

<<敏捷开发一千零一夜>>

图书基本信息

书名：<<敏捷开发一千零一夜>>

13位ISBN编号：9787121208188

10位ISBN编号：7121208180

出版时间：2013-8

出版时间：电子工业出版社

作者：王立杰

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

<<敏捷开发一千零一夜>>

前言

前些日子，有人在微博上发文说，现在IT界有三大俗：云计算、大数据、敏捷，这从一个侧面反映出了敏捷的热度！

如果你最近一年内参加过各类IT过程改进大会、测试大会、研发管理大会、案例研讨大会，一定会发现有很多人在讲敏捷、谈敏捷，而且占据的场次越来越多。

以前是我们这些所谓的敏捷狂热分子在讲，现在连一些CMMI / PMP死硬分子也加入进来讲。

敏捷让以前的CMMI / PMP阵营不得不做出调整，PMP搞了敏捷的认证，CMMI新出了1.3版，据说里面有97处提到了敏捷……这从另外的角度说明敏捷的确已经入“俗”。

中国人素来讲究雅俗共赏，在这个越来越“俗”的过程中，我们该怎么保持那份“雅”呢？

希望大家在看热闹、参与热闹的同时，能够冷静地思考，敏捷到底能为我们带来什么？

我们该如何变得敏捷？

敏捷又将何去何从？

大师们也在思考，2011年，敏捷10年回顾的时候，将“追求技术卓越”放在了第一的位置；10年之后，敏捷又将如何呢？

著名敏捷大师Mike Cohn认为：“在接下来的10年，面向对象世界中发生的事情会再次出现，即我们将不再讨论敏捷。

不久前我们不再讨论对象了，因为它们已经胜出，没有人再会参加针对面向对象的大型辩论。

当然，还有一些应用我们不使用对象，比如有严格性能要求的应用，也有些项目是用非OO语言开发的。

但即使在这些案例中，我也怀疑开发的代码仍然受到对象的影响。

我希望敏捷也能达到这一点，我们不再讨论敏捷，不再说‘敏捷软件开发’，我们仅仅说‘软件开发’，当然一定是敏捷的。

没有人会问我编写的Ruby代码是否面向对象，因为这毋庸置疑；我希望某一天也没有人问我项目中是否使用了敏捷，这也将会毋庸置疑。

”在过去几年中，我们给客户做敏捷培训时，收集到的最多的反馈之一，莫过于学员想听到更多的敏捷实施案例。

于是，慢慢地就有了收集各类敏捷实施案例的想法，并日渐强烈起来。

毕竟，每个人都有偷窥别人隐私的好奇心，不然Twitter、微博也就不会这么火了。

同理，正在实施敏捷的人，无论自己做得好的，做得不好的，其实都希望看到同行是怎么做的。

冯国馨，作为PMBAR IT项目管理公益实践社区发起人，在社区内外的各种敏捷分享和交流过程中，看到大家也非常希望了解、学习和探讨国内不同IT行业、不同特点IT企业敏捷开发的真实案例。

加之，2011年8月PMBAR社区推出的第一部实践案例书籍《IT项目管理那些事儿》在社会上反响热烈，至今图书热卖万余册。

这些极大地增强了我们进一步开展敏捷实践案例书籍撰写的信心！

于是，在王立杰的协调下，PMBAR社区内外的十几位敏捷专家决定共同分享自己的敏捷实践案例，希望通过出书使行业内的敏捷实践者或敏捷观望者有所借鉴和体悟。

本书采用的是叙事的风格，通过分享每个敏捷实践者自身的实践案例，为读者展现一个个鲜活的敏捷实施过程，展示敏捷团队的成长烦恼和喜悦，从而和读者达到共鸣。

这是一个百花齐放、精彩纷呈的故事集，就像《一千零一夜》一样，有十几位作者，他们中有来自国际著名IT公司的敏捷专家、有来自国内著名电商的敏捷教练、有来自大型国有企业的IT总监、有民营高科技企业技术管理者，有纯粹的敏捷实践者、也有CMMI / 敏捷融合实践者等，他们以不同的经历、用不同风格的言语，为我们描述一个又一个精彩的敏捷开发故事。

感谢本书编委会全体成员（王立杰、冯国馨、许舟平、陈勇、杨立东、张克强、杨瑞、杜伟忠、郑贺宸、蔡阿彬、胡峰、范佳莹、马越、史昀）的共同努力，因为他们的努力才有了这本书的诞生。

