中国敏捷软件开发
成功实践案例集

2011 年 08 月

编委专家简介

（按姓氏首字母排序）

陈志波

陈志波博士目前是 Technicolor 中国研究院多媒体实验室主任，视频处理/编码/媒体质量分析领域的专家，国际电气与电子工程师学会(IEEE)多媒体技术委员会成员，并是一些国际多媒体会议的组织委员会和程序委员会成员。作为公司首先启动敏捷式研究管理的项目负责人，有四年以上的利用敏捷式(Agile)管理流程管理研究和创新团队的经验。

单　岚

任职于用友医疗卫生信息系统有限公司，担任研发中心-R 应用开发部开发经理。2001 年 7 月-2004 年 1 月，任职于中科软件集团，担任开发人员。2004 年 2 月-2010 年 4 月，任职于用友软件股份有限公司客户化开发部，担任项目经理。从 2010 年 4 月至今，担任用友医疗卫生信息系统有限公司的开发经理。目前作为 R6 产品的开发经理，主导并实施了项目实施支持网系统，在研产品并不成熟的情况下交付了多个项目，有效的打通了一线实施与研发部门的沟通渠道，并对在研产品的功能和易用性上做了非常大的提高和完善。

高　航

任职于用友医疗卫生信息系统有限公司，担任 G 应用开发部开发经理。从事软件开发 5 年，精通 JAVA 系列技术，熟悉 Delphi 技术。在社保和医疗行业有着丰富的业务建模和系统架构经验。目前专注于软件研发团队的管理、软件研发流程的工具化实践与优化，并积极探索敏捷化开发在工程实践中的应用。

顾　焱

任职于用友软件股份有限公司，担任 NC 产品本部副总经理。2001 年加入用友软件,历任 NC 资金开发部经理,NC 供应链开发部经理,NC 产品本部副总经理。致力于大型管理软件开发 10 余年，在实践中不断尝试改进开发过程，为建立高效适应快速变化市场的开发团队不懈努力。

何　宇

任职于汤森路透，担任 GEDA 部门的 Technical Team Manager。7 年软件行业开发经验，曾服务于欧美日等大型外资企业，从事过外包项目、大型 ERP 系统开发、成熟系统维护改造、以及新系统设计开发等多种类型开发管理工作。熟悉 CMMI、SCRUM 等软件开发流程。在多个项目中推广使用 SCRUM，交付了数十个迭代，积累了宝贵的经验。

黄　方

任职于 Electronic Arts 上海公司，担任 ScrumMaster/Project Manager。CSP,CSD, CSPO, CSM, PMP，十二年 IT 工作经验，七年传统项目管理经验，三年敏捷项目管理经验，带领多个 Scrum 团队从事游戏开发工作。

李春林

任职于东软集团，担任过程改善中心副主任。中国敏捷软件开发联盟副秘书长，资深过程改善顾问，MBA，CSM，A-SPICE Provisional Assessor。1999 年加入中国最大的软件解决方案及服务提供商东软集团，拥有 12 年软件开发和过程改善经验。先后从事嵌入式软件系统研发、测试和项目管理工作，后专职从事过程改善工作，参与了东软集团质量体系文件的编写，曾作为评估组主要成员在 2002 年和 2004 年两次实施了 CMMI v1.1 5 级评估，并作为管理者领导了东软集团 2007 年和 2010 年的 CMMI v1.2 5 级评估。

刘德意

任职于特艺（中国）科技有限公司（Technicolor China），担任北京研究院质量与项目管理部经理。十年以上软件开发及项目管理经验，自 2002 年逐渐转入质量管理、过程改进。先后帮助所在公司通过 CMM-2、CMMI-3 的正式评估，以及公司内部评估。从 2007 年开始，在 Technicolor 中国的北京研究院推行 Scrum，并逐渐走入正轨。通过多年实践，深信过程改进要以人为本，讲求实效。本人是 Agile 的积极参与者，曾在 2010 年的 Agile Tour Beijing 活动中演讲。

刘曙光

任职于广州畅盟信息科技有限公司，担任 IT 部门技术总监，有 10 多年 IT 行业的工作经验，对软件工程及其技术服务有着深厚的理解和认识，曾为电信、电力等行业多个客户提供软件工程咨询及技术支持服务。目前主要专注于 ALM 和自动化测试方向。

卢旭东

任职于广联达软件股份有限公司，担任总裁办研发副总裁。有销售、市场、项目管理、产品管理、技术管理等丰富的经验，曾担任过广联达 PMO 的经理，对敏捷及敏捷咨询和实施有非常独特的理解，是他带领 PMO 在公司实施敏捷并取得显著成效，他是公司敏捷实施成效的有力保障。

马　娜

任职于沈阳东软集团（大连）有限公司，担任基础软件事业部产品管理中心专员。近 10 年的 IT 从业经验，从事过测试、配置管理、项目管理等工作，对于 RUP 和 Scrum 软件研发管理方法有 3 年左右的实践经验，取得 PMP 认证。

宁德军

任职于 IBM，担任 IBM Rational 大中国区技术总监、中国石油大学兼职教授、中国敏捷联盟副主席。有超过 15 年的产品及项目管理和软件工程经验，先后在上海贝尔阿尔卡特比利时研发中心、IBM 工作，拥有丰富的跨国项目和跨国公司产品管理经验。曾为华为、中兴、爱立信、腾讯、上海电力、工商银行、交通银行、中国银行等数十家企业提供咨询和培训服务。目前专注于产品及项目组合管理、敏捷开发过程和企业架构等新技术的研究。

潘　炜

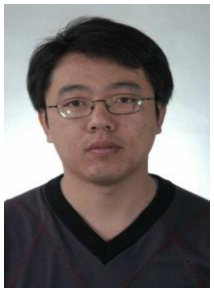
任职于汤森路透，担任 Sales & Trading 部门 Senior Software Engineer。毕业于北京邮电大学，工学硕士，3 年多专业行业软件研发经历，先后就职于斯伦贝谢、汤森路透等 500 强外资企业研发中心。擅长微软.net 平台相关研发技术，熟悉敏捷开发流程，目前担任汤森路透固定收益类新产品研发团队 SCRUM Master。

庞建荣

特艺（中国）科技有限公司机顶盒事业部 R&D Office Leader，14 年软件开发及项目管理经验，8 年质量管理及过程改进经验，曾在大型石油化工行业、移动通讯行业及消费电子产业供职，先后从事软件开发、项目管理、质量保证、过程改进等工作，参与企业的 CMM 评估、ISO9000 内审外审，认证的 ISO 内审员，、CMM 内部评估员、CSM。

彭　渺

任职于广联达软件股份有限公司，担任 PMO 项目管理专员。有 7 年开发经历，在 PMO 从事四年项目管理咨询和实施工作，经历过多种项目和产品的敏捷咨询和实施，对敏捷有深刻的理解。



宋　洋

任职于东软集团，担任 AVNC&IS 事业部过程改善顾问。拥有 9 年的软件开发和过程改善经验，从事过软件开发、软件测试和项目管理工作。以质量体系审核员和 CMMI 1.2 评估组成员的角色参与 ISO9000 外部评估和 CMMI 5 级外部评估工作。目前主要从事过程管理方法和质量管理工具的引入和开发工作，通过引入敏捷的方法和实践，提升项目的过程效率和产品质量。

孙红伟

任职于畅捷通软件有限公司，担任开发资源一部项目经理。
参与新一代业务通 11.0 的设计工作，主要进行销售模块及部分基础档案的设计工作，期间参与平台组件验证，在 11.0 产品研发的中后期开始进行常州研发基地的开发管理工作推动业务通 11.0Beta 版本发布。在用友通 11.2 的产品研发过程中担任生产制造领域的部门经理，推动生产制造产品的研发过程，直到用友通 11.2 产品的发布。

王　鑫

任职于新聚思（Synnex）信息技术有限公司，担任 IT EPG 总监。分管 synnex 公司过程改进、产品质量 8 年以上；拥有 15 年的软件研发、项目管理与质量管理经验；精通 CMMI、敏捷等方法论。作为 EPG 负责人，目前专注于利用 CMMI、敏捷等方法提升组织效率、质量及相关 IT 管理工具的开发。

闻　宇

任职于石化盈科信息技术有限责任公司，担任研发中心技术总监。软件工程硕士，13 年软件开发经验，7 年大型应用系统设计经验，在企业级分布式应用系统的设计及研发方面有独到的见解及特长；敏捷开发领域的积极倡导者和实践者，曾获邀在微软西雅图举办的 MVP 全球峰会上做“基于 Scrum 敏捷开发的技术演讲”，目前正主导公司 CMMI4 过程改进工作，将充分实践敏捷与 CMMI 相结合的新软件开发模式。连续两年（2009、2010）获得微软 MVP（最有价值专家）称号、印度信息软件管理学院（NIIT）系统架构师、中科院高级系统架构师。

许舟平

任职于 IBM，担任软件部 Rational 高级技术顾问。多年敏捷开发和项目管理经验。致力于敏捷开发在中国软件开发中的推广与实践。《敏捷无敌》一书的合作者。

闫建伟

任职于用友软件股份有限公司，担任银行客户事业部开发经理。多年从事银行业务 IT 系统的建设工作，具有丰富的银行专业知识和 IT 系统架构经验。

叶　蓁

任职于英孚教育，担任 EF lab 的高级项目经理/Scrum master。2004 年至今在英孚教育产品研发部门担任项目经理，8 年以上项目管理经验，2 年敏捷开发经验。项目团队成员来自不同国家和地区。早期采用瀑布式项目管理方式进行项目管理，于 2009 年 7 月开始在团队里引进敏捷软件开发模式，2010 年 3 月接受培训成为 scrum master。2011 年 5 月开始，公司进行大范围的敏捷转型，本人所在团队有幸成为首先接受全面培训的先锋团队，希望籍此机会分享团队的转型经验。

张克强

任职于上海宝信软件股份有限公司，担任宝信软件技术中心项目总监。系统分析师，Certified Scrum Master，硕士。现在是负责领导咨询团队把业界最佳实践和自身有效实践应用到宝信的各个研发单元中，并组织和协调各单元的过程改进评估和审核。他曾经在 Intel 的 Christea 团队担当质量保证经理。在软件工程方面拥有 9 年经验，在过程改进、质量保证和测试方面有丰富的实践。帮助组织按 CMM/CMMI 模型进行改进，并通过了 CMM4、CMM5、CMMI5 的评估，从 2007 年起为宝信软件引入了敏捷方法，在 CMMI 的框架下推进敏捷实践，在软件质量和开发效率方面都获得了明显的改善。他参与了 2010 年中国敏捷发展报告的书写。

周　静

任职于东软集团股份有限公司，担任嵌入式软件事业部高级质量工程师。5 年软件开发、5 年项目管理和质量管理经验。曾在毕博大连 GDC 和东软集团担任 Quality Specialist，对 CMMI1.2、6Sigma、质量管理五大工具颇有研究。近年负责敏捷项目质量管理、敏捷开发方法的推广和指导。通过改善，试点项目目标达成率提高 22%。

周瑜傑

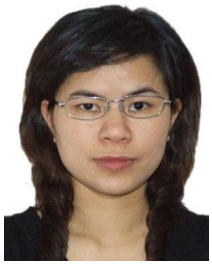
任职于英孚教育，担任 EF – Labs 部门的项目经理、内部教练、Scrum master。5 年开发工作经验，之后转入项目管理。前 5 年的开发经验主要基于 web 的 CRM 平台开发。加入英孚后转战教师日常管理系统。经历了传统模式到 scrum 的变迁，深深体会了 scrum 的特性给团队和客户所带来的利益。内部教练的职责，也让我更深入 scrum。Scrum 没有表面看上去简单，实施也没有想象的容易，但比想象中有效。

Zhuang Wan-Jun(Mike)

He is now working in Shanghai Hewlett-Packard. He served as Agile Service Experience: 4 Years; Certification: CSM,PMP,ITIL, Consultant Capability: Agile Training/Coaching, Organization transformation, Project Management, IT service Management, rocessManagement.

Multinational projects as IT project manager(PMP®)/consultant for 5+ years in foreign companies.5 years IT industry experiences in managing projects across multiple regions and geographies and comfortable with working in a highly distributed and virtual environment. Build high performing teams, lean processes, and achieve organizational agility through Agile transformational journey. Define and establish the foundation and baseline of Agile development process/Agile Implementation Plan for Applications Development Services Offering at HP. Re-configure and fine-tune Agile Methodology and processes to adopt Distributive Agile(Attend 2010 HP Global Tech conference for Distributive Agile topic).

点评专家简介



黄晓倩

任职于特艺（中国）科技有限公司（Technicolor China），担任机顶盒事业部项目经理。十年以上软件开发及项目管理经验，曾在大型通讯行业及消费类电子产业供职，先后从事软件开发、技术 leader，项目管理和 SCRUM Master 等工作。

李忠利

曾先后供职于用友软件和某美国财富 500 强企业。积极的敏捷软件开发方式的热爱者与推动者。历时 5 年的持续努力，成功的将一家跨国软件开发公司从 CMMI3 级的开发模式引导到敏捷软件开发模式，同时还负责企业内部敏捷软件开发管理工具。在企业敏捷导入和实施上，积累了丰富的经验。

钱　岭

任职于中国移动通信有限公司，担任研究院资深研究员。高级工程师，2001 年 1 月毕业于清华大学计算机课学与技术系，获得工学博士学位，主修软件工程方向。毕业后加入贝尔实验室基础科学研究院，参与并负责包括软件质量管理、网络设备、VAS 等产品研发。2007 年 12 月加入中国移动通信研究院，历任广告相关项目经理、云计算 HugeTable 项目经理、大云项目架构师、大云产品研发负责人。在软件工程方法、敏捷开发方法和实践、工具和实践、基于 CMMI 的过程改进、软件质量管理、软件度量等领域有较多的研究和实践工作。

郜晓梅

中国 CSTQB 委员会工作组员，华为公司 10 年测试工作经验,无线系统工程部测试专家，华为 MiniStar 测试技术会议的发起人和组织者。擅长测试分析设计、测试管理、敏捷测试、探索性测试、基于模型的测试、测试过程改进、基于需求的测试、缺陷根因分析等方面。由她提出的 MFQ&PPDCS 软件测试分析设计方法和 T-RCA 缺陷根因分析方法在华为多个产品得到实践应用。

邢　雷

任职于东软集团股份有限公司过程改善中心，任咨询顾问。近十年行业经验，海外工作三年后回国投身于软件项目管理及过程改善工作。曾作为核心人员负责所在组织 CMMI 五级的过程改善和最终评估工作。目前致力于东软集团的敏捷技术研究、部署等相关工作。

张　忠

任职于用友软件股份有限公司，担任集团开发管理部总经理。1995 年开始从事软件开发工作，从 2006 年开始主管用友软件股份的开发管理工作，2011 年开始主管集团开发管理工作。

赵　静

任职于 IBM，担任软件部的高级技术顾问。目前主要关注 Rational 统一开发流程与平台、软件项目配置管理、项目开发管理、项目管理及项目组合管理等。曾经为多个通讯公司、银行、世界 500 强企业、政府相关部门提供过开发过程管理、项目管理、软件配置管理等咨询服务。

编委专家陈志波、高航、黄方、李春林、刘德意同时参与了案例的点评。

序　言

对国内企业敏捷成功实践的采集，是年初中国敏捷软件开发联盟确定的7项认领工作之一，确定此工作是基于联盟对中国敏捷软件开发运动发展状态的准确把握：当前正处于为什么做和怎么做的过渡时期，同时存在两批人：一部分人问为什么要敏捷？而另外一些问怎么做？

与此同时，经过近几年敏捷运动发展，国内一批应用敏捷的先行企业，已经有了不少实践，而且很多取得了明确的效果，把这些实践采集和编辑成册，将有助于回答很多疑问：

中国的软件企业可以成功实施敏捷吗？

敏捷能用于大型软件的交付吗？

敏捷实施需要的文化、制度和人才基础我们具备吗？

有了CMMI和项目管理，还需要敏捷吗？

实施敏捷有真正效果吗？

这种时候，我们推出《中国敏捷软件开发成功实践案例集》，并以此为基础，提炼其中共性方法，制定《敏捷软件开发知识体系》（ADBOK），无疑是有很大的积极意义。

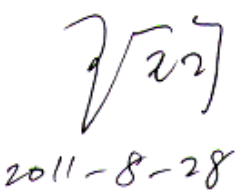
书中这些案例都是由企业将自身实践中行之有效的实践编制而成，联盟组织编委会进行了简单的审核，把明显非敏捷案例去掉，然后由专家加注评语，集合成册，供其他企业参考和借鉴。

本来原计划进行评选，将其中优秀实践识别出来，但过程中发现若要严肃的开展评选，所需要时间显然不够，因此评选留待以后，此次仅仅是把所提交案例结集成册，以便能借助敏捷大会这个难得的场合尽快发布给大家阅读。

成果的取得，离不开团队的协作，我很高兴向读者介绍这个精英团队：工作组长宁德军（IBM Rational技术总监）、工作副组长张忠（用友股份研发总经理）、工作副组长李春林（东软集团过程改善中心副主任），还有21位很棒的成员，项目过程中，处处彰显出大家对敏捷的挚爱、对专业的热情、对行业工作的社会责任感，这些年轻的从业者所体现的精神，正是我们整个软件行业生机勃勃，精彩纷呈的原因。

也许，这些工作还很粗糙，但是毕竟我们已经在路上了——不仅学习于国际社区，而且要在实践中有所创新，勇于分享，争取对国际社区有所回馈。

我也借此机会，代表协会真诚欢迎各路英雄豪杰大侠参与到联盟平台上，共同精化和演进这些成果，推动中国敏捷软件开发运动快速前进，实现我们“以过程改进之能，助企业发展之力”的共同目标。



2011-8-28

编者序

《易经》中的易有三易，不易、变易和简易。领悟不易，才能抓住本质，实现御道而行；洞察变易，才能灵活应变，活学活用；实行简易，才能做到视自己之情境，快速实现敏捷。在敏捷开发知识体系中，“不易”的部分有二：敏捷的价值观和 12 条原则，他们直接影响人们的思想和思维模式。因此，正确的理解敏捷宣言，建立正确的敏捷价值观是成功开展敏捷开发的关键，它充分体现敏捷文化中面向结果、关注价值和以客户为核心的协作创新理念。敏捷开发知识体系中变易的部分则是各种敏捷实践，它即包括例如 Scrum、XP、Lean、OpenUP 等的管理实践，又包括诸如迭代式开发、多级项目规划、持续集成、完整团队等的工程实践，企业只有实事求是的根据团队状况，采用适用的实践并且正确使用，才能真正实现敏捷的价值观。

《中国敏捷软件开发成功实践案例集》中的各种案例，正是对中国企业实施敏捷开发过程中，如何灵活应用各种敏捷实践的成功案例进行总结、分享，以飨来者！

在此我要感谢所有为《中国敏捷软件开发成功实践案例集》作出卓越贡献的企业和朋友：

用友软件股份有限公司：单岚、高航、顾焱、孙红伟、熊昌伟、闫健伟。

IBM：许舟平。

石化盈科信息技术有限责任公司：闻宇。

特艺（中国）科技有限公司：陈志波、刘德意、庞建荣。

英孚教育：叶蓁、周瑜傑。

汤森路透：潘炜、何宇。

Electronic Arts 上海公司：黄方。

广联达软件股份有限公司：卢旭东、彭渺。

广州畅盟信息科技有限公司：刘曙光。

上海宝信软件股份有限公司：张克强。

上海惠普：庄万俊。

新聚思（Synnex）信息技术有限公司：王鑫。

东软集团：李春林、马娜、马云存、毛军、宋洋、周静。

还要感谢联盟秘书处工作人员的努力：黄群、赵倩、夏冰。

目　录

案例一：迭代式开发（IBM）

1. 案例说明

1) 案例来源

IBM

2) 项目基本信息

软件协作平台产品开发项目中使用迭代式软件开发，项目基本信息如下：

- 团队规模：200+人，分布在 7 个地点
- 环境：Windows/Linux/AIX Java

3) 组织结构

- 二级团队：PMC & component teams
- 三层计划 ：Release、 Iteration、 Daily

在 PMC 和 component teams 采用 Scrum；一个大项目经理管理整个项目群；每个人都是技术人员。

4) 面临的问题

发布周期短、市场需求不断变化、新技术、风险较大。

5) 解决方法

采用迭代式开发敏捷成功实践，迭代周期为 4 周。主要活动包括：迭代计划、迭代架构工作、功能增量的开发工作、每周构建、迭代复审。

6) 主要收益

通过迭代确立项目的“心跳”，在过去 3 年内提高团队绩效+15%；支持客户的快速反馈和发布；能够更快速拥抱变更；迭代有助于快速降低项目风险。

2. 定义和特性说明

迭代式开发也被称作迭代增量式开发或迭代进化式开发。在迭代式开发方法中，整个系统开发工作被组织为一系列的短小、固定长度（如 3 周）的小项目，

被称为一系列的迭代。每一次迭代都包括了需求分析、设计、实现与测试，完成系统的一个功能增量开发工作。在整个过程中，不断通过客户的反馈来细化需求，开始新一轮的迭代。随着迭代的不断增加，系统的功能越来越完善。

3. 如何应用成功实践说明

将传统的需求文档转化为一系列的用户故事，然后按照主题进行分组。分析确定每个主题的优先级。根据主题的优先级，确定版本计划。对每个主题里的用户故事确定优先级，构建该主题的产品订单。

用户故事有助于敏捷团队从真正的用户的角度出发去理解需求。通过明确的角色定义，能够有针对性地把利益相关人引入到用户故事的讨论中来。通过以用户的角度来描述用户故事，能够体现用户如何使用我们将要交付产品，反映了用户的实际需求，而不是技术实现方案。而每个用户故事的商业价值为确定用户故事的优先级提供了量化的参考。

为每一个主题组建一个敏捷团队，由敏捷团队对用户故事以故事点进行相对大小的估计。对于高优先级的用户故事，确定任务并进行工作量的估计，预估敏捷团队的交付能力。

在每个迭代周期开始的时候，从产品订单中选出高优先级的用户故事，制定迭代计划。在迭代周期结束，演示已经完成的用户故事，由产品负责人确认用户故事是否完成。

专家点评：

大公司、大团队的典型案例，迭代式开发模式的实践如教科书般规范。遗憾的是大团队的管理方法和运作经验没有做更多的分享，但组织结构的管理模式（如一个经理、二级团队、三层策划）仍然给我们提供了很多借鉴。

点评专家：李春林

案例二：基于 Scrum 框架的迭代开发过程（用友）

1. 案例说明

1) 案例来源

用友软件股份有限公司

2) 项目基本信息

新一代产品已经发布完两个版本 11.0 和 11.2，随着产品上市用户的使用，很多的用户反馈产品的效率和稳定性存在着比较大的问题，尤其是产品的效率让用户不满意，多家客户也因为产品的效率问题要求退货。为了提高客户的满意度，提供伙伴对新一代产品的信心，经项目管理委员会批准在新一代 11.3 项目中不对产品的功能进行任何增加专门在 11.3 中针对产品的效率及稳定性进行修改。

- 团队规模：84 人
- 环境：ASP.Net,JavaScript,C#,IIS,SQL Sever2005

3) 组织结构

整个新一代成员分成如下小组：

- 业务平台 Scrum 组（ScrumMaster 谭坦 ScrumMember：8 人）
- 业务往来生产 Scrum 组（ScrumMaster 赵子东 ScrumMeber：6 人）
- 供应链报表 Scrum 组（ScrumMaster 邹全林 ScrumMember：3 人）
- 库存 Scrum 组（ScrumMaster:司建成 ScrumMember：12 人）
- 总账 Scrum 组（ScrumMaster:慕红利 ScrumMember：7 人）
- 技术平台 Scrum 组（ScrumMaster:李胜国 ScrumMember：8 人）
- 现金银行 Scrum 组（ScrumMaster:蒋勇 ScrumMember：5 人）
- 销售采购 Scrum 组（ScrumMaster:张艳晖 ScrumMember：6 人）
- 需求组：3 人
- 项目经理：2 人
- 测试组：21 人

- 团队物理分布：江苏常州 39 人 北京：44 人

4) 面临的问题

- 产品面临的部署网络环境复杂简陋（要求能够用 2MADSL 接入作为服务器)；
- 团队中缺少 Web 应用特别是基于互联网应用的专家；
- 团队成员分布在异地沟通不直接。

5) 解决办法

- 针对 2M ADSL 的网络环境确定针对降低网络流量的方案，并在系统内各点落实降低网络流量的要求，从技术角度把流量降到最低；
- 团队中缺少 Web 应用及互联网方面的专家，发动团队成员从学习入手，同时吸取一些成功公司的成功案例，进行技术交流；
- 为了加强沟通，除了必要时候直接到异地现场，大多数情况下依靠及时通讯工具以及电话等方式进行沟通，同时在团队内部引入 ScrumWorks 系统，任务的下达反馈更多是通过该系统实现，同时由于很多团队成员分割在两地很多的会议都采用网络会议的形式进行。

6) 主要收益

- 提高了团队的自我管理意识，增加了团队的战斗力；
- 提高了团队的工作效率，以突出阶段行成果为目标的做法，保证了在任何阶段团队都有可提交的成果。

2. 定义和特性说明

Scrum 要求团队在每一次迭代的结尾完成一些可以交付的工作片段。迭代要短，有时间限制。将注意力集中于在短时间内交付可工作的代码，这就意味着 Scrum 团队没有时间进行理论研究。不会花时间用建模工具来画 UML 图、编写完美的需求文档，也不会为了应对在可预计的未来中所有可能发生的变化而去写代码。实际上，Scrum 关注如何把事情做好。这些团队承认在开发过程中会犯错，但是他们明白：要投入实践中，动手去构建产品，这才是找出错误的最好方式；不要只是停留在理论层次上对软件进行分析和设计。

3. 如何应用成功实践说明

总原则：交付可工作的软件；在每一次迭代的结尾完成一些可交付的工作片段；把事情完全做完，达到可以交付的状态；事情只做了一半，它的价值就是 0。

迭代要有固定时长（被称为“时间盒——timebox”），暂定 2 周为一个迭代周期；在每一次迭代的结尾，代码都必须经过验证能够正常工作；团队必须要有燃尽图，而且要了解他们自己的生产率；在一个 Sprint 中，外人不能干涉团队的工作。

Scrum 管理工具：ScrumWorks Pro4.3

Sprint 计划会议：组织实施人为各个 ScrumMaster；确定 Sprint 的工作目标；确定团队成员名单；确定 BackLog 列表；把目标以及 BackLog 分解成任务；团队估算时间，确定 Sprint 的长度；确定 Sprint 的演示日期；确定每日 Sprint 例会的时间和地点。

专家点评：

这是将 Scrum 框架应用在产品升级类项目的一个案例，项目同时具有大团队、分布式开发的特征。项目将大团队拆分成多个 Scrum 小组，严格遵循敏捷开发的原则和流程，采用 ScrumWorks 作为管理工具，很有借鉴意义。但个人认为“Scrum 关注如何把事情做好”并不意味着可以“没有时间进行理论研究”、“不花时间用建模工具来画 UML 图”——这对产品升级类项目没有问题，但产品研发类项目还是不可或缺的。

点评专家：李春林

案例三：迭代式开发（英孚教育）

1. 案例说明

1) 案例来源

英孚教育

2) 项目背景

项目组负责开发的产品属于公司的核心业务，多年来为用户提供 365 x 24 小时不间断的在线课堂服务。用户群是支付昂贵学习费用的来自全球各地，各个年龄层的学生，他们期望的是优质的用户体验。

由于产品的敏感性，本次的产品更新是六年来的首次。立项初期，产品经理（现 product owner）采用传统瀑布式流程，从概念到完成功能需求书已花时半年，预估开发周期为半年，目标是 2011 年 8 月份产品上线。公司管理层非常支持本次产品的更新换代。CEO 相信改善用户体验能显著提高产品使用率，从而创造更大价值。因此，尽快推出产品到市场，公司就能尽早获益。同时，中国学生数量激增，受制于网络带宽问题，老产品已经不能支持激增的用户量，尽快让新产品上线迫在眉睫。

项目市场需求迫切而且不断变化，技术实现风险大，团队进行了评估，与其完成全部功能的开发再上线，倒不如先推出一个简化版本。这样可以降低整体耦合度，分散风险，还可以提前发布。经过团队投票，大家愿意接受此挑战，并决定使用 scrum 作为项目开发管理方式。

3) 项目基本信息

为全球学生提供世界一流的 365 x 24 小时不间断的在线课堂服务。产品包括在线小班课和在线一对一课堂。

- 团队规模：7 人
- 环境：.NET C#；FW4.0；WCF； MVC 3；SQL2008；WIN SERVER 2008

4) 团队组织结构

这是一个国际化的喜欢挑战的年轻团队，团队组建于 2011 年 3 月。

- The team: 成员来自中国各地，战斗力强，热衷编程，讨厌开会；
- 刘晓敏：擅长前台开发，用 code 去优化用户体验；
- 姜严，朱琦，周磊：三人合共超过 25 年开发经验；
- 韦亦君：数学系毕业的测试工程师；
- Scrum master 叶蓁：关心但严厉的教练，有长期项目管理经验，一年半的敏捷团队经验；
- Product Owner：Beverly Chung, 钟路平，美籍华裔 ABC，对技术领域陌生，听到瀑布模型、极限编程就头大。

5) 面临的问题与解决

虽然这是一个新组建的团队，由于大家都愿意接受挑战，一致的目标迅速把团队融合起来。

为了让产品尽快上线，团队选用迭代开发的方式。

每个迭代周期为两周。第一个迭代进行开发，第二个迭代进行测试，第三个迭代作为发布前的准备和调整。每个迭代的开始团队进行迭代计划（sprint planning），迭代结束检视结果并进行总结和改进。短短的一个半月，第一版产品顺利上线。

第一版产品上线后，按照原来的功能需求设计，下一步的开发计划需要三到四个月才能完成，同时第一版的产品也需要不断改进，推广到更多市场。为了能更灵活的迎合多变的市场需求，团队进一步引进用户故事的形式，这样有助于让每个迭代周期更完整从而更独立。在每个迭代里，由 product owner 统一排定每个用户故事的优先级，里边既有对已发布产品的改善，也包括新产品的开发。由于用户故事都是相对完整的功能，从而保证每个迭代的结束，团队完成从设计，开发到测试的整个过程，生产出潜在可发布的产品。

6) 主要收益

对比原项目计划，本次产品更新比原计划提前三个月投入市场。在每一到两个迭代结束后，新产品持续发布到更多市场，三个月间完成全球市场的投放。

新产品上线以来，在全球用户的统计中，产品使用率提高 30%，各个市场都出现明显增长。中国用户的用户体验显著改善。由于改善显著，市场部正计划大

规模展开宣传，推广此在线产品。

以下图表显示在持续的 10 个迭代中，产品持续更新和发布到更多市场的时间表：

图 3-1 产品更新时间表

2. 定义和特性说明

迭代式开发也被称作迭代增量式开发或迭代进化式开发。在迭代式开发方法中，整个系统开发工作被组织为一系列的短小、固定长度（如 2 周）的小项目，被称为一系列的迭代。每一次迭代都包括了需求分析、设计、实现与测试，完成系统的一个功能增量开发工作。在整个过程中，不断通过客户的反馈来细化需求，开始新一轮的迭代。随着迭代的不断增加，系统的功能越来越完善。

3. 如何应用成功实践说明

考虑使您的迭代都具有相同的迭代长度。它有助于建立确立有“心跳”的项目和了解项目团队的绩效，支持创建更准确的估算。一般，在第一个迭代将攻克最困难的问题，但不可超载太多，要确保在每次迭代结束时您都可以展示进展。不要延长迭代，以完成工作，也不可以完成没有任何可展示软件的迭代。如果需要的话，可将工作分解成更小，更易于管理的任务，以实现这种平衡。

专家点评：

产品能够切合市场的需求，快速推出。并随着市场的变化，不断交付符合市场需求的产品，这点令人印象深刻。

点评专家：李忠利

案例四：迭代式开发 （用友软件）

1. 案例说明

1) 案例来源

用友软件股份有限公司

2) 项目基本信息

医疗基层卫生服务系统产品研发。

- 团队规模：20 人
- 环境：JAVA、用友 UAP

3) 组织结构

矩阵式组织架构，包括需求、UE/UI、开发、测试来自不同部门的人员组成产品团队，参照 Scrum 组织架构，产品项目经理为 Scrum Maste。

4) 面临的问题

产品研发设计到的小组多、研发流程环节多，在流程运转中某一环节延误，就会导致整体产品进度延误，并且各个小组沟通协作效率较低。

5) 解决方法

采用迭代式开发敏捷成功实践，4 周为一个迭代，将需求、UE/UI、开发、测试的工作都拆分到每个迭代中，迭代计划，快速交付，及时沟通。

6) 主要收益

产品研发流程运转效率大大提升，避免了产品实现与设计和需求不符的情况。

2. 定义和特性说明

迭代式开发也被称作迭代增量式开发或迭代进化式开发。在迭代式开发方法中，整个系统开发工作被组织为一系列的短小、固定长度（如 4 周）的小项目，

被称为一系列的迭代。每一次迭代都包括了需求分析、设计、实现与测试，完成系统的一个功能增量开发工作。在整个过程中，不断通过客户的反馈来细化需求，开始新一轮的迭代。在产品研发过程中，则根据多级项目计划的细化和调整，安排每个迭代的任务。随着迭代的不断增加，系统的功能越来越完善。

3. 如何应用成功实践说明

1) 因为产品属于大型项目，并且人员分工明细，需要研发流程协作，并且要去文档记录和评审，所以在使用敏捷开发时，不能使用全敏捷。在实践中是整体上遵守瀑布和流程化开发模式，而在每个迭代和每个 team 单独进行时采用 Scrum 敏捷开发框架。

2) 从任务划分上，产品团队分为需求团队、UE/UI 团队、开发团队、测试团队，将整体产品开发划分为以月为单位的迭代周期，每个环节团队根据迭代划分制定各自的迭代计划，并且相互之间的迭代任务要保持统一，这样每个团队的一个迭代就组成整体产品的一个迭代。从需求、设计开发到测试，一个迭代之间采用敏捷开发的 Scrum 框架，从初期的 Sprint 计划会议，到每日回顾的开发过程，再到迭代最后的评审和回顾会议，保证每个迭代交付的产品可构建、可演示。

专家点评：

与小型项目比较，大型项目在应用迭代开发的实践时，不可避免地会遇到团队之间如何沟通协调的问题。用友的一个实践是：每一个小的 team 按照 Scrum 方式开展每个迭代的工作，而在项目整体上依然采用瀑布和流程化的开发模式，这不妨看做整体项目均 Scrum 起来的一种过渡。

点评专家：郜晓

案例五：迭代式开发（东软集团）

1. 案例说明

1) 案例来源

东软集团

2) 项目基本信息

在 XX 产品开发项目中使用迭代式软件开发，市场需求不断变化，要求团队快速反馈。

- 团队规模：30 人
- 环境：Eclipse & Java

3) 组织结构

标准 Scrum 组织架构。

4) 面临的问题

需求不断变化、人员技术水平参差不齐、架构不成熟，进而导致进度难以控制的问题。

5) 解决方法

- 在制定迭代计划时，加入针对某项技术的短期培训任务，集中提升人员技术水平；
- 在每个迭代中，各团队人员局部轮换微调，培养多面手；
- 制定迭代时间切割原则：Scrum 中迭代（Sprint）的划分结合“产品的释放计划”和“部门的例行工作频率”两方面考虑。

基本规则是：

- 在每个月的倒数第 5 个工作日前完成迭代的评审和总结，在部门的资源计划会议后并且在下个月开始之前适时启动一个新的迭代；
- 当需要释放产品版本时，在各个子产品的 Sprint 都结束后需要启动若干个测试类型的 Sprint，进行集成和系统测试。因此在制定产品释放计划时，

需要考虑所有子产品最后的 Sprint 在产品集成测试前“卡齐”.

