

33 | 测试工具：为什么不应该通过QQ/微信/邮件报Bug？

2019-05-16 宝玉 来自北京

《软件工程之美》



你好，我是宝玉。十多年前，当我还是个野路子程序员时，我在外面接私活做项目，客户在使用过程中遇到了 Bug，直接就截个图，或者用 Word 文档整理在一起，从 QQ 或者邮件上把 Bug 信息发送给我，我收到后再修复更新上线。

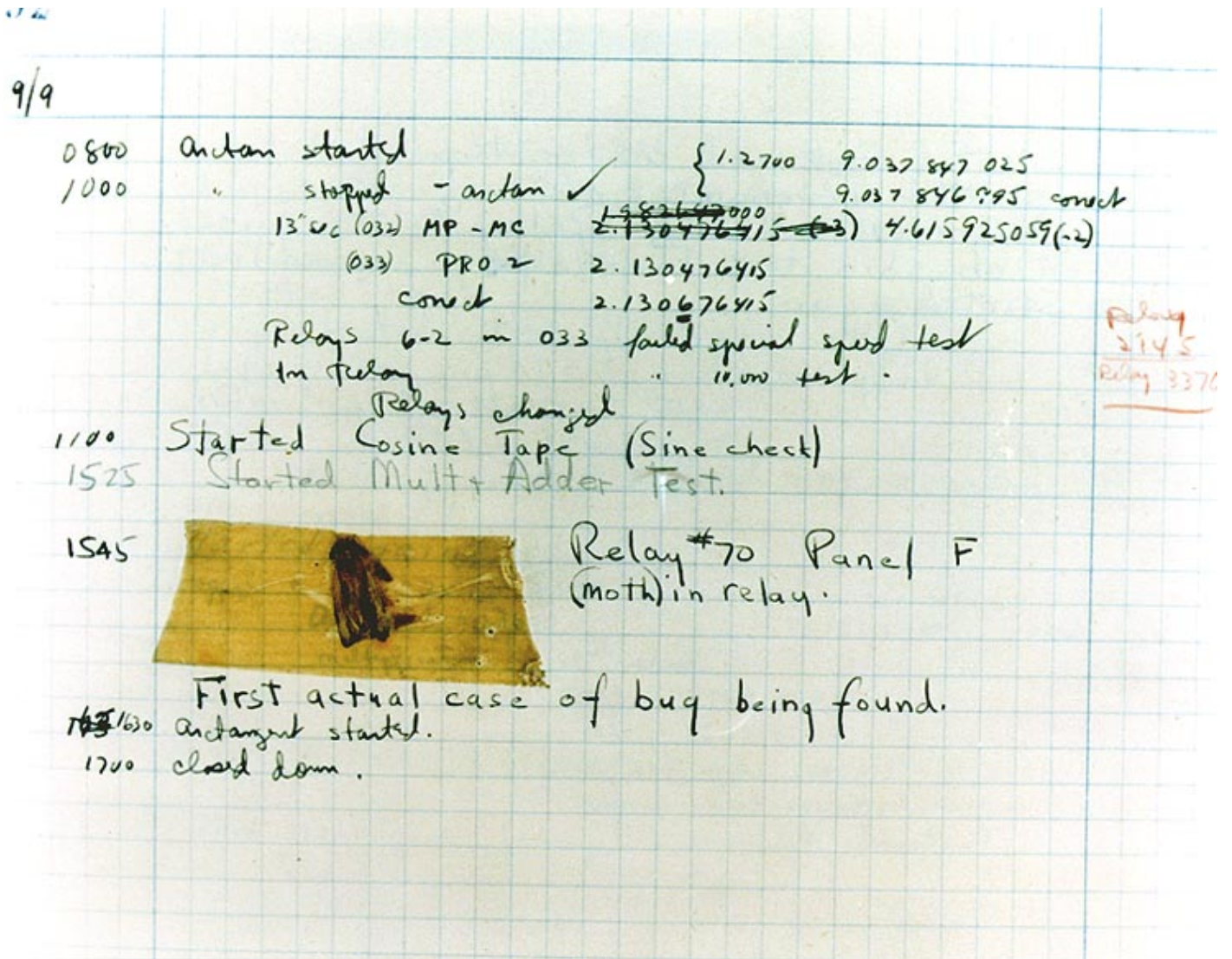
而现在正规的软件项目已经不会再用这种原始的方式来报 Bug 了，而是会借助测试工具来帮助报告和跟踪 Bug，即使你偶尔能看到有项目还在采用原始方式报 Bug，你肯定也会觉得这样做不专业。

