

认证测试工程师

ISTQB®移动应用测试

基础级扩展大纲

2019版

由国际软件质量研究所（iSQI）提供

国际软件测试认证委员会



中文版的翻译编辑和出版统一由 ISTQB®授权的 CSTQB®负责



版权标志

如果此文档的来源是确认的，则可以拷贝此完整的文档或部分。

版权标志© International Software Testing Qualifications Board (以下称为ISTQB®)

ISTQB®是International Software Testing Qualifications Board的注册商标

移动应用专家认证-基础级（CMAP-FL）课程大纲的作者— Jose Diaz, Rahul Verma, Tarun Banga, Vipul Kocher 和 Yaron Tsubery—特此将版权转让给国际软件测试认证委员会(ISTQB®)。此教学大纲用作创建当前文档的基础。

版权© 2019作者是Vipul Kocher (主席), Piotr Wickerski (副主席), José Díaz, Matthias Hamburg, Eran Kinsbruner, Björn Lemke, Samuel Ouko, Ralf Pichler, Nils Röttger, Yaron Tsubery。

本文档由国际软件测试认证委员会移动应用测试工作组的核心团队编写。

Vipul Kocher (主席), Piotr Wickerski (副主席), José Díaz, Matthias Hamburg, Eran Kinsbruner, Björn Lemke, Samuel Ouko, Tal Pe'er, Ralf Pichler, Lloyd Roden, Nils Röttger, Angelina Samaroo, Yaron Tsubery

作者特此将版权转让给国际软件测试认证委员会（ISTQB®）。本大纲作者（当前的版权所有者）和ISTQB®（未来的版权所有者）一致同意下面的使用条款：

本大纲的作者和ISTQB®是公认的原始发起者和版权拥有者，只有在具备ISTQB®理事会认可的国际认证委员会官方授权的前提下，个人或培训公司才可以使用本课程大纲作为培训教程的理论依据。只有在声明承认本大纲作者和ISTQB®是本大纲的原始发起者和版权所有者的情况下，个人和培训公司才可以使用本大纲作为培训课程的基础。如果需要对涉及本大纲的培训材料做广告，那么这些培训材料需要获得ISTQB®认可的国家认证委员会的授权。

在声明承认大纲的作者和ISTQB®作为本大纲的原始发起者和版权拥有者的前提下，个人或团体可以使用本课程大纲作为文章、书籍或其它资料的参考文献或者主要理论依据。

任何经ISTQB®认可的国家认证委员会可以翻译本大纲，同时可以将本大纲（或翻译后的版本）授权给其它组织。

修订历史

版本	日期	备注
Alpha	2018年5月11日	Alpha版本发布
Beta	2019年1月27日	Beta版本发布
GA	2019年3月28日	GA版本发布
V2019	2019年5月3日	ISTQB®发布

目录

修订历史.....	3
目录	4
致谢	6
0.介绍	7
0.1 本文档的目的	7
0.2 移动应用测试认证基础级	7
0.3 业务收益.....	7
0.4 可考核的学习目标	8
0.5 实践能力水平	8
0.6 考试	9
0.7 培训时间建议	9
0.8 报考要求.....	9
0.9 信息资料来源	9
1. 移动世界-业务和技术驱动力-175分钟	10
1.1 移动分析数据	11
1.2 移动应用的商业模式.....	11
1.3 移动设备类型	12
1.4 移动应用类型	12
1.5 移动应用架构	14
1.6 移动应用的测试策略.....	15
1.7 移动应用测试的挑战.....	16
1.8 移动应用测试中的风险	17
2.移动应用测试类型-265分钟	18
2.1 测试与硬件设备的兼容性	20
2.2 应用程序与设备软件的交互测试	23
2.3 测试各种连接方法	26
3.移动应用的常见测试类型和测试过程-200分钟	27
3.1 适用于移动应用的常用测试类型	28
3.2 适用于移动应用程序的额外测试级别	31
3.3 基于经验的测试技术.....	32
3.4 移动测试过程和方法.....	35
4.移动应用平台、工具和环境-80分钟	37
4.1 移动应用的开发平台.....	37

4.2	通用开发平台工具	37
4.3	仿真器和模拟器	38
4.4	建立测试实验室	39
5.	自动化测试执行-55分钟	40
5.1	自动化途径	40
5.2	自动化测试方法	41
5.3	自动化工具评估	42
5.4	建立自动化测试实验室的方法	42
6.	参考资料	44
6.1	ISTQB®文档	44
6.2	参考书目	44
6.3	其他书籍和文章	44
6.4	链接（网络/互联网）	45
7.	附录A-学习目标/知识认知水平	46
7.1	第一级: 牢记(K1)	46
7.2	第二级: 理解(K2)	46
7.3	第三级: 应用 (K3)	46
8.	附录B-领域专业术语表	48

致谢

本文档由国际软件测试认证委员会移动应用测试工作组的核心团队编写。

Vipul Kocher (主席), Piotr Wicherski (副主席), José Díaz, Matthias Hamburg, Eran Kinsbruner, Björn Lemke, Samuel Ouko, Tal Pe'er, Ralf Pichler, Lloyd Roden, Nils Röttger, Angelina Samaroo, Yaron Tsubery

核心团队感谢评审团队的意见和投入。

下列人员参与了本课程大纲和其前作的评审，建议和表决：

Graham Bath, Veronica Belcher, Armin Born, Geza Bujdoso, YongKang Chen, Wim Decoutere, Frans Dijkman, Florian Fieber, David Frei, Péter Földházi Jr., Chaonian Guo, Attila Gyuri, Ma Haixia, Matthias Hamburg, Zsolt Hargitai, Hongbiao Liu, Ine Lutterman, Marton Matyas, Petr Neugebauer, Ingvar Nordström, Francisca Cano Ortiz, Nishan Portoyan, Meile Posthuma, Emilie Potin-Suau, Liang Ren, Lloyd Roden, Chaobo Shang, Mike Smith, Péter Sótér, Marco Sogliani, Michael Stahl, Chris Van Bael, Paul Weymouth, Salinda Wickramasinghe, Minghui Xu

基础级-移动应用测试大纲中文翻译参与者（按姓氏拼音排序）：

陈少锐、戴兵、肖焕阳（组长）

基础级-移动应用测试大纲中文QA评审参与者（按姓氏拼音排序）：

崔哲（QA组长）、刘晓更、陶显锋

致谢企业：广东农信



该文档由ISTQB®于2019年5月3日正式发布。

0.介绍

0.1 本文档的目的

本大纲是移动应用测试认证基础级水平的中文版课程大纲。国际软件测试认证委员会（以下简称 ISTQB®）提供标准的课程大纲：

1. 给各个成员国委员会，翻译成本国语言并且授权给培训机构。成员国委员会可以根据他们特定的语言调整教学大纲，以及引用当地出版物。
2. 给考试机构，根据本大纲的学习目标，出当地语言的考试题。
3. 给培训机构，编制课件和决定相应的授课方法
4. 给认证考试应试者，准备认证考试（认证考试可以是作为培训课程一部分或独立准备）。
5. 给国际软件和系统工程界，促进软件和系统测试的专业化，并且作为书籍和文章的基础之一

ISTQB®允许其他组织为其他目的而使用本大纲，只要他们事先获得 ISTQB®书面许可。

0.2 移动应用测试认证基础级

基础级资格认证的目标对象是从事与软件测试工作相关的，希望拓宽移动应用测试知识的人员，或想要开始专职从事移动应用测试的人员。

在制定本大纲时，ISTQB®认证测试工程师基础级大纲[ISTQB_CTFL_2018]中描述的相关移动应用测试信息已考虑在内。

0.3 业务收益

本节列出了已获得基础级移动应用测试认证的考生的预期业务收益。

MAT-01 理解并审查移动应用的业务和技术驱动因素，以便制定测试策略。

MAT-02 识别并理解与移动应用程序测试相关的关键挑战、风险和期望。

MAT-03 应用移动应用程序特有的测试类型和级别。

MAT-04 将常见的测试类型，如[ISTQB_CTFL_2018]中提到的测试类型，应用于特定的移动环境中。

MAT-05 将移动应用测试所需的具体活动作为 ISTQB®测试过程所述的主要活动的一部分来开展；

MAT-06 识别和使用合适的环境和适当的工具进行移动应用测试。

MAT-07 了解专门用于支持移动应用测试自动化的方法和工具。

0.4 可考核的学习目标

学习目标可以支持商业价值，并且同时用于生成认证移动应用测试员基础级的考试题。学习目标根据 K-level 进行分配。

K-等级，或者说认知水平等级，是根据布鲁姆[安德森 2001]修订的分类方法来对学习目标进行分类。ISTQB®用这种分类方法设计考试大纲。

本大纲包含了三个不同的 K 级（K1 至 K3）。详见第七章。

0.5 实践能力水平

移动应用测试基础级引入了注重实践技能和能力培养的动手操作目标的概念。

实践能力可以通过实际操作来实现，如以下非详尽清单中所示：

- 如同现有的各种 ISTQB®教学大纲一样，使用纸和笔或文字处理软件进行 K3 级学习目标的练习。
- 搭建及使用测试环境。
- 在虚拟和物理设备上测试应用程序。
- 在桌面设备和/或移动设备上使用工具测试或协助测试相关任务，如安装、查询、记录、监控、截图等。

以下级别适用于实操目标：

- H0：包括练习的现场演示或录制的视频。由于这不是由学员进行的，因此严格来说不是练习。
- H1：引导练习。学员按照教练员的步骤顺序进行练习。
- H2：带提示的练习。给予学员一个带有相关提示的练习，使练习能够在规定的时间内完成。
- H3：没有提示的无指导练习。

建议：

- K1 的学习目标一般采用 H0 级，情况需要时采用 H1 或 H2。
- K2 学习目标一般采用 H1 或 H2 级别，情况需要时采用 H0 或 H3。

K3 学习目标一般使用 H2 或 H3 级别，不过 K3 学习目标不一定要有实操练习。如果实操练习设置复杂，或者耗时过长，则使用 H0 级别。

0.6 考试

本教学大纲是移动应用测试基础级认证考试的基础。

考试中涉及到的问题，可能需要用到本大纲的一个甚至多个节的知识。考试的范围覆盖本大纲的所有节，除了简介和附录。标准、书籍和其他 ISTQB®大纲只是作为参考，但其内容除了从这些标准、书籍和其他 ISTQB®大纲中总结出的内容之外，是不作为考试范围的。

移动应用测试基础级考试形式为选择题。共有 40 道题。要通过考试，至少必须正确回答 65% 的问题（即 26 个问题）。实操目标和练习将不作考察。

考试可以作为经认证的培训课程的一部分或独立进行（例如，在考试中心或公开考试）。完成经认证的培训课程不是参与考试的先决条件。

如果有考生想在没有接受认证培训机构培训的情况下参加考试，应阅读认证和能力准则文件[CTFL-MAT-2019-Accreditation-and-Competence-Guidelines.pdf]中提供的的能力准则，并尝试自己进行这些实操练习。这将有助于考生获得经认证的培训机构所传授的能力。需注意的是，这对移动应用测试基础级认证考试没有任何影响，因为考试只基于本大纲及其学习目标。