基于以上规则，每个迭代周期的建议在 3~4 周之间。

6) 主要收益

通过迭代初的详细规划以及执行中的跟踪检查，提高了进度的可控性；通过人员的轮换，培养出一批技术多面手。定义和特性说明全面部署快速迭代式的开发模式，获得项目的高可预见性、高可控性和团队的快速成长。

产品开发总体上可分为三个阶段：规划、开发、服务与支持。三个阶段间以产品规划为核心，驱动开发和支持与服务工作。

- 产品规划：根据多方面的需求和商业目标得出产品释放计划，包括释放日期和需求范围；
- 产品开发：开发产品规划的需求，在目标日期发布可用版本；
- 产品支持与服务：推广产品，提供咨询和技术支持，并跟踪产品的使用情况，收集用户的反馈需求。

在这里我们只对产品开发进行详细描述。

2. 定义和特性说明

全面部署快速迭代式的开发模式，获得项目的高可预见性、高可控性和团队的快速成长。

产品开发总体上可分为三个阶段：规划、开发、服务与支持。三个阶段间以产品规划为核心，驱动开发和支持与服务工作。

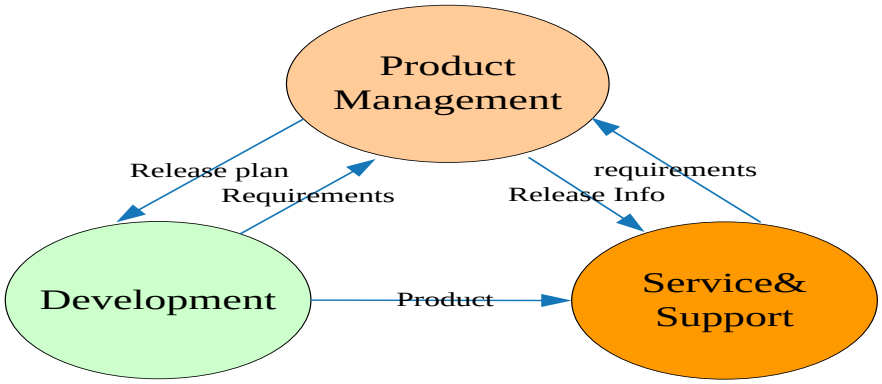


图 5-1 产品研发的三个阶段

1) 产品规划：根据多方面的需求和商业目标得出产品释放计划，包括释放

日期和需求范围。

2) 产品开发：开发产品规划的需求，在目标日期发布可用版本。

3) 产品支持与服务：推广产品，提供咨询和技术支持，并跟踪产品的使用情况，收集用户的反馈需求。

在这里我们只对产品开发进行详细描述。

开发过程的每个迭代分为四个阶段：计划、执行跟踪、检查、总结回顾。

1) 迭代计划（Iteration Planning）：主要工作包括迭代细化任务，估算任务的工作量，制定开发计划和测试计划。迭代的周期通常为 2~4 周。

2) 执行跟踪（Iteration Tracing）：在一个迭代内周期性的跟踪任务的进度，评估剩余工作量，并根据评估结果及时对计划进行微调，确保迭代目标的达成。任务跟踪是一个持续的过程，跟踪周期最长不要超过一周，最短为 1 天，具体可根据项目本身的约束而定，如时间、资源、特定的开发模式等等。

3) 检查（Review）：项目团队配合产品经理、用户等干系人一起检查产品是否满足需求，项目团队在此期间还要收集检查人员的反馈意见，并补充到下一迭代开发的需求中。此阶段通常以验收、产品演示等方式进行。

4) 回顾与总结（Retrospect & Report）：项目团队和 QA 成员一起回顾本次迭代过程中存在的问题和解决措施，制定进一步的过程改进计划，还要总结开发经验和最佳实践，为后续迭代开发做准备。

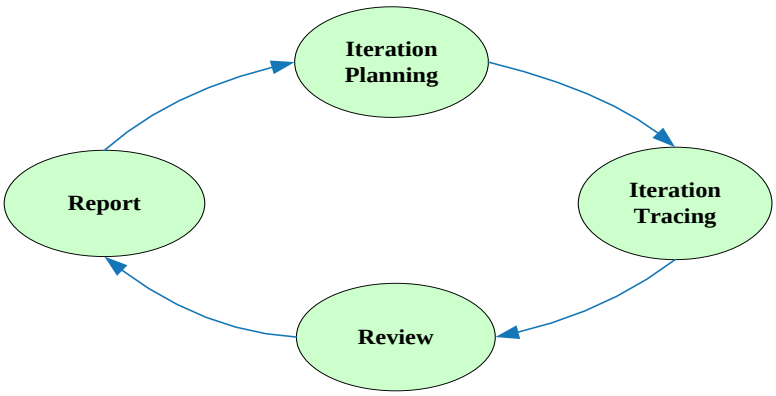


图 5-2 产品开发的一般性敏捷流程

3. 如何应用成功实践说明

这里将描述开发阶段的工作流程，即 Scrum 一般性流程。

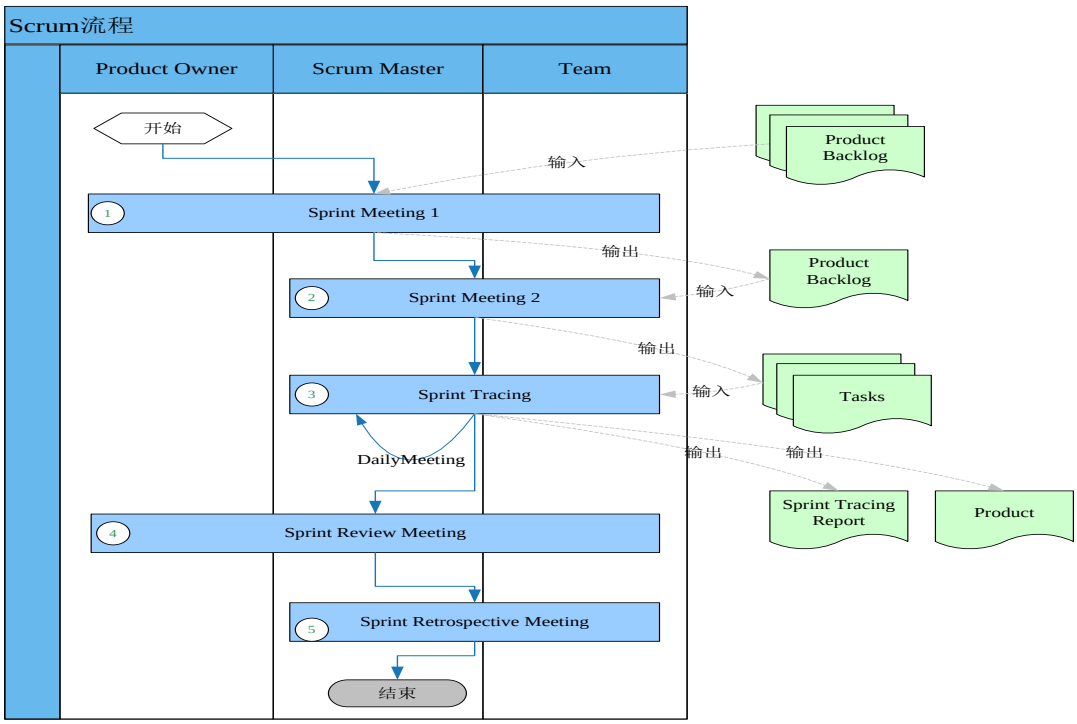


图 5-3 Scrum 一般性流程

1) 产品释放计划会议

产品释放计划会议是产品策划结束后，在迭代开发前进行，由 Product Owner 组织。

- 会议目标：划分迭代（Sprint），确定产品的释放计划；
- 参与者：Product Owner（本次会议主持）、Scrum Master、团队成员；
- 会议准备：Product Owner 对所有的产品 Backlog 都已按优先级从高到低排好序。产品 Backlog 都已发给所有与会人员；
- 会议成果：

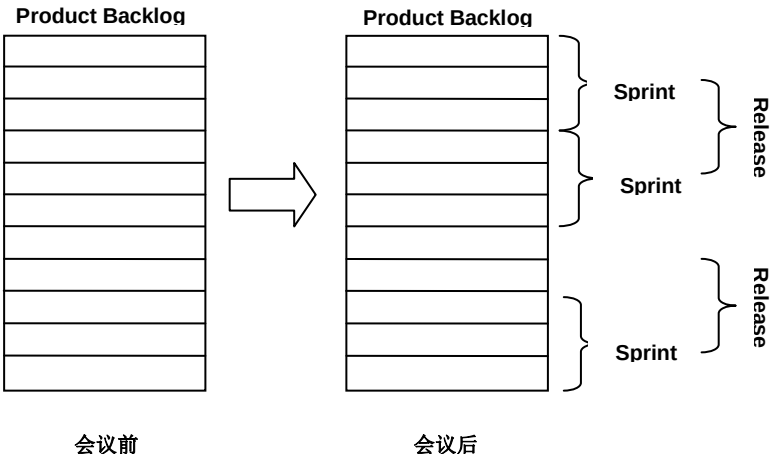


图 5-4 会议成果图（1）

2) 产品迭代计划会议

迭代计划会议在 Sprint 启动时召开，由 Scrum Master 组织。

- 会议目标：确定 Sprint 迭代计划；把即将开始的 Sprint 中的 Backlog 分解成工作任务，调整 Sprint 的 Backlog；
- 参与者：Product Owner、Scrum Master、团队成员；
- 会议准备：所有与会人员都已了解产品的释放计划；
- 会议成果：把 Backlog 分解为具体的工作任务，并形成初始的任务板。

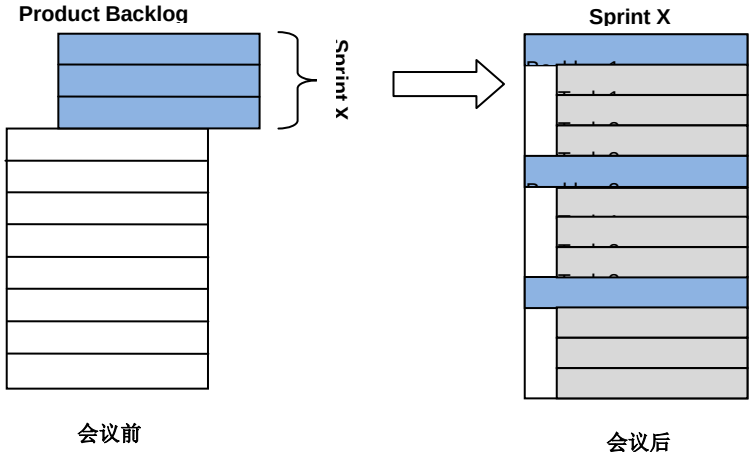


图 5-5 会议成果图（2）

3) 执行跟踪

在迭代周期内强调范围和计划的准确性，坚持两个“不变”。

在 Sprint 迭代过程中要求 2 个“不变”：Sprint 的需求范围不变、Sprint 的目标

结束日期不变。保持两个“不变”可以使团队成员精力更加集中的为达成迭代目标而努力，目标结束日期不变要求团队成员都能对所做的工作有充分准确的估计，需求不变要求 Product Owner 在迭代开始前要对产品做什么和不做什么进行仔细的评估，力求保证产品特性尽量稳定。

为了达成 Sprint 的目标，Scrum 采用每日例会的方式来跟踪任务进展，评估当前工作完成情况，安排当前工作，估计剩余的工作量，对存在的工作障碍进行识别，并给出解决计划。

每日例会(Daily Meeting)要求限定在 15 分钟以内，之所以这么做是为了保证会议的高效性，为此团队成要对 Sprint 内的任务都应有比较深入的了解。

每日例会：

- 会议目标：修订任务状态；安排下一次会议前的工作；识别工作障碍；
- 参与者：Scrum Master（会议主持）、团队成员、其它关心任务进度的人员（可选）；
- 会议准备：为了节省会议时间，所有成员对昨天所做的任务和今天将要做的任务都已做了初步的评估；
- 会议成果：得到最新的任务板、燃尽图和障碍列表，剩余工作量会根据任务板自动计算出来，燃尽图表示一种任务完成的趋势，可以大致估计出在目标日期内是否能够完成 Sprint 内的工作任务。

4) 检查

检查工作在 Sprint 评审会上进行，评审会议的目标是通过产品演示，检查是否达成本次 Sprint 的目标。会议成果是与会人员对本次 Sprint 的结果达成共识，即本次 Sprint 是成功的还是失败的，Sprint 的目标是否达成。

- 会议目标：通过产品演示，检查是否达成本次 Sprint 的目标；
- 参与者：Product Owner、Scrum Master（会议主持）、团队成员、其他干系人（主要是实施方向的人员）；
- 会议准备：与会者都已了解本次 Sprint 的目标和范围；项目组已准备好演示环境；
- 会议成果：与会人员对本次 Sprint 的结果达成共识，即本次 Sprint 是成功的还是失败的，Sprint 的目标是否达成。

5) Sprint 回顾会议

回顾会议的指导原则是在现有资源条件下把事情做到最好。

- 会议目标：回顾本次 Sprint 开发过程的经验和教训，改进过程，为后续开发工作的高效进行提供更有效的指导原则；
- 参与者：Scrum Master（会议主持）、团队成员、其它相关人员（可选）；
- 会议准备：团队所有成员都已对本次 Sprint 做了初步的总结；会议中用到工具都已准备好（红绿颜色贴纸、白板笔等）；

挂板按如下示意图布置好：

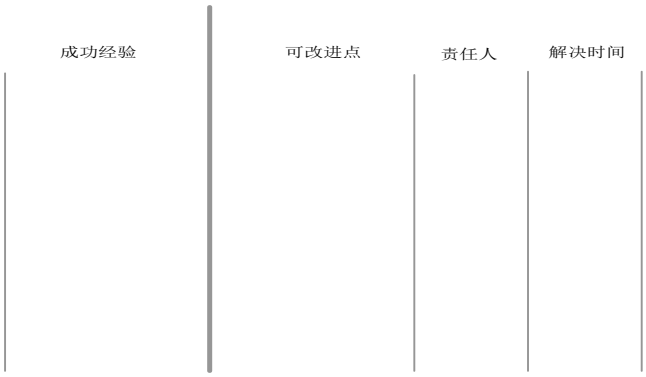


图 5-6 挂板图（1）

- 会议成果：和部门内所有项目组分享项目经验；障碍列表已被更新，并对部门内所有人公开。

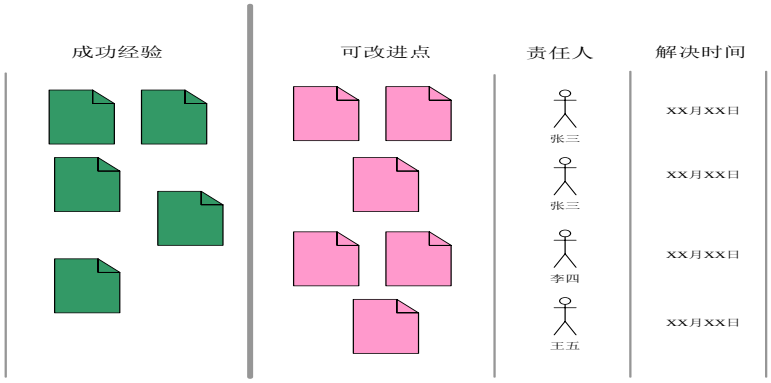


图 5-7 挂板图（2）

专家点评：

迭代开发是 Scrum 三大变革之一（另外两个是可视化和自组织），东软在实施迭代开发的实践时，与标准的 Scrum 中迭代实践相比，根据自身的特点做了灵活的适配，包括：将非产品开发相关的“人员技能培训”的工作列入迭代计划中，这样就可以有效地确保这项工作所需的时间得以分配；每个迭代开始时间根据组织需要并不一定是固定的；为保证迭代任务的完成，除了每日例会的反馈跟踪，还周期性（1 周以内）的跟踪任务进度，及时微调迭代计划；迭代内坚持需求范围不变和迭代结束日期不变；Scrum Master 主持迭代内各主要会议。

点评专家：郜晓梅

案例六：迭代式开发 （汤森路透）

1. 案例说明

1) 案例来源

汤森路透

2) 项目基本信息

在公司下一代核心平台上开发支持中国市场业务的新产品，并以此为基础，未来进一步扩展产品功能直至支持全球市场业务。

- 团队规模：10 人
- 环境：.Net 4.0, C#, WPF

3) 组织结构

标准 Scrum 组织架构。

4) 面临的问题

产品需求不断变化，产品架构复杂，全球协作成本较高，市场响应速度高，产品伸缩性要求高。

5) 解决方法

采用 SCRUM 敏捷开发方式，迭代周期为两周。主要活动包括：迭代计划、迭代评估、功能增量的开发工作、持续构建、每周演示、迭代复审、风险控制。

6) 主要收益

- 迭代支持产品经理的快速反馈；
- 迭代支持快速拥抱架构变更；
- 迭代有助于控制发布范围，降低项目风险。

2. 定义和特性说明

迭代式开发也被称作迭代增量式开发或迭代进化式开发。在迭代式开发方法中，整个系统开发工作被组织为一系列的短小、固定长度（如 3 周）的小项目，

被称为一系列的迭代。每一次迭代都包括了需求分析、设计、实现与测试，完成系统的一个功能增量开发工作。在整个过程中，不断通过客户的反馈来细化需求，开始新一轮的迭代。随着迭代的不断增加，系统的功能越来越完善。

3. 如何应用成功实践说明

考虑使您的迭代都具有相同的迭代长度。它有助于建立确立有“心跳”的项目和了解项目团队的绩效，支持创建更准确的估算。一般，在第一个迭代将攻克最困难的问题，但不可超载太多，要确保在每次迭代结束时您都可以展示进展。不要延长迭代，以完成工作，也不可以完成没有任何可展示软件的迭代。如果需要的话，可将工作分解成更小，更易于管理的任务，以实现这种平衡。

只对马上要开始的迭代进行详细规划。每个迭代开始于由整个项目团队参加的迭代计划会议。在这次会议上，迭代的目标被定义成工作项，工作项又被进一步分解成任务（如图一所示）。团队成员签署并领取任务，给出他们的估算。一旦迭代开始后，应该被允许继续根据计划执行，尽可能减少外部干扰。

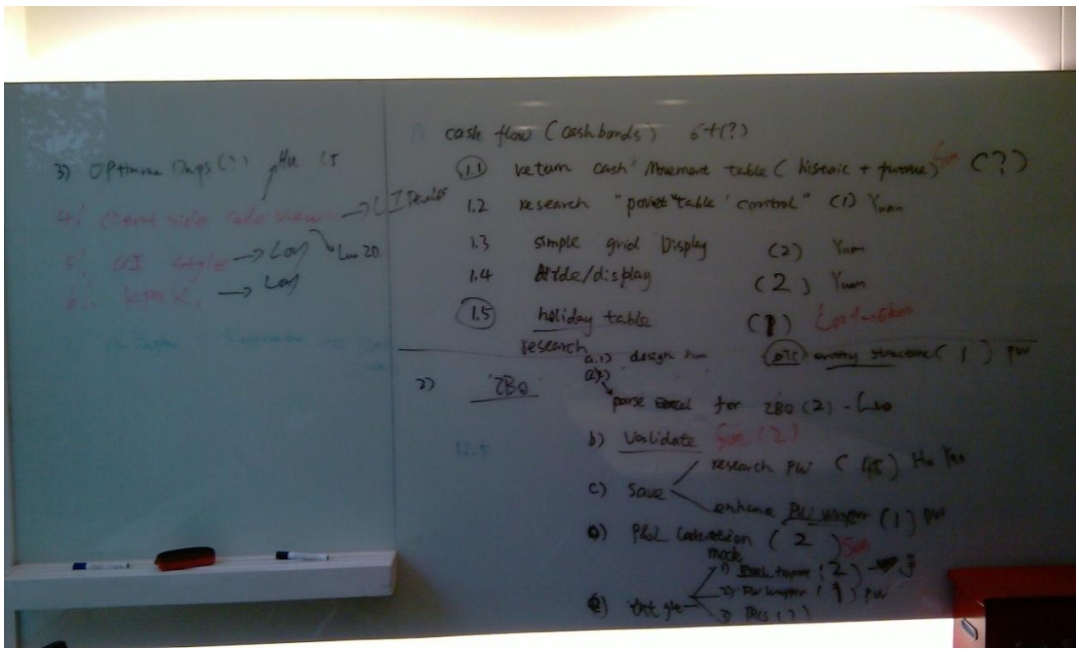


图 6-1 迭代会议中被定义的工作项及相关细分后的任务

在迭代的过程中，团队成员通过会议，提供其任务状态。会议的频率取决于球队：可以每天或一个星期几次，或每周。迭代结束时，任何尚未完成的工作从迭代中删除，并重新分配给下一个迭代。

所有迭代遵循相同的工作模式，这有助于团队把重点放在一个迭代的开始、中间和结束时的具体活动。例如，一个迭代过程中执行的一些常见的活动是：

1) 迭代计划

2) 迭代架构工作

3) 微功能增量的开发工作

4) 建立每周稳定的构建

5) Bug 修复

6) 迭代审查和回顾

专家点评：

这是一个将 Scrum 框架应用于新产品开发的典型案例。通过对敏捷开发原则的严格遵守，如：固定迭代长度、不可超载、每迭代完成可展示软件、只为下一个迭代做详细规划等，使敏捷开发模式的价值得到了充分发挥。

点评专家：李春林

案例七：用户故事 （IBM）

1. 案例说明

1) 案例来源

IBM

2) 项目基本信息

个产品的新版本的开发

- 团队规模：200+人
- 环境：Java,Web2.0, C, DB2

3) 组织结构

Scrum of Scrum 组织架构

4) 面临的问题

发周期确定，需要严格按照计划的时间发布新版本。新的技术架构具有不确定性。行业变化快，要求产品能够迅速响应市场需求。

5) 解决方法

绕用户故事进行计划、估计。产品负责人根据市场需求，随时修订产品订单的优先级，确保敏捷团队在每一个迭代周期能够交付相应的用户故事。

6) 主要收益

- 能够及时调整计划，相应市场变化；
- 迅速交付高优先级的用户故事；
- 避免部分完成的工作带来的浪费。

2. 定义和特性说明

户故事描述了用户将要利用该软件产品实现的目标，获得的商业价值。通过用户故事推动敏捷团队内部以及与所有利益相关人的协作，确保敏捷团队专注于

高优先级的用户故事，在每个迭代周期都有能够交付的功能。用户故事的格式模版如下：

a <role> I want to <goal> so I can <business value>

为<角色>我想要<目标>以便能够获得<商业价值>

3. 如何应用成功实践说明

将传统的需求文档转化为一系列的用户故事。用户故事应该简洁明了，而不是需求规格说明，用户故事应该是可测量的，能够演示的。然后将用户故事按照主题进行分组。分析确定每个主题的优先级。根据主题的优先级，确定版本计划。对每个主题里的用户故事确定优先级，构建该主题的产品订单。

针对每个主题，组建一个敏捷团队，团队成员由具备相应技能的开发人员、测试人员、文档人员组成。

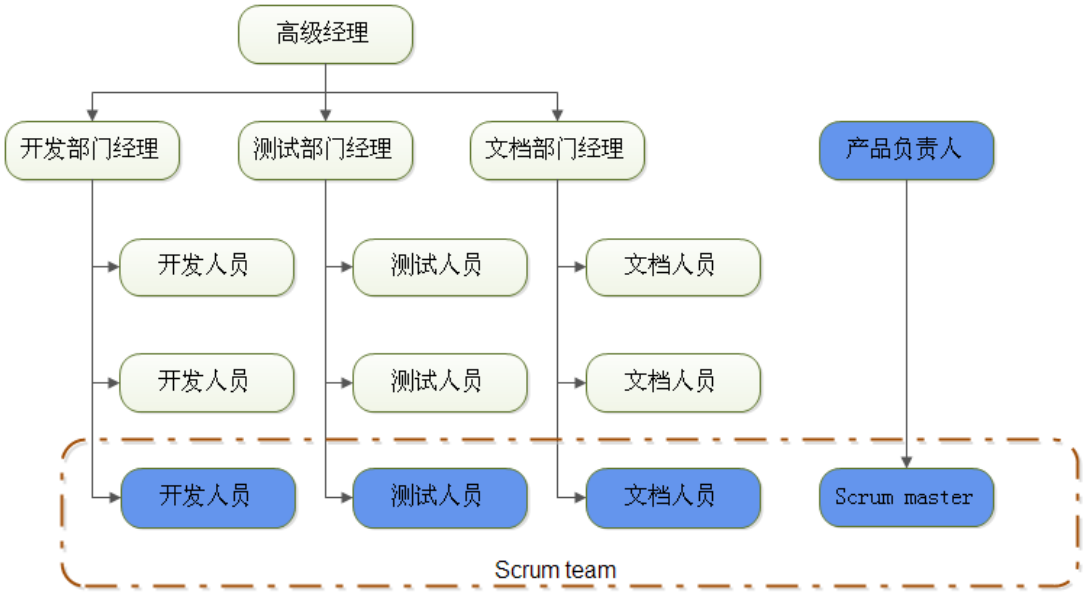


图 7-1 Scrum team

敏捷团队对用户故事以故事点进行相对大小的估计。对于高优先级的用户故事，确定任务并进行工作量的估算，估算敏捷团队的交付能力。在项目进行的过程中，根据实际完成的迭代周期的交付情况校准敏捷团队的交付能力，作为后续的迭代周期的计划的参考，必要时，会在各个敏捷团队之间调动资源，确保版本计划的顺利完成。

在每个迭代周期开始的时候，从产品订单中选出高优先级的用户故事，制定迭代计划。在迭代周期结束时，演示已经完成的用户故事，由产品负责人评审确认是否满足需求。对于没有完成的或是评审中提出异议的用户故事，我们通常有两种处理方式，一种是必需马上处理的，将没有完成的工作或是修改任务做为债务放到下一个迭代的计划中，作为高优先级的任务；另一种是不太紧急的，创建新的用户故事跟踪未完成的任务或是修改意见，放到产品订单中评估优先级。

用户故事有助于敏捷团队从真正的用户的角度出发去理解需求。通过明确的角色定义，能够有针对性地把干系人引入到用户故事的讨论中来。通过以用户的角度来描述用户故事，能够体现用户如何使用我们将要交付产品，反映了用户的实际需求，而不是技术实现方案。通过用户故事推动开发团队与产品负责人之间充分的沟通，并对用户故事的验收标准达成共识。而每个用户故事的商业价值为确定用户故事的优先级提供了量化的参考，确保团队专注于高投资回报比的用户需求。

专家点评：

随着开发进程的推进，不断变化的市场需求、不断细化调整的产品功能、不断增加的要紧急交付的项目数量，这些都在不断的冲击着产品开发的既定规划，如何能够以用户的角度来规划、开发，并持续验证软件是否符合用户场景，缩短开发到交付的周期，写好、用好用户故事是很有效的方法实践。

点评专家：张忠

案例八：用户故事 （用友软件）

1. 案例说明

1) 案例来源

用友软件股份有限公司

2) 项目基本信息

在开发 XX 银行资金交易系统的过程中，需要有一种有效的沟通手段，即能让需求分析人员和开发人员都能够明白客户的需求和意图，这就要求这种沟通手段既不能太偏重于业务背景又不能太偏重于技术名词，通过 UserStory 能够很好的解决这一问题。

项目基本信息如下：

- 团队规模：6 人
- 环境：NC5.6 平台 & ORACLE10G

3) 组织结构

标准 SCRUM 架构

4) 面临的问题

USER STORY 难以编写，并且会占用开发人员的工作量。

5) 解决方法

- User Story 的形式要很简单，这样人们可以很容易地掌握编写 User Story 的方法。这样就可以保证是由与项目相关的领域专家们来写 UserStory，而不是开发人员；
- User Story 有一个通用的公式格式来编写，作为<某个角色>，我可以<做什么>，以完成<什么目的>。

6) 主要收益

通过使用 USER STORY 来描述用户需求的每一个 CASE,使业务需求人员能够更好的描述客户的需求，开发人员也能够最大程度的理解需求人员的意图，使沟通

效率大为加强。

2. 定义和特性说明

用户故事（情景），从用户的角度对系统的某个功能模块进行简短描述。

3. 如何应用成功实践说明

我们通常把 User Story 写在一张小卡片上，同时在卡片上标明它的优先级和预计完成时间，以便开发人员根据任务的优先级来制定 Sprint Backlog。

一个 User Story 的大小和复杂度应该以能在一个 Sprint 中开发完毕为宜。

专家点评：

用友的成功实践再一次证明：用户故事形式简单，故事写作者使用起来方便；用户故事从用户角度出发描述需求，利于用户、需要分析人员和开发人员之间的沟通。

点评专家：郜晓梅

案例九：用户故事 （石化盈科）

1. 案例说明

1) 案例来源

石化盈科信息技术有限责任公司

2) 项目基本信息

在企业能耗评价系统二期开发项目中使用 Scrum 模型，每次 Sprint 周期相对固定，约为 2 周左右，要求团队快速响应。

项目基本信息如下：

- 团队规模：7 人
- 环境：VSTS2010 & sql server

3) 组织结构

Scrum 组织结构。1 名 Product Ower,1 名 Scrum Master 及 5 名团队开发/测试人员

4) 面临的问题

- 团队成员对业务知识掌握不够；
- Product Owner 公务繁多，还负责其他项目，不可能实时支持项目的全过程。

5) 解决方法

每次冲刺前先召开计划会议，由 Scrum Master 主持，会议之前 Product Owner 确认本次冲刺所涉及的用户故事。计划会议一般选择在周二的上午进行，时间为半天到一天。

计划会议上，Product Owner 向项目成员详细解说某一个用户故事，确保团队成员准确把握用户需求。Product Owner 通常还会在计划会议上给团队成员培训业务知识，各成员进行学习与交流。

6) 主要收益

- 在冲刺计划时，团队成员与 Product Owner 就用户故事的理解达成共识，降低了开发过程中由于误解需求造成返工的风险；
- 团队成员聚集在一起学习业务知识，有助于业务能力的提高，加强彼此之间的交流与沟通。

2. 定义和特性说明

用户故事（User Story)描述的是一件用户通过系统完成他一个有价值的目标的事。用户故事只是描述系统的外在行为，完全忽略系统的内部动作。

3. 如何应用成功实践说明

尽量在系统的开始阶段，尝试找出所有的用户故事。然后，为每一个用户故事评估 Story Point（复杂度）和 Stack Rank（优先级）。

由产品经理(Product Owner)确认每一次冲刺需要完成哪些用户故事。召开计划会议，将每一个用户故事分解为工作任务（Task)，由团队成员认领并承诺在规定的时间内完成。

每一次冲刺完成后，召开评审会议，与 Product Owner 进行用户故事的确认，对于确认通过的用户故事，可以将关联的工作任务关闭，对于确认未通过的用户故事，考虑是否将本次冲刺计划延迟或将这些用户故事放到下一轮冲刺计划中。

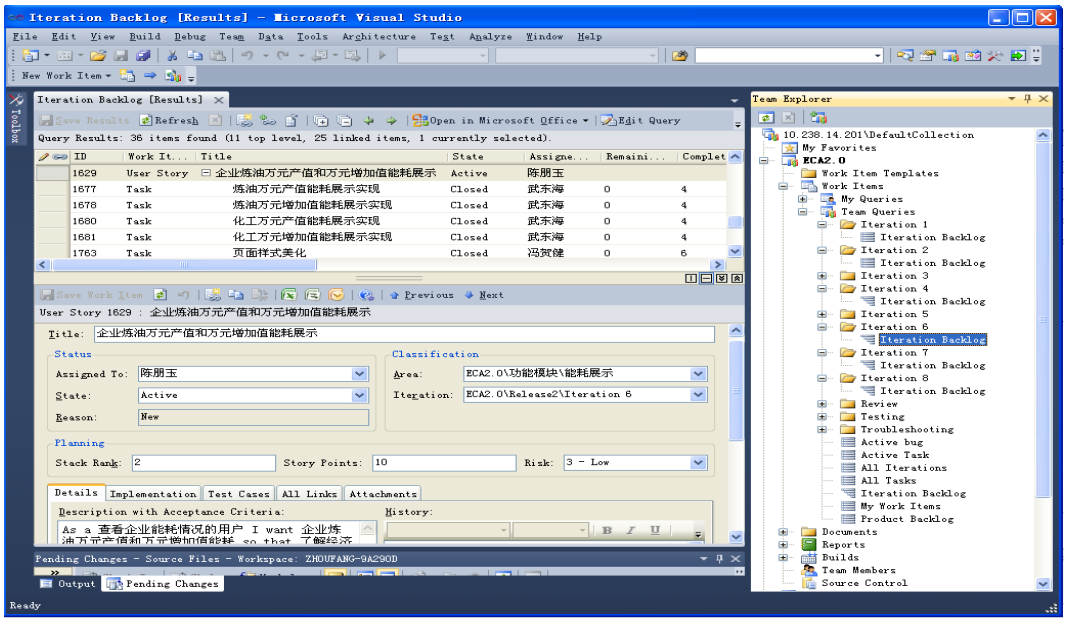


图 9-1 使用 VSTS2010 管理用户故事

专家点评：

本案例清晰地描述了基于自身项目背景和环境，团队在每一个环节怎样使用用户故事。很具体、很规范，是具体操作的有效参考。——只是此方面标准理论和案例已比较普遍。

依个人视角，User Story 本身作为需求描述的技术，我的兴趣首先是“怎样写出好的用户故事”。在最初使用 User Story 技术时是否有什么不适？后续是否找到了改善的途径并有突破？希望作者有机会时和我们进一步分享。

点评专家：刑雷

案例十：Task Board（特艺）

1. 案例说明

1) 案例来源

特艺（中国）科技有限公司

2) 项目基本信息

Research 项目，研究项目“不好做计划”。

项目基本信息如下：

- 团队规模：4-5 人
- 环境：主要交付物：算法、实验用代码、demo、技术调研报告等

3) 组织结构

标准 Scrum 组织架构

4) 面临的问题

在使用 Task Board 之前，由于 research 工作的相对独立性，站立会议中讨论的进展有时相关性不大，也不直观，使得站立会议的效率不高。Retrospective Meeting 中讨论的结果经常忘记实施。

5) 解决方法

Task Board + Sprint Burn-down Chart + Retrospective Result

6) 主要收益

- Sprint 的任务及进展全部显示在公共区域的墙上，使得团队成员对 Sprint 的进展了解得更加直观，有效提高了站立会议的效率；
- Retrospective Result 也贴于明显的位置，提醒大家在当前 Sprint 中需要注意的事项。

2. 定义和特性说明

在 Scrum 团队的共有区域，显示该团队正在进行的 Sprint 的进展。

3. 如何应用成功实践说明

1) 参与人员：Scrum 团队所有成员，由 ScrumMaster 组织。

2) 频率：每次站立会议之中更新 Scrum Board 的内容。

3) 内容：

- Sprint 中的任务：“To Do”，“In Progress”，“Done”；
- Sprint 进展：Sprint Burn-down Chart；
- 本次 Sprint 的重要目标；
- 团队注意事项：Sprint Retrospective Meeting 的结果，包括“What to Keep”，“What to Change”，“What to Try”。

