，将进行评审和可用性评估，在应用实现后和在实际设备上部署时，就要进行测试。

通过对各“单元”内容、功能性或导航性的初步检查，移动 App 的测试可检测到每一维度的质量。在对每一部件进行验证以后，就将移动 App 作为一个整体进行测试。为做到这一点，许多测试是由用户的观点导出的，并且由用例中包含的信息驱动。移动 App 计划制定出来以后，就确定了测试步骤、工作产品（如测试用例）以及评估测试结果的机制。测试过程包含若干不同类型的测试。

内容测试（或评审）关注的是各类内容。其目的在于检查那些影响将内容展示给最终用户的错误。因移动设备限制而导致的性能问题也必须是检查的内容。界面测试检查交互机制和由移动 App 所提供的用户体验。这样做的目的是为了发现当移动 App 不考虑设备、用户或是位置等条件时所导致的错误。

导航测试基于用例，是作为建模活动的一部分而导出的。针对部署移动 App 所用的体系结构框架内的导航设计，以测试用例来运行每个场景。构件测试检查移动 App 中的内容和功能单元。

针对特定设备或网络环境，配置测试力图发现错误和兼容性问题。此后，测试是要发现与每个可能的配置相关的错误。由于移动设备和网络服务提供商的数量巨大，因此将使问题更复杂。安全性测试包含了许多针对移动 App 或其环境中薄弱环节的测试。安全性测试的目的是查找在设备操作环境中或是访问的网络服务中存在的安全性漏洞。性能测试是在需要服务器端的资源容量增加时，评估移动 App 的响应时间和可靠性。最后，移动 App 测试应该解决的性能问题包括：耗电量、处理速度、内在限制、从故障中的恢复能力以及连通性问题等。

581

习题与思考题

- 26.1 在某些情况下，是否可以忽略在实际设备上所作的移动 App 测试？
- 26.2 用自己的话说明测试移动 App 的目标。
- 26.3 兼容性是重要的质量因素。为了确保移动 App 的兼容性，必须要测试什么？
- 26.4 安装一款免费的移动 App 测试工具，对于你所熟悉的移动 App，评价该工具的性能。
- 26.5 移动 App 的什么元素可能在单元测试中得到测试？在移动 App 的元素集成以后才能进行什么类型的测试？
- 26.6 开发正式的书面测试计划是否总是必要的？请作解释。
- 26.7 全部移动 App 的测试策略都是以用户可见元素开始的，并且向技术元素发展。这种说法是否合理？这个策略是否有例外？
- 26.8 验证性测试（certification testing）在传统意义上真的是测试吗？请作解释。
- 26.9 描述对移动 App 所做的与用户体验测试有关的步骤。
- 26.10 假如你正在开发一个访问网上药店的移动 App，为的是满足老年人的购药需求。药店可提供典型功能，而且还为每位客户维护数据库，以便提供药物信息并且对可能的药物间潜在相互作用提出警告。针对这一移动 App 讨论特定的可用性测试。
- 26.11 假定您已实现了网络服务，为 YourCornerPharmacy.com 提供了药物相互作用检查功能（参看习题 26.10）。探讨将要在移动设备上实施的构件级的测试类型，以保证移动 App 正常地使用这项功能。
- 26.12 如何考虑测试移动 App 的环境能力？

- 26.13 在生产环境中测试移动 App 时遇到的每一种配置都可能做到吗? 如果不可能, 应如何选择一组合理的测试配置?
- 26.14 描述 YourCornerPharmacy 药店(习题 26.10)的移动 App 可能要进行的安全性测试, 谁应执行这种测试?
- 26.15 移动 App 的压力测试和 WebApp 的压力测试之间的差别是什么?

扩展阅读与信息资源

有许多描述移动计算的书籍, 其中往往包含着对移动 App 测试的讨论。Kumer 和 Xie 的书(《Handbook of Mobile Systems Applications and Services》, Auerbach Publications, 2012)涵盖了移动服务以及移动计算中面向服务体系结构的作用。普及性计算读物定义了移动计算环境的原则, 包括: Chalmers(《Sensing and Systems in Pervasive Computing: Engineering Context Aware Systems》, Springer, 2011)、Adelstein(《Fundamentals of Mobile and Pervasive Computing》, McGraw-Hill, 2004)以及 Hansmann(《Pervasive Computing: The Mobile World》, 2nd ed, Springer, 2003)。在另外一书中, Nguyen 等人(《Testing Applications on the Web: Test Planning for Mobile and Internet-Based Systems》, 2nd ed., Wiley, 2003)讨论了注重测试可理解性(test accessibility)、可靠性和安全性的移动 App 测试。

有关界面设计的书籍有很多, 其中不少都包含有关测试移动 App 可用性方面的信息。Ben Shneiderman 和他的同事编写的书(《Designing the User Experience》, 5th ed., Addison-Wesley, 2009)在可用性方面是一本最优秀的著作。其他较好的参考书有: Quesenberry 和 Szuc(《Global UX: Designing and Research in a Connected World》, Morgan Kaufmann, 2011)以及 Schumacher(《Handbook of Global User Research》, Morgan Kaufmann, 2009)。Nielsen(《Mobile Usability》, New Riders, 2012)提供了有关如何设计可用界面的建议, 其中也考虑到了移动设备的屏幕尺寸。Colborne(《Simple and Usable Web, Mobile, and Interaction Design》, New Riders, 2010)描述了简化用户交互的过程。Ginsburg(《Designing the iPhone User Experience: A User-centered Approach to Sketching and Prototyping iPhone Apps》, Addison-Wesley, 2012)讨论了采用以用户为中心的方法评估用户体验的重要性。

Meier(《Microsoft Application Architecture Guide》, 2nd ed., Microsoft Press, 2009)提供了有关移动 App 测试的有用信息。而 Graham(《Experiences of Test Automation: Case studies of Software Test Automation》, Addison-Wesley, 2012)为测试自动化提供了良好的背景。Lee(《Test-Drive iOS Development》, Addison-Wesley, 2012)研究了在设计测试驱动的环境中移动 App 的测试过程。

在网上有各种各样关于移动 App 测试的信息源。最新的参考文献可在 SEPA 网站 www.mhhe.com/pressman 上的“software engineering resources”中找到。