软件测试风险识别与防控机制研究

孙 娟

（南京紫金数云信息技术有限公司，江苏 南京 211100）

摘 要 在软件研发全生命周期中，软件测试是保障产品质量的关键环节，但测试过程易受需求变更、资源约束、技术瓶颈等多因素影响，滋生各类风险，进而导致测试效率下降、成本超支甚至产品质量不达标。为有效规避上述问题，本文开展软件测试风险识别与防控机制研究。首先梳理软件测试与风险管理相关理论基础；其次构建包含风险分类、识别方法与指标体系的风险识别体系，在此基础上建立风险评估模型，明确评估维度、量化指标并确定风险优先级；最后设计涵盖总体框架、分阶段策略、针对性措施及动态监控预警的风险防控机制。研究成果旨在为软件测试工作提供参考，进而提升风险管控能力，保障测试工作高效有序推进。

关键词 软件测试；风险识别；风险评估；防控机制；风险管理

中图分类号：TP311.55 **文献标志码**：A **DOI**：10.3969/j.issn.2097-3365.2026.08.001

0 引言

随着信息技术迅猛发展，软件产品应用场景持续拓宽，覆盖日常办公、工业控制、金融核心等关键模块，用户对软件质量的要求也稳步提高，软件测试对研发流程的重要性越来越凸显。软件测试环节有需求分析、用例设计、测试执行、缺陷修复等多个步骤，容易被需求模糊、技术迭代、资源配置不当、跨团队协作不畅等内外部因素干扰，催生各类风险，最终引发测试周期滞后、成本增加或产品质量不达标等问题。多数企业做软件测试风险管控，普遍有识别不周全、评估有偏差、防控措施滞后等不足，无法匹配敏捷开发、DevOps 等新型研发模式需求。本文全面分析软件测试风险识别与防控体系，构建科学识别体系、评估模型及防控机制，实现测试风险全流程管控，以期为提升测试质量和效率、压缩研发成本、适配新型研发模式提供实践参考。

1 软件测试风险相关理论基础

1.1 软件测试核心概念与流程

软件测试即依照规定条件操作软件，找出错误、判定质量、检查是否符合设计要求的流程，核心方向是验证功能完整性、性能稳定性等关键指标，为质量改进提供凭据。这套标准化流程涵盖测试计划制定、需求分析、用例设计、环境搭建、测试执行、缺陷管理与回归测试、测试总结等环环相扣的环节，各环节分别发挥方向规划、边界明确、提供执行依据等核心作用[1]。

1.2 风险与风险管理理论

风险即特定环境中事件产生的不确定性及潜在负面效应。软件测试风险是测试流程里可能引发目标未达成、质量下降、成本超支或周期延误的不确定因素，有传播性、连锁性属性。风险管理理论核心流程为风险识别、评估、应对与监控形成的闭环管理，逐一完成风险梳理、量化分级、针对性防控、动态跟踪调整，守住风险底线。

2 软件测试风险识别体系构建

2.1 软件测试风险分类梳理

结合软件测试流程、项目规模及实际工作场景，将软件测试风险划分为四大类，分别为需求类风险、技术类风险、资源类风险与管理类风险。需求类风险主要源于需求不明确、需求频繁变更、需求文档存在歧义等，这类风险易导致测试范围偏差、测试用例返工；技术类风险包括测试环境搭建困难、测试工具适配性差、技术人员能力不足、自动化测试脚本开发受阻等，直接影响测试效率与测试覆盖度；资源类风险涉及人力、设备、时间等资源配置不合理，如核心测试人员流失、测试设备性能不足、测试周期压缩等，易造成测试工作停滞或质量下降；管理类风险则包括测试计划不完善、沟通协调不畅、缺陷管理不规范、团队协作效率低等，影响测试全流程的顺畅推进。各类风险相互关联、相互影响，任一风险的爆发都可能引发连锁反应，影响测试工作全局。

作者简介：孙娟（1981-），女，本科，工程师，研究方向：软件测试。

2.2 软件测试风险识别方法选择与应用

结合软件测试工作特点、项目类型及团队构成，选用适配的风险识别方法，涉及文献研究法、头脑风暴法、德尔菲法和故障树分析法。采用文献研究法，梳理国内外相关文献、行业报告及项目案例，梳理既有软件测试风险类型与识别经验，为风险识别筑牢理论根基；以头脑风暴法组织测试、开发、需求、产品相关人员开展集中研讨，通过思维碰撞充分挖掘潜在风险，实现风险识别无死角；德尔菲法邀约行业专家和资深测试工程师实施多轮匿名评审，慢慢收拢风险认知偏差，保障风险识别客观精准；故障树分析法以“测试失败”或“产品重大缺陷漏测”作为顶事件，从后往前排查诱发该事件的中间事件和基本事件，精准锁定风险源头和传播路径。在实际应用中，要依据项目各阶段调整搭配多种方法，项目起步阶段采用文献研究法和头脑风暴法，项目中期采用德尔菲法和故障树分析法，增强风险识别的完整性和精准性[2]。

