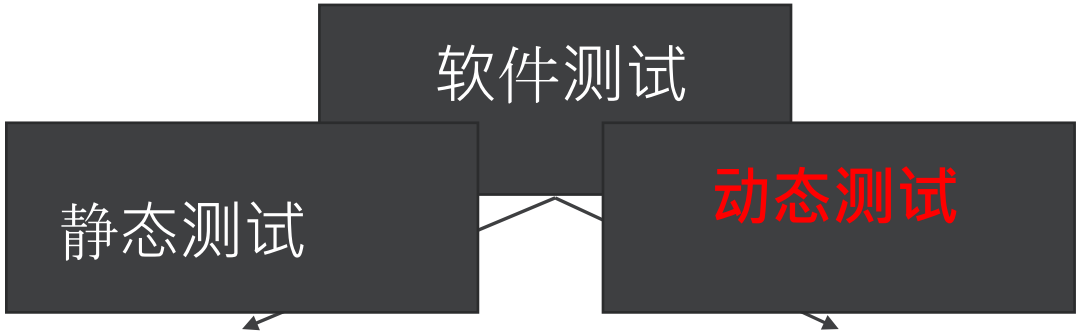
动态测试技术

静态测试与动态测试

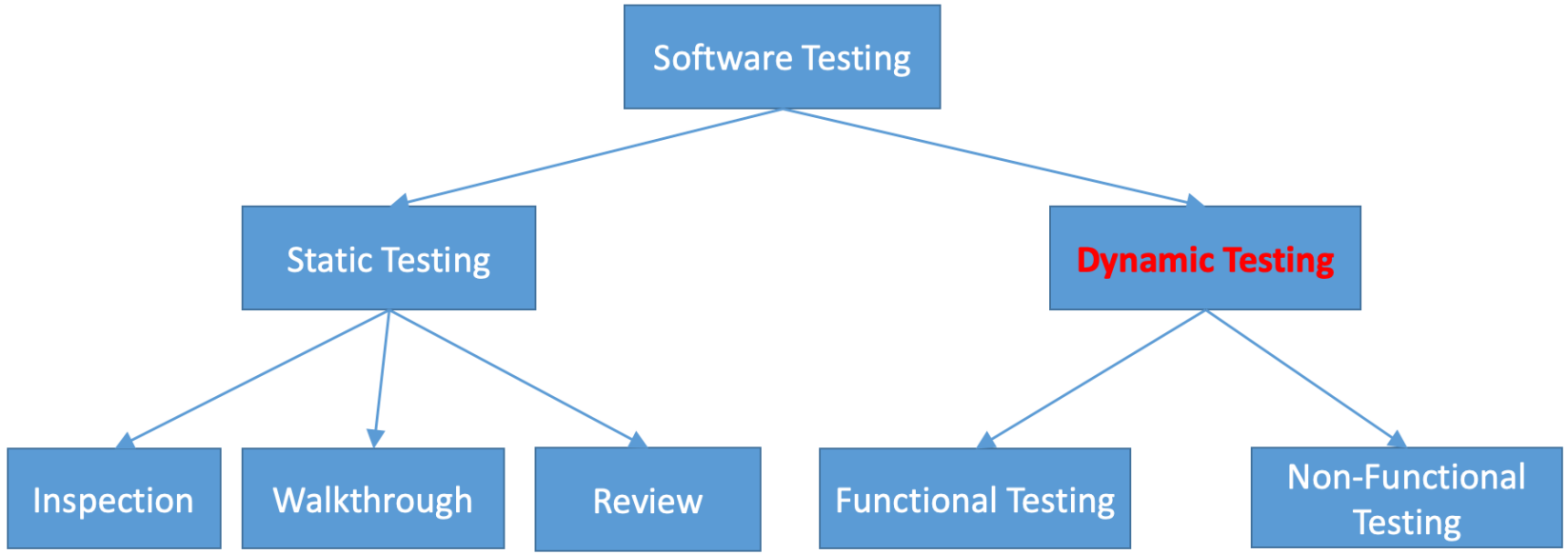


什么是动态测试

- ✓ 在**动态测试**下，一个代码被**执行**。
- ✓ 动态测试检查软件的功能行为
  系统，内存/CPU的使用和系统的整体性能。 的
- ✓ 这种测试的主要目的是确认软件产品的工作**与业务需求相一致**。
- ✓ 动态测试执行软件并验证输出与预期结果。 的

