

《微软的软件测试之道》读书笔记

目录

目录.....	1
引言.....	2
第一部分：关于微软.....	2
第二部分：关于测试.....	3
第三部分：测试工具和系统.....	4
第四部分：关于未来.....	4
写在最后.....	6

引言

这本书的定位是非常明确而又清晰的。它包括了微软普遍的测试实践、工具和技术，在讲解这些专业性较强的内容的同时，也穿插了一些有意思的小故事。因此，无论是对于一个测试人员，还是对于一个对微软测试感兴趣的非测试人员，这本书都是具有很强的可读性的。

这本书分为了四个部分，介绍的内容循序渐进，使得读者更加容易理解。首先介绍了微软内部做测试的大致环境，如：产品、工程师、测试人员、测试的作用等等；紧接着，介绍了许多在微软中较常使用的测试工具和实践案例；第三部分介绍了“测试工具和系统”，较为细致专业；最后一部分探讨了未来微软在软件测试方面的展望及发展的方向。简单来说，这本书描述了微软怎么用很多已知的软件测试技术和工具来改进软件质量。

我们的读书笔记也将按照书中的思路来进行，分为四个部分，在每部分中简单介绍了书中内容及感想。

第一部分：关于微软

首先，讲一讲第一部分“关于微软”。

第一部分中，主要讲述了“微软的软件工程”，“微软的软件测试工程师”以及“工程生命周期”这三方面。

首先，本书在开篇介绍了微软年会的盛况，在其笔下的微软是一个充满着活力与创造力的地方，书中介绍了微软的各个主要部门以及微软的组织结构。而在于微软这样大型的软件工程公司，有条不紊地完成各项工作无疑是个技术活，截止 2008 年，微软在总员工数达到了 10 万，这是一个庞大的数字。因此，微软主要有两个模式来组织工程师队伍，分别是：产品部门经理模式和共享团队模式，这两个模式在一个产品的开发过程中是可以相互转换的，目的是统一的，做出更好的产品，书中对于微软工程队伍和产品组织给了一个很好的比喻：像创作戏剧一样研制和发布软件产品，还类比了爱丽丝梦游仙境这个奇妙的童话故事。

第一部分中，我印象最深刻的应该是比尔盖茨的思考周——每两年一次公司大小小工程师提交自己的思考白皮书，比尔盖茨会亲自审阅并亲手批注，好的 idea 会被全公司共享甚至成为现实。这个故事对于任何人都是触动极大的，这应该也是微软能够保持创造力和活力的原因之一。而在这个部分，这本书还介绍了微软的职能种类，以及微软的全球化运营，其中在中国和印度的从业人员比例是不断上涨的，作为一个中国人，看到还是觉得非常有意思的。这体现出微软发展的全球化趋势。

读完这一部分，我还了解到软件测试工程师在微软中重要的工作职能和可观的职业发展规划等。对于微软来说，软件测试工程师与软件开发工程师的人员比例甚至可以达到 1:1 之多，足以见得微软对软件测试的重视。而对于微软 SDET（软件开发测试工程师）的职业发展方向，微软实行管理和技术轨道交叉发展的模式。对于一名测试工程师来说，你可以选择成为一名测试架构师，测试独立贡献者或者测试管理人员，尽管成为一名管理人员并不意味着升职。

在我看来，与平时课本上了解到的最最基本的瀑布模型、螺旋模型、敏捷开发等模型不同的是，作为一个商业化的企业，微软的开发模型要更加的高效便捷，这在后文中会详细叙述。可靠的开发流程保证了软件的质量与品质。书中将做软件类比做饭，这个有着固定模式但又需要不断调整容许不断尝试的过程，私以为这是十分恰当的，也许做软件的过程并不能说是最好的，但显然，微软是不断追求做好的软件的。这种态度也是我们在软件开发的过程中需要努力实践的，我们不应该只追求走完软件开发的流程，而应该在其中不断精益求精，

产出更优质、能造福社会的作品。

第二部分：关于测试

这部分首先讲的便是软件测试用例设计的实用方法，其中不乏一些我知道的方法：比如黑盒白盒测试、用好坏数据进行测试等等，但是更多的诸如软件测试模式、测试时间估计、测试可行性，甚至还有配对测试等，这些对于一个刚刚接触软件测试的人来说都是很新奇而且有趣的，在阅读这部分的时候也感觉在专业知识方面大有收获。更加重要的是，这里所讲述的一切都告诉我们，没有完美的技术可以保证完全优秀的测试，一个优秀的软件测试工程师必须要有一双善于发现的眼睛和一颗勤于思考的心。

作者从四个方面依次介绍了测试的相关技术，分别是功能性测试、结构测试技术、利用代码复杂度分析风险以及基于模型的测试。

关于功能性测试，作者结合一些典型的例子，介绍了这样一些技术：边界值分析、等价类划分、组合分析以及状态转换测试等等。这些技术系统的分析了一个程序的功能性组件，增加了测试的覆盖比例，不同技术的使用有助于暴露不同的、特定类型的缺陷，这些技术大大的增加了测试的有效性，减少了测试的冗余性，也更加的科学可靠。

