

基于持续集成的软件冒烟测试优化探索

吴云梅¹ 李蕊¹ 查诚¹ 黄华^{1,2} 张润哲¹ 蔡卓航¹ 谢晓如¹

WU Yunmei LI Rui ZHA Cheng HUANG Hua ZHANG Runzhe CAI Zhuohang XIE Xiaoru

摘要

基于持续集成的软件冒烟测试是软件快速开发迭代周期中的重要环节,软件的性能测试又是对客户体验影响很大的测试阶段。结合软件冒烟测试和性能测试的特点、测试流程,多维度深入分析和对比软件冒烟测试和性能测试的差异性,提出将软件性能测试的测试用例左移,适度融合到软件冒烟测试阶段。实践表明,在兼顾软件冒烟测试执行效率的前提下,基于持续集成的软件冒烟测试与软件性能测试的适度融合,实现了软件冒烟测试和性能测试的测试效能与软件质量的双提升。

关键词

冒烟测试;性能测试;测试左移;持续集成;自动化工厂

doi: 10.3969/j.issn.1672-9528.2024.05.031

0 引言

随着近年来计算机软件行业开发技术的飞速发展,软件快速开发迭代的节奏愈发普及,传统的提测—>测试—>回归流程无疑过于死板和迟钝,软件产品中存在的很多问题不能在第一时间反馈给软件开发流程中的各角色,而基于持续集成的软件冒烟测试^[1-3](smoke testing)则是集合了软件开发整个团队的力量共同助力产品的质量,使得在软件代码提交到软件代码库的早期就扫清软件的最基本缺陷,确保软件产品的质量。

软件冒烟测试最早由微软公司首先提出,微软一直提倡每日构建版本(build),软件冒烟测试就是在每日构建版本(build)后,对系统的基本功能进行简单的测试,而不会对软件的具体功能进行更深入更全面的测试,所以软件冒烟测试也叫作BVT(build verification testing)。在长期的软件冒烟测试过程中,笔者及团队发现在软件冒烟测试中融入轻量级的软件持续性能测试,可以达到二者优势互补的作用。以下笔者根据自身工作实际对软件冒烟测试和性能测试的特点、测试流程进行详细阐述,对二者的差异性进行深入比较和分析,提出了将软件性能测试的测试用例部分左移到冒烟测试阶段,将冒烟测试和性能测试合理融合,由真实数据展

现在在实际项目中将软件冒烟测试和性能测试二者结合之后产生的良好效应。

1 软件冒烟测试的特点和测试流程

冒烟测试是针对软件新版本的主分支(trunk)进行详细全面测试之前的预测试(pre-test),执行冒烟测试的主要目的是快速验证每日构建的新软件版本基本功能,确保新版本没有引入阻塞性的基本缺陷。如果冒烟测试出现必然性的用例执行失败,则不必将该版本发布到下一测试环节。此时,需要测试人员和开发人员以最高优先级共同协作定位和调查问题,直至找到有问题的代码,并将有问题的代码直接回退,使产品主分支恢复正常至可提交新版本的状态。

冒烟测试中涉及软件开发工程师、冒烟测试工程师、持续集成自动化工厂等。各个角色的分工和具体职责如下。

(1) 软件开发工程师主要负责新软件代码的设计和提交,软件冒烟测试失败时和冒烟测试工程师协同调查冒烟测试失败的问题,对于由软件问题造成的冒烟测试失败,需要以第一优先级修复软件问题,保证主版本分支的纯净。

(2) 冒烟测试工程师主要负责软件冒烟测试用例的设计、冒烟测试环境的搭建(包括软件和硬件方面的冒烟测试环境)、冒烟测试结果的跟踪及反馈。当冒烟测试失败时,必须以第一优先级和软件开发工程师协同调查问题,针对冒烟测试环境问题,必须以最高优先级修复环境问题,以确保软件冒烟测试流程的正常进行,保证软件主分支实时交付新的软件版本。

