一、用户角度质量属性（8项）

1. 有效性（Effectiveness）

定义：系统能满足用户核心需求的准确性和完整性。  
指标：

问答准确率：LLM应答准确率≥95%

热点问题识别时效：热点问题识别时效≤30分钟

2. 高效性（Efficiency）

定义：系统在资源消耗与响应速度上的优化表现。  
指标：

响应速度：用户提问到首响应≤5秒

并发处理能力：支持≥1000并发会话

LLM推理延迟：LLM推理延迟≤800ms

3. 灵活性（Flexibility）

定义：系统适应用户需求变化或环境变化的能力。  
指标：

多模态交互支持：支持语音、文字、图片等多模态输入

知识库动态更新：支持实时/批量更新知识库内容

4. 完整性（Completeness）

定义：系统覆盖用户所需全部功能和服务场景。  
指标：

服务完整性：覆盖教学、科研、生活等6大类服务场景

功能覆盖率：核心功能（如问答、审核、聊天）100%实现

5. 可操作性（Operability）

定义：用户操作系统的便捷性与直观性。  
指标：

界面友好性：提供分步骤引导、错误提示清晰

终端适配：移动端/PC端统一体验，适配主流浏览器与设备

6. 可靠性（Reliability）

定义：系统在无故障条件下持续运行的能力。  
指标：

系统可用性：24小时服务，可用性≥99%

数据可靠性：每日全量备份，数据恢复时间≤1小时

7. 强壮性（Robustness）

定义：系统在异常输入或环境干扰下的容错能力。  
指标：

非法请求拦截率：≥99%

异常处理能力：系统崩溃自动重启，错误日志可追溯

8. 可用性（Usability）

定义：用户无需培训即可有效使用系统的程度。  
指标：

用户学习成本：新用户3分钟内完成核心功能操作

可访问性：支持无障碍模式（如屏幕阅读器）

二、开发者角度质量属性（4项）

1. 可维护性（Maintainability）

定义：系统代码易于修改、扩展和修复缺陷的能力。  
指标：

代码可维护性：代码复杂度（Cyclomatic Complexity）≤10

模块化设计：核心功能模块独立，接口文档完整

2. 可移植性（Portability）

定义：系统在不同硬件/软件环境中的部署能力。  
指标：

跨平台支持：支持Linux/Windows部署，容器化（Docker）

依赖管理：第三方库版本兼容性验证通过率100%

3. 可重用性（Reusability）

定义：系统组件在其他项目中复用的可能性。  
指标：

组件化设计：≥80%核心功能封装为独立微服务

开发文档完备性：API文档覆盖率100%

4. 可测试性（Testability）

定义：系统支持自动化测试和缺陷定位的便利性。  
指标：

单元测试覆盖率：核心模块≥85%

自动化测试：CI/CD流水线集成，冒烟测试执行时间≤10分钟