作者通过一个关于苹果颜色的小故事，展开了对结构测试技术的讨论，生动的阐明软件隐藏在用户界面之下的计算特征。而结构测试技术通过对函数控制流程的详细分析帮助降低风险，这样的测试对于研究分析程序的部分代码很有效。这属于一种白盒测试设计方法。不同于行为、探索性测试，这种方法不是评估项目外在体验而是真正对于内部问题进行剖析。这里面提到了老师上课所讲的一些方法：比如块测试、决策测试、条件测试、基础路径测试等等，有其思路以及一些具体的代码实现、分析。并且就像老师上课时所讲的，也对这些不同的测试方法进行了一番优劣、包含比较，令人获益匪浅。在这里，作者也指出，行为测试方法、系统化的功能测试方法、结构测试方法是从不同方面根据不同的信息来评测软件，并不能够说某一个是不必要的。

我最喜欢的部分，是用代码复杂度分析风险这一篇章。软件的风险评估与代码复杂度息息相关，而代码复杂度的测定可以根据：代码行数、贿赂复杂度、Halstead 度量、面向对象的度量等，对于测定了复杂度的代码，也会产生误诊以及偏颇的认知，对于复杂但清晰已维护少缺陷的代码，易出现误诊，而对于真心复杂的新代码而言，不妨尝试对代码重新分解。而代码复杂度是识别应用程序中可能存在缺陷的一种基本度量，对于识别代码维护性也有同等价值。

而基于模型的测试强调了模型对于复杂问题简单化、形象化的重要作用，这些介绍过程中用到的大多是数据结构和编译原理的知识，强调了运用。微软目前使用基于模型的几种测试工具有 Spec Explorer、MBT，他们声明，建模的意义在于使用，而任何类型的建模都是有意义的，这样的测试是很强大的，有助于理解复杂的系统、帮助管理风险以及寻找漏洞。具有很强的实际操作意义。

这部分内容引发了我对软件测试技术的一些思考。对于软件的测试，要有的放矢，学会筛选，有针对性的投入测试力量，更加高效、合理的进行。身为一个软件开发测试工程师，所要掌握的知识技能不仅要广，更要精，需要多维度的进行思考与实践，才能保证更好的软件质量，提供更优质的软件服务。这对任何一个对测试工作抱有兴趣的人来说，都是极富挑战性的工作。除此之外，我也意识到，要做一名合格并且优秀的软件开发测试工程师，是需要良好的专业素养的。以微软为例，所有的软件开发测试工程师都是具有计算机科学的知识背景的，而不是我们一般意义上理解的“用户”，测试工程师不仅仅是简单的使用软件来判断现有开发成果中是否存在缺陷，他们更需要缜密的逻辑思维与良好的编码能力，从各个角

度来“审判”一个软件，更全面的找出软件的潜在问题，交付开发工程师来优化。

第三部分：测试工具和系统

这一部分介绍了微软目前广泛应用的缺陷跟踪系统和测试用例管理工具，通过一系列的讨论和实践，介绍了诸多微软测试团队和资深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，让我对软件测试有了更深一步的理解。

书中有这样一句话，“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？”言外之意是无论它是否发出声音，都没有人会注意到。同样的道理，不论是客户还是开发工程师，或者测试工程师，如果软件没有被运行，那么缺陷永远不会被发现。

缺陷跟踪系统是测试团队最开始必不可少的工具。测试人员利用这个系统来记录缺陷。这里就引出了测试自动化。这个概念是很重要的，它提高了测试的效率，机器工作无疑比人眼工作更精确高效。不过，是否需要自动化的测试依情况而定，微软也有自己的一套标准。今天的微软采取的解决方案是——使用测试用具 `test harness` 来运行自动化测试。一个好的测试用具跟一个好的程序是有异曲同工之妙的，它需要具有良好扩展性，也要易于配置。

作者还提到，测试的范畴不仅局限于测试软件本身功能是否正常，还应包括投入市场后是否具有良好用户体验与反馈。

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。书中给我们介绍了微软的很多测试工具，比如改动项总数(`Total Churn`)、源代码管理(`Source Code Management-SCM`)系统。通过利用这些工具，微软保证了高效、优质、系统化的测试水准。

在微软，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，全公司的人都在其中分享自己工作时所需要的各种工具，其中不乏员工自己开发的成熟测试工具。这样一个丰富的工具库是测试工程师拥有的最大财富之一。

微软目前正努力寻找更好的测量方式来分析用户如何看待质量，从中研究用户喜好。他们开展了“用户体验改进计划”，在获取了用户的同意后，用户使用程序方式的匿名资料，会上载到微软公司，供员工分析。部分用户虽然存在安全疑虑，不过已经能够充分体现出微软对软件质量的极大重视。微软类似的策略还很多，诸如错误报告、“发送微笑计划”等。

