

测试演示文稿

软件测试流程示例

演示者：Alex · 日期：2026年7月13日

目录

01

测试计划 (Test Planning)

奠定测试工作的基础，明确目标、范围与资源配置，为整个测试周期制定科学的执行蓝图与管理规范。

02

测试设计 (Test Design)

将业务需求转化为可执行的测试用例，细化测试步骤、预期结果与覆盖场景，确保测试的完整性与针对性。

03

测试执行 (Test Execution)

严格按照测试用例执行测试，真实记录执行结果，及时发现、跟踪并管理软件缺陷，保障测试过程规范。

04

测试报告 (Test Reporting)

系统总结测试结果，量化评估软件质量指标，输出专业的测试报告，为产品上线、迭代优化提供关键决策依据。

测试计划

测试计划是整个测试活动的基石，它定义了“为什么测”、“测什么”和“怎么测”，为测试工作提供清晰的行动指南，确保测试能够有序、高效、全面地开展，最终保障产品交付质量。



01 明确测试目标

核心导向：验证软件是否满足需求规格中的功能、性能、安全及兼容性要求。

具体落地：覆盖功能完整性测试、系统性能压测、多端兼容性适配、安全漏洞扫描及用户体验评估，确保产品核心价值达标。



02 确定测试范围

正向界定：清晰罗列需覆盖的功能模块、核心业务流程及关键用户场景。

反向边界：明确本次不涉及的功能点或非重点模块，有效避免测试范围蔓延，防止资源浪费与工期延误。



03 安排测试资源

资源统筹：配置测试人力、服务器与终端硬件，部署缺陷管理与自动化框架。

进度管控：制定全周期时间表，明确各阶段起止时间与关键里程碑，保障测试资源按需到位，进度可控。

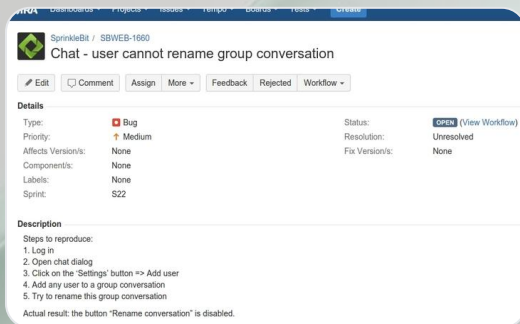
3. 测试执行

测试执行是将测试设计付诸实践的关键过程，不仅是发现软件缺陷的核心环节，更是验证软件功能完整性与质量可靠性的重要手段，直接决定了产品上线的信心。



3.1 执行测试用例

严格遵循测试用例的步骤逐项执行，实时记录每一步的执行结果，并准确标记测试状态（通过、失败、阻塞或跳过），确保测试过程可追溯、结果真实可信。



3.2 记录缺陷

发现异常时，在缺陷跟踪系统中创建报告，清晰描述复现步骤、预期结果与实际结果，并附上截图或日志，帮助开发团队快速定位与修复问题。



3.3 管理测试环境

确保测试环境的独立性、版本一致性和数据清洁，定期进行环境维护与更新，避免环境差异引入的误报，保障测试结果的准确性与可重复性。



感谢观看

Q & A

欢迎就软件测试流程与细节进行提问与深入交流