

第二十五章 软件风险分析

上一章回顾

- 软件测试与软件质量
- 软件测试过程等级
- 软件测试过程改进
- 软件企业良好的软件测试过程

本章学习目标

- 能够运用软件风险分析的方法进行风险分析

本章学习方法

- 复习
- 记忆

本章主题

- 软件风险管理的概念
- 风险管理的重要性
- 软件项目中的风险
- 风险管理过程
- 经典风险管理理论

国家软件人才国际培训基地



软件风险管理的概念

- 软件项目风险是指在软件开发过程中遇到的预算和进度等方面的问题以及这些问题对软件项目的影响
- 软件风险管理：试图以一种可行的原则和实践，规范化地控制影响项目成功的风险

风险管理的重要性

- 有效的风险管理增加项目成功的机会，减少项目失败的概率
- 风险管理可以增加团队的稳固性
- 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。

软件项目中的风险

- 软件项目的风险主要来源于需求、技术、成本和进度。

需求风险

- 已经纳入基线的需求在继续变更
- 需求定义不准确,进一步的定义会扩展项目范畴
- 增加额外的需求
- 产品定义含混的部分比预期需要更多的时间
- 在做需求中客户参与不够
- 缺少有效的需求变化管理过程

计划编制风险

- 计划、资源和产品定义全凭客户或上层领导口头指令,并且不完全一致
- 计划是优化的,是"最佳状态",但计划不现实,只能算是"期望状态";
- 计划基于使用特定的小组成员,而那个特定的小组成员其实指望不上;
- 产品规模(代码行数、功能点、与前一产品规模的百分比)比估计的要大;
- 完成目标日期提前,但没有相应地调整产品范围或可用资源;
- 涉足不熟悉的产品领域,花费在设计和实现上的时间比预期的要多。

组织和管理风险

- 仅由管理层或市场人员进行技术决策,导致计划进度缓慢,计划时间延长;
- 低效的项目组结构降低生产率;
- 管理层审查决策的周期比预期的时间长;
- 预算削减,打乱项目计划;
- 管理层作出了打击项目组织积极性的决定;
- 缺乏必要的规范,导致工作失误与重复工作;
- 非技术的第三方的工作(预算批准、设备采购批准、法律方面的审查、安全保证等)时间比预期的延长。

人员风险

- 作为先决条件的任务(如培训及其他项目)不能按时完成;
- 开发人员和管理层之间关系不佳,导致决策缓慢,影响全局;
- 缺乏激励措施,士气低下,降低了生产能力;
- 某些人员需要更多的时间适应还不熟悉的软件工具和环境;
- 项目后期加入新的开发人员,需进行培训并逐渐与现有成员沟通,从而使现有成员的工作效率降低;
- 由于项目组成员之间发生冲突,导致沟通不畅、设计欠佳、接口出现错误和额外的重复工作;
- 不适应工作的成员没有调离项目组,影响了项目组其他成员的积极性;
- 没有找到项目急需的具有特定技能的人。

开发环境风险

- 设施未及时到位;
- 设施虽到位,但不配套,如没有电话、网线、办公用品等;
- 设施拥挤、杂乱或者破损;
- 开发工具未及时到位;
- 开发工具不如期望的那样有效,开发人员需要时间创建工作环境或者切换新的工具;
- 新的开发工具的学习期比预期的长,内容繁多。

客户风险

- 客户对于最后交付的产品不满意,要求重新设计和重做;
- 客户的意见未被采纳,造成产品最终无法满足用户要求,因而必须重做;
- 客户对规划、原型和规格的审核决策周期比预期的要长;
- 客户没有或不能参与规划、原型和规格阶段的审核,导致需求不稳定和产品生产周期的变更;
- 客户答复的时间(如回答或澄清与需求相关问题的时间)比预期长;
- 客户提供的组件质量欠佳,导致额外的测试、设计和集成工作,以及额外的客户关系管理工作。

产品风险

- 矫正质量低下的不可接受的产品,需要比预期更多的测试、设计和实现工作;
- 开发额外的不需要的功能(镀金),延长了计划进度;
- 严格要求与现有系统兼容,需要进行比预期更多的测试、设计和实现工作;
- 要求与其他系统或不受本项目组控制的系统相连,导致无法预料的设计、实现和测试工作;
- 在不熟悉或未经检验的软件和硬件环境中运行所产生的未预料到的问题;
- 开发一种全新的模块将比预期花费更长的时间;
- 依赖正在开发中的技术将延长计划进度。

设计和实现风险

- 设计质量低下,导致重复设计;
- 一些必要的功能无法使用现有的代码和库实现,开发人员必须使用新的库或者自行开发新的功能;
- 代码和库质量低下,导致需要进行额外的测试,修正错误,或重新制作;
- 过高估计了增强型工具对计划进度的节省量;
- 分别开发的模块无法有效集成,需要重新设计或制作。

过程风险

- 大量的纸面工作导致进程比预期的慢;
- 前期的质量保证行为不真实,导致后期的重复工作;
- 太不正规(缺乏对软件开发策略和标准的遵循),导致沟通不足,质量欠佳,甚至需重新开发;
- 过于正规(教条地坚持软件开发策略和标准),导致过多耗时于无用的工作;
- 向管理层撰写进程报告占用开发人员的时间比预期的多;
- 风险管理粗心,导致未能发现重大的项目风险。

风险管理过程

- 风险识别
- 风险量化
- 风险应对计划制定
- 风险监控

国家软件人才国际培训基地



风险识别

- 内容
 - 确定风险的来源
 - 风险产生的条件
 - 描述风险特征
 - 影响本项目的风险
- 方法
 - 头脑风暴法
 - 面谈
 - Delphi
 - 核对表
 - SWOT

