

标记	数量	修改单号	签字	日期

# Zero-Defect Prediction 软件可靠性分析系统 测试计划 V1.0

(ZDP 软件配置项测试计划)

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 ..... 4

2 依据和引用文档 ..... 4

3 测试环境 ..... 4

    3.1 硬件环境 ..... 4

    3.2 软件环境 ..... 5

        3.2.1 核心依赖库版本 ..... 5

    3.3 安装、测试与控制 ..... 6

    3.4 测试数据准备 ..... 6

4 测试需求分析 ..... 6

    4.1 测试目标 ..... 6

    4.2 测试类型及测试要求 ..... 7

    4.3 测试范围 ..... 9

        4.3.1 包含的测试内容 ..... 9

        4.3.2 不包含的测试内容 ..... 9

    4.4 测试准入准则 ..... 9

    4.5 测试终止准则 ..... 9

5 测试进度及人员安排 ..... 10

    5.1 测试进度安排 ..... 10

    5.2 人员安排 ..... 12

    5.3 测试资源需求 ..... 12

        5.3.1 硬件资源 ..... 12

        5.3.2 软件资源 ..... 12

        5.3.3 测试工具 ..... 12

    5.4 风险评估与应对 ..... 13

6 测试交付物 ..... 13

    6.1 测试文档 ..... 13

    6.2 测试代码 ..... 13

    6.3 测试数据 ..... 13

7 测试终止条件 ..... 14

8 附录 A：软件文档审查清单 ..... 15

    8.1 A.1 需求规格说明审查表 ..... 15

    8.2 A.2 设计说明审查表 ..... 15

    8.3 A.3 用户手册审查表 ..... 16

9 附录 B：测试用例分类统计 ..... 17

---

10	附录 C：缺陷分级标准 .....	18
11	附录 D：测试通过标准 .....	19
11.1	功能测试通过标准 .....	19
11.2	性能测试通过标准 .....	19
11.3	安全测试通过标准 .....	19
11.4	兼容性测试通过标准 .....	19

## 1 范围

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计划文档，适用于 ZDP v0.1.0 版本的功能测试、性能测试、安全测试和兼容性测试。

## 2 依据和引用文档

- 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》
- GB/T 15532-2008《计算机软件测试规范》
- ISO/IEC 25010:2011《系统与软件工程 系统与软件质量模型》
- 《ZDP（Zero-Defect Prediction）软件需求规格说明》（内部文档）
- 《ZDP（Zero-Defect Prediction）软件设计说明》（内部文档）

## 3 测试环境

### 3.1 硬件环境

序号	设备名称	配置规格	数量
1	测试工作站 1	PC 机/Windows 10 64 位, Intel Core i7-10700, 16GB DDR4, 512GB SSD	1 台
2	测试工作站 2	PC 机/Windows 11 64 位, AMD Ryzen 7 3700X, 32GB DDR4, 1TB NVMe SSD	1 台
3	低配测试机	PC 机/Windows 10 64 位, Intel Core i5-8250U, 8GB DDR4, 256GB SSD	1 台

表 1 测试硬件环境配置

3.2 软件环境

序号	软件项名称	版本	测试设备	用途
1	Windows 10 Professional	21H2	工作站 1/低配机	运行平台
2	Windows 11 Professional	22H2	工作站 2	兼容性测试平台
3	Python	3.10.11	所有测试机	主要运行环境
4	Python	3.11.9 / 3.12.5	工作站 1	兼容性测试
5	uv 包管理器	0.5.8	所有测试机	依赖管理
6	ZDP 被测软件	0.1.0	所有测试机	被测对象
7	pytest 测试框架	8.2.2	所有测试机	自动化测试
8	pytest-qt	4.4.0	所有测试机	GUI 自动化测试
9	Microsoft Excel	2019	工作站 1/2	数据准备和验证
10	Adobe Acrobat Reader	DC 2024	所有测试机	PDF 报告验证