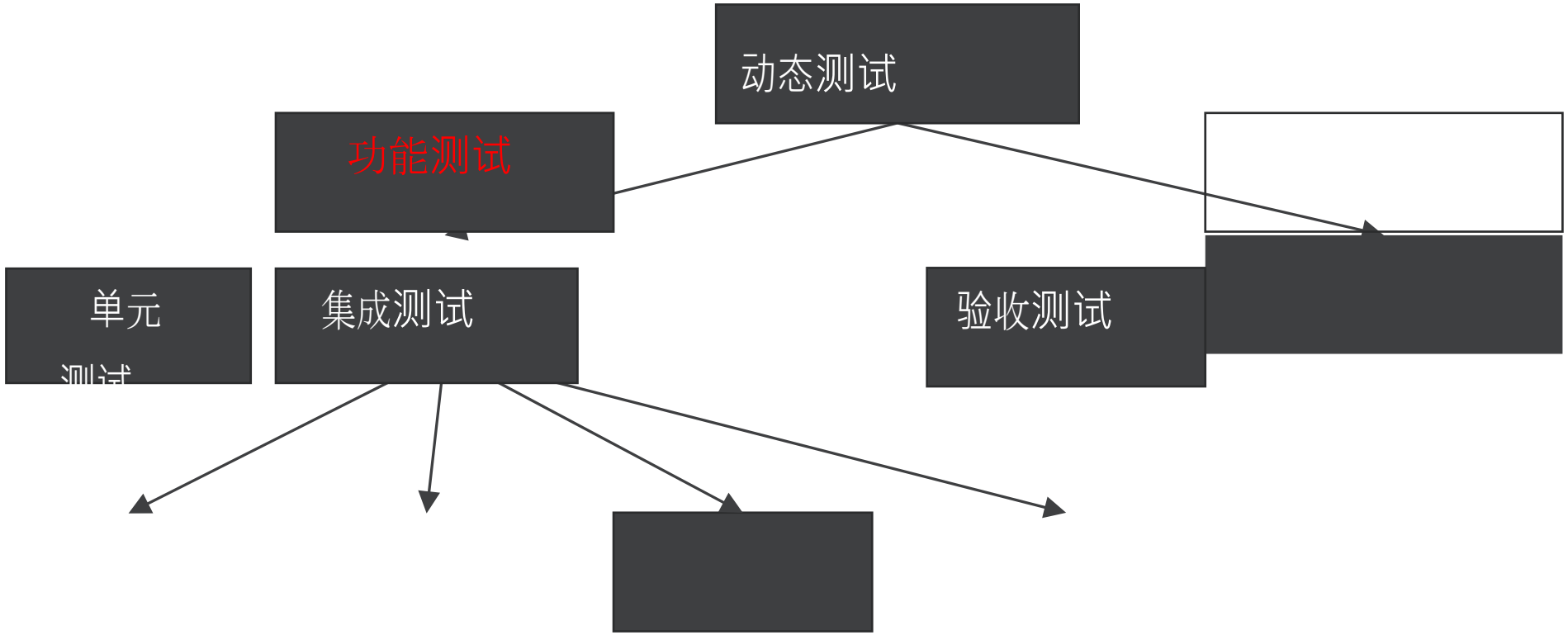
什么是动态测试

✓ 为了执行动态测试的过程，软件测试人员使用两种不同的技术**--功能测试和非功能测试**。



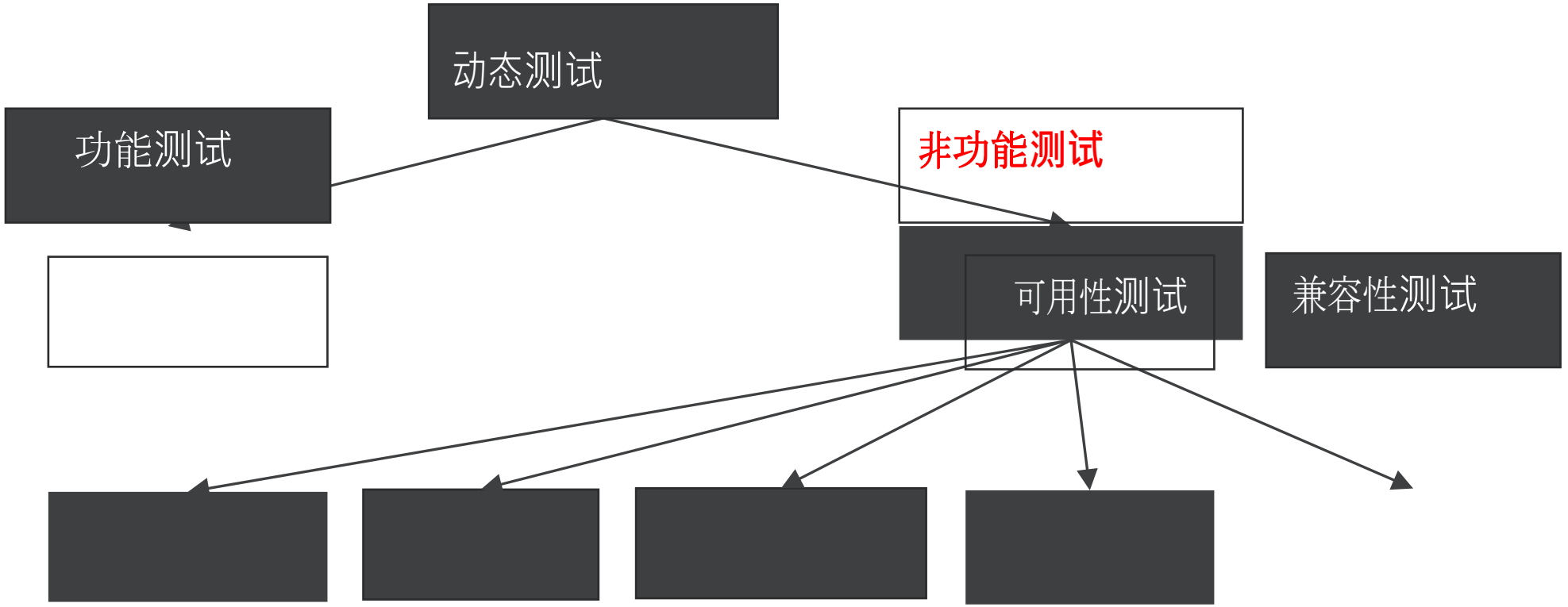
什么是功能测试

- ✓ **功能测试**是软件测试的一种类型，验证了违背功能要求/规范的软件系统。
- ✓ 功能测试的**目的**是**测试软件应用的每个功能**，通过提供适当的输入，根据功能要求验证输出。
- ✓ 功能测试主要涉及黑盒测试，它**不关心应用程序的源代码**。这种测试检查用户界面、API、数据库、安全、客户/服务器通信和其他被测应用程序的功能。


动态测试
功能测试
单元
测试
集成测试
验收测试

什么是非功能测试

- ✓ **非功能测试**被定义为软件测试的一种类型，用于检查**非功能方面（性能、可用性、可靠性。等**）的软件应用。
- ✓ 这种测试技术的重点是**测试软件的非功能属性**，如软件系统的稳健性、内存泄漏、性能等。
- ✓ 非功能测试的一个例子是检查有多少人可以同时登录到一个软件。
- ✓ 非功能测试与功能测试**同样重要**，并影响到客户满意度。


动态测试
功能测试
非功能测试
可用性测试
兼容性测试

问题。

静态测试与动态测试。有什么区别？

Static Testing	Dynamic Testing
Testing was done without executing the program	Testing is done by executing the program
This testing does the verification process	Dynamic testing does the validation process
Static testing is about prevention of defects	Dynamic testing is about finding and fixing the defects
Static testing gives an assessment of code and documentation	Dynamic testing gives bugs/bottlenecks in the software system.
Static testing involves a checklist and process to be followed	Dynamic testing involves test cases for execution
This testing can be performed before compilation	Dynamic testing is performed after compilation
Cost of finding defects and fixing is less	Cost of finding and fixing defects is high

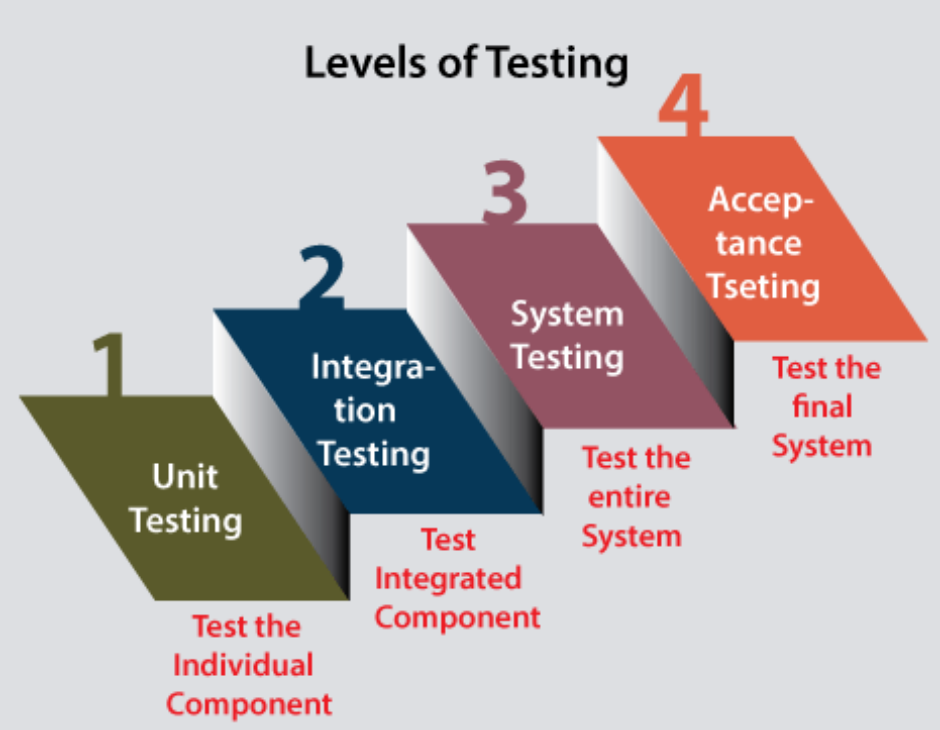
我们今天要讨论的内容

- ✓ 功能测试
  - ► 单元测试
  - ► 集成测试
  - ► 系统测试
  - ► 验收测试

我们今天要讨论的内容

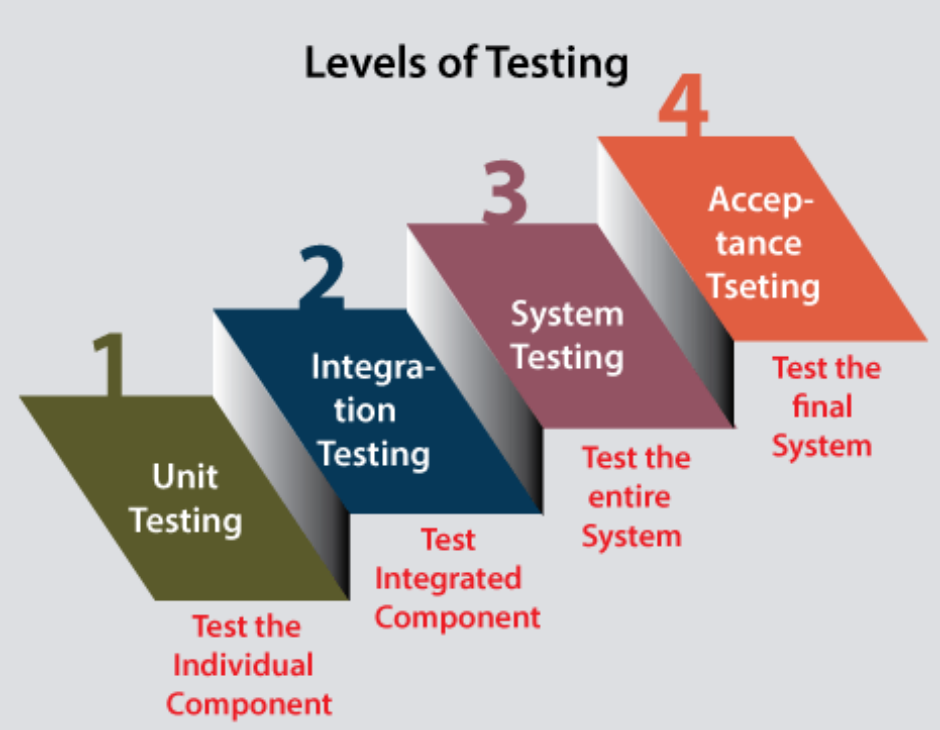
- ✓ 非功能测试
  - ► 性能测试
  - ► 恢复测试
  - ► 安全测试
  - ► 可用性测试
  - ► 兼容性测试

软件测试水平



- ✓ 从**软件开发过程**的角度来看。**软件测试级别**是指软件开发生命周期中进行测试的不同阶段。
- ✓ 在一个程序被批准使用之前，需要完成四个主要层次的测试：**单元测试、集成测试、系统测试和验收测试**。

