Web测试

http://mooctest.net

大纲

- 功能测试概述
- 性能测试概述
- Web测试方法
- 探索式测试方法

功能测试

- 功能测试

  Functional Testing

  根据产品特性和设计需求，验证一个产品的特性和行为是否满足设计需求

功能测试常用步骤

1. 根据需求来细分功能点
2. 根据功能点派生测试需求
3. 根据测试需求设计功能测试用例
4. 逐项执行功能测试用例验证产品

用户界面类型

- 非图形化用户界面
  - 命令行
- 图形化用户界面（GUI）
  - 桌面：Windows风格，单文档、多文档、资源管理器等
  - Web：Html元素等，静态页面，动态页面
  - 移动设备：多触点交互，传感器等
- 其他用户交互方式

相关的测试类型

- 正确性
  - 产品功能是否与需求和设计文档一致
- 可靠性
  - 用户交互是否引发软件崩溃和其它异常
- 易用性
  - 软件产品完成特定任务的难易程度

基于用户界面的功能测试

- 以票务预订系统为例
  - 登录
  - 查询
  - 订票
  - 订单修改
  - 订单取消
  - …

机票查询的测试需求细化

- 路程分类：单程，往返，联程
- 人数分类：1人，多人
- 特殊人群：儿童，婴儿
- 地点选择：国内，国际
- 时间跨度：同月，跨月，跨年等
- 取票方式：
- 舱位：经济舱，商务舱，头等舱

功能测试示例

- 软件需求：功能点
  - 单程国内机票预订
- 测试需求
  - 有直飞航班城市间的预订
  - 无直飞航班城市间的预订
    - 有联程，无联程
  - 单人，多人（带小孩，带婴儿）
  - 有票无票，座位选择等

测试编号	测试环境	输入数据	预期结果	实际结果

性能测试

- 性能测试

  Performance Testing

- 验证产品的性能在特定负载和环境条件下使用是否满足性能指标
- 进一步发现系统中存在的性能瓶颈，优化系统

性能测试度量方法

- 不同的关注对象采用不同的性能的度量方法
- 服务端性能采用CPU、内存等使用率来度量
- 客户端性能通常根据系统处理特定用户请求的响应时间来度量

响应时间

- **响应时间**是指系统对请求作出响应所需要的时间
- **响应时间**划分为：
  - **服务端响应时间**是指从请求发出开始到客户端接收到最后一个字节数据所消耗的时间。
  - **客户端响应时间**是指客户端收到响应数据后呈现/响应用户所消耗的时间。

并发用户数

- 并发用户数取决于测试对象的目标业务场景
- 需要先确定业务场景，然后基于场景采用某些相应方法计算并发用户数
- 并发用户数与同时在线数的区别

吞吐量

- 吞吐量是指单位时间内处理的用户请求数量
- 访问人数/天，页面数/秒，请求数/秒，处理业务数/小时，等等

性能计数器

- 性能计数器是描述系统性能的一些数据指标
- 例如，内存使用率、CPU使用率、进程时间等都是常见服务器性能计数器
- 性能计数器与系统配置情况、系统架构、开发方式等都有密切联系

负载测试

- 负载测试用于验证应用系统在正常负载条件下的行为
- 性能行为通过一些性能指标体现
- 两种方式：
  - 直接到达负载数（如并发用户数，事务数等）
  - 逐步增加负载数（如并发用户数，事务数等）

压力测试

- 压力测试是评估应用系统处于或超过预期负载时的行为
- 压力测试关注的行为不一定是性能行为，可能是某种Bug，比如同步问题，内存泄露等
- 在压力级别逐渐增加时，系统性能应该按照预期缓慢下降，但是不应该崩溃
- 压力测试还可以发现系统崩溃的临界点，从而发现系统中的薄弱环节

Web功能测试方法

1.功能相关性：

- 删除/增加一项会不会对其他项产生影响，如果产生影响，这些影响是否都正确，常见的情况是，增加某个数据记录以后，如果该数据记录某个字段内容较长，可能会在查询的时候让数据列表变形。

