前言

经过软件行业几十年的发展,软件系统变得越来越复杂,传统的软件工程理论使"软件危机"越来越严重。过长的开发周期、超出预算的开发成本、令人担忧的软件质量、频繁流动的开发人员、官僚的体系制度、迅速变化的市场环境等因素,让繁冗、笨重的软件开发过程越来越不能适应现实的需要,软件项目的失败率很高。敏捷开发就是在这种背景下应运而生的。敏捷(Agile)是一种关注价值、消除浪费、以人为核心、迭代、循序渐进的开发方法。

记得在 2002 年的时候,恰逢国内引进了一批 XP(eXtreme Programming,极限编程)敏捷开发的图书,网络上和杂志中也出现了一些早期的相关文章和报道,于是笔者有机会认识了敏捷开发,觉得耳目一新,也很震撼。可惜当时很多讲解敏捷开发方法论的书籍内容比较抽象,也很理论化。那时笔者正在读研究生,所以没有经历过敏捷实践,也就很难有深刻的体会。当时笔者就在想,敏捷开发真的有这么神奇吗?

笔者亲身经历过不同大小、不同类型的公司,也听过许多朋友和同学谈起自己的工作经历,可惜很多都是失败的教训,大家的抱怨大都集中在传统瀑布软件开发流程和一些具有中国特色的企业管理制度和文化上。笔者一直在关注敏捷,也总是在思考这样一些问题:采用敏捷开发的软件公司和软件团队是怎样工作的?不同性质、不同文化的软件公司和软件团队对个人成长的影响又是怎样的?

如今,几年的时间过去了,以 Serum 方法为代表的敏捷思想已经在全球范围内

推广开来。Scrum 一词来自英式橄榄球(Rugby)比赛。敏捷软件开发团队就好比一支橄榄球队:他们有明确的最高目标,而且每时每刻都朝着目标努力;他们熟悉最佳实践,高度自我管理,高度协作,高度灵活地面对各种挑战。大量的调查统计表明,敏捷开发确实大大提高了软件开发效率和软件质量,帮助软件企业提高了效益,并更利于个人的成长。

在现在这个 SOA 和 Web 2.0 当道的时代,国内也迫切需要敏捷思想,然而这些年似乎依然是雷声大、雨点小。国内实施敏捷的阻力主要在于人。因为敏捷的核心就是"以人为本",人的问题上升到了企业管理、企业价值观和文化的层面。片面地关注具体实践,而不去了解它背后的哲学思想,可想而知,是不会取得好结果的。所以说,敏捷决不是一个简单的软件过程。

最近两年,笔者有幸在 IBM 中国软件开发中心的一个 Scrum 敏捷开发团队担任 Scrum Master,期间发生了很多有趣的故事。由于起步比较早,特殊的机遇让我们有 机会给 IBM 的其他大大小小的开发团队进行敏捷培训,分享我们成功的经验和失败 的教训。每次培训时,我们都会被同事们对敏捷的浓厚兴趣深深感染,也发现很多问题和困惑非常有代表性,于是就萌发了写作一本有关敏捷开发的图书的想法。

有一天,同事们在一起突发奇想——为什么不把我们的故事改编成小说,以小说的形式写一本书呢?大家说干就干。我们的开发小组在业余时间就变成了本书的创作小组。本书以敏捷思想为核心,以 Serum 为重点,结合笔者所在的开发小组在 IBM 两年的 Serum 项目实施经验,参考了大量资料,将近百个案例、问题、知识点溶入到我们的故事中。

本书讲述了某外企的一个新团队如何从零开始实施敏捷,经历挫折、失败、进步、成长,直到项目成功结束的故事。

为什么不直接写敏捷的最佳实践,而要写那么多曲折的经历呢?我们认为,这就像解题一样——了解、分析问题的过程比直接知道答案更有趣,也更有用。更何况其实并没有真正的所谓"最佳实践"。在实际工作中,软件开发团队、软件项目往往千差万别,书中讲到的实践不一定都是正确的和都尽善尽美的,它们仅仅代表一种可能的敏捷开发实践。

本书的创作完全是由 4 位作者共同完成的,整个写作过程也是敏捷的:迭代、

自我管理的团队、有条不紊的进度节奏、期间收集潜在读者的反馈继而调整本书的 内容。我们惊喜地发现:敏捷思想真的有效,而且不仅仅是对软件开发。

本书的优势也许就是和大部头的经典著作相比更有趣味性,和纯正的小说相比更具有知识性。本书的定位是介绍 Scrum 敏捷开发的入门书籍。如果您想了解什么是敏捷开发和 Scrum,如果您对软件工程、软件开发流程有诸多困惑,如果您正准备采用敏捷开发但又缺乏实践经验,如果您想了解一些外企的工作模式和企业文化,如果您对自己的职业生涯感到迷茫……希望您能通过这本书得到一些帮助。如果这本轻松、有趣的敏捷开发故事书能在您忙碌的工作和生活中引发一点思考,带来一些价值,就是我们最大的欣慰了。

本书的完成首先要感谢 IBM 中国开发中心及各位同事的指导、支持和帮助。特别感谢所有提前审阅过本书的朋友,他们是:彭雷、岳治宇、汤宇松、钟朝晖、陈昊、窦文敏、唐威锋、郑曙旻,陈川、魏永超、马冀。他们中有的是 IBM 中国开发中心经验丰富的同事,有的是来自其他著名 IT 公司资深的软件工程师和项目经理,他们都对本书提出了大量的宝贵意见。我们同样要感谢家人对我们在工作之外花费时间写作本书给予的理解和支持。本书能够出版还有赖于 IBM 中国开发中心的高级经理闫小兵和电子工业出版社博文视点公司编辑潘昕的辛勤工作。

由于时间和能力有限,最后呈现给读者的内容依然有不少的遗憾。我们欢迎您 任何形式的反馈,以促进我们不断改进——这也是敏捷所倡导的。

本书的故事场景、情节、人物纯属虚构,如有雷同,纯属巧合。本书观点仅代表笔者的个人观点,不代表 IBM 公司。

好了,我们的敏捷故事即将开始。

读

本书是一本敏捷开发的故事书,穿插人物的对话、博客、电子邮件,在轻松、 有趣的故事中生动、形象、场景鲜明地讲解了敏捷开发的众多知识点,并对一些敏 捷开发的思想进行了总结和思考。

为了便于读者阅读,本书特设置了多种图标,用以区分各种不同的内容,具体 如下。



电子邮件



主人公的博客文章和留言回复



Scrum 博士扫盲,讲解 Scrum 的有关知识



资料库,介绍与敏捷开发相关的知识



总结和思考



开发任务



主人公在敏捷大会上的演讲及相关内容

目 录

1 重获新生 000

痛苦的挣扎 002

全新的开始 005

找工作 006

面试 E 公司 ① ① ⑨

焦急的等待 ①①③

E 公司报到 ①①④

2 回首往事 000

回忆 X 公司 0000

欢迎午宴聊开发 000

来自经理的帮助 028

- ⑩③② 敏捷开发培训——什么是敏捷开发?
- ◎④① 敏捷动员大会
- ⑩④④ 招兵买马——打造敏捷开发团队
- ① ④ ⑦ Hello , 温哥华
- ◎④⑨ 初遇 Scrum——XP、RUP 和 Scrum 的比较
- ①⑥③ 产品 Backlog 的制定
- ①⑥⑧ Scrum 管理工具

000 4 Sprint 1

---激动人心的 Scrum 之旅第一站

- ①⑦⑥ Sprint 1 计划
- ①⑦⑥ 投掷"照明弹"
- ◎⑧◎ 需求分析——买土豆的故事
- ⑩⑧④ 初尝每日 Scrum 会议
- ① 9 ① Scrum 之旅, 启程
- ⑩⑨① 改进的每日 Scrum 会议
- ◎ ⑨ ④ 跟踪每天的进度
- ① ⑨ ⑧ 每日 Scrum 会议的真正内涵
- ① @ ② 初尝 Scrum 有感
- ① ② ④ 来自大老板的质疑
- ① ① ⑦ Sprint 1 发慌的收尾
- ① @ 9 Sprint 1 评审——意外的结局
- ①①⑤ Sprint 1 回顾——批评和自我批评

5 Sprint 2——计划与变化 **00**9

棘手的新任务 ①②①

Wiki 上的需求文档 ①②①

没有不可能完成的任务 ①②④

Sprint 2 的计划会议 ①②⑦

Sprint 计划会议是属于整个团队的 ①②⑦

休假问题 ①②⑨

技术问题 ①③①

用扑克牌游戏来做估计 ①③②

任务状态要随时更新 ①④①

得到帮助 ①④②

单元测试 ①④④

Sprint 2 的第二周 ① ④ ⑦

用 Scrum 不能教条 ① ④ ⑦

经理安排的临时任务 ①⑤①

遇到困难 ①⑤⑤

Code Review 的重要性 ①⑤⑦

给经理的项目状态报告 ①⑤⑧

最后一个 Story 可能完不成了 ①⑥①

- ①⑥④ Sprint 2 的第四周
- ①⑥④ 冲刺=加班
- ①⑥⑥ 加班是个很严重的问题
- ①⑥⑧ 不是所有的 Story 都适合做 Sprint 演示
- ①⑦① 评审 Sprint 2
- ①⑦① 演示时不谈技术
- ①⑦② 评审总结和没有完成的 Story
- ① ⑦ ④ 回顾 Sprint 2

●●● 6 Sprint 3——深入 Scrum

- ①⑦⑦ Sprint 3, 出发
- ①⑦⑦ 为敏捷调整座位
- ① ⑦ ⑧ Sprint 3 的计划——精益思想
- ① ⑧ ① Sprint 3 的目标——口号的力量
- ①⑧① 失败的 Sprint 3 计划会议
- ① 8 3 重新计划 Sprint 3
- ①⑧⑥ 测试的介入
- ① 8 9 结对编程
- ① ⑧ ⑨ 先理论,后实践
- ① ② 超强的结对编程工具
- ① ⑨ ⑦ 饭馆中的敏捷思想
- ② ⑩ ⑩ 产品发布计划的变更

现场沟通 ② ② ②

出差到现场解决问题 ② ① ②

临时 Scrum Master ② ① ⑦

Sprint 3, 冲刺 ②①①

关于考核的问题 ②①①

突发事件 ②①⑤

不隐瞒,不放弃 ②①⑦

7 Sprint 4——最后的冲刺 ❷❷❶

延迟开始的 Sprint 4 ②②②

两天"休息"时间 ②②②

以假乱真的需求演示 ②②③

团队听诊工具 ②②④

部门重组 ②②⑦

新的敏捷管理协作工具 ②②⑧

敏捷测试 ②③②

为什么不报 Bug? ②③②

测试团队的新角色 ②③②

系统集成测试怎么办? ②③④

临时员工 ②③⑧

实习生 ②③⑧

CMM 与敏捷 ② ④ ①

现场客户 ②④⑤

- ② ④ ⑦ 小插曲: 装修也 Scrum
- ②⑤② 持续集成

❷❸③ 8 路在前方

- ②⑤⑨ Scrum 之旅大获成功
- ②⑤⑨ 产品顺利发布
- ②⑤⑨ 去敏捷技术大会演讲
- ②⑥③ 全球敏捷使用调查
- ②⑥⑥ 对敏捷的诠释
- ②⑥⑨ 附录 A 案例索引
- ②⑦④ 附录 B 相关概念
- ②⑦⑧ 附录 C 参考资料
- ②⑧① 附录 D 免责声明

名校计算机硕士毕业的关毅几年前加入一家大型民营 IT 企业,在项目开发中遭遇了种种不顺。他反思了自己的职业规划与现实之间的差距,决定换换环境。

经过好友推荐,关毅有机会去了著名的外企 E 公司进行面试。E 公司的面试官发现关毅正是他们想要找的人。这是一个绝好的机会,关毅将采用敏捷开发的思想领导一个全新的团队,他的职业生涯也会有一个全新的开始。



痛苦的挣扎

电脑上的时间显示着凌晨 2 点, X 公司的办公区已经漆黑一片, 但在角落里略显刺眼的台灯下还有一个身影, 正在查看目前的项目进度报表和软件 Bug 列表。这个人叫关毅。他站起身, 伸了个懒腰。再过几天就要给客户试用产品了, 还有这么多问题没有解决, 今晚注定又将是一个不眠夜。

关毅几年前从北京某名牌大学计算机专业毕业,获得了硕士学位。当时拥有一大堆令人羡慕的工作 offer,不过他最终出人意料地选择了 X 公司,让很多同学都不太理解。的确, X 公司是一家知名的民营公司,从事 ERP、OA 等产品的软件开发和软件系统集成。在众多 IT 公司中, X 公司开出的工资只能算是中等,不过,当年面试经理的一席话打动了他:"年轻人要做一番事业,我们这里提供的是没有天花板的平台,公司很快就会上市……"于是在那一年栀子花开的七月,他带着理想,拖着行李,搬进了租来的房子,到 X 公司报到,成为了一名程序员。

到首都北京去也是关毅从小的梦想。况且如今的北京也是信息产业之都,中关村、上地等地方更被人称为中国的硅谷。尽管 X 公司待遇比不上外企,不过机会可能会很多,这也让关毅对未来充满希望。

回想当初,关毅完全是凭着兴趣选择了计算机专业。他从小就痴迷于电脑,喜欢自己 DIY,还订阅了各种电脑技术期刊,钻研各种技术。为了积累经验,他在上大学期间还曾经为一些 IT 公司打工,时常通宵编程。当写程序真正成为自己的工作以后,他认为自己是幸福的。

一晃几年过去了。这几年间,关毅先是在一个系统集成的项目组,半年后升为项目经理,算是非常顺利的。

不过在随后的几年里,做项目的遭遇让关毅非常不爽——没有技术积累,没有成就感,而且非常辛苦。于是他转到了产品组,一直到现在。

目前,关毅是 X 公司的一名开发经理,带领 5 个兄弟一起做一个代号为"F4"的办公自动化平台已经有一年多的时间了,没想到产品组也没有想象中的那么好。

如今的关毅已经很多年不锻炼,看样子运动生涯也基本算是结束了。他已经记不清多久没和同学聚会了,记不清多久没给家里打电话了,也记不清多少次被女朋友埋怨了。工资涨得慢,但是脂肪肝却长得很快。这些关毅都不是很在乎,只是最近比较烦的是,女朋友家里"逼婚"了。其实他内心也想组建一个家庭,老这样飘着也不是个事儿,29岁了,好像身体可以透支的资本也越来越少。

关毅这样的劳模们,为了赶工期,不得不经常超负荷运转,加班甚至熬夜更是家常便饭。项目组的兄弟们也够义气,一起跟着关毅打拼。可是不拼能行吗?关毅只是开发经理,权限非常有限,但是老板们却凶得很,放出的话其实可以翻译成"不好好干,都给我滚蛋"。这年头什么都贵,就是大学毕业会编程的又便宜又好找,况且还有那么多毕业生找不到工作呢。但是关毅够义气,一直待兄弟们不薄,好吃好喝,活自己也一点不少干。最近兄弟们实在太累了,关毅不忍心,于是背着老大们,早早放他们回家了。

凌晨 2 点 30 分,关毅也挺不住了。他太累了,不一会就趴在办公桌上睡着了。睡梦里,他仿佛梦到了自己的将来。公司因为效益不好进行裁员,他年龄大,身体不好,工资又高,于是丢了工作。他躺在医院里,好像没有人来看他。以前记得媒体曾歌颂过一些劳模,做了很多贡献,累倒了,很多人会去慰问,而且还会被当作为先进的楷模学习,光宗耀祖啊。关毅在想,其实我比劳模都劳模,劳模能做到每天都熬到 2 点吗?唉,不过想想看,工作以来好像没做过多少真正有价值的事情。几年了,产品还没有真正开发完成,到现在也没有一个客户真正埋单,理由是没有提供足够多他们想要的功能,而且质量太差。关毅接着想:悲哀,悲哀啊,我居然没有给社会创造任何价值,仅仅在公司干活——仅仅是干活而已,然后每个月领那点薪水——况且那点薪水在北京,没法买房子,没法结婚,甚至连看病都看不起……关毅非常伤心,一切仿佛都太晚了,自己像脏水一样被泼掉了,没了工作,身体也垮了,还一事无成。怎么会混成这样!小时候自己可是年年考第一、月月戴小红花的主儿啊。

想着想着,关毅醒了。他朦朦胧胧地回想起刚才的梦境,有点后怕,也有点失落。好在现在是 29 岁,还没有老到梦里的那种地步。过去,关毅坚信付出就有回报。但事到如今,他有点怀疑了,也许选择比努力本身更重要。

回想这几年的工作,关毅并不是很满意,也许当时选择 X 公司就是一种错误。



不过只要尝试过了,关毅也就不会后悔。他尝试过想改变公司的现状,他从技术人员的角度清楚地明白开发流程肯定有问题,可是技术总监、产品经理乃至管理层却并不以为然。他也明白,目前并没有更好的解决办法,况且,他也没有这个权力——公司的文化就是以销售为导向的,靠做项目维持,对技术和技术人员都不是很重视,尤其是关毅他们这些处于公司最底层的真正干活的程序员。

来公司几年,看着人走了一拨接一拨,才发现原来程序员在管理层眼里和民工 没什么两样,只是招之即来,挥之即去的赚钱工具。关毅自己也一样,只不过他现 在是稍微高级一点的民工而已。虽然公司的项目和产品都不少,但如今市场竞争越 来越激烈,只好靠压低价格赢得订单。目前公司的状况很不好,很多项目都严重亏 损或处在亏损的边缘。当时面试时经理对他的承诺早就无影无踪。

关毅眼神中充满了迷茫,这种迷茫不仅仅是针对 \mathbf{X} 公司未来发展的迷茫,也是对自己前途的迷茫。

全新的开始

昨晚幸亏睡了一觉,不然上午又要请假回家了。

关毅洗了把脸,冲了杯浓浓的咖啡,又坐到了电脑前。这个时候,他的 MSN 闪出了对话窗口,原来是他的好朋友也是大学同学张大力。

张大力说:"关总,这周末有空么?"

关毅说:"张总,估计够呛了,最近项目逼得比较紧。"

大学时他们就喜欢这么总啊总的互相称呼。

张大力说:"就周六中午,咱们大学同学聚会,花不了关总您多少时间"。

关毅愣了好一会儿,才懒洋洋地敲键盘回复:"好吧,在哪儿?"

• • • • •

周六,关毅见到了张大力,还有很多毕业后就没见过面的本科同学。

张大力,当年计算机系的强人,大学期间打工无数,做过很多项目,大四没有考研,毕业以后几乎平均每两年跳槽一次,最近终于稳定下来了,据说去了著名的外企 E 公司,属于世界 500 强。

同学相见,格外亲切,大家互相述说着这几年各自或郁闷或精彩的生活。关毅 有一搭没一搭地听着,时不时地喝一口酒。

张大力就坐在关毅旁边,仿佛看出了他的心思,拍了拍他,说道:"多谢关总给面子过来,我看你气色不太好,最近忙啥呢?"

关毅摇了摇头,对张大力说:"不怎么样,项目做得不爽,正郁闷呢。"

张大力追问如何不爽,关毅就一五一十地把自己在 X 公司的近况告诉了张大力。

発がScrum之が 一教徒开发故事

张大力听后直摇头,说:"我们公司正在招人,你有没有兴趣?"。

关毅有点不解: "E公司?你知道我对外企在中国的研发中心没什么兴趣,技术含量不高,分工太细,我不想当螺丝钉。"

张大力笑道:"我敬爱的关总,你的认识还停留在 10 年前啊。如今外企的中国研发中心,尤其是像 E 公司这样的,已经不是你想象的那样了。因为质量好、成本低,所以 E 公司已经把越来越多的重要工作转移到中国。E 公司鼓励创新,而且正在内部大力推广敏捷开发,每个研发人员对产品的成功都是很重要的。另外,待遇也绝对有竞争力,我们毕竟是世界 500 强嘛,呵呵。"

关毅了解张大力, E公司能吸引他这样的强人肯定有自己的一套。

饭桌上笑声不断。

酒过三旬,关毅悄悄推了推张大力,说道:"我回去想一想。"

张大力半天才反应过来说:"还想啥想,快把简历给我,这次的推荐奖金我是拿定了!拿到之后咱俩平分,哈哈!"

关毅说:"会不会要求很高?"

张大力说:"以你的背景和能力,我想应该问题不大,不过英语要好。"

关毅暗想, 英语也曾是自己的强项, 只是需要尽快捡起来。

关毅举起杯,对张大力说:"大力,奖金拿到后请我吃饭就行。先告诉我是什么 职位?"

张大力喝了一大口啤酒,说:"一个新产品项目的 Team Leader……"

找工作

晚上,关毅回到了家中。他泡上一杯咖啡,坐在电脑前,打开尘封已久的简历。简历已经很久没有更新了。

他花了一个晚上的时间把这几年的经历加了上去,发给了张大力。

星期一的早上,关毅在项目总监的办公室门前踱步。他显得有些紧张,他不知道应该如何告诉产品经理李鸣他准备辞职的决定。实际上他已经想了很长时间了,张大力的推荐不过是个导火索而已。不管去不去得成 E 公司,他已经决定出去看一看,选择更好的机会。

关毅终于敲了敲门, 走了进去。

产品经理李鸣30多岁,平时让人感觉有些难以接近。

看到关毅进来,李鸣还以为他是来沟通项目上的问题,低头继续忙自己的。

关毅开门见山:"我想辞职,不过我保证把手上的工作交接完以后再走。"

听了这话,李鸣放下手里的工作,抬起头,显得有点惊讶。他高声问:"为什么? 准备去哪儿?"

关毅轻轻地回答:"不,不是别的公司已经要了我,只是我想换换环境。"

"没找好就更别急了。我想知道你为什么想走。公司这两年很不容易,但机会是 很多的,现在特别需要你这样的人才。你看这样好不好,你回去再考虑考虑,不要 一时冲动,是不是觉得薪水……"李鸣的语速明显加快了。

关毅摇了摇头:"钱不是最重要的问题,虽然我现在也很需要钱。主要是……"

关毅一迟疑,李鸣把话接了过来:"那好办,你好好做,机会还是很多的。其实 我们早就考虑要给你升职了,你不要着急。"

关毅叹了口气,说道:"我是觉得留在咱们公司没有什么前途。因为公司在管理上有很多问题,我是做技术管理的,经营战略上我不好说什么,但是就开发这一块,这几年我好几次提出的问题都没有被重视,我实在不愿意眼看着开发过程中这么多的不合理、不和谐,但却没有办法解决。公司大、层级太多,在研发决策上效率就很低。咱们公司大,我说不上话,一时也改变不了,这我都能理解,但是我能改变我自己,换个更好的环境总是可以的。"

発がScrum之が 一般接开发故事

接下来是一阵沉默。

李鸣点上了一支烟,不一会儿就在两人之间升腾起一股烟雾,又迅速弥漫开来。虽然公司的人员流动非常频繁,但李鸣还是觉得像关毅这样的人才走了实在可惜。李鸣阅历比较丰富,是 X 公司的元老之一。他有点郁闷, X 公司连关毅这样勤恳、优秀的人都留不住,真是意料之外,不过也在情理之中。X 公司啊,问题到底出在哪里?该反省了。

不过李鸣当然不会就这样放弃,想继续用感情留人。一番长谈后,关毅好像还 是没有回心转意,最后只好不欢而散。

一般的 IT 公司还有最后一招,就是靠高额的违约金、扣押户口甚至学历证书来避免过于频繁的人员流动,把人拴住。不过关毅是老员工,已经不受这种游戏规则约束了。

关毅走出李鸣的办公室时已经快到中午了。他长出了口气,有一点轻松,也有 一点愧疚。

一个星期后的一天上午9点半,因为前一天加班到很晚,关毅还在睡觉。突然, 一阵清脆的手机铃声把他吵醒了,是E公司的面试通知。

关毅一下子睡意全无,无比兴奋,立刻打电话给张大力,咨询面试时都需要准备些什么。

张大力告诉他,不同的经理风格可能会不一样,不过第一面一般都偏重技术,还有英语。

关毅知道那个职位是 Team Leader, 从职位描述上来看, 他觉得自己非常合适, 技术上也没什么问题, 毕竟做了几年的 Java 开发了, 上学的时候自己的基础也很扎实。想到这些, 关毅信心十足。

关毅最后问:"我是不是要穿西服打领带去面试?"

张大力笑道:"我天天穿牛仔裤上班,你觉得呢?"

面试E公司

上地信息技术产业园区地处北京海淀郊区,方圆3公里,号称中国的"硅谷"。这里坐落着无数大大小小的IT公司,E公司的研发中心也在这里。大部分在这里工作的上班族都乘坐城铁,早上火车一箱一箱地把人拉到这里,晚上再拉走。非工作时间,上地就像一个空城。

面试的时间约在上午9点半。

早8点半,关毅来到城铁站。正值上班高峰,站台上人满为患,大家都在等待下一班车。车到站的时候,关毅几乎是被众人推着踏上了开往上地的城铁。车厢里的拥挤程度超乎想象,如果有人吃了大蒜,那周围的人都会遭殃。车厢里的人大都表情麻木,有座位的幸运儿百分之五十都在睡觉,人人充满疲倦,很多人背着重重的笔记本电脑包。在这种"恶劣"的生存环境下,这群人也自嘲为"IT 民工"。

下了城铁, 关毅径直来到 E 公司的办公大楼。地方并不难找, 关毅比约定时间提前了一会儿。他明白大公司的时间观念很强, 于是他在办公大楼一层的茶餐厅休息了一会儿, 正好也缓和一下情绪。

9点29分,他准时拨通了面试主管的电话。电话里的声音非常亲切,关毅由紧张变得有点兴奋。很快,主管很快出来了,是个30岁左右的男人,穿条格衬衫和牛仔裤,看上去显得非常有精神。他先帮着关毅在前台做了登记,然后带着他去面试的会议室。

沿途关毅仔细观察着 E 公司的办公区域——气派的前台,漂亮的秘书美眉。离开前台,进入办公区前,会先经过一个走廊,走廊的墙上贴满了各种讲座的海报,走廊的尽头是健身房和茶水间。透过透明的玻璃隔断,关毅看见健身房里摆放着乒乓球台和跑步机,茶水间里有咖啡机、冰箱、微波炉。走廊的墙上还挂了一些壁画,大概是员工自己旅行拍摄的照片,有西藏、云南等地的风景照。穿过走廊,就是员工的工作区了,有的区域大家讨论得很热闹,有的区域则埋头干活的比较多。工位装修很上档次,空间也够大。

很快, 关毅随着面试主管走进了一间小会议。

面试主管请关毅坐下,自己则坐在了关毅的斜前方。在关毅的正前方还有一个 面试官,年龄稍大,看上去已经等了一会了。现场气氛有点紧张。双方简单寒暄了 一下后,面试正式开始了。

一直坐在会议室的面试官说:"请先简单介绍一下你自己。"

"我叫关毅,毕业两年了,在 X 公司担任开发经理,目前负责代号为 F4 的 OA 产品开发……"关毅说。

"哦,给我们介绍一下 F4 这个产品吧。"

关毅花了 10 分钟的时间,介绍了 F4 产品的来龙去脉,画了系统架构图,对一些重点部分,他还仔细解释了一番。针对面试主管提出的技术细节问题,关毅都回答得非常清晰。两位面试官频频点头。

"你是开发经理,现在花在技术上和管理上的时间比例大概是多少?"

"技术 70%, 管理 30%左右。"

"能不能告诉我们你这几年中最成功的经历?"

"我没有成功的经历。不过相对而言,毕业半年后升为项目经理,带领团队打拼,不管多么困难的技术,多么紧的工期,都能按时交付,我的团队成员都很信任我,转到产品组后我也很快升为开发经理。"

"能具体举一个你觉得比较成功的例子吗?"

"最困难的一次是给 B 集团开发一个人力资源模块。因为是新客户,公司很想搞好关系,所以销售给了很多不切实际的承诺,要求我们在半年之内完成。我们只有 4 个人,我带领大家克服了很多困难,甚至周末也基本上都在加班。因为我的团队成员都比较信任我,所以愿意跟着我一起干,最后还是按时交付了。"

"为什么大家会信任你?"

"我认为,首先我的技术好,能帮助大家解决很多问题。另外,我愿意为团队做事情,身为开发经理,我没有把自己当成管理者,而是把自己当成一个为大家服务

和解决问题的人。"关毅说。

"请谈谈你在工作中最有压力或感觉最不好的一次。"

"也是 B 集团的项目。虽然我们尽了最大的努力按时交付了系统,可是系统最 终没能使客户非常满意。后期大家疲于奔命,忙于应付解决各种各样的 Bug 和按照 客户的要求重新修改系统。而最后这个项目基本上没赚多少钱。"

"哦。你能不能分析一下是什么原因导致的?"

"从我作为程序员和开发经理的角度上来看,我认为问题之一在开发流程和管理上,当然也有合同管理的问题和其他一些问题。首先,销售没有尊重我们的意见,项目进度不切实际,客户那边的需求分析没有做得特别好,没有让客户积极参与进来。当然,需求一开始不可能做得特别准确,不过需求、设计、开发、测试应该是一环套一环的。我们公司的开发流程一直是比较复杂和严格的,采用的是经典的瀑布模型。这样,如果需求上错了,设计也就跟着错了,后面的开发就会错上加错,后果往往无法弥补。如果真的要改需求,由于已经差得太远,只能加班加点靠拼体力来补救,因为交付日期在那放着呢。结果是团队成员都很疲惫,士气下降,代码质量也在下降,形成恶性循环。最近两年公司为了接更多的项目,又在上 CMM,文档是做了不少,可还是没有解决根本问题,反而降低了开发效率。管理层只关心进度,关心短期利益。我知道这些问题一直都是存在的,但是凭我个人的力量是无法在短期内改变一个大公司的开发体系的。公司最大的问题是不重视开发人员,只追求眼前利润,忽视长远的战略发展。"

"假设你有公司的决策权,你将如何改变现状?"

"我之前一直都在思考这个问题。公司属于高科技公司,所以一定要重视每一位 开发人员,以人为本。项目我很久不做了,我说说产品研发。我会从开发流程的改 造着手,使软件开发更加灵活,更早地吸收客户的反馈,这样才能快速满足客户需 求。另外,太过复杂的流程很浪费资源,我会简化流程和文档,把重点放到客户那 边,让我们的产品帮助客户解决实际的问题,真正带来价值,而不是仅仅为了实现 自己的销售利润,因为那只能是短期利益。"

"你认为自己最大的长处和弱点分别是什么?这些长处和弱点对你在企业的工

作业绩会有什么样的影响?"

"我认为自己是一个非常理性和坚毅的人,我能从混乱中整理出头绪,解决复杂的问题,帮助企业更快地实现交给我的目标和任务,并且也能积极地影响周围的同事。缺点是我这个人有时候对自己有点苛刻,容忍不了自己做得不够好……"

接下来是技术面试和英语面试, 关毅都表现得很好。

"你还有什么问题要问我们吗?"一直坐在会议室的面试官问道。

"我想了解一下我应聘的职位具体是什么?"

"这是一个 Team Leader 的职位,将负责一个产品小组的研发工作,并和加拿大那边的工程师一起协作来完成产品的研发。这个产品叫'电子商务通',之前都是在加拿大开发的,北京这边没有研发团队。今年 E 公司决定投资组建电子商务通的北京研发中心,所以这个职位还将参与整个团队的组建。对于这个新团队和新团队的Leader,我们的期望还是比较高的。在开发流程上,我们也想尝试一种全新的模式……"

全新的模式?关毅心想,到底是什么全新的模式,这么神秘。

"谢谢你参加我们的面试,如果没有别的问题,今天我们就先到这吧,我们会尽快通知你结果。"

"谢谢你们!"

"谢谢!"

走出 E 公司的办公大楼,关毅又回头看了一眼,心想这里真够气派的,真希望我也能成为这里的一员,可以天天去门口的小饭馆吃饭。

回到家中,关毅立刻和张大力联系,把面试的经历讲了一遍。张大力告诉他: "不错啊关总,大概你已经有 60%的把握了。"

关毅忙问:"还有四成呢?"

张大力笑笑: "我说 60%你还真信啊!即使过了,还有第二轮面试,主要看你长

得够不够帅和人品怎么样。呵呵,开个玩笑。具体我也不知道,还是要看面试经理。 你放宽心吧。谋事在人,成事在天!"

一个星期后,关毅的收到了二面的通知。

负责二面的是一位和蔼的高级经理,主要问了一些非技术问题。关毅的沉稳和 自信,让那位高级经理非常满意。

焦急的等待

时间过得很快,一转眼二面过去一周了,还是没有动静。

关毅有些着急。他又拨通了张大力的电话。张大力在电话那头安慰道:"我当时进来的时候等了一个月呢,如果通过了,我们公司也还有无数的流程要走。"

关毅说:"我不急,不过行不行总得给个信儿啊。"

张大力说:"我可以帮你打听一下,不过不会那么快。你想啊,有那么多面试的,也许高手如云呢! Good Luck!"

挂了电话,关毅才明白原来自己是那么在乎 E 公司的 offer。除了通过一些"内线"打听消息,他也时常上网看看大家对 E 公司的评价。面试之前,关毅是在挑 E 公司。现在轮到 E 公司挑人了,能不能选上,他心里一点底都没有。

关毅尽量使自己忙碌起来,处理 X 公司的交接工作,同时也在寻找一些新的机会,可惜都不太看得上眼。怪不得经常有人拿找工作和找对象相比,还真有几分相似之处。

两个星期以后,关毅好像已经淡忘了 E 公司这回事。一天中午,不经意间,关毅的手机响了,是 E 公司的号码,一个甜美的声音问他什么时候可以去 E 公司办理人职手续。关毅拿着手机,足足有 5 秒钟没有说话,搞得电话那头的美眉还以为信号出了问题。

幸福来的就是这样的突然,有时候往往就是在一个人即将绝望的时候。



E公司报到

一个月后,经过入职培训,关毅要正式到 E 公司上班了。

报到当天一早,关毅跳上了去往上地的城铁。城铁里的人依然还是那么的多,不过这一次关毅的心情不一样了,他一点也不觉得烦躁,甚至觉得城铁里空气中的每一个分子都是那么的清新。

到了公司,关毅领了分配给自己的机器,是一台笔记本电脑和一台全新的高配置台式机。另外,E公司会为每一位新进公司的员工安排一名 Mentor (导师),关毅的 Mentor 就是面试他的经理,叫徐天。

徐天先是带着他把整个办公区转了个遍。每到一处,徐天都向关毅详细的介绍——这是某某产品开发组,那是某某测试组。同时,徐天还——介绍相关同事给关毅认识。有些产品关毅在进 E 公司之前就耳熟能详,只是想不到在中国也有研发部门。关毅的另外一个发现就是 E 公司所有人的办公区都是一样大的,即使是经理甚至高级经理,也都没有独立的办公室。

关毅在E公司的第一天就在兴奋与新鲜的氛围中度过了。

第二天早上,徐天把关毅叫到一间会议室,先是询问关毅第一天的工作情况,工作环境有没有搭建好,工作适应不适应等,然后认真地对关毅说:"今天是第二天,不过我们正式的工作就要开始了。你即将负责的是我们中国这边刚拿下来的全新项目,这个产品已经有10年的历史了,我们中国研发中心终于参与到这个产品的开发工作中,之前这边只有两位同事在负责产品的测试。等人招齐后,你负责的产品组将有4名成员,包括你在内。目前项目组只有你一个人,我先给你一些文档让你熟悉一下,接下来你的任务是培训、出差和招人,组建你的项目组。"

关毅心想,自己已经好久没参加过培训过了,感觉人都快被掏空了,是时候该 充充电了。于是他好奇的问:"是什么培训?"

徐天笑笑: "还记得面试的时候和你提起的新的开发模式吗?"

关毅点点头,他当然记得,神秘的新的开发模式。

"我说的新的开发模式就是敏捷开发,这次培训就是关于敏捷开发的。现在 E 公司刚刚开始在软件研发部门大力推广敏捷开发。关于敏捷,之前你了解的多吗?"徐天问。

关毅想起张大力在同学聚会的时候也特意提到过敏捷开发。敏捷开发到底有多神奇?

于是关毅回答:"了解不多。但我非常想知道什么是敏捷开发?"

徐天说道:"好的,好好去听吧,我希望你成为咱们部门的敏捷专家。"

那一晚,躺在床上,关毅兴奋得难以入睡。关毅觉得自己的职业生涯又有了一段全新的开始,他已经准备好要抡起袖子大干一场了。

总结和思考

世上充满无数的选择和努力,但对于成功,选择大于努力。敏捷开发 是一种选择。





回首往事

关毅在博客中回忆过去,讲述了在 X 公司软件项目开发过程中的一些遭遇。X 公司 F4 产品混乱的软件开发管理状况从侧面反映了传统软件开发流程所暴露出来的问题。

关毅迅速融入 E 公司 , 并和 E 公司的同事们进行了有关敏捷开发和软件工程的有趣的探讨。

回忆火公司

即将开始的新征程充满了未知,但这一切都让关毅充满了期待。

虽然关毅到 E 公司的时间很短,但从以前的 X 公司到如今的 E 公司,他还是发现了太多的不同:公司规模、管理流程、人际关系……

此时的关毅很想和朋友们交流心中的感受,写博客大概是最好的方式了。

关毅已经很久没有更新自己的博客了,以至于先前注册的空间都已经被清理掉了。关毅有些沮丧。不过,他转念一想,这未尝不是一件好事。一切从头开始,一切皆有可能。

关毅给自己新博客的第一篇文章取名叫做"回忆"。

回忆

本人生于 1980 年。按现在流行的"什么什么后"的说法, 好像不属于 70 后。可我觉得自己怎么也不像 80 后。但是,等 到我踏上了工作岗位才明白,我们这些人的处境是多么的尴尬:比 70 后干的 多,但是没人家资深;比 80 后资深,但是干的可比他们多了去了。没办法, 谁让咱们赶上了呢!我呢,几年前毕业后就加入了 X 公司,分配到一个系统 集成项目组,没过多久就当上了项目经理。我卖力打拼,也确实干得不错, 好几次挽救了项目。

但是我实在不愿意再做项目了,这样的项目经理太难当了。公司里的项目不少都是首先由销售签下了"丧权辱国"的不平等条约,然后他们就撤了,拿奖金去了,我们开发实施团队在得到了客户"正式"提供的几片纸的"需求书"之后,就得凭着自己的理解立即投入开发。每次等"木已成舟"之后,用户对很多地方都是拒绝接受的,然后就是艰难的沟通,做工作,修改,反复修改。开发人员累了,走了,又换新的。用户对系统用之无味,弃之可惜,后期收款遥遥无期。然后,我们和销售互相埋怨,扯皮,项目被分拆,公司



只收部分款,很多开发工作都被浪费了。

一年多以后我调到了公司最具战略意义的产品"F4"担任开发经理的工作。"F4"是个电子办公自动化产品,它对于 X 公司的意义就像核武器一样。在我去之前,就已经秘密开发一年多了,准备两年以后完成开发并正式对外发布。就在我以为我的职业生涯将从此一帆风顺,而 X 公司将成为我这辈子的职业归属时,谁曾想老天爷居然给我开了一个天大的玩笑:在 X 公司工作了三年以后,我离职了,带着无限的遗憾走了。后来我才明白,那三年炼狱般的生活,也是我人生的一笔宝贵财富,不过如果上天再给我一次重新选择的机会,我也许不会选择 X 公司。说句实话,在 F4 最后的日子里,我对自己的职业生涯充满了迷惘。

在 X 公司的日子里,我每天 7 点钟准时出门,8 点之前赶到公司。这里需要声明一下,X 公司是 IT 企业里少有的早上 8 点半上班下午 5 点下班的。可是在 F4 这个部门,8 点之后到,就算迟到,就要扣钱。这是我第一天到 F4 时,负责 F4 的李总亲自向我们强调的。也许有人会说,IT 公司不是弹性工作时间吗?怎么能这样啊?不要奇怪,在 F4 奇怪的事还多着呢!

F4 产品组在 X 公司里是个大组,采用传统的瀑布开发流程加上 CMM。组里百十来号人,需求、设计、编码、测试,人人各司其职。流程上是一环套一环,文档也是一堆一堆的。而实施上,F4 产品组采用的是高压政策,从上到下,每个月都用绩效考核,让每个人的弦都绷得紧紧的。在外人看来,X 公司在国内民营 IT 企业当中算是管理非常严谨和有效的。我身为 X 公司最低级的小领导——开发经理,主要的职责就是带领自己的团队按照设计文档完成 F4 产品整个生产制造模块的编码工作。我的上面就是 F4 研发部的部门经理,部门经理管着所有模块的研发工作。再往上就是产品总监李总了。这里提到的设计文档对我们而言可是有确定意义的,我们拿到时已经整整等了三个月,而那三个月能做的就是每天看看需求分析文档,熟悉生产制造的业务流程,顺便写点儿简单的代码,学习一下新的开发工具 XFF。XFF 是 F4 平台部门用了两年时间研究出来的,可以自动编码。好不好用我就先不说了,如果修改时一不留神忘了保存了,出了问题就只能一切重来,无形间就增加了一倍的工作量啊。

好不容易拿到了设计文档,我们当然是欣喜若狂啊。这三个月都快把我们憋出病了!可是,让我们没想到的是,设计文档居然在许多地方和需求分析文档有非常明显的不一致。这可真是难为我们了。我将问题反映给我的顶头上司——F4 研发部的部门经理,他总是一副很忙的样子,一句"你们先按设计文档做"就把我们打发了。为了能赶上主管 F4 项目李总监的时间表,我们几乎是从正式开始编码就每天加班到晚上 9 点,周六大家也都被迫加班赶进度。就这样,又整整辛苦了三个月,整个生产制造模块的雏形终于清晰了。我带着弟兄们刚刚庆祝完,回到公司得到的第一个通知就是:设计有重大改动,我们的工作有 80%需要返工,而且原计划不变,必须在一个月后完成编码任务,好让测试人员开始功能测试。这回我又去找部门经理要说法,而他则是一脸的无奈,说什么设计的事他也管不了,现在的情况只能辛苦我们了。我当时气得都说不出话了,回去一想,唉,都怪不容易的,忍了!后来的一个月,可真的是地狱式的了。我是除了睡觉,其余的时间都在公司里渡过,最后硬是按期完成了任务。其实最令我欣慰的是,兄弟们都很理解我,居然没人抱怨一句,现在想想都觉得是个奇迹。

F4的 Demo 版给上层领导演示过后,得到了领导的肯定。但是,得到的意见是这个版本还没有一些特别吸引眼球的东西,和现有的产品差距不大。就是这句没有"吸引眼球"的东西,让李总监把我们整个开发部门大骂了一顿,说整个开发工作没有创新、UI 界面设计做得很不到位、业务流程实现过于简单、系统不够稳定等。我一听到这些,首先就觉得奇怪了。业务流程简单和我们有什么关系?这都是已经设计好了的。系统不够稳定?那是平台部门的责任嘛,怎么能一股脑地都扣到我们头上呢?后来,我才从一些元老级的员工那里得知,在 F4,设计部门是最牛的部门,因为设计部门是由 F4 的老大李总监亲自带领的,这是老李的绝对嫡系,是绝对不允许任何人去挑战的。而平台部门的员工基本上都是在 X 公司呆了七八年的老员工,属于有名的"养老院",谁也不会闲着没事去找老人的麻烦吧?所以,最倒霉的就是开发部门了,因为我们的地位是最低的。

接下来发生的远比我预计的要糟糕,整个团队的工作效率只有原先的70%,代码质量更是惨不忍睹。后来,为了提高产品质量,上层领导想出了一个更绝的招儿,要求每个测试人员每天至少开出50个Bug才算完成任务。



这个规定仅仅实行了几天,我们每个人头上就都压了上百个 Bug 需要解决。 Bug 本身的质量不敢恭维,很小的功能缺陷可能就开了一个 Bug, Bug 的说 明就是一大段文字。结果,开发人员浪费了很多时间在阅读 Bug 的说明上, 也要花更多的时间鉴别哪些才是真正的 Bug。测试人员为了达到每天 50 个的 目标,可以说是不顾一切。但事实上,很多 Bug 其实是测试人员没有理解真 正的需求,由于使用不当而导致的。毫不夸张地说,当时整个 F4 基本上到了 失控的地步。

其实,造成这样的局面是太多因素的集合所致。需求和设计为了所谓的精益求精,在最后关头三天一小变,五天一大变。真不知道之前的几年时间他们都干嘛去了!还有就是开发平台了。为了能有自己的开发平台,X公司可真是煞费苦心,耗费了大量的人力物力,结果出来的XFF简直就是个玩具。而为提高质量新出台的所谓每天50个Bug的规定,也让开发和测试部门之间原本就比较脆弱的合作关系彻底破裂了,争吵处处可见。发展到最后,这两个部门就完全变成了敌对关系。可是,最令我佩服的地方就是F4的李总监居然能这么沉得住气,好像一切都跟他无关似的。

问题这么多,我无数次尝试改变,无数次遭受打击。等级森严的管理层次,瀑布式的开发流程,中国式的 IT 企业管理文化像"三座大山"一样挡在我的面前,让我无计可施。中国共产党找到了马列主义,救了中国。我也希望能有什么来救救 X 公司,救救 F4,可惜我目前还没有找到。F4 产品在我走之前还没有结束,不过我已经预见了它的失败。于是,我了断了自己在 X 公司的职业生涯。

欢迎午宴聊开发

新的公司、新的环境,从理论上讲不可避免地会让关毅这个新人感觉到局促、 紧张,甚至压抑,而关毅也对此做好了思想准备。可是,关毅入职后的第三天,欢 迎午宴(Welcome Lunch)就让他不得不重新审视他的新东家,这个业界的IT巨人。

这天一早,关毅用了比平常多一倍的时间来洗漱、"打扮"自己,因为今天他要参加一个特别的活动——他和新团队的欢迎午宴。在这个聚会上,徐天要正式把他介绍给整个团队。而关毅自己心里也很清楚,这是一场遭遇战,要想成功地融入这个团队,今天就一定要充分地表现自己,赢得伙计们的信任。想到这些,关毅不觉热血沸腾起来,就连每天早晨必吃的豆腐脑,今天都觉得好吃了许多。他三下五除二地解决了早餐,迫不及待地出发了。

到了公司后,关毅认真地按照徐天的安排进行文档的学习。时间很快就到了中午 12点。关毅下意识地看了一下时间,不自觉地抬头看了一下徐天的座位,没人。"经理还没有开完会啊?"关毅在心里嘀咕了一句。

"小关,是不是等急了?"徐天的声音从关毅身后响起。

关毅急忙站起来,转过身,"没有,没有着急!"关毅红着脸,显得有些不自然。

徐天微笑着说:"小关,昨天秘书已经通知你了吧。今天的午餐是咱们部门的 Welcome Lunch,算是欢迎你正式加入我们的队伍。走吧,大家都在等你呢!"

关毅整理了一下自己的桌面,随即对徐天点了点头,表示可以走了。

当和徐天一同走出办公楼,关毅深深地吸了一口气,可心中还是不能平静。 Welcome,新的旅途!

每个人都不愿翻开的一些令人不安的记忆。然而,事情的发展却往往不会以人们的意志为转移。有时候你会发现,事情总是会向你能想到的最坏的方向发展——来自墨菲定律。

维松Scrum之旅

——敏捷开发故事

当关毅走入公司附近小有名气的"野猪林"餐厅时,伙计们都已经入座,佳肴也已经摆上了餐桌,一副"万事俱备,只欠东风"的架势。不出意料,他一露面,所有人的注意力就都集中到了他这个新人身上。关毅心里稍微有些紧张,其他人也看出了他的局促。"来,坐在这吧,我叫马如龙,你可以叫我 Franco。我是 SAO Designer Team 的 Leader,我就坐在你后面的那个隔间里。"

关毅有些受宠若惊地坐了下来,他尽量自然地环视一周,和身边每一个人打了招呼。

"好了,人都到齐了,咱们开始吧!咱们今天这次 Welcome Lunch 的主题是欢迎关毅加入我们这个大家庭,大家也正好借此机会一起联络一下感情,放松放松。大家不要客气,吃好喝好!"徐天作为这次的聚餐的"主办方",当仁不让地成为兼职主持。

徐天话音刚落,大家就无所顾忌地吃喝、闲聊起来。关毅感觉轻松了很多,一切都来得那么自然,大家都很随便地和他开着玩笑,之前所有的紧张、担心一下子被抛到了九霄云外,此刻的他就如同沐浴在暖暖的春风中,别提有多么受用了。关毅那消失许久的幽默感终于又回来了,详细的自我介绍后,他便尽情地享受着这个时刻,和伙计们人无所不谈。不知道是不是由于受了这家川菜辣味的刺激,他有些忘乎所以,嗓门都比平时高了八度。

聊着聊着,又聊到了 IT 技术、开发流程的话题,突然有人问了一句:"关毅,你们以前 X 公司的开发流程是用的什么模式啊?你们用 Agile 了吗?"

"Agile? 哦, 敏捷开发, 没, 没有。" 关毅对这个问题没有准备, 听到 Agile 这个词也是愣了一下。

与此同时,关毅的心忽地紧了一下。那个声音继续道:"我就说嘛,真正的大型软件开发是用不得 Agile 流程的!一个需要几百人甚至是上千人一起协同合作的项目,用所谓的 Agile 方式,我个人认为行不通。"

关毅没有搭话,也没有转头去看,他知道那是谁——Franco,用现代人的话说,一个绝对的牛人。在刚才的闲聊中,他特意了解了 Franco 的情况。Franco 几乎具备了一个牛人所应有的全部特质:国内顶尖大学计算机系科班出身,毕业后直接进入

E公司,5年的工作中文章发表了无数,还申请了几项专利,另外,他还是几个著名开源项目的贡献者。任何一条拿出来都得让人羡慕一阵子。不过,从刚才的语气中,关毅似乎听出了些什么,那是对目前 E 公司研发管理战略上的不满——为什么放弃已经被实践证明了的传统开发模式,而去选择被"炒"得火热的 Agile。

"我一直认为……" Franco 发现每个人都竖起耳朵仔细地听,不觉来了兴致,"真正的软件发开流程不可避免地要经历几个阶段:需求、设计、编码和测试,最后是产品面市。这几个过程缺一不可。通过理论研究和实践得来的,是绝不会有错的。"

"可是,即便是 Agile 开发模式也符合这样的流程啊!"不知是谁插了一句。

"当然不一样了,Agile 讲究用小作坊式的模式,尽可能简化需求和设计,侧重于编码和测试。这样的产品如何能满足实用的需要?其实,现在好多公司的产品不是做得不好,而是功能复杂,界面不友好,不能满足用户的需要。说白了,就是需求做得不到位。如果他们把这个开发周期的一半时间用来做客户需求调研和分析、产品结构设计和论证,何愁开发不出好的产品。"Franco继续说道。

听了这些,关毅的思绪如同放电影似的回到了几年前 X 公司 F4 产品的动员大会。同样牛的人用同样的语气说着同样的话,整个会议室的人都在鼓掌,所有在场的人都豪情万丈,梦想用自己的力量创造出世界—流的管理软件,为祖国争光,为民族争气。关毅极力地想忘记这些,然而结果却是他越想忘记,记忆就越清晰。

是啊,过去几年如同炼狱般的生活怎么可能是一句"忘记"就能够解决的呢? 关毅苦笑了一下,逐渐从自己的回忆中回过神来,用自嘲的口吻苦笑道:"呵呵,你们知道吗? X 公司在瀑布式开发流程中引入 Milestone 里程碑机制,正式产品发布之前必须要完成 3 个 Milestone 版本,可效果并不好。"

就在这时,徐天问了一句,"时间不早了,大家都吃好了吧?咱们撤?"

大家一个个收起思绪,恢复了常态,彼此开着一些饭后常用的玩笑,其中一个一边拍着肚子,一边满不在乎地说:"唉,今天我啥也没吃,要不咱们再来碗担担面?"

"哈哈哈……"大家都被他的话逗得开怀大笑。关毅也松了口气。还好,自己没有破坏这次聚会。



关于软件开发流程的争论

在回公司的路上,大家都一反常态地各自想着心事。快到公司的时候,Franco叫住关毅:"关毅,能和我单独聊一会吗?"

"什么事, Franco?" 关毅问道。

"我没别的意思,只是我一直在想你刚才说的,如果时间允许,我们可以在瀑布式开发的基础上多做几次 Milestone 版本。有一点我想问你,你说的 Milestone 版本是个什么样的版本? 是能给公司上层演示的? 还是可以给用户试用的呢? 我觉得这个想法很好啊! 你们以前是不是就是这样做的? 现在几乎所有的公司为了提高市场占有率,都尽可能地缩短开发周期,节省成本,但是结果也显而易见——产品质量无法保证,补丁一个接着一个。唉,想起来这些我都替那些公司心烦啊!"

关毅注视着 Franco, 心情有点复杂。不过他倒是很有兴趣和 Franco 进一步探讨, 把自己的经验教训告诉 Franco。

"Franco, 你说传统软件开发模式的经验是从哪里借鉴来的?"关毅尽量压低音量,以免引起别人的注意。

"这个嘛,我们常说的开发模式其实来自于软件工程研究的一部分。在学校的软件工程课上老师也教过,为了避免软件危机、提高软件生产率及质量,从而引入了传统工业中的工程管理学,一举打破了软件开发一直局限于手工作坊的局面。对了,你可以把开发软件想象成建筑工程,从需求、设计到施工,就是这么回事。"Franco讲得兴起。

关毅微笑着点了点头,"咱们就说建筑这行,你家装修过吗?"

Franco 说:"装修过,记得那次可把我累死了,装修是一个典型的项目。"

关毅接着说:"太好了,我也正要买房呢,最近常在网上看买房和装修的经验。 以后还要向你请教。我想问的是,你在装修的时候是怎么做的呢?是先花大量的时 间做需求分析吗?" Franco 有点明白关毅的意思了,他摇摇头:"当然不是,一开始我们也想不到那么细,只能照一个大概的设计方案来施工。"

关毅笑着说:"对,没错,装修到一半的时候你也许会发现卧室插座的位置不方便,餐厅的墙面应该换一种颜色,还有很多诸如此类的问题。"

Franco 点点头:"的确。可是这和软件开发一样吗?"

关毅说:"软件开发比装修还要复杂的多!"

关毅继续向 Franco 倾诉: "Franco, 你猜得没错。我在 X 公司时, 那里的软件开发流程的确是以一个瀑布开发周期为一个迭代周期, 准备至少迭代 3 次,每个迭代完成后都会发布。第 1 个作为 Demo 版,会给所有的战略伙伴演示;第 2 个作为 Beta 版,给已有用户试用;正式的发布就是第 3 版。听起来如何?"

关毅不等 Franco 开口,就又继续讲道:"在产品真正发布前,我们做了 3 次需求分析,每次都力求完美。"

"这不是挺好的吗?是 X 的哪个产品啊?有机会我一定要见识一下。" Franco 有些迫不及待了。

"F4!" 关毅说。

"F4? 我怎么没听说过呀!" Franco 显然无法相信, X 公司费这么大力气开发的产品,一定是惊世骇俗的,自己怎么可能没听说过呢?这绝不可能!

"是啊,估计除了做过 F4 的,别人都不会知道。这个产品是秘密研发的,已经做了三年了,我离开的时候 Beta 版算是完成了,距离 Release 的目标可以说进了一大步啊!"

"啊……" 听了关毅的话, Franco 惊诧得下巴都快掉到地上了, "怎么会这样! 这三年你们都干什么去了?"

"客户需求的调研、分析,业务逻辑的设计、论证,前前后后就拖了两年。在这两年里,开发和测试部门十分清闲。这些前期工作完成之后,如果开发和测试再花上一、两年,谁能保证 Release 的产品是用户所要的?这样,横竖也是落后了市场好



几年,何谈竞争力啊!而现在的用户,需求基本上都是一天一个样。如果现在用户提出要追加一个大的功能模块,你该怎么办?暂缓开发,继续提需求、做设计?或者一边开发一边修改需求和设计?好像都不合适吧!这样的话,在开发之前进行的那一套调研、分析、设计、写文档的流程又得重来一遍,好些工作等于白做。凭心而论,如果按照传统的开发模式,所谓的需求要尽量做到充分,可能永远都只是一句空话。"

Franco 陷入了沉思。这时,他的心里就像打翻了五味瓶,什么滋味都有——有不服、有无奈,而更多的则是疑惑。要真是这样,X公司岂不是很危险?虽然这个极端的例子比较有说服力,但 Franco 还是觉得敏捷开发仿佛是另一种极端——和小作坊式的开发没有什么区别。

关毅回到自己的座位上,思绪久久不能平静。与 Franco 的争执让他开始重新思考自己在 X 公司的经历和目前应该如何面对 E 公司的新工作。关毅觉得,其实最重要的还是信心。

Agile 开发模式到底能不能保证新产品开发的顺利进行?遇到阻力的时候应该怎么办? Agile 是解决问题的灵丹妙药吗?况且,连 Franco 那样的牛人都在怀疑。

关毅脑子里一直在思考着这些问题,一时间,他有些手足无措,不知道该干什么了。

徐天发现关毅从午饭回来后就有点心不在焉,完全没有了上午的精神头。到底 发生了什么?天知道。不过可以确定的一点是,这样下去肯定是不行的。身为经理, 就有责任解决自己团队成员的各种问题——无论是工作上的,还是生活上的。

为了不让关毅感觉到压力,徐天给他发了封 E-mail。

Hi , 小关!

如果这两天有什么不习惯、不满意的地方,不要觉得难为情,只管告诉我,看我能不能帮助你。如果心里有什么疙瘩,

或者有什么疑问,咱们可以找个时间好好谈谈,你看行吗?

徐天

读完这封 E-mail 后,关毅心里充满了感激。于是,他毅然决然地做出了一个决定——在给徐天的回复中,他把自己的博客地址发给了徐天,说不定徐天真的能给自己一个答案。而且,就算没有答案也没关系,至少让徐天深入了解一下自己的背景也行。

做完了这些, 关毅深深地吸了一口气。



来自经理的帮助

时间到了晚上9点。

徐天结束了一天的工作,坐在家中温馨的书房里,仔细地读着关毅的博客。关 毅幽默的文字让徐天忍俊不已。呵呵,80后?70后?他29岁,的确是一个说大不 大,说小不小的年龄。不过他"遭遇"的那些经历和经验正是徐天看重的,因为徐 天自己也曾有过类似的经历。

读过关毅的博客,徐天仔细查阅了自己明天的工作计划,取消了一些无关痛痒 的会议。他想尽量给关毅预留多一些时间。

第二天,徐天一到公司就直接找到关毅,两个人找了一间会议室,也没有什么 寒暄,就直入正题。

"小关,我看过你的博客了,很有感触,只是有些地方不是很明白,即便这样,我一样可以理解你现在的心情。我也有过类似的工作经历,现在就咱们两个,有什么想说的、想问的、想不通的,咱们都可以好好聊聊,工作上的、生活中的,都可以,怎么样?"

关毅点点头,然后低着头沉默了一会儿,终于下定决心,说出了憋在心中许久的困惑:"徐天,咱们实话实说,自从在 X 公司经历过那些之后,我对大型组织的管理有点丧失信心了,真不知道咱们 E 公司会怎么样。"

关毅调整了一下情绪,继续说道:"这些天我一直在思考,F4 为什么会做成这样?当然,我们不能把所有的问题都推给开发流程。其实,在管理上,F4 的问题更大。前期部门之间自发的协作和沟通基本没有,所以需求、设计的问题到了编码阶段才逐步暴露出来,这时再由程序员们逆流而上去协调、解决,难度之大是可想而知的。我不是说需求和设计人员的坏话,可是如果给了他们绝对的权利,他们却不用承担相应的责任,这样情况就永远不会改变。"

借着关毅提出的问题,徐天说:"是啊,部门之间的合作确实是个让公司比较头疼的问题。咱们公司为了解决这样的问题,倡导实行一种叫做'矩阵式'的管理模

式,就是在需求、设计、开发、测试部门的内部根据独立的模块划分成多个虚拟的项目组,这样就把模块内的需求、设计、开发、测试人员紧密结合起来,可以说是最大化地降低了人员之间的内耗,从而调动起整个项目组的积极性。你觉得怎么样呢?管理上的问题是每个公司都会遇到的,而且随着公司的不断发展,原来的模式一定会有不能适应的一天,所以,公司的管理者同样需要创新。可能 X 公司的高层还没有意识到这一点,或许他们也不是没有意识到这一点,只是需要时间,特别是对国内的企业而言。所以,千万不要对一切都失去信心,失去信心才是最可怕的。"

徐天的话让关毅陷入了沉思,大约过了半分钟,关毅苦笑了一下,说道:"你说的没错,X公司的管理模式有问题,领导的权利太大,而且没有人能制衡他,以至于在决策中出了问题,没办法及时发现和补救,就只能离目标越来越远了。"

"还是面试时候我问过你的,难道你们没有向管理层反映过吗?比如说,你可以告诉 F4 的总监,说设计和需求之间缺乏沟通,这样对产品的开发是有极大隐患的。"徐天有些不解。

"反映?我都不知道反映过多少回了,效果甚微。后来,我们所有的开发经理恳请我们的部门经理去找总经理反映一下困难,谁知道害得部门经理被 F4 产品的总监狠狠 K 了一顿,从此以后,就再也没人敢说什么了。李总监一个人真是有绝对的权威啊!"

提到这些,关毅总是一脸的无奈,"最后的结果是我们开发部门的怨气很大,人员流动极为频繁,我当然也是其中的一个。说到底,就是领导认为我们开发部门的工作是在整个 F4 中技术含量最低的,因为已经有了 XFF 这样的自动开发平台,随便找个玩过电脑的人来都可以胜任。真是可笑啊!"

