

GIS 平台软件自动化测试框架

崔雪¹, 王少华^{1,2}, 蔡文文¹

(1. 北京超图软件股份有限公司, 北京 100015;

2. 中国科学院 地理科学与资源研究所, 北京 100010)

摘要: 针对 GIS 平台软件规模不断增大导致的软件测试工作量加重, 手工测试无法保证 GIS 平台软件高质量运行的问题, 该文提出了一套适用于地理信息领域的软件自动化测试与监控管理体系。在分析 GIS 软件测试流程和需求的基础上, 结合 TeamCity、Sonar、WhiteBox 等 20 多个管理软件, 研发了适用于 GIS 软件的自动化测试和持续集成系统, 构成了软件测试与监控的管理体系。这一自动化测试框架在 GIS 软件生产流程的应用表明, 该框架可保障 GIS 平台软件的高效运行。

关键词: GIS; 平台软件; 自动化测试框架

【中图分类号】TP311.53 【文献标志码】A 【文章编号】1009-2307(2017)04-

Application of the automated testing framework for GIS platform software

CUI Xue¹, WANG Shaohua^{1,2}, CAI Wenwen¹

(1. SuperMap Software Co., Ltd., Beijing 100015, China;

2. Institute of Geographic Sciences and Natural Resources Research, CAS, Beijing 100101, China)

Abstract: Software testing can guarantee and improve the quality of the GIS platform. However, manual testing has no longer an effective way due to the increasing testing workload caused by the growing scale of the GIS platform. To resolve the problem, an automated testing and continuous integration system was developed in this paper based on the process and requirements of software testing. The framework was integrated more than 20 management software, including TeamCity, Sonar, and WhiteBox, which was applied for the whole production process of GIS software. Test results showed that the automated testing framework could guarantee the high quality of GIS platform.

Keywords: GIS; platform software; automated testing framework

0 引言

软件测试是保障和提高软件质量的重要手段^[1]。一个广泛应用的 GIS 平台软件, 软件开发者和使用者必须对软件进行充分的测试, 以确保其正常工作。以往在 GIS 平台产品研发过程中, 软件测试工作量往往占总工作量的 40%以上。随着超图平台产品的不断完善, 软件规模的不断加大, 平台软件总的有效代码数已经突破 500 万行, 代码总量已经相当于半个 Linux。这么大体量的代码, 单纯依靠过传统的手工测试是较难实现的; 并且 GIS 平台软件因为其自身的特点, 如高并发、数据量大, 使人工测试变得无法满足需求。因此为了提高测试效率, 越来越多的测试工作引入了自动化测试的思想和方法^[1]。实践证明, 软件自动化测试技术提高了软件测试的速度和效率, 节省了软件测试成本, 缩短了产品发布周期^[2]。自动化测试可以减少人工干预, 避免人为原因给测试工作带来的误差^[3]。所以, 采用自动化测试方法和相应的测试框架成为了软件开发组织测试工作的重要支撑手段。本文给出了一种实用、有效的自动化测试框架, 并验证了该框架在自动化测试领域的实用性和优越性。

1 GIS 软件测试体系特点

GIS 平台软件因其自身的特点, 决定着 GIS 测试体系不仅具备其他测试体系的特点, 还需满足如下要求: ①能够验证高并发性; ②能够验证数据安全性, 严格保障地理数据的安全性; ③能够验证网络信息安全性; ④能够随时更换不同的测试数据全方位验证功能的正确性和承压性; ⑤能够随时更换功能执行顺序, 执行顺序不同, 结果不同; ⑥由于 GIS 数据量非常大, 因此性能测试指标的重要性与功能正确性同等重要。

2 自动化测试框架的开发原则

根据 GIS 软件测试体系特点, 同时参考国内外主流测试技术的功能, 本文在设计自动化测试框架时, 遵循以下原则^[4-5]:

1) 实时, 有针对性。GIS 平台软件采用敏捷的研发流程^[6], 开发人员每天需提交大量的代码。原有测试原则是每个开发人员先手工测试自己的代码, 测试通过后再提交到测试代码库, 然

后每天全部集成到一起进行编译打包成一个版本统一测试，测试后发现和定位问题都很困难。亟待能提交代码后立即测试，发现问题立即反馈给提交人的框架。

2) 全自动化流程。在整个测试环节中尽量减少人工干预，从代码生产、测试、验证和打包的整个流程都能自动完成。一个流程工作完毕后能够有序地进入到下一个流程中，无需人工干预和驱动，既可节省人力和时间成本，又能提高工作效率。

