

标记	数量	修改单号	签字	日期

Zero-Defect Prediction 软件可靠性分析系统 测试报告 V1.0

(ZDP 软件配置项测试报告)

编制: _____

会签: _____

校对: _____

标检: _____

审核: _____

批准: _____

会签: _____

ZDP 软件项目组

目 次

1 范围 3

 1.1 标识 3

 1.2 系统概述 3

2 依据和引用文档 3

3 测试概述 3

 3.1 测试工作概述 3

 3.2 测试范围 4

4 测试实施情况 4

 4.1 测试实施时间和地点 4

 4.2 测试团队 5

 4.3 测试实施进程 5

5 测试数据分析 6

 5.1 测试用例执行结果 6

 5.2 缺陷趋势分析 6

 5.3 按缺陷严重程度统计 7

 5.4 性能测试结果 7

 5.5 兼容性测试结果 8

 5.6 代码覆盖率分析 8

6 测试结论 8

 6.1 测试结果 8

 6.1.1 文档审查测试 8

 6.1.2 功能测试 9

 6.1.3 接口测试 9

 6.1.4 边界测试 9

 6.1.5 性能测试 9

 6.1.6 安全性测试 9

 6.1.7 兼容性测试 9

 6.2 测试基本结论 10

7 遗留问题说明 10

8 后续工作建议 11

1 范围

1.1 标识

本文档为 ZDP（Zero-Defect Prediction，零缺陷预测）软件可靠性分析系统的测试报告，标识号为：ZDP-2025-001-TR。

1.2 系统概述

ZDP（Zero-Defect Prediction）软件可靠性分析系统是一款基于 Python 科学计算生态和 PySide6 桌面框架开发的软件可靠性分析工具。主要功能包括：

- 故障数据导入与预处理（CSV/Excel 格式，累计故障数/故障间隔 TBF）
- 8 种可靠性模型拟合（GO, JM, S-Shaped, BP, SVR, Hybrid, GM, Plugin）
- 可视化图表生成（预测曲线、残差图、U 图、Y 图）
- PDF 报告导出（支持中文字体）
- 实验管理（导出/导入 ZIP）
- 命令行工具（zdp-cli）
- 插件系统（entry_points 机制）

软件版本：v0.1.0

开发语言：Python 3.10+

运行平台：Windows 10/11

软件质量等级：通用级

2 依据和引用文档

1. GB/T 25000.51-2016《系统与软件工程 系统与软件质量要求和评价(SQuaRE) 第 51 部分：就绪可用软件产品(RUSP)的质量要求和测试细则》
2. GB/T 15532-2008《计算机软件测试规范》
3. ISO/IEC 25010:2011《系统与软件工程 系统与软件质量模型》
4. 《ZDP（Zero-Defect Prediction）软件需求规格说明》，2024.12，内部文档
5. 《ZDP（Zero-Defect Prediction）软件测试计划》，ZDP-2025-001-TP，2025.12.05
6. 《ZDP（Zero-Defect Prediction）软件测试说明》，ZDP-2025-001-TD，2025.12.10

3 测试概述

3.1 测试工作概述

本次测试类型包括文档审查、功能测试、接口测试、边界测试、性能测试、安全性测试、兼容性测试、可用性测试和回归测试。

测试统计数据：

- 测试周期：2025 年 12 月 02 日至 2025 年 12 月 27 日（18 个工作日）
- 测试轮次：4 轮（冒烟测试、全面测试、专项测试、回归测试）
- 设计测试用例：126 个
- 执行测试用例：126 个

- 复用测试用例：0 个（首次版本测试）
- 发现缺陷总数：42 个
 - Blocker：2 个（已修复）
 - Critical：5 个（已修复）
 - Major：12 个（已修复）
 - Minor：18 个（17 个已修复，1 个遗留）
 - Trivial：5 个（遗留）
- 遗留缺陷：6 个（不影响发布）

3.2 测试范围

序号	测试模块	测试项
1	数据导入	CSV 导入、Excel 导入、数据类型推断、缺失值处理、非单调数据处理
2	模型拟合	GO/JM/S-Shaped/BP/SVR/Hybrid/GM 模型、参数估计、收敛性、自动排名
3	可视化	预测曲线对比图、残差分析图、U 图（拉普拉斯检验）、Y 图（算术平均检验）
4	报告导出	PDF 生成、中文字体显示、图表嵌入、多页排版
5	实验管理	实验导出 ZIP、实验导入、数据重现
6	插件系统	第三方模型加载、entry_points 机制、接口契约验证
7	CLI 命令行	参数解析、模型运行、报告生成、错误处理
8	边界测试	空数据集、极小/超大数据集、极端参数配置
9	性能测试	并发执行、大数据集处理、内存占用、响应时间
10	安全性测试	特殊字符注入、超长字段、畸形文件、路径遍历
11	兼容性测试	Windows 10/11、Python 3.10/3.11/3.12、PyInstaller EXE
12	可用性测试	首次使用引导、错误提示友好性
13	回归测试	核心功能回归、性能回归