"小关,我不能对 X 公司的问题评头论足,但是在 E 公司,这样的问题会少一些。"徐天郑重地做出了保证。

关毅轻轻地点了点头:"谢谢。其实我一直认为,在 X 公司的经历是我的宝贵财富。反思一下在 X 公司遇到的那么多开发流程上的问题,其实我也没有提出特别系统的和有建设性的解决方案,所以管理层也就没有足够的重视吧。问题也许大家慢慢就都知道了,但是究竟应该如何解决呢?"



徐天看着有些茫然的关毅,微微地笑了一下,"这个问题我先不回答你,但是你 下周不是要参加关于敏捷开发的培训吗,说不定你能找到答案。"

关毅看着徐天诚恳的眼神,不禁在想,敏捷开发,Agile,真有那么神吗?

总结和思考

X 公司的问题不完全是由传统的开发模式造成的。但是我们可以看到, 传统的开发模式确实大大阻碍了 X 公司的产品,甚至人的成功。





关毅参加敏捷开发培训颇有收获,回来后便给部门全体同事作了一次讲座,算是部门的一次敏捷动员大会。接着,关毅的团队——电子商务通北京开发中心正式开始组建了。关毅和他的经理招聘了3名符合敏捷开发要求的成员。随后,关毅前往电子商务通产品的研发总部温哥华接受培训。在那里,关毅第一次亲眼目睹了Scrum 敏捷开发流程。另外,关毅还和温哥华的同事们讨论了具体的项目需求。与此同时,远在北京的新团队成员们也对敏捷管理工具进行了调研,为今后的开发管理工作打好了基础。



敏捷开发培训——什么是敏捷开发?

时间过得很快,关毅刚刚适应了新公司的工作环境和氛围,新鲜和兴奋劲儿还 没过去,参加 Agile 培训的日子就到了。

培训那天,他特意起了个大早,找到培训地点后,挑了个前排的座位坐下,打 开电脑,又从兜里掏出了 MP3 准备录音。

旁边的一位同事看见了, 打趣地说: "哥们儿, 装备真够全的啊, 就差 DV 了。"

"呵呵, 听说是大牛嘛, 还不得好好瞻仰一下。" 关毅笑道。

话音没落,就看见一位大胡子老外走了进来。首先是一番惯例的自我介绍,一连串英文单词不停地从他嘴里蹦出来,关毅赶紧打起精神仔细听,一边暗自庆幸,还好带了 MP3。

老外叫 David, 从上世纪 90 年代末就开始研究敏捷开发了,有很丰富的敏捷开发经验。他自称很喜欢中国文化,还给自己取了个中文名叫张大卫。

介绍完自己, David 接着说:"我知道你们中国有个重要的节日——春节。在这个节日里, 所有的人不管在哪里都要回到家乡, 对吗?"

大家对老外同志突然提起中国的传统节日感到好奇,纷纷答 Yes。

David 接着说:"那各位都是采取什么方式回家的呢?"

他得到的回答自然五花八门,有的乘火车,有的乘飞机,有的自驾车,还有人 甚至要把几乎所有的交通工具都换个遍才能到家。

David 显然对最后这种需要耗费极大人力物力财力的方式感到很好奇,但是他没有忘记自己的初衷,接着说道:"我想知道那些选择自驾车的朋友们,在驾车回家乡的途中有没有遇到过意外情况,从而延误了你们的行程呢?"

这下课堂里可炸了锅了,大家都想起 2008 年的大雪,事先谁也没有料到会发展

成雪灾——很多开车回家的人都被大雪堵在了半路,进退不能;很多趟火车也晚点 好多个小时,甚至好多天······

David 一直笑眯眯地看着乱成一锅粥的课堂,过了好一会儿才说:"哪位能给大家仔细说说你开车回家时遇到过的意外情况呢?"

一位同事站起来说:"我的老家在江苏,我每次开车回家都要经过南京。南京这些年发展得很快,几乎每年路过都会有意想不到的情况,有时是我习惯走的地方在修路,有时是在修建新的大桥。由于我对南京不熟悉,每次这些情况发生时总要绕很多路才能回到家。"

他刚坐下,另外一位同事也站了起来:"就说发生雪灾的那年吧。那时我还在哈尔滨,而我家在江西南昌。哈尔滨没有直达南昌的火车,所以我一直是先坐火车到北京,然后从北京坐一趟走京广线的火车到南昌。谁知那年湖南暴雪,走京广线的列车全被堵在半路……后来我才知道,当时京九线的情况还不错,如果我选择这条线路也许就能按时到家了。"

大家七嘴八舌地发着言,关毅倒没有说什么,他看着 David,觉得自己好像有点 隐约明白 David 的用意了。

David 听完大家的发言, 笑着说:"对呀, 大城市里最容易发生状况了——修路、堵车、车祸, 你事先设定好的路线计划往往都赶不上实际情况的变化。如果遇到这些情况的时候, 你仍旧要坚持以前制定的计划, 一定要按计划好的路线行驶的话, 那必定会延误你的行程, 花费更多的代价, 甚至到不了你的目的地。"

然后, David 转身在白板上画了张草图 (如图 3-1 所示)。

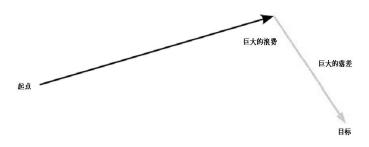


图 3-1 传统的瀑布软件开发效果示意图

维松Scrum之旅

——敏捷开发故事

David 说:"传统的软件开发流程,比如瀑布流程,就像我画的这张图(如图 3-1 所示)一样。从需求、设计,到开发、测试和部署,是一环套一环的,结束一个环节才能开始下一个,中间过程很长,且不允许有变更,等到交付的时候才发现已经与客户的目标相差太远了。这时,要想弥补就要付出巨大的代价,很多工作需要重来,从而产生了巨大的浪费。"

David 接着转身又在白板上画了一个草图(如图 3-2 所示)。



图 3-2 敏捷软件开发效果示意图

这下大家都明白了, 敏捷开发的每一个迭代就相当于 David 画的第 2 张草图(如图 3-2 所示)中的一个小箭头,由于开发工作能够根据需求的变化不断调整,所以整个过程中浪费很少。

David 说:"春节回家,一般情况下,大家都会在出发前制定一个大概的计划,明确你家所在的城市这一最高目标,然后顺应情况不断调整行程计划。"

他又接着说:"好了,假设你们之中有几个人是老乡,现在结伴开车回家,旅途中还有什么是比较重要的?"

"因为通常我会告诉家人我大概到达的时间,所以我想时间很重要。"一位同事 回答。

"很好,到达时间!我们需要根据已有的经验判断我们行驶的速度,算出最终到达的大概时间。我们今天上课的内容是讲敏捷开发,你提到你的家人关注你到达的时间,实际上,你的家人就相当于我们软件开发中的利益相关者(Stakeholders)。我们所有的目标都是为了实现利益相关者的利益。"David 说。

"如果路上时间很长的话,我们一定要找点事做,比如打牌或者玩 PSP。"另一位同事回答。

"很好! 打牌和玩游戏实际上是为了避免旅途太过枯燥。在一个团队里也需要不断的自我激励,以保持一种高昂的士气。敏捷开发强调尽可能让每个成员都做自己认为有趣的事情。还有吗?"

"旅途中需要经常休息吧?"

"太好了!这正好说到了团队开发中非常重要的一点,就是保持一个可持续的进度(Sustainable Pace),绝不能太累了。事实证明,软件开发团队过度疲劳会严重影响生产率,降低产品质量,得不偿失。疲劳驾驶可是要出车祸的。还有吗?"David接着问。

"大家在旅途中遇到问题实际上都是要互相商量的……"

"很好,这实际上说明我们是一个自我管理的团队。"

David 说,"谢谢大家的踊跃发言,刚才谈到的几点非常重要。你们结伴同行的人实际上就组成了一个敏捷团队,一个成功的敏捷软件开发项目就恰恰具有刚才所说的那些特质。看见了吗,很多敏捷的思想都来源于生活。"

David 继续说道: "下面我们做个游戏。这个游戏叫'命令和控制'。首先,我需要两位参与者,其中一位是老板,另外一位是执行者,也就是员工了。老板可以发出6个指令,包括走、停、左、右、快、慢,执行者必须服从老板的指令。老板只能发出指令而不能触碰到执行者。你们的目标就是要使执行者在2分钟之内走完60步。当然,我会在你们的行进途中设置很多障碍。哈哈!好,我们现在开始吧!"

话音刚落,就有两位同事自告奋勇站起来了。等他俩分配好"老板"和"员工"的角色之后,David 让大家起身把椅子腾出来,还故意摆得乱七八糟,这才让他俩在一堆椅子之间开始游戏(如图 3-3 所示)。

游戏开始了,"老板"开始发号施令。由于有障碍物存在,他得不时地让他的"员工"向左走或者向右走以避开障碍物。而另一方面,"员工"虽然知道该如何避开这些障碍物,但他却不得不等待"老板"下命令后才能行动。这样一来,"老板"思考的时间、发出指令的时间和"员工"的反应时间,这些无疑都成了拖延时间的因素,最终那位"员工"也没能在2分钟内走完60步。



图 3-3 混乱中走出 60 步

随后又有几对搭档玩了这个游戏,他们要么也像第1对一样没有按时完成任务,要么就是手忙脚乱地达成了目标。即使完成了目标,也还免不了有一些违规的情况发生:有的"老板"总是忍不住想要亲自带领"员工"行动,还有的"员工"与"老板"意见不一致时总是忽略"老板"的意见……

"好, 谁能谈谈对刚才这个游戏的体会?"游戏过后, David 发问。

一位扮演"员工"角色的同事说:"做员工很累,要不停地接受命令。"

接着,另一位扮演"员工"角色的同事说:"做员工也很没意思,像个机器人,没有自主权,明明知道怎么走却要等命令才能行事。"

"刚才游戏中的'员工'发表了意见,'老板'的代表能谈谈体会吗?" David 说。

"当老板更累,要根据情况不停地发号施令。"一位"老板"说。

另一位"老板"似乎当得有些委屈:"老板不容易,员工有时候不听老板的。我还没发命令,他就自己走了。"

"很好,谢谢大家,我们再接着做第二个游戏。第二个游戏叫'自我导向'。大家听名字就应该明白是怎么回事了,就是你们所有人都是执行者,需要你自己在障碍物之间用 2 分钟走完 60 步。"

David 让所有人参与,同样在乱七八糟的椅子阵中穿行,所不同的就是这次只有"执行者",而没有发号施令的"老板"。结果,所有人都按自己的判断,在选择路径的同时避开了障碍物,很快就走完了60步,用时还不到1分钟。

David 调侃道: "大家看,在第二个游戏里没有'老板'指手划脚,大家完成得多好啊!'老板'总是耽误时间并拖延项目的那个人,呵呵。这两个游戏讲的是自我管理的团队,这在敏捷开发中是一个非常重要的理念。敏捷开发的管理者要充分相信团队能够完成任务,并充分授权让团队进行自我管理。而传统开发过程的理念则恰恰与之相反。这几个例子就当作敏捷开发的引子吧,下面我们来全面介绍敏捷开发。"

"什么是敏捷开发?"它是一种开发方法学(Methodology),可以应对客户快速变更的需求。它强调以人为核心,采用迭代的方式,循序渐进地开发软件。在敏捷开发过程中,软件项目被划分成多个相互联系但也能独立运行的子项目。这就使得每个子项目在开发、测试直至完成的过程中一直保持可使用的状态。这个过程实际上就是要形成开发过程中团队之成员之间更加有效的合作关系,使其灵活性更高,以适应不断变化的需求。敏捷开发过程与传统开发过程的最大的不同之处在于,在敏捷开发过程中,团队是有激情、有活力的,能够适应更大的变化,生产出更高质量的软件。"

"接下来我谈一下敏捷的价值观。这是在 2001 年敏捷联盟成立的时候一些业界 专家共同提出来的。可以说,只要真正领悟了敏捷的价值观,就能懂得什么是敏捷。"

"我们开发软件时的首要任务是通过尽早地、持续地交付有价值的软件来使客户 满意。请看 PPT……"



资料库

• 个体和交互重于过程和工具

敏捷方法认为,人是软件开发中最重要的因素。开发团队成员之间有效的交流、沟通与协作,比单纯的编程能力更为重要。人与人面对面的交流,是最有效、最迅速的交换信息的方式。

• 可以工作的软件重于面面俱到的文档

过多的文档需要开发人员花费大量时间来维护。文档应该是为程序服务的,因此应当短小精悍、易于维护,而且主题突出。敏捷方法认为最根本的文档就是源码。



• 客户协作重于合同谈判

客户对产品的需求是不断变化的,试图在合同中规定项目的细节和进度显然无法应对不断变化的需求。只有开发团队和客户之间真诚的协作,加上频繁的客户反馈,才能让项目走向成功。

• 随时响应变化重于循规蹈矩

客户的需求可能在项目开发过程中不断变化,即使是在合同谈判阶段确定的需求, 也可能在客户看见了逐渐成型的产品之后而发生改变。敏捷方法欢迎并且随时准备应对 变化。制定计划的时候应该尽量简洁、灵活,使其能适应技术和需求方面的变化。可以 说,随时响应变化的能力往往决定着一个项目的成败。

敏捷开发方法的核心思想概括起来就是"适应变化"和"以人为本"。

(1) 敏捷开发方法是面向人的而非面向过程的。

敏捷开发认为人是软件开发中最重要的因素,而且人工作的环境很复杂。它希望使软件开发工作顺应人的天性而非逆之,强调软件开发应当是一项令人愉悦的活动,因此它们注重调动人的积极性、主动性和创造性,并培养人在工作中的自豪感。敏捷开发的理念就是信任开发团队能够很好地完成任务,所有的管理都是围绕这个理念展开的。

(2) 敏捷开发方法是"主动适应的"而不是"预先设定的"。

瀑布模型等传统软件开发过程试图对一个软件开发项目在很长的时间跨度内作出详细的计划,并形成详细的文档,然后依照计划进行开发。这类方法在计划制定完成后拒绝变化,后期的需求变化将会花费极大的代价。而敏捷开发方法则乐于迎接变化,其实,它的目的就是成为适应变化的过程。另外,据统计,很多软件产品的功能中,客户常用的功能只占 20%左右,其他大部分功能是客户很少使用甚至基本不用的。在这种情况下,采用瀑布方式在详细设计阶段所设计出的功能,其实很多是不必要的,这将浪费很多资源。在敏捷开发中,要求客户始终参与整个开发过程,这使得敏捷团队能不断地获得客户反馈,不断适应需求的变更,从而使最终的产品充分符合客户的要求,也极大地减少了资源的浪费。敏捷开发的理念认为未来的开发过程是不可详细预知的。

David 讲解到:"关于这里的第二点,我举一个例子。假如你接到的任务是要为

一群人盖房子,如果按照传统的软件开发方法,你应该挨个采访他们,了解他们对房子的条件和各项设施的要求,并将这些不同的要求详细记录下来。等这些工作完成之后才破土动工,按他们的要求建造房屋。在房子没盖好之前,他们一直没有房子住。等房子盖好了,如果有人对它不满意,就得拆掉重新盖。而敏捷呢?你可以先了解他们的大致需求,快速盖好一大间可以住的房子,然后再请他们提意见,按每人的意见继续改进,经测试满意后再继续满足下一个人的要求。"

"我再澄清一些常见的对敏捷开发方法的误解:"

"有的人对采用敏捷开发是否能真的有效提高效率并生产出成功的产品表示怀疑。他们认为,在敏捷方法中,由于没有经理的管理和约束,团队和项目必然是一团糟,仿佛越是上层越有这种想法。但是玩过刚才的游戏大家应该有切实的体会,正如我刚才所说过的,敏捷开发的理念是信任开发团队,信任每一个人。试想一下,如果你们这个团队对你们的项目充满激情,而老板又充分信任你们,那么你们必然会将更多的时间花在如何有效地提高生产率、如何创新地完成某个功能等方面,而不是按老板的意思按部就班地工作了,这样还会节省很多时间并简化流程,例如开会、向老板报告、请示老板、等老板批准等。就像咱们刚才的那个游戏结果所揭示的那样,充分调动参与项目的人员的主动性,让他们自我组织、自我管理,由团队自身来掌握项目进度,比老板整天催促反而更有效率。"

"当然,敏捷开发的团队也需要管理,但这些管理是非命令式的,很多时候是战略指导和服务性质的。在敏捷开发中,管理者对团队和项目的管理表现在挑选合适的人、为团队创造一个开放而自由的工作环境、经常性的反馈、为团队建立评估和奖励机制、当团队有方向性错误时能及早发现并纠正、容忍错误的发生等。"

"还有一种误解,认为敏捷开发就是完全不需要计划、文档和架构设计。这种看法也是不对的。敏捷开发也需要文档,也同样要计划。但我们要明白,计划赶不上变化,在这样一个不断变化的情况下要做出详细的计划是不可能的。因此敏捷方法认为不值得在这些因素上花费过多的资源,可工作的软件才是敏捷方法关注的重点。敏捷团队依靠变化来获取活力,他们更愿意使设计保持尽可能的干净、简单。基于以上的原因,采用敏捷方法的项目初始设计是比较粗略的,并需要使用许多测试手段作为辅助,这就保持了设计的灵活性和易于理解性。团队可以利用这种灵活性持续地改进设计,以便于每次迭代结束时的系统都具有最适合于那次迭代中需求的设



计。摆脱一切对软件开发不合理的束缚就是敏捷。"

"敏捷联盟的发起人 Martin Fowler 和 Jim Highsmith 曾经这样解释过:敏捷开发 所追求的是一种平衡——我们也建模,但不仅仅是画几个模型图存放在少人问津的 项目文档库里;我们也需要文档,但从不浪费纸张去编造那些极少使用而又没有保 存价值的大部头;我们也做计划,但我们同时也认识到在这纷繁复杂的环境中做计 划是困难的。"

"但是,敏捷开发不是可以解决所有问题的银弹。在实际的项目中,受条件的限制,有些原则实施起来确实有困难,那该怎么办?要知道,敏捷并不要求你们一成不变地遵循这些条条框框,遇到困难时应该灵活地调整策略,这样才真正达到了敏捷的目的。"

.

培训结束了,关毅听得十分过瘾,会后还追着 David 问了不少问题, David 也立刻喜欢上了这个中国小伙子。最后, David 临走前还给关毅留下了联系方式,并表示以后有问题可以尽管和他交流。

敏捷动员大会

时间过得很快, 关毅负责的项目马上就要启动了。

上次培训过后,经过潜心钻研和消化,关毅已经对敏捷理论不再陌生了。

一天下午,徐天召集部门全体员工开会,出乎大家意料的是,关毅这个新人第 一次成了会议的主角。

徐天把投影仪的操纵权交给了关毅,并对大家说:"今天我主要想请关毅给大家 讲讲敏捷开发的基本思想,他刚参加完培训,非常有心得。"

Franco 正用昏睡的眼神盯着投影幕布,当幕布上突然打出"敏捷开发"的标题时,他一下子来了精神。

关毅清了清嗓子,开始了激昂的演讲。他费了不少功夫准备,把两天的培训内容压缩到1个小时,就是为了第一次在大家面前好好表现一下自己。当然,最重要的是要"推销"敏捷开发的思想。

关毅从敏捷宣言开始讲起:"软件开发中优先级最高的事情是给客户提供有价值的、可以工作的软件,因此软件开发最重要的是给客户创造价值。那么,谁来创造价值呢?主要由人来提供。所以在敏捷开发里,人是最重要的因素,而非流程和工具。传统流程过分强调流程和工具,因为传统的软件工程是从传统制造业继承而来的。但是长期以来我们犯的最大错误就是把软件开发比作其他像汽车制造或者建筑一样的工程来看待。其实软件开发是截然不同的,它是随时随地的创造性的工作,尽管可能谈不上艺术,但也绝非普通的工程。所以敏捷开发很伟大地跳出了传统的软件工程,以人为本。"

"敏捷开发认为人与人之间有效的交流和协作是最重要的,透过一切流程和工具 看本质,实际上就是使人能够协作开发软件。如果能够保持最有效的交流和协作, 又何必在乎流程和工具呢?以前我们为了遵守流程而遵守流程,实际上花了很多力 气,但是走偏了。在所有的交流方式中,面对面的交流是最有效的。可以工作的软 件胜过文档,敏捷开发的宗旨是提供可以工作的软件,只有可以工作的软件才是有



价值的。过去,我们要求完备的文档,实际上这些文档并没有创造价值。一切离开了创造价值的活动都是毫无意义的浪费。所以,凡事如果偏离了本质目标,即使再努力,也一定做不好。"说到这里,关毅的脑海里又突然跳出了那句话:人生的选择大于努力。

关毅继续说:"敏捷开发随时拥抱变化、响应变化,而不是恪守计划。敏捷开发认为,在软件开发过程中,未来是不可完全预知的,并且要求软件开发团队进行高度的自我管理,管理者要充分信任开发团队。而传统开发流程认为,未来是可以计划和预测的,另外,管理者也不完全信任开发团队,需要通过命令和控制的手段来管理项目。显然,敏捷更真实地符合现实并顺应了管理的趋势,那种从上到下高压式的控制管理方法已经过时了,不单是在软件开发领域,在很多领域都已经被抛弃了,那其实是工业时代的产物。"

"总而言之,我认为敏捷开发的本质,第一是一切活动以价值为导向,第二是以 人为本。以价值为导向可以帮助我们大大提高软件开发的效益,以人为本的精神可 以帮助我们大大提高软件开发的效率。"

"下面是自由提问的时间。"关毅讲完,望了望大家。

这时候 Franco 开始提问了: "我为什么要用敏捷?我觉得我们现在的瀑布流程 挺好的,能够保证我们这么大的产品成功那么多年。"

关毅笑了笑,说:"发哥的广告看过吧?别人认为他很成功了,他谦虚地说自己才刚上路。所以不要轻易说成功,产品能按时发布到市场上去不代表成功。我问你几个问题。你觉得你们组的产品在市场上卖得好吗?有多少功能是客户觉得有用的,还卖那么贵?对那些没用的功能投入的研发力量是不是一种浪费?你觉得客户对我们产品的质量满意吗?在产品发布之前,有多少测试和开发工程师需要挑灯夜战?在产品发布以后,又有多少技术支持和服务的工程师还在忙着修补漏洞?你觉得你们团队的每个成员都很愉快吗?他们能做到每天高高兴兴按时上班,高高兴兴按时下班吗?每个成员每天都有进步和成长的感觉吗?"

Franco 张着大嘴,一时说不出话来。他没想到,关毅这个新人在这么短时间内就给人锋芒毕露的感觉。

关毅接着说:"当然,实施敏捷开发是很痛苦的,这种转变不是一天两天就可以 完成的。敏捷开发也不是银弹,不能指望它解决所有的问题。"

Franco 没有罢休,接着问:"你刚才提到了在敏捷开发中要保持一个可持续的进度,不提倡加班,每个团队成员似乎都很高兴。那你又说实施敏捷开发是很痛苦的,岂不是很矛盾?"

"人的思想转变过程是最痛苦的。从传统流程转换到敏捷的过程是复杂的,甚至是对传统的管理模式、组织形式的一种挑战,往往不是一个人的力量能完成的,需要从下到上、从上到下地推行,如果碰上阻力,是很麻烦、很痛苦的,尤其是老员工可能不喜欢敏捷这样的新玩意。但是一旦足够敏捷了,每个人都会很愉快。"

"敏捷里讲需要自我管理的团队,那还要经理做什么?你觉得中国人能实现自我管理吗?" Franco 继续问道。

"经理应该相当于足球场上的教练,在场外指导,但是真正踢球的是团队。教练不会告诉队员应该如何来踢每一脚,更不会进入球场自己去踢,顶多急了对着场上喊一嗓子。的确,中国老百姓过去几千年都习惯被'奴役',不过我们可是优秀的70后、80后的代表,崇尚自由,有独立的思想,从小就接受过良好的教育,有团队合作的精神,这样你觉得自我管理行不行?"

徐天咳嗽了一下,看了看表,已经快到时间了,于是对大家说:"感谢关毅给大家带来的精彩演讲。没错,转变是痛苦的,尤其是我们这个大部门,有很多不同的产品组,每个项目都很大,转变不会很快完成。关毅将要组建一个新的团队,相对好办,一切按照敏捷来打造。他的团队也会作为咱们的试验田,我们要随后跟上,不妨从一点一滴敏捷起来。以后大家有问题可以多和关毅交流。"

徐天说:"同学们,我们今天开始就正式迈上敏捷之旅了。我们都没有什么经验,有的只是一些理论,摸着石头过河。但是如果成功了,我们会成为 E 公司中国研发中心的明星团队,可以帮助更多的团队提高他们的软件开发效率和效益,相信这也会为在座各位的职业生涯写上漂亮的一笔。我相信我们一定会成功的!"



招兵买马——打造敏捷开发团队

关毅的敏捷开发介绍很成功,仿佛成了整个部门的敏捷动员大会。不过,这仅仅是一个良好的开始,他现在最重要的就是要着手招兵买马,打造敏捷团队了。

要招聘,先得有招聘启事,最重要的就是职位描述(Job Description, JD),一般的职位描述里会介绍职位的大概职责和对应聘人的要求。关毅这次要招 3 个人,全部是软件开发工程师。关毅的团队将要做的产品是 E 公司大名鼎鼎的在线电子商务系统——电子商务通。E 公司的这个产品目前处于世界领先地位,IT 业内的人都如雷贯耳。所以,光是把产品名称放到职位描述中,就足够有吸引力了。那么,这个职位需要做什么?要设计和开发产品新的功能,要能够撰写设计文档,要修改产品缺陷并和测试工程师一起工作以提高产品质量,要给全世界的客户提供高质量的技术支持……乍一看,好像全世界的软件开发工程师或程序员所做的事情都大同小异,但实际上敏捷开发团队里的每个团队成员都有机会去进行软件设计,每个人都必须有非常强烈的客户和市场意识,根据客户的需求设计最好的软件。

下面该写要求了,做这个产品的开发要求自然也低不了,什么 Java、J2EE、Web 2.0、C++、面向对象技术,一个也不能少。另外,敏捷开发团队成员更需要什么样的软素质的呢? 关毅想了想,把这样一些素质写了上去: "A very good team player, excellent communication skills, open minded, pro-active, and self-motivated." JD 必须是英文,关毅也没办法,就怕应聘的人不能深刻理解这些词的含义。



资料库

• A very good team player

很好的团队合作者。敏捷强调团队,如果只是个人能力强而不懂得合作,这样的人 在敏捷团队里就没法混。

• Excellent communication skills

优秀的沟通能力。这一点的重要性不言而喻,敏捷里最强调的就是沟通,最有效的沟通方式就是面对面的交流。那种只会埋头干活,不会沟通的不要。

• Open minded, pro-active, and self-motivated

敏捷团队成员必须能够敞开思想,随时接受新事物,积极主动,自我激励。

关毅决心建立一个自我管理的团队,所以这些素质都是必不可少的。回想过去, X 公司那些兄弟们在干活的时候还是习惯被人牵着走,现在可不行了,希望 E 公司的平台能够帮助他打造一个梦幻团队。

想到这里,关毅心里一阵高兴。呵呵,不过这样的人得要多少钱一个月啊?不管了,这些让老大徐天去处理吧。关毅又看了一遍写好的 JD,按下了发送按钮。

第二天,徐天找到关毅,对 JD 很满意,但是他想补充一下对产品的介绍,其实说白了就是再"忽悠"一下。现在人才市场竞争那么激烈,稍不留神,潜在的人才就被竞争对手抢走了。

徐天和关毅一起把产品介绍进行了修改,翻译成中文就是:"我们只设计世界上 最优秀的电子商务系统,加入我们,和全球最优秀的工程师一起工作,每一天你都 有机会改变世界"。

这样的 JD 让关毅觉得很爽。毕业以来,他一直想要寻找的就是这种激情。

为了尽可能大面积地撒网,关毅在一些高校的 BBS 上也张贴了招聘启事。

也许是由于 E 公司的声望,也许是招聘启事中的激情感染了应聘者,短短两天功夫,简历就塞满了关毅的邮箱。这些简历有从 BBS 上看到招聘启事发过来的,有HR 推荐的,还有同事、朋友推荐的,甚至还有 E 公司其他部门的员工过来应聘。应聘者中有应届毕业生,也有在其他世界 500 强或者外企工作多年的"老鸟"。参加"海选"的人条件都不错,接下来就是挑人来面试了。

关毅即将打造的这个精英敏捷开发团队,需要每个成员不仅有优秀的技术能力、 优秀的团队协作和沟通能力,还要具备良好的全局观,而且最好能优势互补。

一个星期之后,"海选"般的面试开始了。

长话短说。虽然很多人的简历写得不错,不过一见面就不是那么回事了:有的沟通能力不行,有的技术能力不够深入,有的眼高手低……不过,真正优秀的、有



潜质的人才是需要仔细挖掘的,这些素质在简历上也许并不突出。

经过层层选拔,关毅的团队最后选择了这么3位应聘者:思维敏捷、擅长 Web 和 Java 的应届硕士胡志刚,伶俐大方、各方面技术相对均衡的段小丽,还有活宝余大鹏——胖胖乎乎的,酷爱编程和美食。

总结和思考

招聘实际上是从写招聘启事开始的,而不仅仅是面试。敏捷开发的核心是人,招到合适的人是所有开发环节中最重要的。



Hello, 温哥华

一个月后,段小丽、胡志刚和余大鹏都陆续来报道了。关毅带着他们熟悉环境, 并进行相关的技术准备。

Welcome Lunch 之后,团队气氛和士气空前高涨,大家也都分别有了外号:段小丽长得好看,于是大家都叫她"段美女";胡志刚长了一张娃娃脸,稚气未脱,大家都叫他"小胡";另外,"胖子"的光荣称号就给了余大鹏。

最近,还有件事让关毅兴奋得几乎睡不着觉。由于 E 公司的电子商务通产品是 在加拿大开发的,所以他马上要出差去加拿大接受培训,并且商讨任务的安排。长 这么大,又是头一次出国,关毅很想去看看外面的世界。

一个星期以后,关毅接受了徐天的嘱托,正式出发了。临行前,家里也得安顿好,他给段美女、小胡和胖子3位同学留了不少"作业",并且要求他们每天都要写工作汇报。

在飞机上,关毅打开笔记本电脑,再次查看这次去加拿大的任务安排,主要可以归纳为以下 4 点。

- (1)了解产品的基本情况;
- (2)接受开发技术的培训;
- (3)熟悉开发流程和开发环境;
- (4) 商讨开发工作的安排。

关毅心想,这次还有一个重要目的,就是建立起彼此的信任,和总部的人建立 良好的关系。他听说加拿大也在实施敏捷开发,所以也迫不及待地想去看一看敏捷 开发究竟是什么样的。

经过十几个小时的飞行,这个星期六,关毅只身来到了大洋彼岸的温哥华,异 国他乡的美景让他目不暇接。整个周末,除了倒时差,关毅只简单逛了逛,另外还

発がScrum之旅 ・ 敬捷开发故事

去公司踩了点。关毅凡事都喜欢有所准备。

星期——大早,关毅起床,在酒店里吃了顿丰盛的西式早餐之后,就出发前往公司了。

在路上,关毅猜想着,接下来到底会是怎么样的开场呢?那边的 Team Leader 会是什么样子呢?

初遇 Scrum——XP、RUP 和 Scrum 的比较

关毅开着租来的福克斯,一路畅通,心情舒畅,他心想:要是在北京,今天应该又是一个极其平常的星期一早高峰,这个时间肯定路上很堵。一路上没有看到一个维持秩序的警察,虽然温哥华市中心的车也不少,但一切都井井有条。

关毅想起了培训课上那个"命令和控制"的游戏,觉得颇有几分相似之处。如果每辆车都靠命令和监控行驶,对于一个城市来讲,维持交通运转的成本是不可想象的,而且依然是低效的。其实,只要每个人都自觉地遵守规则,就像今天的道路一样,那该多爽啊!自觉?说得容易。说到底,人还是根本的问题。关毅想到这里,不禁摇了摇头。

15 分钟后, 上午 8 点 50 分, 关毅把车停到了公司的楼门前。

关毅借助保安的引导来到了公司秘书那里。秘书是一个漂亮的印度小姑娘,她好像早有准备,立刻打电话通知本次负责接待关毅的 Team Leader。登记后,关毅坐在沙发上忐忑不安地等待着。不一会儿,走出来一个高个的大胡子,他叫 Peter,是关毅所在产品组的开发负责人。

"Nice to meet you, Guan Yi!" Peter 向关毅伸出了右手。

关毅握着 Peter 的手: "Nice to meet you, Peter!"

寒暄之后, Peter 领着关毅参观他们的研发基地。每到一处, Peter 都向关毅介绍这些团队的工作任务。

关毅在通往茶水间的走廊停了下来,他被走廊墙上的一块白板所吸引。这块白板上贴满了小卡片,这些小卡片上写的好像是各种 Task 的描述。白板上画满了表格,许多的小卡片就分散着粘贴在表格中的不同位置,花花绿绿,甚是好看。

此时, Peter 也回过头来, 笑着对关毅说:"关毅, 这是我们的任务白板, 回头你参加我们的每日 Scrum 会议, 让你看看我们是怎么玩的。"

"什么是 Scrum?" 关毅小声地问。

"你听说过敏捷开发吗? Scrum 是其中的一种敏捷开发方法。" Peter 回答道。

"哦,我知道敏捷开发的一些思想,来之前还专门接受过培训。我听说过 XP,不过还真没听说过 Serum。"

"Scrum 目前是世界上最流行的敏捷开发方法。我们也是刚刚开始采用敏捷开发流程,使用的就是 Scrum 这种方法。关毅,你这两天有时间可以上网了解一下。"

关毅点了点头。

E 公司在加拿大的研发中心的布置和北京差别并不大,倒是茶水间比国内的更厚道,除了免费的咖啡,还有无数好吃的东西和一个台球桌,是个放松的好地方。会议室的名称起得也很有特色,什么夏威夷、马尔代夫,都是度假时最想去的地方,墙上还布置了大幅的风景壁画。在这里开会应该不会感觉到特别累吧,呵呵,关毅心想。

在第一天剩下的时间里, 关毅开始接受 Peter 给他开的一对一的"小灶"。

Peter 专门在一个小会议室里向关毅介绍电子商务通的基本情况。虽然关毅之前就看过一些文档,不过还是缺乏感性的认识。

Peter 所在的产品组主要负责 E 公司的一个拳头产品,这是一个在线电子商务系统,名叫电子商务通。Peter 向关毅详细介绍了电子商务通的光辉历史:它的很多客户都是世界 500 强,每年为公司创造了非常多的利润,产品至今已经有 8 年的历史,一开始是由一家小公司研发并经营的,在前年被 E 公司收购,目前已经发布了 7 个版本,收购之后,E 公司统一了该产品原公司研发部门的组织结构和产品名称,包括版本号。关毅他们马上即将开发的是电子商务通 3.0。

关毅问 Peter: "目前这个产品在加拿大的开发团队有多少人?"

Peter 说:"目前有 6 名开发人员、6 名测试人员、2 名文档和界面人员,另外还有一些产品支持和售前售后人员。"

关毅很惊讶。这么成功的产品只有这么少的团队成员,真是人不在多而在精啊!

Peter 笑道: "我们之前就是小公司,合并到 E 公司之后才有了更多的机遇,可以借助 E 公司各方面的平台得到更好的发展。这不,你们也可以加入进来了。这在过去是无法想象的。"

关毅也告诉 Peter, 在北京的另外 3 名开发人员都是非常优秀的。Peter 听后非常高兴。

晚上回到酒店,关毅第一件事就是上网查询关于 Serum 的资料。趁这个机会, 关毅也想了解所有常用的敏捷开发方法。他想弄明白为什么 Peter 说 Scrum 是最流行 的敏捷开发方法。

很快,所有的资料都在网页中显示出来了,经过整理,关毅把以下内容复制到了一个 Word 文档里。



资料库

● 极限编程 (eXtreme Programming , XP)

极限编程的思想源自 Kent Beck 和 Ward Cunningham 在软件项目中的合作经历。在这里,"eXtreme"的意思是希望将软件开发过程中一些好的方法发挥到极致。XP 注重的核心在于"沟通、简明、反馈和勇气",用一句话来概括 XP 的这 4 个核心价值观就是:通过充分的交流和沟通,使产品的设计尽可能简单明了;同时通过客户经常性的反馈,生产出符合客户要求的软件产品,并且有勇气迎接需求的改变。

另外,极限编程者还总结出一系列经典的实践,形成了 XP 的 12 个主要实践方法,这些方法对极限编程具有指导性的意义,分别是:客户计划的制定、小版本发布、隐喻、结对编程、测试驱动开发、重构、稳定的进度、代码共享、编码规范、简单的设计、持续集成、现场客户。

• RUP (Rational Unify Process, Ratioanl 统一过程)

RUP 试图总结现代软件开发过程中所有好的实践经验,形成一种有很强适应性的软件开发过程。它包括了软件开发中的6大经验,分别是:迭代式开发、管理需求、可视化建模、使用基于组件的软件体系结构、验证软件质量、控制软件变更。



——敏捷开发故事

RUP 将软件开发周期按时间和 RUP 的核心工作流划分为一个二维空间,如图 3-4 所示。

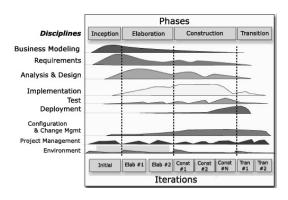


图 3-4 RUP 核心工作流

从图 3-4 中的横轴来看,RUP 把软件开发的生命周期划分为多个循环迭代,每个迭代生成产品的一个新版本,每个迭代由 4 个连续的阶段组成,分别是:初始阶段,了解项目的大致需求,建立业务模型,确定系统范围;细化阶段,设计、确定系统的体系结构,制定工作计划;构造阶段,构造产品并继续演进需求、体系结构、计划直至产品提交;移交阶段,完成产品的最终版本并交付给用户使用。

从图 3-4 的纵轴来看,RUP 的 9 个核心工作流分别是:业务建模、需求、分析与设计、实现、测试、部署、配置与变更管理、项目管理、环境。

RUP 的基本原理是:以满足客户需求、为客户创造价值为最终目标;尽可能早且不断地化解重大风险;把注意力放在可工作的软件上;在项目执行过程中尽可能早地适应变化;在项目早期设计、实现并测试一个可执行的架构;使用组件来构造系统;建立高效、协作的团队;要始终重视产品质量,否则追悔莫及。

以上 RUP 的基本原理构成了 RUP 的灵魂,在这些指导原则下,RUP 又给出了其他一些最佳实践,比如:为了化解业务风险,它始终以用例为中心;不是逃避变化,而是直面变化;为了化解时间风险,它采取迭代开发,尽量在早期探知到时间的延迟,以便采取更灵活的策略;为了化解技术风险,它强调架构设计;为了化解质量风险,它强调测试优先。而这些也恰好是 RUP 的主要特点。在这些最佳实践之后,才是具体的过程组织形式:具体由哪几个阶段组成?而每个阶段又包括哪些工作流,要产出哪些产品?

而且这些形式不是固化的。RUP 只是提供一个模板,可以在开发过程中进行裁减。

● Lean (精益软件开发方法)

精益生产的概念首先出现在制造业中,由日本的丰田公司提出。大规模制造理论认为,一定程度的浪费、一定程度的废品是正常的和被允许的。而在软件开发中,资源浪费、成本居高不下也同样成为软件开发的一大障碍。处于变革的十字路口的软件开发行业,总是能不断地从其他行业中寻找可借鉴的理论。这种借鉴来的思路就被称为精益编程(Lean Programming)。

Lean 方法的主要思路有:消除浪费,将所有的时间花在能够增加客户价值的事情上;延迟决策,在一个复杂多变的环境中进行软件开发,需要根据实际情况保持可选方案的开放性,但时间不能过长;尽早交付,软件交付的周期越快,用户的需求就会越清晰,软件应对需求变化的灵活性就越高,让客户的需求来推动工作的进展;加强学习,承认变化的存在及其不可预见性,加强反馈和交流,在实践中发现问题、解决问题,并最终形成解决方案;授权给团队,正确的决策取决于准确的信息,让开发团队参与决策,让团队成员充分发挥自己的潜力。

无数的经验和教训都已经证明,软件开发中一个巨大的浪费源头就是不注重质量而导致的返工。人们常常为了追赶工期而降低对质量的要求,殊不知这会带来更大的损失。 Lean 强调消除浪费,这正是为了避免低质量和返工造成的浪费。尽管这样做一开始看起来似乎有些麻烦,但它所带来的收益是实实在在的。

如果采用 Lean 方法,还要注意不要钻牛角尖。消除浪费并不意味着扔掉所有的 文档;尽量推迟决策并不意味着拖延决策,不能晚到错过了时机、耽误了工作才作出 决策;尽快交付并不意味着匆忙交付和马虎行事,否则会为日后的系统维护带来更多 的麻烦和浪费,这恰恰是与消除浪费的原则背道而驰的;授权给团队也并不意味着放 弃领导。

关毅收获不小:看来 XP、RUP、Lean 这些开法方法都是符合敏捷精神的。XP 偏重工程实践,缺少框架,好像不是很系统,没有有效地指导我们应该先做什么、再做什么。RUP 的框架太过繁琐,9 个核心工作流程的每一项市面上都有厚厚的一本书,让人望而生畏。尽管 RUP 也是敏捷,不过要想快速上手好像有点困难,倒是很多好的实践可以学习和借鉴一下。还有一些例如 Crystal、DSDM 等方法,用的人

——敏捷开发故事

都比较少,如果相信群众的话,就应该放弃。那么,Scrum 又是什么样子呢?为什么叫"Scrum"这么一个怪名字?它有什么魅力能够成为当今最流行的敏捷软件开发方法?关毅又开始了搜索。

Scrum 博士的第 1 次发言



大家好,我是关毅的个人数字助理 Scrum 博士,从现在开始我会帮助大家解释在本书中遇到的 Scrum 概念。别着急,慢慢你就会熟悉 Scrum。

Scrum 是一种灵活的敏捷软件开发管理过程,这个名词来源于英式橄榄球,如图 3-5 所示。Scrum 方法由 Ken Schwaber 和 Jeff Sutherland 提出,它将软件开发团队比作橄榄球队,全队有明确的最高目标——发布产品的重要性高于一切,团队高度自治,成员们熟悉开发过程中涉及到的各种技术,紧密合作,确保每个迭代都朝着最高目标推进,而且每隔 2~4 周,每个团队成员都能看到能实际工作的软件,并且据此决定是发布这个版本还是继续开发以加强它的功能。



图 3-5 Scrum 的起源

对于那些功能需求可能经常发生变化的项目来说,Scrum 是最为理想的选择之一。在一个采用 Scrum 的项目中,首先要将所有需要完成的工作列在一个 Product Backlog 中,项目开发过程中需求的改变也要写进去。在每个 Sprint 开始之前,要召开一个 Sprint 计划会议。在这个会上,产品责任人(Product Owner)为 Product Backlog 中的各项功能需求确定优先级。随后,Scrum 开发团队按照优先级,从 Product Backlog 中挑选出他们认为能在这个 Sprint 中完成的任务,并把这些任务从 Product Backlog 中挪到 Sprint Backlog 中去。在 Sprint 的进行过程中,Scrum 团队每天都要举行一个简短的每

日 Scrum 会议,以便团队成员了解开发进度。一个 Sprint 结束之后,需要召开 Sprint 评审会议和 Sprint 回顾会议。开发团队在 Sprint 评审会议上把这个 Sprint 的开发成果展示给大家。而在 Sprint 回顾会议上,团队成员们会回顾刚刚过去的这个 Sprint,从中总结经验和教训。

Scrum 的总体结构如图 3-6 所示。

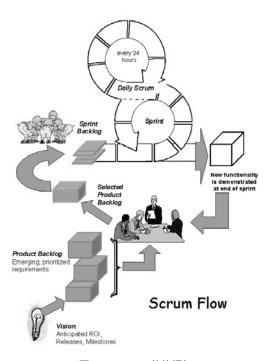


图 3-6 Scrum 整体框架

关毅合上电脑,准备休息了。他想,Serum 除了具有敏捷开发方法所倡导的那些共性的东西,它清晰简单的框架,实际可行的实践指导正是现在最需要的——列举任务、对任务排序、选择优先级高的任务进行工作、每日沟通、定期回顾和总结。

第二天一早,每日 Scrum 会议开始了。会议的地点在走廊的任务版前,就是昨天关毅看到的那个贴满小卡片的白板。关毅躲在一旁,认真观察每日 Scrum 会议是如何进行的。

维松Scrum之旅

---敏捷开发故事

只见电子商务通开发组的全体成员一字排开,面对白板站着,每个人开始轮流讲出昨天都做了什么,今天打算做什么,遇到了哪些问题。Peter 时不时地对问题进行分析。最后,所有成员说完,大家开始挪动小卡片了。在表格的横向上划分了每个任务不同的状态,分别是"未开始"、"被阻塞"、"进行中"和"已完成";在表格的纵向上划分了任务的所有者,即任务的所属人。每张小卡片就代表着一个任务,挪动小卡片就是为了改变它的状态,如表 3-1 所示。

 未开始
 被阻塞
 进行中
 已完成

 麦克

 拉里

表 3-1 Scrum 任务板

有人把小卡片从"未开始"那一栏挪到了"进行中",还有人把贴在"进行中" 栏的小卡片挪到了"已完成"栏中。当"已完成"栏中的小卡片越来越多,表明剩 下的任务越来越少,这样应该会使整个团队成员都感觉很有成就感。

20 分钟过去了。Peter 看了看表,摇摇头,对大家说:"今天开会的时间没有掌握好,某些问题我们讨论得太多了,每日 Scrum 会议我们一定要控制在 15 分钟之内,技术细节可以留在下面讨论。OK,散会。"

会议结束后,关毅问 Peter: "为什么还有很多小卡片不在表格中,而是放在了外面?"

Peter 解释说,"那些是下一个 Sprint 计划要做的,先贴到那里。"

"这么多的小卡片制作起来岂不很花时间?"关毅问。

Peter 笑着说道:"关毅,这是马上要交给你们的任务"。

关毅吓了一跳,难道要让我来做那些小卡片?那么多,连写带画的,岂不是要死人……

Peter 看着关毅满脸疑惑的表情,马上说:"趁你们还不忙,可不可以帮忙调研

一下现在的市面上的 Scrum 管理工具,这样也许我们产品组就再也不用做卡片了。"

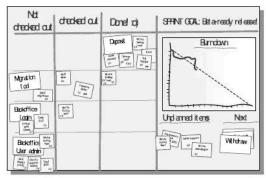
关毅欣然接受了这项任务,准备回去之后马上写信给远在北京的团队,让小段、 小胡和小余每人找一个敏捷开发的工具,并且在他离开加拿大之前写出使用报告。

Scrum 博士扫盲



• 任务板

任务板,说通俗点就是在一块白板上按一定规则张贴任务卡片。任务板的形式可以多种多样,不过通常的样子就像图 3-7一样,有点麻烦哦。



Kniberg, Henrik. Scrum and XP from the Trenches: How We Do Scrum. Version 2.1, Crisp, 5 Apr 2007.

图 3-7 Scrum 任务板实例

继续敏捷开发的话题。

关毅告诉 Peter 目前北京那边敏捷开发的情况。当 Peter 得知关毅他们已经接受了敏捷开发的培训,有点意外。

Peter 对关毅说: "太好了,我们也是刚开始试着采用敏捷开发,应该说正式采用还要到电子商务通 3.0 的开发,目前只是在一个小项目上试用。幸运的是管理层目前都很支持,这样,我安排一个会议,咱们一起讨论一下,也给你介绍一下我们目前的状况。"

两个小时以后,电子商务通的项目经理 Greg 和测试经理 Lisa,还有 Peter、关毅一起聚在了名叫夏威夷的会议室里。



首先由 Peter 介绍目前采用的敏捷开发流程。

Peter 说: "在即将到来的 3.0 开发中,我们准备采用 4 个星期的迭代。Sprint Backlog 争取能在 Sprint 开始的前一周制定好。关毅,我们计划设两个 Scrum 团队,北京和加拿大,这样我们就需要有两个 Sprint Backlog,你觉得怎么样?"

"那很好,毕竟我们离得很远,沟通起来不方便,两个团队好一点。"关毅说。 "不过,应该由谁来制定北京的 Sprint Backlog 呢?"关毅问。

"由于我们还不了解你们团队的能力,所以一开始由我们来制定,以后可以由你们来决定每个 Sprint 做多少个任务。" Peter 回答。

"那 3.0 的产品 Backlog 我们什么时候能准备好?产品开发计划是怎么样的呢?" 关毅接着问。

"呵呵, 3.0 开发计划正在筹备中, 我们会在另外一个会议中一起讨论产品 Backlog。今天是 3 月 10 号, 3.0 计划在 9 月 30 号发布。"Peter 回答。

"关毅,你们和北京电子商务通产品组的测试人员沟通方便吗?"测试经理 Lisa 问道。

"很方便,他们部门就在我们楼里,不过之前联系得不多。"关毅回答。

"太好了,我希望你们能加强沟通。之前北京的测试组和这边的开发团队由于时差和地理位置的缘故,沟通起来很不方便,效率很低,今后我希望北京开发的功能都能由北京的测试团队进行测试。"Lisa说。

"也许我们应该打破过去将测试和开发分为不同部门的惯例,也许按产品分更合适一些。" Peter 说道。

"Peter,采用敏捷开发是不是意味着你们开发组不用再让我们测试组等得很辛苦了?过去要等好几个月才能拿到可以测试的东西,临发布前由于Bug太多,害得我们也跟着一块儿加班。"Lisa不满道。

"呵呵,这是我们的目标。以后你们顶多等一个月就能拿到可以测试的软件,不过不要高兴得太早,我们可不敢保证你们会比以前更轻松。"Peter 笑道。

"我们是不是讨论一下今后如何沟通会更有效?"关毅问道。

"关毅,你有什么好的建议?我知道 E 公司北京研发中心和 E 公司美国总部合作的项目比较多,他们通常都是怎么沟通的?" Peter 问道。

"一般都是每周打一次电话沟通一下状态,仅此而已。不知道我们敏捷了是不是沟通应该更频繁一点?"关毅说。

"让我想想。你们的早上 8 点相当于我们的下午 5 点,不过我们早上 9 点上班,就相当于你们的半夜了。这样吧,我们每周先开两次会,一次是你和我参加,在周一开,主要沟通一些计划和安排、遇到的技术困难等。另一次全体成员的大会,在周五开,包括加拿大和北京电子商务通产品组的全体成员,主要是更新状态,你看怎么样?" Peter 回答道。

关毅点了点头: "OK, 就这么定了。"

总结和思考

在项目的开始制定一个有效沟的通计划至关重要,对于敏捷开发尤其如此。



接下来, Peter 又开始介绍敏捷开发具体流程的实施细节:"当拿到 Sprint Backlog 的时候,我们需要召开 Sprint 计划会议,把每个 Backlog 分解成具体的 Task,就是早上每日 Scrum 会议所采用的小卡片。每个 Task 以小时为计量单位,并安排到人。我们在目前的试用阶段经常是这么分解的,你们可以参考。比如完成一个用户登录校验功能的 Backlog,可以分解成调研校验功能所需技术(3 小时)、设计校验功能(4 小时)、开发校验功能(3 小时)、编写校验功能的单元测试用例(3 小时)、执行单元测试用例(1 小时)、Code Review(1 小时)、更新设计文档(2 小时)、代码重构(2 小时)……"

两个小时的会议很快就结束了,走出会议室的时候,关毅觉得心里有底多了,剩下的就是要解决在电子商务通 3.0 版本的开发工作中北京团队具体要做什么的问题了。

为什么刚才 Greg 发言很少呢? 关毅忍不住问 Peter: "Greg 具体负责什么工作?"



Peter 说:"他是我们的项目经理。由于我们正在试着采用敏捷开发, Greg 提倡团队自我管理, 所以你会发现他只要当团队需要帮助的时候才会出面。Greg 很了不起, 他是刚转到我们组的, 过去我们的项目经理总是喜欢发号施令, Greg 现在还相当于我们 Scrum 里面的产品负责人。接下来我们讨论产品 Backlog 的时候, 他会带着我们一起来制定。"

Scrum 博士扫盲



什么是产品 Backlog?什么是 Sprint Backlog?

产品 Backlog 指根据初始需求分解出的任务列表,包括功能性的和非功能性的所有功能,由 Product Owner 为 Product Backlog 中的任务确定优先级别,当开发团队开始某个任务的时候,再精确定义和分解这个任务。

产品 Backlog 是产品所要具备的所有功能的总纲。当一个项目刚刚开始时,没人能够事先预见到所有的任务和需求,并为之制定一个充分、详细而又包罗万象的计划。可行的方式是,先为一个项目写下所有它应该具备的显著特征和功能,数量不必很多,最好够让团队的第 1 个 Sprint 有活可干。

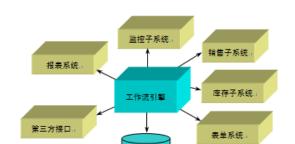
随着 Sprint 的进行,生产出可发布的产品增量,客户对产品的直观认识也会随之加深,他们可以据此建议更改或者添加产品 Backlog 中的任务。

在 Sprint 计划会议上,产品负责人为产品 Backlog 中的任务确定优先级,并向 Scrum 团队描述这些任务。Scrum 团队随后根据团队整体情况,确定他们能在这个即将到来的 Sprint 中要完成哪些功能,并把它们挪到 Sprint Backlog 中去。

下午,Peter 特意给关毅安排了一次关于产品技术架构的讲座,主讲人是 E 公司加拿大电子商务通产品组的架构师 John。John 的技术等级相当于全球副总裁,总之就是传说的"中牛人中的牛人"。

在 E 公司,管理和技术两条职业发展道路都可以走得很远,并且做技术可以像做管理一样成功。北京的 E 公司研发中心还很年轻,随着它的不断成长,在未来 10年,也将会涌现出一批技术大牛。

John 开讲了。



首先,他在白板上迅速画出了电子商务通的系统结构图(如图 3-8 所示)。

图 3-8 电子商务通系统结构图

数据库.

系统的核心是工作流引擎,相当于电子商务通的心脏,它使得电子商务通的业务流程是可定制的,这一点在市场上非常有竞争力。接下来,"心脏"连接着外围系统,如销售子系统、库存子系统、人员角色子系统、监控子系统、报表系统、电子表单系统等。

John 谈到电子表单系统的时候更是激情四射:"我们的表单系统可以动态地生成,完全基于 XML 和 xForms 标准,支持网络安全协议。表单是电子商务信息的重要载体,E 公司的系统可以帮助客户大大减少二次开发的成本,并整合已有的业务系统。"

John 接下来谈了自己对电子商务通未来发展的想法。John 目前对 Web 2.0 和 SOA 颇有研究,打算领导研发小组从 3.0 版本开始逐渐把电子商务通打造成一个基于 Web 2.0 和 SOA 的新型电子商务系统,给客户带来更好的体验和更大的价值。

John 说: "我们必须迅速反应市场的变化,技术也需要不断创新,否则随时都有可能死掉。"

关毅看着 John 潇洒的演讲,不禁陷入了无限的崇拜之中。E 公司是一个以创新为核心价值理念的公司,也是技术大牛们的天堂,他们在这里释放着巨大的能量,设计着一个个超酷的技术架构,指引研发团队一步步朝着目标迈进。对于他们而言,支配各种系统和代码,并不亚于指挥千军万马。



不过关毅心里一直在想,敏捷开发和传统的瀑布开发一个重要区别就在于设计环节。瀑布开发强调先由技术大牛们详细设计出要开发的功能,再由初级的程序员具体实现编码,至少关毅以前工作的 X 公司就是这样做的。在电子商务通 3.0 的开发里,北京团队会全体充当初级程序员的角色吗?于是关毅问 Peter:"3.0 具体的开发模式会是什么样的?北京团队也会参与设计吗?"

Peter 没有说话,好像在考虑什么问题。

关毅接着对 Peter 说: "Peter, 我知道你在担心我们的能力。没关系, 我们不妨按照敏捷开发的精神, 最好我们也能够参与设计。第 1 个迭代我们可以试试看, 如果不行, 再想办法, 可以吗?"

