

以下是基于Scrum框架的项目执行计划，结合提案中的时间表和功能模块进行优化调整：

## Scrum框架设计（总周期：8周，分为4个Sprint）

### 角色分配

- 产品负责人（PO）**：负责需求优先级（由全栈开发者兼任）。
- Scrum Master**：负责流程协调（由前端开发者兼任）。
- 开发团队**：前端、后端、全栈开发者。

### Sprint规划

#### Sprint 1（Week 1-2）：需求分析与原型设计

- 目标**：完成用户需求验证、技术选型与低保真原型。
- 任务清单**：
  - 用户调研（学生/教师访谈）→ PO主导。
  - 技术方案确认（AI模型选型、3D库对比）→ 后端+全栈。
  - 低保真UI原型（Figma）→ 前端。
  - 初始产品待办列表（Product Backlog）→ PO整理。
- 产出**：需求文档、原型图、技术架构图。

#### Sprint 2（Week 3-4）：前端核心功能开发

- 目标**：实现浏览器扩展基础框架与核心交互。
- 任务清单**：
  - 浏览器扩展脚手架搭建（Manifest V3）→ 前端。
  - 文本捕获与高亮功能 → 前端+全栈。
  - 词汇仪表盘静态页面 → 前端。
  - 本地数据库设计（IndexedDB）→ 后端。
- 产出**：可运行的扩展基础版本，支持文本捕获与本地存储。

#### Sprint 3（Week 5-6）：后端与AI集成

- 目标**：完成AI服务对接与数据同步逻辑。
- 任务清单**：
  - 部署错误分析API（FastAPI + OpenAI）→ 后端。
  - 集成间隔重复算法（词汇复习）→ 后端。
  - 用户数据同步逻辑（SQLite + RESTful API）→ 全栈。
  - 前端联调测试 → 全栈+前端。
- 产出**：支持AI反馈与数据同步的MVP版本。

## Sprint 4 (Week 7-8)：高级功能与发布准备

- 目标：完成3D可视化、跨设备同步与发布。
- 任务清单：
  - 3D知识图谱渲染 (Three.js) → 前端。
  - 跨设备同步 (Firebase Auth + WebSocket) → 后端。
  - 性能优化与压力测试 → 全栈。
  - Chrome Web Store提交流程 → PO。
- 产出：正式版扩展包，发布至应用商店。

## Scrum关键活动安排

- 每日站会 (15分钟)：
  - 时间：每天上午9:00。
  - 内容：昨日进展、今日计划、阻塞问题。
- Sprint评审会 (每Sprint最后一天)：
  - 展示增量功能，收集反馈 (邀请教师/学生代表)。
- Sprint回顾会 (每Sprint结束后)：
  - 总结改进点 (如：AI API延迟优化方案)。

## 风险管理整合

- AI延迟问题：在Sprint 3中预留时间实现请求队列与缓存逻辑。
- 浏览器兼容性：在Sprint 4中增加“降级模式”开发任务。
- 用户反馈超载：PO在发布后使用优先级框架 (Bug > 性能 > 新功能) 管理需求池。

## 工具推荐

- 任务管理：Jira/Trello (可视化Sprint Backlog)。
- 协作开发：GitHub (分支策略：feature/sprintX)。
- 文档管理：Confluence/Notion (需求+技术文档归档)。

通过此框架，团队可灵活响应需求变化，同时确保核心功能按时交付。