微软的测试之道part3读后感——测试工具和系统

* 缺陷和测试用例管理

这一部分通过介绍微软广泛应用的缺陷跟踪系统和测试用例管理工具的信息，通过一些讨论和实践，并且介绍了基于成千上万个微软测试团队和测试工程师的工作经验。前辈们通过分享他们遇到的挑战和得到的教训来为我们指点迷津，打开微软测试之道的大门。

“如果在一个没有人的森林里，一棵树倒下了，它会发出声音吗？”言外之意是它发出或没发出声音，是没有人注意到的，这也是这本书里面提到的一句话，同样的道理，不论是客户还是开发工程师，或者测试工程师，知道有人运行代码，我们才会知道缺陷的存在。那么缺陷是什么意思呢？缺陷描述了测试用例的结果。对于软件测试工程师的最大工作成果之一就测试工程师通过检查这些代码和程序发现了这些缺陷。既然发现了缺陷，那么就要对缺陷进行跟踪。缺陷都是有生命周期的，一个软件缺陷的生命可以从软件设计开始，也可以从变成或者其他软件开发的相关过程里。我们可以通过很多种方式来发现和记录缺陷，并且由一个核心小组来审查代码。所以我们就需要撰写缺陷报告，因为缺陷报告可以准确的长期的记录软件的错误和错误处理的决策。缺陷报告也往往成为历史参考资料。在一些场合，缺陷报告可以成为法律辩护的依据。在微软，缺陷的标题也许是缺陷报告中最重要的信息，系统的临时用户可以扫描标题去了解产品中或某一特定领域的缺陷类型，标题也是缺陷系统中搜索最多的字段。高质量的缺陷报告是至关重要的，它能使缺陷会审小组做出正确的决定，并确保缺陷适当的处理路径和优先顺序。一个缺陷报告中也会出现很多的错误，如果处理不得当的话，会失去缺陷报告的价值。

说到这里我们根据书上讲的内容一起来看看到底什么是缺陷跟踪系统呢。缺陷跟踪系统是测试团队最开始必不可少的工具。成功的缺陷跟踪系统常有一些重要的特征。简单好用是缺陷跟中系统中最重要的要素之一。测试人员利用这个系统来记录缺陷。缺陷跟踪系统中重要的特征性质有这几个：可设置性，可靠性，缺陷通知，互操作性，外部用户访问。

* 测试自动化

测试自动化是很重要的，测试自动化提高了测试的效率，机器有时候也能比人眼跟精确，所以能更容易的发现缺陷。不过是否需要测试自动化，这在测试界，一直是一个具有争议的话题。对于有一些代码，可能只需要测试一遍，这个时候完全自己来完成这一项操作，可是如果是要测试一百次，一千次，甚至上万次呢，这个时候适当的采取自动化是十分必要的。自动化的价值就在于帮助测试人员减少繁复的工作压力。那么我们自然会想要了解的问题就是自动化测试中到底包括什么,好的自动化测试会利用计算机的能力去执行人力无法有效执行的测试，首先要能考虑到测试的执行步骤，在运行之前，程序必须处在能够执行测试的状态，在测试执行后，要知道测试是都通过，这三点是自动化测试中不可缺少的三点，其他还有一些细节的问题我们这里暂时先不讨论，我们着重来看一下尾端内部实现测试自动化的途径和方案，在微软一种普遍使用的方法是这样的，大多数团队会采用自动化的程序安装，但是会有一名专职的安装测试工程师来撞门验证整个安装设置测试矩阵。其实很多时候，有些测试工作看起来很复杂，很困难，几乎没有自动化的可能性，但是其实本书贯彻的主张是，不论你遇到任何的问题，要学会集思广益去找找看是否能让其中的一部分变得更有效率。执行测试用例各个步骤就是自动化测试的核心。有各种各样的方式来实现。今天的微软采取的解决方案是，使用测试用具test harnesss来运行自动化测试。测试用具其实就是每个程序运行适合，测试所需要的框架，一个好的测试用具跟一个好的程序是有异曲同工之妙的，要是可以进行扩展的，可以自行配置的。只有这样，才能说这样的测试工具是一个人性化的测试工具。构造成功的自动化测试系统能够让整个程序进行测试的时候，像是按一下开关一样，打开开关，便进行测试的工作。

