

基于 TSP 风险评估的研究

吴鸣宇, 李伟生

(北京交通大学 计算机与信息技术学院, 北京 100044)

摘要:控制软件项目的风险是软件项目管理的重要组成部分。要控制软件项目的风险,首先需要进行软件风险评估,但是目前的软件风险评估方法存在着一些不足。对一些中小规模的软件开发来说,比如 TSP,复杂的、高成本的风险评估方法并不适合。因此,对于类似 TSP 的中小规模软件开发,需要一种相对简单实用的风险评估方法。文中根据 TSP 开发的特点,在分析已有风险评估方法的基础上,提出了一个基于概率分类权重求和的风险评估方法,并在几个中小项目开发中应用了该方法,为项目的风险管理提供了快速有效的风险评估数值。

关键词:群组软件过程;风险评估;基于概率分类权重求和方法

中图分类号: TP311

文献标识码: A

文章编号: 1673-629X(2006)05-0137-03

Research on Risk Evaluation TSP-Based

WU Ming-yu, LI Wei-sheng

(School of Computer and Information Technology, Beijing Jiaotong University, Beijing 100044, China)

Abstract: Controlling risk is a very important part in software-intensive project management. But there are some insufficiencies among the methods of risk evaluation at present. In fact, a complex and high-cost risk evaluation method is not suitable for some medium/small scale software development, including the TSP. Then it needs to design a simple and non high-cost risk evaluation method. Based on the features of TSP and the analysis of present methods, presents a new probability-based classified weighted sum method. This method has been used in several medium/small scale software development and provides effective risk data.

Key words: team software process; risk evaluation; probability-based classified weighted sum method

0 引言

软件风险是指软件开发过程中及软件产品本身可能造成的伤害或损失。风险关注未来的事情,这意味着,风险涉及选择及选择本身包含的不确定性。在软件开发过程及软件产品中都要面临各种决策的选择。风险是介于确定性和不确定性之间的状态,是处于无知和完整知识之间的状态。

当在软件工程领域考虑风险时,要关注以下的问题:什么样的风险会导致软件项目的彻底失败?用户需求、开发技术、目标计算机以及所有其它与项目有关的因素的改变将会对按时交付和总体成功产生什么影响?对于采用什么方法和工具,需要多少人员参与工作的问题,该如何选择和决策?对软件质量要达到什么程度才是“足够的”?因此,要对软件风险进行管理,第一步就是要进行风险识别,然后在风险识别的基础上进行风险评估。

风险识别是试图系统化地确定对项目计划(估算、进度、资源分配)存在的威胁,系统化地识别已知的和可预测

的风险^[1]。通过识别已知和可预测的风险,项目管理者就有可能避免这些风险,且当必要时控制这些风险。它的理论实质也就是有关知识、推断和搜索的理论。风险识别一般运用分解原则,将复杂的事物分解成比较简单的容易被认识的事物,将大系统分解成小系统,这也是符合人们分析问题、认识事物的规律的。在实践中,用于识别项目风险的方法有很多,目前常用的有:头脑风暴法(Brainstorming)、德尔菲法(Delphi Method)、情景分析法(Scenarios Analysis)等。识别出项目风险之后,就去要进行风险评估。常见的风险评估的方法有基于 BBN 的风险评估^[2];基于数据度量的风险评估;基于证据理论的风险评估^[3]等等。但是这些风险评估方法操作起来稍显复杂,对于 TSP 开发来说,并不是很适合。因此,文中根据 TSP 开发的特点,在分析已有风险评估方法的基础上,提出了一个基于概率的分类权重求和的风险评估方法。

1 基于 TSP 的风险评估

1.1 TSP 介绍

TSP(Team Software Process, 群组软件过程)^[4],它指导项目组中的成员如何有效地规划和管理所面临的项目开发任务,并且告诉管理人员如何指导软件开发队伍,始终以最佳状态来完成工作。TSP 实施集体管理与自己管

收稿日期:2005-08-31

作者简介:吴鸣宇(1980-),男,广东茂名,人,硕士研究生,研究方向为软件工程、计算机网络;李伟生,教授,研究领域为数据库、计算机网络、算法的设计与分析。

理自己相结合的原则,最终目的在于指导开发人员如何在最少的时间内,以预定的费用生产出高质量的软件产品,所采用的方法是对群组开发过程的定义、度量和改进。

和大型的软件项目开发相比,TSP 的风险评估方法应该具有相对快速灵活的特点。

1.2 基于概率的分类权重求和方法

各个层次间的风险关系的分析,有的层次的风险发生关系是串联的关系,有的层次的风险发生关系是并联的关系,它下层可能有好几个风险发生,那它们怎么影响到上层,就涉及到风险的加法和乘法问题。一个大型软件的开发大约存在 30 至 40 种风险。如果每种风险都需要 3 至 7 个风险管理步骤,那么风险管理本身也可以构成软件开发过程的一个子项目。这对 TSP 项目同样适用。根据 80-20 理论,即 80% 的风险是由 20% 常见的风险因素造成的。因此快速灵活的风险评估应该把重点放在这 20% 上面。

