

软件自动化测试的探索实践

贺晋宁 杜伟伟 高 静

摘 要: 随着产品应用市场的不断推广, 用户和需求越来越多, 尽管开发人员在设计和自测阶段都尽可能的将影响面考虑周全, 但并非所有的涉及面都能被考虑到, 单纯依靠以劳动密集人工测试, 已不能满足当今软件交付又快又好的要求, 而借助自动化测试, 即在没有人干预的情况下将产品的大部分功能进行回归测试, 可以有效的提高软件版本质量和开发效率, 减少故障泄露。本文通过实践探索, 论证了实施合理有效的自动化测试, 能够减少测试工作量, 缩短测试周期, 降低软件成本, 达到“多”、“快”、“好”、“省”的目的。

关键词: 自动化; 测试

1 自动化测试的发展和趋势

软件测试存在于软件生命开发周期的每一个阶段, 是软件开发质量保障的有效手段。随着计算机技术、通信技术、检测技术的迅猛发展, 智能化的用户界面、高效的数据处理、网络化的信息传递, 逐渐被应用到各种测试系统中, 形成了各种高效便捷的自动测试系统。因此测试自动化已经成为软件测试发展的重要趋势之一: 有效的结合多种测试工具和测试方法, 有效的提高软件测试效率。

软件自动化测试改变了测试的主体, 从人为驱动转化为事件驱动、时间驱动, 不仅解放了人力, 而且降低人为因素的影响。软件自动化测试按照对象不同分为: 测试过程自动化和测试结果分析自动化。测试过程自动化是包括测试环境准备的自动化, 测试用例执行的自动化; 而测试结果分

析自动化覆盖范围更广, 它是指测试中一经出现问题, 不需要人工分析中间结果或者数据流, 系统自动汇总错误信息, 同时按照既定程序抓取关键信息, 便于后续开发测试人员定位问题。

软件自动化测试在某种程度上是能够达到降本提效的作用的, 但是要想从根本上解决质量与效率、投入与产出等矛盾, 必须要有一套体系化的自动化测试框架, 包括自动化测试工具、测试策略、测试方法等, 在不同的发展阶段, 自动化测试的关注点也应该发生变化。因此, 深入研究软件自动化测试技术和方法, 保障软件质量, 已经成为国内外软件行业和相关机构的研究热点。

2 自动化测试的优点

软件自动化测试的优势是显而易见的: 高效、便捷, 被越来越多的项目认可和实践, 总结其优点如下:

1) 提高测试效率

①用于回归测试, 保证范围和效果

回归测试是将系统已经实现的功能再次验收测试, 检查新代码、新修改对系统原有功能的影响, 通常情况下, 回归测试涉及功能较多, 覆盖广泛, 因此回归测试的自动化, 可以极大的提高测试效率, 缩短回归测试时间。

②用例复用

自动化测试通常采用脚本技术, 这样有些用例就可以在现有用例的基础上, 只需做少量的甚至不做修改, 实现在不同的测试过程中使用相同的用例。

2) 实现高难度测试

对于一些非功能性测试, 比如性能测试、压力测试等, 这些测试靠人工实现很困难, 但是自动化测试能够将这些测试变得简单可靠, 比如: 大

呼测试,可以通过自动化测试工具,模拟并发用户数,控制呼叫时长,轻松实现各种效果的测试。

3) 解放人力资源

将测试人员从繁琐重复的测试工作中解放出来,不仅可以提高测试有效性,而且有助于调动测试人员的积极性,测试技术人员可以将更多的精力投入到需求澄清沟通新技术研究等方面,从而提高项目人员整体的研发能力。

3 自动化测试案例

3.1 项目现状

目前项目的现状是:需求月月有,开发常夜班;代码天天改,故障还最多。随着产品的不断推广,需求越来越多,尽管开发人员在设计和自测阶段都尽量将影响面考虑周全,但是人非神,并不是所有的功能,所有的涉及面都能够被考虑到的,而且开发人员经常需要处理一些紧急需求或者紧急故障,开发自测时间比较短,经常出现新合入的代码又引起了新的故障泄露。但是质量是产品的根本,只有高质量的产品才有市场竞争力,而人力资源是有限的,需要控制成本,因此,需要尝试用新的方法、新的思路对现有版本进行覆盖回归测试,将开发人员从繁复的测试工作中解放出来,把有限的精力投入到更多的开发创新工作中。

3.2 解决思路

利用测试工具提供的脚本控制实现功能测试自动化是项目找到的解决方案。以Onetest为例说明,Onetest自动化测试工具,它可以设定测试执行步骤,测试用例执行时间、顺序,提供对测试例状态的判断,提供循环控制等。

