# 

# 国汽大有时空科技（安庆）有限公司

## 测试方法操作指南

编 制：

审 核：

版 本：

国汽大有时空科技（安庆）有限公司对本文件资料享受著作权及其它专属权利，未经书面许可，不得将该文件资料（其全部或任何部分）披露予任何第三方，或进行修改后使用

修改记录

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 版本 | 修订说明 | 修订人 | 日期 |
| 1 | V0.1 | 初始版本 |  | 2022/07/20 |
|  |  |  |  |  |
|  |  |  |  |  |

目录

[国汽大有时空科技（安庆）有限公司 1](#_Toc14898)

[测试方法操作指南 1](#_Toc21418)

[修改记录 2](#_Toc12062)

[1、引言 5](#_Toc2077)

[1.1目的 5](#_Toc21051)

[1.2适用范围 5](#_Toc28415)

[2、测试方法操作说明 5](#_Toc30733)

[2.1功能测试 5](#_Toc3267)

[2.1.1功能测试定义 5](#_Toc5426)

[2.1.2功能测试方法 6](#_Toc26019)

[2.1.3功能测试目的 5](#_Toc16070)

[2.1.4功能测试步骤 6](#_Toc31980)

[2.2性能测试 7](#_Toc11003)

[2.2.1 性能测试定义 7](#_Toc22458)

[2.2.2 性能测试目的 8](#_Toc7018)

[2.2.3 性能测试方法 8](#_Toc26245)

[2.2.4 性能测试步骤 8](#_Toc9954)

[2.3安全性测试 10](#_Toc5168)

[2.3.1安全性测试定义 10](#_Toc18116)

[2.3.2安全性测试目的 10](#_Toc28283)

[2.3.3安全性测试方法 10](#_Toc27695)

[2.3.4安全性测试步骤 13](#_Toc15101)

[2.4兼容性测试 14](#_Toc17280)

[2.4.1兼容性测试定义 14](#_Toc9504)

[2.4.2兼容性测试目的 14](#_Toc16634)

[2.4.3兼容性测试方法 15](#_Toc21672)

[2.4.4兼容性测试步骤 15](#_Toc10135)

**1、引言**

**1.1目的**

本文档编写的目的主要对测试团队测试活动进行了规范，使我们的产品测试活动更加科学有序的进行，旨在提高产品的测试质量，制定完整且具体的测试路线和流程，为快速、高效和高质量的软件测试提供基础流程矿建，最终目标是实现软件测试规范化，标准化。

**1.2适用范围**

本文档适用于公司及外部人员，了解公司的测试方法及流程。

**2、测试方法操作说明**

**2.1功能测试**

**2.1.1功能测试定义**

功能测试，也称为行为测试，根据产品特性、操作描述和用户方案，测试一个产品的特性和可操作行为以确定它们满足设计需求。本地化软件的功能测试，用于验证应用程序或网站对目标用户能正确工作。使用适当的平台、浏览器和测试脚本，以保证目标用户的体验将足够好，就像应用程序是专门为该市场开发的一样。功能测试是为了确保程序以期望的方式运行而按功能要求对软件进行的测试，通过对一个系统的所有的特性和功能都进行测试确保符合需求和规范。

**2.1.2功能测试目的**

功能测试的目的主要如下：

1、应用程序的主要功能是否正确或是否遗漏；

2、功能实现是否满足用户需求和系统设计的隐藏需求；

3、能否正确地接受输入，能否正确地输出结果；

4、验证业务流程是否正确、合理。

**2.1.3功能测试方法**

1、按是否查看程序内部结构分为：

（1）黑盒测试（black-box testing）：只关心输入和输出的结果，不关注被测对象的内部逻辑和结构。

（2）白盒测试（white-box testing）：仅关注被测对象的内部逻辑和结构，不关注被测对象的外在功能。

（3）灰盒测试，既关注被测对象的部分内部逻辑和结构，也关注被测对象的部分外在功能。

2、按是否运行程序分为：

（1）静态测试（static testing）：是指不实际运行被测软件，而只是静态地检查程序代码、界面或文档可能存在的错误的过程。

静态测试包括：

对于代码测试，主要是测试代码是否符合相应的标准和规范。

对于界面测试，主要测试软件的实际界面与需求中的说明是否相符。

对于文档测试，主要测试用户手册和需求说明是否真正符合用户的实际需求。

（2）动态测试（dynamic testing），是指实际运行被测程序，输入相应的测试数据，检查输出结果和预期结果是否相符的过程。

**2.1.4功能测试步骤**

1、确认需求范围

（1）明确各类需求分别是谁做的，找开发总负责人拿到任务分配表。

（2）根据产品需求整理出有哪些功能点，包括业务功能、数据约束、编辑约束、权限需求等，如果是增量版本，确定本次和上次版本的不通电在哪里，需要关注什么，变更部分对其他模块的影响等。

