代码审查缺陷密度量化模型研究

鸦 文,杨沁梅
(中国电子科技集团公司第二十八研究所,江苏 南京 210007)

摘 要:为通过代码审查活动达到对软件产品质量提升的作用,给出了代码审查平台搭建方案,并据此平台策划开展了多个工程的代码审查活动。利用首轮采集的数据,初步分析确立了代码审查缺陷密度模型和影响因子,计算得到了代码审查缺陷密度的基线目标值。该模型可以供研发团队和研发团队所属组织策划确定代码审查基线,并策划开展后续代码审查活动。

关键词:代码审查;代码审查平台;缺陷密度
中图分类号:TN9 **文献标识码**:A DOI:10.16157/j.issn.0258-7998.200912

中文引用格式:鸦文,杨沁梅. 代码审查缺陷密度量化模型研究[J].电子技术应用,2021,47(8):106-109,115.
英文引用格式:Ya Wen,Yang Qinmei. Research on quantitative model of code review density[J]. Application of Electronic Technique,2021,47(8):106-109,115.

Research on quantitative model of code review density

Ya Wen,Yang Qinmei
(The 28th Research Institute of China Electronics Technology Group Corporation,Nanjing 210007,China)

Abstract:In order to improve the quality of software products by code review, this paper gives the construction scheme of code review platform which helps planning code review activities of several projects. Based on the data collected in the first run, the defect density model and influence factors of code review are preliminarily analyzed and established, moreover, the baseline target value of it is calculated. The research and development team and its organizations can use this model to determine the baseline of code review and plan subsequent code review activities.

Key words:code review;code review platform;defect density

0 引言

测试是“使用为发现错误所选择的输入和状态的组合而执行代码的过程”[1-4],代码审查是软件测试的手段之一,是在不执行软件的条件下有条理地仔细审查软件代码,从而找出软件缺陷的过程。代码审查必须依靠具有软件系统开发经验、编程经验、测试经验的技术人员集体审查[5]。进行代码审查的主要目的是提高软件质量,及早发现软件缺陷,避免因这些缺陷造成更大的灾难[6]。在开发过程初期进行软件代码审查非常有价值,不仅可以找出后期软件测试阶段难以发现或隔离的软件缺陷,降低研发成本;同时还可以促进开发团队内部沟通和知识共享,提升团队开发能力。

量化管理是CMMI的主要内容之一,量化管理使得软件管理者拥有决策的客观基础,能在量化的范围内预测性能,可以有效地监控项目过程,处理过程偏差的特殊原因[7-12]。

为此,可以将代码审查过程识别为影响项目目标达成的关键过程之一,通过建立过程性能模型和使用适当的量化分析方法[13-14],来指导代码审查活动的策划和实施,达到提升产品最终质量的目标。统计过程控制中主要利用控制图来记录过程质量[15]。

1 代码审查平台

软件项目管理过程中,组织往往缺少量化的数据支撑,从而导致对于有关量化的改进目标制定或者项目的量化目标制定无从下手[16]。为此,可以通过集成开源的配置管理、代码检测、权限控制工具,建立代码审查平台来进行在线审查和数据采集,为后续的模型建立和量化分析过程提供平台支撑。

一种可行的代码审查平台设计架构如图1所示。

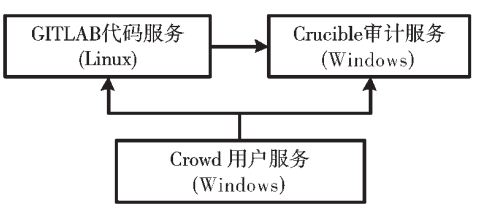


图1 代码审查平台架构

其中：

(1)Crowd 是 atlassian 公司推出的集成化组件，其实现和配置简单，功能齐全，用于用户管理和授权登录。利用此平台可录入代码审查活动相关人员信息并赋予不同的使用和管理权限。