4) Task Board 的地点：每个团队成员都能随时看到的公共区域（最好坐在自己的座位上就能看到，不要在某个封闭的会议室中，否则不够直观）。

图 10-1 Scrum Board

5) 方法：

- Sprint 开始的时候，在 Sprint Planning Meeting 之后，ScrumMaster 负责每个 PBI 和相应的 tasks 都写到卡片上，并且贴到 Task Board 上；
- 在站立会议中，每个成员在说明自己的工作的时候在相应的 Task card 上更新本 task 还需多少小时完成；

- 如果某 task 存在一定风险，则用红笔标出风险内容，以便于跟踪；
- 团队成员，尤其是 ScrumMaster 分析 Sprint Burn-down chart，看当前 Sprint 的进展是否有异常；
- 团队成员一起查看 Sprint Retrospective 的内容，并互相提醒在本次 Sprint 中是否实施。

图 10-2 Scrum 小组成员在 Task Board 前开站立会议

专家点评：

首先肯定，从物理材质上讲，这是我见过的最漂亮的白板！水平泳道和垂直泳道贴得整齐又颜色鲜明。摆在显著位置而且是玻璃的，“透明度”真的很高！

话说回来，本实践的背景描述中有一句“研究项目‘不好做计划’”，我本人对此很有兴趣。非常希望看到对于最难作严谨估算的研究项目，我们的 Technicolor 同仁是怎样从白板获得帮助的。很遗憾没有看到。至少实践中的描述，我觉得更多的是，白板怎么帮助了我们的“监控”，而非“计划”。

我个人武断地猜测一下，该团队很好地利用了白板的 WIP（Work In Progress）属性，——当然这也是很有价值的，——但很可能不是 WIP Limit（需要在限定时间盒内完成既定的任务）。那么这种运作方式未必是可以称作 Sprint 的。

点评专家：刑雷

案例十一：任务看板（英孚教育）

1. 案例说明

1) 案例来源

英孚教育

2) 项目基本信息

整个团队都能随时看到的大白板。

3) 组织结构

团队所有人员

4) 面临的问题

开发的过程中，不光是管理着项目的领导还是团队的成员都希望有一个地方能让他们简单而快速的发现项目的进度，并且能在困难发生前，看出问题，并提早解决。

传统的 mpp 管理方式，不光在更新上费时，同时也并不是每个人都会想到去看，也不是每个开发人员都看的懂。

5) 解决方法

项目看板。它被选在一个项目组成员都能方便看到的地方，可能是小办公室门口，也可能是必经的路口，上面贴满了任务表，并且随着进度的更新，变换着状态。无论是谁，都可以轻而易举的结合看板上的三个模块，获得项目当前的状态。

6) 主要收益

- 团队成员间，作为一个记录板，很容易看到别人正在进行的任务；
- .任何问题，方便的找到责任人；
- 帮助 scrum master 预估所会遇到的困难与阻碍；
- 让团队时刻能更新手头的障碍，并且有预期的看到障碍被解决的时间方

便他们合理安排自己的时间；

- 老板无需和团队沟通，就能知道团队的进度；
- 让团队意识到自己当前的进度，调整开发速度；
- 瓶颈的消除，变的积极，因为谁都不会想在 action board 看到自己的名字。

2. 定义和特性说明

任务看板被分成了三大块：sprint backlog，action board，和燃尽图。

1) sprint backlog

主要张贴此次 sprint 所承担的用户故事，及完成故事所需要的任务列表。

图 11-1 sprint backlog

这块被分成了 5 种状态：用户故事；代办任务；进行中的任务；待测试的故事；完成的故事及任务。

- 在 sprint 计划会议之后，我们将把所有的故事，按照每个故事的责任人，放在用户故事这一栏。 把完成故事所对应的任务，贴在代办任务一栏；
- 在 sprint 开始之后，组员各自把每日的任务挪到进行中任务一栏；
- 在每日站立会议时，挪走完成的任务至完成一栏，并挪入明日的任务到进行中；

- 当所有对应用户故事的任务都被完成时，在待测试故事这一栏打勾。这意味着测试结束，等待 QA 验证。当所有故事都被打上勾而没有被移到完成状态的时候，那就是告诉团队，需要帮助 QA 一起测试了；
- 当用户故事和所有任务都被挪到完成状态时候，sprint 顺利结束。

2) Action board

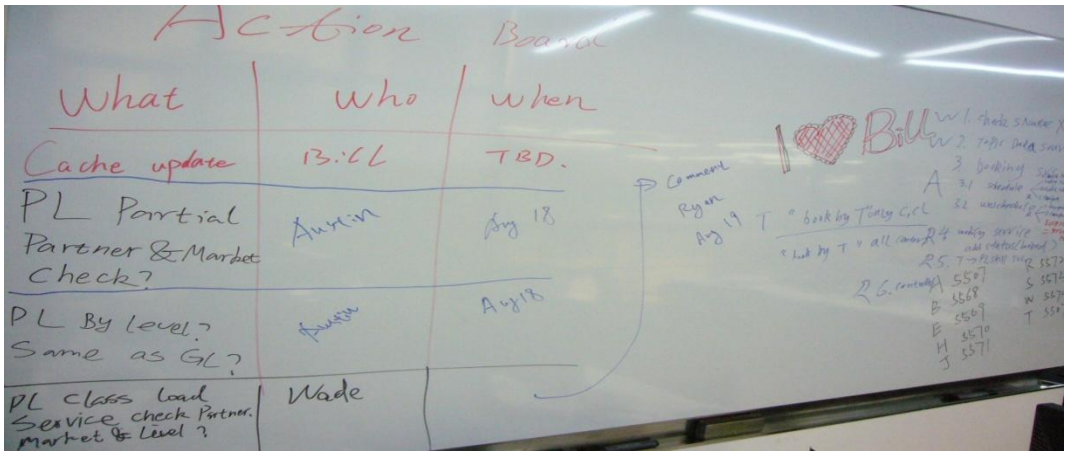


图 11-2 Action board

记录阻碍团队顺利完成 sprint 的任务，上面有三栏，分别是：任务描述；预计解决时间；负责人。

3) 燃尽图

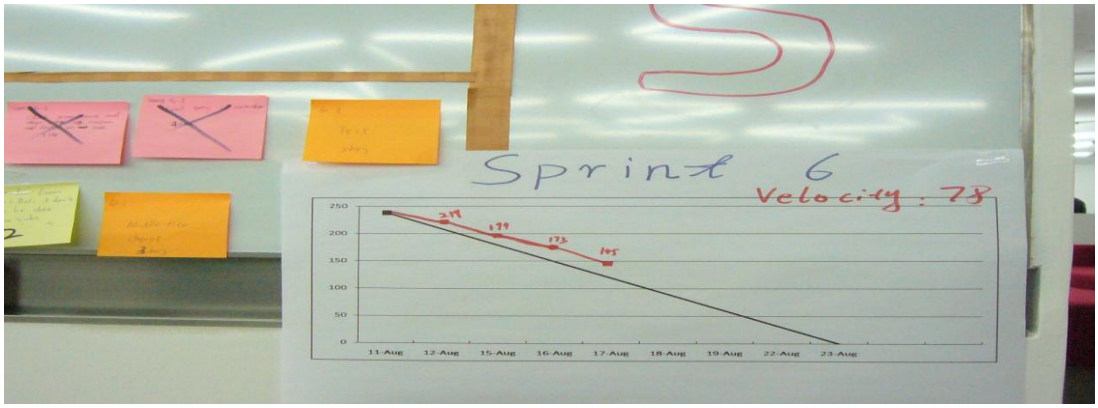


图 11-3 燃尽图

记录团队每日所剩故事点和计划所剩故事点的图表，帮助跟踪团队开发速度。

专家点评：

英孚的任务看板案例将 sprint backlog，action board 和 burndown diagram 结合在一起，是一个有价值，值得分享的案例。通过这个任务看板，Sprint 的执行状态以及团队目前碰到的阻碍一目了然，有助于团队成员了解项目进度，调动工作积极性。

在团队成员能力相当的前提下，SCRUM 团队也可以尝试先将所有 task 集中在一块，团队成员在完成目前任务后，从任务集中根据优先级和自己的兴趣，挑选新的任务。

点评专家：黄晓倩

案例十二：验收测试驱动开发 （广州畅盟）

1. 案例说明

1) 案例来源

广州畅盟信息科技有限公司

2) 项目基本信息

管理平台，B/S 架构，使用迭代式软件开发，3 周为一个迭代周期，要求测试团队能迅速反馈系统质量。

- 团队规模：10 人
- 主要交付物：Java

3) 组织结构

标准 Scrum 组织架构

4) 面临的问题

难以有效开展应用版本的功能测试，每次迭代后界面变化比较大，传统自动化工具录制的脚本维护工作量大，难以快速反馈系统质量状况。

5) 解决方法

采用验收测试驱动开发实践，引入针对业务功能测试的自动化 GUI 测试（关键字驱动框架+IBM Rational Functional Tester）。

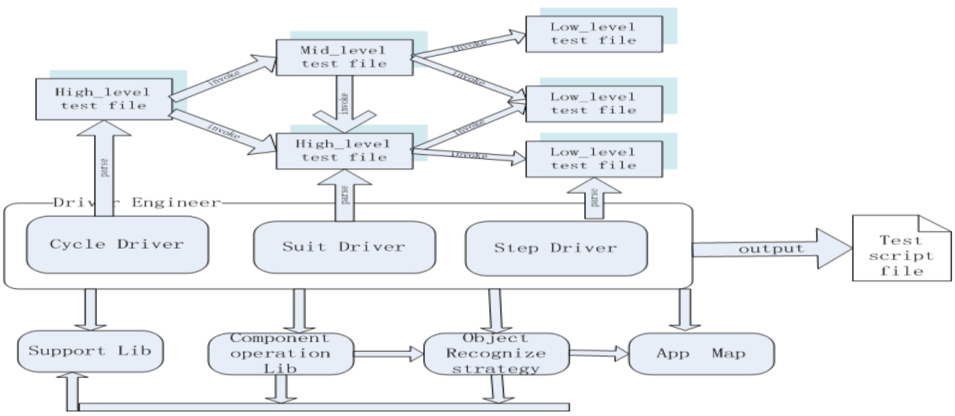


图 12-1 关键字驱动框架图

6) 主要收益

- ATDD 保证开发出的软件满足正确的功能需求；
- ATDD 通过编写验收测试清晰定义了故事完成标准，让团队明了“知道我们在哪儿”及“知道何时停止”；
- ATDD 让团队（客户、开发人员、测试人员等）为一个目标努力，创造了协作性更强环境，而且明晰了验收标准。

2. 定义和特性说明

ATDD(Acceptance TDD 是测试驱动开发核心理念的一个扩展）驱动整个开发过程。在用测试驱动开发实现某个具体功能之前，将会首先编写功能测试或验收测试，从系统功能角度驱动开发过程。

ATDD 能有效促进客户、测试人员、开发人员之间的交流协作。通过增量式开发软件，用面向客户的测试作为讨论和反馈的基础，ATDD 能使整个团队紧密协作。依照客户指定的优先级，在整个项目过程中不断开发出可用的具体功能，赢得客户信任并增强自信，交流也因相互之间有了共同语言而更加高效。

3. 如何应用成功实践说明

ATDD 周期包含挑选故事，给故事写测试，实现测试以及最后实现功能这四个步骤。

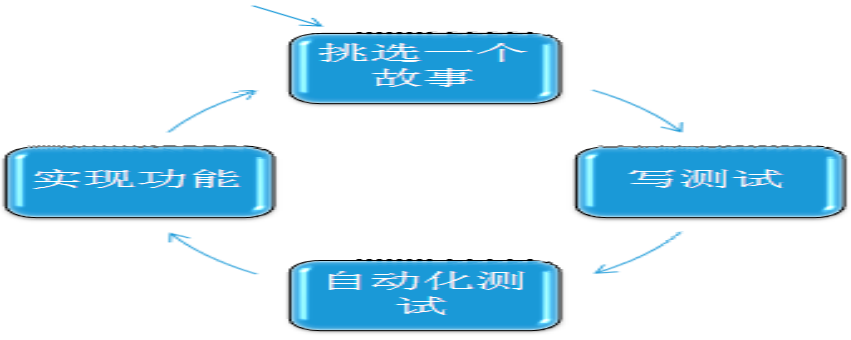


图 12-2 ATDD 周期

一个故事的“4 步周期”如何嵌入整个迭代中，可采用以下方式。迭代开始前的计划会议，这个会议中，客户会决定下一个迭代该做哪些故事，及给故事排优

先级。ATDD 一般情况下可先选择较高优先级的用户故事，然后写测试用例，接着把测试用例转化成自动化的可执行的测试，最后实现功能，让验收测试通过。

在为故事编写测试用例的时候，客户是最适合写验收测试的人，而对于实际情况最好能由客户、开发人员、测试人员共同参与编写验收测试。测试人员和开发人员有时会在没有客户参与的情况下编写验收测试，这时候，必须保证将来会把这些测试同客户进行讨论及完善。在编写测试用例时只关注完成故事必须要做的几件事情，形成简单清单，然后细化测试，添加更多细节，讨论各部分如何工作，确定客户对界面是否有特别的要求等。根据功能的类型，对应的测试可以是一组顺序操作，或者是一组输入及对应的输出。在故事的实现过程中还难免会发现原有测试外的新的测试，合理的做法是，在征求客户意见后，我们应该把新的验收测试加入到清单中。

在完成验收测试用例编写后，就需要把测试用例转化成可自动运行的测试脚本，且执行完成后会返回一组成功或失败的结果（通常通过验证点方式）。在把测试用例转化成可自动运行的测试脚本过程中，团队常会引入一些相对成熟的测试框架及商业化自动化测试工具。这些框架和工具一般可化分为 API 级别功能测试（如采用 Fit 和 FitNesse 框架）和自动化 GUI 测试（如可采用关键字驱动框架和 IBM 自动化工具 Ratioanl Functional Tester 等），团队可以根据应用及自身情况选择使用。API 级别功能测试，帮助客户团队编写和理解驱动开发的面向业务测试，使团队能够在不牵涉 UI 层的情况下测试业务逻辑。而自动化 GUI 测试在实际应用中，自动化测试框架成熟度更显得重要，框架可通过测试用例、GUI 对象、数据的分离，大大减少因为频繁的界面变化而对已有自动化脚本的影响。

最后一步就是实现功能，让新加入的验收测试通过。在实现过程中可采用 TDD 方式。这样在代码层面，采用 TDD 方法以测试驱动的方式编写代码。在软件的特性和功能层面，使用 ATDD 方法以测试驱动的方式构建系统，这两个不同层面上结合使用测试驱动，既能保证软件的内部质量同时能保证开发出的软件能满足正确的功能需求。

另外在使用 ATDD 时候可以同持续集成及自动构建等最佳实践结合使用以最大化达到测试生命周期的自动化。

专家点评：

这一实践可行性很高。从用户故事的挑选、编写验收测试用例、编写自动测试脚本，到功能的最终实现；从高层的 ATDD 到实现层的 TDD，都有很高的可操作性。ATDD 和 TDD 不仅能够避免和减少缺陷的产生，还可以促进用户、开发人员、测试人员之间的沟通交流，避免由于对需求理解的偏差造成的返工。先测试后开发，减少由于需求变化带来的返工。

点评专家：刘德意

案例十三：测试驱动开发（石化盈科）

1. 案例说明

1) 案例来源

石化盈科

2) 项目基本信息

Research 项目，研究项目“不好做计划”。

项目基本信息如下：

- 团队规模：7 人
- 环境： VSTS2010 & sql server

3) 组织结构

Scrum 组织结构，1 名 Product Ower，1 名 Scrum Master 及 4 名团队开发人员。另配置一名软件功能测试人员，来自测试部门。

4) 面临的问题

- 开发人员中有一名为工作年限不长的刚毕业学生，对测试代码的设计能力把握不足；
- 经常需要随时提供可运行程序；
- 功能测试人员通常在系统集成之后才进入项目工作，把握需求的准确度不高。

5) 解决方法

- 项目团队内部自发组织单元测试的学习与培训；
- 在编写完单元测试代码后，编写程序代码，由单元测试代码驱动程序代码的编写；
- 将单元测试发现的代码编写问题更新到代码规范中，在项目团队中共享；

- 功能测试人员参与计划会议，充分理解每一项用户故事，对表述不当的地方尽早提出改进建议，在开发人员编写单元测试用例的同时根据用户故事开始编写功能测试用例；
- 通过每日构建测试，保证服务器代码的稳定行和高质量。

6) 主要收益

- 测试驱动开发，尽早发现程序中存在的缺陷，提升了研发效率；
- 大大提高了代码质量；
- 提高了测试效率；
- 团队成员在学习过程中得到了能力提升，培养了良好的编码习惯。

2. 定义和特性说明

测试驱动开发就是在明确要开发某个功能后，首先思考如何对这个功能进行测试，并完成测试代码的编写，然后编写相关的代码满足这些测试用例。然后循环进行添加其他功能，直到完成全部功能的开发。

3. 如何应用成功实践说明

先编写测试代码，再进行系统功能的代码开发，测试通过后进行代码重构、细化。通过测试来推动整个开发的进行。这有助于编写简洁可用和高质量的代码，并加速开发过程。

过程如下：

1) 明确当前要完成的功能。可以记录成一个 TODO 列表；

2) 快速完成针对此功能的测试用例编写；

3) 测试代码编译不通过；

4) 编写对应的功能代码；

5) 测试通过；

6) 对代码进行重构，并保证测试通过；

7) 循环完成所有功能的开发。

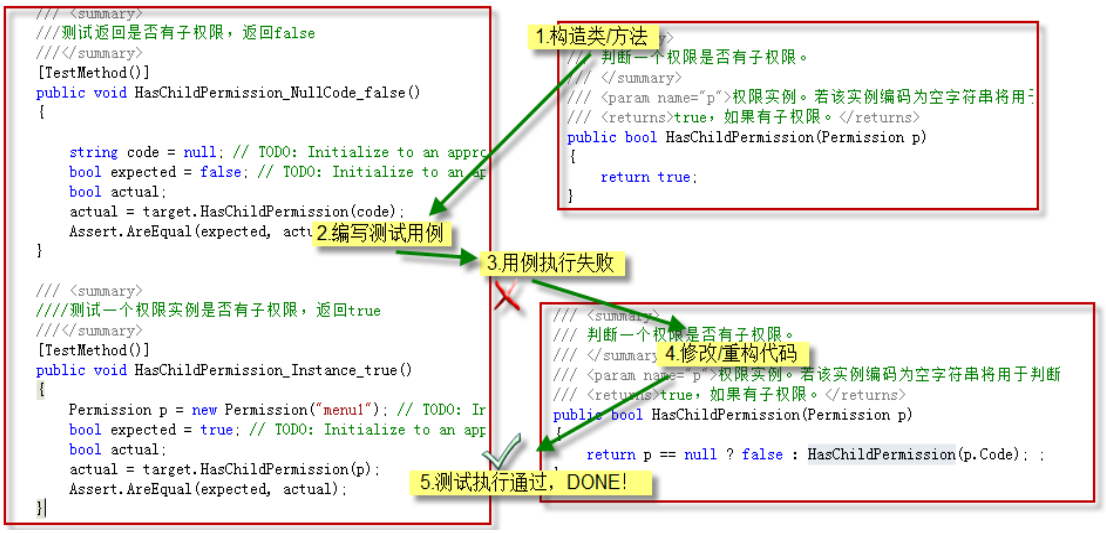


图 13-1 测试驱动开发示例

专家点评：

代码开发过程中，除了依照单元测试代码开发，还可以依照测试人员提早写完的测试用例文档作为指导进行开发，这样就即保证了单个单元的代码正确性，又保证了整体功能的业务正确性。

点评专家：高航

由单元测试代码驱动程序代买的编写，可以有效减少缺陷的发生。随时将发现的代码编写问题总结入代码规范中，是提高代码质量的一个有效途径。功能测试人员早期介入项目，提高测试效率。另外，本实践还没有忽视后期代买重构的工作，确保了代码的高质量。

点评专家：刘德意

案例十四：敏捷测试之测试保护开发（上海宝信）

1. 案例说明

1) 案例来源

上海宝信软件股份有限公司

2) 项目基本信息

这是 BS 架构平台中间件，封装了界面、工作流、数据库、常见控件等等，在宝信各大信息化项目中应用了 200 套以上。

- 团队规模：14-22 人
- 主要交付物：Java, C#

3) 组织结构

普通项目组织+独立测试组

4) 面临的问题

编码时难以避免注入缺陷，新功能开发影响老功能，回归测试不及时，而且没有足够的时间。

5) 解决方法

- 采用每日集成+测试保护开发；
- 主要活动：建设自动化测试集。

6) 主要收益

前期就能发现缺陷，后期能保障代码修改，回归测试开展快速方便。

2. 定义和特性说明

在敏捷开发但有保留独立测试团队的情况下，通过测试保护开发来协同开发团队和测试团队在项目前期的合作，期望后期测试工作量减少，测试人员可以有

更多的工作量投向前期。从 ATDD 本身而言，是不区分开发人员和测试人员的，但要求快速一惯。测试人员先编测试用例，再由开发人员开发的模式是不可行的。

因此提出测试保护开发，做法是：

1) 开发人员根据需求，编写代码，实现界面和接口，对于测试保护开发而言，不强制要求采用 TDD，采用 ATDD 下，再用测试保护开发是没有必要的；

2) 几乎同步，测试人员编写自动化测试代码，主要是黑盒自动化测试，也不排除白盒自动化测试，加入到每日测试当中，在每日集成时一起执行；

3) 一般保证，代码出来后的第 3 天之内，相关的自动化测试代码开发完成。

3. 如何应用成功实践说明

针对敏捷开发，测试团队采用测试保护开发。测试保护开发过程中，有两种形态：

1) 测试人员紧密跟踪开发人员，理解需求分析；开发人员进行功能设计，测试人员同时进行对应测试用例的设计；界面确定后，测试人员马上跟进开发对应的自动化测试用例；在每日测试环境中对每日集成的工作产品进行较验，以达成保护开发的目地。

2) 开发人员按照 TDD 负责编写单元测试和功能代码之后，如果测试人员有资源和能力，测试人员补充单元测试（包括接口测试），全部单元测试（含接口测试）进入每日集成。

测试保护手段对于减少测试缺陷，保证开发质量有着重要的意义。这已经被过程能力基线及各项目组的有效实践所充分证明。现有的单元测试和自动化测试等测试保护手段，对于简单的基本功能或者接口类型的产品效果较好。但是对于复杂业务场景的保护可能还略嫌不够。

一方面，复杂业务逻辑的测试用例的识别存在进一步提升的空间。对于简单的场景，比如一个典型的维护页面，对于它的验证场景比较确定也较有经验。但是对于一个复杂的业务场景，由很多个业务逻辑相互交织，对于它的验证场景相对较复杂，并且依赖于个人的经验和对总体逻辑的把握程度，容易存在疏漏。

另一方面，更重要的是，对于即使识别出来的复杂验证场景，为了组织相应的验证代码，需要投入的精力过大，让很多开发人员望而生畏。尤其是初始数据

的准备工作，往往为了几行核心测试代码，需要配套几十倍的数据准备等辅助代码，让很多人觉得投入产出比不高由于测试保护的作用，很大程度在于持续性和可重复性，往往在后续重构和功能新增的时候才能愈加充分的体现，因此在真正编写测试代码的当时无法深刻体会，结合上述的投入产出问题，很多开发人员会下意识的抵触并对过于复杂的场景采取回避的态度。

对于这个问题，有如下应对措施：

1) 重视并加强对复杂业务场景测试用例的设计能力提升；

2) 研究初始数据准备的方法论和辅助手段，减少开发人员的资源投入，提高投入产出，减少心理抵触情绪；

3) 在方法论上需要提炼和总结，从组织的层面，对开发人员和测试人员的相关工作进行指导；

4) 在工具如测试框架等方面要进行支撑，将开发人员从繁琐的数据准备代码工作中解放出来，将精力保证在业务逻辑验证上。

专家点评：

测试保护强调测试人员和开发人员紧密配合、高度同步，在每一步都有测试的工作参与。这种测试保护策略确保测试的持续性和可重复性，为之后对重构以及新增功能的高效测试奠定了扎实的基础。尤其对比较复杂的系统会有比较明显的效果。

点评专家：刘德意

案例十五：每日站立会议（用友软件）

1. 案例说明

1) 案例来源

用友软件股份有限公司

2) 项目基本信息

新一代产品已经发布完两个版本 11.0 和 11.2，随着产品上市用户的使用，很多的用户反馈产品的效率和稳定性存在着比较大的问题，尤其是产品的效率让用户不满意，多家客户也因为产品的效率问题要求退货。为了提高客户的满意度，提供伙伴对新一代产品的信心，经项目管理委员会批准在新一代 11.3 项目中不对产品的功能进行任何增加专门在 11.3 中针对产品的效率及稳定性进行修改。

项目基本信息：

- 团队规模：84 人
- 产品开发环境：ASP.Net,JavaScript,C#,IIS,SQL Sever2005

3) 组织结构

- 业务平台 Scrum 组（ScrumMaster 谭坦 ScrumMember:8 人）
- 业务往来生产 Scrum 组（ScrumMaster 赵子东 ScrumMeber:6 人）
- 供应链报表 Scrum 组（ScrumMaster 邹全林 ScrumMember:3 人）
- 库存 Scrum 组（ScrumMaster:司建成 ScrumMember:12 人）
- 总账 Scrum 组（ScrumMaster:慕红利 ScrumMember:7 人）
- 技术平台 Scrum 组（ScrumMaster:李胜国 ScrumMember:8 人）
- 现金银行 Scrum 组（ScrumMaster:蒋勇 ScrumMember:5 人）
- 销售采购 Scrum 组（ScrumMaster:张艳晖 ScrumMember:6 人）
- 需求组：3 人
- 项目经理：1 人
- 测试组：21 人

- 团队物理分布：江苏常州 39 人 北京：44 人

Scrum 的另一个成功之处在于大家实际上都站着。 当团队成员起立时，他们会感到不自在，尤其是当他们发言的时候。 如果每个人都站着，则会议将进展顺利可避免冗长的谈话。

会议应在每天同一时间和同一地点准时开始和结束。 这种一贯性可帮助团队确立一个模式。

2. 定义和特性说明

1) 在 Scrum 方法中，Scrum 会议非常重要，整个会议可能会比较混乱粗略，但推进进度的目标却非常清晰明确，并促使团队齐心协力朝共同目标迈进。

2) 团队应召开每日 Scrum 会议，以便确定下一天所需执行的工作，以最大可能地履行其承诺。团队的每个成员都应该描述自上次会议以来所做的工作。

3) 他们计划在当天完成的工作，以及可能对其他团队成员产生影响或需要获得其他团队成员帮助的任何问题或障碍。

4) Scrum 主管严格控制会议结构，确保会议准时开始并在 15 分钟或更短时间内结束。

在此会议中，每个团队成员都需要回答以下三个问题：

- 自上次 Scrum 以来我完成了哪些工作？
- 至下次 Scrum 之前我将完成哪些工作？
- 哪些阻碍性问题或障碍可能影响我的工作？

3. 如何应用成功实践说明

组织设施人：各 ScrumMaster

会议的主要意义：有助于团队进行自我组织，是团队成员间的一个进度协调会议。

1) 例会时间：每天早上 8：25 分，要求每个人必须在 8：25 进入会议状态。

2) 每日会议前每个人要更新自己的任务面板（每日 8：25 前）。

3) 每日会议中决定要签出的任务，并在会议后更新任务面板，并在任务便签上注明任务的签出人。

小结一下昨天完成的任务，主要是完成程度，以及遇到的问题；今天要 CheckOut 的任务。

专家点评：

这是一个比较成功的 SCRUM 日站会案例。SCRUM 组严格遵守 timebox 原则，每天的日站会准时开始，每次都严格的控制在十五分钟之内，会议的进展也严格围绕 daily SCRUM 的三个主题进行。该案例还分享了任务面板的更新方式，赞。
这同时也是一个跨地域，需多个 SRUM 组协调工作的背景下运作的一个案例，如能分享一下日站会里有关这方面的经验和教训，会更好。

点评专家：黄晓倩

案例十六：每日站立会议（英孚教育）

1. 案例说明

1) 案例来源

英孚教育

2) 项目基本信息

Sprint 当中的每个开发日的下班前 15 分钟。团队的任务看板前

3) 组织结构

- 团队所有开发人员（必到）
- PO，上级经理（列席旁听）

4) 面临的问题

Scrum 的实施变化多，速度快，所以团队之间的互相交流就变得由为重要，可能队友间平时在开发过程中已经积极响应并且互相交流，但是很多时候，交互的信息可能仅仅限于自己所开发的那一块业务及代码。

而那些团队信息、可能有互相依赖的信息、架构信息等可能会被忽略或者被小范围传播。

当然其中可能还有一些阻碍进度的困扰，有可能是团队尚未认知，但确确实实由个人发现，影响着或者将要影响团队。

我们需要有一个简短而频繁的聚会，快速收集这些信息，并且在事态扩大化之前，解决这些问题。也希望信息能及时的传递给团队的每一个成员。

5) 解决方法

我们设立了每日站立会议。用规则，限定简短，并且有效实施。

6) 主要收益

- 分享当日最新资讯（架构，业务）；
- 团队成员之间能很清楚自己的同伴在哪些任务上；

- 方便团队个人，总体把握产品进度；
- 及时发现模块之间的互相依赖，及时处理依赖关系。减少互相依赖所带来的依存；
- QA 能考虑到测试方便性，提交意见给相关程序员，调整不同任务的优先级。优先提交测试压力较大的模块，减轻 QA 后期压力；
- 能及时发现项目危机，并及时予以解决。

2. 定义和特性说明

每日站立会议，主要是每天花费 15 分钟以内的时间，大家聚集在一起。对项目进度即开发状态进行沟通。

为了严守 scrum 的时间盒概念，每日站立会议也必须是准时开始，时限内结束，每日固定时间，固定地点，固定问题。

任何与条款无关的话题，不得涉及，但是可以在结束后，分小组讨论。

这样，即保证了每日站立会有的高效率，又能保证任何与问题无关的组员被迫花时间听取别人的讨论。

3. 如何应用成功实践说明

1) 团队成员在规定的时间，去向 team 的任务看板前。轮流回答事先商定好的问题：

- 你今天完成了什么？
- 你明天打算做什么？
- 你有任何困难需要帮忙解决么？

2) 在完成任务的同时，大家移动任务看板上的标签，更新团队进度状态。当然，任何在这 3 个问题以外的问题，将会被记录，并在会后由相关队员单独讨论。

3) 每日站立会以我们将要求限定在组员之间互相沟通当日工作状态，并不涉及问题详细。

同时为了表述沟通而不是汇报，我们会建议领导不参加站立会议，就算参加，也提议他们不要在会议上发言。

4) 由于某些原因，我们组更偏好于把每日站立会议安排在下班前。原因如下：

- 保证自己能清楚且不遗漏的向整个团队汇报今天所做的事情；
- 如果有任何未能完成的任务，及时通告整个组自己所遇到的困难和阻碍，寻求帮助，同时争取在当天，解决问题；
- 这点承接第二点，如果需要当日加班完成今天所未完成的任务。相关组员，可自行安排加班，并留下相应组员。团队发挥自我管理精神解决问题；
- 根据今日任务完成状态，挑选明日任务。获取是由于估算不足需要加任务，或者是在白板上，选一个任务；
- 提交当日 impediment，如需要 scrum master 协助，那加上任务在行动看板上，作为一个任务留给 scrum master 尽快解决。同样，这也是 scrum master 明日的任务；
- 作为一个时间限点，团队成员必须在每日站立会之前完成当日任务（我们所有的任务，都规定不能超过一天，所以每天都会需要有任务完成），激发成员工作积极性。同时也逼迫他们合理的安排每日的时间。不能总用加班去弥补进度；
- 避免加班，避免拖堂。由于是在下班前，大家都会很有效率的汇报完工作，并且下班回家。当然对于那些工作狂人，我们也作为一种善意的提醒，提醒他们下班时间到了。可以回家休息了。（当然，如果真的需要加班，我们也不限制）。

专家点评：

每日站会的召开时间有人倾向于早晨，有人倾向于下班前，可能前者更多一点。其实只要是自己的团队适用，能够带来实际效果，早晨、晚上开都无所谓。文中也列出了一些在下班之前召开站会的自己团队的理由，由此可见团队是真正从每日站会中获得了很大收益的，站会没有流于形式。

点评专家：高航

案例十七：每日站立会议（汤森路透）

1. 案例说明

1) 案例来源

汤森路透

2) 项目基本信息

在公司下一代核心平台上开发支持中国市场业务的新产品，并已此为基础扩展产品支持全球市场业务。

项目基本信息如下：

- 团队规模：10 人
- 环境：.Net 4.0 , C# , WPF

3) 组织结构

标准 Scrum 组织架构

4) 面临的问题

产品需求不断变化，由于架构局限，模块之间耦合性高且单元测试覆盖度低，功能集成效率低下。

5) 解决方法

采用站立会议的方式，除了日常站会内容外，每日会议后，由任务负责人牵头同步任务进度情况。

6) 主要收益

提高集成效率、适应快速需求变化。

2. 定义和特性说明

每日站立会议是 SCRUM 方法中的一条关键实践。

3. 如何应用成功实践说明

站立会议通过每天面对面的沟通，可以:

1) 快速同步进展，让项目组内部的员工互相了解彼此的进展，从而了解本项目的整体进展。

2) 培养团队的文化，让每个人意识到：我不是一个人在战斗，我们是一个团队。所以站立会议不能用邮件替代，不能在电脑旁，必须眼睛看着其他的成员，而不是电脑。

3) 项目的延期源之每天的延期，所以要每天实时跟踪进展，站立会议必须每天定时定地点召开。

4) 每次会议不超过 15 分钟；如果要讨论技术问题，会后单独开会，少数人参与讨论。为了避免开长会，每位成员一般只需回答 3 个问题：

- 昨天你完成了什么？
- 今天要做什么？
- 有什么需要别人帮忙的？

其他问题不在此次会议上讨论，而是会后讨论。

5) 在开站立会议时，不是每位成员给项目经理汇报工作，而是大家是平等的，是给项目组的所有人一起同步进展。

图 17-1 项目进度同步白板

专家点评：

从“How”到“Why”，该案例给出了执行站立会议足够的细节和背后的目的，值得借鉴。

点评专家：李春林

案例十八：每日站立会议（用友软件）

1. 案例说明

1) 案例来源

用友软件股份有限公司

2) 项目基本信息

医疗基层卫生服务系统产品研发

项目基本信息如下：

- 团队规模：20 人
- 环境：JAVA、用友 UAP

3) 组织结构

矩阵式组织架构，包括需求、UE/UI、开发、测试来自不同部门的人员组成产品团队，参照 Scrum 组织架构，产品项目经理为 Scrum Master。

4) 面临的问题

产品研发小组内部或小组之间沟通交流不够，导致需求已完成但设计人员不清楚迟迟不开始的情况，或设计的内容与需求不符，但开发出来才发现，导致修改工作量大。

5) 解决方法

采用每日站会，加强沟通，团队成员互动，了解相互的情况、问题、计划，及时发现协作之间不一致的问题。

6) 主要收益

产品研发流程运转效率大大提升，研发过程中的问题和困难都能及时得到解决。

2. 定义和特性说明

基于 Scrum 的每日站会制度，每日固定地点、固定时间，团队所有人员站立参加，时间不超过 15 分钟。每个人回答三个问题：

- 我昨天完成了什么任务？
- 我今天打算做什么任务？
- 我遇到了哪些障碍或困难？

回答的形式与目的不是向领导汇报工作，而是团队成员之间相互交流，以共同了解项目情况和共同解决问题。

3. 如何应用成功实践说明

1) 每天早上 9 点，固定地点团队成员站立参加，不允许人员迟到，最长时间不超过 15 分钟，对于超过 10 人的团队，拆分成两组进行，Scrum Master 进行主持。

