

# 集群式软件平台敏捷测试<sup>①</sup>

徐淑焕 雷跃明 (重庆大学 软件工程学院 重庆 400044)

**摘要:** 在敏捷开发模式的环境下,对以集群存储业务为导向的集群平台的测试与其他产品的测试有很多不同的地方,需要采用有效的策略和方法。首先介绍了敏捷测试的理论和敏捷测试传统的增量模型,在此基础上结合集群平台敏捷开发中实际的问题提出了适合平台敏捷测试的增量模型和对集群平台用户故事测试时用到的仿真环境的搭建和有效的测试方法。

**关键词:** 敏捷测试;增量模型;集群平台

## Agile Software Test of Cluster Platform

XU Shu-Huan, LEI Yue-Ming

(School of Software Engineering, Chongqing University, Chongqing 400044, China)

**Abstract:** In agile development, there are differences between the test of cluster platform supporting cluster storage and other tests. It needs to find some effective methods and strategies. This paper introduces the speculative knowledge of agile software test and the traditional incremental testing model. Based on the traditional model, the paper presents a new incremental model which adapts to the agile development of cluster platform, the construction of the test environment and the effective test methods.

**Keywords:** agile testing; incremental model; cluster platform

随着信息技术的迅猛发展,对于互联网、流媒体、高性能计算等数据密集型、高并发访问的行业用户来说,容量与性能的线性扩展是必须的,传统的存储方式已经无法满足需求。在用户的迫切需求下,集群存储便应运而生。本文的集群平台就是以集群存储业务为导向,快速构建存储应用软件的平台。它解决了上层管理软件的业务描述与操作系统、软件基础架构平台间的交互管理问题,同时屏蔽了技术细节,使存储业务开发人员能够集中全力关注产品研发中的业务与管理问题,摆脱技术细节困扰,提高产品研发效率。

针对平台软件客户需求不稳定不断变更的实际特点,开发模式采用敏捷方式。用户故事(story)是敏捷开发过程中能独立交付的客户可以感知的最小需求,它描述了对软件客户有价值的功能。在敏捷开发环境中对集群平台软件的测试与产品的测试有很多不同的地方:第一,平台软件开发的产品是为业务服务的,无法独立在真实设备上被测试,需要搭建模拟的集群场

景,而模拟场景与真实环境存在一定的差异;第二,最初迭代中用户故事之间的依赖关系还没有建立,所以对各个用户故事的测试就需要用与开发同样的语言编写测试代码,写大量的桩函数来模拟运行环境;第三,敏捷开发是拥抱变化的过程,开发方案的变动必然会导致测试方案设计和执行的改变。第四,在多节点上进行测试执行,纯手工的测试方法实现比较困难等。针对这些出现的问题,在对集群平台采用敏捷测试前期测试人员需要综合考虑采用合适的敏捷测试模型和有效地测试方法,以尽量提高测试效率和质量。

## 1 敏捷测试理论

### 1.1 敏捷测试实质

敏捷开发的最大特点是高度迭代,有周期性,并且能够及时、持续的响应客户的频繁反馈。敏捷测试即是不断修正质量指标,正确建立测试策略,确认客户的有效需求得以圆满实现和确保整个生产的过程安

① 收稿时间:2009-08-31;收到修改稿时间:2009-10-26

全的、及时的发布最终产品。区别于传统瀑布型模式，在敏捷环境中，从确认客户需求开始，到测试计划、测试用例、测试执行、缺陷跟踪、回归测试，一直到最后软件系统发布，测试贯穿整个流程<sup>[1]</sup>。这样能够让测试人员从客户的角度尽早的使用软件，及早发现与需求相悖的问题，尽量减小因设计实现所带来的缺陷问题所导致的维护成本。