表 1 测试内容一览表

4 测试实施情况

4.1 测试实施时间和地点

- 测试时间：2025 年 12 月 02 日 - 2025 年 12 月 27 日
- 测试地点：ZDP（Zero-Defect Prediction）项目组测试实验室
- 测试环境：Windows 10/11 工作站 + Python 3.10/3.11/3.12 环境

4.2 测试团队

- 测试经理：黄耘青（负责测试计划、进度管理）
- 功能测试工程师：何东泽、赵乐坤（负责功能和接口测试）
- 自动化测试工程师：黄耘青（负责 pytest 脚本开发）
- 性能测试工程师：何东泽（负责性能和压力测试）
- 安全测试工程师：赵乐坤（负责安全测试）
- 配置管理员：何东泽（负责环境配置和版本管理）

4.3 测试实施进程

序号	测试阶段	开始日期	结束日期	工作内容描述
1	测试策划	2025-12-02	2025-12-05	编写测试计划，搭建测试环境，准备测试数据
2	测试设计和实现	2025-12-06	2025-12-12	设计 126 个测试用例，开发 pytest 自动化测试脚本
3	第 1 轮冒烟测试	2025-12-13	2025-12-14	执行 30 个核心功能测试用例，快速验证主要流程
4	第 2 轮全面测试	2025-12-15	2025-12-19	执行所有功能、接口、边界测试用例（70 个），发现主要缺陷
5	第 3 轮专项测试	2025-12-20	2025-12-22	执行性能、安全、兼容性测试用例（26 个），压力测试
6	缺陷修复期	2025-12-15	2025-12-23	开发团队修复发现的缺陷，提交修复版本
7	第 4 轮回归测试	2025-12-24	2025-12-25	验证缺陷修复，执行回归测试用例，确保无新增问题
8	测试报告编写	2025-12-24	2025-12-27	整理测试数据，编写测试报告，提交评审

表 2 测试实施进程

5 测试数据分析

5.1 测试用例执行结果

轮次	测试阶段	测试项数	设计用例	执行用例	通过用例	遗留问题
1	冒烟测试	5	30	30	22	8
2	全面测试	8	70	70	46	24
3	专项测试	5	26	26	20	6
4	回归测试	13	126	126	120	6
	总计	13	126	126	120	6

表 3 测试执行情况一览表

测试用例执行率： $126/126 = 100\%$
测试用例通过率： $120/126 = 95.24\%$
缺陷修复率： $36/42 = 85.71\%$

5.2 缺陷趋势分析

测试期间（2025 年 12 月 13 日至 2025 年 12 月 25 日）共发现 42 个缺陷，缺陷发现和修复趋势如下：

日期	新增缺陷	修复缺陷	累计缺陷	遗留缺陷	备注
12-13	3	0	3	3	冒烟测试开始
12-14	5	2	8	6	
12-15	8	3	16	11	全面测试开始
12-16	9	5	25	15	缺陷高峰期
12-17	7	8	32	14	
12-18	4	7	36	11	
12-19	2	6	38	7	
12-20	3	4	41	6	专项测试
12-21	1	3	42	4	
12-22	0	2	42	2	
12-23	0	4	42	-2	修复完成
12-24	0	8	42	-10	回归测试
12-25	0	0	42	6	遗留缺陷评审

表 4 每日缺陷统计

- 缺陷趋势分析：
- 缺陷发现高峰期在全面测试阶段（12-15 至 12-18），符合预期
 - 从 12-19 开始，缺陷修复速度超过发现速度，呈现收敛趋势
 - 回归测试未发现新缺陷，说明修复质量良好
 - 最终遗留 6 个低严重度缺陷，不影响软件发布

5.3 按缺陷严重程度统计

严重程度	数量	占比	修复状态	典型缺陷
Blocker	2	4.8%	已修复	程序启动时依赖库冲突导致崩溃；大数据集加载 OOM
Critical	5	11.9%	已修复	G0 模型参数估计偏差>10%；PDF 中文全部乱码；多线程数据竞争
Major	12	28.6%	已修复	Excel .xls 格式不支持；CLI 参数解析错误；图表坐标轴标签错误
Minor	18	42.9%	17 个已修复	界面布局不合理；提示信息不够友好；图标显示模糊
Trivial	5	11.9%	遗留	注释拼写错误；README 示例命令有误；日志格式不规范
总计	42	100%	36 个已修复	