2) 成员在回答三个问题时目光要注视着大家，而不是 Scrum Master，否则就变成了向领导汇报工作。对每个人回答的问题有疑问，其他成员都可以提出，而不是只有 Scrum Master 一个人在问。

3) 成员提出的问题或困难其他成员要认真倾听，确定相关的负责人和协作方式，但问题细节不在会议上讨论。

4) 站会结束后，ScrumMaster 一定要知道哪些问题需要帮助团队成员解决。

5) 成功实践每日站会的表格：

主题	正确的模式	正面例子	反面例子	改进建议
自组织	团队成员都参与到其中，对话和讨论发生在团队成员之间。	团队成员争先恐后的从任务版上领取任务,识别不同分类的需求(QA 和 开发人员一起工作)	团队成员不是相互讨论，而是向领导汇报工作	"看着你的鞋子"技巧: 站会主持这每当一个团队成员注视着他汇报的时候，就低头看着自己的鞋子
关注点	团队成员能够锁定什么是最重要的，什么是阻碍	团队成员监督每个人的关注点，并且提醒跑题的人: “让我们会后再讨论."	站会之后才讨论问题的解决方法和其他议题，即使它们很重要	站会主持人在大家讨论超出范围时，要给出提醒
合作	每个团队成员能	“我如何支持你？你	会议主持人必须	第三个问题变成：

	够向其他人提供帮助，并且不羞于向他人寻求帮助	如何支持我？让我们站会之后一起讨论一下.”	提醒团队成员之间的工作任务的相关性，并且指出可以相互支持的内容	“什么在阻碍我们完成工作”，让团队成员认识到在任何任务中他们都不是孤立的
节奏	一人接一人的连贯并迅速的讲话，并不需要表面上的指令让下一个人继续，所有人都在“管理”，因此没有人“被管理”	完美的“爆米花”式讨论，没有团队成员过多的和某一个人讨论	站会主持人必须经常提醒下一个人开始分享，团队成员们也都在等他“下一个进行”的信号	在任务板前开站会（对成员有直观的提示），每天同一时间、同一地点.
勇气	揭露并且面对障碍，当变化来临时，承认并且拥抱它	如果燃尽图没有按照计划更新，团队成员要有勇气讲出来	每个人都避而不谈一些显而易见的问题，避开有难点的讨论	使用注释黑板，放置便利贴去记录关注点和进度
尊重	团队成员，ScrumMaster，product owner 之间相互关心尊重；欣赏站立的价值	参与者准时参加站会并且监督是否超时	团队只向站会主持人或领导问候，没有想到相互间的尊重	禁止或消除站会中存在“领导”，考虑随手记录、时间记录，停止任何礼仪

图 18-1 每日站立会议表格

专家点评：

站立会几乎被所有部署敏捷技术的组织所使用，但如本案例分析得如此具体、透彻的并不易见。作为读者本人受益良多。

每日立会是一种形式，它辅助“每日迭代”的管理。从某种意义上讲，“每日迭代”和“Sprint 迭代”，除了颗粒度外本质上没什么不同。本文阐述的实践如此到位，个人武断地猜一下，这个团队的 Sprint 管理应该也是不错的。

再有，"看着你的鞋子"技巧、“爆米花”式讨论等词汇让我印象深刻。想必此团队对成员的切身感受是很重视的。

奉送一点建议：“产品项目经理为 Scrum Master”，以我个人认知不推崇这样做。这会使得这个人物在团队中过于“特殊”，而且实际上这两个角色各自的职责都已相当繁重。

点评专家：刑雷

案例十九：每日站立会议（IBM）

1. 案例说明

1) 案例来源

IBM

2) 项目基本信息

在分布式敏捷项目中，团队因区域、文化、时间的差异，每日 Scrum 会议仍能够顺利开展。

项目基本信息如下：

- 团队规模：20 人
- 环境：JAVA

3) 组织结构

标准 Scrum 组织架构，但成员分布在不同国家。

4) 面临的问题

与会人员不同角色、不同性格、不同文化、不同经验、不同关系，一开始每日会议非常焦灼。即使在同一个会议室内每日成功开展会议也未必是件易事。

5) 解决方法

（Scrum Master）学习组织会议的方法和技巧。

6) 主要收益

产品研发流程运转效率大大提升，研发过程中的问题和困难都能及时得到解决。

2. 定义和特性说明

每日站立会议旨在让团队统一目标，协调团队内部问题的解决，绝非进度报告；他的特点除了包括站立开会和经典的三个问题：

1) 我昨天完成了什么任务？

2) 我今天打算做什么任务？

3) 我遇到了哪些障碍或困难？

还要求这类会议有效、快速，团队需要全部的参与以及 Scrum Master 有有效组织会议的技能。

3. 如何应用成功实践说明

组织会议其实正是 Scrum Master 一项非常重要的技能；有效的开展会议，能够体现出组织者，也就是 Scrum Master 在控制会场，协调问题的解决，积极推动项目进展和管理团队方面的综合实力了。并且，当敏捷项目分布在不同国家地区、时区时，这类会议的开展更有难度。使用以下推荐的方法和技巧能够帮助成功开展每日站立会议并且将使得 scrum master 变得更加专业：

1) 给 Scrum Master 的几点建议：

- 自信
- 声音洪亮
- 在会议室中适当走动
- 观察团队每个人
- 了解每个人的特点
- 不做无准备之发言
- 保持礼貌
- 前后一致
- 别太幽默
- 避免情绪化
- 感激
- 平衡发言时间
- 请外向型的先发言，请内向型做总结
- 保持正常语速
- 提醒发言者如果发言时间很长，或是经常打断其他人
- 可定期改变发言顺序

- 并非 3 个问题都需要问

2) 如 Scrum Meeting 是在电话会议中则还需注意：

- 发言者需独占话筒
- 未发言者需保持缄默

专家点评：

Scrum 强调自我管理，每个成员都是团队的管理者。但实践中看，在每日站会中，一个好的 Scrum Master 是很重要的，他能保证每日站会正确、高效的执行，尤其是在实施每日站会的初期。文中对 Scrum Master 的建议特别值得学习。对于团队成员不在同一地点还需要电话会议召开每日站会的情况，个人建议反倒不要使用每日站会制度，因为这样就失去了其最核心的高效、敏捷的沟通的核心理念。很有可能就流于形式了。

点评专家：高航

案例二十：每日站立会议（用友软件）

1. 案例说明

1) 案例来源

用友软件股份有限公司

2) 项目基本信息

在开发 XX 银行资金交易系统的过程中，每天坚持使用敏捷开发中的站立会议来沟通开发进度和开发中的困难，使开发过程中的开发团队对项目进度能够及时同步，遇到的问题也能够得到最快的解决。

项目基本信息如下：

- 团队规模：6 人
- 环境：NC5.6 平台 & ORACLE10G

3) 组织结构

标准 Scrum 组织架构。

4) 面临的问题

站立会议的方式是敏捷开发中比较容易执行的，但往往由于内容很容易演变为例行公事的进度汇报，也是最难于坚持的。

5) 解决方法

- 作为 leader，不能仅仅通过例会跟踪、管理问题。例会上要善于发现问题，并号召大家介绍进展、共享关键信息、主动暴露问题；
- 时间不宜过长，15 分钟以上难以让团队成员介绍。冗余的会议表明 leader 对于团队的状况在平时缺乏手段掌控，是危险的信号，同时也浪费团队的时间和精力；
- 例会上不要解决复杂问题，简单的组内协调问题，立即解决；复杂的问题，如果涉及组外协调，给出简单的原则和建议，会后解决，明天通报进展；

- 会上平等，大家是否积极参加会议代表了例会是否成功。

6) 主要收益

- 通过站立会议，项目进度得到了最快的同步；
- 成员的团队观念大大加强；
- 团队成员通常会严格按照任务制定的计划来完成任务。

2. 定义和特性说明

每日站立会议是敏捷开发的一条关键实践。可以加强团队之间的沟通，加强团队意识。

3. 如何应用成功实践说明

每日站立会议中团队成员轮流主动发言，主要就“昨天干了什么”，“今天计划干什么”，“遇到了什么障碍”三个问题进行讨论。团队外成员也可以参与，但没有发言权。通过每天面对面的沟通可以：

1) 快速同步进度，让组内成员相互了解彼此进展，从而了解本项目的整体进展。
2) 给团队成员一种精神压力，要对每日的工作目标信守承诺。
3) 培养团队文化，让每个人意识到我们是一个团队在战斗。

专家点评：

对于站立会议，关键是保证时间的控制和会议的有效性，能及时发现项目进展的问题和潜在风险，并从团队的角度合作解决和处理。本案例的实施符合了敏捷的基本原则，在确保如何不将站立会议演变为例行公事的进度汇报方面还应有更多的实践经验可以尝试和总结。

点评专家：陈志波

专家点评：

虽然这个每日例会也实现了每日的监控循环，但我认为有一点关键的“Bad Smell”——Leader 的存在。

这个特殊人物，使得汇报带有了上下层级间报告的意味，——无论 Leader 是否是同甘共苦的 Partner，——而排斥了“大家报告给大家”的氛围。我想这不是真正的敏捷团队希望看到的。

点评专家：邢雷

案例二十一：每日站立会议（石化盈科）

1. 案例说明

1) 案例来源

石化盈科

2) 项目基本信息

在企业能耗评价系统二期开发项目中使用 Scrum 模型，每次 Sprint 周期相对固定，约为 2 周左右，要求团队快速响应。

项目基本信息如下：

- 团队规模：7 人
- 环境：VSTS2010 & sql server

3) 组织结构

Scrum 组织结构。1 名 Product Ower,1 名 Scrum Master 及 5 名团队开发/测试人员。

4) 面临的问题

团队成员彼此间不熟悉，为第一次合作。工位相对分散，公司会议室资源紧张。

5) 解决方法

每日上午召开站立会议，一般选择刚上班的时间，地点随机，相对灵活。各团队成员分别汇报自己的工作情况，轮流发言，内容围绕着 3 个方面：

- 我昨天完成了什么？
- 我今天准备做什么？
- 我当前工作遇到了哪些问题？

Scrum Master 负责主持会议，记录并跟踪大家提出的问题。面临来自团队外干扰时由 Scrum Master 出面协调。

6) 主要收益

- 通过每日站立会议增进了团队成员之间的了解；
- 有效的提高了团队的信息沟通；
- 有利于及时发现项目问题；
- 提高了团队开发效率。

2. 定义和特性说明

团队每天 15 分钟的检查和调整会议称之为每日站会。会议上每个成员需要回答 3 个问题：昨天都完成了哪些工作？今天准备完成什么？工作中遇到了什么问题？由 Scrum Master 确保会议的举行，并控制会议时间，让团队成员进行简短有效的汇报。此外还要处理其他干系人（主要针对用户）的影响。

3. 如何应用成功实践说明

1) 由 Scrum Master 制定每日站立会议的时间，并把每个人的控制汇报时间在 5 分钟之内。尽量固定场所。

2) 在会议中确认进展，记录问题。对于一些可以快速解决的问题，可以在会议中适当处理。保证团队成员的工作效率。

3) 对于会议中，产品经理产生的干扰，Scrum Master 一定要坚决排除。

4) 一般选择在早上召开会议，通常一天的工作从早晨开始计划。

专家点评：

每日站会的时间和地点应该固定，到了时间团队成员自动汇集到固定地点上，不需要 Scrum Master 召集，这样可以节省召集和选择地点的时间。

点评专家：高航

案例二十二：持续集成（东软集团）

1. 案例说明

1) 案例来源

东软集团

2) 项目基本信息

具备多种项目情态，人数过千的某东软事业部

3) 组织结构

- 矩阵型组织：事业部下，多个开发部，每事业部下存在多个项目；
- 多项目：标准 Scrum 团队+以传统 PM 为核心的团队。

4) 面临的问题

- 项目在开发过程中，每次版本 Release 前的编译和检查工作需要花费较长的时间，并时常伴随的问题的发生；
- 存在问题的代码经常被提交到代码库中；
- 项目代码的稳定性无法度量和保证。

5) 解决方法

引入持续集成工具，对代码进行自动构建，提升构建效率，并结合代码检查工具，如 PClint、Checkstyle、Findbugs 等，自动对代码进行检查。

对代码管理工具进行二次开发，在开发人员向代码库中提交代码时，触发代码检查工具对提交的代码进行检查，如果提交的代码中存在问题，则不允许向代码库中提交。

在事业部层面搭建持续集成服务器，配置专门技术人员，供所有有需求有条件的项目选用。

建立代码质量度量网站，用于收集项目的持续集成结果和代码中的问题，并对代码的稳定性和质量进行度量和分析，作为后续代码质量改善的输入。

6) 主要收益

- 成本节约

持续集成工具每天/每周会自动对版本库中的代码进行构建（包括从代码版本库中取得代码、开发包和动态库的最新版本，代码的编译，代码的静态工具检查等），整个构建过程不需要人工的参与，单次持续集成的效益已比较明显，并且聚少成多，多个项目为事业部带来的效益非常可观。

- 代码质量提升，并利于持续改善

代码库中的代码质量有了显著提升。项目中代码的稳定性有了显著的提高，版本发布前编译过程基本不会出现问题。项目中代码的稳定性得到了量化，通过持续集成工具，可以看到项目中的持续集成过程成功了几次，失败了几次，成功的次数越高，项目稳定性越好。

- 为整个事业部建立集成工作的高效管理

通过建立代码质量度量网站，并通过图表的方式清晰的展现项目持续集成的成功率，随时了解项目中代码的稳定性。给事业部层面的管理，提供了简洁有效的管理视图。

2. 定义和特性说明

在大型事业部组织层面统一部署持续集成，收益巨大

持续集成是一种软件开发实践，即团队开发成员经常集成它们的工作，通常每个成员每天至少集成一次（意味着每天可能会发生多次集成）。每次集成都通过自动化的构建（包括编译，发布，自动化测试)来验证，从而尽快地发现集成错误。许多团队发现这个过程可以大大减少集成的问题，让团队能够更快的开发内聚的软件。

一些原则：

1) 所有的开发人员需要在本地机器上做本地构建，然后再提交的版本控制库中，从而确保他们的变更不会导致持续集成失败；
2) 开发人员每天至少向版本控制库中提交一次代码；
3) 开发人员每天至少需要从版本控制库中更新一次代码到本地机器；
4) 需要有专门的集成服务器来执行集成构建,每天要执行多次构建；

5) 每次构建都要 100%通过；

6) 每次构建都可以生成可发布的产品；

7) 修复失败的构建是优先级最高的事情。

3. 如何应用成功实践说明

1) 建立持续集成环境

- 硬件和软件的准备和安装，硬件包括用于建立持续集成系统的设备；
- 软件包括持续集成工具（Jenkins（原 Hudson））、代码管理工具（CVS、SVN、GIT 等）、代码检查工具（PCLint、Checkstyle、Findbugs 等）、自动构建工具（ant、nant、shell、bat 等），测试工具（CppCheck、Junit 等）、结果展示工具（各种 Report plugin）等。

2) 建立持续集成过程

根据不同的项目类型，建立不同的持续集成过程，包括：

- 代码的更新方式

根据项目代码库中代码管理方式的不同，代码更新分为两种方式：

a) 通过 Jenkins 中的代码管理插件取得最新版本的代码；

b) 通过在 Jenkins 中调用代码更新脚本更新代码，这种方式适用于代码库中引用关系较多的情况。

- 代码的构建方式

a) java 工程使用 ant 脚本实施代码的自动构建；

b) C/C++工程使用 Nant 脚本实施代码的自动构建。

- 代码的检查/测试方式

a) java 工程使用 Checkstyle、Findbugs 和 Junit 实施代码的静态检查和测试；

b) C/C++工程使用 PClint 和 CppCheck 实施代码的静态检查和测试。

- 集成产品的发布方式

通过构建脚本自动将 java 工程生成的 jar/war 包、C/C++工程中生成的 待烧写文件(.out）、动态库(.dll)或可执行文件(.exe)等部署到指定的位置，用于后续的测试。

- 集成结果的呈现方式

通过 Jenkins 中的 report 插件呈现持续集成的结果，呈现的内容包括代码静态检查的结果和问题数量的趋势图。

3) 持续集成结果的管理

■ 持续集成结果通知

a) 通过 Jenkins 中的 Mail 通知功能，将持续集成结果发送给项目成员；

b) 通过登录度量网站，查询持续集成结果。

■ 持续集成结果数据收集

通过在 Jenkins 的自动构建脚本中调用数据库写入工具，将持续集成的结果写入数据库。

■ 代码质量问题数据收集

在开发人员提交代码时，通过代码检查工具自动对提交的代码进行检查，检查的结果会通过数据库写入工具写入数据库。

4) 代码质量度量网站

■ 代码稳定性度量与分析

项目的每一次持续集成的结果都会展示在度量网站中，并通过柱状图展示持续集成的成功率（如附图）。

■ 代码质量问题度量与分析

代码的质量数据会分别通过 开发部视图、项目视图和担当视图 展示在度量网站中，并通过饼图，按照代码的问题类型，对各种问题的分布比例进行展示（如附图）。


A[redacted]事业部 代码质量度量网站

持续集成结果柱状图 PClint检查代码问题比例饼图 Checkstyle检查代码问题比例饼图

开发部视图

业务部	项目名称	累计集成次数	累计集成成功次数	本周集成次数	本周集成成功次数	累计变更代码行数	累计编码问题数	本周变更代码行数	本周代码问题数	累计代码问题密度
国内业务1部	project_1	3	2	0	0	0	0	0	0	0.0
国内业务1部	project_2	7	3	0	0	3	0	0	0	0.0
国内业务2部	project_1	1	1	0	0	0	1	0	0	0.0
国内业务2部	project_2	4	1	0	0	1	4	0	0	4.0
国内业务2部	project_3	3	2	0	0	0	7	0	0	0.0
国内业务2部	project_4	4	3	0	0	0	6	0	0	0.0
国内业务2部	project_5	2	1	0	0	0	3	0	0	0.0
国内业务3部	project_34	3	2	0	0	2	6	0	0	3.0
国内业务3部	project_35	2	2	0	0	16	5	0	0	0.0

每页10行共42行第1页共5页
首页 上一页 下一页 尾页 转到第1页

图 22-1 代码质量数据——事业部视图

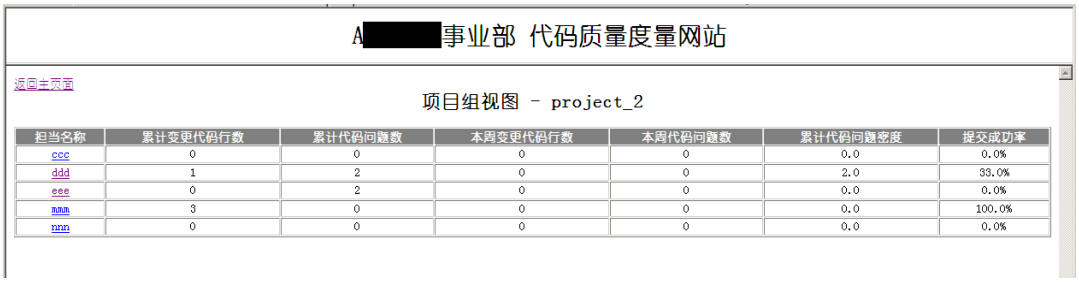


图 22-2 代码质量数据——项目组视图

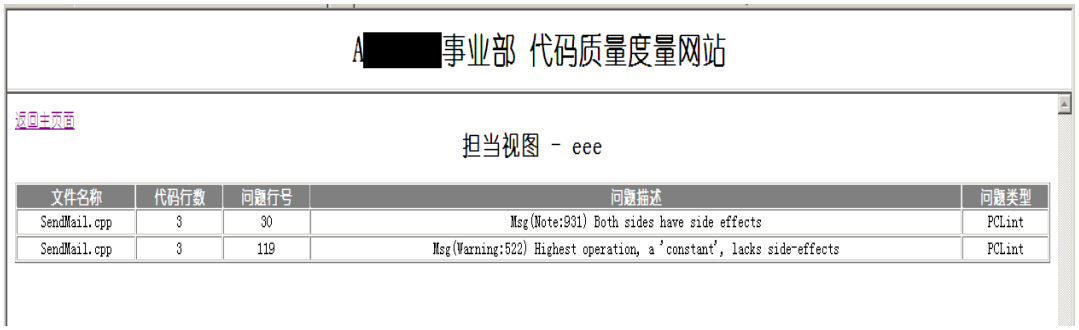


图 22-3 代码质量数据——担当视图

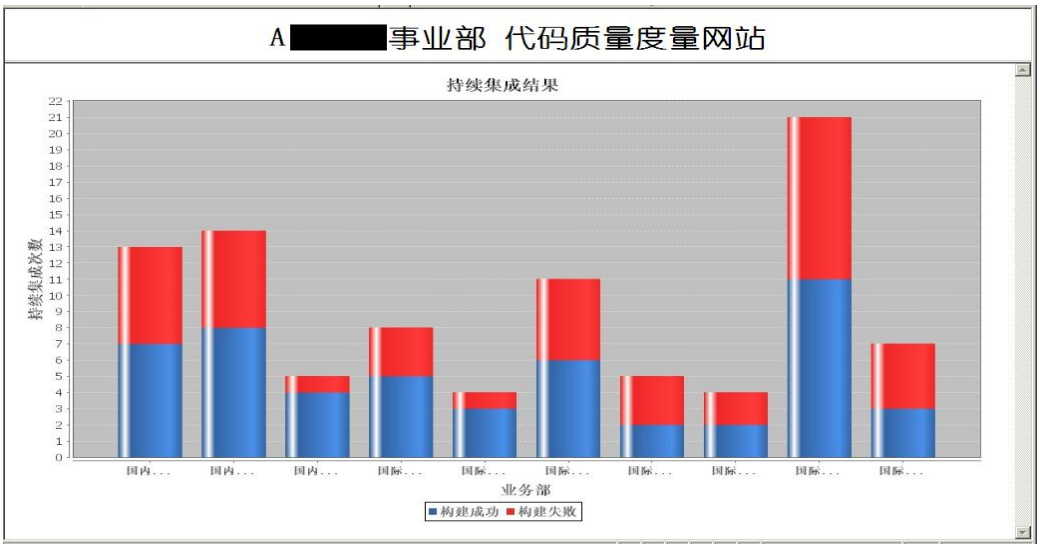


图 22-4 持续集成成功失败的事业部统一视图

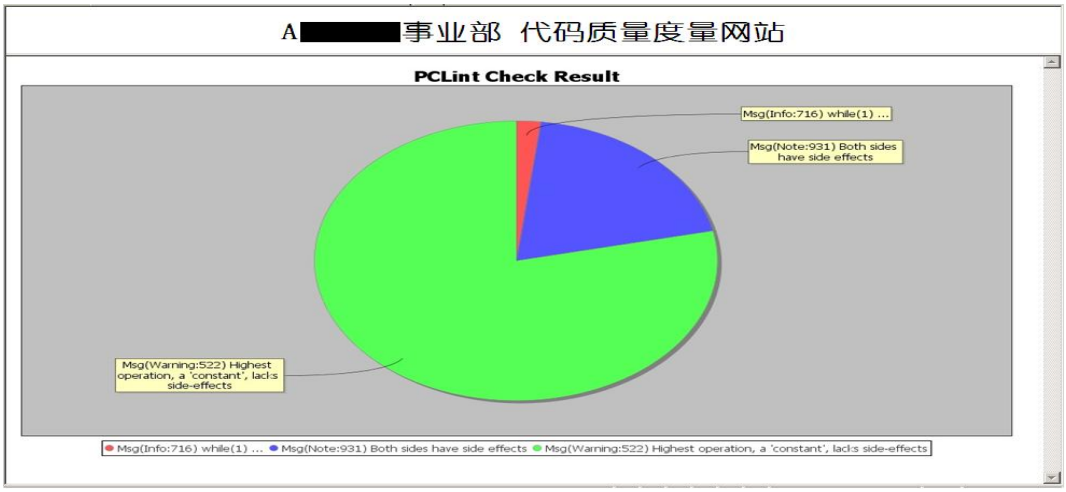


图 22-5 静态检查工具 PCLint 结果的事业部统一视图

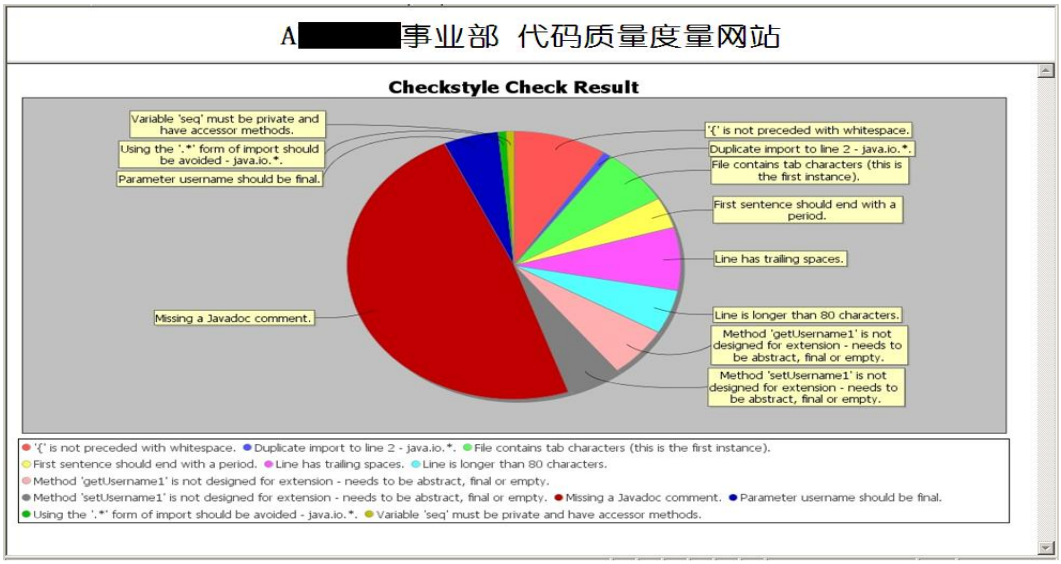


图 22-6 静态检查工具 CheckStyle 结果的事业部统一视图

专家点评：

由于持续集成要求频繁的集成，因此自动化工具必不可少，可以减少很多人工的操作。同时持续集成业要求所有项目成员遵循统一的持续集成的流程。本人认为持续集成不失为一个快捷、有效的管理集成、确保软件质量的方法，由于需要额外的工具、技术、流程的支持，也许更适合比较大规模的系统。

点评专家 :刘德意

案例二十三：持续集成（IBM）

1. 案例说明

1) 案例来源

IBM

2) 项目基本信息

在一个在线报价和订单项目中采用敏捷软件开发，实现快速高质量交付。

项目基本信息如下：

- 团队规模：10 人
- 环境：Java, Web

3) 组织结构

标准 Scrum 组织架构

4) 面临的问题

在敏捷实践中，持续集成的时间成本过大。

5) 解决方法

实现并采用并行持续集成的方法和工具，主要活动是持续集成。

6) 主要收益

- 持续集成时间缩短至原来的 1/2；
- 实现快速交付；
- 提高了软件质量。

2. 定义和特性说明

持续集成由一组软件开发实践、行为和原则构成，目的是自动化软件或部件之间的集成，并持续改进，从而使工程师能够尽早发现和解决问题，交付高质量软件。持续集成是在短时间迭代中交付稳定、可用软件的关键。

3. 如何应用成功实践说明

通常，持续集成由源代码提取（在大型项目中，代码可能由不同源代码库管理）、编译、单元测试、自动部署、构建验证测试等步骤完成。由于步骤间或步骤内可能存在依赖关系，持续集成只能串行完成，从而增加了时间成本。

这里介绍的是一种并行持续集成的成功实践和思路。Apache Ant 提供了并行执行嵌套任务的功能，但是它不拥有任务间依赖检查的能力。而在实际开发中，Ant 任务之间经常会有依赖。而通过定义和配置任务间依赖拓扑，并行执行嵌套任务，则可以扩展 Ant 任务的并行能力，从而实现并行持续集成，提高效率。

1) 并行源代码提取

静态源代码模块之间没有依赖，源代码提取可以很方便地做到并行进行，以提高生产率。

2) 并行自动化构建

模块之间在编译时可能会存在依赖关系。构建工程师通过定义工程间依赖拓扑，以实现并行编译。

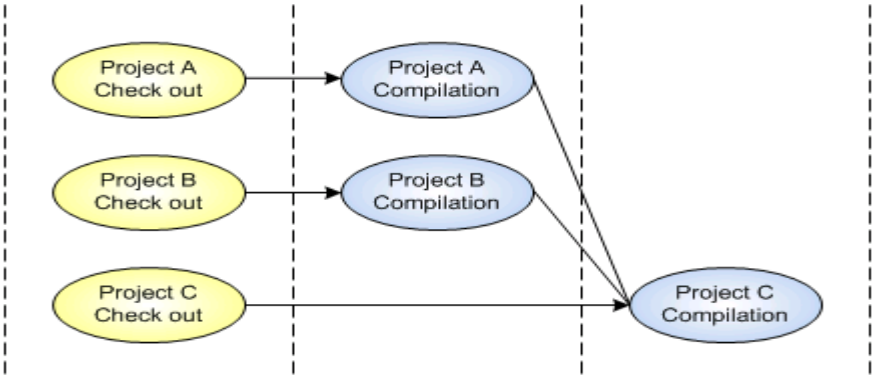


图 23-1 并行编译

3) 并行单元测试

定义一个单元测试框架，为每个模块工程建立单元测试工程，从而实现并行单元测试。

4) 多个应用开发的并行持续集成

对于一个项目组，以上方法可以从单个应用开发的并行持续集成扩展为多个

应用开发的并行持续集成。在持续集成所占用的资源有限的情况下，我们可以对多个应用的不同步骤进行并行，从而有效利用现有资源。例如，当应用 A 在进行步骤 1（主要占用网络资源）时，其他应用等待。当应用 A 在进行步骤 2 和 3（主要占用 CPU 和内存资源）时，从等待列表中唤醒应用 B，应用 B 开始进行步骤 1（主要占用网络资源），等等。构建工程师可以管理不同小组，通过定义线程数量，来进行多应并行持续集成。

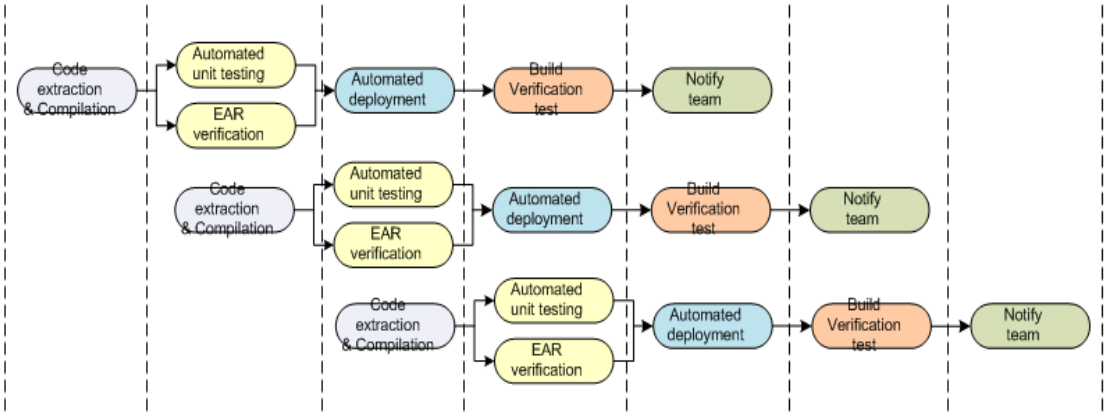


图 23-2 持续集成

专家点评：

在传统开发方法的项目中，集成往往最耗时，但同时由于在项目的后期才开始集成，使得集成的有效时间很短，造成集成不够完整，测试不够全面，遗留缺陷过多的问题。本实践所提出的并行持续集成，对许多困扰于集成所消耗的时间成本的项目团队来说，应该是个启发。

点评专家：刘德意

案例二十四：持续集成 （上海宝信）

1. 案例说明

1) 案例来源

上海宝信软件股份有限公司

2) 项目基本信息

平台中间件，封装了界面、工作流、数据库、常见控件等等，在宝信各大信息化项目中应用了上百套。

项目基本信息：

- 团队规模：14 人
- 主要交付物：Java, C#

3) 组织结构

普通项目组织+独立测试组

4) 面临的问题

代码修改后的质量不能保障。

5) 解决方法

- 采用持续集成+每日集成；
- 主要活动：单元测试，建设持续集成测试集。

6) 主要收益

保障了代码修改的质量

2. 定义和特性说明

持续集成是指当代码提交后，马上启动自动编译、自动部署和自动化测试来快速验证软件，从而尽快地发现集成错误。持续集成的英文是 Continuous integration，在敏捷语境下缩写为 CI。

在持续集成中，团队成员频繁集成他们的工作成果，一般每人每天至少集成一次，也可以多次。每次集成会经过自动构建（包括自动测试）的验证，以尽快发现集成错误。许多团队发现这种方法可以显著减少集成引起的问题，并可以加快团队合作软件开发的速度。

持续集成的目的是自动化软件或部件之间的集成，并持续改进，从而使工程师能够尽早发现和解决问题，交付高质量软件。

持续集成是在短时间迭代中交付稳定、可用软件的关键。

持续集成原则：

1) 所有的开发人员需要在本地机器上做本地编译，然后再提交的版本控制库中。

2) 需要有专门的集成服务器来执行集成,每天可以执行多次集成。

3) 每次集成争取都要 100%通过，如果集成失败，马上修复，修复失败的集成是优先级最高的事情。

4) 每次成功的集成都可以生成可运行的软件。

5) 修复失败的构建是优先级最高的事情。

持续集成能够帮助开发团队应对如下挑战：

1) 软件构建自动化。使用 CI，只要按一下按钮，它会依照预先制定的时间表，或者响应某一特定事件，就开始进行一次构建过程。如果想取出源码并生成构件，该过程也不会局限于某一特定 IDE、电脑或者个人。

2) 持续自动的构建检查。CI 系统能够设定成持续地对新增或修改后签入的源代码执行构建，也就是说，当软件开发团队需要周期性的检查新增或修改后的代码时，CI 系统会不断要求确认这些新代码是否破坏了原有软件的成功构建。这减少了开发者们在手动检查彼此相互依存的代码中变化情况需要花费的时间和精力。

3) 持续自动的构建测试。这个是构建检查的扩展部分，这个过程将确保当新增或修改代码时不会导致预先制定的一套测试方案在构建构件后失败。构建测试和构建检查一样，失败都会触发通知(Email,RSS 等等)给相关的当事人，告知对方一次构建或者一些测试失败了。

4) 构件生成后续过程的自动化。一旦自动化检查和测试的构建已经完成，

一个软件构件的构建周期中可能也需要一些额外的任务，诸如生成文档、打包软件、部署构件到一个运行环境或者软件仓库。通过这样，构件能更迅速地提供给用户使用。

实现一个 CI 服务器需要的最低要求是，一个易获取的源代码仓库(包含源代码)，一套构建脚本和流程和一系列围绕软件构建的可执行测试。

3. 如何应用成功实践说明

利用 Hudson 作为工具，监控 VSTS 的源代码管理。

持续集成(CI,Continuous Integration)是一种实践，目的是缓和和稳固软件的构建过程。CI 能够帮助开发团队应对如下挑战：

1) 软件构建自动化

使用 CI，只要按一下按钮，它会依照预先制定的时间表，或者响应某一特定事件，就开始进行一次构建过程。如果想取出源码并生成构件，该过程也不会局限于某一特定 IDE、电脑或者个人。

