

炫目的敏捷架构师

随着敏捷软件开发的理念和方法论逐渐被越来越多的人接受，敏捷架构师在团队中的地位也越来越重要，那怎么样才能成为一名优秀的敏捷架构师呢？

■ 文 / Vikas Hazrati 译 / 舒克

一直以来，无论是在软件开发组织之内，或是行业广大人士之中，对于敏捷团队是否需要架构师一直存在争论。大家的质疑集中在：既然软件的架构是随着每个迭代而演进的，那一个架构师还能给敏捷项目带来哪些价值呢？这让许多传统的架构师都感受到了威胁，并力图寻找掩护，也为一种新类型的架构师——敏捷架构师——打开了机会的大门。在敏捷项目中，传统架构师的象牙塔已经逐渐成为最薄弱的一环，而他们的许多工作职责也已经被整个敏捷团队所分解。敏捷架构师的出现，正符合了查尔斯·达尔文的“适者生存”理论。在一个团队中，敏捷架构师角色的重要性是毋庸置疑的，而且许多敏捷团队都认为他是任何敏捷软件开发团队中最有价值的成员之一。

那怎么样才算敏捷架构师呢？我们如何识别团队的架构师是否是敏捷架构师呢？嗯，答案其实并不简单；不过敏捷架构师确实具备与众不同的品质，而且这些品质会体现在他们在每天的日常工作之中。如果团队中的架构师能够体现这些品质，那么他肯定就是一个很好的敏捷架构师。不过首先我要警告你：下面的描述听起来也许有些过于理想化了，但他们确实描绘出了敏捷架构师应该达到的境界。**敏捷架构师的首要目标：以最优质量交付可用的解决方案。**

这里有两个很重要的方面。首先：解决方案的质量应该是最优的。理想状态下，质量的需求（也被称为非功能性需求）在敏捷项目的每个迭代中都要有所体现。像安全、性能、代码质量等等这些方面会隐式地包含在一个用户故事之中，或者作为一个单独的用户故事在迭代中出现。然而，在迭代中工作的团队经常会陷入到对业务功能的开发中，而质量方面的要求

置于可工作的解决方案之上，而不是对业务价值没有多大贡献的文档和管理层要求提交的某些可交付物上。

维护概念完整性

他要确保在硝烟四起的战场上，任务能够得以顺利完成。在实施时间表和技术障碍的双重压力下工作，开发人员有些时候做出的决策会让项目偏离原有的业务目标。架构师必须注

对敏捷架构师来说，“可以工作的软件重于求全责备的文档”和“人和交互重于过程和工具”这两条敏捷宣言尤为重要。

就被抛在脑后了。敏捷架构师要保证团队在每次进行构建时，都能从持续集成环境中获得关于静态代码质量、性能统计数据等反馈。他还要注意监控这些统计数据和质量属性，并不断提起团队对它们的注意。

其次，解决方案必须是可以正常工作的。敏捷架构师要与团队评估众多不同的选择，并一起交付一个可以工作并且能够产生业务价值的解决方案。他必须要确保解决方案不只能够独立运行，而且可以与客户现有整个软件环境进行良好的集成。这个解决方案必须足够健壮，可以随着时间变化进行变更和扩展。他必须将注意力

意这种状况，确保任务在整个开发过程中不会受到负面影响。

当系统展现出这样的一致性——所有的东西看起来都可以无缝结合，这样的系统就具备了概念上的完整性。它是通过不同部分中一致性的展现和非理性设计的缺失来体现的。在大多数操作系统中的拖放操作就是一个很好的例子。如果系统在设计时遵循了概念上的完整性，那么拖放操作在任何时候、任何地方都应该是可以运作的。例如，要打开一个文件，可以通过将该文件拖放到合适的应用程序中实现；删除文件可以通过将其拖放到垃圾桶中完成。简单来说，应用概念



敏捷架构师：敏捷架构师要具备扮演“协调者”和“万能胶”角色的能力

完整性可以创造一个易于使用、易于维护，而且在使用中不会给用户带来过多怪异和不便的系统。

与团队一起工作

他会参与项目整个过程。会像其他程序员一样领取开发任务、实现用户故事。类似工作不会占用他一整天的时间，却是每天的固定工作。

他会在整个项目中运用自己的经验，并指导架构完成演化。在最初的几个迭代中，架构是不会完成的，它会随着每个迭代的进行逐渐演进。这就意味着：敏捷架构师与传统的架构师不同，当所谓的架构设计阶段完成后，他也不会转向全新的项目。

