

# 软件项目中的失信因子探讨

孙乐平, 赵佳宝

(南京大学 工程管理学院, 江苏 南京 210093)

**摘要:**随着软件开发规模的急速扩大,软件项目复杂性程度越来越高,软件项目的失败率也越来越高,这与业界缺乏对软件过程的深入理解有关。在软件开发项目中,进度、成本、质量是核心问题,产生这些问题的因素也是多方面的,每一个软件项目都是一个动态的复杂系统,没有对软件开发动态过程及影响软件开发项目绩效的因素的深入理解,就无法制定出提高项目绩效的有效策略。基于此,文中将这些因素统一命名为失信因子,并按照软件生命周期对其进行识别划分,最后提出了一种基于系统动力学思维的失信因子分析方法。通过文中的探讨,在以后的开发项目中相关人员会对失信因子进行重点关注和研究,进而对软件项目进行有效的控制和预防。

**关键词:**软件;失信因子;系统动力学

**中图分类号:**TP31

**文献标识码:**A

**文章编号:**1673-629X(2011)12-0059-04

## Research of Discredit Factors in Software Projects

SUN Le-ping, ZHAO Jia-bao

(School of Management and Engineering, Nanjing University, Nanjing 210093, China)

**Abstract:** With the rapid expansion of the scale of software development, software programs are becoming more complex, and the probability of failures in present software development is higher and higher. Lacking fundamental understanding of software process is the major cause of this. In the software development projects, schedule, cost and quality are three core issues. A variety of factors led to these problems. Each software project is a dynamic software development process, without such an understanding the likelihood of any significant gain in management of software development is questionable. In this paper, called these factors discredit factors, and recognized them according to software life cycle. Finally proposed a method to analyze discredit factors based on system dynamics. Through this study, people will pay more attention to discredit factors and do more research. Then they can effectively manage software projects.

**Key words:** software; discredit factors; system dynamics

## 0 引言

软件项目管理是使软件项目能够按照预定的进度、成本、质量顺利完成,人们对其重视程度越来越高<sup>[1-3]</sup>,也做了很多意义重大的研究成果,像在软件质量这块,研究的比较热的是软件缺陷,人们开发出了很多软件缺陷预测模型,通过预测模型对所开发的软件缺陷数进行预测<sup>[4,5]</sup>,由此判断所开发的软件是否达到要求。但在实际的软件项目中,仍然是存在着进度落后、成本超出预算、质量不达标等严重问题,究其原因,是因为人们仅仅是对结果进行了预测,而并没有对导致这个结果发生的因素进行研究,文中将这些因素统称为失信因子,只有整理出这些失信因子,对它们进

行相关的一些分析,进行有效地控制,这样才能加快软件开发速度、合理控制成本、提高软件质量。

## 1 失信因子识别

### 1.1 需求分析阶段

需求分析阶段是软件开发的第一个时期,是保证软件质量的第一步,这一阶段的主要任务是:用户和软件人员共同来充分理解用户的需求,并把需求明确的表达成一份书面材料。

在这一阶段要关注的失信因子有:

(1) 需求的清晰程度。

(2) 需求的变化程度。

(3) 需求涉及到的规范和文档,如在完成需求规格说明书时,书写错误或者书写格式、内容不规范。

(4) 需求是否可执行,如将需求分析转换为数据流程图时出错。

(5) 需求人员之间的交流,如用户未能正确表述其问题;软件人员未与一线需求人员交流,收集问题的

收稿日期:2011-04-18;修回日期:2011-07-21

基金项目:国家自然科学基金(90718036);江苏省自然科学基金(BK2009232)

作者简介:孙乐平(1986-),男,安徽黄山人,硕士研究生,研究方向为可信软件、系统工程;赵佳宝,博士,副教授,硕士生导师,研究方向为数据挖掘、管理控制一体化。

手段、方法不恰当;用户与软件人员具有不同的知识背景,处于不同的角度,造成相互之间交流困难;用户内部各类人员对问题的陈述可能是不完整的,也可能是矛盾的;软件人员没有良好的概括能力、分析能力和社交活动能力,没有一定的开发软件系统的经验,了解的用户的需求不全面或没有抓住关键。