感谢电子工业出版社博文视点的张月萍，她的“博文视点除了盈利的任务外，还承担着教育用户、传播思想的重任！

<<敏捷开发一千零一夜>>

”一句话为《敏捷开发一千零一夜》这本书画龙点睛。

我们希望这本书能启迪正在或者准备开展敏捷实践的读者，如果它帮助你加快了敏捷步伐，那就是我们最大的欣慰。

再次，真心感谢曾经为此书劳心劳力的各位朋友，愿天下人幸福安康。

编者

<<敏捷开发一千零一夜>>

内容概要

本书以多位作者的亲身经历，再现真实的敏捷实施过程，描述各个企业在实施敏捷的过程中，遇到的种种问题、解决思路及最终得到的经验与教训。

这本案集从不同的视角，为读者展示从大型互联网企业到初创公司、从大型国企到独资外企、从典型的甲方到第三方咨询公司的敏捷历程。

这里面既有大的组织的敏捷转型，也有一个小团队或个人的敏捷历程，还涵盖某个敏捷实践或工具的应用描述。

本书的特色在于由真实实践提炼，对正在实施敏捷的读者具有很高的参考价值。

<<敏捷开发一千零一夜>>

作者简介

陈勇：在工作的17年间，曾任其程序员、项目经理、事业部总监、副总经理、咨询师等各种技术与管理岗位。

开发中获得的一线工程经验和CMMI / FPA功能点估算 / 敏捷开发等跨领域知识，令其可以更广的视角来理解敏捷开发。

当前他作为产品经理、架构师带领一个小型团队，从事“火星人敏捷开发在线平台”的研发工作。

很多课程与咨询中的最佳实践，均来自于其之前及当前参与的实际项目的一线实践。

本书收录的文章就是他们在实际开发过程中，同时应用敏捷开发的用户故事和功能点估算中的功能点时的实际经验总结。

日常工作与培训之余，陈勇在其博客上总结编写了300多篇系列文章，点击量150万次，其中200篇以上是敏捷开发相关的文章，力求逐一解决敏捷开发中的一些似是而非的问题，为一线工作人员及敏捷推广者提供完整透彻的应用思路和方法。

杜伟忠：中国人民大学管理学硕士，京东商城敏捷教练，精通Scrum、Kanban、XP等敏捷实践。

10年电信软件开发经验，6年敏捷实践经验，其中有3年教练经历。

作为教练指导团队超过200人，目前专注于互联网公司敏捷的推广和实施。

冯国馨：网名“谷雨霖”。

天津大学工学博士、PMP 联合永道高级副总裁兼CTOPMBar IT项目管理实践公益社区创始人、天津大学北京校友会秘书长曾任神州数码网络研发中心副总经理、质量管理部总经理，现任联合永道高级副总裁兼CTO、联合创始人。

美国项目管理学会协会PMI会员、中国国家外专局资深专家、中国系统与软件过程改进协会主任专家会员、中国计算机学会软件工程师工作委员会专家组成员、国家应用软件产品质量监督检验中心特聘专家。

长期从事项目管理、产品研发、持续改进与团队管理，对教育信息化建设有一定见解。

创办IT项目管理实践公益社区，长期积极活跃在CSPI、CSDN CTO俱乐部、IT168等IT管理社区，与用友集团、阿里巴巴集团、盛拓传媒、搜狐集团、安博教育集团、神州数码集团等多家IT单位多次开展沙龙交流活动。

王立杰：敏捷爱好者、实践者，独立培训师，专注Scrum与XP。

2006年开始探索敏捷，应用敏捷，于2009年与许舟平合作出版国内第一本小说体敏捷项目管理图书《敏捷无敌》；2010年进入敏捷咨询培训领域，为诺基亚、北电、爱立信、VMWare、阿尔卡特-朗讯等多家公司提供服务，曾在AgileChina、ScrumGathering、AgileTour等行业会议发表多次主题演讲。