编写系统级别的测试

他会撰写底层的测试来检查系统的架构。如果架构上存在漏洞，失败的测试就会将其暴露出来。由于敏捷主张测试优先，他会编写使系统在质量特性方面失败的测试，并修改架构的某些相关部分来通过测试。少数情况下，用代码进行测试也许无法起到作用，他会想出其他的方式来“测试”架构。这可以确保不会在未经证实的设计上耗费精力。任何设计上的决策，无论大小，

必须经过测试用例的检验。

例如，如果要求系统达到同时为一万个用户提供服务的质量要求，敏捷架构师会使用 JMeter 或类似的压力测试脚本，以模拟足够的负载，确保系统可以承担这样的压力。如果测试失败，他会重构架构以通过测试。

参与紧密的协作

他不会单独完成设计和技术上的决策；这些决策是与团队共同协作完成的。他拥有开放的心态，并且坚定认为自己不是全知全能的万事通，而且不会对自己的主意保有盲目的保守心态。当团队坐在一起进行头脑风暴时，就会产生最好的主意；任何人都可能提供好的想法和有价值的见识。开放和诚实的沟通可以帮助人们基于更好的信息做出更好的决策。敏捷架构师是一个善于与人打交道的人，他对任何人都充满敬意，并且会利用每一个机会向别人学习。他会促进共识的形成，并欢迎团队中每一个人为解决方案做出贡献。

如果业务发起者在解释完项目的目标之后，架构师马上就可以提出一个解决方案，那么这个架构师也许不是一个真正的架构师。

做坚定的指导者

他经常与团队一起工作，来提升团队的决策制定能力。根本上，这可以帮助架构师更好地利用大家的智慧，而不是成为项目中单一的决策制定者。通过指导团队使用新的技术、为技术架构构建并提供支持，他与团队可以更好地形成协作与互利关系。他可以确保团队不会因为缺少某些经验或是对某些技术感到不适而简单地拒绝一种设计方案。如果真的发生了类似情况，他会为团队弥补中间的隔阂，从而让大家可以做出更加明智的决策。

正如 Martin Fowler 指出：“这指出了如下这条基本原则：一个架构师的价值与其所做出的决策数目成反比。”

做熟练的协调者

敏捷团队中包括技能熟练、富有激情的团队成员。对于设计决策、实现功能的最佳方式、开发流程的改进措施，团队中经常会出现不同意见。还会带来激烈的争吵，如果不能迅速妥善处理，会影响成员之间彼此的感情。这些场合下，团队成员需要一个拥有丰富经验、足够成熟、并且见多识广的调解人，来帮助团队打破僵局，重新和谐。而敏捷架构师由于其成熟的水平和经验就成为了不二之选。

不做大型的预先建模

通常他不会使用重型的 CASE 工具来做大型的建模设计和决策。一般来说，所有的建模都是“足够”的建模，并且是在白板上完成的。简单的模型会随着每个迭代慢慢成长、逐步改进。他会跟随敏捷建模的概念，对他来说“代码就是模型”。如果需要为了文档生成模型，那么这些模型也是通过一种合适的逆向工程工具从代码生成的。

模型、文档、沟通的方式会以尽可能简洁扼要的方式保存。关键是要将注意力放在内容上，而不是展现或其他为了看起来好看而作的事情，这些东西并不能添加多少业务价值。

寻找大规模重构的机会

他总是在注意观察可能成为严重技术障碍、损害项目架构的问题。随着系统的成长，他确保架构可以跟上发展的脚步。他总是在注意观察可以取得更大收效的变革。举例来说，如果应用程序没有利用足够的CPU，并导致性能低下，那么他可能会试图引入多线程，以使得空闲的资源可以被利用起来提供更好的性能表现。他会首先选择最简单的可用解决方案，接下来会经常对其进行重构，以满足变化的质量和功能需求的要求。

敏捷架构师同样会帮助切分系统。他会推荐采用“先征服再切分(conquer and divide)”而不是“先切分再征服(divide and conquer)”的方式。最开始时，一个小系统是通过小分队实现的。最终，随着架构师经验的不断丰富，团队会将系统按照自然形成的界线进行切分，并且始终在心中包含有系统的整体概念，而且为系统每个不同的部分增加团队成员。

