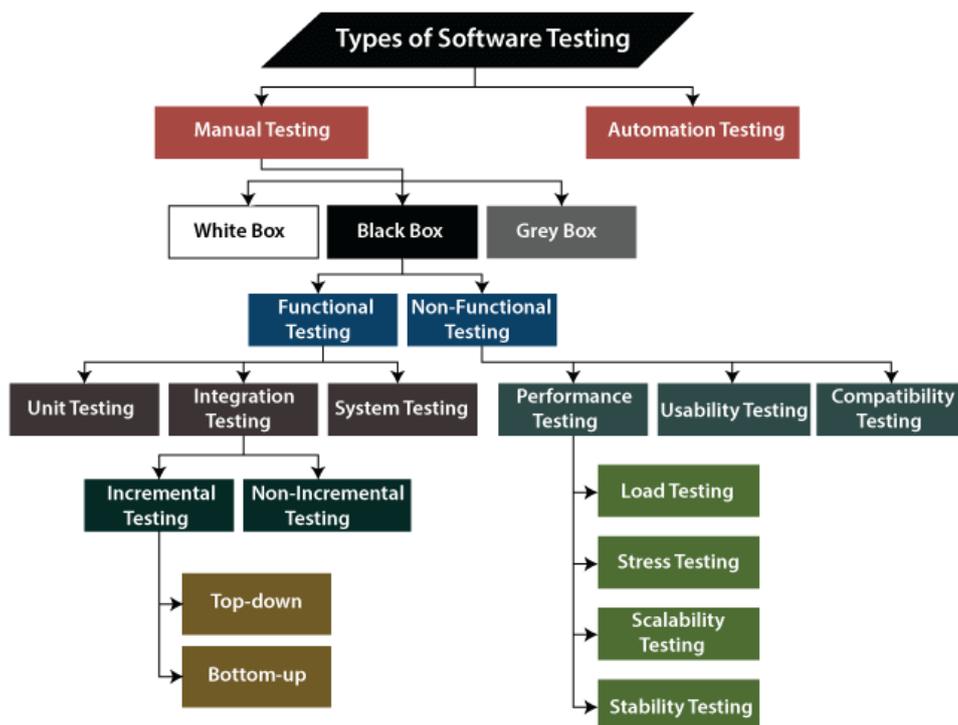


# 软件测试的类型

来源	来自于 JavaTpoint 网站
翻译	Alice (火龙果软件)
说明	在本节中，我们将了解各种类型的软件测试，这些测试可以在软件开发生命周期时使用，下面请仔细阅读本文来学习吧!



MBSE with 火龙果

## 目录

一、软件测试的类型 .....	1
一、 不同类型的软件测试 .....	3
二、 什么是手动测试? .....	4
三、 手动测试的分类 .....	4
1. 白盒测试 .....	5
2. 黑盒测试 .....	6
3. 黑盒测试的类型 .....	7
4. 功能测试 .....	8
5. 功能测试的类型 .....	8
四、 非功能测试 .....	12
6. 非功能性测试的类型 .....	12
五、 灰盒测试 .....	16
六、 自动化测试 .....	17
七、 一些其他类型的软件测试 .....	18
1. 健全性测试 .....	20
2. 回归测试 .....	20
3. 用户验收测试 .....	21
4. 探索性测试 .....	21
5. 临时测试 .....	21
6. 全球化测试 .....	22
八、 结论 .....	23