Peter 很高兴,说道:"关毅,很高兴你有这样的想法和愿望,如果北京团队能够独当一面,这是我们最高兴的。但是你们毕竟对产品不熟悉,需要有一个熟悉的过程。这样,你在这边多留几天,我愿意把我知道的都告诉你,回去之后你的担子可不轻啊!我向项目经理汇报一下你的这个想法,这可能会影响产品 Backlog 的选取,本来 3.0 我们是打算主要由我们来设计,你们主要来编码的。"

关毅很高兴自己能为团队争取到一些机会, 也感谢敏捷开发的推行。

很快,项目经理同意了关毅的请求。敏捷开发强调自我管理的团队,就必须充分信任开发团队能漂亮地完成任务。

产品 Backlog 的制定

又过了一天,终于到了正式讨论产品 Backlog 的时候。这次会议的参加者有产品负责人 Greg、架构师 John、Scrum Master Peter,以及电子商务通产品的市场负责人 Leo、技术支持负责人 Paul,当然还有关毅。把这么多人召集在一起的目的只有一个,就是要按照敏捷开发的理念设计出电子商务通 3.0 的产品 Backlog,并确定它们的优先级。

Scrum 博士扫盲



Scrum 中有 3 种角色,分别是产品负责人(Product Owner), Scrum Master 和 Scrum 团队,他们各自的职责如下。

● 产品负责人 (Product Owner)

Product Owner 需要确定产品的功能和完成时间,并对产品的收益负责,要根据市场需求确定产品功能的优先级。在每个 Sprint 开始之前,Product Owner 可以修改功能需求和优先级。而且,Product Owner 有权决定接受或否决各个 Sprint 的工作成果。

Product Owner 的角色通常由市场部门的人员或开发部门内部主要使用该产品的人员来担任,主要工作是根据市场需求确定产品功能,将其列入 Product Backlog 中,并为这些功能确定优先级。

Scrum 团队按照功能的优先级,将它们从高到低分配到各个 Sprint 中进行开发,这些被分配到一个 Sprint 中完成的功能就形成了 Sprint Backlog。

在产品的整个开发过程中,Product Owner 对于产品的需求可能会发生改变。他可以修改 Product Backlog,以及增加某些功能需求、删除某些功能需求、修改优先级等,但这些行为只能在各个 Sprint 之间进行。

Scrum Master

Scrum Master 的职责是:负责监督整个 Scrum 项目进程,调整项目计划;确保开发团队成员的能力能够胜任产品的开发;促进团队中不同角色的成员间充分交流和沟



通,并负责为项目的进行扫除障碍;保证开发团队不受外力的干扰和阻挠;掌握产品开发进度,参与每日 Scrum 会议、Sprint 计划会议和 Sprint 评审会议。

Scrum Master 通常由传统开发中的项目组长 (Team Leader)来担当。

• Scrum 团队

一般由 5~10 个能全职工作的成员组成较为理想。

团队成员横跨各个职能,通常包含开发、测试、文档设计人员等。

来自市场部的 Leo 谈了一些目前产品的销售情况和市场方面最迫切的功能需求。比如,欧洲的一些大客户使用苹果电脑,希望 E 公司的电子商务通能够在 Safari 浏览器中运行。另外,还有一些客户对打印报表和表单票据的要求特别高,现有系统的打印功能满足不了需求。

负责技术支持的 Paul 谈了客户最抱怨的性能问题。

John 谈了可以利用目前最新的 Web 2.0 技术框架使产品上一个大台阶。

Greg 从内部角度入手,表示自从并入 E 公司后,E 公司的其他产品部门也有迫切地希望与电子商务通集成的需求。但由于电子商务通曾经是独立的产品,所以目前和 E 公司各大产品集成的能力还比较弱。

然后, Greg 总结各方意见, 加上之前已经通过 Wiki 向全体员工和一些客户征集过的最想要的功能列表, 所以这次讨论会的效率比较高。大家淘汰了一些优先级明显比较低的 Backlog, 剩下的就是决定优先级和 Story Point 了。

Story Point 就是每一个 Story 的复杂度, Story Point 的计量单位是人/日,是理想状态下一个人一天能够完成的工作量。Story Point 都是由 John 一个人决定的。关毅不知道这是不是符合敏捷的精神,不过也没敢说什么。这个时候,自己毕竟还是个新手。

会议的最后,决定要做多少个 Backlog 完全取决于电子商务通 3.0 的计划发布时间。根据市场策略,电子商务通 3.0 计划在 8 月 30 日发布,所以要在 4 月 1 日正式开始研发。整个研发过程一共有 5 个月的时间,前 4 个月计划都用来进行迭代开发,

准备 Scrum 之旅

最后一个月用来修复 Bug。虽然产品组计划采用非瀑布式的开发方式,不过大家还是一致认为有必要预留 1 个月的时间进行最后的集中测试和 Bug 修复。毕竟是第一次采用敏捷开发,在 1 个迭代中能否开发完所有的功能并完成所有的测试和 Bug 修复,大家多少有些心里没底。

经过一个上午的激烈讨论, Backlog 出来了。拿到北京团队即将负责的 Backlog, 关毅有点激动, 因为北京团队也负责了一些非常重要的功能(如表 3-2 所示)。

编号	标 题	描述	优先级	Story Point
010100	新 Ajax 框架	用户可以通过 Ajax 框架系统实现 所有的 Web 应用,并达到桌面级的 用户体验。 http://e.tech.spec/wiki/	1	40
020100	PDF精确打印表单功能	用户可以精确打印电子商务通系 统中任何可操作的表单,要求达到 像素级的匹配。 http://e.tech.spec/wiki/	3	20
011000	电子商通 API (工作流部分)	用户可以通过 Java 和 JavaScript 的 API 与其他产品进行集成。 http://e.tech.spec/wiki/	4	40
040100	苹果 Safari 浏览器支持	让用户能够使用苹果操作系统 (Mac)上的 Safari 浏览器访问电子 商务通,使电子商务通完全兼容于 Safari 浏览器	2	20
030100	Web 2.0 工作流设计器	用户可以完全通过 Web 进行电子 商务流程的设计	5	40

表 3-2 北京团队的 Backlog

关毅注意到, Backlog 表中的编号实际上是 6 位组成。Peter 向关毅解释,编号是帮助把 Backlog 进行分类的,前两位代表主题(Theme),比如 "010100 新 Ajax 框架"和 "011000 电子商务通 API(工作流部分)"都属于系统主题,所以前两位编号都是 "01","040100 苹果 Safari 浏览器支持"属于平台相关功能,所以前两位编号是 "04",中间两位为功能属性,最后两位是具体的 Backlog。这份 Backlog 还需要进一步分解为更为详细的 User Story。为了让关毅了解更详细的需求,Peter 告诉关毅不用着急,到 Sprint 计划会议的时候会安排专门的会议,产品负责人 Greg 届时会帮助



整个北京团队制定更详细的产品 Backlog,每项都会是 1 个 User Story,最好是全体成员都参加。

总结和思考

通常在敏捷开发中,制定的产品 Backlog 不是一成不变的。敏捷开发中的每个 Story 都要求是由市场和客户驱动的。要注意的是,User Story 和 Use Case 是两个不一样的概念。Use Case 指的是用例,通常用 UML 图来描述,也是用来描述需求的很好的工具,已经有很多年的历史。Story Point 的计量在一开始不可能做得很准确,只能做一个大概参考,而且由架构师一个人决定也是不妥的。

Scrum 博士扫盲



User Story

Sprint Backlog 里的项目我们通常用 User Story 来描述, User Story 是从用户的角度对系统的某个功能模块所作的简短描述。一个 User Story 描述了项目中的一个小功能, 以及这个功能完成之后将会产生什么效果, 或者说能为客户创造什么价值。

User Story 要由 Stakeholder 来编写。User Story 的形式很简单,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 User Story,而不是开发人员。

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。而且,Stakeholder可以随时更改一个 Story 的优先级,那么此时开发人员就应该相应地调整 Story 的开发次序。

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。如果 User Story 太大,可能会导致对它的开发横跨几个 Sprint,这种情况是需要避免的,此时就应该将这个 User Story 分解。

User Story 有一个通用的公式格式,大家可以套用一下试试,很简单。

作为<某个角色>,我可以<做什么>,以完成<什么目的>。

准备 Scrum 之旅

例如:作为一个病人,我可以预约一个医生,让他给我看病。

这种表达方式清晰明了,提供了足够的信息以供测试。更详细的实现细节会在要完成这个 User Story 的 Sprint 开始之前确定下来,并补充到 Sprint Backlog 中去。这是一种把客户需求分解为可测试的且有优先级的任务的有效方式。

为了能及时、高效地完成每个Story Scrum 团队会把每个Story 分解成若干个Task。 每个 Task 都是可以在明确的时间内完成的,而且时间是以小时为计量单位的。

特别提示:每个 Task 的时间最好不要超过 8 小时,就是要保证在 1 个工作日内完成,如果做计划时发现有些 Task 的时间超过了 8 小时,就说明 Task 的划分有问题,需要特别注意。

Peter 看出了关毅的兴奋,好像恨不得马上就要开始工作,于是他对关毅说:"呵呵,关毅,不要着急,我还有好多课要讲给你呢, John 的系统架构只是开了个头。我们的电子商务通里面的技术够讲一年的,以后更多的还要靠你们边做边熟悉。"

从那天下午开始,Peter 每天都会花一定的时间给关毅讲解系统的各个部分。关 毅本来学东西就很快,再加上浓厚的兴趣,进展十分顺利。



Scrum 管理工具

在关毅去加拿大的这段时间里,远在北京的团队对 Scrum 工具的调研工作也进展得很顺利。小段很厉害,首先带头在 Google 上搜索到了最常用的几个工具,包括 ScrumWorks、Rational Team Concert、XPlanner。于是他们做了分工,每人负责调研 1 个。几天后,关毅就收到了调研报告。



资料库

ScrumWorks

ScrumWorks 是一个专门针对 Scrum 项目管理的商业软件,详见 http://danube.com/ScrumWorks,专业版的价格大概是每用户每年几百美金,软件的授权费采用固定的期限,到期还需要继续购买。ScrumWorks 有桌面和 Web 两种界面,都是基于 Java的,服务器的安装需要 JBoss 的支持。Product Owner 和 Scrum Master 通常使用桌面应用程序入口,而普通团队成员则可以使用 Web 应用程序入口。ScrumWorks 的桌面应用程序入口界面如图 3-9 所示。

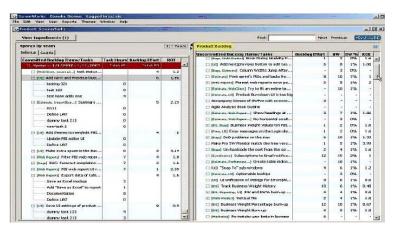


图 3-9 ScrumWorks 的桌面应用程序入口界面

如图 3-9 所示,右边部分记录了所有的 Backlog,左边是每个 Sprint 对应的产品 Backlog 列表。每个 Backlog 都可以用 User story 来描述,如图 3-10 所示。



图 3-10 User Story

针对每个 Backlog, ScrumWorks 提供了丰富的属性,比如商业权重(Business Weight)投资回报率(Return On Investment, ROI)收益(Benefit)惩罚(Penalty)等。根据这些属性可以帮助制定 Backlog 的优先级。

当制定好了产品 Backlog 以后,制定 Sprint Backlog 就非常方便了,只要把 Backlog 从窗口的右边列表拖到左边即可,如图 3-11 所示。

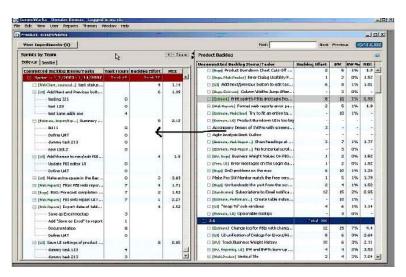


图 3-11 制定 Sprint Backlog



对每个 Backlog 可以进一步分解成任务,层次结构在软件中一目了然。

ScrumWorks 提供了较强大的可视化报表功能,整个产品的 Burndown Chart 如图 3-12 所示,其中,虚线为每个迭代完成 Backlog 的下降趋势线,实线为 Backlog 增加/删减的趋势线,这样,两条线的交点所在的时间就是预测出来的产品交付时间。

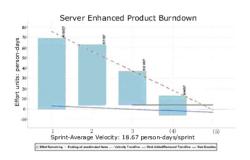


图 3-12 ScrumWorks 的 Burndown Chart

对于整个团队的日常工作,ScrumWorks 提供了方便的 Web 界面,所有 Backlog 和任务展现形式完全模拟任务板,如图 3 - 13 所示。

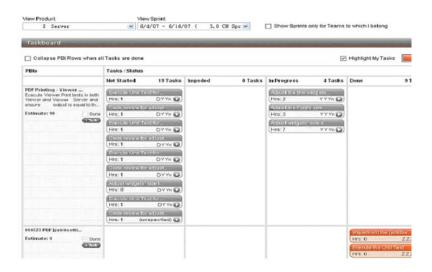


图 3-13 ScrumWorks 中的任务板

团队成员可以在任务板中选择一个自己要做的任务,并每天更新状态。每个任务都有4种状态,即未完成、被阻塞、进行中和已完成。根据当前所有任务的状态,可以实

准备 Scrum 之旅



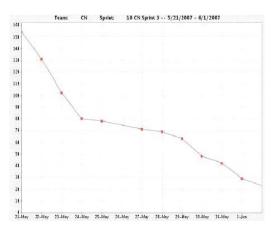


图 3-14 Sprint 的 Burndown Chart

• IBM Rational Team Concert

IBM Rational Team Concert 是一个十分强大的开发协作平台,属于商业软件,详见 http://jazz.net。Rational Team Concert 是基于一个名为 Jazz 的 IBM 开源平台开发的,Jazz 本身是基于 Eclipse 的,而 Eclipse 本身是跨平台的,所以 Rational Team Concert 给用户在不同平台上带来的体验是相同的。Rational Team Concert 最强大的地方在于可以轻松地与 IBM Rational 软件配置管理工具进行集成,比如著名的代码管理工具ClearCase、软件缺陷管理工具 ClearQuest 和 Build 管理工具 BuildForge 等。Rational Team Concert 自身也带有一个代码控制管理(Source Control Management)工具。

由于采用了 Eclipse 平台 ,所以开发团队甚至可以把 Rational Team Concert 也作为一个开发 IDE。这样,整个软件开发的生命周期,例如编码、代码控制管理、测试、管理、调试、Build 集成等,都可以在一个系统中完成,实现 All-in-One,非常方便,如图 3-15 所示。

Rational Team Concert 对 Scrum 支持功能的也很强大,可以建立不同的 Scrum 团队,为每个 Scrum 团队制定不同的 Sprint Backlog,每个 Sprint Backlog 也提供了丰富的属性。另外,Team Concert 支持把每个 Backlog 项进一步分解成任务,并分配到人。Team Concert 还可以生成大量的报表,如图 3 - 16 所示。

---敏捷开发故事

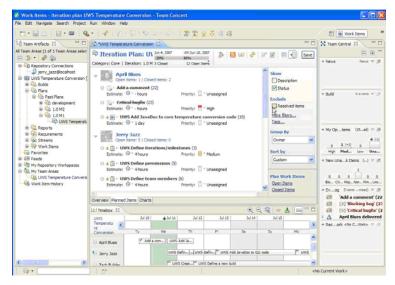


图 3-15 IBM Rational Team Concert 界面

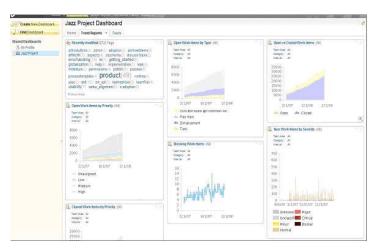


图 3-16 Team Concert 生成的报表

在 Sprint 视图中有一个小柱状图 ,可以一目了然地提示团队当前的工作进度是否比计划落后了 , 如图 3 - 17 所示。



图 3-17 团队工作进度

准备 Scrum 之旅

在图 3-17 中,右上角柱状图中的黑色部分表示团队目前落后于计划进度的工作量。

Rational Team Concert 提供了较为强大的历史记录功能,可以查看过去所有的 Backlog 和任务的变更记录,这个功能可以很好地跟踪团队进度。

当然,Rational Team Concert 还属于比较年轻的产品,不能像 ScrumWorks 一样为用户提供一个更为直观的任务板。不过,只要团队平时就习惯在 Eclipse 平台下工作,用 Eclipse 的视图展示 Backlog 和任务,接受起来应该也还算自然。

• XPlanner

XPlanner 是一种针对敏捷开发,尤其是 XP 的项目管理工具,详见 http://xplanner.org。不过,XPlanner 对 Scrum 也同样适用。XPlanner 本身是一个轻量级的开源项目,且完全免费。虽然它是基于 Java 的 Web 应用,不过安装起来也很容易。用户只要下载一个自带 Tomcat 和数据库的独立运行包(standalone),解压后执行"bin"目录下的 startup.bat 文件,就可以启动 XPlanner 服务器,然后,打开浏览器访问服务器地址就可以使用了,非常方便,如图 3-18 所示。

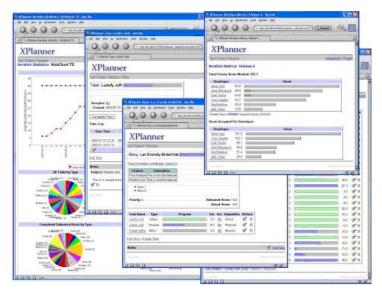


图 3-18 XPlanner 的操作界面

XPlanner 的计划模型比较简单,支持迭代开发。在项目开始的时候,首先建立一个



迭代,然后可以在迭代里面建立 Story,以小时为评估,每个 Story 可以进一步分解成任务,每个 Story 和任务都可以分配到人。XPlanner 也有一些迭代统计和监控功能,可以根据当前的进度情况生成柱状图,甚至还支持 Scrum 里面的 Burndown Chart。

不过, XPlanner 给人的总体感觉比较简单, 比较适合非常小的团队。

总结和思考

工欲善其事,必先利其器。找到一个合适的敏捷项目管理工具并不难,就算没有,Excel 也是一个很好的替代工具。



Peter 和关毅对调研的结果都很满意。由于 ScrumWorks 中的"任务板"功能最接近他们现在的工作模式,所以 Peter 决定让整个团队先试用一下 ScrumWorks。这意味着在接下来的 Sprint 里, 所有的 Scrum 数据北京和加拿大都可以通过 ScrumWorks来进行管理并实现共享。

在加拿大的日子每天都很充实, 关毅有点乐不思蜀了。

不知不觉已经过去两周了, 所有的任务都已经圆满完成, 关毅必须要回国了。

临行前,Peter 和整个电子商务通的团队请关毅吃了顿大餐。关毅知道西方人一般是很少请人吃饭的,所以对此十分感动。的确,彼此间在短短两周内建立起了深厚的友谊,想必将来的合作也一定会非常顺利。最后,关毅也盛情邀请Peter 和他的团队有机会到北京做客。

飞往北京的班机起飞了,关毅和他的 Scrum 敏捷开发之旅也即将开始了。



Sprint 1

——激动人心的 Scrum 之旅第一站

充电完毕的关毅带着他的团队满怀热情地投入到了第一个 Sprint 的开发工作中,其中有一系列的实践,包括 Sprint 计划、每日 Scrum 会议、Sprint 评审以及 Sprint 回顾等。但是新手就是新手——他们遇到了很多意想不到的问题。

就在紧张忙碌的 Sprint 1 总算要结束的时候,哪知道最后评审的结局也很意外。但是,这些经验和教训都为今后的成功指明了方向。



Sprint 1 计划

回国和大家见面之后,关毅做的第一件事就是了解了一下他们的状况,结果令 他非常满意。于是,趁热打铁地开展工作就是顺理成章的事情了。

关毅给团队成员们进行了一次速成的培训,从什么是 Serum 讲到产品技术概要, 花了大半天的时间。

为了尽快开始 Sprint 1 的工作, Sprint 计划刻不容缓。

不过一提到 Sprint 计划,关毅心理就打起了小算盘,段美女他们对要做的产品毕竟不熟悉,恐怕很难在任务的分解及其开发时间的估计中给出太多的帮助,不如这次的计划就由我一个人来做吧,既节省时间,又能保整质量。Agile 嘛,本来就是强调随机应变!

总结和思考



由 Team Leader 一个人来进行 Sprint 计划合适吗?

关毅开始了 Scrum 之旅。

投掷"照明弹"

作为一个 Serum 新手, 关毅第 1 次制定 Sprint 计划的最好办法就是按照理论步骤一步步来。其实很简单, 先确定 Sprint 的目标。

应该如何制定 Sprint 计划呢? 既然现在所有的产品 Backlog 都已经有了,那么从 所有的 Backlog 里面选出优先级最高的就可以了。

关毅从头到尾看了一遍,仔细地研究了一下,优先级最高的是 Ajax 框架。

关毅打开了 Wiki 上的需求说明。



Sprint 1 开发任务

需求概要

电子商务通是一个成功的电子商务平台,但是随着 Web 应用技术的不断发展,要想使电子商务通继续领先于我们的竞争对手,就必须在用户操作体验、产品易用性上有所突破。我们可以通过 Web 2.0 达到客户端应用程序级别的用户体验,使用户享受简单、快捷的服务。

开发方案

为了能提高电子商务通的易操作性和用户体验,我们要重写整个前台逻辑,要求采用最新的 Web 2.0 的 Ajax 技术实现前台界面的动态交互。我们希望能够提供一个可扩展的 Ajax 框架,方便以后新的业务流程的开发,或者用户的二次开发。

从 Wiki 上的需求看,采用 Ajax 技术重写原先的前台交互逻辑是现在工作的重点,而同时最好的办法是先实现一个 Ajax 框架作为整个前台系统的支撑。但是这里有个问题,关毅实在是对 Ajax 框架没有特别深入的了解,一时间也搞不清楚在 1 个 Sprint 里能不能完成。但是,如果告诉温哥华那边,中国的团队在 1 个 Sprint 里面连 1 个 Story 都完不成,那还不让人笑掉大牙啊!

总结和思考



如果真的遇到一个特别大的 Story , 应该适当地将 Story 分解。



Scrum 博士扫盲

• Sprint 计划 (Sprint Planning)

在每个 Sprint 开始之前,需要召开 Sprint 计划会议,会议时间一般为 4~8 小时,参加人员有产品责任人、Scrum Master、Scrum 团队和其他感兴趣的人,比如管理层人员和客户代表。Product Owner 从产品 Backlog 中挑选高优先级的任务,并与 Scrum 团队一起决定在这个 Sprint 中需要完成多少功能。Scrum 团队将这些任务分解成小的功能模块。Scrum 团队成员详细讨论如何能按需求完成这些功能模块,并估计完成每个功能模块所需的大概时间。



关毅在自己的座位上有点儿着急,想找个人请教一下。可是看看周围别组的同事,一个个都忙得不可开交,实在不好意思去打扰他们。

就在关毅抓耳挠腮的时候,突然发现自己有封新的 E-mail, 打开一看,恰好是敏捷培训老师 David 的回访邮件,他希望大家把目前敏捷开发的进行状态和疑问作一个反馈。

看到这里,关毅灵机一动,心想:真是谢天谢地,救星来了!他二话不说就给David回复了E-mail,而且特别提到了自己面临的困难:如果采用Serum,当遇到了我们从未接触过的工作时应该如何去做计划,如何安排Sprint的工作,特别是如何精确地计算所有Sprint任务所需要花费的时间。

E-mail 发出去了, 关毅心里稍稍地平静了一点。

当他正想整理一下在温哥华接受培训的资料,然后给自己的团队也进行一次培训时,徐天走过来叫他去参加一个每周的例会,这是只有 Team Leader 和经理参加的例会,主要是回顾一下上周的工作,每个 Team Leader 都要介绍一下自己 Team 的工作状况和下一步的工作计划。

关毅硬着头皮开完了会,就迅速地跑回自己的座位,还没来得及反思一下自己在会上的表现,就收到 David 的回信了。

关毅:

我看过了你的邮件,大致上了解了你目前的状况。的确, 当我们进入一个新的领域或者开始一些从未涉及过的工作时,

感觉无从下手是很正常的,就像我们走在一条漆黑的山路上,我们不知道前面的情况,甚至连这条路是否能到达山顶都难以确定。而我们一般会怎么做呢?点起火把,拿起手电,摸索着前行。只是,这样的话,我们的视野只局限在眼前,如果我们走错了,等到发现的时候,可能已经无法回头,接下来的命运,我想不用我再赘述你也能够想象。所以更好的是,我们可以打一颗照明弹,尽可能地往远处打,说不定就能发现我们正在走的路是否能通往我们的目标。同时,对于路上的困难,我们也会有一个感性的认识。

敏捷开发的价值观认为计划是不可能精确估计的,所以你也没必要担心。

这也是习惯了传统流程的人再转到敏捷开发时会不太适应的地方。但是慢慢 地你就会体会制定一个粗略的计划反而更符合实际,它可以帮助我们把重点 放在更有价值的事情上。

希望我说的这些会对你有帮助。

David

虽然接受过敏捷开发的培训,可还是不自觉地回到老路上来,关毅看完后有点 惭愧。

过去在 X 公司里,关毅经常要为自己的开发小组制定无比详细的开发计划,计划里虽然有无数不确定的因素,但是一但定下来就必须严格执行,计划逐层上报,然后汇总。最后的产品发布对外宣称的日期 (Deadline) 也就是这样产生的——这个 Deadline 不知道害了多少人。

想到这里,关毅不愿意再继续想下去了,因为这会勾起他痛苦的回忆。David 给出的"照明弹"的建议非常好,关毅很受启发。

关毅想: 既然电子商务通的 Ajax 框架是所有工作的基础, 那就把它放在 Sprint 1 里去完成。而且, 为了能把这个 Sprint 做成功, 我们需要一颗"照明弹", 这就作为 Sprint 1 里优先级最高的 Story 吧。通过这颗"照明弹", 我们一定要搞清楚实现电子商务通的 Ajax 框架需要做些什么!

关毅在心里暗暗地盘算着,要搞定这个工作,首先一定要了解电子商务通的现有系统,同时还要去了解已经被开发出来的成熟的 Ajax 框架中有什么是我们可以借鉴的。

总结和思考



在混乱中建立秩序是 Sprint 开始阶段的目标。

关毅下定了决心,一定要来个漂亮的开门红。

目标已经明确,下一步的计划也就顺理成章,关毅有些迫不及待地打开了 ScrumWorks,做起了Sprint计划。



需求分析——买土豆的故事

关毅以"迅雷不及掩耳盗铃"之势完成了 Sprint 1 的计划,如图 4-1 所示。



图 4-1 Sprint 1 计划

关毅长出了一口气,看了看时间,才用了两个小时。他对自己的工作还算满意, 美中不足的地方就是现在还无法细化随后的两个 Story。

这样美了一会儿,关毅也暗地里有些难为情了。同时,他也从这种自我陶醉中醒了过来。计划是做好了,但是这个计划做得是否符合 Scrum 的要求,他心里可是一点也拿不准。

关毅再次想到了 David。他按照电子邮件上的联系方式拨通了 David 的电话: "Hello,是 David 吗? 我是关毅。"

"Hello,关毅,你好吗?"电话里传来 David 热情的声音。

"我很好,谢谢你给我回信,我们已经开始考虑加入'照明弹'了。然后,我把 所有的任务都按照 Scrum 的要求进行了任务分解,还有什么需要注意的吗? Sprint 计划这样就算完了吗?"关毅说。

"关毅,你听说过买土豆的故事吗?故事说的是两个同时进公司的员工小甲和小乙,小甲晋升很快,不到一年就已经从普通职员升到了项目经理,而且已经是公司的重点培养对象,前途一片光明,小乙却一直在原地踏步。对此小乙很不服气,于是就找他的老板抱怨说,公司对他不公平。而他的老板并没有正面回答这个问题,只是把小甲也叫了过来,吩咐他们两个去买点土豆回来。小乙二话不说,径直就跑

了出去,5分钟不到,他就急急忙忙地跑了回来,气喘吁吁地问老板,土豆有1块钱一斤的和1块5一斤的,要哪个?老板说你自己看吧。小乙又急忙跑了出去,而且这次回来得更快,他上气不接下气地问老板,刚才忘了问您了,要买多少斤啊?老板笑了,你在这里等等吧,看小甲是怎么做的。就这样,他们等了1个小时。小甲不急不忙从外面走了进来,对老板说,我去了离咱们公司最近菜市场,看到那里的土豆有1块钱一斤的,但是个头小、卖相差,虽然1块5一斤的个头大、卖相好,但是我问了一下他们的货源,他们都是在稍微偏僻一些的一个蔬菜批发市场里进货,我当时就想,说不定到那里能买到更好、更便宜的土豆,于是我直接去了那个蔬菜批发市场,果然,发现之前要1块5一斤的土豆只要1块2一斤,我又和卖土豆的砍了砍价,他答应1块1卖给我,不过我走的时候有点急,忘了问要买多少,于是我就把那个卖土豆的叫到咱们公司楼下,老板,要不您和我一块下去看看?关毅,如果你是这家公司的老板,你觉得小甲和小乙他们谁做得好啊?"

David 话音刚落,关毅就接上了话:"这个故事我好像听过,这个其实是让我们 多站在老板和客户的角度上思考问题,同时要对各种情况有所准备。"

"是啊,关毅,你说得很对。不过你不觉得这个故事有问题吗?故事里的成功者居然只是通过老板的一句话,就替老板假设了那么多的情况,而且所有的假设都没有得到老板的反馈意见,就直接替老板决定了,这岂不是很危险的事?但是你可能会说,总不能每次遇到需要抉择的时候都去找老板请示吧。是的,我们当然不能这样。所以呢,我们要在事情没做之前,从老板那里收集到尽量多的信息,例如买土豆是要做什么啊、要买多少啊。这样我们才能够迈出坚实的第一步。好的开始等于成功的一半。关毅,从你的 E-mail 中我没法了解你们和 Product Owner 有多少沟通,但是一定要充分、有效。" David 说。

总结和思考



买土豆的故事值得反思。小甲真的做得好吗?

听到 David 的这一席话,关毅频频点头:"David,我明白了,谢谢,太感谢了,我知道你的意思了!我会尽快和温哥华再沟通确认的,我也会保证我的团队成员都会理解这些需求的来龙去脉。我一定要把这些记录下来,作为珍贵的文档保存好的。"

"我的建议是,文档是必须要有的,但是不要把它看得太重要了。文档只是沟通

雅松Scrum之旅

的一种形式,而且还不是最有效的,一、两页纸就够了。Serum 推荐大家用最直接、最简单的交流方式——口头讨论——因为这是效率最高的方式。"David 说。

"David,这个我明白,培训的时候都讲了的。不过我还是有点不理解,如果需求之类的内容都不用文档记录下来,到时候大家忘了怎么办?还有,如果团队成员有调整,假如没有详细的文档,又怎么让新来的成员快速上手呢?"关毅问。

"这样的话,你就需要好好回想一下,在过去的项目当中,哪些详细的文档最终排上了用场呢?" David 说。

关毅的眼前像过电影一样,闪过从前一幕幕的场景。这样的例子太多了。

曾经有一次,他维护一个详细的客户需求文档,可谓面面俱到。后来客户的需求一直不停地改来改去,因为公司对文档的要求很高,要求文档的内容要非常详细,结果把团队里一个最有耐心的弟兄小 A 都弄烦了,最后索性不改了。可奇怪的是,大家一直都有没发现,其实谁也不看这个文档。还有一次,来了一个新人小 B,他的第一项工作就是看文档。小 B 看了两天,关毅一检查,非常生气。因为小 B 看的文档是旧的,内容都是错的,连问都不问一下。当时弄得小 B 也很委屈,心想,我怎么知道文档是错的。

想到这里, 关毅说: "David, 也许的确如此, 谢谢你的建议……"

David 听到这些,呵呵笑了一声:"关毅,以后有什么问题想问我们再一起交流吧,我今天还有些事情要忙,咱们就先聊到这里吧,再见!"

总结和思考



敏捷开发倡导面对面的交流。可以工作的软件胜过面面俱到的文档。

关毅放下电话,静下心来,仔细地琢磨着 David 的话。

要想尽可能早地得到反馈,最直接的办法就是尽快和温哥华沟通一下。但是, 北京和温哥华有 16 个小时的时差,如果现在给他们发电子邮件,得到回复最快也要 明天早上了。那么今天这一整天,我们是该等待,还是先开始 Sprint 呢?如果什么 都不做而只等着反馈,这也太愚蠢了。但是如果明天得到的建议是要大范围地修改

之前的计划,有很多工作可能就白做了。最好的情况是只有一天的工作是白做的,如果这样的情况多了,可是会非常打击团队的士气的,更加严重的问题是不可避免 地会降低自己的威信。

关毅忽然发现这似乎是个无解的难题。然而更令他难以理解的是,他居然感觉有些兴奋——这是猎手遇到猎物时的兴奋,这是对自己自信的表现。

关毅重新浏览了已经做好的 Sprint 计划,对于目前的 Team 来说,当务之急就是要尽快熟悉电子商务通这个产品,分析已有的 Ajax Framework,尽快提出一个实验原型,完成"照明弹"的工作。这个任务可以说是整个 Sprint 1 的基础,也是最重要的一点,而且这个工作一定不会因为计划的改变而受到影响。在完成这个任务的过程中,完成与温哥华的沟通应该不成问题。

想到这里, 关毅拿定了注意, 他给团队成员发了一封电子邮件。

Team:

我们的 Sprint 1 明天就要正式开始了,我已经在 ScrumWorks 上创建了这个 Sprint 需要做的 Story。同时,我也把 Story 分解了 Task。我个人认为,在所有的 Story 中,优先级最高的就是为电子商务通作出一个可用的 Ajax 框架,而这个 Story 我们可以再细分成若干个 Task,例如调研已有的 Ajax Framework、理解电子商务通原有 Client 端的交互逻辑等。而且按照 Scrum 的精神,大家可以根据自己的特长或兴趣自行选择这些 Task 开始各自的工作,大家意下如何?

我们明天尝试第一次每日 Scrum 会议(Daily Scrum)。我们的每日 Scrum 会议时间就定在每天下午 4 点半,地点固定在茶水间。

关毅



初尝每日 Scrum 会议

关毅一边整理着从温哥华带回来的资料,一边不时地看看 ScrumWorks 上的更新情况。对于 ScrumWorks,关毅不禁感叹:有这样工具真是太好了!虽然只用了一天,关毅已经觉得离不开了。有了它,总是给你可以操控全局的感觉。从 ScrumWorks 的任务白板上,关毅看到小胖、小胡和段美女都已经选择了自己感兴趣的 Task (如图 4-2 所示)。



图 4-2 Task 分配结果

一切看上去似乎都很顺利,可是也带来了另一个问题,就是有几个非常重要的 Task,例如调研和实现电子商务通的 Server 端业务处理逻辑以及 Code Review 无人问 津。这也不能怪他们,大家都认为这个 Task 非关毅莫属,因为他刚刚从温哥华培训 回来,对电子商务通应该最了解,工作起来一定是事半功倍,而且关毅又是 Team Leader,Code Review 的最终结果一定得由他来拍板,所以,大家都很知趣地把这些 Task 留给了他。不过,关毅的想法却和他们有些出入。关毅原来的想法是在 Sprint 1 开始之前尽量再找个时间给大家来一次更详细的电子商务通的技术培训,而且,关 毅是想把这个任务加到 Sprint Backlog 里的,只是他从心里面觉得培训的事情不能算 真正工作的一部分,是可以挤出时间来做的,特别是关毅认为 Scrum Master 是不能 把过多的精力放在特别具体的开发任务上的,而要用更多的精力去控制整个项目的 进度,负责各种资源的调配以及对项目进行中出现的异常情况的处理。鉴于以上的 考虑,关毅觉得需要和大家讨论一下这个问题。他低头看了看时间,已经是下午 4 点钟了,还有半个小时就要开每日 Scrum 会议了,于是他决定把这些事情放在每日 Scrum 会议上来讨论。乘着开会前的间隙时间,关毅继续为电子商务通的培训做着

准备。与此同时, 小胡他们也都已经开始了各自的调研工作。

总结和思考

敏捷和 Scrum 倡导团队自我管理,在任务分配上提倡每个人按兴趣和能力自主选择任务。



下午 4 点半,关毅、小胡、段美女和胖子准时出现在了茶水间。他们站成了一个圈,看样子第一天的工作似乎不怎么顺利,一个个都垂头丧气的。

关毅作为 Serum Master,第一个在每日 Serum 会议上发言:"现在我们开始进行每日 Serum 会议,也叫 Daily Serum (如图 4-3 所示)。由于今天是第一次,我还是要再交代一下,每个人需要回答 3 个问题。第一,你今天都做了什么?第二,你明天准备做什么?第三,工作中有什么困难使你无法继续下去?下面从我开始。我今天主要的工作是整理电子商务通的培训资料,我会抓紧时间在明后天再给大家做一次培训,当然,时间不会太长,应该只是上次整体介绍的补充,当你们在工作中遇到了相关的问题时再来找我吧。今天晚上我会把资料发给大家,我希望大家在培训开始之前先看看,如果遇到问题一定要记录下来,要带着这些问题来参加培训,好吧!"

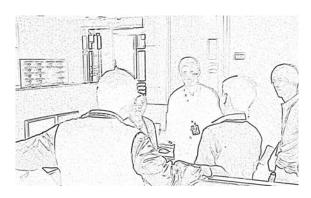


图 4-3 每日 Scrum 会议

大家不约而同地点了点头。

关毅继续说:"我今天还没有遇到什么问题,小胡……"

关毅正要让小胡继续下去时,徐天走了过来:"在开会吗?我可以参加吗?"



关毅急忙回答道:"欢迎欢迎,今天是第一次每日 Scrum 会议,正好您也可以给我们提点意见。"

徐天笑了笑,说道:"太好了,听说过 Scrum 有这样的会,就是没见过。不要被我打断了你们会议,你们继续吧!"

"好的,那下面就从小胡开始回答那 3 个问题吧!" 关毅继续开始组织每日 Serum 会议。

小胡有点紧张,皱了皱眉头,把声音压得很低:"我今天主要的工作是在网上搜索已有的 JavaScript Unit Test 框架,找到了一个 JSUnit 项目,下载下来后,简单地运行了几个示例,然后我就开始研究 JSUnit 源代码。明天我会继续研究 JSUnit 源代码。"

关毅皱了皱眉,对小胡说:"不用研究 JSUnit 源代码,我们只要拿来用就可以了。 另外,你对 Unit Test 的了解有多少?"

小胡摇了摇头。然后,关毅向小胡提出了建议,让他明天从了解什么是 Unit Test 开始。

段美女很自觉地接着小胡的话继续说:"我今天也是在网上搜索已有的 Ajax 框架,找到了几个网上评价比较高的,但是那些好像都不是免费的,而且代码也做了混淆,基本上是没法看的,明天我还是得继续这些工作。"

徐天听到这里,忍不住问了一句: "Ajax 具体是个什么意思啊?我只听别人说得很多,没有深入了解它。"

关毅看到小段跃跃欲试的样子,把到了嘴边的话咽了下去,点头示意小段继续。 段美女也没和关毅客气,"Ajax 嘛,简单点说就是 JavaScript 采用异步方式访问后台 服务器,然后根据返回的结果动态地修改前台页面的 DOM 节点,而不再是使用传统 的 Post 和 Get 方法来得到数据,从而实现以页面的局部刷新代替整个页面的刷新, 可以说是 Web 应用上的一次飞跃,使 Web 客户可以享受到如同桌面客户端一样的 体验。"

徐天点了点头,又开口问道: "JSUnit 和 JUnit 有什么关系吗?"

"名字听上去差不多,思想也差不多。JSUnit 是为 JavaScript 而设计的,JUnit 是为 Java 程序设计的,都是自动单元测试的框架。我以前一直以为单元测试是测试员该干的工作,今天才明白原来得开发来做。"小胡回答得挺实在。

大家都安静了下来,目光都集中到了胖子的身上,胖子被看得有些不自在:"我今天的工作本来是要开始调研电子商务通客户端的实现逻辑,不过,我照着 Wiki 上的指示来配置开发环境,配了一天也没搞定。"

徐天听到这里,转向关毅,说道:"你们的开发环境还没有配好是吗?"

关毅心里也是咯噔一下,怎么把这事给忘了!他急忙说道:"不好意思,是我的失误。Wiki上的开发环境的配置方法是 2.7 版的,3.0 版都已经不再沿用了。我带回来的资料里有关于新的开发环境的配置方法,我一会就发给大家。"

徐天继续问道:"那你们的开发工具是什么?都装上了吗?"

"还没有。我们会用 Rational Application Development 来开发和调试后台程序,对于前台 JavaScript 的开发,我们会用 Mozilla Firefox 的调试工具 Firebug 来调试。我们争取今天晚上就把这些开发工具都装好。大家还有什么问题吗?"

关毅说完后环视了一周,看没有人说话,于是继续自己的话:"有个问题我想和大家商量一下,就是 Task 的分配情况。虽然我们建议大家来领取自己感兴趣的或擅长的 Task,可是鉴于咱们第一周的工作是以熟悉环境、调研已有的系统为主,所以,我个人的意见还是你们 3 个为主力,我作为你们的技术支持,怎么样?现在我已经列出的 Task 里还有 3 个没人认领,Code review 这个算我的,其他的两个,你们看怎么办?"

"这样吧,小胡,你把调研 JavaScript Unit Test 框架的工作让给我吧,你负责电子商务通 Server 端的 Task,我知道你的 Java 和 C++都不错,所以,做 Server 的开发是最好不过的了。"关毅说。

小胡之前还有些顾虑,就是担心如果选择这几个 Task 会抢了关毅的风头,才勉为其难地选了一个自己本来就不是很想做的 Task,而且一天下来搞得头大如斗。这下可好了,他立刻来了精神:"既然大家都这么说,我也没有什么好说的。行,我就做剩下的 Task 了!"

"好的,既然大家都同意,那么一会回去之后,我希望大家把 SerumWorks 上的 状态都更新一下。然后呢,趁着大家都在,我给大家的说一下配置电子商务通需要

注意的几点事项。徐天, 你要是有事可以先离开……"

徐天示意关毅继续,关毅就把如何访问温哥华的代码服务器、如何配置本地的 编译环境等问题简单地讲解了一下。等关毅讲完,大家站得腿都有点麻了。关毅看 了看时间,已经5点半了。天啊,第一次每日 Scrum 会议开了1个小时!

关毅宣布散会,徐天示意关毅单独留下来谈一谈,因为他对关毅的做法有点不满。首先是开会也没通知他,另外,对于 Sprint 的计划,应该让他评审一下,毕竟他是经理。E 公司就像所有的大公司一样,流程以科学和严谨著称,怎么能说做就开始做了呢?

关毅只好道歉,他没想到徐天第一次这样对自己发脾气。他刚开始采用敏捷开发,接受了一些自我管理的思想,以为自己是 Scrum Master 就可以掌握一切了,但竟然忘记了过去自己多年恪守的职场规则,忽视了自己身边一个最重要的人,更何况自己还是个新人,更要虚心一点。过于自信、求胜心切真是要不得。慢慢来,关毅告诫自己并暗想:一定要影响徐天,让他真正从内心理解并接受敏捷的思想,虽然口头上他是非常支持敏捷的推广的。

总结和思考



虽然是强调自我管理,可千万别犯关毅的错。

关毅故作镇定地回到座位上,把准备好的资料通过电子邮件发给了大家,并且 在邮件中强调,要求大家今天务必把开发环境搞定,不要延误了已经制定好的工作 计划。于是,大家就开始从温哥华那边的服务器上下载代码、安装开发工具。等到 所有人的开发环境都配置好了,也已经是晚上8点多了。

关毅和大家一起走出公司,一路上没有人说话,每个人都在想着自己的心事。 这一刻他们最想说的一句话就是:这就是 Serum?

回到家后,关毅总觉得如鲠在喉,不吐不快。他打开自己的 Blog, 记下了 Scrum 第一天的感受。

万事开头难

万事开头难,真是一点也没有说错啊!第一天正式上路就让我手忙脚乱了:好不容易做好了Sprint 计划,心里还是感觉没底。"照明弹"计划是否成功无法预知,而且这个计划是否能顺利进行,还要看温哥华那边的脸色,不知道能否批准呢。计划,总以为可以做到最好、最细致,把一切都考虑在内,反倒是出了大纰漏,居然忘了"万丈高楼平地起",没有配置好开发环境就要求大家开始工作了,这个错误真是要不得啊。第一天就让 Team 加班,实在是有些过意不去啊。

特别让我心里有些郁闷的是,老大今天参加了我们的每日 Scrum 会议,挑出来不少骨头,真是丢人啊!虽然明天我们的 Sprint 1 就要正式开始了,但是我总是觉得大家的准备工作还差很多。不过,第一次嘛,出些问题是正常的,这样更好,问题暴露得越早,我们就有越多的时间应对,如果问题都到最后才出来,到时候哭爹喊娘也没用了。

关毅,不要放弃,加油啊!

Scrum 博士扫盲



• 每日 Scrum 会议 (Daily Scrum)

每日 Scrum 会议(Daily Scrum),即团队每日例会,条件允许的话,每天都应该在同样的时间和地点,组织所有成员站立举行,如图 4-3 所示。由于是以站立的状态开会,因此时间比较短,一般为 15 分钟左右。这个会议最好是在每天的早晨开,有利于团队成员们安排好当天的工作计划。只有团队成员可以在每日 Scrum 会议上发言,其他人员如果对项目进度有兴趣也可以参加,但只能旁听而不能发言。

会议由 Scrum Master 主持, Scrum 团队的所有成员轮流回答以下 3 个问题。

- (1) 昨天我完成了什么工作?
- (2) 今天我打算做什么?
- (3) 我遇到了什么障碍?



通过每日 Scrum 会议,团队成员之间可以彼此相互熟悉工作内容,充分了解项目进度,相互帮助解决问题。Scrum Master 除了倾听团队成员的发言外,还有责任设法解决团队成员在会上提出的困难,尽快扫除阻碍他们工作顺利进行的障碍。即使有的问题 Scrum Master 没有能力解决,例如某些技术细节问题等,他也应该找到团队中或其他团队的成员来帮助快速地解决问题。另外,还有两点需要注意的地方:其一,这是团队成员之间的交流,也是相互的承诺,并不是用来向老板汇报工作进度的;其二,这也不是一个专门用于解决各种问题的会议,团队成员们遇到的问题可以在会上提出来,而有能力解决这些问题的人可以在会后帮助他们解决问题。

总结和思考

每日 Scrum 会议是 Scrum 的精髓 ,最简单又最复杂 ,如何更有效地召开 ,需要不断地改进和摸索。



Scrum 之旅, 启程

生活就是一个七日接着又一个七日, Serum 的 Sprint 迭代也是这样, 所有的酸甜苦辣都浓缩在其中, 慢慢地品味起来, 又是那么回味无穷。

改进的每日 Scrum 会议

Sprint 迭代正式开始的第二天,好消息就来了,关毅收到了温哥华的正面反馈。

关毅:

我们收到了你的 E-mail。我和大家一起讨论了一下,我们认为你的决定是非常正确的。对于一个未知的领域,的确需要有一颗"照明弹"来指引我们前行。当然,有可能我们还需要不只一颗"照明弹",这对我们绝对是至关重要的,我们这边也会努力推广这个策略,在此我要向你表示感谢!

对于你们的 Sprint 计划,我们认为做得还不错,看来你们对这个功能理解得很深刻,在这么短的时间内能达到这个水平,我个人表示很钦佩。我想等到你们的"照明弹"计划完成后,应该会有更好的消息告诉我,对吧!

Peter

"照明弹"计划被批准了,并得到了加拿大那边的高度评价和强烈支持,看样子他们也要广泛地推行这一策略,Sprint 计划从整体上看是没有问题的。这时候,徐天也对整个Sprint 的计划没有多少意见,他只是提醒关毅一定要把需求彻底搞清楚。总之,可以正式开始Scrum 迭代了。

这一天的白天,关毅在大家饭后挤出了一点时间,给团队做了关于电子商务通的进一步介绍。整个过程进行得相当成功,大家也提出了不少很有价值的问题。凡是关毅能回答的,都做了尽量详细的回答,回答不了的,他也都一一记录下来,准备发 E-mail 和温哥华方面沟通一下。



一天下来,同志们的工作也都有了眉目。

在当天的每日 Scrum 会议上,关毅首先再次强调了规则:"昨天大家第一次做每日 Scrum 会议,表现都不错。不过有个问题要说一下,一个是时间方面,我会注意,争取短一些,另外,要求的 3 个问题一定要回答得有内容,不能太简单了,目的就是要给别人提供有价值的信息,而不是简单的汇报,明白了吗?

大家都点了点头。

先是段美女,在浩瀚的 Internet 海洋中,她找到了一颗璀璨的明珠——K 项目,这是一个真正意义上的 Ajax 框架系统,同时,最难能可贵的是,它是完全开源的,这让团队成员们感觉像穷小子走进了一个装满了珍宝的山洞,而且满洞玲琅满目的宝贝还任你挑选。不过,徐天对此也给出了提醒:"我们不能借鉴任何他们的东西,更不能 Copy 他们的代码,只能学习其思想和精髓。如若不然,我们将会承担巨大的风险。如果遇到了知识产权的问题,真的闹到法庭上,不但电子商务通的发布会被Kill 掉,而且 E 公司的声誉也会受到巨大的影响。这个责任可是我们谁也担当不起的啊!"

徐天接下来的话对于关毅来说, 也绝对是个好消息。

徐天接着说:"对于 JSUnit 框架也是一样,不过,我会和公司的律师联系一下,看看能不能直接使用 JSUnit。毕竟这是个工具,大不了我们去买些软件许可回来。"

接着是小胡的发言:"我今天在调研电子商务通的后台逻辑时,先是看了关毅给我们的材料,大体上了解了整个后台处理的流程。在我熟悉代码时,发现后台的代码格式比较乱,一个类的方法居然有 1000 多行的代码。说真的,我真想来一次彻底的重构,我看了两三个小时,也只能把一个简单流程串下来,效率有些低。不过,我也想好了对策。我用了一下午的时间配好了 Server 端的 Debug 环境,下一步,我打算采用单步调试的方式读代码。这样,即便代码再乱我也能顺利地看完所有业务流程的逻辑了。"

对小胡的发言,徐天又做出了指示:"我想,代码的重构能不能也在你们的这个 Sprint 里完成啊?毕竟是早晚都要做的,晚做不如早做。你看呢,关毅?"

关毅一时也不知道该如何回答。重构整个后台的代码,那可不是一时半会儿能

搞定的。

于是关毅也玩起了太极:"这个嘛,等我们完成这个 Story 时再说吧。等我们这个'照明弹'的结果出来,是否可以进行代码重构的工作就有答案了。对了,小胡,你这次做得很好,一会你给大家介绍一下配置调试环境的经验,我觉这个得是大家都需要掌握的。"

小胡答应了一声,接着就轮到了胖子:"今天我的工作进展不是很多,不过我也大概了解了电子商务通前台的交互逻辑。例如,虽然对于用户的输入验证都是通过 JavaScript 在前台进行的,但都是通过 Server 端预先生成静态页面。可以说,完成任何一次操作都需要从 Server 端重新生成一个新的静态页面。我也把这些和已有的采用 Ajax 框架的 Web 应用比较了一下,还是有很大的差距的。我明天的工作就是要整理出电子商务通的各个业务场景的前台交互流程。我的话完了。"

徐天非常高兴,对关毅说:"今天大家都干得不错嘛!看来每日 Serum 会议名不虚传,每天我都能知道大家的进度,并且发现新的问题!我还有点其他的事,你们继续吧,我先撤了。不过我有个建议,你们可不可以每天把每日 Serum 会议的讨论记录下来,会后转发给大家,也好有个备份啊。还有,记得也给我发一份啊!"

总结和思考

给经理写会议记录无可厚非,但是如果变成了汇报工作就违背了 Scrum 的初衷。有些团队经理甚至把每日 Scrum 会议变成了监督和指挥团 队的会议,这完全违背了 Scrum 的精神。



徐天说完就急匆匆地离开了。

关毅问大家是否愿意以后轮流做每日 Serum 会议的记录? 大家都表示,既然老大发话了,没问题,无条件服从!

接下来,关毅又趁热打铁,带着胖子、小段一起到小胡的座位上,让小胡给大家演示了如何配置 Server 端的调试环境。等大伙都搞明白了,也到了下班的时间。

关毅看着大家询问的眼神,有些不好意思,他说:"大家今天工作得非常好,今 天的每日 Serum 会议就我由我来记录了,大家没什么特别的事情就可以回去了。而



且我也在这里声明一下,Scrum 的精神是不提倡加班的,大家以后要尽量提高白天的工作效率,我们争取做到永远不加班,好吧!"

大家听到这些,一阵叫好,惹得旁边的几个 Team 一阵埋怨。他们几个赶紧夹着尾巴撤了。即使他们如此狼狈地逃走了,一路上,其他的 Team 成员还是向他们投来了无数羡慕的目光。

跟踪每天的进度

在这周接下来的两天,大家的进度简直是一日千里。

先是徐天告诉大家, JSUnit 可以直接使用, 没有 License 的问题。

段美女也完成了对 K 项目的初步调研,基本的框架已经搭建完成。经过和胖子的讨论, 段美女得出了结论: K 项目不能满足电子商务通的需求, 有些前台的 Widget 需要定制。同时,她提议是否可以先采用 E 公司内部的一个开源项目 D 来补足基本框架的不足。最令人鼓舞的是,D 项目已经提供了关毅他们需要的一些 Widget 的基本实现,只要稍加改进就可以使用了。

胖子也已经完成了电子商务通前台交互的方案,下一步就是要和段美女一起完成前台的实现。

与此同时,小胡也已理顺了后台的整个处理逻辑,但是,他提议不要完全复用原来的代码,这样会使最终的代码更加难以理解,可以尝试采用新的架构,实现电子商务通的 Ajax 框架的后台逻辑,以便以后在代码重构时,能够在把老的代码移植到新的架构上时不影响整个 Ajax 框架。

关毅在分析这些成果时,脑海里逐渐有了电子商务通 Ajax 框架整体的雏形。

不知何故,徐天出席了前两次的每日 Serum 会议后就没再露面,这让关毅他们 开会时的气氛轻松了不少。不过,每日 Serum 会议的记录还是每天按时发给了徐天。 只是这件事总让关毅觉得有些问题,到底是什么,他也说不清。

一转眼就到了周五,第一周的最后一天。按照计划,这个 Story 到下周一才算正式结束。不过,关毅担心过了个双休日,大家会把一些体会和心得忘掉,他觉得有

必要对大家进行提醒。



Team:

今天是周五了。我们这几天的工作做得非常好,希望大家继续努力,在今天取得更大的成果。

接着有一个双休日,大家可以好好放松一下,不要来加班。

还有一点我想提醒大家,我们还是需要及时更新一下 ScrumWorks 上的 状态,我看现在只有小段记得每天进行更新,其他人也不要不把这个当回事,好吧!

关毅

这一天对于关毅来说,可真是够漫长的。

上午关毅被徐天叫去开会,是由徐天的老板 Chris 主持的一个技术讨论会,要求 Chris 下面所有的 Team Leader 和经理必须参加,议题并不固定,可以是有关产品的 发展方向,也可以是对最新技术的讨论。今天的议题是敏捷开发,由关毅主讲。会 议进行了整整两个小时,关毅说话说得嗓子都哑了。会后,Team Leader 和经理的提 问也非常踊跃,大家对敏捷开发的质疑很多,而且对关毅的回答也不是很满意。说 到底,就是由于关毅所讲的 Serum 经验很大一部分还是停留在培训的理论层面,但 这也不能怪他,这仅仅只是第一周而已。不过,关毅倒是发现了一个很重要的信息,就是大家对 Serum 和敏捷有很浓厚的兴趣。说到这里,当然不能不提 Chris。作为徐天的老板,Chris 倒是对使用 Serum 有点担心,害怕出岔子,这也让徐天和关毅的压力陡然增加了不少。

吃过午饭,关毅就马不停蹄地开始了余下 Story 的分析工作。登录 ScrumWorks,关毅看到小胡和胖子也都已经更新了各自 Task 的状态,只是由于他们把几天的工作一次做了更新,整个燃烧曲线(Burndown Chart)就显得有些异常,任务量平缓几天,接着突然下降很多,并不符合实际的情况。关毅心想,一定要让团队成员养成更新ScrumWorks 的习惯,其实每天只要花 2 分钟就可以完成。只是关毅不明白,为什么他们不喜欢做更新呢?



Scrum 博士扫盲



我们来认识认识一些 Scrum 辅助图示工具。

• Burndown Chart

Burndown Chart 的横轴表示整个 Sprint 的总时间,纵轴表示 Sprint 中所有的任务, 其单位可以是小时、人天等。一般来说, Burndown Chat 有 Sprint Burndown Chat 和 Release Burndown Chat 之分。Sprint Burndown Chart 的示例如图 4-4 所示。

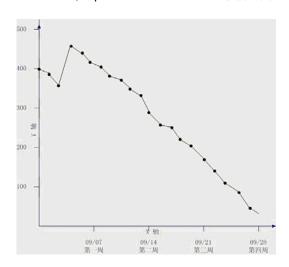


图 4-4 Sprint Burndown Chart 示例

从图 4-4 中可以看出,这是一个 Spirnt Burndown Chart。

- ◆ 在 8 月 31 日 Sprint 刚开始时,这个 Scrum 团队预计这个 Sprint 的总任务量大 概是 400 小时;
- ◆ 到了9月4日,这条曲线不降反升,这是正常的现象,说明在实际开发过程中, 团队认为之前对某个 Story 的估计不够准确,他们根据实际情况调整了其中的 任务和相应的估计时间值;
- ◆ 曲线的中间有波动,但是总体趋势是随时间倾斜向下的,而且团队在 9 月 28 日 Sprint 结束时完成了这个 Sprint 的任务。

Sprint Burndown Chart 可以体现 Sprint 的进度。如果 Sprint Burndown Chart 一直是上升状态,或当 Sprint 进行一段时间之后,Sprint Burndown Chart 上当前点的 Y值仍然与 Sprint 刚开始时相差无几,就说明这个 Sprint 中的 Story 过多,要拿掉一些 Story 以保证这个 Sprint 能顺利完成。如果 Sprint Burndown Chart 下降得很快,例如 Sprint 刚过半时 Y值已经接近零了,则说明为这个 Sprint 分配的任务太少,还要多加一些任务进来。在 Sprint 计划会议上,如果团队对即将要做的任务理解和认识不充分,就很可能导致这两种情况的出现。

• Release Burndown Chart

Release Burndown Chart 用于记录整个 Scrum 项目的进度,它的横轴是这个项目的所有 Sprint,纵轴是在各个 Sprint 开始前所有尚未完成的工作,它的单位可以是个 (Story 的数量) 人天等。

在如图 4-5 所示的 Release Burndown Chart 示例图中可以看出,这个 Scrum 项目预计需要用 6 个 Sprint 来完成,项目开始时,预计总共有 300 个 Story,在有些 Sprint 点,这个 Story 数还稍有上升,就像在图 4-4 的 Sprint Brundown Chart 中我们看到的曲线波动一样,这是因为 Scrum 团队增加了一些 Story,但曲线的总体趋势是平缓向下的。

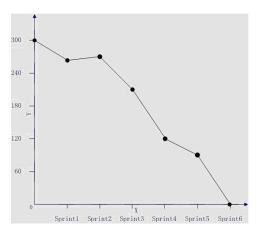


图 4-5 Release Burndown Chart 示例

周五的每日 Scrum 会议,大家开得很兴奋。今天的收获确实不小:他们已经完



成了一个简单业务流程的 Ajax 实验原型,虽然简单,但是"麻雀虽小,五脏俱全"。 所有的技术难题都已被攻破,怎能让人不兴奋!

不过,关毅仍然适时地跳出来给大家泼了一盆冷水:"我知道大家已经取得了很大的突破,但这还只是一个原型,只能说明这个方向是对的,对于整个 Sprint 1 的目标来说,我们还是在原地踏步,明白吗?接下来的几周时间才是非常重要的,我们可不能骄傲自满啊!现在的原型做得再好,到 Sprint 1 结束时,我们没有拿出真正的东西,可就意味着我们工作会大打折扣的。不过,今天我还是要祝贺大家,Great Job!"这句话是关毅的心声,他已经有好久没有这么激动了。

这一刻,整个团队紧紧地团结在了一起,每个人都充满了信心,前途一定是光 明的。

当关毅和大伙一起走出公司大楼时,他们心里又问了一句:这就是Scrum吗?

总结和思考

Burndown Chart 是常用的衡量团队进度的可视化工具。敏捷开发可以 给项目提供更多的可视性。



每日 Scrum 会议的真正内涵

Serum 第一周的经历让关毅感慨颇深,最重要的体会是每日 Serum 会议很容易使团队产生凝聚力,不再是过去那种"你干你的,我干我的"的状况了。

回到家后,关毅草草地吃过晚饭,再次翻出 Scrum 的培训资料,想看看通过一周的实践,再与理论相结合,能不能有新的发现。

他注意到了资料上的一句话:每日 Scrum 会议一般是 15 分钟左右,最好在每天的清晨开。

看到这里,关毅苦笑了一下: 15 分钟左右,我一直都明白,可每次都开到了 1 个小时,原因是自己总是会忍不住和大家深入讨论技术问题。按照建议,技术细节的讨论应该在每日 Scrum 会议后进行。他有些愧疚,每日 Scrum 会议时间太长,让

大家站了那么久,腿都酸了。原来每日 Serum 会议站着开的目的就是为了避免会开得太久,所以关毅开始怀疑徐天不再出现在每日 Serum 会议上是由于受不了那么长时间的站立。只不过这几天他确实有那么多事需要交代,而且大家也需要对一些问题进行讨论,时间长一点应该是不是可以?以后应该如何避免发生这样的情况?关毅越想心里越没底,还是找 David 吧。

David:

又要向您请教了。我们已经完成 Scrum 第一周的开发,感觉上还不错,只是我个人还是觉得其中有些问题有待解决。第 1 个问题,就是我们的每日 Scrum 会议每次都要开 1 个小时左右,不过,这主要是由于我们刚开始开发,有很多事情需要讨论,其他时间我们都在忙各自的工作,所以就只好趁这个时间来讨论了,估计以后等大家都熟悉这个项目了,我们就不会再花这么多时间讨论了。第 2 个问题呢,也和每日 Scrum会议有关,就是我们是在每天的下午开会每日 Scrum 会议,而不是早上,这也是我的提议。我的想法是一大清早来,经过一夜的休息,有可能就忘了昨天工作中的一些细节,同时也无法想清楚当天要继续做什么,改在下午开,大家对一天的工作都会有一个非常清楚的概念,同时对第二天的工作计划也能做到心里有数。不知道你觉得如何?

我非常渴望能尽快得到您的回答,非常感谢!

关毅

这一休息就休息到了周一,关毅一上班就看到了 David 的回信。

关毅:

你提到的问题,我之前也遇到过。项目开始阶段总是会遇到很多的问题,而且大家都很忙,所以在每日 Scrum 会议的时候难免会多讨论一些问题,不然都会感觉自己一天什么也没做,心里很有愧疚感。这些都是很正常的,也是 Scrum 特有的激励手段。你们把开会时间改在下午本来无可厚非,不过正因为这样,可能你们每天开会时大家都会不自觉地多说两句。可是你有没有想过每日 Scrum 会议的真正意义呢?



话说回来,每日 Scrum 会议设立的主要目的是让团队里的所有人都能及时地理解别人都做了什么?准备做什么?有什么问题?其实也就是整个项目都完成了什么?还要完成什么?有什么问题阻碍了项目的进行?自己是否有好的想法来解决这些问题?通过大家提供的信息,Scrum Master 以及每个人都可以非常明确地知道整个项目的进度,如果整个进度已经落后于之前的计划,那大家就要想办法来赶上进度了。而且实践证明,这个方法很有效。如果我们的工作比计划提前了,就可以适时地去调整后面的工作,例如可以增加一些任务。

说到底,每日 Scrum 会议是一个让大家同步信息的平台,而不是一个交流问题、讨论问题的渠道,这也是为什么让大家站着开会的初衷。那么应该在什么时候去讨论问题和解决问题呢?这就是 Scrum Master 的一个非常重要的职责了。而且,如果在会上有人说自己遇到了一个很麻烦的问题,刚巧有另外一个知道解决方法,这时候,作为 Scrum Master,你就需要在会后把他们两个召集在一起,一同解决这个问题,而不是在会上让他们任意地讨论——毕竟还有别人也在参加这个会议,不要浪费别人的时间。当然,如果没有人对出现的问题有好的解决方案,你一定要记得不能在会上对这样的问题做过多的讨论,这是在浪费所有人的时间,你要做的是会后联系其他的资源来解决这个问题。

关毅,如果你们真的遇到一些问题需要大家一起来讨论的话,你完全可以另找个时间,找一个合适的地点专门来讨论嘛,千万不要让大家在每日 Scrum 会议上站得太久。有些团队非常抵制 Scrum 的最大的根源就是他们的 Scrum Master 会把每日 Scrum 会议开得很久,导致大家每天都要在辛苦的工作之外付出更多不必要的辛苦。千万不要继续做这样的事,切记!

David

关毅看完邮件,不禁吸了一口凉气。"好家伙,这么严重啊!"他不自觉地说了一句。关毅心中默念: "不管 David 说有人会为了每日 Scrum 会议开得太久而放弃 Scrum 是不是真的,对于每日 Scrum 会议的真正意义我还真是'只知其然,不知其所以然'啊!看来 Scrum 的学问还大着呢!"

总结和思考

固定的时间和固定的地点也有助于团队把每日 Scrum 会议坚持下来,至于召开的时间则没有定论。大多数团队选择早上一上班的时间,因为头脑清醒;如果是跨国开发团队,比如关毅他们,下午下班前召开比较合适,因为信息可以更及时地反馈给比北京晚 10 多个小时的加拿大团队。

知错就改,不断调整,敏捷嘛!

在第二周的第 1 次每日 Serum 会议上,大家非常高兴地把自己的工作情况作了 汇报,整个系统的原型已经顺利完成,关毅也及时地表扬了整个团队的工作。

就在大家想继续讨论接下来的工作时,关毅马上打断了大家: "我们的第 1 个 Story 做得非常成功,我也根据这个 Story 的成果,结合我们实际的工作状况,为接下来的工作做了计划。我明天正式开始对大家的代码进行评审,希望你们今天下班前提交自己的代码,并在 ScrumWorks 上更新自己的状态。明天一早你们会看到新的 Task,大家不要客气,尽管去挑吧!咱们今天的每日 Scrum 会议就开到这里,你们记得把最近的感受都写下来,发个电子邮件给我,谢谢了!"

关毅看着大伙有些诧异的表情,微微一笑: "以后我们的每日 Serum 会议一定不要超过 15 分钟,大家记得监督我哟!"

