

---

# 目 录

---

对敏捷开发的一些思考.....	01
浅谈探索性测试中用到的一些方法.....	06
云计算知识介绍.....	16
《赢在测试 2-中国软件测试专家访谈录》读书笔记.....	39
《云服务测试-如何高效地进行云计算测试》读书笔记.....	56

# 对敏捷开发的一些思考

◆ 作者：顾翔

## 一、简介

敏捷软件开发 ( Agile software development ) , 又称敏捷开发 , 是一种从上世纪九十年代开始逐渐引起广泛关注的一些新型软件开发方法 , 它是应对快速变化的需求而产生的。它的具体名称、理念、过程、术语都不尽相同 , 相对于“非敏捷” , 共同点是更强调程序员团队与业务专家之间的紧密协作、面对面的沟通 ( 认为比书面的文档沟通更有效 ) 、频繁交付新的软件版本、紧凑而自我组织型的团队等等 , 它能够很好地适应需求变化的代码编写和团队组织 , 更注重软件开发中人的作用。“敏捷” ( Agile ) 一词来源于 2001 年初美国犹他州雪鸟滑雪圣地的一次敏捷方法发起者和实践者 ( 他们发起组成了敏捷联盟 ) 的聚会。敏捷开发是软件工程经过原始模型 , 大棒模型 , 瀑布模型 , 迭代模型后产生的。

敏捷组织有自己宣言 :

- 人和 ( 人与人的 ) 交互 : 优先于过程和工具。
- 可以工作的软件 : 优先于求全责备的文档。
- 客户协作 : 优先于合同谈判。
- 随时应对变化 : 优先于循规蹈矩。

## 二、敏捷开发的优点

我认为敏捷开发有以下优点：

1，采用敏捷，可以快速提高软件发布周期。许多软件公司以前采用瀑布模型：从业务建模，需求调研，需求分析，设计，编码，测试到最终送交给客户使用，需要经历很长的周期。而采用敏捷开发，可将一个产品或项目分解为多个 sprint，开发小组每周或每两周提交一部分产品，而这部分产品都要保证可运行，可使用的。经过多模块集成及集成测试，客户每一到两个月就可拿到产品，这样可保证客户尽早发现产品问题所在，及时反馈回来。此外客户对产品的重要部分可优先使用起来。敏捷开发提倡客户参与到项目中去，尽管实施起来比较困难，但若能够真正做到这一点，这样就可更加体现这个优点。另外若再采用火车模型<sup>[1]</sup>进行产品发布，则更可缩短产品发布的周期。

2，采用敏捷开发，测试能够尽早参与进项目中来。缺陷预防优于缺陷发现，缺陷发现是亡羊补牢，虽说为时未晚，但是软件测试能够尽早参与，可把众多的缺陷消灭在萌芽状态，这一直是软件测试人员梦寐以求的事情，就像中医所说的：上工治未病，让疾病隐患消灭在萌芽状态，不让它发出来。采用敏捷开发，测试人员可以在第一时间内参与产品在这个 sprint 的需求，由于每一个 sprint 的功能被细化得比较简单，测试与开发可比较清楚地了解这个 sprint 的需求，并与开发人员达成一致的观点。在开发人员写功能代码的时候，测试人员就可以设计测试案例。当开发代码写完，测试案例设计也完成以后就可执行测试案例了，与此同时可以进行探索性测试(ET)<sup>[2]</sup>，及早发现问题并反馈给开发人员，以便及时解决，及时总结。对于一些问题可避免在下一个 sprint 中再犯;而对于系统架构体系中的问题，可以在第一时间内进行优化，重构或重建，缩短了开发时间，从而达到了敏捷开发的目的。

3，采用敏捷开发，可以减少许多不必要的文档。记得我当时作 QA 工作的时候，领

导要求我和我的同事撰写《公司产品开发测试流程规范》，这要涉及到流程，干系人与文档这三个部分。在文档部分就要从业务建模，需求调研，需求分析，概要设计，详细设计，单元测试，集成测试，系统测试，验收测试到软件部署，共定义了五十几个模版。但在实施过程很困难，大部分文档开发，测试人员写得很草率甚至不写，问及原因，无非是时间紧或者是没必要，他们会说：“设计和开发都是我个人一个人，一切都在我脑子里”。敏捷宣言中说：“可以工作的软件：优先于求全责备的文档”。敏捷小组中的员工可以根据产品或项目的具体情况来决定应该写哪些文档，不应该写哪些文档，灵活决定。但并不是说敏捷开发不需要写文档，否则员工休假甚至离职，接手的员工如何工作？

4，采用敏捷开发，结对编程与结对测试是有利于提高产品质量并且有利于培养新员工很好方式。结对编程与结对测试，一个人工作，另一个随时检查，及时沟通，遇到问题立即解决。

5，采用敏捷开发，日立会（Standup meeting）是一种很好的沟通形式。日立会既每天早上开发小组内的所有同事在一起，每人各自介绍昨天做了哪些工作，今天计划做哪些工作以及工作中所遇到的各种问题。日立会是敏捷开发中的一项重要内容，它是一项很好的沟通活动。通过日立会，开发小组内的同事可以及时了解小组的工作情况。对于遇到的问题，若小组内其他同事可以帮助解决，这个问题可交给这位同事处理；若小组内其他成员都无法解决的问题，可由小组负责人上报给上级领导，让上级领导来决定如何处理。这样可以有效提高工作效率。

任何一种方法有优点也会有不足之处,下面来谈一谈敏捷开发的缺点,这是本人的一些体会，不一定正确。

### 三、敏捷开发的缺点

1, 采用敏捷开发, 开发团队的人员素质要求比较高。敏捷开发的首要任务是快速, 目前提出的“全栈软件工程师”<sup>[3]</sup>, 它要求软件开发人员在开发的各方面: 从需求, 设计, 编码, 测试一直到部署都要求是行家里手, 可以减少因彼此沟通带来的时耗, 这样才能保证他在一个 Sprint 中能独立完成产品中某个特定的任务。显然这样的软件开发人员的素质一定要求很高的, 而在软件开发行业中, 人员流动率高, 新手多的情况下, 要做到这一点是比较困难的。

2, 采用敏捷开发, 开发人员与测试人员混为一体, 彼此分工不明晰。敏捷开发要求软件人员会测试, 测试人员会开发, 这在实施起来是比较困难的。这是因为开发和测试人员关注的重点是不同的: 开发关注与技术实现比较多, 一般都采用正向思维; 而测试关注业务比较多, 多采用逆向思维。所以一个产品要保证有高的品质, 就必须要有专门的测试人员, 应该把测试和开发有比较清晰的分工。正如古话所说: “术业有专攻, 问道有先后”。

3, 采用敏捷开发, 是“短平快”的开发方式, 由于产品发布周期短, 所以产品的维护, 升级等操作的频率也增加了。这必然增大开发人员, 测试人员以及运行维护人员的工作压力, 在高压的环境下工作容易出错, 影响产品的质量。

4, 采用敏捷开发, 不利于文档的建立和修改。敏捷开发有一句口号: “拥抱变化”。然而客户需求的变更是经常变化的, 这正如当今社会流行所谓“唯一不变的是变化”。为了缩短版本发布周期, 特别是在版本发布的之前, 当客户的需求发生了变更时, 敏捷开发团队仅是修改代码而没有时间修改所对应的文档, 这就造成产品和文档的开发不一致性。给产品后期优化, 调整或二次开发带来极大的麻烦, 在人员频繁流动中更是灾难性的。

#### 四、总结

软件工程无非好坏，先进与落后。敏捷开发是个新方法，存在优点，也存在缺点，我们不要一味赶时髦，凡事都一窝蜂。我们要根据自己的企业现状和产品特性，选择符合自己的软件工程方法论，只要这个方法可以给企业提高质量，带来效益，就是一个好的方法。敏捷的特点是版本发布速度快，然而中国又有一句古话：“慢工出细活”。活干得快，往往会影响质量，所以我认为对于一些版本发布频率要求不高，或者涉及到严格质量要求的产品，比如航空，航天，金融等领域的产品，不一定采用敏捷开发的方法，可采用更适合于自身产品特性的软件开发方法。我有一位同行，在美国工作，从事金融软件的开发业务，可以想象，这种产品的质量要求是很高的容不得半点差错，所以他们仍旧采用传统的瀑布模型开发方法，他每天从设计师处拿到设计文档，该文档写得很详细，然后按照设计文档进行编码，可以在家里通过互联网工作（公司省去了为员工租用 Office 的开销），公司效益很好，已经维持了近二十年。

[1] 火车模型：软件发布像火车一样，有固定发车时间，新功能是否可发布取决于它是否可赶上这班火车。

[2] 探索式测试（Exploratory Testing）：是一种自由的软件测试风格，强调测试人员展开测试学习、设计测试、测试执行和测试结果评估等活动，以持续优化测试工作。

[3] 全栈工程师：软件从建模，需求，设计，开发，测试到部署，维护都由一个工程师承担。

# 浅谈探索式测试中用到的一些方法

◆ 作者：顾翔

"探索式测试"是测试专家 Cem Kaner 博士在 1983 年提出，并受到语境驱动测试学派 ( Context Driven Testing School ) [1]的支持。随着近年来敏捷开发的出现，探索式测试的理论由于符合快速提交的理念，也被重新提出，并且受到了广泛的重视。其实我们以前在测试中也有意无意地使用到探索式测试中的一些方法，就是没有把它升华到一个理论的高度。探索式测试 ( Exploratory Testing )：是一种自由的软件测试风格，强调测试人员展开测试学习、测试设计、测试执行和测试结果评估等活动，以持续优化测试工作。本文不涉及到探索式测试的详细定义，思想以及一些相关的模型（这些详细定义，思想和模型可以通过互联网或者目前出版的一系列书籍中找到），仅将在多年测试工作中用到的一些探索式测试方法介绍给大家，希望能对从事软件测试的同行有所帮助。

**注：**本文描述的测试产品对象大都数以基于 B/S 结构的产品为主，少部分会考虑其他类型的产品。

**1、对表单输入的测试探索：**输入表单在产品中经常出现，比如我们要成为某个网站的会员，我们就要注册一些个人信息，然后通过表单页面上的[提交]按钮，存储到数据库中去。对表单元素输入的测试探索中我们经常需要考虑两个方面：对超长字符或特殊字符（比如只允许输入数字）输入的探索以及对保留字符的输入探索。

1.1 对超长或不符格式字符输入的测试探索。由于表单输入的数据最终一般都是存

入到数据库中的，所以对于输入的数据一定要对输入字符的长度或者类型进行限制。一个正常的操作方法为：输入一个超长或不符格式的字符串，会有如下三种处理方式：

a) 超长或不符格式的字符在输入界面中被锁定：比如表单要求输入最长长度为 40 个字符串，当输入第 41 个字符串的时候，页面是不允许输入的；再如，表单只允许输入阿拉伯数字，如果你输入了一个字符'A'，页面也是不允许的。

b) 超长或不符格式的字符在输入界面中被输入，但在提交表单的时候，输入界面会提示表单中有超长或不符格式的字符，且表单不允许被提交。

c) 超长或不符格式的字符在输入界面中被输入，但在提交表单后，会在另外的页面中提示表单中有超长字符或不符格式的字符被输入，且表单不允许被提交。

我们如果在表单输入超长或不符格式的字符，提交后界面如果没有任何提示信息，甚至出现系统崩溃，或者出现数据库发生异常的英文日志显示在页面上，那么这显然是一个 bug。另外如果我们输入一个超长或特殊字符，符合上述 3 种处理方法，我们的测试并未结束，这时候你不妨可以问问开发人员对应的代码，查看一下代码，尤其是查看在存储到数据库之前，程序是否对输入字符的长度或类型进行再一次校验，这是从安全性角度的考虑的。举个例子：我们输入页面的文件名为 table.htm,后台存入数据库操作的文件名为 insert.jsp。一个黑客为了破坏这套系统，它用其他软件制造了一个表单页面代替 table.htm。在他的页面中输入超长或者不符格式的数据，然后通过你的 insert.jsp 提交到数据库中，由于 insert.jsp 没对输入字符进行校验，从而造成系统的瘫痪。这种案例在从事网站开发的时候经常遇到。