众所周知，软件测试是根据客户的先决条件分析应用程序功能的过程。

如果我们想确保我们的软件没有错误或稳定性较好，我们必须执行各种类型的软件测试，因为测试是使我们的应用程序避免出错误的唯一方法。

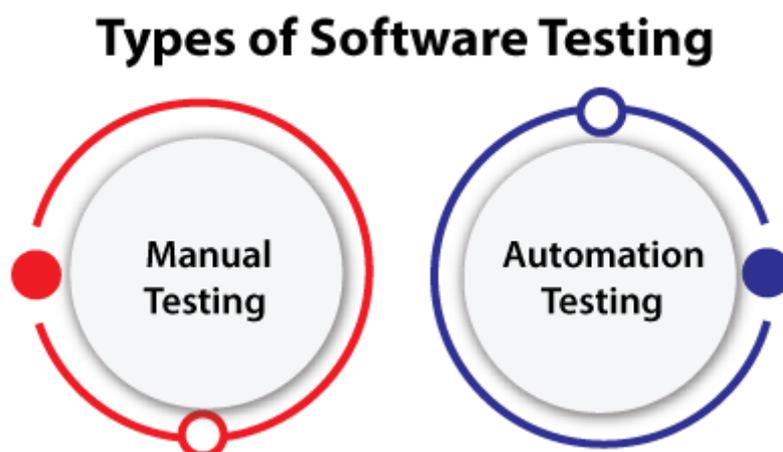
## 一、不同类型的软件测试

软件测试的分类是各种测试活动的一部分，例如测试策略、测试可交付成果、定义的测试目标等。而软件测试就是执行软件来发现缺陷。

具有测试类型的目的是确认 AUT（被测试应用程序）。

要开始测试，我们应该有一个可用的、应用程序就绪的必要资源。为了保持问责制，我们应该将相应的模块分配给不同的测试工程师。

软件测试主要分为两部分，具体如下：

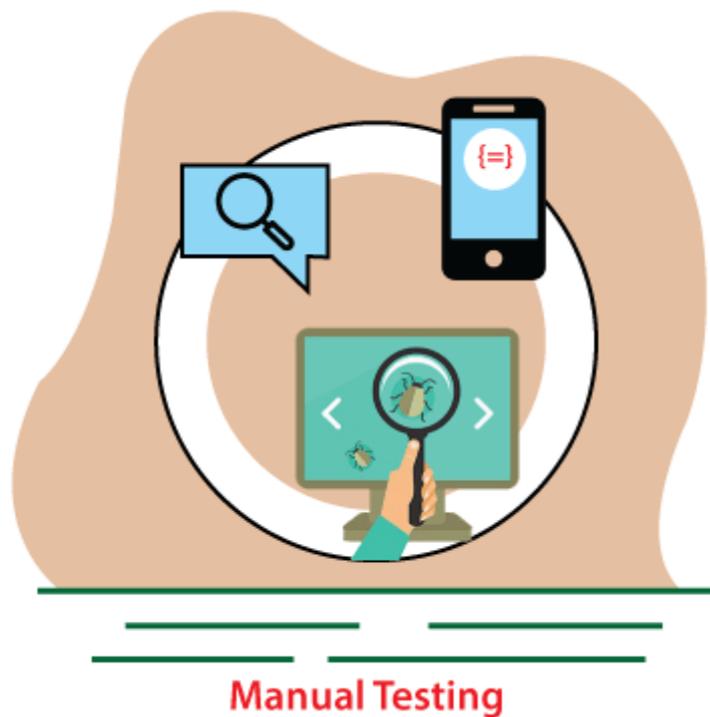


- 手动测试
- 自动化测试

## 二、什么是手动测试？

根据客户的需求测试任何软件或应用程序，而无需使用任何自动化工具，称为手动测试。

换句话说，我们可以说这是一个验证和确认的过程。手动测试用于验证应用程序或软件的行为是否符合需求规范。



我们不需要任何测试工具的精确知识来执行手动测试用例。我们可以轻松准备测试文档，同时对任何应用程序执行手动测试。

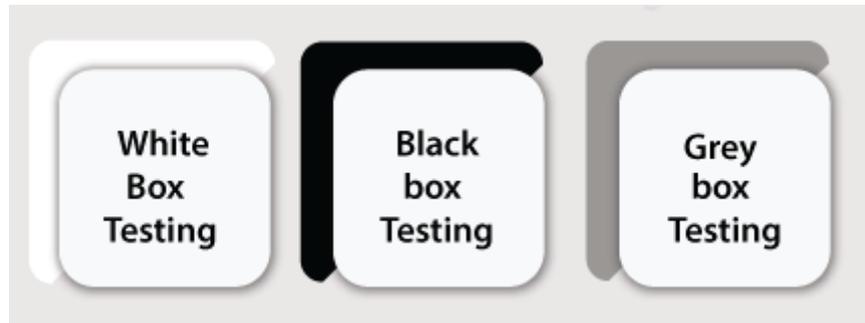
要获取有关手动测试的详细信息，请单击以下链接：

<https://www.javatpoint.com/manual-testing>。

## 三、手动测试的分类

在软件测试中，手动测试可以进一步分为三种不同类型的测试，具体如下：

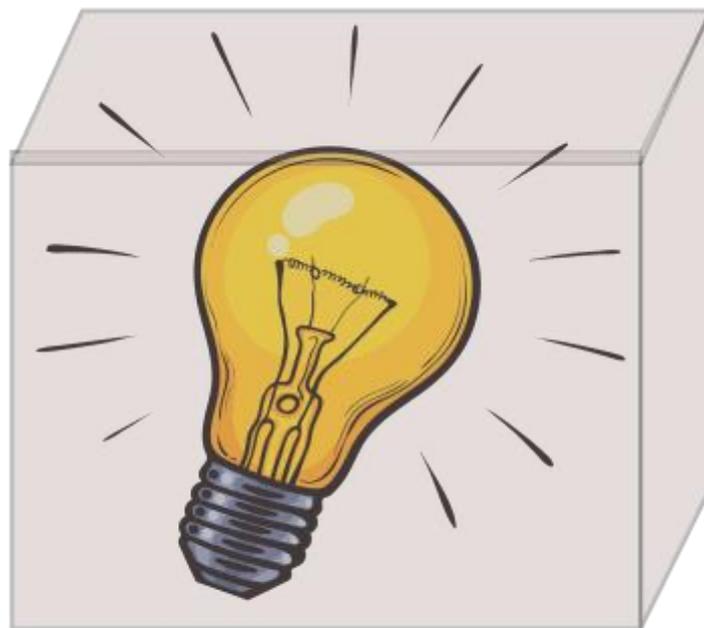