敏捷架构师是万能胶

他会在团队成员之间、不同的厂商之间以及利益相关者之间逐步沟通架构、设计决策、任何技术障碍之类的东西，来充当万能胶的角色。他是业务世界和技术世界之间的媒介。他能够在业务上下文中通过技术的角度和问题发现潜藏的业务问题。他能够以业务价值的角度解释当前的采纳技术，让利益相关者明白这些技术的价值所在。例如，他可以使用人员利用率、更快的周转率、节省的资金等业务价值用语，来说明为什么在两种技术中选择其中一种。针对某个项目时，他会同时思考如何回答“为什么”和“怎么做”这两个问题，以回答业务和技术上的疑问。大多数传统的架构师会关注如何实现一个解决方案，而不去思考为什么这个解决方案是最佳的。敏捷架构师会试图与业务人员一起，并找出为什么需要某种特别的解决方

案，是否有比业务人员提出的建议更简单的方式解决手上的问题？

最后但并不是最不重要的，敏捷架构师拥抱变化。

他拥抱架构和角色上的变化。真正架构上的敏捷度，是指快速应变而不损害架构的能力，同时对其他部分的影响尽可能小。要确保这一点，他要从好的设计开始，并让其随着项目的进展而演进。他常常会抽象出通用的元素，并将其封装在稳定的接口之后，因此将变化带来的影响最小化。

真正的角色敏捷度是可以扮演多个角色。在一天的工作中，他可能从事程序员、系统测试人员、指导者、协调者、团队成员、万能胶，甚至有可能是管理层等多种角色的工作。

结语

架构师象牙塔的根基已经被动摇了。新一类的“敏捷架构师”正在逐步涌现出来，并与团队一起工作，为团队做贡献。他们确实是在按照丰田原则【注】的第十二条在生活：“亲临现场查看，通过一手资料来彻底了解情况”。他们是开发团队的第一等公民，而且不只为团队工作，还在团队之中工作。因此，要想成为能够与开发团队共进退的敏捷架构师，成熟、经验以及上述的能力不可或缺。

注：美国密西根大学教授，杰弗里·耐克尔（Jeffrey Liker）根据观察和研究，撰写出了关于丰田汽车公司管理的14条原则。■

作者简介



Vikas Hazrati, 在金融、制造、制药、健康等行业有多年的开发和项目管理经验。目前在Xebia公司担任首席咨询师。曾在多个组织中参与基于敏捷的高效开发过程和环境的搭建。

■ 责任编辑:郑柯 (zhengke@csdn.net)

为什么? 他们选择 msup

因为我们能够帮您建立一支
—— 高效国际化研发团队

详情登陆: www.msup.com.cn

我们的使命

- ▲ 为整体提升中国软件业研发管理水平;
- ▲ 为搭建中国与全球软件精英之间交流平台;
- ▲ 为使中国软件业发展成为中国未来的核心产业;
- ▲ 为使中国未来能够出现像微软一样伟大的软件公司!

部分国内知名企业定制课程精彩案例

- ★ 北京金山软件有限公司 微软软件测试定制培训
- ★ 趋势科技(中国)有限公司 SQL Server2005的超强开发特性定制内训
- ★ 金蝶软件(中国)有限公司 软件测试人员定制内训
- ★ 阿里巴巴旗下淘宝和支付宝联合 软件自动化测试定制内训
- ★ 上海惠普有限公司软件服务分公司 微软Biztalk实践定制内训
- ★ 东软集团 Windows Vista培训
- ★ UT斯达康通讯有限公司深圳分公司 微软卓越软件测试人员定制培训
- ★ 中国工商银行珠海软件开发中心 微软软件测试人员定制内训
- ★ 北京中软国际信息技术有限公司 微软Project IT项目管理、应用培训
- ★ 深圳中兴通讯股份有限公司 软件测试工程师定制内训
- ★ 上海普元信息技术有限责任公司 软件测试和项目管理

麦思博(msup)有限公司
北京市朝阳区霄云路现代汽车大厦17层
电话: 010-6410 8466
传真: 010-6410 8466 转 605

msup全国服务热线: 800-818-8020
详情登陆: www.msup.com.cn