

# 敏捷开发在战机飞控系统软件研制中的应用

邵山 付滕强 姚方圆 郑义

(中航工业沈阳飞机设计研究所 辽宁 沈阳 110035)

**摘要** 战机飞行控制系统软件是战机最高等级——关键级软件,其高质高效的研制直接影响到战机安全性、可靠性和研制进度。目前存在着提高软件质量的同时缩短研制周期的矛盾。为解决该矛盾,提出一种将敏捷开发方法与军用软件研制能力成熟度模型(GJB5000A)相结合的开发模式,并成功应用于某型战机飞控系统软件研制中,为关键级军用软件项目应用敏捷方法提供了一个范例。

**关键词** 敏捷开发 极限编程 飞控系统软件 GJB5000A 标准 工程应用

**中图分类号** TP311.5 **文献标识码** A **DOI**:10.3969/j.issn.1000-386x.2015.01.011

## APPLICATION OF AGILE DEVELOPMENT IN DEVELOPING FLIGHT CONTROL SYSTEM SOFTWARE FOR FIGHTERS

Shao Shan Fu Chengqiang Yao Fangyuan Zheng Yi

(Shenyang Aircraft Design and Research Institute of AVIC, Shenyang 110035, Liaoning, China)

**Abstract** Fighter flight control system software is the highest level, the critical level, flight software of fighters. Its development with high quality and efficiency has a direct effect on the security, reliability, and developing progress of fighters. Meanwhile, there is a contradiction between improving the software quality and shortening the development cycle. To solve this contradiction, we raise a development pattern which combines the agile development method with the military software research and development capability maturity model (GJB5000A), and have successfully applied it in the development of the flight control system software for certain fighter, thus provide an example for applying the agile method in critical-level military software project.

**Keywords** Agile development Extreme programming Flight control system software GJB5000A Engineering application

### 0 引言

随着现代战机数字化、信息化程度的不断提高,涉及战机飞行安全和性能的飞行控制系统软件正面临着巨大的挑战,其研制周期越来越短,同时软件质量要求也越来越高,这迫使战机飞控系统软件的研制必须采用更高效、更规范的软件开发过程模式,才能同时满足这两方面要求<sup>[1-5]</sup>。军用软件能力成熟度模型(GJB5000A)在飞控系统软件研制过程中的全面应用为解决这一矛盾奠定了基础,但某类型飞控系统软件项目,其工程特点使得其按GJB5000A体系执行时,延期和不能按节点提交工作产品的风险较大,且规避该风险可能影响产品质量,这对安全性等级为关键级的飞控系统软件是不可接受的,而执行GJB5000A为总装备部对军用软件研制单位的硬性要求,因此,需要在符合GJB5000A体系的基础上,探索能符合该类飞控系统软件项目工程特点的新方法。

本文基于某型飞机飞行控制系统软件开发项目特点,进行了软件开发方法研究,形成了敏捷开发与GJB5000A相结合的新开发模式,并进行了工程应用,最后,给出了项目实施建议。

### 1 开发模式建立

#### 1.1 敏捷开发与GJB5000A

敏捷开发的概念于2001年提出,强调以人为核心,进行迭

代、适应性、循序渐进的开发<sup>[6]</sup>,已成功在民用企业广泛应用。GJB5000A是总装备部在美国软件工程研究所(SEI)提出的能力成熟度模型集成(CMMI)1.2版本的基础上,经裁剪和改进而形成的软件工程经验集,军用航空软件开发必须遵守。两者的比较情况如表1<sup>[7-9]</sup>所示。

表1 敏捷开发与GJB5000A比较

比较项	敏捷开发	GJB5000A
结构特点	包含多种敏捷方法,如极限编程(XP)、自适应软件开发(ASD)等,一般不对软件开发过程作强制规定,而是提供一系列指导具体活动的实践	是精细复杂的软件工程经验集,覆盖面广,但仅提出需做什么,未说明具体怎么做,实施组织需将标准进行本地化定义
主要目标	快速响应变更、快速交付	项目的可预见、稳定、可靠性
计划	逐步做计划,按迭代周期分次发布	一般为计划驱动开发,计划需更为详细、周密
适用规模	更适合中小规模	应用于大型项目效果更明显

续表 1

比较项	敏捷开发	GJB5000A
项目沟通	更倾向于人员间沟通、基于阅读代码的沟通,直接、快捷,但证据不易留存,沟通成果不易传播	更倾向于基于文档的沟通,有据可查,但相对耗时
测试	提倡测试驱动开发	无明确的方法规定
团队	对成员技能要求较高,自由度相对较高,提倡正常的工作时间	对成员技能要求相对较低,靠清晰完备的过程定义指导成员活动