敏捷测试也需要高度迭代工作、频繁得到客户的反馈，需要动态调整测试计划、测试的执行。面对采用诸多开发最佳实践的开发者以及快速迭代出新功能的软件系统，测试人员要保持快速的响应，并实时地调整测试过程。这对测试人员是个考验，即使是采用了相同敏捷实践的开发者也不会表现出一样的生产力，只有测试工程师根据整个团队的特点，总结出一套最适合团队的测试模式，才是最理想的<sup>[2]</sup>。

## 1.2 传统敏捷测试增量测试模型

测试是敏捷开发过程重要的环节，自始至终测试贯穿于每个迭代。敏捷开发生命周期一般可以分为4个阶段<sup>[3]</sup>，即初始阶段，项目的建设阶段，产品发布阶段和产品的维护阶段，在关键的项目建设阶段中，测试可以被分成两个部分，确认测试(confirmative testing)和系统综合测试(investigative testing)。下图是传统的敏捷测试增量模型。

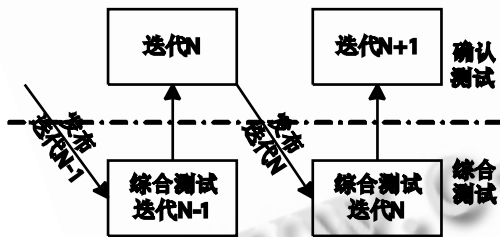


图1 传统敏捷测试增量模型

确认测试就是对版本的有效性和关键的功能是否正确进行验证，测试人员依据测试用例和测试脚本的完整测试是工作的重心。确认测试包含了开发人员的单元测试和被称之为敏捷接受测试的测试部分，这段时间的测试任务用来保障迭代的必须输出的质量。基本的功能、非功能的任务，凡是在迭代开始时制定的计划中承诺的高优先级需求，都要被充分的测试和验证。

系统综合测试是对确认测试的补充和更广泛的测

试活动，测试团队在完成确认测试后来做这部分测试，它包含功能测试，文档测试和系统测试以及和其他产品、环境之间产生的必然的集成测试，还有探索性测试，测试人员采用多种不同途径去尝试测试产品，使用各种手段来考验版本和产品的稳定和正确性等。

## 2 集群平台敏捷测试模型和测试流程

### 2.1 适合集群平台的敏捷增量测试模型

针对敏捷项目开发实际情况，测试流程在传统的增量模型上进行了适当改进，将确认测试和系统综合测试两部分都放在当前迭代的计划内完成。因为敏捷测试团队针对当前迭代的任务计划应服务于当前的产品，过去的迭代产物，或者因为需求变更不再适用，又或者因为未解决的质量缺陷使得实际测试效果不佳。同时团队的目标是能够高效的完成当前迭代的任务，完成更高优先级的工作，每个迭代的考核也比较清晰。为了完成服从当前的高优先级任务和计划，测试人员对以往版本的测试都归入以前迭代的任务而不是当前迭代计划。如果测试以往的产品而不是最新的，敏捷测试的管理会变得有些困难，同时测试团队关注的问题也只能是过时的，低优先级的问题。因此，本项目测试团队使用改进后的敏捷增量测试模型，即在当前迭代仅仅完成当前迭代的计划，所有测试都围绕最新的产品版本进行。

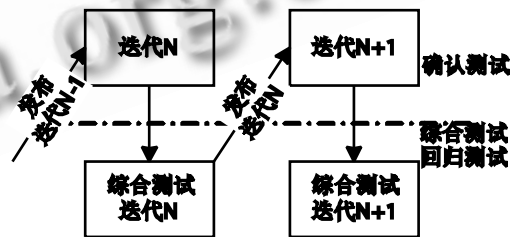


图2 改进的敏捷测试增量模型

在这种增量测试模型中确认测试包含对需求的静态测试，开发人员做的单元测试，以及版本验证测试，功能测试(仅限于对当前迭代功能)和重要的其他类型测试。通过确认测试的产品版本就可以在迭代结束时发布。系统综合测试是为了产品拥有更好的质量，也为了完成对那些较低优先级的质量验证的需求得以确保成功的实现，有针对性的做的系统测试，压力，并发等测试。另外回归测试也放在这个阶段，使产品不

