

软件测试技术是软件开发过程中的一个重要组成部分，是贯穿整个软件开发生命周期、对软件产品（包括阶段性产品）进行验证和确认的活动过程。其目的是尽快尽早地发现在软件产品中所存在的各种问题，与用户需求、预先定义的不一致性。检查软件产品中可能存在的 Bug，并且编写缺陷报告，交于开发人员修改。软件测试人员的基本目标是发现软件中的错误。

软件测试技术就相当于软件测试人员的武器。作为软件测试人员，必须要清楚了解可以通过哪些手段去保障产品的质量。只有知道了这些，才能更好的完成测试的工作。

软件测试分类

软件测试的分类可以按照不同的维度去划分，一般来说可以按照下面的这些维度去划分。



按开发阶段分类

- 单元测试
- 集成测试
- 冒烟测试
- 系统测试
- 验收测试



按测试实施组织分类

- α 测试：非正式验收测试
- β 测试：内测后的公测



按测试执行方式分类

- 静态测试：不启动被测对象的测试，比如代码走读，代码评审，文档评审，需求评审等。
- 动态测试：启动被测对象的测试，比如白盒测试，黑盒测试等。



按是否查看代码分类

- 黑盒测试：指的是把被测的软件看作是一个黑盒子，不去关心盒子里面的结构是什么样子的，只关心软件的输入数据和输出结果。
- 白盒测试：指的是把盒子盖子打开，去研究里面的源代码和程序结果。



按是否手工执行分类

- 手工测试：由人去一个一个的去执行测试用例，通过键盘鼠标等输入一些参数，查看返回结果是否符合预期结果。通常用于黑盒测试方法或系统测试阶段。
- 自动化测试：把以人为驱动测试行为转化为机器执行的一种过程。



按测试对象分类

- 性能测试：检查系统是否满足需求规格说明书中规定的性能。
- 安全测试：各种的攻击手段，例如 SQL 注入、XSS 等。
- 兼容性测试：软件和硬件之间是否能够发挥很好的效率工作，会不会影响导致系统的崩溃。
- 文档测试：测试软件产品中的各类文档。
- 易用性测试：用户体验测试。
- 业务测试：测试人员将系统的各个模块串接起来运行、模拟真实用户实际的工作流程，满足永续需求定义的功能进行测试的过程。
- 界面测试：也称为 UI 测试。测试用户界面的功能模块的布局是否合理，整体风格是否一致、各个控件的放置位置是否符合客户的使用习惯，还要测试操作界面操作便捷性、导航简单易懂性、页面元素的可用性，页面元素的可用性、界面中文字是否正确，命名是否统一，页面是否美观、文字、图片组合是否完美。
- 安装测试：测试程序的安装、卸载。



其他分类

- 回归测试：修改了旧代码后，重新执行测试以确认修改后没有引入新的错误或导致其他代码产生错误。
- 随机测试：指测试中的所有输入数据都是随机生成的，其目的是模拟用户的真实操作，并发现一些边缘性的错误。
- 探索性测试：可以说是一种测试思维技术。它没有很多实际的测试方法、技术和工具，但是却是所有测试人员都应该掌握的一种测试思维方式。探索性强调测试人员的主观能动性，抛弃繁杂的测试计划和测试用例设计过程，强调在碰到问题时及时改变测试策略。

黑盒测试

黑盒测试又叫功能测试、数据驱动测试或基于需求规格说明书的功能测试。该类测试注重于测试软件的功能性需求。

采用这种测试方法，测试工程师把测试对象看作一个黑盒子，完全不考虑程序内部的逻辑结构和内部特性，只依据程序的需求文档，检查程序的功能是否符合它的功能说明。测试工程师无需了解程序代码的内部构造，完全模拟软件产品的最终用户使用该软件，检查软件产品是否达到了用户的需求。