* 非功能测试

根据书本的例子，我们可以知道，当你发布一款软件的时候，这个软件最初的评价是正面的，并且在开发过程中整个团队都合作的十分顺利，但是可能软件发行以后一段时间，开始有用户对软件有各种各样的意见，各种各样的需求，这样就迫使开发团队不停的开发一个又一个的补丁。所以在功能测试后，还有一个很重要的测试环节就是非功能测试。包括可靠性，可用性，可维护性，可移植性等等这些方面。还有属性测试，包括可依赖性，可测性，可扩展性，可适应性。微软测试团队经常会有专门的队伍负责这些属性测试甚至在可用性中分离出一个工程职种，有一个叫做可达性角色的名词，用来代表某类用户和他们如何使用我们产品的虚构人群。使用角色有利于产品团队集中精力设计和开发适合这些用户的功能。还有性能测试，压力测试，兼容性测试，可达性测试，可用性测试，安全测试等等各种类型的测试。所以说功能测试是十分重要的，然后用来有效完成功能测试的许多技术和方法也是同样重要的，很多用户并不会关心你到底找了多少个缺陷，这并不是用户关心的事，用户只是希望自己在使用一款app的时候可以放心大胆的使用，不用担心密码账户泄露等等的安全问题，并且可以用这款然间方便轻松的完成自己想做的事，在能接受的响应速度中达到自己需要通过这款软件达到的目标和功能，所以，非功能测试必不可少。对于一个软件是否能够成功，非功能测试面临着更大的挑战。不过最大的难点就在于，这些测试必须要在玉户使用过后，才能真正开始。

* 其他工具

相信大家都听过这样的故事，一个师傅有两个弟子，分别给了他们一人一把一样钝的镰刀，吩咐说，你们比比看用一天时间谁砍得柴多，一个弟子早早去了山上一直砍柴，但是另外一个弟子就在院子里细心的磨刀。在磨好刀后才上山砍柴，看起来好像是他耽搁了很多时间，但是实际上磨刀不误砍柴工。因为镰刀锋利很快就超过了另一个人，最后当然在这次师傅给的难题中，当了第一。所以工欲善其事，必先利其器。我们要很好的进行测试，那么一定要熟悉一些测试工具，才能更好的进行测试。微软的 Visual Studio Team System 定义了“改动项总数”(Total Churn)指标,该指标是增加的代码行数,删除的代码行数和修改的代码行数这三者的总和。源代码控制管理(跟踪源代码的所有变化)对于开发人员是非常重要的,在微软,它对测试人员也同等重要。在软件行业几乎所有的开发团队都要使用某种源代码管理(Source Code Management-SCM)系统,微软开发团队也不例外。在某种程度上,每个微软的测试团队不仅使用了传统意义上的源代码管理,还会把它应用于一些特定的测试任务。

在微软公司,各测试团队在源代码控制系统的使用上已经经历了各个阶段,也在不断的完善中。微软在早年的时候,很多测试团队的源代码控制至少是做得非常不正规的。当年作者在微软的时候，每个测试人员负责编译自己的测试源代码,同时要负责把编译产生的二进制代码文件复制到同一个共享的网络文件夹中,这样出现的问题就是如果有一个人删除了文件夹大家都会受到影响。微软的员工不仅仅善于使用工具，更喜爱创建工具。几千种不同的测试工具从辅助测试程序库到 Outlook 的插件等等都被放在一个大家可以轻易拿到的工具库里。微软的工程师们喜欢使用这些工具来帮助他们更好地工作,很多的人都愿意和本部门以及全公司的人共享他们的工具。员工们可以在这个工具库里(ToolBox)查找自己需要的工具。也可以预订 RSS 源,这样当新的工具被加进来的时候就可以得到通知。每一个工具都会列出关于这个工具的 名字、作者和用途等信息。而且每一个工具都能在工具库的游览器里被打分以及添加评论以帮助以后的使用者更好地做出选择使用哪个工具的决定。微软的员工完全可以自己开发测试工具来解决问题。所以一个丰富的工具库,加上正确使用工具的知识和技能,是测试工程师拥有的最大财富之一。