（3）根据整理的功能点，提取每一个功能点中的细节需求，确定测试目标的优先级。

1. 测试前准备

（1）熟悉完产品需求后，需要准备测试用例，根据常见的等价类划分法、边界值分析法、判定表法等方法设计测试用例

（2）准备测试需要的账号等数据，需要的账号自己注册，切勿与其他测试人员混合使用，易造成数据不对等其他问题。

（3）确认再什么环境下测试，包括软件和硬件以及网络要求等，对于数据库有无要求，数据量有无要求，操作系统有无要求，存不存在制约软件使用的硬件等。

1. 测试任务分配

（1）根据每个成员的情况，测试任务的难易程度，以及对整个业务的影响程度，为每条测试任务标注优先级，优先级高的测试任务需要优先测试。

（2）量有无要求，操作系统有无要求，存不存在制约软件使用的硬件等。

1. 测试任务执行

（1）测试人员根据分配的测试任务在测试用例的基础上执行用例。

（2）对比实际结果与预期结果是否一致。

**2.2性能测试**

**2.2.1 性能测试定义**

性能测试是通过自动化的测试工具模拟多种正常、峰值以及异常负载条件来对系统的各项性能指标进行测试。负载测试和压力测试都属于性能测试，两者可以结合进行。通过负载测试，确定在各种工作负载下系统的性能，目标是测试当负载逐渐增加时，系统各项性能指标的变化情况。压力测试是通过确定一个系统的瓶颈或者不能接受的性能点，来获得系统能提供的最大服务级别的测试。

**2.2.2 性能测试目的**

1、验证系统是否满足预期需求;

2、验证系统在高压下的表现;

3、验证系统是否能持续稳定的运行;

4、探测系统的瓶颈和产生瓶颈的原因;

5、探测系统设计与资源之间的最佳平衡，改善并优化系统的性能。

**2.2.3 性能测试方法**

根据测试目的的不同，性能测试通常可分为负载测试、压力测试和稳定性测试等几种比较常见的方法。

* 负载测试：找到系统稳定时（或满足性能需求下）的最大吞吐量；（要有响应时间、成功率的限制，比如定义：99.9%的响应时间必须在1ms之内，平均响应时间在1ms以内，100%的请求成功）
* 稳定性（通过浸泡测试soak test）：以系统稳定时的最大吞吐量（或满足性能需求时的最大吞吐量），长时间对系统进行测试，以检查系统是否稳定。
* 压力测试： 找到系统极限值，系统瓶颈（系统崩溃临界值）（要求：响应时间可以变慢，但系统不能崩溃）。

**2.2.4 性能测试步骤**

1、性能需求分析。

性能需求分析是整个性能测试工作开展的基础。

2、性能测试计划。

确定明确的需求之后，要做的工作就是制定性能测试计划。对性能测试过程中所有需要工作制定与规划。

3、测试环境搭建。

测试环境搭建，分硬件环境与软件环境，硬件环境主要是向上级审批硬件配备，在某些大型性能测试，可能需要公司购置或租用硬件设备来进行。或者是将来原有设置进行调配与重组，这个时候就需要网络工程师的参与或协助。

4、性能工具的引入。

走到这一步才需要引入性能测试工具，在日常的工作中往往是先选定好测试工具然后再分析需求，制定计划进行测试。这样在做性能需求分析的时候往往会考虑所选的工具是否能实现，无法实现可能就放弃这个需求或改变这个需求。这样以某一工具为基础点做出的性能测试结果可能是不准确的。