表 2 ZDP 软件可靠性分析系统测试环境软件项

3.2.1 核心依赖库版本

序号	库名称	版本	用途
1	PySide6	6.7.2	GUI 框架
2	pandas	2.2.2	数据处理
3	numpy	1.26.4	数值计算
4	scipy	1.11.4	科学计算
5	matplotlib	3.8.4	数据可视化
6	scikit-learn	1.4.2	机器学习模型
7	torch	2.2.2	神经网络
8	reportlab	4.1.0	PDF 生成
9	statsmodels	0.14.2	统计分析

表 3 Python 依赖库清单

3.3 安装、测试与控制

安装或测试项名称	安装计划	测试计划
被测软件 ZDP	2025-12-03	按第 5 章进度
Python 环境配置	2025-12-02	2025-12-03
依赖库安装 (uv sync)	2025-12-03	2025-12-03
测试数据集准备	2025-12-04	按第 5 章进度
自动化测试环境	2025-12-04	按第 5 章进度

表 4 安装和测试计划

3.4 测试数据准备

测试所需数据集位于 data/samples/ 目录，包括：

标准测试数据集：

- nhpp\_goel\_okumoto.csv - NHPP 累计故障数据（50 点）
- tbf\_jm\_synthetic.csv - JM 模型故障间隔数据（40 点）
- tbf\_s\_shaped.csv - S 型曲线数据（60 点）
- tbf\_jm\_nonmonotonic\_solvable.csv - 非单调 TBF 数据（35 点）
- field\_weekly\_counts.csv - 现场周故障数据（52 点）

边界测试数据集（需生成）：

- 空数据集（0 点）
- 单点数据集（1 点）
- 极小数据集（2-3 点）
- 大数据集（10000 点）
- 超大数据集（100000 点）
- 含缺失值数据集
- 含负数数据集
- 非单调累计故障数数据集

4 测试需求分析

4.1 测试目标

本次测试的主要目标是：

1. 验证 ZDP 软件的所有功能模块符合需求规格说明
2. 确保 8 种可靠性模型的拟合精度和收敛性达到设计要求
3. 验证系统在边界条件和异常情况下的鲁棒性
4. 评估系统的性能指标（响应时间、内存占用、并发能力）
5. 确保系统在目标平台上的兼容性（Windows 10/11, Python 3.10-3.12）
6. 验证 PDF 报告生成和中文字体显示的正确性
7. 确保系统的安全性，防范恶意输入攻击

## 4.2 测试类型及测试要求

序号	测试类型	测试要求
1	文档审查	对需求规格说明、设计文档、用户手册、API 文档进行审查，确保文档完整性、准确性和一致性（见附录 A）。
2	功能测试	依据需求规格说明对以下功能模块进行全面验证： <ul style="list-style-type: none"><li>• 数据导入（CSV/Excel，累计故障数/TBF）</li><li>• 8 种模型拟合（GO, JM, S-Shaped, BP, SVR, Hybrid, GM, Plugin）</li><li>• 可视化图表（预测曲线、残差图、U 图、Y 图）</li><li>• PDF 报告导出</li><li>• 实验管理（导出/导入 ZIP）</li></ul>
3	接口测试	验证以下接口的正确性： <ul style="list-style-type: none"><li>• 模型插件接口（entry_points 机制）</li><li>• CLI 命令行接口（zdp-cli）</li><li>• 内部模块接口（数据加载、模型编排、可视化）</li></ul>
4	边界测试	测试系统在边界条件下的行为： <ul style="list-style-type: none"><li>• 数据边界（空集、单点、极小/超大数据集、缺失值、负数、非单调）</li><li>• 参数边界（BP/SVR/Hybrid 模型的极端参数配置）</li></ul>
5	性能测试	验证系统性能指标： <ul style="list-style-type: none"><li>• 多模型并发执行时间 &lt; 60 秒（8 个模型，5000 点数据）</li><li>• 大数据集加载时间 &lt; 5 秒（10000 点）</li><li>• 模型拟合时间 &lt; 60 秒（BP 模型，10000 点）</li><li>• 内存占用 &lt; 2GB（峰值）</li><li>• GUI 响应时间 &lt; 200ms（界面操作）</li></ul>
6	强度测试	连续运行 8 小时不崩溃，执行以下操作循环： <ul style="list-style-type: none"><li>• 加载数据 → 运行所有模型 → 生成报告 → 清空 → 重复</li></ul>
7	安全性测试	验证系统对恶意输入的防护： <ul style="list-style-type: none"><li>• 特殊字符注入（SQL/XSS/命令注入）</li><li>• 超长字段攻击</li><li>• 畸形文件攻击</li><li>• 路径遍历攻击</li></ul>
8	兼容性测试	验证系统在不同环境下的兼容性： <ul style="list-style-type: none"><li>• 操作系统：Windows 10 (1809+) / Windows 11 (21H2+)</li><li>• Python 版本：3.10.x / 3.11.x / 3.12.x</li><li>• 打包程序：PyInstaller 单文件 EXE</li></ul>
9	可用性测试	评估系统的易用性： <ul style="list-style-type: none"><li>• 首次使用引导的清晰度</li><li>• 错误提示的友好性和可操作性</li><li>• 界面布局的合理性</li></ul>
10	回归测试	在缺陷修复后执行回归测试，确保： <ul style="list-style-type: none"><li>• 修复的缺陷不再出现</li><li>• 未引入新的缺陷</li><li>• 核心功能无退化</li></ul>