## 1.2 设计阶段

充分了解了用户的需求之后,就进入了设计阶段,此阶段的主要任务就是在软件需求规格说明书的基础上,设计出满足需求分析的软件系统结构,即系统各个功能模块。设计阶段应尽量考虑各种可能的设计方案,对每个方案的质量进行全面评价,根据项目要求选择较好的一个方案。

要关注的失信因子有:

(1)设计的清晰性及合理性,如设计方案的选择;数据结构设计不合理;数据库设计时模型的选择;详细设计时,算法过程的设计;程序逻辑结构设计不合理或出错。

(2)系统的模块结构关系,如系统模块划分;各模块功能设计;模块之间调用关系;模块之间传递的数据关系。

(3)系统设计是否满足软件需求,如将需求分析阶段数据流图转化为软件结构出错。

(4)可测试性和可追溯性,如总体设计文档编写错误;人机界面设计时未形成良好一致性;用户操作步骤设计过多,不合理,用户不能很方便地完成操作步骤;设计时用户进行输入操作后无反馈信息,导致用户操作出错;设计时未提供撤销操作,导致用户出错时不能纠正;设计时未考虑系统容错性、鲁棒性,导致用户出错后系统不能进行误操作处理;未提供联机参考资料,难学,用户使用出现问题时无从参考,导致操作出错。

## 1.3 编码阶段

编码阶段的任务是按模块说明书的要求为每个模块编写程序,每个模块都由某种编程语言实现。编程语言的选择受多种因素的影响,比如说可移植性、语言必须要适合所编写的模块、语言编译器或解释器的质量,以及使用不同语言编写同一模块时,必须具有兼容的编译器,且编译后的代码具有合适的接口。

在这一阶段要关注的失信因子有:

(1)需求分析合理的程度。

(2)设计合理的程度,如编程语言选择;数据结构的错误设定;程序的参数错误;函数错误。

(3)规范,如其它项目中使用的约定不明;处理不当的决定和遵循步骤;指定了错误数目的重复循环;重用现有的代码;省略判断或判断遗漏。

(4)开发能力,如开发人员掌握的开发过程的知识;项目平台的经验;开发环境的经验和熟悉程度;开发工具和语言的经验和编写;不良编程习惯;拼写错误;代码设计人员本身水平问题,未能完成设计时的功能;代码编译人员未能正确理解设计人员的设计意图;错误标志的使用;标签定义遗漏;使用了错误的标签;字符或整数转换错误;内部故障(如语法等);接口故障(与其它系统组件的交互)。

(5)领域知识,如开发人员理解架构和解决相关风险的程度;类似应用开发上的经验;开发人员拥有的产品测试业务方面的知识;开发人员拥有的关于这个产品领域内的知识;开发人员在用户环境和操作方面的经验;应用环境的知识;应用经验。

(6)编码人员的交流,如项目组内部的交流;项目组之间的交流。

## 1.4 测试阶段

测试阶段的工作量约占开发期的一半,测试一个大型系统的难度,有时比设计这个系统还要大。前面的阶段可能产生各种类型的错误,测试阶段的任务就是发现并排除这些错误。该阶段应注意的原则有:避免由程序作者本人来测试;测试集应该由输入数据和预期的输出结果组成;既要选用合理的输入数据作为测试集,也要选用不合理的输入数据作为测试集;既要检查程序是否完成了它应该做的工作,也要检查它是否做了不该做的事;应该保留测试数据。

在测试阶段要关注的失信因子有:

(1)测试涉及到的计划的明确程度,如测试什么,何时何地该由谁来测试;为测试计划、测试规范和设计所作的假设的程度;计划、监测和控制测试活动的的能力;测试计划方法的可用性;测试计划和管理中的精度指标;用来支持测试计划和管理的历史数据的可用性;建立和遵守测试计划。