- 白盒测试
- 黑盒测试
- 灰盒测试



为了更好地理解，让我们——看：

## 1. 白盒测试

在白盒测试中，开发人员将检查每一行代码，然后将其交给测试团队或相关的测试工程师。

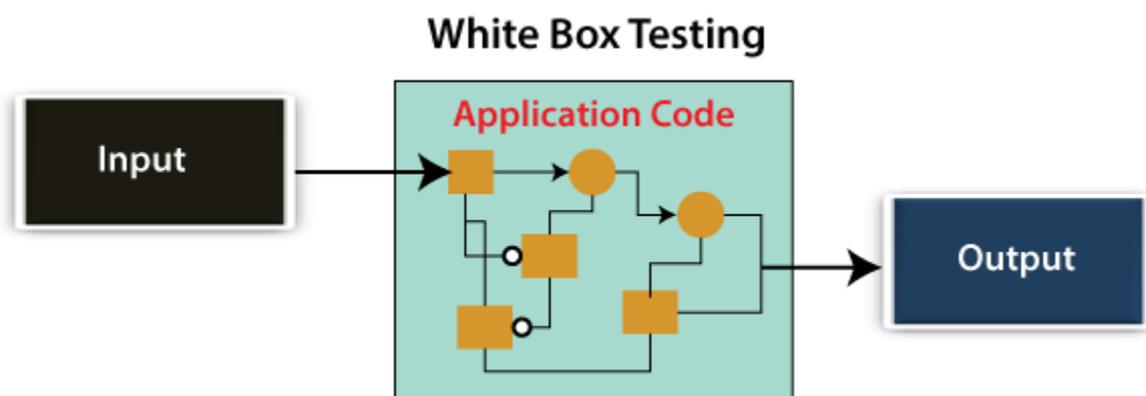


## White Box Testing

随后，在整个测试过程中，开发人员可以注意到代码;这就是为什么这个过程被称为 WBT (白盒测试)。

换句话说，我们可以说开发人员将对特定软件执行完整的白盒测试，并将特定应用程序发送给测试团队。

实施白盒测试的目的是强调软件上的输入和输出流，并增强应用程序的安全性。



白盒测试也称为开盒测试、玻璃盒测试、结构测试、透明盒测试和透明盒测试。

要深入了解白盒测试，请参阅以下链接：

<https://www.javatpoint.com/white-box-testing>。

## 2. 黑盒测试

另一种类型的手动测试是黑盒测试。在此测试中，测试工程师将根据需求分析软件，识别缺陷或错误，并将其发送回开发团队。

然后，开发人员将修复这些缺陷，进行一轮白盒测试，并将其发送给测试团队。

在这里，修复错误意味着缺陷已解决，并且特定功能正在根据给定的要求工作。

实施黑盒测试的主要目标是指定业务需求或客户的要求。

换句话说，我们可以说黑盒测试是根据客户要求检查应用程序功能的过程。源代码在此测试中不可见;这就是为什么它被称为黑盒测试。



有关黑盒测试的更多信息，请参阅以下链接：

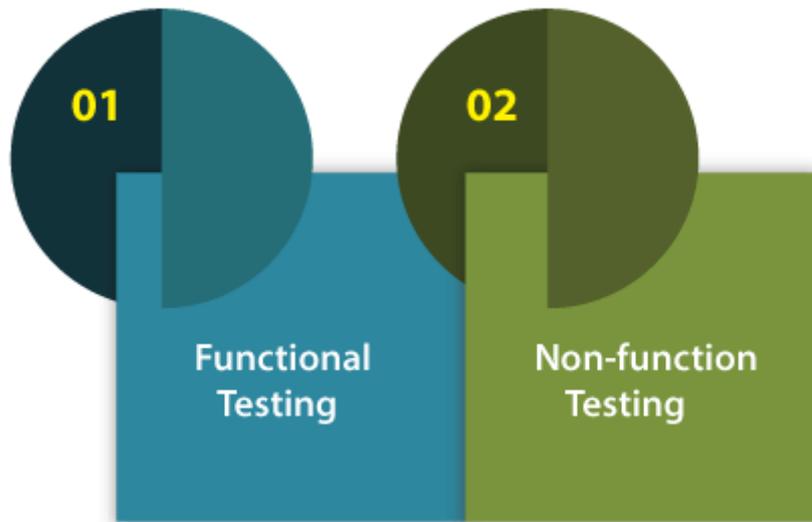
<https://www.javatpoint.com/black-box-testing>。