1.2 对保留字符的输入的测试探索。对于表单输入数据除了存储到数据库中去之外，还会显示在界面上。下面拿查询个人信息作为一个例子，此时界面会把你以前输入到数据库中的数据显示到界面上来，这样便于你对信息中不合适的地方进行修改。我们要对

这类种型的测试进行探索，首先要搞清楚这套产品对表单输入数据的是使用何种方式进行输出的？这种输出方式中存在哪些保留字符？我们比如输出的方法是用 HTML 语言显示的，HTML 保留字符为：<, >, ", ', &, 空格, 回车这七种字符，HTML 语言显示这些字符会用其他另外的字符串进行替换的。一个正常的操作是从数据库中输出保留字符以后，程序对这些保留字符通过正则替换转码为 HTML 中对应的可以显示的字符串，如'<'转为 &lt;，空格转为&nbsp;:...，如果你在表单输入这些字符，甚至可以输入一段含有这些字符的 HTML 代码或者 javascript 代码，如：<a href="http://www.sina.com.cn">进入新浪</a>，然后在显示页面中进行查看，看看输出的格式是否正确。如果输出的格式不正确，或者出现显示页面错乱，甚至执行了 javascript 语句，这肯定是一个 bug。

**2、对于模糊查询输入框输入数据的测试探索。**目前除了一些专业的搜索引擎，比如 Google, 百度以及目前流行的大数据存储方式采用非关系性数据库或者 NO SQL 技术进行存储，对于大多数软件系统通常还是采用传统的关系性数据库进行存储。若一个用户想查询标题中含有'云计算'的标题，再通过点击相应标题查看文章内容。通常所对应的 SQL 查询语句为：select url,title , content from paper where title like '%\$title%' (\$title 为模糊查询文本框提交过来的字符串,在这里为"云计算"),让我们来深入讨论一下 SQL 语句中%这一个关键而特殊的模糊查询字符，假设用户在模糊查询输入框中输入%，而程序在处理的时候没有对%进行特殊处理，上述的 SQL 语句就变为 select url,title from paper where title like '%%%'，这样的查询输出结果不是把文章标题中含有%字符的标题输出，而是把这个数据库表中所有的记录都给输出出来了。如果在测试中发现这样的问题，可以告诉程序员把输入字符中的%，通过正则替换把%转意为/%,就可以了。

**3、对于并发操作的测试探索。**这里举一个例子，比如被测系统是一个需要审核员审核的一个博客系统。注册该网站的会员可以在博客上发布自己的博文，但是发表的博文

必须通过该网站的审核员审核通过后可以正式发布到网上去。

在这个产品中，网站会员具有的权限是：a)书写博文并提交给审核员进行审核；b)对审核员退回的博文进行修改然后重新提交；c)对已提交未审核的博文可以随时进行修改或删除；

审核员具有的权限是 a)审核通过，博文被正式发表在网上 b)审核退回，审核员需要书写退回理由。让我们来设想可能会发生如下这样过程，审核员 A 正在审核一篇博文，而这个时候书写这篇博文的会员 B 觉得这篇博文不太合适，要将它删除了。此时审核员 A 进行审核通过或者审核拒绝的时候会出现什么情况？我们希望出现的情况是：在审核员 A 进行操作以后，系统会出现相应的提示信息"该博文已经被发表者删除，请与作者联系"的提示信息。

然而在测试过程中我们经常会出现数据库发生错误，页面显示系统调用的用户看不懂的英文错误日志信息。让我们再进行一次探索，比如这个时候有 2 位审核员 A 和 C 同时对同一篇博文进行审核，审核员 A 进行了一次审核通过的操作，过了几分钟后，审核员 C 进行了一次审核退回的操作，我们期望审核员 C 看到的也是一条友好性的提示信息："该博文已经被审核员 A 审核通过，请您与审核员 A 联系"，而不是在页面中出现一些用户看不懂的英文错误日志信息或者这篇博文被后操作的审核员 C 退回了。上述例子是审核通过在先，审核退回在后。同样道理，我们可以考虑审核退回在先，审核通过在后的情形。

**4、对于页面刷新功能的测试探索。**几乎所有的人机交互页面都具有刷新功能，刷新功能的目的在于当开发人员把页面作了变化后，用户通过按刷新按钮，就可以及时地看到最新页面的内容。但是由于刷新功能的存在，往往给产品带来一定的隐患，暗藏着一些 bug，下面是可能遇到的两种情形：

4.1，系统进入表单填写页面，填写内容，提交表单，页面出现"表单数据提交成功"的提示信息。就在这个提交成功页面按下刷新按钮，页面会出现用英文提示数据库错误日志，大致内容是表单中的某某字段不允许相同。显然，在刷新的时候，系统把表单数据又重新的进行了一次提交操作。正确处理办法是：刷新后，不再重新提交数据库，仍旧显示表单数据提交成功的提示信息，或者告诉用户不允许同时提交相同的信息。

4.2，查询到一条记录，点删除按钮，页面出现对应"记录删除成功"的提示信息。就在这个删除成功提示页面按下刷新按钮，页面用英文提示数据库或程序中出现了空指针异常的错误日志，显然，在刷新的时候，程序对同一条记录又进行了一次删除操作。正确的办法是：不再重新删除记录，仍旧显示表单数据删除成功的提示信息；或者告诉用户修改的记录不存在，无法删除。

**5、对于非常用功能的测试探索。**这里我们还是以表单提交页面做个例子。在一个表单提交页面中往往会出现 2 到 3 个按钮 ( button )，一个是[提交] ( Submit ) 按钮，一个是[重写] ( Reset ) 按钮 ( 这个按钮往往在有些页面被删除 )，还有一个是[取消] ( Cancel ) 按钮。对于[提交] ( Submit ) 按钮，几乎每一个测试人员都不会遗忘，而对于[重写] ( Reset ) 和[取消] ( Cancel ) 按钮，几乎无人问津，而在操作[重写] ( Reset ) 或者[取消] ( Cancel ) 按钮操作以后，往往会发现一些隐藏的 bug。

**6、对于突发事故的测试探索。**比如一个程序正在进行工作：或许是在备份数据；或许通信系统正在进行通话；或者一个用户正在浏览一个网站...，就在这个时刻，事故发生，比如事故为发生了停电现象 ( 在测试的时候你可以拔掉电源 )，事故被解除后 ( 在测试的时候你可以再次插上电源 )，系统被重新运行，这个时刻你要仔细检查一下整个系统是否仍旧可以启动？各个功能是否可以正常运行？是不是存在着数据丢失现象？...

**7、对于界面链接的测试探索。**一个复杂的系统往往存在数十个甚至上百个页面，这

些页面存在着互相链接依赖的关系，我们在浏览一些网站，甚至浏览某些著名的网站的时候，通常会点击一个链接，系统会告诉你这个页面不存在，这就是由于界面链接测试没有做好，为了避免这种情形的出现，可以用下面做法：是把所有的页面画在一张纸上，如果 A 页面有链接可以链接到 B 页面中，就从 A 画一条直线链接到 B，在直线 B 处画一个箭头；同样如果 B 页面有链接可以反向链接到 A 页面中，就从 B 再画一条直线平行于上述直线链接到 A，在直线 A 处画一个箭头。全部直线，箭头都画好后，选择一个起点页面，一个终点页面，然后设计一条路径，在软件上一步一步地按照计划好的路径进行操作。经过几个这样的路径操作后，图中所有的路径都会被覆盖，测试结束。虽然目前市场上存在检查 WEB 页面是否存在空链接的工具，但是采用这种方法，在操作过程中往往会发现这样或那样的其他 bug。

**8、对于需要多步操作来完成一个事物的测试探索。**有些操作往往需要进行多步操作才可以完成，比如提交一份你的求职简历，第一页填写你的基本信息，第二页填写你的教育经历，第三页填写你的工作经历，最后一页填写你的其他信息。在测试这样的软件产品的时候你可以尝试提交第一，二，三步以后回退到第一步，重新输入，软件会出现什么情况？再尝试提交第一，二步后放弃提交，退出，然后再一次进入，输入和上次相同的信息，软件会出现什么情况？

**9、对于老功能的测试探索。**对于一些产品中一些老的功能，我们经常很容易忽视，有些老测试工程师会说，这些功能我们那个时候测试了不下几十次了，肯定没有问题，这样造成的结果是这些功能模块几个月甚至几年都不去测试一下。可以设想，当其他功能的接口，框架，结构...都在进行调整，变化，包括数据库的结构也可能作过相应的调整。这些调整或许会对老功能造成一定程度的影响，所以我们需要不时地对这些老功能进行测试，有时会发现一些意想不到的问题。

**10、对于重灾区的测试探索。**我们经常会听到这么一句话：80%的 Bug 出现在 20% 的功能点上，所以你在测试的时候，对于经常出现 Bug 的功能点，你一定要小心，小心再小心；认真，认真，再认真地进行测试。在这里你往往你会找到更多的 bug。

**11、对于强迫症测试法的测试探索。**强迫症测试法是 James A. Whittaker 在他的著作中《探索式软件测试》中描述的，采取强迫症测试法往往会发现程序中出现的系统内存泄露等问题。记得几年前我在测试一个产品角色功能模块的时候，需要不断的登出，登入系统，后来发现经过多次的登出，登入操作后系统就会出现死机的现象，最后经过程序员排查，发现是程序中一个指针用完以后没有及时释放造成的。由于我们不可能对所有模块都进行强迫症测试法，所以在哪里进行强迫症测试法？这需要凭测试人员的经验，也要看运气，具有一定的偶然性。当然我们还可以借助某些工具来进行强迫症测试法，这样可以节省很多精力。

**12、对于产品广告中提及功能的测试探索。**产品广告往往是由市场人员作为产品推销而书写的，对于广告中提及的功能，我们要与市场和销售人员进行及时沟通，比如这条语句在那个模块的哪个功能点实现的，然后在产品上具体操作一下，看看是否是那么回事情。广告具有一定的夸大性，这是在所难免的，但是对于夸大成份过大的部分，测试人员有提出修改建议的义务。

**13、对于产品说明书的测试探索。**产品说明书对于用户，特别是一些新用户是了解产品的一个强为有力的工具，所以作为测试人员应该认真对待，我们需要对用户手册中每一条功能进行严格核实。除此之外，我们应该站在用户角度去思考，考虑有些注意事项是否需要告诉用户，用户手册是否便于用户阅读，用户手册的书写逻辑是否合理以及手册中章节的前后顺序是否需要进行调整，以使用户能够更好的熟悉产品...