0.7 培训时间建议

本大纲定义了每个学习目标的最小培训时长，每章建议的总授课时间标记在章节标题后。

培训机构需注意，其他 ISTQB®大纲根据不同的 K 等级分配了固定的标准培训时长。本移动应用测试大纲无需严格采用这种形式。培训机构可根据每一学习目标的最小培训时长更灵活更实际地调整实际的培训时长。

0.8 报考要求

获得 ISTQB®基础级认证的考生方可报名参加本考试。

0.9 信息资料来源

本课程大纲中使用的术语，在 ISTQB®软件测试术语表[ISTQB®_GLOSSARY]文档中做了详细定义。

第六章节列出了一些关于移动应用测试的推荐书籍和文章。

1. 移动世界-业务和技术驱动力-175分钟

关键词

风险分析, 风险缓解, 基于风险的测试, 测试策略

业务和技术驱动力的学习目标

1.1 移动分析数据

MAT-1.1.1 (K2) 描述如何将可用的移动分析数据用作测试策略和测试计划的输入

HO-1.1.1 (H3) 根据从一个或多个分析数据源(地理位置、平台、操作系统版本和设备类型分布)收集的数据, 选择需要测试的设备类型及其相应的优先级。

注: HO-1.1.1 和 HO-1.7.1(参见下文)可合并。

1.2. 移动应用的商业模式

MAT-1.2.1 (K2) 区分移动应用的各种商业模式。

1.3 移动设备类型

MAT-1.3.1 (K1)回顾不同类型的移动设备。

1.4 移动应用类型

MAT-1.4.1 (K2) 区分不同类型的移动应用。

1.5 移动应用架构

MAT-1.5.1 (K2)区分不同通用架构类型的移动应用

1.6 移动应用的测试策略

MAT-1.6.1 (K3) 运用移动应用市场的特点和具体情况来准备测试策略

1.7 移动应用测试面临的挑战

MAT-1.7.1 (K2) 举例说明与移动应用程序测试相关的挑战

HO-1.7.1 (H1) 收集市场数据, 如设备或选定区域的操作系统市场份额。收集屏幕尺寸和密度的数据。生成一个五种设备的清单, 并计算该清单的预期市场覆盖率。

注: HO-1.1.1(参见上文)和 HO-1.7.1 可以合并。

1.8 移动应用测试中的风险

MAT-1.8.1 (K2) 说明如何降低移动应用特有的风险。

1.1 移动分析数据

移动世界中许多利益相关者，包括制造商、平台提供商、操作系统（OS）提供商、市场数据提供商、工具提供商，当然还有应用开发商和测试人员。

为了有效地促进测试计划讨论和测试分析，移动应用测试人员应了解和熟悉以下因素：

- 分销平台的商业影响
- 每个平台的应用下载量
- 操作系统版本的数量和分布
- 各种设备类型的市场分布情况，包括基于地理位置的变化。
- 不同的屏幕尺寸和分辨率
- 不同的输入方法
- 摄像头的种类

上述信息有几个来源，包括免费的和商业的。这些来源包括 StatCounter GlobalStats[URL1]、操作系统供应商本身和其他第三方来源。

移动分析数据用于选择适合目标市场的设备组合进行测试执行。根据设备的重要性在设备上测试应用程序。与设备及其特殊功能（如果有的话）相关的数据也可以用来设计针对设备类型的测试。例如，带有心跳传感器的设备可能需要特殊的测试案例。

1.2 移动应用的商业模式

有几种模式可用于给创建移动应用程序的工作带来价值。这些模式包括但不限于：免费增值、基于广告的、基于交易的、基于收费的和企业应用。此外，应用内购买也可应用于其中一些模式。

这些方法都有一定的优势和劣势，测试人员在测试移动应用时，应该牢记这些商业模式。

在免费增值模式中，应用程序本身通常是免费的，但如果用户需要使用额外的功能，则必须付费。应用程序需要提供足够的功能以吸引用户，同时提供大量用户愿意付费的高级功能。

基于广告的应用程序会在用户与应用程序交互时在屏幕上显示广告。如果应用程序的使用时间相对较长，这种创收策略就比较有效。用户界面设计者在显示广告时必须注意，它们必须足够突出，且不遮挡应用程序的基础部分，同时他们必须确保用户不会因广告分心以及不喜欢使用应用程序。

基于交易的应用程序向用户收取费用，模式包括每笔交易收费、固定费用或交易价值的百分比或类似方式。这类商业模式只适合数量有限的应用，通常适用于商业和金融应用，如移动钱包。

收费型应用程序要求用户为下载和安装应用程序付费。决定采用收费的商业模式时，应该经过深思熟虑，因为大多数应用类型都存在大量的免费或免费增值选择。如果这种应用提供了出色的功能或可用性，或者在没有竞争性应用的情况下，用户购买这种应用的概率就会增加。

免费和企业应用程序不向用户收费。企业应用程序是为组织内部使用而开发的，为所提供的服务提供接口。银行或电子商务公司等机构有许多这样的应用程序。这些应用一般不注重应用本身的盈利能力，而是通过引导用户使用企业提供的服务来获得收入。

目前有很多种移动设备可以支持不同类型的应用。

典型设备包括：

1.3 移动设备类型

- 基础型手机
- 功能手机
- 智能手机
- 平板电脑
- 辅助设备--包括可穿戴设备和一些 IoT（物联网）设备。

在测试时，应牢记每种类型的设备都有特定的功能，以满足特定的需求。

基础型手机仅用于电话和短信，提供极少的内置应用程序和游戏。无法安装应用程序或浏览网页。

功能手机提供有限的应用程序和应用程序安装支持。它们通过内置的浏览器提供互联网访问，并可能拥有一些额外的硬件，如相机。

智能手机指带有多个传感器的手机。操作系统支持应用程序安装、多媒体和浏览等功能。

平板电脑与智能手机类似，但体积更大。通常在需要或期望更大的显示屏的情况下会使用平板电脑，同时平板电脑的电池寿命更长。

辅助设备和一些物联网设备是由计算机驱动的设备，通常与智能手机或平板电脑一起使用，用以扩展可用功能或以更方便的方式访问手机或平板电脑上的数据。

可穿戴设备是指可以由消费者佩戴的设备。这些设备可以作为现有设备的辅助设备，也可以独立运行。流行的可穿戴设备的例子有手表和健身手环。

1.4 移动应用类型

移动应用主要有三种类型。

- 本地型
- 基于浏览器
- 混合型

每种类型的应用都有一定的优势和劣势，需要在应用开始开发前做出商业决策。

本地应用程序是使用平台特定的软件开发包(SDK)、开发工具和平台特定的传感器和特性开发的，它们可从供应商商店下载、安装和更新。这些应用程序可能需要在所有支持的设备上进行测试。

本地应用程序一般性能更佳，可充分利用平台特性，并符合他们所开发平台的期望。开发成本通常较高，并可能面临额外的挑战，如使用多个平台以及在大量设备上安装和测试。

基于浏览器的应用是通过手机浏览器访问的。由于这些应用使用典型的网络开发技术和浏览器，所以容易支持多个平台，且开发成本通常较低。

移动网络应用的创建方式主要有四种。

- 网站和应用程序的特定移动版本（这些也被称为 **m(.)** 网站）。通常而言，当移动浏览器访问应用程序时，提供了移动版本的应用程序。例如，当从移动设备访问 **facebook.com** 时，会重定向到 **m.facebook.com**。
- 响应式网络应用确保设计能适应形状系数和屏幕大小，通常以视图端口的形式表示。
- 自适应式网络应用会根据一些预定义的尺寸对设计进行调整。这些尺寸有不同的设计，用户可使用的功能通常是可以调整的。
- 渐进式网络应用允许在移动主屏幕上创建特定网页的快捷方式。它们看起来像本地应用程序，有时甚至可以离线工作。

移动网络应用是使用通用网络技术创建的，通常这使其比本地和混合应用更容易开发和管理。然而，它们可能不像本机或混合应用程序那样功能丰富，而且对平台的原生应用程序编程接口（API）的访问有限，同时对移动传感器的访问也受到限制。移动网络应用不需要对设备进行可安装性测试，但需要进行浏览器兼容性测试。

混合应用程序是本地应用程序和网络应用程序的结合。它们使用包含网络视图的原生应用包装器，在本机应用中运行网络应用。这些应用程序从供应商商店下载，可以访问所有的设备功能。它们相对容易开发、更新和维护，而且无需更新设备上安装的应用程序。开发这些应用程序所需的技能几乎与网络开发相同。这些应用程序可能存在的缺点包括由于使用包装器而导致的性能问题，以及由于平台特定方面的原因，可能与预期的观感有差异。

本地和混合应用程序是实际安装在设备上的，因此即使设备没有互联网连接，用户也可以随时使用。相比之下，基于浏览器的应用则需要访问互联网。

有些应用是预装在移动设备上的，有些应用则可以通过各种分销渠道安装，如苹果应用商店、Google Play 商店、企业应用商店（仅在企业网络内部提供）和第三方应用市场。

对于不同应用程序的测试可能需要不同的方法。需要考虑的参数包括：

- 支持不同类型的设备
- 需使用的传感器和设备功能
- 各种网络条件下的可用性
- 可安装性、兼容性、性能效率和易用性。

1.5 移动应用架构

设计一个移动应用有多种解决方案。

在选择特定的架构或设计决定时，需要考虑的因素包括：

- 目标受众
- 应用类型
- 对各种移动和非移动平台的支持
- 连接需求
- 数据存储需求
- 与其他设备的连接性，包括物联网设备。

架构决策包括：

- 客户端架构，如瘦客户机或胖客户机。
- 服务器端架构，如单层或多层。
- 连接类型，如 Wi-Fi、蜂窝数据、近距离无线通信(NFC)、蓝牙。
- 数据同步方法，如存储转发、拉取推送、同步和异步通信等。

瘦客户端应用程序不包含针对设备定制的应用代码，并尽量减少对移动操作系统功能的使用。这些应用程序通常使用网络浏览器作为前端，使用 JavaScript 作为实现客户端逻辑的语言。

瘦/厚客户机应用程序可能有多层应用程序代码，并可能使用移动操作系统功能。这些是典型的本地或混合应用程序。

服务器端架构包括以下几种可能：

- 单层架构是整体的，所有服务器都在同一台机器上。它们的可扩展性较差，而且难以保证安全。
- 多层架构将服务器端组件分布在不同的单元中。两层架构涉及单独的网络和数据库服务器，而三层架构也包括一个应用服务器。多层架构允许职责分离，提供数据库专业化，并提供更好的灵活性、可扩展性和安全性。然而，与单层架构相比，多层架构的开发、管理和托管费用可能要高得多。

连接方法有很多种。移动设备可通过 Wi-Fi 等连接类型或通过 2G、3G、4G 和 5G 等蜂窝数据连接到服务器。移动应用通常以以下三种模式之一运行。

- 从未联网的应用程序可以离线工作，不需要连接。一个简单的计算器就是这样一个应用的例子。
- 始终联网的应用程序在运行过程中需要永久的网络连接。所有的移动网络应用都属于这一类，尽管有些应用在部分连接时也可以有限的方式运行。
- 部分联网的应用程序需要连接才能完成数据传输等任务，但在没有连接的情况下也可以长时间运行。