2.数据相关性：

- 下拉列表默认值检查，下拉列表值检查，如果某个列表的数据项依赖于其他模块中的数据，同样需要检查，比如，某个数据如果被禁用了，可能在引用该数据项的列表中不可见。

Web功能测试方法

3.检查按钮的功能是否正确

- 如新建、编辑、删除、关闭、返回、保存、导入，上一页，下一页，页面跳转，重置等功能是否正确。常见的错误会出现在重置按钮上，表现为功能失效。

4.字符串长度检查

- 字符串长度检查：输入超出需求所说明的字符串长度的内容，看系统是否检查字符串长度。还要检查需求规定的字符串长度是否是正确的，有时候会出现，需求规定的字符串长度太短而无法输入业务数据。

Web功能测试方法

5. 字符类型检查
   - 在应该输入指定类型的内容的地方输入其他类型的内容(如在应该输入整型的地方输入其他字符类型)，看系统是否检查字符类型。

6. 标点符号检查
   - 输入内容包括各种标点符号，特别是空格，各种引号，回车键。看系统处理是否正确。常见的错误是系统对空格的处理，可能添加的时候，将空格当作一个字符，而在查询的时候空格被屏蔽，导致无法查询到添加的内容。

Web功能测试方法

7．特殊字符检查

- 输入特殊符号，如@、#、$、%、!等，看系统处理是否正确。常见的错误是出现在%‘\这几个特殊字符。

8. 中文字符处理

- 在可以输入中、英文的系统输入中文，看会否出现乱码或出错。

Web功能测试方法

9. 检查信息的完整性

- 在查看信息和更新信息时，查看所填写的信息是不是全部更新，更新信息和添加信息是否一致。要注意检查的时候每个字段都应该检查，有时候，会出现部分字段更新了而个别字段没有更新的情况。

10. 信息重复

- 在一些需要命名，且名字应该唯一的信息输入重复的名字或ID，看系统有没有处理，会否报错，重名包括是否区分大小写，以及在输入内容的前后输入空格，系统是否作出正确处理。

Web功能测试方法

11. 检查删除功能

- 在一些可以一次删除多个信息的地方，不选择任何信息，按“delete”，看系统如何处理，会否出错；然后选择一个和多个信息，进行删除，看是否正确处理。如果有多页，翻页选，看系统是否都正确删除，并且要注意，删除的时候是否有提示，让用户能够更正错误，不误删除。

12. 检查添加和修改是否一致

- 检查添加和修改信息的要求是否一致，例如添加要求必填的项，修改也应该必填；添加规定为整型的项，修改也必须为整型。

Web功能测试方法

13. 检查修改重名

- 修改时把不能重名的项改为已存在的内容，看会否处理，报错。同时，也要注意，会不会报和自己重名的错。

14. 重复提交表单

- 一条已经成功提交的纪录，返回后再提交，看看系统是否做了处理。对于WEB系统来说，可以通过浏览器返回键或者系统提供的返回功能。

Web功能测试方法

15. 检查多次使用返回键的情况
    - 在有返回键的地方，返回到原来页面，重复多次，看会否出错。

16. 搜索检查
    - 有搜索功能的地方输入系统存在和不存在的内容，看搜索结果是否正确. 如果可以输入多个搜索条件，可以同时添加合理和不合理的条件，看系统处理是否正确，搜索的时候同样要注意特殊字符，某些系统会在输入特殊字符的时候，将系统中所有的信息都搜索到。

Web功能测试方法

17. 输入信息位置

- 注意在光标停留的地方输入信息时，光标和所输入的信息会否跳到别的地方。

18. 上传下载文件检查

- 上传下载文件的功能是否实现，上传文件是否能打开。对上传文件的格式有何规定，系统是否有解释信息，并检查系统是否能够做到。下载文件能否打开或者保存，下载的文件是否有格式要求，如需要特殊工具才可以打开等。上传文件测试同时应该测试，如果将不能上传的文件后缀名修改为可以上传文件的后缀名，看是否能够上传成功，并且，上传文件后，重新修改，看上传的文件是否存在。

Web功能测试方法

19. 必填项检查
    - 应该填写的项没有填写时系统是否都做了处理，对必填项是否有提示信息，如在必填项前加“*”；对必填项提示返回后，焦点是否会自动定位到必填项。
20. 快捷键检查
    - 是否支持常用快捷键，如Ctrl+C、 Ctrl+V、 Backspace等，对一些不允许输入信息的字段，如选人，选日期对快捷方式是否也做了限制。
21. 回车键检查
    - 在输入结束后直接按回车键，看系统处理如何，会否报错。这个地方很有可能会出现错误。