单元测试

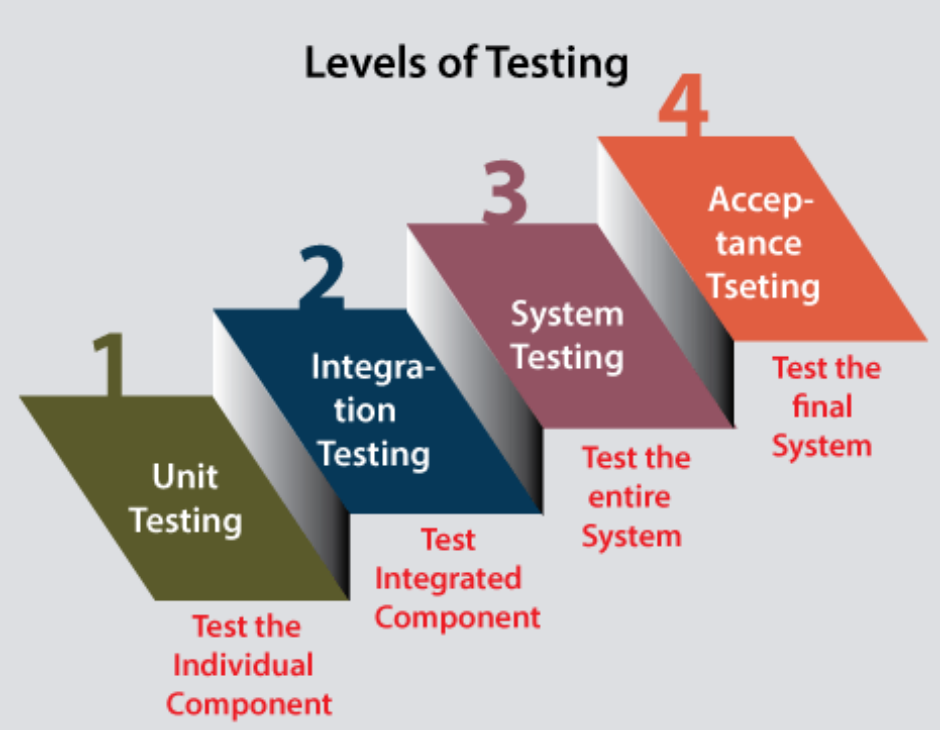


✓ 单元测试

► 最基本的测试类型是单元，或组件测试。

► 单元测试的目的是通过隔离软件的每一部分来验证它，然后进行测试以证明**每个单独的组件**在满足要求和所需功能方面**是正确的**。

单元测试



✓ 单元测试

► 这种类型的测试是在开发过程的**最早阶段**进行的，而且在许多

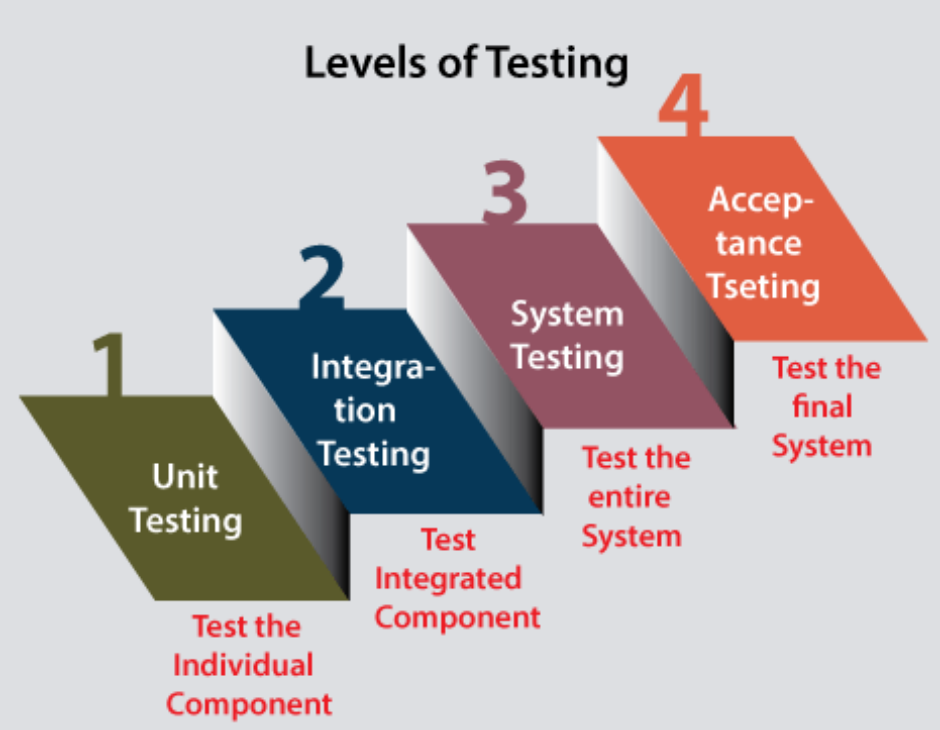
在某些情况下，它是由开发人员自己执行的

在把软件交给测试团队之前，我们要做的事情就是把软件交给测试团队。

► 检测任何错误的优势在于

尽早使用软件的好处是，通过这样做，团队可以将软件开发的风险降到最低，以及在程序接近完成时不得不回头解决程序中的基本问题所浪费的时间和金钱。

集成测试

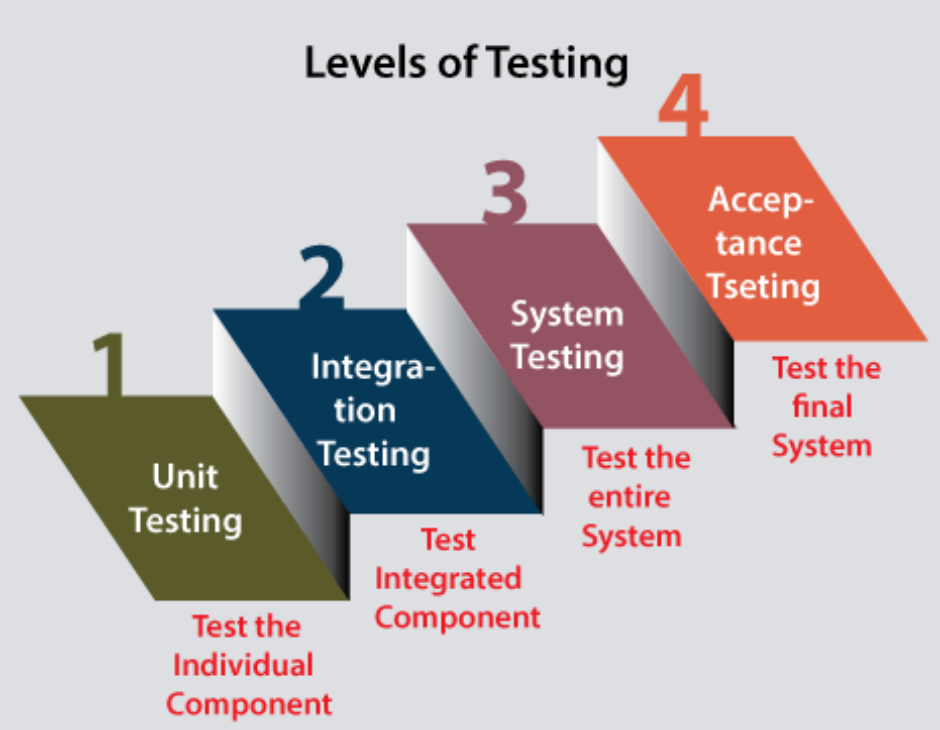


✓ 集成测试

► 集成测试是软件测试过程的第二个层次，在单元测试之后。

► 在这种测试中，软件的单元或单个组件**在一组中被测试**。

集成测试

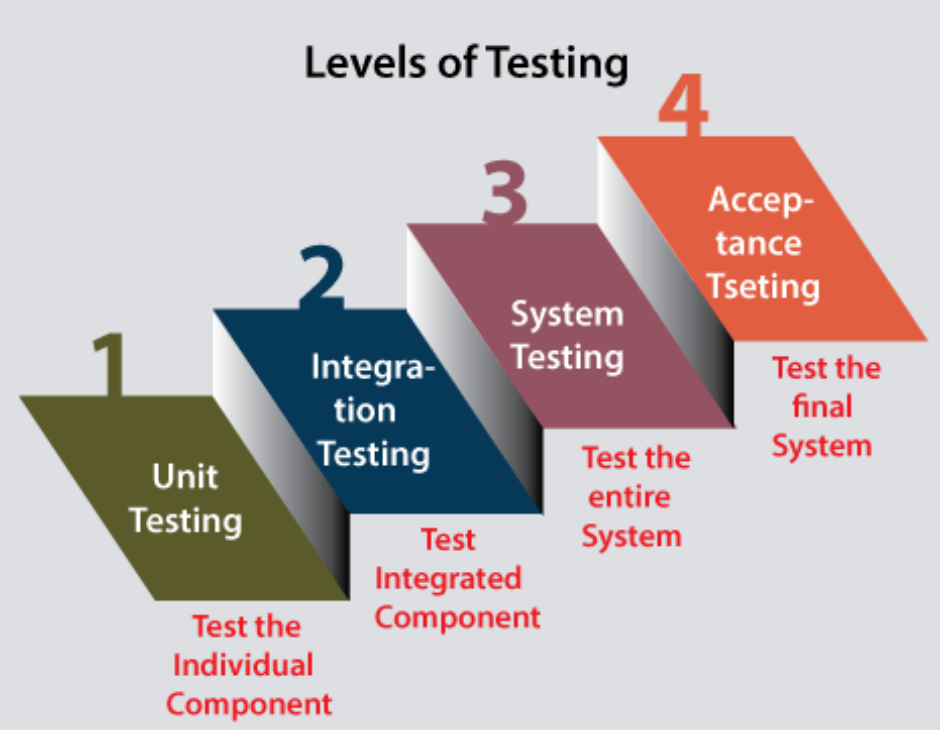