客户端和服务器之间的数据同步可以通过以下模式进行。

- 连续模式指的是数据一提交就会被传输。

- 存储和转发模式是指数据在传输前可能被存储在本地，特别是在未连接的情况下。

数据传输可以通过以下两种方式进行。

- 当调用函数在返回之前等待被调用函数完成时，会进行同步数据传输。
- 异步数据传输时，被调用的服务器函数立即返回，在后台处理数据，完成任务后再回调调用客户端函数。这样可以给用户更多的控制权。然而，在实现过程中，当服务器发起回调时，握手机制增加了有关客户端或网络可用性的复杂性。

1.6 移动应用的测试策略

为移动设备制定测试策略，要求测试人员需考虑到迄今为止本章列出的所有参数。此外，还必须考虑本节讨论的风险和第 1.7 节所述的挑战。

典型风险，例如：

- 在不知道扩散数据设备的具体地理位置情况下，无法选择需要在哪些设备上持续测试应用。
- 在不知道业务模式的类型的情况下，无法测试应用程序的行为是否适合该业务模式。

创建移动应用测试的测试策略还需要考虑以下具体的风险和挑战。

- 种类繁多的移动设备，部分设备上存在特定缺陷。
- 内部设备的可用性或通过使用外部实验室测试
- 在应用生命周期内引入新技术、设备和/或平台。
- 通过各种渠道对应用本身进行安装和升级，包括保留应用数据和参数选项。
- 可能影响应用的平台问题。
- 网络覆盖及在全球范围内对应用的影响。
- 使用各种服务提供商的网络进行测试的能力。
- 使用移动仿真器、模拟器和/或真实设备进行特定的测试级别和类型的测试。

这些挑战在第 1.7 节中有更详细的描述。

测试策略需要考虑到的风险和挑战。如：

- 测试策略可规定在开发的早期阶段使用移动仿真器/模拟器，然后在后期阶段使用真实设备。某些类型的测试可在移动仿真器/模拟器上进行，但不是所有类型的测试都适用此种情况。关于这一点的更多情况，将在第 4.3 节中说明。
- 测试策略需考虑大量不同的设备带来的挑战，采用以下方法之一：
 - 单一平台方法：将范围缩小到单一类型的设备、操作系统版本、运营商和网络类型。
 - 多平台方法：根据移动流量或其他分析数据，将范围缩小到选择目标市场上具有代表性的、大多数客户所使用的设备和操作系统
 - 最大限度的覆盖方式：覆盖所有操作系统版本、设备、制造商、运营商和网络类型。基本上

可称为穷举测试，这通常在经济上是不可行的，特别是考虑到市场上的众多设备和操作系统版本。

- 测试策略需利用外部资源，考虑到设备、网络或现实生活条件不可用所带来的挑战，如：
 - 远程设备访问服务。这是一种通过网络访问不属于自己的设备的方式。
 - 众测服务。这是一种可以接入庞大的志愿者群体及其设备的方式。
 - 类似朋友或同事等的个人网络。这种方式是利用自己的私人社交网络。
 - 捕捉缺陷。这是一个使用公司的志愿者或公众志愿者的游戏化的测试活动。

除了[ISTQB_CTFL_2018]中描述的测试级别外，测试策略还考虑了用于移动应用的常见测试类型(见第 3.1 节)和所需的其他附加的测试级别(见第 3.2 节)。

1.7 移动应用测试的挑战

在移动世界中，存在着许多额外的挑战，这些挑战在桌面软件或服务器软件中是不常见或不重要的。测试人员必须觉察到这些挑战以及它们是如何影响应用程序成功的。

移动世界的典型挑战包括：

- 多平台和设备碎片化：多种操作系统类型和版本，屏幕尺寸和显示质量。
- 不同设备的硬件差异：各种不同的传感器以及因 CPU 和 RAM 资源受限的测试环境所带来的困难。
- 平台所需的各种软件开发工具。
- 用户界面设计和用户体验（UX）预期与平台的差异。
- 多种网络类型和供应商。
- 设备的资源匮乏。
- 应用的不同发布渠道。
- 多样化的用户和用户群体。
- 具有不同连接方式的不同应用类型。
- 由对用户影响很高的错误所导致的高反馈越明显，这可能很容易导致他们在在线市场上发布反馈。
- 在诸如 Google Play 或苹果 App Store 等市场所有者的应用发布需要额外的审批周期。
- 无法使用移动仿真器/模拟器进行模拟的新设备。

这些挑战的影响包括：

- 大量的组合需要测试。
- 测试需要大量的设备，导致成本上升。
- 需要向下兼容，以便在旧版本的平台上运行应用程序。

- 每一个版本的底层操作系统都会发布新功能。
- 不同平台所需要考虑的准则。
- 资源匮乏的 CPU 以及有限的内存和存储空间。
- 不同网络的带宽和跳动变化。
- 基于数据计划更改可用的上传和下载速度。

以下两个例子说明了典型的挑战及其潜在影响。

- 不同的设备有不同类型的传感器，测试时需要考虑到这些因素。每一个新加入硬件的传感器都可能需要额外的向下兼容性测试。
- 通过使用适当的缓存和预取策略，即使在不同的网络条件下，也可以适当地应对一些网络挑战。不过，这也是有代价的，由于大多数应用都会让用户在服务器上保持登录状态，因此大量的开放连接会影响服务器端的性能。

1.8 移动应用测试中的风险

1.7 节中提到的挑战可能单独出现，也可能与其他挑战一起出现。这可能会给移动应用带来额外的风险。

测试人员必须能够为产品风险分析做出贡献。[ISTQB CTFL_2018]第 5.5 章中讨论的常见风险分析和缓解方法也可应用于移动环境。此外，还存在以下针对移动应用的风险和缓解措施：

风险	可能的缓解措施
市场分化	选择合适的设备进行测试执行，例如，测试最常用的设备。
多个平台的成本测试费用支持	进行分析，了解最常用的平台，从而避免对那些已不在范围内的平台进行测试。
新技术、平台和设备的引进	使用这些技术的预生产版本。
缺乏用于测试执行的设备	运用远程设备接入服务或众测服务。
远程使用移动应用的预期使用模式带来的风险	采用适当的测试方法，如现场测试

2.移动应用测试类型-265分钟

关键字

共存性，兼容性，连通性，跨浏览器兼容性，互操作性，被测系统（SUT），测试类型，易用性

移动应用测试类型的学习目标

2.1 设备硬件的兼容性测试

MAT-2.1.1 (K2)描述测试时应考虑的设备的特征和硬件

HO-2.1.1 (H1)在多个移动设备上测试应用程序及验证被测系统的功能是正确的

MAT-2.1.2 (K3)准备对应用程序的屏幕尺寸、宽高比和屏幕密度的兼容性测试。

HO-2.1.2 (H3)在多个移动设备上测试一个应用程序(虚拟或物理)，以显示分辨率和屏幕大小对应用程序用户界面的影响。

MAT-2.1.3 (K2)描述测试如何发现设备过热对被测系统的潜在影响。

MAT-2.1.4 (K1)回顾不同的测试类型，用于测试移动设备中使用的各种输入传感器。

MAT-2.1.5 (K1)回顾对各种输入方法进行的测试；

HO-2.1.5 (H0) 测试一个应用程序的各种类型的输入，包括安装多个键盘的键盘相关测试，手势相关测试和（可选）摄像头相关测试。

MAT-2.1.6 (K2)描述当改变屏幕方向时，如何测试用户界面显示问题。

HO-2.1.6 (H3)测试一个应用程序，检查方向变化对应用程序功能的影响，包括数据保留和用户界面的正确性。

MAT-2.1.7 (K3)为使用典型移动设备中断的应用程序准备测试。

HO-2.1.7 (H3)测试一个应用程序在使用过程中的多个移动设备中断。

MAT-2.1.8 (K3) 为改变应用程序所要求的设备功能的访问权限准备测试。

HO-2.1.8 (H3)通过允许和拒绝请求的权限来测试应用程序的权限管理，并观察安装时拒绝文件夹和传感器权限设置或安装后更改权限的行为。

MAT-2.1.9 (K3)为验证应用程序对设备功耗的影响以及其电源状态对应用程序的影响。

HO-2.1.9 (H3)在不同的电池电量下测试应用程序，找出消耗数据，测试低电量和无电状态下的表现。

2.2. 测试应用程序与设备软件的交互

MAT-2.2.1 (K3)准备测试，检验测试系统如何处理通知

HO-2.2.1 (H2)测试应用在前台和后台时接收通知的效果。测试更改通知设置对应用功能的影响。

MAT-2.2.2 (K2)描述测试如何验证快速访问链接的正确功能。

HO-2.2.2 (H3) 测试一个应用程序的快捷方式/快速访问功能。

MAT-2.2.3 (K3) 为操作系统提供的用户偏好设置对应用程序的影响准备测试。

HO-2.2.3 (H3) 通过改变操作系统提供的偏好设置的输入值选项来测试正在运行的应用程序。

MAT-2.2.4 (K2) 区分原生、web 和混合应用所需的不同测试。

HO-2.2.4 (H0) (可选) 根据应用类型，确定应用所需的测试。

MAT-2.2.5 (K1) 回顾在多个平台或操作系统版本上提供的应用程序所需的测试。

MAT-2.2.6 (K1) 回顾与其他应用程序共存和互操作性所需的测试。

2.3 测试各种连接方法

MAT-2.3.1 (K2) 总结连通性测试，包括切换网络、使用蓝牙、切换飞行模式。

HO-2.3.1 (H0) (可选) 当手机根据可用信号强度在 Wi-Fi 和手机数据连接之间切换时，对正在向服务器传输数据的应用程序进行测试。

2.1 测试与硬件设备的兼容性

2.1.1 设备功能测试

不同类型的设备具有不同的功能，意味着必须对大量设备进行兼容性测试。这就需要对测试的目标设备进行优先排序。对于优先级市场数据，正如 1.1 节所讨论的，用于选择最适合目标市场的设备组合。设备组合的选择通常是市场覆盖率、成本和风险之间的折中。

应用程序可以安装在具有以下功能的不同类型的设备上：

- 不同的关机方法
- 不同的导航方式
- 硬键盘和软键盘的使用
- 各种硬件功能，如：
 - 无线电/收音机
 - USB
 - 蓝牙
 - 摄像头
 - 扬声器
 - 麦克风
 - 耳机接口

这些功能都不应该对应用程序的操作产生负面影响。

设备功能有许多变化，甚至在同一制造商生产的不同设备型号之间也会有所不同。它们通常被用于区分不同的细分市场，并且会随着时间的推移会迅速发生变化。例如，目前高端和中端设备有指纹传感器是很常见的，而低端设备则没有。这种情况会随着时间的推移而发生变化。几年前，指纹传感器根本不包含在任何移动设备中。由于这种变化性，测试人员需要清楚地了解设备及其用户所期望的特性。测试人员需要创建设备组合，并据此设计相应的测试。

一般来说，仅仅测试应用程序是否能正确使用预期的功能是不够的。此外，还需要测试如果某项功能缺失，应用程序是否仍能按照预期工作。

例如，一个支持使用前置和后置摄像头的应用程序，如果安装并执行在有多摄像头、只有一个摄像头或根本没有摄像头的设备上，都应该不会崩溃。

2.1.2 测试不同的显示器

设备显示器有不同的屏幕尺寸、视角尺寸、宽高比和分辨率，以每英寸像素（ppi）和每英寸点数（dpi）来衡量。设备碎片化需要优先进行处理。应该在目标市场最常见的不同屏幕尺寸、分辨率和宽高比的各种设备上运行用户界面，创建测试。