许舟平：本一北国布衣，求学于西，工作于都，躬耕于代码十余载，苟全技术于IT间，不求闻达于海内。

亲实践，远空谈，此敏捷之所以兴隆也；实践者，敏捷之本。

盖闻流程者莫高于Scrum，实践者莫高于XP，皆因敏捷而成名。

今天下贤者智能，岂特西洋者乎？

好友立杰，邀士十余，原始察终，见盛观衰，论考之行，历经春秋，终成此集。

敏捷者，其行虽不轨于瀑布、CMMI等，然其言必信，其行必果，已诺必诚。

周易曰：天下一致而百虑，同归而殊途。

窃以为，敏捷之事，内立法度，务实践，修研发之具；外连横而协客户，则事可成已。

至于XP、Scrum、Lean、Crystal等，各有所用，若欲循观其大旨，可阅此书。

布衣之人，不害于政，不妨百姓，取与以时而行敏捷，仅愿读者有采焉。

杨立东：PMP，美国加州州立大学富尔顿分校硕士，中欧国际工商学院EMBA，华中科技大学特聘教授。

北京暴风科技股份有限公司董事、首席技术官。

2011年第二批“中关村高端领军人才聚集工程”科技创新人才。

发表学术论文4篇。

<<敏捷开发一千零一夜>>

曾从事过6年一线Java开发工作，作为项目经理带领过多个大型软件项目，2006年开始从事CMMI咨询和评估工作，服务过的软件企业超过30家，多次举办项目管理，软件度量，敏捷项目管理的公开课，培训人数超过15000人。

2008年开始专注于公司级管理工作，2010年开始总结自己多年的研发管理经验，专注于互联网公司敏捷项目管理的推广和实施。

杨瑞：个人简介：厦门英睿信息科技有限公司联合创始人，目前致力于敏捷的推广，研发管理咨询、培训及移动开发。

10年软件工程管理经验，在基于传统和敏捷的开发管理方面具有丰富经验。

曾任三五互联产品技术中心总监、福州网龙程序中心高级经理等职位，擅长软件研发管理、过程改进咨询、培训及实施，精通CMM、CMMI及Scrum。

2011、2012年ScrumGathering分享嘉宾，2011、2012年AgileTour厦门站&福州站组织者、分享嘉宾，Agile China2013程序委员会专家，厦门敏捷社区发起人、技术社区积极分子。

张克强：思碧睿inspearit高级顾问，高级程序员、系统分析员、IPMP、CSM。

本硕毕业于清华大学。

曾在宝信软件担任过测试经理、EPG组长、项目总监，曾在Intel担任过QA Manager，曾在DNV担任过资深咨询师。

在软件工程/系统工程方面拥有10年多经验，主要经历在组织过程改进、质量保证和测试方面，帮助组织参照CMM/CMMI/Agile/Scrum等进行改进，在软件开发技术方法论和流程管理两方面都积累了丰富的经验，熟悉OOAD，UML，TDD，测试等等。