5、测试的执行。

运行性能监控工具，按照测试场景执行并记录测试结果。往往需要通过多轮测试，才能找到并发用户数的具体值。

6、测试结果的分析。

根据测试工具采集的各种数据和图表分析系统的瓶颈、响应时间是否达标、超时是否在接受范围内、硬件资源消耗情况是否合理等维度去考虑。

7、软件硬件配置调整与优化。

根据分析结果指导调优方案，再次执行测试并采集数据，分析优化后的结果性能问题是否有改善。

**2.3安全性测试**

**2.3.1安全性测试定义**

安全性测试是指有关验证应用程序的安全等级和识别潜在安全性缺陷的过程。应用程序级安全测试的主要目的是查找软件自身程序设计中存在的安全隐患，并检查应用程序对非法侵入的防范能力，根据安全指标不同测试策略也不同。

**2.3.2安全性测试目的**

安全性测试的目的是查找软件自身程序设计中存在的安全隐患，并检查应用程序对非法侵入的防范能力，根据安全指标不同测试策略也不同，如果遵循相同的原则，去证明软件的安全性，将有利于软件安全测试的工作规范的进行，有利于软件安全测试工作的发展。

**2.3.3安全性测试方法**

安全测试的常用的方法有以下几种：

(1)静态代码检查

静态代码检查主要是通过代码走读的方式对源代码的安全性进行测试，常用的代码检查方法有数据流、控制流、信息流等，通过这些测试方法与安全规则库进行匹配，进而发现潜在的安全漏洞。静态代码检查方法主要是在编码阶段进行测试，尽可能早地发现安全性问题。

(2)动态渗透测试

动态渗透测试法主要是借助工具或手工来模拟黑客的输入，对应用程序进行安全性测试，进而发现系统中的安全性问题。动态渗透测试一般在系统测试阶段进行，但覆盖率较低，因为在测试过程中很难覆盖到所有的可能性，只能是尽量提供更多的测试数据来达到较高的覆盖率。

(3)扫描程序中的数据

系统的安全性强调，在程序运行过程中数据必须是安全的，不能遭到破坏，否则会导致缓冲区溢出的攻击。数据扫描主要是对内存进行测试，尽量发现诸如缓冲区溢出之类的漏洞，这也是静态代码检查和动态渗透测试很难测试到的。

从用户认证、网络、数据库和Web 四个角度进行安全性测试，需要注意以下几个方面：

(1)用户认证安全性测试

1) 系统中不同用户权限设置;

2) 系统中用户是否出现冲突;

3) 系统不应该因用户权限改变而造成混乱;

4) 系统用户密码是否加密、是否可复制;

5) 是否可以通过绝对途径登录系统;

6) 用户退出后是否删除其登录时的相关信息;

7) 是否可以使用退出键而不通过输入口令进入系统。

(2)网络安全性测试

1) 防护措施是否正确装配完成，系统补丁是否正确;

2) 非授权攻击，检查防护策略的正确性;

3) 采用网络漏洞工具检查系统相关漏洞(常用的两款工具为NBSI 和IPhackerIP);

4) 采集木马工具，检查木马情况;

5) 采用各种防外挂工具检查程序外挂漏洞。

(3)数据库安全性测试

1) 数据库是否具备备份和恢复的功能;

2) 是否对数据进行加密;

3) 是否有安全日志文件;

4) 无关IP 禁止访问;

5) 用户密码使用强口令;

6) 不同用户赋予不同权限;

7) 是否使用视图和存储过程;

(4)Web 安全性测试

1) 部署与基础结构;

2) 输入验证;

3) 身份验证;

4) 授权;

5) 配置管理;

6) 敏感数据;

7) 会话管理;

8) 加密;

9) 参数操作;

10) 异常管理;

11) 审核和日志记录;

**2.3.4安全性测试步骤**

1. 识别系统中有价值的数据

了解了业务以后，我们需要考虑系统中会有什么有价值的数据。这是为下一步加入恶意用户需求做准备。对于一个网上商城，有价值的数据可以包括产品信息、订单信息、用户信息、支付，等等。

2. 在需求阶段加入恶意用户需求

恶意用户需求是用来记录恶意用户想要在系统中达到的目的。与普通用户需求的区别是，我们不是要去实现它，而是使用它帮来助我们远离对系统使用者“不恰当的信任”。通常我们需要针对每一个合法用户需求来增加一个或多个相对应的恶意用户需求。

3. 针对“恶意用户需求”设计测试用例

现在我们需要做的是努力把自己限制在“恶意用户”的角度做头脑风暴：“到底有什么方法可以使买家无法上传图片信息呢？让页面无法正确显示买家秀图片又怎么做到？”嗯，也许最直接的办法就是让服务器所在的机房断电、断网之类的。这是些不错的想法，虽然执行难度有点大。没关系，记录下来。