通过对敏捷开发的多种方法研究发现,极限编程(XP)作为其中最具代表性的一种方法,与 GJB5000A 二级过程域的满足和支持程度较好(见表 2)<sup>[10]</sup>,决定在 GJB5000A 二级体系中引入敏捷开发的极限编程方法,建立新的开发模式。

表 2 XP 实践对 GJB5000A 二级过程域的满足和支持情况

序号	GJB5000A 二级过程域	相关的 XP 实践	满足程度
1	需求管理	用户素材,现场客户,持续集成	较好
2	项目策划	计划游戏,小型发布计划	较好
3	项目监控	每日例会,开放的工作空间	较好
4	过程和产品质量保证	结对编程,持续集成,测试驱动开发,验收测试,编码规范	部分满足
5	配置管理	集体所有权	部分满足
6	测量与分析	每日例会	部分满足
7	供方协议管理	无	未提及

### 1.2 敏捷开发与 GJB5000A 融合

某型战机飞行控制系统软件项目的特点如表 3 所示。

表 3 飞控系统软件项目特点

序号	特点描述
1	安全性等级为关键级,对软件开发、测试、管理有着军用软件最高要求
2	研制任务紧,软件规模为中小规模
3	功能复杂,需求变化大
4	根据型号节点要求快速提供中间供试验版本

根据这些特点,基于项目要求、人员、设备等条件限制,不断地对 XP 实践进行裁剪和改进,并与单位 GJB5000A 二级体系融合,提出了如图 1 所示的新开发模型<sup>[11-13]</sup>,该模型覆盖 GJB5000A 二级的全部七个过程域。

模型采用两周一次迭代,迭代周期在时间上可重叠,但每个迭代周期的评审节点仍应保证顺序进行。在每个迭代周期中,首先由项目经理确认并发布本次迭代周期中需完成的工作内容,随后,由开发人员将需求分析、设计与实现都融合在“结对编程”活动中完成,与此同时,测试人员同步编写测试用例、代

码并进行测试,保证代码的健壮性。

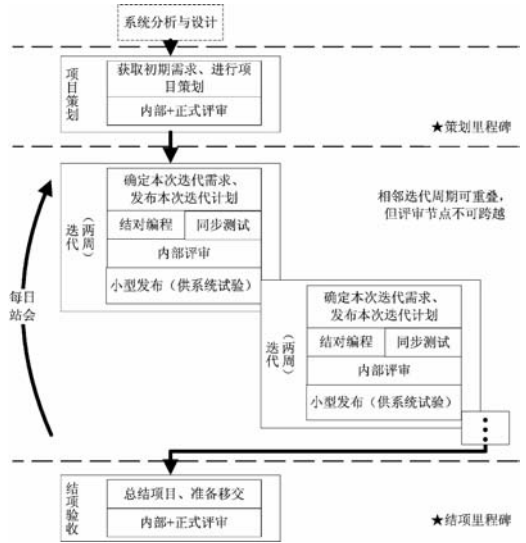


图 1 敏捷开发与 GJB5000A 融合的新开发模式

## 2 开发模式应用

### 2.1 应用过程

新开发模式在某型战机飞行控制系统软件项目研制过程中成功应用,以下为工程应用经验:

#### (1) 人力分配

由于此类项目本身研制任务紧、重,节点要求高,一般为从后向前倒排进度,若采用各工程阶段串行的开发过程,在某时间段内,可能出现开发人员任务极为繁重、不得不采用加班、临时补充人员等措施保证节点,而相关测试人员任务则相对轻松的情况,在另一时间段,情况则可能恰恰相反,这种人力安排不适合类似本项目需要“集中兵力、短时击破”的项目。

新开发模式采用工程活动高度并行、迭代周期可重叠但评审节点不跨越的方式,大大降低测试对开发的依赖程度,且若项目组某人员本次迭代中安排的任务已完成,即可着手下次迭代中相关任务,个人的可调配程度加大,有效避免了因分工不同导致的单位时间内人力资源分配不均或闲置,提高了单位时间人员利用率和工作效率,减少加班,为项目提速、缩短开发周期打下基础。