2.3 软件测试风险识别指标体系构建

基于风险分类与识别方法，构建软件测试风险识别指标体系，明确各类风险对应的具体识别指标，为风险识别提供量化依据。具体指标体系如表1所示。

3 软件测试风险评估模型建立

3.1 风险评估核心维度与指标量化

软件测试风险评估的核心维度有风险发生概率和风险影响程度，两者一同判定风险的优先级。风险发生概率是特定风险在测试流程中出现的概率，需整合历史项目数据、团队经验及项目当前状态进行综合判断；风险影响程度指风险出现后给测试目标、测试质量、测试成本和测试周期带来的负面影响大小，可从影响范围、影响时长、损失大小等维度评判。采用李克特5级评分法量化指标，把风险发生概率、影响程度各划分为5个等级，1级代表最低级别，5级为最高等级，对应发生概率极高、影响极大，以加权求和方式算出各风险综合评分，按各指标对测试工作的重要程度设定权重，依托层次分析法，组织多名行业专家及资深项目负责人打分得出权重值，保障权重分配科学且合理[3]。

3.2 软件测试风险评估模型构建

参照风险评估核心维度和指标量化数值，打造多维度风险评估模型，风险识别指标体系作为模型输入，依靠量化评分计算各风险的综合风险值，综合风险值 = 风险发生概率评分 × 概率权重 + 风险影响程度评分 × 影响程度权重，按照综合风险值的大小，将风险归类为高、中、低3个等级，综合风险值≥4分归为高风险，2～3分判定为中风险，≤1分属低风险，该模型能实现各类风险的精准量化评估，为风险防控提供依据。

3.3 风险优先级排序方法

通过风险矩阵法给识别出的风险划分优先级，这个方法清晰明了、实操性强，贴合软件测试项目的风险管控，将风险发生概率作为横轴，风险影响程度作纵轴，搭建3×3风险矩阵，把横轴和纵轴全部分为3个等级：高、中、低，划分9个象限，归属于高概率—高影响象限的风险是最高优先级，需及时落实防控措施，防控风险爆发；落在高概率—低影响或低概率—高影响象限的风险，归为中优先级，需打造针对性防控方案并全程监控；低概率—低影响象限内的风险归为低优

表1 软件测试风险分类及特征表

风险类别	识别指标	指标说明
需求类风险	需求文档完整性、需求变更频率、需求理解偏差率	需求文档完整性指核心需求描述的全面程度；需求变更频率指测试阶段需求变更的次数；需求理解偏差率指测试人员与需求方对需求理解不一致的比例
技术类风险	测试环境搭建成功率、测试工具适配度、技术人员技能匹配度	测试环境搭建成功率指按要求完成测试环境搭建的比例；测试工具适配度指测试工具与被测软件的兼容程度；技术人员技能匹配度指测试人员具备的技能与测试需求的契合程度
资源类风险	人力资源缺口率、设备资源利用率、测试时间冗余度	人力资源缺口率指实际所需测试人员与现有人员的差值占比；设备资源利用率指测试设备的实际使用时长与可用时长的比例；测试时间冗余度指计划测试时间与预估最短测试时间的差值占比
管理类风险	测试计划完备率、跨部门沟通效率、缺陷修复及时率	测试计划完备率指测试计划中包含的关键要素（目标、范围、流程等）的完整程度；跨部门沟通效率指不同部门间信息传递与问题解决的平均时间；缺陷修复及时率指在规定时间内完成修复的缺陷占比

先级，可归入常规风险管理，降低防控资源浪费，为提升排序精准性，可结合项目核心目标微调风险矩阵。就时间敏感型项目而言，可适度上调“影响测试周期”类风险的权重，把防控资源都集中投去应对关键风险。

