

文章编号: 1005-8451 (2007) 03-0010-03

基于统一过程原则的软件自动化测试研究

常 征, 陆 守 一

(北京林业大学 信息学院, 北京 100083)

摘 要: 软件自动化测试以较高的效率, 广泛的覆盖率, 成为测试行业的主流。介绍统一过程 (RUP) 和自动化测试及手工测试, 总结目前自动化测试过程中存在的主要问题。重点分析在 RUP 原则的指导下, 如何解决这些问题, 实现测试自动化, 从而高效率地进行软件测试, 促进软件自动化测试的发展。

关键词: 统一流程处理; 自动化测试; 生命周期; 迭代开发

中图分类号: TP31 : TP206 **文献标识码:** A

Research on software automatic testing based on RUP principle

CHANG Zheng, LU Shou-yi

(Institute of Information, Beijing Forestry University, Beijing 100083, China)

Abstract: With high efficiency and wide coverage, software automatic testing was the main stream of testing industry. How to successfully implement automatic testing was one research focus. It was introduced Rational United Process (RUP), automatic testing and manual testing, summarized the main problems at present during automatic testing process. In order to perform software testing with high efficient, and accelerate software automation testing's development, it was mainly analyzed how to overcome these problems and succeed to implement test automation.

Key words: rational united process; automatic testing; lifecycle; iteration development

随着计算机应用的日益普及和深化, 软件规模越来越大, 逻辑也越来越复杂。针对传统手工测试效率低、成本高的缺点, 软件自动化测试应运而生, 引起了人们的广泛关注。如何有效、成功地进行自动化测试是自动化测试研究的重要主题之一。

统一过程 (RUP) 思想涉及有关软件开发生命周期、开发准则和方法、开发成员的责任和作用、开发过程的管理和文档管理等很多方面的内容, 形成一套系统的方法。本文提出结合 RUP 原则, 解决软件自动化测试中遇到的主要问题。把自动化测试项目视为软件工程项目, 是自动化测试项目成功实施的基础。

1 统一过程处理与自动化测试

1.1 统一过程简介

统一过程 (RUP) 是一种新的面向对象的软件开发过程, 它对软件开发过程中的技术因素和管理因素进行了全面考虑, RUP 提出的开发过程同样可以有效应用到自动化测试中。RUP 关注的是开发过

程中的生命周期, 通过迭代开发的方式, 化解开发过程中的重大风险, 满足客户变化的需求。

RUP 中的软件生命周期在时间上被分解为 4 个顺序的阶段, 初始阶段、细化阶段、构建阶段和交付阶段。任何一个软件工程方法的引入都是为了减少风险, 增加成功的可能性, RUP 也不例外, 在每个生命周期内都管理相应的风险, 如图 1 所示。

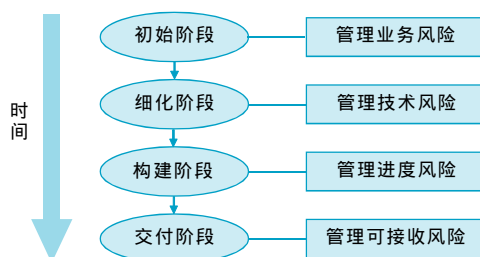


图 1 软件生命周期

RUP 中的每个阶段可以进一步分解为迭代, 通过迭代开发软件得以增量演进, 每个迭代都以一个可执行的产品发布而结束。一次迭代可包括以下活动: 初始计划、定义需求、分析设计、实现、测试和部署, 据其在开发周期的位置不同, 所占比重也不同。

收稿日期: 2006-08-17

作者简介: 常 征, 在读硕士研究生; 陆 守 一, 教授。

1.2 手工测试与软件自动化测试

手工测试的特点就是简单，但是它存在的问题比其带来的好处要多得多。手工测试可能引入人为的输入错误，尤其在数据量大的情况下；大量重复性的手工测试可能成本较高，如果考虑软件发生改动而需要重复手工测试的情况，这个成本还会更高；手工测试的覆盖面不广，只能够测试系统的输入和输出；没有办法对组件进行隔离的测试，从而导致发现问题和解决问题的成本都太高。针对手工测试的缺点，自动化测试也随之产生。