<<敏捷开发一千零一夜>>

书籍目录

第一篇 组织篇第1章 暴风敏捷项目管理实践2一、暴风的项目危机2二、组织级的敏捷初体验3三、再次组织级探索12四、持续优化的力量18五、组织改进中人的因素27第2章 淘宝的敏捷实践与过程改进35一、背景介绍35二、不同背景下的解决方案36三、ScrumMaster心得50四、敏捷与过程改进51五、工具的支持54第3章 从CMMI5到敏捷57一、案例背景57二、敏捷导入过程60三、敏捷优化改进过程70四、整体回顾81第4章 从装甲兵团到特种部队83一、引子83二、实施过程88三、反思95第二篇 产品篇第5章 火星人一千零一夜98一、第一个月：一个产品的诞生98二、第二个月：框架优先，还是故事群优先101三、第三个月：故事树103四、第四个月：用户故事的颗粒度（上）105五、第五个月：用户故事的颗粒度（下）108六、第六个月：用户故事的分类110七、第七个月：分类语法114八、后记115第6章 从敏捷到精益117一、背景117三、破窗理论121四、敏捷宣言错了吗125五、南辕北辙127六、MVP才是王道129七、跨越鸿沟135八、小结136第三篇 团队篇第7章 敏捷英雄传之火烧赤壁140一、人物介绍140二、故事梗概141三、引子142四、CEO孙权的故事：计划永远赶不上变化144五、CTO周瑜的故事：究竟是变好，还是变得更烂148六、产品经理太史慈的故事：一个大版本经理的困惑152七、编码狂人凌统的故事：主刀，就是能上得天堂，下得地狱的主儿157八、测试经理陆逊的故事：敏捷测试，就是明知不可为而为之161九、产品集成主管甘宁的故事：持续集成的烦恼162十、臭皮匠的话167第8章 打造学习型自适应团队168一、背景介绍168二、团队实践过程169三、回顾与反思181第9章 从传统软件开发到敏捷的初体验183一、背景183二、迈出第一步187三、对第一次迭代的改进192四、关键的第三次迭代196五、第四次迭代：低耦合软件设计197六、总结199第10章 敏捷在传统软件与互联网中的应用200一、背景介绍200二、敏捷实施过程201三、回顾与反思207第四篇 实践篇第11章 敏捷无它，唯持续改进210一、背景介绍210二、我与敏捷的第一次亲密接触210三、敏捷原来是这样的211四、第一个挑战213五、团队的好奇心213六、更多挑战214七、团队的惯性215八、镜子215九、从一开始就要高标准216十、TDD的争议216十一、“道”与“术”217十二、程序员文化217十三、程序员与建筑工人218十四、TDD不是目的，“拉”与“推”218十五、Coding Dojo219十六、让实践落地219十七、程序员的产出220十八、权威220十九、认同权的建立：无私，勇于说不知道221二十、成就感：点燃程序员的热情221二十一、PDD：痛苦驱动开发222二十二、排除障碍，创建舒适的技术环境223二十三、投资回报率223二十四、让领域模型裸奔224二十五、架构224二十六、你做什么就是什么（You are what you do）225二十七、Scrum是不行的，如果只有Scrum225二十八、做正确的事情 vs 正确地做事情226二十九、问题和解决方案，5-whys226三十、“为什么”227三十一、估算（动词）很有帮助，但估算（名词）往往没有228三十二、守破离228三十三、测试优先228三十四、QA vs QC229三十五、分享：Wiki、博客、书籍、技术讨论、编程练习229三十六、没有银弹，只有持续改进230三十七、敏捷宣言230第12章 网龙持续集成实践231一、案例背景231二、案例分享232

<<敏捷开发一千零一夜>>

章节摘录

1. 展示板和信息系统的巧妙结合敏捷例会的展示板每个团队都会实施得不一样。

对于进度跟踪,可以采用观察燃尽图的趋势,也可以融合信息系统的做法。

只要不苛刻地挑剔,很多工具都支持Scrum敏捷项目管理,从最简单的白板加Excel,到ScrumWiki、Xplanner、GNATS都能支持Scrum的实施。

暴风的团队开始就尝试了Redmine这个缺陷管理工具来配合展示板实施敏捷项目管理。

没有好不好用的系统,只要运用合理,抱着极简的思维去使用工具,就能最低成本地发挥工具的作用,至于使用信息系统遇到的各种缺陷或者不能满足管理的功能,还是应该报以宽容的态度。

但理想和现实往往经常事与愿违,将就用了一段时间后团队终于不能忍受系统功能的不足,于是开发自己的信息系统的呼声愈演愈烈,最终还是决定开发自己的系统。

还是本着极简的思维方式,系统只围绕包括Sprint任务导入、任务完成更新(工时填报)、燃尽图自动生成、问题的生命周期管理这样几个核心功能进行开发。

这样也使得任务板和信息系统的信息完全一致,障碍(问题)也得到了有效的跟踪和管理。

2. 配置管理的敏捷实践活动对于暴风影音的官网版本来说,每个月会有多个官网版本发布,一种是提供给用户新特性验证的版本,根据用户体验后真实的反馈信息,进行后续产品的优化和完善,另一种是将已经完成验证的功能和新特性合并到稳定的官网版本中。

基本每两周就要交付一个版本,每个项目都感觉时间紧、任务重。

配置管理在敏捷实践中起到怎样的作用呢?

暴风转码的敏捷团队在版本V2.1刚刚完成用户故事的开发工作,也就是说,产品已经能运行,系统测试工作还未开展。

按照估计测试和修改Bug的周期为一周计算,此一周的时间研发人员会很饱和,按照公司对人员使用率的要求,通常会启动V2.2版本的开发工作。

这样会提高研发人员的利用率,但也会带来两个甚至多个版本同时开发的问题,如果没有好的配置管理,代码会混乱不堪,很多人同时并行两个版本的代码开发工作,一方面改V2.1版本的Bug,另一方面开发V2.2版本的新功能。