2) 持续自动的构建检查

CI 系统能够设定成持续地对新增或修改后签入的源代码执行构建，也就是说，当软件开发团队需要周期性的检查新增或修改后的代码时，CI 系统会不断要求确认这些新代码是否破坏了原有软件的成功构建。这减少了开发者们在手动检查彼此相互依存的代码中变化情况需要花费的时间和精力。

3) 持续自动的构建测试

这个是构建检查的扩展部分，这个过程将确保当新增或修改代码时不会导致预先制定的一套测试方案在构建构件后失败。构建测试和构建检查一样，失败都会触发通知(Email,RSS 等等)给相关的当事人，告知对方一次构建或者一些测试失败了。

4) 构件生成后续过程的自动化

一旦自动化检查和测试的构建已经完成，一个软件构件的构建周期中可能也需要一些额外的任务，诸如生成文档、打包软件、部署构件到一个运行环境或者软件仓库。通过这样，构件能更迅速地提供给用户使用。

实现一个 CI 服务器需要的最低要求是，一个易获取的源代码仓库(包含源代

码)，一套构建脚本和流程和一系列围绕构件构建的可执行测试。

Hudson 是一个界面友好，功能强大，拥有高可扩展性的持续集成监控系统，一般来说 Hudson 主要可以做以下两件事情：

■ 持续构建软件项目

像 CruiseControl 一样，hudson 提供自身的持续集成系统，目前可以自身集成了 subversion?， cvs?，maven?等一些插件，利用它可以构建或者定时构建 软件项目，如果有测试失败，可以将测试报告以丰富的形式展现出来。

■ 监控外部任务的执行情况

可以监视没有在 hudson 进行的配置的定时任务，甚至是远程机器上的任务。Hudson 很好的满足了 iPlat 项目组对持续集成工具的要求。

专家点评：

宝信的持续集成体现目前 CI 实践的典型样式：Hudsonà提交即触发集成动作一定要配套自动测试à最优先处理集成问题。——使得开发总是在“干净”的平台上稳步推进。

我对宝信的 CI 实践印象深刻原因有二：它是我个人最早见到的完整 CI 案例，在 2007 年时既已部署；再者，Night-Build 后团队早来后第一件事就是看前一天晚上的结果。“绿灯集体欢呼，红灯由出错者买雪糕”的规则，让我对敏捷团队的欢乐气氛有了最初的体验。

点评专家：邢雷

本实践的实施，需要两大前提：所有持续集成的要素（统一的代码库、自动编译、自动部署、自动测试），持续集成的工具。个人理解，持续集成除了要求有工具和服务器支持外，还要求开发人员和测试人员有较高的专业能力，并且一致遵从严格的持续集成制度。

点评专家：刘德意

案例二十五：需求订单（汤森路透）

1. 案例说明

1) 案例来源

汤森路透

2) 项目基本信息

在某自动化处理配置应用中，通过制定、维护需求订单，分解了需求的同时，使得开发人员对项目有了总体的了解， 避免了需求理解偏常以及需求变更所造成的浪费。

3) 组织结构

标准 Scrum 组织架构。

4) 面临的问题

项目大，需求粒度大，变更多，需求需要进一步挖掘。

5) 解决方法

维护需求订单：需求早期为粗粒度需求，迭代开始前细化高优先级需求，基于细粒度需求详细讨论。

6) 主要收益

由于有维护良好的需求订单和充分的沟通，项目开发工程中基本没有返工，以及不必要的资源浪费，及时满足了客户的需求。

2. 定义和特性说明

需求订单是一张记录用户需求的列表。 列表中的每一项都包含了需求编号，需求标题，详细描述，重要性，以及故事点（理想人日）数。

3. 如何应用成功实践说明

建立一张准确的需求订单有助于团队成员了解项目概况，理解具体需求。同时各相关人可根据需求订单了解项目进展，以及产品的概貌。

在项目开始初期，通常只记录最粗粒度的需求并给出大致的优先级。优先级是一个数字，数字越大代表优先级越高，初始给出的优先级间应留有一定的空间，便于日后添加介于两者优先级之间的需求。在项目初期不细化需求的原因在于避免将时间用于暂时不开始的需求，以及避免日后需求变更所导致需求重新分析所带来的时间浪费。

在迭代开始前，产品负责人按优先级选定需求同团队成员讨论，如果需求粒度太粗，则先细化需求。确定一个需求粒度是否太粗，可有以下标准：

- 开发人员无法确定故事点数，
- 故事点数大于某个特定数字（我们的团队认为 6 是最大可接受的需求粒度）。

需求细化之后，开发人员须与产品负责人确定每一个细节。通常一个迭代前，需要 4-8 小时的讨论，方可达成共识。需求讨论得越细致，产品最终和用户的需求越接近，也就避免了由于需求理解偏差而产生的返工，变更所消耗的时间。最终，需求细节要记录在需求订单中。

在迭代过程中及迭代后，用户及产品负责人如有新需求，可将需求加入需求订单。

2401	GR2401	(Generic IFFM) Lock Mechanism	Implementing a lock mechanism in order to handle parallel processing case.	Waiting for approval		JAVA_GENERIC_IFFM		10		
2402	GR2402	(Generic IFFM) Upload file to Unix server from Windows Server	As the application server will be deployed to WTS mechine, there must be a way to upload file to Unix server.	Completed	Herry_luo	JAVA_GENERIC_IFFM	2	15	9	4/29/2011
2403	GR2403	(Generic IFFM) Support new keyword in derive rule	The key word 'curr.' and 'last.' need to be supported in order to easily refer to current instrument data and last message data seperately.	Waiting for approval		JAVA_GENERIC_IFFM		12		
2404	GR2404	(Generic IFFM) Remove messages in message editor	Some messages may not need to be processed. The remove message function should be provided in order to delete unwanted message.	Completed	Qiang_wan	JAVA_GENERIC_IFFM	2	15	9	4/29/2011
2405	GR2405	(Generic IFFM) Configuration for disable Autoprocessing for specific action	We should make the iffm configurable for auto processing specied actions, e.g. only auto process Add/Change action.	Completed	Qiang_wan (REMA)/Herry_luo (RAM)/Jacky_chen (RA)	JAVA_GENERIC_IFFM	6	15	9	4/29/2011
2406	GR2406	(Generic IFFM) Bulk processing specific action in Message Editor	This function is supported in legacy emea IFFM GUI. In some exchange, user may want to bulkly process the unprocessed actions.	Waiting for approval		JAVA_GENERIC_IFFM		6		
2407	GR2407	(Generic IFFM) Enlarge holding table to handle more pipes	Currently the holding table can handle max 50 pipes in a feed file, while for project "Linking Holiday to GEDA" the feed file may contain max 66 pipes. So holding table should be enlarged by adding at least 16 PIPE columns.	Completed	Qiang_wan	JAVA_GENERIC_IFFM	1	15	9	4/29/2011
2408	GR2408	(Generic IFFM) Housekeeping for non-category record in Hold table	The record which is not belong to any category should be deleted after few days.	Completed	Herry_luo	JAVA_GENERIC_IFFM	1	15	9	4/29/2011
2409	GR2409	(Generic IFFM) Performance: Create View for Hist table and feed message	Using bind variable in the partion table may cause performance issue in case the data in diff partition is not balanced. In order to solve this issue, all of backgroud processing should base on view for diff partition.	Completed	Qiang_wan	JAVA_GENERIC_IFFM	3	15	9	4/29/2011
2410	GR2410	(Generic IFFM) Support Drop processing	Support Drop processing, which is requested by Bucharest.	Waiting for approval		JAVA_GENERIC_IFFM		10		
2411	GR2411	(Generic IFFM) Support Key change Logic	Request by Bucharest	Completed	Qiang_wan (REMA)/Herry_luo (RAM)/Jacky_chen (RA)	JAVA_GENERIC_IFFM	5	15	9	4/29/2011
2412	GR2412	(Generic IFFM) Non-Functional: schema based release	A new approach for release, which will 1) Release new version to B schema if A is in using. 2) Change synonymous to point to B. 3) Clear A after few days. 4) Next release will switch from B to A.	Completed	Herry_luo	JAVA_GENERIC_IFFM	6	15	9	4/29/2011
2413	GR2413	(Generic IFFM) Invoke Mifid API		Completed	Herry_luo	JAVA_GENERIC_IFFM	1	15	9	4/29/2011
2433	GR2433	(Generic IFFM) Disable mail report for Future Option processing	As Future Option will have its own report, the Generic IFFM report will always be null.	Waiting for approval		JAVA_GENERIC_IFFM		10		
2444	GR2444	(Generic IFFM) Integrate with VDI system		Waiting for approval		JAVA_GENERIC_IFFM		5		
2445	GR2445	(Generic IFFM) Version control for configuration items	1) Be able to track the configuration change 2) Be able to fallback configurations to any version	Waiting for approval		JAVA_GENERIC_IFFM		8		
2446	GR2446	(Generic IFFM) Webservice for getting configurations by Feed		Waiting for approval		JAVA_GENERIC_IFFM		5		
2457	GR2457	(Generic IFFM) Support populate default action for Incremental file	If there is no action field for a Incremental file, IFFM should support populate 'A' for all of the messages.	Completed		JAVA_GENERIC_IFFM		10	9	
2474	GR2474	(Generic IFFM) Support TAB delimited file	1) Support TAB delimited file loading 2) Configurable for choosing delimiter	Completed	Herry_luo	JAVA_GENERIC_IFFM		15	10	5/6/2011
2475	GR2475	(Generic IFFM) Support generating .f file for Legacy Asia IFFMs	Legacy Asia IFFM called GEDA Interface for trigger IFFM processing. Need support this case.	Completed	Herry_luo	JAVA_GENERIC_IFFM		15	10	5/6/2011
2477	GR2477	(Generic IFFM) Support Customized processing time for local files	Currently, only Remote files can be setting a job for processing in certern time point. ICE need support it even for local files in order to wait for whole batch completed.	Waiting for approval	Qiang_wan/Herry_luo	JAVA_GENERIC_IFFM		15		

图 25-1 需求订单

专家点评：

本案例中团队有效地贯彻了对 Product Backlog 使用的诸多基本原则，并且根据自己实践中问题形成了非常具体的操作规范（如颗粒度选择）。体会很具体，操作可借鉴性很强。

另外一个角度：本文中没见到 Product Backlog、Product Owner、Sprint 等标准英文术语，取而代之的是产品负责人、需求清单、迭代等中文词汇。——这不也正是理论已有效落地的明证吗？

点评专家：刑雷

案例二十六：需求订单（石化盈科）

1. 案例说明

1) 案例来源

石化盈科

2) 项目基本信息

在企业能耗评价系统二期开发项目中使用 Scrum 模型，每次 Sprint 周期相对固定，约为 2 周左右，要求团队快速响应。

项目基本信息如下：

- 团队规模：7 人
- 环境：vs2010 & sqlserver2008.

3) 组织结构

Scrum 组织结构。1 名 Product Ower,1 名 Scrum Master 及 5 名团队开发/测试人员

4) 面临的问题

早期需求不稳定，用户意见零散，业务目标不明确。传统的需求文档表述繁琐，后期维护的工程较大。

5) 解决方法

用 Product Backlog 作为产品的需求规格；周期性收集用户的业务需求并表述为用户故事；与团队成员一同建立和细化用户故事；每一次迭代结束后，与用户确认，对需求订单进行维护。

6) 主要收益

用需求订单取代传统的需求规格说明书。表述清晰明确，团队成员易于理解，易于维护。

2. 定义和特性说明

也称为产品订单(Product Backlog)，是整个项目的概要文档。包括产品所有所需特征的粗略描述。

3. 如何应用成功实践说明

需求订单由 Product Owner 向用户收集，并分解为用户故事。需求订单是开放的，由 Product Owner 创建，团队每一个成员都可以编辑和维护。

在整个项目开发生命周期内，需求订单需要不断维护，管理与跟踪需求变更。

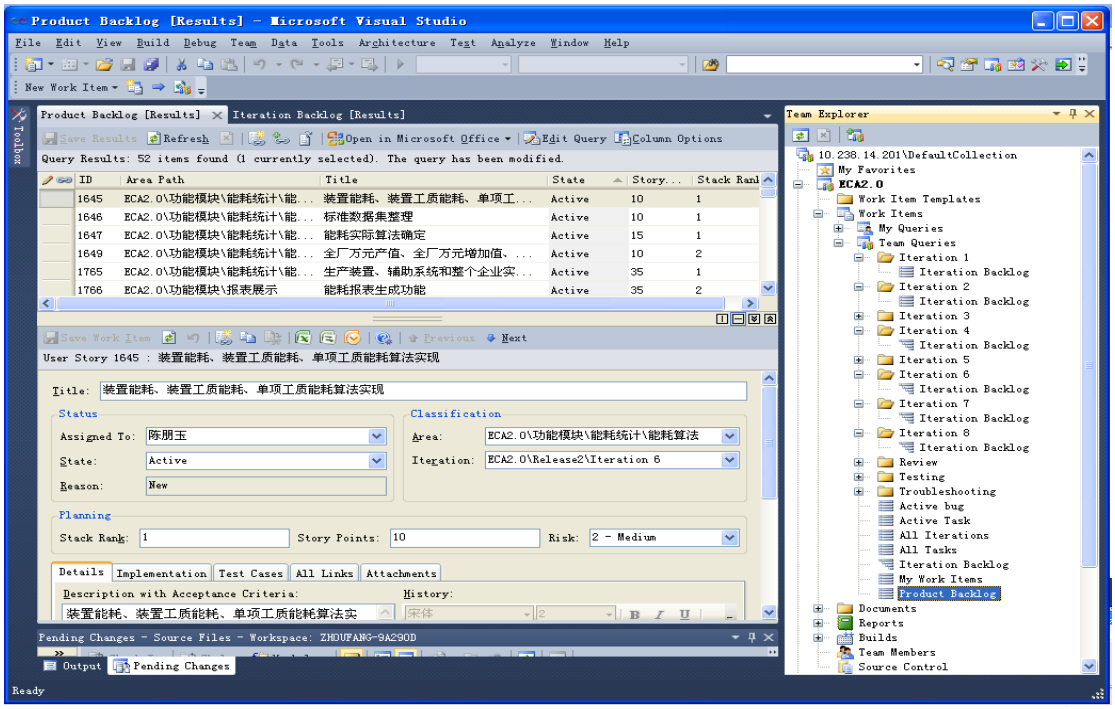


图 26-1 使用 VSTS2010 进行需求订单管理

专家点评：

敏捷项目的短迭代下，如何解决需求零散、维护难的问题？使用 Scrum 里的 Product Backlog 交付件代替传统的原始需求描述文档，尤其在小项目里，这不失为一个选择。Product Backlog 是 Product Owner 的财产，Product Owner 负责随时随地收集整理来自项目外部的需求，对每一项原始的需求进行初步的分析，包括其应用背景、所解决的用户问题等。经过初步分析的并且已经更为清晰一些的需求项目将可以进入迭代规划中，Product Owner 和团队成员一起做估计、分析优先级。

点评专家：邰晓梅

案例二十七：Retrospective Meeting（特艺）

1. 案例说明

1) 案例来源

特艺（中国）科技有限公司

2) 项目基本信息

Research 项目，研究具体应用，现有技术的调研，新技术、算法的分析和。创新，实现现有产品的升级和新应用的推广。研究的具体方向和内容（项目需求）随项目的进展，由模糊逐渐清晰。

项目基本信息如下：

- 团队规模：4-5 人
- 主要交付物：算法、实验用代码、demo、技术调研报告等

3) 组织结构

标准 Scrum 组织架构

4) 面临的问题

对 research 工作缺乏计划，没有前期的借鉴；Retrospective Meeting 中总是少数较活跃的成员发言，经过几次 Retrospective Meeting 之后，团队成员反应没有新的内容可以总结。

5) 解决方法

用 PostIt，每个人分别写下自己的观点，内容不局限在“流程”、“Scrum”方法本身，可以讨论任何对个人、团队、项目有益的内容。并且采用更为多样化的形式来促进团队成员进行思考。

6) 主要收益

Retrospective Meeting 更加活跃，集思广益，总结出团队成员真正觉得满意的/需要改进的/想要尝试的事项。

2. 定义和特性说明

通过 Retrospective Meeting，总结上一个 Sprint 中的活动，团队成员一起确定哪些需要保留，哪些需要改进，有哪些想要在新的 Sprint 中尝试。

3. 如何应用成功实践说明

1) 与人员：Scrum 团队所有成员，由 ScrumMaster 组织。

2) 频率：每个 Sprint 结束后，通常在 Sprint Review Meeting 之后进行。

3) 内容：

- What to Keep
- What to Change
- What to Try

不仅仅局限于流程、Scrum 方法本身，可以是任何对个人和团队效率提高、对项目有益的内容。

4) 方法：

为保证每个人有机会提出自己的观点，并不受他人影响：

- 每人三张 PostIt，分别写下“What to Keep”，“What to Change”,“What to Try”。每张纸最多写三；
- 将 PostIt 按类贴在白板上，由 ScrumMaster 主持统计，如下图所示：

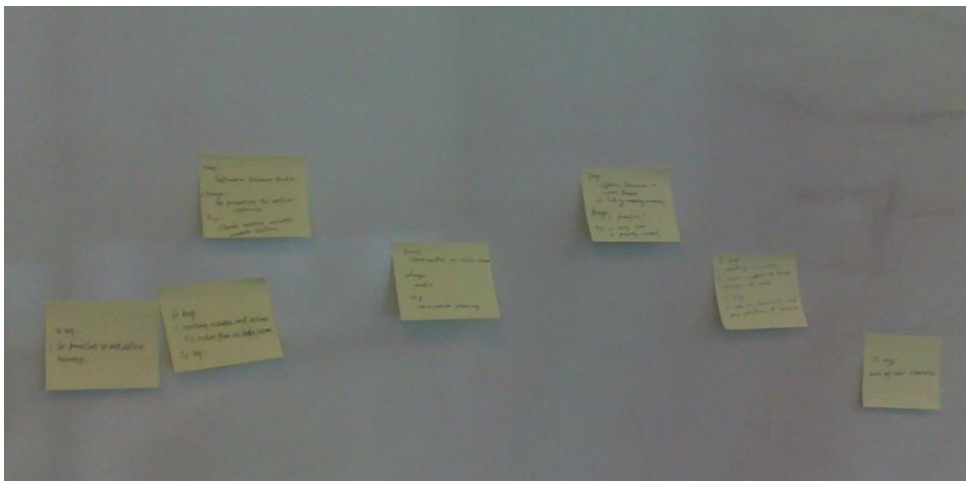


图 27-1 PostI

- 所有团队成员选出每一类中的前三条，作为下个 Sprint 中要执行的项目；
- ScrumMaster 将结果记录在 Scrum 管理工具中（ScrumWorks），并且将其贴在任务板(Task board)上供所有团队成员在进行站立会议(Daily scrum)时进行反省和互相提醒在实践中是否实施。

专家点评：

冲刺回顾会议是团队绩效不断提升的关键活动。本实践首先通过简单的方法引出团队成员的想法和建议，并由团队共同确定改进的优先级，最关键的步骤是在管理工具和每日站立会议上进行跟踪和反省，构成了 PDCA 持续改进的良性闭环。

点评专家：李春林

案例二十八：Sprint 回顾总结会议（英孚教育）

1. 案例说明

1) 案例来源

英孚教育

2) 项目基本信息

每个 sprint 最后一天的下午。

3) 组织结构

团队所有开发人员，由 Scrum master 主持。

4) 面临的问题

Scrum 由快速并且连续的迭代所组成，团队间不断的交流并产出产品，但是同样也会面临许多的问题需要解决。如果这些问题不被及时的反馈并交流，那很有可能影响到今后的开发速度和产品质量。同时也会影响团队氛围。我们要让团队，始终保持一种向上的并且能自我约束和管理的状态，那势必需要不断的交流，互相反馈和鼓励。

5) 解决方法

回顾会议，在产品发布之后，让大家去回想过去一个 sprint 所发生的事情，大家畅所欲言，肯定优势，改进不足。每次一个小时。

6) 主要收益

- 团队之间的互相肯定，保持了积极向上的氛围；
- 团队成员互相发现问题，并且提供平台，给他们互相提出意见，这样不仅是团队，就算个人，也是不断的回顾自身状态，不断改进。团队进步速度迅速；
- 鼓励发现对方亮点，互相学习；
- 团队成员开始在平时日常工作中，就开始留心需要改进的方面，从另一

方面来说，监督力度变大也更有效；

- 问题快速的解决；
- 无效的解决方案将在每次回顾会议说中被提及，并做相应改进，问题的解决，非常有效率。

2. 定义和特性说明

回顾总结会议，是每个 Sprint 结束后，把团队成员聚集在一起，谈论关于过去一个 Sprint 的感想。

大家各自发言，回顾团队的优势与弱势。并对弱势和不良习惯，提出个人意见及建议，由团队归纳，获得解决方案。

回顾为团队提供了一个机会：让大家能够作为一个团队，聆听来自每个人的声音、整合大家的想法并达成向下一个阶段迈进的共识。

3. 如何应用成功实践说明

团队回顾会议被安排在了每个 Sprint 最后一天的下午部分。安排在这个时候，也正是因为大家在忙活了整个 Sprint 之后，能有一个相对轻松的氛围，让大家来讨论。

会议仅仅邀请开发团队参加，由 scrum master 组织。一般来说，我们 2 周长度的 sprint 花费 1 个小时的时间，去完成回顾总结。会议大致分成 5 个部分。

1) 准备开始阶段（5 分钟）

由 scrum master 完成对过去一个 sprint 的简短总结，强调大家在 sprint 中那些做的不错的地方，主要是调节气氛振奋团队情绪，引导团队开始思考。同时也总结一下上次回顾会议之后所留改进项目的状态。

2) 收集反馈（10 分钟）

每个团队成员在记事贴上，留下他们对上一 sprint 的不满或者满意点，并贴在白板上（每个人可能有很多贴，但是可能有重复，采用这种方式，如果自己和别人的意见重复，那就可以不要再贴了）

3) 选取最重要的进行讨论（20 分钟）

大家在自己最关心的问题上，做记号。最终我们会选择整个团队最关系的 3 个事件进行讨论。讨论将会收集团队成员对此问题进行头脑风暴，试着想出结果。

4) 解决方案确定（15 分钟）

依照各方建议，对于 3 个最关心事件，依次找出一个适合团队的行动计划，并在下个 sprint 中，予以使用。

5) 选取本次 Sprint 的最佳表现成员（5 分钟）

团队成员互相投票选出本 sprint 中，最佳表现成员，之后互相说出，为什么会选这名队员。此举主要帮助我们去发现队友身上的闪光点，并且作为借鉴，去学习。

6) 总结，结束（5 分钟）

对优点再次肯定，予以表扬；收集所有剩余提出意见，备用。

专家点评：

这是一个充分体现民主与集中的较成功的 sprint 回顾总结案例。该案例分享了各成员组如何充分地总结这个 sprint 中的成功和失败之处，并将大家最关心的 3 个事件放在下一 sprint 中执行。

记得 SCRUM 中有个规则，defect 永远具有最高优先级。总结回顾大会也应该对产品本身的 defect 进行讨论并放入下一 sprint 的高优先级的 product backlog 里。

点评专家：黄晓倩

案例二十九：轮值 ScrumMaster（特艺）

1. 案例说明

1) 案例来源

特艺（中国）科技有限公司

2) 项目基本信息

Research 项目，每个团队成员有各自主攻的研究方向，相对独立。

项目基本信息如下：

- 团队规模：4-5 人
- 主要交付物：算法、实验用代码、demo、技术调研报告等。

3) 组织结构

标准 Scrum 组织架构

4) 面临的问题

由于 research 工作的相对独立性，团队成员对其他成员的工作不够了解，缺乏有效交流。

5) 解决方法

轮值 ScrumMaster

6) 主要收益

- 通过担任 ScrumMaster，每个团队成员对项目有了整体的认识和了解，加强对其他成员工作的了解，从而促进整个团队的有效沟通；
- 额外收益：作为部门领导，通过轮值 ScrumMaster 可以很容易识别出有领导潜力的成员，有助于进行针对性的培养。

2. 定义和特性说明

每个 Sprint 由各 Scrum 团队成员轮流做 ScrumMaster。

3. 如何应用成功实践说明

1) 参与人员：Scrum 团队所有成员。

2) 频率：每 Sprint 轮换一次。

3) 内容：每次 Sprint 由 Scrum 团队中的成员轮流担任 ScrumMaster。

4) 方法：

ScrumMaster 负责组织 Sprint Planning，Sprint Review，Sprint Retrospective，Scrum Meeting；总结项目周报告和月度报告；记录会议的重要结论、action plan 等。负责解决 Sprint 中遇到的各种问题，跟踪风险，确保 Sprint 顺利完成。

5) 注意：

只有经过 3-5 个 Sprint，当所有团队成员对 Scrum 方法及 ScrumMaster 的职责有了一定了解后，才可以使用这种方法。

专家点评：

看题目就感觉很新颖，“轮值 ScrumMaster”是一个有趣的尝试，当然，这个尝试也很有意义。案例中分享的收益是“每个团队成员对项目有了整体的认识和了解，加强对其他成员工作的了解，从而促进整个团队的有效沟通”，而我理解还有一个重要的收益是，这可以加强每个团队成员的责任感、使命感，加深每个成员对 Scrum 原则的理解和发自内心的认同感。

点评专家：李春林

案例三十：敏捷导入之 Scrum Master 培养（新聚思）

1. 案例说明

1) 案例来源

新聚思（Synnex）信息技术有限公司

2) 项目基本信息

在六个月内，培养 10 个以上合格的 Scrum Master。建立起 Scrum Master 认证体系。

3) 组织结构

- 5 个开发部门，超过 30 人的项目经理团队；
- 3 个资深 Scrum Master, 1 个敏捷培训师。

4) 面临的问题

- 项目经理对敏捷/Scrum 的了解程度参差不齐；
- 不知道拥有什么样的技能才能被称为 Scrum Master；
- 公司内、外的统一培训针对性不强，准 Scrum Master 难以获得有针对性的帮助。

5) 解决方法

- 建立公司的 Scrum Master 能力需求模型；
- 进行初步的评估，确定 Scrum Master 培养名单；
- 根据评估后的差距分析，做有针对性的统一培训；
- 制定个人能力提升计划，以三个敏捷项目的评估为里程碑；
- 执行评估；
- 最后资质认证（即最后一次项目评估）；
- 发给资质证书。

6) 主要收益

建立了 Scrum Master 培训及认证的标准过程。有效地培养了大量合格的 Scrum Master，使公司敏捷导入不存在人才上的障碍。

2. 定义和特性说明

做为敏捷项目团队的精神领袖，Scrum Master 能力在某种程度上决定了敏捷项目的成败。Scrum Master 的挖掘及培养也在某种程度决定了企业敏捷导入的成败。

本实践建立了对 Scrum Master 的能力需求模型，通过规范化的个人能力评估方法，找出个人差距并加以培训。

3. 如何应用成功实践说明

1) 根据组织需求建立对 Scrum Master 的能力需求模型

一个合格的 Scrum Master，不但要掌握敏捷、项目管理等方面的知识、工具，还面要具备较强的沟通、领导力等软件技能。在敏捷导入阶段，能力需求可能为分以下几方面：

- Scrum 会议

在 Scrum 项目中，Daily Meeting,Planning Meeting, Demo Meeting 和 Retrospective Meeting 是项目沟通的几个重要会议，它们贯穿了整个项目的生命周期。Scrum Master 对这些会议的理解、会议内容及会议中可能出现的各种情况的处理能力，决定了这些会议是否能够达到预期目的。

- 计划与估算

计划与估算实际上是项目管理的基本能力要求，在敏捷开发中，估算方法及要求又有不同。其内容涉及 Backlog 及 Story 的分拆，各种协作模式下的规模(size)及工作量估算，发布计划、Backlog/Story 优先级确定，任务认领以及对异常情况的处理。

- 可工作的软件

在 Scrum 开发鼓励按 Story 交付。Scrum Master 需要通过引导项目成员的工作方式，加强测试，持续集成等做到这一点。

- 可见性

项目的高可见性可以保证项目的目标不被误解，项目的状态能被上级、客户、团队充分了解，项目的问题能够得到充分的沟通并被解决。敏捷方法提供了许多工具可以保证项目的可见性，如各类会议，白板，燃尽图等。Scrum Master 需要确认这些工具被正确的使用以保证项目的可见性。

- 有效的学习与交流

Scrum Master 对团队成员的培训，团队内部及与客户、上级的沟通与交流。Scrum Master 自身也需要保证对敏捷方法的持续学习。

2) 规范化的个人能力评估方法

- 根据能力需求模型，结合组织内客户、项目管理、流程的具体情况，针对每项能力需求，建立检查列表；
- 由资深 Scrum Master 及 Scrum 教练组成评估团队；
- 在评估个人能力时，采用访谈与项目回顾相结合的方式，定期进行。每次评估，找出差距，并给出改进建议；

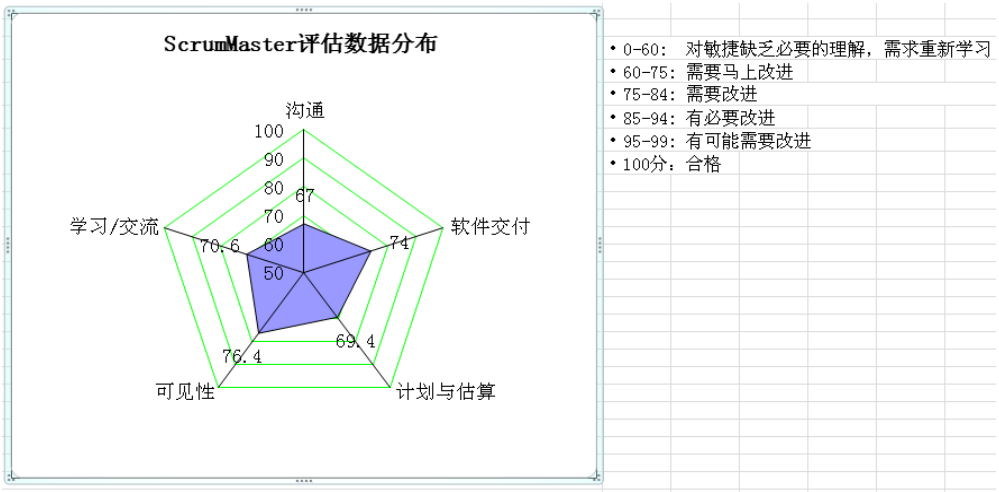


图 30-1 评估图

- 每位准 Scrum Master 必须经过至少三个项目的评估，且最终通过评估。

3) 认证体系

- 对每个通过评估的 Scrum Master 给予资质证书；
- 在认证有效期内，必须完成一定数量/规模的敏捷项目；
- 在认证有效期到期前，需要再进行一次评估，若通过，则给予新的资质证书。

专家点评：

非常认同本案例所提出的“Scrum Master 的能力在某种程度上决定了敏捷项目的成败”，而合格的 Scrum Master 需要具备敏捷基本知识和沟通及领导力。本案例中的 Scrum Master 评估和资质认证的流程很有意义，而且强调 Scrum Master 保持对敏捷方法的持续学习和实践总结也是非常重要的。

点评专家：陈志波

案例三十一：分布式敏捷开发（上海惠普）

1. 案例说明

1) 案例来源

上海惠普公司

2) 项目基本信息

Service Catalog Management 子模块开发。

项目基本信息如下：

- 团队规模：20 人 美国 6 人 UI 设计， 欧洲 4 人 上海 10 人开发 app
- 环境：.Net

3) 组织结构

分布式 Scrum

4) 面临的问题

在全球化的背景下,如何高效集中全球团对完成大型软件开发项目。

5) 解决方法

采用分布式开发敏捷成功实践；迭代周期：2 周,主要活动：集中训练营,Scrum of Scrums Planning, Scrum of Scrums meeting, Scrum of Scrums review.此模式特别适合在不同地方人员比较多的大项目使用 采用一下对应的模式。

6) 主要收益

通过 Scrum of Scrums 的实践,由于及时地沟通 rework 减少 8%,bug 减少 10%,大部分开发工作在中国,所以成本降低。

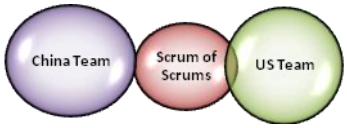
7) 局限

模块间的依赖度不能太大。

2. 定义和特性说明

分布式敏捷开发一个共同的软件系统，但是研发人员却受制于距离和多个不同时区、的文化和语言差异的影响。

3. 如何应用成功实践说明



客户需要一种流程,方法或者实践能够帮助全球化团队开发同一个项目,我们推荐 Scrum of Scrums 的实践给来帮助客户,专注于集中沟通,协作。定义了统一沟通的时间 Scrum of Scrums meeting,由和 2 个时区有交集的时区的 ScrumMaster 来组织会议沟通,不同时区的 team 派出 ScrumMaster 和主要核心成员参加会议,在会议上沟通问题,功能间制约,用同样的工具共享 product documents and source code, demo 成果。

开发采用日不落方法,美国休息,中国在开发,反之美国开发,中国休息,欧洲作为之间沟通枢纽。在项目初始阶段让不同地方的人员集中到一个地方开始集中训练营活动，之后回到各自时区去传授经验。

专家点评：

在全球化进程日益加速的背景下，如何有效协调地处不同时区、不同地域的全球开发团队，快速交付高质量的产品是当前众多企业所面临的一个重要课题。上海惠普案例很有代表性，他们的集中训练营活动、Scrum of Scrum 等实践给我们提供了很好的借鉴；而利用全球时差、采用日不落的开发方法更是化弊端为优势典型案例。

点评专家：李春林

在日益全球化的今天，这种分布式的敏捷开发给目前越来越多的跨地域开发的项目提供了一个实用的解决方案。个人认为前期的集中训练营非常重要。由于语言、文化的差异，也许这种敏捷项目所需的书面文档要比普通项目多一些。如果有机会希望能够了解更具体的实施方法。

点评专家 :刘德意

案例三十二：Sprint Champion（特艺）

1. 案例说明

1) 案例来源

特艺（中国）科技有限公司

2) 项目基本信息

Research 项目，每个团队成员有各自主攻的研究方向，相对独立。

3) 组织结构

- 团队规模：4-5 人
- 主要交付物：算法、实验用代码、demo、技术调研报告等

4) 面临的问题

各团队成员的工作相对比较独立，对其他成员的工作了解不多。不利于整个项目的发展，每个成员容易陷入研究的死胡同。需要鼓励大家互相交流的积极性。

5) 解决方法

通过 Sprint Champion 评选出最活跃的提问“明星”

6) 主要收益

促进了团队成员间主动交流。通过这种积极的交流，产生出许多新的想法，对研究内容很有帮助。活跃团队气氛，使大家工作更愉快。

2. 定义和特性说明

选择团队中需要改进的地方，总结出度量方法和标准，在 Sprint Retrospective 之后进行评选。

3. 如何应用成功实践说明

1) 参与人员：Scrum 团队所有成员，由 ScrumMaster 组织。

2) 频率：每个 Sprint 结束后，通常在 Sprint Retrospective Meeting 之后进行。

3) 内容：根据团队的需要，制定出评选的内容。比如，在 Technical Review 中，评选出提问 Champion。鼓励大家认真听并思考其他成员的研究内容/成果，并提出有价值的问题，从而拓展对所研究内容的思路，使大家互相借鉴。

