

Scrum 中软件缺陷管理方法的研究与应用

文俊浩^{a,b}, 田 清^a, 李 朋^b

(重庆大学 a. 软件工程学院; b. 计算机学院, 重庆 400044)

摘 要: 针对某公司互联网产品的开发项目, 分析基于 Scrum 的软件敏捷开发过程, 给出其缺陷管理目标, 即敏捷地响应缺陷和实时地改进开发过程, 设计相应的缺陷管理方法, 包括缺陷管理流程模型以及缺陷度量方法。在此基础上, 开发缺陷管理工具 RQMS 并加以应用。应用结果表明, 该缺陷管理方法在基于 Scrum 软件敏捷开发中具有较好的效果。

关键词: 软件缺陷管理; Scrum 开发方法; 敏捷开发; 缺陷管理流程; 缺陷度量

Study and Application of Software Defect Management Method in Scrum

WEN Jun-hao^{a,b}, TIAN Qing^a, LI Peng^b

(a. College of Software Engineering; b. College of Computer Science, Chongqing University, Chongqing 400044, China)

【Abstract】 According to Scrum software development process and the Internet product development practice in a company, this paper analyzes the goals of software defect management of Scrum agile development, that is, the agile team can response the defects and optimize the development process very quickly and suitability. Then it puts forward how to manage the software defects by two ways, one is designing the defect management model, the other is measuring the software defects. It develops a software management tool called RQMS. By using RQMS and the practical application data, it proves that the management methods are quit efficient.

【Key words】 software defect management; Scrum development method; agile development; defect management process; defect measurement

DOI: 10.3969/j.issn.1000-3428.2011.19.010

1 概述

软件缺陷管理已成为软件工程领域的重要课题。互联网用户需求不断变化, 使敏捷开发成为互联网产品的主要开发方法。

软件敏捷开发伴随着“高度迭代、周期性、持续响应客户频繁反馈”的特点, 决定了它的缺陷管理与传统缺陷管理存在着差异。但目前还未专门针对软件敏捷开发进行过缺陷管理方法的研究。因此, 各种软件缺陷管理工具如 Mantis^[1]、Bugzilla, 以及 IBM 公司的 Rational ClearQuest 等也都未曾考虑敏捷开发的缺陷管理。本文对 Scrum 软件敏捷开发中的缺陷管理方法进行了研究和工具实现。

2 Scrum 软件敏捷开发过程

Scrum 是敏捷开发框架之一^[2], 其定义了产品拥有者、利益相关者、Scrum 专家以及开发团队 4 种角色。依据公司产品开发过程, 敏捷团队由 1 名产品经理、1 名项目经理、1 名 UI 设计师、3 名开发人员以及 1 名测试人员组成约 7 人组成。产品经理即是产品拥有者, 产品项目经理、开发人员、UI 设计师、测试人员组成开发团队。而互联网产品的利益相关者是互联网用户, Scrum 专家是公司多年来积累的最佳实践。因此, 在公司并不设置利益相关者和 Scrum 专家。

Scrum 将开发过程分为多个 Sprint 周期, 每个周期时长约 2 周~4 周。产品需求被分成不同的产品需求积压条目, 在 Sprint 计划会议上, 最重要产品需求被优先安排到下一个 Sprint 中, 并在对现阶段需要实现的需求进行设计和任务分配。在每个 Sprint 中, 每天均有一个 15 min~20 min 的站会。每个 Sprint 周期结束时, 将一个可使用的系统交付给用户。用户根据使用情况提出意见, 意见又将作为需求的形式保留。

3 Scrum 软件敏捷开发中的缺陷管理方法

在传统软件开发中, 缺陷管理的目标是确保每个被发现的缺陷都能够被解决, 并收集缺陷数据, 以识别测试过程的阶段或作为过程财富。敏捷开发的缺陷管理目标与传统缺陷管理相比, 有以下 2 个要点: (1)必须符合“敏捷”特点, 即能保证敏捷团队能快速响应缺陷; (2)缺陷数据能实时对软件开发过程的改进进行指导。