我们通过配置管理工具的代码分支、代码合并来解决代码管理问题。

为V2.1开出代码分支,这样两个版本的代码相互隔离,不会彼此干扰。

V2.1版本上的Bug修复工作,对于V2.2的开发同样也适用。

这样实现了版本的交替开发,提升了开发人员的使用率。

通过代码的分支与合并也能处理好“新特性开发”与“产品稳定”的关系。

两种版本经常存在交叠。

用分支支持交叠,即防止了前者干扰后者,也防止了后者阻滞前者,这样维持了开发效率和产品质量的平衡关系。

在敏捷开发过程中,变更是被鼓励的。

而变更一定会体现在代码中,因此在SVN里,加一些控制以保证开发人员做且只做该做的事。

方法是让变更集不由开发人员创建,而是由专门的配置管理员创建,然后指定给相应的开发人员完成。

配置管理员会和产品经理、开发负责人和测试负责人沟通达成一致,然后在SVN中完成相应的设置。

对于一些比较大的任务,需要多个人协同完成,可以为此开分支,分支也由配置管理人员创建,然后要求开发人员在分支上工作,完成相应任务,同时控制修改代码之后的提交。

对于分支版本,只有符合本次版本要求的分支才允许合并到主干。

这个工作是配置管理员和敏捷团队协作完成的。

配置管理实践中最重要的是要摒弃原来所有的控制思想,让配置管理员作为敏捷项目团队中的一员,和团队共同协商如何管理代码、管理变更,而不像实施CMMI过程中有各种控制和审批环节。

3. 敏捷中实施Code Review(代码走查)技术评审活动是不管是否实施敏捷项目管理都需要进行的一种活动,很多人从来没有搞清楚什么是技术评审、什么是管理评审。

<<敏捷开发一千零一夜>>

技术评审包括需求评审、设计评审、代码Review、测试用例评审等和项目直接结果相关的工程活动的评审。

管理评审指的是敏捷项目管理的三个仪式：Sprint规划会、每日例会和Sprint评审会。

传统项目管理中的项目立项评审、里程碑评审和验收评审也属于管理评审。

简单地总结，技术评审是针对软件工程活动的评审，其目的是通过缺陷预防提升软件的质量，管理评审是针对阶段活动是否可认定为结束、下阶段如何开始，其目的是项目管理活动是否有效，交付的成果是否可接受。

如果借助敏捷项目管理的思想“交付可用的软件”，技术评审在敏捷活动中显得尤为重要，从评审的正式程度上划分，技术评审分为正式评审、技术审查和代码走查三种类型，如下表所示。

评审种类计划准备会议修正确认 Inspection（正式评审）有有有有有 Technical Reviews（技术审查）有有有有有 Walkthrough（代码走查）有有有有有 正式评审通常由项目经理主持，邀请项目的核心成员和外部有经验的专家参加，目的在于定位并消除软件中的缺陷。

技术审查或称内部评审，通常由技术负责人召集相关人员参加，一般是在模块或功能完成时进行，目的在于通过对交付模块或功能的技术审查，提出改进意见。

代码走查又称代码走读，由代码作者来组织，主要是代码质量分析和编程规则检查。

一般是在模块或功能完成的早期进行，作者有一定的想法时，希望从其他开发者那里获得一些帮助或补充。

由作者主持，目的是进行缺陷预防，提高代码的质量，很多公司已经成功地实施了Code Review和结对编程。

总结第一轮实施敏捷的试点项目，并没有要求实施代码走查活动，项目负责人和开发人员也没有意识进行Code Review。

要么是找借口说进度太紧张时间不够用，要么是不知道如何进行代码走查。

第一种纯粹是意识问题，要回答不知道怎么进行代码走查的问题则从代码走查的种类开始着手。

代码走查分为两种：一种是交叉代码审查，自己的代码由他人来检查，就像检查作业一样；另一种是代码会审，就是以会议的形式，大家共同审核代码的质量。

代码走查主要审查的内容包括：是否符合代码规范，确保所有人写的代码基本一致并符合代码规范；代码满足基本用户故事的要求，检查代码的逻辑实现，确认能实现用户故事；代码满足性能等非功能性需求，非功能性需求一般需要借助一定的工具和Code Review人员的能力，针对编码中对于性能影响的瓶颈给出解决方案；代码简化，提高代码的可读性，能够用公用的方法和公用API实现，则无须每人定义自己的实现代码。

随便抽出某开发人员的1000行代码检查的结果是：代码注释不充分，很多实现逻辑的类和方法没有注释；有些代码性能差，频繁地读/写磁盘I/O；有些代码虽然实现了功能需求，但从逻辑上写得太死，不便于扩展；在约定使用统一代码的写日志功能代码作者定义了自己的代码来实现。