### 3. 黑盒测试的类型

黑盒测试进一步分为两部分，如下所述：

- 功能测试
- 非功能测试

## Types of Black Box Testing



### 4. 功能测试

测试工程师将根据需求规范系统地检查所有组件，称为功能测试。功能测试也称为组件测试。

在功能测试中，通过给出值、定义输出并使用预期值验证实际输出来测试所有组件。

功能测试是黑盒测试的一部分，因为它强调应用程序需求而不是实际代码。测试工程师只需要测试程序而不是系统。

要获取有关功能测试的详细信息，请参阅以下链接：

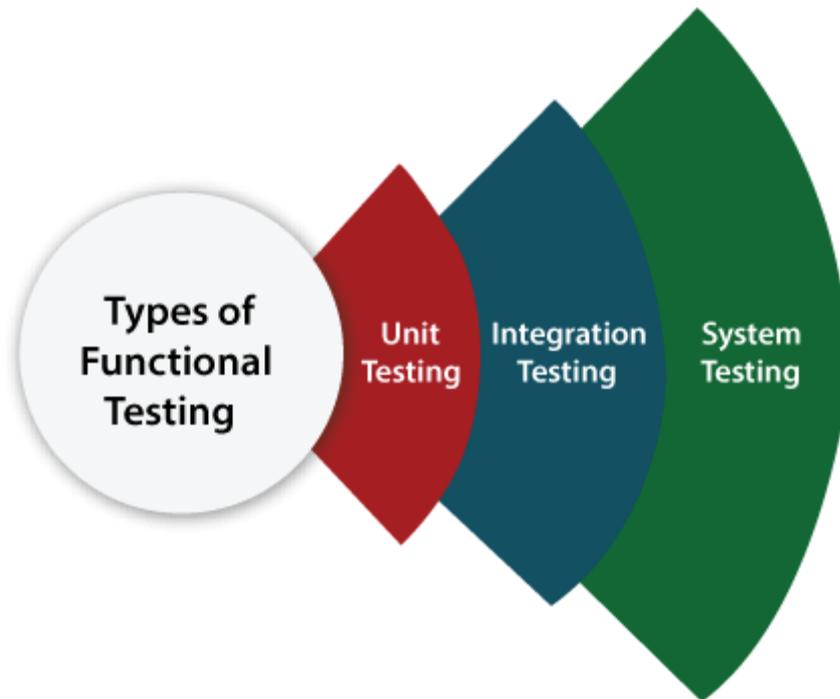
<https://www.javatpoint.com/functional-testing>。

### 5. 功能测试的类型

就像另一种类型的测试分为几个部分一样，功能测试也分为不同的类别。

不同类型的功能测试包含以下内容：

- 单元测试
- 集成测试
- 系统测试



现在，让我们——了解它们：

## 5.1 单元测试

单元测试是功能测试的第一级，用于测试任何软件。在这种情况下，测试工程师将独立测试应用程序的模块或测试所有模块功能称为单元测试。

执行单元测试的主要目标是确认单元组件的性能。在这里，单元被定义为软件或应用程序的单个可测试功能。并在整个指定的应用程序开发阶段进行验证。

单击下面的链接以获取有关单元测试的完整信息：

<https://www.javatpoint.com/unit-testing>。

## 5.2 集成测试

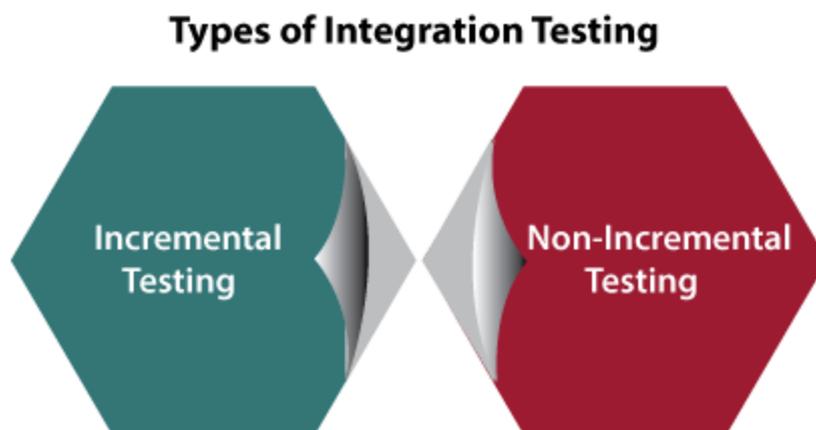
一旦我们成功实现了单元测试，我们将进行集成测试。它是功能测试的第二级，我们测试依赖模块之间的数据流或两个功能之间的接口称为集成测试。