但不知道你有没有仔细想过这个问题，为什么现在不通过 QQ/ 微信 / 邮件报 Bug，又有哪些测试工具可以帮助你更好地发现、报告和跟踪软件中的 Bug 呢？今天我们来展开讨论这个问题。

Bug 跟踪工具

我想你对于 Bug 这个词一定不陌生，它是我们软件中的缺陷或错误。这个词的诞生也很有意思。

1947 年 9 月 9 日，一只小飞蛾钻进了哈佛大学的一台计算机电路里，导致系统无法工作，操作员把飞蛾贴在计算机日志上，写下了“首个发现 Bug 的实际案例”。



图片来源：Wikipedia

虽然 Bug 的历史已经有 60 多年了，然而 Bug 跟踪工具却没有出现太久。软件项目中最早也是通过邮件、即时通讯等原始方式报告 Bug，直到 1992 年才有第一个专业的 Bug 跟踪软件 [GNATS](#)。

在这之后才逐步有了像 Bugzilla、Jira、MantisBT 等专业的 Bug 跟踪工具。而现在，Bug 跟踪工具已经成为软件项目中必不可少的工具之一。

那么，Bug 跟踪工具是怎么逐步替代 QQ、邮件等方式来处理 Bug 的呢？

为什么要使用 Bug 跟踪工具？

我们在上一节学习了软件测试相关的理论知识，软件测试的主要工作就是发现 Bug、报告 Bug 和跟踪 Bug。测试人员发现 Bug 只是第一步，还需要报告 Bug 让开发人员可以知晓和定位，并且跟踪整个 Bug 修复的过程。

用 QQ 或者邮件报 Bug 的这种方式，看起来快捷简单，但是问题很多：

- Bug 不能有效被跟踪，不知道一个 Bug 是不是已经被修复了；

- 效率很低，开发人员频繁的被这样的报 Bug 的消息打断，不得不停下手头的工作去甄别 Bug；

- 不能直观的了解当前项目的 Bug 状态，比如说：修复了多少，还有多少没有修复，近期 Bug 数量是增加了还是减少了。

不难看出，通过 QQ 等方式报告的 Bug，都是文字配合图片等信息，很难检索和分类，而 **Bug 跟踪工具**，采用结构化的数据来定义 Bug，每一个 Bug 都有一些关键的信息可以对 **Bug 进行分类和检索**。