看来为了提高质量，代码走查是实施敏捷项目管理一定要执行的工作。

4. 被修订的测试报告测试报告是集成测试阶段每天都需要测试工程师编写的一份重要的信息沟通工具，它的目的是让团队所有成员了解当前Sprint交付软件的质量情况，每个开发人员身上有多少未解决的Bug，以及交付模块的质量情况。

回顾试点项目的项目测试报告，大概是从百度上搜索的模板，冗长而不知所云。

报告中什么编写目的、背景、测试对象、测试概要、测试用例、测试环境等，光标题就会让人眼花缭乱。

从正规性上讲，这样的测试报告提交给最终用户无可厚非，但是给团队内部看就没必要这样讲求全面性和规范性了。

再引用敏捷思想“可用的软件重于完备的文档”。

所以修订测试报告应该是为团队减轻负担、提高沟通效率的一项优先级非常高的优化活动。

问问团队需要什么样的报告。

产品团队回答需要了解当前软件的质量来决策是否可发版，以及有些不好解决的缺陷拿出来让团队共同来决策Bug的处理策略。

<<敏捷开发一千零一夜>>

研发团队回答需要了解每个人身上有多少个未解决的缺陷，Bug整体解决情况和各模块的质量。

测试团队关注的是Bug整体解决情况及分解下的按模块、按人员缺陷情况。

统一思想后的测试报告包括一个测试情况的基本总结和三个部分。

基本情况总结是整个测试报告中最重要的部分，应该用粗体和红色来描述。

要求尽量用一两句话描述清楚当前测试的总体情况。

例如，当前测试遇到未解决的严重级别缺陷大于3个，不符合发版需求，其中组件下载功能如果点击取消，必崩。

第一部分是Bug移除率的统计，其中累计移除率是了解当前Bug整体解决情况的最好的说明，如下表所示为2010年9月23日某项目测试统计数据。

日期当前遗留本日新增本日移除本日新增未解累计新增累计移除本日移除率累计移除率
9/1412087805831419492 . 0%61 . 8%9/1513683675839726180 . 7%65 . 7%9/16135818338478343102 . 5%71
. 8%9/17105477629525420161 . 7%80 . 0%9/1810787844561250596 . 6%82 . 5%9/2113973414568554656
. 2%79 . 7%9/22137676950752615103 . 0%81 . 8%9/2315271495882367169 . 0%81 . 5%

成功实施敏捷项目管理的团队一定有疑问，为什么实施敏捷后还有集成测试阶段，看上面的数据为什么质量会那么差，似乎和敏捷倡导的交付可用的软件有冲突。

带着这些“十万个为什么”不妨思考一下。

没有实施代码走查导致了大量缺陷，造成的直接结果是必须有个集中测试的阶段来把Bug找出来，每个功能和功能之间并不是独立的，开发人员在开发过程中并没有及时沟通导致了接口不一致等诸多Bug，产品人员没有及时确认交付的功能或模块是否可用甚至提出一些变更也导致了一些缺陷。

如此多的原因导致这么多缺陷已经解释了上面的为什么，结论是如果敏捷项目管理实施成这样，基本上可以界定为是一场“形似而神不似”的失败的敏捷尝试了。

但也可以肯定的是，这份测试数据总结对团队了解项目当前的总体质量起到了一目了然的作用。

第二部分为Bug人员分布，第三部分为Bug模块分布，分别是按人员和按功能模块的缺陷数据总结。

按人员的分布可以看出每个人当日解决的Bug数和剩余Bug数，便于提醒团队及时地解决Bug或者相互协助。

按模块的Bug数总结便于提醒团队哪个模块的Bug多，除了了解模块的质量外还有助于提示团队及时进行代码走查来彻底解决质量问题。

5. 发布规则，绝不让步经过再一轮的组织级探索，让团队对于敏捷项目管理的基本原则有了更深入的认识。

软件开发的残酷现实告诉我们，即使在敏捷项目管理框架下，没有规则的软件开发过程带来的只可能是无法预料的结果。

不管是多么有经验的项目团队，在“响应变化胜于遵循计划”一项上都难以做到。

一次次失败的教训告诉我们，我们需要在敏捷框架下设定一些基本规则，而在这些基本规则面前，在保证敏捷思想的前提下不能向规则让步。

软件可用规则：上述的代码走查和测试数据告诉我们，在很多情况下我们交付的用户故事可用实际上是带有很多缺陷的，即使强调了PCT原则，团队交付的功能模块还是不完全可用，所以严格的方法是在燃尽图统计时，所有未完成的（即使是P和C状态都完成了的功能）都不进行统计，这和传统的“0 - 100”进度统计法则一致。