表 5 缺陷严重程度分布

- 缺陷集中模块：
- 数据导入模块：10 个缺陷（Excel 格式、缺失值处理、类型推断）
 - 模型拟合模块：9 个缺陷（BP/SVR 收敛问题、参数估计精度）
 - PDF 报告模块：8 个缺陷（中文字体、图表嵌入、排版）
 - CLI 命令行：6 个缺陷（参数解析、错误处理）
 - 其他模块：9 个缺陷

5.4 性能测试结果

测试场景	目标值	实测值	结论
数据加载（10000 点）	< 5 秒	3.2 秒	√ 通过
8 模型并发执行（5000 点）	< 60 秒	42.7 秒	√ 通过
BP 模型拟合（10000 点）	< 60 秒	48.3 秒	√ 通过
图表渲染（5 模型）	< 10 秒	5.8 秒	√ 通过
PDF 生成（8 模型+图表）	< 15 秒	11.2 秒	√ 通过
峰值内存占用	< 2GB	1.4GB	√ 通过
GUI 响应时间（点击按钮）	< 200ms	85ms	√ 通过

表 6 性能指标测试结果

性能测试结论：所有性能指标均达到或优于目标值，系统性能表现优秀。

5.5 兼容性测试结果

测试项	测试环境	测试结果
操作系统兼容性	Windows 10 21H2	√ 通过
	Windows 11 22H2	√ 通过
Python 版本兼容性	Python 3.10.11	√ 通过
	Python 3.11.9	√ 通过
	Python 3.12.5	√ 通过
打包程序兼容性	PyInstaller EXE (Windows 10)	√ 通过
	PyInstaller EXE (Windows 11)	√ 通过
DPI 缩放兼容性	150% DPI	√ 通过
	200% DPI	√ 通过

表 7 兼容性测试结果

兼容性测试结论：系统在所有目标平台上运行正常，兼容性良好。

5.6 代码覆盖率分析

模块	行覆盖率	分支覆盖率
zdp.data (数据加载)	92%	85%
zdp.models (模型实现)	88%	78%
zdp.services (服务编排)	91%	83%
zdp.visualization (可视化)	85%	72%
zdp.reporting (报告生成)	87%	75%
zdp.gui (图形界面)	78%	68%
zdp.cli (命令行)	95%	88%
总体覆盖率	87%	78%

表 8 代码覆盖率统计 (pytest-cov)

- 覆盖率分析：
- 总体代码行覆盖率 87%，超过目标值（85%），达到良好水平
 - CLI 模块覆盖率最高（95%），因为参数解析逻辑相对简单
 - GUI 模块覆盖率相对较低（78%），主要是部分异常分支和边界情况未覆盖
 - 核心模型和数据处理模块覆盖率均超过 85%，质量有保障

6 测试结论

6.1 测试结果

6.1.1 文档审查测试

测试组对 ZDP 软件的需求规格说明、设计说明、用户手册、API 文档进行了审查。审查结论：

- 需求规格说明：格式规范，内容完整、准确，需求描述清晰，可追溯性良好
- 设计说明：架构设计合理，模块划分清晰，接口定义完整
- 用户手册：安装说明完整，功能说明详细，截图时效，可操作性强
- API 文档：接口定义准确，类型注解完整，示例代码可用

6.1.2 功能测试

- 数据导入功能：支持 CSV/Excel 格式，数据类型自动推断准确率 100%，缺失值和异常数据处理正确
- 模型拟合功能：8 种模型全部支持，参数估计精度符合要求，收敛成功率>95%
- 可视化功能：4 类图表生成正确，数据一致性良好，渲染效果清晰
- 报告导出功能：PDF 生成成功率 100%，中文字体显示正常，报告内容完整
- 实验管理功能：导出/导入功能正常，数据重现准确
- CLI 命令行：参数解析正确，功能完整，错误处理友好

6.1.3 接口测试

- 插件系统：entry_points 机制工作正常，第三方模型加载成功，接口契约验证通过
- 模块接口：数据加载、模型编排、可视化等内部接口调用正确，数据传递准确

6.1.4 边界测试

- 数据边界：空数据集、极小/超大数据集、缺失值、负数、非单调数据等边界情况处理正确
- 参数边界：BP/SVR/Hybrid 模型在极端参数配置下运行稳定，无崩溃