针对手工测试，自动化测试的优势很多：规范测试流程、提高测试效率，测试覆盖率等。很多人对自动化测试存在误区，将其理解为找到一种自动化测试工具，把它应用到软件工程项目中，自动化测试工具只是被看作是一种录制和回放的工具。事实上自动化测试远不止这么简单，按照成熟度自动化测试可以被划分为5个级别，录制和回放仅是自动化测试中的最低级别，如图2所示。

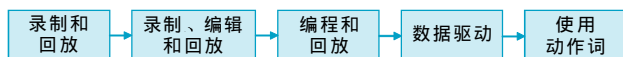


图2 成熟度自动化测试级别

2 目前软件自动化测试存在的主要问题

自动化测试的优势很多，很多单位都引入了自动化测试工具，实施自动化测试。但是在实施过程中却普遍存在很多问题，导致整个测试工程失败。普遍存在的问题如下。

2.1 盲目选择测试工具

选择测试工具目的性不强，比较盲目。选择哪种测试工具应该受被测软件的特点和测试的目的及经济承受能力。一般测试工具的软件厂商会强调其工具的优势、有利的或成功的一面，可能对此软件的缺点和实施过程中遇到的困难却只字不提。结果，由于对测试工具认识不清，导致此次测试的失败。

2.2 对自动化测试期望过高

认为所有的测试工作都应该自动化实现，自动化测试能发现很多错误。实际上100%的自动化测试是一个理论上的目标，要全部实现自动化，代价十分昂贵。另一方面，发现错误的数量主要是与测试用例的设计有关，与具体用什么测试工具关系不是很大。

2.3 轻视系统分析与设计

认为自动化测试就是录制/回放脚本，不需要系统进行分析与设计，只需录制测试脚本。例如GUI测试，产品界面的改变对脚本的正常运行影响较大。如果不在开始对系统进行很好的设计，很容易导致在界面发生改变以后，以前的测试脚本全部作废。所以在项目开始初期，应利用面向对象的方法或其他对项目进行分析与设计，保证项目的可扩展性和组件的重用性。

2.4 忽视脚本质量对测试的影响

默认开发的测试脚本不会有错误。其实测试脚本本身就是程序，所以说就有可能产生错误。很多时候测试项目时发现的错误是测试脚本的问题。所以一定要重视测试脚本的质量，保证其正确性。

2.5 忽略需求变化对测试的影响

测试的目的就是为了保证产品的质量，满足用户需求。如果发现最后的产品偏离了用户的需求，这个问题就很棘手，测试也变得没有意义。在实际的项目中，很少有需求在整个项目固定不变的情况。因此需求的变化对测试的影响很严重，应重视这个问题。

3 基于RUP原则解决软件自动化测试中的主要问题

软件自动化测试工程和软件开发过程很相似，测试脚本就相当于开发程序。因此，RUP原则也可以运用到自动化测试中来。RUP的5个主要策略，可以解决目前软件自动化测试存在的5个主要问题，减少自动化测试的风险。这5个主要策略为：

- (1) 初始计划—促使正确选择测试工具；
- (2) 定义需求—矫正对自动化测试的过高期望；
- (3) 基于组建的体系架构—为系统设计指明方向；
- (4) 持续的质量验证—保证脚本质量；
- (5) 管理项目需求和迭代开发—应对需求变化对测试的影响。

3.1 初始计划

初始计划解决盲目选择自动化测试工具等问题。RUP提出计划就是投资。凡事预则立，不预则废。初始计划对自动化测试项目的成功实施奠定了基础，计划可以统筹整个项目。一个完整的测试计划应该考虑一下几个方面。