在 Bug 跟踪工具使用中，一个基本的 Bug 信息包括：

- 标题；

- 描述（包括期望结果、实际结果和重现步骤等关键信息）；

- 优先级；

- 指派人；

- 状态（New、Open、Rejected、Fixed 等）；

- 其他。

那这样的话，就很容易的对 Bug 进行分类和检索，比如说：

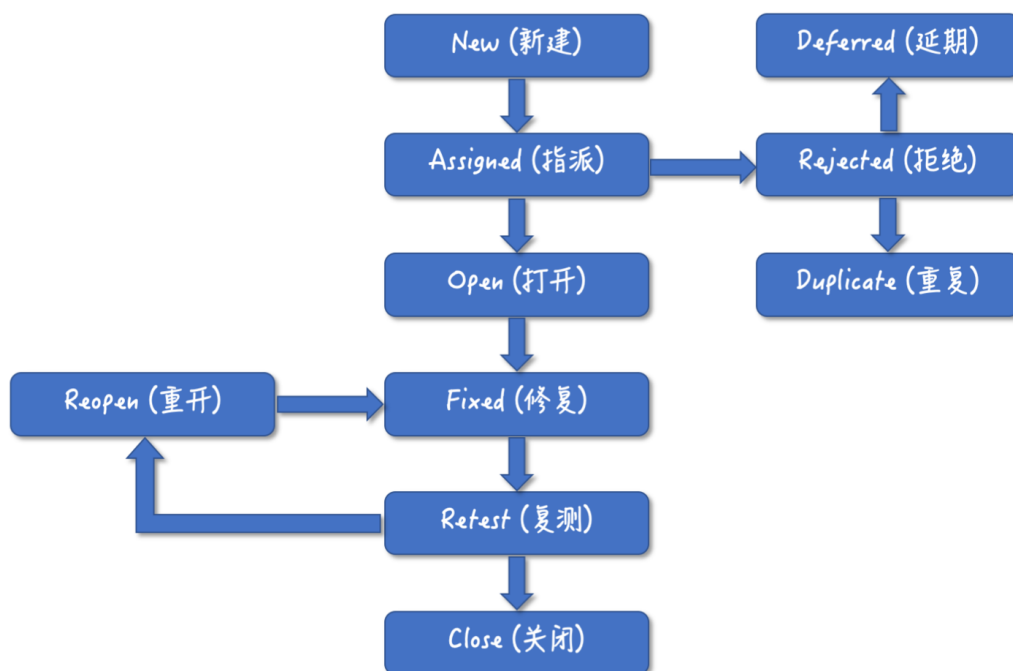
张三想查看所有分配给他的 Bug，那只要列出所有指派人是张三的 Bug；

想列出所有未解决的 Bug，只要列出所有状态不是 Close 或 Rejected 的 Bug 即可。

这样对于开发人员来说，可以直观的看到自己有哪些 Bug 需要处理，Bug 的描述信息也可以帮助重现 Bug、快速定位到 Bug 的原因；对于项目经理或者测试人员来说，可以直观的看到哪些 Bug 还没解决，及时了解项目进展。

另外，我在《[🔗12 | 流程和规范：红绿灯不是约束，而是用来提高效率](#)》这篇文章中提到了项目中的流程和规范，**在软件项目中，要把好的实践流程化，把好的流程工具化。Bug 跟踪工具则很好的贯彻了这一点，将 Bug 的解决过程流程化。**

你平时在 Bug 跟踪系统中看到的 Bug 状态，看起来只是一个有限的状态列表，但背后其实是一套解决 Bug 的流程。就像下面这张图表示的这样，一个 Bug 从创建到最后结束，其实是有一个完整的流程的。



通过这样的流程，开发人员就可以集中对 Bug 进行分配、按照优先级分别解决，而测试人员则可以第一时间知道 Bug 处理的状态变化，及时验证，方便跟踪整个过程。

使用 Bug 跟踪工具的注意事项

报告 Bug 的目的是为了能跟踪 Bug，以及帮助开发人员重现直到解决问题。要想做到测试和开发高效协作，这里面有一些需要注意的事项。

首先，所有的 Bug 都应该通过 Bug 跟踪系统管理和跟踪，不应该再通过 QQ/ 微信 / 邮件的方式跟踪 Bug。如果客户、同事通过 Bug 跟踪系统之外的其他途径反馈 Bug，应该统一提交到 Bug 跟踪系统管理跟踪起来。

然后，不能把多条 Bug 合并成一条，一个 Bug 创建一个独立的 Ticket。我遇到过有些测试为了省事，把几条 Bug 合并成一个 Ticket 来报，导致的问题就是，必须这几条 Bug 都修复了，这个 Ticket 才能改变状态，如果其中一个 Bug 没有验证通过，需要 Reopen 整个 Ticket。