3) 数据与关键字驱动测试^[7-8]。不能将数据与测试用例的执行顺序在自动化测试框架中固定，需能够随时更改，做到数据驱动和关键字驱动，这样的框架能够更加灵活实用。测试框架易于扩展、维护。测试框架实现高度模块化，可提高框架的维护性，并且该框架要易于扩展，以适应不断更新的平台产品和不断变化的测试方式。

4) 测试脚本所使用的测试语言与框架相互独立。对于 SuperMap GIS 平台产品，既有 Java 应用程序，也有 .NET 应用程序，该测试框架应独立于测试脚本所使用的语言，不能统一规定仅适用某一种语言。

5) 框架简单易用。测试人员的精力应投入到设计高效实用的自动化测试案例上，真正对产品功能和性能起到验证作用的是测试用例本身而非测试框架，不应该让框架的复杂性分散测试人员的精力。

3 持续集成和自动化测试框架

3.1 框架概括

依据自动化测试的开发原则，首先分析软件生产的流程。软件从生产到发布，要经历代码编写、提交、编译、审查、测试、打包、验证和发布等环节，各环节依次执行。在执行的过程中任一环节都有可能报错，比如在“编译”过程中，若开发人员代码编写有问题，就会造成编译出错，这就需要该测试框架能够针对性地提示产生错误的开发人员，使其立即修改。针对流程和需求，利用包括开源、商业和自主研发的 20 多个管理软件，如 TeamCity、Sonar、WhiteBox 等^[9]，本文研发了一套兼具自动化测试和持续集成^[10]的软件测试与监控管理体系。该体系由一个或多个自动化测试基础模块、自动化测试管理模块、自动化测试统计模块等组成，用以管理整个 GIS 软件生产的所有活动。

研发的 GIS 平台软件自动化测试系统框架如图 1 所示，其中，英文部分为执行该操作所使用的工具软件名称。

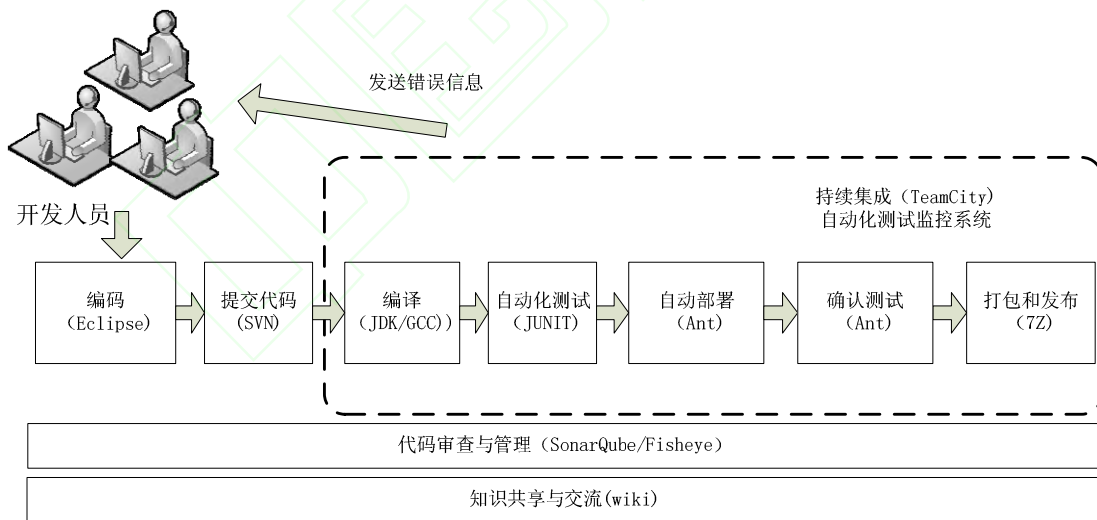


图 1 持续集成和自动化测试框架流程图

Fig.1 Framework of the Continuous Integration and Automated Testing

流程说明如下：

1) 开发人员使用 Eclipse 等开发工具编写代码，然后提交到 SVN 服务器上。

2) TeamCity 自动监测到新提交的代码，若 15 min 后没有其他人提交，TeamCity 将调用脚本通过 JDK 或 GCC 等工具执行代码编译。

3) 编译通过后，TeamCity 运行脚本调用 JUNIT 执行自动化测试，验证该功能的正确性。该测试属于单元测试。

4) 除每次提交代码后立即测试外, 该框架会在每天凌晨 0 点自动启动编译全部代码, 通过 Ant 脚本将完整的软件包部署到一台单独的机器上, 该机器模拟客户真实环境, 拥有客户类似的软硬件环境。