✓ 集成测试

► 集成测试的目的是**发现模块/功能之间的接口缺陷**。这特别有利，因为它决定了各单元在一起运行的效率如何。

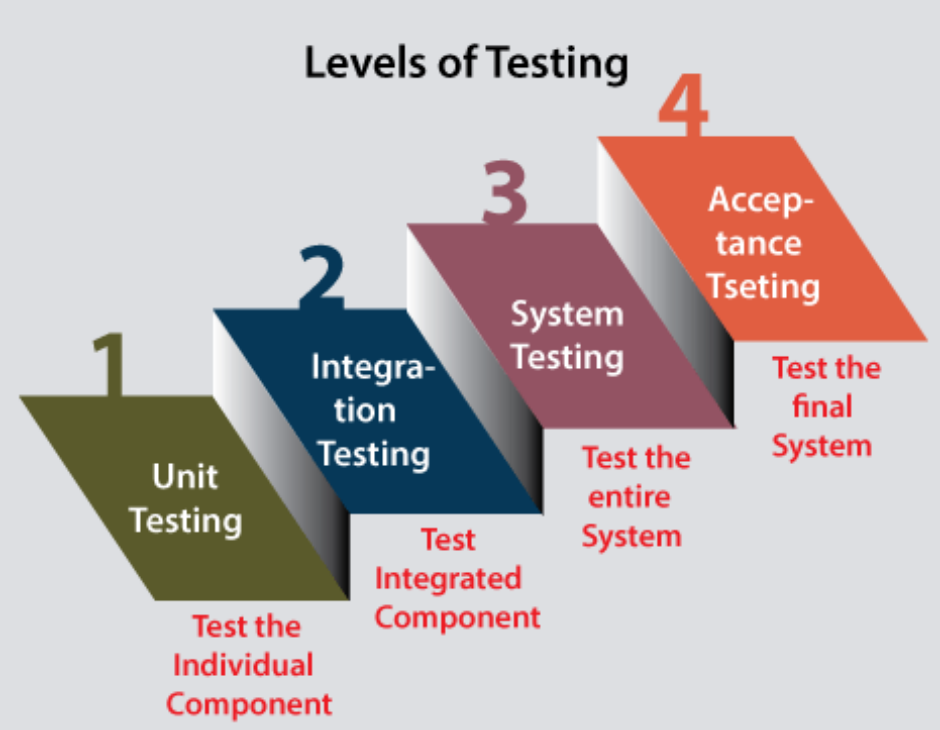
►

集成测试



- ✓ 集成测试
- ► 为什么需要集成测试？
- ► 无论每个单元的运行效率有多高，如果它们没有适当地整合，就会影响软件程序的功能。

集成测试



- ✓ 集成测试
- ► 为什么需要集成测试？

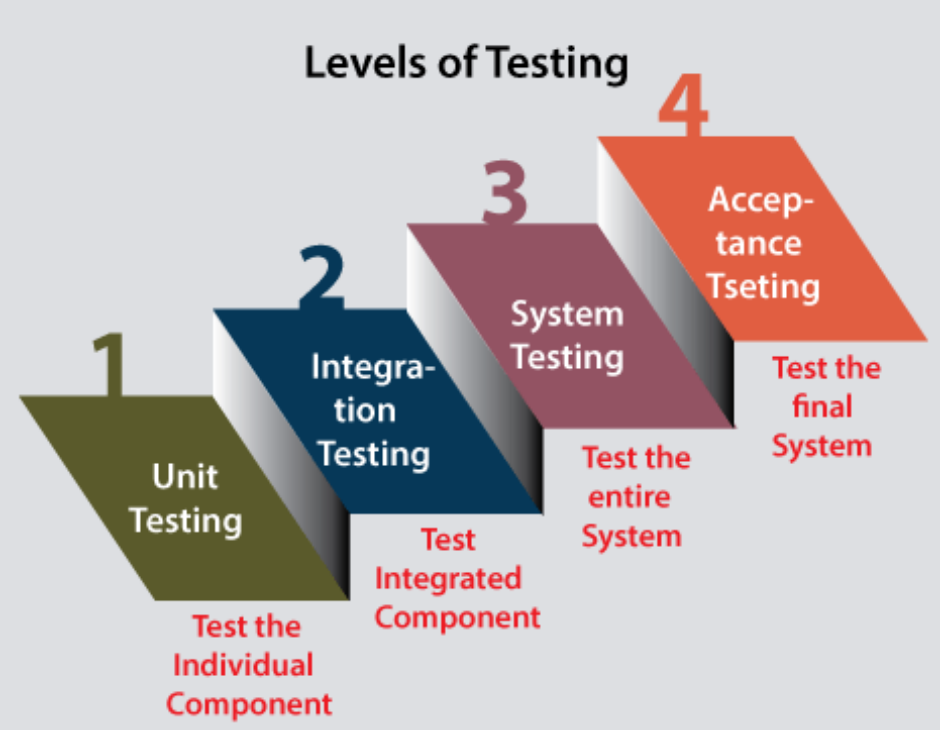
举例说明。

1999年9月，在成功运行了41周和4.16亿英里之后，火星气象轨道器在即将进入火星轨道之前失败了。美国投资50,000美元，以调查其原因。

事故，并发现当太空科学家使用英国（磅）的加速度数据时，美国的实验室使用公制（牛顿）的加速度数据进

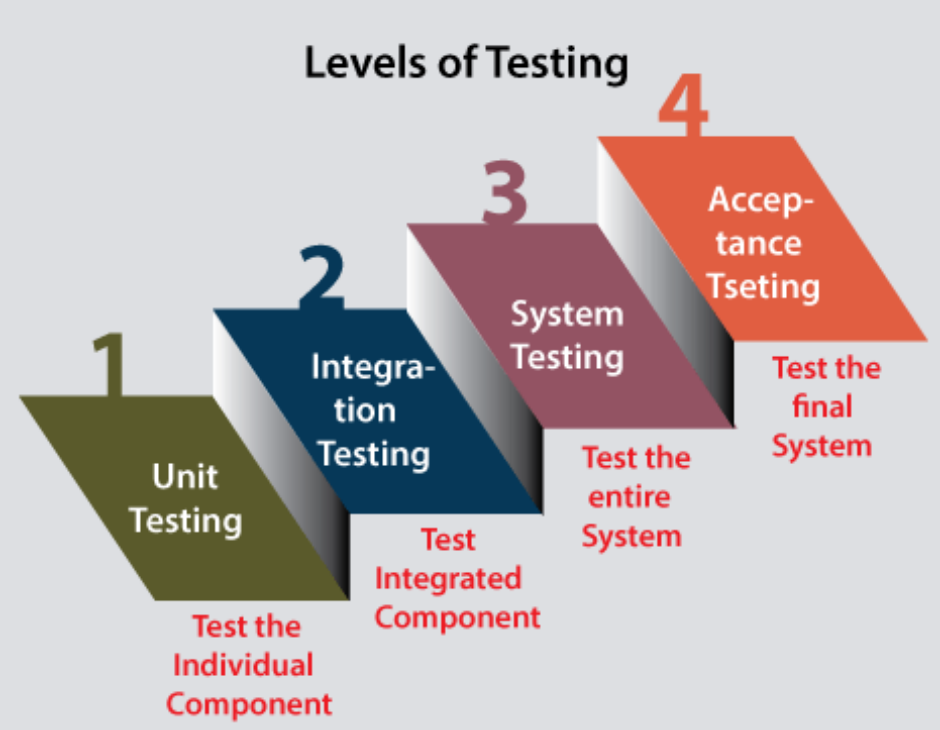
行计算。

系统测试



- ✓ 系统测试
- ► 系统测试是第一个层次，其中**完整的应用程序被作为一个整体进行测试。**
- ► 顾名思义，软件的所有组件都作为一个整体进行测试，以**确保整个产品符合规定的要求。**