Web功能测试方法

22．刷新键检查

- 在Web系统中，使用浏览器的刷新键，看系统处理如何，会否报错。

23．回退键检查

- 在Web系统中，使用浏览器的回退键，看系统处理如何，会否报错。对于需要用户验证的系统，在退出登录后，使用回退键，看系统处理如何；多次使用回退键，多次使用前进键，看系统如何处理。

24．直接URL链接检查

- 在Web系统中，直接输入各功能页面的URL地址，看系统如何处理，对于需要用户验证的系统更为重要。如果系统安全性设计的不好，直接输入各功能页面的URL地址，很有可能会正常打开页面。

Web功能测试方法

25．空格检查

- 在输入信息项中，输入一个或连串空格，查看系统如何处理。如对于要求输入整型、符点型变量的项中，输入空格，既不是空值，又不是标准输入。

26．输入法半角全角检查

- 在输入信息项中，输入半角或全角的信息，查看系统如何处理。如对于要求输入符点型数据的项中，输入全角的小数点（“。”或“．”，如４．５）；输入全角的空格等。

Web功能测试方法

27．密码检查

- 一些系统的加密方法采用对字符Ascii码移位的方式，处理密码加密相对较为简单，且安全性较高，对于局域网系统来说，此种方式完全可以起到加密的作用，但同时，会造成一些问题，即大于128的Ascii对应的字符在解密时无法解析，尝试使用“uvwxyz”等一些码值较大的字符作为密码，同时，密码尽可能的长，如17位密码等，造成加密后的密码出现无法解析的字符。

28．用户检查

- 任何一个系统，都有各类不同的用户，同样具有一个或多个管理员用户，检查各个管理员之间是否可以相互管理，编辑、删除管理员用户。同时，对于一般用户，尝试删除，并重建同名的用户，检查该用户其他信息是否重现。同样，提供注销功能的系统，此用户再次注册时，是否作为一个新的用户。而且还要检查该用户的有效日期，过了有效日期的用户是不能登录系统的。容易出现错误的情况是，可能有用户管理权限的非超级管理员，能够修改超级管理员的权限。

Web功能测试方法

29. 系统数据检查

- 这是功能测试最重要的，如果系统数据计算不正确，那么功能测试肯定是通不过的。数据检查根据不同的系统，方法不同。对于业务管理平台，数据随业务过程、状态的变化保持正确，不能因为某个过程出现垃圾数据，也不能因为某个过程而丢失数据。

30. 系统可恢复性检查

- 以各种方式把系统搞瘫，测试系统是否可正常迅速恢复。

31. 确认提示检查

- 系统中的更新、删除操作，是否提示用户确认更新或删除，操作是否可以回退（即是否可以选择取消操作），提示信息是否准确。事前或事后提示，对于Update或Delete操作，要求进行事前提示。

Web功能测试方法

32．数据注入检查

- 数据注入主要是对数据库的注入，通过输入一些特殊的字符，如“’ ”，“/”，“-”等或字符组合，完成对SQL语句的破坏，造成系统查询、插入、删除操作的SQL因为这些字符而改变原来的意图。如select * from table where id = ‘ ’ and name = ‘ ’，通过在id输入框中输入“12’-”，会造成查询语句把name条件注释掉，而只查询id=12的记录。同样，对于update和delete的操作，可能会造成误删除数据。当然还有其它一些SQL注入方法，具体可以参考《SQL应用高级SQL注入.doc》，很多程序都是基于页面对输入字符进行控制的，可以尝试跳过界面直接向数据库中插入数据，比如用Jmeter，来完成数据注入检查。

Web功能测试方法

33．刷新检查

- web系统中的WebForm控件实时刷新功能，在系统应用中有利有弊，给系统的性能带来较大的影响。测试过程中检测刷新功能对系统或应用造成的影响（白屏），检查控件是否回归默认初始值，检查是否对系统的性能产生较大影响（如每次刷新都连接数据库查询等）。

34．事务检查

- 对于事务性操作，断开网络或关闭程序来中断操作，事务是否回滚。

Web功能测试方法

35．时间日期检查

- 时间、日期验证是每个系统都必须的，如2006-2-29、2006-6-31等错误日期，同时，对于管理、财务类系统，每年的1月与前一年的12月（同理，每年的第1季度与前一年的第4季度）。另外，对于日期、时间格式的验证，如2006年2月28日、2006-2-28、20060228等。日期检查还要检查日期范围是否符合实际的业务，对于不符合时间业务的日期，系统是否会有提示或者有限制