(2)测试涉及到的规范的清晰程度,如采用严格的需求分析方法;进行严格的配置管理;测试步骤选择错误;技术要求规范;测试计划的规范性;程序严谨的审查过程。

(3)测试阶段所涉及人员的水平,如审查员没有经过文件层次结构审查的培训;关于整合计划的不正确假设(延迟或被迫跳过测试);测试人员的心理压力承受度;测试人员的态度和承诺;测试团队的凝聚力和关系;测试团队的连续性和稳定性;测试人员提供测试活动的准确估计的能力;测试团队有足够的人员;测试工具的经验;测试团队的经验和对应用的熟悉度;测试人员对要测试的系统的了解;测试人员对要测试的产品的业务知识的了解;测试人员的技能和技能水平;测试团队的经验和专业知识;测试人员关于测试技术、方

法、策略、测试设计和测试计划等方面的知识;测试人员有关测试过程及与测试相关的开发过程的知识;测试人员的职业道德问题,出于某种目的,未能及时指出某些问题;对从测试团队之外获得的测试代码的解释;缓慢的沟通和变化的分布;被其他人或分项目隐藏的信息;在何种程度上的测试结果传达给直接利益相关者;测试团队以及团队之外各方的交流;测试者和开发者之间交流的程度;没有及时访问其他项目执行情况;调试时不能将测试时暴露出的问题完全改正;调试时改正现有错误后却引进了新的故障;未修复的缺陷没有通报其他项目。

(4)测试工具的使用程度,如合适的测试工具;没有足够的测试环境,为其它项目产生的对象使用(例如过于昂贵);项目之间测试环境的差异性;在其它项目中使用的自动化测试工具不可用。

(5)测试用例选择的恰当程度,如“低概率”现象;检测到,修改后,成为“低概率”;测试用例不完整;测试方法选择不正确;单元测试时测试驱动程序选择不恰当;集成测试时策略选择不当,或没有使用自顶向下及自底向上的混合策略;在测试时,未覆盖所有程序语句;单元测试时代码审查有问题,并未查出大部分逻辑错误和编码错误;调试策略选择不正确。

(6)测试中组织管理的水平,如测试进度报告;其它项目未报告的问题阻碍了适当的集成测试;没有相关项目的利益者参与规范审查;对其它项目工作人员检测有效性的不公正信任;错误的利益相关者参与了验收测试的执行;整体项目组织的复杂性;高级管理的管理风格;在测试中明确责任;明确测试所界定的组织;测试团队的组织;参与测试的测试小组的数量;测试团队的大小;依靠几个关键人员;群体、分项工程及有关人士之间的文化差异;组织的成熟度;集成测试的责任没有明确定义;承包商对测试工作的支持程度;用户参与测试活动的程度;客户端支持测试工作的程度;最终用户支持测试工作的程度。

## 2 失信因子分析

软件开发过程是一个动态的复杂系统,失信因子之间的关系也是错综复杂的,它们会随着项目的进展

不断变化,所以分析失信因子最好的方法就是采用动态建模,这样,随着活动的发生,通过过程模型就可以观察资源和产品发生了什么情况。这种动态过程的视图能够模拟过程,研究失信因子对整个项目的影响,并在实际消耗资源之前能够进行修改。系统动力学研究的问题是动态的,它能处理高阶次、非线性、多重反馈复杂时变系统的问题<sup>[6]</sup>,能对系统进行动态模拟,以观察系统在不同的组织形态、不同的参数、不同的政策因素输入时所表现的行为和趋势。基于上述考虑,可以结合系统动力学与软件过程的知识,来对失信因子做一些相关研究。

系统动力学究竟如何运用其中呢,可以分析一下软件开发过程的一些因子是如何影响生产率的<sup>[7,8]</sup>。可以构建包括开发人员时间在内的各种活动的描述性模型,然后考察模型中失信因子的变化是怎样增加或减少需求分析、设计、编写和测试代码所用的时间的<sup>[9-11]</sup>。首先,必须确定哪些因子对总生产率有影响。图 1 所示为一简化的影响生产率的因子模型。

