日志管理系统改造测试

测试计划

XXX 公司

二〇XX年X月

1、项目概述

日志管理系统包括日志查询、日志接入、日志导出、监控告警、全局扫描等功能，针对本次日志管理系统的日志查询、导出、告警功能进行改造优化，故开展本轮测试，保障系统质量和稳定。

2、测试目标

日志管理系统当前业务需求的功能要求如下：

日志查询模块：

1. 关键字查询：满足不同数据集，列出关键字联合查询出结果。
2. 日志级别查询：通过 INFO/WARN/ERROR 等日志级别查询出结果。
3. 查询时间：满足最近 7 天之内，可以进行日志查询的需求，自定义起始结束时间。
4. 性能目标：千万级铺地数据量，简单查询<1 秒，复杂查询<3 秒。

日志导出模块：

1. 导出范围：支持当前查询结果导出、也支持按时间范围导出。
2. 导出格式：支持 CSV、Excel、TXT 等多种格式
3. 大文件处理：支持百万级日志异步导出、进度追踪、完整性校验。
4. 性能目标：导出进程 CPU、内存占比低于 20%。

监控告警模块：

1. 告警配置：支持关键字频率、错误日志比例阈值配置。

2. 通知渠道：支持邮件、钉钉、企业微信、短信等通知告警。
3. 准确性：触发条件时及时告警、不能误报漏报。

总体质量目标：

功能：核心功能测试覆盖率 100%，P0/P1 级 Bug 为零。

稳定性：系统支持 7*24 小时全天候查询。

进度：按需求评估计划于 20 个工作日内完成全部测试。

3、测试范围和重点 5H2W

维度	分析内容	详细说明
Why	为什么要测？	验证日志查询、导出、告警功能改造后的功能正确性、数据准确性及系统性能，确保上线后不影响现有业务，提升检索效率。
What	测什么功能？	日志查询：关键字联合查询、日志级别筛选、时间范围筛选（最近 7 天/自定义）。 日志导出：CSV/Excel 格式、同步/异步导出、大数据量导出。 监控告警：阈值配置、通知渠道（邮件/短信/Webhook）。 底层接口：查询 API、导出 API、配置 API。
Where	在哪里测？	功能测试环境：用于日常功能验证、接口自动化测试。 性能测试环境：独立环境，用于高并发及大数据量压测（需隔离）。 预发布环境：用于上线前的最终回归验证（连接生产只读库或镜像库）。
Who	谁来测？	测试团队：负责用例设计、执行、缺陷跟踪。 开发团队：协助构造测试数据、修复 Bug、解释技术逻辑。 运维/DBA：协助监控服务器资源、分析慢查询 SQL、环境部署。
When	何时测？	功能测试：开发提测后即刻开始，持续至上线前。 性能测试：功能测试收敛后（Bug 率<5%）介入。 回归测试：每轮 Bug 修复后及上线前 24 小时内。
How	怎么测？	功能测试：黑盒测试、边界值分析、场景法（手工 + 自动化）。 性能测试：使用 JMeter 模拟并发，Prometheus 监控资源。 数据构造：编写 Python 脚本生成千万级模拟日志数据。
How Much	测到什么程度？	用例覆盖率：核心功能测试覆盖率 100%。 性能指标：千万级数据下查询响应<3s，导出百万级数据不超时。 质量标准：P0/P1 级缺陷清零，无严重内存泄漏。

测试范围：

功能：

1. 日志查询需要覆盖单条件、多条件查询；
2. 日志导出支持同步、异步导出；
3. 告警配置及触发验证；
4. 权限控制，不同用户拥有不同数据集。

接口：

覆盖查询接口、日志导出接口、告警配置接口。

性能：

重点针对高并发查询、大数据量导出进行测试。

稳定性：

执行 24 小时稳定性测试、监控系统是否存在资源占用高、内存泄漏等风险。

兼容性：

主流浏览器谷歌、Edge、火狐页面展示兼容性验证。

测试重点：

1. 复杂查询的准确性：验证时间+级别+关键字多条件查询、结果是否正确、是否存在数据不准确的情况。
2. 大数据量性能表现：在千万级铺地数据时、验证极端情况性能表现。
3. 导出功能稳定性：验证大文件导出时是否导致 OOM 或连接断开。
4. 告警实时性：验证日志达到告警标准到触发告警延迟<1 分钟。

4、测试策略

采用功能、性能、自动化、测试策略。

4.1 功能测试

思路：采用黑盒测试、基于需求设计测试用例。

方法：

等价类划分：针对时间范围、日志级别进行分类组合测试。

边界值分析：7 天边界、最大导出能力、关键字长度限制。

场景法：查询-发现异常-导入日志。调整告警配置-触发告警。

工具：

使用 jira、禅道管理测试用例。

4.2 性能测试

思路：模拟生产数据量级、逐步增加并发、寻找系统瓶颈。

方法：

基准、负载、混合、稳定性、高可用

工具：

jmeter、loadrunner、监控工具 promethus+grafana、AWR 报告等

4.3 自动化测试

思路：采用接口、UI 自动化结合的测试策略。

方法：

接口自动化：使用 pytest+requests 框架，将查询、导出接口纳入 Jenkins 自动化流水线，代码变更自动回归。

UI 自动化：核心查询关键字条件、使用 selenium 编写冒烟脚本。

工具：

pycharm

5、项目里程碑

任务	开始时间	结束时间
需求了解/评审	T-4	T-2
测试用例设计	T-2	T
TC 评审	T+1	T+2
自动化开发/测试工具开发	T+3	T+5
冒烟测试	T+6	T+6
功能测试	T+7	T+10
性能测试	T+11	T+14
第一轮回归测试	T+15	T+16
第二轮回归测试	T+17	T+18
预发布验证	T+19	T+19
发布	T+20	T+20

6、测试资源

6.1 人力资源

测试参与人员及角色

角色	人员	职责
项目经理	张三	进度把控、资源协调
测试工程师	李四	用例设计、脚本开发、执行测试、跟踪缺陷、总结报告
开发工程师	王五	环境部署、Bug 修复

6.2 环境资源

1）列出此项目中所涉及到的测试环境、包括测试机名称及用途

功能测试环境：用于功能验证、部署最新代码。

性能测试环境：用于性能压测、配置与生产环境 1:4 或 1:2 缩放、确保数量级足够。

pre 预发布环境：上线前验证、连接只读生产库、或生产库镜像。

2）测试环境部署方案

部署方法：采用 docker/k8s 部署，保证环境一致。

数据准备：模拟生产日志开发生成日志的脚本、在性能环境进行数据铺地。

注意事项：压测如果存在远超生产的增量表，需定时清理以模拟实际情况。

7、风险列表

风险描述	可能性	影响程度	解决方法
性能测试数据准备不充足，测试结果失真	高	高	尽可能模拟生产数据量级、数据多样性，保证测试脚本正确性及请求参数多元化，以避免该风险。
需求频繁变更，测试进展受变更而延期	中	中	加强需求或用例评审细节、将问题尽可能暴漏在前期，减少变更情况。如无法避免，申请延期。
环境资源紧张，性能测试资源不足。	中	高	若性能测试环境资源不足，使用单节点或降低配置后测试，待资源充足时，考虑加入扩展性场景，辅助评估系统性能。