文章类型：论文|刊号（ISSN）：2737-4505(P) / 2737-4513(O)

# 软件测试中白盒测试方法的实际应用研究

罗宜欣
新疆大学
DOI:10.32629/etd.v7i6.21004

[摘　要] 面向电商平台登录模块的复杂控制流与数据流,研究以白盒测试为核心,分别在语句、分支、路径三个层面构建覆盖体系,并在控制器、服务与数据访问层进行插桩与故障注入。通过空值与异常输入构造、短路拆分、循环上界控制以及路径谓词求解,形成可追踪的用例集合与覆盖证据链。实验结果显示,语句覆盖率达到95%,分支覆盖率达到88%,路径覆盖率达到75%。共定位逻辑错误5项、边界错误3项、输入验证错误2项,相比同基线黑盒测试的6项缺陷,白盒测试在语句与路径层面显著提升了问题暴露与定位精度。研究进一步给出路径优先级与自动化执行策略,并提出将异常场景外显化以把路径覆盖率提升至85%的优化方向。
[关键词] 白盒测试；语句覆盖；分支覆盖；路径覆盖；电商登录模块
中图分类号：TD77+2　文献标识码：A

## Applied Research on White-Box Testing Method in Software Testing

Yixin Luo
Xinjiang University

[Abstract] Targeting the complex control flow and data flow of the e-commerce platform login module, this study takes white-box testing as the core, and constructs a coverage system at three levels: statement, branch and path. Instrumentation and fault injection are implemented in the controller layer, service layer and data access layer. By constructing null and abnormal inputs, short-circuit splitting, loop upper bound control, and path predicate solving, a traceable test case set and coverage evidence chain are formed. The experimental results show that the statement coverage reaches 95%, branch coverage 88%, and path coverage 75%. A total of 5 logical errors, 3 boundary errors and 2 input verification errors are identified. Compared with 6 defects detected by black-box testing under the same baseline, white-box testing significantly improves the defect exposure and localization accuracy at the statement and path levels. This study further proposes path priority and automated execution strategies, and puts forward an optimization direction of externalizing exception scenarios to increase path coverage to 85%.
[Key words] White-box Testing; Statement Coverage; Branch Coverage; Path Coverage; E-commerce Login Module

## 引言

电商平台的登录模块是账户鉴权与风控拦截的第一道关口,现实实现常包含验证码校验、行为风险评估、口令核验、账号状态判定、权限加载、失败计数与冻结、单点登录冲突处置及第三方绑定等多要素,多分支与跨组件协作导致控制流圈复杂度高、数据依赖与副作用路径复杂,黑盒方法难以触达异常通道与边界情形,也无法验证短路求值、重试折叠、故障降级下的内部一致性。为突破上述局限,本研究采用白盒视角,以覆盖为度量基准,通过语句、分支与路径三级覆盖体系将质量评价从外部行为迁移到可量化内部观测,目标是建立工程化可追踪实践,整合插桩采集、故障注入、时间门控与约束求解,形成输入到语句与路径的映射,发现外部难以显现的逻辑冲突与副作用不一致问题,为电商登录等关键模块可靠性提升提供方法论与执行框架。

## 1 白盒测试方法的理论基础

白盒测试的核心思路是把程序的内部结构作为直接观测对象,测试人员可以看到控制流和数据流的全貌,在此基础上设计测试用例并审查实现细节,其价值在于将缺陷定位精确到语句、分支乃至路径层面,把质量评价从“看结果对不对”推进到“看内部跑了哪里”的可量化层次。语句覆盖要求每条可执行语句

文章类型：论文|刊号（ISSN）：2737-4505(P) / 2737-4513(O)

至少被运行一次,目的是找出那些从未触及的代码实现;分支覆盖围绕判定节点的全部结果展开,要求真假两条路都必须走到,从而发现条件组合上的遗漏和短路求值时的误判；路径覆盖把入口到出口的可行路径作为检验单位,依赖控制流图同时限制循环展开次数来防止路径数量爆炸式增长。结合电商平台的用户登录模块,链路包含验证码以及行为检测、口令核验、账号状态以及权限判断,此外还涉及失败计数以及冻结、单点登录冲突处置以及第三方账号绑定这三个环节。进行白盒测试需要在控制器、服务以及数据访问层插桩,围绕空值输入、验证码过期、缓存失效以及数据库异常构造覆盖集[1]。