5) 部署到指定机器后, 接下来运行的就是确认测试。确认测试使用客户的真实数据, 将执行软件的全部功能, 以确认整个包的正确性。

6) 确认测试通过后, TeamCity 会调用打包工具 7Z, 生成完整安装包, 最后发布。

7) 代码审查与管理是利用 SonarQube^[11]或 Fisheye^[12]等工具随时查看代码的提交情况, 检查代码的规则符合度, 统计测试用例单元覆盖度和每个开发人员的代码编写数量和质量等信息。

8) Wiki^[13]用于交流的平台, 便于经验的分享与交流。

整个持续集成均由 TeamCity 控制, 所有环节的运行情况均可从自动化测试监控系统查看。每天早上开发人员便会收到一封邮件, 告知自动化测试的结果, 对于发现的错误会立即查找原因并修改。

3.2 代码审查

静态代码检查本身非常重要, 比如 C++代码, 对于变量应该赋初值, 若不赋初值, 就会在运行过程中出现问题, 而且这种问题事后查找非常困难; 再比如记录集 Recordset 用完需要释放, 否则运行一段时间会出现崩溃, 且出现这种崩溃时在 Debug 调试阶段定位相关问题非常困难, 错误指针会不确定地指向某一个没有问题的代码行。在静态代码检查中, 除了使用 SonarQube 或 Fisheye 等商业软件外, 本文研发了 WhiteBox 和 CppCheck 两个工具。当开发人员提交代码后, 就会触发这个检查器的运行, 静态检查代码的规则符合度, 若发现代码不符合规范, 就会显示不符合规范的代码行, 并发送错误信息给代码提交人。检查规划部分内容如表 1^[14]所示。

表 1 C++代码编写规范
Tab. 1 C++ Coding Specifications

规则名	规则含义	说明
MemberVariableMustSetFirstValue	成员变量要在构造函数中赋予初值, 除非该成员变量是有构造函数的对象类型	Debug 下没有问题, Release 下就是有问题; 因为 Debug 下编译器会自动赋予默认值。这类问题出现后非常难查, 最开始预防好就能为后续节省很多时间
ReleaseMemoryInFunction	函数中内存应该被释放, 除非是作为函数返回值, 或引用型参数等	这类问题最主要的现象是有些代码分支没有释放内存, 如果一个函数稍稍复杂, 容易出现该问题; 如果一次调用, 哪怕只有几个字节的内存泄露, 服务器产品持续调用一个月后, 也很可能把内存耗光了
MustReleaseRecordset	记录集用完必须释放	记录集占用了数据库的资源, 若没有得到及时的释放, 之后再 Query 将得不到记录集, 服务器产品出现白图
MemoryNewDeleteMatch	内存的分配和释放要配对, 如 new、delete、new[]、delete[], malloc、free	如果不匹配, 有可能导致内存释放不彻底, 等同于内存泄露; 还有可能导致程序崩溃
JudgeNotNullAfterQuery	记录集在查询得到后, 要判断是否为 NULL; 不为 NULL 才能使用	如果传入进行查询的条件非法, 查询返回的记录集指针会为 NULL, 如果不判断就直接使用, 程序将出现崩溃

3.3 自动化测试

单纯运行单元自动化测试用例的框架有 4 种, 即模块化测试框架、测试库框架、关键字驱动的测试框架和数据驱动测试框架。模块化测试框架需要创建小而独立、可以描述的模块、片断以及待测应用程序的脚本, 这些树状结构的小脚本组合起来, 就能组成用于特定测试用

例的脚本。测试库框架与模块化测试脚本框架类似，并具有同样的优点；不同的是测试库框架把待测应用程序分解为过程和函数而非脚本，该框架需要创建描述模块、片断以及待测应用程序的功能库文件。关键字驱动框架需要开发数据表和关键字，这些数据表和关键字独立于执行它们的测试自动化工具，并可以用来驱动待测应用程序和数据的测试脚本代码。数据驱动测试框架通过捕获工具或者手工生成的代码脚本被载入到变量中，在这个框架中，变量不仅被用来存放输入值，还被用来存放输出的验证值，整个程序中，测试脚本来读取数值文件，记载测试状态和信息。