关毅正要转身离开,胖子拦住了他:"老大,既然咱们的每日 Serum 会议以后不会超过 15 分钟,那能不能晚点开啊?如果我们还是在 4 点半开会,就 15 分钟,然后再继续回去工作,这样也不太好啊,既打乱了我们的工作思路,也不能正确反映我们一天的工作情况。说实话,每天一开完每日 Serum 会议,我就没有再继续工作的心思了,觉得一天的工作已经结束了。"

关毅看了看其他人,大家都不约而同地表示同意。关毅转念一想,是啊,自己 何尝不是这个感觉啊。

关毅说:"好吧,咱们把时间改到 5 点半,这样大家应该没问题了吧!"大家全票通过。



初尝 Scrum 有感

开完每日 Scrum 会议,关毅也没闲着,他继续在 ScrumWorks 上完善着下几个 Story 的计划。

不一会儿, 关毅就收到了他们几个的 E-mail。

第一封是胖子的。

老大:

对于刚刚过去的第 1 个 Story 的开发, 我觉得收获颇多。 我这是第一次经历 Agile 的开发模式, 感觉很新颖, 同时也有 很大的压力, 特别是头两天, 看到小段和小胡的进展那么快, 我心里觉得挺 不舒服的。不过还好啦, 我终于迎头赶上, 顺利地完成了所有的 Task。

我不知道是不是因为我们采用了 Scrum,我总是觉得我在这一周内,每 天都充满了活力,对自己的工作有从未有过的激情,我希望我们能继续这样 快乐下去。

于大鹏

关毅看过胖子的 E-mail, 会心地笑了笑, 又打开了段美女的 E-mail。

领导:

这一周的开发,让我和 Scrum 做了第一次亲密接触。我对 Scrum 早有耳闻,现在好多知名的大公司都在推广采用它,但 我其实一直对它有所怀疑。可是虽然只有一周的时间,已经足以打消我心中的疑虑。我觉得自己提高很快,每天的计划性很强,主动权也很大。

段小丽

最后一封电子邮件不用问也知道是小胡的,关毅倒是很希望能从里面看到一些 不一样的东西。

关毅:

这是咱们第一次合作。自从我在面试时见到你的第一眼开始,我就知道:跟着你,成功是必然的。这从你的名字里就能感觉到。于是乎,对于我们是否采用 Scrum,我一直都不是很在意,因为我也知道无论是什么开发模式,都是要由人来实施,人在这里才是最关键的。而咱们中国有句俗话,"物以类聚,人以群分",能被你选中的人,应该不会差到哪里去。

但是,如果真的说到我对 Scrum 的感受,可能这才一周,我还真的不能说有太深刻的体会,唯一的感受就是我觉得现在的每一天都过得很充实,每过一天,我都觉得我们离成功近了一步,这是我以前从未感受过的,我想这可能就是我最大的收获。

胡志刚

关毅看完这些 E-mail 后,默默地沉思了一会儿,然后郑重地回复了 E-mail。



亲爱的团队成员们:

我由衷地感谢你们对我的信任。虽然我们离成功还很远, 但是我保证,那一定是属于我们的。

关毅

第二周的第 2 天,大伙一来到公司就登录 ScrumWorks,查看后面几个 Story 的 具体计划。看到新的计划,大伙连想都没想,就各自选了自己的 Task。小胡选的是继续 Server 端的开发,胖子则是完成 Client 端代码的编写,段美女就去编写整个电子商务通的 Unit Test Case,关毅还是老任务——代码评审(如图 4-6 所示)。

总结和思考

Burndown Chart 是常用的衡量团队进度的可视化工具。敏捷开发可以 给项目提供更多的可视性。



维松Scrum之旅

——敏捷开发故事



图 4-6 剩余的 Story

来自大老板的质疑

一切都看起来都是那么顺利,那么和谐。关毅在和老板们的每周例会上非常乐观地估计:"我们的工作一切进展顺利,按照目前的情况,我们应该在第三周就可以完成这个 Sprint 的所有工作。"

徐天的老板 Chris 有些不客气地说了一句:"能问个问题吗?你们这个 Sprint 的目标是什么?"

关毅不假思索地回答道: "完成电子商务通的 Ajax 框架的开发。"

Chris 又问道:"那怎么算是完成呢?"

"完成一个可以工作的电子商务通的 Ajax 框架。" 关毅有些不解 Chris 为什么要这么问。

"'可以工作的',这句话我是不是可以理解为能够拿给用户使用了?" Chris 还是不依不饶。

"应该是。"关毅有点底气不足了。

"应该是?我想再问一下,你们能保证完成所有的测试,包括功能测试和系统测试吗?" Chris 的这个问题让关毅有些猝不及防。

"这个嘛……目前测试工作还没有介入,不知道他们是怎么安排的。不过晚一点也好,这样能够保证测试人员拿到一个更完整的产品。"关毅的回答让他自己都觉得有些不自信。这也难怪,他根本就没有来得及考虑这个问题,虽然理论上他清楚应该怎么做。

Chris 是二线老板,徐天的顶头上司,不好对付,在 IT 外企研发圈里摸爬滚打了十几年,非常有经验,管理风格比较严谨。

Chris 继续说:"噢,如此说来,你所说的可以拿给用户体验的、可以工作的产品,居然是没有经过测试的?我是不是可以这样理解呢?"

"这个……"关毅答不上来了。

"你们的产品在完成所有功能后,最后安排测试的时间了吗?" Chris 问道。

"按照敏捷的流程,最后应该是没有单独测试的时间的。不过我们还是计划留了 1个月的测试时间,最后集中测试。"关毅回答。

"胡闹!我看到的是,你们的产品没有一个完整、详细的开发计划,1个月的测试时间够吗?你们如何保证测试的质量?你们在迭代里面也不做测试,最后怎么保证能按时和按质量发布?在 E 公司是绝对不允许这样的,真不知道加拿大那边是怎么搞的!还有,你们产品组,每天一下班,所有人都没影了,这样的态度,工作怎么可能做得好!"Chris 很生气地说。

总结和思考

组织中,有些老板不喜欢员工一下班就走。在敏捷实施中,让老板认同对敏捷的价值观是最重要的。



关毅这时候汗都出来了,他下意识地看看了徐天。徐天明白,Chris 这样也相当于在给自己施加压力。

徐天赶忙出来圆场:"计划主要还是加拿大他们在把关,他们心里有数。关毅他们刚刚采用敏捷开发,还在转型期,经验不够,不过他们还是非常认真的,你提的建议我们回去后会好好考虑。"



这个周的例会就这样不欢而散。

会后徐天找到关毅,在一间小会议室里,徐天告诉关毅不要在意:一来 Chris 对敏捷开发一直有成见;二来 Chris 属下的一个项目最近出了一个比较大的状况,产品马上就要发布,结果测出了一个非常麻烦的问题,北京和美国两边的团队一起忙了一周,居然发现是当初设计上的一个漏洞,而且两边都不愿对这个问题负责,这几天他们都在扯皮,闹得挺严重,Chris 也很头疼。不过,Chris 提到的问题也需要好好地思考一下,尤其是对测试这个部分。关毅点点头。徐天苦笑了一下,就起身离开了,这段时间也够他发愁的了。

对今天发生的事,关毅是无论如何也不会无动于衷的。在当天的每日 Scrum 会议上,关毅向小胡他们提起了会上发生的事,并且希望大家不要有什么压力和负担,按照既定的计划工作,不要被别的事情所干扰。不过,对小胡他们的工作情况汇报,关毅还是相当的满意,反倒是对自己的 Task,关毅需要做一下检讨了。代码评审的工作,他进行得有些慢了。现在看来,Scrum Master 如果给自己安排了的很多开发任务,和他本身的职责也是有冲突的。关毅觉得这是一个很好的问题,可以留在后面解决。

说来奇怪, 关毅自从开始电子商务通 3.0 的开发以来就从来没和测试组联系过, 还说什么敏捷呢?这确实有点说不过去, 也难怪 Chris 责问。

在接下来的几天里,关毅找到了测试组的组长赵平,互相认识并沟通了一下。 赵平告诉关毅,他们现在正忙着测试几个客户定制的版本,所以还无暇顾及电子商 务通 3.0 的测试。不过赵平对敏捷的概念也很感兴趣,他答应关毅回去以后自己先 好好扫扫盲,等忙完了这边的事就立刻开始电子商务通 3.0 的测试,不再像过去瀑 布流程那样等开发全部结束后才开始测试了。

另外,关于测试的一些问题,关毅还需要和温哥华的 Peter 沟通一下。关毅出差 在那边的时候,和 Peter 定下了每周两次电话会议的计划,也正好可以利用这个机会 讨论。

于是,这周电话会议的议题就是有关测试的问题。

所谓 Sprint 完成的条件就是不但完成产品功能的开发,而且测试工作也要完成,

最后没有特别严重的 Bug。经过热烈的讨论,大家认为,目前对于电子商务通团队来说,这是一个很大的挑战。一来是测试组的资源明显不够,没有时间在当前 Sprint 测试了,而且由于之前都是采用瀑布式的开发流程,一系列测试用的规范流程不可能一下子改过来。另外,开发这边也有问题,任务安排得太满,恐怕即使测试那边有时间,开发也没有能力在 Sprint 结束前给测试留出足够的时间。当初制定计划的原则就是要尽快地完成产品的功能,只要在 Sprint 结束前完成所有的功能就行。但是如此一来还是敏捷吗?这样不还是像过去一样,先集中开发,然后再集中测试?不过基于现状,也只能希望测试组能尽快跟进。测试工作可以比开发滞后一个或者两个Sprint,开发也要从下一个 Sprint 开始给测试留有余地。

关毅对这次的会议的结果非常满意,这个头疼的问题可算是告一段落了,而且对 Chris 那边也有了交待。

在周五的技术讨论会上,关毅就着重介绍了这次和 Peter 讨论的成果,以及如何让测试和开发在 Scrum 里协同起来。Chris 看起来心情不错,不住地点头。

总结和思考

在很多组织中,敏捷开发都是从下到上推行的。尽可能多地得到上层的支持是非常关键的。初期采用敏捷开发的团队对于迭代式开发的实质(可以工作的软件而不是仅开发完功能)要求很容易忽略或觉得难以达到,不妨循序渐进。

Sprint 1 发慌的收尾

经过了两周的工作,电子商务通的 Ajax 框架已初具雏形。

关毅这几天也很兴奋。一天中午吃饭时,他和测试组的老张谈起这些,一直在 夸采用 Ajax 框架的电子商务通在客户体验已经提高到另一个境界了。

就在关毅夸夸其谈时,老张提出了一个很简单的问题:"小关,你们的电子商务通是不是还是在 Web 上应用啊?你们都支持什么浏览器啊?"

关毅一下子就愣住了,心想不好,客户端开发都用的是 Firefox,在 IE 上只是简

発**がScrum之**派 一教徒开发故事

单验证过,没仔细测过。

关毅不再和老张瞎聊, 三下五除二地吃完了饭, 就急忙赶回办公室了。

回到座位上,关毅马上打开微软的 IE 试着登录了电子商务通,完成了几个基本的业务操作,悬着的心终于落地了,看样子问题不大,在 IE 上还没出现什么问题。可关毅还是有点不放心,他又用 IE 执行了已经写好的 JSUnit Test Case,也没有什么问题。这下关毅终于放心了。在 Windows 上,现在 IE 和 Firefox 占了超过 90%的份额,只要能搞定这两个浏览器就大局已定。

关毅这会儿真想立刻去告诉老张这个好消息,顺便能向老张好好展示—下这个 无比强大的电子商务通,准能让老张佩服得五体投地。

关毅愉快的心情一直持续到周三的每日 Serum 会议。这几天的会都开得特别快, 大家的工作都很顺利,照段美女的话说,现在的工作完全是体力劳动了,根本不用 思考就能做了。

这段时间,关毅也觉得每天的每日 Serum 会议开得很没意思,大家过来都是说已经完成几个点的编码了,总共完成百分之几了,明天继续,如此这样。还真不如照 Franco 说的,发个电子邮件说一声就完了,免得大家觉得麻烦。可是想归想,在这个问题上,David 已经给关毅强调过每日 Serum 会议的重要性。每日 Serum 会议不仅仅是更新各自的状态,更重要的是解决困难、协调和指引方向。作为 Serum Master的关毅始终牢记 David 给他的这些建议,不断地改进每日 Serum 会议,不断地提醒团队成员在回答 3 个问题时一定要言之有物。另外,关毅自己也时刻掐着表,把会议时间控制在 15 分钟以内。

这个 Sprint 已经接近尾声了,接下来已经可以开始准备 Sprint 评审和 Sprint 回顾了。关毅心里有些暗暗发慌,总是觉得有些事没有做完。但是,是什么事情呢?他又说不清楚,索性也就不再去想那些没用的了。还是先把现有的事做好吧!

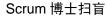
Sprint 1 评审——意外的结局

到了最后一周,这天是周四,是关毅和 Peter 商量好要开 Weekly Scrum 的日子,Sprint 也即将结束,双方都要在这个会上展示一下各自的成果。就是他们第一次尝试传说中的 Sprint 评审会议(Review Meeting)。今天到场的还有徐天、Product Owner Greg 以及 Peter 的经理 Michael。在会的每个人都有些激动,大家都对第 1 个 Sprint 的工作成果非常期待。

会议开始了, Peter 请所有的与会人员登录 E 公司内部的网络会议平台,接着选择正确的会议 ID, 进入 Peter 事先预订的网络会议室。登录网络会议室对关毅他们来说可是头一回啊,4 个人忙活了10 分钟才登录上去。

Michael 显得有些不耐烦: "Peter,可以开始了吗?我们只有1个小时的时间,下了班我还得去接孩子回家。你知道,回去晚了,我老婆可是不会跟我客气的。"

Peter 只有再次询问关毅。关毅汗都下来了,急忙说:"OK,我们可以开始了,没问题了。"





• Sprint 评审会议

Sprint 评审会议在 Sprint 结束时召开,由开发团队展示这个 Sprint 中完成的功能,长度为两个小时左右,不需要 PPT,一般是已经完成功能的 Demo,而且客户、管理层、Product Owner 以及其他开发人员等都可以参加。

在每个 Sprint 结束时,应该组织一次 Sprint 评审会议。Scrum 开发团队将在会上展示他们在这个 Sprint 中所做的工作,一般采用向大家演示产品新功能的方式来展示。

相对来说,Sprint 评审会议不必很正式。通常不需要用到 PPT,而且长度最好控制在两个小时之内。也就是说,不要让 Sprint 评审会议成为 Scrum 团队的负担,不必让他们花太多时间来准备这个会议。



Sprint 评审会议的参与者包括所有对该产品感兴趣的人,可以是产品责任人、Scrum 团队、利益相关者、管理层人员、客户,甚至是来自其他项目的开发人员等。

在 Sprint 评审会议上, Scrum 团队用 Demo 的形式展示产品的新功能之后,与会人员依据在 Sprint 计划会议上确定的这个 Sprint 的目标来评审具备了这些新功能的产品。

为什么一定要用 Demo 的形式来展示产品的新功能呢?因为这种方式有很多好处。首先,参与会议的人都能直观地看到 Scrum 团队的工作成果;其次,利益相关者们可以据此提出更切合实际的反馈意见;第三,为了完成 Demo,团队会努力完成并发布那些功能,即使只是发布在测试环境下,也比没完成的好。如果不做 Demo,也许不少功能会停留在"已完成 99%"的阶段。相比较起来,在有 Demo 的情况下,也许"已完成"的功能数量会相对少一些,但它们是确确实实完成了的。否则,那些"99%"的貌似完成的功能势必还要拖到下个 Sprint 来解决。

假如有一个刚从传统的开发模式转而采用 Scrum 的团队,由于不习惯这种自我约束、自组织的方式,在 Sprint 中并没做多少工作,那么在会上演示的时候,他们会觉得很尴尬。没准老板会因为看到他们花了这么多时间只做了这么少的工作而感到很生气。发生这种情况当然是大家都不想看到的。但良药苦口,在下一个 Sprint 中,这个团队肯定会吸取教训。他们会明白,无论什么情况下都必须做 Demo,那么他们必然会很好地完成这个 Sprint 并演示给大家看。

Peter 说:"好的,现在我们正式开始 Sprint 1 的评审。第 1 件事,我们要开始演示这个 Sprint 的成果。先由我们这边开始吧!Bryan,开始吧!"

Bryan 是 Peter 的组员,他的工作是完成工作流引擎对整个电子商务通流程的支持。例如,对用户提交表单状态的控制,同时保证数据流双向进行的正确性等。Bryan 漂亮地完成了演示,在场的每个人都为他鼓掌祝贺。就这样,Peter 他们很快就完成了演示。

轮到关毅上场了。关毅清了清嗓子:"大家早上好!"关毅一激动,忘了温哥华的时间已经是傍晚了,只好接着说:"今天我很高兴在这里和大家一起检验我们的这个 Sprint 所取得的成果,我们在这个 Sprint 里主要的工作是完成对电子商务通整个前台交互逻辑的重构,为此我们为电子商务通实现了一个全新的 Ajax 框架,所以现在的电子商务通已经不再像过去那样,是一种需要用户不停刷新页面来完成数据获

取和计算的 Web 应用了,我们已经赋予了它新的生命。本来我是想来给大家做简单演示的,但是我现在觉得,与其看着我做,不如大家亲自体验一下吧!请大家在自己的浏览器里输入我们的测试 Server 的地址,http://e-commerce:9080,按下回车,你就可以感受到桌面级的用户体验了。"



图 4-7 Sprint 评审会议

关毅这番话说得大家都不由得替他捏了一把汗,毕竟现在的产品还没有经过正式的测试,问题肯定是有的,特别是在几个老大都在的情况下,还是应该稳妥一点。关毅何尝不知道这样做会有很大的风险?这次的演示让他费尽了心思。怎样才能做得让大家印象深刻?才能彻底征服大家?关毅不知道在心里问了多少次。最后他觉得应该把参加演示的人当成客户来看待,而客户在接触一个新的产品时,最能让他记忆深刻的方法莫过于让他亲身体验一下了。这次关毅也是下定了决心,如果得到好的评价,就是真正的成功,但是如果反响不好,那就是工作没做好,需要继续努力。所以说,关毅是铁了心冒这个险。只是,他的脑子装满的都是对庆功场面的幻想,失败,他压根就没考虑过。

古人有云:乐极生悲。

关毅还在幻想着会得到大家的一致喝彩时,他听到的是 Michael 冷冰冰的声音: "关毅,我的浏览器进入你提供的页面后,根本打不开表单,怎么会这样啊?"

接着, Peter 也开始了"责难"。关毅这时候恨不得找个地缝钻进去。而且,两边相隔1万多公里,谁知道那边到底发生了什么?

雅松Scrum之旅

小胡看不下去了,就插了一句:"我们这里都没问题啊,所有的都是好的啊!"

Michael 有些不高兴了:"难道你们是说我在开玩笑吗?"

这时候,徐天也坐不住了:"Michael,不好意思啊,千万别误会,我们没有这个意思,只是可能的确是有些地方出问题了,我们会后一定立刻进行检查,确定是出了什么问题!"

Peter 也帮忙说起了好话: "Michael,我们这边也会帮忙一起来重新评审他们的 代码,看问题是出在哪里。"

Michael 听到这里,语气稍微缓和了一下:"OK,这个事情可不能马虎了,今天的演示就到这里吧!"

关毅也长出了一口气。今天这次可是把脸丢尽了。

Sprint 评审会议结束后,紧接着就是 Weekly Scrum 了,就是每周一次 Peter 的团队和关毅的团队一起汇报工作状态。Peter 登录到 ScrumWorks,让大家开始逐一汇报自己的情况。

轮到关毅他们的时候, Michael 又问了一句: "从 ScrumWorks 上看, 你们的 Story 都已经是 Done 了, 你们的代码都跑过 Unit Test 了吗?"

关毅小声回答道:"一部分跑过了!"

"代码都经过评审了?" Michael 继续问道。

"是的,都做过了。"

"都通过功能测试和集成测试了吗?"

"这个,肯定没有。测试的问题我们不是讨论过吗?暂时还没有考虑。"

"既然是这样,我们以后都不能说 Story 是 Done 了。以后咱们两边需要达成一个 共识,就是每个 Story 都必须是所有的代码都完成,所有的代码都做过 Unit Test,并 且都提交到代码服务器上,经过测试,才能称作'Done',你看呢?"

Michael 的这些话让关毅无话可说,除了回答"是"。

总结和思考

由于 Scrum 具有高度的可视性,所以团队中谁的状态都隐瞒不了。关于任务 Done (完成)的标准是个非常重要的问题。在一开始总以为代码完成了就算 Done 了,其实这离 Scrum 的标准还差得很远。



"关毅你别误会,没有责怪你们的意思,我们这边也有同样的问题。第1个Sprint 留给我们很多教训,回头我们要利用Sprint回顾会议好好总结一下。"Peter 接着说。

电话会议结束后,关毅看了看徐天。徐天安慰了一下关毅:"小关,别难过,今 天的事不用太放在心上,没事的,以后尽量避免再出现类似的情况就行了。现在还 有点时间,大家看看有什么想说的?咱们简单地讨论一下吧!"

小胡早就憋不住了,几乎在徐天的话音刚落,他就接上了:"今天他们那边怎么会出这么多问题呢,不应该啊,咱们这边好像一直都是好的啊!真是想不通!"

段美女也觉得奇怪: "是啊,真是奇怪,到底出了什么问题? 怎么会搞成这样!"

胖子低下头,沉思了一会儿,猛地说:"都怪你关毅,事先准备好了,你来演示 不就不出问题了。"

徐天摇了摇头,对大家说:"关毅就应该这么做。胖子,你那么做不叫演示,而是掩饰!另外,我想这次出问题,除了计划不足、准备不足,从技术上讲,是不是没有考虑环境的差别?比如 Michael 他们的系统环境会不会和我们不一样?"

到底是徐天。关毅恍然大悟,刚才忘了问了,Michael 他们用的是什么系统环境,怎么会出现那么多错误。岂止是刚才忘了问,在一开始就应该把需求彻底搞明白,不能想当然。做 Sprint 计划的时候不能光图完成得快,还应该把方方面面的问题都考虑到。这句话说起来容易,要做到完美还真是很难。好在现在是一个 Sprint 结束后就可以暴露问题,而过去要很晚才能发现。

关毅突然想到下一个 Sprint 很有可能在苹果的 Mac OS 上开发,自己和团队对苹果电脑一无所知,如果能早拿到苹果的机器,就可以提前熟悉一下。另外,关毅十

维松Scrum之旅

——敏捷开发故事

分怀疑 Michael 他们用的都是 Mac OS,如果拿到机器,可以立刻试一下,说不定能 赶在 Sprint 结束之前就把问题解决掉。于是他对徐天说:"徐天,可能要麻烦你尽快 帮我们找 3 台 Apple 的机器,上次和你提到过的。我们想在上面简单地测试一下,说不定能重现 Michael 他们的问题,并且在最短的时间内解决这些问题。"徐天听了关毅的话,表示他立刻就去想办法。

关毅接着对大家说:"今天下午咱们继续召开我们的 Sprint 回顾会议。这次没有时间和温哥华他们一起开,那咱们自己来吧!我一会儿去订个会议室。徐天,下午你来吗?"

徐天查看了一下自己的日程,面露难色:"我今天下午还要去别的办公地点开会,恐怕不能参加你们的会了。"

关毅笑着说:"没关系,下次一定要来哦。回头把会议记录发给你。"

整个过程,关毅觉得徐天并没有责怪自己和团队,一下子感觉好多了。

总结和思考

发现问题不是坏事。Sprint 评审可以帮助团队尽早地发现问题,尽早地得到使用反馈。



散会后,徐天就去找其他部门的同事借苹果电脑了,关毅他们则开始准备下午的 Sprint 回顾会议,这样 Peter 他们就可以看到了,也能帮忙检查一下其他的问题。

安排好其他事情,关毅给 Peter 写了一封 E-mail。



Peter:

不好意思,今天的演示我们准备得不够。我们的疑问是,你们所用的系统平台是不是和我们的不一样啊?

关毅

发完了邮件,关毅就马不停蹄地去订会议室了。还好,有间刚好容得下 4 个人的会议室还空着。

Sprint 1 回顾——批评和自我批评

中午,关毅简单对付了一顿午餐,想尽快看看徐天那边的消息,可是结果真令 人失望。徐天去开会了,走的时候什么也没有留下,看样子机器是没借到。

关毅有些失落。他漫不经心地打开电脑,发现 Peter 已经回复了邮件。

Guan:

需要道歉的应该是我们。我已经和 Michael 交流过今天的事了,Michael 在事先有些情况没有了解清楚的时候就下了结论,特向你们道歉。Michael 用的是 Mac 下的 Safari 浏览器,关于是否支持这个浏览器,我们在这个 Sprint 开始计划的时候没有特别声明。不过管理层在最早的时候已经明确决定要支持 Mac 下的 Safari 浏览器了,这已经作为另外一个 Story,要在下一个 Sprint 完成。这是一个很大的 Story,不会很快做完。所以,请不要担心,更不要急着赶在这个 Sprint 结束前把 Michael 发现的问题改掉。

希望你不要介意,Michael 是我的经理,他非常关心项目的进展情况。实际上,我们的 Product Owner Greg 还是很满意的。我简要地看了一下你们提交的代码,整体的风格和语法都相当不错,只是还有些小问题,我们会帮你们做代码重构。

对于你们的工作,我由衷地表示感谢,请继续努力!

还有,马上就要到下一个 Sprint 了。除了把剩下的工作和问题解决掉, Greg 会在明天把下一个 Sprint 的工作发给你们,你们就可以开始着手做 Sprint 计划了。

Peter

看过 Peter 的邮件,关毅觉得好过多了。



在 Sprint 回顾会议上,除了继续反思这次 Sprint 评审的失败,关毅还向团队传达了 Peter 邮件的精神。小段他们都表示理解,只是小胡有些不服气,觉得代码重构的工作完全应该由自己来做,如果由 Peter 他们来做,岂不是被别人抢了功劳。关毅还是非常客观地评论了这件事:"以我们目前对产品的理解,确实不适合做代码重构。我认为咱们也不用想得太复杂了,如果我们有好的想法或提议,一样可以向他们提出来啊!我们还只是刚起步,让 Peter 他们完完全全地信任是不可能的,我们只有靠以后的工作来不断地证明自己,你们说呢?"

听到这里, 所有的人都不说话了, 大家都觉得关毅有道理。

看到大家都沉默下来,关毅转身在白板上写下了"批评与自我批评"7个字,随即说道:"今天呢,我想我们的这次回顾,事实上是第一次回顾会议。先说教训,咱们来一次批评与自我批评活动,大家可以畅所欲言,可以无所顾忌,在座的任何人的问题,只要你觉得有问题,尽管说出来。开始吧!"

Scrum 博士扫盲



• Sprint 回顾会议

Sprint 回顾会议由产品责任人、Scrum 团队和 Scrum Master 参加,会议中需要讨论有哪些好的建议或方法应该被采纳,在 Sprint 中有什么做法不可取,有哪些做法效果很好,应该继续下去。

Sprint 结束后,Scrum 团队回顾刚结束的 Sprint,对其进行总结和反思,使整个团队能持续成长。

Sprint 回顾会议的形式可以比较随意,主要做到以下这些方面就可以了。

Scrum Master 首先给大家看 Sprint Backlog,总结这个 Sprint,大家讨论在这个 Sprint 中发生的一些比较重要的事件。与会人员轮流发言,每个人都有机会发表自己的 意见:他认为哪些方面做得好、哪些方面需要改进、如何改进等。还要对比 Sprint Backlog 中各个 Story 的估计值与它们的实际完成时间,如果差距太大,就应该好好分析出现这种情况的原因。

在会议即将结束之前, Scrum Master 总结会上的讨论成果,即"团队如何才能在

下个 Sprint 中做得更好 "。

总之, Sprint 回顾会议的宗旨就是: Scrum 团队如何在下一个 Sprint 中做得更好!

小胡第一个站了起来,关毅一看,乐了:"不用站着发言,这算是讨论,坐着就可以了。"

小胡看了关毅一眼,并没有坐下:"我还是觉得站着说好一点,至少感觉上好一点。先说一下我自己吧!我的最大问题可能就是过于自负,一看到原来电子商务通的代码后,满脑子想的都是如何重构它们,甚至想重写它们,结果就没有去仔细研究代码的实现逻辑,以至于后来出了问题。回过头一检查,发现在原来的代码里其实都有相关的说明和实现。我想,以后的工作中再也不会发生类似的事情了。我的话完了。"

小段接着小胡的话说了下去:"我觉得现在是有个问题,就是给我们的需求一点也不详细,工作起来都不知道是否能满足当初的设计,咱们的工作对文档的重视还是不够的。其他的,也没什么了。"

接下来,胖子的话就让关毅有些不好意思了:"我觉得一个很大的问题就是,我们的计划都是老大一个人做的,而且时间也都是老大定的。其实我早都想问了,Sprint计划是不是我们也应该参加啊,我觉得有些问题大家一起讨论一下,结果可能会更好吧。"

关毅接过话:"胖子说的很对,这个我会考虑,以后我们争取能一起来做。这次主要是因为我对 Scrum 相对比较熟悉。大家还有别的问题吗?"

- "我们对敏捷和 Serum 的认识还需要继续深化啊。"
- "一开始做 Sprint 计划的时候把学习的时间考虑得太少。"
- "每日 Serum 会议的时间太长。"
- "加班不对。"
- "我检讨,有时候没有及时更新 ScrumWorks。"

维松Scrum之旅

——敏捷开发故事

"我也检讨,有时候没有及时沟通,自己一个人闷头干。"

"我也检讨……"

在一片检讨声中, Sprint 1 就这样结束了。

总结和思考

Sprint 回顾会议通常是最容易被忽略的。然而,Sprint 回顾会议其实是非常有用的,它是整个 Scrum 开发框架中第二重要的事件(最重要的是 Sprint 计划会议),因为它是让 Scrum 团队成长和进步的最好的机会。如果不开 Sprint 回顾会议,不久以后你就会发现,你的团队在不断地犯着同样的错误。在本章中,关毅尽管已经了解了不少关于 Scrum 和敏捷的知识,不过,离真正的理解和运用娴熟还差得很远。



Sprint 2——计划与变化

关毅他们开始了第 2 个 Sprint 之旅。然而他们还是遇到了很多问题:首先是临时休假的小胡打乱了他们的计划;然后,Sprint 进行中遇到的技术问题又难住了他们一阵子;还有经理徐天临时安排给关毅的任务也够他们受的;最后,他们发现有一个 Story 可能完成不了了,是加班,还是……



Sprint 2 的目标

棘手的新任务

墙上的时钟飞快地旋转着,转眼就到了周五。下周一就要开始新的 Sprint 了。不过,今天这个"逃亡星期五"也绝对不轻松,因为关毅要和加拿大那边的 Product Owner Greg 商量决定新的 Sprint, 也就是 Sprint 2 的任务,时间就在上午 10 点。

到了约定的时间,关毅临时找了一个会议室通过电话和 Greg 进行 Sprint 计划会议。关毅发现自己的团队成员好像都很忙,事情也都很多,就没有叫大家一起开会。

总结和思考

不应该有理由和借口抛弃团队。



这个会议的主要内容是商讨和决定下个星期即将开始的 Sprint 2 的任务。不过, 虽说是一起商量决定,实际上关毅是说不上什么话的。所有的需求都在温哥华,目 前他们只能跟着做,况且自己这边本身对产品也不是很熟悉。

Greg 根据产品的 Backlog 以及优先级的排序,很快就决定把完成"Mac Safari 浏览器支持"以及"PDF 精确打印表单"这两个大的任务分配给了关毅的开发小组。看上去,Greg 对整个产品要做什么、要先做什么都是很明确的,估计之前很早就有了这个 Sprint 的计划。在这两个 Story 中,果然列出了"Mac Safari 浏览器支持"的任务。而关毅也没有提出太多的意见和建议,考虑到自己目前的状况,他觉得还是不要发表太多意见得好。大概是温哥华那边考虑北京这边有与 Mac 相关的开发小组,经验应该会比较丰富,所以才把这个 Story 分配给关毅他们吧。可是温哥华那边哪里晓得,关毅他们小组实际上连见都没见过苹果机!不过尽管困难很多,关毅还是兴奋地接受了这些任务。他此时想的是任务越多越好,越难越好,这样才能显示自己团队的战斗力。但是,关毅在会议的最后还是告诉了 Greg 他们团队目前对 Mac 的了解情况。Greg 竟然一点都不吃惊,甚至还很欣赏关毅能够这么坦诚地告诉他实际的情况。不过,Greg 仍然觉得北京这边有技术优势,毕竟关毅他们可以很容易地联系

Sprint 2——计划与变化

到 Mac 开发组。当然,沟通得怎么样,能得到多少帮助,就要看关毅的本事了。

总结和思考





关毅打开 ScrumWorks,这个 Sprint 的 Story 已经罗列在上面了(如图 5-1 所示), "Mac Safari 浏览器支持"以及"PDF 精确打印表单"这两个大的功能被分解成了 4个 Story。虽然这些 Story 在 Wiki 上都有相对简明扼要的说明和需求分析,尽管刚才也已经和 Greg 有了一些基础的讨论,但关毅还是觉得不是那么清楚。他沏了一杯绿茶,盯着电脑仔细看了一会儿,微微地感到一阵目眩。

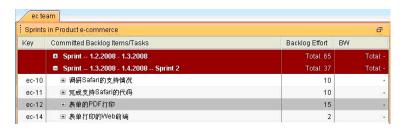


图 5-1 Sprint 2 中的 Story

ScrumWorks 也已经计算出了总的计划人天数,也不知道 Product Owner 估计的人 天数对不对,能不能完成。这些人天数都是由 Product Owner 决定的,感觉上最多也 只能是估计,很难做到精确吧。整个 Sprint 加起来是 37 人天的样子,从上个 Sprint 大家工作的实际时间以及效率上分析,关毅感觉还算差不多。

Wiki 上的需求文档

关毅从会议室出来,看着屏幕上的 Story 列表,任务真是艰巨啊。轻轻地叹了口气之后,他打开了 Wiki,开始详细地了解这次 Sprint 任务的背景。

Wiki 上的任务需求写得很是简洁扼要,这让关毅很不习惯。要知道,自己以前的公司,写个需求分析文档都要几十页,而如今眼前的这份需求文档,看上去最多两页。





Sprint 2 开发任务背景之 Safari 浏览器支持

用户需求

上一个版本,我们的产品支持了 Firefox,从而让我们的产品从单独的 Windows/IE 平台延伸到了 Linux 平台,这使得使用 Windows 平台以及 Linux 的终端用户访问电子商务通都没有什么障碍。但是尽管 Mac 平台的用户可以使用 Firefox 浏览器访问电子商务通,但绝大多数的 Mac 用户还是习惯于使用机器自带的 Safari 浏览器。由于我们的电子商务通没有测试过该浏览器,肯定会有一些或大或小的问题,比如页面布局等。为了支持 Mac Safari 用户,需要对其进行调研,加强产品的浏览器兼容性,扩展电子商务通的客户领域。

开发方案

- (1) 使用 Safari 调研和测试电子商务通。
- (2) 记录并归类各种不兼容现象,简要地给出解决方案。
- (3) 更改代码,增强电子商务通的浏览器兼容性,使其支持 Safari 浏览器。



Sprint 2 开发任务背景之 PDF 精确打印表单功能

用户需求

浏览器自身具有的打印网页的功能,是可以用于我们产品的打印的,但是我们的用户认为浏览器自身对打印出来的纸张加了其他一些无关的东西,诸如页边距、网络地址信息等。另外,用户往往希望只对网页其中的某些报表、表单进行打印,而对一些外围修饰性的图片等不予打印,比如打印用户的订单等。用户希望有一个像 Word 打印一样的能精确控制的打印功能,而这一功能不会受到浏览器的限制。在此之前,电子商务通的某个客户也曾经提出需要将这些报表、表单等信息保存为 PDF 以便对账等的需求。

综合以上的用户需求,我们需要开发 PDF 精确打印表单的功能。用户通过点击页面上表单内的某个打印按钮,激活电子商务通后台的打印逻辑,按照当前所处网页中表单的内容以及布局,精确地输出 PDF 文档,并提示用户下载和保存。

开发方案

- (1) 利用某种 PDF 第三方库生成 PDF 文档。
- (2) 收集表单数据。
- (3)打印表单布局元素。
- (4)打印表单数据信息。

关于 PDF 的内容,请参见......

关于收集表单数据等内容已经有一些具体的技术文档,请参见.....

总结和思考



Wiki 是个不错的敏捷项目文档管理工具。

虽然是一个看上去简洁明了的开发背景说明,关毅还是觉得有些头皮发麻,他想:这可真是 Agile 啊,文档写得实在是太精简了。

关毅看完了项目的需求背景,总算是心里有了一点底。不过有了第 1 个 Sprint 比较失败的经验,他还是有些不放心。

收拾了一下电脑桌面,关毅又给 David 发了封电子邮件,算是交流一下最近关于 Serum 的心得体会。另外,关毅也想看看他对今天的这个会议有什么意见或者建议。关毅生怕 Sprint 2 再次出现不必要的遗憾。



Hi, Davida

我们今天刚刚开始第2个Sprint了。

今天上午我和 Product Owner 开了个短会,主要是商议下个 Sprint 的内容,大致定下了 4 个 Story 的任务。我决定下周一召集团队成员继续开这个计划会议,进行细分任务、估计时间和分配人员等,然后就算是正式启动这个 Sprint 了。



目前我还有一些 Scrum 实践上的困惑。比如说,今天的 Scrum 计划会议是我和 Product Owner 两个人商讨新的 Sprint 计划。因为我们的团队成员都比较忙,我就没有招呼大家一起开会。你也知道,人太多反而不太容易做决定。不过这样的做法是不是有问题呢?Sprint 计划会议到底需不需要大家共同参与呢?

另外,在与 Product Owner 的沟通中,在 Story 的选择上,他的主导性要大很多,我们基本上只有听的份,能不能让我们的团队来选择 Story 呢?

暂时就这两个问题了。期待你的回复!致谢。

关毅

没有不可能完成的任务

中午, 关毅胡乱吃了顿工作餐, 就开始了下午紧张有序的工作。

首先,他给小组成员发了封电子邮件,概括地通报了上午的会议内容。他知道 大家都没有什么相关的开发经验,于是他要求大家利用周末的时间稍微熟悉一下有 关的内容。因为下周一他准备召开组内的 Sprint 2 计划会议,大家总应该有些准备 吧!不打无准备之仗,何况往下的工作看起来还是如此的艰巨。

写完邮件,关毅走到茶水间冲了杯咖啡,凝神望着窗外阳光下闪闪发光的树叶,想稍微整理一下思绪。突然,他想起来有个以前认识的同事在 E 公司另外一个产品的研发小组,已经做了快两年了。他隐约记得他们的那个产品有输出 PDF 的功能,所以这个同事应该在这方面有比较多的开发经验。想到这里,关毅兴冲冲地回到座位,找到对方的电话拨了过去:"是王宇吗?我是关毅啊。最近忙吗?"

"还行吧!有事?"电话那头的王宇说。

"是这样,我们要为产品添加一个使用 PDF 打印表单的功能,我在这方面没什么经验。我记得你们的产品好像有这个功能,你了解吗?能给我们讲讲吗?"

"呵呵,那个不难,而且那个打印功能我也参与了开发。PDF 打印有一些第三方的类库可以用。这样吧,找个时间,你安排一个小会吧,我做个 PPT,随便讲讲,

算是给你们做个简单的入门讲座吧。讲的内容是很简单的,讲得不好不要怪我啊,呵呵。"

"怎么会,太好了!谢谢了,到时候我再打电话给你, Bye。"

关毅放下电话,感觉轻松了不少。有这位同事的帮忙,PDF 打印上的技术问题以后都可以问他了。关毅不由得露出了一个安慰自己的笑容。不过,虽然第 2 个问题有了帮手,但是对于支持 Safari 的任务,他还是不敢怠慢。

关毅开始上网搜索一些相关的资料,包括 Safari 浏览器的特性、对标准的支持等,虽然有些资料和文档讲得不是太细,但总归还是比较全面的。

关毅忙着搜索材料、组织文档,还在苹果的官方网站上逛了半天。他跃跃欲试,想要实现 Safari 对网页的呈现效果,这才猛然想到,没有苹果电脑怎么开发啊?他拍了一下脑袋,赶紧问徐天有没有搞到苹果电脑。

徐天的进展很顺利,用他的人格魅力很容易就借来了 3 台,都是别的部门的暂时闲置的。

于是,关毅找来胖子和小胡,3个男人从库房里拖出3台机器。然后,每个小组成员就都领到了崭新的苹果电脑。大家都从来没有用过苹果电脑,拆了包装,接上电源,感觉就是不一样,每个人都很兴奋,开机之后都对它爱不释手。除了这3台机器,徐天还告诉关毅一个好消息:E公司购买了苹果的服务,有问题可以寻求苹果官方的技术支持。

看着大家试用苹果机的高兴劲,想到拥有了苹果官方的技术支持,关毅觉得现在的自己全身充满了力气,对将要到来的下周一也充满了期待……

总结和思考



善于寻求帮助是个好习惯,不仅仅是针对敏捷开发项目。

回到家,关毅在自己的博客上记下了今天的流水账。他总是希望有对 Scrum 感 兴趣的人给他留言,然后一起讨论。不过,至今访问的人还只是精确到个位数。可能在国内 Scrum 还不是很火吧,至少关毅是这么认为的。

発**がScrum之**が 一般接开发故事



关毅的 Scrum 日记: Sprint 2, 收到任务安排

星期五,阴,有小雨。

今天是 Sprint 1 的最后一天了,我收到了新的安排。下一个 Sprint,我们要开始做 Safari 浏览器的支持以及表单的 PDF 打印功能。到底应该怎么做,我一点经验也没有。我跟胖子、小胡、段美女大致说了一下,他们也很茫然,因为他们和我一样,也都没有经验。唉,只能走一步算一步了。自己是 Leader,总不能说不会就放弃吧!不能够、不放弃、不抛弃,三多精神永远是我的榜样,嘿嘿:-)

阿汤哥不是说了吗,Mission Impossible,哈哈,世界上没有不可能完成的任务。

下午我上网简单搜索了一下相关的技术文档,虽然没有什么头绪,不过至少有了一些线索。下个星期我会跟胖子他们开个会,详细讨论一下这些问题。但愿我们能有好运吧!

BTW,今天跟 Product Owner 开会,感觉有一些小地方还是有点不太明白,是不是要发个电子邮件再确认一下呢?

另外,我们这边的任务现在基本上都是他们那边直接安排过来的,以后 我们能不能有自主权呢?还有,这种异地开发的模式,Agile 的开发方法,真 的很好吗?

我感觉我的问题真的还有很多,不过这也许不是短期能够解决或者知道 答案的。继续努力吧!

Sprint 2 的计划会议

日历撕掉了两页,放松了一个周末。周一,Sprint 2 出发!

Sprint 计划会议是属于整个团队的

到了公司,关毅像往常一样先整理周末两天的电子邮件。看到 David 的回信赫然在列,关毅便迫不及待地打开阅读了。



关毅:

你好!

恭喜你们已经进入 Sprint 2 的开发,想必你一定进步很大。敏捷开发就是需要不断实践,每个团队遇到的问题都不会完全一样,你需要不断思考。

你在周五与 Product Owner 商议了 Sprint 2 的开发计划,没有召集团队成员,这不是一个好的 Scrum 实践。召开这样的计划会议,团队成员应该是要到场的。要知道,你虽然是他们的 Team Leader,但是毕竟你一个人的作用是有限的,而且团队成员的意见往往更具有代表性和针对性,你一个人不可能面面俱到。要时刻记住,在敏捷的理念里面,一直都在反复强调自我管理的团队,要淡化层级观念。Sprint 任务的选择,主要决定权属于团队。需求都是从 Stakeholder 那里来的,而 Stakeholder 包括了你们的客户,你们的管理层,还有你们自己。Product Owner 有权维护产品 Blacklog,而 Scrum 团队有权从产品 Backlog 中选择适当的任务作为 Sprint Backlog。所以,由 Product Owner 一个人完全主导是不对的,你们应该参与到讨论需求和计划的活动中。

另外,你决定在周一召开组内 Sprint 计划会议的下一个阶段会议并细分任务,这个时间间隔有些长。一般而言,像你遇到的这种情况,最好是在上午与 Product Owner 开完第一阶段的计划会议后,紧接着召开第二阶段的会议。当然,灵活一点的话,你也可以在下午和团队成员一起消化一下第一阶段会议的任务安排,然后接着召开第二阶段的会议,这样任务就能够及时地



下达并明确下来。当然这也只是 Scrum 最佳实践的要求之一,具体情况也要另外分析。不过,由于这两个阶段的会议中间有周末的两天时间,打断了连续性,所以应该不是一个正确的实践方法。我建议你以后开计划会议最好连着开,压缩一下第一阶段的会议时间。

至于选择 Story 的主导性问题,由 Product Owner 主导是没有问题的。但是你和你的团队应该充分地提出意见和看法,以确保自己的任务合适,并能够在预期时间内完成。

其他的也需要你自己慢慢在实践中体会。就这样吧,下次再见,我期待 着你更大的进步。

David

看完 David 的邮件,关毅认认真真地做了一些笔记。他若有所思地端起茶杯,回忆起上周开会的过程。关毅觉得这种一对一的与有着丰富 Scrum 经验的人交流的感觉真是再好不过了,以后这种交流应该更频繁一些。

关毅在心里默默地想着,不过他还是很快回过神来,因为预定的开会时间到了。David 说这两个阶段的会议应该连在一起开,现在已经算是晚了。想到这里,关毅赶紧夹着笔记本电脑来到了预定的会议室。虽然他比预定的时间稍微早到了几分钟,不过小胖、小胡、段美女都已经在那里等他了,他们一个个好像都有很多话要说,不过又都很小心的样子。关毅不慌不忙,他接上投影仪,打开了 SerumWorks,4 个 Story 投射在白板上,排列得很整齐。

关毅说:"好吧,我们现在准备开会了。首先我要向大家道歉,周五的会议没有叫上大家一起,这是我的过错。我们可能会对这个 Sprint 的任务有很多疑问,如果大家都在现场的话,我们就可以及时沟通并与 Greg 协商了。我以后会注意的。今天的主要议程是讨论我们这个 Sprint 要完成的计划任务,这个会也就是计划会议的第二阶段了。大家都知道,我们的任务已经确定了,是做 Safari 浏览器支持以及 PDF打印功能的开发。我知道这次的任务有些重,大家也没有相关经验,但是这是一个机会,是一个考验我们的机会,也是体现敏捷精神的机会。敏捷不仅仅是流程上的敏捷,也要求人员自身的敏捷。快速掌握新的技能也应该是敏捷的精神之一嘛!呵呵,大家说是吧?"

关毅的一席话,大家听了都很受用,脸上都露出了笑容。

休假问题

会议刚开始,小胡有点不好意思地插了句话,因为他准备在这个 Sprint 的第三 周请假一周,回老家探亲。

关毅心里咯噔一下,有点急了。工作刚刚上了轨道,新的任务又这么重,怎么能休假呢?不过他转念一想,这也是没办法的事情,这是员工的个人福利。但是 Sprint 怎么办呢?整个计划是不是应该在这个前提上修改一下呢?

关毅拿出了一张纸,开始估算这个 Sprint 的工作量和大家的工作时间。上一个 Sprint, 关毅还不太会做这个工作, 这次他显然熟练了许多。他在纸上计算着: 1个 月的迭代周期总共是20天,4个人总共是80人天(5×4×4=80),但是由于召开 计划会议、最后的评审会议以及每天的每日 Scrum 会议等,都需要花费一些时间和 精力,这样实际的工作时间基本上只剩下 75 人天。关毅本人作为 Scrum Master,经 常会有一些与开发工作不相关的管理、向上级汇报、讲座等活动,所以他自己大概 只能有 50% 的时间投入在 Serum 流程中, 所以只剩下 65 人天(之前关毅的工作量 是按照 20 人天计算的, 折算为 50% 后, 是 10 人天)了。而现在小胡探亲一周, 也 就同样减少了5人天的工作时间,那么就只剩下60人天了。每天是8小时工作制, 最理想的状况是大家每一分每一秒都在 Serum 上的任务中, 但这显然是完全不可能 的。大家每天都有很多电子邮件要处理,杂事也同样不算少。而且关毅带领的是新 团队,在做 Scrum 上的新功能的时候,同时还要花时间了解整个产品架构、跟外国 团队进行沟通等。所以,关毅认为应该要乘以一个比较小的系数来计算用于这个 Sprint 的工作时间。关毅好像之前看到培训教材上说,这个系数大概是 50%~70%。 于是,他决定采用最小的系数来计算。关毅认为,随着大家对 Scrum 了解的深入和 对产品认识的加深,这个系数会逐步提升上来,这样大家的工作效率也就相应地提 高了。这样的话,大家用在这个 Sprint 中的时间是 $60 \times 8 \times 50\% = 240$ 小时,再换算 到每天8小时的完美状态,则是整整30人天。

关毅知道这个效率肯定是相对较低的,但是他看了看 ScrumWorks 上 4 个 Story 的时间加总: 10+10+15+2 = 37 天,也差不了太多。看来 Product Owner Greg 也认为



——敏捷开发故事

大家的效率会比较低,所以也没有把工作量安排得太大。不过关毅还是觉得事实上可能 Greg 估算的工作量会比较准确,也许真的需要 37×8,也就是将近 300 个小时。如果小胡不请假的话,可能状况会相对好一点。

关毅在想,自己是不是应该去商量一下,再调整任务呢? 比如说减少 PDF 打印中的某个子功能,安排到以后再做。但他感觉很难,而且他也不是很好意思和 Product Owner 去说。所以关毅只能提醒大家:"以后大家如果需要请假,最好多往前提前点计划,早点说,这样我们的 Sprint 计划才比较好安排。"

不过,段美女显然没有给老大面子:"咱们不是 Agile 嘛,我觉得请假这种例外处理自然应该考虑在我们的流程上面。"

关毅虽然也有些认同她的观点,不过他还是将这个问题记在了心里。

休假的问题就这样过去了,会议开始正式进入主题——技术问题。

技术问题

关毅找来一只白板笔,刷刷刷地在白板上列了几行字,内容是 3 个技术方面的问题。

- (1) Safari 浏览器的支持。
- (2) PDF 第三方类库的使用。
- (3) 表单元素的打印。

这 3 个问题是目前他们遇到的比较大的困难。关毅说:"我知道大家都不熟悉 Safari 浏览器,苹果机就更别提了。对 PDF,我们也只是使用阅读器来浏览一些技术 文档而已,没有生成 PDF 的技术经验。所以我想让大家结合这 3 个技术问题谈一下 自己的看法,说说自己这两天的学习成果,评估一下技术风险。胖子,你说说吧。"

胖子认为 Safari 的支持并不是那么困难,毕竟大家之前实现了支持 Firefox,尤其是对于一些 Web 标准,也都有一些经验了。至于 Mac,小胖认为这基本上不会影响开发,大家只要关注 Safari 浏览器就好了。

Sprint 2——计划与变化

小胖说完,小胡接着他的话说开了:"周末我大概在网上搜了搜相关的内容。我认为,可以分别给这 3 个技术难点打个难度分数。Safari 浏览器的支持,难度 5; PDF 第三方类库的使用,难度 4; 表单元素的打印,难度 7。总而言之,对这次 Sprint 我还是很有信心的。但是我感觉我们还是缺少一个很完整的需求文档。比如说,对于 Safari 浏览器的支持到底需要支持到什么版本?表单的精确打印到底需要做到什么样的程度?这些我都不是很明白。我知道 Agile 的流程对文档不是很看重,但是没有那些文档,我们怎么做呢?以后如果有新人加入团队,要怎样才能了解项目背景和功能状态呢?"

接下来轮到段美女。她基本上同意小胖、小胡对于 Safari 浏览器支持的意见,也认为这不是很困难的工作。她另外补充了一些关于 PDF 打印的内容: "我在网上找了几个第三方 PDF 输出类库,看了一下他们的例子,好像都差不多,我还不是很清楚我们要使用哪一个。"

关毅仔细听完大家对技术问题的分析,综合自己这几天的学习情况,大致上也和大伙的想法差不多。不过针对 PDF 打印的第三方类库的选型问题,他还是比较小心的。关毅认为暂时还是不要急于下结论,决定用哪一个类库进行开发,目前应该找出不同类库的优缺点,然后再最后选择一个正式进行开发。当然还有王宇的讲座,他应该会提供一些非常有用的信息。

而对于小胡提出的 Agile 文档的问题, 关毅告诉团队: "过去我没到 E 公司之前, 在 X 公司上班, 文档多如牛毛, 而且要求特别具体。虽然花了大量的精力去编写文档, 但最后很多文档都没人去看, 所以很多文档实际上是没有价值的。Scrum 作为一个敏捷开发框架, 文档的实际地位确实不像在瀑布式的开发流程中那么高, 但是某些简要的文档还是需要的, 毕竟文档能规范我们的开发, 提供一些用户的需求。但是这种文档的形式是可以多变的。比如说, 我们可以使用 Wiki 来管理项目文档, 不管是技术上的问题, 还是用户的需求, 都可以采用多人协作的方式来完成。我们还可以使用会议纪要的方式将遇到的问题或者得到的答复记录下来, 作为一个开发的凭证。而文档提供的具体时间或者文档撰写的负责人, 我认为也可以是很灵活的。因为一旦采用了 Agile 的流程, 我们就要适应这种时时有可能发生的动态改变。也许我们可以在完成了代码之后再来编写这个文档, 甚至可以说, 代码里面的注释、方法名等都是文档。所以关于文档的问题, 我觉得也同样是敏捷的。至于小胡说的很具体的关于 Safari 浏览器支持版本的问题, 可以下次开会的时候讨论, 然后以会议

维松Scrum之旅

---敏捷开发故事

记录的方式写下来,贴到 Wiki 上去,修改目前那个特别不完善的需求背景的 Wiki 页面。而且关于对 Safari 浏览器的支持,我们还有一个很重要的前期调研任务,这 基本上可以看作是一个完成最终需求文档的任务。大家明白我说的这些吗?"

大家点了点头,都表示理解了。关毅给大家打了打气:"大家提出的问题都很好,也对我们要做的工作很有信心,这都非常好!至于需求之类的内容,也是可以慢慢明晰的。待会儿我们来估计一下工作量,分解一下这4个Story,然后大家挑一下任务,可不能太挑三拣四啊,呵呵。"

总结和思考

到目前为止,出现了不少概念。Story、Backlog、Task(任务),它们之间的关系是怎样的呢?在 Scrum 中,产品要完成的功能清单叫做产品 Backlog。每一个 Backlog 项通常也叫 Story,因为它是由 User Story 来描述的,一个 Story 是由一个完整的 User Story 来描述的。有时候,一个比较复杂的 Story 也可以分解若干个成更小的 Story。Task 是任务,在具体实现每个 Story 的时候都要将其分解成具体的任务,比如编码、测试、调研、Code Review 等,这些都是 Task,而不能称为 Story。

用扑克牌游戏来做估计

技术上面的问题,大家其实都很清楚了,只是这些问题目前还没有十分确切的答案而已,而这也带来了任务评估上的难度。每个 Story 都要细分成 Task,而且每个必要的功能点完成后都要有 Code Review 这样的 Task。

针对第 $1 \uparrow Story$,调研 Safari 的支持情况,关毅和大家七嘴八舌地将它细分成了以下 $6 \uparrow Task$ 。

- (1) 使用 Safari 运行电子商务通, 检查基本功能。
- (2) 完成代码, 保证基本功能的运行。
- (3) 深入调研 Safari 上功能的可靠性。

- (4) 深入调研 Safari 上显示效果的问题。
- (5)针对功能问题,总结方案文档。
- (6)针对显示问题,总结方案文档。

由于暂时不确定 Safari 到底有没有破坏电子商务通的功能,在显示效果上和其他浏览器差异大不大,具体的细节也只能等调研之后才能知道,所以感觉这些 Task 定得有些笼统,但是目前也只能做成这样。

第 2 个 Story, 完成支持 Safari 的代码,则依赖于第 1 个 Story 的调研。相应地,大家定出了 5 个 Task。

- (1) 完成功能方面的代码。
- (2)完成显示效果的代码。
- (3)完成技术开发文档。
- (4)单元测试。
- (5) Code Reviewo

而针对剩下的关于 PDF 打印的两个 Story, 在具体开发层次上面, 分的 Task 大体上和这个 Story 差不多, 也有完成代码、Code Review 等。不过由于 PDF 的牵涉面相对大一些,不太需要调研方方面面的内容, 所以 Task 分得较为细致, 也相对多。

Task 就这么确定了。接下来,就是估计时间、分派任务了。

估计时间是最让大家困惑的。如何估计每一个 Task 所需要的时间,并且能够保持一定的准确性,实在是一件让人伤脑筋的事情。

实际上,现在每个 Story 的人天数都是 Greg 自己估计的,并没有与小组成员有太多的商量,所以就目前的情况来说,准确性也绝对不会很高。因为每个 Story 的时间毕竟是所有 Task 加总的结果,没有经过任务的细化,没有准确评估每个任务的时间,最后完成 Story 的时间又怎么会很准确呢?

维松Scrum之旅

----敏捷开发故事

只有对每个任务进行评估,最后才会得到更加合理的计划。对于每个任务而言,有的时间估计得多,有点估计得少,加起来可能会出现折中的现象,所以最后的结果也许会近似准确。但是如何让每个任务评估尽可能准确呢?关毅打算尝试一下新玩意儿。

只见关毅神秘地笑了笑, 摸了摸口袋, 掏出一副扑克。

关毅带来的是"计划扑克"(Planning Poker),这是在敏捷培训课结束的时候,培训老师 David 送给他的。

Scrum 博士扫盲



• 计划扑克 (Planning Poker)

所谓"计划扑克"(Planning Poker) 是一种标有各种数字的扑克牌。参加游戏的人每人各拿一叠扑克牌,牌上有不同的数字。

客户或者产品责任人为大家挑选 1 个 Story (Backlog), 并简单解释其功能,以供大家讨论。

每个游戏参加者按自己的理解来估计完成这个 Story 所需的时间,从自己手中的牌里选 1 张合适数字的牌,并发给大家看。游戏参加者各自解释选择这个数字的原因,尤其是数字最大和最小的人。

根据每个游戏参加者的解释,重新估计时间并再次出牌,直到大家的估计值比较平均为止。

在这个游戏中需要注意的是:首先,这不像普通的扑克游戏,不是轮流出牌,而是 大家考虑好之后同时出牌,这样就可以避免后出牌的人被先出牌的人干扰;其次,要告 诉团队成员,他们需要估计所有的 Story,而不仅仅是他们自己将要做的那些部分,比 如测试人员不能只估计测试工作所需要的时间。

关毅在参加 Serum 培训时,还和培训小组成员一起做了一个关于计划扑克的小游戏,让他记忆犹新。

培训老师 David 首先让大家分组,每5个人一组,并发给每个小组一副专用的

扑克牌。这不同于普通的扑克牌,每张牌的分值分别为 1、2、5、7、10、20 和 100。 然后, David 转身在白板上写了 4 种动物,分别是鱼、大象、狗和猫。大家都觉得非常奇怪。

David 说: "每个小组 Sprint 计划的任务就是给出衡量这 4 种动物大小的数值。 在这个练习中,每种动物就相当于 1 个 Story,动物的大小就相当于完成这个 Story 所需要的时间。我给你们 5 分钟的时间,我是 Product Owner,有问题请问我。"

大家心想,这还不简单?开始吧!

关毅所在的小组你一言我一语地开始了。

关毅建议先找一个参照系作为 1, 那就找鱼吧, 因为它最小。但马上就有人质疑:"鱼也有大鱼和小鱼之分, 体积大的鱼甚至比猫和狗都大。"

关毅大声说:"这样的话,狗和猫也有大小之分,我们就没法估计了,不如就按一般情况和通常的大小估计吧。从鱼开始吧!鱼就是1,大家有意见吗?"

关毅小组的其他的人都面面相觑,不置可否。

关毅接着说:"我们接着讨论猫吧。"

于是,小组中的每个人想了一会儿,一起出牌。关毅出了 10,有人出 20,最少的人才出了 2。选 20的人和选 2的人需要分别阐述理由,大家争论了很久也没有结论。关毅一看时间,不好,还剩 1分钟,时间快不够了。于是他大声制止了这样无休止的争论,让小组成员重新出牌,结果还是有意见不统一的。选 2的哥们儿还是坚持己见,怎么办?

这时候组里一个嗓门大的老兄说话了: "算了,少数服从多数,选 10 的多,那 就 10 吧。反正时间不多了。"

意见不统一的人虽然心里不爽,但也只能这样了。接下来的"狗"、"象"基本 上大家都是少数服从多数,一次出牌就解决问题了。

游戏结束, David 开始统计每组的结果, 发现每组的结果都相差很远, 千奇百怪, 于是就问大家有什么感想。

其中一位培训学员说:"我觉得很难估计,世界上有那么多的狗,那么多的鱼,每个人都有自己的理解。"

David 说:"我很遗憾,你们没有一个组来问我具体的需求,我作为 Product Owner 是如此的孤独。当需求不明确的时候,你们为什么不问呢?我可以告诉你们是什么鱼,什么狗啊!"

