

软件项目风险评估方法应用探讨^{*}

张俊光¹, 吕廷杰¹, 马晓平²

(1. 北京邮电大学, 北京 100876; 2. 武汉大学, 湖北 武汉 430072)

摘 要: 对软件风险管理中非常关键的部分, 软件项目风险的评估进行分析, 研究了几种定性的且目前比较适用的软件项目风险评估方法。从理论上来说, 提出了一套从易到难进行研究的风险评估研究思路和方法; 从实践上来讲, 为软件组织进行项目的风险评估提供了一系列逐步改进的过程和方法, 体现了过程管理的思想, 对软件项目的开发具有指导价值, 对同行具有借鉴意义。

关键词: 风险评估; 可能性; 严重性; 优先级

中图法分类号: TP311. 5 文献标识码: A 文章编号: 1001-3695(2006)10-0076-02

Application Research on Evaluation Methods of Software Project Risk

ZHANG Jun-guang¹, LV Ting-jie¹, MA Xiao-ping²

(1. Beijing University of Posts & Telecommunications, Beijing 100876, China; 2. Wuhan University, Wuhan Hubei 430072, China)

Abstract: Analysis is implemented on software project risk evaluation which is the crucial part of software project risk management. Several software project risk evaluation methods that are both qualitative and applicable are studied. In terms of theory, a suit of risk evaluation study method and idea are put forwarded for study purpose from easy to difficult sequence; In terms of practice, a series of processes and methods are provided for continuous improvement of software organization's risk evaluation, which incarnates the idea of process management and has guidance value for project development. The outcome is instructive to the people devoted to the same industry.

Key words: Risk Estimate; Possibility; Impact; Priority

软件项目风险管理是软件开发过程中一项非常重要的工作, 是关系到软件项目成败与否的一个很重要的因素。软件项目风险管理主要包括项目风险的识别、评估、应对和控制四项主要活动, 即首先要识别出项目的风险, 评估风险之间的关系及风险发生的可能性、后果及影响范围; 根据评估的结果制定风险应对计划, 并对风险进行跟踪和控制。软件项目评估的主要目的是量化项目开发活动中可能发生的各类风险, 估计风险的可能性和对项目开发的影响程度, 划分风险的优先级, 为制定项目风险管理计划及对风险进行监控提供依据和参考。由此可见风险评估在软件项目风险管理中起着非常重要的作用, 是风险管理成败与否的关键。本文主要是按照由易到难, 由简单到复杂的过程, 对风险的可能性、严重性及优先级进行分析, 从而形成了各种不同的风险评估方法。从国内外的研究现状来看, 本文具有一定的实用创新性。

1 简易的定性风险评估

软件组织在刚刚开始进行项目风险管理时, 应本着先易后难, 逐步推进的思路, 先做起来, 再逐步地完善、细化及改进。刚开始时, 风险的可能性与严重性均只定义为高、中、低即可。

(1) 风险可能性: 指风险可能发生的程度。风险可能性根据风险在项目中发生的概率从高到底划分为三级, 如表 1 所示。

(2) 风险严重性: 指风险对项目造成的危害程度。风险严重性根据风险对项目工作量增加程度的大小分为三级, 如表 2 所示。

根据以上识别的风险的可能性及严重性, 可以定义出一个管理风险的优先级。

(3) 风险优先级: 指风险对项目造成的影响, 定义为风险严重性和风险可能性的一个综合值。风险优先级的确定如表 3 所示。

表 1 风险可能性

等级	等级说明
1	可能性高及极高
2	可能性中等
3	可能性低及极低

表 3 风险管理的优先级

影响	可能性	高	中	低
	高	高	高	中
	中	高	高	低
	低	中	低	低

表 2 风险严重性定级

等级	等级说明
高	风险影响项目承诺的实现, 必须由高级管理者参与处理
中	风险可能会影响项目承诺的实现, 但项目经理可以处理
低	风险不影响项目承诺的实现

注: 项目承诺指的是项目的交付日期、项目的需求实现及项目的费用与工作量情况等。

项目根据上述确定的风险优先级, 来确定相应的风险管理措施及不同措施所对应的措施负责人, 并据此对风险进行管理和监督。

2 具有数值依据的简易定性风险评估

对风险的评价主要有影响和可能性两类, 其中影响主要分为高、中、低三种。低影响是指对承诺没有影响的影响; 中影响

是指将影响到承诺,但项目负责人能够缓解,并不影响承诺的实现;高影响是指将影响到承诺,高级管理者将不得不介入来解决问题的影响。合理的影响通常以工作量来表示,是介入最坏的情况和最可能的情况之间。这种高、中、低的划分标准举例如表 4 所示。