(2)Crucible 工具是一个代码检测工具，能检查、注释、编辑代码，并记录结果，帮助开发人员发现和纠正缺陷，提高代码开发效率。代码审查活动主要依托共用平台 Atlassian Crucible 模块实施。

(3)Gitlab 是一个用于仓库管理系统的开源项目，使用 Git 作为代码管理工具，并在此基础上搭建起来的 Web 服务。该模块用于浏览源代码和代码缺陷管理。开发团队可使用该模块浏览历史版本，也可以利用其代码片段收集功能实现代码复用、查找。该模块非强制使用，由项目组/业务室选择使用。

2 代码审查触发时机

代码审查由项目软件负责人或专业/业务负责人在完成以下代码研发和自测后组织执行。可重点针对以下对象进行审查：

(1)被定义为关重件配置项的代码；

(2)核心算法、主要业务逻辑、关键数据处理、提供对外接口的代码；

(3)新员工产生的代码；

(4)前期测试过程中问题较多的软件代码。

3 代码审查

3.1 审查准备

3.1.1 建立专家库

项目研发组织首先应建立代码审查专家库，主要成员可以是组织内各领域的资深程序员和各专业/业务的技术骨干。

每次代码审查前，由各项目/专业/业务负责人依据被审查代码的专业/业务类型从部门代码审查专家库中选取除被审代码编写者之外的代码审查人员，成立代码审查组，并指定代码审查组长。

3.1.2 设置用户权限

依托代码审查平台可以实现用户的添加和权限设定，主要步骤如下：

(1)由配置管理员在 Crowd 模块中录入人员信息(支持分组)，并设定各模块管理员和使用者角色；

(2)配置管理员将 Crucible、Gitlab 模块的权限信息与 Crowd 相连，确保 Crowd 模块中的用户信息在各个模块中可见；

(3)各项目/专业/业务负责人在需使用的模块中设定人员权限。

3.1.3 选取和提交审查代码

代码审查前，各项目/专业/业务依据第 2 节选取需审查代码(但不限于)。

3.2 执行代码审查

代码审查专家在 Crucible 上开展代码审查活动，审查重点为代码整体架构、风格、逻辑质量等，审查项及主要要求参考表 1。

表 1 首轮采集

工程对象编号	审查专家能力	开发人员能力	审查人时	软件规模/千行	发现缺陷个数	整改缺陷个数	审查速率/(千行/小时)
1	5	5	4	2.7	8	6	0.675
2	5	5	3	3.2	5	5	1.067
3	5	5	3	3.1	8	8	1.033
4	3	5	5	4.5	2	2	0.9
5	5	5	4	3.7	2	2	0.925
6	3	5	5	4.3	3	3	0.86
7	5	3	2	1.3	21	21	0.65
8	5	3	4	3.4	2	2	0.85
9	3	3	2	0.8	7	7	0.4
10	5	1	2	1.3	3	3	0.65
11	5	3	2	2	1	1	1
12	5	1	2	1.2	3	3	0.6
13	3	1	2	1.2	4	4	0.6
14	3	3	2	1.3	6	6	0.65
15	5	5	5	3	5	5	0.6
16	3	3	2	1.2	4	4	0.6
17	5	3	5	6.3	4	4	1.26
18	5	1	8	7.1	7	7	0.888
19	3	5	8	5.6	1	1	0.7
20	3	3	8	6.1	5	5	0.763
21	5	5	8	7.7	17	17	0.962

具体方法：

(1)代码审查专家登录“lzb.cru.com:8060/cru”，打开代码开展审核；

(2)点击有问题的代码，并输入审查意见，点击“Comment”提交。

3.3 审查问题整改归零

(1)代码审查后，开发人员根据代码审查专家提出的问题对代码进行修改；

(2)完成修改后，重新提交代码审查；

(3)代码审查人员进行回归确认。

3.4 审查结果处置

项目代码审查活动结束后，由代码审查组长汇总、分析代码审查数据，并及时将代码审查过程中遇到的典型编码缺陷在开发团队或全部门集中宣贯，减少典型缺陷重复发生。

4 代码审查过程性能模型建立

4.1 模型定义

代码审查缺陷清除模型预测值为代码审查发现缺陷的密度，分析确定影响部级代码审查质量的因子是：审查专家能力、开发人员能力和代码审查速率。即：代码

审查发现缺陷的密度=f(审查专家能力,开发人员能力,代码审查速率)。

模型因子取值如下:

(1)审查专家能力取值:5 表示能力高,从事软件编码工作 5 年及 5 年以上;3 表示能力中,从事软件编码工作大于等于 3 年且少于 5 年;1 表示能力低,从事软件编码工作少于 3 年。

(2)开发人员能力取值:5 表示能力高,从事软件编码工作 5 年及 5 年以上;3 表示能力中,从事软件编码工作大于等于 3 年且少于 5 年;1 表示能力低,从事软件编码工作少于 3 年。

(3)代码审查速率=代码审查代码规模÷代码审查总工时。

4.2 数据采集

首轮采集了 21 组有效数据,见表 1。

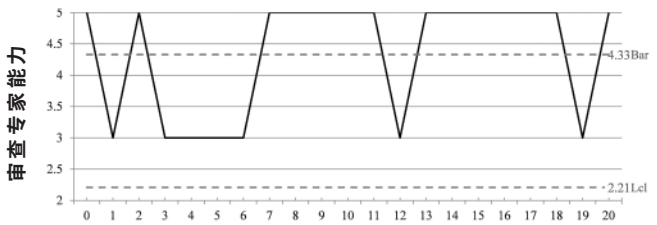
4.3 模型试算

4.3.1 正态性检验

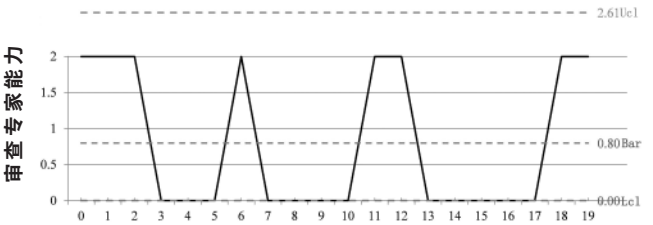
经使用 Mintab 工具计算得到的审查专家能力、开发人员能力和代码审查速率这 3 个参数的正态性检验结果如图 2 所示。

4.3.2 相关性分析

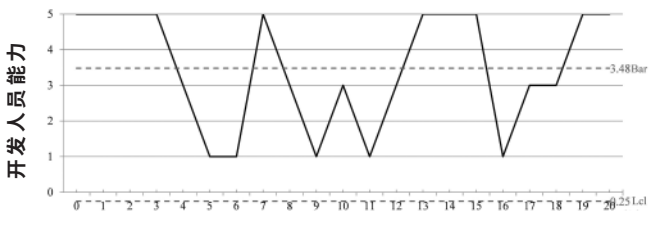
经使用 Mintab 工具计算得到的审查专家能力(x1)、开发人员能力(x2)、审查速率(x3)和代码审查缺陷密度(Y)这 4 个参数的相关性结果如图 3 所示。