4 软件测试风险防控机制设计

4.1 风险防控机制总体框架

构建“事前预防—事中控制—事后改进”的全流程风险防控整体框架，事前预防阶段主打风险识别和评估，构建完备的识别体系与评估模型，先期识别潜在隐患，拟定预防办法；事中控制阶段核心推进风险动态监控和预警，实时跟进风险变动，即刻启动防控措施，制止风险扩散；事后改进阶段对风险防控效果做总结评估，总结经验教训，优化风险识别指标及防控方案，搭建风险管理闭环。

4.2 分阶段风险防控策略

围绕软件测试全流程各阶段核心任务和风险特征展开，制定分阶段风险防控举措，做到精准防控。测试计划阶段重点管控需求类风险与管理类风险，加大需求调研力度、召开多轮需求评审会、启用需求追溯矩阵，落实清晰明确的需求；同步优化测试计划，厘清各环节责任、资源配置与时间节点，防止计划缺漏。测试设计阶段紧盯技术类风险，提前检测测试工具和环境的适配程度，开展技术预研和可行性验证，给复杂测试场景配套专项技术方案；同步开展技术培训强化人员能力，把控测试设计质量，测试执行阶段重点管控资源类风险与缺陷管理风险，构建资源动态调配机制，结合测试进度和问题反馈调整人力、设备资源；规范缺陷提交、评审、修复与回归测试的操作流程，明确缺陷修复先后等级，杜绝缺陷积压；测试总结阶段对全流程风险防控效果做全面评估，总结经验教训，升级风险识别指标库与防控策略库，构建风险管理闭环，为后续项目筑牢基础[4]。

4.3 典型风险针对性防控措施

聚焦不同类型的典型风险，定制专属防控办法，应对需求频繁变更风险，构建需求变更管控流程，理清变更审批权限和流程，评估变动对测试工作的作用；针对测试环境搭建困难风险，提前梳理测试环境需求，构建通用测试环境模板，安排专业技术人员承担环境搭建和维护工作；聚焦人力资源缺口风险，预先落实人员安排，构建内部人员培训和外部人员储备体系；应对沟通协调不畅风险，制定定期沟通会议制度，构建跨部门信息共享平台，保证信息传递无阻。

4.4 风险动态监控与预警体系

构建风险动态监控指标体系，围绕风险识别指标敲定关键监控点，实时汇聚各风险识别指标的相关数据，如需求变更次数、测试环境故障频率、缺陷修复时长等，通过风险评估模型实时核算各风险的综合风险值，出具风险动态监控报表，设定分级风险预警临界值，按风险等级划定一级（高风险）、二级（中风险）、三级（低风险）预警阈值，某一风险的综合风险值触及对应预警阈值时，自动启动预警机制，以邮件、短信、项目管理平台弹窗等多种方式告知相关责任人。责任人收到预警后，要按要求时限分析风险变化原因，高风险 2 小时内完成，中风险 12 小时内完成，调整防控举措报送处理进展，每周按期召开风险复盘会议，总结监控和预警实绩，优化监控指标、预警阈值及防控手段，增强风险管控的及时性、有效性和适应性[5]。

5 结束语

软件测试风险管控是筑牢软件质量、提升研发效率的关键，也是适配新型研发模式的核心依托。本研究构建软件测试风险识别体系、风险评估模型、全流程风险防控机制，形成一套科学全面的软件测试风险管控方案。该方案明确了风险识别的具体指标和方法，实现风险精准评估及优先级排序，制定全流程分阶段防控策略和动态监控预警机制，可有效提高软件测试风险识别的全面性和评估的精准度，增强风险防控的针对性与及时性，协助企业解决测试周期延误、成本超支等问题。未来可进一步引入人工智能、大数据等技术，依托历史测试数据分析，提前识别风险并做智能评估；强化跨行业、跨项目的风险管控经验共享，推出标准化风险管控指引，为不同类型软件项目制定更匹配的风险管控方案，全面提升软件测试风险管理能力。

参考文献：

[1] 周逸宁，池志杰. 对软件项目管理里及风险评估的研究与探讨 [J]. 网络安全技术与应用，2022(02):63-64.
[2] 赵中芳，汪亦伦，苗森，等. 一种软件需求变化的测试风险识别方法及装置 :CN202310572602.3[P].2023-08-22.
[3] 许晓飞，李昊. 软件测试过程风险分析与预防探讨 [J]. 现代通信技术，2021(01):25-27.
[4] 段继鑫. 嵌入式软件回归测试的风险控制策略研究 [J]. 电子通信与计算机科学，2024,06(06):124-126.
[5] 甘晨. 探析软件工程中软件测试的重要性 [J]. 数字化用户，2019,25(12):99.