

基于 CMMI 的软件风险管理

王天青 潘金贵

(南京大学软件新技术国家重点实验室 南京大学计算机科学与技术系 南京210093)

摘要 大型软件项目在开发过程中经常会伴随一些风险,而这些风险往往会对项目造成不利的影响,因此需要对风险进行管理。在 CMMI 模型中,在严格定义级给出了一个软件风险管理的过程域,本文介绍的软件风险管理实现就是基于这个规范的。该实现把软件风险管理分成五个步骤:风险识别、风险评估、风险计划、风险跟踪和风险控制,进而对风险进行全面管理。

关键词 CMMI 软件风险管理,软件过程改进框架

CMMI Based Software Risk Management

WANG Tian-Qing PAN Jin-Gui

(National Key Laboratory of New Software Technology, Computer Science of Nanjing University, Nanjing 210093)

Abstract Large software projects will often be accompanied by some risks, which might have some adverse impacts on projects, so we need to manage risks. In Defined level of CMMI criterion, there is a process area of Risk Management, which is the base of our software risk management implementation. We divide Software Risk Management into five steps: Risk Identify, Risk Analyze, Risk Plan, Risk Track and Risk Control, thus we total manage the risks.

Keywords CMMI, Software risk management, Software process improvement framework

1 引言

Capers Jones 曾经说过:“明天的问题就是今天的风险。软件风险管理正越来越广泛地被应用,因为它被证明能够在风险变为问题之前得到有效的识别和消除”^[1]。

通常定义下的风险是指一种受到损失或者伤害的可能性。软件风险指的是潜在的还未发生的问题,这些问题发生时,往往会对项目的成本、进度和技术实施等方面造成不利的影响。由此可见,软件风险包含两个重要的特征:

① 不确定性——包含风险的事件可能发生也可能不发生,没有100%发生的风险。

② 影响性——如果风险变成了问题,就往往会对项目产生不利影响。

Peter Kulik 认为:“大型软件项目都会伴随一些风险,但是如果风险能够降低到一个可以接受的程度,这就是一个好的开始”^[2]。

目前,风险管理被认为是 IT 软件项目中减少失败的一种重要手段,这是因为:

① 有效的风险管理可以提前发现那些潜在的问题。

② 提前对风险制定对策,就可以在风险发生时迅速做出反应。

③ 有效的风险管理可以帮助项目经理抓住工作重点,将主要精力集中于重大风险,将工作方式从被动救火转变为主动防范。

2 CMMI 的软件风险管理

美国卡内基梅隆大学软件工程研究院(SEI)提出的能力度成熟模型(CMM)已经被软件业广泛使用。在使用的过程中 CMM 中也暴露了一些缺点。而 CMMI 正是在这样的背景下

开发出来的。CMMI 被看作是把各种 CMM 集成为一个系列的模型中。CMMI 的基础源模型包括:软件 CMM 2.0 版(草稿 C), EIA-731 系统工程,以及 IPD CMM (IPD) 0.98a 版。

2.1 CMMI 的五级模型简介

初始级 代表了以不可预测结果为特征的过程成熟度。过程包括了一些特别的方法、符号、工作和反应管理,成功主要取决于团队的技能。

已管理级 代表了以可重复项目执行为特征的过程成熟度。组织使用基本纪律进行需求管理、项目计划、项目监督和控制、供应商协议管理、产品和过程质量保证、配置管理,以及度量和分析。

严格定义级 代表了以组织内改进项目执行为特征的过程成熟度。强调级别2的关键过程域的前后一致的、项目级的纪律,以建立组织级的活动和实践。风险管理(检测、优先级,相关问题和意外的解决方案)作为一个独立的过程域被加入了进来。

定量管理级 代表了以改进组织性能为特征的过程成熟度。3级项目的历史结果可用来交替使用,在业务表现的竞争尺度(成本、质量、时间)方面的结果是可预测的。

优化级 代表了以可快速进行重新配置的组织性能,和定量的、持续的过程改进为特征的过程成熟度。

2.2 CMMI 的风险管理

风险管理是一个持续的、前瞻的过程。它是业务和技术管理过程的一个重要组成部分。

风险管理应当找出那些可能危害目标的因素。一个持续的风险管理实现必须有效地预见和减轻那些会对项目造成严重影响的风险。风险管理可以分成以下三个部分:

风险管理准备:① 确定风险来源和风险分类;② 确定风险参数;③ 建立风险管理的策略。识别和分析风险:① 识别

王天青 硕士研究生,研究方向为软件工程、软件过程改进等。潘金贵

教授,博士生导师,从事中间件、Agent 技术及多媒体远程教学系统等的

风险;② 评估、分类和区分风险的优先级。减轻风险:①确立风险减轻计划;② 执行风险减轻计划。

虽然风险管理过程的主要着重点是项目级上的,但是同样可以管理组织级的风险。

3 基于 CMMI 的软件风险管理的实现

3.1 SPIF 系统简介

SPIF (Software Process Improvement Framework) 是一

个基于 CMMI 的 Web 化的软件过程管理与改进系统,它采用 B/S 的系统架构,它提供了一组相关工具和知识库用来辅助对软件开发的各个过程进行管理与控制,软件工程过程小组可以通过对知识库的维护来实现组织级的过程改进活动。其结构框架如图1所示。

3.2 风险管理过程

SPIF 系统中项目管理工具中提供对风险管理过程的支持(如图1所示)。

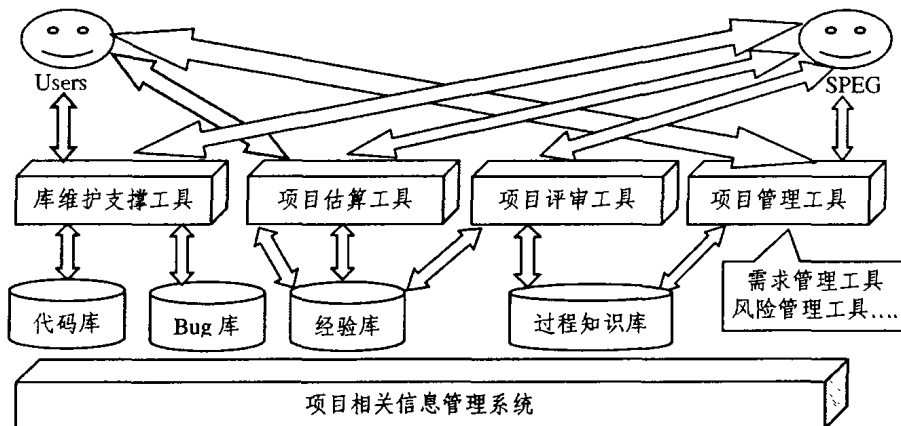


图1 SPIF 体系结构

3.2.1 风险形式化描述

定义1 风险可以用风险来源、风险类型、可能性、影响性、风险系数、阈值和当前状态信息来刻画。故可令 $R(isk) = (Source, Type, Probability, Effect, Coefficient, Threshold, Status)$ 。简记为: $R = (So, T, P, E, C, T, St)$ 。

定义2 风险可能性可以根据其发生的概率划分为五个等级:很高(≥ 0.8)、比较高($0.61 \sim 0.80$)、中等($0.41 \sim 0.60$)、比较低($0.21 \sim 0.4$)和很低(≤ 0.2)。可以量化为(5, 4, 3, 2, 1),即 $P = \{5, 4, 3, 2, 1\}$ 。

定义3 风险影响可以根据其产生的影响划分为五个等级:很高、比较高、中等、比较低和很低。可以量化为(5, 4, 3, 2, 1),即 $E = \{5, 4, 3, 2, 1\}$ 。

定义4 风险系数表明了风险的综合影响,定义 $C = P * E$ 。

定义5 风险状态 $St = \{S1, S2, S3, S4, S5\}$ 。

S1:新引入风险,即已经发现风险,但还未对其进行评估。

S2:对风险进行评估之后,发现风险系数低于阈值,表明该风险不是急迫的。

S3:对风险进行评估之后,发现风险系数高于阈值,表明该风险是急迫的。

S4:风险已经演化为问题,此时需要问题跟踪进行应对。

S5:通过风险应对,风险已经消除。

定义6 阈值:又称临界点,它用来表明某个风险是否可以接受,或者说是否急迫,当风险的风险系数高于阈值时,则表明该风险是急迫的,应当优先处理。计算方法如下:

$$T = \Sigma(C / N) + \delta$$

其中 C 为历史数据中相同类型风险在第一次风险评估时得到的风险系数, N 为同类型风险的个数, δ 为 SEPG (Software Engineering Process Group) 人员根据经验得出一个校正值得(默认为零)。