依据上文提到的自动化测试框架的开发原则，结合 GIS 平台软件的自身特点，本文采用关键字驱动和数据驱动相结合的方式。将自动化测试用例存放到不同的测试机器上，每个测试用例有自己的优先级，所有的测试用例信息都存放在一个公共的表中，每个用例用到的测试数据文件名称和路径也存放在一个公共的 xml 文件中，更改测试数据仅需更改这个指定的文字即可。JUnit 自动化测试框架会先从数据库中读出测试用例信息，然后生成一个待执行的序列，按照这个序列依次执行测试用例，执行过程如图 2 所示。

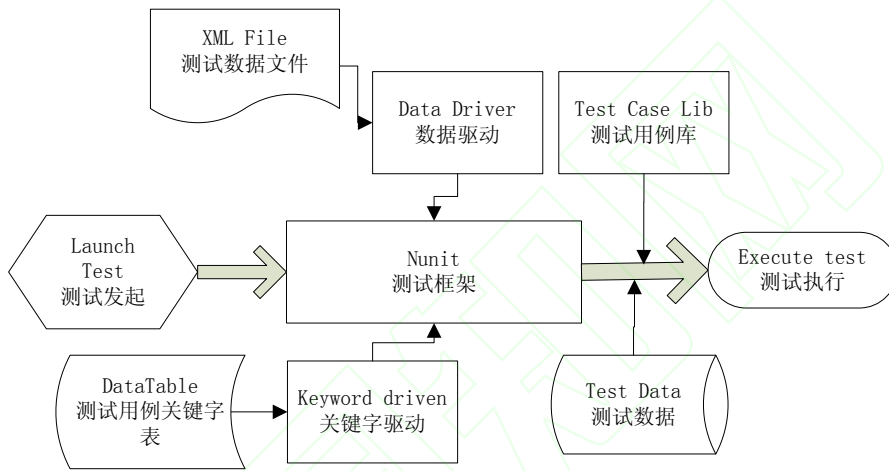


图 2 关键字驱动和数据驱动执行过程图

Fig. 2 Execution Process Driven by Keywords or Data

4 性能自动化测试框架

对于 GIS 平台软件，功能的正确性固然重要，但性能也是决定一个软件是否可用的重要指标^[15]。性能的评判标准有很多，其中一个重要的指标为客户的可接受程度，但因人而异，每个人的接受程度不一样^[16]。比如一幅地图缩小一次需要 3 s，有人认为可接受，有人认为很慢不能接受。针对这种情况，本文采用对比测试方法，即同一功能在同样的环境下执行同样的操作，将本文软件与业内主流软件的耗时进行对比。

采用 python 作为自动化测试的脚本驱动语言，利用数据驱动与关键字驱动混合的方式。脚本里依次编写所要执行的测试案例的关键字，数据库表中存放每个测试用例使用的测试数据，同时脚本要记录每个功能完成的起止时间，将测试结果写入到一个的数据库表中。对于组件产品，数据库中仅记录执行某一功能所耗费的超止时间；对于服务器产品，需要记录 5 个指标，即“90%用户时间”“吞吐率”“点击率”“内存使用率”和“CPU 占用率”，然后对这 5 个指标按重要性的高低，采用加权平均的方式计算得出平均分，实时展现结果。

评价性能时，系统根据执行耗时的不同会给出一个分值，性能高者定义为 100 分，性能低者根据耗时与性能高者的倍数关系确定分值。如执行 FindPath 功能，A 软件用时 2 ms，B 软件用时 0.5 ms，那么 B 软件的得分即为 100 分，A 软件得分为 25 分。

5 自动化测试框架在 GIS 平台软件中的应用

配合敏捷开发流程，SuperMap 的平台产品已在此测试框架下 7×24 小时不间断运行，实现了代码编写、提交、编译、审查、测试、打包、验证、发布的全自动化流程。

目前功能自动化测试用例的编写主要包括：软件新增的接口和函数，测试时或是用户使用时发现的缺陷，测试人员模拟用户使用过程设计的一套自动化测试用例。所有的测试情况和统计值都可实时展示，每个团队都可看到自己团队以及其他团队的测试情况。