**14、对于领域专家提及功能的测试探索。**比如我们正在做一个 4G 手机产品软件的

时候，听到一个新闻，国际上某某通信专家提出以后 4G 手机产品必须具有 X 功能。我们要和需求人员，公司领导一起讨论确定这条新闻是否属实，这位专家是否在业内影响力有多大等信息。然后再查看我们的产品是否具有 X 功能？如果没有，规划立刻把 X 功能安排进开发计划中；否则，我们是否需要重新审视一下 X 功能，看看 X 功能是否需要做优化。

**15、对于用户并发性的测试探索。**在互联网的时代，用户并发性是个常见的现象。这在制订需求的时候我们就应该定义好同时并发的最大数，然后测试人员根据产品的具体情况，选择合适的工具或者自己开发相应的工具，以便于到产品完成后按照性能指标要求用工具进行并发量测试。

**16、对于稳定性的测试探索。**稳定性也是用户关心的一项性能指标，如同用户并发性一样，我们同样需要结合自身产品特性，选择合适的工具或者自己开发相应的工具，以便于到产品完成后按照性能指标要求用工具进行稳定性测试。

**17、对于用户友好性的测试探索。**什么叫用户友好的，什么又叫用户不友好的，这是个仁者见仁，智者见智的问题，没有统一的规范。这项测试在许多企业很难展开。我采取的方法是如果没有客户，让一个模块由多个人进行测试，大家都认为不友好，那么作为不友好 bug 提出。若有客户，则优先考虑客户的意见。另外我们认为相同模块处理结果的一致性为用户友好性的一项衡量指标，比如提交客户信息后出现的是 A 风格的界面，提交产品信息后出现的是 B 风格的界面，这样的用户友好性肯定差。

**18、对于兼容性的测试探索。**兼容性测试分为产品自身前后版本兼容性，产品在各个显示器或浏览器上显示兼容性，对运行的操作系统兼容性以及与其他产品的兼容性。关于这些兼容性的细节我在这里就不再提及的，大家到互联网上可以很清楚地看到其定义。

**19、对于升级 ( upgrade ) 的测试探索。** 一个产品不可能一次就能开发得满足用户的需求，肯定需要经过多个版本，尤其是敏捷开发概念提出以后，版本发布越来越频繁。现在产品大都支持在老产品的基础上，不删除产品，直接运行升级脚本，从而完成升级目的。对于一些实时性要求很高的产品，比如通信产品，在升级的过程中还需要考虑不影响通信业务的正常运行。另外测试完毕升级操作后，我们都要将产品进行还原(Restore)操作，还原到升级之前的版本，以保证维护人员在客户处万一升级失败，用户仍旧可以在老版本上继续进行使用。

**20、总结：**探索式测试强调在一个测程内进行测试设计，测试执行，然后对测试结果进行分析总结，从而进一步对业务知识，测试技巧，测试工具...进行学习。重新调整测试策略，然后进入下一个测程。所以及时地总结测试方法在探索式测试中是非常重要的，大家应该把测试方法放在一个大家都可以看到的地方，比如 Wiki，便于大家都可以及时进行查看和学习。测试人员学习测试方法可以提高自己的测试技能；开发人员学习测试方法可以在开发的时候尽量避免错误的发生，从而起到缺陷预防的效果。

# 云计算知识介绍

作者：顾翔

云计算这几年已经席卷了各个角落，在 IT 行业内兴起了一股新的潮流，它必将在这几年内统领 IT 行业的发展方向。对于广大从事与 IT 行业相关的工作的人士，了解云计算的历史沿革和发展现状是十分必要的，本文首先介绍云计算的历史，云计算建立基础，云计算定义，由 NIST 规定的关于云计算的五个基本特征、三个服务模型和四个发布模型；然后分析了云计算的优缺点；接下来我们介绍了云计算的两个关键要素：云服务器和云存储，并且介绍了当今比较流行的几个云平台。最后简要介绍了一下云计算与大数据之间的关系。

## 1、云计算历史

**首先让我们来回顾一下云计算的历史：**

最早提出云计算概念的人是斯坦福大学的科学家 John McCarthy 在 1960 年提出"计算机可能变成一种公共资源"；

1966 年，著名的 ARPAN(Advanced Research Projects Agency Network)的负责人 J.C.R.Licklider 提出了"从任意点通过网络访问计算机程序"的设想；

1983 年，太阳电脑（Sun Microsystems）提出"网络是电脑"（"The Network is the Computer"）；

1999 年 Salesforce.com 是现在公认的最早的云计算公司，提供基于云的 SaaS 服务；

2006 年 3 月，亚马逊（Amazon）推出弹性计算云（Elastic Compute Cloud；EC2）服务；

2006年8月9日，Google 首席执行官埃里克·施密特（Eric Schmidt）在搜索引擎大会（SES San Jose 2006）首次提出“云计算”（Cloud Computing）的概念。Google“云端计算”源于 Google 工程师克里斯托弗·比希利亚所做的“Google 101”项目；

2008年2月1日，IBM（NYSE: IBM）宣布将在中国无锡太湖新城科教产业园为中国的软件公司建立全球第一个云计算中心（Cloud Computing Center）；

2008年7月29日，雅虎、惠普和英特尔宣布一项涵盖美国、德国和新加坡的联合研究计划，推出云计算研究测试床，推进云计算。该计划要与合作伙伴创建6个数据中心作为研究试验平台，每个数据中心配置1400个至4000个处理器。这些合作伙伴包括新加坡资讯通信发展管理局、德国卡尔斯鲁厄大学 Steinbuch 计算中心、美国伊利诺伊大学香槟分校、英特尔研究院、惠普实验室和雅虎；

2008年8月3日，美国专利商标局网站信息显示，戴尔正在申请“云计算”（Cloud Computing）商标，此举旨在加强对这一未来可能重塑技术架构的术语的控制权；

2010年7月，美国国家航空航天局和包括 Rackspace、AMD、Intel、戴尔等支持厂商共同宣布“OpenStack”开放源代码计划，微软在2010年10月表示支持 OpenStack 与 Windows Server 2008 R2 的集成；而 Ubuntu 已把 OpenStack 加至 11.04 版本中；

2011年2月，思科系统正式加入 OpenStack，重点研制 OpenStack 的网络服务。

## 2、云计算基础知识

这一届将介绍一下建立云计算所用到的一些基础知识：

### 2.1 互联网技术

云计算是互联网技术的一个延伸，云计算背后的互联网技术是基于一系列的标准和协议的，它使得客户可以在任何地方访问任何设备中的数据。

## 2.2 虚拟化技术

虚拟化是一个广义的术语，在计算机方面通常是指计算元件在虚拟的基础上而不是真实的基础上运行。虚拟化技术可以扩大硬件的容量，简化软件的重新配置过程。CPU 的虚拟化技术可以用单个 CPU 去模拟多个 CPU 并行，允许一个平台同时运行多个操作系统，并且应用程序都可以在相互独立的空间内运行而互不影响，从而显著提高计算机的工作效率。

虚拟化技术可以在云服务器上创建更多的逻辑硬盘，内存等虚拟硬件设备以及操作系统，中间件等虚拟软件设备。关于更多的虚拟化技术的知识请参看参考文献<sup>[1]</sup>。

## 2.3 面向服务的体系架构 ( SOA )

面向服务的体系结构，是一个组件模型，它将应用程序的不同功能单元(称为服务)通过这些服务之间定义良好的接口和契约联系起来。接口是采用中立的方式进行定义的，它应该独立于实现服务的硬件平台、操作系统和编程语言。这使得构建在各种这样的系统中的服务可以以一种统一和通用的方式进行交互。

SOA 基于标准格式与标准协议提供服务，为云计算中各个服务提供统一的接口做出了很大的贡献。关于更多的 SOA 的知识请参看参考文献<sup>[2]</sup>。

## 2.4 网格计算

网格计算即分布式计算，是一门计算机科学。它研究如何把一个需要非常巨大的计算能力才能解决的问题分成许多小的部分，然后把这些部分分配给许多计算机进行处理，最后把这些计算结果综合起来得到最终结果。

显而易见，通过网格计算可以通过一系列小功能的机器实现大规模的数据运算提供了可能，网格计算在云计算中起到了非凡的作用。关于更多的网格计算的知识请参看参

考文献<sup>[3]</sup>

## 2.5 按量使用付费模型

云计算是通过使用互联网流量以及存储器使用空间进行收费的。典型的按量使用付费模型是我们日常生活中使用的水，电，气。

## 3、云计算定义

那么到底什么是云计算呢？由于云计算的概念太抽象了，就像当年如何定义"无线局域网"或"802.1X"一样，业界有各种各样的定义标准，并且每个标准都有一定的意义。到目前为止关于云计算的定义就超过了 100 中。在这里，我们在这里给出 CSA ( Cloud Security Alliance ) 云计算安全联盟在 Security Guidance For Critical Area of Focus In Cloud Computing V3.0 的定义：

"云计算的本质是一种服务提供模型，通过这种模型可以随时，随地，按需地通过网络访问共享资源池的资源，这个资源池的内容包括计算资源，网络资源，存储资源等，这些资源能够被动态地分配和调整，在不同用户之间灵活划分。凡是符合这些特征的 IT 服务都可以被称作云计算服务。"

## 4、云计算的标准

NIST ( U.S. National Institute of Standards and Technology ) 美国国家标准与技术学院给出了一个云计算的标准---"NIST Working Definition of Cloud Computing/NIST 800-145"，它由五个基本特征、三个服务模型和四个发布模型组成。

### 4.1 五个基本特性

#### 4.1.1 按需自助服务

视客户需要，可以从每个服务提供商那里单方面地向客户提供计算能力，比如，服务器时间和网络存储，而这些是自动进行无需干涉的。自助服务是区分 B/S 架构与云计算的重要的标准。

#### 4.1.2 广泛的网络访问

具有通过规范机制网络访问的能力，这种机制可以使用各种各样的客户端平台（例如，PC、笔记本电脑以及 PDA）。比如您是位作家，即使您手头的机器上没有安装 MS Word 等类似编辑文件功能的软件，或者使用的是 PAD 或者智能手机，您也可以随时随地的可以登录到 Google Doc 中去进行您的写作工作。

#### 4.1.3 资源共享

提供商提供的计算资源被集中起来通过一个多客户共享模型来为多个客户提供服务，并根据客户的需求，动态地分配或再分配不同的物理和虚拟资源。资源包括以下类型：存储，计算能力，内存，网络带宽和虚拟环境等。

#### 4.1.4 快速的可伸缩性

具有快速的可伸缩性地提供服务的能力。在一些场景中，所提供的服务可以自动地，快速地横向扩展，在某种条件下迅速释放、以及快速横向收缩。对于客户来讲，这种能力用于使所提供的服务看起来好像是无限的，并且可以在任何时间、购买任何数量。

#### 4.1.5 可度量的服务

云系统通过一种可计量的能力杠杆在某些抽象层上自动地控制并优化资源以达到某种服务类型（例如，存储、处理、带宽以及活动用户账号）。资源的使用可以被监视和控制，通过向供应商和用户这些被使用服务报告以达到透明化。并且以此作为收费的基础依据

## 4.2 服务模型

### 4.2.1 软件即服务 (SaaS)

客户所使用的服务商提供的这些应用程序运行在云基础设施上。这些应用程序可以通过各种各样的客户端设备所访问,通过瘦客户端界面像 WEB 浏览器(例如,基于 WEB 的电子邮件)。客户不管理或者控制底层的云基础架构,包括网络、服务器、操作系统、存储设备,甚至独立的应用程序机能,在可能异常的情况下,限制用户可配置的应用程序设置。

### 4.2.2 平台即服务 (PaaS)

客户使用云供应商支持的开发语言和工具,开发出应用程序,发布到云基础架构上。客户不管理或者控制底层的云基础架构,包括网络、服务器、操作系统或者存储设备,但是能控制发布应用程序和可能的应用程序运行环境配置。

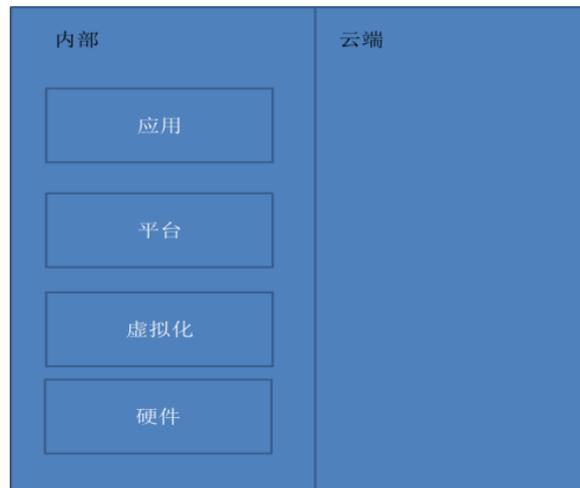
### 4.2.3 架构即服务 (IaaS)

向客户提供处理、存储、网络以及其他基础计算资源,客户可以在上运行任意软件,包括操作系统和应用程序。用户不管理或者控制底层的云基础架构,但是可以控制操作系统、存储、发布应用程序,以及可能限度的控制选择的网络组件(例如,防火墙)。