#### (2) 项目监控和测量分析

未引入敏捷方法时,监控和测量分析的时机均为双周和评审节点。为融入敏捷方法,本项目提高了项目监控频度和效率:一是以迭代周期中每日 5~15 分钟的精简的站立会议监控项目情况,讨论存在问题,以简捷的会议纪要形式记录解决的问题以备查;二是在每评审节点前整理该时间段项目研制情况,以完备的项目进展报告形式记录。正式测量分析的时机改为每迭代周期中单周、每评审节点;迭代周期中的每日站会时,仅关注关键路径情况,若有问题直接在会上提出。以上方式与未引入敏捷的 GJB5000A 相比,形式稍显繁复,但由于问题发现时间提前、响应速度变快、对关键路径的关注频度提高,项目进度更平稳可控。

#### (3) 客户参与

以往项目中,客户一般仅参加评审会和少数临时协调会,本项目中,客户在迭代周期中还需在每周一、三参与团队每日站

会,尽早了解项目情况、修正需求,这种方式缩短了需求变更的响应链,减少了由变更引起的代价,降低了项目延期的风险。

#### (4) 文档

敏捷开发方法提倡少且精的文档。而飞控系统软件开发文档需遵循《军用软件开发文档通用要求》(GJB438B-2009),因此本项目未过多裁剪开发文档的种类,但对文档形成时机做了调整:项目策划期,编制详细、规范的开发计划;在迭代周期中,只要求有相应文档的相应内容即可(如该次迭代内容对应的计划、需求规格说明、测试报告等,均在对应工程活动过程中同步产生),不对文档的规范性作要求;迭代周期结束,评审时,对相应文档内容做抽样检查;结项验收阶段中,完善和规范所有文档。

做了以上调整后,项目组人员在迭代周期中,集中精力实现需求并进行测试,不同职责的人员间需沟通时,采取直接沟通而不是通过阅读文档获取信息,减少了迭代周期中用于文档编制和维护的时间,压缩了版本小型发布的周期,但结项验收阶段的文档完善与规范工作量较大。下一步工作中,需着重关注和调整融合敏捷开发后的文档编制方法。

#### (5) 编程

以往项目中,编程人员间的工作相对独立,本项目修改性应用了XP的结对编程、简单设计、重构、编码规范实践,要求编程人员结对,使用相邻两台电脑,遵循统一的编码规范、采用适度的简单设计(需保证一定的代码灵活性和可移植性,而非XP提倡的最简原则)和重构原则编程,结对人员每天互查代码,发现问题随时交流,这种方式在形式上弱化了详细设计,但实际上在编码过程中强化了详细设计,强调在编码中不断重构,提高代码质量和可读性,降低修正代码缺陷的代价。

#### (6) 测试

与以往的先开发后测试不同,敏捷开发要求测试驱动开发,直接根据需求编写测试用例和代码,再编写被测试的代码,确保所有代码都可被测试,能避免过度设计,能迅速定位代码缺陷,代码可靠性更高。由于条件限制,本项目暂采用与开发同步的测试方法,待人员熟悉且工具支持度更高时,逐步向测试驱动开发转变。飞控系统软件作为关键级软件,还必须全部进行软件单元测试和软件配置项测试,在本项目中,第一、二迭代周期侧重于单元测试,第二、三周期侧重于配置项测试。

#### (7) 集成

引入并改进XP的持续集成。以往项目的集成时机一般为评审节点前。本项目选定每周三、五下午,以及每迭代周期编码任务完成后、实施评审前集成的方式,以所有代码通过测试为该次集成完成的标志。这种方式能尽早发现代码缺陷,降低修正代价,提高软件透明度,增强项目组及客户的信心。

#### (8) 供方管理

本项目有部分软件以供方协议形式外包,需按GJB5000A执行供方协议管理,目前,国内仅少数几家军工单位覆盖该过程域。本项目供方协议中,要求对供方实施单周电话监控、双周现场监控以及评审点监控,项目方按既定的检查单检查,同时,供方需按项目方提出的测量要求,提供原始测量数据,由项目方核实并进行数据分析。

#### (9) 工具

为保证规范度和效率,本项目使用了项目管理、配置管理、需求管理等辅助工具。

#### (10) 其他

未提及的其他活动仍按GJB5000A体系规定执行。

## 2.2 应用效果

本项目采用在GJB5000A体系中引入敏捷方法的开发模式,成功地在进度紧张的情况下高质量完成了研制任务,证明了新开发模式的实用性和一定的先进性。表4为同一项目团队在前期按GJB5000A体系执行的某飞控系统软件项目与本项目的研制情况对照。