表 4 风险严重性定级

风险严重性等级	权值	描 述
高	3	该风险所造成的影响占最可能工作量估计的比例大于 20%
中	2	该风险所造成的影响占最可能工作量估计的比例为 5% ~20%
低	1	该风险所造成的影响占最可能工作量估计的比例小于 5%

对风险的处理标准则必须参照一定的标准执行,举例如表 5 所示。

表 5 风险处理标准

影 响	可能性	措 施
低于 5% 的额外工作量	0 ~50%	不采取措施
低于 5% 的额外工作量	> 50%	确定缓解措施并执行,缓解的费用不超过额外的工作量,在周项目报告中报告缓解的进展情况
5% ~ 20% 的额外工作量	0 ~33%	确定缓解措施并执行,缓解的费用不超过 1 /4 的额外工作量,在周项目报告中报告缓解的进展情况
5% ~ 20% 的额外工作量	33% ~75%	确定缓解措施并执行,缓解的费用不超过 3 /4 的额外工作量,在周项目报告中报告缓解的进展情况
5% ~ 20% 的额外工作量	75% ~100%	确定缓解措施并执行,缓解的费用不超过额外工作量,在周项目报告中报告缓解的进展情况
> 20% 的额外工作量	0 ~30%	确定缓解措施并执行,缓解的费用不超过 3 /4 的额外工作量,在周项目报告中报告缓解的进展情况,每月向高级管理者报告,直到被缓解到影响或可能性不再属于这一类别
> 20% 的额外工作量	30% ~100%	确定缓解措施并执行,缓解的费用不超过额外工作量,在周项目报告中报告缓解的进展情况,每周向高级管理者报告,直到被缓解到影响或可能性不再属于这一类别

一旦风险的处理标准得到确定,则在进行风险策划及监控时,必须严格按上述标准执行。但管理层有特权关注那些低于极限值,特别是他们因为别的原因而仍然感到很重要的风险时,如一个对组织来说非常重要的客户的满意度等。

3 具有数值依据的定性风险评估

在这种方法体系下,三个参数的定义标准如下:

(1) 风险严重性。风险严重性根据风险对项目工作量增加程度的大小分为五级,举例如表 6 所示。

表 6 风险严重性定级

风险严重性等级	权值	描 述
很高	5	为缓解风险,附加工作量占原有工作量比例大于 20%
比较高	4	为缓解风险,附加工作量占原有工作量比例大于 15% ,小于 20%
中等	3	为缓解风险,附加工作量占原有工作量比例大于 10% ,小于 15%
比较低	2	为缓解风险,附加工作量占原有工作量比例大于 5% ,小于 10%
很低	1	为缓解风险,附加工作量占原有工作量比例小于 5%

需要说明的一点是,对于本文方法“2”及“3”中,风险严重性的确定标准,是以工作量作为确定标准来进行的。在项目具体进行风险评估时,还可以以进度及其他指标作为衡量的标准,而且这一标准的确立应以软件组织的实际情况为依据,没

有统一的标准。

(2) 风险可能性。风险可能性根据风险在项目中发生的概率从高到底划分为五级,举例如表 7 所示。

表 7 风险可能性定级

风险可能性等级	权值	描 述
很高	5	风险发生的几率为 80% ~100%
比较高	4	风险发生的几率为 60% ~80%
中等	3	风险发生的几率为 40% ~60%
比较低	2	风险发生的几率为 20% ~40%
很低	1	风险发生的几率低于 20%

(3) 风险系数。指风险对项目造成的影响,定义为风险严重性权值与风险可能性权值的乘积。风险系数如表 8 所示(呈梯形状)。

表 8 风险系数表

风 险 系 数		风 险 可 能 性				
		很高 5	比较高 4	中等 3	比较低 2	很低 1
风 险 严重性	很高 5	25	20	15	10	5
	比较高 4	20	16	12	8	4
	中等 3	15	12	9	6	3
	比较低 2	10	8	6	4	2
	很低 1	5	4	3	2	1