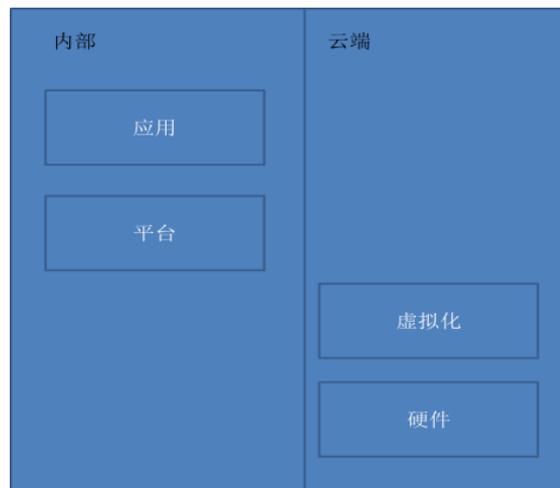
### 4.2.4 小结

为了更好解释这三种服务模型,我们看看图一。图一(A)是没有购买云服务的情形,所有设备都部署在组织内部;图一(B)是购买了 IaaS 后的情形,用户仅使用云端的硬件设备;图一(C)是购买了 PaaS 后的情形,用户不仅使用云端的硬件设备,还使用了云端的平台(比如数据库,开发工具,WEB 服务器等等);图一(D)是购买了 SaaS 后的情形,除了硬件系统,平台搭建在云上,而且把应用也部署在云上了,也就是说购买

了 SaaS 后，开发商把所有的软硬件设备都构建在云端。



A



B





D

图一

## 4.3 发布模型

### 4.3.1 私有云

云基础架构被一个组织独立地操作，可能被这个组织或者第三方机构所管理，可能存在于某种条件下或者无条件存在。

### 4.3.2 社区云

云基础架构被几个组织所共享，并且支持一个互相分享概念（例如，任务、安全需求、策略和切合的决策）的特别的社区。可能被这些组织或者第三方机构所管理，可能存在于某种条件下或者无条件存在。与公有云相比，社区云的的目的性更强，社区云的发起者往往具有共同的目的和利益的机构。比如软件测试机构为了对软件测试技术和经验有更好的研究和推广，作为研究，学习软件测试的目的，把相关的软件测试的资料，工具，论文等资源放在建立的一块云上，这样的云就叫做社区云。一般来说社区云比公有

云要小。

#### 4.3.3 公有云

云基础架构被做成一般公共或者一个大的商业群体所使用，被某个组织所拥有，并出售云服务。

#### 4.3.4 混合云

云基础架构是由两个或者两个以上的云组成，这些云保持着唯一的实体但是通过标准或者特有的技术结合在一起。这些技术使得数据或者应用程序具有可移植性。例如，在云之间进行负载平衡的 Cloud Bursting 技术。

### 5、云计算的优点与缺点

现在让我们来介绍一下云计算的优点和缺点。

#### 5.1 优点

##### 5.1.1 降低成本

以前搭建一个 IT 系统我们需要购买一堆服务器，网络设备以及一些相关的平台软件（如操作系统，数据库）和应用软件（如编辑软件，编译软件），并且可能还需要拿出一间屋子作为机房使用。现在我们使用了云，我们就可以把这些工作都可以交给云供应商，而不去管理了。

##### 5.1.2 扩展性强

使用了云服务，我们就不要再为内存不够，硬盘空间不够，CPU 速度太慢，网速太慢等类似的问题而担忧了。因为正如 4.1.4 讲到云具有的很强的扩展性，它会根据系统的使用状况，动态地为我们系统分配这些资源设备。

### 5.1.3 高可靠性

云具有很高可靠性，云存储设备一般都具有相应的备份设备，另外我们存储在云设备的数据可能在世界各地的云存储设备上都具有备份。比如我们把多年积累的照片放在移动硬盘上，一旦移动硬盘发生了问题，那么上面的所有照片全都丢失了。而我们把这些照片放在云设备上，如果云设备发生了问题，没有关系，我们还可以从备份设备上拿到我们的照片；更糟糕的是如果备份设备也发生了问题，同样不用害怕，我们的照片或许在地球的某个地方，比如阿拉斯加的云设备上存储着。

### 5.1.4 可以远程访问

云是基于互联网技术的，通过互联网你可以在任何时间，在任何地方使用我们存储在云设备上的数据或应用。比如我公司在中国上海，某天公司要求我到智利去出差，如果我们公司的业务都是基于云上的，这样我只需要带上一台可以连接到互联网的很普通的一台笔记本电脑，甚至是一台 Pad，甚至于仅仅是我的智能手机。当我到了智利，我只需要用我带上的设备连接到公司业务所在的云上，就可以处理我的工作。也就是说云计算实现了当年微软提出的"工作新世界" ( New World of Work ) 的设想。

### 5.1.5 模块化

云计算通常使用模块化的方法提供服务的。我们可以单独使用云上的 OFFICE，邮件，CRM，ERP ...软件或服务，我们也可以按照自己的需求使用云上多项软件或服务，甚至与云上的所有软件或服务。

### 5.1.6 高等级服务

如果我们没有使用云计算，我们需要雇佣一到多名系统管理员来维护我们的系统。而使用了云，我们就可以把这些任务都交给云供应商了，因为云供应商肯定会有一批专

家级的系统工程师来维护我们的系统的。换句话说，一旦我们使用了云，我们也就使用了云提供给我们的高等级的服务。

## 5.2 缺点

### 5.2.1 安全性

也许人们所意识到的云开发最大的不足就是给所有基于 web 的应用带来麻烦的问题：安全性。基于 web 的应用长时间以来就被认为具有潜在的安全风险。由于这一原因，许多公司宁愿将应用、数据和 IT 操作保持在自己的掌控之下。

### 5.2.2 数据隐私

如何保证存放在云服务提供商的数据隐私不被非法利用，不仅需要技术的改进，也需要法律的进一步完善。

### 5.2.3 数据丢失

也就是说，利用云托管的应用和存储在少数情况下会产生数据丢失。尽管可以说，一个大的云托管公司可能比一般的企业有更好的数据安全和备份的工具。然而，在任何情况下，即便是感知到的来自关键数据和服务异地托管的安全威胁也可能阻止一些公司这么做。

### 5.2.4 网络传输问题

云计算依赖网络并且目前网络的速度也不是很稳定，使得云应用的性能不高。所以说云计算的普及依赖网络技术的发展。

### 5.2.5 缺乏统一的技术标准

云计算的美好前景让传统 IT 厂商纷纷向云计算方向转型。但是由于缺乏统一的技术

标准，尤其是接口标准，各厂商在开发各自产品和服务的过程中各自为政，这为将来不同服务之间的互连互通带来严峻挑战。

#### 5.2.6 宿主离线所导致的事件

尽管多数公司说这是不可能的，但它确实发生了，亚马逊的 EC2 业务在 2008 年 2 月 15 日经受了一次大规模的服务中止，并抹去了一些客户应用数据。(该次业务中止由一个软件部署所引起，它错误地终止了数量未知的用户实例。)对那些需要可靠和安全平台的客户来说，平台故障和数据消失就像被粗鲁地唤醒一样。更进一步讲，如果一个公司依赖于第三方的云平台来存放数据而没有其他的物理备份，该数据可能处于危险之中。

#### 5.2.7 降低了系统的可测试性

使用了云计算，除了客户（或者叫用户），厂商以外还增加了供应商这个角色。尤其对于 IaaS, PaaS，当系统出现了问题就需要考虑引起问题的原因是在公司内部还是在云设备供应商处。另外由于云中运行某个固定的服务可能发生在云中任何的一台机器上，所以不管是 IaaS, PaaS，还是 SaaS，在 troubleshoot 的时候搞不清出现的问题发生在哪台机器上，这就给查看出错日志文件带来了很大的困难。此外供应商在未告知厂商的情况下对云中的软硬件进行升级，打补丁等操作造成厂商产品无法运行的例子也经常发生，这就需要系统厂商经常性的对产品进行回归测试，从而降低了系统的测试性。

## 6、云服务器

云服务器与云存储是云计算中两个关键元素，这一节让我们先来谈谈云计算中的一个关键的元素：云服务器。

### 6.1 云服务器概念

云服务器（Elastic Compute Service, 简称 ECS）是一种处理能力可弹性伸缩的计

算服务，其管理方式比物理服务器更简单高效。云服务器帮助我们快速构建更稳定、安全的应用，降低开发运维的难度和整体 IT 成本，使您能够更专注于核心业务的创新。

云服务器是云计算服务的重要组成部分，是面向各类互联网用户提供综合业务能力的服务平台。平台整合了传统意义上的互联网应用三大核心要素：计算、存储、网络，面向用户提供公用化的互联网基础设施服务。

云服务器平台的每个集群节点被部署在互联网的骨干数据中心，可独立提供计算、存储、在线备份、托管、带宽等互联网基础设施服务。

### **集群节点由以下硬件构成：**

**管理服务器：**采取双机热备的方式，对整个节点的所有计算服务器、共享存储、网络进行管理，同时对外提供管理整个节点的 API。管理服务器上提供：管理服务（管理节点的计算服务器，对外提供管理接口）、DHCP 服务（为计算服务器的网络启动分配管理网段的 IP）、tftp 服务（为计算服务器的网络启动提供远程启动映像）、nbd 服务（为计算服务器提供网络块设备服务）。管理服务器上还会运行一个数据采集程序，他定时将各种性能数据采集下来并发送到中央的数据采集服务器上  
**存储服务器群：**存储服务器可以是 ISCSI 或内置存储容量比较大的 x86 服务器，通过 集群文件系统组成一个统一的存储池，为节点内的虚拟机提供逻辑磁盘存储、非结构数据存储以及整合备份服务。

**计算服务器群：**计算服务器是高配置的八核以上服务器，计算服务器无需安装操作系统，但必须具备网络引导功能，其上运行一个 Linux 微内核、云计算软件、一个与管理服务器进行通讯的 Agent

**交换机：**按不同功能和节点性能要求配备多个三层交换机，分别负责管理网段、公网交换网段、内部交换网段、存储网段等

## 6.2 云服务器的特点

云计算服务器具有以下特点，即高密度（High-density）、低能耗（Energy-saving）、易管理（Reorganization）、系统优化（Optimization）。

### 6.2.1 高密度（High-density）

未来的云计算中心将越来越大，而土地则寸土寸金，机房空间捉襟见肘，如何在有限空间容纳更多的计算节点和资源是发展关键。

### 6.2.2 低能耗（Energy-saving）

云数据中心建设成本中电力设备和空调系统投资比重达到 65%，而数据中心运营成本中 75% 将是能源成本。可见，能耗的降低对数据中心而言是极其重要的工作，而云计算服务器则是能耗的核心。

### 6.2.3 易管理（Reorganization）

数量庞大的服务器管理起来是个很大问题，通过云平台管理系统、服务器管理接口实现轻松部署和管理则是云计算中心发展必须考虑的因素。

### 6.2.4 系统优化（Optimization）

在云计算中心中，不同的服务器承担着不同的应用。例如有些是虚拟化应用、有些是大数据应用，不同的应用有着不同的需求。因此针对不同应用进行优化，形成针对性的硬件支撑环境，将能充分发挥云计算中心的优势。

## 6.3 云服务器所要解决的问题

### 6.3.1 适应更高环境温度

在云计算数据中心中，空调系统的核心理念是注重 IT 设备的温度要求，高效解决区

域化的制冷，在云时代，IT 设备在适应温度方面变得更强壮。据了解，目前欧美许多公司已经将云服务器放在北欧，加拿大等高纬度地区，以适应温度的影响。目前的通用服务器设计标准为 35°C 进风温度，天地超云科技有限公司推出的高温节能服务器，最高可在 47°C 的环境下正常运行。IT 设备的优化工作直接导致了数据中心空调温度标准的改变。ASHRAE (美国暖通空调协会) 在其数据中心标准中发布了 2008 版本和 2011 版本；在 2008 年版本中，数据中心的温度推荐标准：温度范围为 18°C~27°C；而在 2011 年的推荐标准中，对高温 IT 设备，则扩展到了 A1-A2 框体，温度范围为 0°C~35°C。所以，云服务器需要不断提高环境温度的适应能力，能耐高温、节能成了云服务器的一个发展方向。