3.2.2 风险管理过程系统结构图 SPIF 风险管理过程由五个逻辑模块组成,分别为风险识别、风险分析、风险计划、

风险跟踪和风险控制。风险管理过程结构如图2所示。

各部分说明如下:

(1) 项目信息:包括项目名称、编号、客户等。

(2) 风险分类表:描述了风险的类型。

(3) 风险记录列表:每个风险都有一个从产生(发现风险)到消失(风险消除或者转化为问题)的生命周期。在风险生命周期的每一个阶段,都有一个风险记录与其对应。风险记录是对风险定义(定义1)的具体实现,包括:编号、识别日期、识别者姓名、风险标题、风险来源、风险类别、可能性、后果、风险系数、阈值、当前状态等。

(4) 风险应对计划:包括风险编号、应对策略、行动步骤、负责人、完成日期、批注等。

(5) 评估标准:描述了评估风险可能性和风险影响性的标准。

(6) 应对策略:包括接受、避免、保护、减少、研究、储备和转移等方法。

(7) 项目级风险数据库:存放了与项目相关的风险管理的信息,包括风险记录列表、风险应对计划等信息。

(8) 组织级风险数据库:存放了与整个组织有关的风险管理的信息,包括风险分类表、评估标准、应对策略等信息。

阈值参见定义6,风险状态参见定义5,详细的软件风险分类参见文[3]。

3.2.3 工作方式 风险识别:风险发现者填写风险来源和选择风险类型等相关信息(这时风险的当前状态为 S1)。

风险分析:系统根据历史数据给出该类型风险的平均可能性、平均影响性和阈值。然后 SQA 人员、SEPG 人员以及项目组成员通过讨论,根据以往经验得出该风险的可能性、影响性和阈值的校正值得 δ 。

系统根据风险可能性和风险影响性计算出风险系数,然后和阈值进行比较,设置风险的状态。接着系统更新项目级风险数据库和项目级风险数据库。

风险计划:SQA 人员、SEPG 人员和项目组成员,通过讨

(下转第155页)

为逐步建立小卫星综合测试床打好基础。

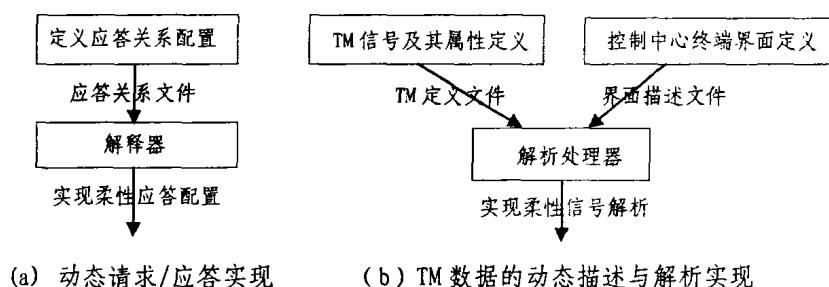


图3 综合模拟测试系统关键部分实现结构

参考文献

- 1 Jain R. the art of computer systems performance analysis: techniques for experimental design, measurement, simulation and modeling [M]. John Wiley & Sons Inc. 1991
- 2 骊山微电子研究所. 嵌入式系统综合设计技术[R]. 2002
- 3 Kono J, Santana C E. SCD1: One Year In Orbit. In: Proc. of International Symposium on Spacecraft Ground Control and Flight Dynamics[C], São José dos Campos, Brazil, Feb. 1994
- 4 Kono J, Santana C E. A Low Cost Portable Telemetry and Telecommand Station for the Brazilian Data Collecting Satellites. In: Proc. of the 10th Annual AIAA/USU Small Satellite Conf. [URL] <http://www.dea.inpe.br/papers/artmtt.html> 1996
- 5 Saturno M E, Alonso J D D. On-board computer data handling testing experiences for the second Brazilian data collecting satellite (SCD-2). [URL] <http://www.dea.inpe.br/papers/artesp1.html> 1996
- 6 Barry R M, Bakkes P J. Design of a Distributed Ground Support System for Small Satellites. SSC01-Viia-3. In: 15th AIAA/USU Conf. on Small Satellites [C], 2002
- 7 Groleau N, Kiser L. A Fully Implemented Semi-Automated Ground-Control System for the TERRIERS Satellite. [URL] <http://ssl.berkeley.edu/euve/sci/>, 1996