需要针对不同的显示器进行测试，检查以下内容：

- 应用根据当前的屏幕密度和大小，对所有用户界面元素进行缩放。
- 用户界面元素不重叠。
- 不发生易用性或触摸问题。
- 不会因为高 dpi/ppi 而出现图像收缩的问题。

2.1.3 设备温度测试

与台式电脑不同，移动设备对设备温度升高的反应不同。

移动设备过热的原因有很多，比如电池充电、高强度工作负荷、后台运行的应用程序、连续使用蜂窝数据、Wi-Fi 或 GPS。

过热可能会影响设备，因为设备会试图减少发热和保护电量。这些操作可能导致包括 CPU 频率下降、释放内存和系统的部分功能关闭。

如果发生这种情况，也会影响到应用程序的功能，因此在规划测试时必须考虑到这一点。测试的设计必须消耗大量的电量，从而导致设备在长时间，不间断的时间内产生热量。然后，被测试的软件必须无异常行为。

2.1.4 设备输入传感器的测试

移动设备从使用 GPS、加速器、陀螺仪和三轴磁力计等传感器或对压力、温度、湿度、心跳、光线或非接触式输入作出反应的传感器中接收各种输入类型。

对不同设备的输入传感器进行测试，检查以下内容。

- 应用中每个可用的传感器是否都能按预期工作。例如，该应用程序需要测试各种类型的运动，如圆周运动和前后运动（在步行中）。
- 对外部光照有反应的功能在各种光照条件下都能正确反应。
- 声音输入和输出与软硬音量按钮、麦克风、有线和无线扬声器以及在各种环境声音条件下都能正确响应。
- 在以下条件下，位置是准确的：
 - 打开或关闭GPS
 - 不同的GPS信号质量
 - 应用是否需要备选方法确定位置，包括Wi-Fi、信号塔位置或手动输入位置。

2.1.5 各种输入方法测试

测试设备中不同的输入方法，检查以下内容：

- 考虑到手机允许安装各种软键盘，该应用应至少能够与各大设备厂商提供的软键盘以及被广泛使用的软键盘配合使用。
- 应用应确保在需要时，键盘会默认弹出相应的布局和按键。
- 当用户将一个或多个手指放在触摸屏上时，应用程序将该模式解释为一个特定的手势或命令。典型的手势包括按/触摸、双触摸、多点触摸、滑动、点击、双点击、拖动和放大/缩小。
- 应用程序的每一屏都需要正确响应适合该屏幕的手势或其他输入方式，并忽略所有不支持的手势或输入。
- 应用程序使用的摄像头能够捕捉图像和视频，扫描条形码、二维码和文档，并可测量距离。
- 在有前置和后置摄像头的情况下，会默认开启相应的摄像头。例如，在视频聊天需要默认开启前置摄像头的情况下，需要对应用使用摄像头输入和不使用摄像头输入的情况进行测试。此外，测试必须确保被测软件在只有一个（前置或后置）摄像头而不是两个摄像头的情况下正常工作。如果被测软件使用了一个特定的摄像头，而缺少的正是这个摄像头，则更应如此。

2.1.6 屏幕方向变化测试

移动传感器用于检测方向的变化，并触发横向和纵向模式之间的切换（反之亦然），并根据需要在用户界面上进行布局更改。

测试改变屏幕方向后的应用，检查以下内容：

- 当切换到纵向或横向模式时，正确的使用性和功能表现。
- 应用程序保持其状态。
- 输入数据字段保留已采集的数据。
- 输出数据字段在保持当前会话的同时显示相同的数据。

屏幕方向改变后的测试不应该只关注单一的切换，因为渲染或状态问题不一定会在单一的切换后显示出来。因此，应该在纵向和横向模式之间进行多次不间断的切换来进行测试。

应该在用户界面的各种状态下设计多次切换方向的测试，有数据和无数据。应用程序应该如预期一样的运行，保持状态，不丢失或更改任何数据。

2.1.7 典型中断的测试

常见的设备中断类型包括语音呼叫、信息、充电器开启、低内存或其他通知。用户发起的中断来自于应用程序切换或在应用程序运行时将设备设置为待机操作引起的。

对中断测试进行一下检查，检查下列内容：

- 该应用正确处理了上述所有的中断，同时不会对应用行为产生负面影响。
- 无论哪个中断发生，应用程序都能继续正常运行，保持其状态、数据和会话。

- 如果设备具有可抑制通知的“请勿打扰”的阻断模式，应用程序必须确保可以正确使用各种条件。当关闭长时间处于“请勿打扰”活动状态的模式时，也必须进行这些测试。以上将导致设备会同时收到许多通知。
- 测试应该针对应用使用过程中接收中断的情况进行设计，确保中断不会产生负面影响。例如，用户在使用 app 的时候接听电话，又回到中断时的状态不应受到影响。

2.1.8 设备功能访问权限测试

应用程序需要访问各种文件夹，如联系人和图片，以及相机和麦克风等传感器。如果在安装时拒绝访问或安装后改变访问权限，可能会影响应用程序的行为。

测试访问权限，检查以下内容：

- 应用程序可以在减少权限的情况下工作；可要求用户授予这些权限，并且不会以不明原因的方式失败。
- 权限仅授予与应用程序功能相关的资源请求，不允许对不相关的资源有广泛的权限
- 如果在安装过程中权限被撤销或拒绝，应用功能可正确响应。
- 任何由应用程序发出的许可请求都是正确和合理的。

为了测试访问权限，测试人员需要知道为什么应用程序需要每个权限，以及如果在安装过程中撤销或拒绝授权，功能会受到什么影响。应该设计在安装过程中拒绝权限，以及在安装后授予权限的测试。

2.1.9 电量消耗及状态测试

测试电量消耗和状态，检查以下内容。

- 电池电量状态和放电相关缺陷。
- 在低电量和无电状态下的数据完整性。
- 应用活动时的功耗，以及在大量使用和少量使用下的功耗。
- 应用在后台时的功耗。

这些测试需要仔细规划，因为这些测试需要在长时间内不间断地运行。例如，可能需要让设备在没有使用的情况下让后台或前台的应用程序自动运行。需要使用日志分析器等工具来提取有关电池消耗模式的信息。

2.2 应用程序与设备软件的交互测试

2.2.1 测试通知信息

操作系统使用各种机制来显示通知。有时，操作系统为了优化功耗，会延迟显示通知或根本不显示通知。必须考虑以下测试条件。

- 当应用处于前台或后台时，正确处理收到的通知，特别是在电量不足条件下。

- 如果通知允许与应用内容直接交互，（即无需打开应用本身），那么用户交互必须由应用在稍后的时间提供。例如，如果用户对通知做出响应，那么必须能够在以后的时间可以从应用内访问该响应。
- 如果通知允许访问应用程序，那么当通知包含到该页面的深度链接时，必须打开的是应用程序的相应页面而不是主屏幕。

2.2.2 快速访问链接测试

测试中的软件可能会提供快速访问链接，如 Android 中的应用快捷键和 iOS 的 Force-touch 或 3d-touch。这些功能可以从主屏幕上执行应用功能的子集，而无需实际启动整个应用。

必须考虑以下测试条件：

- 当某些功能只在某一特定版本的操作系统上可用时，如果被测系统安装在提供或不提供这些功能的操作系统版本上，该系统都必须表现正确。
- 在快速访问链接中执行的操作在打开时可正确地反映在应用程序中。

2.2.3 对操作系统提供的用户偏好设置测试

操作系统提供给用户的任何偏好（设置）都必须经过测试。如果某个偏好设置不被应用遵守，会给用户带来负面体验。例如，如果设备被设置为静音，应用就不应该播放声音。

必须考虑以下测试条件：

- 用户可以修改典型的偏好选项，如声音、亮度、网络、省电模式、日期和时间、时区、语言、访问类型和通知。
- 应用程序通过相应的行为来遵守设定的偏好。

2.2.4 不同类型应用程序的测试

可根据移动应用的类型进行具体测试（见 1.4 节）。必须考虑以下测试条件：

- 对于原生应用：
 - 设备兼容性
 - 使用设备功能
- 对于混合应用：
 - 应用与设备原生功能的交互。
 - 抽象层带来的潜在性能问题。
 - 与相关平台上的原生应用相比，易用性如何（观感）。
- 对于 Web 应用：
 - 测试以确定应用对各种常见移动浏览器的跨浏览器兼容性。
 - 功能不会因为使用不同版本的JavaScript受到影响

- 使用操作系统的功能（如日期选择器和打开适当的键盘）。
- 与相关平台上的原生应用相比，易用性如何（观感）。

2.2.5 多个平台和操作系统版本的互操作性测试

软件公司的应用通常支持多个操作系统。每个移动操作系统都有自己的限制，在测试应用程序时需要考虑这些限制。测试人员必须了解每个被测试的平台具体情况，以确保应用程序能够按照预期工作，同时仍然符合平台的观感。

必须考虑以下测试条件：

- 处理中断、通知和优化(如节能)。
- 多平台应用程序共享一些代码或使用跨平台开发框架创建的应用功能正确与否。需要注意的是，如果应用程序没有共享代码，那么就好比是在测试两个不同的应用程序，一切都需要测试。
- 如果一个平台使用了不同的操作系统版本，则要进行向下兼容性测试。
- 对平台新的或改变的功能进行测试。例如，在安卓系统中，引入 Doze 框架，需要在支持该框架和不支持该框架的不同版本的操作系统上进行测试。

2.2.6 设备上其它应用程序的互操作性和共存性测试

安装在设备上的应用程序之间的互动是很常见的。典型的例子是联系人和电子邮件应用程序。

必须考虑以下试验条件：

- 被测系统与使用的应用之间的数据传输正确。
- 对存储在应用程序中的任何用户数据都不会造成伤害。
- 相互冲突的行为。例如，一个应用程序可能会关闭 GPS 以节省能源，而另一个应用程序则会自动打开 GPS。

由于市场上有数以百万计的应用程序，不可能实际地对所有这些应用程序进行共存测试。然而，应根据其风险考虑和测试这些潜在的问题。

2.3 测试各种连接方法

移动设备可以使用各种方法连接到网络（见 1.5 节）。这些方法包括 2G、3G、4G 和 5G 等蜂窝网络，以及 Wi-Fi 和其它无线连接类型，如 NFC 或蓝牙。

在进行连通性测试时，应考虑以下备选方案：

- 设备仿真器/模拟器可以模拟各种网络连接，一些远程设备访问服务提供商在其功能中包含了这些功能。但是，仿真器/模拟器的价值有限。
- 配置自己的移动网络以支持各种连接类型，然后控制带宽。这是一个花费很高的选择。
- 实地测试可能是更具成本效益的替代方法，但在测试的复现方面受到限制。

在实际使用中，连接方式不同。用户可以使用一种特定的模式持续连接，也可以在不同的模式之间切换，比如从 Wi-Fi 到蜂窝网络（例如，当用户在使用该应用的过程中离开家）。用户可以在各种 Wi-Fi/蜂窝网络和版本之间切换，也可以在 GSM 手机之间切换。在移动过程中，他们甚至可能会遇到根本没有网络的死角。此外，用户还可以通过切换到飞行模式等方式故意断网。

连接性测试必须确保考虑以下测试条件：

- 在不同的连接模式下，应用的功能均能正确运行。
- 模式之间的切换不会导致任何意外行为或数据丢失。
- 如果由于网络连接有限或没有网络连接，或者带宽较低，功能受到限制，则应向用户提供明确的信息。该信息应说明限制及其原因。