$$C_s=\frac{N_e}{N_t}\times 100\%$$

其中,$C_s$表示语句覆盖率(单位：%),$N_e$为被执行的可执行语句条数(单位：条),$N_t$为模块内可执行语句总条数(单位：条)。

## 2 白盒测试方法在电商登录模块的应用实践

2.1语句覆盖测试在电商登录模块的应用

语句覆盖测试的目的是确保登录链路中每条可执行语句都被实际运行到,是验证最小实现单元的基础手段。由于电商登录模块横跨控制器层、认证服务、缓存组件和数据访问层,本研究在控制流完全可见的前提下梳理代码并标注验证码校验、行为检测、口令核验等语句集合形成可执行清单(见图1)；围绕清单,将输入空间细化为空值、超长等七类具体情形并映射到对应语句,用例设计采用双向追踪方式,借助插桩工具记录每条语句的命中情况,一旦发现未覆盖的语句就回溯补充对应用例,执行阶段还对返回码、异常类型等关键断言加以约束,以便核查代码执行时产生的副作用是否符合预期。

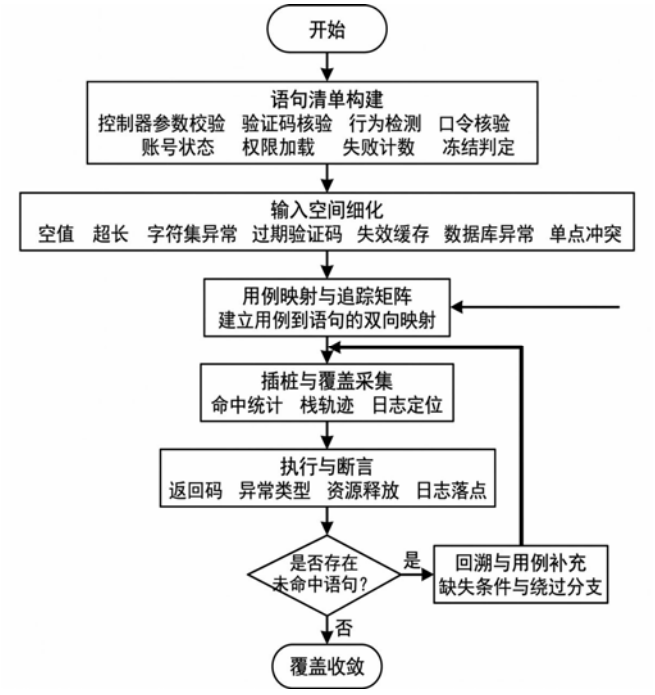


图1 电商登录模块语句覆盖测试执行流程图

执行过程中发现,控制器参数绑定处未对空账号字段进行早期拦截,这会给后续服务层留下了空指针的隐患；失败计数自增语句在异常通道未被触发,也会导致冻结阈值的统计出现偏差；缓存失联场景下的降级分支语句虽被触达,但缺少超时重试后的清理动作。针对上述发现,研究对输入构造方式和插桩点位置做了相应调整,同时让语句清单和用例库保持版本同步,确保覆盖结果的稳定性和可解释性[2]。

2.2分支覆盖测试在电商登录模块的应用

面向电商登录模块的分支覆盖,研究以判定节点(由布尔结果控制后续流向的代码位置)为度量单元,结合从控制器到数据访问层的完整实现,梳理出用户名非空判断、验证码有效期校验、行为检测得分阈值比较等十余个关键决策点；为了保证真假两条路都能被跑到,在构造输入值和环境状态时下了不少功夫,对于包含短路求值的复杂合取或析取条件,会按照短路特性将其拆分为若干可独立控制的子情形来分别驱动,例如验证码有效且行为风险低的合取用首项真次项假与真两组用例驱动,口令匹配或账号锁定的析取安排口令错误但账号可用及正确但冻结两类路径。为精确度量复合条件分支的内部逻辑严密性,本研究引入修正条件判定覆盖(MC/DC)思想。针对电商登录中诸如“验证码有效且(行为得分达标或通过二次验证)”的嵌套布尔表达式,常规分支覆盖易掩盖单一变量的独立影响。为此,研究在抽象语法树层面解耦复合节点,提取独立的原子谓词。构造用例时严格遵循控制单一变量原则,固定其他布尔算子状态,强制目标子条件的独立翻转能够唯一决定外层分支走向。此策略不仅规避了短路求值引发的安全校验绕过风险,还在未显著膨胀用例规模的前提下,精准暴露出因逻辑运算符优先级错位导致的风控失效隐患,深化了对代码内部逻辑一致性的验证。执行时依靠分支探针记录命中情况,对于还未覆盖到的分支边,通过回溯其前置条件来补充用例,同时借助时间控制手段来模拟验证码过期和缓存断联等异常触发场景,整体来看,发现的缺陷大多集中在分支级的行为一致性问题上,比如密码错误时日志没有记录、异常通道里失败次数没有累计等,研究对此维护了用例与分支的双向可追踪清单,以保持覆盖的长期稳定[3]。