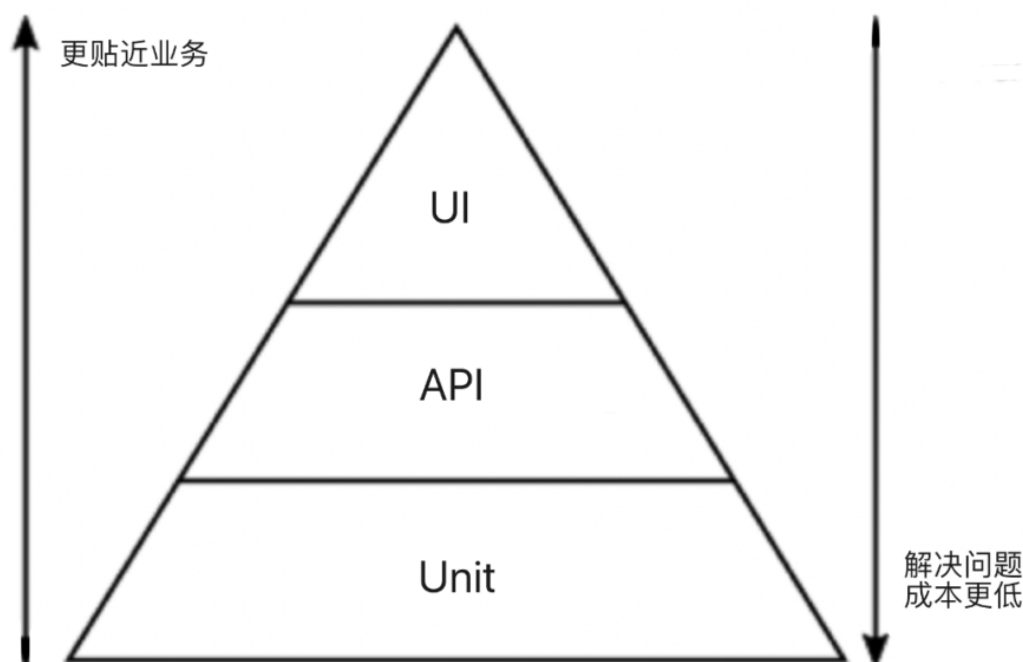
黑盒测试方法能更好、更真实地从用户角度来考察被测系统的功能性需求实现情况。在软件测试的各个阶段，如单元测试、集成测试、系统测试及验收测试等阶段中，黑盒测试都发挥着重要作用，尤其在系统测试和确认测试中，其作用是其他测试方法无法取代的。

白盒测试

白盒测试又称结构测试、透明盒测试、逻辑驱动测试或基于代码的测试。白盒法可以全面了解程序内部逻辑结构、对所有逻辑路径进行测试。

白盒测试常用的方法有代码检查法、静态结构分析法、静态质量度量法、逻辑覆盖法、基本路径测试法。

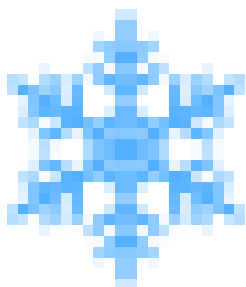
分层测试体系



其中 Unit 代表单元测试，API 代表接口测试，UI 代表页面级的系统测试。分层的自动化测试倡导产品的不同层次都需要自动化测试，这个金字塔也正表示不同层次需要投入的精力和工作量。对于测试金字塔，越靠下越容易自动化，越靠下成本越低，越靠下效率越高。

分层测试顾名思义就是分多个层次一个层次一个层次的测试，比如先测完中间接口层，再测最上层的界面。当然了，也可以同时测试。

分层测试的测试方法还是原来的测试方法，但对测试人员的代码能力还有自动化测试水平有较高要求，同时要求测试人员和开发团队真正的理解敏捷开发和敏捷测试，甚至要求开发团队达到开发即测试、测试即开发的能力。



对软件中的最小可测试单元进行检查和验证。具体的说就是开发者编写的一小段代码，用于检验被测代码的一个很小的、很明确的功能是否正确。通常而言，一个单元测试是用于判断某个特定条件（或者场景）下某个特定函数的行为。



接口测试

接口测试是测试系统组件间接口的一种测试，主要用于检测外部系统与系统之间以及内部各个子系统之间的交互点。

测试的重点是要检查接口参数传递的正确性，接口功能实现的正确性，输出结果的正确性，以及对各种异常情况的容错处理的完整性和合理性。

接口测试可以更早介入，介入越早越能更早的发现问题，还可以缩短项目周期，能够发现更底层的 Bug，减少开发成本。

因为不同端（前端，后端）的工作进度不一样，所以要针对最开始出来的接口，以及需要调用其他公司的（银行，支付宝，微信，QQ 等）一些接口进行接口测试及验证数据，从安全层面来说，只依赖前端进行限制已经完全不能满足系统的安全要求（绕过前面实在太容易），需要后端同样进行控制，在这种情况下就需要从接口层面进行验证。前后端传输、日志打印等信息是否加密传输也是需要验证的，特别是涉及到用户的隐私信息，如身份证，银行卡等。



UI 测试

UI 测试测的是应用中的用户界面是否如预期工作。比如，用户的输入需要触发正确的动作，数据需要能展示给用户看，UI 的状态需要发生正确变化等。

对于 UI 测试，可以采用静态测试方法，也可以采用动态测试方法。

对于用户界面的布局，风格，字体，图片等与显示相关的部分测试应该采用静态测试，比如点检表测试，即将测试必须通过的项用点检表一条一条列举出，然后通过观察确保每项是否通过。

对用户界面中各个类别的控件应该采用动态测试，即编写测试用例或者点检表，对每个按钮的响应情况进行测试，是否符合概要设计所规定的条件，还可以对用户界面在不同环境下的显示情况进行测试。

UI 测试需要关注的内容包括通过浏览测试对象可正确反映业务的功能和需求，这种浏览包括窗口与窗口之间、字段与字段之间的浏览。各种访问方法（Tab 键、鼠标移动和快捷键）是否支持。还有窗口的对象和特征，比如菜单、大小、位置、状态和中心等都符合标准。