4) 方法：在 Technical Review Meeting 中，指定一个成员记录每个人提问的次数，最后在 Sprint Retrospective 之后评选。Sprint Champion 会得到一个小礼物，并且免费享用庆祝 Sprint 完成的工作餐。

图 32-1 小礼物

专家点评：

敏捷开发的核心价值观之一就是“以人为本”，即“个体和互动 胜过 流程和工具”。该实践正是通过一些小仪式、小礼物来激发个体参与的积极性，创造相互交流、轻松活泼的团队气氛，是从细节上将敏捷价值观落地。

点评专家：李春林

案例三十三：并行测试（用友软件）

1. 案例说明

1) 案例来源

用友软件股份有限公司

2) 项目基本信息

在用友 NC 供应链产品 6.0 开发中使用并行测试。

项目基本信息：

- 团队规模：74 人
- 环境：Java

3) 组织结构

需求 8 人程序员 50 人测试 18 人

4) 面临的问题

产品模块间接口多，流程长，前后关联度高，一个接口问题会导致后续接口无法测试。整个流程接口全部测试周期过长，整体稳定慢。

5) 解决方法

建立统一的基础数据准备，编写详细到数据的接口测试用例，在统一的基础数据上作接口用例的业务数据准备。所有数据准备完成脚本化，在新测试环境搭建后插入。安排实习生执行，快速回归接口用例。两天内可以回归所有用例。每名实习生每天可以回归 100 个用例。

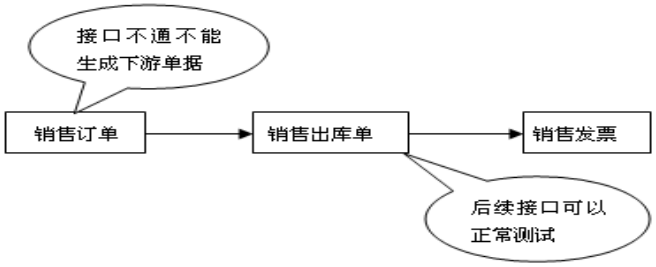


图 36-1 流程

主要活动：数据准备、用例编写、测试环境搭建、用例回归。

6) 主要收益

快速全面回归接口用例（1 个月缩短到 2 天），根据数据准备程序员可以快速重现问题并解决。

2. 定义和特性说明

在有数据准备的前提下，有前后依赖关系的一系列用例，可以安排不同人员或设备同时测试。

3. 如何应用成功实践说明

测试用例需要明确到数据细节.需要统一使用提前规划好的基础数据.数据准备使用工具抽取成数据库脚本.在测试前插入使用的测试环境数据库.数据准备需要按照测试功能的前一步数据状态进行准备.插入数据后可以直接进行当前要测试的功能测试。

专家点评：

在敏捷开发中，测试很容易成为“瓶颈”。除了提高自动化测试能力之外，并行测试也是很好的思路。用友医疗建议的前期统一数据库准备和详细到数据层面的接口测试用例，是不错的测试实践。

点评专家：赵静

案例三十四：结对编程（用友软件）

1. 案例说明

1) 案例来源

用友软件股份有限公司

2) 项目基本信息

在开发 XX 银行资金交易系统的过程中，项目基本信息如下：

- 团队规模：6 人
- 环境：NC5.6 平台 & ORACLE10G

3) 组织结构

标准 SCRUM 架构

4) 面临的问题

- 遭到其他技术更高的程序员更具“威胁”的程序员，会担心自己被视为无能；
- 结对编程之前，程序员需要更多时间去考虑，但他往往在仔细考虑自己的想法之前就被强迫开始结对编程了；
- 内向的程序员喜欢独自工作(不合群的人)；
- 程序员可能互相讨厌对方。

5) 解决方法

结对编码的所有问题和阻力都在于人，在品尝到结对编程的好处之前对期不解也是正常的。尽早让团队感受到结对编程的优势，另外 master 多在团队成员中做工作，相信面临的问题都可以解决。

6) 主要收益

- 持续不断的头脑风暴、更大的知识库、在理解上有更少的差异、有更多的脑力解决设计问题；

- 更少的漏洞、想法的即时认证、始终如一的方法并更加遵守团队会议中的要求；
- 经验共享与知识共享、对于为什么做、怎么做和做什么有更深入的理解；
- 更好地集中精力及更高的工作强度、彼此促进并激励来达到最好的结果、更少的拖延和时间浪费；
- 大多数人喜欢分小组工作并且共同解决有趣的问题。

2. 定义和特性说明

程序员肩并肩地坐在同一个工位合作完成同一个设计。

3. 如何应用成功实践说明

1) 两个人共用一个工位工作，方便沟通和交流；

2) 两个人共同完成一个功能模板的开发。在开发过程中两个人一起讨论，相互交替编码，虽然同时只有一个人在编码，但另一个人会在一旁指导，同时也可以重新审视已完成的代码的逻辑和算法，使代码评审工作在编码阶段就已经完成，提高了不少效率和代码质量。

专家点评：

结对编程，可能还有很多人认为是在浪费时间，一个人可以完成的工作，为何要两个人做？用友医疗的案例，很好的解释了这个问题。我们更看重交付代码的“质”，而不仅仅是“量”。

点评专家：赵静

案例三十五：结对编程&代码 Review&团队变更管理

（用友软件）

1. 案例说明

1) 案例来源

用友软件股份有限公司

2) 项目基本信息

医疗基层卫生服务系统产品研发。

项目基本信息如下：

- 团队规模：20 人
- 环境：JAVA、用友 UAP

3) 组织结构

矩阵式组织架构，包括需求、UE/UI、开发、测试来自不同部门的人员组成产品团队，参照 Scrum 组织架构，产品项目经理为 Scrum Master。

4) 面临的问题

产品开发人员对相互业务不了解，产品代码质量较低，如果团队人员变动，开发活动接续困难。

5) 解决方法

用 Product Backlog 作为产品的需求规格。周期性收集用户的业务需求并表述为用户故事。与团队成员一同建立和细化用户故事。每一次迭代结束后，与用户确认，对需求订单进行维护。

采用变形的新结对编程方法。举例说明，两个开发人员甲和乙，形成结对编程小组，共同负责两个功能 A 和 B。工作开始，甲进行 A 的设计工作，并给出设计文档。乙进行 B 的设计工作，并给出设计文档。两人的设计工作完成后，互相阅读对方设计文档并进行交流，两人要保证对对方的设计内容了解清楚，并指出

对方文档中描述不当的地方让其修改，最终保证设计内容是清晰明确的，而且甲、乙两人对 A、B 功能的设计都熟悉明确并思想一致。接下来开始代码开发工作，甲按照乙的设计文档开发功能 B，乙按照甲的设计文档开发功能 A。代码开发工作进行到某个段落后，甲对乙开发的功能 A 代码进行审查，检查代码格式和规范以及代码实现是否符合甲设计的业务逻辑，如有问题，则请乙进行修改并复查，乙同样反之对 B 进行审查，最终达到代码开发完毕，A、B 两个功能都进行过多次审查和修改，而且甲、乙两人都确认两个代码是合格的，并符合业务需求的。这样，功能 A、B 的设计和代码实现就是有甲、乙共同协作完成的，两人共同对两个功能负责，并且两人对两个功能的设计和实现都清楚，更为重要的是甲、乙两人的工作是相互并行配合的，不会产生冲突，更为容易推广施行。

6) 主要收益

按照可实践的结对编程模式完成软件开发，不仅高效的完成了代码检查，提升了软件质量，还带来了一些额外的好处：1、每个模块至少两个人熟悉，这样在有人请假的时候都有 backup，不会造成严重的耽误工作进度。如果有人要离职，交接工作通常也会非常轻松，几乎没有什么需要特别交接的。2、进行结对的两个开发人员相互阅读设计文档和代码实现，实际也是一个相互学习的过程，所以结对编程也起到了一些知识传播、培训的作用。

2. 定义和特性说明

在敏捷软件开发的各种实践中，结对编程（Pair Programming）是特别有争议的，尤其是在国内的软件企业或团队，几乎看不到有真正实践、施行，并带来效果的。相比于其他敏捷软件开发方法的火热及它们带来的成效，结对编程可以说备受冷落。它看上去很美，但想要成功的实践，可谓障碍重重。那么实用为王，我们是不是可以秉承结对编程优秀的核心理念和思想，而将它的实践框架加以改造，让它更适合我们的工作流程和习惯，最终成功实践并提升研发效率和质量呢？

1) 传统结对编程定义及特性

两位程序员形成结对小组，肩并肩地坐在同一台电脑前合作完成同一个设计、同一个算法、同一段代码或同一组测试。

结对编程可以看作是一种敏捷化的代码检查（Code Review），其最终目标是提高软件产品的质量。代码检查工作是公认的提高软件产品质量的重要活动之一，但在实践中，成功应用的也并不多见，原因是传统的代码检查方法只能在某个功能完全开发完毕后才进行检查，效率不高。而更大的问题是，检查人对被检查人的代码所实现的业务功能并不十分清楚，所以只能检查代码的格式、语法是否规范，对实现逻辑是否满足业务需求往往无法检查。这导致了很多企业和团队的代码检查都流于形式。

结对编程就解决了这一问题，它不需要代码都写完才开始检查，而是在两个人相互协作过程中，实时的进行多次交叉检查，大大提高了效率，而且因为两人同时完成设计和代码，对业务需求也都了解，所以检查时可以验证代码是否满足了需求、是否符合业务逻辑。但是，传统的结对编程模式也有一些显而易见的问题导致其难以实践，例如人们担心两个用一台电脑做同样一件事情，是不是浪费了人力资源？两个人的个性、习惯、水平都不同，在对等的地位上同时做一件事情会不会产生冲突和矛盾？他们之间的沟通和协作会顺畅吗？会不会感到不自在？

2) 新结对编程定义及特性

两位程序员形成结对小组，每人一台电脑，坐在临近的工位上，两人合作完成一组功能（可以是两个或多个独立的模块）的设计、代码实现。但对于某一个模块来说设计和代码是分开的，一个人负责设计，另一个人负责写代码，对于其他模块则反之。当某个功能阶段性完成后，由其设计人员对代码进行检查。目前，很多敏捷开发的倡导者们都在积极的寻找可实践的结对编程的变形体，以打破结对编程这种看上去很美，摸上去扎手的情况。而上述这种形式的新结对编程，也是我们几经探索总结出来的一种可实践的结对编程形式，并在产品团队中成功应用。它继承了结对编程的核心理念，让代码检查更加高效、深入。也在两人合作模式上，更适应实际工作情况和工作习惯，以做到真正的可实践、可应用。

3. 如何应用成功实践说明

在可实践的结对编程中，两名开发人员结对形成一个小组，临近而坐，共同负责一组功能模块，在具体的某个功能或任务上，两个人所扮演的角色是有承接

关系的，一个是设计者、另一个是开发者，一个是审查者、另一个是被审查者。在单个点上，两人互不干扰，独立运行，都有自己的发挥空间。而在在总体上，两人又形成一个整体，要分别对对方的设计和代码熟悉了解、审查及负责，而且两个人的关系又是平等的，因为设计与开发、审查与被审查对于两个人来说都是相互的。这样就能够让两人在工作中共同协作，提高效率和质量，而且又同时避免了一些可能会产生的冲突与矛盾。

结对编程中的设计环节，也是敏捷化的。敏捷开发与传统瀑布型开发一个很重要的区别就是：在研发活动中各环节的流转的承接时，瀑布型开发是以文档为主，思想的交流与沟通为辅。而敏捷开发是以思想的交流与沟通为主，文档为辅。所以在瀑布型开发中，对设计文档的格式和规范等要求比较高。而敏捷开发中的设计文档，只是要描述清楚设计思想，并作为一个辅助手段，将关键内容存留作为依据即可，不需要规范的模板。可以根据设计者的习惯，以最高效的形式描述清楚设计内容，并且清晰明确即可。在结对编程的实践中，所涉及到的设计文档的书写，都是以此种方式进行的。

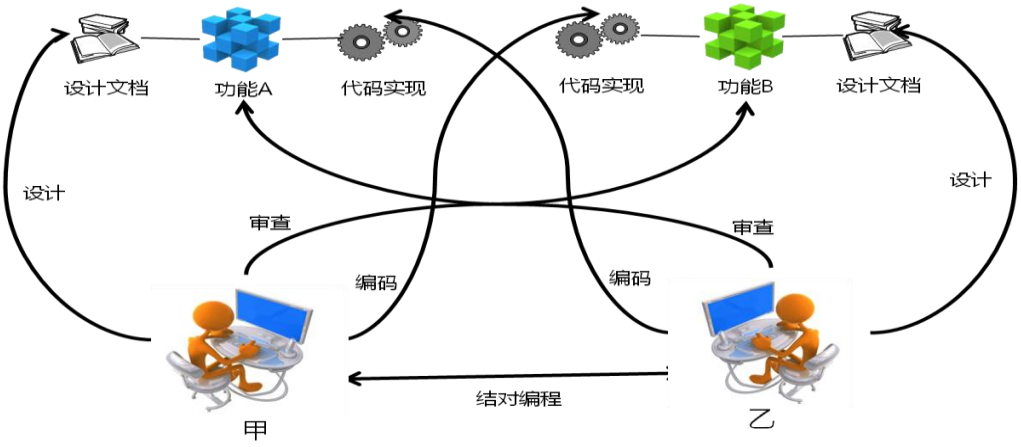


图 35-1

专家点评：

结对编程是一种实时的 review，在提高代码质量、促进人员成长等方面有显著收益。可是真正要实施结对编程还是有很多困难的，包括资源支持、性格搭配、工作习惯等方面。用友的这套“新结对编程”方案思路新颖，不仅在编码阶段能够充分利用结对编程和 review 带来的好处，还把这种理念推广到了设计阶段，并且在面临团队成员变更的风险管理方面也大有裨益，值得一试。

点评专家：郜晓梅

案例三十六：演进的设计(汤森路透)

1. 案例说明

1) 案例来源

汤森路透

2) 项目基本信息

在某自动化处理配置应用中，通过设计的演进，快速交付了初始版本，并在后续功能开发中，逐步对设计进行改进，即争取到客户的信任，又避免了过度设计的问题。

3) 组织结构

标准 Scrum 组织架构

4) 面临的问题

- 系统开发初期，需求不完整，无法预知全部需求；
- 后期需求变更引起设计变更。

5) 解决方法

模块早期划分；最小化，最简化设计原则；灵活的接口设计和ＤＢ建模。

6) 主要收益

避免了早期过渡设计的浪费；增加系统灵活性；快速交付可用产品。

2. 定义和特性说明

演进的设计是指软件组件的设计随软件开发而不断增强进而满足新的需求。

3. 如何应用成功实践说明

在项目早期，架构师/程序员结合需求订单，从而对软件有总体的大致理解，从而进行模块划分。基于此模块划分，开展对各模块的设计。

设计须遵循以下原则：

1) 设计仅满足已提出或可预见的需求。例如，如果仅有顺序处理调用的需求，就没有必要实现有循环，分支需求的工作流引擎。

2) 设计尽可能简单，直接。

3) DB 建模时应保持一定灵活性。

4) 模块接口的设计应具有一定的灵活性。

在每次迭代开始前，针对已确定的需求，对可能设计变更进行讨论。例如，如果用户要求处理流程有循环，分支控制功能，则考虑是否引入工作流引擎。

在设计变更时，除考虑上述原则外，还应：

1) 回顾需求变更历史，挖掘相似需求。

2) 检查其他子系统，提取相似需求，实现共通组件。

Artifact ID	GR2796
Title	(Generic IFFM)New generic workflow
Description	1.DataModel and package. Defined activity and transition.Defined rule and parameters. 2.Migrate Generic IFFM and test 3.Change java data layer code 4.Change savefeed and import etc. java service layer code 5.Check and change java action layer and javascript code
Sprint	15
Component	JAVA_GENERIC_IFFM
Category	1 - Service Support
Importance	12
Estimation	7.5
Developer	Jacky Chen
QA	
Team	IFFM
Status	Development
Req. Type	New Requirement
Comments	
Release Date	12/31/2011 12:00 AM
Number of ST test cases	
Number of ST automation test cases	

图 36-1 迭代

图 36-2 演进图

专家点评：

演进的设计，是迭代开发的重要实践之一。在确保简单设计、灵活接口的基础上，提前预估变更，并且随时提取相似需求，实践共通组件。这种设计思路，在有效保障尽早提交可运行的软件的同时，不失对整个系统架构的渐进式地加强，确保系统的强壮。是一种循序渐进、增量式的设计开发思路。

点评专家：刘德意

案例三十七：质量风险前移（广联达）

1. 案例说明

1) 案例来源

广联达软件股份有限公司

2) 项目基本信息

产品质量不好，问题多，而且因为质量还造成项目的产能也存在较大的问题，远远不能满足销售部门及市场的需要，而项目则深陷“反馈处理”及“救火”的泥潭。

项目基本信息如下：

- 团队规模：30 人（分 5 个小组）
- 环境：Windows

3) 组织结构

30 人，分 5 个小组，各组之间都基于一个产品平台做针对不同客户群但业务差不多的产品，而各组间相互独立又偶尔有一些关联。

4) 面临的问题

产品质量不好，问题多，反馈多，产能严重不足。

5) 解决方法

逐步实施“质量风险前移”，提高平台质量，提高产品质量，提高团队的凝聚力和战斗力。

6) 主要收益

- GBQ 产品的客户满意度：87.85，比上半年提高 2.44；
- GBQ 的产品开发产能比上一年增加两倍多（按对市场发布的版本数，本年：上一年为 92:28；按发布一个版本的开发工日，本年：上一年为 40:138）；

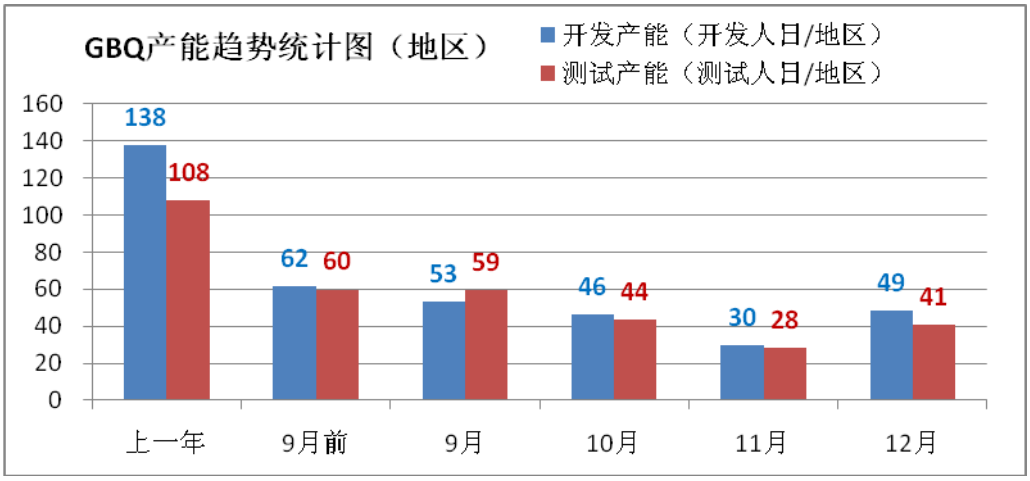


图 37-1 GBQ 产能趋势统计图

- 计划目标的延迟率从一季度的 57%下降到四季度的 25%；

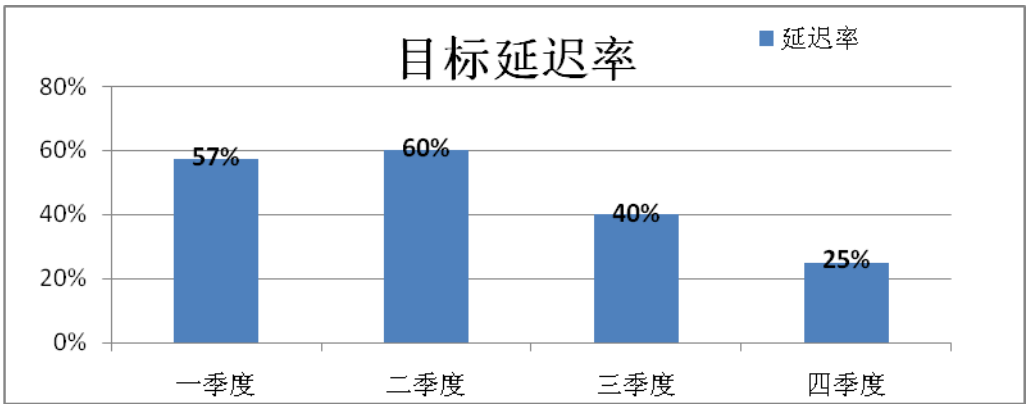


图 37-2 计划目标延迟（季度）

- 通过质量风险前移一些活动的实施（高质量的需求交底、严格的自测、高覆盖度的自动化测试等），提高了产品质量，并且提高了团队的产能，之后就基本满足了销售部门及市场对产品的需要；
- 团队形成良好的质量意识和质量习惯，进入正向循环，产品质量越来越高，计划越来越可控。

2. 定义和特性说明

就是在一个项目的过程中，把有可能发生的质量风险尽量提前引发、克服、应对。尽早在风险和代价小的时候解决问题，对项目结果最有益（投入产出比）。

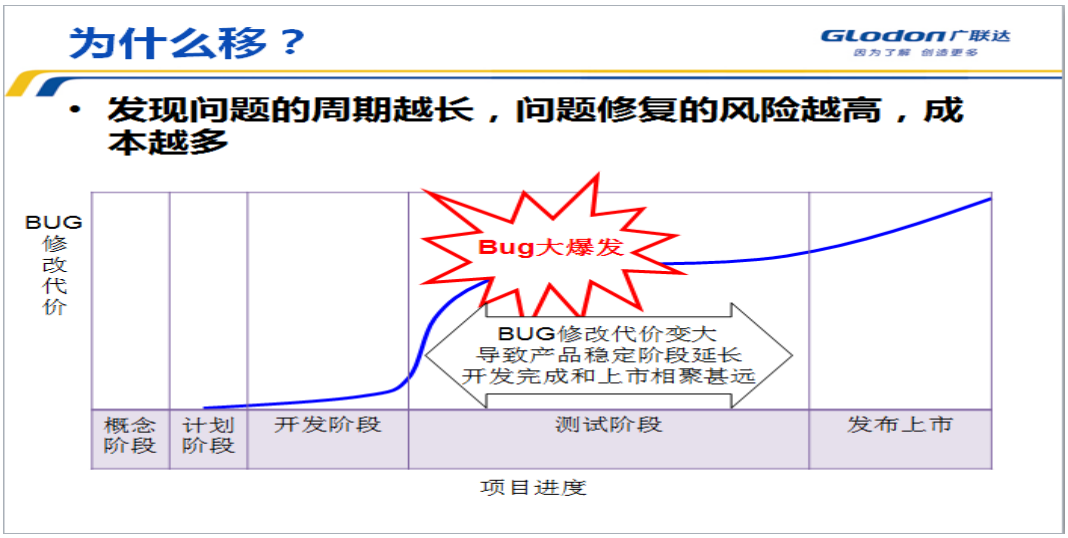


图 37-3 所有提升质量的活动都可以作为质量风险前移的操作活动（1）

图 37-4 所有提升质量的活动都可以作为质量风险前移的操作活动（2）

3. 如何应用成功实践说明

在项目早期，架构师/程序员结合需求订单，从而对软件有总体的大致理解，从而进行模块划分。基于此模块划分，开展对各模块的设计。

要在一个项目成功实施质量风险前移，一定要有一个整体的实施规划，一般有如下几个步骤：

1) 破冰松土：一般要先从破冰松土开始，先给大家建立起质量风险前移的概念，让项目组认识到质量风险前移是什么、为什么

2) 选试点：选试点分两个维度，一个维度是选试点项目团队，要选积极响应改进意识和动力强的团队作为试点；另一个维度是选质量风险前移的活动，而选择试点的改进活动有一个原则：要选择“简单、对项目干扰小、易见效”的活动来进行改进实施

3) 界定成绩，树立标杆：过程中让所有可能参与变革的人，都明确的知道每一个小变革所产生的作用和带来的价值，从而增强其坚持变革的信心。另外一定要记住：榜样的作用是无穷的！

4) 总结经验,巩固成果：实施过程中不断总结经验吸取教训，以便后续的变革过程更顺利进行。为保障团队能持续的坚持优化后的方法，过程中需不断的加深大家的理解，巩固成果，直至该方法转化为团队习惯

5) 推广,循环前四步：试点成功后，方法在组织内推广使用，并利用改进的成效有效促进后续其他组、其他质量活动改进的顺利实现，一定要形成正向的循环。

专家点评：

“质量风险前移”是一个原则、一种理念，而要将其落实下来则是一种文化。具体采用哪些活动前移质量风险并不重要，而如何采用一套组合拳来建立这种文化才是问题的关键，这正是本案例所分享的。

点评专家：李春林

案例三十八：工具助力 Scrum 执行（东软）

1. 案例说明

1) 案例来源

东软集团

2) 项目基本信息

某音响设备的软件开发项目，客户要求采用 Scrum 开发模式进行管理。

- 团队规模：15 人
- 项目特点：全新项目，没有历史经验，分布式团队开发，新员工/中级员工/老员工分布比例为 1：1：1。
- 开发语言：C＋＋/C，JAVA
- 环境： Android 2.2

3) 组织结构

标准 Scrum 组织架构

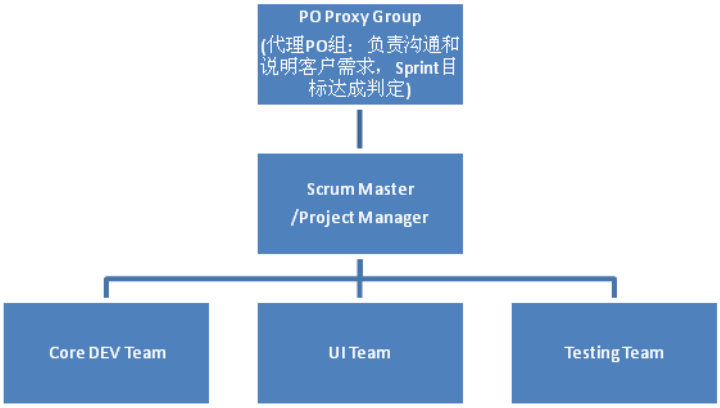


图 38-1 标准 Scrum 组织架构

4) 面临的问题

分布式团队协作、需求不明确、需求变更频繁、Kanban 信息只适于存档（不适于再拷贝、粘贴、计算等）。

5) 解决方法

采用 Scrum 成功实践；引入 Rally/Scrumworks Basic 1.8.3 工具进行敏捷项目管理。

主要活动：

- 三个工件(Product Backlog、Sprint Backlog、Burndown Chart)使用工具管理和生成日会结合工具的看板进行工作内容的展示；
- Planning Meeting、Review Meeting 以及 Retrospective Meeting 结合工具进行；
- 分布式团队通过工具可以看到工作进展。

6) 主要收益

- 实现管理工具的自动化；
- 实现分布式开发管理；
- 实现工具和白板的完美融合；
- 通过工具的引入，Team 在日会上花费的时间减少了 33％。

2. 定义和特性说明

选用合适的 Scrum 管理平台软件，既保证了 Scrum 管理项目的高效灵活，也解决了分布式开发的实际难题。

通过在线的 Scrum 管理工具，帮助项目组进行 Product Backlog、Sprint Backlog 等的跟踪管理。选择适合项目的 Scrum 工具，对项目的效率提升尤为重要，所谓“适合”的工具必须具备以下几个特性：具备灵活的需求管理功能；支持分布式开发场景；能够和 Scrum 本身（至少三个工件）完美融合；能支持白板展示功能。基于以上四点，最终选择了 Scrumworks Basic 以及 Rally。

3. 如何应用成功实践说明

1) 背景

在实施 Scrum 项目初期，通过白板进行项目跟踪，用于每天的 tracking，还是非常不错的。但白板对于 Product Backlog 和 Multi-site 团队的支持不是很理想。也曾使用 Excel 进行，但是在成员多的情况下，修改时会相互冲突，不好同步。在这种情况下，考虑引入 Web 方式的 Scrum 管理工具。

2) 准备阶段

工具导入前，对工具进行评估，分析其优缺点和项目适用性。

整理工具和流程配套使用的相关规则说明，指导项目组在工作中使用工具，重点在如何让 Scrum 中三个角色（PO，Scrum Master 以及 Team）知道自己需要使用工具干什么，以及五个仪式和工具如何结合需要说明清楚。

指定工具维护和管理人员。

3) 导入阶段

Team 培训，在准备培训资料时，要求不只单纯的介绍工具的使用，而是要求结合工作流程介绍工具的具体使用方法，将流程和工具融为一体，明确工具如何和五个仪式融合。

示例一：Scrumworks 与五个仪式的融合

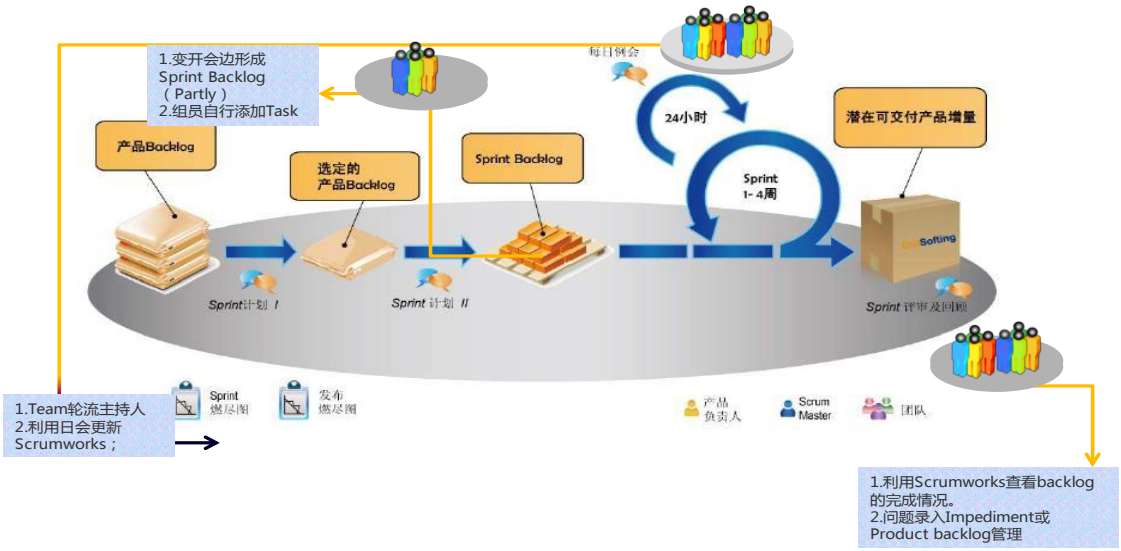


图 38-2 Scrumworks 与五个仪式的融合

示例二：Rally 和 Scrum 五个仪式的融合

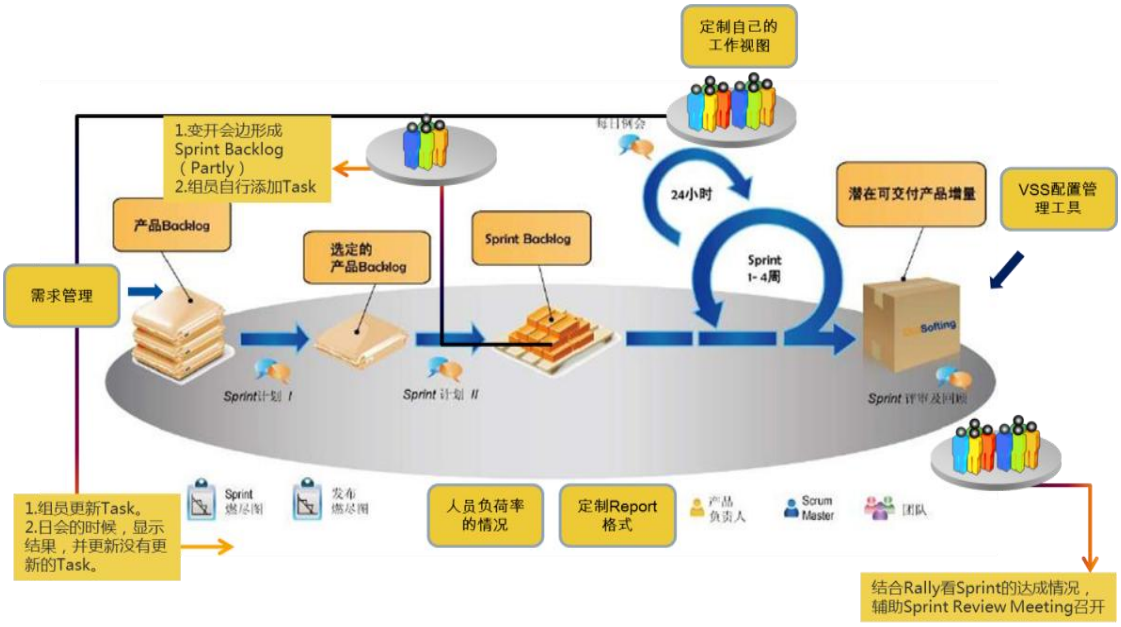


图 38-3 Rally 和 Scrum 五个仪式的融合

4) 试运行

目的：通过试运行，保证既定规则在项目组实际应用和适当调整；将工具不会使用以及规则不熟悉的问题解决。

Scrum 工具使用初期，需要每天监控，必要时和 Team 中每个人沟通。

尽量发现不适宜的规则，并及时调整，同时向 Team 明确。

监控和保证五个仪式和工具的结合使用。

监控并保证三个工件通过工具形成。

注：这里的规则可能包括如何更新、临时任务如何处理、没有完成的 Backlog 如何处理等等，尽可能把大家工作中遇到的困惑进行明确的说明，并需要不断的补充。

5) 正式上线

可以利用工具收集一些数据，做必要的度量分析。

示例一： Sprint Burndown Chart（Scrumworks）

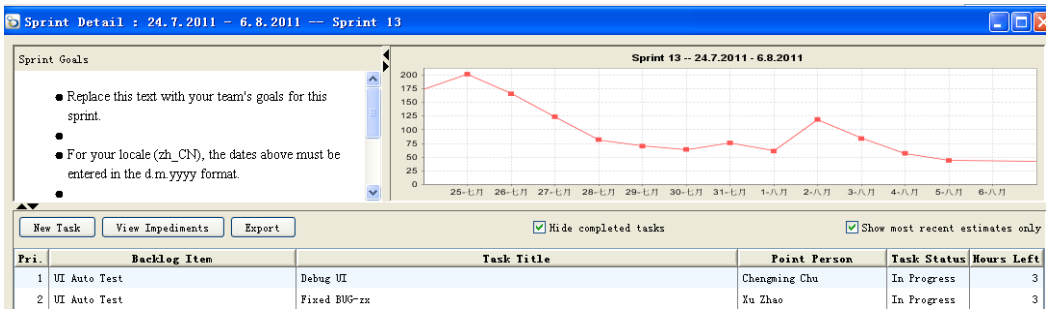


图 38-4 Sprint Burndown Chart（Scrumworks）

示例二：Taskboard（Rally Kanban）

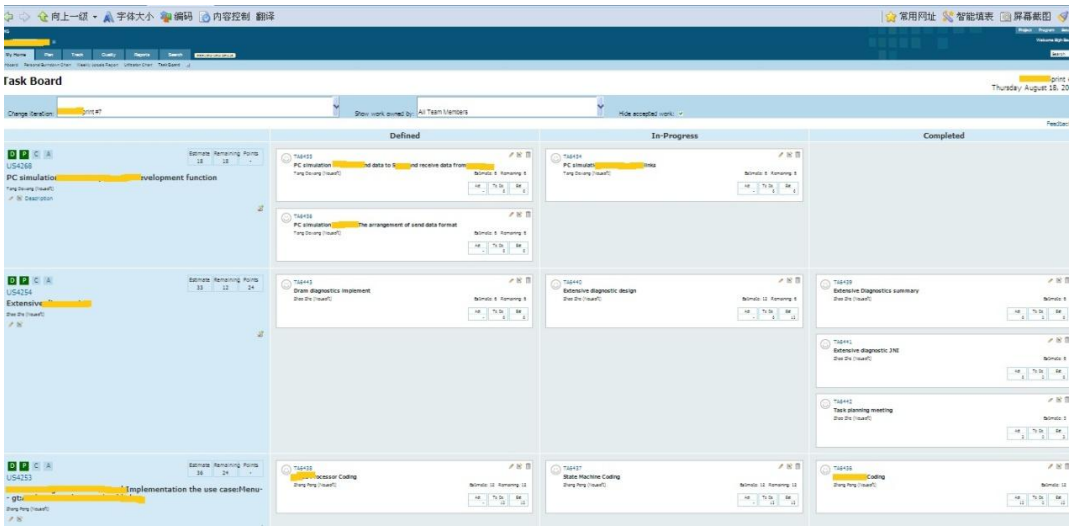


图 38-5 Taskboard（Rally Kanban）

示例三：人员 workload 分析（Rally）

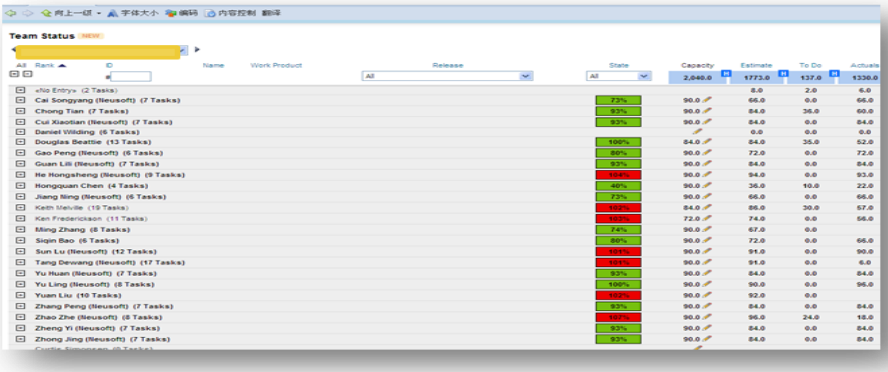


图 38-6 人员 workload 分析（Rally）

示例四：目标达成率分析(数据来自 Scrumworks)