项目根据上面定义的风险参数,评估每个已识别风险的严重性、可能性计算风险系数,根据风险系数大小排列优先级。制定风险应对措施:项目制定项目内部风险管理的方法,如对于各级别风险的处理措施可如表 9 所示。

表 9 风险处理措施

风 险 系 数	响 应 措 施
风险系数 < 5	不响应
5 风险系数 10(浅色部分)	项目内部制定补救、缓解措施
10 风险系数(风险系数表中的深色部分)	制定补救、缓解措施并上报组织高级管理者,由高级管理者审批风险管理措施

4 其他定性及定量的评估方法

以上介绍的是三种相对较简单的定性风险评估方法,随着组织项目风险管理的实施,风险管理方法可不断地进行细化和量化,如在方法“3”中,可将可能性和严重性进一步细分为 10 个级别来进行分析;也可用其他的定量分析方法,如决策树分析、盈亏平衡分析及模拟法等。在进行风险评估时,如能收集组织项目风险的历史数据,据此来确定风险发生的影响及概率,类似项目可能发生的风险等,将使风险分析具有更大的意义和可行性,从而产生更大的效益。但通常情况下因风险较少发生,因此历史数据的收集具有一定的难度。

5 结束语

根据风险识别和评估的结果对风险的管理一般分为四个层次:危机管理(风险已经造成麻烦后才着手处理它们)、风险缓解(事先制定好风险发生后的补救措施,但不制定任何的防范措施)、着力预防(将风险识别和风险防范作为软件项目的一部分加以规划和执行)、消灭根源(识别和消灭可能产生风险的根源)。根据这四个层次的划分,我们应该根据风险评估的结果对风险进行预防,但同时我们也必须认识到不是所有的风险均能够预防,所以还必须建立一个应付意外事件的计划。

总之,软件项目风险识别出来后,对于风险的评估及拟定纠正措施就显得至关重要。一般来说,对于风险(下转第 81 页)

该程序在 VC6.0 下的运行结果是程序中止, 引发程序异常, 且无任何提示; 该程序在 Linux 下的运行结果是程序可以运行, 无输出, 且无任何提示。

不管在以上任何编译器下, 用户得到的结果均是错误且不可理解。在 TC 下 cp2 所指的串被修改了, 在 VC6.0 和 Linux 下程序运行无结果, 且无提示。由此说明在程序中存在错误, 仔细阅读和理解程序后, 发觉这一错误正是由于字符串数组的越界引起的。对于这样的程序在 BIST 系统中模板会插入强保护函数。考虑到本例相对来说比较简单, 因此对这一小程序没有进行测试点的设置, 相应部分可参考文献[2] 。

```
#include <stdio.h>
#include <string.h>
main()
{
    char* cp1,* cp2;
    cp1="12345";
    cp2="abcdefghij";
    BIST_strcpy( cp1, cp2);
    printf( "cp1 = %s\n cp2 = %s\n", cp1, cp2);
}
```

其中 BIST_strcpy(cp1, cp2) 是模板插入的函数, 被组织在 BIST_string 中。其基本结构如 3.1.1 中所示。具体对于这个程序来说代码如下:

```
BIST_strcpy( char * des, char * src)
{
    /* strlen( src) 和 strlen( des) 分别是 BIST 系统通过对源代码的扫描得到的常量, 此处这样写是为便于理解 */
    if( strlen( src) <= strlen( des) )
    {
        strcpy( des, src);
    }
    else
    {
        while( * des) * des++ = * src++;
        printf( "警告: 源串没有完全拷贝到目的串中, 发生了截断");
    }
}
```

插装后的程序在 TC2.0, VC6.0 以及 Linux 下的运行结果相同, 如下:

```
警告: 源串没有完全拷贝到目的串中, 发生了截断
cp1 = abcde
cp2 = abcdeghij
```

由此可以看出, 使用 BIST 系统不仅可以保证程序的正确性, 而且有较好的可移植性, 消除了不同编译器的细微差别。通过这个简单的例子, 演示了模板的使用, 同时验证了 BIST 思

想的可行性。

7 小结

在软件内建自测试中, 模板是系统的基石, 其内容关系到整个系统的性能和效果。本文主要探讨了根据故障模型的类型, 插入强保护函数、弱保护函数和检测函数; 还研究了针对特定故障模板的处理方式。借鉴了其他集成系统的特点以及结合软件 BIST 的特色对模板函数进行了组织。最后针对字符数组越界故障给出了一个示例并通过 BIST 系统进行测试和验证。

参考文献:

[1] 凌良合, 徐拾义. 基于软件内建自测试的模板和基准程序设计[J]. 同济大学学报(自然科学版), 2002, 30(10): 1159-1163.