David 继续说:"扑克牌的想法很好,可是在实际操作中,当谁也说服不了谁的时候,大家更喜欢用少数服从多数的方法作出决策,但事实往往是真理掌握在少数人手里,这些人在少数服从多数的情况下没有机会表达自己。"

"在 Sprint Planning 中,扑克牌游戏的目的就是让掌握真理的少数人有机会表达自己,说服大家,避免团队独裁或出现不理性的决策。" David 说。

"我们经常会发现时间不够用,尤其是在争论比较多的时候。所以 Scrum Master 一定要控制好时间,避免偏离方向的争论。如果真的时间不够用,也不要应付了事,可以另找时间讨论。但我建议最好不要这样。" David 说。

不过针对第 1 个 Story, 关毅并没有使用扑克来估计, 他想试验一下所谓的"取平均值"的方法。

关毅通过搜集大家的意见,挨个对第 1 个 Story 进行了时间估计。对于大家的意见有较大出入的,他采取每个人估计 1 个值取平均数的方法。当然,有些太夸张的估计值——比如认为某个任务太简单,一两个小时就可以搞定,又比如夸大估计某个任务的难度,认为需要非常多的时间——这些夸大或者缩小的估计值都被关毅舍弃了。

关毅笑着给大家解释:"这就像跳水比赛的裁判打分,去掉1个最高分,去掉1个最低分,最后算平均分,呵呵。"

不过这样一轮下来,大家还是觉得大部分 Task 估计的时间量感觉偏小。最后关毅接受了大家的部分建议,调整了个别 Task 的时间量。

现在,每个 Task 感觉上都预留了相对多的缓冲时间。比如"Safari 的调研"这个 Story,时间量最后是这么定的:

- (1)使用 Safari 运行电子商务通,检查基本功能: 8小时。
- (2) 完成代码, 保证基本功能的运行: 15 小时。
- (3) 深入调研 Safari 上功能的可靠性: 20 小时。
- (4) 深入调研 Safari 上显示效果的问题: 20 小时。
- (5)针对功能问题,总结方案文档:12小时。
- (6)针对显示问题,总结方案文档:12小时。

在对后面 3 个 Story 进行估计时,关毅秀出了自己的扑克。他右手拿着扑克牌在空中晃了晃,大家都很感到新奇。

关毅分完牌,给大家讲解了具体的操作步骤。大家便开始拿着分给自己的代表时间量的扑克牌,挨个对每个任务给出自己的估计,一直要到大家都有一致的结果,这一轮估计才算完。有些简单点的任务,大家给的时间估计基本上还比较接近,甚至一次出牌就通过了。不过有些相对模糊的任务,大家的分歧很大,经过3、4轮才最终确定下来。

关毅笑着说:"瞧,这就是这个游戏的目的所在。它让所有的人都有机会表达自己的意见,掌握着'真理'的人最后总会说服大家的。"

使用扑克牌估计时间的办法花的时间明显比取平均值的方法多了很多,但是毕 竟这次大家的意见都得到了充分的尊重,而且最终得出了大家都认为合理的结果。

总结和思考

扑克牌背后的敏捷思想是团队里没有绝对的权威,每个人都有可取之处,要避免少数服从多数。不过如果经过3、4轮也不能确定下来,应该怎么办呢?



在人员分配上,关毅之前已经和大家大概讨论了安排,不过最终还是由大家根据自己的能力所长,通过互相沟通选择了各自的任务。



----敏捷开发故事

经过最后的整理,一张完整的任务列表就展现在大家面前了(如图 5-2 和图 5-3 所示)。

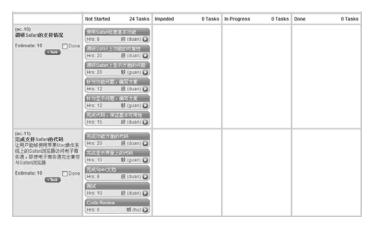


图 5-2 Sprint 2 任务列表 (1)

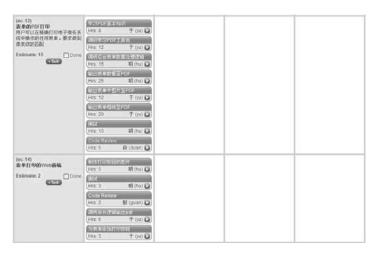


图 5-3 Sprint 2 任务列表 (2)

大家看着最后的计划,有点兴奋。激动人心的 Sprint 2 真的要正式开始了!

回到家,关毅感觉完成了一个重大的任务,条件反射似地打开了自己心爱的博客,写了起来。



关毅的 Scrum 日记: Sprint 2, 开计划会议

星期一,晴。

今天开计划会议,Sprint 2 的计划会议。我、小胖、小胡、段美女 4 个人终于将那 3 个该死的 Story 给分解了。

我们像分猪肉一样将这 3 个任务大卸八块,哦,不,不止八块,总共十几块呢。我们用刀十分小心,我认为已经割得很精细了:-)

我们也最终确定了每个 Task 的估计时间。我们很难确定那些被我们分走的猪肉多长时间可以吃完,消化透,我感觉还是给他们留了一定的冗余时间的。希望大家都能按时完成任务吧!毕竟猪肉里面也还有骨头要啃。最后的时间加总起来跟总的人天数差不了太多,我真的希望我的 Product Owner 估计的总时间是对的。

在人员分配的问题上,我感觉我们真的是 Agile 了,不存在摊派指标的问题,大家各自承担各自的责任,选择自己感兴趣的并且有一定经验基础的 Task。当然,总有一些烂骨头没有人愿意要。不过,最终还是有人站了出来。呵呵,Agile 也要有一定的奉献与拼搏精神。

今天晚上我要睡个好觉了。战斗已经打响了,而胜利的集结号将在 4 周 后的今天吹响。



Sprint 2 的第一周

周一,大家都比规定的上班时间提前不少到了公司,看来干劲确实很足。关毅 瞅了一眼大家的状态,感到很满意。

任务状态要随时更新

于胖子已经开始学习 PDF 文档的一些基础知识。他在 Adobe 公司的官方站点下载了而一些关于 PDF 格式内容的文档阅读了起来, ScrumWorks 中的"学习 PDF 的基本知识"这个任务也被拖到了"In Progress"栏中。

小胡正在研究一些后台数据的相关代码,并开始思考如何收集前台表单数据用于打印,他将"调研分析前台表单数据收集"拖到了"In Progress"一栏。由于这部分内容比较容易,后台的代码也有很多现成的东西,Wiki 上也有些技术文档,小胡还是感觉比较轻松。

段美女则在那儿升级自己的 Safari 浏览器,她的机器自带的 Safari 浏览器已经比较陈旧了。当然,她也启动了自己的电子商务通编程环境,并用 Safari 进行访问。

关毅则因为有其他事情等着要做,暂时没有太多精力跟进他负责的任务,不过他还是忙里偷闲跑过去问了一下段美女那边,了解了电子商务通的基本功能在 Safari 上的运行情况。在得到了一些答案之后,关毅也为自己以后做显示效果上的兼容做了准备。

第一天总是过得很快,大家感觉相当充实,不过另一方面也还是感觉心里沉甸甸的,毕竟前路漫漫啊!

下班前,关毅召集大家开 Scrum 每日会议。

关毅说:"先从我开始吧,大家依次轮着来。我今天主要有几个部门相关的杂事要做,没有很多精力做 Scrum 上的 Task,不好意思,我明天会看一下关于 HTML、CSS 的一些基本内容,再巩固一下知识,等段美女把她那块的基本功能支持做完了,

我就可以开始做了,目前还没法做。"

段美女说:"我今天把我的机器升了级,网络太慢了,花了我半天的时间。然后 我用 Safari 访问了我们的电子商务通,发现连首页都打不开,只提示我不支持该浏览器。因为以前我们的产品不支持 Safari,所以系统就会直接做出判断,如果用户使用 Safari,就提示用户不支持该浏览器。后来我把系统里的判断代码删除了,就打开了。其他基本功能方面的细节我还要继续看。"

关毅问:"大家遇到了什么阻碍或者其他问题吗?我们每天的会议需要大家回答 3 个问题:今天做了什么?有什么困难?明天要做什么?大家有问题最好及时提出来,别堆积在那里。如果有阻碍迈不过去,或者长时间对其他 Task 有依赖,大家可以及时地将 Task 状态置为 Impeded,转而跟进其他的 Task,不要吊死在一棵树上。"

段美女说:"哦,技术上的我暂时没有,不过我有一个 Serum 上的问题。我今天调试和升级机器就花了半天时间,像这样的工作量,Serum 怎么对待?需要作为一个 Task 来看待吗?"

关毅说: "呵呵,这个问题不错,我现在也拿不太准。虽然升级机器这样的事情不属于代码开发的范畴,但也是关系我们这个项目成功的一个小的部分。没有这个过程,我们的程序是开发不出来的,所以我觉得这可以看成是一个 Task,你等会可以加上去,比如叫做'准备开发环境',然后为它设定一个时间,比如你今天花了3个小时左右的时间,最后把它的状态设置为'已完成'。我觉得这样才能正确反映出大家的实际工作量。最后,系统能够统计出我们总共花了多少时间。等有机会我再问问 Serum 专家 David。"

总结和思考

其实类似升级机器这样杂七杂八的事情还有很多,有些就可以算在计划预留的缓冲里面。



小胡接着说:"我这边在看后台收集数据的相关代码逻辑,还不算太难,毕竟有一些基本的文档在那里,我明天还会继续做这方面的工作。"

胖子说: "我今天在网上学习了一些 PDF 的基本知识,并且搜索了 PDF 打印的



第三方类库,发现大部分项目都是用 CCE 工具包来打印 PDF 的,而且我查了一下它们的 License,发现是开源的,并且能用于商用,所以这个肯定是没有问题的。另外,我看了一下它的文档,还算比较详细,我把它下载下来,运行了一个小例子,很顺

利,打印出了一个 PDF 文档,我想我们就用这个吧!明天我会继续 PDF 的工作。"

关毅说:"嗯,不错,大家今天的进展都还可以,尤其是胖子。不过也能太急于求成,基础的东西还是要打牢,比如学习第三方 PDF 工具的 Task,一定要深入下去,对一些常用的 API 要比较熟悉。另外, Safari 的工作由于前期调研任务比较多,所以段美女也要稍微加点劲。还有就是大家还是要及时更新一下 ScrumWorks 上面的 Task的状态,段美女,你的状态就没有改,'使用 Safari 运行电子商务通,检查基本功能'应该是在进行中了,现在这个 Task 还是计划中的状态,回去记得改一下,而且要重新估计一下剩余的时间。就这样,大家继续努力,散会吧。"

得到帮助

之前关毅联系过的王宇在周三上午给大家讲了一堂有关PDF打印的深入浅出的讲座,也顺带介绍了一些关于 Safari 浏览器特性之类的内容,当然主要还是讲 PDF 打印。

王宇从PDF文档的一些自身特性开始谈起,然后列举了一些业界常用的工具包,并且描述了它们各自的优缺点。他很同意小胖的看法,工具包 CCE 是其中比较方便、好用而且功能齐全的开源免费工具包,他们之前的项目也采用它进行开发。他以这个工具包为例做了几个展示,很简单,经过他的解释,大家很快就明白了。关于 Safari, 王宇了解得比较有限,他只是说 Safari 浏览器其实也是很符合标准的浏览器,跟 Firefox 相近,所以没有太多特别的。

小胡、段美女等人边听边做笔记,时不时地提几个问题,王宇也毫不保留地做了回答。大家都感觉学到了很多,对以后的工作肯定有不少的借鉴意义。段美女最后还请王宇到她的座位上帮她看了看在使用 Safari 浏览电子商务通时出现的一些问题,并且做了些如何调试的示范。

有了王宇的帮助,进展好像一下子顺利了许多。小胖已经很接近完成"调研和学习 PDF 工具包"的任务了,他正在使用 CCE 工具包做一个打印数据表的原型,并

且有了初步的样子,估计这周这个任务应该可以完成了。

相对于其他人, 段美女的进展感觉有些慢, 毕竟她的工作更琐碎一些。

而关毅本人还是在熟悉后台的代码,顺便也使用CCE工具包编写了几个小例子。 虽然项目本身没有真正看得见的进展,关毅还是觉得自己有些进步。他在心里笑着 给自己打了个比较高的分数。

回到家,关毅发现他那常年长草的日记体式的被他戏称为"Web 2.0 时代先锋者+Serum 学习型新手"的博客竟然有了两条留言,都是有关 Serum 的问题。虽然关毅自己也是新手,但对方显然也只是对 Serum 感兴趣的从事 IT 的新手而已,基本对 Serum 没有什么太多的了解。关毅鼓起勇气简略地回复了他们的留言。



博客问题的回复

Q:你好,我们是一家非常小的软件外包公司,只有10个人左右。你认为像我们这样的小型公司采用Scrum怎么样?

关**毅**:小公司其实是更容易采用 Agile 的流程的。10 个人的团队,最好划分为2个左右的小组。每个 Scrum 团队,我认为4~10 个人比较合适,但是5个人左右是最有效率的。当然,大公司、大团队也同样可以用Scrum,只是需要把大团队分解成一个个不超过10个人的小团队。

Q:你好,我们也正在使用 Scrum 管理软件开发团队,我们刚开始 1 个多星期,每天大家都开站立会议,大概 1 个小时左右,讨论很多技术上的问题。我觉得 Scrum 确实很好,每天都能了解大家的状况,老板也参加这个会,他也能够每天得到汇报。我感觉 Scrum 还是很新的东西,以后大家多交流吧。

关**級**:你好,很高兴你们也采用 Scrum 做你们的项目管理。但是你提到你们的站立会议要开1个小时,我感觉这是不恰当的,一般最多15分钟就好了,更多的技术细节、实现细节不要说得太多。另外我发现你使用了"汇报"这个词,我感觉这是不好的。站立会议的意义不在于向老板汇报,我们主要关注的是团队自身和项目进展本身。



单元测试

周五是这一周的最后一天,大家紧张有序地继续着自己手中未完成的任务。

下午,小胖最终完成了"学习 PDF 的基本知识"以及"调研和学习 PDF 工具包"的任务。小胡的"调研后台数据处理逻辑"任务也总算做完了。小胖开始给小胡讲解一些学到的新知识,毕竟小胡也要用那个 PDF 工具包将数据输出并转换成 PDF格式。对比 PDF 打印这个任务,"Safari 浏览器支持"的进展情况也还可以。段美女完成了"使用 Safari 运行电子商务通,检查基本功能"以及"保证基本可用性"这两个 Task。关毅则开始了"调研 Safari 上显示方面的问题"这个 Task。

不过,显然关毅的任务进行得不算顺利。他在他的测试环境中新建了几个用户,模拟他们在电子商务通中交易的场景,包括注册用户、提交订单等,想粗略地看看显示上到底问题有多少。但是很快关毅就发现自己进行不下去了:后台管理订单的页面用 Safari 浏览器怎么也进不去,网页总是显示错误,他根本没法调查后台的这个显示页面。他有点惊讶了,于是就把段美女叫来一起调试。调了一阵子,还是一样的错误。

"看来你的'保证基本可用性'这个 Task 还是不过关啊, 页面打不打得开应该算是最基本的吧。"关毅说。

段美女不好意思地笑了笑:"我等会再看看。"

然后关毅问了问段美女具体是怎么调查的, 段美女说就是用浏览器点击各个链接, 走一走基本的流程, 看有没有基本的问题。

关毅听到这里, 觉得这肯定不是正确的方法, 于是赶紧招呼大家过来说两句。

关毅说: "各位,我刚才想开始我自己的 Task,'调研显示效果问题',但是我随便测了测,便发现有的后台页面没法打开,这应该是段美女上个 Task'保证基本可用性'没有做到位的缘故。另外,小段说她就是用 Safari 浏览电子商务通的各种页面,看有没有问题,并没有用到单元测试工具。我觉得她的这种方法很不好。当然,这种纯粹手工的调研和测试方式也是需要的。但是我认为,这种方式总会有遗漏,也免不了出错。不是有 JSUnit 吗? 你为什么不做单元测试呢?这一点我们在第

1个 Sprint 里就已经开始强调了。"

段美女说:"我们这次大部分功能都是用网页上的 HTML、JavaScript 写的,如果都用单元测试困难很大啊!"

胖子也替段美女说好话:"这个单元测试,或者测试先行 TDD 等内容,应该是另外一种敏捷方法 XP 的一贯做法吧,咱们 Scrum 是不是不需要啊?段美女只要再稍微仔细一点,估计也不会出错的,嘿嘿。"

关毅说: "我不同意你的看法。单元测试怎么能不要呢? TDD 测试先行的原则确实是 XP 的原则之一, Serum 与它的关系, 我还真的说不好。但我认为做单元测试这种最基本的原则肯定是不会有错的。小胡, 你怎么看?"

小胡说:"我不是很清楚,要不我们还是问问吧。"

关毅说: "只能这样了。另外,由于这个 Task 我这边测试不通过,小段,你一会儿还是把这个 Task 的状态改回'进行中'吧!等测试完全通过,没问题了,再把它的状态改成'已完成'。就这样吧,大家继续干活。"

下班的时的每日 Serum 会议上,在大家各自讲了今天一天的工作、下周将进行的工作以及今天遇到的问题之后,关毅表扬了大家这一周的干劲,当然也指出了诸如今天发生的问题。然后,关毅表示自己将尽快弄清楚如何对网页程序进行全面的单元测试,以及 TDD 与 Serum 的关系等问题,其他人也保证自己以后会更加小心注意自己的代码质量。

Sprint 2 的第一周就这样结束了。星期五总是令人感到高兴的日子,收拾完东西,关毅拎着包,跳上了人来人往的地铁,满身是汗地回到了家。吃完饭休息了一会,他迫不及待地打开了电脑,登录 MSN,想看看王宇在不在线,他打算问问王宇对单元测试的意见,还要咨询一些技术方面的问题。在等待王宇上线的时间里,他砌了一杯浓茶,开始给 David 写邮件。



Hi, David.

前两天收到你及时的回复,感觉自己还是有很多不足啊! 我会继续努力的。不过这次我还是有几个问题向你请教。



首先,我在周一召开计划会议的第二阶段的时候,我们团队的一个成员 突然提出来需要休假一周,这很突然,而且不可能说不让他休假,但是很明显,这打乱了我们的 Scrum 计划。你怎么看待这种问题?

其次,我们的团队成员现在使用 Scrum 流程进行开发,但是好像开始忽视了诸如单元测试之类的步骤。他们显然不是不知道,而是不情愿做。他们找到理由推托,认为在 Scrum 的字典里面,单元测试是不存在的。但我认为这是不对的。另外,TDD、测试先行的这种方法能跟 Scrum 结合使用吗?我有点混乱。我记得好像 TDD 这样的实践是 XP 的思想,它跟 Scrum 有什么样的关系呢?在我的直觉上这应该没有什么问题,但还是不能确定。

关毅

刚把给 David 的邮件写完,MSN 跳出的小窗口告诉关毅,王宇上线了。关毅赶忙打开了和王宇聊天的窗口。

王宇认为,网页环境中确实不好做全面的单元测试,但是至少可以对 JavaScript 做部分的单元测试,至于 TDD,他表示自己没有用过这样的流程。

关毅有些失望,看来确实有些困难,不过像段美女那样一点都不做测试也是说 不过去的。

突然,墙上的时钟发出的整点报时声打乱了关毅的思绪。他回过神来,觉得笔头强过心记,应该好好整理一下最近与 David、王宇等人讨论的问题,以及团队小组成员使用 Serum 开发的历程和自己的想法,这些对于自己来说是一笔财富,当然也能帮助更多的人。于是关毅又打开了自己的博客,凝神写下了一篇长文。文章中主要谈到了 Serum 的实践问题、跟 XP 的关系等内容。写完博客,已经很晚了。关毅虽然觉得累,但是自己的思路和记忆经过这样一番整理之后变得很清晰,反而不觉得困了。关毅得意地笑了,他想,干脆趁这个机会再多看看书,补充点新知识。哈哈,看来今天又是一个不眠夜了……

Sprint 2 的第二周

一个周末的调整让关毅的精力很充沛。虽然心中对 Serum 还有不少的疑问,让他看上去似乎很有心事的样子,但是关毅相信:世上无难事,只要肯攀登。

用 Scrum 不能教条

周一来到公司,关毅跟大家打了个招呼,然后第一件事就是打开电脑,查看 ScrumWorks 上的最新的项目情况。



项目进度

已经标记为 " Done " 的 4 个 Task 是:

- (1) 使用 Safari 运行电子商务通,检查基本功能。
- (2) 学习 PDF 的基础知识。
- (3)调研和学习 PDF 工具包。
- (4)调研后台数据处理逻辑。

正在进行中的 Task 有:

- (1)调研 Safari 显示效果上的问题。
- (2)完成代码,保证基本可用性。

时间过去了一周,目前的状况是:有4个 Task 已经完成了,段美女的"保证基本可用性"的 Task 应该也差不多了,其他的还没有开始。在关毅看来,这样的情况总体还行,算是马马虎虎吧。

检查了一下电子邮件,没有收到 David 的回信,这让关毅有点失落。不过他振奋了一下精神,马上开始了今天一天的开发工作。他发邮件要求大家使用 JSUnit,

発**がScrum之**派 一教徒开发故事

尽量做好单元测试工作。做完了这一切,他开始继续调研显示问题的 Task。

没想到中午吃完饭, 关毅就收到了 David 的回信。



关毅:

你好!

上个周末就收到了你的电子邮件,但是我有些忙,所以今天才回复你。

你的问题很好,有些问题我在之前的培训材料里没有讲得很透,所以我顺便利用这个机会整理了一下我的培训材料,将你提出的一些问题写到了教案中,你不反对吧:-)

你的问题主要是 Scrum 与单元测试、TDD、XP 这些概念之间的关系。 其实有的我在培训的时候已经讲过了。你首先要理清 Scrum 与其他敏捷方法 的着重点的不同之处。要知道,Scrum 看重的是敏捷项目管理上的实践,着 重点在于管理,比如如何计划、如何迭代、怎样评估进度、在什么时候遵循 怎样的规则开什么样的会等,它仅仅是一个框架,没有过多地关注具体的工 程实践,也就是说,它与 XP 这样的具体的软件开发实践关注的重点不同。 Scrum 没有教我们怎样进行结对编程、怎样做 TDD 等,而 XP 在做这些工作。 你可以简单地认为,Scrum 的概念更宏观一些,而 XP 则更关注微观。

你明白了上面我说的这个区别,那我就来一条一条地分析你的问题。

- (1)单元测试:单元测试在现代软件开发的流程中基本上属于不可缺少的一环,敏捷性质的软件开发更是如此。在 Scrum 管理的软件开发项目中,单元测试的使用与否与 Scrum 无关。但是从软件开发质量的角度和 Scrum 管理软件开发的成效来看,我建议你们要多多使用单元测试,不要轻视它。
- (2) TDD: 很高兴你提到了测试先行的做法,主要是 XP 在提倡它。正如我刚说的,你在一个 Scrum 管理的软件开发项目中,使不使用 XP 的实践方法,包括 TDD,完全是由你们的软件开发项目所决定的。
 - (3) XP: 我想我在说到 TDD 的时候就已经回答了你的问题了: -)

我想重点说的是,你们虽然是在采用 Scrum,但更重要的是你们在接受敏捷开发的思想来指导项目开发。敏捷最忌讳的就是教条,凡事都按照条条框框来进行。你们在用 Scrum,但绝对不要认为 Scrum 概念之外的实践就不用理睬。一切好的实践,如果符合团队的需要、对团队有帮助,就可以采用。

值得一提的是,我在我的培训教程中加入了"XP@Scrum"这样一个小节,简要阐述了如何将 XP 实践方法运用在 Scrum 软件开发中。

至于你说的关于请假的问题,这也正是你们团队没有一起去参加计划会议的第一阶段所造成的。如果当时那位请假的成员在安排 Sprint Backlog 的时候在场的话,我想他会提出来的,然后大家就可以重新定制任务了。当然,他当时也许还没有休假的计划。如果是这样,那么 Sprint 的计划则有可能完不成。你可以跟 Product Owner 事后商量,看能不能移走一些任务,或者你们在这个 Sprint 结束之后,再将未完成的工作挪到下个 Sprint 或者 Product Backlog 中。但是你们一定要保证不要破坏每个功能组件,要保证软件的可用性。还记得结束一个 Sprint 的标志吗?就是保证要拿出一个真正可以给用户使用的软件来。

下次有机会我再跟你谈谈。今天时间有点晚了,再见!

David

总结和思考



机械地实施敏捷是一开始比较容易犯的错误。敏捷并不是一种过程。

关毅觉得 David 的这封邮件写得很好,为他解答了一个很大的困惑,于是就转发给了大家。大家也很踊跃地发表了自己的看法,其中,小胡的看法很有代表性。



我很认同 David 的观点。确实,我感觉 Scrum 只是一个管理框架,实施的细节可以灵活考虑。我想对于单元测试,我们大家都要认真对待,以保证我们的产品质量。不过像 TDD,

我个人认为还是不要实行了,毕竟大家在习惯上很难改变过来。不过 XP 的 其他实践方法,比如结对编程,我们可以考虑考虑。

维松Scrum之旅

大家基本上都一致同意小胡的意见。

讨论完这个事情,大家开始继续完成未尽的任务,一片热火朝天的景象。

很快,一天就要过去了。临下班时,大家开始了 Sprint 2 的第二周第 1 天的每日 Scrum 会议。

胖子首先发言: "我先说。我今天开始了'将表单中的图片输出至 PDF 中'的任务,这些图片主要是表单中的一些物品图片或者交易商家的一些照片等,明天我会继续做这方面的代码。"

段美女继续说道: "Safari 上的基本功能我已经调通了,也添加了几个 JSUnit 的测试,目前来看应该没有什么问题了。我今天开始对整个功能细节进行大的调研,目前没有什么问题,明天会接着做。"

关毅说:"是吗,那太好了!我今天还是继续粗略地看了一些页面在 Safari 上的显示情况,发现前台的页面相对好一些,后台的页面问题比较多,具体的我明天会继续看。估计明天,最迟后天,就能做完这个 Task。另外,我的那个写解决方案的 Task 也同步开始了。我想发现一处问题就记录一处问题,最后再总结归类。"

小胡说:"我周末在家也看了些资料。目前我正在进行'输出表单数据至 PDF'的工作,感觉有点难,而且我这块的数据需要填充到表结构中,但是于胖子还没有开始做那个'输出表格框线到 PDF'的任务,所以我这块做的还只能是实验性质的。我想这两天等我的实验基本差不多了,然后就把这个 Task 设为'Impeded'。再等胖子做完他的那个部分,就可以接着在他的基础上完成最后的步骤了。"

关毅说:"呵呵,你说的很对。暂时先这样吧!胖子那边做完了,可以和小胡一起完成那个部分,看看行不行。那好,大家还有什么补充的?没有的话,今天就这样吧。"

经理安排的临时任务

每个工作周最出成果的是中间的两三天,这周也不例外。小胡因为自己下周就要休假了,做事很用心。

周四的每日 Scrum 会议,关毅的经理徐天也来了,大家有些意外。

徐天从头到尾听完了大家的发言,基本了解了大家的进展情况,然后说:"大家的进展还可以,比上个 Sprint 有进步。关于对 Scrum 的理解以及对项目的熟悉程度,我也要跟着大家一起进步。"

听徐天说完, 关毅他们笑了。

不过,徐天紧接着又提到了一件事情:"下周有一个大老板要过来,我想让他了解一下我们的产品,你们谁有空能帮忙安装一下我们最新的产品,并且准备一下PPT,给老大们讲讲?另外,你们每天开的这个会,别忘了继续发会议记录给我。"

听完徐天的这番话,关毅想了想,还是点了点头。

下班回家的路上,关毅一直都在盘算着自己的时间和下面的计划安排。

徐天临时安排的任务打乱了他的 Serum 任务安排,这让关毅感到有点吃力。至于会议记录,关毅觉得那倒没什么。不过关毅还是想写信问问 David,这样的情况下怎么做比较合适。自己作为一个 Serum Master,自己的经理又应该是什么角色? 经理这样的安排是不是有问题呢?

关毅决定把这些先记录下来,等到这周结束,看看还会累积哪些问题,然后一 并给 David 写信。

关毅回到家,收了一下邮件,发现有两封系统自动发出的电子邮件,通知关毅这些信件来自他的博客。关毅有点吃惊:看来大家对 Serum 的热情还是有的,自己不是一个人在战斗。

关毅有点儿兴奋, 赶紧打开了自己那个"Web 2.0 时代先锋者 + Scrum 学习型新手"的博客。每次看见自己给博客起的名字, 他都觉得有些好笑, 不过要的就是这个效果。

打开博客一看,和上次一样,有两个人在他之前发的文章后面提了两个小问题,都是一些简单的常见问题。关毅凭着自己的一点经验,认真地给他们回复。



博客问题的回复

Q:为什么每日 Scrum 会议大家一定要站着开?

关毅:站着开,大家容易放得开,而且时间上不会太长,效率会高一点。

Q:你好,我们是一家小公司,想用 Scrum 来管理我们的软件开发流程,但是市面上就没有看见一本讲 Scrum 的书,就算有,也是纯翻译国外的,而且翻译得还很不好。所以我想问你,你们有这方面的资料吗?另外,我有一个关于 Scrum 的小问题。我们的团队认为每日 Scrum 会议太繁琐了,有时候一天也干不了什么事情,没什么好说的,他们认为可以减少开这个会的频度,一周5天,周一、三、五开就可以了,你认为呢?

关級:我们的学习资料大多来自公司内部的培训以及国外的一些相关资料,不过你如果要中文的资料的话,也许我们可以写一本 Scrum 的书,呵呵。至于你说的每日 Scrum 会议开会时间频度的问题,我是这么认为的:Scrum 是一个 Agile 流程,Agile 注重的是项目中的动态变化,拥抱这些变化。每日 Scrum 会议能够让我们及时发现项目中的变化。而你们扩展到两天开一次这样的会议,虽然在项目前期看来可能很好,但是项目进行得越深入,问题就会越多,两天一次积累的问题也会比较多。所以我认为每天开一次会议是比较好的,大家容易把握方向,也利于掌握进度,应对实时的、经常的变化。

周五到了。

小胡明天就要休假回家了,他今天主要的工作就是将手头上的代码做个清理, 争取做到不影响下个星期的开发。等到最后一周小胡回来之后,他还可以继续完成 他的任务。

小胡的"输出表单数据至 PDF"的任务被他标注成"被阻塞"(Impeded)状态,这个任务主要还是依赖于胖子的"输出表单框线至 PDF"的任务,而胖子的这个任务只能到下一周才能做完了。

关毅负责的"Safari 显示 UI"方面的调研已经做好了,下周就可以正式开发代码,完成显示上的代码兼容了。

而段美女的进度则相对慢一点,功能上的调研报告还有 5 个小时的工作量等着她。如果把这个做完了,那么"调研 Safari"的 Story 就算完成了。

下午下班的时候, 关毅正准备收拾东西回家, 却发现大伙还都没有要走的意思。

关毅走到胖子身边问:"怎么,加班吗?"胖子点了点头。

关毅说:"我们还是尽量不要加班的好。任务做不完是 Serum 计划安排得有问题,现阶段我们才刚刚开始 Serum 使用,计划做得肯定不是很好,等到以后,大家不需要加班是最好的。"

胖子笑了:"那是当然。你先回去吧,我再待一会,很快就走了。"

关毅回到家,晚上9点,习惯性地开始给 David 写起了电子邮件。



David:

你好。

上次你谈到 XP、单元测试等实践方法与 Scrum 的关系,我收获不少啊,呵呵。团队成员们都觉得你说的不错,很有启发。我给你转发一个我们团队成员小胡的看法,他很认同你的观点,他感觉 Scrum 只是一个管理框架,实施的细节可以灵活考虑。对于单元测试,我们大家都要认真对待,以保证我们的质量。但是诸如 TDD 这样的,他个人认为还是不要实行了,毕竟大家在习惯上很难改变过来。不过 XP 的其他实践方法,比如结对编程,我们可以考虑考虑。

他的意见代表了我们大家的共同意见,最后我与我的团队讨论了一下,还是要求单元测试越完备越好,做得越勤越好。但是对 TDD,我和我的团队成员好像都没有概念和信心,所以大家还没有要用的意思。不过你说的 "XP@Scrum"的概念,我感觉很形象,很好。当然也正如小胡说的,还有一些其他 XP 的实践方法,包括结对编程之类的。对于这些,你还有什么要特别补充的吗?这是我今天的第 1 个小问题。

另外,你说会把我的问题整理到你的讲义中去,真是荣幸之至啊,呵呵。



不过我今天的第 2 个问题与技术无关,主要是在管理上面,希望你能指点迷津吧。

我的经理今天参与了我们的每日 Scrum 会议,简要地提了一些建议和意见,但是也给我安排了一些与项目无关的工作,稍稍打乱了我的计划。我知道这是不可能避免的,但我还是很好奇,一个 Scrum Master 的经理或者任意一个上司在 Scrum 中扮演的是怎样的一种角色呢?我应该怎样配合我的经理呢?另外,假如我自己是某个 Scrum 团队的 Scrum Master,应该怎么做比较好呢?

致谢。

关毅

Sprint 2 的第三周

进入第三周, 也就说明时间过去一半了。

关毅观察了一下 ScrumWorks 上的 Sprint Story, Safari 的调研工作已经基本完成了,代码工作还没有开始,而 PDF 打印的后台代码完成了 70%,前台代码还没有做。

关毅看了一下目前的 Burndown Chart (如图 5-4 所示), 燃烧曲线已经下降了一半左右, 但是由于小胡已经请假了, 后面可能就不会那么快了。总之, 感觉还是不太妙。另外, 小胡请假回家后, 由他负责的"输出表单数据至 PDF"的 Task 只能等他回来做了, 还有一些测试之类的工作也是他的任务。

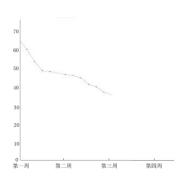


图 5-4 目前的 Burndown Chart

总结和思考

燃烧曲线是衡量团队进度的重要工具。但是不要过分依赖它作为监督和考核的依据,否则就会变味。因为团队会把重点放在生成漂亮的曲线上,而不是项目本身。



遇到困难

今天是星期一。关毅知道,每周的周一是很重要的,每日之计在于晨,每个星 期也是如此。



关毅早上把大家召集在一起,说了一些鼓舞士气的话,要求大家抓点紧,毕竟这已经不是第1个Sprint了,大家再也不能说类似"之前一点经验都没有"的话了。

这一天关毅也很忙碌,他恨不得把时间掰成两半用。一方面他需要跟进项目,根据之前自己写的方案,一点一点地推进 Safari 上的显示问题的解决;一方面他还受经理所托,要编写产品的 PPT,另外还要录制一段产品演示的视频。他知道这些是 Scrum 流程中的异常情况,难道要自己加班加点做这些工作吗?但是不加班又能怎么办呢?关毅真是觉得有点累。

段美女今天终于完成了最后剩余的一点点调研工作,正式进入了 Safari 功能上支持代码的编写工作。

胖子则还在啃他那块硬骨头——输出表单框线至 PDF,好像进展不大。关毅看他剩余时间还是那么多——10 个小时,心想胖子应该是遇到困难了。

果不其然,这天的每日 Serum 会议上,小于说他那块确实很难,有些算法之类的还需要多想想,再考虑考虑,估计一时半会不会有太大的进展。

总结和思考



Scrum Master 要有敏锐的"嗅觉",及时发现团队成员遇到的问题。

关毅虽然急,但也只能看着,因为他自己也不懂,没有任何办法去帮助胖子。 关毅知道项目中途出现拦路虎是很正常的,但是关毅不知道,如果出现了团队成员 对团队中其他人做的代码基本不了解的情况到底正不正常,毕竟有时候真的只能分 工去做,不可能做到全部覆盖。他决定把这些写到自己的博客里,先记下来,顺便 也看看网上的朋友们怎么说。



关毅 Scrum 日记: Sprint 2, 时间过去了一半

星期一,阴天有小雨。

Sprint 2 过去了一半。前一个阶段虽然进展不算快,但还是顺利的。不过 今天胖子遇到了阻碍,他估计这个问题花 1~2 天都很难完成,我们其他人完 全没有参与他的那块内容,爱莫能助。我知道做项目总是会有一些问题的,

但是我想,要是团队成员都对其他人的工作都有比较清楚的了解的话应该会好一些吧,这样其他人也能提些建议和意见。

不管怎样,加油吧。

Code Review 的重要性

周二上班,关毅继续完善着给大老板们看的 PPT。他也不是那种应付事的人,做事很细致,找这个数据,画那个图表,力争做好这个 PPT。另外,ScrumWorks 上的列表就像一个盯着自己看的稻草人,时时刻刻都在那里,令他压力倍增。

不过,他还是忙里偷闲,看了一下自己的博客,一条留言赫然在目。

博客上的留言

每个项目都是一样的。不管采不采用 Scrum 这样的流程,大家都只能分工合作,齐头并进,不可能都关注同一个任务。 而且团队成员的技术水平各有差异,技术结构也不尽相同,所以分工是必然的结果。

但是,我认为你可能忽略了一些在敏捷软件开发、甚至是一般软件开发中的基本要素。你们有没有代码复审,也就是 Code Review 的机制呢?或者你们有没有用结对编程(Pair Programming)这样的敏捷方法呢?我想这样两种机制至少能够保证团队成员在某个软件功能方面不至于只有唯一的一个人了解。结对编程能保证两个人对代码的熟悉程度基本上是一样的,而代码复审机制则能保证团队全体成员都对代码有一定程度的熟悉。你可以开一些代码复审的小会议,要求所有人参与,不管团队成员中的人跟这块功能有没有关系,你最好都把他们拖进来,这样有问题也能及时发现。这样,即使团队组织结构发生了变化,也不会对项目开发产生太大的影响。

◎希望以后多交流吧。

看了这篇留言,关毅很受启发,Code Review 这样的工作看来意义真的很大。他想,如果胖子这周之前不能解决问题,那就看能不能组织开个不长的会议,让大家



帮他看看代码,集思广益,也许能找到一个好办法。就这么定了!

尽管在这件事上自己有些疏忽,不过正是因为 Serum, 才让大家能在第一时间发现问题, 而不至于延误战机。

给经理的项目状态报告

周五一大早,关毅就来到了公司。这已经是这周的最后一天了,也意味着整个 团队的时间所剩不多了。

前两天,也就是周三和周四,是大家干劲最足的时候,项目的进展也还比较顺利,但就是不知道胖子今天能不能解决他那个棘手的问题。

关毅舒了一口气,他揉了揉红红的双眼,努力想集中自己的注意力。关毅昨天晚上熬得有些晚,不过终于完成了给大老板们汇报用的 PPT,也录完了产品演示的视频。他今天想全身心地投入到自己的 Serum 任务中去,毕竟自己也是开发团队中很重要的一环,他负责的"Safari 浏览器显示方面的支持"的 Task 还需要不少的时间去完成。

到了中午,关毅收到了 David 的电子邮件,主要是对他上次提出的一些问题的集中答复。



关毅:

你好!

我发现这种一问一答的方式真的很好,很多问题以前没有讲得很透彻,但是在和你写信的过程中,却能越辩越明,我真的很喜欢这种感觉。

关于你问的"XP@Scrum"中的其他一些问题,比如说结对编程等,我想,这些敏捷的实践方法你们团队都可以试试。Scrum 并没有规定什么敏捷方法不能用,什么方法能用。所以像结对编程这样的好的实践,你们确实可以试试。不过你们同时也可以关注一下其他的敏捷思想,比如 RUP 这样的敏捷过程,其中的一些敏捷理念,你们也都可以运用在自己的开发过程中,融会贯通。

至于你说的你们经理的问题,以及对你们经理的做法的一些意见,我想这个在我们之前的培训讲座里面已经说了一些内容了。你做 Scrum 的时候一定要记住"鸡"和"猪"的故事,一个 Scrum Master 的经理也许是"鸡"的角色,如果他不参与项目的开发的话。所以,我认为经理如果擅自安排那些额外的事情给团队成员做是不恰当的。但是换个角度思考,我们又不能说这些事情的安排不重要,做 PPT 和演示产品这样的事情,总需要有人来完成,而且这也肯定是对产品有益的事情。

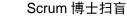
所以我想你们在计划整个 Scrum 的 Backlog 的时候,首先一定要对未曾 预料到的中途出现的事情有所预防。简单地说,就是要留出一定的时间来处 理这些异常事件。你们之前的计划会议也许预留了一些缓冲时间,但事实上,这样的时间是不能随随便便估计的,必须要按照每次 Scrum 的经验来量化地估计。当然,你们也只是刚开始,所以以后你们对这样的事情肯定会更有经验。另外,如果突发事件需要花费团队成员大量的精力,那最好还是拒绝,以保证 Sprint 的正常进行。如果这件事足够重要,有很高的优先级,那么可以放到下一个 Sprint 的 Backlog 里面。

当然,单从你们的经理参与每日 Scrum 会议来说,作为经理,他如果比较频繁地参与每日 Scrum 会议,说明他对产品很关心,急切地想知道目前整个项目的进展情况。换言之,除了参加这个会议,他可能还没有找到其他更有效的方式来获取这些信息,他觉得自己有点孤独了,呵呵。

所以,作为 Scrum Master,你也需要协调好开发团队与经理之间的关系。我给你提两点建议。我知道你们在使用 ScrumWorks,第一,你可以为你的经理在 ScrumWorks 上建立一个类似观察者身份的账号,那么他就可以随时随地掌握整个产品以及当前 Sprint 的进展情况,也就可以实时观察 Burndown Chart 的变化,这样他就不需要太频繁地参与会议,提出一些干扰 Scrum 进行的建议和批评了。第二,我想很重要的是,或许你的经理觉得仅仅拿到一个账号来查看那些枯燥无味的数据是没有太大意义的,那么我认为你可以每隔几天给经理发一份当前工作的简报,在简报上面你可以采用一些图表,比如饼图、柱状图,来多方位、多角度、更加全面地体现当前工作的进展,这种展现方式,可以认为是对 ScrumWorks 的 Burndown Chart 的一个极大的补充。这样的话,我想你的经理应该会满意很多。

做到了这两点,我想你的经理在掌握了这么多的团队工作信息,了解了你们的任务有多重之后,就自然不会给你们的团队成员安排太多的所谓"杂事"了。这样的话,你们将会有更多的时间完成 Scrum 任务,也能更高质量地完成自己的代码工作。这也是相得益彰,两全其美的方法。

David





• "鸡"和"猪"的故事

Scrum 把软件开发项目中的各种角色形象地分为两类,一类是"鸡",一类是"猪"。 故事的来历是这样的。

一只鸡和一头猪是朋友。一天,鸡对猪说:"咱们合伙开个餐馆吧!"猪觉得挺有意思,说道:"这个主意不错,那咱们的餐馆该叫什么名字呢?"鸡说:"叫火腿和鸡蛋吧。"猪马上不干了,说:"那谢谢了,我不参与了。如果要开这个餐馆,我得把自己全部贡献出来,需要全身心地投入,而你只需要投入一部分就行了(如图 5-5 所示)。

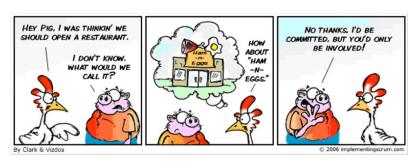


图 5-5 "鸡"和"猪"的故事

在 Scrum 中,不实际参与项目开发的一些管理者是"鸡", 而全身心投入的项目团队成员是"猪"。

关毅一口气看完信,觉得非常痛快。David 的电子邮件写得很畅快,而且很实际, 关毅看了感觉相当兴奋,他决定立刻就开始实践。

关毅首先以管理员的身份登录到 SerumWorks 中, 为徐天建了一个账号, 然后发

orint 2——计划与变化

了一封电子邮件给徐天,告诉他现在可以随时登录 ScrumWorks 并了解项目的进展状况了。

然后,关毅新建了一份 Excel 文档,开始做 David 的电子邮件中所说的工作进展统计简报。他首先搜集了当前每个 Task 的估计时间、已完成时间、每个 Task 从开始到完成所用的实际天数等信息,然后利用 Excel 自带的图表功能绘制了几个图表,比如每个团队成员工作量的对比、每个团队成员工作效率的对比、每周的进展速度等。总而言之,这个表格让一直身处项目之中的关毅看了都觉得对当前的工作进展有了新的认识。

关毅开心地笑了,他决定这样的简报以后可以每半个星期发一次,而且不仅要 发给经理徐天,还要发给每个团队成员。大家建立同样的认识,对项目肯定是很有 好处的。

总结和思考

如果觉得这种报告比较麻烦,只使用项目管理工具提供的报告功能就 足够了。



最后一个 Story 可能完不成了

周五,很快到了下午快下班的时候,每日 Serum 会议开始了。由于是这一周的最后一天,关毅拿着自己的笔记本电脑过来了。他打算稍微多说两句,总结一下。

于胖子说:"今天我先说吧,呵呵。我终于把那个在 PDF 中划线的问题解决了,我试了试,效果很好,我已经将代码提交了,等下周小胡来了,把他那一块最后剩下的内容补上,做完测试的工作,基本就算完成了。我下周来了之后也会开始做 PDF 打印的 Web 前台。"

关毅说: "太好了,辛苦了,呵呵。找个时间我们大家一起来做一个 Code Review。 我这块实在是有很多杂事要忙,这一周的进度都不怎么样,针对 Safari 的显示效果 的调试算是完成了5成,我下周只有接着做了。段美女,你今天怎么样?"

段美女说:"我们的产品在 Safari 上的功能部分我已经调完了,然后简单的单元



——敏捷开发故事

测试也跑得差不多了,同时文档也在写。估计下周我的任务应该能完成,没有什么问题。不过我看 PDF 这块内容,尤其是前端的代码,好像还没有开始写啊!我有点担心。"

关毅打开了 SerumWorks,招呼大家一起来看现在的进度(如图 5-6 所示)。确实如段美女所说,PDF 前端的东西还没开始做。虽然于胖子下周可以开始这个任务,但是他还要配合小胡完成后台的代码,所以估计PDF 前端输出的任务很难完成了。

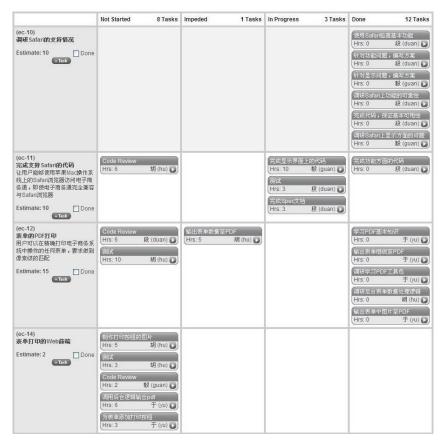


图 5-6 第三周进度图

关毅舒了一口气说:"总之,情况不容乐观。从图上看,最后一个 Story 可能完不成了。但是为了能顺利演示,我们至少要保证前 3 个 Story 100%的完成。我不打算现在就告诉温哥华,因为他们也能查看我们的 ScrumWorks 状态,其实可能也已经

了解了。我是想下周努一把力,说不定我们还有希望。但是我要再次强调,还是先 把前 3 个 Story 完成。"

会后,徐天找关毅谈话。大概是通过 ScrumWorks 了解到了进度情况,也或许是 关毅的状态报告起了作用,徐天看到了当中的风险,也就是最后一个 Story 有可能完 不成,所以徐天要了解关毅下一步的计划。

总结和思考

在这里,保证前3个 Story 的完成是个正确的决定。敏捷开发可以使项目具有高度的可视性,并能够及时发现问题。



关毅和徐天单独在小会议室里商讨对策。关毅想在老板面前表现得信心十足,于是他告诉徐天,如果努把力的话,任务应该是能完成的。徐天则建议关毅还是提前沟通一下为妙,至少要打个招呼,解释一下原因。徐天并没有对关毅有可能完成不了最后一个 Story 而生气,因为他知道,在整个 Sprint 2,大家其实都尽力了。但是他不希望在沟通上出什么差错,何况已经是 Agile 了。关毅看徐天并没有责怪自己和团队的意思,就壮着胆子发了封电子邮件给温哥华的 Product Owner Greg 和 Peter。

下班回家的路上,一路小雨,淅淅沥沥的,整个团队成员的心里既有对这几天 工作成果的满足感,又有对任务总量完成不了的担心,作为 Scrum Master 的关毅更 是如此。回到家,他连上网写博客的心情都没有了……

这个周末,大家过得都不是很痛快。



Sprint 2 的第四周

第四周, Sprint 2 冲刺的一周。

周一,大家都来得很早,小胡也按时归队了。大家的心情有些急切,毕竟任务 还有很多。

关毅发现徐天并不给自己加压,感觉轻松了不少,反而更能够加足马力应付自己的 Serum 任务。

小胡今天开始做他最后剩下的"输出表单数据至 PDF"的任务,他好像在家也没有闲着,补了些功课,看上去今天完成这个任务没有什么问题。

于胖子由于想尽快完成"表单的 PDF 打印"这个 Story, 把这个 Story 的单元测试任务也接了过来,毕竟小胡还有一些 Code Review 的工作要做。

段美女的事情也很多,她主要是在做 Safari 上的测试工作,她不光要测试自己的之前做的功能方法的那部分内容,同时也要顾及关毅做的显示效果上的内容。

冲刺=加班

毕竟是接近扫尾工作了,最后剩下的任务都比较细碎,然而大家却干得非常有 效率。

不过,下午照常开完了这天的每日 Serum 会议之后,关毅自己加了点班,他想稍微再加把劲,把代码质量再提高提高,对以前的代码做一些简单的重构。他观察了一下,段美女、小胡和胖子也没有走。几个人很有默契,都干到晚上9点才回去。

回到家,关毅觉得非常疲惫,趴在床上眯了一会,感觉随时都可能睡过去。不过他还是想起上次自己发现于胖子加班之后,还批评了他,可是今天自己和段美女也都加了班。没办法,形式所逼啊。

他又想到了向 David 求救。



David,又打扰了。

关于加班的问题,我想再向你咨询一下。

我们的 Sprint 2 到了尾声,这个周末就要开始评审,但我发现进展还是不尽如人意,有一个 Story 估计完不成了,至少到现在还没有开始。虽然这个 Story 不是一个特别关键的功能,但我们还是觉得有些失望。

这几天,大家都或多或少地出现了加班的情形,没办法,害怕完不成,所以要赶赶进度。我知道敏捷是不提倡加班的,敏捷倡导可持续的开发进度。我们在 Sprint 一开始做的计划是比较认真的,尽可能地想在正常情况下完成所有的任务,可最后还是比较失败,眼看 Sprint 就要结束了,还有很多任务没有完成。我们该怎么办?

写完信,关毅舒展了一下自己疲惫的身体,收拾收拾东西,就倒在床头沉沉睡去了。他实在是太累了。

周三,大家开始进入了一种加速开发的状态,都在满负荷运转,剩余的任务也 在一点一点被蚕食。这可能就是一场战斗吧!大家都是勇士,关毅这样想着。

经理徐天也偶尔过来看看大家的情况,为大家加油,不过这次明显没有打扰到 大家。

这一天,大家基本上完成了前面 3 个 Story 的遗留任务,在每日 Serum 会议上,大家都显得比较活跃。

小胡说:"我的'支持 Safari 的 Code Review'这个任务做完了,发现了一些代码上的问题。比如,很多地方明显有拷贝、粘贴的痕迹,代码没有做到复用,还有几处声明的变量到最后没有用到等,当然也发现了两个小的算法方面的问题,我都写在 Wiki 上面了,待会儿发给大家。明天的话,我看是不是动手做做最后一个 Story,虽然肯定完不成了,但是做一点是一点嘛。你们认为呢?"

关毅说:"这样也好。不过因为后天,也就是周五,要开评审会议,你明天下午抽出点时间来熟悉一下流程,准备演示吧。另外,我今天完成了 Safari 显示效果兼容性代码的最后一处扫尾工作,还不错,没有在最后关头拖累大家。不过,小胡,



你做的 Code Review 实际上还是依赖于我那个 Task 的,所以你有可能需要再看看我今天提交的代码。明天的话,我会根据小胡目前 Review 的结果和明天可能查出来的新的问题做修改,算是清理战场吧。"

段美女说:"我今天主要修改了一下 Safari 支持的规格文档,然后简单试了试 PDF 打印的功能。现在 PDF 打印的 Code Review 刚做了一点点,没有什么问题,明天上午应该会做完,估计问题不多。另外,我明天还要针对小胡 Code Review 提出的意见进行修改,估计明天的时间会很紧张。"

关毅说: "不要急,一项一项来,明天我也可以帮你点忙。胖子,你今天呢?"

于胖子说:"今天我把 PDF 打印的单元测试的 Task 完成了,主要是补充了一些单元测试,覆盖了几个不常见的测试用例,简单调整了代码,没有发现什么大问题,几个小问题随手就改掉了,我明天会根据段美女的 Code Review 的结果再做一些代码修改。"

关毅说:"现在进入最后的阶段了,大家一定要安排好时间,处理好 Task 之间的依赖关系,什么先做,什么后做,一定要有比较清晰的考虑。好吧,今天就到这里,明天我们开始清理战场!"

加班是个很严重的问题

关毅所形容的"战场",实际上并没有想象中那么的狼藉一片。大家的代码质量与上一个 Sprint 相比有了一些改观,而且单元测试的广泛使用也保证了大部分代码的可靠性。所以,今天主要要完成的 Code Review 的工作以及随之而来的代码修改,在前期大家的共同努力下,并没有花费很多时间。

周四上午段美女就完成了 PDF 打印的 Code Review,发现了几处代码的问题,主要是一些代码格式规范和代码复用的问题,有的地方她随手就改了,所以于胖子、小胡他们的修改工作很是轻松。

而小胡发现的"Safari 支持"这个功能上的问题稍多一点,主要集中在关毅做的这块。关毅看着 Wiki 上小胡记录的问题,觉得有点不好意思。自己作为 Scrum Master,代码质量还需要再提高。不过小胡他们却还是觉得关毅在项目之外的乱七八糟的事

情这么多,做的代码上出点小问题是可以理解的,何况也都不是什么大问题。

临近中午,关毅抽了点空,收了收自己的邮件,看到了 David 对他上次提出的加班问题回信。

关毅:



昨天看到了你的信。你这次提了个很有趣的问题,但是又 是一个很普遍的问题。

你知道在一个软件开发项目中要做到团队成员 100% 不加班基本上是相当困难的,可以说,无论是大型企业还是小型公司,都很难做到这一点。不过我还是基本同意你的看法,不提倡加班。

Agile 原则上是不支持加班的。因为敏捷的软件开发不是一直以冲刺的力量跑,也不是动不动就像抽了筋似地一阵猛跑,而且敏捷要求团队成员的密切合作,所以大家的速率必须基本保持一致,才能使整个项目有序地顺利向前推进。加班对于团队成员的工作效率来说也是一把双刃剑,今天晚上加了班,提高了工作的产出,然而你第二天白天就可能会由于昨天工作太晚没有休息好导而致效率低下,加总起来,可能还不如不加班的效率高。长期加班对整个团队的士气和战斗力都是一种极大的摧残。

加班产生的源头,其一是计划不充分、失准、中途有或大或小的变化等。而 Scrum 的流程、每日 Scrum 会议、Scrum 计划会议、短的迭代周期,这些无一不是为了克服传统软件开发中的弱点的。传统软件开发中的弱点主要有流程无序、管理困难、难以变更、难以拥抱变化等。造成加班的另一个原因可能是不好的习惯或观念,认为只有加班才能体现对工作的热情,这是非常幼稚的。殊不知,对一个人来讲,工作和生活如果不能平衡,时间长了一定会出问题。敏捷开发是以人为本的,对于团队,如果人出了问题,那项目失败的可能性就会大大增加。对每个人来讲,如果出了问题,他的职业生涯又将如何在一个倡导敏捷的时代有所发展呢?

我注意到你的信中只是提到了一些零散的加班现象,而没有一些长时间 大规模的加班,我想这是好的。零星的加班无法避免,长时间的加班是完全

発松Scrum之旅

——敏捷开发故事

不能被 Scrum 所接受的。当然,到了项目的尾声,在可以预计的情况下,如果加班能够完成剩余的工作量,不加班则有困难的时候,你们也可以酌情考虑。但是从原则上讲,不应该提倡加班。我想,以后随着你们的计划越做越好,随时都能拥抱变化,随时都能与 Product Owner、项目投资人进行密切联系的话,这种问题就不再是问题了。

David

关毅收到信,倒吸了一口凉气,不禁回忆起他在 X 公司日夜奋战的经历。那场噩梦仍历历在目:他孤零零地躺在病床上,什么都没有了,工作,爱情,生活,一切!关毅心想,绝对不能再回到过去的老路上了。

想到这里,关毅立刻决定将这封信和大家分享一下,然后顺便要求大家也照着 这封信的精神来做,不要随意加班。



Hi, Team.

前两天我问了我的培训老师 David 一些关于加班的问题, 下面就是他的回信,大家可以仔细看看,体会一下。

我想跟大家说的是:这次就算了,我们还是要尽力完成所有的任务。从下一个 Sprint 开始,大家能不加班就尽量不要加班。我们需要做的还是要在一开始就把计划做好,而不是透支我们有限的青春。我们是一个和谐的、讲究可持续发展的、积极向上的团队。

关毅

总结和思考

是否应该为了赶进度而加班?



不是所有的 Story 都适合做 Sprint 演示

周四下午,关毅和段美女主要集中攻克 Review 之后的问题,而小胡和于胖子则 在临近中午的时候就已经做完了修改的工作,开始考虑怎样演示他们的 PDF 打印这

块内容。小胡有点犯愁,由于前台的代码没有写,都不知道该怎么演示。他和于胖子试着编了编前台的代码,想看看能不能快速做出一个原型,调用后台的代码,输出 PDF。不过他们很快就发现时间不够用了,本来想加班,却不曾想关毅中午竟然给大家说了一通 Agile、Scrum 对加班的态度,要求大家不到万不得已最好不要加班。这下小胡他们两个没了主意,跑过去问关毅的意见。

小胡说:"能不能不要这个演示啊,费时又费力,有什么用啊,而且一点意思都没有,有些东西也没办法演示。最后做好了大家试试不就行了吗,何必呢?"

关毅说: "你们忘了之前让你们学习的有关 Serum 的内容了吗?这个演示一方面能让其他人对我们的工作有一个认识,而且能及时收到来自项目负责人、Product Owner 等人的反馈,也能让我们有成就感。"

关毅想了想,接着说:"我看那个前台的功能你们就是加班也不一定能弄好,这样仓促做出来的代码肯定不行,到头来还是要返工,这不就相当于糊弄大家嘛!这样吧,你们干脆给大家展示单元测试算了,虽然不好看,但毕竟这是目前为止唯一的办法了。不过,直接在机器上跑单元测试,都是代码,不容易看。这样,你们可以用 Flash 的方式把你们后台的单元测试过程录下来,比如输出某个表单的数据之类的过程,然后打开输出的 PDF 展现一下。录完后附在邮件里发给我,明天我再做个 PPT 给他们演示。"

"好主意,我这就去做Flash。"小胡说。

下午 3 点, 关毅和段美女完成了 Safari 方面的代码修改, 还算比较顺利, 主要是其中一个算法花的时间长了一些, 其他都是些小问题。

这个工作做完之后,关毅开始和段美女商量明天评审会议的事情,主要是具体 谈谈如何演示"支持 Safari"的这个功能。最后,他们决定各自演示自己所做的内容。

关毅选了几个常见的页面,准备明天通过视频会议共享自己的电脑桌面,然后用 Safari 浏览器打开这几个页面,给大家展现一下效果,接着再用其他浏览器打开同样的页面,供大家对比。

段美女也基本上和关毅一样,不过她更注重电子商务通的运转流程和功能,她准备给大家展示使用 Safari 浏览器从选购商品、下订单,一直到后台处理、发货等



一系列流程,完成功能方面的展示。

时间过得很快,到了每日 Serum 会议的时间,大家七嘴八舌地讲起了今天打扫战场的成果。从 SerumWorks 上看,前面 3 个 Story 都很漂亮地清理完毕了,很干净,一点遗留的问题都没有,大家都很有成就感。唯一遗憾的就是最后一个基本没动的 Story,大家看着它,觉得没有完成任务,很没有面子,有点忐忑不安,不知道明天的评审会议 Product Owner 以及其他人会怎么批评自己。

即使这样,关毅还是安慰了大家:"大家做的已经比第 1 个 Sprint 有很大的进步了,回去之后好好休息,明天振作精神,准备评审 Sprint 2。"

评审 Sprint 2

周五上午 10 点,评审会议正式开始了。加拿大那边除了 Product Owner,还来了几个关毅都没有听说过的整个产品的大老板。在北京,关毅也请自己的经理徐天参加会议。

演示时不谈技术

一看这个阵势,虽然经历过第 1 个 Sprint 的演示,但大家心里还是有些紧张。 关毅其实也差不多。一开始,他给大家展示了 ScrumWorks 上的内容,大致概括了这 个 Sprint 所需要达到的目标以及自己团队目前完成的现状。当然,说到最后一个功 能的时候,他很不好意思地说这个 Story 没有时间完成了,不过由于提前沟通过,所 以也没有受到什么质疑。然后,他打开 PPT,开始做演示。他的 PPT 有 20 页左右, 主要是一些项目背景、功能截图之类的东西。

放到一半, Product Owner Greg 有点不耐烦了: "嘿, 关毅, 直接进入主题吧! 给我们看看你们做的最终内容吧! 以后这样的评审会议就不要用这么长的时间讲PPT 了, 大家更关心的是实际的内容。"

又被教训了一下,关毅心里一阵紧张, 赶紧给大家展示小胡他们制作的 PDF 输出的单元测试的 Flash, 并且向大家解释, 由于前台代码没做, 没有比较好的演示方式, 所以做成了现在这个样子。另外, 由于之前他听小胡介绍了一些这个功能的技术实现细节, 于是他一边演示, 一边讲起了这些细节内容。

不过关毅刚讲了两分钟,就又被 Product Owner 打断了。大家显然不是那么关心技术细节的实现。

Product Owner 说:"技术细节不是这个会议所关心的,我们主要关心的是最后的成品。你这个功能没有完成前台的代码,所以采用 Flash 的方式展现单元测试的结果是一个不错的方式,尤其是在需要演示一些不容易好演示的功能的时候,这种方法值得大家学习。但是技术细节方面的东西,还是少讲一些为好。"



关毅点了点头,很快讲完了 PDF 打印的功能,开始讲解 Safari 支持的功能。这一次,由于完成了所有的工作,又有实际的场景可看,所以讲得很快,关毅自己也很有成就感。

关毅讲完之后,段美女接着给大家展示了她的全程使用 Safari 访问并完成一个 网上购物流程的工作。

在分别对两个演示提了几个小问题之后,大家都很高兴,Safari 支持的内容完成的质量很高,温哥华的经理们一个劲地称赞。

演示做完了,关毅心里七上八下的。虽然获得了一定的褒奖,但是总归没有完成任务,而且第一个功能的演示又做得那么糟糕,这让他很是忐忑。

评审总结和没有完成的 Story

关毅他们做完演示之后,大家又各自在自己的电脑上试了试新的功能,七嘴八舌地讨论了几分钟,期间提出了两个小的改进:一个是关于 Safari 支持上的改进,对方希望开发团队能够在首页中明确显示当前电子商务通所支持的浏览器,给用户一个提示,让他们自己选择适合的浏览器运行电子商务通;另外一个是在输出 PDF的时候给用户一些额外的选项,提示用户可以支持横向打印,而不仅仅是默认的纵向打印。这些意见都很中肯,关毅都一一记了下来。

Greg 说这两个改进马上会添加到产品的 Backlog 中,而且以后有这样的问题或者意见都可以及时和他沟通,根据产品的需要添加到产品的 Backlog 中去。

最后, Greg 做了 Sprint 2 评审会议的总结: "总体来看, Sprint 2 我认为可以打75 分的分数, 有欠缺的地方, 也有很不错的地方。确实, 有一个 Story 没有完成, 而且基本上没有开始动手做这个 Story。我想这既是计划的问题, 也是执行的问题。考虑到这个 Story 的优先级不是那么的高,以后有时间可以再拿过来做,或者直接在下个 Sprint 里做也可以,所以影响还不是很大。另外, Safari 这块的功能做得很好,很细心,我们大家看了都觉得没有什么问题,这是你们团队的一个进步。总而言之,失败的教训值得我们多思考、多学习,成功的经验也值得我们多总结。作为第 2 个 Sprint,你们还缺乏一些经验,能做成这样,已经非常不错了!希望大家继续努力,

越做越好!"