3.1.1 测试风险

测试风险是各种各样的。例如，需求可能迅速

地变更,这可能会破坏可测试性或设想的测试成本。后继的精细化阶段可能会利用新的测试难点的解决方案来解决技术风险。

3.1.2 可测试性

可测试性与可行性直接相关。一些需求是非可测试的,因为他们是主观的,或者不是有助于度量或量度的。一些是不可测试的,因为从技术上很难进行合适的测试。例如,一些战略防御计划,由于格外昂贵的硬件而不能很容易地进行测试。

3.1.3 准备测试环境

准备测试所需的硬件、软件和数据等资源。硬件包括所需的计算机、网络等;软件包括被测软件、自动化测试软件,测试所需的操作系统等;数据包括用于性能、功能和可用性等方面需要的数据集。

3.2 定义需求

定义需求解决对自动化测试的过高期望等问题。自动化测试需求描述了自动化测试的内容,定义哪些需要自动化测试,哪些需要手工测试。很多人认为自动化测试显然是一件好事情。但是,他们不愿意对自动化测试的内容给出清晰的描述。这样会造成目标不统一,在实施软件自动化测试过程中会偏离原来需求的初衷。

RUP 推荐以下几个适合实行自动化测试的类型:

- (1) 重复性最大—例如数据的边界值测试、回归测试等;
- (2) 冒烟测试—每个发布版本提交测试前的基本功能确认;
- (3) 配置测试—需要在不同支持平台的测试;
- (4) 复杂的测试—难以手工执行,或者容易出错,也难于自动化测试,但可做相应考虑;
- (5) 需要对测试结果做电子记录的测试。

3.3 基于组建的体系架构

基于组建的体系架构为系统设计指明了方向。设计模板告诉设计者怎样处理最重要的问题。没有一个坚实的架构,或设计模板,就不会有项目大幅度的前进。

基于组件的体系架构有以下的优点:

- (1) 软件在本系统和其他系统中的重新使用更加方便;
- (2) 允许高效的并行开发,在系统设计中提供了方便的抽象概念;
- (3) 使得系统更加容易升级和维护。

3.4 持续的质量验证

持续的质量验证用于保证测试脚本的质量。RUP 分为4个阶段,每一个阶段的重点放在开发周期内一个特定的方面:起始,精细化,构建和产品交付。

在每一个阶段中,RUP 提供了在开发中验证项目质量的机会,从而尽早找到错误和潜在的改进。在RUP 中,每一次迭代都要检验上一次增加的质量。首先检验问题是否被正确理解,并且存在一个可靠的业务场景。如果不是,则继续检验,直到达到目的,或者停止这个项目,节省时间和金钱。在验证系统功能之前,也检验是否已经建立了一个十分牢固的架构。

3.5 管理项目需求和迭代开发

管理项目需求和迭代开发控制了需求变化对测试的影响。RUP 在开发周期内进行需求管理。这意味着项目需求会被反复和逐渐的确定、证明、评估和改进。功能性的需求使用用例的术语来描述。软件系统中非常重要的非功能性的需求,也应该被确定和管理。这些需求不是通过用例来描述的,但他们通常对系统有很大影响,包括系统可用性、可靠性和可支持性等。

用瀑布式开发策略时,项目周期是顺序的,它将项目中潜在问题的发现时间推迟到项目完成和测试之时。所有隐藏在需求、设计和编写代码过程中的问题在项目结束时突然显现出来。而迭代开发允许项目小步进行或增长,逐渐完成整个系统。

4 结束语

软件自动化测试效率高、覆盖率高,但自动化测试过程中存在的问题制约了其发展。结合RUP 软件工程方法论思想,解决自动化测试过程中的主要问题,无疑给自动化测试的成功提供了一个重要的途径,但是软件自动化测试取代不了手工测试,手工测试和自动化测试将相辅相成,共同促进软件测试业的发展。

参考文献:

- [1] Philippe Kruchten. Tutorial: Introduction to the Rational Unified Process[C]. Proceedings of the 24th international conference on Software engineering. New York, USA: ACM Press, 2002.
- [2] Ivar Jacobson, Grady Booch, James Rumbaugh. 统一软件开发过程[M]. 周伯生,冯学民,樊东平,等. 北京:机械工业出版社,2001.