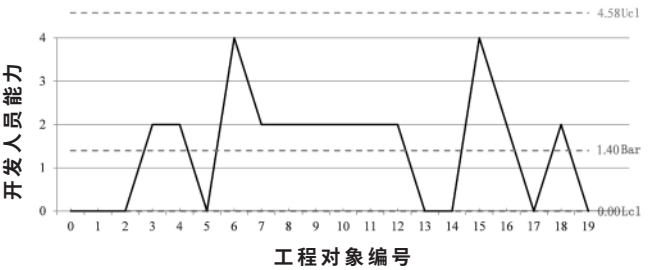
(a)“审查专家能力(x1)”单个观测值控制图



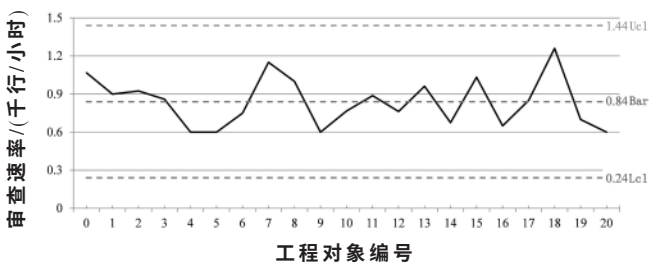
(b)“审查专家能力(x1)”观测值移动极差控制图



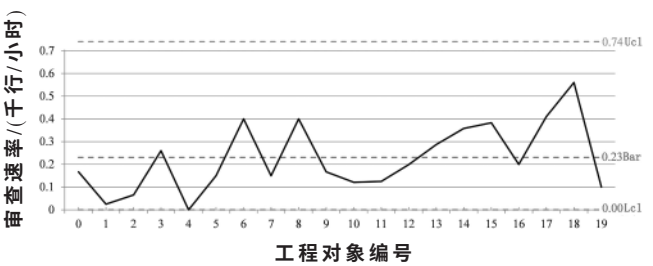
(c)“开发人员能力(x2)”单个观测值控制图



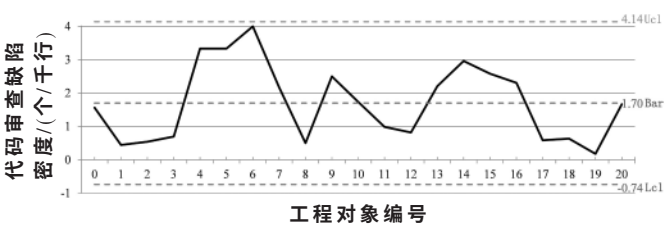
(d)“开发人员能力(x2)”观测值移动极差控制图



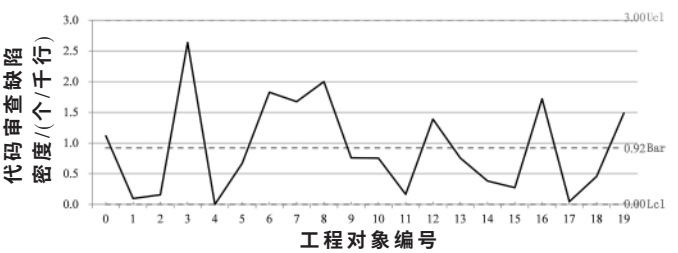
(e)“审查速率(x3)”单个观测值控制图



(f)“审查速率(x3)”观测值移动极差控制图



(g)“代码审查缺陷密度(Y)”单个观测值控制图



(h)“代码审查缺陷密度(Y)”观测值移动极差控制图

注:Bar - 均值;Ucl - 基线上限;Lcl - 基线下限。

图 2 正态性检验结果

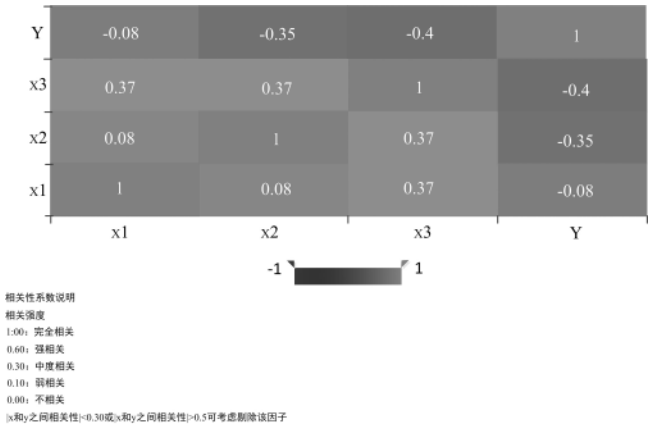


图 3　相关性分析结果

4.3.3 回归分析结果

进一步使用 Mintab 工具计算，得到初步建立的模型算法，如图 4 所示。即：

代码审查缺陷密度(个/千行)=3.583+0.076×审查专家能力−0.159×开发人员能力−1.977×代码审查速率