再有，描述清楚如何重现 Bug 非常重要。一个 Bug 如果无法重现，也没有日志、截图等辅助信息，那是非常难以定位的，会浪费很多开发人员定位 Bug 的时间。

最后，不要把 Bug 跟踪系统当成讨论板用。在项目中一个常见的场景是，一个 Ticket 下面，跟讨论版一样添加了很多留言，开发认为不是 Bug，测试认为是一个 Bug，开发又觉得是产品设计没定义清楚，应该让产品经理来讲清楚，皮球踢来踢去，最后问题还没解决。

Bug 跟踪系统的主要功能是用来跟踪 Bug 的，不是用来讨论和扯皮的。遇到上面的情况，其中一方就应该主动一点，拉上相关人面对面讨论，当面确认清楚这个 Bug 到底是什么问题，然后马上解决掉。

自动化测试工具

除了 Bug 跟踪工具，软件测试中还有很重要的一个工具就是自动化测试工具，虽然我在《[29 | 自动化测试：如何把 Bug 杀死在摇篮里？](#)》中已经有了较多篇幅说明，但这里还是想继续提一下，因为我觉得，**未来自动化测试会占据越来越多的比例，很多手工测试的工作会逐步被自动化测试代替。**

像美国 Facebook、Google、Amazon 这些大厂，单纯的手工测试职位在减少，一些手工执行测试用例检查的工作外包到了人力成本更低的像中国、印度、罗马尼亚等国家，而美国本土主要招聘的都是能写自动化测试的软件测试人员，或者直接就是开发人员来写这些自动化测试代码。

这就意味着对于软件测试人员来说，要求越来越高了，不仅要会设计测试用例，还要能写自动化测试脚本。同时对于开发人员来说，不仅要写功能代码，还需要实现一定量的自动化测试代码。

这些年自动化测试工具的快速发展，也降低了自动化测试的实现难度，可以方便地搭建自动化测试环境，通过简单的脚本语言就可以模拟人工操作。

但很多团队还是不愿意投入在自动化测试的开发上面，宁可雇佣更多的初级测试人员手工测试。

其实这个问题还是要整体来看，这就像修路，如果你从一个地方到另一个地方（类比测试所有用例），偶尔走几次，那么可以不修路（手动测试），如果你未来一段时间需要频繁的在两个地方通行（反复测试），那么最好现在就开始修建高速公路（自动化测试），这样可以节约你大量通行的时间（测试时间）。

当然更多的情况其实是团队不知道该如何实施自动化测试，比如说测试人员不会写程序，开发人员太忙，或者开发人员不会写测试用例，或者不知道该选择什么样的自动化测试工具。

对于这种情况，我的建议是：

测试人员可以学习一些基本的编程知识，尝试自己实现自动化测试。自动化测试所需要的技术，主要是对 API 的调用，并不需要复杂的逻辑，其实学习门槛并不高，而且这种技术在工作效率、薪资、个人职业发展等方面的投资回报都是巨大的。

从项目的角度，应该加大对自动化测试的投入，让开发人员参与到自动化测试代码的开发中。增加自动化测试代码的覆盖，对于提升软件质量是有明显好处的，通过自动化测试可以提升测试效率，及时发现软件质量问题。

对于开发人员来说，如果已经有了测试用例，完成自动化测试并不复杂，这个投入其实比做一些重要性不高的功能回报更高。

自动化测试工具的选择，需要根据你的软件的特点，去找出来适合你软件自动化测试的几款，然后自己搭建环境试用一下。在本文后面的附录中，我会列出一些自动化测试工具供参考。

其他帮助发现 Bug 的测试工具

软件测试的一个主要工作就是发现 Bug，而要发现 Bug，就需要对软件的各个领域进行测试，比如说有性能、安全性、兼容性等领域。

这些不同领域的测试，要求也不一样，比如说性能测试要求能测试出软件是否有性能瓶颈，能达到多少用户的访问量，需要模拟大量用户并发访问；安全性测试则要求对软件可能存在的安全漏洞进行扫描、验证；兼容性测试则要针对不同环境不同设备，对软件进行测试，以确保不会因为环境不一致导致功能不正常。