验收测试



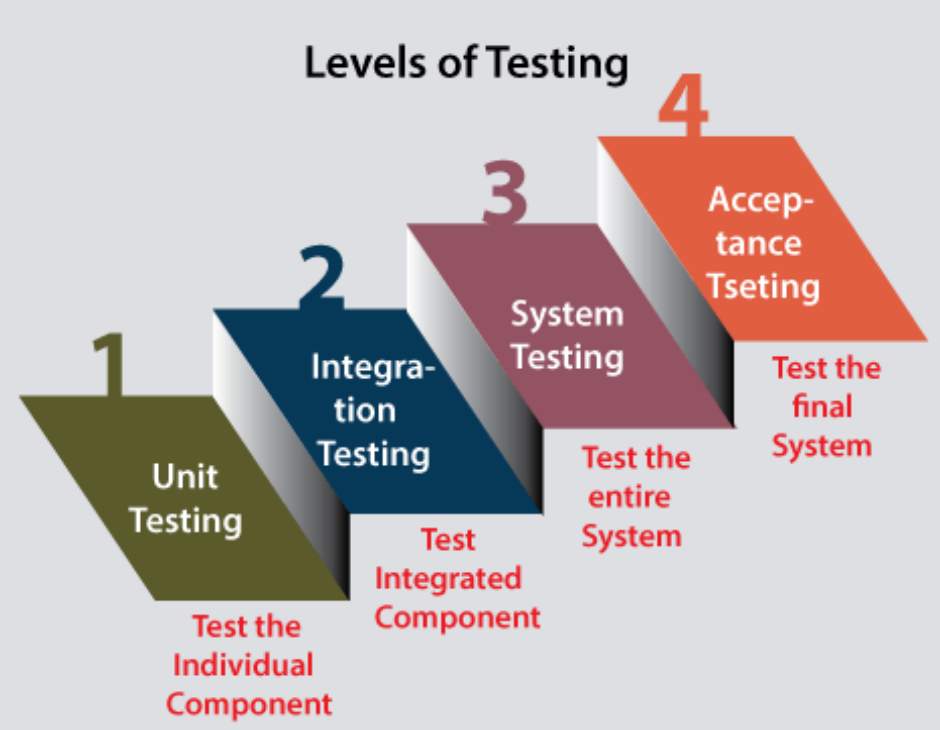
✓ 验收测试

► 最后一个层次，验收测试（或用户验收测试），是为了**确定系统是否可以发布**。

► 在这个最后阶段，用户将对系统进行测试，以了解该应用程序是否符合他们的业务需求。

► 一旦这个过程完成，软件通过，该程序就会被交付生产。

验收测试



- ✓ 验收测试
- ► 验收测试有两个阶段，**Alpha测试**和**Beta测试**。

验收测试

✓ 阿尔法测试通常由**内部员工**进行，在实验室环境中进行。

✓ 阿尔法测试确保产品真正工作，并**做它应该做的一切。**

验收测试

- ✓ Beta测试是一个让**真正的用户**在生产环境中使用产品的机会，以便在全面发布之前发现任何错误或问题。
- ✓ Beta测试者是 "真正的 "用户，在生产环境中进行测试。

验收测试

- ✓ Beta测试可以是**公开的**，也可以是**封闭的**。
- ✓ 在公开测试中，**任何人都可以使用该产品**，通常会有一些信息表明该产品处于测试阶段，并给出一个提交反馈的方法。
- ✓ 在封闭测试中，测试仅限于**一组特定的测试者**，这些**测试者**可能由现有客户、早期采用者和/或付费的测试者组成。
- ✓ 测试可以持续一段时间，也可以运行到新的问题不再被报告，所有重要的问题都

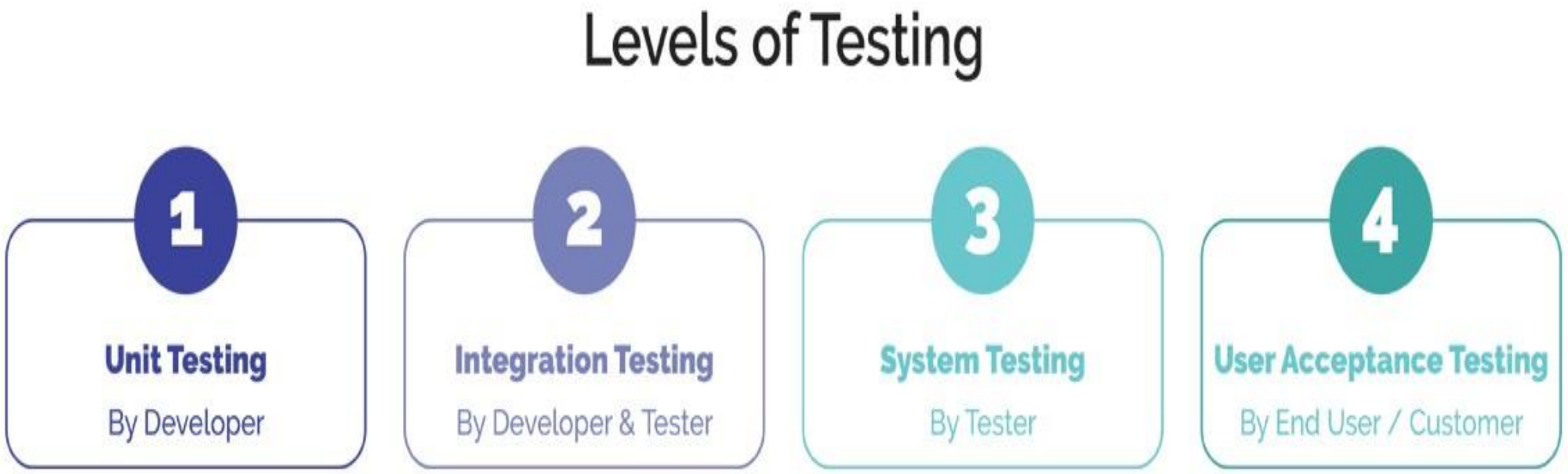
得到解决。

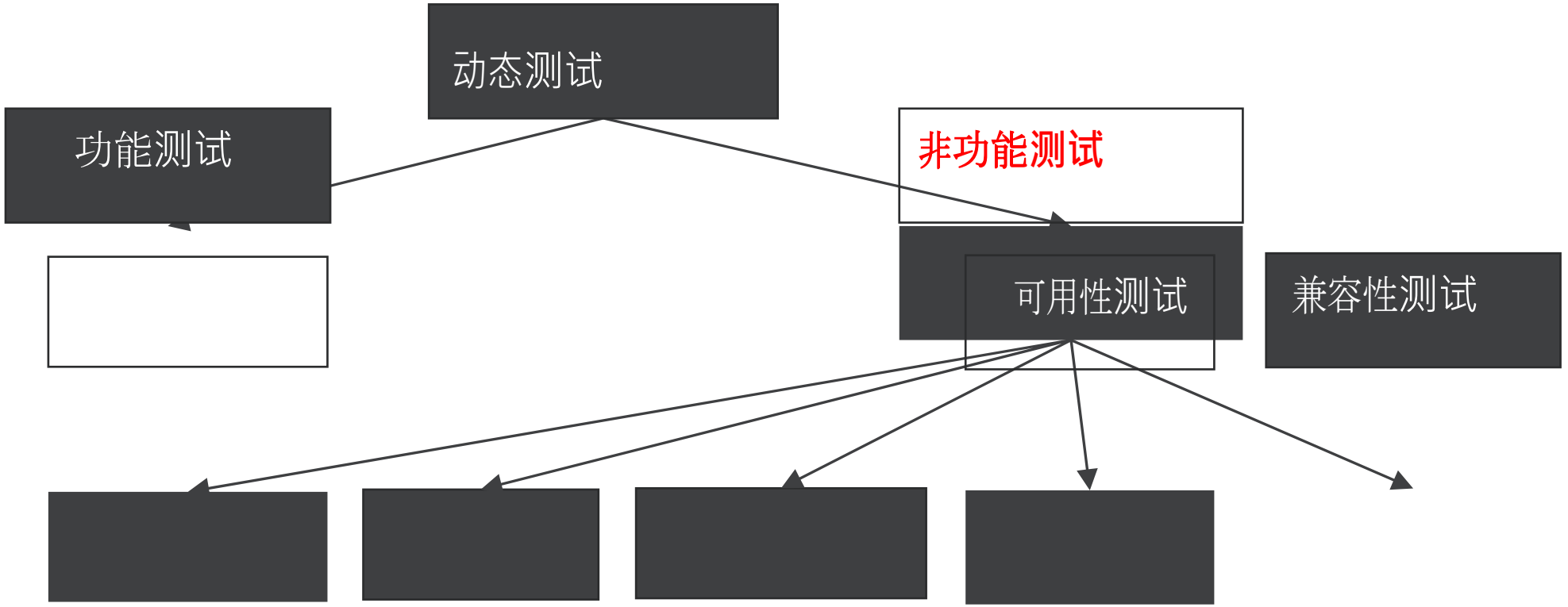
问题。
α测试和β测试的主要区别是什么？

验收测试

- ✓ alpha测试和beta测试的主要区别是**谁在做测试。**
- ✓ 阿尔法测试通常由**内部员工**在实验室环境中进行， 而β测试则由**实际用户**在生产环境中进行。