### 6.3.2 更加节约空间

当前不计成本的高性能计算时代已经一去不复返了，解决尖端问题的高端系统同样也必须降低成本。为了避免由于服务器爆炸性增加而造成机房面积过快扩大以及随之而剧增的各种运行维护费用，数据中心要求大幅度缩小服务器的占地面积、提高计算密度、发展高密度计算。

## 7、云存储

这一节，让我们来看看云计算上另外一个关键元素：云存储。

### 7.1 云存储概念

云存储是在云计算(cloud computing)概念上延伸和发展出来的一个新的概念，是指通过集群应用、网格技术或分布式文件系统等功能，将网络中大量各种不同类型的存储设备通过应用软件集合起来协同工作，共同对外提供数据存储和业务访问功能的一个系统。当云计算系统运算和处理的核心是大量数据的存储和管理时，云计算系统中就需要

配置大量的存储设备，那么云计算系统就转变成为一个云存储系统，所以云存储是一个以数据存储和管理为核心的云计算系统。简单来说，云存储就是将储存资源放到云上供人存取的一种新兴方案。使用者可以在任何时间、任何地方，透过任何可连网的装置连接到云上方便地存取数据。

## 7.2 云存储模型

云存储系统的结构模型由 4 层组成。

### 7.2.1 存储层

存储层是云存储最基础的部分。存储设备可以是 FC 光纤通道存储设备，可以是 NAS 和 iSCSI 等 IP 存储设备，也可以是 SCSI 或 SAS 等 DAS 存储设备。云存储中的存储设备往往数量庞大且分布多不同地域，彼此之间通过广域网、互联网或者 FC 光纤通道网络连接在一起。

存储设备之上是一个统一存储设备管理系统，可以实现存储设备的逻辑虚拟化管理、多链路冗余管理，以及硬件设备的状态监控和故障维护。

### 7.2.2 基础管理

基础管理层是云存储最核心的部分，也是云存储中最难以实现的部分。基础管理层通过集群、分布式文件系统和网格计算等技术，实现云存储中多个存储设备之间的协同工作，使多个的存储设备可以对外提供同一种服务，并提供更大更强更好的数据访问性能。

### 7.2.3 应用接口

应用接口层是云存储最灵活多变的部分。不同的云存储运营单位可以根据实际业务类型，开发不同的应用服务接口，提供不同的应用服务。比如视频监控应用平台、IPTV

和视频点播应用平台、网络硬盘引用平台，远程数据备份应用平台等。

#### 7.2.4 访问层

任何一个授权用户都可以通过标准的公用应用接口来登录云存储系统，享受云存储服务。云存储运营单位不同，云存储提供的访问类型和访问手段也不同。

### 7.3 云存储分类

**云存储可分为以下三类：**

#### 7.3.1 公共云存储

像亚马逊公司的 Simple Storage Service(S3)和 Nutanix 公司提供的存储服务一样，它们可以低成本提供大量的文件存储。供应商可以保持每个客户的存储、应用都是独立的，私有的。其中以 Dropbox 为代表的个人云存储服务是公共云存储发展较为突出的代表，国内比较突出的代表的有搜狐企业网盘，百度云盘，乐视云盘，移动彩云，金山快盘，坚果云，酷盘，115 网盘，华为网盘，360 云盘，新浪微盘，腾讯微云,cStor 云存储等。

公共云存储可以划出一部分用作私有云存储。一个公司可以拥有或控制基础架构，以及应用的部署，私有云存储可以部署在企业数据中心或相同地点的设施上。私有云可以由公司自己的 IT 部门管理，也可以由服务供应商管理。

#### 7.3.2 内部云存储

这种云存储和私有云存储比较类似，唯一的不同点是它仍然位于企业防火墙内部。至 2014 年可以提供私有云的平台有：Eucalyptus、3A Cloud、minicloud 安全办公私有云、联想网盘等。

### 7.3.3 混合云存储

这种云存储把公共云和私有云/内部云结合在一起。主要用于按客户要求的访问，特别是需要临时配置容量的时候。从公共云上划出一部分容量配置一种私有或内部云可以帮助公司面对迅速增长的负载波动或高峰时很有帮助。尽管如此，混合云存储带来了跨公共云和私有云分配应用的复杂性。

## 8、云计算平台

了解了以上知识以后让我们来看看目前比较流行的一些云计算平台。

云计算平台也称为云平台。云计算平台可以划分为 3 类：

以数据存储为主的存储型云平台；

以数据处理为主的计算型云平台；

以计算和数据存储处理兼顾的综合云计算平台。

### 8.1 开源的云平台

#### 8.1.1 AbiCloud ( Abiquo 公司 )

AbiCloud 是一款用于公司的开源的云计算平台，使公司能够以快速、简单和可扩展的方式创建和管理大型、复杂的 IT 基础设施（包括虚拟服务器、网络、应用、存储设备等）。Abiquo 公司位于美国加利福尼亚州红木市，它提供的云计算服务包括为企业创造和管理私人云服务、公共云服务和混合云服务，能让企业用户把他们的电脑和移动设备中的占据大量资源的数据转移到更大、更安全的服务器上。

#### 8.1.2 Hadoop ( Apache 基金会 )

该计划是完全模仿 Google 体系架构做的一个开源项目，主要包括 Map/Reduce 和

## HDFS 文件系统

### 8.1.3 Eucalyptus 项目 (加利福尼亚大学)

创建了一个使企业能够使用它们内部 IT 资源 (包括服务器、存储系统、网络设备) 的开源界面, 来建立能够和 Amazon EC2 兼容的云

### 8.2.4 MongoDB (10gen)

MongoDB 是一个高性能、开源、无模式的文档型数据库, 它在许多场景下可用于替代传统的关系型数据库或键/值存储方式。mongodb 由 C++ 写就, 其名字来自 humongous 这个单词的中间部分, 从名字可见其野心所在就是海量数据的处理。关于它的一个最简洁描述为: scalable, high-performance, open source, schema-free, document-oriented database。

### 8.2.5 Enomalism 弹性计算平台

它提供了一个功能类似于 EC2 的云计算框架。Enomalism 基于 Linux, 同时支持 Xen 和 Kernel Virtual Machine(KVM)。与其他纯 IaaS 解决方案不同的是, Enomalism 提供了一个基于 Turbo Gears Web 应用程序框架和 Python 的软件栈

### 8.2.6 Nimbus (网格中间件 Globus)

Nimbus 面向科学计算需求, 通过一组开源工具来实现基础设施即服务 (IaaS) 的云计算解决方案

## 8.3 商用云平台

### 8.3.1 微软

技术特性: 整合其所用软件及数据服务

核心技术：大型应用软件开发技术

企业服务：Azure 平台

开发语言：.NET

### 8.3.2 Google

技术特性：储存及运算水平扩充能力

核心技术：平行分散技术 MapReduce , BigTable , GFS

企业服务：Google AppEngine , 应用代管服务

开发语言：Python , Java

### 8.3.3 IBM

技术特性：整合其所有软件及硬件服务

核心技术：网格技术 , 分布式存储 , 动态负载

企业服务：虚拟资源池提供 , 企业云计算整合方案

### 8.3.4 Oracle

技术特性：软硬件弹性虚拟平台

核心技术：Oracle 的数据存储技术 , Sun 开源技术

企业服务：EC2 上的 Oracle 数据库 , OracleVM , Sun xVM

### 8.3.5 Amazon

技术特性：弹性虚拟平台

核心技术：虚拟化技术 Xen

企业服务：EC2、S3 , SimpleDB、SQS

#### 8.3.6 Salesforce

技术特性：弹性可定制商务软件

核心技术：应用平台整合技术

企业服务：Force. com 服务

开发语言：Java , APEX

#### 8.3.7 旺田云服务

技术特性：按需求可定制平台化软件

核心技术：应用平台整合技术

企业服务：netfarmer 服务提供不同行业信息化平台

开发语言：Deluge ( Data Enriched Language for the Universal Grid Environment )

#### 8.3.8 EMC

技术特性：信息存储系统及虚拟化技术

核心技术：Vmware 的虚拟化技术，一流存储技术

企业服务：Atoms 云存储系统，私有云解决方案

#### 8.3.9 阿里巴巴

技术特性：弹性可定制商务软件

核心技术：应用平台整合技术

企业服务：软件互联平台，云电子商务平台

#### 8.9.10 中国移动

技术特性：坚实的网络技术丰富的带宽资源

核心技术：底层集群部署技术，资源池虚拟技术，网络相关技术

企业服务：BigCloude-大云平台

## 9、云计算与大数据的关系

最后来让我们简单讨论一下大数据与云计算的关系，大数据也是目前正在新兴的一门学科。随着云时代的来临，大数据（Big data）也吸引了越来越多的关注。“著云台”的分析师团队认为，大数据（Big data）通常用来形容一个公司创造的大量非结构化数据和半结构化数据，这些数据在下载关系型数据库用于分析时会花费过多时间和金钱。大数据分析常和云计算联系到一起，因为实时的大型数据集分析需要像 MapReduce 一样的框架来向数十、数百或甚至数千的电脑分配工作。现在需要处理的大数据在几年前是无法估算的，让我们来看看大数据究竟大道哪个程度？

**按照进率 1024 ( 2 的十次方 ) 来计算：**

1Byte = 8 bit

1 KB = 1,024 Bytes

1 MB = 1,024 KB = 1,048,576 Bytes

1 GB = 1,024 MB = 1,048,576 KB

1 TB = 1,024 GB = 1,048,576 MB

1 PB = 1,024 TB = 1,048,576 GB

1 EB = 1,024 PB = 1,048,576 TB

1 ZB = 1,024 EB = 1,048,576 PB

1 YB = 1,024 ZB = 1,048,576 EB

1 BB = 1,024 YB = 1,048,576 ZB

1 NB = 1,024 BB = 1,048,576 YB

1 DB = 1,024 NB = 1,048,576 BB

可见，处理如此之大的数据，如果使用单一的计算机是根本完成不了的，但云计算的产生，为大数据处理成为了可能。

### 参考文献

[1] <http://baike.baidu.com/view/13605.htm>

[2] <http://baike.baidu.com/view/453197.htm>

[3] <http://baike.baidu.com/view/20049.htm>

[4] <http://baike.baidu.com/view/1316082.htm>

[5] [http://baike.baidu.com/link?url=R1EE7a8gsOA\\_MX1c-aSG\\_-UvD3CSI1WoYIPvHKKqIA7uz4fvHPaQDEg2OMN6q7mvlNM0tRmdeGmSTJi0\\_66BIK](http://baike.baidu.com/link?url=R1EE7a8gsOA_MX1c-aSG_-UvD3CSI1WoYIPvHKKqIA7uz4fvHPaQDEg2OMN6q7mvlNM0tRmdeGmSTJi0_66BIK)

[6] <http://csrc.nist.gov/groups/SNS/cloud-computing/cloud-def-v15.doc>

[7] <http://www.3testing.com/new/d2.htm>

[8] 《云计算和大数据时代网络技术揭秘》 徐立冰著

[9] 《云服务测试：如何高效地进行云计算测试》 Kees Blokland Jeroen Mengerk Nartin Pol 著

# 《赢在测试 2- 中国软件测试专家访谈录》读书笔记

◆ 作者：顾翔

## 1、阿尔卡特-朗讯测试架构师 郑文强

1) 作为一个合格的测试人员首先需要具备深入了解测试对象，它的架构，功能以及客户是如何使用他们的业务知识的。火车模型[1]，TCL 语言 ( Tool Command Language )

2) 结对测试 Pair Testing ( 多人测试 Mutiple Testing )