这些测试要么人工很难完成，例如模拟大量用户并发访问；要么需要很深的专业知识，例如安全性测试；要么需要大量的设备和巨大的工作量，比如做兼容性测试。所以这些领域的测试，就需要借助工具的帮助才能进行测试，从而发现问题。

应用这些测试工具其实并不难，毕竟都有很成熟的 API，网上也有很多教程，真正需要的是去执行。另外如果想要最大化工具的价值，及时发现问题，还要考虑将测试工具的应用自动化，加入到你的持续集成流程中去。

以压力测试为例，你用 Jmeter 完成了压力测试脚本后，还可以考虑和 CI 集成，在每次构建时，运行一遍压力测试代码，可以在构建完成后看到直观的图表，还可以设置性能数据的阈值，如果性能指标低于阈值，会导致构建失败，这样就可以第一时间发现性能问题，缩小问题范围，并及时解决。

在这里，我也帮助搜集了一些相关的测试工具供参考，具体可以查看附录。

附录

Bug 跟踪工具

在项目管理工具那一篇文章中，我已经给你介绍了一些任务跟踪系统，比如说 [🔗 Jira](#)、[🔗 禅道](#)、[🔗 TAPD](#)、[🔗 云效](#)等，都可以用来跟踪 Bug。

Bugzilla

[🔗 Bugzilla](#) 是由 Mozilla 公司提供的一款开源免费的 bug 跟踪系统。这是一款历史很悠久的产品。

MantisBT

[🔗 MantisBT](#) 是一个简单但功能强大的开源 bug 跟踪系统，可以通过各种插件来扩展其功能。

Redmine

[🔗 Redmine](#) 是一款开源的综合性的项目管理工具，不仅可以用于 Bug 跟踪，还可以用来跟踪项目进度。

自动化测试工具

除了传统的桌面应用外，现在移动设备的普及，要测试的终端也越来越多。借助一些自动化测试工具，可以帮助简化多设备的测试。下面简单介绍几个自动化测试工具。

Selenium

[🔗 Selenium](#) 是一个 Web 端的自动化测试工具，直接运行在浏览器中，用来模拟用户操作。类似的还有 [🔗 WebDriverIO](#) 和 [🔗 Nightwatch.js](#)，支持 Javascript，API 更简单更方便。

Appium

🔗 **Appium** 是一个开源、跨平台的自动化测试工具，用于测试移动原生应用，支持 iOS, Android 系统。

Macaca

🔗 **Macaca** 是阿里巴巴开源的一款面向多端的自动化测试工具，支持桌面端、Web、移动端、真实设备和模拟器。

更多自动化测试工具可以参考：《🔗 **Best Automation Testing Tools for 2019 (Top 10 reviews)**》(🔗 **中文版**)。

压力测试工具

很多软件在上线后，需要面对巨大的用户访问量，但如果等到上线后才发现程序性能不行，访问量一大就会导致服务崩溃，那就太晚了。所以最好是在测试阶段，就能测试出来程序的性能如何，瓶颈在哪里，然后在发布前对程序进行优化，确保能满足性能要求。

对程序性能的测试，就需要借助压力测试工具来模拟大量用户并发访问的场景。下面简单介绍一下几款常用的性能测试工具。

Apache JMeter

🔗 **JMeter** 是一款开源的压力测试工具，纯 Java 应用程序。

LoadRunner

🔗 **LoadRunner** 是惠普旗下的一款商业自动负载测试工具，可以通过录制的方式制作测试脚本，上手容易功能强大，可以方便的监控和分析性能测试结果。

阿里云性能测试 PTS

阿里云性能测试 [PTS](#) 是基于云端的压力测试服务，可以模拟从全国各地域运营商网络发起的流量，真实地反映使用情况，生成有价值的性能测试报告。

WebPageTest

[WebPageTest](#) 是一个可以用来测试和分析网页性能的在线工具，支持不同浏览器，支持 API。可参考《[WebPagetest H5 性能测试工具入门详解](#)》。