36．多浏览器验证

- 越来越多的各类浏览器的出现，用户访问Web程序不再单单依赖于Microsoft Internet Explorer，而是有了更多的选择：Maxthon、Firefox、Tencent Traveler等，考虑使用多种浏览器访问系统，验证效果。

Web功能测试方法

37．安装测试

- 对于C/S架构的系统，安装程序的测试是一个重要方面，安装程序自动化程度、安装选项和设置（验证各种方案是否都能正常安装）、安装过程中断测试、安装顺序测试（分布式系统）、修复安装及卸载测试。

38．文档测试

- 主要是对用户使用手册、产品手册进行测试，校验是否描述正确、完整，是否与当前系统版本对照，是否易理解，是否二义性等。

39．测试数据检查

- 事实告诉我们，测试数据比代码更有可能是错的，因此，当测试结果显示有错误发生的时候，怀疑代码错误前要先对测试数据检查一遍。

Web功能测试方法

40．请让我的机器来运行

- 在某些项目中，出现一个病态的问题：系统没有问题呀，它在我的机器上是能够通过的。这就说明了其中存在着和环境相关的BUG。“是否所有的一切都受到了版本控制工具的管理？”、“本机的开发环境和服务器的环境是否一样？”、“这里是否存在一个真正的BUG，只不过是在其他的机器里偶然出现？”。所有的测试必须在所有系统要求的机器上运行通过，否则的话，代码就可能存在问题。

41．Ajax技术的应用

- Ajax有很多优点，但也有很多缺点，如果利用优点、避免缺点，是我们对新的Web2.0应用的一个挑战。而Ajax的应用最直接的问题就是用户体验，用户体验的效果直接关系到是否使用Ajax技术。“会做，并不意味着应该做、必须做”，这就是对Ajax技术的很重要的注解。

Web功能测试方法

42．Ajax技术的应用

- Ajax采用异步调用的机制实现页面的部分刷新功能，异步调用存在异常中断的可能，尝试各种方法异常中断异步的数据调用，查看是否出现问题。在这里遇到的一个问题就是对日期控件的操作，已经如果页面数据较多的时候的刷新。

43．脚本错误

- 随着Ajax、IFrame等异步调用技术的发展，Javascrīpt技术也越来越受到开发人员的重视，但Javascrīpt存在调试困难、各浏览器存在可能不兼容等问题，因此在Web系统中，可能会出现脚本错误。同时，脚本错误造成的后果可大、可小

探索式测试方法

- 探索式测试(ET)是一种帮助测试人员如何在需求不完善的情况下尽早发现更多软件质量风险的测试手段。
- 探索式测试的思想和方法源于美国测试专家们测试经验的抽象提取具有高效的质量保障效果。根据软件质量标准不同既可以与传统结构化测试结合使用，也可以100%采用探索式测试，无论哪种方式都会帮助提高单位时间内的测试效率和质量。尤其适合测试新人快速上手发现缺陷。

产生过程

- 目前缺陷的检测有两种方式：自动化测试 和 手工测试。
- 自动化测试：
  - a、有代码维护成本、学习成本，而且测试程序本身也存在问题。
  - b、只要能在合理的场景中使用自动化还是能提高效率的，并不是所有的测试都适合自动化。
- 手工测试：
  - 需手脑并用，发挥聪明才智，才能设计出导致软件失效或符合软件设计效果的真实场景。如果想发现与应用程序业务逻辑相关的缺陷手工测试是最佳选择。
  - 缺点：a、慢 b、无规律 c、不易复现 d、不可反复使用 e、无可借鉴的经验

- 基于以上缺点诞生了手工测试的技术 ------ 探索式测试

探索式测试的特点

- 它是一种测试风格，测试思维，而不是具体的测试技术
- 它强调测试人员的个人自由和责任，其目的是为了持续优化其工作的价值
- 它建议在整个项目过程中，将测试相关学习作为相互支持的活动并行执行
- 它是一种目的明确，过程规范的测试，它精心策划以防万一，同时留有发挥空间，让测试人员随机应变