Product Owner 话音一落,大家都鼓掌了。有进步,还要再接再厉!

总结和思考

Sprint 评审会议实际上是个非正式的会议,但可以邀请高层参加。会议的气氛可以活跃一点,避免变成严肃的报告会。Sprint 评审会议上要避免过多地谈论技术细节,而是要重点关注最后的成果。





回顾 Sprint 2

回顾总是好的,对过去有个总结,对自己有个认识,对未来也会有所帮助。

关毅在周五下午主持召开了这个 Sprint 的回顾会议,基本上就算是经验总结大会了。

在这次的 Sprint 回顾会议上,大伙都显得比较放松,毕竟这个 Sprint 已经过去了,静下心来思考一下自己在这个 Sprint 里做的工作、犯的错误、取得的进展等,都是很好的一个过程。会议由段美女做记录。

于胖子说:"我觉得在这个 Sprint 里,我们总体上比上个 Sprint 更加像模像样了,有点敏捷开发的味道了。我承认,有一个 Story 没有完成是我们的问题,在这方面以后应该更加注重计划安排和实时沟通,我认为这是我们以后最应该克服的问题。另外,现在的每日 Serum 会议,我认为开得挺好的,这应该算是我在这个 Sprint 里的收获吧!"

关毅说:"嗯,总体而言,这个 Sprint 确实比上个 Sprint 有进步,大家也很努力。 其间我与 David 讨论了很多关于 Scrum、Agile 的问题,受益匪浅。比如说 Scrum 与 XP 的关系、经理在 Scrum 过程中的角色、加班的问题等,这些细节性的问题我想往 往是一个 Scrum 成败的关键,我们在这个 Sprint 弄清楚了,这就是一个很好的经验。"

小胡说:"我觉得我自己做的不好的地方主要是请假时间没有比较早的提出来,导致整个 Sprint 缺了一个人周的时间,破坏了计划的执行,对不起大家,以后我会注意的。另外,正如关毅所说的,我自己也从那些问题的讨论中学到了很多关于Scrum 的知识和实践思想,我想这确实是我们的一个很大的进步。"

段美女说:"最后一个 Story 没有完成,其实我们应该尽可能再提前一点和 Product Owner 进行沟通,这样可能会更好一点。"

关毅说:"的确,最后一周才提出来是有点晚,因为我们太在意最后得到一个好的结果了。本来是想尽全力去完成,觉得如果很早告诉他们我们做不完会很没面子。幸运的是,这次 Product Owner 和那边的高级经理没有提出多少质疑,是因为恰好我

们做 Story 的顺序都是从优先级最高的开始,从高到低往下做,最后剩下的那个 Story 重要性相对不是那么高。但是这绝对是个教训,以后有什么事情都应该尽早提出来。如果在第二周结束就提出来,说不定 Product Owner 就会对任务作一些调整。"

小胡也忙着道歉:"怪我,要是我不休假,说不定我们可以完成的。"

关毅说:"呵呵,够了,你刚才已经道过一次歉了,怎么又来了?不过我们还回避了一个非常大的问题,就是测试。按照 Serum 的要求,一个迭代结束的条件是功能全部结束,测试也要完成,并且没有严重级别为 1 和 2 的 Bug 存在,只允许有少量的严重级别为 3 的 Bug。在下一个 Sprint 里,我们一定要努力让测试介入进来,否则还是没有做到真正的敏捷开发,我们的迭代仍然不完整。"

大家都点了点头。

关毅接着说:"我认为我们可以讨论得更加细致一些,大的方面说了,也可以谈一些更加细节的方面……"

Scrum 回顾会议持续了 1 个多小时,大家感觉都很有收获,一个个自信满满地 走出了会议室。

回到座位上,关毅放松地看着窗外的景色:太阳已经快要下山了,晚霞映在天边,很是好看。



Sprint 3——深入 Scrum

这是关毅他们第3个Sprint之旅。

为了增强沟通,关毅在这个 Sprint 的一开始就经过种种努力,将团队成员的座位调整到了一起。由于对任务非常不了解,在随后进行 Sprint 计划时,他们遇到了一些困难,不过好在涉险过关。结对编程喊了很久,这个 Sprint 团队才终于尝试了使用工具来进行结对编程。就在项目进展越来越顺利的时候,金融危机的阴影也笼罩了整个 E 公司。电子商务通的团队是敏捷开发团队,所以采取了更敏捷的对策。 Sprint 3 的第 3 周,他们要做的任务当中有一个需要与另一个国外团队进行集成,为了提高效率,关毅出差到现场解决问题。可是他回来后没多久,就发生了一件棘手的事情。

Sprint 3, 出发

为敏捷调整座位

又是周五,不过这已经是 Sprint 2 的最后一天了。

为了庆祝 Sprint 2 的结束,中午关毅带着团队一起出去 FB 了一顿。饭桌上大家都很兴奋,继续回顾着 Sprint 2 的点点滴滴。胖子建议干脆以后吃饭的时候做 Sprint 回顾得了,大家一致赞同。

饱餐一顿,人有点昏昏欲睡。关毅则不敢懈怠,因为他还要做下一个 Sprint 的 计划。关毅坐在座位上,想起刚才饭桌上大家愉快的场景,又环视一下四周,突然 觉得异常的寂静——他的周围坐的都是其他 Team 的成员。两个 Sprint 了,关毅一直 觉得心里有块石头,好像一直有什么东西阻碍着 Team 的有效沟通,但又说不太清楚,这时候他才恍然大悟。

想到这里,关毅马上去找徐天,要求把 Team 成员都挪到一起坐,互相挨着,这才能做到敏捷式的沟通。可是这个要求却让徐天皱起了眉头。虽然他内心是非常支持的,不过关毅旁边坐的都不是自己的人,要想调整座位需要先和别的部门经理沟通,但他表示愿意试一试,不过万一有"钉子户",还真是不太好办。在关毅的座位周围有 3 位同事,但他们竟然分别属于 3 个部门,而且资格都比关毅老得多。E公司中国研发中心曾经一度扩张得很厉害,历史原因造成了新员工只能"见缝插针"地就坐,同一个部门的员工散坐在办公区域的各个角落是非常正常的事情。

到底还是徐天面子大,和人家老板好说歹说,有两个同意换了,小胡和段美女的问题解决了。最后还真碰到一个"钉子户",死活也不愿意换座位。没办法,只好"谈判"。"钉子户"的条件是:换也行,但想要一个角落,安静一点的,因为他对将要换到的胖子现在的那个位置不满意。最后徐天想了个办法,把自己部门 Franco 的位置让给他,然后让 Franco 坐胖子的位置。Franco 的位置风水最好,绝对让懒人垂涎三尺。正好 Franco 生性好动,喜欢热闹,对胖子的位置也非常感兴趣。这下好了,三赢啊!



两个小时后,终于折腾完了。关毅看看左右,心里那叫一个爽,以后再也不用 写很多无聊的邮件了,有什么事情直接张嘴就可以了,效率啊!

关毅正美着呢,突然段美女提醒他,Sprint 计划会议的时间快到了。关毅心里一惊,不过他还是马上想起来,Sprint 计划会议已经改了时间,变成了下周一。由于在上个 Sprint 的时候,David 建议关毅和 Product Owner 的 Sprint 计划会议,还有自己 Team 的 Sprint 计划会议,这两个会不要相隔太长的时间,并且最好让大家都参与,于是关毅在上周就通知 Product Owner 更改了 Sprint 计划会议的时间,以后 Sprint 计划会议都改到了周一的上午。想到这里,关毅马上通知周围的兄弟姐妹周一上午9点准时开 Sprint 计划会议,和 Greg 一起讨论 Sprint 3 的计划,不许迟到。对于小胡、段美女和胖子来说,这还是头一次。

总结和思考

敏捷强调面对面的沟通,创造一个有利于敏捷沟通的工作环境至关重要。有时候,动动脑筋就会大不一样。



Sprint 3 的计划——精益思想

星期一一大早,大家静静地等待着和 Product Owner Greg 的会议。对关毅他们来说是星期一的早上,不过 Greg 所在的温哥华还是在周日的晚上,属于加班,这对老外来讲非常不容易。Greg 进入电话会议后,继续肯定了大家上一个 Sprint 的工作,然后交待了这个 Sprint 的主要任务,希望关毅他们完成电子商务通的 API。接着,Greg 列出了所有希望完成的接口。

关毅听得头都有点大了。哇,这么多! 虽然关毅在加拿大的时候, Product Owner 提起过所有新版本要实现的功能的具体需求来源,不过关毅还是想让 Greg 再原汁原味地讲给团队听。原来最重要的需求来自两个方面,一个是与客户现有业务系统集成的需要,另一个是与 E 公司其他产品集成的需要。关毅以往有很多和客户打交道的经验,于是便问 Greg,客户最想要的是哪种集成的场景? 关毅一边看着接口列表,一边听 Greg 描述使用场景,同时大脑飞速地运转。Greg 说:"我们想把电子商务通做成一种面向服务的软件,API 是非常重要的功能,所以优先级很高。但是由于 API 的接口很多,所以我不认为你们还有时间再作别的 Story。"

这时候关毅收回了思绪,他说:"我们 E 公司过去的产品一向大而全,客户实际能用到的功能很少。这次我们可不可以采用精益软件开发的思想,从而避免一些浪费?"

Greg 一下子被说蒙了。什么?浪费?

关毅解释说:"按照我目前对产品的了解,结合刚才听你讲解的应用场景,我们完全可以只关心现有接口列表中三分之一的内容,具体我们可以再和架构师 John 深入探讨。"

Greg 很高兴,不过这还是有点出乎他的意料。Greg 说:"关毅,你们等我一会儿,我把架构师 John 也加入到电话会议中来。现在是周日,不过我想他应该能理解,因为没有什么比 Sprint 计划会议更重要的了。"

5 分钟后, John 果然加入到电话会议中。他有点不高兴地说:"关,你欠我一个人情,我还在享受周末呢。"关毅他们都笑了。

关毅说:"没问题,下回你来北京,我请你吃北京烤鸭。"

John 了解了一些情况后,对关毅他们的想法大为赞同。不但如此, John 还放权给关毅的团队,让他们自己对接口的重要性进行排序,并且可以自主选择当前 Sprint 的 Backlog。

这时候段美女插话了,几个人里她英语最好,当时关毅给他们进行 Scrum 培训的时候她也学得最认真。段美女说:"Greg,我是 Duan,我有个问题。在敏捷开发里,Story 不是应该按照用户的角度进行描述吗?而不应该从技术的角度进行描述。一个典型的例子就是用户应该以某个角色,通过某项功能或者操作,实现某个结果,而且结果一定是要有商业价值的东西。"

关毅有点偷偷冒汗, 段美女讲的不错, 不过他还是不太习惯。因为关毅过去所在的 X 公司等级森严, 哪有下边人说话的机会?

让关毅没想到的是, Greg 听后大悦, 还一直夸奖段美女说得好。

接着,Greg带着大家根据典型的应用场景把API这个需求分解成了若干个Story,



----敏捷开发故事

每个 Story 都是按照段美女所说的,以用户的角度,从商业价值考虑的。然后,Greg 根据经验排列了各个 Story 的优先级。

由于接口比预想中简化了很多,Greg 还想给关毅的团队再增加一些工作内容。 Sprint 计划会议本来就应该由团队来主导,于是关毅表示希望整个团队经过详细的 Sprint 计划会议以后再作决定,明天答复给 Greg。Greg 同意了,并把产品 Backlog 的列表给了他们,也是按优先级排列顺序的。

最后, Greg 和关毅他们商定, Sprint 3 最后演示的时间是在 1 个月以后, 演示的内容要求能够用实例程序展现 API 的功能。

总结和思考

避免不必要的浪费是精益软件开发的思想,也同时是敏捷开发所倡导的。Sprint 计划会议还要确定 Sprint 最后演示的时间和每个 Story 演示的方式。



Sprint 3 的目标——口号的力量

就这样,1个小时很快过去了,Greg 匆匆告别了团队,去和家人共享晚餐了。

北京时间是周一的上午。到了下午 1 点,属于团队的 Sprint 3 计划会议正式开始,会议时长两个小时。会议一开始,关毅就重申了 Sprint 计划会议的目标:制定一个 Sprint Goal (目标)和进行任务分解。Sprint Backlog 已经有了,现在要做的就是详细分解,把每个 Story 分解成若干个 Task,并且估算出大致的完成时间。

制定一个一两句话的 Sprint Goal 是 Scrum 倡导的,这其实就相当于标语和口号,但是区别在于标语和口号通常比较虚,然而 Sprint Goal 必须是实实在在的,这样整个团队才可以清晰地知道 Sprint 里最重要的是什么。

关毅他们在前两个 Sprint 里面没有制定 Sprint Goal,带来的问题是感觉缺少一种大的方向感,也少了一点对未来成就的期待感,虽然有 Sprint Backlog 和 ScrumWorks 提供信息,但是还是感觉来得不那么直接。存在即为合理。因此,类似于标语和口号之类的东西肯定是有价值的。想当年,毛主席在指挥百万雄狮过长江的时候就曾

6

经用过"打过长江去,解放全中国"这样的口号,不知道鼓舞了多少战士的士气。

所以, 关毅决定从这个 Sprint 开始强调 Sprint Goal。

Sprint Goal 应该怎么写才好呢?

胖子在那里抓耳挠腮,不一会儿,他笑眯眯地对大家说:"不如我们就写'做完 API 功能,让老板满意',如何?"

关毅摇了摇头,说:"不好,我们干活怎么能仅仅让老板满意?"

胖子抢着说:"哦,那就加上一个让客户满意。"

关毅笑笑: "不是这个意思。Sprint Goal 应该是实在一点,别玩儿虚的,要表明我们为什么要做这个 Sprint,应该达到一个什么样的结果。如果结果是老板满意,我们上一个 Sprint 已经让老板们满意了,是不是这个 Sprint 就可以放假了?"

小胡是个实在人,想了想,说:"具体一点、实在一点的话,是不是可以这样写? '利用 SOA 和 REST 技术完成商务通 API,包括的接口如下……'这样一二三四的列出来。"

关毅听后还是摇头:"你这也太实在了,最好别用技术术语,要从商业角度来谈。"

最后归纳了一下,关毅把 Sprint Goal 写成了"完成电子商务通集成接口,满足第三方系统与电子商务通集成的需求"。大家一致表示同意。

总结和思考

Sprint Goal 是个鼓舞士气的好工具。



失败的 Sprint 3 计划会议

好不容易有了 Sprint Goal, 关毅看了看表,已经过去 30 分钟了,他只好加快会议进度。

下一个任务是制定 Sprint Backlog。

维松Scrum之旅

Sprint 3 里面到底要放到多少个 Story? 因为有前两个 Sprint 作参考, 这次关毅他们更有底了, 因为他们可以计算出团队 Sprint 的速率。关毅调出了 ScrumWorks 的界面, 通过投影仪投射在墙上。

根据前两个 Sprint 的经验,每个 Sprint 大概需要完成 40 个 Story Point。在上一个 Sprint 回顾会议里,大家总结最需要改变的就是让测试真正加入到 Sprint 中来,因此必须给测试预留一定的时间。

他们初步计划用最后一周的时间来进行测试,这样将会有 3 周的时间用来设计和开发,那么在这个 Sprint 里,也至少能够完成 30 个 Story Point。电子商务通 API 的预估值为 40 个 Story Point,由于在 Sprint 计划会议中删减了一大部分内容,API 的估值肯定会小于 30。但具体减少到多少,关毅他们心里都没有底。他们只是觉得,必须把任务分解成足够小才可以。

最难的部分就是任务分解了。由于大家事先没有特别的准备,所以分解起来格外困难。关毅把这个问题记了下来,等 Sprint 回顾会议的时候一定要提出来,否则 Sprint 计划会议的效率就太低了。

眼看两个小时就要过去了,但是任务还是没有分解完。这可怎么办? 延长时间还是先就此作罢? 关毅发觉大家脑子已经转不动了,甚至有点麻木了,所以他果断地做出了决定: 今天到此为止。

大家你看看我,我看看你,有点沮丧,又有点不知所措,心里都在想,没弄完怎么办?明天怎么向 Greg 交待?

关毅看着大家,平静地说:"今天太失败了,我们过分低估了 API 的难度,没有充分的准备,这是根本无法做好 Sprint 计划会议的。"

总结和思考

在这里,Sprint 计划会议的时间拖得再长也难有好的结果,应该果断结束。在 Sprint 计划会议召开之前应该有所准备。



散会以后,大家都有挫败感,关毅也一样,而且他觉得自己是要负主要责任的,他觉得自己没有事先计划好。于是,回到座位上,关毅给 David 写了封信。

Hi, David.

我们现在开始做第 3 个 Sprint 了。没想到一开始在 Sprint 计划会议上就遇到了点麻烦,主要是整个团队,包括我自己,

都对 Sprint Backlog 不是特别熟悉。我们这个 Sprint 要做的工作还是很有挑战的,单从设计上讲就简单不了,可是设计还没有出来,如何做详细的任务分工呢?现在万分焦急,眼看第 1 天就要过去了,Sprint Backlog 还没有最终定下来。我们召开了 Sprint 计划会议,花了两个小时,可最后还是没有结果,主要是不知道最终团队能完成多少工作。请问 Sprint 计划会议遇到这种情况时应该怎么办?没有详细的计划可以开始一个 Sprint 吗?

关毅

写完上面的话,关毅发出了邮件。这两个多月里,关毅把 David 当成救命稻草,遇到问题总是向他求援。这一次,关毅没有像以前那样感觉可以放松一下,他心里还是不踏实,因为这次的时间比较急,通常 David 都会在第二天回复邮件。

关键时刻还要靠自己。关毅打起精神,开始仔细研读电子商务通 API 的相关需求文档。

重新计划 Sprint 3

仔细看过两遍需求文档后,关毅突然想起在第 1 个 Sprint 里用到的"照明弹" 思想。假如计划不了一个月,至少可以先计划半个月的内容。

半个小时以后,关毅重新把大家召集在一起。由于现在团队成员们的座位都挨在一起,所以非常容易沟通。这次又是临时决定开会,订会议室已经来不及了,所以大家正好在各自的工位上围坐在一起进行讨论。关毅再一次体会到了换座位的决定是多么的明智。

关毅拿着笔记本电脑,让所有人用网络会议软件连接他的机器。他打开ScrumWorks,一边说一边飞快地敲击键盘,不一会儿,一张系统的简易时序图就画出来了。大家顿时觉得豁然开朗。经过一段时间的休息,大家的脑子也好用多了,纷纷你一言我一语地拆解任务。然后,大家拿出扑克牌开始估算时间。



不过这样一弄,惹得路过他们工位的同事都忍不住多看他们两眼,甚至一个嗓门大的还叫了出来:"干嘛呢?上班时间不许玩牌。"

胖子回敬道:"我们没玩牌,我们这儿赌博呢!参观要给钱啊!"

关毅赶紧制止胖子,让他不要胡说,否则真进行不下去了。

赌博!关毅转念一想,在过去,规划软件的交付时间不也是一种赌博吗?时间就是赌注,而且这场游戏输的概率非常大。在传统的软件流程里,对计划的依赖已经达到了极致,人们往往要在计划上花费大量的时间,希望能够预先看到每一个细节并计划好。大量的研究也都想帮助人们如何更好地进行计划,如何计划得更准确。这在工业时代也许奏效,但是对于软件行业,几十年来一直屡屡碰壁。长时间周期的计划是很难精确的。然而我们可以尽量把风险控制在一个比较小的时间范围内,这就是迭代。如果一个迭代周期对于计划来讲仍然显得有点长,那么关毅他们这次就应该先计划出一个更短时间内的任务安排。

下班时间快到了,从所有人脸上洋溢的满足感和成就感可以看出,关毅他们的 Sprint 计划会议最终成功了,而且效果比关毅预想的要好很多。他们完成了这个 Sprint 中大部分任务的分解和计划,少数不明确的和复杂的部分留到下次继续。经过扑克 牌的估算,整个团队已经没有余力做除了 API 之外的工作了。为了要给测试预留 1 周的时间,所以,所有的开发工作都必须要在 3 周之内完成。

当晚,关毅把 Sprint 计划会议的结果汇报给 Product Owner Greg,Greg 也支持团队的决定。

第二天一早, 关毅收到了 David 的回信。

Hi,关毅。

你的问题越来越有水平了,说明你们的 Scrum 实施在不断深入。其实很多问题是没有标准答案的,寻找适合自己的方法才是最正确的。我建议你们的团队最好有一个人能在 Sprint 开始前的几天就提前进行一些准备,这样整个团队就可以顺利地过渡到新的 Sprint 中。Sprint 计划会议主要是为了制定 Sprint Goal 和 Sprint 计划,具体开多长时间,要以最终达成目标为准。但是如果事先大家谁都没有准备,那么这个会议开多长

时间也是没有效果的,只能是浪费时间。任何会议都需要准备,更何况像 Sprint 计划会议这么重要的会议。

还有一点需要再次强调的是,在做计划的时候,任何需求不明确的任务都要及时向产品负责人提出来。还记得上次培训时做的那个猜动物大小的练习吗?同样是狗,大小差别可能很大,如果不搞清楚,就会出大问题。

另外,你提到在没有详细计划的情况下可不可以开始一个 Sprint。当然可以,但是你不能没有任何计划。 Scrum 的价值观就是未来是不可详细预知的,也就是说你不可能做出详细的计划。 不过只要有计划,并且这个计划足够支持团队开始有效且有序的工作的时候,就完全可以开始一个 Sprint,然后再不断完善计划就可以了。

David

看完 David 的邮件,关毅不由自主地点了点头,他立刻把不明确的内容再次发给了 Greg。另外,关毅发现,自己的一些想法居然和 David 的建议比较一致。看来只要领会了敏捷开发的精神,剩下的应该就是灵活应用了,没有那么多的条条框框。

关毅转身看着身边的胖子、段美女、小胡,他们早早就来上班了,而且忙得不 亦乐乎,这让他感到很欣慰。



测试的介入

关毅今天的事情还很多,其中比较重要的就是和测试团队一起讨论如何在 Sprint 中进行测试的问题。关毅到现在才知道惭愧,之前一直以为自己的团队已经是敏捷团队了,已经是 Serum 团队了,可是在前两个 Sprint 迭代里面连测试都没有做,只有开发,开发完了就扔在一遍了。这在本质上和过去有区别吗?不过也不能全怪开发团队,测试团队一直都在忙着干别的事情,无暇顾及电子商务通新版本的开发。谁知道这算不算借口呢?总之,虽然说在一个 Sprint 里应该包含测试,但是大家却都没有把它当成高优先级的事情,自然不会把它当回事。

下午两点,关毅和北京这边的测试组长赵平,还有整个测试团队,开会讨论如何进行敏捷测试的问题。

赵平加入 E 公司 3 年了,刚刚被提升为 Leader,干劲十足。赵平的测试团队曾经测试过不少 E 公司的产品,今年年初转到电子商务通项目组,同样需要向加拿大汇报工作。赵平的团队在北京也有对应的测试经理,和关毅属于不同的两个部门。赵平的团队除了赵平之外还有 6 个人,另外,加拿大的电子商务通测试组也有两个人。这样看来,测试工作实际上是以北京为主的。赵平团队的 6 个人各有分工,有做功能测试的,有做系统测试的,总之人手还是比较紧的。赵平的团队在电子商务通 3.0 开发的前两个月里一直都在忙着这个产品 2.5.1 版本的测试工作,最精刚刚忙完。从 Sprint 3 开始,他们就可以全力投入电子商务通 3.0 的测试了。

赵平对敏捷开发有一些了解,所以关毅只是交待了目前开发团队实施的状况和问题,赵平就全都明白了。不过要想在迭代中进行测试,对于赵平他们也是个不小的挑战。过去他们有好几个月的时间准备测试用例,可以说,那几个月还是比较爽的,上上网,学点想学的东西,有点像旁观者一样"幸灾乐祸"地看着开发人员忙得底朝天。但是当开发结束,进入测试阶段后,赵平他们就开始进入疯狂的状态,加班就成了家常便饭。好在开发人员也不轻松,修改 Bug 的速度总也赶不上测试人员发现新 Bug 的速度。3 年了,赵平他们已经习惯这样了。

古人说得好,"一张一弛,文武之道"。赵平是个很有能力的人,当初选择了做

测试,就是想在 E 公司做出一番成绩。国内的公司普遍对测试人员不够重视,但是 E 公司却有非常好的测试人员成长通道。赵平一直很努力,成绩也很突出,自从升为 Leader 以后,他也不想再按老路平平淡淡地混下去。这不,恰恰赶上一个这么好的实施敏捷开发的机会,所以,他显示出了极高的热情。

赵平说:"刚才我们也讨论了一下,所以我们的目标就是在1个 Sprint 里完成所有的功能开发和测试。最后,那些功能所带来的 Bug 不能有严重级别为1和2的,可以有少量严重级别是3的。如果想要在1个 Sprint 里完成测试,在过去,我们有完备的测试用例,测试用例的编写是根据系统的需求和详细的设计文档来进行的,可是现在你们压根就没有那么详细的文档了,而且即使有,准备测试用例也来不及了,这一点让我觉得非常困难。"

关毅对测试流程不是特别了解,尤其是刚加入 E 公司不久,对这样一个大公司的产品测试流程更没有特别明确的概念。

会议室陷入了沉默。

过了一会儿, 赵平想了想, 说:"除非……"

关毅急切地问:"除非什么?"

赵平说:"除非我们打破 E 公司传统测试流程的常规约定。我们不再编写大而全的测试用例,其实过去很多的测试用例在早期根据文档编写完了以后,由于开发变动非常大,实际上已经变成废纸了。这次我们只写最简化的测试用例。不过你们没有详细的文档,这怎么办?"

关毅马上回答:"这个好办!敏捷强调面对面的沟通,既然我们都在一个地方办公,沟通起来比加拿大方便多了。有什么事能帮你们的,您说句话,一定效劳。"

赵平说:"能不能在你们开发功能前把需求文档准备好?不管这个需求文档有多 么简单。即使没有文档,你们也肯定会对需求有所了解。可不可以在那个时候就跟 我们详细讲一讲?这样我们好准备测试用例。"

关毅点点头,说:"这个没问题。"

雅松Scrum之旅

——敏捷开发故事

赵平接着问:"不过,我们什么时候可以开始测试?"

关毅说:"最后我们预留了1周的时间专门修改 Bug。也就是说,前3周就可以保证把所有功能开发完。"

赵平很高兴,这正合他意,于是对关毅说: "好的,那我们在 3 周后正式开始 Sprint 3 的测试。前两周的任务也很紧,因为我们需要把已经落下的'作业'补上,前两个 Sprint 的任务还等着我们去测试呢! 所以这段时间我们会随时骚扰你们的,呵呵。"

关毅笑道:"随时欢迎……"

不过关毅的心里还是有一点顾虑:如果开发任务很紧,大量频繁的沟通会不会影响开发效率?不过他又转念一想,情况应该不会太坏,良好的沟通会大量避免今后的麻烦,反而能节省总体时间。再说,这总比"少量的沟通 + 大量的文档"要强得多。

会议结束的时候,关毅心想,又扫除了 Serum 道路上的一个障碍,终于和测试部门建立起联系了。不过将来的测试问题一定也会很多,自己还需要对测试流程多加熟悉才好。

总结和思考

在 Scrum 中,实际上要求 Scrum 团队是跨职能的。一个 Scrum 团队 应该包含开发人员、测试人员、美工及文档人员。敏捷的开发流程迫切需要一个跨职能的团队。



结对编程

先理论, 后实践

电子商务通 API 的开发工作正在如火如荼地进行着,但是一个大的问题也随之而来——目前在组内真正有过 API 开发经验的只有关毅和小胡,胖子和段美女离关毅的要求还差得很远。

在每日 Scrum 会议上,大家的进度好像都还不错,可关毅却总是觉得有些不对头。于是,他找时间好好看了看胖子和段美女已经完成的代码。真是不看不知道,一看吓一跳:胖子的测试用例基本上没怎么写;段美女的问题也不小,她居然犯了实现 API 的禁忌——在实现 API 方法 A 时调用了 API 方法 B,甚至更多的 API 方法,而一般的原则应该是把多个 API 方法的公用处理逻辑抽象出来,封装到一个工具类里。像段美女这样直接调用其它 API 接口的做法会大大提高 API 接口间的依赖性,同时也增加了 API 的维护难度。关毅原本不错的心情这下子算是跌到了谷底。

如何在最短的时间内最大程度地提高胖子和段美女的 API 编程能力呢?目前的情况是时间紧、任务重,如果再花大量的时间去培训,那这个 Sprint 的目标可就真成了"目标"了。

有了前两个 Sprint 的开发经验,关毅已经成熟了许多。这次他没有再自乱阵脚,而是静下心来,耐心地查阅以往收集的资料,希望能从中能得到一些启发。关毅翻着翻着,不自觉地又看到了"结对编程"。说句实话,当初关毅在学习结对编程时,从心里认为这个所谓的"结对编程"都是在瞎搞,既浪费资源(对于软件企业来说,最大的资源是什么,不就是人吗?让两个人去做一件事,这不是浪费是什么?),又浪费时间("人月神话"早已被证明是个笑话,两个人同时做一件事绝对不会比一个人做少用一半的时间!),绝对是出力不讨好的事。所以可想而知,关毅是一点都没听进去。这会儿,他又看到了"结对编程"。

在上一个 Sprint 里,虽说关毅也很想做更多的尝试,可惜结果还是不理想。忽然,关毅又注意到在结对编程的简介中有一句话,让他实在不能不好好看看详细的



内容。"结对编程对于团队成员间技能的互相学习很有帮助。"这句话简直就像黑夜中的一堆篝火,照得关毅浑身暖暖的。

可是结对编程究竟是什么呢?难道就是两个人共用一台电脑,一个写程序,另一个在一旁指挥?当然不是。关毅仔细地读完了资料里关于"结对编程"的介绍和讲解,在他的脑海里已经能够浮现出"结对编程"的经典画面。两个人一起讨论,相互交替着编码,虽然同时只有一个人在编码,不过另外一个人也会在一旁指导,同时也可以重新审视已完成代码的逻辑和算法,使代码评审的工作在编码阶段就已经完成。从某种意义上说,这样做实际上会节省不少的时间,并且能提高效率。最重要的是,在目前的情况下,这个方法可以说是最能解燃眉之急的。

关毅拿定了主意,既然问题已经刻不容缓,早下决定就意味着早一步掌握主动权。"先发制人,后发制于人",今天就得搞定结对编程!

"说起来容易做起来难"可真是一句至理名言,这次关毅算是领教了。原以为采用结对编程会大大提高目前的工作质量,同时又可以尽可能快地培养起胖子和段美女的业务能力,谁曾想,一天下来,搞得大家筋疲力尽。

按照关毅的安排,关毅和胖子结对,小胡和段美女结对。开始工作前,关毅特意交待小胡:"今天的工作让他们两个做主力,咱们俩辛苦一点,在他们身后做指导,这样有利于他们提高,好吧?"

小胡一听, 乐了:"这有什么辛苦的?放心吧,老大!"

就这样,大家都忙活起来了。胖子和段美女打起了十二分精神,一丝不苟地写着代码。关毅和小胡分别坐在他们的身后,也都注视着屏幕上的已经写好的代码,出于本能地挑他们的"骨头"。

"等一下,别着急往下写了。你看看,说过多少次了,不要调用别的 API,你怎么又用了?"半个钟头内,这已经是小胡第3次警告段美女了。

段美女也有点不耐烦了,理直气壮地反问道:"这样多省事啊,既减少了代码量, 又增加了程序的可读性,为什么不能这么用啊?既然是 API,不就是让别人用的吗? 连自己人都不让用,还写它干嘛?" 小胡正在思考如何重构段美女的代码,经她这一番抢白,思绪被搅得支离破碎。 小胡这个气啊,心里说:"嘿,这人怎么狗咬吕洞宾,不识好人心啊。"

关毅在旁边也清楚地听到了段美女的埋怨,觉得苗头不对,急忙转过身,对段美女说:"段美女,小胡说的没错。在实现 API 方法时,不要把别的 API 方法引入进来,这是一个基本的常识。这也是主要考虑到 API 是要给别人调用的,所以一定要保证每个 API 方法的准确性、高效性、可维护性。不然的话,如果让大家随意地在API 方法里调用别的 API 方法,不但会增加被调用 API 方法的负载,也会增加 API 的出错率。比如,如果有个 API 方法被调用了,结果问题就出在这个方法里,那么我们从外部所看到的就是所有与这个有问题的 API 方法有关的方法都出问题了,修改起来就要要花费很大的力气了。你说呢?"

段美女不好意思地点点头,继续工作。

小胡向关毅吐了吐舌头,做出一副无可奈何的样子。关毅拍拍小胡的肩膀,小 声地说:"革命尚未成功,同志仍需努力。别急,慢慢来!注意态度啊!"

终于到了每日 Scrum 会议的时间。这天的每日 Scrum 会议都有一点诉苦大会的 味道了,主题就是"结对编程"。

胖子率先发难:"这简直是在摧残我们。我们本来经验就不多,之前的任务也不少,这下可好了,又加了一倍的工作量。而且工作的时候,背后还有人看着,这还让不让人活了。"

"说得没错",段美女接过话头:"还有就是,要知道,每个人都有自己的工作节奏,进入状态以后都是不希望被人打扰的。哪像今天,我就没消停过,时不时地被打断,你们说还怎么让我工作?"说完她狠狠地瞪了小胡一眼。

这时候,小胡的脸色也十分难看:"这能怪我吗?我当然也不想这样啊,可是我看到你的代码里有问题,我也是出于好心才及时告诉你的啊!"

关毅也头疼了一天,这会儿他可真的有些后悔了:"事已至此,也就只有两条路了,放弃或坚持,咱们举手表决吧!"

关毅环视大家。没有人说话,整个场面显得有些冷清,和平时热闹的景象产生

了巨大的反差。

过了一会儿,小胡打破了沉默:"我知道大家今天过得都不愉快。不是我们做得不好,我想可能是我们还不习惯两个人如此'亲密'地一起做同一件事。可是,我想大家静下心来。平心而论,这一天的时间,我们还是得到不少启发,不是吗?我的意见是,结对编程本身是没有什么问题的,问题出在我们自身上。我个人觉得,还是得坚持下去。"

小胡的一席话让大家都开始思考,只是这次大家想的不是要不要继续"结对", 而是如何才能更好的"结对"。

总结和思考

结对编程的阻力在于人,没有尝到结对编程的好处之前对其不理解是 正常的。



超强的结对编程工具

这一天的每日 Scrum 会议就这样结束了,大家都带着问题离开了。

关毅有些精神恍惚地往座位上走去,在路上不小心撞上了赵平,赵平笑着说: "干嘛呢?梦游啊。"

关毅抬眼一看,是赵平,就开玩笑地说:"别理我,烦着呢!"

赵平也没介意,继续问道:"到底怎么啦,哭丧个脸,谁又借你钱啦?"

"唉,要是借钱这么简单的事就好了。我们最近想采用结对编程来提高团队的战斗力,不成想,就一天光景,把大家搞得鸡飞狗跳的,接下来可怎么办啊!"关毅这次可真的是犯难了。

赵平收起了玩笑,一脸严肃地对关毅说:"结对编程可是敏捷开发中的一个亮点,我想你们应该是在执行上出了问题。敏捷的思想是不要拘泥于形式和及时的调整。从最近的接触中,我发现你和你的团队的个人自尊心和荣誉感都很强,所以我觉得你们可以换一种方式来结对。我前些日子看到 Eclipse 官方网站上有关于 ECF 的最

6

新消息, 你可以去看看, 或许会对你有帮助的。我还有会要开, 先撤了。"

这是关毅第一次听到 ECF 这个名词,有些不明所以,还没来得及细问,赵平就已经跑了。

回到座位上,关毅 Google 了一下"Eclipse ECF 结对编程",打开第1个搜索结果,是一个视频。看了一半,关毅整个人都傻了——这正是解决目前困境的钥匙啊!



资料库

ECF

ECF 的全称是 Eclipse Communication Framework , 是一个基于 Eclipse 的协同工作框架 , 它让开发人员在开发产品时 , 相互间可以及时地沟通和交流。而最新的消息是 , 有爱好者开发出了可以在 ECF 上共享同一个编辑器的插件。也就是说 , 两个人可以同时编辑一个文档 , 其中任何一个人的修改另外一个人都能实时捕获到 , 就如同是在面对自己桌面一样。

像 ECF 这样的插件这简直就是结对编程的福音啊! 关毅激动的心情真是难以用语言来形容,他现在唯一想做的就是尽快搭好这样的开发环境,让大家都能享受 ECF的最新插件带来的无比刺激的快感。不过遗憾的是,关毅花了一晚上的时间也没能搞定。无奈之下,关毅又把 Eclipse 官方网站提供的配置材料从头到尾读了一遍,还是实在想不出哪个地方做得不对,只好作罢,心有不甘地回家了。

到家之后,关毅还是一直惦记着如何尽快地配置好 ECF 协同工作的开发环境。转念一想,他突然发现自己对类似开发环境配置等"体力活"从来都没有重视过,团队里的其他人也是如此。虽然有时候大家确实会在这些"小事"上花费很多精力,但是在 ScrumWorks 上却从来没有单独作为 Story 或者 Task 列出来过,所以这些工作量也只有关毅他们自己清楚,别人是完全不知道的。这样一来,就容易给人一种错觉,好像这段时间内大家什么都没做一样。这样下去可不行。看样子,以后在做计划时一定要把这种工作作为一个单独的 Story 来对待,明天到公司后,一定要先去建一个新的 Story,计划配置 ECF 的时间和工作量。可是像这样的工作时间有该如何来估计呢?关毅心里嘀咕着,看来还是需要 David 帮忙了。

発**がScrum**之が 一教徒开发故事



Hi, David.

长夜漫漫,无心睡眠,不知道 David 兄是否也有同感啊,呵呵。

开个玩笑,不要介意啊。咱们言归正传。我们现在遇到了一个不大不小的问题,就是我们总是会把配置服务器和开发环境的之类的工作看作是每个人的个人问题。但是,这些"个人问题"往往是令我们非常头疼的事情,属于出力不讨好的工作。当然,我们也不会把这类工作列入到我们的 Sprint Backlog 里。所以,对于类似的工作,我们基本上都是在义务劳动,不会计入我们的工作量里。虽然目前的情况还好,只是我担心时间一长,这种工作一多,我的团队成员难免会有怨言。同时,我们的 Product Owner 也会奇怪为什么我们会在某些时候似乎什么也没做?特别是我们和他地处两个半球。

我的想法是,我们能不能把这种工作也放到 Sprint Backlog 里?另外,我们应该如何计划和评估这类 Story?毕竟我们预先是不知道具体需要花费多少时间的。你也知道,配置环境这种事,一般来说都是相当麻烦的。

期待着你的回信:)

关毅

E-mail 发出去了, 关毅的心情轻松了许多, 躺在床上翻了几个身就进入了梦乡。 一睁眼, 天已经亮了, 阳光照进了房间。关毅不禁感叹:"又是一个艳阳天啊。"

到了公司之后,关毅就立刻把小胡他们叫到了一起,然后简要介绍了 ECF 这个项目,告诉大家可以花上半天时间研究一下如何使用它,最好能尽快地配置好相关的环境。至于自己的"失败"经历,关毅是只字未提,他期待着小胡他们能给他一个惊喜。在这一点上,他们还真的没让关毅失望过。

世事无绝对,这次真的让关毅失望了。小胡他们 3 个人一起,严格地遵循 ECF 官方文档的指示,用了一上午的时间,一步步做下来,结果竟然和关毅昨晚一样。这不由得让关毅开始怀疑 ECF 官方文档的正确性了。可是如果 ECF 官方提供的文档都有问题,那岂不是说这个方案肯定行不通了?如此一来,提高结对编程效率的努力也都白费了。这样的打击对关毅来说可以算是史无前例的,他整个人都仿佛掉进

了冰窟窿里, 从头凉到了脚。

正当关毅努力地想爬出那个冰窟窿时, David 的 E-mail 不期而至。

Hi,关毅。

你提到的这个问题,我早就注意到了,这可能是软件开发人员的一个固有的习惯。不过,对于配置环境、安装工作软件之类的工作,我们也是需要区别对待的。简单点说,如果对于简单的软件安装,几分钟或者几十分钟之内能搞定的,我们当然可以不用再单独创建 Story了。但是,对于事前就知道至少需要在这上面花费几个小时的工作,我们一定要事先做好计划。这里的计划可不仅仅是估计时间和分配人员这么简单了,要考虑需要的问题很多。例如,我们要配置一台 AIX 的服务器。首先呢,我们要确定这台服务器的硬件工作是否正常,一旦发现硬件有问题,要及时联系厂商。接着呢,指要定一个全职的管理人员,在自己的团队里找一个也可以,让他负责这台服务器的所有工作,从学习 AIX 操作系统的操作开始,在学习的最初还可以联系服务器厂商的技术支持给他做一下指导,毕竟这样是最快的方法,以便尽快配置好这台服务器。在这里我想强调的是,一定要把配置过程中出现的问题和正常的步骤记录下来作为以后工作的指导,这样即便服务器管理员更换了,我们也不会手忙脚乱的。

对于时间的评估,如果你们是第 1 次做的话,评估起来会比较困难。这时候,我的建议是可以去请教有过相关经验的人,他们会告诉你他们会用多少时间,而你需要做的就是在这个基础上增加一些缓冲,毕竟他们是" 熟练工"嘛。

祝你好运!

David

看完了 David 的 E-mail, 关毅后悔自己没有再去找赵平。关毅总认为赵平是做测试的,应该没用过这个工具,可他现在才想到,既然赵平推荐了 ECF,他至少了解的要比自己多才是啊!

关毅终于抓到了一根救命稻草。二话没说,他拿起电话,拨通了赵平的分机:

维松Scrum之旅

---敏捷开发故事

"喂,老赵吗?我是关毅。咱们长话短说。你昨天告诉我的协同开发工具 ECF,你还记得吗?我现在遇到了一个非常棘手的问题。我们按照他们的官方文档去配置环境,却怎么也配不好。你能不能来给我们指导一下啊?"关毅开门见山地说道。

"哦,这个事情啊。伙计,不是兄弟我不帮你,实在是我也没有真正接触过这个工具啊。不过我很奇怪,你怎么舍近求远啊?"赵平说。

"舍近求远?这怎么讲?"关毅被赵平搞糊涂了。

"老哥,你们那边不是有个牛人 Franco 嘛,他可是个 Eclipse 专家。你找他咨询一下,一切问题不就都解决了?"赵平说。

听到这里,关毅笑了,他还真是忘了 Franco 这头大牛。越是身边的,越容易被忽略。

关毅赶忙站起身走到了 Franco 的座位旁,毕恭毕敬地说:"Franco,打扰一下,有个问题需要你帮忙。"

Franco 回过头,看到是关毅,一脸的惊讶,下意识地说道:"什么事啊?需要我做什么?"

关毅把遇到的问题从头到尾说了一遍, Franco 低着头沉思了一会, 说道:"这样吧, 马上就到吃饭的时间了, 你们先去吃饭吧, 我先试试在我这里配置 ECF, 有什么消息我再通知你, 行吗?"

看到 Franco 一口答应,关毅心里别提有多高兴了。这块石头终于落地了。

总结和思考

工欲善其事,必先利其器。其实帮助敏捷开发的小工具有很多,遇到问题不妨 Google 一下。



饭馆中的敏捷思想

关毅回到座位,发现赵平已经在等自己了。

赵平说:"走,今天中午带你去吃一个新开的新疆饭馆,拉条子,烤羊肉串,味道好极了。"

听赵平这么一说,关毅还真是有点馋了,叫上小胡他们,就跟着赵平一起去了。

到了饭馆,人不是很多,可能是刚开业,人气还不是很旺。关毅他们一落座, 大家就开始讨论准备吃点什么了,可是等了好一会,也没有服务员来招呼他们。这 时,小胡忍不住大声喊道:"服务员,我们点菜了,快点过来啊!"

这时,正在招呼关毅他们邻桌的高个子服务员说道:"不好意思,您稍等一会, 他们马上就点完了。"

关毅有些纳闷,因为在离他们不远的地方就站着另外一个个子矮一点的服务员,可是那个服务员似乎根本就没有理会他们的意思。

关毅对那个个子矮一点的服务员说道:"嘿,我们这边点菜。"那个服务员依然没有任何反应。

关毅对赵平他们无奈地摊摊手,实在搞不懂这是怎么回事。又等了一会,高个子服务员跑了过来:"几位吃什么?"

大家各自点了自己想吃的,时间不长,饭菜就上齐了。关毅他们早就饿坏了, 一阵风卷残云,饭菜一扫而空。

关毅叫服务员买单。那个高个子的服务员一来,关毅就好奇地问道:"服务员,怎么那边那个服务员就在那里站着,什么也不做啊!我们刚才叫她,她也不理我们,是不是对我们有意见啊?"

高个子服务员急忙说:"哪里,哪里,她怎么敢对您有意见啊!这是我们老板定下的规矩。为了能更好地服务顾客,我们每个人只负责划分好的4张桌子,别的桌

発移Scrum之が 無様开发故事

子的事情我们是不能过问的,还请您见谅啊!"

关毅他们结完账,走出饭馆,说有笑地往公司走去。

聊着聊着,胖子忽然冒出了一句:"这家饭馆想得还真'周到',一个服务员只负责固定的4张桌子,这样是不是就好像我们只负责自己的Task一样啊!"

胖子的一番话触动了关毅的神经: "只负责自己的 Task,那不就是说如果自己的 Task 全部完成了,也不会去主动地去解决别人的 Task,就像那个矮个子服务员一样?"关毅说出了自己心里的担心: "按照咱们目前的情况,大家每个人都有一堆 Task,能够在计划的时间内完成就已经很难得了,所以还不存在这种有人在计划时间内完成所有 Task 的情况。不过如果真的出现了这种情况,我们该怎么办呢? 毕竟有些时候大家做的工作在技能要求上会有很大的差别。这就好比餐厅的服务员和厨师,虽然厨师的工作很忙,但有些服务员却闲着。可我们也不能要求这些闲着的服务员去做厨师的工作啊!"

关毅说完,赵平点点头:"老关担心的很对。从刚才在饭馆看到的情况,我们确实可以发现一些问题。虽然不能让跑堂的伙计和掌勺的师傅相互协作,但是至少在伙计之间、厨师之间还是可以尽量多一些协作吧!比如麦当劳的服务员就经常会招呼在别的队伍的顾客过来,那才叫敏捷,呵呵。"

关毅笑了笑:"唉,现在怎么都成这样了,吃个饭咱们居然都能和 Agile 结合在一起了,真成职业病了。呵呵!"说到这里,大家都开心地笑了起来。

回到公司,关毅就收到了一个利好消息——Franco 已经成功地配置出了 ECF 环境,这让大家着实激动了一把。关毅几乎把能说的感谢话一股脑都给了 Franco。Franco 潇洒地回了一句:"没什么,小意思。有什么需要我帮忙的尽管说,我一定尽力而为!"

事情终于按照关毅的计划开始进行了。配置好 ECF 环境后,结对编程进行的非常顺利,胖子和段美女的能力也在极快地提高,团队的生产力水平也由此达到了一个新的高度。

那个新疆饭馆味道确实不错,当然除了服务。第二天,关毅他们忍不住又去了。 显然,第一天的问题老板可能也发现了,为了提高服务质量和服务效率,老板亲自 坐镇。他站在大堂中间,像指挥作战一般,看见哪个服务员动作慢了就催两句,看 见谁站在那发愣就训斥一声安排新的任务。经过他的协调,局面确实有所好转。关 毅他们一入座,老板立刻就让一个小姑娘过来点菜。不过好景不长,毕竟老板一个 人管不过来,而且所有的安排也不一定都是最合理、最高效的。这个饭馆最大的问 题是服务员没有眼力劲儿,什么倒水啦、送餐巾纸啦,还是跟不上。关毅不禁想起 了 David 带他们做的"命令与控制"的游戏,当每个人都是自己的老板,当一个团 队实现自我管理的时候,才能发挥最大的生产力。

后来,关毅和女朋友又去了一次那个新疆饭馆,发现那里的管理也已经有了些改变。老板不在,每个服务员固定负责 4 张桌子这一点还是没变,但如果某个服务员的"辖区"内没有客人,她就可以去支援客人比较多、服务员较少的"辖区"。虽然改进不多,但还是很有效的。通过一些细微的变化就能满足客户的需要,这不就是敏捷精神吗?

总结和思考



敏捷本质上也是一种先进的管理思想。



产品发布计划的变更

眼看就要到周末了,没想到这一天发生了件大事。

一大早,大家就收到了公司群发的电子邮件,E公司第一季度的财务报告出来了,受全球金融危机的影响,公司的销售状况不如预期的好,要求所有相关的产品部门采取相应的对策。看到这里,关毅心想:这么大的公司,应该和自己关系不大吧!即使有关系,也不会这么快就影响到自己。这几个月肯定还是踏踏实实做我的开发,毕竟项目计划是早就定好了的。

可是关毅错了。

第二天,温哥华的管理层召集温哥华和北京的所有经理以及 Scrum Master 开会, 关毅也参加了。这个会议首先传达了上面的精神:由于整体经济形势不好,电子商 务通的销售业绩也受到了很大的影响,不想办法肯定是不行了。其实,电子商务通 产品组的管理层得到消息要更早一些,他们一直在考虑对策。

对于研发,出乎所有人意料的是,敏捷开发恰好可以应对这场危机。这个会议上果断地宣布了一项决定:有两类 Story 被定为最高优先级,一类是如果发布的版本里有这个 Story,那么客户就会购买或者升级的产品,另一类是如果发布的版本里没有这个 Story,那么客户就不会购买的产品。也就是说,研发会变得更受市场和销售驱动了。

眼看金融危机愈演愈烈,日子将会越来越难过,不适应快速变化的市场就无法 生存。

接着,大家在会上考虑如何变更。以往这么做是有繁琐的流程和返工费用作为代价的。由于电子商务通已经开始尝试敏捷开发了,迭代式的开发使需求的变更相对容易得多。所以管理层决定不改变这个 Sprint, 从下个 Sprint 开始, 计划将会按照这个原则重新安排优先级,以制定 Sprint 的 Backlog, 所以,这个 Backlog 也许会包含原来不在发布计划中的 Story。

会后,徐天让关毅在这段时间里找机会多收集一些来自中国重要客户的需求或

者潜在需求,好为下一个 Sprint 计划会议做好准备。关毅这时才不敢掉以轻心,他想:如果研发跟不上,销售就上不去;销售上不去,就没有研发经费;没有研发经费,大家就都得回家了。

总结和思考

将投资和回报最大化,快速反映市场的风云变换,敏捷开发是应对危机 的最好武器。





现场沟通

出差到现场解决问题

又是新的一周,这已经是 Sprint 3 的第三周了。

关毅走在上班的路上,一边盘算着自己今天该干点什么,一边想着这个 Sprint 到目前为止进行得还算顺利,虽然也还陆续有这样那样的问题出现,但前两个 Sprint 中曾经犯过的一些低级错误已经没再犯了。关毅想,看来 Sprint 回顾会议还是很有用的。

刚到办公室, 段美女就冲关毅抱怨: "老大, XX 产品组提供的新的 Build 好像有问题,接口不对呀!"

原来这个 Sprint 要做的 API 有一部分就是使电子商务通与 XX 产品集成,形成一个完整的解决方案。由于电子商务通是新加入 E 公司软件家族的,所以 XX 产品原来提供的接口不能很好地符合电子商务通的接口需求,这就要求 XX 产品专门为电子商务通进行一些定制。

在这个 Sprint 的开始,关毅他们就开始跟 XX 产品组的人沟通。XX 产品的开发人员都在美国,而且他们还没有采用敏捷开发的方式,跟他们的沟通基本靠电子邮件。由于时差的关系,相互之间沟通的效率不是很高,往往中国这边发一封邮件过去,最快也要到第二天才会有回音。如果再赶上双方理解有误,就又要耽误上两三天。这不,今天又出问题了!

小胡在一边接着说:"是不是上周咱们发给他们的邮件里有些话没讲明白,让他们误会咱们的意思了?他们给的东西并不是我们想要的。我觉得现在这种发邮件的沟通方式不太好,一来一去的太耽误工夫了。而且他们不用敏捷开发,好像不太适应这种每日构建的方式。你看,咱们希望他们最好能每天提供一个新的Build,他们顶多一周才能出一个……"

关毅点点头,陷入了沉思。敏捷开发是最注重人与人之间的沟通的,而现在在

沟通上出了问题,该怎么办?想来想去,最好的方式莫过于面对面的沟通了,让双方都多了解一下对方的产品,先把以前存在分歧的地方达成一致,以后再通过邮件和电话沟通,应该就会好多了。

关毅是个典型的行动派,一想到这里,他就立刻找徐天商量去了。

听了关毅的想法,徐天也表示赞同,他也能感觉到这样下去不是办法。两人商量的结果就是关毅尽快去一趟美国 XX 项目组。

徐天说:"这样吧,今天就开始走出差流程,最快这周就可以过去了。正好刚才我们几个经理一起开电话会议,XX的项目经理Simon和我也达成了一致,就是希望你能尽快过去。"

关毅之前就有美国的签证,还没有到期,所以这次不用等签证,非常方便。

这么快? 关毅好一会儿才反应过来, 很高兴地答应了。

总结和思考

面对面的沟通是最有效的方式。在一个异地开发项目的初期,核心人员进行面对面的接触是非常有必要的,尽管费用会增加一些,但是从长期效果来看,这样会节省很多的沟通成本。



回到座位上,关毅把这个消息告诉了大家。大家之前都被跟美国项目组的沟通 问题搞得郁闷无比,听到这个消息都很高兴。

胖子打趣说:"老板就是老板啊,以'迅雷不及掩耳盗铃之势'就把我们老大打发到美国去了。老大,你这次去得跟他们的人搞好关系啊,打入他们内部,好让他们积极主动配合咱们的工作,这是多么光荣而艰巨的任务啊!"

关毅笑着说:"小胖,你的嘴皮子还是留着和他们的人交流的时候去练吧。对了, 我还得请你临时代理这个 Scrum Master 呢。"

胖子受宠若惊,心想,怎么会是我!他一直以为段美女甚至小胡在关毅心目中的座次比都自己靠前,不过他马上回过神来,一个立正敬礼:"遵命!"



其他 3 人都被他滑稽的动作和神态逗得哈哈大笑,早上被错误的 XX Build 带来的有点郁闷的心情一扫而光。于是大家接着埋头工作。

出差审批流程很顺利,部门秘书也很顺利地帮关毅订好了去西雅图的机票。接下来,关毅整理了一下自己这部分的工作,为他不在的那几天安排好 Backup,而最重要的 Scrum Master 的职责,就请胖子在这几天里替他担任。

"唉, 计划又被打乱了。" 关毅有点郁闷: "果然是计划赶不上变化啊!"

在出发前的时间里,关毅从公司内部网上找了不少 XX 产品的资料来看,又准备了一些用于向 XX 项目组介绍电子商务通的材料。

临行当晚,关毅在家收拾完行李,看时间还早,就打开了自己的博客,赫然发现这几天多了很多新的评论,有给他提意见的,也有向他提问题的,可见现在关注敏捷和 Scrum 人越来越多了。看到这么多人关注他的博客,关毅很兴奋,他认真地挨个看着。在他的那篇关于 Scrum 回顾会议的文章下面,有这样一篇留言。

博客上的留言

关毅,你好。通过这些天关注你的博客,觉得你在应用 Scrum 方面很有心得。我有个问题想请教你。我们公司也在采

用 Scrum 开发方法,刚刚开始几个月,可是我发现我们团队的开发效率并不高,不像你们那样能看出明显的进步,而且我们这里很多人都觉得在开过了 Sprint 评审会议之后,再开 Sprint 回顾会议没什么必要,认为太多的会议是浪费时间,所以都不怎么参加,以至于后来就直接取消这个会了。我能看出你们很重视 Sprint 回顾会议。请问它的重要意义在哪里呢?有必要每次都坚持开 Sprint 回顾会议吗?谢谢!

关毅心想:大哥,Sprint 回顾会议这么重要你们还不开,那怎么成啊!他马上回复了这篇留言。

博客问题的回复

回复楼上的问题。事实上, Scrum 十分强调 Sprint 回顾会议的重要性。从我们团队实施 Scrum 这段时间以来的经验和教

训来看,我也认为定期的回顾是很有必要的。

Sprint 回顾会议,顾名思义,是为了在 Sprint 刚刚结束的时候及时总结这个 Sprint 中的得与失,使团队在各方面持续成长。这就像一次比赛,在中场休息的时候,教练为队员们指出比赛中的亮点和不足,让他们在接下来的比赛中能更好地发挥。这跟 Sprint 评审会议有本质上的不同,想必你也了解,我就不多说了。

在项目开发过程中,有一个原则是,问题被发现和处理得越早,付出的代价就越小。但问题是我们在紧张的开发任务中有时候很难发现这些错误,更意识不到这些错误会带来的严重影响。Sprint 回顾会议就为团队提供了集中发现和解决问题的好机会。我个人觉得,在 Sprint 回顾会议上应该注意的几点是:

第一,不要流于形式。如果把 Sprint 回顾会议开成一个没有主题的漫无目的的讨论会,则既浪费时间,又没有效果。同时也要注意不要把它开成一个批判大会。Sprint 回顾会议的最终目的是学习,而不是绩效考核。

第二,总结出经验和教训只是第一步,如果只有总结没有行动,那么总结出来的东西最终只能是废话一堆。所以在 Sprint 回顾会议上发现问题之后,Scrum Master 应该督促团队在下一个 Sprint 中吸取教训,持续改进。这一点我是有切身体会的。我们的团队之所以成长得很快,都是因为大家懂得利用Sprint 回顾会议的机会发现问题、总结经验,持续成长。

祝你和你的团队在 Scrum 之路上越走越好。

第二天, 关毅一路顺利, 在当地时间的上午就到了西雅图, Simon 也早就开车到机场等着他了。

Simon 怕关毅因为时差的关系会觉得疲劳,建议他先去宾馆休息一天。关毅笑着说:"不必了,我精神得很呢。时间就是金钱,咱们直接去公司吧!"

Simon 载着关毅来到公司,让他稍事休息。

下午,关毅先给 XX 项目组的同事们全面介绍了电子商务通,包括产品的历史、现有功能、新特性以及这些新特性希望 XX 产品能提供哪些支持等。

维松Scrum之旅

听完关毅的介绍, Simon 他们很诚恳地说:"关毅,你这次来确实对咱们的产品整合很有帮助,之前因为对你们的产品不了解和沟通交流上的不方便,导致出现了一些错误和误解,我们感到很抱歉,我们现在知道应该怎么做了。"

XX 组的 Team Leader Alex 接着说:"我们都对你们采用的敏捷方法很感兴趣,不如你找个时间给我们讲讲敏捷吧。"

关毅想,这正合我意,省得你们不理解我们的做法,于是很高兴地答应下来。

周三,也就是关毅到西雅图的第二天, Alex 向关毅详细介绍了 XX 产品。

XX 产品家族其实很庞大, Alex 讲完后还特意给关毅留了不少练习题, 他说多用是熟悉一个产品最快的方式。等关毅把这些练习都弄明白, 也已经是下午了。

这一天跟 XX 的"亲密接触",让关毅受益良多,有些原来不太清楚的地方,经过 Alex 的解释也都弄清楚了。

周四的安排是关毅给 XX 项目组的同事们讲敏捷开发以及 Scrum。

结合电子商务通团队采用 Serum 开发这几个月以来的经验,关毅又比刚培训完那会儿给自己部门的同事们讲得好多了,内容丰富生动,听得 Alex 他们频频点头,又不停的地问问题。这样一来,关毅足足讲了3个多小时。

Serum 知识普及完毕, Simon 对他说: "关毅, 你们的 Serum 方式听上去确实很神奇, 我想我们也可以考虑采用这种开发方法来试试, 不过还得等这个版本发布之后才行。这样吧, 我们配合你们的 Serum 方式, 在你们开发 API 期间, 我们每天下班之前总结一下我们一天的主要工作发给你, 就像我们参加了你们的每日 Serum 会议一样, 就从今天开始! 然后咱们再商量一个彼此都比较方便的时间, 每周通两次电话, 尽量加强双方的联络。对话要比邮件沟通好得多。你看怎么样?"