执行集成测试的目的是测试每个模块之间的语句准确性。

## 5.3 集成测试的类型

集成测试还进一步分为以下几个部分：

- 增量测试
- 非增量测试



## 5.4 增量集成测试

每当模块之间存在明确的关系时，我们都会进行增量集成测试。假设，我们采用两个模块并分析它们之间的数据流是否正常工作。

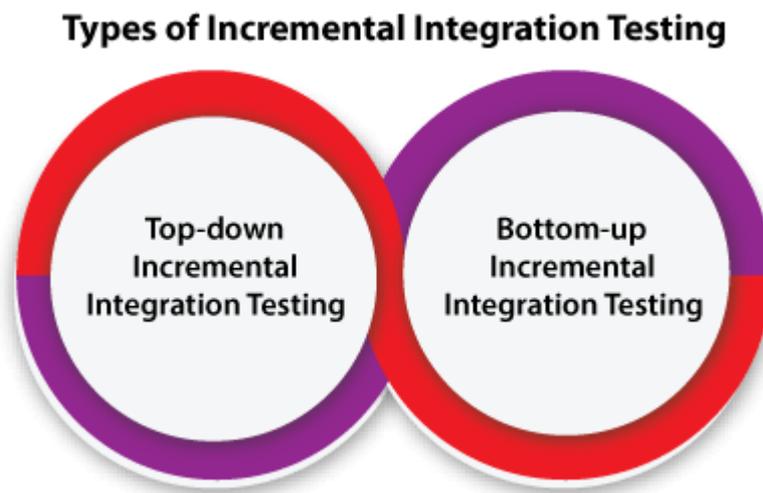
如果这些模块工作正常，那么我们可以再添加一个模块并再次测试。我们可以继续相同的过程以获得更好的结果。