问题。
在不同的测试级别中，谁进行测试？




动态测试
功能测试
非功能测试
可用性测试
兼容性测试

非功能测试

- ✓ 非功能测试是指软件的工作情况。而功能测试则是验证软件的作用。

非功能测试

- ✓ 性能测试
- ✓ 恢复测试
- ✓ 安全测试
- ✓ 可用性测试
- ✓ 兼容性测试

非功能测试

- ✓ 性能测试
- ✓ 恢复测试
- ✓ 安全测试
- ✓ 可用性测试
- ✓ 兼容性测试

什么是性能测试

- ✓ 性能测试的重点是**一个系统如何在特定的负载下运行**。

  一般来说，这个测试定义了**服务器对用户请求的响应速度**。

投票

最多可选1项

设置

点开朋友圈图片，你最多等几秒？

A 什么垃圾网速，等1秒还打不开就不看了

B 可以接收3-5秒的加载时间。

C 我有好耐性，我可以等10秒

D 为了看到高清无码大图，我愿意一直等到海枯石烂

提交

当我们使用性能测试时

- ✓ 一旦软件稳定并投入生产，我们将进行性能测试，并且可以由多个用户访问。

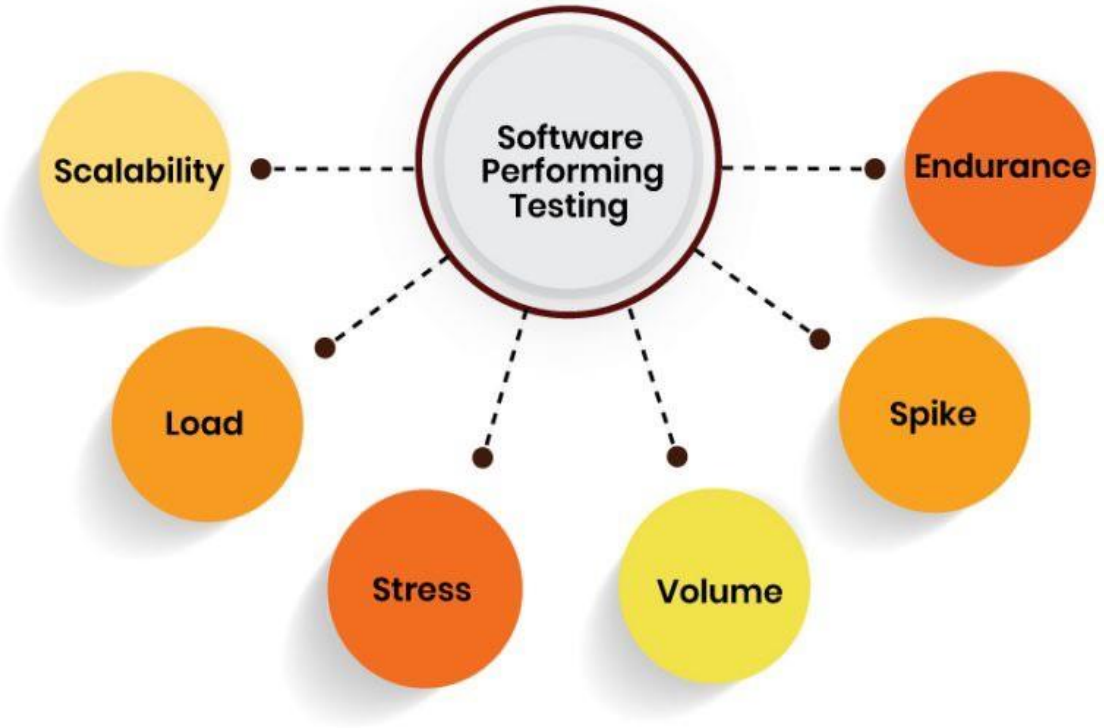
  同时进行，由于这个原因，可能会出现一些性能问题。对

  为了避免这些性能问题，测试人员进行了性能测试。

- ✓ 在应用程序功能稳定的情况下，我们会去进行性能测试。

性能测试的类型

- ✓ 每个公司都根据测试环境使用不同类型的性能测试。这里我们列出了流行的性能测试
  - 。



负载测试

- ✓ 负载测试是用来检查性能的

  通过施加一些负载，使应用程序的

  小于或等于所需的负载，被称为负载测试。

- ✓ 换句话说，它验证了应用程序在**正常和峰值负载条件**下的行为。

负载测试

- ✓ 负载测试确定了软件应用程序在被多个用户同时访问时的表现。
- ✓ 进行负载测试是为了验证我们的应用程序能够满足我们的预期性能目标。

压力测试

的一个

- ✓ 压力测试检查行为

  通过施加**大于期望负荷的负荷**来应用。

- ✓ 换句话说，压力测试检查软件**在非正常情况下**的表现。这就决定了软件会在多大的限度内出现故障。

- ✓ 重要的是要弄清楚当系统处于压力之下时会发生什么。是否显示正确的错误信息？系统会失败吗？它将如何恢复？

耐力测试

- ✓ 耐久性测试是对软件在巨大的**预期负荷**下**持续很长一段时间的**正常工作表现的评估。

问题。

我们为什么要做耐力测试

为什么要进行耐力测试

- ✓ 进行这种测试的主要目的是为了识别任何潜在的**内存泄漏**。因此，在这个测试中，内存利用率被密切监测。
- ✓ 内存泄漏是指软件程序在释放废弃的内存时出现故障，导致性能受损或失败。
- ✓ 内存泄漏可能不会产生短期影响，但从长期来看，它使系统变慢，没有空闲内存，从而最终导致应用程序或系统崩溃。

为什么要进行耐力测试

- ✓ 内存泄漏是一个在一段时间后才变得明显的问题。
- ✓ 因此，我们需要耐力测试来发现这些问题。
- ✓ 可能有这样的情况：在巨大的负载下，你的应用程序在一段时间内工作良好，例如1小时。但是，当连续暴露在相同数量的负载下较长时间，例如3-4小时，你的应用程序就会由于资源问题和磁盘空间不足而崩溃了。

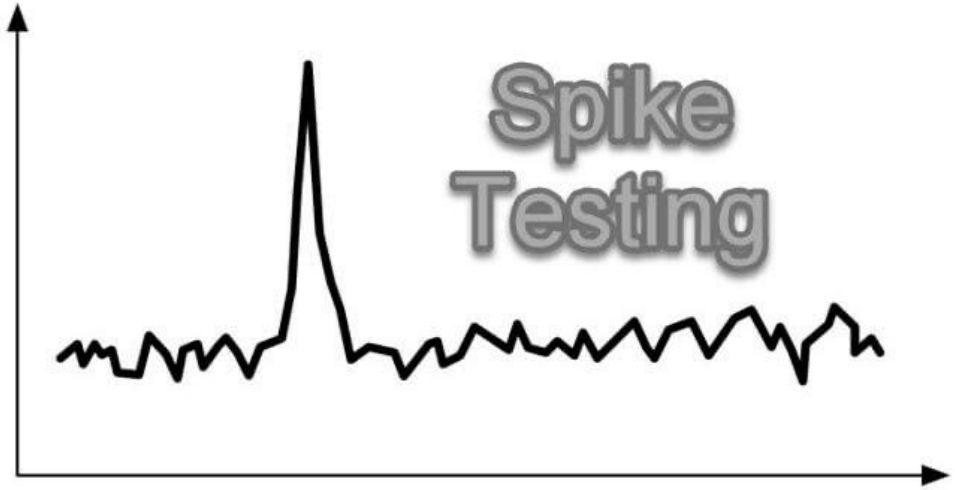
为什么要进行耐力测试

- ✓ 耐久性软件测试所要识别的另一个重要问题是其**性能下降**。需要确保长期使用后的吞吐量或响应时间与测试开始时相当。
- ✓ **数据库连接问题**也在耐力测试中被发现。如果数据库连接没有成功关闭，那么可能会导致系统崩溃。