业界内有两种含义，一是指两个人一起做测试，二是指一种测试设计技术，在有多种参数需要测试情况下，应用结对测试减少测试用例数，提高测试效率

3) 今天分享了经验，同时需要一段时间去消化。的确，很赞同，分享很重要

4) 项目成功的观点：沟通，分享，合适的测试过程，测试与开发的紧密合作。

5) 如何成为优秀的测试工程师

A. 深刻了解测试对象

B. 熟悉研发流程

C. 如何有效地去做

D. 培养软技能

6) 探索性测试(Exploratory Testing)是 Cem Kaner 在 1983 年提出的，是软件测试的一种方法。与脚本化测试 ( Scripted Testing, ST ) 相比,探索性测试将更高的认识水平放在测试执行方面，同时更强调测试人员学习，设计，执行与结果分析等测试活动并行，相互反馈与相互支持。与脚本化测试相比，探索性测试更强调测试人员的思维自由度与主观能动性

7) 软件发布之前的评估

缺陷

测试通过率

测试覆盖率

信心

8) 缺陷检测百分比 DDP ( Defect Detected Percentage ):

客户现场发现缺陷数/ ( 发布前测试团队发现缺陷数+客户现场发现缺陷数 ) \*100%

9) 如何做好测试设计

明确参考输入：需求，开发文档，标准与规范，测试经验知识库

关注功能之间的交互

"相对于开发人员来说，功能交互是测试人员的优势，我们要在这方面好好发挥"，我非常赞成这个说法

采用合适的技术与方法

做好评审

10) 思维导图：Mind Mapping,是表达放射性思维的有效图形工具<sup>[3]</sup>

成为测试牛人

制定目标

技能要求

立即行动

11) 面试中考虑点

产品知识与业务知识

测试流程，技术及方法

性格特征

12) ISTQB 测试认证沙龙

ISTQB 测试认证<sup>[4]</sup>

13) 测试人员提升自己水平

A. 自动化测试与手工测试之争

B. 探索性测试与脚本化测试之争

C. 敏捷开发下的敏捷测试

## 附录

[1] 软件开发中的火车模型发布模式

<http://www.biaodianfu.com/release-train.html>

[2] TCL 语言

[3] 思维导图

[4]ISTQB

## 2、广联达 彭月

1) 有多种业余爱好是一笔宝贵的财富，可修身养性，也可打发时间

2) 测试一定要比客户更专业

3) 挑战 1，特殊用户使用的场景

挑战 2，专业知识深度不够

挑战 3，实验室环境与客户环境不同

4) 敏捷中保证测试工程师独立而独特的视角，坚决同意

5) CMM5,千行代码 bug 率在千分之 2.53 之下

6) IBM ODC Orthogonal Defect Classification 正交缺陷分类<sup>[1]</sup>

7) 绩效分为基本绩效和贡献

8) 软件测试做 2 个纬度

验证我们的产品是否达到设计要求

在第二步基础上再去发现更多缺陷，并且是否可以优化

9) 作为一个管理者，首先要有经营意识

10) 支持鼓励渐进性创新，而不是变革性创新。注：可操作性强

11) 测试框架

云计算服务层

核心驱动层

自动化测试框架

12) 招聘看中

聪明

有激情

该学的都学好

13) 突破发展前景

14) 保持危机意识

15) 实现专业生存

16) 保持开发心态

### 附录

[1] 正交缺陷分类 ( ODC ) 流程简介及应用经验分享

### 3、独立测试顾问 邵晓梅

1) MFQ&PPDCS [1]

2) StartWest/StartEast [2]

3) 测试压倒开发和开发压倒测试一样，不是好的项目状态

4) 预防测试 PreventiveTest 完整团队 ( whole Team )

5) 启发

第一，如果一个产品或项目有大量 bug 暴露出来，作为项目管理者要注意了，这意味着项目本身有很大改进空间，产品质量不容乐观。

第二，测试流程只起到辅助型的作用。

测试流程是个启发式的(Heuristics),遵守流程，测试不一定做得好；不遵守流程，测试也不一定做不好，测试流程更多起到辅助型而不是决定性作用。(测试流程是否需要，作用在哪里，是否可以取消？)

第三，做任何测试工作，首先要做到 Know Your Mission(知道你的任务所在)

6) 测试认为的三个阶段：

以 bug 为中心

以流程为中心

以人为中心

7) 测试深度图 ( Test Depth Graph )

深度思考 ( Focused Thinking )

广度思考 ( Defocused Thinking )

8) 基于需求的测试 ReqBT ( Requirements Based Testing ) [3]

快速软件测试 RST(Rapid Software Testing)[4]

James Bach 和 Michael Bolton 讲述的一门课程，侧重于如何在测试进度紧张，测试资源有限等情况下快速而有效地开展测试工作。

Quality is the value to someone who matters

9) 对于软件测试，可以先实践后理论的路子

10) 如何学习软件测试

多实践，多思考

人与人之间最有效的沟通是面对面的交流

三步法

描述目标

要实现目标，具备哪些知识和技能

掌握这些和技能

测试就是学习 ( Testing is learning )

11) 如何把握软件质量

1, 找到问题的客户是谁？

2, 明白客户的在意的价值事啥？

3, 制定测试策略-----优先风险高的，不要平均用力

12) GOM(Goal-Question-metrics)

13) 问问题三步法

A) Huh? 恩？

B) Really? 真的吗？

C) So? 接下来如何做？

14) 不是如何提高测试设计能力，而是提高分析能力

15) 基于模型的测试(Model Based Test)

[http://wenku.baidu.com/link?url=6jBquzGXriyxWihZlrbz2r6Bt\\_Sz2FAgFJvWM6Mgy\\_sQ4ukQJ82y6WZCeMLTz2EvHuP9CVoDWj6JMLqgevOikLQECP26CmfveoZdY0QdkFy](http://wenku.baidu.com/link?url=6jBquzGXriyxWihZlrbz2r6Bt_Sz2FAgFJvWM6Mgy_sQ4ukQJ82y6WZCeMLTz2EvHuP9CVoDWj6JMLqgevOikLQECP26CmfveoZdY0QdkFy)

16) 测试的独立性

测试可以

- A) 让开发人员做
- B) 让专业的测试人员做
- C) 独立的第三方

#### 附录

[1] MFQ&PPDCS:

[2] 09 年美国 StarWest 和 StarEast 举办了全美国最大的软件测试领域的展会及展会部分资料

[3] 专家观点： James Bach 讲如何成为软件测试专家

[4] 认识软件测试中的黑天鹅

#### 4、迈瑞 肖利琼

1) 作测试逆向思维很重要，测试是要找出软件有哪些问题，而不是证明软件没有问题

2) 成为一个优秀的测试工程师

- A) 积极主动的心态
- B) 责任心强
- C) 善于总结，乐于分享
- D) 全局意识强

3) 测试方案：

- A) 切实了解和掌握用户使用场景
- B) 了解开发人员实现原理
- C) 确定测试范围-> 深入分析测试对象-> 确定测试点和测试方法

4) 测试对象分析法

1) 三层架构模式分析法：

功能应用层

模块接口层

系统接口层

2) 多叉树结点分析法

3) 业务状态变迁分析法

4) 代码更改追溯分析法

5) 做好评审

A) 用例编写者自评

B) 测试主管审核

C) 参加评审会

D) 高频用例；中频用例；低频用例

E) 测试用例的交叉执行会是一个检查和提高测试用例的好办法。A 执行 B 的测试用例；B 执行 A 的测试用例-----非常同意

6) 嵌入式自动化测试

A) 数据层：负责与脚本程序打交道，读取数据，分析数据等

B) 逻辑层：处理自定义脚本关键字的识别，与系统的动作做交互

7) 内存泄露软件：top, valgrid

8) 把握软件质量

A) 制定好测试策略

B) 做好测试过程控制 IBM ODC Orthogonal Defect Classification, 缺陷正交分析法[1]

C) 邀请客户试用

9) 优秀管理者

A) 培养人才

B) 合适的人作合适的事情来分配工作---good

- C) 胜任人才招聘和人才识别
- D) 维护一个好的团队氛围

附录

<http://www.ithov.com/linux/122683.shtml>

## 5、百度 董杰

- 1) 发现实验室里所看重的功能不是客户所要的；而用户所要的却不是开发和测试看重的重点
- 2) 对我看到四五十岁的人仍然在技术一线工作，对创新保持者热情市，内心感动
- 3) 如何开展测试
  - 减少测试用例的用余---从设计提高效率
  - 减少漏测---从设计提高质量
  - 提高测试评审质量---提升测试本身的质量
- 4) 《基于缺陷的测试技术体系》
- 5) 变化分为两类：
  - 基于社会，行业，产业的变化
  - 技术的发展
- 6) 因为教是最好的学
- 7) 如果你已经做到了测试的高级管理职位，我恭喜你，同时也提醒你不要完全脱离具体的测试工作
- 8) 如何开展测试工作
  - 首先测试工程师要做好被测对象的分析和建模，要理解清楚业务
  - 其次，应用多种测试方法来做测试
  - 第三，展开缺陷预防工作
- 9) 探索性测试适合新人快速掌握业务。探索性测试大粒度，小粒度
- 10) 探索性测试不是自由测试，只有测试有两个软肋：

没有计划，随机性强

经验无法积累，无法给别人做培训

11) 对于预防缺陷：

发现缺陷

建立质量保证体系，预防缺陷

A) 需求方面：结对需求评审是很好的需求缺陷预防的很好方法

B) 架构设计方面：收集公司里的设计缺陷。

灰盒测试分析法：在开发人员写好设计文档以后，测试人员阅读设计文档并创建产品设计的二叉树表，画出产品从一种状态迁移到另一种状态的过程，每个过程都要画出成功与失败两条路径。

C) 编码的缺陷预防：白盒测试技术

12) 如何做好测试设计

A) 做好测试分析，做好测试对象的分类与建模

B) 采用多样化的测试手段

C) 准备多种测试数据

D) 做一些该产品领域特有的测试

13) 个人发展

A) 坚持爬坡

B) 不要局限在自己小圈子里

C) 可以从宽度和广度上来发展

## 6、金山 赖炳新

1) 作产品经理的体会

A) 要用运营的是未来高铝产品质量

B) 软件研发过程修复所有缺陷是不可能的，优先解决用户经常用到的问题，以客户为核心

2) 如何成为优秀的测试工程师

- A) 对所从事的职业有正确认识
  - B) 耐心，细心
  - C) 钻研精神
  - D) 要有恐惧之心
- 3) 如何成为优秀管理者
- A) 从一线起来
  - B) 沟通能力
  - C) 了解团队成员的特点
  - D) 身先士卒
- 4) 如何测试安全产品
- A) 安全产品影响面广，不光是自身，重视兼容性
  - B) 性能很重要
  - C) 安全类软件的本身安全性很重要
- 5) 安全类软件测试的三个技术要求
- A) 对 Windows 机制要有深入了解
  - B) 安全软件测试对代码能力有一定的要求
  - C) 对病毒，木马要有一定的了解
- 6) 安全软件性能测试
- A) 自身的性能
  - B) 与同类产品比较的性能
- 7) 招聘每个新人的要求
- A) 技术方面要有一定要求
  - B) 更看重人的品格
  - C) 学习能力好
- 8) 非安全软件的安全测试
- A) 到软件最核心的地方去做
  - B) 通过一个好的开发流程来保障软件的安全性