更多性能测试工具介绍可以参考：《[10 大主流压力 / 负载 / 性能测试工具推荐](#)》。

安全性测试工具

软件的安全性是非常重要的指标，有时候开发人员缺乏安全意识，就可能会导致程序存在安全漏洞。安全领域也是开发和测试之外的一个技术领域，中小公司一般不会有自己专业的安全团队，就需要借助一些安全性测试工具来帮助对软件进行安全性检测。

HP Fortify On Demand

[Fortify On Demand](#) 是惠普旗下的一款安全检测工具，可以通过分析源代码、二进制程序或者应用程序 URL 检测程序安全漏洞。

Sqlmap

[Sqlmap](#) 是一款开源免费的检测 SQL 注入的工具。

IBM Application Security APPScan

[APPScan](#) 是 IBM 旗下的一款漏洞扫描工具，支持网站和移动 App。

更多安全性测试工具介绍可以参考：《[11 款常用的安全测试工具](#)》《[安全测试工具篇（开源 & 商业）](#)》《[最受欢迎的软件安全性测试工具有哪些？](#)》。

浏览器兼容性测试工具

网站开发最苦恼的问题之一就是浏览器兼容问题，不仅要兼容 Chrome、IE/Edge、Firefox 三大主流浏览器，还得考虑桌面设备和移动设备上的不同表现。如果人工对所有浏览器做兼容性测试，工作量比较大。好在也有一些不错的工具可以帮助做兼容性测试。

Browsera

🔗 **Browsera** 可以对不同浏览器下的布局提供报告，包括截图和 Javascript 错误。

Browslring

🔗 **Browslring** 可以针对不同浏览器进行测试，它在虚拟机中运行真实桌面浏览器，还可以人工进行交互。

更多浏览器兼容性测试工具可参考《🔗 **10 个免费的顶级跨浏览器测试工具**》

测试用例管理工具

我们在上一篇里面已经学习了，设计测试用例是软件测试很重要的工作，有专业的工具帮助管理测试用例，也可以起到事半功倍的效果。

TestRail

🔗 **TestRail** 是 TestRail 是一个专注于管理测试用例的工具，可以用它来创建测试用例和用例集，跟踪测试用例的执行和生成报告。

飞蛾

🔗 **飞蛾** 是 Coding 旗下的测试管理工具，对中文支持好，界面美观。

更多测试用例管理工具可以参考：《🔗 **有哪些比较好的测试用例管理工具？**》

总结

今天，我带你一起学习了软件测试工具的相关知识。软件测试，主要工作就是发现 Bug、报告 Bug 和跟踪 Bug。软件测试工具，也是围绕这三方面来帮助我们提高效率的。

Bug 跟踪工具，不仅可以方便的报告 Bug 和跟踪 Bug，更可以帮助开发人员将 Bug 的解决过程流程化。

自动化测试工具是发展趋势，未来自动化测试会占据越来越多的比例，很多手工测试的工作会逐步被自动化测试代替。

除了 Bug 跟踪工具和自动化测试工具，软件测试中还有性能测试工具、安全性测试工具、兼容性测试工具等，这些工具都可以更好的帮我们发现软件中的质量问题。

如果想要最大化工具的价值，及时发现问题，还要考虑将测试工具的应用自动化，加入到你的持续集成流程中去。

课后思考

你的项目中使用了哪些软件测试工具？看完本文，你觉得还可以在哪些方面加强对软件测试工具的应用？欢迎在留言区与我分享讨论。

感谢阅读，如果你觉得这篇文章对你有一些启发，也欢迎把它分享给你的朋友。

© 版权归极客邦科技所有，未经许可不得传播售卖。页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

精选留言 (13)



果然如此

2019-05-17

对于自动化测试一直有些疑问，如果每次发布都对所有方法自动化测试：

1. 一定会耗费很多时间；
2. 数据库产生很多测试历史数据；
3. 写测试用例能达到覆盖率高的写代码技巧，如边界测试代码、幂等测试代码如何实现。

作者回复: 1.自动化测试确实会耗费很多时间

自动化测试代码通常是金字塔结构:

单元测试(小型测试) 代码最多, 执行也最快, 占总比例的70%左右, 通常1分钟内

集成测试(中型测试) 代码其次, 执行比较快, 占比20%左右, 控制在10分钟以内

端对端测试(大型测试) 最少, 执行慢, 占比10%左右

一般CI里面跑单元测试和集成测试, 耗时10-15分钟左右, 其实还可以接收。

2. 跑自动化测试, 数据库有不同策略

单元测试不访问数据库, 完全模拟

集成测试只访问本机数据库, 或者模拟的内存数据库, 每次创建新数据库, 或者使用完清空数据库

端对端测试, 每次创建唯一数据(例如增加固定数据+时间戳), 连接真实的测试环境, 可以不清理数据

3. 高覆盖率的关键在于在写代码就注意让代码方便的被测试。也不必过于追求100%覆盖, 70%以上我觉得就不错了。



👍 9



易林林

2019-05-16

在集中办公的时候, 很多人喜欢用微信/QQ去聊工作上的事情, 一会儿@这个, 一会儿@那个, 一些重要的信息很快就被淹没了, 很难去挑出对自己有用的信息。对于这种情况, 我一般很少去理会别人发了什么信息, 也不会随时在电脑上开着微信/QQ等着闪亮的出现, 要么通过各种管理工具沟通(邮件、Bugfree、Jira等等), 要么当面或电话沟通, 这样问题才能得到有效的、高效的解决。但在这样做之前, 我会在合适的场合说出这样做的原因和相应的解决办法, 避免让他人误解。

作者回复: 确实, 这样的频繁打扰特别影响效率。

不过我在用了Slack之后, 还是改变了看法, 觉得从沟通的角度看, 尤其是异地办公。

沟通很及时, 跟当面沟通一样

通过Channel可以对消息进行区分和过滤

通过Thread可以展开讨论以及避免对不想干人员的打扰



宝宝太喜欢极客时间了

2019-05-16

老师，对测试这块一直很疑惑，测试脚本、测试用例、测试数据这三者如何配合一起通过CI进行自动化测试？

作者回复：是这样的，CI本质上只是一个像流水线传送带，你的代码提交了，流水线传送带开始工作，你可以在传送带上面添加任务。

简单来说，你可以想象成CI的一个任务启动后，给你一个干净的虚机（实际是运行Docker Container），然后帮你把当前代码下载下来，帮助你配置好运行环境，然后你就可以在里面安装任何软件、服务和运行任何脚本。

举例来说，你可以在传送带上增加以下任务：

1. 安装所有的依赖包
2. 运行模拟服务（比如一个内存数据库）
3. 运行单元测试
4. 运行集成测试

如果上面所有任务都成功了，那么这一次的CI任务就成功了，其中一个失败这一次的CI任务就失败了，然后你就要检查什么原因导致的，然后修复再重新执行，保障CI任务成功执行位置。



4



一路向北

2019-05-18

用qq和微信等作为bug管理，与其说是用它来做管理，不如说是用它做驱动。很多问题是在客户那边发现，因此他们第一反馈就是即时通讯工具反馈，我们也就根据这个来解决bug问题。对于小公司（10来个人）来说，管理一套系统和维护一个流程，总觉得需要付出比较大的代价，是不是这样呢？以前试过禅道，用了一段时间之后也放弃了。需要专门的人员来维护推动。

作者回复：QQ作为Bug反馈是没有问题的，但是无法有效对Bug跟踪，另外效率太低。及时是通过QQ报的Bug，也应该重新录入到Bug跟踪系统，进行管理和跟踪。

自己维护一套Bug和任务跟踪系统成本是有点高，建议直接使用在线托管的，Gitlab、GitHub、腾讯、阿里云、Coding、微软都有现成的，小团队的话价钱也不高。