会出现在代码重构后的质量回归现象，以涵盖对以往关键功能的再度验证。

### 2.2 测试流程

在集群平台敏捷的开发环境下，为了保证当前迭代的质量，采用上述改进的增量测试模型，下图是集群平台实际敏捷开发和测试中将敏捷实践串起来，在一个迭代中一个可独立交付的用户故事的开始和结束流程。

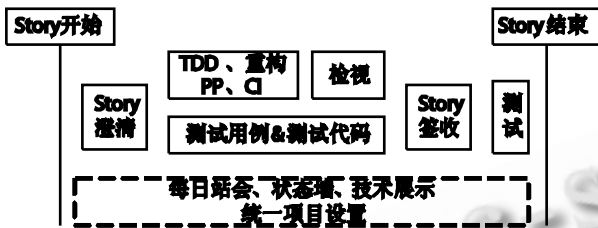


图3 story 测试流程

按照敏捷测试的要求和改进的增量测试模型，从用户故事开发的初期，测试人员就需要参与用户故事方案设计和制定过程，尽量从用户的角度考虑方案的完整性和有效性，提出可测试性需求。用户故事开始开发的前期开发人员和测试人员进行澄清，澄清的目的就是将这一迭代要开发的用户故事的需求和功能明确下来，之后开发人员做相应的开发和单元测试，测试人员在用户故事开发完成之前要做的工作是搭建测试环境、设计测试用例和编写测试代码。测试用例编写完成后要由开发和相关人员评审，以避免漏测和不必要的测试，以便提高测试质量和效率。

用户故事开发完成后转测试，测试人员进行可接受性测试，测试该迭代中产品基本功能和高优先级需求。倘若基本功能和高优先级需求都测试无误，确认测试完成。然后进行系统综合测试，测试人员采取一系列适合敏捷的测试方法来对产品进行测试，考虑方案中异常现象发生的场景，确保较低优先级的质量验证的需求得以成功的实现，有针对性的做系统测试、压力、并发等测试确保版本和产品的正确和稳定性。综合测试完成后，就标志着一个迭代中一个用户故事的完毕。

### 3 集群平台敏捷测试方法

对集群平台用户故事的测试时需要考虑很多问题。集群场景通常采取用虚拟机模拟的方式进行，

但模拟场景与真实场景存在差异。另外最初迭代中用户之间的依赖关系还没有建立，所以对各个用户故事的测试就需要打大量的桩函数来模拟运行环境。随着迭代的不断进行，累加的功能越来越多，受测试资源和场景的限制，不可能一直全面地测试到软件系统的所有方面。此外还要考虑多个节点的相互通信和并发运行的现象，若是纯手工测试，是无法满足需求的，需要使用适合的工具来完成。最后，自动化测试是保证不断迭代后继续保证软件质量的不可缺少的途径，在前期的迭代中，因为需求和方案的不断变更，实现自动化的可能性不大，但在后期迭代中，当客户需求和方案设计稳定后，测试人员就要逐渐实现自动化测试，以方便后面迭代版本的回归测试。

#### 3.1 测试模拟环境的搭建

对集群平台的测试需要多节点的环境，真实的场景通常是无法实现的。为了模拟这种环境，且资源有限情况下，采用的方法是用四台 linux 虚拟机来模拟四个节点的集群场景，组网图如下：