3.移动应用的常见测试类型和测试过程-200分钟

关键字

异常终止, 无障碍性, 代码注入, 探索性测试, 现场测试, 启发式, 可安装性, 性能效率, 性能测试, 发布后测试, 安全测试, 基于会话的测试管理, 压力测试, 测试级别, 测试过程, 测试金字塔, 向导, 易用性实验室, 易用性测试

移动应用通用测试类型和测试过程的学习目标

3.1 适用于移动应用的常见测试类型

MAT-3.1.1 (K3)准备移动应用的可安装性测试

MAT-3.1.2 (K3) 准备移动应用的压力测试

MAT-3.1.3 (K2) 举例说明与移动应用程序相关的安全问题。

MAT-3.1.4 (K1)回顾移动应用的时间和资源行为的注意事项

MAT-3.1.5 (K3) 准备移动应用的易用性测试。

HO-3.1.5 (H2)选择向导、助记符或启发式方法, 使用基于会话的测试管理对一个应用程序进行易用性测试。

注: HO-3.1.5 和 HO-3.3.1、HO-3.3.2 和 HO-3.3.3 可合并使用。

MAT-3.1.6 (K1)识别移动应用数据库测试所需的测试类型。

MAT-3.1.7 (K2)总结移动应用国际化(全球化)和本地化测试所需的测试。

MAT-3.1.8 (K2)总结移动应用测试中进行无障碍测试的必要性。

3.2 适用于移动应用程序的附加测试级别

MAT-3.2.1 (K2) 说明附加的测试级别, 如现场测试, 以及有效的移动应用测试所需的相关附加活动。

MAT-3.2.2 (K2)描述进行应用商店审批发布应用程序所需的测试。

3.3 基于经验的测试技术

MAT-3.3.1 (K1) 在探索性移动测试的背景下, 回顾基于会话的测试管理、角色和助记符。

HO-3.3.1 (H2) 选择一个移动应用测试专用的向导(或其中的一部分), 使用基于会话的测试管理对应用进行测试。

注: HO-3.1.5 和 HO-3.3.1、HO-3.3.2 和 HO-3.3.3 可以一起进行。

MAT-3.3.2 (K2) 描述参观和启发式方法作为移动应用测试的探索性技术的用法。

HO-3.3.2 (H2) 选择一个特定的移动启发式方法来测试移动应用。

注: HO-3.1.5 和 HO-3.3.1、HO-3.3.2 和 HO-3.3.3 可以合并使用。

MAT-3.3.3 (K3)利用移动专用向导(如功能向导)来测试移动应用。

HO-3.3.3 (H2) 选择一个移动特定的向导方法来测试移动应用。

注：HO-3.1.5 和 HO-3.3.1、HO-3.3.2 和 HO-3.3.3 可以合并使用。

3.4 移动测试过程和方法

MAT-3.4.1 (K2)将[ISTQB_CTFL_2018]中描述的测试过程与移动应用测试的需求相匹配。

MAT-3.4.2 (K2)描述各测试级别的测试方法，具体到移动应用测试。

3.1 适用于移动应用的常用测试类型

3.1.1 可安装性测试

测试人员需要使用以下方法关注应用的安装、更新和卸载。

- 应用商店

根据应用的用户情况，安装过程可能会有所不同。用户可以从 Google Play Store 或苹果的 App Store 等市场商店安装应用。企业级应用的用户需要通过链接，或者 HockeyApp 或 App Center 等分发服务进行安装测试。

- 文件安装（复制并安装应用程序）

一些操作系统提供了安装应用程序的选项，将其复制到移动设备上，然后从文件中安装。

- 桌面应用

桌面应用程序，如苹果 iTunes(iOS)或 Android App Installer，可用于在智能手机上安装应用程序。测试人员需要在这个应用程序中下载应用，并使用数据线将其从桌面应用安装到智能手机上。这些桌面应用程序大多还允许卸载应用程序。

可以使用以下方法进行安装。

- 通过 Wi-Fi 或蜂窝数据的 OTA（空中传输）。

- 数据线

可以考虑的一些测试条件包括：

- 在内部和外部存储器上安装、卸载和升级（如果支持）。
- 在上一次卸载选择“保留应用数据”选项时，重新安装应用。
- 在上次卸载没有选择“保留应用数据”选项时，重新安装应用。
- 取消或中断安装或卸载，例如，在安装过程中关闭移动设备或断开互联网。
- 取消或中断后，恢复中断的安装、卸载和升级。
- 权限相关的测试。例如，一些应用程序请求使用通讯录的权限。如果用户拒绝权限，这个重要的测试必须验证应用的行为。例如，是否向用户发送了相应的消息？
- 更新应用程序，并确认没有丢失数据。

一些应用程序需要越狱（iOS）或 root（Android）的设备，这些设备赋予用户对设备的管理权限。大多数平台供应商不支持越狱/root，因为这可能会有法律后果。不需要越狱/root 的应用可能不需要对越狱/root 设备进行测试。

3.1.2 压力测试

压力测试的重点是确定应用程序在承受超出正常负载条件下的性能效率。这种情况下的压力测试只针对移动设备。后台的压力测试在 ISTQB®性能测试大纲([ISTQB_CTFL_PT_2018])中进行了描述，因此进一步信息可以根据需要参考大纲获取。

压力测试考虑的测试条件包括：

- 高 CPU 使用率
- 内存不足
- 低磁盘容量
- 电池压力
- 失效
- 带宽不足
- 过多的用户互动（可能需要模拟现实世界中的网络条件）

其中一些压力条件可以使用诸如 **Monkey** 这样的工具来创建。这是一个命令行工具，通过 **ADB shell** 命令行[URL3]运行，如果可以的话，也可以手动运行，例如使用大文件或其他 CPU 使用率高或内存消耗大的应用程序。

3.1.3 安全测试

由于安全测试是一个复杂的话题，ISTQB®对此有单独的专业大纲[ISTQB_CTAL_SEC_2016]。移动应用的主要安全问题包括：

- 访问设备上的敏感数据。
- 未加密的信息传输或不安全的存储。

安全测试考虑的测试条件包括：

- 测试输入是否有代码注入和溢出。
- 对传输的数据进行加密。
- 对本地存储的数据进行加密。
- 使用后或异常结束后删除临时数据。
- 清除密码字段中的文字。

来自开放网络应用安全项目（OWASP）的十大移动相关漏洞也应该被探讨[URL2]。

3.1.4 性能测试

如果用户安装应用后，应用响应的速度不够快（如小于或等于 3 秒），它可能会被卸载，用户会转而使用其它替代应用。时间和资源消耗方面是一个应用成功的重要因素，性能测试就是为了衡量这些方面。

除了与后台系统和其它移动设备的交互外，还需要在设备本身进行性能效率测试。

整个系统的性能测试应按照测试策略中的定义进行，并不针对移动设备。详情请参考 ISTQB®性能测试专家大纲[ISTQB_CTFL_PT_2018]。

应用程序本身的性能测试应包含最重要的工作流程的时序测试。网上银行应用工作流程的一些例子是："登录"、"更改地址"或"使用 PIN 和 TAN 进行银行转账"。然后，测试人员应将这些时间记录与类似的应用程序进行比较。

除了计时测量外，考虑用户的感知性能也很重要。用户体验会对用户愿意为完成某项功能等待多长时间产生巨大影响。

3.1.5 易用性测试

易用性对于移动应用来说是非常重要的，因为数据显示，大量的用户在安装应用后的几分钟内就会因为易用性或性能不佳而卸载应用，见[URL4]。

因此，建议用户体验(UX)设计要考虑使用应用平台的观感。如果用户体验不符合用户对其选择平台的期望，会产生强烈的负面影响。因此，测试人员应该了解所使用平台的观感。

易用性测试可以由测试人员使用各种可用的启发式方法和向导测试来进行。考虑用户角色也是对易用性测试的一个有益支持。如果需要的话，易用性实验室也可以用于这个目的。

在实际项目中，易用性测试的发现往往只是发现而不是缺陷。测试员必须有能力向团队/产品负责人或其他利益干系人解释这些发现。要达到满意的可用性，应用必须达到：

- 一目了然直观易懂
- 允许用户犯错
- 文字和行为保持一致
- 遵从平台的设计准则
- 在不同屏幕尺寸和类型下都让需要的信息可见可用

更多细节请参考 ISTQB®易用性测试专业教学大纲[ISTQB_FLUT_2018]。

3.1.6 数据库测试

许多应用程序需要使用各种数据存储机制将数据存储在本地，如普通文件或数据库。移动应用的数据库测试需要考虑的一些测试条件包括：

- 验证数据存储问题：
 - 同步化

- 上传冲突
- 数据安全
- 对数据的约束
- CRUD（创建/读取/更新/删除）功能
- 检索
- 对设备提供的数据（如通讯录）或第三方应用程序提供的数据（如图片、视频和消息）进行数据集成测试。
- 设备上数据存储的性能。

3.1.7 全球化和本地化测试

应用程序的国际化（I18N）/全球化测试包括对不同地点、日期格式、数字和货币的格式以及用伪字符串替换实际字符串的应用程序进行测试。

本地化（L10N）测试包括针对本地化字符串、图像和特定区域工作流的应用进行测试。例如，俄语和德语单词可能比其他语言的单词长得多。由于移动设备具有不同的屏幕尺寸和分辨率，有限的屏幕尺寸可能会导致字符串翻译的问题。这些问题应作为标准的全球化/本地化测试进行检查。

需要检查的一个很重要的方面是使用的日期格式，如年-月-日或日-月-年。

3.1.8 无障碍测试

进行无障碍测试是为了确定残疾用户使用某个组件或系统的难易程度。对于移动应用程序，可以通过使用设备无障碍设置和测试应用程序的每个设置来完成。

平台供应商提供了无障碍指南，应该使用这些指南。例如，谷歌[URL5]和苹果[URL6]都为各自的平台发布了无障碍指南。从需要易访问性的人那里获得反馈也是很有帮助的。

对于移动网络，万维网联盟已经发布了一份无障碍指南，应予以考虑[URL7]。

3.2 适用于移动应用程序的额外测试级别

除了[ISTQB_CTFL_2018]中描述的从组件到验收测试的常规测试级别外，还需要为移动应用测试提供额外的测试级别。

3.2.1 现场测试

一些移动应用程序需要进行现场测试，以确保它们在实际用户的场景下能按照预期正确运行。这可能包括在各种网络 and 不同类型的通信技术（如 Wi-Fi 或蜂窝数据）上进行测试。

现场测试应在应用程序正在使用的过程中使用包括移动信号塔、有线网络、Wi-Fi 和蜂窝数据进行数据切换。测试应在不同的下载速度和信号强度下进行，并包括对盲点的处理。

现场测试需要仔细规划和确定进行测试所需的所有项，如适当的设备类型、Wi-Fi、各种运营商的蜂窝数据套餐以及获得充分覆盖所需的各种运输方式。此外，还需要安排运输方式和路径以及一天中进行测试的时间。