体积测试

- ✓ 体积测试是在以下情况下进行的一 **高** 种测试 **数据量**。
- ✓ 在批量测试中，对数据量大的软件产品或应用程序进行测试，如系统中大量的输入文件、数据记录或大量的数据库表大小。
- ✓ 它也被称为 "洪水测试"。

钉子测试



- ✓ 在**尖峰测试**中，一个软件应用程序在**流量负载的极端增量和减量下**进行测试。
- ✓ 尖峰测试的目的是看**系统如何对用户负载的意外上升和下降做出反应。**
- ✓ 尖峰测试有助于确定突然出现高负荷时的系统性能恶化。
- ✓ 尖峰测试的另一个目标是**确定恢复时间**。在两个连续的用户负载高峰之间，系统需要一些时间来稳定。
  这个恢复时间应该尽可能的短。

可扩展性测试

- ✓ **可扩展性**测试通过**增加或减少特定规模的负载**来检查一个应用程序的性能。
- ✓ 规模可以是数据量，在特定时间间隔内使用系统的用户数量，等等。
- ✓ 可扩展性测试的基本目标很简单：**确定应用程序在哪一点上停止扩展，然后找出如何修复它**。

非功能测试

- ✓ 性能测试
- ✓ 恢复测试
- ✓ 安全测试
- ✓ 可用性测试
- ✓ 兼容性测试

恢复测试

- ✓ **恢复性测试**是软件测试技术，该技术验证了软件的　　能力**从失败中恢复，如软件/硬件崩溃，网络故障等**。
- ✓ 该技术包括**使**系统**失效**，然后验证系统恢复是否**正常**进行。
- ✓ 为确保系统具有容错性并能从故障中很好地恢复，恢复测试是重要的表现。一个系统被期望从故障中恢复，并在预先规定的时间内恢复其

工作。

恢复测试

- ✓ **为了进行恢复测试，软件/硬件被强行失败验证。**
  - ► 如果恢复成功与否。
  - ► 是否可以进行软件的进一步操作。
  - ► 恢复运作所需的时间。 的
  - ► 丢失的数据可以完全恢复或不恢复。
  - ► .....
- ✓ 在进行这项测试之前，要进行备份并保存到一个安全的位置，以避免在数据不能成功恢复的情况下出现任何数据损失。

恢复测试

✓ **应进行恢复测试的常见故障。**

- 网络问题
- 停电
- 服务器没有响应
- 外部设备没有反应
- 无线网络信号损失
- .....

非功能测试

- ✓ 性能测试
- ✓ 恢复测试
- ✓ 安全测试
- ✓ 可用性测试
- ✓ 兼容性测试

安全测试

- ✓ **安全测试**发现了系统的漏洞。
- ✓ 它确保软件系统和应用程序**不存在任何可能导致损失的威胁或风险**。
- ✓ 任何系统的安全测试的重点是找到系统的**所有可能的漏洞和弱点**，这可能会导致信息或组织的声誉损失。

安全测试

- ✓ 安全测试的目标是。
  - ► 识别系统中的威胁。
  - ► 衡量系统的潜在漏洞。
  - ► 帮助检测系统中每一个可能的安全风险。
  - ► 帮助开发者通过编码修复问题 安全。

非功能测试

- ✓ 性能测试
- ✓ 恢复测试
- ✓ 安全测试
- ✓ 可用性测试
- ✓ 兼容性测试

可用性测试

- **可用性测试**也被称为用户体验(UX)测试，是一**种**衡量**软件应用程序是否方便和用户友好的**测试方法。

可用性测试

- ✓ 美学和设计是很重要的。一个软件应用程序/网站可能由于以下原因而失败------。
  - ► 我应该在哪里点击下一步？
  - ► 哪个页面需要导航？
  - ► 哪个图标代表什么？
  - ► 错误信息不一致或没有有效显示。

可用性测试

✓ 这种测试的目的是为了满足用户，它主要集中在系统的以下参数上。

► **系统的有效性**

- 该系统是否容易学习？
- 所使用的内容、颜色、图标、图像是否具有美感？

► **效率**

- 要到达所需的屏幕或网页，应该很少需要导航，滚动条应该不经常使用。
- 统一你的应用程序/网站中的屏幕/页面的**格式**。

可用性测试

► **准确度**

- 不应存在过期或不正确的数据，如联系信息/地址。
- 不应存在断裂的链接。

► **用户友好性**

- 所使用的控制装置应该是不言自明的，而且必须不需要培训就可以操作
- 应该为用户提供帮助，使其了解应用程序/网站。

例如：我们如何锁定和解锁我们的手机？

非功能测试

- ✓ 性能测试
- ✓ 恢复测试
- ✓ 安全测试
- ✓ 可用性测试
- ✓ 兼容性测试

兼容性测试

- ✓ 兼容性测试检查你的软件是否能够在**不同的硬件、操作系统、应用程序**上运行。

  网络环境或移动设备。

- ✓ 一旦应用程序稳定，我们将其转移到生产中，它可能会被不同平台上的多个用户使用或访问，他们可能会面临一些兼容性问题，为了避免这些问题，我们会做兼容性测试。

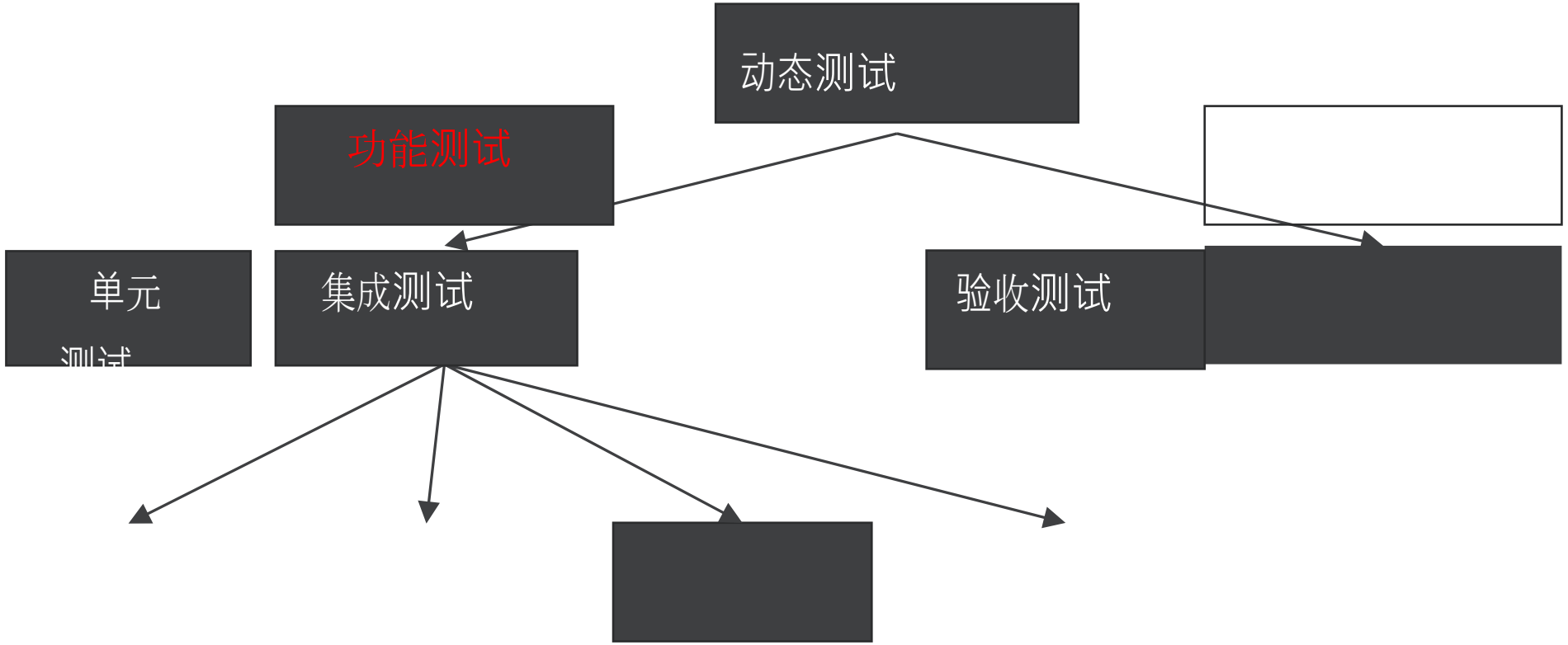
兼容性测试

- ✓ **兼容性测试的例子：**
  - ► 在PC上测试，使用不同的浏览器，如Safari, Chrome, Firefox, IE。
  - ► 在不同的移动设备上测试，这些设备有不同的平台，如iOS、Android或Windows。
  - ► 在4G、3G或WIFI等网络上测试。
  - ► 在多个操作系统上测试，如Mac、Windows、Linux。