1.2.1 基于概率的分类权重求和方法

对开发过程中可能遇到的风险,可以按以下的分类来建立风险条目检查表:

- 产品规模——与要建造或要修改的软件的总体规模相关的风险。
- 商业影响——与管理或市场所加诸的约束相关的风险。
- 客户特性——与客户的素质以及开发者和客户定期通信的能力相关的风险。
- 过程定义——与软件过程被定义的程度以及它们被开发组织所遵守的程度相关的风险。
- 开发环境——与用以建造产品的工具的可用性及质量相关的风险。
- 建造的技术——与待开发软件的复杂性以及系统所包含技术的“新奇性”相关的风险。
- 人员数目及经验——与参与工作的软件工程师的总体技术水平及项目经验相关的风险。

根据项目的具体情况,可以请专家对软件项目的风险进行分析归类,对所有风险因素赋予一个表示风险影响程度的权值。权值为 10,表示影响最大,灾难性的;权值为 1,表示可忽略的;中间数值依次类推。另外,对每一个风险因素,给出一个发生概率。概率最大值总会小于 1,如果等于 1,就不能算是项目风险了,应该叫做项目约束了。

因此,对于一个特定的 TSP 项目来说,首先要进行风险识别,依靠经验识别出这个工程的主要和关键的风险。最好请专家来进行这一环节。如果必要的话,还可以对这些主要和关键风险因素进行分类。对每一类都规定一对 i -threshold 和 p -threshold。其中 i -threshold 表示该类风险因素的估计阈值。最简单的情况,就是对各个风险因素不分类,这时就只有一对 $\{i$ -threshold, p -threshold $\}$ 。当一个风险因素属于该类,并且其风险影响值大于 i -threshold、发生概率大于 p -threshold 时,才对这个风险

因素进行具体的评估。这样做,可以减少在开发基于 TSP 的项目的过程中的风险管理花费,节省人力和物力。这个方法对于基于 TSP 的项目开发尤为适用。

1.2.2 基本步骤

具体来说,基于概率分类权重求和方法应用的基本步骤如下:

1) 针对一个 TSP 项目,请专家对本项目进行风险的识别,把可能涉及的风险分成大类(class),类的数目不限,但一般说来,按风险条目表分成不多于 7 大类为佳。然后把可能的每一个风险归到大类中。

2) 对每一个风险,给出一个二元组 $\{i_j, p_j\}$,其中 j 表示风险的序号, i_j 表示其对项目的影响程度, p_j 表示其发生的概率。

3) 对每一个二元组,判断其影响程度和发生概率落在图 1 中的哪一个区域。

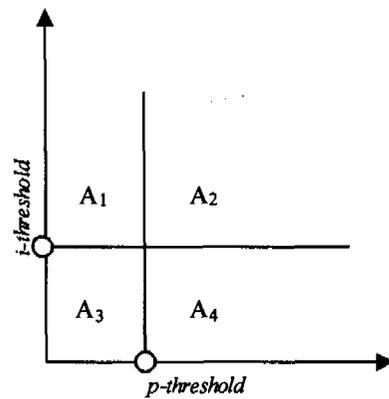


图 1 风险因素区域分布图

判断如下:

- $i_j > i$ -threshold, $p_j < p$ -threshold, $\{i_j, p_j\} \in A_1$
- $i_j > i$ -threshold, $p_j > p$ -threshold, $\{i_j, p_j\} \in A_2$
- $i_j < i$ -threshold, $p_j < p$ -threshold, $\{i_j, p_j\} \in A_3$
- $i_j < i$ -threshold, $p_j > p$ -threshold, $\{i_j, p_j\} \in A_4$

如果落在区域 A_1 和 A_3 ,可以不必进行处理;如果落在区域 A_2 和 A_4 ,就要进行相应的处理了。

4) 对落在 A_2 和 A_4 区域的风险因素,尤其是 A_2 区域的,要给予重视,因为这类风险不仅发生概率大,而且其对项目的影响也很大。对这个风险因素,如图 2 所示进一步分析其衍生的子风险因素 F_j ,识别出 F_j 的风险评估二元组 $\{i_j, p_j\}$ 。

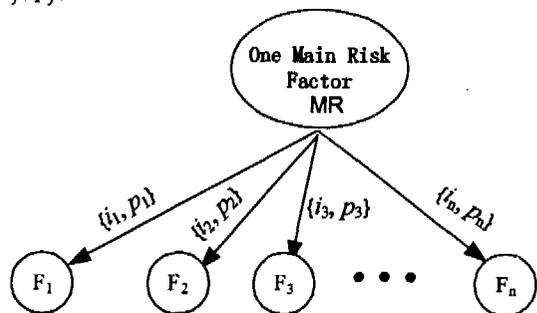


图 2 风险衍生分析图

5) 计算 MR 的风险值: $MR = \sum_{j=1}^n F_j = \sum_{j=1}^n i_j \times p_j$ 。这个 MR 值就是当前分析的这个风险的风险评估值。对其它主要风险因素也进行相似的处理, 得到多个 MR 值。将这些 MR 值相加, 即 $\sum MR$, 就是整个项目当前的风险评估值。