应用的易用性是在进行现场测试时需要涵盖的另一重要方面。测试应结合环境因素，如温度和与使用场景相关的类似条件。

3.2.2 应用商店审批和发布后的测试

在应用程序被发布之前，必须通过一些检查表/基于检查的测试，以确定应用程序在商店中能够获得批准。如果发布的是升级版，那么还应该运行升级相关的测试。

检查列表通常基于指导方针，例如针对操作系统的指导方针，基于用户界面设计，以及使用应用程序存储提供的库和 API。

提交后，审批过程可能需要一段时间。如果在审批过程中发现任何问题，可能需要提交新的版本，这将需要额外的时间来解决。这种情况在项目规划和测试期间需要仔细考虑。

另一个层次的测试是“发布后”测试。这个级别的测试包括从应用商店下载和安装应用程序。

3.3 基于经验的测试技术

3.3.1 人物和助记符

人物是代表真实顾客的虚构人物，他们有动机、期望、问题、习惯和目标。当需要模仿真实的用户行为时，使用他们会有帮助。

一个人物可以有名字、性别、年龄、收入、教育背景和位置。在移动背景下，他们可能会使用其它应用，每小时查看移动设备 x 次，还可以有其它设备和个人特征。

助记符是记忆的辅助工具。在测试的背景下，助记符中的每一个字母都代表着一种技术、一种测试方法或一个测试的重点。SFIDPOT[URL8]就是一个助记符的例子。助记符中的字母具有以下含义。

S - 结构（如用户界面元素、其它应用元素及其顺序和调用层次）。

F - 功能(例如，所需的特性是可行的，可用的，并按照规定运作等)。

i--输入（例如，所有需要的输入都是可用的，并按其应有的方式进行处理，如来自键盘、传感器和摄像头的输入）

D--数据(如：数据的存储(包括在 SD 卡上)、修改、增加和删除，按需求定义)

P--平台（如：根据设备设置，具体的操作系统功能，包括下载应用的商店，是否可以使用）

O--操作(如正常的用户活动都可以进行，例如在移动运营商网络和 Wi-Fi 之间转换)

T - 时间(例如，处理和显示时区、时间和日期)

一个专门用于移动的口诀和启发式助记符是 I SLICED UP FUN[URL9]。该助记符中的字母具有以下含义

I - 输入

S - 存储

L - 地点

I--互动和中断

C--交流

E - 人体工程学

D - 数据

U—易用性

P--平台

F--功能

U--用户场景

N--网络

3.3.2 启发式方法

启发式方法是一种"经验法则", 是一种采用实用的方法来解决、学习和发现问题的方法。这并不能保证该方法是最优的或完美的, 但可以认为对实现眼前的目标是足够的。

移动测试的启发式方法有很多。大多数助记符都可以作为启发式方法, 但并不是每个启发式方法都是助记符。

3.3.3 向导法

向导法可以用于探索性测试, 以使应用程序能够从特定的角度和重点进行探索。它们可以被执行, 以了解应用程序如何工作, 并为工作流程创建模型。向导为现场测试提供了一种有效的方法。

一个向导的例子是地标之旅, 用户模仿游客在一个城市的访问, 去知名的地标。下表显示了如何将向导中的访问和移动测试中的步骤进行类比。

地标之旅中的访问	移动测试的类比
历史性街区	遗留代码
商业区的高峰时段	业务逻辑; 应用启停
旅游区	新用户使用的部分应用
酒店区	只在睡眠模式下活动的部分应用。

将基于会话的测试(见 3.3.4 节)与向导相结合, 包括使用启发式和记忆法, 有助于提高移动应用测试的有效性。

下表显示了一些应用测试的好例子，以及它们所覆盖的领域，用于给出测试思路。其中一些可以在 [Kohl17] 中找到。

应用程序测试之旅	涵盖的主题
超模	观感以及易用性
地标	应用中最重要功能
破坏	鲁棒性
特征	新功能
场景	整个工作流程在应用中与用户故事相结合
连接性	使用的连接，如Wi-Fi、GSM
位置	正确的语言、日期、数字
光线	在不同的光照条件下的可见度，如黑暗、室外、红光。
低电量	由低电量水平引起的应用程序中的数据损失。
更多的应用程序测试之旅	[Kohl17] 中涵盖的主题
手势	尽可能使用所有的手势
方向	改变方向
改变想法	回退
动作	做各种类型的操作
位置	四处走动
连接性	通过移动改变连接类型或位置
比较	与其它类型的设备比较
一致性	检查屏幕、图形用户界面（GUI）的一致性

3.3.4 基于会话的测试管理（SBTM）

基于会话的测试管理（SBTM）使探索性测试能够以（限定时间）时间盒的方式进行管理。一个会话由三个任务组成：

- 会话设置
- 测试设计和执行
- 问题调查和报告

SBTM 通常使用包含测试章程的会话表，规定测试目标。此外，会话表还用来记录所进行的测试执行活动。

探索性测试是一种基于经验的测试技术，它可以成为测试移动应用的有效方法。基于经验的测试技术在 [ISTQB_CTFL_2018] 中有所描述。

3.4 移动测试过程和方法

3.4.1 测试过程

ISTQB®测试过程的主要活动在[ISTQB_CTFL_2018]中描述，也适用于移动应用测试。

还有其它一些特定于移动测试的方面，应作为 ISTQB®测试过程的一部分加以考虑。

测试过程中的一组主要活动	移动测试需要考虑的典型领域	参考大纲
测试计划	<ul style="list-style-type: none"> 需要测试的设备组合。 使用移动仿真器和移动模拟器作为测试环境的一部分 移动应用测试中的特殊挑战。 移动应用测试明确需要的测试类型。 	<ul style="list-style-type: none"> 第4.3节 第1.7节 第3.2节
分析和设计	<ul style="list-style-type: none"> 应用商店审批测试。 实地测试 设备兼容性。 可以使用的实验室种类 移动应用测试明确需要的测试类型。 	<ul style="list-style-type: none"> 第3.2.2节 第3.2.1节 第3.2节
测试实施和测试执行	<ul style="list-style-type: none"> 现场测试 发布后的下载和安装测试 基于经验的技术 	<ul style="list-style-type: none"> 第3.2.1节 第3.1.1节 <p>[ISTQB_CTFL_2018]</p>
测试实施和测试执行	<ul style="list-style-type: none"> 基于用户界面和应用商店的平台准则的测试。 基于准则的测试通常由平台提供商为其应用商店的审批流程运行。 建议在交给平台提供商之前，先以应用提供商的身份进行运行，以避免被拒绝的可能。 	

3.4.2 测试方法

移动应用测试包括了开发人员以及测试人员所要进行的活动。

确定每个测试级别适当的测试深度，（即组件测试、集成测试、系统测试、现场测试、应用商店批准、发布后和用户验收测试）对于交付高质量的产品非常重要。每个测试级别所需的测试深度取决于许多因素，如应用架构、应用复杂性和预期用户受众。

移动开发平台提供了各种工具来支持不同级别的测试。了解这些工具以及了解如何在特定级别上应用这些工具是非常重要的。例如，如果需要利用平台提供的框架和仪器 API，可以在组件测试级别中使用移动模拟器和/或移动仿真器。此外，当实际设备不可用时，移动模拟器和/或移动仿真器可用于系统测试级别。这样可以对功能、易用性的有限方面以及用户界面进行测试。

此外，早期实施可以作为一个关键点，确保设备的正确设置和所有执行的前提条件将按时满足。

单元测试和集成测试以及手动测试也很重要，（尤其是在现场测试阶段）。移动应用翻转测试金字塔是非常常见的[Knott15]。这意味着可以有很多手动测试。

4.移动应用平台、工具和环境-80分钟

关键字

仿真器，现场测试，近距离测试，远程测试实验室，模拟器

移动应用平台、工具和环境的学习目标

4.1 移动应用的开发平台

MAT-4.1.1 (K1)回忆移动应用开发所使用的开发环境。

4.2 通用开发平台工具

MAT-4.2.1 (K1) 回忆作为应用开发平台的一部分而提供一些常用工具。

HO-4.2.1 (H1) 使用软件开发包中的工具进行截图，提取日志和模拟传入事件。

4.3 仿真器和模拟器

MAT-4.3.1 (K2)理解仿真器和模拟器的区别。

MAT-4.3.2 (K2)描述如何使用仿真器和模拟器进行移动应用测试。

HO-4.3.2 (H1)创建并运行一个模拟/仿真设备，安装应用程序并在上面执行一些测试。

4.4 建立测试实验室

MAT-4.4.1 (K2)区分建立实验室的各种方法。

4.1 移动应用的开发平台

市场上有很多集成开发环境(IDEs)可用于各种移动应用程序的开发。这类 IDE 有各种工具，可以帮助设计、编码、编译、安装、监控、模拟、记录和测试应用程序。

例如，安卓应用开发可以使用 **Android Studio**，iOS 应用开发可以使用 **Xcode**。这些与普通 IDE 的不同之处在于它们为移动平台提供了额外的支持。

也有一些有助于移动应用开发的跨平台开发框架。这些框架运行在多个平台上，不需要特别编码。

4.2 通用开发平台工具

软件开发工具包通常提供对开发和测试应用程序很有帮助的各种实用程序。这些实用程序的用途很广，如拍摄屏幕截图、提取日志、向设备发送随机事件和通知、监控各种参数，如内存和 CPU 利用率，以及创建虚拟设备。

这类工具的一些例子是安卓虚拟设备 (AVD) 管理器，安卓调试桥 (ADB)，以及安卓设备监视器和 iOS Instruments。

4.3 仿真器和模拟器

4.3.1 仿真器和模拟器概述

在本大纲的上下文中，“仿真器”和“模拟器”是指移动仿真器或移动模拟器。术语模拟器和仿真器有时被交换使用，但并不准确。有关定义请参考第 8 章的术语表。

模拟器模拟运行时环境，而仿真器则模拟硬件并利用与物理硬件相同的运行时环境。在模拟器上测试的应用程序被编译成一个专用版本，在模拟器中工作，而不在真实设备上工作。因此，它是独立于真实的操作系统的。

相比之下，编译后在仿真器上部署和测试的应用程序被编译成实际的字节码，这些字节码可以被真实设备使用。

模拟器和仿真器在开发的早期阶段非常有用，因为这些通常与开发环境集成，并允许快速部署、测试和监控应用程序。

在测试中，模拟器有时也被用来替代真实设备。然而，这比使用仿真器局限性更大，因为在仿真器上测试的应用程序与将要分发的应用程序在字节码级别上是不同的。

仿真器还用于替代真实设备进行部分测试以降低测试环境的成本。仿真器不能完全取代设备，因为仿真器的行为方式可能与它试图模仿的移动设备不同。此外，一些功能可能不被支持，如(多点)触摸、加速度计等。部分原因是由于运行仿真器的平台的限制。

4.3.2 使用仿真器和模拟器

由于多种原因，使用仿真器和模拟器进行移动测试是很有帮助的。

每个移动操作系统的开发环境通常都有自己附带的仿真器和模拟器。也可以使用第三方仿真器和模拟器。

测试人员可以使用任何适合他们目的的仿真器或模拟器。使用时需要启动仿真器或模拟器，并在它们上安装必要的应用程序，然后像在实际设备上一样测试应用程序。

通常仿真器和模拟器允许设置各种使用参数。这些设置可能包括不同速度的网络模拟、信号强度和数据包丢失、方向改变、中断产生和 GPS 位置数据。其中一些设置会非常有用，因为使用真实设备实现会难度很大或造价昂贵，例如全球 GPS 位置或信号强度。