为实现敏捷开发缺陷管理的目标, 提出了以下方案: (1)对软件敏捷开发中的缺陷管理流程进行建模, 即通过分析敏捷的缺陷管理流程, 减少缺陷的冗余信息, 使团队成员快速地响应缺陷, 便捷地对缺陷进行追踪管理; (2)对软件缺陷建立度量模型, 即通过分析敏捷开发中各角色所需的信息产品, 实时地指导敏捷团队的工作。

3.1 软件缺陷管理流程模型

基于“敏捷”的特点, 模型以“对缺陷从产生到解决的流程”为基础, 定义足够但不冗余的属性, 简化沟通流程, 建立高效的软件缺陷管理流程。

缺陷的生命周期是缺陷处理流程的关键属性, 本文定义了 New、Open、Reopen、Postponed、Won't Fix、Assigned、Rejected、Fixed、Closed 共 9 个缺陷生命周期中的状态。

测试用例不仅能帮助缺陷解决人员重现缺陷, 而且, 顺

基金项目: 国家科技支撑计划基金资助项目(2007BAF23B0302); 重庆市科委自然科学基金资助项目(2010BB2244)

作者简介: 文俊浩(1969—), 男, 教授、博士、博士生导师、CCF 会员, 主研方向: 服务计算, 面向服务的软件工程; 田 清, 硕士研究生; 李 朋, 博士研究生

收稿日期: 2011-04-29 **E-mail:** qing_798004@163.com

延“缺陷→用例→需求”跟踪次序,还能帮助产品经理明确缺陷与需求的关系。缺陷解决方案能保证重复缺陷或相关缺陷的修复效率。关联缺陷的相关信息能帮助缺陷的解决。因此,还需记录测试用例、解决方案、关联缺陷 3 个缺陷信息。

根据缺陷属性定义,建立如图 1 所示的管理模型。敏捷团队发现缺陷后,则在缺陷库中查询是否存在重复缺陷,若存在,则将缺陷重新打开;若不存在,则记录缺陷,关联或补充缺陷对应的测试用例,设缺陷状态为 New。每日站会中,产品经理组织团队成员对缺陷进行评审,设无效缺陷状态为 closed,无须改正的缺陷状态为 Won't Fix,优先级不高的缺陷状态为 Postponed,并将 Postponed 状态的缺陷放入需求条目;设通过评审的缺陷状态为 Open。项目经理对于 Open 状态的缺陷进行分配,设缺陷状态则为 Assigned。开发人员和 UI 设计师修改被分派给自己的缺陷,修改完毕后,将缺陷状态设为 Fixed,并记录解决方案,提交测试人员验证,若验证通过,测试人员将关闭缺陷,若验证未通过,则测试人员提交给关闭缺陷,相应解决者重新修改;对于不能重现的缺陷,解决者将状态设置为 Rejected,并且提交测试人员验证,若验证未通过,则重新开启缺陷,并且提交给产品经理组织的评审会再次审核;若验证通过,测试人员关闭缺陷。