* 用户反馈系统

一款好的软件必然需要是用户友好的，一个软件的评判标准必须和用户是息息相关的，一个开发团队的软件即使通过了所有测试，但是如果没有深入到用户群体，没有通过用户的测试，那么这个软件也是失败的，在微软的用户反馈系统中，这本书主要介绍了微软公司搜集用户和合作伙伴数据的工具和技术，这些工具和技术能让我们的产品质量得到提升，也可以改进我们的测试方法。用户并不关心你这款软件的相关数据，在发布前修复了多少缺陷等等，用户需要的是简洁明了的说明和简单上手的操作。微软目前也是在努力寻找更好的测量方式来预测用户如何让看待质量，研究用户喜好，在如今大数据的时代，我认为进行有用数据的提取，在基数大的情况下去研究用户，会有一定的成效。这里的测试困难的地方在于每个用户反应的一些数据是带有主观色彩的，这使我们经常无法处理和理解所有数据。了解你的客户如何使用软件的最好方式，就是去观察他们的实际工作，可以去观察哪些功能他们最常使用，哪一些功能他们不常常使用，微软有一个计划叫做“用户经验改进计划”，在争取了用户的同意后，可以开启这个计划，当你的计算机闲置时，有关你使用程序方式的匿名资料，会尚在到微软公司去，在微软的内部，为了做出高质量的产品，需要尽最大的努力去了解产品。不过我比较担心的依旧是安全问题，如果我同意微软的这个计划，我的隐私和信息是否真正能得到很好的安全保证。微软还有错误报告这个形式，当你在windows系统中应用程序或者系统发生了故障，你就可以想微软公司发送一份报告。微软获得了很多这样的报告，并且简历起了很有效的数据管理系统，用于研究这些报告，了解哪些驱动程序有问题。还有微软的“发送微笑计划”，坦白说，我从来没见过这个，但是这个概念是很新的，确实是很多情绪并不能用文字很好的表达，但是图标可以。比尔盖茨在2003年专业开发者大会中说过这样一句话“今天，我们已经看到了许多驱动程序制造商所作出的改进，但我们希望在应用程序的层次能看到更多的工作，这回促进我们共同努力去改进用户还不满意的任何方面”

* 测试“软件加服务”

有很多面向服务的架构和软件作为服务的书已经出版了，这本书的作者建议读者能够度过某一本这一类的书，因为对仍和一个跟服务打交道的人来说，这个都是很重要的准备工作。软件加服务，在这里我私自把他称为双s服务，这是微软的服务战略，微软加服务是微软创造的属于，它展示了我们队分布式软件拥有巨大价值的信念。服务是整个软件生态系统中一个重要组成部分，也有很多人认为，做服务，才是今后做软件的主流。如今有三个准则在驱动着这个领域的根本转变，他们都在某种程度上与服务有关，这是2005年雷。奥奇提出的三点，第一要基于广告的经济模式的能力，第二要有新的推出和采纳模型的有效性，第三，要有令人瞩目的，集成化的，拿来就可以立即使用的用户体验。这样才可以从大规模成长为超级到规模，要跟着创新的浪潮。运营也是至关重要的一环。微软一直致力于为互联网开发产品和服务。在我看来，这个也是整个互联网发展应该需要坚持的准则。高质量的服务也是意味着我们需要不断的在经历的错误中，慢慢学习不断成长，在遗漏的缺陷中西区教训，所以一个软件的测试可以说是一直存在的，只要软件不消亡，软件的一生便是测试的一生，这一部分我的理解就是这样。