(3) 持续集成自动化工厂^[4-5]主要负责自动化编译新的软件包,实时为冒烟测试提供测试对象和统一的测试环境,

1. 浙江经济职业技术学院智能控制研究所 浙江杭州 310018
2. 湖州荣登智能科技有限公司 浙江湖州 313009
[基金项目] 2022年度浙江省教育厅一般科研项目(Y202250380);浙江经济职业技术学院校级项目(X2023040);浙江省省级工业新产品开发项目(202302A00059)

冒烟测试成功后及时将合格的新软件版本自动交付给下一级测试。

2 软件性能测试的特点及测试流程

软件性能测试^[6] (performance testing) 属于软件测试中黑盒测试的一种类型, 它关注的是软件功能之外的系统各项指标。软件性能测试具体可以分为: 负载测试 (load testing)、压力测试 (stress testing)、并发测试 (volume testing)、基准测试 (capacity testing)、稳定性测试 (stability testing) 和可恢复测试 (recovery testing), 具体见图 1。

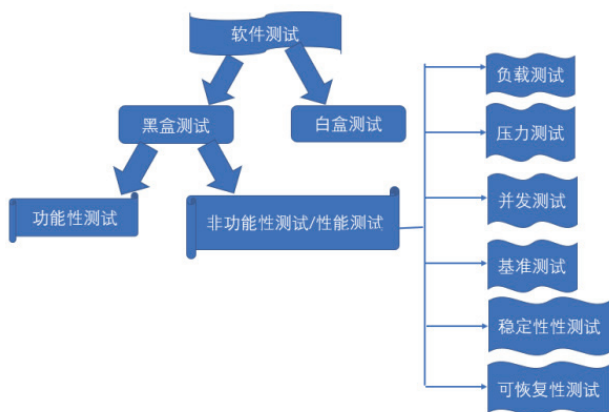


图 1 性能测试

(1) 负载测试: 指对系统不断地增加压力或增加一定压力下的持续时间, 直到系统的某项或多项性能指标达到安全临界值, 使系统达到瓶颈, 为系统调优提供有效的参考数据。

(2) 压力测试: 分为稳定性压力测试和破坏性压力测试。稳定性压力测试是在不同的给定的条件下 (比如内存的使用、一定时间段内有多少请求等), 系统表现出来的处理和反应能力; 破坏性压力测试是不断加压, 直至系统崩溃, 从而得出系统的最大承受能力的阈值。

(3) 并发测试: 测试多个用户同时访问同一个应用、同一个模块时, 是否存在死锁或者其他性能问题。

(4) 基准测试: 在给系统施加较低压力时, 查看系统的运行状况, 得出系统的基础参考值或者基准范围。

(5) 稳定性测试: 在给系统加载一定业务压力的情况下, 使系统运行一段时间, 以此检测系统是否稳定。

(6) 可恢复性测试: 针对有多余备份和负载均衡的系统设计, 检测如果系统局部发生故障, 系统能否继续使用。

综上, 性能测试与功能测试的区别在于, 功能测试关注的是软件能“做什么”, 而性能测试关注的是软件“做得怎么样”。

软件性能测试的流程与功能测试的流程类似。首先需要

收集软件性能测试的需求, 基于软件性能测试的需求文档, 由软件性能测试工程师设计出具体软件性能测试的测试文档, 软件性能测试工程师基于软件性能测试的测试文档设计出性能测试用例, 如果采用自动化测试的方式, 还需要软件性能测试工程师设计出性能测试的自动化用例, 在新的软件版本发布的阶段, 开始系统的执行软件性能测试用例, 软件性能测试用例执行完毕后生成软件性能测试报告。性能测试工程师基于性能测试报告分析和调查问题, 属于软件问题, 性能测试工程师推动软件开发人员解决软件问题, 属于测试环境问题, 需要软件性能测试工程师修复测试环境问题。