让任何一个团队成员都知道积极参与，因为每一个功能交付如果说可用就一定要可用。

目标导向原则：目标导向是充分调动员工工作积极性的最有效的方法，每个目标都设定卓越和及格的标准，在制订敏捷计划时，就需要明确项目如何做到卓越，并让指标认领到个人。

例如，手机端每个触屏操作的卓越标准是任意滑动的展现在0.6秒以下全部完成。

对于开发者来说，近期的Sprint目标总是比远期目标来说更容易看到和达到，这会使每一个Sprint员工的工作效率维持在一个较高的水平。

代码提交原则：很多开发人员在提交代码到配置管理系统时往往不注意是否通过了单元测试和自己代码的质量，这样会给其他人员带来或多或少的麻烦。

所以为了提升团队效率，代码的提交必须满足两个硬条件，其一是提交的代码必须经过自己的单元测

<<敏捷开发一千零一夜>>

试，其二是确保提交代码后整体代码可编译通过。

在这个原则的基础上，如果能自动化完成的工作绝对要减少人工干预，例如，持续集成和自动化测试尽量按策略自动完成。

有话好好说原则：项目成员理解和赞同用户故事的范围、项目目标吗？

开发人员大多数时间都不喜欢发表自己的看法和言论，大家不发表言论意味着他们认同项目范围和目标吗？

这一点在项目沟通中极为重要，敏捷教练一定要问每个人，让每个人表达自己的观点，让大家有话好好说，有话桌面上说，不可认为不发言就认为是同意。

在对项目目标没有一个共同认识的前提下，团队是不可能成功的。

文档极简原则：敏捷思想并不鼓励书写文档。

敏捷宣言也宣告“可用的软件重于完备的文档”，但如果一个软件是长期更新和维护的，还是需要必要的文档来将所有有价值的历史信息记录下来的。

你如果刚加入一个敏捷团队来接手一个已经迭代了9个Sprint的软件，如果什么都没有你从哪里下手？但是不希望文档过于庞大，能把用户故事的关键点、设计的关键点记录下来就够了，特别是代码每次变更必须用注释描述清楚。

精英团队原则：众所周知国内软件的成熟度偏低，一个公司能数出几个好的产品经理？

又有多少技术高手？

其实资源很有限。

如果实施敏捷项目管理，必须抱着精英团队的原则，不鼓励团队规模过大。

团队规模过大，按照 $n(n-1)/2$ 的沟通渠道计算公式，沟通成本增加，效率极大降低。

特别是很多需要技术难点攻关比较多的项目，人数多根本解决不了任何问题。

总之，在不违背敏捷思想的前提下，一些必要的规则是保证项目目标能够达到卓越的必要约束，也是促进团队建设和高效协作的基础。

<<敏捷开发一千零一夜>>

媒体关注与评论

敏捷是一种方法，更是一种组织能力。

那些不懂得敏捷不利用敏捷方法的研发团队，最终将在快速变化、竞争激烈的互联网环境中落败。

拥抱敏捷吧，从现在开始！

——京东商城高级副总裁李大学这是一本很好的讲敏捷的书，本书的作者们并没有生硬地罗列令人费解的理论，而是结合自身的经历，给读者生动展现了敏捷在IT行业的各种应用场景，令人受益匪浅，非常值得一看。

我们也可以从中了解到，任何一种工具或者方法应用到实际工作中时，都不能生搬硬套，而是需要根据具体所在的行业、公司的企业文化以及研发团队的状况进行适当调整，亦可以称之为“本土化”——通过在实践中不断的解决问题，持续优化，使其更适用，更实用，真正的帮助到团队。

——汽车之家CTO 廖志强

<<敏捷开发一千零一夜>>

编辑推荐

覆盖组织转型、产品管理、团队建设、工程实践，解析精彩过程，分享经验教训，揭示敏捷背后的奥秘。

汇集IBM、淘宝、京东、暴风影音等IT名企的敏捷专家经验分享。

<<敏捷开发一千零一夜>>

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>