换句话说，我们可以说增量添加模块并测试模块之间的数据流称为增量集成测试。

## 增量集成测试的类型

增量集成测试可以进一步分为两部分，具体如下：

1. 自上而下的增量集成测试
2. 自下而上的增量集成测试



让我们简要介绍一下这些类型的集成测试：

1. 自上而下的增量集成测试

在这种方法中，我们将逐步或增量添加模块，并测试它们之间的数据流。我们必须确保我们添加的模块是早期模块的子模块。

2. 自下而上的增量集成测试

在自下而上的方法中，我们将增量添加模块并检查模块之间的数据流。并且，请确保我们要添加的模块是早期模块的父模块。

## 5.5 非增量集成测试/大爆炸方法

每当数据流复杂且很难对父项和子项进行分类时，我们将采用非增量集成方法。非增量方法也称为大爆炸方法。

## 5.6 系统测试

每当我们完成单元和集成测试时，我们就可以继续进行系统测试。

在系统测试中，测试环境与生产环境并行。它也被称为端到端测试。

在这种类型的测试中，我们将经历软件的每个属性，并测试最终功能是否根据业务需求工作。并将软件产品作为一个完整的系统进行分析。

# 四、非功能测试

黑盒测试的下一部分是非功能测试。它提供了有关软件产品性能和使用技术的详细信息。

非功能性测试将帮助我们最大限度地降低软件的生产风险和成本。

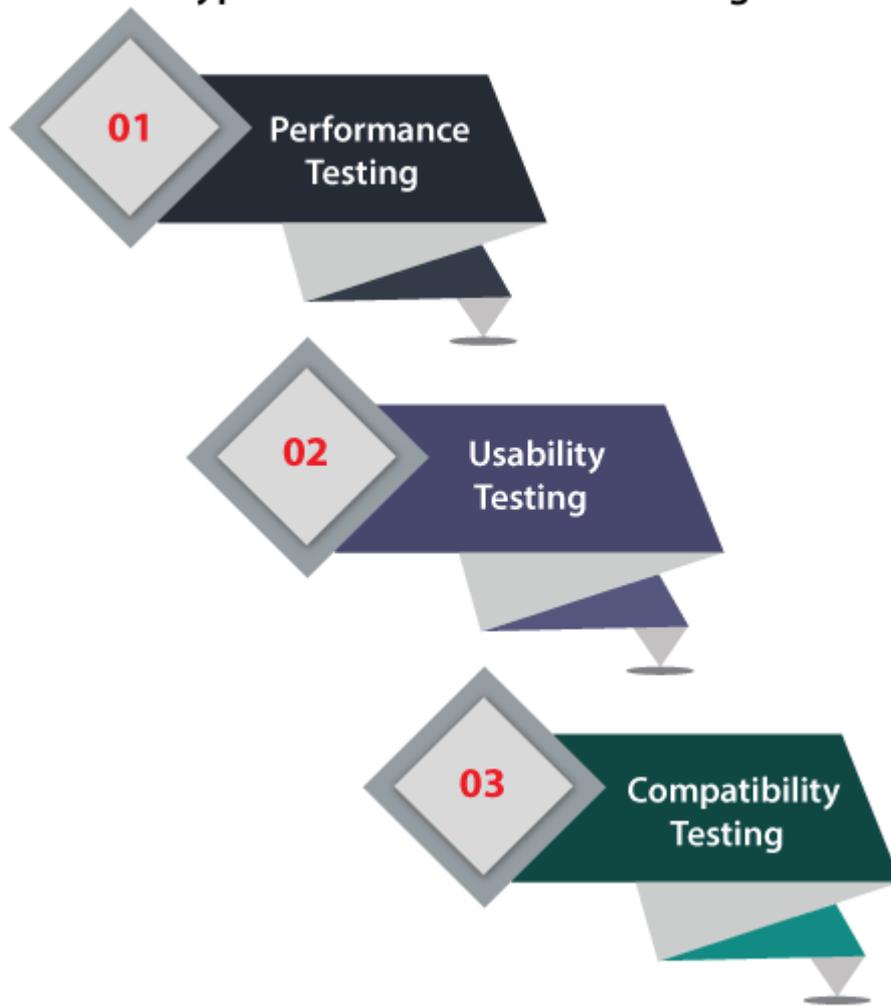
非功能测试是性能、负载、压力、可用性和兼容性测试的组合。

## 6. 非功能性测试的类型

非功能性测试分为测试的不同部分，我们将进一步讨论：

- 性能测试
- 可用性测试
- 兼容性测试

## Types of Non-functional Testing



### 6.1 性能测试

在性能测试中，测试工程师将通过施加一些负载来测试应用程序的工作。

在这种类型的非功能性测试中，测试工程师只会关注几个方面，例如响应时间、负载、可扩展性以及软件或应用程序的稳定性。

性能测试的分类

性能测试包括各种类型的测试，如下所示：

- 负载测试
- 压力测试
- 可扩展性测试
- 稳定性测试

### Classification of Performance Testing



- 负载测试

在执行性能测试时，我们将在特定应用程序上应用一些负载来检查应用程序的性能，称为负载测试。在这里，负载可以小于或等于所需的负载。

它将帮助我们检测软件的最高操作量和瓶颈。

要获取与负载测试相关的完整信息，请参阅以下链接：

<https://www.javatpoint.com/load-testing>。

- 压力测试

它用于分析超出常见功能限制的软件的用户友好性和稳健性。

压力测试主要用于关键软件，但也可用于所有类型的软件应用程序。

请参阅以下链接以获取压力测试的深入知识：

<https://www.javatpoint.com/stress-testing>。

- 可扩展性测试

为了进行分析，通过增强或减少特定平衡中的负载来增强应用程序的性能称为可伸缩性测试。

在可扩展性测试中，我们还可以检查系统、进程或数据库满足向上需求的能力。在这方面，测试用例被有效地设计和实现。

单击以下链接以获取与可伸缩性测试相关的详细信息：

<https://www.javatpoint.com/scalability-testing>。

- 稳定性测试

稳定性测试是我们通过在精确的时间内施加负载来评估应用程序性能的过程。

它主要检查应用的恒定性问题和开发产品的效率。在这种类型的测试中，即使在压力大的情况下，我们也可以快速找到系统的缺陷。

要获得有关稳定性测试的详细信息，请参阅以下链接：

<https://www.javatpoint.com/stability-testing>。

## 6.2 可用性测试

另一种类型的非功能性测试是可用性测试。在可用性测试中，我们将分析应用程序的用户友好性并检测软件最终用户界面中的错误。

在这里，术语用户友好性定义了应用程序的以下方面：

- 应用程序应该易于理解，这意味着所有功能都必须对最终用户可见。
- 应用程序的外观和感觉应该很好，这意味着应用程序应该看起来令人愉悦，并让最终用户感觉到使用它。