表 5 测试类型和方法



## 4.3 测试范围

### 4.3.1 包含的测试内容

1. 所有用户可见功能（GUI 和 CLI）
2. 所有 8 种可靠性模型的核心算法
3. 数据加载和预处理逻辑
4. 可视化图表生成
5. PDF 报告导出（含中文字体）
6. 插件系统（entry\_points 机制）
7. 多线程并发执行（QThread）
8. 异常处理和错误提示

### 4.3.2 不包含的测试内容

1. 第三方库（pandas, numpy, scipy 等）的内部实现
2. 操作系统底层功能
3. Python 解释器本身的功能
4. 非公开 API 或实验性功能

## 4.4 测试准入准则

测试活动开始前，必须满足以下条件：

1. 被测软件版本已构建并通过基本冒烟测试
2. 所有依赖库已正确安装（`uv sync --all-extras` 成功）
3. 测试环境已按第 3 章要求配置完毕
4. 测试数据集已准备就绪
5. 测试用例设计已完成并通过评审
6. 自动化测试脚本已开发完成（pytest 套件）

## 4.5 测试终止准则

测试活动满足以下条件时可以终止：

1. 完成所有计划的测试项（126 个测试用例）
2. 测试用例执行率  $\geq 98\%$
3. 测试用例通过率  $\geq 95\%$
4. 所有严重缺陷（Blocker/Critical）已修复并验证
5. 遗留的一般缺陷（Major） $\leq 3$  个，且不影响核心功能
6. 遗留的次要缺陷（Minor/Trivial） $\leq 10$  个
7. 代码覆盖率  $\geq 85\%$ （行覆盖率）
8. 性能指标全部达标
9. 测试文档齐全、符合规范，通过评审

## 5 测试进度及人员安排

### 5.1 测试进度安排

序号	活动	内容	开始时间	完成时间
1	测试策划	编写测试计划文档	2025-12-02	2025-12-05
2	环境准备	搭建测试环境，安装依赖，准备测试数据	2025-12-02	2025-12-05
3	测试设计	编写测试说明、设计测试用例（126 个）	2025-12-06	2025-12-10
4	测试实现	开发自动化测试脚本（pytest）	2025-12-08	2025-12-12
5	第一轮冒烟测试	执行核心功能测试用例（约 30 个），快速验证	2025-12-13	2025-12-14
6	第二轮全面测试	执行所有功能、接口、边界测试用例（约 70 个）	2025-12-15	2025-12-19
7	第三轮专项测试	执行性能、安全、兼容性测试用例（约 26 个）	2025-12-20	2025-12-22
8	缺陷修复跟踪	开发团队修复发现的缺陷	2025-12-15	2025-12-23
9	测试总结报告编写	编写测试总结报告，整理测试数据	2025-12-24	2025-12-26