Simon 他们积极配合的态度让关毅很感动,关毅和他们很容易就达成了一致。

关毅很兴奋,觉得自己这次来对了,算是圆满完成了任务。总之,关毅这几天过得很愉快,这天晚上,他还开着租来的小车在西雅图逛了几圈,看了看夜景。而 关毅不知道,这些天可把在北京的胖子给折腾坏了。

临时 Scrum Master

关毅走之前请胖子替他当几天 Scrum Master,胖子满口答应下来,他觉得自己好歹也算是有不少 Scrum 经验了,临时顶替几天 Scrum Master 还不是小菜一碟嘛。

周三是关毅出差的第一天,也就是胖子担任临时 Scrum Master 的第一天。前两周赵平的团队给他们找出了不少的 Bug,而电子商务通中国开发组这边又少了一个关毅,所以在这一周里,剩下 3 个人的任务就更重了,既要做新功能,又要改 Bug,忙得大家晕头转向。这不,已经到了每日 Scrum 会议的时间了,大家都忙得忘了时间,压根就没想起来。而往常的这个时候,就算大家没想起来,关毅也会适时地提醒大家。

到了快下班的时候,胖子习惯性地打开浏览器登录 ScrumWorks,当他打算更新自己今天的工作进展的时候,才猛然想起每日 Scrum 会议这回事来。他拍脑袋大叫一声:"哎呀!忘记开 Daily Scrum 了!"

小胡和段美女也一副恍然大悟的表情:"对呀!我说怎么老觉得少了什么似的!" "你这个 Scrum Master 失职了啊。"

胖子有点不好意思地挠挠脑袋说:"是我的错。不过今天好像大家都在改 Bug 吧?反正也没啥好说的,要不咱们把 ScrumWorks 上的状态更新一下,今天就算了吧……"小胡和段美女觉得他说的好像有点道理,也就作罢了。

结果,周四一大早就出问题了,预示着这又将是忙乱的一天。

早晨上班一打开邮箱,胖子他们3个就看见关毅发来的电子邮件,告诉他们XX产品组在明确了他们的需求之后,又提供了一个新的Build,已经上传到专门存储Build的服务器上,希望大家在这个新Build的基础上继续工作。

段美女动作很快,看完邮件就去连 Build 服务器,连了几次却发现连不上了, 急得她大喊:"胖子同志,Build 服务器怎么连不上了,你快看看!"

胖子只好放下手头的工作到"寒风凛冽"的机房去检查 Build 服务器,查来查去发现是 IP 冲突了,可能别的部门的人给服务器配置网络的时候不小心写错了,写



成 Build 服务器的 IP 了。这可不好找啊!胖子只好一边联系公司的 IT 管理员,一边给所有人发邮件,看有谁错用了这个 IP。好在他的邮件发出去没多久,就有个家伙回信说:"对不起,是我弄错了……"

虽然那个搞错 IP 的家伙回信很快,可这毕竟也费了胖子不少时间。胖子不禁大发感慨:"唉,真是不当不知道,原来 Scrum Master 真不容易啊。'要负责解决团队成员遇到的障碍和问题',时间都被这些各种各样的突发状况瓜分得很零碎。看来当 Scrum Master 还得有很强的时间管理能力才行!关毅啊,老大啊,你快回来吧……"

"哈哈,没事吧你。"胖子的感慨正好被过来找他的赵平听见:"快别发感慨了,我还有事要问你呢。"

胖子的脸顿时严肃了起来:"啊……"

赵平说:"哎,说正经的,你们前两个 Sprint 开发的新功能,我们上周就已经测得差不多了,这两天一直在等你们的消息呢。什么时候可以开始测这个 Sprint 的功能啊?你们得给我一些文档啊什么的,我们好写测试用例然后开始测呀。"

胖子愣住了:"文档?我们这里也没什么详细文档啊。Scrum 并不推崇详细的文档,这你不也知道嘛。"

"对呀,这我知道,"赵平接着说:"可是我们对你们的新功能一点都不清楚,那也没法测呀。之前的两个 Sprint 开发的功能,关毅都给我们讲了讲。我们得知道你们完成了哪些功能,才能写出专门针对这些功能的测试用例。那要不这样吧,你也给我们讲讲你们这个 Sprint 准备完成什么新功能,这些功能都有哪些作用。我们心里有个谱,写测试用例的时候就容易多了。"

"行,我整理一下现在手头已有的文档发给你们,下午给你们大致讲讲,然后咱们再商量商量还有什么更好的增进双方沟通的办法。"胖子说。

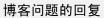
"嗯,我同意!"赵平满意地回去了。

下午,胖子给赵平的测试团队讲完 Sprint 3 的新功能后,赵平对他说:"胖子, 上午回去后我想了想,我觉得如果以后每个 Sprint 都要特意找时间让你们给我们讲 解,也不是长久之计,而且这样笼统的讲一次毕竟有些地方一时顾及不到,可能会 遗留一些问题。Scurm 不是强调团队中除了开发人员,还应该有各种其他职能的人员吗? 所以我觉得咱们应该尽量像一个团队那样行动。比如你们的 Daliy Scrum,如果我们也能每天一起参加的话,咱们彼此都能更好地了解对方的状态。我们测试人员能清楚地了解开发人员都在做什么,测试新功能时都要注意哪些地方,你们开发人员也会更好地了解到测试人员都发现了哪些 Bug,是怎么发现的……"

赵平的话说得人情人理,又提出了一个能有效解决长久以来开发和测试沟通不够问题的解决方案,胖子激动得一拍大腿:"哎呀,你说得太好了!其实关毅也曾经想过这个问题,不过当时因为 Sprint 3 才刚开始,你们还有测前两个 Sprint 功能的任务,工作跟我们并不同步,大家觉得没必要就作罢了。现在完全有这个必要了,我怎么就没想到呢!不如咱们就从今天开始吧。我再给你们申请一个 SerumWorks 的账号,我们每天都会在 SerumWorks 上更新自己负责的模块的工作进展和状态,你们每天登录进去看一下,不就能更清楚地了解我们的开发状态了吗?"

解决了与测试团队沟通交流不畅的大问题,胖子十分兴奋,他一时玩心大起, 也不给关毅发邮件汇报这个新情况了,竟然直接跑到关毅的博客上冒充读者留言。 他把情况改头换面大致说了一遍,询问关毅的意见。

第二天再打开关毅的博客, 关毅果然回复了他的留言。



你好。你所遇到的问题和我的团队遇到的问题很相似。在 敏捷开发中如何做测试?我认为这是敏捷中最困难的部分之

一,也是最容易被忽视的一个环节。

各种敏捷开发方法都强调 TDD (Test Driven Develop),就是所谓的"测试驱动开发",这主要是针对开发流程来说的。它要求开发团队在开始开发一个模块或功能之前,要先写好针对这个功能或模块的测试用例,然后再编写代码,并保证代码能通过这些测试。这种方式能有效地提高开发人员的代码质量,同时也能提高测试工作的效率。

而 Scrum 还强调 Scrum 团队的组成人员应该是跨职能的,也就是将开发人员、测试人员和其他职能的人员组成一个团队,形成一个整体。测试人员



加入到 Scrum 团队中来,通过参与 Scrum 实施的各种环节,比如 Sprint 计划会议、每日 Scrum 会议等,他们可以深入了解产品的各项功能和需求,了解开发人员到底在做什么、怎么做;他们可以在开发人员测试的薄弱环节投入更多的精力,比如开发人员主要做单元测试,这样就不太容易测试 UI 界面,那么测试人员就可以花费更多的努力在表现层的测试上。

总之,测试人员加入到 Scrum 团队中来,所带来的好处是不言而喻的。 我觉得你的处理方法非常好。我现在在出差,准备回去以后也提议我们的测试人员加入到开发团队中来。虽然我们目前不在一个部门,但是我们可以组成一个 Scrum 虚拟团队。欢迎你以后常来逛逛,咱们还可以在这里交流经验,看看测试人员加入之后的效果如何,呵呵。

转眼到了周五,关毅周末就要回来了。这 4 天的代理 Scrum Master 当下来,胖子挺累的,但收获颇多。

赵平和他的测试团队每天都来参加每日 Scrum 会议,这让他们对所要测试的模块了解得十分透彻,已经开始测试 Sprint 3 的新功能了。由于这个迭代结束后要出Beta 版,所以赵平还打算在功能测试之外再做一些系统测试和性能测试,以保证增加的新功能不会破坏原有系统的稳定性,也能确定电子商务通 Beta 版所能承受的压力大概是多少。不过这得等关毅他们将 Bug 改得差不多的时候才能进行,恐怕得到下周了。

总结和思考

胖子代理 Scrum Master,可以帮助我们进一步了解 Scrum Master 的职责。当然,即使是关毅也还仅仅是个新手。真正的 Scrum Master 要能够排除开发人员和产品负责人之间的障碍、确保 Scrum 达成目标、实现投资回报最大化、确保团队进度、确保团队状态具有高度的可视性、激发团队创造力、提高团队开发水平、采用各种优秀的工程实践、提高生产力等。

Sprint 3 冲刺

关于考核的问题

Sprint 3 的最后一周,关毅也从西雅图回来了,他们开始进行一些扫尾的工作。 马上要出 Beta 版了,大家也加紧了修改 Bug 的工作。与此同时,测试部门的同事们 也在赶时间测试 API 的相关功能,争取做到大的问题没有,小的问题能尽早发现就 尽早发现,并交给开发部门解决。

今天是周一,除了一些小的功能补充、完善之外,关毅团队的任务主要就是修改 Bug,并且做好一定的代码重构工作。事实上,有几个 Bug 出自代码设计上的问题,由于架构不是很科学,导致有些地方没有顾及到,在一些边缘情况的检测中出了问题,但是这样的问题又不是简单地增删一两行代码就能解决的,这经常需要做通盘的考虑,从一个比较高的高度来审视代码,做重构。

代码重构的工作实际上是很细节,但又很影响全局的工作。做得好能为产品功能增色不少,并且提高代码的质量,增加代码的可读性;做得不好就有可能导致程序出错,失去功能性,产生新的问题。这也是为什么关毅要求大家做重构的时候一定要同时进行 Code Review 的原因。不过有了 Eclipse ECF 的帮助,做起这件事情来也变得轻松有趣多了。

段美女和小胡两个人今天就对几个 Bug 用 ECF 做了一次结对,进行重构编程。 针对某些方法过于臃肿的问题,段美女提取出一个帮助类方法,只有较少代码的重 复。小胡则在一旁一边提醒段美女注意方法参数的问题,帮助她修正方法,一边借 鉴工厂模式的设计方法,将这个方法重新进行封装。针对一些变量名称的命名问题, 小胡也是考虑再三,采用了更加可读的、易识别的命名……

很快,重构的工作做得差不多了,小胡他们并没有针对那几个 Bug 而刻意做什么特别的工作,整体架构对了,有些东西就是自然而然、水到渠成的事情。就像这几个 Bug,很轻松就搞定了。当然,重构工作也不能追求过分的完美,毕竟交付时间在那摆着,软件的可用性还是第一位的,尤其是对关毅团队的电子商务通来说,



过完这一周很快就要发布 Beta 版了,以后还有的是时间进一步的完善和改进。

在周一的每日 Serum 会议上,开发小组的 4 个人加上测试小组的 4 个人(包括 所有功能测试成员和系统测试成员的代表)开始总结自己这一天工作,并对明天的 工作安排进行陈述,当然还有自己目前所遇到的障碍以及问题。

总体来说,目前的状况还是令人放心的。API 这个部分的功能基本上接近完成了,而一些功能上关系重大或者紧急的 Bug 都已经修复或者已经找到了解决办法,肯定能在这个 Sprint 结束之前修复。至于性能上的问题,虽然负载还不能达到很大,但目前来说也算可以了。

关毅对这样的情况比较满意,他最后说:"不错,我感觉这个 Sprint 大家做得都很好,虽然前期有一定的困难,不过大家都克服了。尤其是自从测试小组加入 Scrum中之后,我们成了一个紧密的团队,很多有价值的、比较紧急的问题,开发部门都能迅速得知情况,尽快地修正,我想这也是我们每天开每日 Scrum 会议所带来的好处吧。不管是开发人员还是测试人员,大家都能全盘地了解情况。明天我会和 Product Owner 以及客户代表讨论一下 Beta 版发布的具体情况,希望不会出什么岔子。大家不要松劲,继续加油!"

下了班,关毅一路哼着小调回了家。心情不错,慰劳一下自己吧!他买了两瓶啤酒拎了回去……

吃完饭,休息了一下,关毅想起了自己的博客。关毅的这个博客最近由于关注度上升很快,在这个以软件开发管理为主题的博客社区网站中渐渐有了一些小名气。虽然关毅的工作比较忙,写得不多,但是访问量还不错。也许都是冲着 Serum 来的吧!关毅心里暗暗想着。

打开博客首页,关毅发现又有网友留言了。



你好,上次你写的关于每日 Scrum 会议的内容,我看了觉得比较有意思,我已经说服老板也用 Scrum 来管理团队开发流

程了,以后有问题再请教你。谢谢!

6

关毅憨憨地一笑,看来自己也算是推广 Scrum 的先锋了。那个谁, Ken Schwaber, Scrum 的创始人,找到他一定要让他付我工资,哈哈。

下一条留言是一个提问,关毅扫了一眼,神情严肃了起来。

博客上的留言

你好。我最近一直在关注你写的博客,而且我自己也在网上看了一下关于 Scrum 的资料,确实觉得作为一种敏捷的思想实践,Scrum 既有理论也有实践,对软件开发团队管理很是有用。但是有一个地方我还是不太明白,就是说对于一个 Scrum 团队来说,怎样衡量每个团队成员的绩效呢?应该由谁来衡量呢?是 Scrum Master 吗?希望你能帮我答疑解惑。谢谢了。

这确实是一个很大的问题,实际上关毅自己也还没有相关的经验,毕竟刚来了 几个月,不过他也确实曾经考虑过这个问题。

一般的敏捷团队特别关注整体,强调可交付使用的软件产品,要求团队成员都有比较高的能力,然后参与到团队中来,最大限度地提升产品的功效。在组织管理中,敏捷强调的是整体团队,对每一个个人的付出与回报确实相对关注得比较少。这方面,敏捷社区也有很多不一样的声音。

关毅对于这个问题不敢妄下结论,赶紧问问 David 才是正道。他也很高兴在自己还没有碰到这样的问题之前,就有别人先提出来了,这无形中也能帮助关毅了解这件事情,防止以后自己出错。

David:

今天的问题,不是我提出来的,是我的博客上面有人留言,问了一个关于 Scrum 与个人绩效评估的问题。他问我,在一个 Scrum 团队中,应该如何评价每个人的绩效?由谁来评价? Scrum Master 应该是什么样的角色?我觉得这个问题非常好,但是以我目前的水平,怕答错了,所以问问你。不过,还是先说说我的看法吧!

一般的公司,每年或者每半年会做一次员工的绩效评估,主要是反馈给



人力资源部门,用来有针对性地调整员工薪水或者员工职位。当然我们公司 也差不多。我刚来不久,还不是很熟悉,但是我大致知道这一般是由经理打 分然后统计上报给有关部门的。

首先,我觉得一个 Scrum 的敏捷团队,从敏捷的角度出发,应该是一个有机的整体。那么,是不是可以把每个人的绩效评价为一样,从而反映团队的绩效就可以了呢?换句话说,经理可以评估每个 Scrum 团队的绩效,而无法评价每个团队成员的绩效。我不知道这种方式的可行性以及正确性如何。

其次,如果真的要按每个人来评价绩效的话,我想目前虽然经理还是我的经理,但是采取了 Scrum 的管理之后,我作为 Scrum Master,跟团队成员的接触和了解自然要比经理多得多,似乎由我来评价团队成员的成绩更合适——当然,如果经理同意这么做的话。但是从敏捷和团队精神方面来说,我又觉得自己不应该参与这件事情,毕竟这不是我的职责,我的任务就是建设好团队,提升团队每个人的技能,带领他们完成最终的目标。

以上是我的一些想法,但是没有找到自己认为满意的答案。呵呵,期待你的高见。

关毅

写完信,关毅就准备睡了,一天的工作还是很累人的。

周二是个大晴天,这样的天气最近不是很常见,尤其是前一阵子,总是会下点 小雨。

一上班,大家都很快各就各位了,API的工作正在有条不紊地进行。

段美女正在完成剩余的几个代码接口的实现工作。小胡正在做 API 的一些打包、集成相关的工作。而小胖则在那里改 Bug,一些是上几个 Sprint 遗留下来的 Bug,一些就是这个 Sprint 最新的 Bug。昨天测试部门又开了几个 API 相关的 Bug,不过看上去小胖信心满满,应该只是一些犄角旮旯的小问题。

关毅则和段美女一起进行接口相关的实现工作,也比较顺利。中午休息的时候, 关毅收到了 David 的回信,他快速地扫了一眼,信不算短,看来 David 对这个问题 也很重视, 很认真地做了答复。

关毅:

绩效评价这件事情,每个公司具体的做法不是很一致。敏捷提倡的是以团队绩效取代个人绩效,当然这有点绝对,其实可以以团队的分数作为平均分,然后再给每个团队成员评分。这样,就既有团队绩效,又包含个人绩效了。

Scrum Master 在绩效评价这件事情上的角色是起至关重要的作用的。只有 Scrum Master 才充分地了解每个团队成员到底做了什么、做得怎么样、进步大不大等,经理可能只能掌握个大概,所以由 Scrum Master 来评价是合适的。不过一般还是建议由经理来主导评价这个事情,然后 Scrum Master 给一定的客观的反馈意见。

注意,是"一定的",也就是说,Scrum Master 只需要给出一个大概的反馈就可以了。

总而言之,这个问题没有普遍的情况,也没有标准的答案,要结合不同 公司的不同文化灵活采用。

David

总结和思考

Scrum 专家建议对 Scrum 团队整体进行考核,不过在现实操作中,还是会对每个人进行考核。



突发事件

这一天很快就过去了, 关毅觉得自己很充实。

周三一大早,关毅就来到了公司,不过,一件意想不到的事情打乱了他的阵脚——Product Owner Greg 给他发了封邮件。



关毅:

还记得上次我跟你说的我们的大客户 BOU 公司吗?在这个 Sprint 开始的时候,我告诉他们我们将会在这个 Sprint 结束的时候发行 Beta 版,提供 API 集成的功能。BOU 的决策层很高兴。他们由于之前采用我们的电子商务通平台时没办法利用一些已有的资源将他们的信息系统盘活,所以走了一些弯路,效果还不是很理想。他们拿到 Beta 版之后会做一些评估,然后会集成我们的正式版,向公众发布集成后的新版本。

我刚刚看到他们已经在官方网站上做了一些宣传了,甚至在当地的一些 电子商务杂志上也都做了广告,说是新集成的电子商务网站将扩大容量,能 够支持更好的客户体验,并且将最终的上线时间明确定在3个月之后。

可问题出来了,他们发布了这个消息之后,网站的注册商家和普通用户人数都有了非常大的增长,虽然我们的系统整体上前端的访问量还能承受得住,不过按照你目前提供给我们测试的 API 的性能来看,使用 API 访问后台集成信息系统的并发访问还只是一个比较小的数量,毕竟我们用了远程网络访问等一些技术,提高了 API 的可伸缩型,但同时也导致了性能的下降。所以如果 3 个月后正式发布的话,按照 BOU 公司统计的在线人数和注册用户的比例关系,他们的同时在线用户数量将会远远超过我们的 API 的访问能力。但是如果推迟发布的话,BOU 公司的信誉问题不说,预期盈利也会下降,甚至会导致客户的流失,这个问题就大了。BOU 是我们的长期战略客户,非常重要!

关毅,我希望你尽早告诉我该怎么办,如果有什么问题,我们这边的架构师都会随时帮忙。这件事情非常紧急!

Greg

看到这封邮件,关毅也慌了。3个月,电子商务通 3.0 也才刚发布,哪有时间发布新版本啊!而且目前这段时间也不太可能把性能提高到那么高,更何况还有优先级更高的 Story 要做。

关毅走进茶水间,给自己倒了一杯冰水,擦了擦头上的汗,坐在位子上想了半

天。该如何是好呢?要不直接隐瞒这个信息,不告诉 BOU 公司这个问题,以后在给他们做升级的时候偷偷修改好。这样合适吗?这时候,关毅头上冒出了点点冷汗,他觉得这样肯定是不行的。如果真的这么做,出了问题他可担待不起这个责任。最后关毅决定招呼大家一起来商量,想出一个对策来。

这时候,小胡、段美女他们正有说有笑地聊着什么呢,看到关毅招呼大家的神情,他们心里都飘过一层乌云:看来事情不妙了。

来到会议室,关毅首先将 Greg 的邮件转发给大家,让大家先看看,了解情况, 他自己也简要介绍了一下 BOU 公司的情况,然后把自己刚才的想法告诉了大家。

大伙儿一开始都有些措手不及。本来一个好好的 Sprint,都做得差不多了,而且还很顺利,突然发生了这样一个情况,大伙的情绪确实都受到了很大的影响。

关毅安慰大家:"困难总是在顺利的时候出现,我们大家都应该有所准备才是。"

小胖相对属于乐天派一点,他抿着嘴笑着说:"我说各位,都提起点精神,没什么大不了的,先想办法吧。我觉得不如直接隐瞒这个信息,别影响我们的 Sprint。"

段美女接过话茬:"嗯,大家先想办法吧!不过胖子提出的这个想法——隐瞒信息——我看基本上就不要讨论了。关毅你觉得呢?责任太大,我们担当不起啊。我看我们还是实际一点,从目前的状况出发,看看什么应该争取?什么应该放弃?风险有多大?投入有多少?给出几个备选方案出来。"

小胡听了连点头:"我赞同段美女的看法。实际上这个问题是一个风险/回报模型的问题,也可以说是投资回报率(Return On Investment, ROI)的问题。我们应该采取什么的措施?这个措施有什么样的风险?这个措施有什么样的价值?需要投入多大的力气?最终我们来选取一个最好的方案,或者我们自己不做决策,而是把这些备选方案提供给客户或者 Greg。让他们来做决定,我觉得会比较好。"还是小胡有思想。

不隐瞒,不放弃

大家你一言我一语地交谈了一阵子,决定最后汇总几个备选方案发给 Greg。经



过和徐天一起商量,关毅他们最终总结的方案大致有如下4个。

方案 1: 我们应该直接告诉 BOU 公司导致这样的问题产生是系统 API 的访问容量的问题。我们也没有考虑到 BOU 公司的访问量会随着广告宣传增加这么多,我们会在下一个发布版中解决这个问题,设计并开发一个稳健、快速的 Transaction 并行任务处理系统和队列系统。在下一个版本的开发工作的开始,我们会立刻启动这项工作。目前来说,估计完成这项工作大概需要两个月的时间。所以我们建议 BOU 公司立刻撤下已有的广告宣传,推迟 BOU 公司的集成 Beta 版的推出时间,或者更改宣传口号,改变其中的发布时间。

方案 2:我们可以在较短的时间内实现一套简单的虚拟排队系统。这样每次用户登录后,系统在调用 API 访问后台的信息系统时,如果发现 API 的访问容量已经达到顶峰,便进行排队等候,前台将给用户"正在排队中,请等待"的提示信息。这样做的好处是系统可以在预定时间内上线,而且不影响功能。但是这样做的缺点也显而易见,系统将会变得很慢、很不可靠,用户的体验也会大大下降。这样的一套排队系统需要 1 个星期左右的开发测试时间。

方案 3: BOU 公司应该购买最新的高端服务器,并重新规划自己的网络设施,提高主机、网络的计算能力。这样的话,我们的 API 的访问性能和速度会加快很多,从而提高 API 的访问容量。我们推荐 BOU 公司及时更换 XXX 公司的服务器,并且组建多个镜像网站用来支持高访问量。这个方案对 BOU 公司的成本投入要求比较高,但是回报是长期的,BOU 公司的系统性能会得到进一步提升,而且一段时间之内也不需要再进行硬件的更新。

方案 4: BOU 公司可以考虑只开放部分接口,而不是所有接口来连接后台的被集成系统,其他接口的功能可以继续采取以往一直使用的旧的功能来实现(如果需要的话)。另外,BOU 公司也可以只为一些高级客户(比如长期往来的客户、销售额高的客户)提供这个新的 API 集成接口,而不是同时为所有客户开放这个功能。BOU 公司也可以在新注册用户中开启试用的功能并设置一定的限制条件(比如访问次数、频度等)。通过以上这些设置,可以大大降低对 API 的访问,从而满足我们的需求。而与此同时,我们可以在实行这种办法的同时进行设计和开发,提高 API 的能力,并且可以根据试用的反馈信息进行部分的修改。如果采取这样的方案,BOU在一个多月之后就可以全面开放 API 了。

第二天, 美毅把这 4 个方案发给了 Greg。

周四 Greg 就回信了,看来这确实是一个件十分紧要的事情。



关毅:

你帮了我大忙了。这次你做得非常好!

你给我提出了 4 个方案, 我认真的看了一下, 客户的成本以及回报都很清楚, 你甚至给 BOU 公司提出了一些运营方面的建议, 这非常好。

在目前这个状况之下,我们不能向客户隐瞒我们产品的缺陷或者问题,客户在敏捷开发中也是成员之一,这次我们学到了一个很好的经验。看来我们以后非常有必要让我们的客户在开发前期就参与进来,这样能让我们的产品更加接近客户的需要,并且能做到及时的沟通,那样的话就不会出现今天这么一个大家都很被动的情况了。

不过,这次我们提出的方案我已经交给 BOU 公司了。初步看来,他们虽然对我们前期没有明确、详细地说明 API 容量的事情有点抱怨,不过他们也承认自己有点急于求成了,他们也有责任。他们拿到我们的方案之后感到很高兴。BOU 公司的 CIO Kevin 告诉我,他们正在研究我们的方案,很快就会有结果。Kevin 说对我们的敏捷开发流程很感兴趣,他也希望以后能多参与我们的会议,参与到开发的过程中来,这样 BOU 公司也能更好、更快地采用我们的新的产品。

关毅,在没有拿到客户的最终决定之前,你还是按部就班地按照这个 Sprint 的计划继续做下去吧。明天我们开 Sprint 评审会议。

Greg

看完 Greg 的邮件,关毅松了一口气。他把邮件转发给了大家,要大家也各自体会一下。

下午,在这一天的每日 Serum 会议上,每个人都谈了自己的想法,大家都认为客户参与敏捷开发对 Serum 确实是非常重要的。



晚上回到家,关毅也没有忘记把这个突发事件记录在博客上。关毅觉得,作为一个 Scrum Master,这次的经历确实又让自己成长了不少。

到了周五,照例是做 Sprint 的评审和回顾。有了前两个 Sprint 的经历,大家都比较有经验了。这次会议 Greg 给了团队比较高的评价,虽然还是或多或少地有些小问题,但是瑕不掩瑜。关毅也很高兴,因为他在这个 Sprint 里经历了不少事情。Sprint 回顾会议后,大家都在心里暗暗较劲,下一个 Sprint,我们会做得更好!

总结和思考

Scrum 的可视性能够保证及时发现问题。不要隐瞒风险。要用非技术语言给客户提供足够的信息以便他们提前做出决策。



这是关毅团队的第 4 个 Sprint 之旅。

这次关毅团队拿到的 Sprint 4 的任务需求不仅仅是文字的,而是一段以假乱真的演示录像。另外,在 Sprint 4 开始之前,关毅他们意外地得到了两天"假期"。这两天他们做了很多事情:学习、调研、发现新的工具。不但如此,期间关毅还竟然努力促成了开发和测试人员的部门合并,在真正意义上实现了开发和测试是一个团队的梦想。由于一些原因,关毅他们重新选择了用 IBM Rational Team Concert 来管理他们的 Scrum 流程。在 Sprint 4,关毅有机会了接触了一系列的 Scrum 问题,比如临时团队成员、CMM、现场客户、低素质团队的管理、持续集成等,他对敏捷开发的理解也逐渐深入了。



延迟开始的 Sprint 4

两天"休息"时间

在 Sprint 3 的最后一个周五,关毅收到了产品负责人 Greg 的电子邮件。在邮件里, Greg 高度赞扬了关毅团队的成果,并对下一个 Sprint 的需求进行了描述。是啊,在经历了 3 个 Sprint 的开发工作以后,电子商务通产品小组胜利在望,产品 Backlog 里面属于北京负责的 Story 只剩下 Web 2.0 工作流设计器了。但是由于管理层已经宣布了要更改计划,看来想按部就班地执行产品 Backlog 里面剩下 Story 的可能性已经不大了, Product Owner Greg 正在和市场部门的人一起进行紧张的评估。

这个周末,关毅趁机放松了一下,好好陪女友逛了逛街,没有再想工作的事。 这样做一来是因为 Sprint 3 已经顺利结束,他要犒劳一下自己,另外也是因为下周 一他们不会马上开始一个新的 Sprint。

原来,关毅这次和 Peter 商量了一下,决定推迟 Sprint 4 的开始时间。在 Sprint 4 到来之前额外增加两天的时间,用来计划和思考。也就是说,以后两个 Sprint 中间会将有额外的两天时间不算在 Sprint 里面。

关毅和 Peter 一致认为需要给团队一个喘息的机会。"Sprint"的意思是快跑,但总是快跑谁也受不了。敏捷开发最佳实践里有一条就是保持可持续的进度(Sustainable Pace)。所以,在这两天里,团队不用担心燃烧曲线走得够不够好,也不用担心一些技术小尝试、小创新会耽误项目进度,Sprint 计划会议也更轻松了。关毅鼓励团队好好利用这两天的时间,不是休息,而是做一些以前想做却没时间做的事情。对关毅和 Peter 的这项决定,管理层也很支持。由于预留了一定的缓冲时间,所以这两天的修整不会影响最终的发布日期。关毅希望这样做会带来积极的回报。

总结和思考

•

缓冲时间在两个 Sprint 之间起到了承上启下的作用。

以假乱真的需求演示

周一到了公司,大家虽然不着急开始新的 Sprint,但 Sprint 计划会议仍然照常进行。Greg、Peter、John 和关毅的团队一起召开电话会议,讨论 Sprint 4 的计划任务。

由于北京这边即将负责的 Web 2.0 工作流设计器任务正好是某个欧洲大客户需要的,并且承诺如果有这个功能,他们就会考虑购买,这将会是一笔很大的生意,所以北京的任务不变,这是大家之前没有预料到的。而温哥华的任务确实做了一些相应的调整,删掉了一些不是非常关键的 Story。

会议开始了。出乎意料的是,这次 Greg 没有过多地凭空解说 Story 的需求,而是边借助演示边讲解。Greg 大概是从 Sprint 演示会议上得到了灵感,他准备了一个 Flash 视频,展示 Web 2.0 工作流设计器的界面和操作流程。Greg 的 Flash 展示了一个很酷的 Web 2.0 风格的界面,工作流各项属性的设计简单明了,每个流程活动之间通过动态的线条连接起来。

关毅他们看得有点傻眼。没搞错吧!不是 Sprint 计划会议吗?还没开始做怎么就能演示出这样的效果呢?

Greg 演示完,问关毅: "怎么样,清楚要做的东西了吗?"

关毅立刻回答:"太清楚了,感觉都像做完了。这是怎么回事?"

Greg 笑道: "看来这个 Flash 做得还挺成功的,连你都被骗了。我找了个 Flash 高手,仅仅作了个演示而已,但是你们要做一个真的出来。怎么样,有难度吗?"

关毅摇摇头,笑道:"呵呵,看来我们还是回去学 Flash 算了。需求我们大体明确了,只是技术方面还需要花时间调研……"

Greg 接着关毅的话说:"相信你们能找到办法。过去的 3 个 Sprint 已经证明了你们的实力,这次也一样。设计、开发全归你们了!"

胖子问: "能不能把那个 Flash 发给我们,回去参考一下。"

Greg 笑道: "当然没问题!"



团队听诊工具

在接下来的两天里,关毅和团队成员一方面深入进行计划和调研,另一方面进行了"头脑风暴",讨论了目前电子商务通系统存在的不足、可能的创新点以及有什么地方可以申请专利等。他们还开了 4 次研讨会,每个人都和大家共同分享一个自己拿手的技术话题,最好是与电子商务通相关的。关毅讲了工作流的标准,其实是为 Sprint 4 的工作做铺垫;胖子讲了 Safari 浏览器插件编程;小胡大讲特讲 Web 2.0;段美女对流程和开发工具最感兴趣,她这两天发现了一个 Sprint 回顾(Retrospective)的工具,感觉特别好用,于是推荐给大家。

段美女推荐的工具叫"团队听诊器"(Team Pulse Survey),实际上是一个用Microsoft Excel 做的问卷调查表。每个团队成员拿到表格以后,需要花几分钟时间填写,由 Scrum Master 收集并汇总,表单会自动给出团队成绩,自动生成图表,帮助团队迅速找到自己的问题和短板。这样,团队实施 Scrum 的总体状态就一目了然了。

关毅觉得这个工具太好了,当场就决定演练一下。于是让段美女把表格发给大家,不一会儿数据就出来了(如表 7-1 所示)。

实 践 得分 在你们的开发中,你们使用利益相关者目标来描述你们开发的 Vision 目标吗 User Stories and Use 你们是否使用 Use Cases 或 User Stories 来定义需求 8 Cases Unit Test 开发人员经常跑自动运行的 Unit Test 吗 5 Stakeholder Feedback 你们经常从利益相关者(Stakeholder)那里拿到反馈信息吗 7 你们会从一个具有优先级排序的功能列表中为下一个迭代选择 9 Prioritized Backlog 工作内容吗 你们会根据以往迭代的速率(完成的工作量)来重新计划当前 Estimating 8 的迭代吗 你们是否重视构建高质量的软件,在迭代结束的时候只剩下少 Working Software 6 量不重要的 Bug Time-Boxed Iterations 你们是否遵守迭代结束的日期,根据情况调整工作的内容 9

表 7-1 Team Pulse Survey 调查结果表

续表

实 践	解 释	得分
Daily Scrum	你们是否召开15分钟的每日会议,讨论今天做了什么,明天打	9
	算做什么和遇到了哪些困难	
Test Early	测试是否很早就介入?测试人员是否参与迭代计划并在早期的	6
	迭代就开始测试	
Reflections	你们是否召开经验总结会来持续提高你们的流程	9
Continuous	你们是否经常构建?一天好几个 Build 分数高,相反地,一周 1	3
Integration	个 Build 分数就低	
Self Directing Teams	团队是自我管理的吗? 经理只相当于教练的作用	9
Sustainable Pace	团队的进度是长期可持续的吗	9
Non Solo	你们是否召开代码评审、结对编程等会议	7

根据大家的回答, 生成的柱状图是这样的(如图 7-1 所示)。

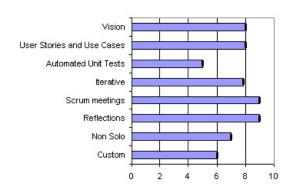


图 7-1 Team Pulse Survey 调查结果柱状图

大家对这个模版真是相见恨晚,这回段美女立了大功了。

关毅他们一商量,干脆重新召开 Sprint 回顾会议,而且马上就开。以前的回顾会议总是批评和自我批评,抓不到重点,也不系统,现在好了,有一个标准放在那里,做得怎么样真是一目了然。当然,除了模板中的内容,大家还是可以畅所欲言。关毅看了看数值,鼓励大家说:"在各项里面,我们得分高的不少,说明我们现在在敏捷实施方面越来越成熟了。当然问题也是有的,主要有 4 个:持续集成、自动化的 Unit Test,还有 Working Software 和 Test Early。下面我们来讨论一下这 4 个问题。

随着我们的开发工作临近尾声,持续集成变得越来越重要了。还有自动化 Unit Test,要是每次做完 Build 都能自动跑一遍 Unit Test 就好了,现在的问题一个是覆盖率太低,另一个是没有自动化。Working Software 和 Test Early 这两项的分数都很低,其实是一回事,不过测试组已经在跟进,所以我们暂且不提。那么,持续集成和自动化 Unit Test 就成了我们这个 Sprint 要改进的重点。"

接着,讨论开始了。

胖子首先发言:"以前只是提出了问题,不知道该怎么改,而且有时候问题还特别多,没有头绪·····"

关毅说:"其实这个模版就是一个标准的敏捷开发实践清单。我们接着讨论如何 用具体的行动来提高那两个弱项吧!持续集成,第1步,我们先找到工具,第2步 再讨论具体实施。"

段美女说:"找工具是我的强项,我来负责吧。"

关毅点点头,说:"很好,不过这自动化 Unit Test……"

胖子笑着说:"我来好了,我专啃硬骨头。"

小胡在一旁有点儿郁闷,说:"好活儿都被你们抢了,我怎么办?"

关毅笑笑:"对了,上周我刚接到通知,说 ScrumWorks 这个工具的授权快到期了,从上面得到的消息,我们不打算再对这个软件进行续费了。你来找一找可以替代的工具吧。"

"Yes, Sir!" 这下小胡乐了。

最后,关毅作了总结:"在接下来的时间里,大家不但要高质量地完成开发任务,还不要忘了开发任务之外的这些工作。这些工作虽然重要,但不紧急。不过一旦做好了,会大大提高我们的开发效率。"

大家会后都各忙各的去了,关毅一个人坐在座位上静静思考。这两天的收获还 真不小,这次算是开了个好头。怪不得人家 Google 这么成功,老有很酷的东西做出 来。Google 的员工周五可以不用做项目,完全自由安排,想干什么都可以,就这样,

异想天开、天马行空才有可能,才能做出最棒的东西来。

总结和思考

Team Pulse Survey 的各项实际上告诉我们应该如何成为一个成熟的 Scrum 团队。



部门重组

正在这时,从旁边会议室传出的争吵声打断了关毅的思绪。怎么回事?好奇心 驱使关毅站起来,悄悄地走了过去。

会议室里坐着一群经理,好像在为什么事情争吵,具体内容听不太清楚,但矛盾似乎集中在开发和测试之间。关毅过去在 X 公司,由于条件有限,测试部门经常为了抢机器、抢时间之类的事情和开发部门打架,没想到 E 公司也是这样。

关毅回到座位上,心想:现在我们和测试组都属于一个 Serum 团队了,为什么还从属于两个部门?要是能部门重组就好了。部门按照产品划分,而不是像现在这样按照职能划分。电子商务通的人都划分在同一个部门,那多方便,多敏捷啊!不过想法虽好,但实施起来肯定有困难,还是先跟徐天说说吧。

关毅把这个想法告诉了徐天。徐天摇了摇头,面露难色。部门之间的调整可不是小事情啊!于是他对关毅说:"你的想法不错,但要说服我们的大老板 Chris 做这项决定可不容易啊。你想想,如果看不到特别多的好处,而且这个部门调整又会牵扯到很多资源的重组,还有人事上的调整,恐怕是有点难度的。"

关毅点了点头。这件事的确不容易。怎么才能说服 Chris 呢? 什么时候提出来比较好呢?

徐天又说:"我们每年都有机会对组织进行一次调整,都由 Chris 最终决定,这一次调整的时间也快到了,我们可以向 Chris 提这个建议。你先准备一下,把各种利弊写清楚,然后给我看看,帮你改改。由我来发出去比较好。"

听完徐天的话, 关毅高兴了。

雅松Scrum之旅

----敏捷开发故事

写东西是关毅的强项。敏捷的武器再加上徐天的建议,关毅的邮件写得非常漂亮,"Yes"和"No"非常清晰,最后留给 Chris 的事情就是做选择题。当天这封邮件就通过徐天之手发给了 Chris。

总结和思考

按照产品整合部门可以大大提高执行效率,开发和测试在行政上属于一个部门有利于使他们结合成一个真正的 Scrum 团队。



新的敏捷管理协作工具

第二天,关于敏捷工具的调研也有结果了。小胡把报告发给了关毅,关毅看后非常兴奋——IBM Rational Team Concert,就是它!在电子商务通 3.0 正式开发之前,小胡他们就做过一次调研,这次小胡更详细、更深入地调查了 IBM Rational Team Concert 这个产品,收获颇丰。



资料库

• Jazz

IBM Rational 推出的 Jazz 是一个创新的团队协作平台,它将成为下一代 Rational 软件开发平台的基本框架。Jazz 平台的结构如图 8-2 所示。

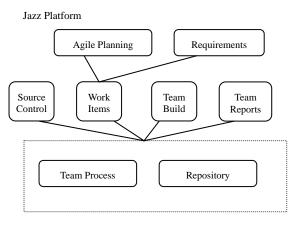


图 7-2 Jazz 平台结构图

IBM Rational Team Concert (RTC)是第一个基于 Jazz 技术的商业产品,也是 Rational 的下一代协作软件交付平台。

RTC 特别适合敏捷开发团队,主要提供了 3 大块内容,分别是工作项管理(Work Items)版本控制管理和构建管理。RTC 是基于 Eclipse 的,所以采用 Eclipse 开发 IDE 的团队使用 RTC 将是非常方便的。这样做,RTC 提供的 3 大块内容又和开发本身无缝式地集成在了一起。

RTC 能够带来很大程度上的信息透明化,各种角色的用户都可以获取自己感兴趣的信息,还可以通过 RSS 自动获取。用户可以随时了解团队开发过程中的各种动态变化:在今天的开发任务中谁提交了新的代码?一共做了多少次构建?哪些失败,哪些成功?到现在为止我们的迭代开发完成了多少?用户甚至可以通过制定查询条件(Query)随时查询自己经常需要的结果。

RTC 的核心是基于 Work Item (工作项)的沟通协作。在过去没有一个好的工具的时候,大量的精力都花在通过电子邮件、文档、绘制各种报告等进行沟通上了。但这些都不能解决这样一个问题,就是所有的文档和工作依然是脱节的。而通过 Work Item 机制,在 RTC 中,开发人员完全可以将全部的需求、代码开发、测试等内容作为工作项的一部分进行统一管理。

关毅很兴奋,看来 RTC 不但具有 ScrumWorks 的基本功能,最重要的是,它还有强大的协作功能,它与现有开发工具和管理工具的无缝集成功能实在是太"彪悍"了。

关毅当晚就和 Peter 商量使用 RTC 这件事。Peter 也非常兴奋,最后的决定是立刻上!一番周密的布置后,任务安排是北京负责数据的迁移,加拿大负责系统的安装和配置。

在 jazz.net 注册账号以后,就可以下载 RTC 的服务器端和客户端产品,系统配置起来也不是很复杂,配好以后,就可以开始 Serum 的 RTC 之旅了。

关毅先创建了一个迭代,也就是 Sprint 4 (如图 7-3 所示)。Sprint 4 迭代建立好以后,就可以创建 Work Item 了,也就是创建 Scrum 里的 Story (如图 7-4 所示)。

维松Scrum之旅

——敏捷开发故事



图 7-3 在 RTC 中创建 Sprint 4 迭代



图 7-4 在 RTC 中创建 Sprint 4 的 Work Item

在 RTC 中, Work Item 可以是 Story, 还可以是 Retrospective、Impediment、Defect 或者 Task。关毅把 Story 的名称定为"Web 2.0 Workflow Editor",接下来就是任务的细化和分配了。

关毅他们在 Sprint 计划会议之后制定了全部的任务,并把这些任务分配到每个人,一切都进行得非常顺利。

开始工作了。

每天,大家都会登录到 Rational Team Concert 上, Eclipse 的环境使团队成员倍感亲切。和过去一样,关毅他们依旧是每天花 10 分钟更新状态,以便随时看到剩余的工作量。每一项工作的状态,是开始、进行中,还是结束,关毅一目了然。另外,他们还使用 RTC 的沟通协作功能,对于每一个任务,大家都在一个类似于BBS 的区域讨论,然后把所有中间的讨论过程自动存档,形成有价值的文档(如图7-5 所示)。



图 7-5 Rational Team Concert

一周以来, RTC 为团队敏捷保驾护航, 一切都进行得很顺利。

就在这周末的时候,关毅和所有人都收到了大老板 Chris 发的通知: 电子商务通小组的测试组由原来的测试部门调整到徐天的部门,原先测试小组的员工统一向徐 天汇报。

关毅盯着邮件愣了半晌,居然没发现徐天已经走到他的身后。他猛一回头,吓了一跳。

徐天笑着说:"关毅你真棒,居然把 Chris 说服了。不过因为我们是敏捷开发团队,所以才作为试点破例调整,别的产品暂时不动。不管怎么样,说明 Chris 很支持敏捷开发。"

关毅说:"太好了,不过赵平他们最好也能和我们坐在一起。"

徐天笑了:"他们正准备搬呢,下周一就能搞定了。"

总结和思考

IBM Rational Team Concert 是一个功能强大的、能够支持各种敏捷框架的项目管理工具。





敏捷测试

为什么不报 Bug?

电子商务通的开发和测试团队成功合并之后,虽然两个团队的座位都搬到了一起,测试人员也每天按时参加每日 Scrum 会议,可是关毅总觉得似乎还有些地方不太对劲,这样的感觉以前也有过,但具体是哪儿不对劲,他却一时说不出来。想了一会儿,他摇了摇头,决定等头脑稍微清醒一点的时候再考虑这个问题。

这时,他听见隔壁工位负责产品功能测试的吴霞大声喊小胡:"小胡,你做的这个地方好像出问题了,你能不能过来一下? 我懒得去 Bug 管理系统里写了。"

听到美女呼唤,小胡乐颠颠地跑了过去,一边还嘴硬:"你可真懒,现在不愿写, 以前怎么愿意写呢,还给我报了那么多个 Bug,弄得我头大。"

"你以为我愿意啊?"吴霞辩解道:"以前不是因为咱们的座位离得远嘛,报一个 Bug 那么麻烦,要写一大堆字说明 Bug 是怎么发现的、发现时的环境设置、重现 Bug 的步骤等。有时候可能因为描述得不够清楚,还让你们开发人员误解了我们的 意思,告诉我们这个 Bug 已经解决了,等到下一个 Build 出来我们再去验证的时候,却发现原来的 Bug 其实还在……这样一来一去,要耽误很多时间,还浪费精力。说实话,到目前为止我对这个敏捷方法虽然还没有多大的感触,但至少咱们的座位搬到一起之后,我还是觉得方便了很多。"

小胡一看自己一句话引出她那么多话来,不敢再说,闭上嘴巴老老实实看问题 去了。

说者无心,听者有意。吴霞的一席话却让关毅恍然大悟:"哦!我明白了!"关 毅一拍脑袋,终于想明白到底是什么地方让他觉得不对了。

测试团队的新角色

原来两个团队合并之后,虽然开发和测试组成了一个敏捷开发团队,可是测试

人员好像并没能马上适应他们的新角色。也就是说,他们还不够敏捷。比方说,他们在参加 Sprint 计划会议、每日 Serum 会议的时候很少发表意见,好像认为这些会议主要是开发人员参与的,他们只是列席旁听而已。可能也是由于在传统的开发模式下,开发和测试之间一直有隔阂,似乎测试人员是专门给开发人员找茬的,所以现在赵平的测试团队也很少主动参与到开发过程中来,与关毅他们交流,像今天吴霞这样主动找小胡过去讨论问题的情况很少见。不过这也是个好现象,说明测试团队正在朝敏捷的方向前进,他们正逐渐敏捷起来。

下午5点,又到了每日Scrum会议的时间,大家陆续往开会的讨论区走去。

负责产品系统测试的李杰说他的主要工作还是在为正式测试准备测试环境。关 毅皱了一下眉头,他记得好像在上周的几次每日 Serum 会议上,李杰也是这么说的。

下一个是吴霞,她说她在新 Build 上测试新功能的时候偶然发现一些原有的功能也出现了问题,可能是添加的新功能对以前的功能有所影响。

听到这里,一直没说话的赵平开口说道:"嗯,这个问题是我忽略了。因为咱们的电子商务通 3.0 是在原有系统的基础上继续开发的,那么在测试新功能的时候必须先对原来的老功能做回归测试,以保证新加进来的功能不会破坏系统原有的功能和完整性。还有,咱们应该尽量实现自动化测试,这也是事半功倍的。这两个问题我们会后再找个时间详细讨论一下。"

关毅最后一个发言,他说:"我趁这个机会说点跟每日 Scrum 会议无关的题外话吧!我觉得自从咱们这两个团队合并成一个敏捷开发团队之后,大家好像还没从原来的角色中转变过来,咱们的团队还是不够敏捷啊!敏捷团队中需要有各种不同职责的人,测试人员更是承担着重要的角色,他们和其他所有的团队成员一起,为产品的最终成型负责。测试人员和开发人员不是两个对立面。咱们在一个团队中,应该增强沟通和协作。不知道大家有没有注意到今天小吴叫小胡过去讨论问题的情景?小吴发现了一个 Bug,但是她不想在 Bug 管理系统里写一大堆文字来描述 Bug,然后等着小胡在 Bug 管理系统里看到之后才去修改,最后等到新 Build 出来再去验证……这些都非常好。既然大家坐得这么近,就应该多倡导这样的主动沟通,不用去绕那么大一个弯子了!你们看,今天小胡过去看了之后,回来稍微调试了一下就解决了那个 Bug,这样一来,双方都节省了大量的时间,又大大提高了效率,何乐



而不为呢? 当然,对于简单的 Bug,这样做是可以的,复杂的还是记录一下比较好。 所以说,大家要充分利用现有的条件,加强彼此间的沟通。敏捷很强调沟通和协作。 大家一起加油,让咱们的团队真正敏捷起来吧!"

听完关毅的话,吴霞和小胡对视一眼,很有感触地频频点头,其他人也纷纷陷 入了沉思……

回到座位上, 赵平给测试小组成员们发了一封电子邮件。

Hi, 各位。

刚才关毅的话说得很有道理,咱们已经是敏捷团队中的一份子,大家要及时调整心态,积极主动地与开发人员沟通,同时主动参与到项目开发中来。

敏捷开发定义一名好的测试人员在于他不仅能做好自己本职的测试工作,还愿意并且能够帮助团队的其他成员解决问题,以实现团队的共同目标为己任。在敏捷团队中,还要求测试人员在项目时间紧迫的情况下能转变职能,完成其他角色的一些任务,达成团队的最终目标。当然,敏捷也要求开发人员在测试任务紧要时也能转变角色去帮助完成测试任务。我想,这样的要求无论对团队还是对团队成员都很有帮助。因为大家在职能转变的过程中,自身的各项技能可以得到锻炼和增强,从而获得更大的发展。

因此,大家完全可以积极一点,主动熟悉产品的系统框架、架构设计, 甚至代码实现等。随着大家对产品的理解越来越深入,相信会对大家设计测 试用例和提高技能等方面有帮助。

赵平

系统集成测试怎么办?

赵平刚写完邮件,关毅就过来找他了:"赵平,我注意到李杰好像从上周开始就一直在做诸如准备环境、看文档之类的工作,一直没有开始测试。是不是有什么问题啊?"

赵平说:"我正想跟你商量这事呢。在系统测试方面我们确实遇到了一些问题,就是关于系统测试的介入时机问题。因为咱们的产品还在持续开发阶段,功能不太稳定,对于功能测试还好说,可以在一个功能完成之后同步开始测,但系统测试就不太一样,它主要注重的是产品的整体性能方面。我们以前都是在一个相对稳定和正式的版本上进行系统测试的,进入 Serum 之后,我就一直在思考系统测试开始的最合适的时间,但总是无法找到最适宜的时机,因为这中间不确定的因素太多了。"

"第一,如果在 Sprint 4 开始之初就做,这时候新功能还没完成,功能测试都通不过,更别说系统性能测试了。第二,如果在这个 Sprint 中测前 3 个 Sprint 的功能,那就有可能出现有的问题在当前的 Build 中没有,而在以前的 Build 中才有,如果遇到这种问题就纯属浪费时间了。第三,系统性能测试不像功能测试那么灵活,修改测试用例比较容易,可以随时响应需求和功能的变化,它的前期准备工作比较复杂,不大可能说变就变。" 赵平继续说。

"嗯……确实挺让人头疼的。"关毅低头努力地回想曾经看过的关于敏捷测试的资料:"我也看过一些关于敏捷测试的东西,关于系统测试,比较常见的说法就是最好在第 N+1 个迭代里测第 N 个迭代的功能。但是就咱们目前的情况来说,还不太一样。咱们这已经是最后一个 Sprint 了,除了 Workflow 还差一点之外,其他的功能都差不多完成了,而且它对其它的功能的影响也不大。我觉得可以先开始跑一些 Case,等 Workflow 做完了,我们再出一个完整的 Build 给你们测。按照计划,最后我们还会有 1 个月的时间留给系统测试,这样测试的压力不会太大。"

赵平听完点头说道:"嗯,也好。既然无法兼顾,那就这么办吧。正好可以给以后的测试减轻一些压力,及早发现问题。"

第二天,赵平和李杰就开始着手制定系统测试的计划。他们设计了几个典型的应用场景,还需要覆盖电子商务通支持的所有系统平台,工作量不小。

时间在大家的忙碌中很快过去了。

周三下午 5 点,在每日 Scrum 会议上,吴霞那里又出现了一些问题,她说:"我从前天开始,在原来的老系统上为原有的功能做回归测试,遇到了一些问题。电子商务通的老版本在开发的时候因为不是采用敏捷,也没有测试先行的概念,所以现在代码库里可以自动测试的代码特别少,已经几乎不能用了。我还曾经试图在老版



本上实现自动化测试,但是费了挺大的劲,却没什么进展。"

会后,赵平和吴霞一起找原因,发现她在一些很琐碎的配置任务上花费了大量的时间。这样就算在真正的测试阶段实现了自动化,这些前期的配置也极其浪费时间,很影响效率。

看来不能太急于求成。所以,赵平他们决定一步一步来,先开发一些小脚本、小程序来做那些测试前的准备工作,例如安装产品、调整参数等。完成这一步的工作后,再准备将测试步骤自动化。在自动化测试的工具方面,吴霞决定选用 IBM 的 Rational Function Tester (RFT)。RFT 的基础是针对 Java、.NET 的对象技术和基于 Web 应用程序的录制、回放功能,能为测试人员的活动提供自动化的帮助,例如"数据驱动测试"功能,就可以大大提高脚本的易用性和可维护性。

电子商务通的测试工作在赵平他们的努力下有条不紊地进行着,而且自从周一 关毅与测试人员们沟通之后,大家明显感觉到开发和测试人员之间的沟通和交流变 多了,发现和解决 Bug 的效率也比以前大大提高了。测试组的同事们越来越多,也 越来越深入地参与到开发过程中来。现在,赵平他们在测试之余,空闲的时候就坐 到关毅他们旁边,跟他们结对编程。有时候,他们觉得开发人员在某个功能的写法 上可能会有问题,就特意针对这个功能写几个测试用例,不仅针对性强,也更容易 及早发现问题。

对于这样的变化, 赵平的感觉比关毅更明显。

以前,尤其是没有采用敏捷开发之前,他们往往要等到一个较为正式的版本出来才能开始测试,而在开发人员开发产品的很长一段时间里,他们都只能做一些诸如准备环境、读文档之类的事情,一旦开始测试,测试人员就陷入十分忙碌的状态:跑 case、报 Bug、验 Bug、回归测试……

而现在呢? 赵平他们和开发人员就坐在一起,往往是关毅他们做完一个功能后抬头喊: "赵平, XX 功能我刚跑了一遍没啥问题, 你要不要来试试?" 赵平他们要是试出了 Bug, 关毅他们马上就可以改, 都不用去 Bug 管理系统里费劲地写一大堆文字了。这样, 有些 Bug 一发现就能马上被解决, 大大提高了工作效率。而且这样面对面地把问题说出来, 也降低了大家误解彼此意思的几率。只有遇到特别难解决或者需要花特别长的时间去解决的 Bug, 赵平他们才到 Bug 管理系统里提交 Bug 报

告用来跟踪。

还有些时候,一些 Bug 在开发人员的环境中总是不能重现,因为他们的环境可能不够纯净,影响了运行结果,使 Bug 无法重现。而现在,开发人员可以直接在测试人员的测试环境下重现 Bug,然后跟踪调试,这样也能准确地发现问题并修正 Bug。

这样一来,很多 Bug 在还没被提交到 Bug 管理系统之前就已经解决了,不知道 节省了开发和测试人员多少的时间和精力。这让一开始还不习惯自己的新角色的测 试人员们感叹,敏捷确实有它的过人之处,不仅减少了沟通成本,提高了开发效率, 也让测试团队真正融入到整个 Scrum 团队中来,成为其中不可分割的一部分,和团 队一起快速成长。

总结和思考

性能测试也是系统测试的一种。关于系统测试的介入时机问题,常见的做法是在第 N+1 个迭代测试第 N 个迭代的功能。





临时员工

实习生

Sprint 4 的第三周了,关毅的团队还在有条不紊地推进着项目的进度。

周一早上出门的时候,阳光格外明媚,能见度也特别好,相隔很远都能望见城郊的山脉。关毅心情不错,他想:今天天气这么好,那么远的山都看得见,也许也预示着我们的项目离成功不远了吧。

到了公司, 关毅整理了一下手头的工作, 登录 RTC 检查了一下最新的项目状况。 一切似乎都在掌控之中, 当然, 不可能一切都那么的完美。

让关毅有点担心的还是上个 Sprint 电子商务通 Beta 版发布之后大家的反响。不过因为这才刚刚发布半个月左右,可能还得等待一段时间才能有反馈。但是关毅也知道,上次 BOU 公司提前宣传他们自己的电子商务平台 Beta 版的时候出了些问题,给 BOU 公司和关毅的团队都带来了一些麻烦。所以,关毅也比较关心 BOU 公司目前的情况,他有些急切地想了解最新的状况。不过这些问题的答案,他目前也无从知晓。加拿大那边没有什么最新的消息过来,还是等两天吧,应该快有些消息了。关毅上周末已经给 Greg 发了一封电子邮件询问这个事情,而且他也把上次讨论 BOU公司 Beta 版事件处理方案的一些想法又重复了一下。

在给 Greg 的邮件里,关毅主要重申了让客户及时参与到 Scrum 中的重要性,他认为这是 Agile 所倡导的。不过说归说,做归做,实际上关毅也不知道 BOU 公司那边方不方便,也不是特别清楚究竟应该如何让他们参与进来。根据关毅在 X 公司与客户打交道的经验,一旦客户真的参与进来,估计会有很多预想不到的情况,如果客户提出新的功能需求,也够关毅他们忙活的了。

关毅在心里想着的这些事,确实都不是一下子就能搞清楚的。慢慢来吧,以后有的是机会。他站起来观察了一下大家,每个人都在干手里的活,每个人都很认真、仔细。关毅感到很欣慰,那一丝担心又暂时被他放到了脑后。

每日 Scrum 会议,按照惯例,关毅先开头。他先讲了一下自己手中的事情,然后大家依次轮着说。

大家的进展情况都还不错: Web 2.0 Workflow 的功能已经基本上完成了, 并且能够兼容各种浏览器。

会开到一半,大家正说着,徐天突然走了过来。要知道自从关毅按照 David 的建议,给徐天安排了很多了解项目的途径之后,徐天已经很久没参加每日 Scrum 会议了。关毅看他走过来,心里一紧:不会又有什么乱七八糟的事情吧?

果然,等大家都说完,徐天跟大家说了一件事:"大家都听说过咱们公司的学校常青藤计划吧!说白了就是招一些实习生到公司来做做事,体验体验公司文化,宣传宣传公司形象。目前这个计划已经招了一些学生,我准备分配其中的一个人到你们小组。这个学生我之前参与过面试,技术很好,也有上进心。关毅,你让他学习一下电子商务通的基本知识,参与你们的 Scrum,做一些简单的、容易上手的工作吧,你看怎么样?"

关毅听了,心里想又有新人要来了,很高兴,于是问了一下细节:"他什么时候来啊?实习多久啊?"

徐天告诉他: "后天就来报道了,实习大概两个月。"

关毅有点吃惊:"才两个月啊,来了估计也没什么用。"

徐天说: "先这样吧, 你考虑考虑怎么安排, 有问题到时候再说。"

散会了,胖子显得有点兴奋:又有新人来了,虽然只是临时实习的,不过也是新人新气象啊,呵呵,不错。

关毅却感到有些为难。项目已经进行到收尾阶段了,实习学生来了怎么安排呢? 刚来肯定先要熟悉产品,这个时间就要花去很多,但是他的实习时间又那么短,来 了基本做不了什么事情。而且,让他加入 Serum 团队能行得通吗?他能做些什么呢?