有关可用性测试的更多信息，我们可以参考以下链接：

<https://www.javatpoint.com/usability-testing>。

## 6.3 兼容性测试

在兼容性测试中，我们将检查应用程序在特定硬件和软件环境中的功能。一旦应用程序功能稳定，那么我们才会进行兼容性测试。

在这里，软件意味着我们可以在不同的操作系统和其他浏览器上测试应用程序，硬件意味着我们可以在不同的大小上测试应用程序。

要全面了解兼容性测试，请参阅以下链接：

<https://www.javatpoint.com/compatibility-testing>。

## 五、灰盒测试

手动测试的另一部分是灰盒测试。它是黑盒和白盒测试的协作。

因为，灰盒测试包括访问用于设计测试用例的内部编码。灰盒测试由懂编码和测试的人执行。



换句话说，我们可以说，如果一个人的团队同时进行了白盒和黑盒测试，则被认为是灰盒测试。

要获取有关灰盒测试的详细信息，我们可以参考以下链接：

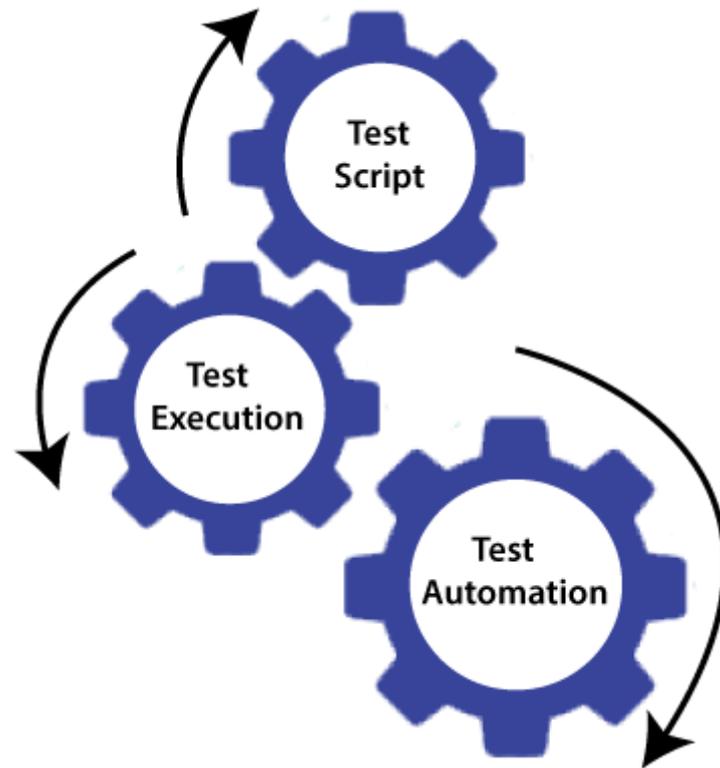
<https://www.javatpoint.com/grey-box-testing>。