表 6 测试工作进度安排

总测试周期：18 个工作日（2025-12-02 至 2025-12-27）

5.2 人员安排

序号	角色	职责	人数
1	测试经理	负责测试计划制定、资源协调、进度跟踪、风险管理	1 人
2	测试工程师（功能）	负责功能测试用例设计、执行和缺陷记录	2 人
3	测试工程师（自动化）	负责自动化测试脚本开发和维护（pytest）	1 人
4	测试工程师（性能）	负责性能测试、压力测试的设计和运行	1 人
5	测试工程师（安全）	负责安全测试用例设计和漏洞扫描	1 人
6	配置管理员	负责测试环境配置、版本管理、测试数据管理	1 人

表 7 测试团队人员安排

总人力：7 人

5.3 测试资源需求

5.3.1 硬件资源

- 高性能测试工作站：2 台
- 低配测试机（最低配置验证）：1 台
- 备用测试机：1 台

5.3.2 软件资源

- Windows 10/11 操作系统授权
- Microsoft Office (Excel)
- Adobe Acrobat Reader DC
- Python 3.10/3.11/3.12 环境
- PyCharm / VS Code IDE

5.3.3 测试工具

- pytest：单元测试和集成测试框架
- pytest-qt：GUI 自动化测试插件
- pytest-cov：代码覆盖率统计
- ruff / black：代码质量检查
- mypy：类型检查
- 性能监控工具：Windows Performance Monitor
- 内存分析工具：Python memory\_profiler

5.4 风险评估与应对

序号	风险描述	影响	应对措施
1	测试环境搭建延迟	中	提前 1 周开始环境准备，准备备用方案（Docker 容器化）
2	依赖库版本冲突	中	使用 uv 虚拟环境隔离，锁定依赖版本（uv.lock）
3	第三方模型拟合时间过长	低	设置超时机制（60 秒），对超时模型跳过并记录
4	缺陷修复周期长	高	建立快速缺陷响应机制，每日缺陷评审会议
5	性能测试数据不足	中	使用脚本生成大规模合成数据集（10000+点）
6	测试人员不足	中	优先执行高优先级测试用例，低优先级用例延后
7	中文字体缺失（PDF）	低	测试前验证目标机器字体，准备字体安装包
8	PyInstaller 打包失败	中	提前验证打包流程，准备 spec 文件模板

表 8 测试风险与应对策略

6 测试交付物

6.1 测试文档

- 1. 《ZDP 软件测试计划》（本文档）
- 2. 《ZDP 软件测试说明》（包含 126 个测试用例）
- 3. 《ZDP 软件测试报告》（测试结束后提交）
- 4. 《ZDP 缺陷报告汇总》（Excel 格式）
- 5. 《ZDP 测试覆盖率报告》（pytest-cov 生成）

6.2 测试代码

- 1. pytest 自动化测试套件（tests/ 目录）
- 2. 测试数据生成脚本（scripts/generate\_sample\_data.py）
- 3. 性能测试脚本（tests/test\_performance.py）

6.3 测试数据

- 1. 标准测试数据集（data/samples/）
- 2. 边界测试数据集（自动生成）
- 3. 回归测试基准数据（JSON 格式）

## 7 测试终止条件

满足以下所有条件时，测试活动可以终止：

1. 完成所有测试项，执行完所有测试用例（126 个）
2. 测试用例通过率  $\geq 95\%$
3. 测试过程中发现的所有严重缺陷（Blocker/Critical）已修复并通过回归测试
4. 遗留缺陷已评审，确认不影响软件发布
5. 代码覆盖率达到目标（ $\geq 85\%$ ）
6. 性能指标全部达标
7. 测试文档齐全、符合规范，通过评审
8. 测试经理和项目经理批准测试终止