3 软件冒烟测试与性能测试的差异点

软件冒烟测试必须以效率为先, 软件冒烟测试用例的执行和持续集成自动化工厂的设计^[7-8] 必须以尽量缩短冒烟测试时间, 提高测试效率为宗旨, 目的是在保证冒烟测试质量^[9-12] 的同时, 尽可能缩短软件迭代开发的周期。而性能测试更着眼于系统在特定的工作负载下的响应性、稳定性、伸缩性、可靠性, 直接关乎实际用户的体验感。以下笔者从测试频度、测试阶段、测试时效性、测试用例的深度、调查问题的复杂度、调查问题的优先级、问题解决和推进的速度, 多个维度深入比较和分析冒烟测试和性能测试的差异性。

(1) 测试频度方面

冒烟测试是每日每个 Build 执行^[13-14]; 性能测试仅仅是跨多版本定期执行。

(2) 测试阶段方面

冒烟测试是每个新版本提交到版本库的初期开始执行测试; 性能测试经常拖到各个测试阶段的最后期开始执行。

(3) 测试时效性方面

冒烟测试的执行时效性要求很高, 必须及时反馈软件测试结果, 以便更快发布新版本; 性能测试执行时效性要求一般, 只需定期反馈软件测试结果。

(4) 测试用例的深度方面

冒烟测试只测试系统最基本功能, 避免后期出现阻塞性问题; 性能测试需要在基本功能正常的基础上, 详细测试系统的各方面性能, 以保证客户有良好的使用体验。

(5) 测试用例的执行时间方面

冒烟测试的是在每一次的 build 生产之后执行, 只有执行通过, 才能将 x 新版本发布, 所以必须短时间内就要拿到冒烟测试的结果, 这意味着冒烟测试用例的执行时间必须短, 一般最长不超过 1.5 h; 性能测试时要求不高, 测试用例设计只需考虑性能的覆盖率, 不需要考虑测试用例的执行时间。

(6) 调查问题的复杂度方面

冒烟测试是针对单个新包问题排查, 调查对象单一, 问

题推进速度快,性能测试是针对跨版本问题调查,调查对象复杂,问题推进速度缓慢。

(7) 调查问题的优先级方面

冒烟测试的问题往往会影响到主分支正常版本迭代和发布,阻塞功能测试的进行,所以开发测试人员必须以最高优先级调查和修复软件或者环境问题;性能测试的问题往往没有对系统造成功能性破坏,问题的影响面有限,导致各人员重视力度不够,调查问题的优先级不高。

(8) 问题解决和推进的速度方面

冒烟测试的问题解决迅速,由于问题是在软件开发的初期就得以解决,所以问题解决的成本相对较低;性能测试的问题往往难以推动,经常出现问题的乒乓效应,导致问题解决成本很高。

4 软件冒烟测试和性能测试合理融合

基于以上诸多软件冒烟测试和性能测试的差异点,笔者和团队将软件冒烟测试和性能测试进行合理融合,在软件冒烟测试用例中加入了轻量级的性能测试检查点,可以很好地融合二者的优势。但由于软件冒烟测试对测试用例执行时间和执行时效性都有很高要求,意味着软件冒烟测试只能选择非常基本的测试用例和检查点,以保证冒烟测试的效率,能够快速展现每次冒烟测试执行后的测试结果,以便提交代码的工程师可以在短时间内获取当次的测试报告,及时定位提交代码的问题。因此,在冒烟测试中融入性能测试时,需要综合考虑两大原则。一是需要考虑整体冒烟测试用例的执行时间,所以不能选择执行时间过长的性能测试用例,不能因为性能测试用例的引入大大增加了整个冒烟测试执行的时间,以至于增加整个持续集成的周期;二是需要考虑性能测试用例的具体执行场景,尽量在正常冒烟测试用例能够覆盖的场景内,适度融合合理的性能测试用例,不能选择需要复杂场景的性能测试用例。综上,如何在冒烟测试用例中合理加入轻量级的性能测试?只能选择以满足冒烟测试用例的场景为前提,并且执行时间相对较快的性能测试用例作为软件持续集成中冒烟测试与性能测试融合的基准。