6 结束语

自动化测试框架的全面使用,使 SuperMap GIS 平台软件的测试人员将注意力从原始而繁重的手工测试转变成了自动化测试框架的开发、自动化测试用例和测试场景的设计工作。此工作对测试人员的要求也在逐渐提高,产品的质量保障工作较之以前有很大的提高。

依靠该自动化框架来完成 GIS 平台软件的测试,使一个 500 多万行的代码测试从不可能完成到 72 h 内全部测试完成。同时该框架可以有效避免同一个缺陷在不同的版本里反复出现,可极大减少人工干预,提高测试的稳定性。另外该框架能够执行多用户多进程压力测试,这是人工测试所不能做到的。目前该框架运行稳定,为 GIS 平台软件的高质量保驾护航。

参考文献

- [1] DUSTIN E, RASHKA J, PAUL J. Automated software testing: introduction, management, and performance[M]. New York: Addison-Wesley Professional, 1999: 1-22.
- [2] 朱少民. 软件测试方法和技术[M]. 3版. 北京: 清华大学出版社, 2014: 1-90.(ZHU Shaomin. Method and technology of software testing[M]. 3rd Edition. Beijing: Tsinghua University Press, 2014: 1-90.)
- [3] 牛璐. 手机软件自动化测试方法研究与应用[D]. 郑州: 信息工程大学, 2012: 1-74.(NIU Lu. Research and application of automatic testing for mobile phone software[D]. Zhengzhou: Information Engineering University, 2012: 1-74.)
- [4] BERTOLINO A. Software testing research: achievements, challenges, dreams[C]//IEEE Computer Society. FOSE '07 2007 Future of Software Engineering. Washington: IEEE Computer Society, 2007: 85-103.
- [5] YOUNG M, PEZZE M. Software testing and analysis: process, principles, and techniques[M]. New Jersey: John Wiley & Sons, 2008: 1-479.
- [6] ABRAHAMSSON P, SALO O, RONKAINEN J, et al. Agile software development methods: review and analysis[M]. Finland: VTT Publications, 2002: 1-107.
- [7] LAUKKANEN P. Data-driven and keyword-driven test automation frameworks[D]. Finland: Helsinki University of Technology, 2006: 1-112.
- [8] ZAMBELICH K. Totally data-driven automated testing[J]. Retrieved April, 1998(4): 1-17.
- [9] MEYER M. Continuous integration and its tools[J]. IEEE Software, 2014, 31(3): 14-16.
- [10] YÜKSEL M, TÜZÜN E, GELIRLI E, et al. Using continuous integration and automated test techniques for a robust C4ISR system[C]//ISCIS. The 24th International Symposium on Computer and Information Sciences. North Cyprus: ISCIS, 2009: 743-748.
- [11] MAURIZIO M, ANDRAES J, TOBIAS L, et al. Source code analysis of flight software using a SonarQube based code quality platform[J]. Ada User Journal, 2015, 36(2): 99-103.
- [12] SIDOR T R, JENSEN K. Fisheye views of source code using dynamic degree of interest functions[EB/OL]. (2007-08-01)[2015-11-16]. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.118.4390&rep=rep1&type=pdf>.
- [13] LEUF B, CUNNINGHAM W. The Wiki way: quick collaboration on the Web[EB/OL]. (2001-04-13)[2015-11-16]. <http://www.citeulike.org/group/1136/article/114322>.
- [14] HOFF T. C++ coding standard[EB/OL]. (2008-03-01)[2015-11-16]. <http://www.possibility.com/Cpp/CppCodingStandard.html>.
- [15] DUVALL P, MATYAS S, GLOVER A. Continuous integration: improving software quality and reducing risk[M]. New Jersey: Pearson Education, 2007: 1-275.
- [16] 百度百科. 软件性能的各界视角 [EB/OL].(2014-07-15)[2015-11-16]. <http://baike.baidu.com/view/1812806.htm>.

作者简介: 崔雪 (1982—), 女, 山东即墨人, 硕士, 主要研究方向为地理信息软件技术应用。

E-mail: cuixue@supermap.com

基金项目: 北京市科技专项 (Z151100003615012, Z151100003115007, Z141101004414011); 中国科学院重点部署项目 (KZZD-EW-07-01-001); 国家科技支撑计划项目 (2013BAC03B00); 资源与环境信息系统国家重点实验室自主研究项目 (088RAC00YA); 测绘公益项目

(201512015); 朝阳区博士后基金项目

收稿日期: 2015-12-07

通信作者: 王少华 博士 E-mail: wangshaohua@supermap.com



中国知网