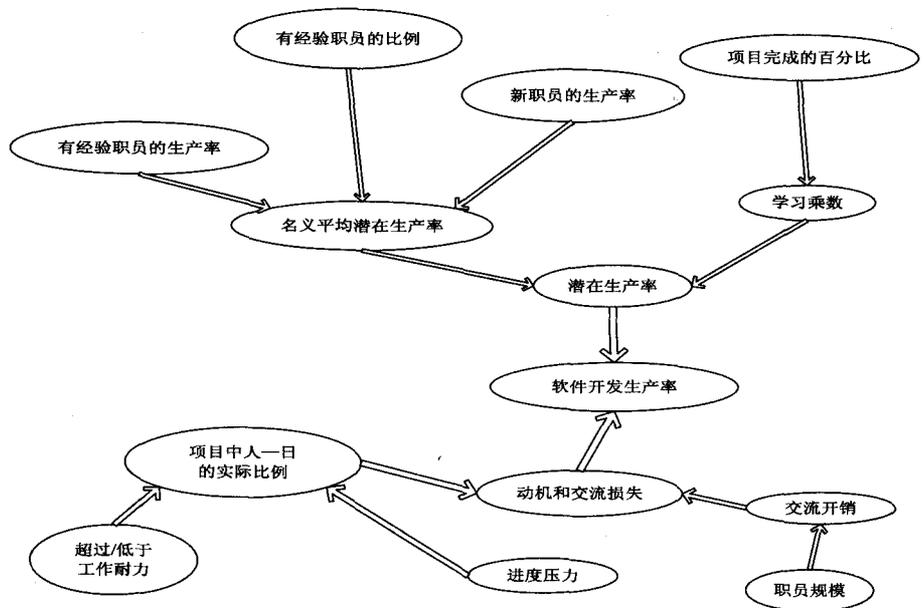


图 1 影响生产率的因子模型

箭头表明一个因素中的变化是如何影响另一个因素中的变化的。例如,如果在分配给项目的人员中,有经验的职员所占比例从 1/4 增加到 1/2,那么,预期平均生产率也会有所提高。同时,职员所占的比例越大(反映在职员规模上),那么用于项目团队成员之间交流的时间就越多(交流开销)。

从图中可以得知,名义平均潜在生产率(名义潜在生产率代表在同一开发阶段中不变的、与开发任务和资源有关的因素所决定的最大生产率)受三种因素影响:有经验职员的生产率、有经验职员的比例、新职员的生产率。同时,在整个项目进行过程中,新职员必须一步步了解这个项目,项目完成的部分越多,则新职

员在他们能够成为团队中高生产率的成员之前,就必须了解的越多。其他问题也会对软件开发生产率产生影响,首先,必须考虑每个开发人员每天对项目贡献所占的比例,当然进度带来的压力会影响这个比例,开发人员对工作量的承受力也会影响这个比例,职员规模会影响生产率,但是职员越多,项目团队成员用于交流信息的时间就越多。交流、动机及生产潜力合在一起给出了一种概括的软件开发生产率关系。

因此,使用系统动力学方法的第一步是将实证性证据、研究报告和直觉这三者结合在一起来标识这些要考察的失信因子的关系。第二步是量化这些关系,量化可以包含直接的关系。如职员规模和交流之间的关系。如果给一个项目分配  $n$  个人,那么可能有  $\frac{n(n-1)}{2}$  对人员必须彼此交流和协调。然而对某些关系,尤其是那些涉及随时间变化的资源的关系,必须进行一些分配来描述资源的增加和减少。例如,在一个项目中,很少会出现每个人都在第一天开始工作的情况。需求分析人员首先开始工作,当大量需求和设计构件文档化之后,编程人员才加入到项目中。因此,这种分配就描述了资源的增加和减少。