5 性能测试用例部分左移到冒烟测试的优势

在传统的瀑布式软件开发流程中,开发人员和质量保证专家在具有不同角色和职责的团队中工作。即使在敏捷方法中,测试也通常排在最后^[15]。这就导致软件测试问题发现和修复滞后,而问题发现越晚,意味着问题的修复成本越高。根据 Capers Jones 在《全球软件生产力和质量度量分析报告》中给出的图表数据,假设在编码过程中修复一个缺陷的成本是 1,在单元测试中修复一个缺陷成本是 4,而功能测试和系

统测试的单个缺陷修复成本分别是 10 和 40。如果遗漏线上产品,修复成本会涨到 640。这正是持续集成(CI)要解决的问题,要快速在软件开发的最初阶段发现尽可能多的问题,特别是严重的问题,必须及时修复。在确保持续集成一定周期的前提下,将能够与冒烟测试用例场景复用的部分性能测试用例左移到冒烟测试阶段。在每次新代码提交阶段,可以及时发现性能测试问题,如内存泄漏、App 偶然性崩溃等严重性能问题,测试人员可以快速复现问题,提供及时的问题现场,供软件开发人员快速定位性能测试问题,及时做出反馈,此时的软件修复成本大大降低。

在冒烟测试用例中合理加入轻量级的性能测试,其优势有以下几点。

(1) 在保证冒烟测试执行效率的同时,有效拦截了由新软件提交而引入的部分性能测试的问题。

(2) 性能测试的问题在软件开发的最初期得到最高优先级调查解决,问题调查成本降低。

(3) 性能测试的用例可以在冒烟测试中多次频繁地重复测试,性能测试的质量得以明显提升。

(4) 性能测试的加入提高了冒烟测试的质量,使得冒烟测试不仅仅覆盖最基本的功能点。

(5) 性能测试的基本问题在冒烟测试中解决后,性能测试阶段有更高的执行效率。

(6) 明显降低了后期性能测试中调查和修复问题的成本,达到了冒烟测试和性能测试的优势互补。

在过去 12 个月中,在冒烟测试阶段合理融入性能测试检查点之后,冒烟测试发现的各类型问题分布,主要分为软件问题、冒烟测试环境问题(包括自动化测试脚本和硬件环境)、自动化工厂问题,占据的失败比例分别为 94%、4%和 2%(图 2);软件问题中主要分为基本功能性问题和性能问题,占比分别为 72%、28%(图 3)。由此可见,软件性能问题导致的冒烟测试失败的比例不容忽视,且性能测试在冒烟测试中的合理融入,的确可以有效提升软件迭代开发过程中的软件质量。

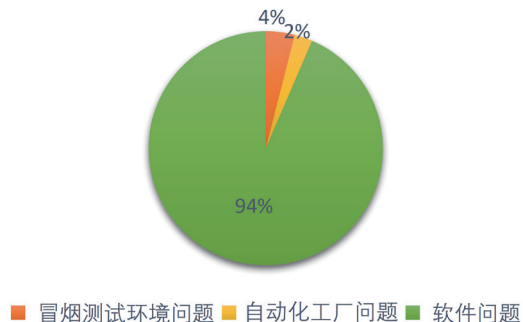


图 2 冒烟测试失败问题分布

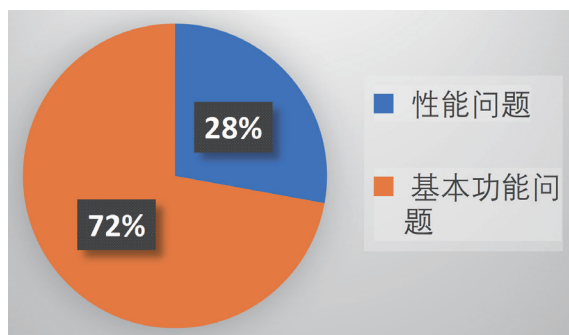


图3 软件问题类型分布

6 结语

长期的项目实践经验证明,在快速迭代的软件开发过程中,在基于持续集成的自动化冒烟测试用例中合理融入性能测试检查点,将性能测试的部分测试用例合理左移,不仅可以快速及时定位和发现产品最基本功能性问题,更能将有关性能问题在软件开发的早期得到及时反馈和纠错,有效避免在后期的测试环节中严重的性能问题大爆发,无疑大大提高了软件的质量和软件的开发效率,明显提升了客户的体验感和产品的口碑。