项目的功能分为信令处理和媒体处理,自动化测试重点关注的是信令层面的流程和功能测试。如图1所示为整个自动化设计和测试的过程。对于项目用例收集,主要从以下两方面入手,具体如图2所示。

- 1) 最小测试集测试,包含目前X的所有基本功能点;
- 2) 功能点测试,针对目前每个功能点进行一些特殊的测试,比如异常测试、兼容测试等。

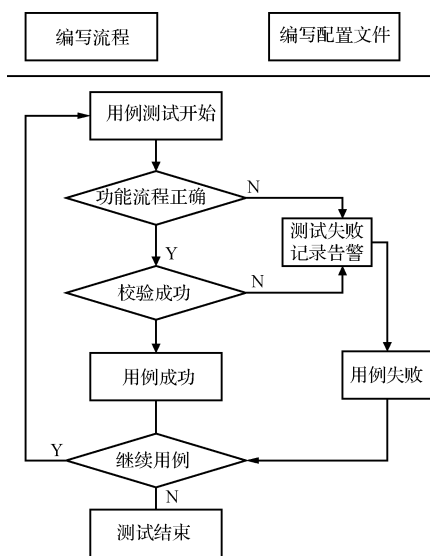


图1 自动化测试的完整过程

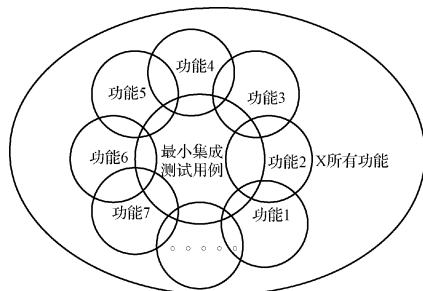


图2 功能覆盖关系

3.3 实践情况

自动化测试设计的原则是:可移植、可维护、可扩展等,下面使用测试工具Onetest介绍项目的自动化测试过程。

1) 根据功能编辑流程,如图3所示。对于整个项目的功能进行归纳,形成最小集成测试用例,据此对于产品进行一个整体的基本测试;其次,根据产品的每个功能点进行整理,包含功能重合、边界场景、异常场景等;将自动化测试用例和用例文档一一对应,这样以避免用例的遗漏,保证后期的追溯;流程执行过程中执行是否成功可以通过一些正则表达式组合判断,如图4所示。

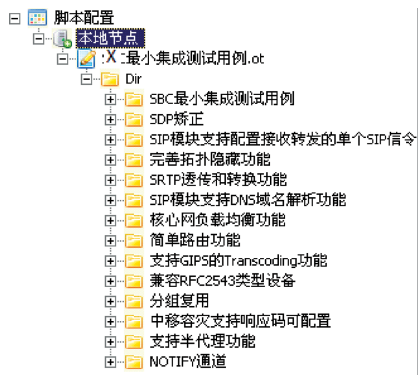


图3 自动化测试功能组合

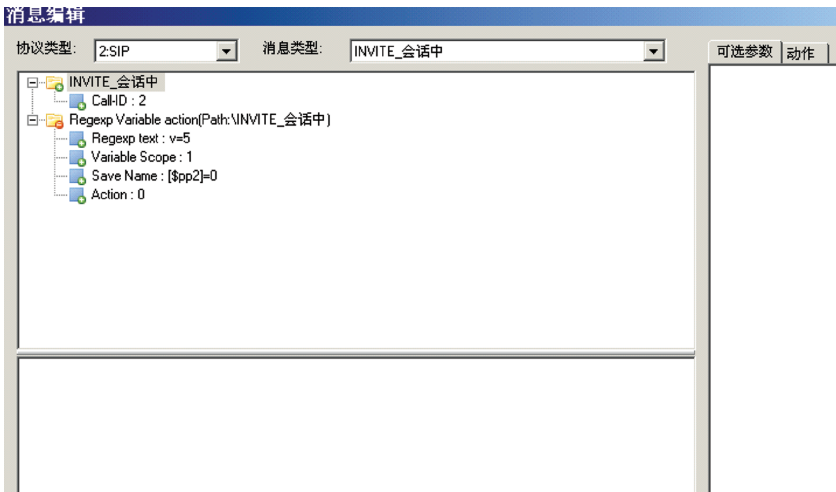


图4 正则表达式

2) 对于不同的功能，项目的配置是不同的。在自动化的执行过程中需要能够自动修改配置并且功能测试完可以自动恢复，保证不同用例之间的独立性。项目的配置修改首先需要telnet连接，下发命令，完成配置修改。OTSend命令是OneTest2.0中扩展的Tcl命令，可编写用户的Tcl脚本中，用于通过OneTest2.0的Telnet客户端向前台服务器发送命令。如图5所示为脚本内容，使用txt文件编写，保存为tcl文件，将其插入到编写的脚本中以便后续执行，如图6所示。