探索式测试的指导方法

- 局部探索式测试法（小范围的，比如一个对话框等）
  - 在运行任何一个测试用例时都都需要作出很多细微的战术层面决定
- 全局探索式测试法（总体测试，整个程序功能）
  - 基于“漫游”概念，如同一个导游带领旅游团队参观大都市中的一系列著名景点，漫游测试法提供的路线可以指导测试人员如何探索软件的方方面面。
  - 全局测试法对于制定完整的测试策略给出了指导意义
- 混合探索式测试法（把探索式测试的实质和使用脚本的手工测试合并起来）
  - 使用正式脚本可以为探索式测试设立一个明确框架范围，探索式测试可以提高脚本测试的有效性，为脚本中的测试用例提供更多种多样的变化。

局部探索式测试

- 概述：测试人员在测试中了解各种可以变化的东西，就可以更好的进行探索式测试。局部探索性测试从用户输入、状态、代码路径、用户数据、运行环境五个方面考虑软件的变化。
- 用户输入：分为原子输入和抽象输入。需要考虑：输入的变量、输入的组合、输入的顺序
  - 合法输入 和 非法输入
  - 错误处理程序的三种方式，输入筛选器、输入检查、使用异常
  - 常规输入 和 非常规输入
    - 按键组合、特殊字符、保留名称
  - 默认输入 和 用户提供的输入：置空默认
  - 用输出指导输入选择：内部变量、数据结构的初始化

局部探索式测试

- 状态
  - 状态可以是临时的也可以是长久保持的。状态和输入有很大关系，可以使用状态信息来帮助寻找相关的输入，可以使用状态信息来辨识重要的输入序列。
- 代码路径
  - 多用于白盒测试
- 用户数据
  - 用户的真实数据
- 运行环境
  - 各种硬件及软件环境

全局探索式测试

- 漫游测试：使用游客来比喻测试
  - 商业区：用户所要使用的软件特性和功能（热门区域，最重要的功能）。
  - 历史区：就是前版本遗留下的代码。
  - 旅游区：特定的特性和只对新用户非常有吸引力，而老用户不再使用他。
  - 娱乐区：辅助特性和功能。
  - 旅馆区：长时间的运行。
  - 破旧区：令人容易忽略的漏洞。

商业区测试法-用户所要使用的软件特性和功能

- 指南测试法（用户说明书）：要求测试人员通过阅读用户手册并严格按照手册的建议执行操作。（不仅验证了手册所描述的各种特性，同时也验证了用户手册的准确性）
  - 演绎版本：博客测试法：遵循测试第三方的建议来测试
  - 专家测试：根据抱怨者的方法来编写测试用例
- 卖点测试法：观摩销售演示，根据产品演示步骤来自己执行一遍将吸引用户的特性重点测试
  - 演绎版本：质疑测试法：用户不断的要求这样那样做，打断原有的顺序及做法。
- 地标测试法：首先选择地标，确定他们的先后顺序，然后从一个地标执行到另一个地标来探索应用程序，直到访问完所有的地标。（此过程中，需要记录已经使用过那些地标，并创建一个地标来覆盖图来标识工作的进展）

商业区测试法-用户所要使用的软件特性和功能

- 极限测试法：通过某些特定的输入和内部数据来挑战软件的运行能力。
  - 演绎版本：麻烦测试法：测试人员故意设置各种障碍来看软件的应付能力。
- 快递测试法：测试人员必须专注于数据，应该确认那些被存储起来的数据并“跟随”它们走遍软件。
- 深夜测试法：在特定的时间里（除正常的运营时间）运行程序
- 遍历测试法：选定一个目标（例如所有菜单项，错误消息和对话框），然后使用可以发现的最短路径来访问目标包含的所有对象。

历史区测试法-遗留代码

- 恶邻测试法：测试缺陷横行的代码（随着测试的深入，发现和报告越来越多的缺陷，把这些缺陷数目同产品特性联系起来，从而跟踪究竟在哪些地方会出现产品的缺陷），这是由于缺陷经常扎堆出现。
- 博物馆测试法：查看代码库、程序二进制文件的创建日期，找到那些很长时间都没有动过的遗留代码。因为这些代码容易发生失效的情况。
- 上一版本测试法：验证用户已经熟悉的功能可以继续使用，新添加或删除的功能也应该在根据用户手册测试。