## 六、 自动化测试

软件测试最重要的部分是自动化测试。它使用特定的工具来自动化手动设计测试用例，而无需任何人为干预。

自动化测试是提高软件测试的效率、生产力和覆盖范围的最佳方式。

它用于重新运行手动、快速和重复执行的测试方案。



换句话说，我们可以说，每当我们使用某些工具测试应用程序时，都称为自动化测试。

当应用程序或软件上出现各种版本或几个回归周期时，我们将进行自动化测试。如果不了解编程语言，我们就无法编写测试脚本或执行自动化测试。

有关自动化测试的更多信息，我们可以参考以下链接：

<https://www.javatpoint.com/automation-testing>。

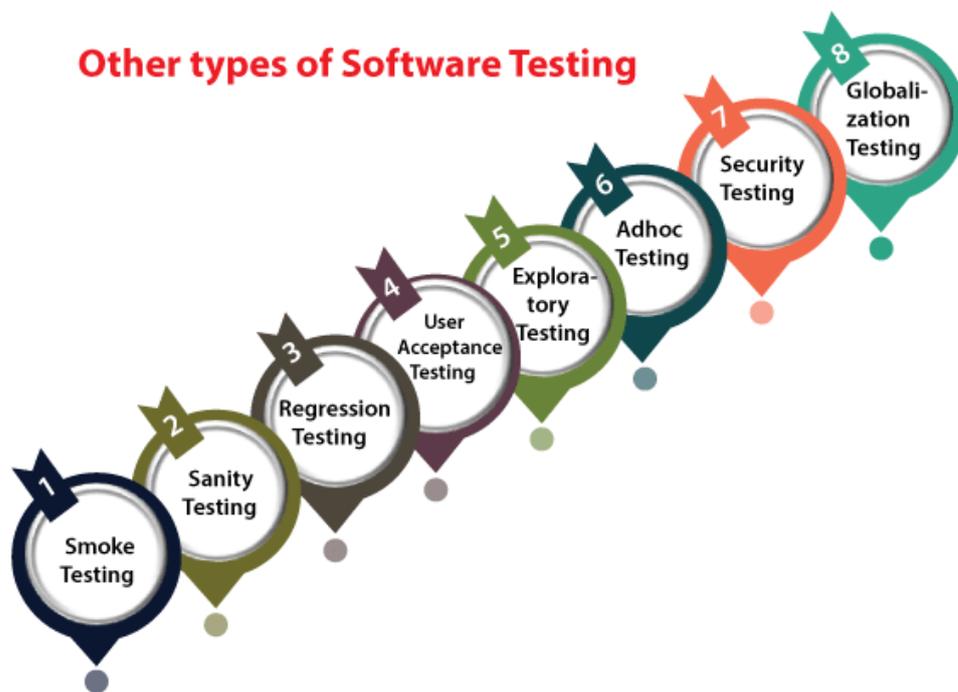
## 七、 一些其他类型的软件测试

在软件测试中，我们还有一些其他类型的测试，它们不属于上述任何测试的一部分，但在测试任何软件或应用程序时都需要这些测试。

- 冒烟测试
- 健全性测试

- 回归测试
- 用户验收测试
- 探索性测试
- 临时测试
- 安全测试
- 全球化测试

让我们——了解这些类型的测试：



在冒烟测试中，我们将测试应用程序的基本和关键功能，然后再进行一轮深入而严格的测试。

或者在检查所有可能的正负值之前称为冒烟测试。分析应用程序核心和主要功能的工作流程是执行冒烟测试的主要目标。

有关烟雾测试的详细信息，请参阅以下链接：

<https://www.javatpoint.com/smoke-testing>。

## 1. 健全性测试

它用于确保所有错误都已修复，并且不会因这些更改而存在任何附加问题。健全性测试是无脚本的，这意味着我们无法记录它。它检查新添加的功能和组件的正确性。

要获取有关健全性测试的详细信息，我们可以参考以下链接：

<https://www.javatpoint.com/sanity-testing>。

## 2. 回归测试

回归测试是最常用的软件测试类型。在这里，术语回归意味着我们必须重新测试未受影响的应用程序的这些部分。

回归测试是最适合自动化工具的测试。根据项目类型和资源的可访问性，回归测试可以类似于重新测试。

每当开发人员修复错误，然后测试由于错误修复而可能模拟的应用程序的其他功能时，称为回归测试。

换句话说，我们可以说，每当某个项目有新版本时，我们就可以执行回归测试，并且由于新功能可能会影响早期版本中的旧功能。

要获得与回归测试相关的全面知识，请参阅以下链接：

<https://www.javatpoint.com/regression-testing>。

### 3. 用户验收测试

用户验收测试 (UAT) 由称为领域专家/客户或客户的个人团队完成。在接受最终产品之前了解应用程序称为用户验收测试。

在用户验收测试中, 我们分析业务场景, 以及称为 UAT 环境的独特环境中的实时场景。在此测试中, 我们将在 UAI 之前测试应用程序以供客户批准。

有关用户验收测试的更多信息, 请单击以下链接:

<https://www.javatpoint.com/acceptance-testing>。

### 4. 探索性测试

每当缺少需求时, 都需要早期迭代, 当我们有关键应用程序时, 测试团队会有经验丰富的测试人员。新的测试工程师进入团队, 然后我们进行探索性测试。

要执行探索性测试, 我们将首先以各种可能的方式浏览应用程序, 制作测试文档, 了解应用程序的流程, 然后测试应用程序。

单击以下链接以获取有关探索性测试的完整信息:

<https://www.javatpoint.com/exploratory-testing>。

### 5. 临时测试

一旦生成按检查顺序随机测试应用程序, 称为临时测试。

它也被称为猴子测试和大猩猩测试。在临时测试中，我们将检查是否符合客户要求的应用程序;这就是为什么它也被称为阴性测试。

当最终用户随便使用应用程序时，他/她可能会检测到错误。尽管如此，专业测试工程师还是会彻底使用该软件，因此他/她可能无法识别类似的检测。

请参阅以下内容以获取有关临时测试的详细信息：

<https://www.javatpoint.com/adhoc-testing>。

## 安全测试

它是软件测试的重要组成部分，用于确定软件应用程序中的弱点、风险或威胁。

安全测试的执行将帮助我们避免来自外部人员的恶劣攻击，并确保我们软件应用程序的安全性。

换句话说，我们可以说安全测试主要用于定义数据将是安全的，并承受软件的工作流程。

要获取有关安全测试的完整详细信息，请参阅以下链接：

<https://www.javatpoint.com/security-testing>。

## 6. 全球化测试

另一种类型的软件测试是全球化测试。全球化测试用于检查开发的软件是否为多种语言。

在这里，全球化一词意味着启发各种语言的应用程序或软件。

全球化测试用于确保应用程序支持多种语言和多种功能。

在当前场景中，我们可以看到几种技术的增强，因为应用程序已准备好在全球范围内使用。

请参阅以下链接以获取与全球化测试相关的完整信息：

<https://www.javatpoint.com/globalization-testing>。

## 八、结论

在本教程中，我们讨论了各种类型的软件测试。但是仍然有超过 100 + 类别的测试列表。

但是，并非所有类型的项目都使用每种测试。

我们已经讨论了最常用的软件测试类型，如黑盒测试、白盒测试、功能测试、非功能测试、回归测试、临时测试等。

此外，在不同的组织中使用了替代分类或流程，但总体概念到处都是相似的。

当项目、要求和范围发生变化时，这些测试类型、流程和执行方法会不断变化。