#	Rsquare	平均绝对误差	推荐公式	推荐子集
0	0.20996014715325861	0.8748690731120002	y=3.583+0.076*x1+-0.159*x2+-1.977*x3	["x1","x2","x3"]
1	0.2064235019055739	0.8687259012446533	y=3.794+-0.161*x2+-1.826*x3	["x2","x3"]
2	0.16371154241888475	0.8697306565395022	y=3.407+0.093*x1+-2.516*x3	["x1","x3"]
3	0.15831576276106007	0.8804668345621336	y=3.665+-2.342*x3	["x3"]
4	0.1260012778955245	0.9220153061224489	y=2.79+-0.061*x1+-0.237*x2	["x1","x2"]
5	0.12335079162861629	0.913954022988506	y=2.537+-0.24*x2	["x2"]
6	0.0064756973005035 69	0.9714285714285715	y=2.113+-0.095*x1	["x1"]

图 4　回归分析结果

4.4 基线建立

经 Mintab 工具计算得到代码审查缺陷密度概率图如图 5 所示，计算得到的 P 值大于 0.05，显示样本数据服从正态分布。

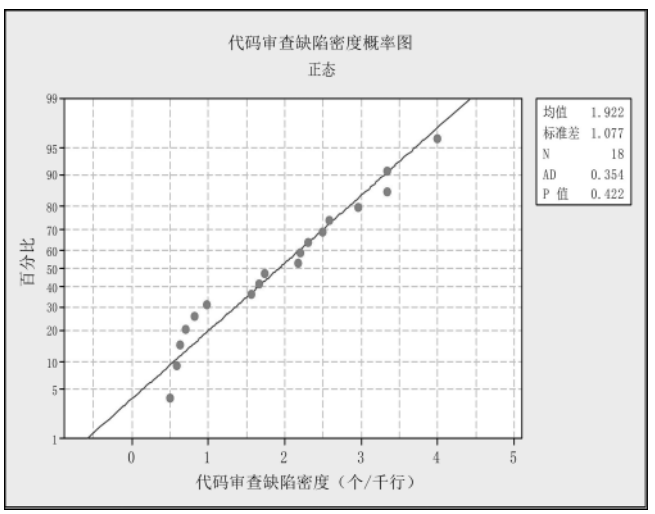
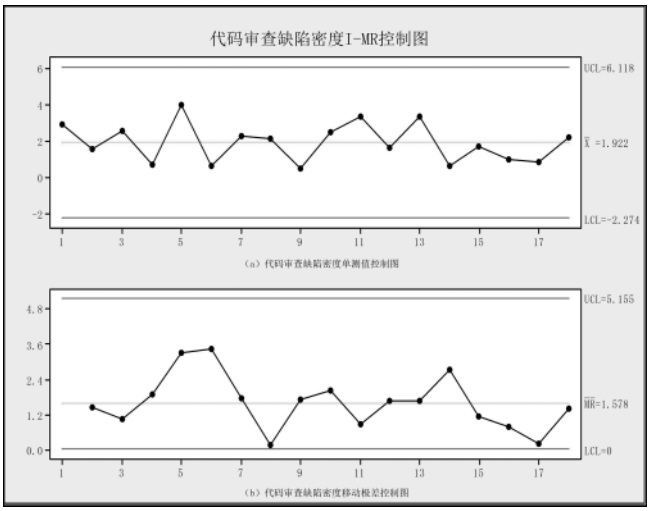