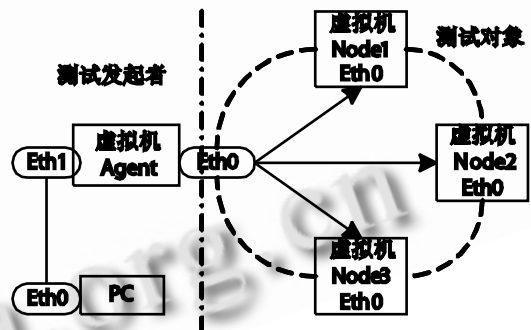


图4 测试组网图

如图4，测试对象节点属于一个专有网络，节点间互通，测试发起者环境和测试对象环境中节点属于不同网络，只能通过一个代理虚拟机通信。

代理虚拟机是指个人PC的一台虚拟机，它有两个网卡，一个网卡eth0和其余三个测试节点属于同一网段，模拟集群的四个节点的环境。另一个网卡eth1和个人PC网卡属于另外一个网段，这样避免不同测试环境的节点混乱。因为测试人员的个人PC属于同一网络，如果个人PC和测试对象节点在同一网络，那么不同测试人员的测试对象节点间就可以通信，会产生混乱。用这种组网方式，不同测试人员可以将个人PC连接到特有的测试节点上，由代理虚拟机登

陆到测试对象环境的各个节点上,进行对节点的配置和测试执行操作。

### 3.2 测试用例的设计和维护

在多节点的环境下,对测试用例的设计用到 Pair-wise 的工程方法<sup>[4]</sup>。因为多节点测试涉及到的场景比较复杂,测试时需要考虑用户故事涉及到的各个功能点和具体执行代码产生的行为,如果选择测试用例完全覆盖,测试用例的数量将会很多。选用 Pair-wise 方法可以将要覆盖的每个参数的每个值只与另两个参数至少配对一次就够了,这样大大减少了用例的冗余,同时覆盖率并没有受到影响。采用 Pair-wise 工程方法就可以实现用尽量少的用例对用户故事进行较全面的测试。

同时在执行测试阶段中,测试人员需要对已有的测试用例进行及时的维护。通常以下两种情况需要新增一些测试用例:一是对于当初测试设计不周全的领域,在集群平台敏捷测试中,前期迭代各个用户故事还不完善,之间的依赖关系是通过打桩来实现的,但是后面的迭代中,用户故事间相互依赖关系实现,彼此交互运行时就发生了很多的异常问题,这些异常是之前测试人员和开发人员都没有考虑到的,出现问题时就要及时补充完善用例,提高用例的质量和覆盖面;二是当产品的功能设计出现更改时(敏捷项目中方案设计的更改是频繁的),所设计的测试用例也要相应地修改,使测试用例保持和现有的功能需求同步。

### 3.3 测试代码的编写和加载

伴随着频繁的变化,大量的测试代码会给适应变化时代码的修改和维护带来巨大麻烦,为了适应敏捷的特点,在编写代码时,我们采用逻辑和数据分离的方法,用命令行传参数,尽量用较少的代码来完成尽量多的用例覆盖,这样就减少了代码的数量,方便后期的回归测试和代码的维护。

在对平台用户故事的测试中,以前常采用一个用例对应一份测试代码的方法,经过多次实践的证明,这样为后期代码的改变和维护埋下了巨大的隐患。采用命令行传参数的方法就可以减少很多的代码编写。比如对集群平台全局锁的测试,开发人员提供模块注册全局锁功能的接口为: `int request(int lockType)`,在测试用例中会有正常和异常的情况,代码的逻辑是相似的,所以只要写一份代码,然后在加载和执行测试程序时,输入相应的参数即可。在用户态下,如果

测试人员的测试执行程序是 `requestlock`,程序中可以设置命令行参数锁类型和执行结果,在某一节点执行测试代码时,输入命令行:

```
./requestlock 3 s
```

传入的参数分别表示该用例是申请 3 锁,且申请该是成功的。执行异常用例时,只要在执行脚本中输入:

```
./requestlock 7777777 f
```

