

## 基本概念

软件包括源代码，用户手册，配置数据。

软件测试是指通过人工或自动的手段，对软件进行检测的活动，其目的在于发现软件与客户需求之间的差异，或者说弄清楚实际结果与预期结果之间的差异。

软件测试的原则：

- 1、测试证明软件存在缺陷；
- 2、不可能执行穷尽测试；
- 3、软件缺陷存在集群现象（二八定律）；
- 4、测试应尽早启动，尽早介入；
- 5、杀虫剂悖论；
- 6、不同的测试活动依赖于不同的测试背景；
- 7、软件不存在缺陷的谬论。

## 软件测试的手段

1. 单元测试。单元测试是对软件组成单元进行测试，其目的是检验软件基本组成单位的正确性，测试的对象是软件设计的最小单位：函数。并且使用假资料测试不同状况下功能使用情况，单元测试还有助于开发人员编写更好的代码。单元测试是基于代码的可读性、可测试性，它们与开发代码的构建方式密切相关。因此开发人员最清楚哪些测试最有意义。
2. 集成测试。集成测试也称综合测试、组装测试、联合测试，将程序模块采用适当的集成策略组装起来，对系统的接口及集成后的功能进行正确性检测的测试工作。其主要目的是检查软件单位之间的接口是否正确，集成测试的对象是已经经过单元测试的模块。
3. 系统测试。系统测试主要包括功能测试、界面测试、可靠性测试、易用性测试、性能测试。功能测试主要针对包括功能可用性、功能实现程度（功能流程&业务流程、数据处理&业务数据处理）方面测试。
4. 回归测试。回归测试指在软件维护阶段，为了检测代码修改而引入的错误所进行的测试活动。回归测试是软件维护阶段的重要工作，有研究表明，回归测试带来的耗费占软件生命周期的 1/3 总费用以上。
5. 模糊测试（fuzz testing）是一种安全测试方法，是一种介于完全的手工渗透测试与完全的自动化测试之间的安全性测试类型。它充分利用了机器的能力：随机生成和发送数据；同时，也尝试将安全专家在安全性方面的经验引入进来。

执行过程：

- 1、测试工具通过随机或是半随机的方式生成大量数据；
  - 2、测试工具将生成的数据发送给被测试的系统（输入）；
  - 3、测试工具检测被测系统的状态（如是否能够响应，响应是否正确等）；
  - 4、根据被测系统的状态判断是否存在潜在的安全漏洞。
6. 白盒测试。白盒测试相对于黑盒测试而言具有一定透明性，原理为根据软件内部应用、源代码等对产品内部工作过程进行调试。测试过程中常将其与软件内部结构协同展开分析，最大优点即为其能够有效解决软件内部应用程序出现的问题，测试过程中常将其与黑盒测试方式结合，当测试软件功能较多时，白盒测试法亦可对此类情况展开有效调试。其中，判定测试作为白盒测试法中最为主要的测试程序结构之一，此类程序结构作为对

程序逻辑结构的整体实现，对于程序测试而言具有较为重要的作用。

此类测试方式针对程序中各类型的代码进行覆盖式检测，覆盖范围较广，适用于多类型程序。实际检测中，白盒测试法常与黑盒检测法并用，以动态检测方式中测试出的未知错误为例，首先使用黑盒检测法，若程序输入数据与输出数据相同，则证明内部数据未出现问题，应从代码方面进行分析，若出现问题则使用白盒测试法，针对软件内部结构进行分析，直至检测出问题所在，及时加以修改。

7. 黑盒测试。黑盒测试，顾名思义即为将软件测试环境模拟为不可见的“黑盒”。通过数据输入观察数据输出，检查软件内部功能是否正常。测试展开时，数据输入软件中，等待数据输出。数据输出时若与预计数据一致，则证明该软件通过测试，若数据与预计数据有出入，即便出入较小亦证明软件程序内部出现问题，需尽快解决。

## 重要挑战

### 测试环境的多样性

应用程序和 Web 应用程序往往同时从数千个设备通过不同操作系统和不同浏览器平台组合访问。毋庸置疑，测试团队需要尽可能构建可能的环境，使应用在最广泛使用的组合中完美无瑕。但是，考虑到市场上已经存在的桌面设备和移动设备数量越来越多，浏览器/操作系统的版本也在不断迭代更新，在推出的每个最新设备/操作系统/浏览器上进行测试应用是一个挑战。

虽然使用仿真器和模拟器可以在测试初始阶段有所帮助，但他们不能保证真实用户条件下的应用性能。只有在真实设备上测试软件时，才会检测到实时错误。这意味着团队需要访问设备实验室，允许在各种设备上进行测试。然而，建立现有设备实验室要求巨额投资，可能对中小型公司不可能经济上可行。

### 测试现代的真实系统

许多测试技术，特别是尚未进入工业实践的学术层面的技术，都是以一些传统的软件为目标的，这些软件是用单一语言编写的、同构的、非分布式的，在某些情况下甚至很小。不幸的是，今天的许多软件系统与这些传统系统有很大的不同，使得这些学术上发现的新技术却对现代的工业界没有什么贡献。具体地说，许多现代软件系统由不同性质和来源的组件组成，彼此之间具有不同程度的耦合，并且它们通常是分布式的和高度动态的。许多日益流行的应用程序(如移动应用程序、Web 应用程序、软件产品线、面向服务的体系结构和基于云的应用程序)都具有这些特征。因此，如何与时代接轨，充分考虑当下最新的软件的特点，并针对性的设计出具有有效用的软件测试方法、机制，是软件测试领域需要考虑的一个重大问题。

### 测试软件的非功能性属性

到目前为止，关于测试的大部分研究都集中在功能正确性测试上，在某些方面被称为“调试测试”。然而，软件系统还有其他属性。

工程师可能希望通过测试来评估这一软件系统的其他属性，比如性能，但到目前为止，许多测试在研究文献中得到的关注相对较少。几乎所有软件系统都可能出现性能问题，但对于某些类别的系统来说，这些问题非常重要。这些类别包括 Web 应用程序和 Web 服务，对于这些应用程序和 Web 服务，响应时间可能是用户采用的一个关键因素。硬实时系统也十分重视性能，这是一段时间以来一直令人担忧的问题，随着移动平台的日益相关和广泛使用，其重要性正在与日俱增。一些研究人员已经提出了基于对系统的分析来估计最坏情况下的中断延迟(WCIL)的方法。然而，这些方法都是保守的，可能会过度逼近 WCILs，这就影响了其结果的精度和适用性。相反，基于测试的方法可能低估了 WCIL。总的来说，在测试软件正确性的同时，如何可以精确的测试出软件的性能、并发性等非功能性属性，也是软件测试需要考虑的方面。