图 5　代码审查缺陷密度概率图

进一步使用控制图检查数据的稳定性，如图 6 所示，数据点随机地分布在中心线(即基线均值)上下，且在基线上下限之间，表明数据基本稳定。



注：Ucl−基线上限；$\overline{X}$−样本均值；$\overline{MR}$−移动极差均值；Lcl−基线下限。

图 6　数据稳定性检查

最终获得的基线值如图 7 所示，即代码审查缺陷密度平均为 1.921 9 个/千行代码。

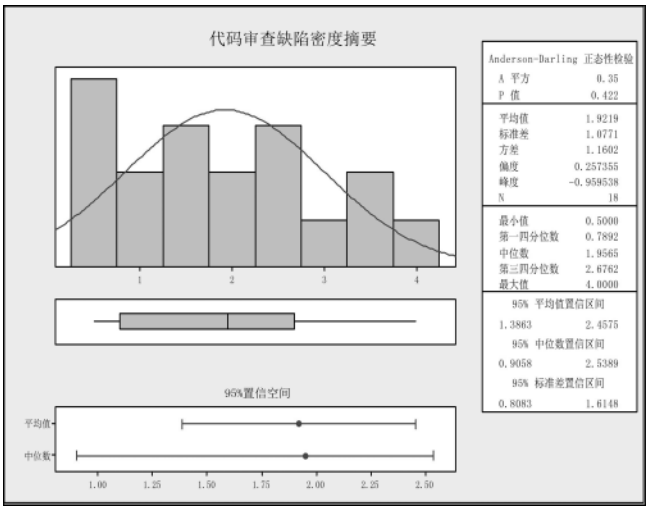


图 7　代码审查基线值

5 结论

研发团队所属组织可以根据组织对产品质量的要求和团队实际情况，参考以上得到的代码审查缺陷密度平均值来设定组织级代码缺陷目标，并使用建立的模型，依据代码审查的代码规模、选定的审查专家能力，计算得到代码审查工时的估计值，并据此估计值进行代码审查活动的策划。通过所属项目后续的测试、检验活动，可以再逐步分析代码审查遗留问题的影响，并重新修正

(下转第 115 页)

introduction[J].International Journal of Computer Vision, 1996, 17(2): 107-112.
[4] 苏锋.带有浮台的移动式水下监控装置: 103501414[P]. 2014-01-08.
[5] 殷莉甜.水下双目视觉定位系统开发与应用研究[D].广州: 华南理工大学, 2016.
[6] 殷莉甜, 谢小鹏, 彭泽林.水下双目视觉系统的设计与密封技术研究[J].润滑与密封, 2016, 41(11): 94-99.
[7] 王新梅, 肖国镇.纠错码——原理与方法(修订版)[M].西安: 西安电子科技大学出版社, 2001: 276-291.
[8] MORELOS-ZARAGOZA R H.The art of error correcting coding[M].2nd ed.New York: John Wiley &. Sons, 2006: 143-150.
[9] 赵登路.水下机器人通信系统的设计与实现[D].哈尔滨: 哈尔滨工程大学, 2015.
[10] 张美琪, 胡国文, 仇荣鑫, 等.基于 P89V51RD 的通用云台控制器设计与实现[M].现代电子技术, 2008, 31(18): 168-170, 173.
[11] 凌好, 刘荣忠, 郭锐, 等.基于 ARM7 的太阳自动跟踪简易控制系统设计[J].计算机测量与控制, 2011, 19(8): 1889-1891.
[12] 王祎晨.增量式 PID 和位置式 PID 算法的整定比较与研究[J].工业控制计算机, 2018(5): 123-124.
[13] 王开宇.基于 C# 的数据与视频监控上位机软件设计[J].现代电子技术, 2017, 40(10): 62-64.
[14] MISTRY J, NAYLOR M, WOODOCOCK J.Adapting Free-RTOS for multicores: an experience report[J].Softw. Pract. Exper, 2014, 44(9): 1129-1154.
[15] BARRY R.Using the FreeRTOS: real time kernel[M].Washington: Amazo, 2010: 10-50.
[16] 牛洪海, 臧峰, 周绪贵.基于 DMA 的高速 UART 串口通信设计与实现[J].自动化仪表, 2018, 39(9): 45-48.
[17] 吴学文, 王新光, 周金陵.基于 Modbus 通信协议的水闸计算机监控系统[J].计算机工程, 2005, 31(13): 195-197.
[18] 程雪婷, 王海峰.解析 Modbus-RTU 协议关键内容及其在智能电器中的应用[J].电器与能效管理技术, 2010(1): 23-25.
[19] 方少雷, 李鹏, 苏跃龙.STM32 系列单片机多串口通讯系统简述[J].山西电子技术, 2016(3): 55-57.