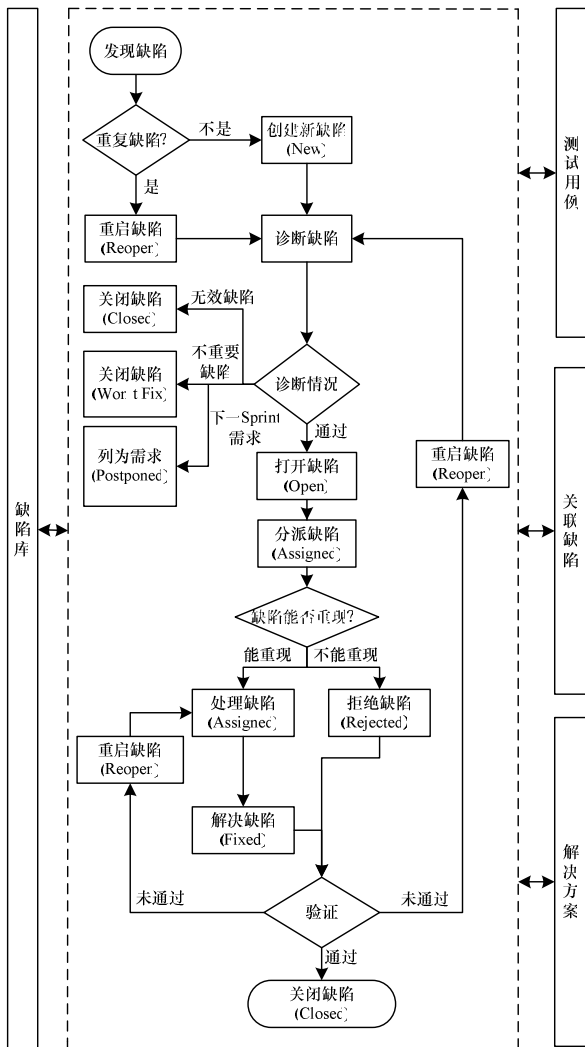


图 1 基于缺陷管理流程的缺陷管理模型

3.2 软件缺陷度量模型

敏捷开发中的软件度量需体现“敏捷”的特点,即“实

时”地将缺陷数据转化为简明有效的信息产品反馈给敏捷团队成员,以便敏捷团队成员时刻了解项目状况,及时调整项目方案。下面将从团队角色关注的信息产品着手分析,构建缺陷度量指标^[3]。

产品经理需明确产品需求中的问题,根据市场情况、投入产出比等对缺陷优先级进行判定,并及时调整需求:若需求重要程度高,缺陷数量较多,则考虑放弃本 Sprint 中重要程度低的需求;若用户需求不够明确,缺陷数量较多,则综合用户调研,考虑需求变更。这就需要明确针对某一需求缺陷状况。本文通过测试用例的记录来跟踪需求:测试人员在制定测试计划时,标注针对某一需求的测试用例,在执行测试计划时,测试用例对应的缺陷,便是需求所对应的缺陷。

项目经理需了解项目进度和团队人员的工作情况。信息可通过缺陷数量,缺陷修复率和缺陷模块当量来展示。开发团队(开发人员、UI 设计师和测试人员)需明确自己所负责的缺陷状况。此外,项目经理和开发人员还需预测缺陷发展趋势,以改进以后的软件开发过程。

缺陷数量在每日站会中作为项目进度和工作状态直接指标。项目经理根据 Sprint 周期中不同阶段设定阈值。若超过此阈值,项目经理需分析原因,或调整团队工作状态,或调整项目人力分配等。

在每个 Sprint 结束后将发布一个可用系统,因此,敏捷开发中缺陷修复率每日需达到 70% 以上。设所有被发现的缺陷数量为 N_d ,已修复的缺陷数量为 N_{dr} ,设缺陷修复率为 R_d :

$$R_d = N_{dr} / N_d$$

缺陷模块当量以缺陷严重程度为基础,反映团队工作状态和缺陷修复成本。为避免繁杂,将缺陷严重程度设定为 4 个等级。设缺陷不同严重程度权值向量 $W(W_{s0}, W_{s1}, W_{s2}, W_{s3})$, 其中, W_{si} 为严重程度 S_i 类型缺陷的权值。设缺陷数量向量 $N(N_{s0}, N_{s1}, N_{s2}, N_{s3})$, 其中, N_{si} 为严重程度 S_i 类型缺陷的缺陷数量。设缺陷模块当量为 DME , 则:

$$DME = W \times N$$

在一个 Sprint 期间,因缺陷数据量较小、时间周期较短,因此缺陷预测意义较小。缺陷预测价值体现在前几个 Sprint 的缺陷数据对于下一个 Sprint 的指导。此度量利用灰色系统理论 $GM(1,1)$ 预测模型,预测不同来源的缺陷数量发展趋势。基于敏捷原则以及实践经验,本文将缺陷来源类型分为代码错误、用户界面、需求变动、新增需求、设计文档、配置相关、安装部署、安全相关、性能压力、外部依赖、测试脚本、产品建议^[4]。对缺陷预测模型的说明如下:

(1)提出了一个称为模块缺陷率 MBR (Module Bug Ratio) 的概念,则:

$$MBR = DME / LOC$$

其中, DME 是缺陷基于来源的分类后的 n 个 Sprint 缺陷权值向量与缺陷数量向量的内积; LOC 表示代码行数。

(2)设某一类型的缺陷数量在 n 个 Sprint 结束后的缺陷数据 $X(X_1, X_2, \dots, X_n)$, 进行累加生成新序列为 $X'(X'_1, X'_2, \dots, X'_n)$, 其中, $X'_i = \sum_{t=1}^i X(t)$, $i = 1, 2, \dots, n$ 。

(3)对 X 向量进行光滑性检测,则:

$$r(i) = X(i) / X'(i-1) \quad (i > 3)$$

若 $\rho(i) \in [1, 1.5]$, $\delta = 0.5$ 成立,则光滑性条件满足。通过以上验证,便建立了 $GM(1,1)$ 模型。

(4)计算响应方程,则:

$$X'(i+1)=[X(1)-u/a]e^i+u/a$$

其中,待估参数 u 、 a 由向量 $A=[a,u]^T$ 利用最小二乘给出:

$$A=(B^TB)^{-1}B^TY_n$$
$$Y_n=(X(2),X(3),\cdots,X(n))^T$$
$$B=\begin{bmatrix}-1/2(X'(2)+X'(1))&1\\-1/2(X'(3)+X'(2))&1\\\cdots&\cdots\\-1/2(X'(n)+X'(n-1))&1\end{bmatrix}$$

4 Scrum 软件敏捷开发中的缺陷管理方法应用

针对敏捷开发中的缺陷管理方法的研究,设计实现了基于 Scrum 软件缺陷管理工具 RQMS,主要实现以下 2 个功能:(1)缺陷管理流程;(2)缺陷度量信息可视化^[5]。

如图 2 所示,为提升缺陷管理流程中每日站会效率,设计了基于缺陷属性字段选择的搜索机制。搜索条件可通过下拉框来进行选择;搜索条件数量的可通过图 2 中的+和×来进行增加和删除。

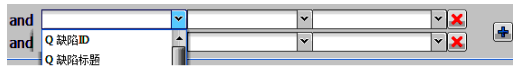


图 2 RQMS 基于缺陷属性字段的搜索机制

如图 3 所示,某一产品一个 Sprint 周期(3 周)缺陷修复率可视化图形。项目经理可清楚地判断每日团队工作情况和产品质量。

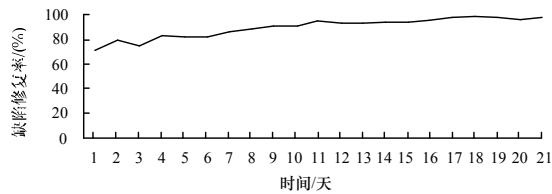


图 3 缺陷修复率度量图

从 2010 年 3 月 RQMS 开始投入使用。表 1 是通过针对其中一款产品,对比了分别使用 RQMS 与传统缺陷管理工具

Bugzilla 的效果。

表 1 RQMS 和 Bugzilla 的使用效果对比

对比指标	RQMS	Bugzilla
每日站会缺陷审核数	44	23
录入 bug 平均时间/s	30	120
缺陷修复平均时间/h	0.8	1.5
每个 Sprint(3 周)各角色维护缺陷管理流程平均时间/h	6	10
产品迭代总次数	7	9
每个 Sprint 用户平均满意度/(%)	90	74