3) 在测试工具配置脚本后，使14个文件，180个用例依次运行，一共3 h左右的测试，不再需要人为干预，测试结束时即可获取结果文件。

3.4 效果评价

目前，项目的自动化测试定期运行（每周二和每周五）并且已经运行了一段时间，期间发现了不少故障，

包括功能点异常、进程挂起等。

1) 使用简单

开发人员将用例拷贝到本机，使

用时只需将现有用例中的IP地址批量替换成本地IP，即可使用验证修改的代码对于原有功能的影响。

表1 运行效果评价

运行时间/月	发现故障/个	严重故障/个
2	11	3

2) 扩展方便

虽然现在测试用例覆盖的功能只有14个，用例180个左右，但是随着项目的扩展和后续用例的完善，功能点会不断的增加，但是目前的用例每个功能点均按照文件夹的形式构造，后续开发人员每开发一个功能，均增加一个文件夹并根据集成测试文档构造对应的用例；如果需要将此用例应用

```
OTSend 10.44.61.17 "end" 1000
OTSend 10.44.61.17 "conf t" 1000
OTSend 10.44.61.17 "sbc" 1000
OTSend 10.44.61.17 "signal-portal" 1000
OTSend 10.44.61.17 "signal-group 2" 1000
OTSend 10.44.61.17 "no signal-nexthop 2" 1000
OTSend 10.44.61.17 "end" 1000
OTSend 10.44.61.17 "conf t" 1000
OTSend 10.44.61.17 "sbc" 1000
OTSend 10.44.61.17 "system" 1000
OTSend 10.44.61.17 "activate signal-group 2" 1000
OTSend 10.44.61.17 "exit" 1000
OTSend 10.44.61.17 "signal-portal" 1000
OTSend 10.44.61.17 "signal-attribute" 1000
OTSend 10.44.61.17 "sip" 1000
OTSend 10.44.61.17 "link-check nexthop nni-side disable" 1000
OTSend 10.44.61.17 "end" 1000
```

图5 脚本内容

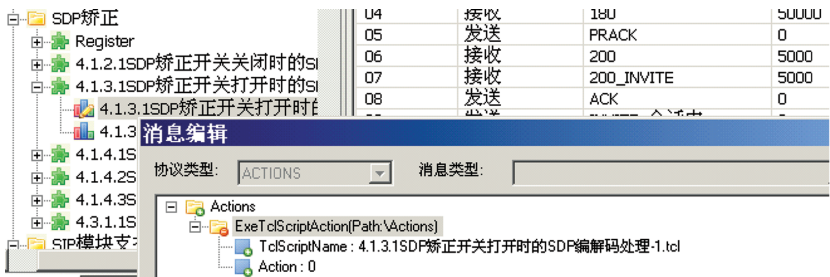


图6 用例中嵌入配置文件

到其他版本,也只需要修改一些配置文件即可。

3) 复用性强

不同的大功能,可能都需要测试某些小功能,比如项目的最小集成测试用例和核心网负载均衡功能中都考虑了紧急呼叫测试,此时两者的用例是相同的,但是配置是不同的,因此只需要修改配置文件和配置文件的说明。

3.5 推广建议

机架真实环境自动化测试的开发,是基于真实环境与版本,全部采

用真实模块,没有进行任何接口打桩,在涉及到与流程相关的配置修改时采用工具自动发消息实现人机命令的配置修改,批量执行,测试结果正确与否一目了然。此自动化测试方式由于是基于真实的环境,所以测试结果相对VC工程的单模块测试更全面,也更可靠,当用例覆盖所有功能后也可保证所有功能回归测试的全面性,后续推广后可大大节约集成测试的人力投入,意义重大。

4 总 结

实施自动化测试能对软件开发各

个阶段进行快速、全面的测试,从而提高软件开发效率,节约开发成本,保证开发质量,最终缩短软件交付周期。但实际上,并非所有的自动化测试技术都可以达到测试活动的预期目标,所以在实施自动化过程中,要有所为,有所不为,结合自身特点,寻找合适的切入点,在质量和效率、投入和产出之间寻找平衡点。

参考文献

- [1] 呼晓黎.软件自动化测试技术的研究与实现[D].成都:西南交通大学,2007.