



需求文档：Web 性能分析与测评工具 (WebPerf Scout)

项目代号：WebPerf Scout

目标用户：前端工程师、产品经理、UX 研究员、SEO 优化师

核心价值：通过自动化爬取 + 性能指标采集，提供可量化、可对比、可追溯的 Web 性能测评报告

一、产品目标 (Vision)

构建一个基于 Specification-Driven Development 的 Web 性能测评平台，支持用户通过声明式配置 (YAML/JSON Spec) 定义待测网站、设备模拟条件、性能关注点，系统自动执行爬取、性能采集、分析与报告生成，确保结果可复现、可审计、可持续维护。

二、核心功能需求

2.1 输入：声明式测评规范 (Spec)

用户通过 `perf-spec.yaml` 文件定义测评任务，例如：

```
version: "1.0"
targets:
```

```
- url: "https://example.com/home"
  name: "首页"
- url: "https://example.com/product?id=123"
  name: "商品详情页"

deviceProfiles:
- name: "Mobile_4G"
  device: "iPhone 12"
  network: "4G"
  cpuThrottle: 4
- name: "Desktop_Fast"
  device: "Desktop"
  network: "WiFi"
  cpuThrottle: 1

metrics:
- lcp
- fcp
- tti
- totalBlockingTime
- cls
- resourceLoadTime

output:
  format: "html+json"
  reportPath: "./reports/"
```

✅ **SDD 体现**：所有行为由 spec 驱动，AI 可据此生成爬取脚

本、性能采集逻辑、报告模板。

2.2 自动化爬取与性能采集

- 使用 Playwright (支持 SPA、登录态、Cookie 注入)
- 每个 target × deviceProfile 组合独立运行一次
- 在真实浏览器环境中采集 [Web Vitals](#) 核心指标
- 支持重试机制 (如页面加载失败最多重试 2 次)

2.3 性能分析与评分

- 对每个页面生成性能评分 (0-100) , 参考 Lighthouse 逻辑但可配置权重
 - 支持自定义阈值告警 (如 LCP > 2.5s 标红)
 - 对比历史数据 (若启用存储) 识别性能退化
-

2.4 报告输出

- **HTML 报告** : 含瀑布图、指标卡片、截图、资源列表 (使用现有开源模板如 Lighthouse Report)
 - **JSON 报告** : 结构化数据 , 便于 CI/CD 集成或二次分析
 - 支持多页面横向对比视图 (如“首页 vs 商品页”在 Mobile_4G 下表现)
-

2.5 可扩展性设计 (企业级约束)

- 所有逻辑模块化 : 爬取器、指标采集器、分析器、报告生成器
 - 支持通过 spec 扩展新指标 (如自定义 JS 执行时间)
 - 日志结构化输出 (便于接入企业日志系统)
 - 安全 : 禁止执行任意 JS (防 XSS) , 限制爬取深度
-

三、非功能性需求

类别	要求
可维护性	所有业务逻辑由 spec 驱动 , 代码与 spec 强绑定 , 变更只需改 spec

类别	要求
可复现性	相同 spec + 相同环境 → 相同报告
性能	单任务 (1 页面 × 1 设备) 执行时间 ≤ 15 秒
兼容性	支持 Chrome、Firefox (通过 Playwright)
部署	支持 Docker 容器化运行，提供 CLI 入口

四、交付物 (MVP 范围)

1. spec-kit 兼容的项目结构 (含 .spec/ 目录)
2. CLI 工具 : webperf-scout run --spec perf-spec.yaml
3. 自动生成 HTML + JSON 报告
4. 示例 spec 文件 (覆盖 2 个页面、2 种设备)
5. README.md : 说明如何编写 spec、运行、扩展

✅ 符合赛题要求 : 可实际投产运营 (例如集成到前端 CI 流程 , 每日自动测评)

五、SDD 与 AI 协同设计建议

• AI 角色 :

- 根据 spec 生成 Playwright 爬取脚本
- 生成指标计算逻辑 (如 TTI 计算)
- 生成 HTML 报告模板 (Handlebars/EJS)

• 开发者角色 :

- 定义 spec schema (JSON Schema)

- 实现执行引擎 (调度、错误处理)
 - 集成安全与日志模块
-

六、后续演进方向 (加分项)

- 支持登录态 (spec 中声明 auth cookies / token)
 - 集成到 GitHub Actions , PR 中自动评论性能变化
 - 多版本对比 (如 v1.0 vs v1.1)
 - 导出 PDF 报告
-

七、附录：术语说明

- **LCP** (Largest Contentful Paint) : 最大内容绘制时间
- **FCP** (First Contentful Paint) : 首次内容绘制
- **TTI** (Time to Interactive) : 可交互时间
- **CLS** (Cumulative Layout Shift) : 累积布局偏移
- **Web Vitals** : Google 提出的核心用户体验指标集