从表 1 可看出应用 RQMS 后,在投入人力基本不变的情况下,工作效率明显提升,为互联网产品争取了时间和用户。

5 结束语

本文从互联网产品实施基于 Scrum 软件的敏捷开发过程入手,建立了“敏捷开发软件缺陷管理流程模型”和“敏捷开发软件缺陷度量模型”来对缺陷进行管理。通过设计实现 RQMS 缺陷管理工具对管理方法进行应用,证明该管理方法能保证敏捷团队充分沟通,缺陷快速解决,产品尽快交付。本文提出的敏捷开发缺陷管理方法对于敏捷团队开发实践具有较好的指导意义。但由于本方法建立在开发实践基础之上,要达到最佳实践水平还需进一步探索及经验积累,这将在今后的研究和实践中进行改进。

参考文献

[1] 罗 霄,侯 红. 基于开源的软件缺陷管理工具的改进策略[J]. 计算机工程, 2009, 35(1): 65-67.

[2] Rising L, Janoff N S. The Scrum Software Development Process for Small Teams[J]. Software, 2000, 17(4): 26-32.

[3] 张 瑞,郝克刚. 软件缺陷度量[J]. 计算机应用研究, 2005, 22(4): 54-57.

[4] 李新军,刘晓明,黄 松. 基于软件过程度量的正交缺陷分类技术[J]. 计算机工程, 2009, 35(23): 30-33.

[5] Henderson C. Managing Software Defects: Defect Analysis and Traceability[J]. ACM SIGSOFT Software Engineering Notes, 2008, 33(4): 1-3.

编辑 任吉慧

(上接第 34 页)

住宿、景点和餐饮 3 个领域 12 个种类的服务实例进行模拟。同时根据经验和相关领域业务规范编制了 30 条规则作为本体规则库,以 0.6 作为匹配阈值,各规模下进行多次搜索,取平均值后得到了如表 1 所示的实验结果。

表 1 服务查询结果

服务数目	访问率/(%)	查全率/(%)	时间比
100	31	92	100
200	36	90	132
300	33	90	143

由实验结果可知,不同规模下的访问率基本保持稳定,会出现周期性小幅波动,这是由于随着服务数目增多,Chord 网络可能动态调整拓扑以控制代表服务概念的 Chord 网络规模,因此保证了稳定的访问率;查全率相对较高并且网络规模对其影响不大,这是由于采用了语义匹配和规则推理,整个服务查询的搜索方向较为准确;时间比是以服务规模为 100 的查询时间作为基值,各种规模下查询时间与之相除再乘以 100 所得的数值,可以看到,时间消耗增速逐渐放缓,服务数目增加的影响由于网络的动态调整策略得到了缓解,使整个系统具有良好的响应性。

5 结束语

本文提出了服务本体映射多 Chord 的服务组织方式,利用本体规则推理高效地为用户推荐需要和潜在需求的服务。下一步将重点研究服务的描述匹配推理,进一步提高服务推荐和选取的准确性。

参考文献

[1] 徐 萌,陈俊亮,彭 泳,等. 基于服务关系本体的服务生成[J]. 软件学报, 2008, 19(3): 545-556.

[2] Wang Zhixiao, Meng Fanrong. Hierarchical Semantic Overlay Network[C]//Proc. of International Conference on Computer Science and Software Engineering. Wuhan, China: IEEE Press, 2008.

[3] 杨 峰,郑伟民,余宏亮. 一种基于 DHT 混合型对等发现服务的算法设计[J]. 计算机应用研究, 2007, 24(3): 99-102.

[4] 刘传昌,陈俊亮. 目标 Web 服务描述本体和服务发现模型[J]. 计算机工程, 2007, 33(18): 187-189.

[5] 潘善亮,张迎新,俞晓锋. 基于 OWL-S 的 Web 服务匹配研究[J]. 计算机工程, 2010, 36(12): 48-51.

编辑 张 帆