C) 提高产品的迭代速度，这样可以及时修复和改进产品

9) 快速迭代中保证质量

A) 发给内部测试用户

B) 发到论坛上

C) 发给更大的客户群

D) 发给所有用户

10) 做事的原则和方法

A) 去一家公司工作，要相信它

B) 相信你选择的行业

C) 要与同事相互信任

D) 软件行业不是传统行业，不要有过于机械的等级关系

## 7、淘宝 高翔

1) 在微软期间认识到：

A) 测试工程师应当持续培养自己的代码能力

B) 测试工程师不仅是找 bug，而且还要想到如何帮助开发

C) 要相信测试可以做得更规范，更好

2) 块，好，准，稳

CTC(Common Test Center) 把互联网里面公共测试用例放入 CTC，集中维护，共同管理，避免重复设计测试用例-----我非常赞同

3) 测试与开发亦敌亦友，有合作的地方，也有对抗的地方。如果你和开发没吵过架，你就没有真正站在客户立场上-----有同感

4) 优秀的测试

A) 学习流程和规范

B) 学习行业内最佳实践

C) 拓宽视野

- D) 分享总结
- 5) 测试用例设计 6 个阶段
  - A) 啥都不懂
  - B) 开始做
  - C) 如何更好地做好测试用例的设计
  - D) 继续提升
  - E) 融合起来
  - F) 学习探索式测试
- 6) 评审：内部评审，外部评审  
开评审会，让愿意发言的人去-----同意
- 7) 对于互联网来说，自动化率超过 50%就很好了
- 8) 测试技术三大方向
  - A) 探索式测试
  - B) 基于需求的测试
  - C) 基于模型的测试：根据软件系统的规格说明建立模型。根据模型自动生成测试用例，自动运行。常见工具：Spec Explorer
  - D) 基于风险的测试 RBT(Risk Base Testing)
- 9) 底层自动化测试，上层探索性测试
- 10) 敏捷好处
  - A) 提高测试效率
  - B) 提高自动化测试框架的质量
- 11) 测试工程师发展方向
  - A) 提高自己开发能力，转向专为测试开发
  - B) 转向项目管理
  - C) 转向产品需求分析与收集
- 12) 如何学习
  - A) 相信自己的梦想

- B) 做好分析和总结
- C) 沟通

## 8、CA 步琼

1. 用 Perl 实现 Web Service & CLI, 用 Selenium 和 JAVA 来做 GUI 测试
2. 对同事要求特别高, 这是双刃剑, 用的时候要小心
3. 优秀测试工程师
  - A) 必须爱钻研
  - B) 很强的逻辑思维
  - C) 具备较轻的技术能力
  - 掌握常用的测试方法
  - 掌握常见的测试工具
  - 掌握编程语言
  - D) 掌握产品知识
  - E) 细心
  - F) 良好沟通能力
  - G) 良好工作态度
4. 做好测试用例
  - A) 了解类似功能
  - B) 有权限的话看看代码
  - C) 与相应开发人员沟通
  - D) 把自己的测试思路画在纸上, 看有无遗漏
5. 评审会
  - A) 会议要有主持人
  - B) 要形成改进措施
  - C) 最好做复审

6. 可以没有开发背景，但应当懂编程
7. 性能测试的建议
  - A) 之前制定好 Baseline，就是基本定义
  - B) 注重附加影响，比如 CPU 占用率一直很高
  - C) 熟悉一些概念，比如区分内存使用中的 Working Set & Virtual Size/Virtual Bytes
  - D) 掌握一些常用工具
8. 推行在测试中采用基于风险测试，根据风险等级排优先级
9. 即使敏捷，测试人员在开始回比较闲，要介入设计，不要等
10. 要想深入，至少持续做 3 年
11. 永远保持学习的热情
  - A) 信守承诺
  - B) 不要随波逐流，保持自己方向
  - C) 坚持不懈
  - D) 对学习始终保持热情

## 9、用友 薛继国

1. 新人加入公司，导师制-----同我们公司
2. 兴趣是最重要的，总结能力也很重要
3. 管理技术两条线  
普通测试工程师----> 高级测试工程师---> 测试专家-----> 高级测试专家  
|--> 测试经理
4. 提高创新能力的另外一个小小尝试是交换测试
5. 做好团队管理
  - A) 管理者起表率作用
  - B) 让员工认识到自己的积极性不够
- 绩效考核 PBC Personal Business Commitment 分为 A B C D 四个级别
6. 每一个管理岗位配备一个副手

7. 大会上少表扬，日常工作多表扬少批评
8. 很少看候选人检验，看素质
9. 做测试用例：测试经理与测试专家，测试设计人员。参照别人的测试用力提高自己的水平-----我同意
10. 自动化测试的要求：核心产品，核心场景
11. 测试技术并不难，难的是业务系统-----我同意
12. 如何学习业务，通过产品去学，强调自学
13. 我们是做测试的，要敢于提出问题，敢于反对；不要制造矛盾

## 10、奇虎 360 王东

1. 经历往往是财富
2. 绿萌三个主要产品
  - A) 入侵检测
  - B) 漏洞扫描
  - C) 防 DOS 攻击
3. 作为测试工程师，不要满足于文档，要多做些文档没有提到的东西考虑不同的场景。我们测试不只是证明软件工作正确，这只是初步目标，我们还要考虑什么情况下会出错。-----200%同意
4. 走向优秀
  - A) 学习和总结能力是关键
  - B) 做好沟通
  - C) 积极向上的精神
5. 提炼自己的方式学习
  - A) 找一个高手辅导自己，这样学习更快
  - B) 提炼自己的方式学习
  - C) 不要忽视软技能
6. 安全软件测试重点是理解安全需求，安全软件的稳定性及容错性很重要

7. 如何做好非安全产品的安全测试
  - A) 审核自己软件里开放接口和界面
  - B) 多做代码 review, 从内部做好预防
  - C) 可以请测试工程师作暴力测试
8. 设计好的测试用力是理解功能
  - A) 理解功能
  - B) 测试人员要主动提问
  - C) 然后主动去学
9. 安全领域的自动化测试
  - A) 提高测试正确性
  - B) 提高测试效率
  - C) 有些地方是手工测试无法作的, 要用自动化
10. 自动化测试包括
  - A) 回归测试;
  - B) 性能测试;
  - C) 主要功能点的测试
11. 提高测试设计能力
  - A) 召开测试用力设计讨论会
  - B) 注意举一反三
  - C) 生活中要注意观察用户
12. 典型应用场景是性能测试的重要步骤
13. 面试
  - A) 积极的工作态度
  - B) 对自己有合适的认识
  - C) 具备良好的沟通表达能力
  - D) 学习能力
14. 管理---管好两头

- A) 管理领头的优秀人才
- B) 抓好所谓拖后腿的
- 15. 管理者不应该做的事情
  - A) 不要孤立自己
  - B) 不要脱离业务
  - C) 不要推脱责任
- 16. 测试人员与开发人员走向融合，开发要会测试，测试要会开发---不同意
- 17. 软件测试从业人员面临的挑战
  - A) 来自于公司高管对软件测试的看法
  - B) 不同的开发技术和测试技术如何融合问题

# 《云服务测试- 如何高效地进行云计算测试》读书笔记

◆ 作者：顾翔

## 1、什么是云计算

### 1.1 前题

互联网

虚拟化技术

面向服务架构体系 SOA

网格计算

付费模型：按使用量付费

### 1.2 云计算定义

NIST 美国国家标准与技术研究院

#### 1.2.1 云计算定义

1.2.1 5 个特征：按需的自助服务

广泛的网络访问

资源池

快速的弹性

可度量的服务

#### 1.2.2 3 个服务模型

架构即服务 IaaS

平台即服务 PaaS

软件即服务 SaaS

### 1.2.3 4 个实施模型

私有云

社区云

公有云

混合云

## 2、测试经理角色

### 2.1 一般任务

#### 2.1.1 风险分析

#### 2.1.2 获取供应商信息与供应商达成一致

#### 2.1.3 端到端的测试

- a) 端到端的测试范围大于系统集成
- b) 端到端测试关注点：IT 环境，系统场景，IT 环境下系统与服务详细功能
- c) 创建端到端测试用例
- d) 执行端到端测试用例
- e) 自动化端到端的测试执行
- f) 端到端的回归测试

#### 2.1.4 给出建议

### 2.2 选型，实施，生产阶段任务

#### 2.2.1 选型阶段

#### 2.2.2 实施阶段

#### 2.2.3 生产阶段

### 2.3 借助云帮助测试

#### 2.3.1 使用 TOGA 将测试外包给云计算

启动

准备

转移

## 实施

### 2.3.2 众包测试

### 2.3.3 云端的测试环境

创建虚拟环境的快速，

创建典型的测试环境，

为移动资源执行可移置性测试

### 2.3.4 生成负载

1，生成重负载

2，来自世界各地的负载

## 3、从风险到测试

### 3.1 性能风险

与其他设备公用，某地点性能差，其他客户影响，扩展性不好，供应商改变服务，无法执行性能测试。

### 3.2 安全风险

工作新世界要求 BYOD，携带自己的设备工作，供应商访问数据，数据丢失，权限模型不完善，供应商出现事估导致数据无法访问，数据被意外删除。

### 3.3 可用性，可持续性风险

某服务不可用，事故可多快速度恢复，有无替代品，某服务受到拒绝服务 DDoS 相关的其他产品是否会受影响，服务供应商破产是否有备份，数据丢失在某处。

### 3.4 功能性风险

要求兼容不同硬件及软件平台供应商作不到。供应商部署新版本没有通知。迁移操作(迁移回来，迁移到其他云服务商。请求，获取，支付附加服务容量不够，导致业务中断。

### 3.5 可维护性风险

服务供应商对服务执行各种服务的维护，而服务使用者对维护毫无控制权。使用服务遇到问题找谁？如何提交记录问题？

### 3.6 法律和法规风险

本地和全球的隐私保护方面的法律法规。法律法规自身风险 ,公司把数据放在 A 国 ,根据 A 国法律可拿到公司的数据。

### 3.7 供应商和外包风险

风险主要与供应商依赖有关。合同没有覆盖部分不确定。对性能 , 安全性等术语理解不同。业务冲突下如何供应商处理 ? 多个供应商出现推卸责任。

## 4、测试方法

### 4.1 选型阶段的测试

往往会考虑两三家服务供应商 , 仅依赖一家供应商 , 即厂商锁定 ( VendorLock-in )

#### 4.1.1 决定选型需要考虑的云计算相关特征

见测试风险。

#### 4.1.2 确定选择标准的完整性与可控性

SMART 原则 : 具体的 , 可测量的 , 一致的 , 切合实际的 , 有时间约束的。

服务的选择标准 , 如何将业务需求转换为需求和用于 IT 解决方案 ( 我们的云解决方案 ) 的验收标准。

供应商的标准 , 云认证专家 Cloud Certified Professionals。

#### 4.1.3 评估服务和供应商

检查规范与条款

寻求参考和认证

NIST 2012 年 7 月颁布 3 组标准 : 互通性 , 可移植性 ( 被遗忘的权力 , 数字迁移后真被删除了吗) , 安全性

执行审计和审查

概念验证 : 执行一个概念验证 POC , 临时使用服务 , 模拟生产环境。

模拟端到端业务流程 : 模拟引入的新服务。

#### 4.1.4 给出建议

### 4.2. 性能测试

新的性能测试 : 弹性测试

压力测试不可影响该服务其他客户

首先确保基础设施上的性能，为转移到云端作参考。SaaS 性能关键主要于服务提供商。

#### 4.2.1 负载测试

像用户一样操作，确定是否有足够响应速度及否定发发生错误。不同时间，负载测试结果不同。方法：尝试从供应商获取其他用户的负载结果。及在规定的时间内执行，比如 8:30，但结果可能也不同

#### 4.2.2 压力测试

找出超出服务峰值时被测系统行为。注意会影响相关服务其他客户，确保使用该服务的使用条款。变种，测试供应商承诺负载水平边缘的行为。

#### 4.2.3 耐力 / 容量测试

短时间内加入更多负载，以更快获得结果。要注意超出官方允许的负载。供应商自行测试服务稳定性是个明智行为。

#### 4.2.4 弹性测试

目标确定服务的性能是否满足客户的整个负载范围，以及分配服务容量是否与服务负载相称。按使用付费模式往往与服务弹性相关。方法：执行负载测试先超出边界(扩展)再将边界降低到边界下(缩减)，全手工完成。发现问题可能：容量扩展不符合需求，容量缩减不符合需求，扩展或缩减中功能发生问题，基于使用付费的信息不符合需要。

#### 4.2.5 建立测试用例

性能测试用例基于负载剖面。

建立运行剖面。使用云意味着选择供应商开发的方案。运行剖面是对系统实践中如何被使用或将要如何被使用的描述。运行剖面由一组操作组成，每个操作产生可能性。运行剖面 5 步骤：