4. 参与启动恶意需求的开发（evil story kickoff）

在开发人员开始开发合法用户需求之前，我们需要跟业务分析人员、开发人员一起沟通需求的内容。在敏捷软件开发项目中我们叫它story kickoff，即用户故事启动。

5. 在开发环境验收恶意需求的实现

100%预防软件的缺陷与漏洞是不太可能的，所以这个环节的存在是为了提早反馈。

6. 在测试环境中进行安全测试

到了运行测试的阶段。可能这个时候我们之前想到的测试用例已经被开发人员给解决。如果是这样那就太好了。但是，事实并非有这么美好。第一，可能这些用例只是在开发环境上成功通过了，但是在理想的测试环境里，也就是类产品环境里，这些用例可能并不能完全通过；第二，肯定还有其他需要探索的地方。这时我们就可以用OWASP Zap、Burp这样的工具来辅助我们把之前的安全测试用例执行一次，同时还再可以对系统的安全性做一下探索测试。

7. 向团队反馈所发现的安全漏洞

测试完成后，我们就可以向团队以及相关干系人汇报安全测试的结果了。跟非安全测试不同的地方是，当我们反馈安全漏洞的时候，要考虑是否不同漏洞结合起来会增加系统的安全风险。

**2.4兼容性测试**

**2.4.1兼容性测试定义**

软件兼容性测试是指检查软件之间能否正确地进行交互和共享信息。兼容测试，指对所设计程序与硬件、软件之间的兼容性的测试。一般来说，兼容性指能同时容纳多个方面，在计算机术语上兼容是指几个硬件之间、几个软件之间或是软硬件之间的相互配合程度。

**2.4.2兼容性测试目的**

兼容性测，主要目的是为了兼容第三方软件，确保第三方软件能正常运行，也能够不受影响。

具体表现如下：

1、待测项目在不同的操作系统平台上正常运行，包括待测试项目能在同一操作系统平台的不同版本上正常运行；

2、待测项目能与相关的其他软件或系统“协调工作”；

3、待测项目能在指定的硬件环境中正常运行；

4、待测项目能在不同的网络环境中正常运行。

**2.4.3兼容性测试方法**

Web端和APP端的兼容性测试，有两种方法：

（1）人工测试即全手工测试兼容；

（2）借助第三方兼容性测试工具。

兼容测试包括：

(1)浏览器兼容测试：测试程序在不同浏览器上是否可以正常运行，功能能否正常使用；

(2)屏幕尺寸和分辨率兼容测试：测试程序在不同分辨率下能否正常显示；

(3)操作系统兼容测试：测试程序在不同的操作系统下面能否正常运行，功能能否正常使用，显示是否正确等；

(4)不同设备型号兼容测试：针对于APP，现在移动设备型号五花八门，主要测试APP在主流设备上能否正常运行，会不会出现崩溃的现象。

**2.4.4兼容性测试步骤**

（1）Web兼容性测试

* 首先开展人工测试，测试工程师测试主流浏览器和常用操作系统测试主流程和主界面，看看主流程和主界面是否有问题，如果存在问题，那么记录下bug情况，以及浏览器型号和版本，以及操作系统，准确定位bug产生的原因，提交bug，告知开发人员修改。所有的主流设备都需要进行测试，只关注主流程和主界面，毕竟每个系统主流程和主界面不是很多，所以这个工作量还是可以承受的。
* 其次借助第三方测试工具，借助第三方测试工具，找到bug产生的位置，分析测试结果，告知程序员调整。

（2）APP兼容性测试

* APP的兼容性测试和Web测试类似，首先开展人工测试，测试工程师借助测试设备对主流程和主功能，主界面进行测试；收集所有的能收集到的不同型号的测试设备测试主流程和主界面，看看主流程和主界面是否有问题，如果存在问题，综合考虑设备的使用率等因素，看看是否需要调整，如果需要，那么记录下bug情况以及测试设备的型号和操作系统，准确定位bug产生的原因，提交bug，告知开发人员修改。
* 其次借助第三方测试工具，对于APP的兼容性测试，测试工程师把打包好的apk或者IPA文件，上传到测试平台，选择需要测试的设备型号，开始任务即可；等待一段时间。测试完成后会生成一份测试报告，可以查看错误页面和错误日志，如果需要调整，那么提交bug，告知程序员修改即可。