表4 研制情况对照表

对照项	前期某项目	本项目	应用敏捷开发的效果	产生该效果的原因分析
延期情况	未延期,有延期风险	未延期,无延期风险	项目进度更平稳可控,规避了项目延期风险	新模型缩短了需求响应时间,提高了人力利用率和效率,减少了转阶段时错误
文档缺陷数	9个	12个	增加了33.3%,文档质量有降低	文档编写时机提前并缩短,对人员个人能力要求增加,导致文档质量不稳定
代码缺陷率	16.5个/千行	9.05个/千行	降低了45.2%,代码质量更可靠	结对编程、小型发布等机制在源头避免了部分代码缺陷

## 3 结语

通过敏捷开发与GJB5000A相结合的新开发模式在某型战机飞行控制系统软件项目中的成功应用,可以得出如下结论:

1) 根据项目具体情况修正后,敏捷开发方法可与GJB5000A融合。敏捷开发方法属于并行、精细化管理,比传统方式对各类软件人员素质要求更高;

2) 本项目所采用的融入敏捷开发的GJB5000A实施方法,可用于需求变化较大、研制周期短的中小规模关键级军用软件项目的开发,能规避延期风险、提高代码质量,初次应用文档质量可能有所降低;

3) 敏捷开发方法与GJB5000A结合后可应用在关键级军用软件项目中,后续项目可根据其工程特点酌情应用敏捷开发,并在应用中不断探索如何取得最佳效果。

## 参 考 文 献

- [1] 金敏,周翔.高级软件开发过程;Rational统一过程,敏捷过程与微软过程[M].北京:清华大学出版社,2005:5-10.
- [2] 隋立江.敏捷方法在软件开发过程中的实现[J].航空制造技术,2011(10):64-67.
- [3] 韩鹏.小规模团队的敏捷开发研究[J].软件工程师,2011(7):42-46.
- [4] 刘从越,孙刚,仲里. Scrum 与 CMMI 在中小型安全关键软件中的应用[J].计算机工程与应用,2011(13):55-58.
- [5] 李艳红,段鹏宇.信息系统敏捷性的实现和提升研究[J].微型机与应用,2013(5):63-65.

反复五次,使得每位参与者提供的数据都被测试一次,最终输出平均测试性能)对语义化访问地点挖掘进行评测,评测结果如表 1 所示。从表中可以看出,基于访问时间特征的分类器对具有较强私人含义的地点(如家、工作地点)分类性能较好,而对公共地点(如超市、商店)分类性能较差。这说明用户通常对私人地点具有较为一致的访问时间特征,而对公共地点的访问时间特征较为多样化。针对此问题,可采用如下方法提高公共地点识别性能:首先,基于现有的 GIS 查询服务(如 Google Place)构建指定区域内的语义化兴趣地点数据库(包含区域内地点的类型、坐标、名称等信息)。然后,基于访问点坐标将挖掘得到的访问地点表示为一个多边形,并采用空间数据库 INTERSECT 操作查询与访问地点交叠的兴趣地点集。最后,访问地点类型被指定为交叠兴趣地点集中出现次数最多的地点类型。现有 GIS 数据库包含了大部分的公共地点,而基于时间特征的分类器可准确地识别私人地点,因此结合两者可提高访问地点的总体识别性能。然而,本文提出的社会关系类型估计方法只考虑三类总体地点类型,即家、工作地点和其他。实验结果表明基于时间特征的分类器可准确地对三类总体地点类型进行分类。

表 1 语义化访问地点识别性能(具体类型识别性能指对具体地点类型的识别准确率,总体类型识别性能指对总体地点类型的识别准确率)

总体地点类型	具体地点类型	具体类型识别性能	总体类型识别性能
家	家	100%	100%
工作地点	工作地点	95.7%	95.7%
其它	餐厅	93.8%	89.6%
	超市	42.9%	
	娱乐地点	28.5%	
	商店	65.7%	
平均性能	73.2%	96.1%	