娱乐区测试法-辅助特性和功能

- 配角测试法：紧邻重要功能的的这些容易被人注意，应该给予这些特性足够的重视。
- 深巷测试法：测试哪些不可能被用到的或是哪些最不吸引用户的特性。
- 通宵测试法：测试应用程序能够坚持的时间（多长时间能够持续运行，处理数据不崩溃）。让程序一直保持运行状态，而不区关闭它。

旅游区测试法-特定的特性

- 收藏家测试法：建议我们收集软件的输出（越多越好），测试员到达哪些所有能够到达的地方并把观察的结果记录下来。
- 长路径测试法：访问离应用程序尽可能原的特性。哪个特性需要经过最多的界面才能访问，选定它然后进行测试
- 测一送一法：它只测试同时运行同一应用程序多个拷贝的情况。例如用不同拷贝同时打开同一个文件等。如果你在一个拷贝上发现了一个缺陷，你就在所有拷贝上发现同样的缺陷
- 苏格兰酒吧测试法：与用户进行沟通，并参与他们的讨论

旅馆区测试法-长时间的运行

- 取消测试法：启动操作然后停止它。
- 懒汉测试法：接受所有默认值，保持输入字段继续为空。测试人员没有作出任何实际的操作。软件必须处理默认值，它必须运行处理空白输入的代码。编程中经常会出现没有对默认值进行处理的情况。

破旧区测试法-漏洞

- 破坏者：强迫软件做一些操作，掌握软件成功完成操作必须使用的资源，在不同程度上可以使用那些资源或限制使用那些资源
- 反叛测试法：输入最不可能的数据，或者已知的恶意输入

  逆向测试法：输入最不可能的数据

  歹徒测试法：输入一些不应该出现的数据

  错序测试法：以错误的顺序做事
- 强迫症测试法：反反复复地执行同样的操作。

混合探索式测试法

- 场景和探索
  - 将探索式测试的思维方法与传统的基于场景的测试方法结合起来，打破了脚本测试所固有的那种死板，提供了更多的灵活性
  - 基于场景探索式的想法：能够覆盖单一场景所无法覆盖的情况，并能更准确地模拟真实的用户，使测试更加的准确。

混合探索式测试法

- 基于场景的探索式测试背后的想法是:
  - 就像探险家们在穿过荒野或者穿越其他不熟悉的地域时需要使用地图，这里就使用现有设计好的场景。场景就像地图一样，是关于在测试时该做什么事情的一般性指导建议，告诉我们选择哪些输入，执行哪些代码路径，但这些又不是完全绝对的。探索式测试人员在执行一个场景时也有多条路径可以选择。我们不是要找到最短的路径，是要找到更多的路径。我们测的路径越多越好，这样会让我们有更多的理由相信，在软件被交付到那些不按常规操作的用户手里后也能可靠地运行

混合探索式测试法

- 有价值的场景：
  - 讲述用户故事：场景记录用户使用软件的动机、目的、以及具体动作
  - 描述需求：如何使用产品来执行这些功能
  - 演示产品功能 ：具体指出使用哪个菜单、单击哪个按钮、输入哪些数据、考虑每个细节
  - 演示基层场景：与其他应用程序相集成或共享信息
  - 描述设置和安装：安装说明描述了初始安装过程，安装和配置，帐号创建或者其他一些管理任务等
  - 描述警告和出错情形：描述故障排除和维护过程的文档可以用来创建很好的场景

混合式探索测试法

- 场景操作和漫游测试：多场景操作引入变化
  - 1）插入步骤：
  - 增加更多的数据 ：当场景需要增加10条数据库记录时，测试人员应该把它提高到20个或者30个，这样才能讲得出道理
  - 使用附加输入 ：当场景需要测试人员输入一系列值时，测试人员可以看看是否能够给出一些没有提到 的额外输入值
  - 访问新界面 ：当场景需要测试人员调用某些屏幕或对话框时，测试人员应该找到其他相关的一些屏幕或对话框，并把它们加入到场景中。
  - 2)删除步骤：我们可以去掉冗余和可选的步骤，这个操作的想法是使场景的步骤尽可能地减少
  - 3）替换步骤：如果场景中某些步骤可以有多个方法完成，就可以用替代步骤的场景操作来修饰这个场景

混合式探索测试法

- 4）重复步骤：重复步骤的场景操作通过重复单独的步骤或重复一组步骤来改变这个顺序
- 5）替换数据
- 6）替换环境
  - 替换硬件
  - 容器（如浏览器等）
  - 版本
  - 修改本地设置

谢谢！