8 附录 A：软件文档审查清单

8.1 A.1 需求规格说明审查表

序号	评审内容与评判标准	满足	不满足	不适用
1	完整清晰地描述了引用文件和标准	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2	确切给出了所有专用术语定义（NHPP，TBF，RMSE 等）	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3	功能需求描述完整（数据导入、模型拟合、可视化、报告）	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4	性能需求量化明确（响应时间、内存占用、并发能力）	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
5	明确提出安全性要求（输入验证、异常处理）	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
6	明确提出兼容性要求（操作系统、Python 版本）	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
7	用户界面需求描述清晰（GUI 布局、交互逻辑）	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
8	数据接口定义完整（CSV/Excel 格式、列名约定）	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
9	模型算法需求详细（8 种模型的数学描述）	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
10	可追溯性矩阵完整（需求-设计-测试用例）	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

表 9 需求规格说明审查表

n 审查结论： (☒) 合格      (☐) 不合格  
审查人员签字：      日期：2025 年 12 月 05 日

8.2 A.2 设计说明审查表

序号	评审内容与评判标准	满足	不满足	不适用
1	系统架构设计合理（分层架构：data/models/services/gui）	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2	模块划分清晰，职责单一（单一职责原则）	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3	接口设计完整（ReliabilityModel 基类定义明确）	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4	数据结构设计合理（FailureDataset，ModelResult）	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
5	异常处理机制完善（自定义异常类）	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
6	并发设计安全（QThread + 模型 clone 机制）	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
7	可扩展性良好（插件系统 entry_points）	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
8	设计文档与代码实现一致	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

表 10 设计说明审查表

审查结论： (☒) 合格      (☐) 不合格  
审查人员签字：      日期：2025 年 12 月 05 日

8.3 A.3 用户手册审查表

序号	评审内容与评判标准	满足	不满足	不适用
1	安装说明完整（uv sync, Python 版本要求）	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2	快速入门指南清晰（示例数据、基本操作）	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3	功能说明详细（每个功能模块独立章节）	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4	CLI 使用说明完整（参数列表、示例命令）	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
5	截图时效性（与当前版本 GUI 一致）	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
6	故障排除指南实用（常见问题 FAQ）	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
7	术语表完整（NHPP, TBF, RMSE 等）	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

表 11 用户手册审查表

审查结论： (☒) 合格      (☐) 不合格  
审查人员签字：      日期：2025 年 12 月 10 日



9 附录 B：测试用例分类统计

序号	功能模块	用例数量	优先级分布
1	数据导入	6	P0: 3 个, P1: 2 个, P2: 1 个
2	模型拟合	8	P0: 5 个, P1: 3 个
3	可视化图表	4	P0: 3 个, P1: 1 个
4	报告导出	3	P0: 2 个, P1: 1 个
5	实验管理	2	P1: 2 个
6	插件接口	2	P1: 2 个
7	CLI 命令行	3	P0: 2 个, P1: 1 个
8	边界测试	9	P0: 4 个, P1: 3 个, P2: 2 个
9	性能测试	4	P0: 2 个, P1: 2 个
10	安全测试	4	P1: 4 个
11	兼容性测试	4	P0: 2 个, P1: 2 个
12	可用性测试	2	P2: 2 个
13	回归测试	2	P0: 2 个
14	文档审查	3	P1: 3 个
总计	126 个用例	P0: 27, P1: 28, P2: 5	

表 12 测试用例按功能模块分类

- 优先级说明：
- P0（高优先级）：核心功能，必须通过，影响发布
  - P1（中优先级）：重要功能，应该通过，影响用户体验
  - P2（低优先级）：辅助功能，可以延后，不影响主要使用