## 2.2 社会关系类型估计评测

社会关系类型估计评测在所有地点语义被正确标注的基础上进行。我们计算得到所有主体用户-目标用户对(即每个参与者与其手机检测到的所有蓝牙设备)的邻近特征向量,并采用 10 折交叉验证(90% 正确标注了社会关系类型的邻近特征向量作为训练集,另外 10% 的邻近特征向量作为测试集)进行评测(同时对训练集和测试集进行后处理保证两个数据集中均包含属于每种社会关系类型的邻近特征向量)。评测指标包括正确率(被正确估计的邻近特征向量数量与总邻近特征向量数量的比值)、准确率(被正确估计为指定社会关系类型的邻近特征向量数量与估计为指定社会关系类型的邻近特征向量数量的比值)和召回率(被正确估计为指定社会关系类型的邻近特征向量数量与标注为指定社会关系类型的邻近特征向量数量的比值)。评测结果如表 2 所示,其中正确率的准确率较高说明提出的方法可以过滤掉绝大多数与主体用户没有相应社会关系的用户。然而,召回率相对较低(特别是对“朋友”社会关系类型的估计)说明提出的方法可能会遗漏一些与主体用户有社会关系的用户。通过对实验数据的分析,我们发现这个现象是由于参与者与其部分朋友的物理交互(相遇)时间较短,且参与者常常关闭手机的蓝牙功能(如出于隐私或省电考虑),导致算法无法检测到足够的物理交互量来判断其与某个目标用户是否具有相应的社会关系类型。

表 2 社会关系类型估计性能

社会关系类型	正确率	准确率	召回率
家人	99.5%	88.9%	80%
同事		95%	95%
朋友		86.7%	68.4%

## 3 结 语

本文提出了一种面向移动社交网络环境的用户真实社会关系估计方法。该方法通过分析智能手机收集到的用户位置和邻近数据估计其与所遇到的蓝牙设备所代表的用户间的三种常见的社会关系类型(家人、同事和朋友)。基于真实数据的实验证实了该方法的有效性。

## 参 考 文 献

- [1] Lugano G. Mobile social networking in theory and practice[J]. First Monday, 2008, 13(11).
- [2] Lin C, Jin B, Long Z, et al. On context-aware distributed event dissemination[J]. Personal and Ubiquitous Computing, 2011, 15(3): 305-314.
- [3] Mokhtar S B, Capra L. From pervasive to social computing: Algorithms and deployments[C]//ICPS 2009: Proceedings of the International Conference on Pervasive Services, 2009.
- [4] Ying J J, Lu E H, Kuo W, et al. Urban point-of-interest recommendation by mining user check-in behaviors[C]//UrbComp 2012: Proceedings of the SIGKDD International Workshop on Urban Computing, 2012.
- [5] Cranshaw J, Toch E, Hong J, et al. Bridging the gap between physical location and online social networks[C]//UbiComp 2010: Proceedings of the ACM International Conference on Ubiquitous Computing, 2010.
- [6] Li Q, Zheng Y, Xie X, et al. Mining user similarity based on location history[C]//GIS 2008: Proceedings of the SIGSPATIAL Conference on Advances in Geographic Information Systems, 2008.
- [7] Eagle N, Pentland A. Reality Mining: Sensing complex social systems[J]. Personal and Ubiquitous Computing, 2006, 10(4): 255-268.
- [8] Haslam N, Fiske A P. Relational models theory: A confirmatory factor analysis[J]. Personal Relationships, 1999, 6(2): 241-250.

### (上接第 46 页)

- [6] Robert C. Martin. 敏捷软件开发原则、模式与实践[M]. 北京:清华大学出版社, 2003: 9-16.
- [7] 匡松, 周启海, 陈森玲, 等. 敏捷软件开发的认识偏差与推广瓶颈浅析[J]. 计算机科学, 2007, 34(12): 294-295, 303.
- [8] 总装备部. 中华人民共和国国家军用标准 GJB5000A-2008 军用软件研制能力成熟度模型[S]. 北京: 总装备部出版发行部, 2008.
- [9] 徐俊, 彭章纲. 敏捷开发过程与 CMMI 实施融合研究[J]. 现代计算机, 2011(31): 21-23.
- [10] Mark C Paulk, 李凌. 从 CMM 角度看待极限编程[EB/OL]. [2006-08-21]. <http://tech.it168.com/oldarticle/2006-08-21/200608211750935.shtml>.
- [11] 谢东强. 敏捷软件开发的双迭代模型[J]. 计算机应用与软件, 2012(6): 176-178.
- [12] 胡文生, 赵明, 杨剑峰, 等. 敏捷开发过程中的迭代策略分析[J]. 微电子学与计算机, 2012, 29(5): 165-169.
- [13] 陆亮亮, 荣国平, 邵栋. XP-PSP 集成软件过程模型[J]. 计算机工程与设计, 2012(8): 3074-3077.