为了安装的目的连接到仿真器可能需要使用命令行工具，如对安卓的安卓调试桥（ADB）或从集成开发环境中连接的，如 Xcode 或 Android Studio。

4.4 建立测试实验室

建立移动试验室的方法有以下几种：

场内实验室

通过场内实验室，所有的设备、仿真器和模拟器都在现场。可以根据各种因素进行设备的选择，如设备的排名（在谷歌或其他分析中发现的）、操作系统和版本、屏幕尺寸和密度、可用性和成本、特殊功能以及对目标受众的重要性。

场内实验室的优势包括基于近距离的具体地测试和具有特定传感器的设备的可用性，如电池、触摸和增强安全性。

建立这种类型的实验室可能需要大量的预算，这取决于需要采购和维护的设备。其它挑战包括需要及时提供设备以及在不同地点和环境下进行测试所遇到的困难。

远程测试实验室

当设备或网络无法在现场实际使用时，这些实验室对测试而言非常重要，很有帮助。远程设备访问（RDA）允许通过网络连接访问托管在提供商数据中心的各种设备。这里需要评估每个潜在的 RDA 提供商是否符合要求，尤其是在安全方面。

一些远程实验室提供以下附加功能：

- 专用物理设备版本（如三星移动设备实验室）。
- 仅适用于特定操作系统和版本的通用设备。
- 执行触摸和手势相关操作的机械臂。
- 虚拟专用网络 (VPN) 连接，以提供对设备的访问。
- 与各蜂窝网络提供商的手机连接。
- 自动化工具和服务；

在使用远程测试实验室时，需要注意的因素包括设备响应速度慢以及与设备交互的选项有限，如多点触控和手势。对于偶尔的使用来说，这可能是划算的，但如果长期使用多种设备，一般来说是比较昂贵的。

其它因素包括平台按需可用，不像本地实验室需要采购没有的设备，以及实验室的可扩展性，可以随着项目进展扩大/缩小远程实验室规模。

包括 NFC/蓝牙或电池消耗等传感器的测试场景通常很难在云端进行测试。然而，远程实验室的不同地理位置可能有助于需要网络和 GPS 连接的测试。

测试实验室可以根据需要进行的测试类型，采用其中一种或两种方法的组合进行测试。

5. 自动化测试执行-55分钟

关键词

基于设备的测试，测试报告，基于用户代理的测试。

自动化测试执行的学习目标

5.1 自动化途径

MAT-5.1.1 (K2) 区分常见的自动化方法和移动应用测试框架。

5.2 自动化方法

MAT-5.2.1 (K2) 描述移动应用测试的各种自动化方法。

5.3 自动化工具评估

MAT-5.3.1 (K1) 回忆在评估移动测试自动化工具时需要考虑的各种参数。

5.4 建立自动化实验室的方法

MAT-5.4.1 (K2) 区分建立测试实验室的常见方法，以及在测试自动化方面的优缺点。

5.1 自动化途径

目前有各种自动化方法和框架可用于移动应用测试。方法的选择部分取决于应用程序的类型。

常用的测试自动化方法有以下两种：

- 基于用户代理的测试
- 基于设备的测试

基于用户代理的测试利用浏览器发送的用户代理标识符字符串来模拟特定设备上的特定浏览器。这种方法可用于执行移动 web 应用。另一方面，基于设备的测试需要直接在设备上运行被测试的应用程序。这种方法可用于所有类型的移动应用程序。

应用程序类型也可以决定适合该应用程序的测试自动化框架。移动 web 应用可以使用桌面上常用的网络应用自动化工具进行测试，而原生应用可能需要特定的工具。平台提供商也可以提供平台专用的自动化工具。

用于传统应用的自动化方法通常也适用于移动应用。这些方法包括采集/回放、数据驱动、关键字驱动和行为驱动测试，如 ISTQB® 基础级大纲 [ISTQB_CTFL_2018] 和 ISTQB® 专家级-自动化测试工程师大纲 [ISTQB_CTAL_TAE_2016] 中所述。

移动应用测试框架通常应包括的主要功能有：

- 对象识别
- 对象操作
- 测试报告

- 应用编程接口和可扩展能力
- 合适的文档
- 与其他工具的集成
- 不依赖于测试开发实践

5.2 自动化测试方法

为了开发自动化测试脚本，测试人员需要了解自动化脚本录制或创建机制，以及如何访问应用程序的图形对象，如按钮、列表框和输入字段。

有几种方法可以识别用于移动测试自动化的图形对象。这些方法包括图像识别、OCR/文本识别和对象识别（网络或原生，取决于应用程序类型）。

移动应用测试人员不仅需要练习图形化的对象检测和识别，还需要了解哪种对象识别方法最能使测试成功地在大量的各类移动设备上并行和连续运行。

脚本创建方法的主要区别是：

比较项	对象识别	图像/OCR比较
可靠性	只要标识符不变，屏幕布局就可以改变。风险在于，对象在代码中可以被识别和交互，但对用户是隐藏的。这可能会导致假阴性的测试结果。	图像可以根据屏幕大小进行缩放，但一旦布局改变，测试就会失败。
用户体验	通常需要手工编写脚本，至少是为了提高录制脚本的可读性和可维护性。	完全基于图形用户界面（GUI）的测试，无需脚本。
执行速度	往往比图像/OCR快，特别是在使用系统制造商提供的本地工具时。	由于需要将屏幕像素与基线图像逐个进行比较，往往速度较慢。
维护性	取决于测试脚本的质量。	主要在于提供变化的基线图像。
编写挑战	需要掌握脚本语言和软件设计方法，以建立一个可持续的自动化解决方案。	生成基线图像，特别是当应用程序经常变化时。

5.3 自动化工具评估

为了成功创建测试自动化解决方案，测试自动化团队需要选择一套合适的工具。需要考虑了解现有工具的主要差异及其对项目要求的符合程度(另见[ISTQB_CTAL_TAE_2016])。

测试自动化工具的评价参数可以分为两类：

- 组织匹配
- 技术匹配

组织匹配参数在 ISTQB®基础级大纲[ISTQB_CTFL_2018]第 6.2 章中有所描述。

技术匹配参数包括：

- 测试自动化需求和复杂度，如应用使用 FaceID、指纹和聊天机器人等新特性。
- 测试环境需求，如不同的网络条件，导入或创建测试数据，以及服务器端虚拟化。
- 测试报告和循环反馈功能。
- 框架在本地或云端的测试实验室中大规模管理和驱动执行的能力。
- 将测试框架与本组织使用的其他工具集成。
- 为当前和未来的升级提供支持 and 文档。

5.4 建立自动化测试实验室的方法

在执行移动应用测试时，开发人员和测试人员可以围绕设备测试实验室进行选择，他们会将测试自动化作为目标。

- 场内设备测试实验室
- 远程设备测试实验室

应用时可以对这些方法进行各种组合。这些方法的主要特点在第 4.4 节中进行了描述和比较。

场内设备测试实验室一般维护难度高且非常耗时。但这种实验室是实体设备、仿真器和模拟器并行的模式，最适合用于移动应用的早期开发和测试阶段。

当应用程序开发达到一个更高级阶段时，测试团队需要执行完整的回归测试、功能测试和非功能测试。这些测试最好在一个完整的设备实验室上执行。云端是对远程设备测试实验室进行管理、持续更新和维护的地方。

这种远程设备测试实验室是对场内设备测试实验室的补充，以确保设备和操作系统的充分组合是可用的和最新的。通过利用常见的远程设备测试实验室，团队可以获得更多的支持功能，包括更丰富的测试报告和高级测试自动化功能。

最后，当通过测试自动化框架或通过持续集成作业(CI)进行大规模执行时，整个测试实验室的稳定性是测试效率和可靠性的关键。这种实验室的设计通常是为了确保设备和操作系统始终可用并且稳定。

在应用程序的后期开发阶段，远程设备测试实验室并不总是必要的。设计和维护良好的场内设备测试实验室的使用效果可以和使用任何远程设备测试实验室一样好，甚至更好。

中国软件测试认证委员会 (CSTQBB®)

6. 参考资料

6.1 ISTQB®文档

- [ISTQB_CTFL_2018]:
- ISTQB® 认证测试工程师-基础级大纲-2018 版本
- [ISTQB_FLAT_2014]:
- ISTQB®认证测试工程师-基础级拓展大纲-敏捷测试工程师-2014 版本
- [ISTQB_FLUT_2018]:
- ISTQB®认证测试工程师-基础级专家大纲-易用性测试-2018 版本
- [ISTQB_CTFL_PT_2018]:
- ISTQB®认证测试工程师-基础级专家大纲-性能测试-2018 版本
- [ISTQB_CTAL_SEC_2016]:
- ISTQB®认证测试工程师-高级专家大纲-安全测试-2016 版本
- [ISTQB_CTAL_TAE_2016]:
- ISTQB®认证测试专家-高级专家大纲-自动化测试工程师-2016 版本
- [ISTQB_GLOSSARY]:
- ISTQB®软件测试专业术语对照表, 3.2 版本

6.2 参考书目

- [Knott15] Knott, D., “Hands-On Mobile App Testing”, Addison-Wesley Professional, 2015, ISBN978-3-86490-379-3
- [Kohl17] Kohl, J., “Tap into mobile application testing”, leanpub.com, 2017, ISBN978-0-9959823-2-1

6.3 其他书籍和文章

- Boris Beizer, “Black-box Testing”, John Wiley & Sons, 1995, ISBN0-471-12094-4
- Rex Black, “Agile Testing Foundations”, BCS Learning & Development Ltd: Swindon UK, 2017, ISBN978-1-78017-33-68
- Rex Black, “Managing the Testing Process”(3e), John Wiley & Sons: New York NY, 2009, ISBN978-0-470-40415-7

- Hans Buwalda, “Integrated Test Design and Automation”, Addison-Wesley Longman, 2001, ISBN0-201-73725-6
- Lee Copeland, “A Practitioner's Guide to Software Test Design”, Artech House, 2003, ISBN1-58053-791-X
- Rick David Craig, Stefan P. Jaskiel, “Systematic Software Testing”, Artech House, 2002, ISBN1-580-53508-9

6.4 链接（网络/互联网）

免责声明：截至 2019 年 1 月 5 日，所有链接都有效。

- [URL1]<http://gs.statcounter.com/>
- [URL2]www.owasp.org
- [URL3]<https://developer.android.com/studio/test/monkey>
- [URL4] <https://www.google.de/amp/s/techcrunch.com/2016/05/31/nearly-1-in-4-people-abandon-mobile-apps-after-only-one-use/amp/>
- [URL5]<https://www.google.com/accessibility/>
- [URL6]<https://www.apple.com/uk/accessibility/>
- [URL7]<https://www.w3.org/WAI/standards-guidelines/mobile/>
- [URL8] <https://www.slideshare.net/karennjohnson/kn-johnson-2012-heuristics-mnemonics>
- [URL9]<http://www.kohl.ca/articles/ISLIGEDUPFUN.pdf>

7.附录A-学习目标/知识认知水平

下面定义的学习目标适用于本大纲。大纲中的每个专题将根据其学习目标进行考查。

7.1 第一级: 牢记(K1)

考生应识别、牢记和记忆术语或者概念的内容。

关键字: 识别、牢记、回忆、记忆、识别、认知

例子:

能够识别“失效”的定义:

- “不能向最终用户或其他干系人提供服务”，或
- “组件或者系统的实际运行情况与期望的发布、服务或结果背离”。

7.2 第二级: 理解(K2)