[2] 徐拾义, 李文. 软件内建自测试中测试点的设计及实现[J]. 同济大学学报(自然科学版), 2004, 32(8): 1057-1060.

[3] 李文锋, 徐拾义. 软件内建自测试中的测试数据生成方法[C]. 北京: 第十届全国容错计算学术会议论文集, 2003. 263-266.

[4] 朱荣, 徐拾义. 软件测试中故障模型的建立[J]. 计算机工程与应用, 2003, 39(17): 69-71.

[5] 宫云战. 软件测试的故障模型[J]. 装甲兵工程学院学报, 2004, 18(2): 1-5.

[6] 孙昌爱, 金茂忠. 基于程序插装的动态测试技术实现[J]. 小型微型计算机系统, 2001, 22(12): 1475-1479.

[7] [美] Charles W, *et al.* Visual C++ 程序员实用大全[M]. 邓劲生, 张晓明. 北京: 中国水利水电出版社, 2001. 85.

[8] Wang Y, King G, Fatad M, *et al.* On Built-in Test Reuse in Object-Oriented Framework Design[J]. ACM Journal on Computing Survey, 2000, 32(1): 7-12.

[9] Wang Y, King G, Court I, *et al.* On Built-in Tests in Object-Oriented Reengineering[C]. Proceedings of the 5th ACM Symposium on FSE/ The 6th European Conference on Software Engineering/Workshop on Object-Oriented Reengineering(FSE/ESEC/WOOR '97), 1997.

[10] Erich Gamma, *et al.* 设计模式可复用面向对象软件的基础[M]. 李英军, 马晓星, 蔡敏. 北京: 机械工业出版社, 2000. 91-146.

[11] 陈意云. 编译原理和技术(第 2 版)[M]. 合肥: 中国科学技术大学出版社, 2002. 195-196.

作者简介:

彭成寒(1980-), 男, 河南南阳人, 硕士研究生, 主要研究方向为软件测试; 徐拾义(1941-), 男, 上海人, 教授, 主要研究方向为计算机容错、测试。

(上接第 77 页) 的评估与分析是一项既重要又比较难的工作。对于一个刚刚开始进行这项工作的组织来说, 必须要分阶段、分步骤, 本着先易后难, 先简单后逐渐复杂的过程来进行。因此, 刚开始的时候, 风险只定义为高、中、低即可。另外, 在根据评估结果对风险进行跟踪的时候, 一般认为缓解风险的费用低于风险发生所造成的损失, 才认为风险缓解是有效的, 因此刚开始时可只对较高层的风险进行跟踪。

参考文献:

[1] David Hillson. Extending the Risk Process to Manage Opportunities[J]. International Journal of Project Management, 2002, 20(3): 235-240.

[2] David Baccarini, Richard Archer. The Risk Ranking of Projects: A Thodology[J]. Intemational Journal of Project Management, 2001, 19: 139-145.

[3] Stephen Ward, Chris Chapman. Transforming Project Risk Management into Project Uncertainty Management[J]. International Journal of Project Management, 2003, 21(2): 97-105.

[4] T Raz, E Michael. Use and Benefits of Tools for Project Risk Management[J]. Intemational Journal of Project Management, 2001, 19(1): 9-17.

[5] Tom DeMarco, Timothy Lister. 与熊共舞: 软件项目风险管理[M]. 熊节, 马珊珊, 等. 北京: 清华大学出版社, 2004. 212-213.

[6] Bob Hughes, Mike Cotterell. 软件项目管理[M]. 周伯生, 廖彬山, 任爱华, 等. 北京: 机械工业出版社, 2004. 135-136.

[7] 王长峰. 重大研发(R&D) 项目过程管理综合集成与过程风险管理模式研究[D]. 合肥: 中国科学技术大学, 2004.

作者简介:

张俊光(1972-), 男, 博士研究生, 主要研究方向为软件项目管理、软件工程; 吕廷杰(1955-), 男, 教授, 博导, 主要研究方向为通信企业政策与电子商务。