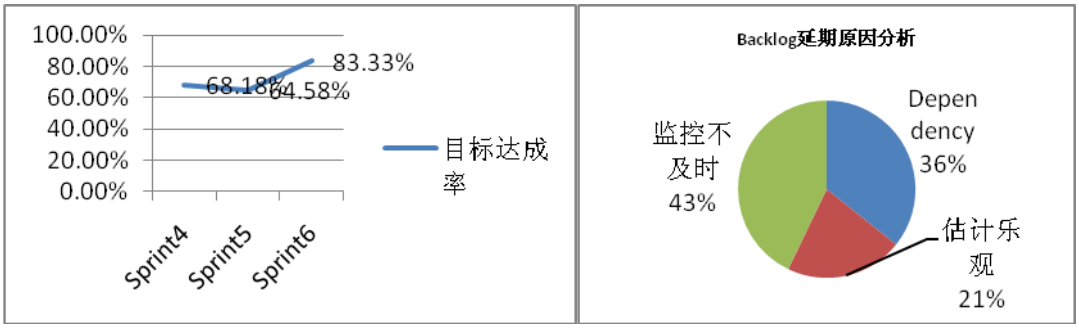


图 38-6 目标达成率分析

专家点评：

有效地利用工具却是可以一定程度上提升团队的工作效率，本案例提供了一些实际开发过程中的有益经验。从 Scrum 的本质来说，使每个团队成员全身心投入项目的计划，实施和管理，培养自我组织和自我激励的团队精神，是建立高效团队的核心基础。

点评专家：陈志波

“具备灵活的需求管理功能；支持分布式开发场景；能够和 Scrum 本身（至少三个工件）完美融合；能支持白板展示功能。”，如何通过较全面的工具支撑让 Scrum 开发模式落地生根，东软做了非常有益的尝试并积累了实际经验，再让 Scrum 融会贯通到开发过程的同时，使开发进程与持续提升的收益变得可视化了，值得任何实践 Scrum 的开发团队借鉴学习。

点评专家：李春林

案例三十九：测试管理（IBM）

1. 案例说明

1) 案例来源

IBM

2) 项目基本信息

在大型敏捷项目、新产品的开发中，在矩阵式的组织架构下，更加要求测试团队需要有效融入敏捷团队，及时并充分做好功能测试、压力测试，回归测试等等。

项目基本信息如下：

- 团队规模：40 人
- 环境：Java, Web2.0, C, DB2

3) 组织结构

Scrum of Scrum 组织架构

4) 面临的问题

迭代增量式开发使得测试点随之增加，代码的重构和需求变更带来测试需求的不断变化。 新技术的运用却将传统的测试方法和工具遥遥抛到身后，测试人员要有快速的学习和创新能力。设计、开发、文档都需在一个迭代周期内完成，各个环节的拖延给都给测试带来压力。

5) 解决方法

Scrum 团队中，测试人员不仅仅用测试技术、工具、执行测试来保障产品质量，也被要求参与设计完整性和正确性的讨论、scrum 迭代任务里程碑及时间先后的安排，并被要求领导和组织 scrum 团队迭代演示。测试人员要求和开发人员工作的更加紧密，参与关键部分代码级检查，并组织完整团队检查测试用例和数据。并行的大部分测试与开发节奏和心跳保持一致，并允许特殊测试跨度多个迭代。合理规划自动化测试投入，达到令人接受的 ROI。

6) 主要收益

- 并行测试赋予测试人员新使命，提高测试绩效 100%；
- 并行测试用更少的代价保障了迭代的正确性和准确性，测试在需求设计阶段介入，因而在代码之前就找到需求的问题，这部分缺陷的数量占总缺陷数量的 20%，大大减少了项目的投入；
- 并行测试团队拥抱变化，能快速响应，有效降低了项目风险。

2. 定义和特性说明

敏捷开发的特点是高度迭代，有周期性，并且不断接受客户的反馈。敏捷测试即是保障每个周期的产品质量，确认客户的有效需求在指定时间内得以完成。测试管理需有效的保障测试活动的以下几个特点：

1) 快速；

2) 有效控制测试投入；

3) 测试工作项和开发工作项一并体现在团队的计划中；

4) 既验证正确性，也能保障准确性；

5) 测试不多不少，不早不晚。

3. 如何应用成功实践说明

1) 敏捷测试的重点

首先，尽早的开始测试，开始系统测试，不可等待到功能完全做好才开始。了解计划中的待实现的功能，了解其权重分配，设计系统测试和功能测试用例。测试执行的一开始可以是针对部分功能的，之后可以逐步扩展。接着开始采用迭代的过程完成测试任务，即将测试任务划分为多个周期，一开始可以做些关键的功能性测试，可以对代码中的可复用部分（组件，构件）做完整的安全测试，性能测试，压力测试，并发测试，全球化测试等。接着的周期可以做边缘的功能测试和其他测试，最后的周期应该用于回归测试，和关键的性能和稳定性测试。

然后策略性的进行自动化测试，设计并开发可以用于日后 regression 和 acceptance 的自动化脚本，持续维护与开发。

自动化测试为团队带来的是长期效益，自动化测试的开发也应该首先选择部分测试对象，例如，API，框架等比较稳定和关键的功能做功能测试的自动化；对产品的性能指标，压力测试也要较早的制定自动化测试的计划和准备脚本。另，选择成熟、易用、有可靠技术支持服务，并经历了市场考验的测试工具。

最后，要做静态测试，做好需求分析，做好设计的分析。

Tester 要更多的思考需求的整体实现，将自身作为第一用户积极参与项目和系统的需求分析，设计和开发。积极地参与前期工作，并迅速反馈给设计和开发测试结果。

2) 敏捷测试计划

以四周一个迭代为例，第一周的工作是静态测试，确定测试策略和制定可行的测试计划，划定测试范围。这个阶段的前提是敏捷团队确定了当下迭代周期内需要完成的 backlog 项目。

第二周的工作是准备开始测试的执行，因此要准备好测试用例和测试环境。期间团队达成测试用例共识。

第三周的中间第一个可以执行的 Build 已经能够发布了（这个前提需要 developer 的密切配合）我们开始功能测试了。

第四周我们要结束测试，包括 constructive 的测试部分和 investigative 的测试。在 Build Review 的前一天团队要通力合作解决能够解决的问题，保障 backlog 的如期完成。

专家点评：

测试的早期介入，可以在早期避免缺陷。本实践介绍的测试策略很值得借鉴。从局部重点功能测试，可复用部分的测试，到安全测试、性能测试、压力测试等，以及之后的边缘的功能测试，随后的回归测试，和关键性能和稳定测试，直到最后策略性的自动化测试、以及编写用于日后回归和验收的自动化脚本。提供了一套很系统的测试策略，并且很好地与敏捷开发相适应。该团队强调培养测试人员的全局观、整体观，从整个系统的角度出发，安排测试任务，使测试工作更加高效。

本实践提出的敏捷测试计划，很好地与敏捷开发相呼应，对敏捷团队中的测试人员是个很好的借鉴。

点评专家：刘德意

案例四十：完整团队（用友软件）

1. 案例说明

1) 案例来源

用友软件股份有限公司

2) 项目基本信息

医疗基层卫生服务系统产品研发。

- 团队规模：20 人
- 环境：JAVA、用友 UAP

3) 组织结构

矩阵式组织架构，包括需求、UE/UI、开发、测试来自不同部门的人员组成产品团队，参照 Scrum 组织架构，产品项目经理为 Scrum Master

4) 面临的问题

团队成员对产品总体计划和目标不清楚，每个人只关心自己的任务和进度，一旦与之承接的任务发生问题，对整体进度影响很大。对于开发经理制定的开发计划，成员可能认为完不成却无法修改，导致延迟。

5) 解决方法

让团队所有人员参与到项目管理中，一同定制任务和计划，并在研发过程进行中，一同监督和保证任务进度。

6) 主要收益

开发计划定制更为合理，项目进度完全按照计划进行，不会延迟。所有团队成员共同监督承接任务进度，研发效率得到提升定义和特性说明。每个团队成员都是项目的管理者，在项目的全生命周期的各阶段中，都有所有项目人员的参与和决策。

2. 定义和特性说明

每个团队成员都是项目的管理者，在项目的全生命周期的各阶段中，都有所有项目人员的参与和决策。

3. 如何应用成功实践说明

1) 产品研发初期，产品经理制定一级需求目标时，所有团队成员参加并进行评审，让所有人了解整体产品需要完成的事情。

2) 二级开发计划制定及工作量评估时，所有团队成员参加，对于某一个任务采用投票形式，每人写下评估的任务工作量，最后取平均值，汇总后形成总计开发计划。

3) 三级开发计划(单个按钮级别功能点)进行任务分配时，所有团队成员参加，任务分配结合每个人的特长和意愿，与之共同协商指定（在标准 Scrum 框架中，任务都是由团队成员自我领取，但在实际实践中，由于团队成员水平差别及某些人员可能较为内向而无法真正的执行下去，所以我们采用的是领导指派和自己领取相结合的方式）。

4) 研发过程中，团队成员都对项目进度进行管理。对于前置任务，要去跟踪和催促进度，对于后续任务，自己的任务完成后要及时通知让其开始。

专家点评：

将每个团队成员是为项目的管理者，参与项目的目标设定和过程管理的流程，对于真正构建自我组织和自我激励的团队是非常重要的。本案例介绍了一个很好的激发团队成员参与项目制定规划的实施方案，并针对分级开发的不同情况进行区别的处理和对待，具有很好的可操作性。

点评专家：陈志波

显然，此团队在自管理、跨职能的团队建设方向上已“尝到了甜头”。这可能为我们众多担心“传统 PM 一旦放手内部管理，团队就将一塌糊涂”的组织，提供了鼓舞。“一级需求目标→二级项目计划→三级开发计划”，似乎是用友的独创，作为同行觉得概念新颖但链条清晰。在每个点上的具体实践，可操作性很强，值得参考。只是这个完整团队，没有看到另一个关键贡献者“PO”的身影。略感遗憾！

点评专家：刑雷

案例四十一：在 Sprint 内部管理需求变更（EA）

1. 案例说明

1) 案例来源

Electronic Arts 上海公司

2) 项目基本信息

B2B 电子商务平台开发， 社交网站开发，基于 IBM WebSphare 的内容管理平台，（美国）医疗保险申报/受理等多个应用开发项目。

项目基本信息如下：

- 团队规模：7-15 人
- 环境：Java，PHP + Drupal， IBM WebSphare Commerce Server + Java， IBM WebSphare Content Management + Java 等

3) 组织结构

标准 Scrum

4) 面临的问题

如何管理在一个 Sprint 内部客户提出的需求变更。

5) 解决方法

- 够用就好的 Planning；
- 尽可能缩短 Sprint 长度；
- 变更冻结；
- 重排优先级以及变更累加总量控制；
- 使用看板（Kanban）等不固定 backlog 的敏捷开发方法。

6) 主要收益

有序的 Sprint 内部变更管理。

2. 如何应用成功实践说明

理想情况下，一个 Scrum 团队应当受到充分的保护以确保在一个 Sprint 内部不发生需求变更。但是理想与现实总是相距甚远。我们还是经常会在一个 Sprint 中间应对需求变更这个难题。从过去 3 年运作 Scrum 中积累的经验，我们觉得以下做法在通常情况下是有效的：

1) JIT（Just In Time）Sprint 计划

Ken·Schwaber 在他的书 *Agile Project Management with Scrum* 中是这样描述 Sprint 计划的：

（对于一个 30 天的 Sprint 来说）Sprint 计划会议应当是一个 8 小时的时间盒。会议分为两个时长为 4 小时的部分。第一部分是从 Product Backlog 中选择用户故事；第二部分是准备 Sprint Backlog。

但是如果我们已知在一个 Sprint 中无论如何都会遭遇用户故事的变更，为什么我们还要花时间来计划那些将要变化的用户故事呢？我们应当只专注于那些优先级最高，并且确信在交付前不会发生变化的用户故事。将这些用户故事分解为任务并进行估算 — 这似乎更符合“应对变化胜于遵循计划”的敏捷宣言。

2) 尽可能缩短 Sprint 周期

一个 Sprint 通常为 2 至 3 周，在某些情况下这样的周期还是无法及时满足客户快速交付的需求。如果这是导致客户在 Sprint 内部变更需求的原因，那么我们或许应当缩短 Sprint 长度。我们并不需要频繁的变更 Sprint 长度，因为我们通常保持恒定的交付节奏来保证团队逐步形成稳定的 velocity，但是在团队内部可以把每周看成一个内部的子 Sprint。这样做可以帮助团队缩短响应客户要求的周期；实际上根据 20 / 80 原则，在第二周交付客户变更的用户故事已经足以满足大部分的客户了。

3) 变更冻结期

需求变更本身发生变化，这种情况也相当普遍。我们遭遇了好几次客户频繁变更需求的情况，在团队花费巨大努力将需求 A 变为 B 后，客户又要求将 B 变为 C。从客户的角度我们需要承认这也是合理的，因为在拿到可用的软件之前客户没有办法准确的知道他们到底需要什么。一个有用的解决方案是维护一个变更 backlog — 我们在一个清单中纪录所有变更，让客户排出优先级，并且说服客户在动工前等待若干天以确保所有的信息都已经得到确认，不会再有新的想法涌

入并导致对变更清单的变更，并且在同一时间只安排变更 backlog 中优先级最高的 3 项。一般情况下客户对于三至五天的变更冻结期还是可以接受的。

4) 不使用 Scrum

有些时候变更发生的过于频繁，使得我们甚至无法维持一个 Product Backlog。这种情况下我们需要认真考虑是否继续使用 Scrum。敏捷开发有很多种方法论，除了 Scrum 之外我们还有其他的选择，例如看板（Kanban）就是一个非常适用于“请求驱动（Ticket Driven）”型开发项目的方法论。

专家点评：

无论如何短的迭代周期，总是可能发生变更，本实践给出了两种处理策略：一种是继续采用 Scrum 方法进行管理，但是在一个周期内部以周为单位（子 Sprint）进行变更管理；另外一种是放弃使用 Scrum。从实际应用看，这是一种对 Scrum 的有效使用。另外一种缓解频繁变更的方式是结合快速开发工具、演示工具进一步引导客户稳定需求。

点评专家：钱岭

案例四十二：基于项目的在研产品开发（用友软件）

1. 案例说明

1) 案例来源

用友软件股份有限公司

2) 项目基本信息

在 R6.2 产品在研过程中，某医院正式上线物流模块，上线中期为确保阶段性验收，实施团队反馈回很多物流模块细节的应用需求和报表查询。

项目基本信息如下：

- 团队规模：6 人
- 环境：XP & VB

3) 组织结构

标准产品研发组织架构，包括需求、开发、测试、质量、实施；

基于项目的在研产品开发流程图及各个角色职责：

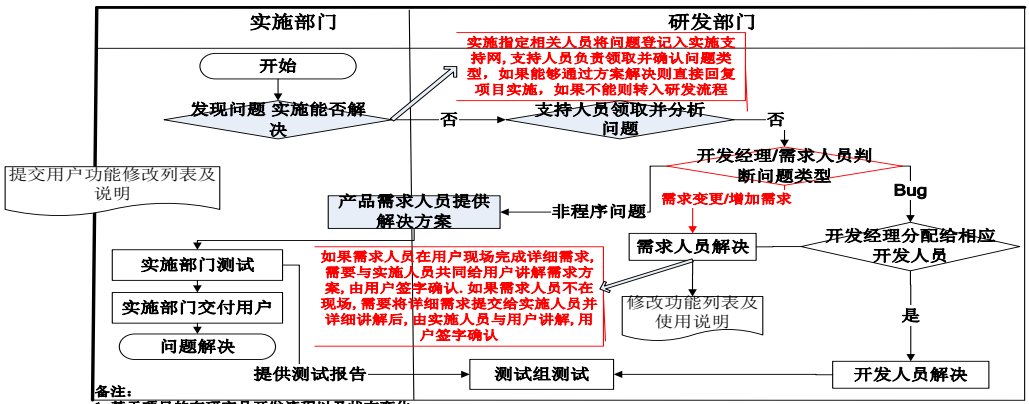


图 42-1 基于项目的在研产品开发流程图

4) 面临的问题

用户需求紧迫且繁杂

5) 主要收益

- 通过实施支持网确立项目的重点和关键以及所属阶段，提高团队绩效10%；
- 提前做到原型产品验证；
- 在研版本的样板用户需求可以有效管理；支持客户的快速反馈，提高客户满意度；
- 支持快速响应需求变更，使产品更完善；
- 缓解研发内测试的压力；
- 实施顾问提早见到产品，方便样板客户实施的最佳实践形成；最终提交的产品覆盖更多的用户需求。

2. 定义和特性说明

产品正在研发进程中,已有若干个样板项目针对该产品进入实施上线阶段,因此在推进样板项目过程中针对在研产品提出更贴近用户需求的问题反馈,目的更高效的支持项目同时完善产品,使产品能支持后续更多的项目,通过在研产品的实施支持网进行项目需求和问题的反馈\跟踪\发布\统计。

3. 如何应用成功实践说明

考虑到所有项目反馈的问题优先级别不同,并且项目上线的模块和侧重点不同,在研产品不可能在某一个迭代周期中解决所有项目的所有问题.通过项目实施支持网结合 UFSDP 平台应用,以项目反馈问题(或需求)驱动开发\需求\测试\支持\实施顾问多种角色,使问题在不同的阶段都有人负责,并明确问题在不同阶段的时间点和解决方案,最终达到问题的解决和产品的完善,使产品在后续发版中更贴近用户需求,同时更多应用于其他项目,逐渐在研产品和项目支持进入良性循环。

所有问题遵循相同的工作模式,这有助于团队把重点放在一个关键问题的开始、处理和结束时的具体活动。例如，一个问题（需求）在整个处理过程中执行的常见活动是：顾问提交问题；支持领取并分析问题；研发需求人员提供解决方

案；开发修改代码；测试验证问题；支持重复验证；顾问最终验证；顾问关闭问题。

应用实践案例

1) 实施支持网提交问题

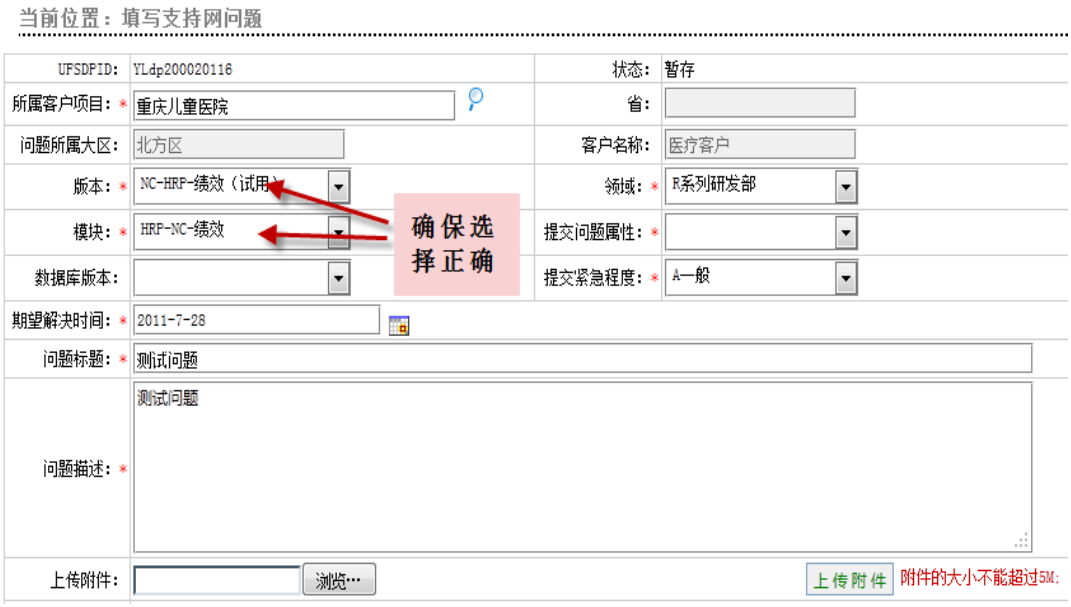


图 42-2 支持网提交问题图

2) 实施支持网查询问题

图 42-3 支持网查询

3) 跟踪在研产品的处理进度及结果

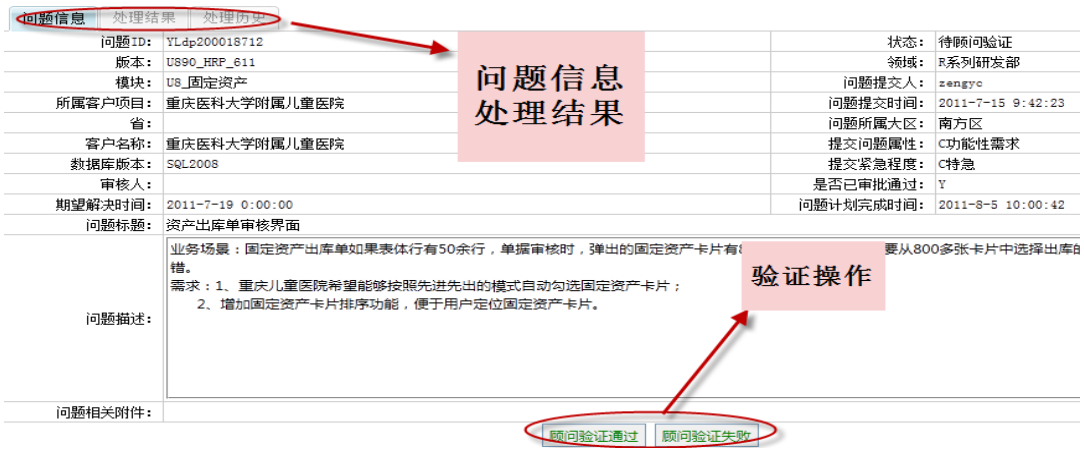


图 42-4 跟踪处理进度及结果

4) 基于项目的在研产品状态统计查询

图 42-5 统计查询

专家点评：

敏捷开发不仅仅是流程和人的协作，工具的支撑必不可少，从本实践中我们可以看到在一个产品支撑多个项目的复杂研发场景中，通过跟踪工具提高了研发生产率。

点评专家：钱岭

案例四十三：多级项目规划(用友软件)

1. 案例说明

1) 案例来源

用友软件股份有限公司

2) 项目基本信息

医疗基层卫生服务系统产品研发。

- 团队规模：20 人
- 环境：JAVA、用友 UAP

3) 组织结构

矩阵式组织架构，包括需求、UE/UI、开发、测试来自不同部门的人员组成产品团队，参照 Scrum 组织架构，产品项目经理为 Scrum Master。

4) 面临的问题

产品研发计划总是不能得到正确执行，任务分配可能会有遗漏或责任人不清。

5) 解决方法

采用多级项目规划，每到一个阶段，将计划和任务分配细化并调整。

6) 主要收益

产品研发进度与计划能够匹配，任务分配明确清晰。

2. 定义和特性说明

项目的计划和任务分配都是在不同阶段，分多级进行。包括：一级计划(业务场景级)；二级计划（菜单功能级）；三级计划（按钮功能或事务功能级）。在不同的阶段制定不同级别的计划，并且根据细化之后产生的变动进行随时调整。

3. 如何应用成功实践说明

1) 项目启动初期，由产品经理定制一级需求目标，描述产品要实现的业务场景或业务点，每一块完整的业务为一个独立内容。

2) 一级需求目标评审通过之后，由架构师或主设计师分解系统功能，给出二级开发计划，它描述对应的系统功能，并对每一个功能评估工作量，给出整体的开发计划，二级开发计划不指定每个任务的负责人。

3) 二级开发计划确定后，对其再进一步细化，明细到某一个菜单功能、按钮或后台算法。对每一个任务给出相对准确的工作量评估和完成时间（可以对二级开发计划的评估有调整），接着为每一个任务分配负责人员。

4) 完整产品研发流程与多级项目规划示意图：

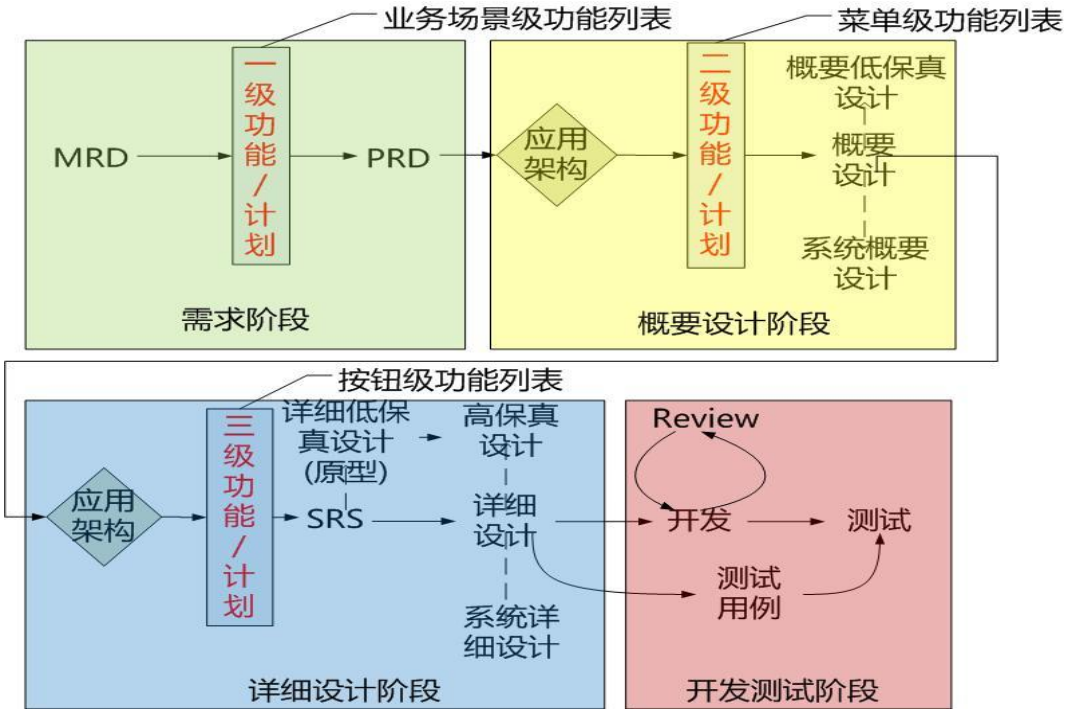


图 43-1 完整产品研发流程与多级项目规划示意图

专家点评：

如何在矩阵式组织构架中有效地实施敏捷开发是一个很有意义的问题。本案例的多级项目规划策略在每一个阶段，将计划和任务分配细化并调整。但是在分级之后，如何有效进行不同级层之间的沟通和管理，实现真正分级敏捷管理体系还需要更进一步探索和总结。

点评专家：陈志波

案例四十四：单元测试（用友软件）

1. 案例说明

1) 案例来源

用友软件股份有限公司

2) 项目基本信息

在开发 XX 银行资金交易系统的过程中，为了代码质量和压缩开发后期集成测试的周期，在开发人员开发代码的同时就进行单元测试，使代码质时不时提升，很多问题在集成测试之前就已经被解决，缩短了开发的周期。

项目基本信息如下：

- 团队规模：6 人
- 环境：NC5.6 平台 & ORACLE10G

3) 组织结构

标准 SCRUM 架构

4) 面临的问题

对于单元测试，程序员通常认为浪费了太多的时间，不愿意做单元测试。

5) 解决方法

在实践工作中，进行了完整计划的单元测试和编写实际的代码所花费的精力大致上是相同的。一旦完成了这些单元测试工作，很多 Bug 将被纠正，在确信他们手头拥有稳定可靠的部件的情况下，开发人员能够进行更高效的系统集成工作。这才是真实意义上的进步，所以说完整计划下的单元测试是对时间的更高效的利用。而调试人员的不受控和散漫的工作方式只会花费更多的时间而取得很少的好处。

6) 主要收益

- 程序中的每一项功能都是测试来验证它的正确性，我们也可以轻松的增

加功能或更改程序结构，而不用担心这个过程中会破坏重要的东西。而且它为代码的重构提供了保障。这样，我们就可以更自由的对程序进行改进；

- 编写单元测试将使我们从调用者观察、思考。特别是先写测试（test-first），迫使我们把程序设计成易于调用和可测试的，即迫使我们解除软件中的耦合；
- 单元测试是一种无价的文档，它是展示函数或类如何使用的最佳文档。这份文档是可编译、可运行的，并且它保持最新，永远与代码同步。

2. 定义和特性说明

单元测试是在软件开发过程中要进行的最低级别的测试活动，在单元测试活动中，软件的独立单元将在与程序的其他部分相隔离的情况下进行测试。单元测试不仅仅是作为无错编码一种辅助手段在一次性的开发过程中使用，单元测试必须是可重复的，无论是在软件修改，或是移植到新的运行环境的过程中。因此，所有的测试都必须在整个软件系统的生命周期中进行维护。

3. 如何应用成功实践说明

开发者编写的一小段代码，用于检验被测代码的一个很小的、很明确的功能是否正确。通常而言，一个单元测试是用于判断某个特定条件（或者场景）下某个特定函数的行为。

专家点评：

磨刀不费砍柴工，前期的充分准备，既帮助开发人员换种角度、跳出提高了代码的质量，同时又降低了缺陷在项目后期才被发现的高修复成本。用友医疗能够深刻认识单元测试的好处，并愿意“花费”时间去认真执行（在实践工作中，进行了完整计划的单元测试和编写实际的代码所花费的精力大致上是相同的），难能可贵。

点评专家 :赵静

案例四十五：代码规范（用友软件）

1. 案例说明

1) 案例来源

用友软件股份有限公司

2) 项目基本信息

在用友 NC 供应链产品 6.0 开发中在业务开发框架基础上，制定代码规范。

项目基本信息如下：

- 团队规模：74 人
- 环境：Java

3) 组织结构

需求 8 人；程序员 50 人；测试 18 人。

4) 面临的问题

团队规模快速扩张，新人比例高，新版本需要重写旧版本所有代码（旧版本规模在 5 百万行代码左右）。

5) 解决方法

在实践工作中，进行了完整计划的单元测试和编写实际的代码所花费的精力大致上是相同的。一旦完成了这些单元测试工作，很多 Bug 将被纠正，在确信他们手头拥有稳定可靠的部件的情况下，开发人员能够进行更高效的系统集成工作。这才是真实意义上的进步，所以说完整计划下的单元测试是对时间的更高效的利用。而调试人员的不受控和散漫的工作方式只会花费更多的时间而取得很少的好处。

逐步完善业务开发框架对可以进行标准化的代码，提供代码片断作为代码规范。提供完善的文档供员工培训。

主要活动：新功能代码收入开发框架、标准代码评审、文档编写、培训、代码讲评。

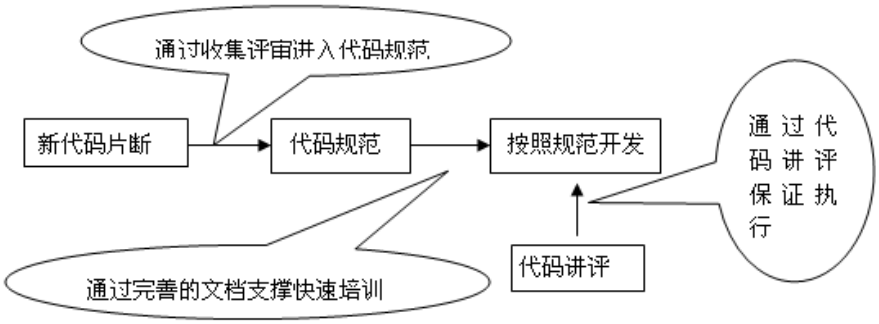


图 45-1 完善业务开发框架

6) 主要收益

新人培训时间短（1 到 2 周）即可发挥做用；标准化覆盖范围的代码功能推进迅速；标准化覆盖范围的代码质量高；维护成本低；不同领域的程序员可以互相维护这部分代码。

2. 定义和特性说明

对代码编写处理一类问题给出的标准样例（分两类一类是功能实现样板，一类是对易出错代码的正确样例）。

3. 如何应用成功实践说明

功能实现样板类的代码规范,需要在统一开发框架代码可标准化的基础上制定。适用于大量进行业务产品开发的团队.业务规则以外的代码可以逐步总结出标准样例.程序员按照标准代码开发产品对开发资源在开发过程中互换,和后期维护的成本降低都有好处.另外一类易出错代码规范是避免程序员范错误,是开发经验的积累,让程序员少走弯路.代码规范的执行,可以通过代码讲评和工具检查来实现。

专家点评：

代码规范是对“代码共同拥有”这个 XP 核心实践的关键支持。用友 NC 的例子很好地向我们诠释了二者之间的关系；代码范本也同时成为了高效的培训资料，令我这个同行有些“喜出望外”，有些等不及要仿效了。作为读者，还有一点希望了解的信息：重写旧代码与编写新代码两种场景，在实施时是否有所差异？