基于这部分的阅读，我对软件测试的内涵有了更进一步的理解。从微软对软件测试的态度看，软件测试包括两个方面。其一，是软件本身的质量，意即软件目标功能是否实现，软件性能是否达到预期目标等一系列硬性标准。其二，也是我们常常忽略的一点，软件测试也包括对软件服务质量的考核，软件测试的生命周期应该与软件的生存周期是吻合的，只要用户还在使用软件，那么测试就永不停息。还需要提到的是，软件测试的方法不应该是单一的，既不能局限于低效的人工测试，也不能过分依赖自动化测试。更进一步讲，各种测试工具在测试过程中的重要作用是不能被忽视的，合理的使用能使测试工作事半功倍，不仅减轻了测试人员的工作压力，也提高了测试效率，缩短了整个软件开发的周期。

第四部分：关于未来

在这部分中，作者谈及，与软件开发相比较，软件测试这一行业并未成熟，还属于一个相对较新的领域，它仍旧处于不断发展的过程中。而行业本身也是一个不断遇到问题，解决问题的过程重复。

作者提到，如果只是在问题发生时采取思考并研发解决的措施显然是为时已晚。因此有必要在“今天”预见到“明天”的问题，并提前思考解决方式，由此才能使测试的艺术，工艺和科学继续进步和发展。

书中从自动失败分析、机器虚拟技术、代码评审和检验和工具等四个方面，谈论了微软目前面临的测试问题，以及测试工程师们为了解决他们而选择的方向。

以自动失败分析来论。问题的由来是小小的失败率（类似 2%）就可能导致足够多的调查而引起“分析瘫痪”，这样可能令一个测试团队需要花与运行测试同样多的时间来调查测试失败。从自动化测试的思路延伸开来，不难想到自动化失败分析这一点。而在自动化失败分析中，最重要的技术实现集中在失败匹配上，意即建立“失败数据库”，详细存储每一次测试失败的原因，做好失败日志记录实践。

说到这里，就不难理解一个具有好的结构的日志记录对失败匹配，乃至对于失败分析自动化的重要性。高质量和一致的日志实践能区别“垃圾”自动化测试和能运行十年甚至更长时间的测试。微软的日志实践给我们提供了这样一个思路：

1. 日志成功时需简短，而失败时需极其详细；
2. 当测试失败时，记下所看到的失败前的成功操作；
3. 日志应跟踪产品信息；

.....

在这一部分的介绍的最后，作者提到了自动失败分析集成化的重要性。大型测试的自动化要想有非常好的投资回报，就必须理解集成对于工作的重要性，尽可能少的减少手动干涉分析测试结果和失败的需要。而集成化的另一个不得不提的有点就是，能够分析测试失败的走势，例如可能指出参数验证错误占测试总错误的百分比，或是计时问题所占的百分比。这样一来，分析产品风险、了解产品经验就更加得心应手。

除了提前预见问题并解决外，微软测试站在了测试行业的制高点，分析了软件测试行业当前的作用（提高软件质量），并且提出了这样一个问题——软件测试的下一步应该向何处走？书中从三个方面来回答了这个问题：前瞻性思考的需求、测试领域的领导力、卓越测试、为未来创新。

通读这一部分，不难得出这样的结论。

将软件质量方在产品周期的最后，或期待质量发生在产品周期的最后是愚蠢的。如果想从测试中得到高质量的产品，首先应该有高质量的产品用来测试。因此，我们应该正视这样一个难题：如何才能得到较高质量的产品。质量成本这一概念在论述中就不得不提及了，这是个被广泛误解的词，大多数人认为它就以为着生产优质产品或服务的花费，但事实上，质量成本以为这生产优质产品或服务的代价，每次需要返工时，质量成本就提高了。因此在未来发展中，还需要不断的思考，究竟怎样才能降低质量成品的同时提高软件质量。

那么，在未来的发展中，软件测试到底会演变成一个怎样的角色呢？书中这样回答我们。在未来的发展过程中，质量将会在设计阶段被充分考虑并集成，直接达到大大减少软件缺陷的目的。这样一来，测试团队的工作重心将不再是查找软件缺陷，而将转移到集中模拟用户场景、同事评审和校验方面。

微软在 03 年创建卓越工程团队，宗旨除了技术培训外，是发现和分享整个公司的工程最佳做法。卓越测试团队认为微软的每一个测试人员都是他们的客户，测试团体整体上的成功时这个团队的驱动力。而这个团队最关键的角色就是预计微软测试工程师未来的需求，并积极主动的确定关于未来软件测试的长远规划。

未来的世界，软件将无处不在，处处渗透我们的生活，影响我们生活的方方面面。而软件的质量将被希望是“零问题”，这是一个困难的过程，作者这样提到。

写在最后

读完这本书，受益匪浅。对微软的测试团队、测试过程、过程中使用的工具和系统以及他们对未来的展望都有了一个简单直观的理解。一方面对微软这个团队的视野与技术前瞻性感到赞叹，另一方面也重新理解了软件测试在软件开发过程中的重要地位，而最最重要的，应该是纠正了从前对软件测试的一些误解与偏见。很庆幸自己在这个时间读到了这本书，也希望能有更多读者与我共同交流对书中观点的理解。