10 附录 C：缺陷分级标准

级别	名称	定义	示例
Blocker	阻塞	导致系统崩溃、数据丢失或无法继续测试的缺陷	程序启动即崩溃；数据导入导致程序卡死；模型拟合计算错误导致结果不可用
Critical	严重	核心功能无法实现或严重错误，无替代方案	GO 模型参数估计偏差 >10%；PDF 导出中文全部乱码；多线程执行导致数据竞争
Major	一般	主要功能存在问题，有替代方案或影响有限	图表坐标轴标签错误；Excel 导入不支持.xls 格式；CLI 参数解析错误
Minor	次要	次要功能缺陷或界面问题，不影响主要使用	界面布局不美观；提示信息不够友好；图标显示模糊
Trivial	轻微	文档错误、拼写错误等不影响功能的问题	注释拼写错误；README 示例命令有误；日志输出格式不规范

表 13 缺陷严重程度分级

优先级	名称	修复要求
P0	立即修复	必须在下一个构建版本中修复，阻塞测试进行
P1	高优先级	应在本轮测试周期内修复，影响发布
P2	中优先级	可在下一版本修复，不影响当前发布
P3	低优先级	可延后修复，视资源情况而定

表 14 缺陷优先级定义

## 11 附录 D：测试通过标准

### 11.1 功能测试通过标准

模块	通过标准	验收指标
数据导入	<ul style="list-style-type: none"><li>成功导入 CSV/Excel 文件</li><li>正确识别累计故障数/TBF 类型</li><li>缺失值、非单调数据正确处理</li></ul>	<ul style="list-style-type: none"><li>支持格式覆盖率 100%</li><li>数据类型推断准确率 100%</li><li>异常处理覆盖率 100%</li></ul>
模型拟合	<ul style="list-style-type: none"><li>8 种模型全部支持</li><li>参数估计误差 &lt; 5%（基准数据集）</li><li>收敛成功率 &gt; 95%</li></ul>	<ul style="list-style-type: none"><li>模型覆盖率 100%</li><li>RMSE 与理论值偏差 &lt; 5%</li><li>拟合时间 &lt; 60 秒</li></ul>
可视化	<ul style="list-style-type: none"><li>4 类图表全部生成</li><li>图表数据与模型输出一致</li><li>坐标轴、图例正确显示</li></ul>	<ul style="list-style-type: none"><li>图表类型覆盖率 100%</li><li>数据一致性检查通过</li><li>渲染时间 &lt; 5 秒</li></ul>
报告导出	<ul style="list-style-type: none"><li>PDF 成功生成</li><li>中文字体正常显示</li><li>报告内容完整</li></ul>	<ul style="list-style-type: none"><li>生成成功率 100%</li><li>中文显示正确率 100%</li><li>章节完整性 100%</li></ul>

表 15 功能测试通过标准

### 11.2 性能测试通过标准

测试场景	目标值	最低要求	测试环境
数据加载（10000 点）	< 3 秒	< 5 秒	标准配置
8 模型并发执行（5000 点）	< 45 秒	< 60 秒	标准配置
BP 模型拟合（10000 点）	< 40 秒	< 60 秒	标准配置
图表渲染（5 模型）	< 5 秒	< 10 秒	标准配置
PDF 生成（8 模型+图表）	< 10 秒	< 15 秒	标准配置
峰值内存占用	< 1.5GB	< 2GB	标准配置
GUI 响应时间（点击按钮）	< 100ms	< 200ms	标准配置

表 16 性能指标基准

### 11.3 安全测试通过标准

- SQL/XSS/命令注入测试：100% 防护，无代码执行
- 超长字段测试：优雅降级，不崩溃
- 畸形文件测试：正确识别并提示错误
- 路径遍历测试：禁止访问父目录
- 敏感信息保护：无明文密码或敏感数据泄露

### 11.4 兼容性测试通过标准

- Windows 10/11：所有功能正常

2. Python 3.10/3.11/3.12: 依赖兼容, 功能正常
3. PyInstaller EXE: 独立运行, 功能完整
4. DPI 缩放 (150%/200%): 界面清晰, 无模糊

— 测试计划文档结束 —