点评专家：刑雷

案例四十六：自动构建（石化盈科）

1. 案例说明

1) 案例来源

石化盈科信息技术有限责任公司

2) 项目基本信息

在离线油品调合软件开发项目中使用 Scrum 模型。

项目基本信息如下：

- 团队规模：9 人
- 环境：vs2010 & sqlserver2008

3) 组织结构

Scrum 组织结构。1 名 Product Ower,1 名 Scrum Master 及 7 名团队开发/测试人员。

4) 面临的问题

项目进度紧，核心模块和其他模块耦合性较大。需要保证签入代码的质量。随时提供可运行的程序版本。

5) 解决方法

- 首先将代码主线和开发分离开。然后要求核心模块和关键功能需要编写自动测试脚本；
- 在代码签入方面加入了的签入验证；
- 在每天凌晨 2 点执行主线的每日构建。

6) 主要收益

- 签入验证保证了签入代码的质量；
- 每日构建保证了主线代码的稳定性和可用性；
- 自动化测试脚本在编译后自动执行，极大的提升了测试的效率；

■ 项目组可以随时提供可运行的软件产品。

2. 定义和特性说明

自动构建就是在特定条件下，如每天、每次签入代码等，在服务器端利用编译工具自动编译代码，生成解决方案的过程。有时候还会在编译完成后，加入自动部署，运行测试脚本的步骤。通过编译是否能成功，编译出的程序是否能通过测试脚本的验证，来保证源代码的稳定性和可用性。

3. 如何应用成功实践说明

首先需要有规范的版本管理，包括分支管理。将产品代码的主线和开发分支分离。其次，需要明确签入操作的条件：功能编写完成后签入代码，如对验证有要求，需要同时签入该功能的验证脚本。

实践一：签入验证

在每次开发签入代码时，由服务器自动编译、测试（可选）。成功后才准许签入。失败则拒绝签入操作。

图 46-1 使用 VSTS2010 进行代码签入验证

实践二：每日构建

针对系统主线代码，采取每日构建。也就是每天固定时刻（一般是当天晚上或次日凌晨），通过由服务器自动编译、测试（可选）。并将执行的结果反馈给干系人。负责人需要在每天早上确认前一天构建结果是否成功。当发现构建失败时，需要第一时间协调解决。

这样可以保证系统主线代码的稳定性和可用性。

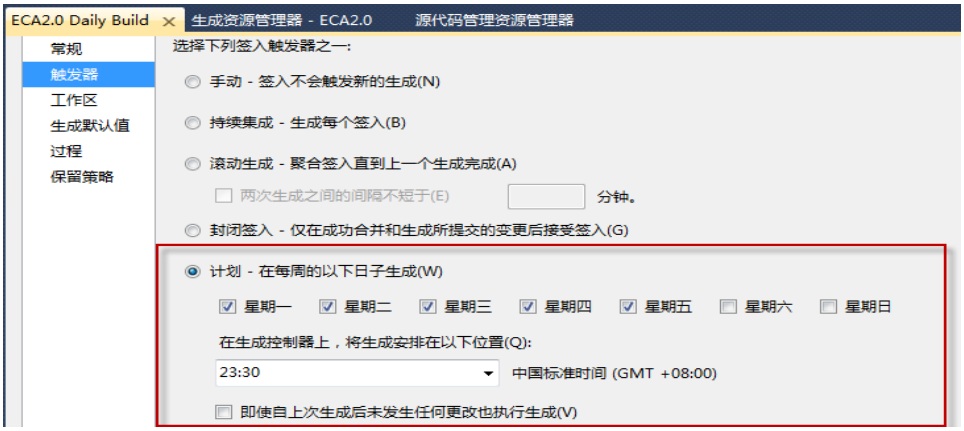


图 46-2 使用 VSTS2010 进行每日构建（1）

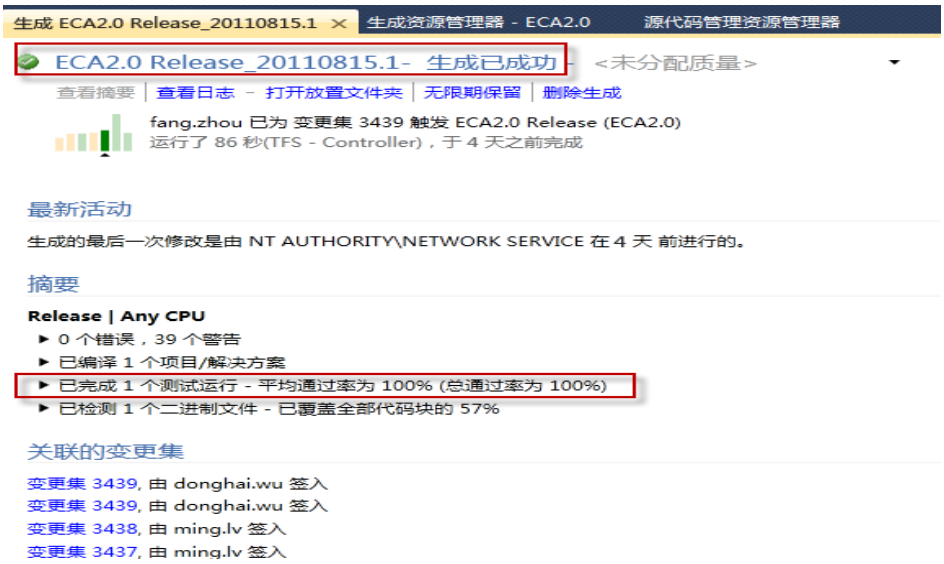


图 47-3 使用 VSTS2010 进行每日构建（2）

专家点评：

每日构建是很多开发团队有效推进进度、保障质量而追求的目标之一，能做到每日构建、并及时将所发现构建问题已经很不容易了，更能在构建时、甚至是签入时将自动化测试脚本结合起来进行测试级验证，就更上了一层楼。能够看出，这个实践背后一定有很多有力的管理方法支撑，很值得大家交流学习。

点评专家：张忠

案例四十七：Agile Estimating 2.0（EA）

1. 案例说明

1) 案例来源

Electronic Arts 上海公司

2) 项目基本信息

B2B 电子商务平台开发， 社交网站开发，基于 IBM WebSphare 的内容管理平台，（美国）医疗保险申报/受理等多个应用开发项目，数据挖掘与报表（ETL／Reporting）项目等

项目基本信息如下：

- 团队规模：7-25 人不等；
- 环境：Java，PHP + Drupal， IBM WebSphare Commerce Server + Java， IBM WebSphare Content Management + Java， ETL Cognos 等。

3) 组织结构

标准 Scrum 或 Scrum of Scrum 组织结构

4) 面临的问题

- Sprint Planning 会议耗时太长；
- 3 周的 Sprint 平均需要 3 小时以上时间完成 Planning。

5) 解决方法

采用 Agile Estimating 2.0 取代 Planning Poker 进行估算，缩短会议时间。

6) 主要收益

3 周的 Sprint Planning 会议时间缩减到 1.5 小时以内。

2. 定义和特性说明

Agile Estimating 2.0 是 Brad Swanson 和 Björn Jensen 在上海 Scrum Gathering（2010/4/19）上提出的一种新的敏捷估算技术，与 Planning Poker 同源但是可以显著提高估算效率。

3. 如何应用成功实践说明

Agile Estimating 2.0

计划扑克(Planning Poker)作为一个被广泛接受的估算技术，被大部分的 Scrum 团队所使用。计划扑克“将专家意见，类比，和分解这三种常用的估算方法捆绑到一个令人愉悦的估算过程中，来较快速的产生可信的估算”。” (Agile Estimating and Planning, Mike Cohn).

但是，我们发现有时候计划扑克玩起来耗费比较多的时间，尤其是在规模较大的团队中。Ken Schwaber 在他的书《Agile Project Management with Scrum》中指出，在进行 Spring 规划时，Spring Planning 环节应该控制为一个 4 小时的时间盒，但是从战术的角度看，如果一个会议持续 4 小时，我相信大部分的参会者会有精疲力竭的感觉，并且很难保持持久的注意力。这时很难再说计划扑克是一个令人愉快的游戏。

为了解决这个问题，我们尝试了不同的办法。我们首先确保团队作了足够的准备工作以减少会议时间；我们让每个成员在会前阅读需求来缩短需求澄清花的时间；我们也设定了合理的会议规则来确保讨论（有时是辩论）处于有序状态，但是会议依然花费很长的时间。设想一下，当 backlog 里放着 15 个用户故事，并且有 10 个组员坐在一起，这种情况下 10 人团队经常需要花费 5 分钟来结束一轮讨论以达成一致，何况为了对一个用户故事做一个估算的决定，经常会需要 2-3 轮讨论来解决不同的问题。因此仅计划扑克环节就经常耗费 2 小时以上。

这个状况困扰了我们许久，直到 Brad Swanson 和 Björn Jensen 在上海 Scrum Gathering （2010/4/19）上介绍了 Agile Estimating 2.0 技术。这个新的估算技术同样基于专家意见，类比和解聚，同样适用 Fibonacci 数列，但是它可以显著的缩短会议时间。

第一步是由 PO 向团队介绍每一个用户故事，确保所有需求相关的问题都在做估算前得到解决。

然后整个团队一起参与这个游戏。只有一个简单的游戏规则 – 一次仅由一个人将一个用户故事卡放在白板的合适位置上：规模小的故事在左，大的在右，一样大的竖向排成一列。整个团队轮流移动故事卡，直到整个团队都认同白板上故事卡的排序为止。

第三步，团队将故事点（Story Point）分配给每个用户故事（列）。在我们团队，我们更倾向使用投票来决定每个用户故事分配到哪一个 Fibonacci 数字。

还有最后一步 – 使用不同颜色来区分影响估算大小的不同方面，并且重新考虑是否需要修改估算值。例如，我们使用红色来表示那些无法被自动化测试脚本覆盖的用户故事，因此那些用户故事需要一个更大的数字来容纳手工回归测试的代价。

在多次实践 Agile Estimating 2.0 之后，我们对于结果还是相当嗨皮的。团队对于估算的准确性更有信心了，并且只耗费原先的 1/2 时间。请阅读以下资料，或许你也会想在团队中尝试一下。

http://properosolutions.com/wp-content/uploads/2010/05/agile_estimation_2.0-for-pdf.pdf

专家点评：

同样汇集了团队成员各方面的智慧，同样利用了 Fibonacci 序列与估计在某些方面存在的相似性，同样是一个有趣的估计过程，Agile Estimating 2.0 版本较 Planning Poker 在花费时间上有着一定的优势，对于那些使用 Planning Poker 估计 Story 遇到困难的团队，值得尝试。

点评专家：郜晓梅

案例四十八：协作应用生命周期管理(IBM)

1. 案例说明

1) 案例来源

IBM

2) 项目基本信息

软件协作平台产品开发项目中使用迭代式软件开发，项目基本信息如下：

- 团队规模：200+人，分布在 7 个地点
- 环境：Windows/Linux/AIX Java

3) 组织结构

- 二级团队:PMC & component teams
- 三层计划 (Release, Iteration, Daily)
- 在 PMC 和 component teams 采用 Scrum
- 一个大项目经理管理整个项目群
- 每个人都是技术人员

4) 面临的问题

- 发布周期短、市场需求不断变化、团队生产力不足；
- 团队分布在全球不同的 7 个地点，如何解决跨地域的团队协作。

5) 解决方法

采用协作应用生命周期管理方法，基于 Rational Team Concert 协作平台，全面实现软件开发过程的自动化、全生命周期的追踪和团队透明的协作开发（自动化报告能力），内置 Web2.0 的协作能力，包括 Wiki,IM,Feed 等。

6) 主要收益

- 通过团队透明协作能力的建立，在过去 3 年内提高团队生产力+15%；
- 软件按时发布率提升+13%；

- 缺陷减少 +10%。

2. 定义和特性说明

ALM（Application Lifecycle Management）是指软件开发从需求分析开始，历经项目规划、需求管理、开发过程的工作项管理和配置管理、测试管理等阶段，直至最终被交付或发布的全过程管理。主要特点包括：

1) 全生命周期的追踪；

2) 过程自动化；

3) 项目状态透明可见；

4) 团队协作。

3. 如何应用成功实践说明

实现协作应用生命周期管理是实现规模化敏捷的前提和基础，它帮助实现：

1) 过程自动化：固化并自动化贯穿活动的流程。

2) 可追踪能力：管理活动使用或生成的工件之间的关系。

3) 整体报告能力：从项目整体上报告开发工作的进展情况。

在大规模敏捷团队中，缺乏协作生命周期管理能力，团队就无法有效协作、无法实现透明管理；团队缺乏透明管理，就无法建立相互信任的自组织、自管理的机制，更无法实现遵守标准化的组织级治理流程和团队执行灵活性、效率的要求。

协作的、整合的应用生命周期管理平台是打造应用生命周期管理能力的第一步；可度量的、自动化的绩效管理平台是建立应用生命周期管理的第二步。通过 ALM 能力的实现和自动化手段，可以最大程度的提升透明的协作带来的团队效率和质量，帮助团队快速诊断错误，提供客户的快速反馈。

在协作应用生命周期管理实现过程中，要强化完整团队最佳实践和团队协作创新理念，通过透明的治理建立相互信任的团队氛围，从而打造自组织、自管理的完整团队。

专家点评：

高效的应用生命周期管理是实现规模化敏捷的重要保障，本案例中基于 Rational Team Concert 协作平台来实现协作应用生命周期的管理是一个很好的实践，但如何实现则需考虑更多的细节。

点评专家：李春林

案例四十九：故事点估算与故事点燃尽图（EA）

1. 案例说明

1) 案例来源

Electronic Arts 上海公司

2) 项目基本信息

企业应用程序开发 (Customized Ent App) + ETL + Reporting

项目基本信息如下：

- 团队规模：23 人, 3 个 Scrum 团队
- 环境：.NET, ETL (Informatica), Reporting (Cognos)

3) 组织结构

Scrum of Scrum

4) 面临的问题

基于 Ideal Hours 的燃尽图不能准确体现真实状态

5) 解决方法

- 使用故事点估算；
- 为每个故事定义清晰的完成标准(Definition Of Done)；
- 使用基于故事点的燃尽图。

6) 主要收益

准确，清晰的进度度量。

2. 定义和特性说明

燃尽图（burn down chart）是在项目完成之前，对需要完成的工作的一种可视化表示。燃尽图有一个 Y 轴（工作）和 X 轴（时间）。理想情况下，该图表是

一个向下的曲线，随着剩余工作的完成，“烧尽”至零。燃尽图向项目组成员和企业主提供工作进展的一个公共视图。

3. 如何应用成功实践说明

在看这两幅从我们的 Scrum 管理工具中直接拿过来的图之前，先来介绍一些背景：

我们是怎么估算/跟踪的？

- 我们使用计划扑克来做 planning，用故事点对每个用户故事做估算；
- 我们还需要把用户故事划分为任务，并且客户指定必须对任务用小时做很细的估算。

我们使用客户提供的 Jira/GreenHopper 来记录/跟踪任务的小时估算，并得出一个小时的燃尽图。同时我们在团队工作区中，放一块白板用来手工画出用户故事的故事点燃尽图。

简而言之，我们同时使用 GreenHopper 和白板两套跟踪系统，分别跟踪小时和故事点的燃尽情况。

这也给我们提供了一个很好的机会来对比这两种估算方法的优劣。现在让我们来看看真实的图。

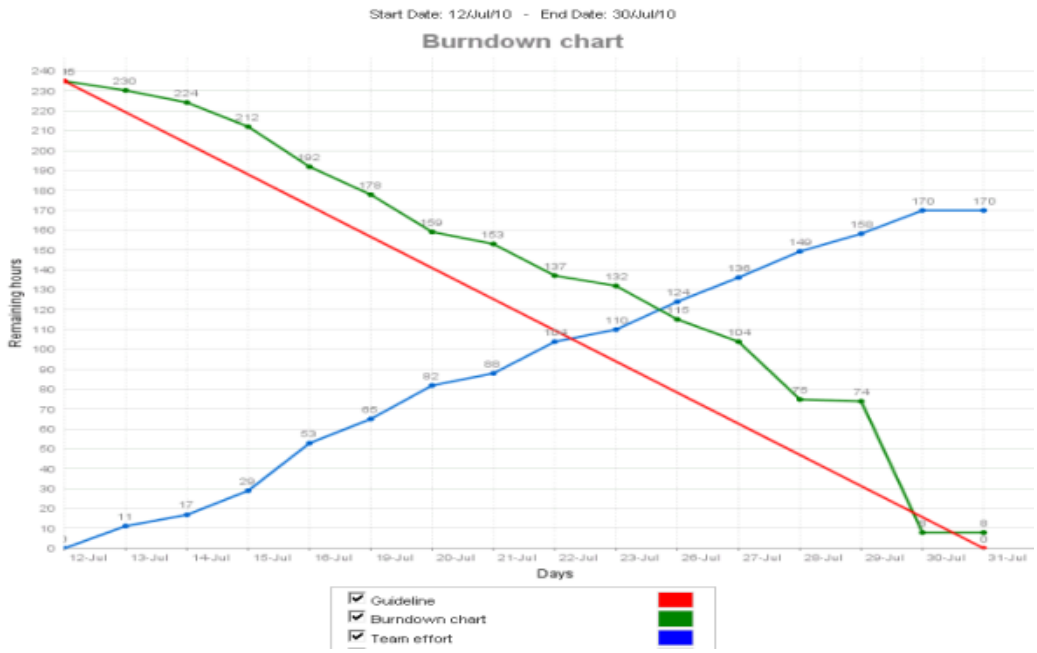


图 49-1 基于小时的燃尽图

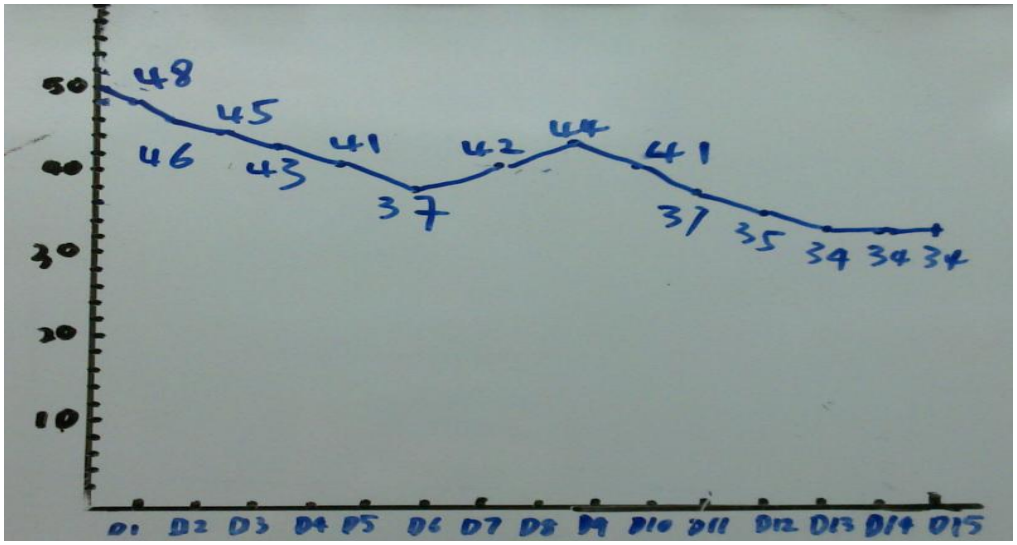


图 49-2 基于故事点的燃尽图

如果我们只看上面的 Jira/GreenHopper 自动产生的小时燃尽图，可能我们会觉得一切顺利，燃尽图曲线几乎跟 Scrum 教材上的一样，到 Sprint 的最后一天，总共计划的 240 小时只有 9 小时剩下。

但是如果我们再看一下故事点燃尽图，情况就完全不一样了。根据估算所有计划的用户故事总大小是 50 个故事点，到最后一天我们只交付了 16 个点 – 这绝对是一个干净利落的失败。

对于这种情况团队进行了分析。当我们使用 Jira 生成的小时燃尽图时，系统并不真正在意用户故事是不是真正完成了，它只是把团队记录在单个任务上的所有小时累加起来，并使用那个累加值代表“已完成工作”。Jira 在使用这样的公式：

团队在任务上所花的时间/所有计划的时间=已完成工作占所有工作的百分比

但是这个公式并不能告诉团队实际的情况，因为最终的交付与团队花的时间之间并没有关联。

- 一个团队/个人花 10 小时能完成的任务，另一个团队/个人或许需要 100 小时；
- 也有可能团队花了 100 小时但是什么都交付不出来；
- 作为一个用户故事，只有“完成”与“未完成”两种状态，不能使用一个百分比来表达“部分完成”，例如我们不能燃烧掉一个用户故事 1/3 的故事点来表达它已经完成 1/3 了；

团队的结论是：

- 基于小时的估算是不精确的，不同人在同一个任务上要花不同的小时数；
- 花了几个小时在一个任务上不等于这个任务有一部分被完成了，“部分完成”是一个隐藏问题的危险状态，应该避免使用；
- 故事点估算更简单和敏捷，虽然它很难使用。

如果 Google 一把你会发现有很多关于如何使用故事点估算的讨论，我们之前也有这个问题，并且现在还有团队成员在纠结于到底要不要坚持使用故事点，因为用起来着实不容易。但是这个 Sprint 之后，大家彻底意识到需要放弃小时估算，而坚持使用故事点，因为小时估算带给我们错误的感觉和错误的方向。

专家点评：

基于“小时估算”及基于“故事点”监控的差异，一直是一个具体而微妙的问题，令很多初涉敏捷的组织迷惑。（引用作者的用词，）这真是一个“干净利落”的案例。

能正确使用“故事点”来估算及后续监控，理应不会出现任何异议。但学界也仍然有诸如 Mike Cohn 等导师级人物仍认为 Ideal Hours 是可用的。我个人的理解，其思路与作者有一点关键不同：在燃尽图更新时，视角并不是记录“已消耗工时”，而是再次估算“剩余工时”。换个角度看，燃尽图的更新体现了一个不断估算的过程。再者，遵守“只有彻底完成的 Task 才能在燃尽图上获得分数”的规则，也可能有效的避免此问题。

个人体会:从 Burn down 的价值取向分析，（彻底）完成了多少、（到底）剩余多少，是重要的；既有的成本投入，不太重要。

点评专家：刑雷

案例五十 :测试驱动需求 - 新的软件过程 V 模型(EA)

1. 案例说明

1) 案例来源

Electronic Arts 上海公司

2) 项目基本信息

B2B 电子商务平台开发， 社交网站开发，基于 IBM WebSphare 的内容管理平台，（美国）医疗保险申报/受理等多个应用开发项目

项目基本信息如下：

- 团队规模：7—15 人
- 环境：Java，PHP + Drupal， IBM WebSphare Commerce Server + Java， IBM WebSphare Content Management + Java 等

3) 组织结构

标准 Scrum

4) 面临的问题

传统需求文档驱动开发无法良好的驱动 Scrum 模型。

5) 解决方法

使用 ATDD：编写接收测试用例取代需求文档；不同层次的测试用例驱动 TDD 开发活动，接收测试自动化。

6) 主要收益

精炼、有序的需求文档，敏捷的相应需求变更，需求与测试驱动开发无缝衔接。

2. 定义和特性说明

测试驱动开发（Test-driven development）是现代计算机软件开发方法的一种。

利用测试来驱动软件程序的设计和实现。测试驱动开始流行于 20 世纪 90 年代。测试驱动开发是极限编程中倡导的程序开发方法，方法主要是先写测试程序，然后再编码使其通过测试。

测试驱动开发的目的是取得快速反馈并使用“illustrate the main line”方法来构建程序。

测试驱动开发的比喻。开发可以从两个方面去看待：实现的功能和质量。测试驱动开发更像两顶帽子思考法的开发方式，先戴上实现功能的帽子，在测试的辅助下，快速实现正确的功能；再戴上重构的帽子，在测试的保护下，通过去除冗余和重复的代码，提高代码重用性，实现对质量的改进。可见测试在测试驱动开发中确实属于核心地位，贯穿了开发的始终。

ATDD（Acceptance Test Driven Development 接收测试驱动开发）已被包含入了 TDD 实践中。

3. 如何应用成功实践说明

我们对于软件过程的 V 模型都非常熟悉：

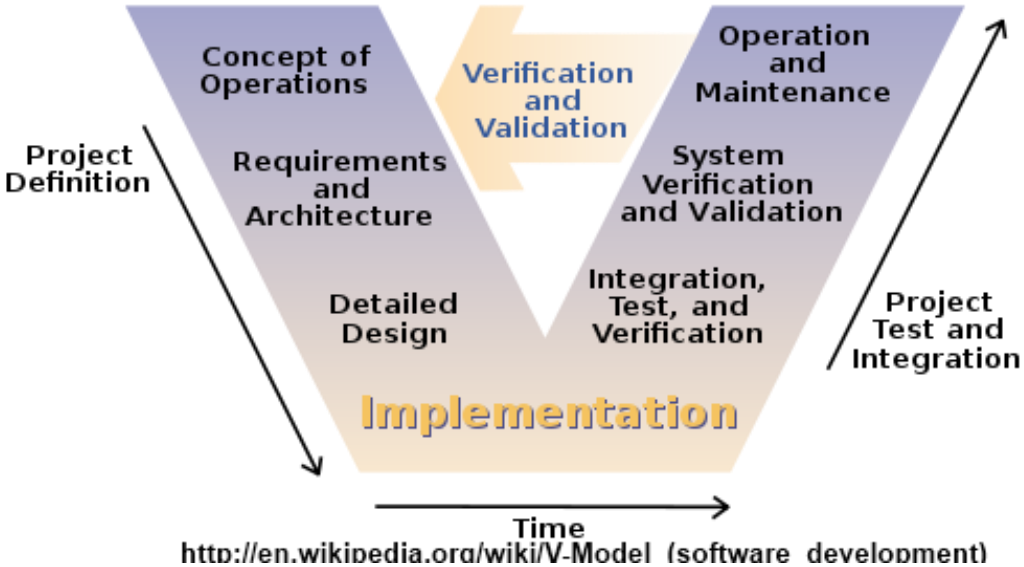


图 50-1 软件过程的 V 模型

即便使用敏捷的软件开发生命周期模型，有些时候我们依然会某种程度的参考传统的 V 模型来定义一些开发 / 测试活动。我们必须承认，很多精通 UML 和 RUP 的资深开发 / 测试工程师，对于使用传统的“交付物 / 文档”来定义项目产出

的方式已经驾轻就熟；对于一个 Scrum 团队来说，我们也经常会发现需求定义文档和测试用例被定义为必需的工件。它们往往被分别用来驱动对应开发和测试工作。

在若干不同项目中我见过如下形式的文档在 Scrum 团队中被采用：

工件	格式	用途
RUP SRS (Software Requirement Specification)，或其他需求文档	UML Use Cases + 自然语言描述	从最终用户的角度定义产品特性
功能性测试用例	测试步骤＋预期结果	从最终用户的角度定义软件行为
界面设计	页面模型、截图	为补充说明需求与测试用例提供补充

SRS 或需求文档通常会被用作开发和测试团队的唯一正式工件，来作为后续程序设计／实现以及测试用例设计／测试执行的输入条件。以下是在传统 V 模型下不同活动间的关系：

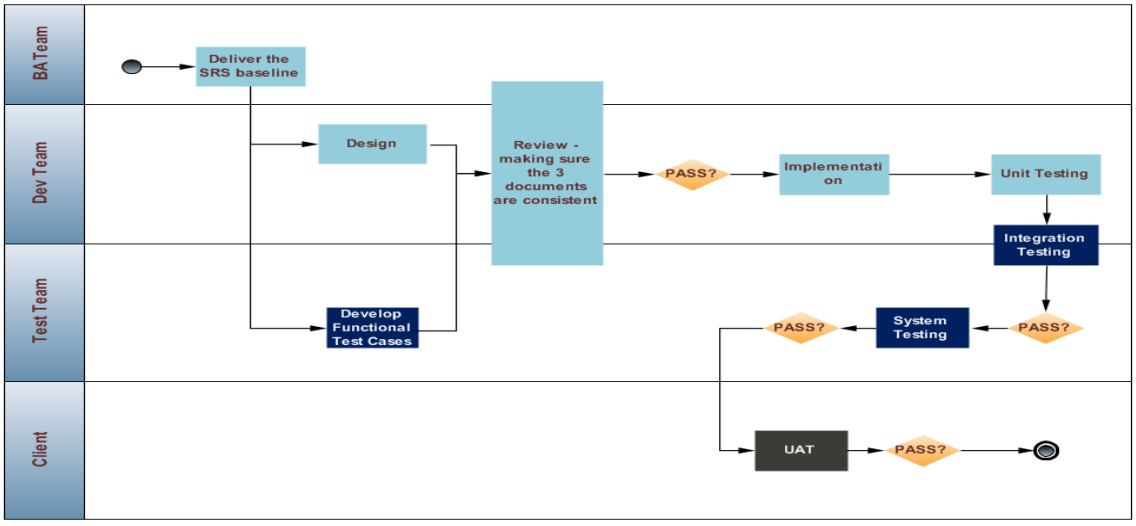


图 50-2 V 模型活动间关系图

很显然这个过程并不适用于 Scrum 团队，在一个 Sprint 里面所有的特性是按照 User Story 组织的，Scrum 团队不会按照技术层／开发步骤来制定开发任务，因此在 Sprint 中 Scrum 团队不会按步就班的执行这些活动。这也是 Scrum 过程与传统过程的一个典型区别。

在实践 Scrum 过程中我们经常困惑于如何将敏捷的方法应用到需求文档／测

试用例中去。团队花费比较大的精力来同步这两份重要的文档：SRS（需求规格说明书，或需求文档），和测试用例。

- 我们花费大量时间召开走读与评审会议来找出需求与测试用例中不匹配的地方；
- 即便两份文档保持一致，开发团队和测试团队也经常对同一需求有不同理解；
- 团队对于需求文档详细程度的依赖性越来越大 — 随着更多的问题被发现，团队对于需求文档详细程度的要求越来越高；
- 问题总在最后一分钟暴露 — 当执行系统测试并且发现了 bug 之后，团队开始争论现有的实现是否符合需求；
- 几乎不可能始终维护需求文档和测试用例文档，使它们一直保持一致。很多需求的细节保留在邮件，Wiki，口头的电话，缺陷跟踪系统或其他媒介，但是我们的测试用例则必需经常更新。在 3 到 4 个 Sprint 后，针对需求变更追踪测试用例的变更历史变得尤其困难。

我们开始思考如何应对这些问题，并决定尝试使用“测试用例驱动需求”。促成这个决定的主要原因是我们意识到，从开发 / 测试的角度来看，测试用例规定了软件每一个功能的具体行为，而这恰恰是最详细的需求规格定义。传统的需求文档更多的是从用户的角度提出对软件功能的需要，其粒度应保持在一定的高度以确保用户始终专注在商业价值上。

我们在 2 至 3 个项目中尝试了这一实践（某些项目是需求复杂度相当高的大型项目），经验表明测试用例驱动需求对于 Scrum 团队来讲确实可以简化文档工作，提高生产效率。

- 在一个项目中，我们依然保留了“用例（Use Case）”形式的需求文档，但是我们确保这一文档只记录高层次的需求，关注商业价值和最重要的业务逻辑；
- 我们为每个“用户故事（User Story）”定义一系列“用户接收测试用例（User Acceptance Test Case，UATC）”，用 UATC 来描述用户故事的具体行为；
- 将 UATC 细化成“功能性测试用例（Functional Test Case，FTC）”，UATC 与 FTC 之间是一对多的层次关系；

- 开发与测试一起修订 FTC。他们结对工作为 FTC 增加更多详细规格直到大家都觉得足以开工，两个团队工作在同一份文档基础上，因为 FTC 已经成为了记录详细需求的唯一文件；
- 需求讨论会议大大减少了。FTC 的描述方式（测试步骤，预期结果）无二意性的特点保证了所有对于需求细节的讨论都非常明确清晰：在这一测试步骤下，预期结果应当怎样。

实际上通过实践“测试用例驱动需求”，我们获得了一种新的，适用敏捷开发的软件过程 V 模型：

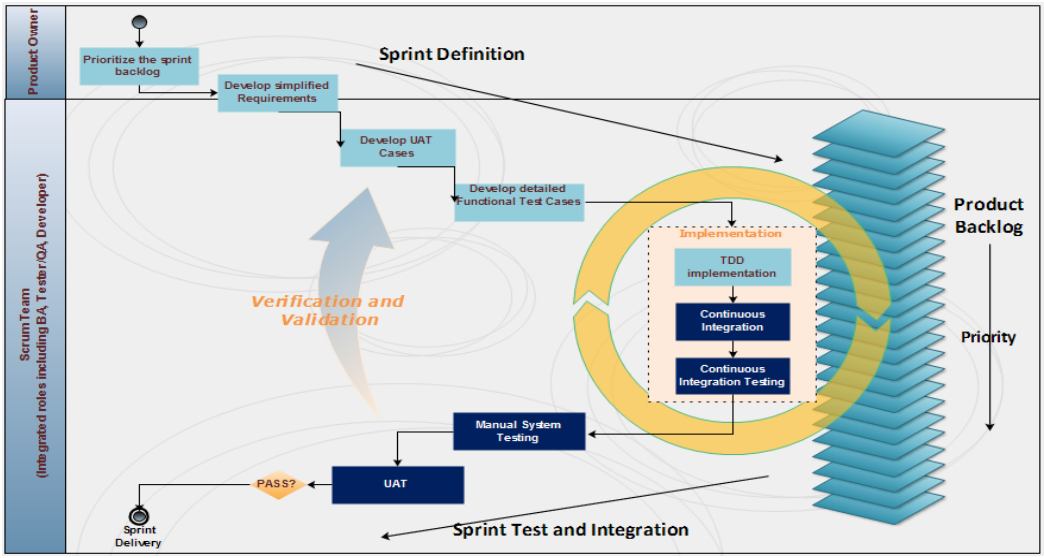


图 50-3 V 模型

一些最佳实践:

- “集成的开发 / 测试团队”对于成功应用这一模型非常重要。打破开发与测试间的功能性团队界限后，使用测试用例来定义需求的方法会更容易被接受，这份文档作为所有团队的唯一法定输入方能成为可能；
- 除了 FTC 外，我们依然可以使用其他格式的文档来描述需求，但是需要确保那些文档保持在较粗的粒度。

一个简化的用户故事 / UATC 模板：

Feature Description:

As a "Role", I'd like to perform "an operation", so that "business value"

Main flow:

Preconditions:			
Execution Steps		Expected Results	

Alternative flow 1:

图 50-4 UATC 模板

专家点评：

本案例很好的阐述了故事点的使用和“完成”状态的严格定义对于项目进展中如何用然进图更好地估计进度和规避风险。而且在 Scrum 的实施中，一个基本原则是要在每次审核时估计项目任务完成所需的剩余时间，而不是用预计时间减去已经花掉的时间。

点评专家：陈志波

ATDD 很好的解决的一些软件项目（特别是大型的企业级应用）在使用敏捷方法（Scrum）中遇到的困境，他们往往要求有完备的需求文档和测试用例，而 ATDD 就使我们在使用 Scrum 时不再担心“文档”的问题，既有指导开发的需求内容，又避免了维护和同步模式化文档的低效率。

点评专家：高航

这一最佳实践很透彻地描述出了很多开发/测试人员在维护不同阶段文档之间一致性的困扰。这种用测试用例代替需求文档的方法非常实用，应该能够大量减少维护不同文档所需的工作量。另一方面，测试用例的早期开发，起到了对需求变更的缓冲作用，减少由于需求变更带来的开发工作的返工，适应现在需求不断变更的众多项目。更重要的，开发人员和测试人员共同开发测试用例，能够使大家都很好地了解需求，并减少对需求理解的偏差。

点评专家：刘德意