谢禾急文

2019-06-16

看到这篇文章，我想到了我之前的一个想法，就是通过用一个工具记录我自己开发过程中遇到的所有bug，通过记录、分析、反思这些bug，能够有助于提升我的编程能力，有助于避免犯同样的错误。我觉得你上面说的那些工具，能够满足我的需求。如果有一个网站，能够提供bug记录、分享、解答的工能，是不是能够满足某些用户的需求(好像stackoverflow就是这样的工具)。

作者回复: 我觉得是有帮助，但这个问题的关键在于分析反思Bug。

自己对自己Bug的反思才是价值最大的，其他人看过之后不一定能有那么大的共鸣，因为一个Bug都有复杂的业务背景，是很难被记录，缺少上下文也很难理解。

StackOverflow是很有价值的，因为它是从问题切入，而问题是有很多共性的，很容易引起共鸣。



宝宝太喜欢极客时间了

2019-05-17

老师，测试用例英文test case为什么叫测试用例而不叫测试例呢？有没有test use case一说？

作者回复: 抱歉这个问题我真不知道，test case的翻译也算是约定俗成的吧。

test use case我也没见过🙄

共 2 条评论 >



邢爱明

2019-05-16

自动化测试，个人理解最大的困难是开发人员的心理和认知，理解测试工作不只是测试人员的职责，开发人员应对软件的交付质量负起更大的职责。

作者回复: 像开发写自动化测试这样的事，一方面要提高开发人员的认知，另一方面还必须要有一些规章制度强制要求执行。



calvins

2020-04-07

受教了，很多测试都明白，就是没有执行力，这篇分享了很多测试相关工具，很nice，可以更多的了解，丰富知识广度。

作者回复: 对的，学习就是这样，知道并不难，但是真用起来就是另一回事了，建议把这门课的知识在日常项目中一点点用起来，体会一下其优点和局限性，什么场景下该应用哪些知识，这样才真的算掌握了。



成

2019-05-16

服务端通过编写单元测试没啥问题，app方面好像少不了人工测试，现在测试人员主要覆盖app测试，并且每个迭代测试时间比较长，上线后问题还不少。app测试如何更好引入自动化测试？

作者回复: 可以试试App的自动化测试框架，比如说appium、katalon。



yu

2019-05-16

程式碼就像程序員的名片，要對寫出來的代碼負責，最好的負責方式就是寫測試代碼，讓每次代碼變動，都不會影響到其他代碼的運行，避免所謂的改 A 壞 B，節省迂迴的時間浪費。也為 CI/CD 做好準備，無論目前有沒有。

程序員不想寫測試在我們公司的原因大多是，不知道怎麼開始寫，不知道重點應該測試什麼。先寫測試的開發模式讓他們覺得不習慣，但這些都是過程，培養良好的撰寫測試代碼習慣後，開發品質更有保證，提升開發效率，提升個人能力，我想都是有幫助的。

程序難免有 Bug，透過追蹤軟件，良好的管控 Bug 數量與修復進度，並且補足測試。

作者回复: 🍌謝謝你的精彩分享





ifelse

2022-07-04

如果想要最大化工具的价值，及时发现问题，还要考虑将测试工具的应用自动化，加入到你的持续集成流程中去。--记下来



小老鼠

2019-10-02

1、软件测试不仅仅是发现Bug，另外更重要的是预防Bug，还有为领导者作决策。2、DevOps中测试还可以左移(预防)和右移(在生产环境中测试)。3、测试工具不是银弹，作为一名优秀的测试工程师，主力应该在分析与设计测试用例。

作者回复: 👍赞总结。

帮你补充一条：

4. 自动化测试很重要

共 2 条评论 >



ຈິນ

2019-06-27

目前java 主流的测试工具有哪些，比较稳定的，测试web及测试接口

作者回复: 一般都是自己写框架，基于TestNG + Selenium，应该也有很多其他的选择，但是抱歉我对Java了解不多，建议可以问问其他人或者通过搜索引擎搜索一下。

共 2 条评论 >