即为申请不存在的锁,且申请该是失败的。这样就将两个用例的代码用一份代码来完成,减少了代码的编写和维护工作。

另外在代码编写时,因为需要模拟测试的场景,就用到其他模块来搭建场景,开发的代码是测试人员常常首先要借鉴的,但是因为敏捷开发是拥抱变化的过程,用户故事也会随需求的改变而改变方案,这样对应的测试代码也要做相应改变,如果没有相应的措施,测试人员的测试代码也就要做大量的整改。针对这一问题,在测试代码中采用对开发代码进行适配的方法,即封装测试对象的外部接口。在测试对象外部接口变动不是非常巨大的情况下,只要改变这个接口的封装代码即可,不需要进行大量的改动,减少了测试人员的工作量,且适应敏捷频繁改变的特性,为改变做好各种准备。

在 linux 环境下测试代码完成后编译成测试模块,然后将测试模块动态加载到被测系统,使测试代码与被测系统原有代码融合(下图中虚线表示)。测试完成后,将测试模块从被测系统中卸载,还原被测系统。如下图:

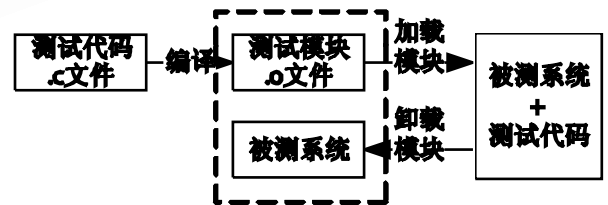


图 5 测试代码加载图

### 3.4 测试自动化的实现

自动化测试在敏捷项目中是重要的内容,每个迭代都会有一些新的功能被开发出来,累加的功能越来越多,测试资源有限,不可能一直全面地测试到软件系统的所有方面,自动化测试是保证不断迭代后继续

保持软件质量不可缺少的途径。在敏捷项目的早期阶段因为进度和方案的变更开展自动化测试通常是很困难的,但是到后期迭代时,前期的用户故事已经稳定下来,测试人员就可以在前期手动的基础上实现自动化测试。

在对集群平台的测试中,需要在多个节点上进行操作,经常会需要模拟时间间隔很小的并发性操作,纯手工测试实现是比较麻烦的,所以在测试中采用了可以同时登陆多台 linux 虚拟机并发送命令的测试工具,该工具可以通过工程形式管理 TCL 脚本,创建测试序列,定义测试套参数,执行脚本,并自动将测试结果填写到对应的测试用例中去。测试自动化<sup>[5]</sup>的脚本由 TCL 调用 shell 命令来完成,在测试代码和 shell 脚本编写完成后,就可以用 TCL 脚本调用测试代码模拟各个测试用例的测试环境和执行步骤,并判断最后结果是否为预期结果,实现整个测试过程的自动化。

自动化测试为迭代的回归测试提供服务,提高了回归测试的效率和质量,节约了大量的时间。在增量模型中,它归于系统综合测试阶段,能够有效保证迭代版本的质量和稳定性。

## 4 结语

本文提出的增量敏捷测试模型和具体测试方法经过实践证明,能对集群平台敏捷开发的用户故事进行有效测试,与传统的测试模型和方法相比,可以将 bug 尽早的发现并解决,为保证产品的质量做下更好的准备。本文提出的敏捷测试模型和方法可以为进行集群平台敏捷测试的测试人员提供极好的指导意义和参考价值。

### 参考文献

- 1 Elisabeth Hendrickson. Agility for Testers. Pacific Northwest Software Quality Conference, 2004,10(1): 38-41.
- 2 叶俊民.软件工程.北京:清华大学出版社,2006.
- 3 敏捷测试的最佳实践.软件测试[2008-7-04]. <http://www.51testing.com>.
- 4 Harrell JM.李正新译,正交表测试策略(OATS)技术, [bbs.51testing.com](http://bbs.51testing.com)
- 5 Dustin E, Rashka J, Paul J.自动化软件测试—入门管理与实现.北京:清华大学出版社,2002. <http://www.c-s-a.org.cn>