(上接第141页)

论得出风险应对策略、风险规避方法或缓解措施和行动步骤,同时确定风险应对计划的负责人。

风险跟踪:对于每一个风险,都有一个风险跟踪信息列表。对于风险信息的每一次更新,都会在风险跟踪信息列表中增加一条信息。

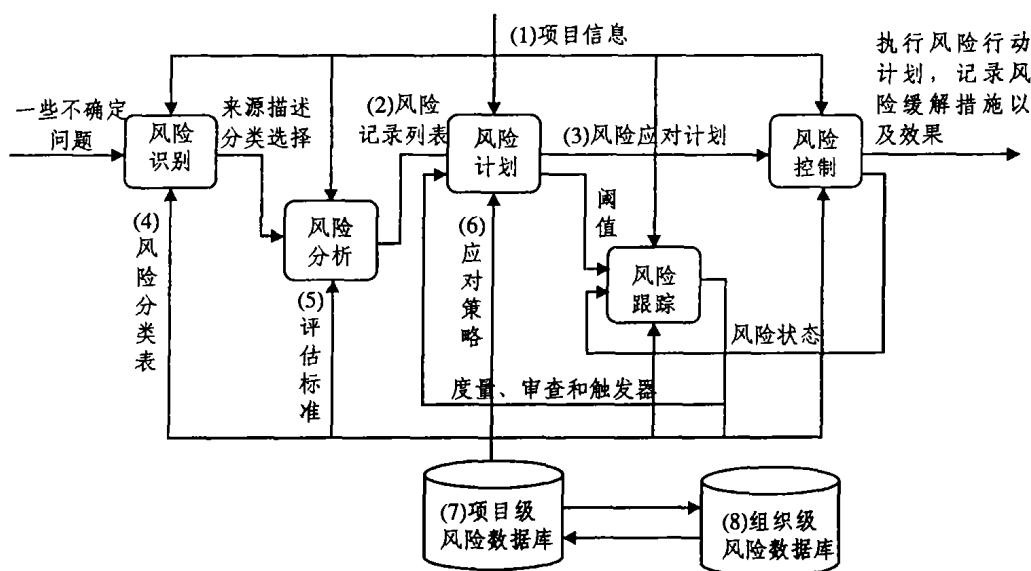


图2 风险管理过程系统结构图

每隔一段时间(触发器),或者风险计划执行完毕之后,相关人员对每一个现有的风险进行重新评估,得出每一个风险的当前可能性和当前影响性。然后系统重新计算风险系数,和阈值进行比较,得出风险当前状态。接着系统更新项目级风险数据库。

风险应对:相关人员在 SQA 的监督下执行风险应对计划。

进一步的工作 目前的风险评估主要由有经验的人员进行。因而需要一个合适的风险评估模型,包括风险可能性(或者是概率)模型和风险影响性模型。目前正在考察使用(进度,费用)这两项指标对风险影响性进行评估的模型。阈值目前是按照同类型风险的平均风险系数加上一个校正值。考虑到同一风险对不同类型的项目有不同的影响性,应当使用一个加

权平均而不是算术平均。这就需要对历史数据进行分析,以期找到加权系数。最后,柔性工作流^[4]整个系统提供更合理的工作流程,我们将在风险管理已有工作流程的基础上,引入柔性工作流。

参考文献

- 1 Capers J. Minimizing the Risks of Software, Cutter IT Journal: The Journal of Information Technology Management, 1998, 11 (4): 13~21
- 2 Peter K. What is Software Risk Management? (And why should I care?), 2000 <http://www.acceleraresearch.com/>
- 3 Carr M J, et al. Taxonomy-Based Risk Identification, SEI, 1993
- 4 范玉顺, 吴澄, 著. 一种提高系统柔性的工作流建模方法研究. 软件学报, 2002, 13(4): 833~839
- 5 CMMI 1.1规范 <http://www.sei.cmu.edu/cmmi/>
- 6 Higuera R P, Haimen Y Y. Software Risk Management. SEI 1996