关毅下意识地打开了邮箱,又给 David 写了一封求助信。



David:

我有个小问题。公司有个跟高校合作的常青藤计划,也就是招学生到公司实习,体验公司文化,宣传公司形象,当然也帮忙干点活。这次上头安排了一个学生到我的团队,是经理分配过来的,据说这个学生能力还不错,经理希望能让这个学生参与到我们的 Scrum 开发团队中去,发挥点作用,但是我认为这不是很合适,他刚来也做不了什么,而且他还不能保证全职的时间。

我也不知道该怎么跟经理谈,你帮我分析分析,想想办法吧。

关毅

写完邮件,关毅收拾了一下就回家了。这个星期才刚刚开始,过一阵子工作量 也许会增加。

星期二,关毅收到了经理徐天转发给他的关于那名实习学生的简历:国内名牌大学研二的学生,计算机专业,多次获得奖学金,叫王卡非。名字听着怪怪的,不过看简历写得还不错。关毅对他有了些未见面之前的好感,不过紧接着就收到了David 的回信,他迫不及待地读了起来。



关毅:

你好。

关于实习学生或者兼职员工的话题,看着是个小问题,其实也不能小视它。我想我们可以在这里再讨论讨论。

首先,我们需要明确的是,兼职员工或者实习学生如果不能保证全职的工作,那么他就最起码不可能完全参加每日 Scrum 会议,这是 Scrum 最基本的要求,达不到的话会伤害到整个 Scrum 的过程。

其次,同样因为他不能保证全职的工作时间,Scrum 计划会议以及最终的回顾会议,他都非常有可能缺席,因为这往往是一个 Sprint 的开始和结束,而兼职人员往往觉得这些对他们不重要,这也是对团队不好的因素。

关毅,要知道 Scrum 的原意是打橄榄球的意思。试想一下,一个有时不来训练的队员,或者关键时候找不到他的队员,能够对 Scrum 团队产生很好的作用吗?肯定是不能的。

尤其是对于你说的情况,你的团队目前并不缺少人手,而这种被动的人员的变动是非常不被 Scrum 的理念所接受的。如果你的团队缺少人手的话,我建议你还是寻找全职的员工,如果实在找不到具备合格素质的人才,而具有较高水平的员工不能全职工作的话,我建议你找一个具有基本素质的全职的员工全身心地投入到 Scrum 中来,作为那个具有比较高技术水平的员工的"影子员工",并且让那个具有比较高的技术水平的人才作为技术支持,即幕后人员,而这个幕后人员可以由其他部门来管理。

当然,对于公司上层分配下来的这名学生,你也可以做一些其他的安排, 具体怎么做就看你的团队的需要了。我相信你能处理好的。

David

关毅看完信,很有感触,不过还是有点苦恼。到底应该怎么样安排王卡非这个新人呢?而且这又是经理安排的任务,应该怎么跟徐天沟通呢?他决定将这封信转发给自己的经理徐天,跟他说清楚利害关系,同时,他也开始思考怎么安排好王卡非的问题。

想了半天,关毅决定让他做一些原型开发工作,为销售部门做一些电子商务通的应用案例或者集成的案例,这样也是一举两得,没有浪费人力。想到这里,他很兴奋,抓起电话就给徐天打了过去,把自己的看法以及对王卡非的安排说得很清楚,徐天显得很开明,说:"关毅,我又不是扔给你一个炸弹,你搞得还挺认真。呵呵,就按你说的办吧。以后你跟 David 的讨论,方便的话也给我转发一份吧,我也学习学习,呵呵。"

CMM 与敏捷

打完电话,这个事情算是处理完了。关毅还是没有怠慢手里的编程工作,打开 RTC,大致浏览了一下,继续接着做昨天刚刚开始的 Task。这个 Task 看上去难度不



大, 关毅做得很是胸有成竹。

中午休息的时候,关毅登录了自己的博客。他准备在博客上把今天与 David 讨论的兼职员工问题的过程记录下来,跟大家分享一下。打开博客之后,他发现了一条新的留言,又是一个关于 Scrum 的问题,是一个叫做"软件工匠"的网友的留言。

博客上的留言

Hello ,我看了你之前写的一些关于 Scrum 的知识和实践经验 ,觉得很有用。我有个比较大的问题想问问你:我们公司去年下半年刚刚通过 CMM5 认证 ,对于国内企业来说 ,通过这个认证需要相当的实力和软件管理水平。当然 ,从软件管理上来讲 ,我们采用的是传统的瀑布模式 ,但是目前我们也想在公司的一些部门推广使用敏捷的办法 ,尤其是Scrum。公司现有的规章制度不能改变 ,还是要严格遵守 CMM/CMMI 的一些要求 ,我不是特别清楚 Scrum 与 CMM/CMMI 的关系 ,另外还有 ISO9001的这样的质量标准 ,他们之间有冲突吗 ?

关毅眼前一亮。好问题! 他隐约记得之前查看一些国外的 Scrum 学习组织的材料时好像提到过这个东西。他又临时查了些资料,学习了一下。



资料库

• ISO9001、CMM 和 CMMI

ISO9001 是一个国际标准,即《ISO9001:2000 质量管理体系——要求》。ISO9001 标准起源于制造业,其标准结构、质量体系特点都与制造业非常吻合。随着市场经济的发展和市场竞争的愈加激烈,除了机械、电子、汽车、建筑等传统产业的企业认为 ISO9001 质量体系认证必不可少外,其他行业,如 IT、通信、服务业(酒店、物业管理等)等也在如火如荼地推行 ISO9001 认证。

而 CMM 是由美国卡内基·梅隆大学的软件工程研究所(SEI)开发的软件成熟度模型,美国的软件企业往往选择取得 CMM 等级证书。

在形式上, CMM分为5个等级(第1级级别最低,第5级级别最高),与ISO9000 审核后只有"通过"和"不通过"两个结论相比, CMM是一个动态的过程,企业在取

得低级别的证书后,可根据高级别的要求确定下一步的改进方向。在基本原理方面, ISO9001 和 CMM 都十分关注软件产品质量和过程改进。尤其是在 ISO 9001 标准增加 了持续改进、质量目标的量化等方面的要求后,在基本思路上和 CMM 更加接近。

而所谓的 CMMI 是在 CMM 的基础上相继开发出了系统工程、软件采购等各方面的能力成熟度模型,并整合而成的一种更为复杂、宏大的模型系统,但基本思想和 CMM 是没有很大区别的......

通过一段时间的上网学习后,关毅在博客上谈了自己的看法。



博客留言的回复

你好,你的问题非常好。

根据我的理解,ISO9001 以及 CMM 或者 CMMI,这些都是比较接近或者类似的软件管理模型方法,而这些方法相对于 Scrum 这种敏捷的软件开发管理方式而言,还是有比较大的差别的。比如,CMM 等强调过程的可观测性,Agile 则强调可观测的结果(可运行软件),CMM 等方法比较强调文档,而 Agile 方法却不是那么在乎文档等。

总体而言,CMM/CMMI与 Agile 是两种不同的软件研发管理和过程体系,前者比较重量级,后者更为轻量一些。Agile 包含了更多具体、实用的软件工程技术方法,而 CMM/CMMI提供了更多以数学统计为基础的过程管理和质量控制技术方法。但是这并不是说 Scrum 在一个 CMM 管理的企业中一定会带来冲突。

我认为,合理、恰当的结合运用 Scrum 以及 CMM 的一些管理理念是不会有太大问题的。因为在很多软件过程中,使用 Scrum 能够在一些专门的领域大大减少工作量,比如修改 Bug、软件设计等方面。也就是说,可以在大的、更为宏观的层面上使用 CMM 来控制质量和管理流程,而在一些小团队的具体分工中采取 Scrum 这样的敏捷方法。

所以,从总体上说,在实施层面上Scrum与CMM/CMMI应该是兼容的。

但是从价值观来看,敏捷和CMM/CMMI又是冲突的。CMMI的价值观

维松Scrum之旅

是过程重于人,文档重于可运行的软件。敏捷则正好相反。敏捷是要消除一切阻碍创造价值的浪费。而在 CMMI 的大量关键过程域中,只有很少的东西和开发有关,大多数 CMMI 的实施都会带来很大的浪费。另一方面,CMM/CMMI 不关心团队协作,不关心是否能招到合适的人,更不关心你的代码是否应该重构……

当然,我搜索了一下目前网络上对这个问题的看法,争论很大,况且我所在的公司因为没有采取 CMM 这样的方法来管理软件开发流程,所以我也谈不上实践的经验。但是我想我们可以在以后的时间里更深入地研究、实践这个问题,甚至是一些更为细节的管理问题,比如 CMM 要求的文档的问题 Scrum 团队应该如何应对等。我想应该可以找到一个很好的平衡点。

总之,我们大家一起探索提高吧。

一天就这么过去了。

周三,关毅到公司稍微晚了一点点,发现实习的学生王卡非已经提前到了,看 上去聪明能干的样子,徐天正带着他办一些手续。

很快,手续办好了,徐天把王卡非领到关毅的座位上,挨个给他介绍同事。不一会儿,大家都对这个新人有了好感。关毅则交给王卡非一些材料,主要是关于电子商务通的材料以及 Serum 的学习资料,要求他这一周把这些材料看完,下周正式进入工作状态。

关毅决定新开一个产品集成和应用案例的原型设计的 Scrum 项目,主要就是他和这个王卡非两个人组队。关毅负责一些设计和文档的工作,而主要的代码测试、集成的工作由王卡非来完成。

关毅希望通过这样安排,能做到既不影响原有团队的工作,又不浪费人力资源。 关毅也可以同时领导两个 Serum 项目。两个人也是可以组成一个 Serum 团队的。

总结和思考

Scrum 要求在一个 Sprint 中团队成员高度稳定。



现场客户

周三的工作,由于新人王卡非的加入,整个团队都显得很轻松,大家有说有笑地度过了紧张忙碌的一天。不过,临下班时,关毅却收到了加拿大 Product Owner Greg的来信。

信的内容是关于客户的。前几天关毅向 Greg 询问了一些上次 Beta 版发布之后的 反馈以及让客户参与到 Scrum 中来的事情,这封信算是一个回复。

关毅:

最近实在是非常忙。咱们的 Beta 版本发布之后,我们一些的客户试用之后都觉得比较满意,当然小问题总是有的,比如定制化的问题、国际化的问题等。但是你也知道,上次 BOU 公司用了我们之后发布的版本,由于访问量大,而我们的 API 还不是很强,所以不得不临时增加硬件,推迟了发布日期,这也给客户带来了一些不快。

当然,我们也讨论过了,这也不完全是我们的问题。客户没有参与到Scrum 中来,不能更详细地了解产品功能也是原因之一,所以我觉得你上次提出来让BOU公司参加我们的Scrum是一个不错的决定。上个星期我已经邀请他们了,主要是让BOU的一个代表参与一些前期的计划会议以及最后的评审会议,如果可能的话,他们当然也可以旁听每日Scrum会议。但是BOU公司回复说他们的人手也比较紧,忙不过来。听他们的口气,好像不是很愿意参与进来。

不过没关系的,我想我可以进一步向他们说明利害关系,毕竟这对他们只有好处没有坏处。当然对于我们的开发团队来说,能更直接地了解客户的需求和反馈意见也是非常好的。我会尽快和他们再谈一谈,我想他们最后应该会同意的。

如果确实是由于人手的原因,我想我们甚至可以雇用他们公司的员工, 支付一定的薪金来达到让客户参与的目的。不过这是我们的最后一招了,我 们还是希望客户能够主动地加入进来,这样对大家都有好处。



关毅,以上就是关于客户的一些信息了,我想这些信息应该对你很有用的。我们下次再聊。

Greg

关毅看完邮件,觉得 Greg 做事真是很用心,竟然能想到雇佣客户的员工来让客户参与到 Serum 中来,让他又学到了不少。不过关毅还是有点儿担心,如果让客户参与到 Serum 评审会议中来,那岂不是把很多产品内部的缺陷或者问题都暴露给客户代表了吗?传出去也不好吧!但是如果对客户遮遮掩掩,关毅也认为不妥。左思右想,他还是给 David 又去了封信,问问 David 在这方面的意见。

充实的一天就这样过去了。关毅回到家,又把今天关于客户参与的内容写在了自己的博客上。关毅数了数自己的文章,还不错,已经有十几篇了,而且读者的留言也有一些了。

时间过得飞快。很快就到了周五,这一周的最后一天。大家心情不错,因为过 完今天又是周末了。关毅也想出去转转,轻松轻松。

刚刚上班, 关毅就收到了 David 对客户参与评审会议的意见的邮件。



关毅:

很高兴看到你在 Scrum 的实践上取得的成果。

关于客户参与的内容,我想这部分我可能不能说得太多,毕竟这非常依赖于真实的实践和公司与客户之间的互动关系。

但是我想有些基本原则还是有的,要记住我们为什么而做。我们是为了我们的客户在做事情,客户永远是上帝,任何时候客户都要摆在很重要的位置上来考虑。让他们参与到 Scrum 中来,不代表我们要把所有信息都透露给客户,毕竟有些内容,尤其是技术上的内容,是公司的核心机密,是不能透露出去的。而客户事实上也不关心这个,他们只关注他们能得到什么。所以无论是在 Scrum 计划会议,还是在评审会议的时候,如果客户在场,就一定要让他们有充分了解产品的权利。他们应该知道这个产品的功能,当然我们也应该告诉他们产品目前存在的问题。不隐瞒问题才是对客户的尊重。这样

也就不会出现上次 Beta 版本发布时由于沟通不及时出现的问题了。

其实这些问题你大可不必担心。也许你更需要担心的是如何与客户打交道,沟通产品功能。因为客户的素质往往千差万别,敏捷的软件方法又有如此高的动态性,客户也许会提出很多临时性的、很特殊化的要求。在这些时候,作为团队的领导,你应该分辨出哪些是真正存在的需求,哪些不是。这些也许是以后客户真正来到之后你会遇到的最大的挑战。

放手去干吧,关毅,给自己信心!

David

关毅看完邮件, 很受鼓舞。

总结和思考

无论以什么方式,尽早让客户参与到 Sprint 中来,无疑可以增加成功的砝码。



小插曲: 装修也 Scrum

下班回到家后,关毅就直接把自己埋进了沙发里,紧张了一周的神经终于得到 了些许放松。想着不久即将面市的电子商务通,关毅的嘴角露出了一丝微笑,他对 这几个月辛苦工作的回报感到满意、知足。

渐渐地,关毅的注意力转移到了自己刚刚装修完的新房上。这个花了一个多月的"浩大"工程也终于告一段落了。看着自己的新房,关毅真是感触颇深,自己也算是亲历了一次客户的角色。

故事还得从关毅的新房装修开始说起。

买房结婚是中国人一成不变的传统观念,关毅也不例外地加入了购房大军的行列中,几经"磨难",终于买到了性价比还说得过去的房子。关毅原想终于可以喘口气了,可是,当装修开始后,关毅终于明白了"买房易,装修难"这句至理名言。



装修工程一开始,关毅就对装修工人的工作态度十分不满。根据装修公司的规定,这些工人每天正常的工作时间是早上8点到下午6点,中午有两个小时的吃饭和休息时间。而在真实的执行中,关毅发现,虽然他们都吃住在工地上,但是每天上午9点能正式开始工作就不错了。到了下午5点半左右,工人们就开始聊天或者整理工具了。

几天过后,关毅才了解到问题的根源:装修公司为了保证工程的进度和质量,都会安排"包工头"进行监督和巡视,而这些人的工作时间要比工人们相对灵活一些。不过,"包工头"的工作时间最早不会早过上午9点,最晚不会晚于下午5点。而且,装修公司和工人们的工资结算规则是根据工种按天结算,换句话说,同一个工种的工人们,每天干多干少都会拿到一样的报酬,这就难怪工人们会磨洋工了。

关毅发现这个问题后,当然非常不高兴。他找装修公司的监督人员投诉了几次,但结果并没有像预期那样有所改善,反而,他的做法似乎刺激了这些工人们,他们采取了更加消极的态度来对抗。在工作时间上,工人们的确是严格按照公司的规定,但完全是出工不出力,工程进度还是在不停地滞后。

关毅意识到自己处理事情的方法可能有问题,抱怨和投诉是解决不了问题的。这时候,电子商务通的开发工作已经如火如荼地展开了,关毅对 Scrum 也有了更深的理解。他有时就在想:可不可以把 Scrum 应用到自己家的装修工程中来?除了自己每天下班后都坚持往工地跑,不断提出自己需求,充当现场客户外,怎样才能影响施工团队的效率呢?不过想到要向那些工人们解释什么是 Scrum, Scrum 能带来什么,关毅只能无奈地摇摇头——这太离谱了。

当工期过半时,关毅家的装修工程实际上只完成了计划的不到三分之一。这下 装修公司的工程总监再也坐不住了,整天急得像热锅上的蚂蚁。为了赶进度,他要 求工人们每天至少加1个小时的班,这无疑使工人们的不满情绪进一步加大了。

关毅的心里也开始焦虑起来,虽然他表面上看起来还是那么的从容不迫。他考虑了好几天,终于下定决心要用敏捷的思想干预一下装修团队的管理。如何把 Scrum本土化、"庸俗化",绝对不是一件轻而易举的事,关毅需要一个突破口。

正当关毅发愁的时候,"突破口"自己找上门了。

这一天,关毅一下班就急匆匆地赶到了工地,正巧碰上了工程总监老钱。老钱一照面就试探性地问起了中期款的事,因为在关毅的装修合同上明确写着,工程中期需要再交付给装修公司一部分费用用来购置原材料。因为工程的工期拖得比较厉害,所以都过了一周了,老钱才敢向关毅开口。关毅一听,顿时来了精神,心里很是高兴:正愁着找不到机会呢,老钱居然自己送上门来了。

关毅对老钱说:"老钱,这事好办。但是,我想还是找个地方,咱们好好聊聊吧,你看怎么样?"

老钱听到关毅这么说,点头答应,忙问去哪儿。关毅看了看时间,说:"到吃饭时间了,走,一起吃饭吧。"于是他们在附近找了家饭馆,准备边吃边聊。

刚上第一道菜,关毅就板起脸了。他开门见山地告诉老钱:"老钱,今天你既然说到了要我付中期款的事,咱们就把话说开了吧。你看看你们现在的工程进度!不是我说你们,你们当初向我做的保证能实现吗?你们能按时完工吗?工人们现在的工作态度简直让人失望透顶,他们不是在比谁干得好、干得快,而是在比谁干得更慢!我现在就把款给你们,你们还不定给我搞成什么样呢!"

老钱面如死灰,心里暗想,这下完了,钱是要不到了,但嘴里却不停地道着歉。

关毅看着老钱的表情,觉得火候差不多了,就忍着笑,"语重心长"地对老钱说: "老钱,按理说,你们装修公司的事我是不该管的,但是,现在你们管理的混乱已经 影响到我的利益了,这一点我想你应该是非常清楚的,照着目前的进度干下去,结 局你我心里都很清楚,工程一定会延期的。按照咱们的合同,你们是要支付违约金 的。当然,这是我们都不希望看到的。眼下呢,我作为局外人,对你们在管理上的 问题,还是想多说两句。"

老钱点了根烟,没说什么。

关毅喝了口茶接着说:"这些日子我看出来了,工人们都很怕你,你在和不在差别很大。有好几次你骂他们,可够狠的!"

老钱还挺自豪,笑了笑,说:"那帮小子,就是贱骨头,不好好干,就得骂,这也是对你负责任。"



——敏捷开发故事

关毅摇了摇头:"现在的情况是你只要不在,就都在那里磨洋工,你在的时候恐怕也是装装样子,但你又不可能整天盯在工地上。"

老钱苦笑道:"没办法,每天就那些钱,我还有很多工地要看。其实延期我比谁都着急。我们公司比较小,在管理上确实没什么经验。我也是从木工干上来的,没怎么管过人,你帮我出出主意。"

关毅说:"你首先要在他们当中挑选一个表现好的当'头儿',给他高一点的工资,让他全权负责。另外,还要跟他们讲清楚,延期的罚款会平摊到每个人的头上,那个当'头儿'的罚得更重一些,并且延期的工作没有工资。这样做是狠了点,不过目的就是他们每个人都负起责任来,让他们知道,好好干岂不是更好?"

老钱说:"这方法好是好,不过就怕他们不同意,一生气干到一半就跑了,那可怎么办?"

关毅说:"老钱,我看你是个明白人。你以后少骂他们两句,都是老乡,怪不容易的,以德服人,知道吗?现在房地产、经济都不景气,你要多鼓励他们跟着你一起干才有前途。你自己也是,做大了,活多了,大家一起发财,不好好混怎么行。"

老钱服气地点了点头。

关毅接着说:"对了,你们可以在每天开始工作前把大家集中一下,开个碰头会, 先说一下昨天自己都干了什么,今天打算干什么,有什么困难需要其他人帮助。你 可以不用参加,但一定要让那个'头儿'组织大家讨论一次,几分钟就可以,后有 事情、有问题及时让那个'头儿'跟你讲或者跟我讲。"

看菜吃得差不多了,老钱赶忙暗示中期款的事情。

关毅说:"这样吧,下周我一定给你。这周你先按我的方法试试。我这周工作确实比较忙,工地上还需要你老哥帮我多照看一下了,麻烦你了啊!从下周开始,我可以破例每周付你一次钱,不过在每周开始的时候我们要一起订一下计划,周末我评审通过就把钱给你,否则只能晚一点付给你了。"

"放心,放心。我一定尽力。最近资金周转比较紧,工人的工资是每周一付的,如果你能每周付,那可真是帮我了大忙。"老钱笑着回答道。

一周后,等关毅再次走进工地,好家伙,一切都不一样了。

以往一进大门,最先看到的就是满地乱放的建筑材料,现在所有的材料都井然有序地摆放好了。工人们的干劲可以用空前高涨来形容。虽然只有短短的一周时间,可是整个屋子已经有了翻天覆地地变化,工程进度超出了关毅的想象,这可真让他大吃一惊。

接下来的日子里,每周关毅都会评审一次。虽然中间也发生了很多次的需求更改,不过因为更改发生得早,所以避免了大规模的返工。每天的例会也很有用,工人缺什么、需要什么都早早提地了出来,避免了等待和浪费时间。另外,他们之间互相监督,互相比赛,干劲十足。还有一点也很关键,他们干得好坏和经济利益挂钩了。

最后,工程居然按期完工,并且关毅的这间房子还作为样板间给这家装修公司 揽下了不少客户,连关毅自己都惊讶不已。

总结和思考

这个小插曲仍然说明,在生活中,敏捷可以无处不在,Scrum 其实蕴含着先进的管理思想。





持续集成

经历了一个轻松的周末,关毅他们迎来了第 4 个 Sprint 第四周的工作。尽管之前的工作中有过困难和波折,但是他们坚强地挺了过来,朝着成功继续前进着。

关毅到了公司,刚打开电脑,就看到了 Scrum Master Peter 上周五发出的邮件。

Hi,关毅。

我们这边的高层经理们知道我们的工作进展很快,他们很高兴。同时,他们希望我们能给他们做一次整个产品的演示,时间就定在这周五,所以最近提交进代码库的时候一定要小心,要确保产品的稳定。

谢谢!

Peter

看完邮件,关毅心里也很高兴。如果能在高层面前成功地完成电子商务通的演示,体现出电子商务通的真正价值,那不仅会增加公司对整个产品的信心,也会提高电子商务通在公司整个产品线中的地位,而最直接的好处自然是可以追加更多的预算了。

关毅把这个消息及时地向大家做了"通报",同时千叮咛万嘱咐,最近的 Build 非常重要,一定要在代码做完单元测试以后再提交,一定要万无一失。

大家得到消息后既兴奋又紧张,兴奋的是这表明电子商务通 3.0 的开发工作即将结束了,紧张的是怕出问题。目前电子商务通并没有做持续集成,只是在每天下班前出一个 Build,也就是每日集成,再有就是像现在这样的按需集成。赶在这个紧要关头,万一 Build 坏在谁的头上,谁都担当不起。

就在这时,关毅的手机突然响了。他按下了接听键,一阵急促的声音闯进了他的耳朵:"关先生吗?请您尽快回来一趟好吗?你家的管道可能出问题了,你家楼下

的天花板都开始滴水了。"

这几句话关毅听起来简直犹如晴天霹雳,脑袋里忽地一片空白。在物业的催促声中,关毅恢复了神志:"好的,我知道了,我这就回去!"

关毅放下电话径直走向徐天:"徐天,我家里出了点问题,我必须马上回家一趟, 处理一下。"

徐天看着关毅焦急的神情,赶紧说道:"快回去吧,别着急,路上小心。如果需要帮忙的话就打我的电话。"

关毅感激地点点头,风一般地走了。

离开公司前,关毅也没忘了向胖子他们强调,所有的代码务必要在下班前提交。

第二天一大早, 关毅来到公司后, 算是真正体会到了什么叫做"祸不单行"。

Peter 下班后居然没有回家,而是在"遥远"的温哥华等着他。关毅刚登上了公司内部的即时通信工具,Peter 就找到了他,而且直接要求和关毅通电话,这个举动让关毅有些措手不及。同时,在潜意识里,关毅敏锐地觉察到,应该是有什么事发生了,看情形,这事还绝对小不了。

"Hi,关毅。我有个问题需要你回答。我们今天已经看到你们提交的代码了,只是我们不明白为什么有个基础配置文件会被你们改得面目全非,我们做过的修改全部丢失了,这直接导致了我们无法正确地编译出安装包。我们把这个文件恢复到你们提交前的状态,同样没有办法正确地生成 Build。很快我们就要演示了,这到底是怎么回事啊!"Peter 的情绪很激动。

关毅一时无言以对,他的确不知道究竟发生了什么: "Hi, Peter。非常抱歉,不过我现在还真的不知道问题出在什么地方。你看要不这样吧,我去调查一下,一有消息立刻通知你,好吧?"

Peter 也意识到自己有些失态: "关毅,我对我刚才的态度表示歉意。但是我真的很着急,我现在就在办公司里等着你的消息,不管有什么消息,好的、坏的,都请及时通知我,谢谢了。"

维松Scrum之旅

关毅一口答应了 Peter, 结束了这次通话。

之后,关毅立刻把小胡、小于和小段召集在一起,开门见山地问:"你们昨天提交代码前,有没有先做一下比对?谁提交的代码涉及基础配置?"

小胡他们个人你看看我,我看看你,有些莫名其妙。

小胡先说:"我提交的代码都是产品服务器上没有的,所以根本用不着做比对,就直接提交了。我好像是对类加载的配置文件做了一点修改,不过,我觉得这个应该不会影响到其他模块才对啊。"

关毅听到这里, 脑子里嗡的一声, 几乎要被小胡给气昏了: "你觉得不会?兄弟,这种事情能够觉得吗?现在整个Build都已经Break了!你说吧,怎么办?"

小胡他们从来没见过关毅发这么大的脾气,这次可算是见识了。

咆哮声引来了不少同事诧异的目光,关毅也意识到了自己的失态。终于,理智重新占据了上风,关毅阴沉着脸,尽量压低声音对小胡说:"现在你先把你修改的文件回滚到上一个最新的版本,然后在那个版本上做一下修改,改完后立刻通知我,我会告诉 Peter,让他再做一次 Build,看看是不是能过。听清楚了?"

小胡这时好像还有点委屈,不过看关毅那么凶,哪还敢提出异议。他满口答应, 转身就去工作了。

关毅又环视了一下胖子和小段:"你们两个确定没有动过配置文件?"

胖子和小段头摇得像拨浪鼓。

关毅挥了挥手, 打发他们两个人去工作了。

关毅有点后悔,心里还是觉得不该对他们那么凶。可是,这件事的确是太严重了,如果不给他们点教训,恐怕类似的事情还会发生。

关毅这会儿也顾不得多想了, Peter 还在等着自己, 关毅稳定了一下情绪, 拿起电话拨给 Peter。几声提示音后, 电话接通了: "Hello, Peter。问题找到了, 我们正在全力处理, 不过可能还需要一点点时间, 麻烦你还得等上一会儿。真是不好意思!"

Peter 听到这个消息,长舒了一口气: "OK!关毅,听到这个消息,我真的是很高兴,我会一直在这里等着,有什么消息或者需要我帮忙的话,只管跟我说就行了。"

关毅这时候都不知道该说些什么才好,他沉默了几秒钟才憋出了一句:"Thank you very much!"

小胡忙活了一上午,终于完成了文件的修改。文件提交后,Peter 立即重新做了安装文件,终于成功了! Peter 高高兴兴地回家了。

这件事虽然有了这样一个相对完满的结局,但还是让关毅觉得十分不舒服。他 找来所有人开了一个小会,想继续探个究竟。

小胡这时候才委屈地说:"实际上我在本地环境里都编译通过了啊,谁知道提交后就会有问题?这件事真对不起。不过我们以后应该怎么办?"

关毅点了点头,不好意思地说:"应该说对不起的是我,我不该对你那么凶,是我太急了。责任不能完全怪你,现在我们应该想想办法,至少要解决两个问题。第一是要大大增加构建的频率,这样问题才能尽早暴露,最好做一个持续集成系统,这是敏捷开发里非常重要的一点。第二是要解决本地环境和正式构建系统不一致的问题,很多时候我们在本地开发环境构建时也许是好的,可是一到了正式的构建系统,问题就暴露出来了,这有可能是我们的疏忽,忘记提交某些代码,本地有、构建服务器上没有,也可能真的是本地环境的问题,因为本地环境可能和服务器环境不一致,所以,最好每个人都能到实际的构建服务器上测一测再提交。"

"持续集成的工具能帮助我们做到这两点。"段美女说。

关毅停了一下,接着说:"看来持续集成的工具我们是非上不可了。测试工作的节奏会越来越快,如果新的代码不能及时进行Build,我们还叫什么敏捷团队?"

中午吃饭的时候,关毅和赵平提到了这个问题,赵平咧着嘴嘿嘿一笑,说道: "现在做自动 Build 的工具简直多不胜数,你居然还在这里犯难。告诉你吧,咱们公司内部使用最多的就是 IBM 的 BuildForge,这个你可以到内部的 FTP 上去下载,上面有说明文档,你有空的时候好好研究一下吧。"

有了赵平的提醒,关毅算是找到一扇窗户。他从 FTP 上找到了 BuildForge 的说



明文档,仔细地读了一遍,不由得感叹这东西绝对是现在整个电子商务通团队所需要的。

不过,关毅知道,这件事还是需要和 Peter 商量。



Hi, Peter.

今天真的是非常对不起,让你牺牲了自己的时间来帮助我们解决问题,我为此非常抱歉。

不过,从今天的事情上来看,我对我们的 Build 系统还是有些担心。你看,现在大家如果对代码做了修改,我们只是在本地的开发环境中通过了编译,但难免会出现忘记提交文件或者本地环境和服务器不一致的问题,这样就有可能在正式构建的时候出现问题。我的意见是:我们需要搭建一个自动构建的系统,持续集成,一旦有人提交了代码,就立刻进行构建,如果有 Build Break 就会立刻给当事人发通知。这样一来,如果有问题,我们都可以及时发现。你觉得怎样?

我建议我们可以采用 IBM 公司强大的自动化构建工具 BuildForge,它可以很好地使用我们现有的编译脚本,让所有的流程自动化,而且能够很好地实现我刚才说的那些问题。详细的资料我稍后再发给你,你也可以先 Google一下。

敏捷开发不是强调" Delivering Fast"和" Working Software"吗?我认为,没有一个敏捷的构建系统,就谈不上真正的敏捷。

关毅

阳光总在风雨后。这次突发事件的确给了关毅和 Peter 很大的触动。关毅的提议 让 Peter 感同身受,于是结果都已经在大家的预料之中。

在这周双方的例会上,Peter 作为会议的主席发了言:"在温哥华时间的今天上午,我们给大佬们做了演示,非常非常的成功,所有人都很激动,也都很期待我们产品的发布,他们都认为电子商务通有可能会成为公司一个新的增长点。在这里我要再次感谢大家的努力。我汇总了以往 Sprint 回顾会议的问题列表,在持续集成这

项上,大家的看法很一致,都觉得我们做得很不好。为此,我和咱们的系统架构师以及技术支持人员详细讨论了一下,决定采纳关毅的提议,搭建 BuildForge 环境,实现真正的持续集成。我们会让 IT 部门在一周之内完成这个工作。"

这次例会结束后,在最后的一个多月的时间里,大家都开足了马力,全力以赴,为了产品能"保质保量"的发布而努力。

经过 3 个 Sprint 的洗礼,关毅和他的 Serum 团队终于慢慢走向成熟。这不正是关毅梦寐以求,徐天一直在追寻的吗?

总结和思考

持续集成是敏捷开发中核心的工程实践,它是敏捷产出"可以工作的软件"(Working Software) 的有利保障。





路在前方

关毅带领团队成功完成了这次 Scrum 之旅。关毅不但在全球的敏捷大会上分享了自己团队采用 Scrum 的经验 ,而且也了解到目前全球最新的敏捷开发使用情况。关毅回到北京后 , 开始为下一个版本进行准备。忽然有一天 , 以前的产品总监李鸣找到他 ,和他一起吃饭 ,聊到敏捷开发 ,李鸣来了兴趣 , 关毅则根据自己的理解对敏捷作了一个诠释。

Scrum 之旅大获成功

产品顺利发布

历时5个月,经历了5个迭代,关毅他们的电子商务通3.0终于顺利发布了。

电子商务通 3.0 对外发布那天,公司上下欣喜若狂。大家没想到电子商务通 3.0 的发布会这么顺利,拥有这么多的功能,而且只用了这么短的开发时间,这在过去几乎是不可能的。电子商务通 3.0 比上一个版本增加了一倍的功能量,所有的开发和测试也几乎只用了上一个版本一半的时间。更神奇的是关毅他们越到后面越轻松,越不用加班。关毅想起自己在学生时代老师常说的一句话:"不考不玩,小考小玩,大考大玩"。身为好学生的他深谙此道。平时抓紧了,哪用临时抱佛脚呢?

不但如此, 敏捷开发也让关毅他们在整个 E 公司中国研发中心出了名, 别的团队纷纷要关毅过去给他们普及敏捷开发的知识, 分享敏捷开发的经验。

去敏捷技术大会演讲

让人兴奋的是,关毅他们投给国际上一年一度的最知名的敏捷技术大会的稿件 被采纳了,关毅随即将前往芝加哥进行演讲。

在敏捷技术大会上,关毅用流利的英语讲述了他们产品组采用敏捷开发的经验 和好处。

关毅的演讲

敏捷开发的核心价值观是,软件开发最重要的是给用户提供有价值的、可以工作的软件。如何保证提供有价值的软件,

是通过反馈机制来完成的。这一点,我们体会很深。自从采用敏捷开发以后,我们比以前更有意识地希望得到各种反馈,包括来自外部和内部的。我们产品的大部分功能都直接来自客户的需求,并按优先级排序。我们有 Beta 项目,



在开发中期就给客户试用并得到反馈。并且我们在公司的内部网上也部署了 最新的版本,并不断更新,得到了大量的用户反馈。

可以工作的软件,含义就是软件是可交付给客户使用的。我们每4周一个Sprint,即迭代。在迭代结束的时候,就会产生一个可以交付给客户使用的版本,这个版本里包含所有新增功能的实现,并且通过所有的测试。这样带来的好处就是可以大大缩短软件开发周期,提高软件质量。并且在临近发布的后期,我们也没有出现特别紧张的现象。每个Sprint即将结束时的Sprint评审会议可以帮助我们关注可以工作的软件,并能及时得到反馈。

每日 Scrum 会议,我们觉得是非常有效的。团队所有成员都通过这个会议同步信息,每天的问题都能及时发现,并能用最快的速度解决。另外,能够形成一种相互之间的压力,每个人要对自己当天承诺要完成的事情负责,每天都不敢怠慢。过去我们通常只是一周开一次例会,汇报一下状态,效率就低多了。Daily Scrum 还给整个团队带来了凝聚力,每个人都觉得自己是团队不可缺少的一员。

采用敏捷开发,对于分布式开发的团队来说,沟通也大大加强了。我们就是这种情况。我们有北京和加拿大两个研发团队,还有分散在世界各地的商业合作伙伴协助开发。采用敏捷促使我们不再过分依赖于文档进行交流,我们主要通过电话、视频,甚至现场进行沟通。另外,我们使用了一些 Scrum管理工具,如 ScrumWorks 和 Rational Team Concert 这些可视化的工具,可以清楚地看到每一个远程团队任何时候的状态,比如 Sprint Backlog、Task 的分工及完成情况以及燃烧曲线的情况等,这些都大大提高了沟通的效率和效果。即使是本地的团队,沟通也比以前更有效。比如我们特意安排整个团队都坐在一起,大家的交流变得非常方便。我们有意识地简化文档,而更多采用面对面的交流。交流多了,工作也变得更有趣了。

我们采用敏捷开发,测试很早就介入了,这样就再也不会出现半年多测试部门无事可做,一旦介入便疯狂加班的情况。开发人员也不会由于要更改很久以前的 Bug,而把很多时间和精力花费在回忆上。而且一旦发现重大的问题,也不会出现疯狂加班的情况。由于测试的提早介入,开发人员可以立即解决问题,还能够有效地调动测试人员的积极性和主动性,让他们也参与

路在前方

到软件的设计中来。在敏捷开发里,测试和开发的交流更多了。测试和开发不再是两个孤立的部门,而是真正意义上的团队,这有效避免了过去测试和开发之间一些消极的争斗和不合作的局面。

在文档管理方面,我们摒弃了过去烦冗的文档管理,节省了大量的时间和精力。需求、计划以及设计文档我们通常只保留一个一页多的版本,并且放在 Wiki 这样的敏捷文档系统上,任何人、任何时候都可以更新它。而且,由于文档只记录必要的信息,所以不需要再像过去那样花大量时间保持同步,不断修改。

由于采用敏捷开发,测试人员的积极性大大提高。过去测试人员一直处于被动的角色,而现在,在开发中就需要测试人员的大量参与,参与需求、参与设计。测试人员对产品的影响力大大增强,也更有利于他们的个人成长。

自我管理的团队是敏捷里的一个比较"酷"的理念,我们也实施得很好。我们的老板也由一开始事无巨细地干预到最后完全变成了教练的角色。老板轻松了,从琐碎的项目管理中解放出来,把精力放到了更重要的带领我们进行技术创新和帮助每个人更快地成长上。我们每个团队成员也更放松,没有人指指点点,自己有更多的决策权,可以调动更大的潜能。每个人在召开Spring 计划会议的时候可以领取自己最感兴趣的任务。事实上,真正运行良好并自我管理的团队,效率远远高于过去习惯被动地接受命令的团队。

Scrum 回顾会议也帮助我们不断地提高。我们采用了一个叫做"Team Pulse Survey"的模版作为我们 Retrospective 的框架。在每一个迭代结束后,Retrospective 框架都帮助我们总结哪些地方做得不好,哪些地方做得好。当总结成为一种思维习惯的时候,想不提高都难。

现在都讲可持续发展,软件开发也一样。如果加班可以有效果,那也只限于一个短的时间范围内。但是任何公司、任何软件产品都不希望只是短命的。所以,我们必须有条不紊、有张有弛,决不提倡加班。我们也曾经加过班,但是发现效果并不好,出现代码质量下降、人员士气低落、精力无法集中等问题,这些都是得不偿失的。后来我们没有再加班,而且由于代码质量高,人员士气也跟着高涨,总体效率当然会比以前有很大的提高。

由于每个 Backlog 都有预设的 Story Point ,因此过去每个迭代一共做多少个 Story Point 就非常清楚。我们可以通过以往 Sprint 的速率来计划当前的 Sprint ,这些数据会帮助我们制定当前 Sprint 需要完成的工作量 ,即放入多少个 Story。

敏捷开发的工具,我们用了很多。Scrum 项目管理工具,向大家推荐 IBM Rational Team Concert 和 ScrumWorks。这两种工具都可以帮助我们有效地管理 Scrum 流程,尤其适合地理位置不在一起的多个团队使用,能够大大增强敏捷流程的可视化。在分布式开发的情况下,如果没有好的管理工具,很难想象会如何敏捷起来。至于敏捷的工程实践、管理 Build 方面,我们采用了IBM Rational BuildForge 这个强大的持续集成工具。另外,我们还用了结对编程工具 ECF、自动重构和自动单元测试工具 JUnit 等。

下面向大家推荐一个我们实施 Scrum 这种敏捷方法的清单列表。

首先是拥有一个具有优先级顺序的 Product Backlog ,每个 Backlog 用 User Story 来进行描述,制定它的优先级和预估完成工作量。

其次是确定团队构成,确定团队是地理分布的还是在同一个位置,最好不要把地理位置不同的一群人算作一个团队。团队成员由开发、测试、文档人员组成,选出一个 Scrum Master,每个团队人数最好是 3~10 人。

接下来,确定每个 Sprint 的周期。根据项目情况的不同,两周到两个月都是可以的,但最好不要超过两个月。

然后,召开Sprint 计划会议,把每个Sprint 的要做的工作从Product Backlog中按照优先级顺序排序,确定需求,将任务进一步分解,如人员分工等。

无论是否采用敏捷或者 Scrum,从现在开始就可以尝试每日 Scrum会议,每天选择固定的时间和固定的地点,15 分钟就可以了。如果选在下班前开这个会,那么每个人陈述的 3 个问题就是:今天我做了什么?明天我打算做什么?我遇到了哪些困难?

应该从 Sprint 的第 1 天就开始准备测试用例。当 Sprint 结束时,不但功能完成,测试也将同时结束。

路在前方

在每个 Sprint 结束的时候召开 Sprint 评审会议和 Sprint 回顾会议。

根据团队情况,不断采用敏捷的工程实践,比如 Unit Test、Code Review、结对编程、持续集成等。

多学习别人的经验。网上有很多资源,还有很多书籍。

Agile 并不是银弹,它不可能解决你所有的问题。Agile 本身并不是一种软件开发流程,而是一种理念,一组行为方法。只有真正理解了 Agile 的精髓,才能真正 Agile。

You don't do Agile, you are Agile.

关毅讲完后,台下响起了热烈掌声。接下来是问答环节。

对关毅演讲的反馈



"谢谢你的演讲。听你讲了很多采用 Scrum 的好处,能不能再简要概括一下,如何才能非常清楚地知道一个团队采用敏

捷是否成功了?"

"采用敏捷之后,软件最终的销售利润是否有所提高?软件的质量是否有 所提高?每个人对工作的满意度是否有所提高?"

.

全球敏捷使用调查

提问环节结束后,散会了。可是依然有不少人把关毅围住,提了不少的问题, 弄得他好不容易才脱身。

关毅急匆匆地赶往下一个会场,不是去讲,而是去听。

这个会场的主题是"敏捷在全球的使用调查"。关毅非常想了解敏捷在世界范围内的使用情况。

维松Scrum之旅

——敏捷开发故事

演讲开始了, 主讲人打开 PPT, 调出了某著名调查公司的数据。



资料库

• 全球敏捷使用调查

不同规模的敏捷项目的成功比率如图 8-1 所示,深色代表项目成功项目的数量,浅色代表尝试敏捷项目的数量,纵坐标是团队的规模(人数),横坐标是参加统计的团队数量。可以看出,团队人数越少,成功的概率越大,但对于超过200人的团队,采用敏捷反而容易成功。

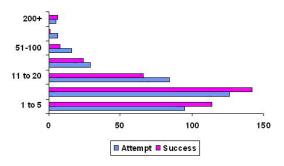


图 8-1 敏捷开发成功概率调查结果图 (数据来源: Dr Dobb's 2008 Agile Adoption Survey)

不同形式的 Agile 项目的成功比率如图 8-2 所示。参加调查的有 214 个本地 (Co-Located) 项目,项目成功率为 83%,看来团队如果在同一个地点,采用敏捷会有 非常大的优势。210 个异地协同(Not Co-Located)项目的成功率为 72%,说明异地 敏捷实施的难度要大一些。129 个外包(Offshoring)项目的成功率为 60%。

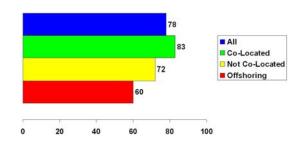


图 8-2 不同类型的敏捷团队成功率 (数据来源: Dr Dobb's 2008 Agile Adoption Survey)

路在前方

图 8-3 也是通过调查大量团队得出的统计结果。在生产率方面,有 60% 的团队认为有所提高,有 22% 认为大大提高,有 13% 认为没有变化,有 4% 认为有所下降,有 1% 认为大大下降。在软件质量方面,有 48% 的团队认为有所提高,有 29% 认为大大提高,有 14% 认为没有变化,有 6% 认为有所下降,有 3% 认为大大下降。在商业利益相关者满意度方面,有 47% 的团队认为有所提高,有 31% 认为大大提高,有 15% 认为没有变化,有 4% 认为有所下降,有 3% 认为大大下降。在系统开发成本方面,有 32% 的团队认为有所下降,有 5% 认为大大下降,有 40% 认为没有变化,有 18% 认为有所提高,有 5% 认为大大提高。

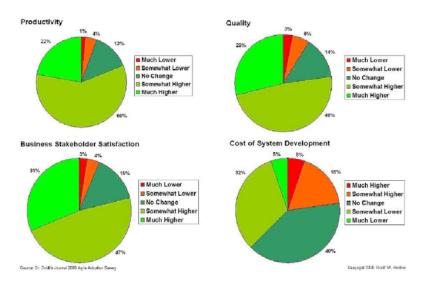


图 8-3 敏捷团队民意调查 (数据来源: Dr Dobb's 2008 Agile Adoption Survey)

为什么要采用敏捷开发?因为它确实有效!从以上方方面面的统计来看,大部分项目采用敏捷后都取得了成功。敏捷开发有效地提高了生产率,改善了软件质量,增强了利益相关者的满意度,在开发成本方面,也仅有23%的团队认为成本提高了。

接下来,关毅听了很多关于敏捷开发工程实践和开发工具的演讲和报告,还有一些其他公司使用敏捷的经验。

关毅此行收获不小,回国以后便忙着和所有同事分享这些精彩的内容。另外,他也想好了要如何在即将到来的下一个版本中改进以往的问题,继续使用 Serum,使用敏捷。



对敏捷的诠释

一年后的一天, 关毅接到一个电话, 居然是原来 X 公司的产品经理李鸣。

寒暄了几句之后,李鸣问关毅有没有空一起出来吃个饭。关毅答应了,不过心 里直犯嘀咕,过去李鸣是自己的高层主管,私下里打交道并不多,这一年多也没什 么联系了,估计这次是有事情。

一个阳光明媚的周六,在五道口的一家装修很有格调的餐厅里,关毅坐在李鸣的对面。

李鸣没有抽烟,喝了口茶,问关毅:"这么长时间没见了,你现在在 E 公司还好吗?"

关毅说:"我很好, 你呢?"

李鸣淡淡地说:"我从 X 公司辞职了。F4 失败了,我现在自己出来开公司了。公司正处在筹备阶段,融资、招人……"

接下来,李鸣详细介绍了自己软件公司的想法,说得很有激情。关毅也觉得很好,时不时地提出自己对公司管理的一些看法,并向李鸣介绍一些敏捷开发思想。

李鸣显然是来了兴趣,于是问关毅:"敏捷开发最近好像业内宣传的挺多,似乎能解决很多传统软件开发不能解决的问题。能不能概括地给我讲讲什么是敏捷?刚才听你讲得挺好。E公司的开发流程和开发方法,我们得好好学学。"

关毅笑了笑:"敏捷不是灵丹妙药,不能包治百病。敏捷也不仅仅是一堆软件开发方法和开发流程,它的本质是一种哲学思想,是一种做事情的方法论。我的体会,敏捷之道就在两点,以价值为本,以人为本。价值为本,要时时刻刻想着客户,想办法创造真正对客户有价值的软件。当做到了能更快速地提供价值的时候,就是敏捷了。帮助客户成功,你也就赢了。以人为本,说到底,价值还是由人来创造。一家新公司或者一个新团队的关键是人才,有了人一切都好办,对人才要好好珍惜。当人的能动性超越了一切死的流程,就是敏捷了。记得有一句话是这么说的,'You

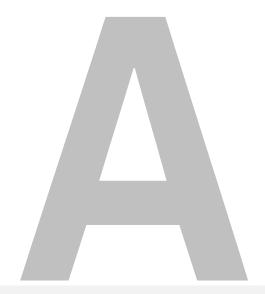
don't do Agile, You are Agile!'"

李鸣不住地点头,犹豫了一会儿,他对关毅说:"关毅,过去在 X 公司,你的很多建议没有被采纳,有我的不对,也有很多公司比较为难的地方。X 公司的问题,其中很重要的一点,现在看来就是太不敏捷了。是这样的,我今天和你一起吃饭就是想请你加入我的公司,我会给你股份,你可以尽情发挥,我们一起努力把它做大。"

刚才听李鸣"忽悠"了半天,关毅的内心也在激荡。

沉思了好一会儿,关毅想清楚了,他的人生理想就是使自己的价值最大化,所以他知道自己现在还不能离开 E 公司。在 E 公司,他可以更好地发挥自己。全球化的平台、明星级别的产品、几百万的企业用户、无比可爱的团队、工作和生活的平衡——这些加在一起对目前的关毅更有吸引力。

关毅抬起头,举起了杯子,对李鸣说:"李总,谢谢你。不过我的敏捷开发 Serum 之旅还没有结束,我在 E 公司还有很多事情要做。不过,还是欢迎你随时找我,我们一起交流。"



附录 A 案例索引



第 4 章 Sprint 1 ——新手上路

- 1. 当开始研发新产品或者已有产品的新模块时,由于各方面的原因,整个团队没有能力在 Sprint 的开始就做出一份非常详实的计划,因此,采用"照明弹"策略绝对不失为一个好办法。
 - 2. 对于每一个 Story, 要尽可能了解它的需求。
- 3. 在开发过程中,为了提高交流的效率,要尽量避免把精力浪费在不必要的文档上,取而代之的是要提倡团队之间面对面的直接交流。
- 4. 在实际工作中, Serum 提倡团队自我管理, 在任务分配时每个人都可以按自己的兴趣来选择任务。
 - 5. 团队成员的技能培训是要在做 Sprint 计划时就考虑在内的。
- 6. 虽然每日 Serum 会议的持续时间会根据整个团队的大小而有所不同,但是,请不要超过 15 分钟。
 - 7. 经理应该充分信任开发团队,不该把每日 Scrum 会议当成每日汇报。
- 8. 在 Serum 工具的使用上,一定要确保每天都进行准确的更新,只有这样才能 让团队的其他成员以及项目经理掌握及时、准确的项目进度信息。
 - 9. 通过 Burndown Chart, Scrum 将给项目带来更多的可视性。
 - 10. 每日 Scrum 会议举行的时间应该在早晨还是下午?
- 11. 在每日 Serum 会议上不要过多地讨论技术难题的细节,如果有团队成员遇到无法解决的困难时,Serum Master 应该将其记录下来,会后再调配资源去解决。
 - 12. Scrum 如何解决开发与测试工作同步的问题?
- 13. 每个 Sprint 结束后,最基本的目标是得到一个可以工作的产品。Sprint 结束时的 Sprint 评审会议和 Sprint 回顾会议都至关重要。

案例索引

第5章 Sprint 2——计划与变化

- 14. 在 Sprint 计划会议中, Scrum Master 应该主动与 Product Owner 一起制定和 讨论这个 Sprint 的工作。
- 15. 在 Sprint 计划会议中, Scrum Master 不应该抛弃团队成员, 团队成员必须全体参与到计划的讨论中去, 并且及时对不明白的地方以及用户需求等进行提问。
 - 16. 采用 Wiki 进行文档管理。
 - 17. 在 Sprint 计划会议中应该如何制定具体的计划?
 - 18. 在 Sprint 进行过程中,遇到临时的员工变化,比如请假,应该怎么办?
- 19. 在 Sprint 计划会议中,应该如何更准确地估计每个 Task 的时间?应该怎样用计划扑克来估计时间?
- 20. 在 Serum 中,例如调试机器、准备开发环境这样的任务的工作量也应该体现在 Sprint 的工作分配中。
 - 21. 关于 Serum 团队大小的讨论。
 - 22. Scrum 与 XP 以及其他一些敏捷方法的关系是怎样的?
 - 23. 每日 Scrum 会议为什么需要团队成员站着开,而不能坐着?
- 24. 每日 Serum 会议需要每天进行,而不能因为每天的任务相似就不召开。Serum Master 应该在每日会议中敏锐地发现问题,并积极地鼓励团队成员进行讨论。
- 25. 作为一个 Scrum Master,对于自己上司临时安排的非 Sprint 计划的任务,应该尽量提前在 Sprint 的计划阶段就留出一定的缓冲时间用于处理这些事情,并应该尽量协调安排好,避免过多非 Sprint 计划的任务出现。
 - 26. 作为一个 Scrum Master, 应该如何有效地向经理汇报项目进程?
 - 27. Scrum 不鼓励加班。



28. 对于不好演示的功能,可以用运行测试用例等其他方式在 Sprint 评审会议中进行展示。

第6章 Sprint 3——深入 Scrum

- 29. 创造一个敏捷的工作环境, 让每个 Serum 团队成员都坐在一起工作。
- 30. 敏捷开发的思想之一也包括避免不必要的浪费, 在做 Sprint 计划时应该放弃实现一些不必要的功能。
 - 31. 如何制定一个 Sprint 的目标?
 - 32. Sprint 计划会议需要团队成员事先进行很充分的准备。
 - 33. 测试应如何介入 Sprint 中?
 - 34. 尝试"结对编程"。
- 35. 在敏捷开发中,应该尽可能地寻找有效的工具提高效率。"工欲善其事,必 先利其器。"
- 36. 对于异地团队,增强沟通是最关键的。初期最好的沟通方式是面对面的交流。核心人员最好能进行一些面对面的接触。
- 37. Scrum Master 的职责之一是在完成日常工作的同时,还需要随时帮助团队成员排除障碍。
- 38. 测试人员应该融入 Serum 团队,这样做有利于开发团队与测试团队之间的沟通,以共同保证产品的质量。
- 39. Serum 强调团队协作,并且在业绩评定上一直都是以整个团队的成果来衡量的,这似乎对团队中的某些成员不太公平,毕竟术业有专攻。这种情况下应该怎样衡量每个人的绩效价值呢?
- 40. 由于团队成员的每个个体在技能、精力、经验上的差别,所以必然会导致效率的差别。那么,应该如何消除这种差别带来的问题?

41. 对于项目进行中有可能影响项目结果和交付日期的突发事件,绝不能隐瞒,而是应当从客户的立场出发,告诉客户实情,并在现有的条件下尽可能满足客户的需求,同时为客户提供多个解决方案作为备选。

第8章 Scrum 4——最后的冲刺

- 42. 在一个 Sprint 结束和新一个 Sprint 开始的中间,应该有一定的缓冲时间。
- 43. 采用 Serum 回顾模板"团队听诊工具"进行 Serum 回顾。
- 44. 利用演示工具有效地传递 Sprint 需求。
- 45. 采用 Scrum 后的新部门组织结构。
- 46. 推荐敏捷工具 IBM Rational Team Concert。
- 47. 如果测试人员不习惯自身的角色转换,应该鼓励他们转变思维方式,主动 参与开发过程。
- 48. 关于性能测试的介入时机问题,常见的做法是在第 N+1 个迭代测试第 N 个 迭代的功能。
 - 49. 测试新功能前要对老功能做回归测试,以保证新功能不会破坏老功能。
 - 50. 关于自动化测试,应该尽力而为,但不能急于求成。
- 51. 一个产品的成功与否需要市场来检验。在产品正式发布之前,如果能有客户尽早地参与到开发过程中进来,无疑会增加成功的砝码。
- 52. Serum 强调在一个 Sprint 中团队的稳定,一个 Sprint 中间的人员变动会对 Sprint 不利,即使加进来足够多的人往往也不能起到提高效率的作用。
 - 53. 如何管理素质较低的团队?
 - 54. 传统的软件管理体系 CMM/CMMI 的理念与 Scrum 的关系。
 - 55. 持续集成的实践。



附录 B 相关概念

Agile

敏捷开发。

Backlog

一项工作。

Build

指已经编译、构建好的一个可运行的软件版本。

Burndown Chart

用来显示当前还剩下多少工作未完成的图形化工具。通常以时间为横轴,以本 次迭代要完成的工作为纵轴。

Code Review

代码审核,通常由非代码编写者完成。

Daily Scrum Meeting

每日 Scrum 会议。每天 15 分钟的每日例会,团队中的每个成员都要回答以下 3 个问题:上次例会到现在我完成了哪些工作?下次例会前我将完成哪些工作?有没有什么事情阻碍了我的工作?

In Progress

进行中。

Product Backlog

产品功能特性列表,主要由产品责任人负责维护并定义优先级。

Product Backlog Item

产品功能特性列表中的条目,每个条目就是一个工作单元,其大小必须限制在

维松Scrum之旅

团队可以在一个迭代之内完成。一个工作单元可以被分解成多个任务。

Product Owner

产品责任人,负责确定 Backlog 中各条目的优先级,同时解决所有关于需求的问题。

Safari

苹果操作系统上的浏览器。

Scrum

Scrum 一词来自英式橄榄球,它把软件开发团队比作橄榄球队。Scrum 是当今流行的敏捷开发方法之一。

Scrum Master

负责管理每日 Scrum 流程的人,是 Product Owner 和 Team 之间的桥梁,要推动双方的合作,负责为 Team 成员解决障碍和问题,保证他们工作的顺利进行。Scrum Master 相当于传统软件开发项目中的项目经理或主管。

Sprint

Sprint 代表 Scrum 的一次迭代,周期通常是 30 天,期间不能给 Team 增加额外的需求,以确保迭代结束时能够获得预期的结果。

Sprint Planning Meeting

Sprint 计划会议,在一次迭代开始时召开,由 Team 与 Product Owner 一起商讨本次迭代的目标,决定本次迭代要完成哪些工作。

Sprint Review Meeting

Sprint 评审会议,在一次迭代结束时召开,一般以 Demo 的形式由 Team 展示这个迭代中完成的功能。

Sprint Retrospective Meeting

Sprint 回顾会议,在 Sprint 评审会议之后召开,由 Team 与 Scrum Master 共同讨论这个迭代中哪些地方做得比较好,哪些地方需要改进,使团队持续成长。

Stakeholders

利益相关者,是项目成败对他们影响不大的一类人,他们参与提出产品的需求 并积极提出反馈意见。

Task

任务。

Team

跨功能的 Serum 团队,人数限制在 3~10 人,可能包括的角色有开发人员、架构师、测试人员、UI 设计师等,是一个自组织的团队,由团队成员自己决定如何更好地满足用户需求,并承担相应的责任。

User Story

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。

Welcome Lunch

欢迎午宴。

Wiki

维基百科,一种开放和共享的在线文档编辑系统,任何人都可以在这个系统中 编辑和修改文档,最早的应用是在线的开放式百科全书,现在广泛应用于各种文档 系统。



附录 C 参考资料

- Agile Software Development with Scrum [Pearson Education, Ken Schwaber/ Mike Beedle, 2004].
- [2] Scrum 敏捷项目管理, Ken Schwaber 著, 李国彪译, 清华大学出版社, 2007.
- [3] Scrum and XP from the Trenches, [Hendrik Kniberg, 2007, C4Media Inc. online edition, http://www.infoq.com/minibooks/scrum-xp-from-the-trenches].
- [4] Scrum Checklists, 2006, [online book: http://www.infoq.com/minibooks/scrum-checklists].
- [5] 测试驱动开发(中文版), Kent Beck 著, 孙平平等译, 中国电力出版社, 2004.
- [6] 敏捷估计与规划, Mike Cohn 著, 宋锐译, 清华大学出版社, 2007.
- [7] 解析极限编程:拥抱变化, Kent Beck 著, 唐东铭译, 人民邮电出版社, 2002.
- [8] 重构——改善既有代码的设计, Martin Fowler 著, 侯捷等译, 中国电力出版社, 2003.
- [9] http://www.scrumalliance.org.
- [10] http://www.implementingscrum.com/.
- [11] Scrum and CMMI: A High level assessment of compatibility, http://www.cioindex.com/nm/articlefiles/71934–ScrumCMMI.pdf.
- [12] What is the difference between Scrum and RUP? http://www.scrum-breakfast.com/2008/02/what-is-difference-between-scrum-and.html.
- [13] Applying scrum and other agile development techniques, http://www.computerweekly. com/Articles/2007/09/04/226507/applying-scrum-and-other-agile-development -techniques.htm
- [14] 极限编程网站, http://xp123.com.
- [15] ScrumWorks 产品官方网站, http://danube.com/scrumworks.
- [16] xPlanner 产品官方网站, http://xplanner.org/.



- [17] Jazz 产品官方网站, https://jazz.net.
- [18] Advantages of User Stories for Requirements, http://www.mountaingoatsoftware.com/article_view/27-advantages-of-user-stories-for-requirements.
- [19] http://www.controlchaos.com/.
- [20] RUP in the dialogue with Scrum, http://www.ibm.com/developerworks/rational/library/feb05/krebs/index.html?S_TACT=105AGX52&S_CMP=cn-a-r.
- [21] Eclipse Communication Framework, http://www.eclipse.org/ecf.
- [22] William Krebs1, Laurie Williams2, Theresa Kratschmer1, Per Kroll1, Ed Richard1, Kay Johnson1, "Introducing the Deployment Evaluation Framework v2.9", 2007 IBM Academy of Technology Conference on Agile.
- [23] Dr Dobb's 2008 Agile Adoption Survey, http://www.drdobbs.com/.



附录 D 免责声明



本书不带有任何明示或暗含的保证。本书提供的建议或最佳实践只作为一般的 经验分享,只在作者特定的环境下验证过。作者不保证这些建议或最佳实践在任何 情况下都有效。本书的内容有可能不太准确或包含错误,作者对此深表歉意。本书 中任何带有主观性的陈述都只代表本书作者的观点,不代表 IBM 公司的官方立场。

本书为了内容介绍的方便,提到了一些情节、测试数据、人名、网站名称或公司名称。这些情节、人名、网站名称或公司名称纯属虚构,如果与任何实际的情景和名称相同,则纯属巧合。