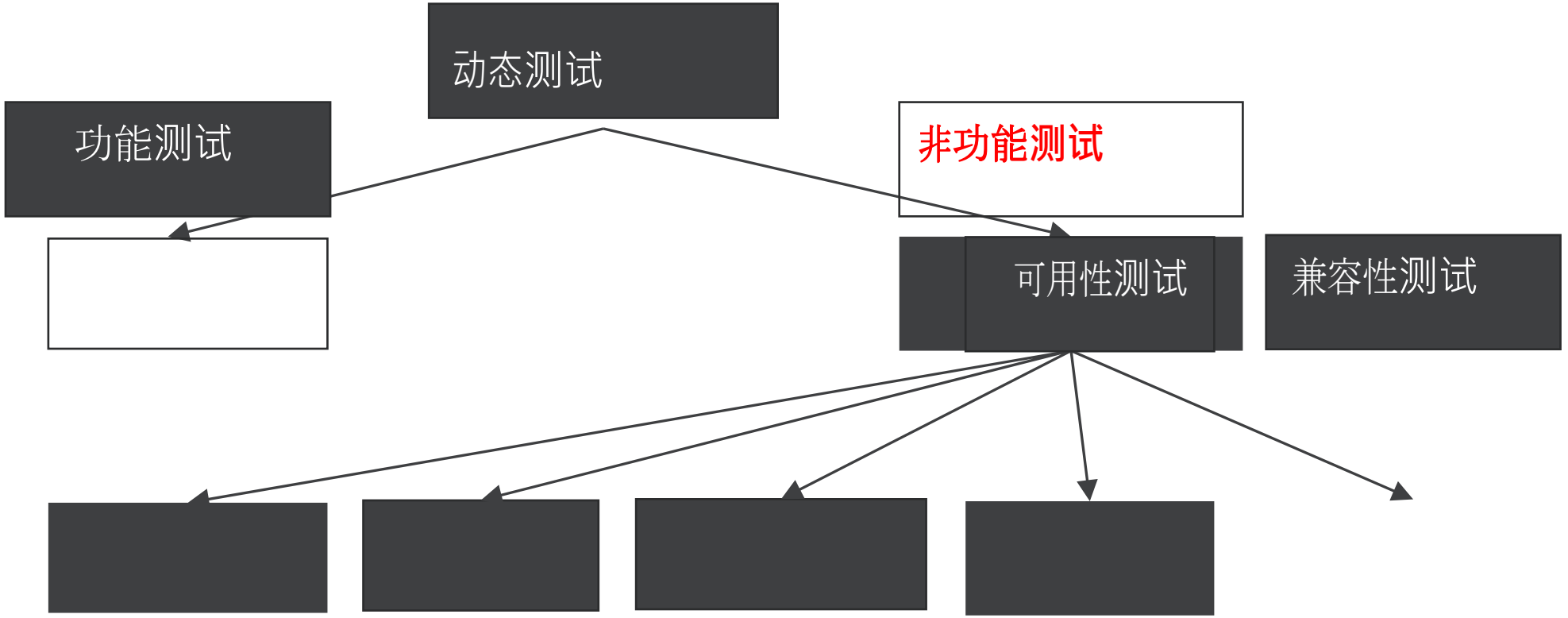
问题。

为什么迪斯尼（狮子王游戏的制作者）没有进行这种测试？

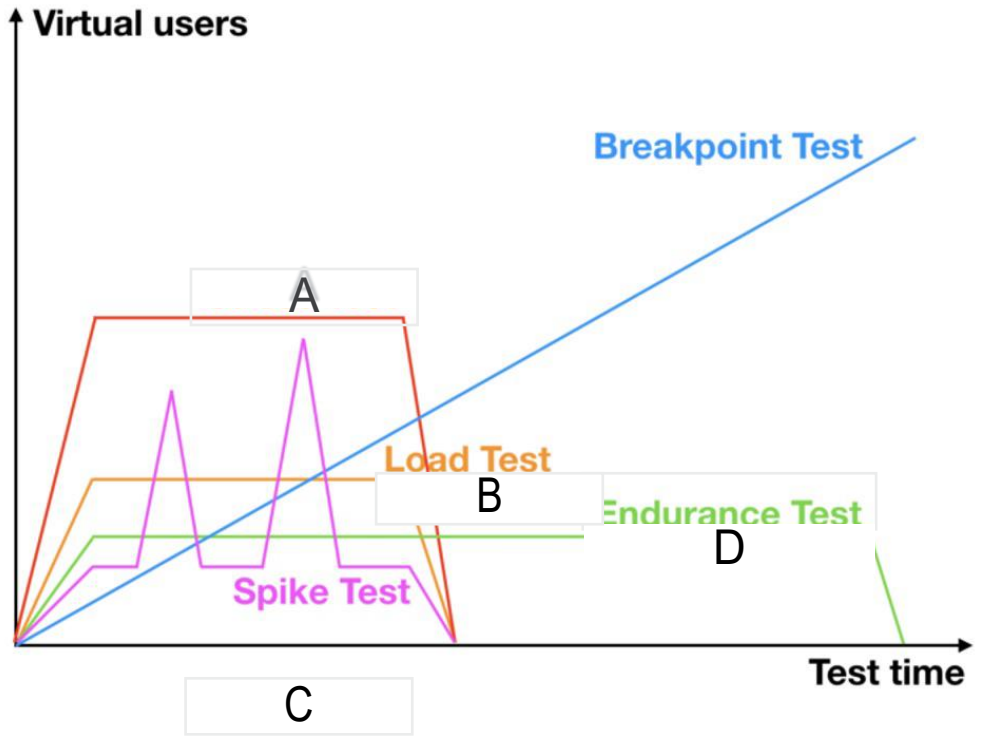
摘要



摘要

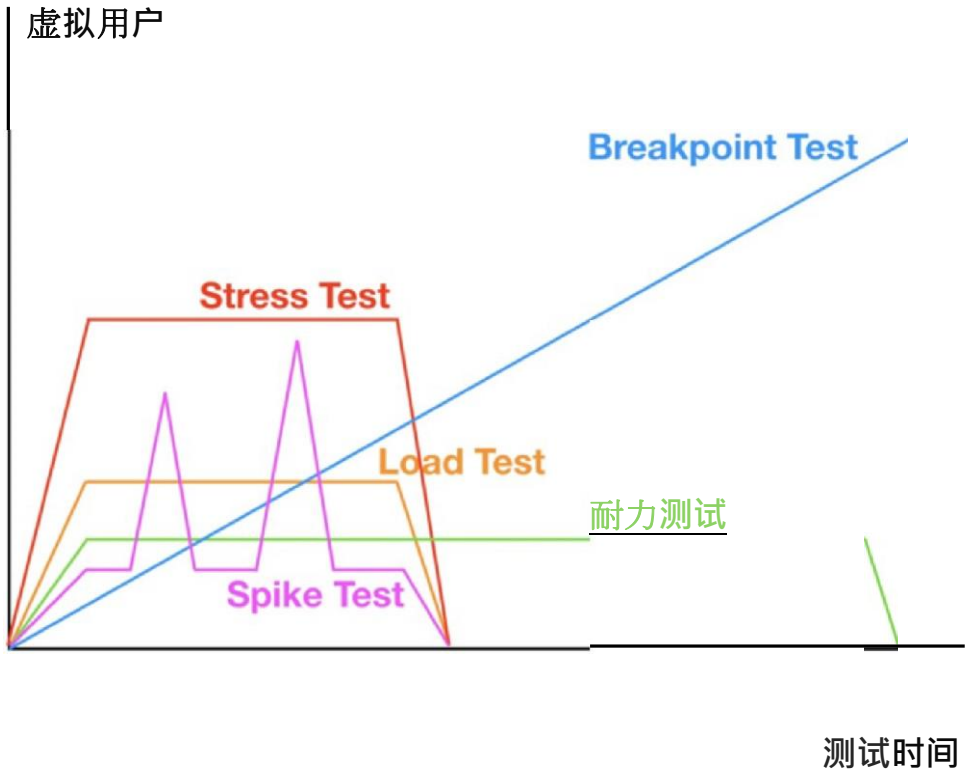


小组讨论



- ✓ 耐力测试
- ✓ 压力测试
- ✓ 负载测试
- ✓ 钉子测试

小组讨论



种类

性能测试

负载测试

olves
under the

负载测试是一种测试类型，涉及评估系统在预期工作负荷下的性能。

压力测试

压力测试是性能测试的一种类型，我们评估应用程序在比预期负载高得多的负载下的性能。

耐力测试

耐力测试也被称为"浸泡测试"。它的目的是确定系统是否能长期维持连续的预期负荷。

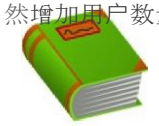
I

钉子测试

在尖峰测试中，我们分析了系统在突然增加用户数量时的行为。

Jlc_

时间



体积测试

体积测试是通过向应用程序输入大量数据来进行的。

检验