6.1.5 性能测试

所有性能指标均达到或优于目标值：

- 数据加载时间：3.2 秒（目标<5 秒）
- 并发执行时间：42.7 秒（目标<60 秒）
- 内存占用：1.4GB（目标<2GB）
- GUI 响应时间：85ms（目标<200ms）

6.1.6 安全性测试

- 特殊字符注入：100% 防护，无代码执行风险
- 超长字段：优雅降级，不崩溃
- 畸形文件：正确识别并提示错误
- 路径遍历：禁止访问父目录

6.1.7 兼容性测试

- 操作系统：Windows 10/11 兼容性良好
- Python 版本：3.10/3.11/3.12 均正常运行
- 打包程序：PyInstaller EXE 独立运行无问题
- DPI 缩放：150%/200% DPI 界面显示清晰

6.2 测试基本结论

经过 18 个工作日、4 轮全面测试，ZDP 软件可靠性分析系统 v0.1.0 版本测试结论如下：

总体评价：

- 1. 软件文档格式规范，内容完整、准确，可维护性好
- 2. 测试项的各功能实现与需求描述一致，功能完整度 100%
- 3. 126 个测试用例全部执行，通过率 95.24%，达到验收标准
- 4. 发现的 42 个缺陷中，36 个已修复，6 个低严重度缺陷遗留（不影响发布）
- 5. 所有严重缺陷（Blocker/Critical）已修复并验证通过
- 6. 代码覆盖率 87%，超过目标值（85%）
- 7. 性能指标全部达标，系统运行稳定
- 8. 兼容性测试全部通过，支持目标平台

软件质量评估：

- 功能性：优秀（功能完整，准确性高）
- 可靠性：良好（稳定性好，缺陷密度低）
- 易用性：良好（界面友好，错误提示清晰）
- 效率性：优秀（性能指标优秀）
- 可维护性：良好（代码规范，文档完整）
- 可移植性：优秀（跨版本兼容性好）

测试结论：ZDP（Zero-Defect Prediction）软件可靠性分析系统 v0.1.0 版本通过测试验收，建议发布。

7 遗留问题说明

ID	严重程度	缺陷描述	处理建议
1	Minor	GUI 界面“图表模型”下拉框选中效果不明显	v0.2.0 版本优化 UI
2	Trivial	README.md 中 CLI 示例命令少了 --time-column 参数	v0.1.1 补丁版本修复
3	Trivial	模型拟合日志输出格式不统一	v0.2.0 版本规范化
4	Trivial	部分注释中存在拼写错误（2 处）	v0.1.1 补丁版本修复
5	Trivial	pytest 测试输出中有 1 个 DeprecationWarning	v0.2.0 版本升级依赖库
6	Trivial	用户手册第 3.2 节缺少截图	v0.1.1 补丁版本补充

表 9 遗留缺陷清单

遗留缺陷影响评估：

- 全部为 Minor/Trivial 级别，不影响核心功能
- 不影响软件正常使用和发布
- 建议在后续版本（v0.1.1/v0.2.0）中修复

8 后续工作建议

1. 加强测试入口准则检查：建议在下一版本测试前，明确测试入口准则检查表，包括代码静态分析、单元测试覆盖率等，确保被测软件质量达到测试要求后再进入正式测试。
2. 测试尽早介入：建议测试团队在需求分析阶段就介入，参与需求评审，提前识别需求中的歧义和风险，减少后期返工。
3. 完善自动化测试：当前 `pytest` 自动化测试覆盖率约 60%，建议继续扩充自动化测试用例，尤其是 GUI 自动化测试 (`pytest-qt`)，提高回归测试效率。
4. 建立性能基准库：建议建立性能基准测试数据库，记录每个版本的性能指标，用于版本间性能对比和性能回归检测。
5. 加强安全测试：建议引入专业的安全扫描工具（如 `Bandit`），定期进行安全漏洞扫描，尤其是对数据导入和插件加载模块。
6. 持续集成/持续部署（CI/CD）：建议建立 CI/CD 流程，每次代码提交自动触发单元测试和集成测试，自动生成覆盖率报告，提高开发效率。
7. 用户体验优化：建议在下一版本中增加更多的用户引导功能，如首次使用向导、交互式教程、智能提示等，降低学习成本。
8. 国际化支持：当前软件主要面向中文用户，建议在后续版本中考虑国际化支持（i18n），支持英文界面和文档。

— 测试报告文档结束 —

测试报告编写人：黄耘青
测试经理审核：何东泽
项目经理批准：赵乐坤
日期：2025 年 12 月 27 日