2.3路径覆盖测试在电商登录模块的应用

立足路径覆盖以入口到出口可行执行序列为检验对象的特性,本研究围绕电商登录模块构建控制流图,梳理跨控制器、认证服务等组件的关键路径簇,涉及的核心路径包括正常登录成功、用户名不存在、口令不匹配、验证码过期、行为风险二次校验、账号冻结拦截、多因子追加校验、缓存降级重试、数据库异常返回、单点会话踢出及第三方账号绑定引导,对于失败计数更新和令牌签发过程中存在的循环重试逻辑,研究将循环次数设定了上界,并把结构相同的重试路径折叠为等价子路径,在不失一般性的前提下有效压缩了路径组合的规模,也让可行性判断的过程更加稳定。

在用例设计上,把每条路径编码为可观测边序列,同时把路

径谓词拆解为输入、状态与时间三类约束；借助符号约束求解与桩件配置来生成可控的系统状态,通过时间冻结与故障注入来构造验证码过期、服务断联的情境,在各层植入探针收集签名以此来比对覆盖完整性;从测试结果来看,路径级别的缺陷大多出现在多条件组合时的逻辑冲突和副作用不一致问题上,例如验证码刷新后绕过了风险校验、失败计数被重复累加、缓存降级时遗留了过期的评分数据、单点登录冲突时引入了僵尸令牌、第三方绑定流程未被计入失败次数、权限加载失败后没有回滚审计日志;为避免路径数量爆炸,研究采用了等价类合并和不可达路径裁剪的方式来控制候选规模,以基础路径集叠加关键跨模块长路径校阅覆盖闭合性,拆分短路求值为显式子路径触达未求值分量,常态化维护路径到用例的双向追踪清单,随着组件迭代同步调整探针和桩件配置,将覆盖稳定性和可解释性纳入版本基线管理[4]。

## 3 白盒测试应用效果与优化建议

### 3.1白盒测试应用效果评估

针对电商登录模块的实际代码结构与可观测控制流,本研究选取覆盖率与缺陷分布当作评估白盒测试成效的核心指标。统计结果表明,语句覆盖率达到95%,分支覆盖率达到88%,路径覆盖率达到75%。以缺陷类型为维度,白盒测试共定位逻辑错误5个、边界错误3个、输入验证错误2个,见表1。仔细分析各类缺陷的成因,逻辑错误大多数是由短路求值导致的分支长期不被求值、跨组件副作用时序不一致以及单点冲突撤销顺序异常引发；边界错误集中在失败计数自增的异常通道与重试上界处理；输入验证错误则主要涉及空账号的提前拦截和非法字符的处置逻辑。与在同一代码基线下开展的黑盒测试对照,黑盒仅发现缺陷6个,并且类型以表层提示与接口返回异常居多,由此可以推导出白盒测试把缺陷定位粒度推进到语句与路径层面,并且把难以凭外部行为显现的内部一致性问题纳入观测范围。

表1 电商登录模块缺陷类型分布表

| 缺陷类型 | 数量 | 主要触发位置 |
|---|---|---|
| 逻辑错误 | 5 | 短路分支未被求值 旧会话撤销时序 权限加载与审计写入不一致 |
| 边界错误 | 3 | 失败计数上界 重试上界 用户名长度与验证码长度边界 |
| 输入验证错误 | 2 | 空账号字段早期拦截 字符集异常输入校验 |