这个方法和基于 BBN 的风险评估方法相比起来, 更加简单, 计算方便。一般来说, 一个项目的主要风险不会是很多个的。对每一个主要风险, 利用公式 $MR = \sum_{j=1}^n F_j = \sum_{j=1}^n i_j \times p_j$ 计算其 MR 值。然后算出整个项目的风险评估值。可以针对具体项目制定 $MR - threshold$ 。如果 $\sum MR > MR - threshold$, 那么这个项目在启动之前就要考虑清楚了, 当前的时机是否适合进行这个项目的开发。因为当前存在的风险超过了项目的风险阈值。在软件风险分析中, 风险参考水平值存在一个点, 称为参考点或临界点, 在这个点上, 决定继续进行该项目或终止它(问题太多)都是可以接受的。

1.2.3 评估方法的操作流程

图 3 是整个评估方法的流程图。

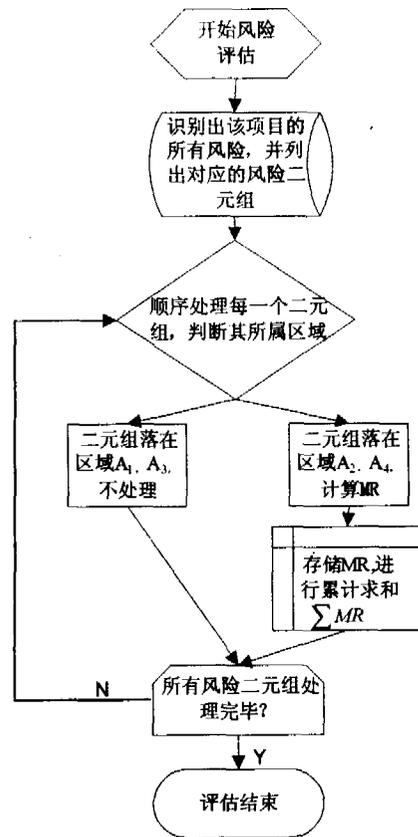


图 3 评估方法流程图

2 小 结

软件开发是一个受控迭代的过程^[5], 对于中小规模的软件开发的风险管理来说, 一个简单快速的风险评估方法是必要的。在实践中, 基于概率分类权重方法, 应用于几个中小项目开发, 为风险管理提供了快速有效的风险评估数值。对于群组软件过程来说, 基于概率分类权重求和的风险评估方法还是比较实用的。

参考文献:

[1] 文亚栋. 软件项目的风险管理[J]. 计算机系统应用, 2002, 17(2): 722-741.

[2] Fan Chin-Feng, Yu Yuan-Chang. BBN-based software project risk management[J]. The Journal of Systems and Software, 2004, 73: 193-203.

[3] 鞠彦兵, 冯允成, 姚李刚. 基于证据理论的软件开发风险评估方法[J]. 系统工程理论方法应用, 2003(9): 218-223.

[4] Humphrey W S. Introduction to the Team Software Process [M]. America: Addison Wesley Longman, Inc, 1999.

[5] 张南平, 曾昭毅, 钟 璐. 采用 RUP 的软件开发方法[J]. 微机发展, 2004, 14(1): 81-83.

(上接第 136 页)

中心实际情况的计算公式。

4 结 束 语

充分利用统计学原理和数据挖掘中的数据预处理原理, 针对呼叫中心的随机性数据预测难的问题, 给出一种预测呼叫放弃率的模型。它具有很强的实用性, 特别适用于小型呼叫中心, 在不增加设备的前提下, 最大限度地利用现有数据, 为管理者改善呼叫中心的运行机制提供有力的呼叫放弃率的预测结果, 从而有助于达到提高呼叫中心的的服务水平的目的。

参考文献:

[1] 郭晨东. 重新认识呼叫中心[DB/OL]. CTI 论坛, 2005. 7.

http://www. ctiforum. com/forum/2000/07/t0707. htm, 2005-07.

[2] 袁道唯. 呼叫中心的负荷管理与预测[DB/OL]. CTI 论坛, 2005. 7. http://www. ctiforum. com/resource/columns/yuandaowei/yuandaowei22. htm, 2005-07.

[3] 客户世界研究院. 2004 年中国呼叫中心发展年度报告[DB/OL]. 客户世界, 2004. 12. http://www. ccmworld. net/article/2764. html, 2004-12.

[4] 朱德通. 最优化模型与实验[M]. 上海: 同济大学出版社, 2003.

[5] 宋美娜, 段云峰, 宋俊德. 呼叫中心排队模型综述[J]. 中国数据通信, 2003(8): 12-14.

[6] Koole G. Call Centre Mathematics[A]. A scientific method for understanding and improving contact centers[C]. Amsterdam: [s. n.], 2005.