这种分析方法是站在一个较高的层面上来研究项目管理过程,可以灵活地把与某一策略相关的各种失信因子、影响综合起来进行研究。通过对不同的情形进行模拟,可以帮助相关人员对未来事件形成较为清晰和现实的认识,从而对软件项目进行有效控制和预防<sup>[12-18]</sup>。

### 3 结束语

文中针对软件项目,提出了失信因子的概念,对软件开发过程中的一些失信因子进行了识别,在以后的工作中,相关人员可以重点关注这些因子并对其进行深入研究。最后提出了一种基于系统动力学模型的失信因子分析方法,系统动力学模型的强大功能是显而易见的,这种分析方法在软件项目相关领域值得深入研究,但是也要谨慎使用这种方法,模拟的结果依赖于量化的关系,但量化的关系常常是启发式的或含糊不清的,它不是明确基于实证性研究的。

#### 参考文献:

[1] 李友仁. 软件过程与软件质量分析[M]. 北京:电子工业出

版社,1987.

- [2] 苏秦,何进,张谏贤. 软件过程质量管理[M]. 北京:科学出版社,2008.
- [3] 林广艳,姚淑珍. 软件工程过程[M]. 北京:清华大学出版社,2009.
- [4] 吴超,许建平,陈丽容. 基于生命周期的软件缺陷预测技术[J]. 计算机工程与设计,2009,30(12):2956-2959.
- [5] 王青. 软件缺陷预测技术[J]. 软件学报,2008,19:1566-1577.
- [6] 贾仁安,丁荣华. 系统动力学:反馈动态性复杂分析[M]. 北京:高等教育出版社,2002.
- [7] Humphrey W S. PSP<sup>SM</sup> 软件工程师的自我改进过程[M]. 吴超英,译. 北京:人民邮电出版社,2006.
- [8] Madachy R J. Software process dynamics[M]. Hoboken, NJ: Wiley; Piscataway, NJ: IEEE Press, 2008.
- [9] 郑志明,马世龙,李未,等. 软件可信性动力学特征及其演化复杂性[J]. 中国科学 F 辑:信息科学,2009,39(9):946-950.
- [10] 栾雪剑. 基于系统动力学的软件过程评价[D]. 杭州:浙江大学,2006.
- [11] 崔晓明,马力. 软件项目进度控制方法研究[J]. 计算机工程与设计,2010,31(12):2754-2757.
- [12] 吴明晖. 基于系统动力学方法的软件过程建模与仿真[J]. 计算机时代,2007(9):1-3.
- [13] 何满辉,杨皎平. 基于系统动力学的软件项目进度管理[J]. 科技和产业,2007,7(5):11-13.
- [14] Abdel-Hamid T K. The Dynamics of Software Project Staffing: A System Dynamics Based Simulation Approach[J]. IEEE Transactions on Software Engineering, 1989, 15(2): 109-119.
- [15] Andersson C, Karlsson L, Nedstam J, et al. Understanding software process through system dynamics simulation; A case study engineering of computer-based systems, 2002[C]// Proceedings of Ninth Annual IEEE International Conference and Workshop. [s.l.]:[s.n.], 2002:41-48.
- [16] Stamboulis Y, Adamides E, Malakis T. A system-dynamics study of a resource-based approach to process development strategy[C]//Engineering Management Conference, 2002. IEMC '02. [s.l.]:[s.n.], 2002:419-424.
- [17] Kellner M I, Madachy R J, Raffo D M. Software Process Simulation Modeling: Why What How[J]. Journal of System and Software, 1999, 46:91-105.
- [18] Rodrigues A. The role of system dynamics in project management[J]. International Journal of Project Management, 1996, 14(4): 213-220.

## 《计算机技术与发展》来稿须知

- (1) 论文用 Word 排版,以电子邮件方式投至本刊电子信箱:ctad@vip.163.com;
- (2) 投稿时请写明详细通信地址、邮政编码、联系电话、Email 信箱等各项必备内容;
- (3) 获国家或省部自然科学基金或列入各类重要科技计划项目资助的可优先安排发表。