考生应能选择与大纲主题相关的陈述理由或解释，同时对测试概念能够进行总结、比较、分类并举例说明。

关键字: 总结、概括、摘要、归类、比较、映射、对比、举例说明、解释、翻译、描述、推断、结论、分类、构建建模

例子:

请解释测试分析与设计应该尽早进行的原因:

- 在缺陷移除成本低的时候就发现它们;
- 尽早发现那些最重要的缺陷。

请解释集成测试和系统测试之间的异同:

- 相同: 需测试多个组件, 可以包括非功能性的测试类型;
- 不同: 集成测试关注于接口和交互, 而系统测试则关注于从全系统角度进行测试, 如端到端的流程。

7.3 第三级: 应用 (K3)

考生应能够选择测试概念或者技术的正确应用, 并将其应用到给定的场景中。

关键字: 实施、执行、运用、遵循流程、使用流程。

例子:

- 可以确认有效和无效等价类的边界值。
- 能从给定的状态转换图选择测试用例, 以覆盖所有的状态转换。

参考文献（针对学习目标的认知水平）：

Anderson, L. W. and Krathwohl, D. R. (eds) (2001) A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives, Allyn& Bacon: Boston MA

中国软件测试认证委员会 (CSTQBB®)

8. 附录B-领域专业术语表

术语	定义
2G	第二代移动无线通信技术。
三维触控	见“压力感应触控”
3G	第三代移动无线通信技术。
4G	第四代移动无线通信技术。
5G	第五代移动无线通信技术。
ADB	安卓调试桥（ADB）--允许与设备进行通信的命令行工具。
广告型应用	一种应用盈利模式，开发机构通过应用内显示的广告赚钱。
安卓设备监视器（ADM）	为安卓应用调试和分析工具提供了用户界面的独立工具。
Android Studio	安卓的官方集成开发环境（IDE）。安卓开发工具提供了在各种类型的安卓设备上构建应用程序的工具。
应用快捷方式	安卓7.1或更高版本上的应用开发者在应用中定义的一组特定动作的快捷方式。
应用商店	一个应用分发平台，开发者可以在这里上传应用，用户也可以在平台上搜索应用进行下载安装。
宽高比	显示屏或图像的宽度与高度之比。
异步通信	一种间歇性而不是稳定地传输数据的通信方式。
AVD	安卓虚拟设备的缩写
后端系统	为其它系统提供功能的服务器系统。
后台应用	在后台运行的应用程序。
向下兼容性	应用程序在早期版本的平台上运行的能力。
条码	一种光学的、机器可读的数据表示法。
基本型手机	具有拨打电话、存储电话号码、发送短信、时钟、闹钟等基础功能的手机。
盲点	没有无线电信网的地方
屏蔽/勿扰模式	移动设备的一种操作模式，用户可以激活它来禁止某些功能—例如常见的通知和语音通话。
蓝牙	一种近距离无线通信技术。
字节码	为软件解释器有效执行而设计的指令集。也称为可移植代码或p代码。
蜂窝数据	通过蜂窝网络传输的数据。
蜂窝网络	蜂窝网络是由多个独立但相互连接的区域组成的网络。
配套设备	设计成与附属智能设备一同工作的计算机设备。
CPU频率	处理器的时钟频率
跨平台开发框架	使用相同代码库为不同平台开发应用的框架。
CRUD	应用于数据的创建/读取/更新/删除的缩略。
数据完整性	数据在其整个生命周期内的准确性和一致性，包括存储、处理和检索。

术语	定义
数据同步	将两个或多个来源的数据纳入同一状态的过程。
数据验证	评价数据是否正确、准确、一致和可用。
死角	见“盲点”。
设备分布	可用设备的多样性，其独特的硬件配置，以及其对应用程序和用户体验的影响。
DPI/PPI	Dots/pixels per inch（每英寸点数/像素）的缩写--表示显示器密度的数字，可以是点数或像素。
仿真器	一个模仿硬件行为的软件应用。
企业应用	为组织内部使用而创建的应用程序，不提供给公众使用。
外部存储器	通过标准接口添加到设备上的一种额外内存。目前手机中最常见的是SD卡。
胖客户机	在客户机/服务器应用程序中，被设计为进行部分或大部分数据处理的客户机。
功能手机	一种比基本手机提供更多功能的手机，如浏览器，但不提供智能手机的全部功能。
平面文件	没有内部层次结构的文件。
飞行模式	移动设备的一种特殊操作模式，在该模式下，无线电发射器被停用，以防止干扰飞行器的操作/通信系统。
Force Touch	苹果公司开发的一项技术，使触控板和触摸屏能够区分施加在其表面的不同力度。
前台应用	在设备前台运行的应用程序，用于直接与用户交互。
免费增值应用	一种商业模式，用户无需支付任何费用即可下载应用，并提供可选择的应用内购买。
手势	一种交互模式，如捏住或轻扫，以激活设备的定义功能。例如，捏一捏通常用于放大或缩小智能设备的屏幕。
全球化	见国际化
GPS	全球定位系统的缩写 -- 在全球范围内，一个卫星网络发出时间信号。通过接收至少3颗卫星的信号，接收器可以通过三角测量法计算出其与卫星的相对位置。
GSM	GSM(Global System for Mobile Communications，原为Groupe Spécial Mobile)的缩写，是欧洲电信标准协会(ETSI)制定的标准，用于描述移动设备使用的第二代数字蜂窝网络的协议。目前是最通用的移动通信标准。
GSM 手机	GSM网络的一部分，可以通过其独特的单元ID识别。
混合应用	一种结合了原生技术和网络技术的应用。通常情况下，混合应用程序使用安装在设备上的原生框架与设备库等进行交互。此外，还可以展示从网络服务器接收的内容。
l18N	国际化的数字名（数字型词）。

术语	定义
IDE	集成开发-一个为计算机程序员提供全面软件开发设施的软件应用。
应用内购买	额外的内容和功能可直接从应用中获得。
Instruments	作为Xcode工具集一部分的性能分析和测试工具。
内部存储器	硬件设备中包含的存储器。
国际化	为适应各种本地化版本而准备应用程序的过程。
中断	在另一事件中发生的事件
物联网设备	物联网。一个设备，或者，例如一个连接到互联网的传感器。
越狱	一种权限升级，目的是消除操作系统施加的软件限制。通常用于iOS系统的术语。类似于root一个安卓设备。
L10N	本地化的代号。
横向模式	显示宽度大于高度的设备方向。
库	计算机程序使用的非易失性资源的集合，即应用程序的功能。
本地化	通过翻译和格式调整等动作，将应用或产品调整到某个地区的过程
观感	对事物的视觉和情感印象。
记忆法	记忆的辅助工具
移动应用测试	测试移动应用
移动设备类型	按移动设备的基本特征进行分类。常见的分类包括基本手机、功能手机、智能手机、平板电话、平板电脑和可穿戴设备。
移动仿真器	硬件平台的虚拟表示。例如，安卓模拟器是运行真实安卓操作系统镜像的虚拟硬件。同样的操作系统镜像可以部署到硬件上，并且可以工作，因为它是真实的操作系统。
移动操作系统	一种专门为移动设备设计的操作系统。
移动平台	围绕移动操作系统的生态系统，通常包括开发工具、操作系统本身和应用分发渠道。
移动空间	一个概括性的术语，包括从市场和参与者到设备和应用的任何与手机设备技术有关的内容。
移动模拟器	一个虚拟的运行环境。例如，iOS模拟器假装是iOS，但实际上不是真正的iOS。
多平台应用	为在多平台上运行而设计和开发的应用程序，对所有平台使用相同的代码库。
多层	一种后端应用设计方法，由2台或更多的服务器提供专门的功能。
多点	一种使用各种并行的触摸事件与设备进行交互的类型。
原生应用	专门为某一平台开发的应用程序，通常使用平台API和平台提供的开发工具。

术语	定义
NFC	近场通信--近距离无线电通信技术。
通知	设备发出的公告。
OCR	光学字符识别。识别电子图片中的文字图像，并将其转换为机器编码的文字。
场内实验室	一个实验室，其物理位置与实验室的使用者在同一个地方。
方向	移动设备中对象的摆放位置，常用来表达设备的使用方式。它可以是横向的，也可以是纵向的。
OTA	空中传输。通过无线电信号进行数据传输，通常指的是不通过电缆连接的信号源直接向设备安装应用。
溢出	传入的数据超过所能容纳的范围的情况。
付费应用	通过在应用商店中销售来实现盈利的应用。
角色	某一用户组的模型/原型。
纵向模式	显示高度大于宽度的设备的方向。
功耗	消耗的能源量。
省电模式	移动设备的一种操作模式，可由用户或设备本身激活，以节约能源。
功率状态	用户定义或预定义的关于功耗的配置文件，可以在移动设备上激活。
偏好	一般设备或应用的配置参数，可以由用户更改。
预装应用	由设备制造商安装的移动应用程序。通常，用户无法卸载这些应用程序。
二维码	QR码(Quick Response Code的缩写)是一种矩阵条码(或二维条码)的商标。
远程设备访问	与位于与用户不同地点的设备进行互动，通常通过互联网进行。
保留应用数据	当应用程序被卸载时，用户生成的数据和/或应用程序的内容将保留在设备上，以便其它应用程序访问，或当同一应用程序被重新安装时。
Rooting	获得设备操作系统的root权限的过程。该术语通常用于Android平台。类似于iOS平台的越狱。
运行时环境	执行模式的实现。又称运行时系统。
屏幕空间	显示器提供的空间大小。
软件开发包(SDK)	为某一平台开发软件的一套工具和库。
敏感数据	需要特别保护的数据，如密码和个人数据。
传感器	一种设备、模块或子系统，其目的是检测环境中的事件或变化，并将信息发送到其它电子装置，通常是计算机处理器。
会话	一个（限定时间）时间盒事件。
会话表	一个文件，用于确定范围和记录测试环节。
侧载	通过应用程序商店以外的方式加载/安装应用程序。

术语	定义
单层	一种后端应用程序设计方法，由一台服务器为应用程序提供所有需要的服务。
智能手机	手持式个人电脑，带有移动操作系统和集成的移动宽带蜂窝网络连接，用于语音、短信（短信息服务，通常称为文本）和互联网数据通信。
软键盘	在软件中实现的虚拟键盘，通过显示屏呈现给用户。
存储和转发	一种数据同步方式，数据存储在本地，当有适当的网络连接时，再转发到服务器。
同步通信	一种数据流以相同的速度发送(上游)和接收(下游)，并以定时信号间隔的数据传输方法。
平板电脑	一种移动设备，常用于屏幕为7"以上的设备。
瘦客户机	在客户机/服务器应用中，客户机设计得特别小，以便大部分数据处理发生在服务器上。
第三方市场	不由平台商运营的应用发布平台。
基于交易的应用程序	用户按交易付费的应用程序。
上传冲突	试图上传一个已经存在于上传目的地的文件时出现的错误。
视角尺寸	浏览器用来调整屏幕布局的虚拟屏幕尺寸。
虚拟专用网络（VPN）	通过公共网络的加密专用通道。
可穿戴	佩戴在身上的电脑设备，如手表或眼镜。
网络应用	通过浏览器在互联网上托管的应用程序。
Xcode	苹果提供的集成开发环境，用于开发OSX和iOS应用。