(收稿日期: 2021-03-24)

作者简介:
任福深(1976-), 男, 博士, 教授, 主要研究方向: 智能装备及其控制理论。
王茜(1993-), 女, 硕士研究生, 主要研究方向: 水下机器人智能控制。
刘均(1976-), 男, 博士, 副教授, 主要研究方向: 嵌入式系统、智能仪器、井下装置等。

(上接第 109 页)

模型和目标, 进而逐步达到通过有效代码审查提升代码质量的目的。

参考文献

[1] 张亦含, 沈敏.CMM4 中量化管理的解决方案[J].计算机工程与设计, 2016, 27(5): 742-743.
[2] 郑人杰, 王纬, 王方德, 等.基于软件能力成熟度模型(CMM)的软件过程改进——方法与实施[M].北京: 清华大学出版社, 2003.
[3] 井涛.代码审查在软件工程实施中的重要性[J].电子技术与软件工程, 2017(12): 43-44.
[4] 袁政江.浅谈软件静态测试中的代码审查[J].计算机光盘软件与应用, 2012(3): 202-204.
[5] 方晓宁, 孙纪敏.代码审查-实施软件工程化不可忽视的环节[J].无线电通信技术, 2003(1): 36-38.
[6] 张如云.代码审查在软件开发中的应用研究[J].电脑开发与应用, 2014(6): 50-56.
[7] 丁岳伟, 刘玉敬.软件项目量化管理的实践[J].计算机系统应用, 2012, 21(6): 254-257.
[8] Capability Maturity Model Integration(CMMISM), Version 1.3 for Software Engineering(CMMI-SW, V1.3)[Z].Staged Representation, 2010.
[9] SUNETNANTA T, NOBPRAPAI N, GOTAL QUANTITATIVE O. CMMI assessment for offshoring through the analysis of project management repositories[C]//3rd International Conference on Software Engineering Approaches for Offshore and Outsourced Development(SEAFOOD), Zurich, Switzerland, 2009: 32-44.
[10] 张栋, 任爱华.实施量化管理和持续性改进过程的研究[J].计算机软件与算法, 2008, 29(19): 4887-4889, 4913.
[11] TARHAN A, DEMIRORS O. Apply quantitative management now[J].IEEE Software, 2012, 29(3): 77-85.
[12] BILL C, SESHAGIRI G V, DONALD R, et al.The Case for quantitative process management[J].Software, IEEE, 2008, 25(3): 24-28.
[13] 张旭, 刘浩驰.基于 CMMI 的量化管理在项目中的应用与研究[J].电脑与电信, 2016(4): 62-65.
[14] 孙晶.基于 CMMI 的软件质量度量模型及工具原型[D].大连: 大连海事大学, 2006.
[15] 杜庆峰, 马慧珺.SPC 在软件过程度量中的应用及改进[J].计算机工程, 2009, 35(24): 103-104.
[16] 徐俊, 李军.软件研发过程性能基线和模型建立方法及应用分析[J].现代计算机, 2013(7): 14-17.

(收稿日期: 2020-09-18)

作者简介:
鸦文(1980-), 女, 硕士研究生, 高级工程师, 主要研究方向: 大型信息系统工程项目管理。
杨沁梅(1978-), 女, 本科, 高级工程师, 主要研究方向: 大型信息系统工程项目管理。

版权声明