从缺陷分布与覆盖的映射关系看,语句覆盖的提高更有助于及早暴露输入验证类问题,分支与路径覆盖的扩展则把复杂条件组合与异常链路中的潜在冲突暴露出来。结合覆盖数据可以看到,较高的语句与分支覆盖促使常规通道得到充分触达,而75%的路径覆盖已囊括关键长链路与异常通道,剩余未覆盖的路径主要是那些被判定为不可达,或者因外部依赖仿真成本过高而暂时搁置的情形。需重点关注的是,覆盖率高并不等于缺陷一定发现得多,控制流的复杂程度和数据依赖的密度都会对边际收益产生明显影响,为此,本研究把覆盖到用例的追踪关系常态化维护,借助插桩证据来提升结论的可解释性与稳定性,同时也尽量把缺陷定位和修复的往返周期压缩在较短窗口内完成。

### 3.2白盒测试应用优化建议

由于路径覆盖在长链路和异常通道方面仍存在一定缺口,本研究把异常场景系统性外显为可执行用例集,目标把路径覆盖提升至85%。具体来说,围绕缓存断联、数据库超时、验证码服务降级、单点会话冲突以及权限载入失败,补充带时间门控与故障注入的组合用例,并把同构重试按等价类折叠,将循环次数上限和短路拆分情况明确记录为独立子路径,从而让候选路径的数量处于可控范围,也更容易被充分驱动测试[5]。为降低人力投入,建议在单元与服务层选用JUnit 5参数化与Mockito桩件完成白盒用例的自动执行,在界面与联机链路侧引入Selenium驱动登录页、二次校验与单点踢出流程,同时把覆盖采集探针集成为Jenkins流水线的构建步骤,设置覆盖阈值与变更范围差分执行以减少无效回归。

在路径优先级排序方面,可以把圈复杂度、历史缺陷热区以及外部依赖扇出作为综合排序依据,配合符号约束求解与数据工厂生成账号、口令与令牌的可控样本,其中符号约束求解把输入抽象为符号并求解满足路径谓词的取值,再叠加时钟冻结以稳定复现时序相关缺陷。为避免探针侵入引起时序扰动,可把分支与路径命中写入异步缓冲并进行剖面采样校准；而在经验迁移层面,建议把用例到语句、分支、路径的追踪清单与服务虚拟化配置纳入代码库版本管理,迁移至支付、注册与找回口令等同类模块时同步复用,并补充跨组件副作用校阅与日志一致性断言,形成可复制的工程化实践。

## 4 结语

研究以覆盖率为核心主线,为电商登录模块构建了一套完整的白盒测试方法,将语句、分支与路径三级度量贯通起来,配合插桩、故障注入与约束求解,形成了测试用例与代码实现之间的双向追踪闭环。结果显示,整体覆盖率处于较高水平,并切实发现了短路长期不求值、异常通道失败计数遗漏、单点冲突撤销时序不当以及缓存降级清理缺失等隐蔽一致性缺陷,验证了以内部代码结构为基础的测试策略在定位精度和可解释性方面的明显优势。同时也观察到覆盖与缺陷发现并非线性关系,控制流的复杂程度和数据依赖的密度对边际收益的影响不可忽视,这提示后续需要在路径裁剪、等价类折叠和不可达路径判定等方面持续优化。面向演进实践,建议将异常场景系统化沉淀为可执行套件,结合参数化与服务虚拟化降低维护成本,将覆盖阈值与差分执行纳入流水线以稳定回归效率,并以异步缓冲减轻探针对时序的扰动。未来工作将围绕长链路与外部依赖的可仿真能力、路径优先级自适应排序以及跨模块复用机制展开,力求在保持覆盖稳定性的同时进一步提升路径覆盖与缺陷捕获的综合收益。

## [参考文献]

[1]秦畅,陈赛,李坤.面向MC/DC覆盖的白盒单元测试用例自

动生成技术[J].科学技术与工程,2024,24(30):13039-13047.

[2]穆爽,张权,徐欣.一种基于白盒密码的Java源码防篡改方法[J].计算机应用与软件,2025,42(02):61-65+164.

[3]常振轩,郑之涵,梅傲寒.一种面向固件网络应用的高效灰盒模糊测试方法[J].信息网络安全,2025,25(04):654-663.

[4]徐邑江,高庆,陈立果.定向灰盒模糊测试技术研究进展[J].计算机学报,2025,48(05):1244-1272.

[5]安翔,黄健,李岱林.ATE在FPGA测试中的应用与优化研究[J].中国集成电路,2025,34(07):73-77.

**作者简介：**

罗宜欣(2005--),男,汉族,江西九江人,本科,研究方向：边缘计算、软件工程与智能物联网。