参考文献:

- [1] YOO S, HARMAN M. Regression testing minimization, selection and prioritization: a survey[J]. Software testing, verification and reliability, 2012, 22(2): 67-120.
- [2] KAZMI R, JAWAWI DNA, MOHAMAD R, et al. Effective regression test case selection: a systematic literature review[J]. ACM computing surveys, 2017, 50(2): 29.1.
- [3] 于波, 姜艳. 软件质量管理实践—软件缺陷预防、清除、管理实用方法[M]. 北京: 电子工业出版社, 2008.
- [4] SRIKANTH H, CASHMAN M, COHEN M B. Test case prioritization of build acceptance tests for an enterprise cloud application: an industrial case study[J]. The journal of systems and software, 2016, 119: 122-135.
- [5] CAMPOS J, ARCURI A, FRASER G, et al. Continuous test generation: enhancing continuous integration with automated test generation[C]// Proceedings of the 29th ACM/IEEE International Conference on Automated Software Engineering. New York, NY, USA: Association for Computing Machinery, 2014: 55-66.
- [6] NGUYEN CD, PERINI A, TONELLA P, et al. Automated continuous testing of multiagent systems[C]// Agent-Oriented Software Engineering X. Cham: Springer, 2009: 180-190.
- [7] 李坤宁. 单元测试和持续集成在企业级软件开发中的设计与实现[D]. 成都: 电子科技大学, 2011.
- [8] 王琦. 探析计算机软件测试方法[J]. 科学技术创新, 2021(4): 93-94.
- [9] YOO S, HARMAN M. Regression testing minimization, selection and prioritization: a survey[J]. Software testing, verification and reliability, 2013, 22(2): 67-120.
- [10] 肖利琼. 软件测试之魂核心测试设计精解[M]. 北京: 电子工业出版社, 2013.
- [11] 齐海燕, 黄少敏. 自动化回归测试在软件产品项目中的应用[J]. 工业控制计算机, 2018, 31(3): 43-44.
- [12] 李坤宁. 单元测试和持续集成在企业级软件开发中的设计与实现[D]. 成都: 电子科技大学, 2011.
- [13] MCCONNELL S. Daily build and smoke test[J]. IEEE software, 1996, 13(4): 143-144.
- [14] MYERS G J, BADGETT T, SANDLER C. The Art of Software Testing, 3rd Edition[M]. Hoboken, New Jersey: John Wiley & Sons, Inc, 2012.
- [15] 性能测试--1、为什么进行性能测试[EB/OL]. (2020-03-08) [2024-01-24]. <https://www.jianshu.com/p/359f3206ec37>.

【作者简介】

吴云梅(1986—), 通信作者(email: wuyun_mei@126.com), 女, 浙江杭州人, 硕士, 高级工程师, 研究方向: 5G+、软件开发与测试、软件质量管理等。

李蕊(1984—), 女, 浙江杭州人, 硕士, 高级工程师, 研究方向: 物联网、软件开发与测试等。

查诚(1979—), 男, 贵州贵阳人, 博士, 讲师, 研究方向: 情感计算、软件测试。

黄华(1979—), 男, 浙江杭州人, 博士, 工程师, 研究方向: 智能控制开发。

张润哲(2001—), 男, 浙江杭州人, 大专, 研究方向: 物联网应用技术开发与测试。

蔡卓航(2005—), 男, 浙江宁波人, 大专, 研究方向: 物联网应用技术开发与测试。

谢晓如(2005—), 女, 浙江江山人, 大专, 研究方向: 物联网应用技术开发与测试。

(收稿日期: 2024-03-15)