- a) 确定操作发起者。
- b) 列出操作。
- c) 评审列出的操作。
- d) 确定操作的频度。
- e) 确定每个操作发生的可能性。

#### 4.2.6 针对特定瓶颈的测试用例

#### 4.2.7 在测试用例中包含云的特性

比如，当确定供应商和用户大概物理位置后，可能在性能测试中使用位置信息。

客户使用的资源(设备)：PC，笔记本，智能手机，平板电脑

使用网络：GPRS，GSM 中的 EDGE，ADSL，WiFi

#### 4.2.8 压力测试的测试用例

确定负载上界，在上界进行边界值测试；没确定上界，可选择远高于峰值负载的负载剖面。

#### 4.2.9 耐力/容量测试的测试用例

耐力测试在负载峰值 70%作为负载剖面，在一段时间内持续。容量测试设计负载剖面本质上压缩时间，比如每天一次动作，每分钟 100 次，记住供应商签订合同很重要

#### 4.2.10 测试弹性的测试用例

自动/手工发生容量扩大或缩减

用变化的负载剖面进行负载测试。

在扩大或缩减的边界进行边界值测试

对管理流程进行循环测试，包括按使用付费。

假设使用容量与费用之间有直接关系，目标峰值达到当前峰值 2 倍，结果又执行了压力测试。

#### 4.2.11 设置性能测试

度量工具，用于确定相应时间。

负载发生器，用于在服务器上增加负载。

监视器，用于衡量基础设施的特定要素。如负载是否发生错误信息。

云服务优势在于它可通过互联网访问，并使用适合互联网的协议。

#### 4.2.12 代表性的测试环境

云计算有益特性：可在与生产环境相似环境下测试。但由于环境每天都在变，所以性能测试应加入 CI。有时买双倍服务容量也比用非云搭建类似运行环境强。

### 4.3 测试安全性

ISO27001 定义 3 个维度

数据保密性与非授权用户能查看到数据风险。谁可访问

数据的集成性与数据被意外修改或丢失的风险。用户相信的数据正确吗

数据可用性与数据及服务在需要时不可用风险。用户可随时访问吗

授权与认证

软件与硬件质量

在不同层次上进行加密

通过欺骗人来获取数据访问权限

记录所有数据事务及所有获取数据访问权限的尝试

确定服务器安全级别

了解分布在全球不同地点服务所在安全性

服务供应商应提供服务相关数据的安全性

#### 4.3.1 网络安全性

对互联网安全性要求不要太高。协议层，SSL TSL 等，对 WiFi WAP2

#### 4.3.2 列出供应商安全清单

检查，服务所在位置的安全。检查供应商如何组织授权与认证过程

a)认证。

使用用户名，密码

使用令牌系统生成接入码

使用访问卡

使用生物数据(指纹，视网膜)

对于云身份即服务 IDaaS，DIGID 由荷兰政府提出及 OpenID 都是身份管理例子。

基于 XML 安全断言语言 SAML 用于交换认证与授权信息

b)授权

c)日志文件与审计跟踪记录

#### 4.3.3 列出客户安全性清单

客户侧不安全行为

在计算机中使用记住密码

不为移动设备加锁

口令漏洞

易受社会工程攻击的个人行为

4.3.4 测试加密。

有许多测试工具验证是否加密

4.3.5 测试认证。

使用功能测试技术,语法测试(合法/非法登陆数据),过程循环测试(发布认证流程),数据循环测试(认证的生命周期),用户不安全行为。混合认证机制更好保护

4.3.6 测试授权。

用功能性测试方法

4.3.7 测试面对互联网攻击时的安全稳定性

目录遍历：尝试读写访问目录之外的目录

XML 外部实体攻击：XML 中包括坏数据

SQL 注入：通过操作 SQL 请求，向系统请求或修改数据

跨站脚本 XSS：在用户不知情的情况下向其他网站发信息

会话挟持：在会话中跳过验证步骤

需深入的知识及对测试工具丰富经验

4.3.8 测试日志文件及跟踪记录

4.3.9 对及时应用安全性补丁进行测试

服务侧，客户侧都要测

4.3.10 执行审计。

除 ISO27001 外，还有 SAS70

在云中需关心不同用户数据如何隔离

#### **4.4 测试可管理性**

可管理性本质：我们如何保持服务和围绕它的过程正常工作。

可管理性与可维护性非常相关

可管理性可通过检查表进行测试，静态形式

是否有可用文档

是否及时更新文档

是否将修改好的文档发给正确干系人

#### 4.4.1 供应商侧的规范

##### a) 接口规范

服务与不运行在云上的系统接口

服务与服务间接口

##### b) 客户资源规范

##### c) 平台规范

##### d) 基础设施规范

#### 4.4.2 客户侧规范

##### a) 基础设施规范 IaaS

##### b) 系统规范

##### c) 架构文档

##### d) 业务过程文档

##### e) 业务需求

#### 4.4.3 用户文档

客户如何配置，使用，维护服务，FAQ，已知错列表，在线课程。

定期更新平台软件，安装安全补丁，修复 bug，会影响其他客户服务。

#### 4.4.4 测试环境可用性

a) IaaS/PaaS 中的测试。开发，测试，验收环境都要从云端获得。

b) SaaS 用户的测试环境。测试用户的活动不应该影响供应商或其他客户，所以开发，测试，验收在仿服务器上，生产在云上。

#### 4.4.5 测试文档

##### a) 客户测试文档

##### b) 供应商测试文档，有理由分享给客户

#### 4.4.6 事故管理流程

a) 供应商解决事故情况。服务更新应先发布在测试环境中，没测试环境，不得不在生产环境中测试，服务更新应该包含影响客户的变更信息

b) 客户解决事故情况

修改服务配置

不修改服务本身修改业务过程以解决问题

c) 事故无法解决的情况

转向另外服务，发生数据迁移

4.4.7 变更流程与版本控制

4.4.8 软件可管理性 PaaS, IaaS

#### 4.5 测试可用性 / 可持续性

可用性是可持续性前提

SaaS 服务供应商是黑匣子

PaaS, IaaS 客户供应商交在一起

4.5.1 失效模式影响分析 FMEA

由于代价高，仅对高失败风险业务进 FMEA

找到失效机制

确定失效原因

确定失效影响严重性

设计活动(目的：防止失效，减少发生几率，减少影响)，与产品风险分析 PRA 关注点一致。FMEA 是个迭代过程。

4.5.2 架构作用

SaaS 黑匣子，PaaS, IaaS 可画整体结构

4.5.3 硬件可靠性。

通过测试监控工具及审查日志

4.5.4 软件可靠性。

是持续性重要部分

4.5.5 承诺和 SLAs

a) 承诺，99.99%

b) SLAs

平均故障时间 MTBF

平均修复时间 MTTR

c) 要点

可用，不可用定义

列出所有相关部分

决定如何处理服务供应商之外的失效

客户与供应商之间责任边界需明晰

4.5.6 可用性机制影响

镜像

4.5.7 因特网与因特网连接

可能供应商的因特网好于客户因特网连接

4.5.8 测试失效修复

SaaS 供应商测试失效大多数情况

PaaS, IaaS 客户与供应商交互

a) 失效测试用例

测试到失效用的时间

结束失效用的时间

修复可能的破坏所用时间

要在类似生产环境下测

b) 失效恢复测试检查点

c) 测试管理视角的失效恢复测试

d) 失效恢复过程模拟

4.5.9 测试在离线状态下工作

## 4.6 测试功能

在功能性方面，获取 SaaS 服务可类比购买软件包

服务的软件质量是供应商的责任，他们必须符合确保合适的软件开发过程，以保证客户不会被软件缺陷所困扰。

#### 4.6.1 服务与业务过程的兼容性

这个问题与用户验收测试 ( UAT ) 的目标类似

过程循环测试 ( PCT )

#### 4.6.2 测试服务质量

探索性测试是一种有效的方法，作为探索性测试的一部分，测试者需要给予假设和以前的经验持续确定预期结果。

#### 4.6.3 测试用户有好性

包括用户文档

#### 4.6.4 测试与其他系统的接口

a)接口使用的技术

b)接口使用的语法

c)接口的语义

d)非功能层面

有许多测试工具可以测试接口

#### 4.6.5 测试服务配置

a)通过审查发现

b)需要有配置专员进行测试

c)由一组配置专员进行测试

#### 4.6.6 供应商定制化

供应商针对所有客户订制服务

#### 4.6.7 客户定制化

客户在客户侧进行测试

#### 4.6.8 测试 Web 服务

a)WSDL

b)工具

c)测试用例

#### 4.6.9 多平台测试

工作在新世界 ( New World of Work ) 和 BYOD 需要多平台测试，据我参加 2014 CSQTB 会议获知 HP 已经成功研发了这种测试的测试工具

a) 供应商侧的考虑

b) 客户侧的考虑

4.6.10 测试应用本身，以及使用应用来测试服务

在移动设备上测试应用：移动设备存储空间小内存更小，且考虑因特网连接实效

4.6.11 测试离线功能

4.6.12 回归测试

建议使用 CI

4.6.13 创建测试依据

a) 过程流

b) 用况

c) 分类树

d) CRUD

e) 授权表

f) 接口规范 ( 协议 )

4.7 测试迁移

a) 转移到云中，引用保持原样(IaaS PaaS)

b) 转移到 SaaS

c) 从一个 SaaS 转移到另一个 SaaS

d) 从云端迁出

转移过程经常需要转移数据

4.7.1 迁移测试策略

成功迁移描述 ( 略 )

4.7.2 最小化业务中断

降低风险可以采取模拟迁移

4.7.3 IaaS 和 PaaS 中正确的数据迁移

检查比如日期格式是否发生变化"月-日-年"变为"日-月-年"

#### 4.7.4 SaaS 中正确的数据转换

- a) 需要工具来帮助
- b) 转换规则是否正确工作
- c) 对输入数据的转换工作正常吗
- d) 是否有数据丢失
- e) 是否有未完成的事务丢失（迁移过程中正在进行的事务）
- f) 转换工作之后，是否有敏感数据可被访问
- g) 数据迁移是否传输了过多的数据到服务中去

#### 4.7.5 迁移的性能

- a) 速度
- b) 基础设施容量
- c) 完全压力情况下的稳定性

转换大量数据的时候，云计算允许用户从云端临时获得处理资源，因此性能不一定是瓶颈

#### 5.7.6 数据清理

#### 5.7.7 测试环境迁移

#### 5.7.8 并行运行于模拟运行

模拟运行与并行运行都和昂贵，模拟运行需要两个环境，最终用户需要并行使用两套系统

模拟/并行运行中的一个重要因素是确定切换到服务的标准

#### 4.8 测试法律法规

发现法律法规问题不可以修复软件来解决

##### 4.8.1 法律法规清单

- a) 隐私法律

《荷兰数据保护法》，《欧洲数据保护条例》，《欧盟个人数据保护指令》，《美国-欧盟安全港》

- b) 数据位置

云计算数据不是存储在一个地方，而是存放在不同的地方

- c) 附加协议
- d) 其他法律
- e) 其他业务
- f) 冲突的法律与规则
- g) 不清晰的法律与规则

#### 4.8.2 检查法律法规

- a) 如果风险高，可以执行审计
- b) 当理解的所有风险和可能的瓶颈后，高层管理者部分给予测试经理的建议，决定是否可以再组织中使用服务
- c) 测试与检查数据安全性是检查法律法规的基础工作

#### 4.9 在生产环境中的测试

##### 4.9.1 变更情况下生产的持续性

- a) 服务的变化
  - 有通知的更新
  - 得到关于变更的时候通知
  - 没有通知
- b) 其他系统的变更
- c) 供应商侧的变更
- d) 业务过程发生变化
- e) 增长：云计算的一个重要优势是其扩展服务容量以满足需求
- f) 连接服务的设备发生变化
- g) 因特网的变化

##### 4.9.2 度量供应商的承诺

- a) 可用性
- b) 性能
- c) 可扩展性

d)安全性

4.9.3 原有选型标准评估

4.9.4 实践中的注意事项

a) 端到端的回归测试：建议采用 CI

b) 关键指标的度量方法