头脑风暴法(Brain-Storming)

- 无限制的自由联想和讨论的代名词，其目的在于产生新观念或激发创新设想。最早是精神病理学上的用语，指精神病患者的精神错乱状态而言的。
- 头脑风暴法是由美国创造学家A·F·奥斯本于1939年首次提出、1953年正式发表的一种激发性思维的方法。
- 头脑风暴法有可分为直接头脑风暴法（通常简称为头脑风暴法）和质疑头脑风暴法（也称反头脑风暴法）。前者是在专家群体决策尽可能激发创造性，产生尽可能多的设想的方法，后者则是对前者提出的设想、方案逐一质疑，分析其现实可行性的方法。

面谈

国家软件产业基地
无锡计划软件产业基地
国家软件人才国际培训基地
国家数字媒体技术产业化基地



Delphi法

- 德尔菲

- 德尔菲法(Delphi Method)系以一系列问卷向各类专家征询意见，依据所有专家对原问卷的答复再拟定下一份问卷，再次向各类专家征询意见，直到大多数专家的意见看法趋于一致才获至结论，此法预测通常较适合作长期预测。

Delphi步骤

- 成立一个团体委员确定问题及设计研究问卷。
- 选择专家匿名质问预测。
- 把质问的问题回答收回做成结果。
- 反复再预测，把问题重新修整再做预测，如此反复预测至少2次以上

核对表

- 核对表是基于以前类比项目信息及其他相关信息编制的风险识别核对图表。核对表一般按照风险来源排列。利用核对表进行风险识别的主要优点是快而简单，缺点是受到项目可比性的限制

SWOT技术

- 主要是用来分析组织内部的优势与劣势以及外部环境的机会与威胁，在市场营销管理中经常使用的功能强大的分析工作。
- SWOT 分析法的基本分析步骤
 - 分析环境因素
 - 构造SWOT 矩阵
 - 制定行动计划

风险量化

- 内容
 - 确定哪些事件需要制定应对措施
- 方法
 - 因子计算
 - PERT估计
 - 决策树分析
 - 风险模拟

因子计算

- 因子量化

PERT估计

- PERT (Project Evaluation and Review Technique)
 - PERT对各个项目活动的完成时间按三种不同情况估计：一个产品的期望规模，一个最低可能估计，一个最高可能估计。用这三个估计用来得到一个产品期望规模和标准偏差的Pert统计估计。Pert估计可得到代码行的期望值E，和标准偏差SD

风险应对计划制定

- 制定风险应对策略主要考虑以下四个方面的因素：
 - 可规避性
 - 可转移性
 - 可缓解性
 - 可接受性
- 接受，即不改变项目计划(或没有合适的策略应付风险)，而考虑发生后如何应对。
- 缓和，即将风险事件的概率或结果降低到一个可以接受的程度。
- 回避，即通过变更项目计划消除风险或风险的触发条件，使目标免受影响。
- 转移，即不消除风险，而是将项目风险的结果连同应对的权力转移给第三方(第三方应该知道这是风险并有承受能力)。

风险监控

- 方法
 - 核对表
 - 定期项目评估
 - 挣值分析
 - 挣值分析法又称偏差分析法，是一种分析目标实施与目标期望之间差异的方法。挣值法的优点是能同时判断项目预算和进度计划的执行情况，以预算和费用来衡量工程的进度

经典风险管理理论

- Boehm模型
- CRM模型
- Leavitt模型

国家软件人才国际培训基地
国家软件人才国际培训基地
国家软件人才国际培训基地
国家软件人才国际培训基地



Boehm模型

- 两步骤：风险评估和风险控制
- 核心是维护和更新十大风险列表
- 包括人员短缺、不切实际的工期和预算、不合时宜的需求、开发了错误的软件功能、开发了错误的用户界面、过高的非实质性能要求、接连不断的需求改变、可外购部件不足、外部已完成任务不及时、实时性能过低和计算机能力有限
 - 在软件项目开始时归纳出现在项目的十大风险列表，
 - 在项目的生命周期中定期召开会议去对列表进行更新、评比。
 - 十大风险列表是让高层经理的注意力集中在项目关键成功因素上的有效途径，可以有效地管理风险并由此减少高层的时间和精力

CRM模型（持续风险管理模型）

- 要不断地评估可能造成恶劣后果的因素
- 决定最迫切需要处理的风险
- 实现控制风险的策略
- 评测并确保风险策略实施的有效性
- CRM模型要求在项目生命期的所有阶段都关注风险识别和管理，五个步骤：风险识别、分析、计划、跟踪、控制。强调对风险管理的各个组成部分的沟通

Leavitt模型

- 将形成各种系统的组织划分为4个组成部分：任务、结构、角色和技术。
- 4个组成部分和软件开发的各因素很好地对应起来：角色覆盖了所有的项目参与者，例如软件用户、项目经理和设计人员等；结构表示项目组织和其他制度上的安排；技术则包括开发工具、方法、硬件软件平台；任务描述了项目的目标和预期结果。
- Leavitt模型的关键思路是：模型的各个组成部分是密切相关的，一个组成部分的变化会影响其他的组成部分，如果一个组成部分的状态和其他的状态不一致，就会造成比较严重的后果，并可能降低整个系统的性能。

总结

- 软件风险管理的概念
- 风险管理的重要性
- 软件项目中的风险
- 风险管理过程
- 经典风险管理理论

国家软件人才国际培训基地
国家数字媒体技术产业化基地

