准备阶段：第1周

项目范围定义：与利益相关者定义项目的范围，明确必要的业务需求和技术需求。

技术选型：选择合适的技术栈，包括前端框架、后端语言、数据库和第三方服务。

风险评估：识别项目可能面临的风险，制定相应的缓解措施。

资源分配：确定项目资源，包括人员、设备和预算。

第一增量版本：构建核心功能

第2周至第4周：功能开发

用户认证系统：

安全策略：设计一个多层次的安全模型，包括密码加密、二因素认证。

后端实现：使用JWT（JSON Web Tokens）或OAuth进行用户身份验证和授权。

前端实现：开发用户登录和注册界面，提供友好的用户交互体验。

课程管理系统：

课程数据库设计：设计一个能够存储课程信息、教师信息和用户反馈的数据库架构。

API开发：实现RESTful API，支持课程的增加、查询、修改和删除操作。

前端实现：创建直观的课程管理界面，使教师和学生能够轻松管理和浏览课程。

第5周：集成与部署

自动化测试：开发自动化测试脚本，持续集成和部署。

性能评估：进行负载测试，确保系统在高负载下的表现符合预期。

文档编写：更新API文档和用户手册，为用户和开发者提供详细的使用和维护指南。

第二增量版本：增加高级功能

第6周至第7周：高级功能开发

学习进度跟踪系统：

数据模型设计：设计用于跟踪和分析学生学习活动的数据模型。

推送通知：开发系统通知功能，实时向学生推送学习进度更新。

个性化推荐系统：

算法选择：选择适用于教育领域的推荐算法，如协同过滤或内容基于推荐。

系统实现：集成推荐系统到现有平台，提供个性化学习资源。

第8周：优化与发布

安全审计：进行全面的安全审计，确保没有安全漏洞。

用户培训：组织在线或现场的用户培训会，确保用户能够有效使用新系统。

发布准备：准备发布通告，组织发布日的技术支持和监控。

持续迭代与调整

性能监控：持续监控系统性能，根据反馈优化系统配置。

用户反馈循环：建立一个反馈机制，定期收集和分析用户反馈，以指导未来的产品迭代。

技术债务管理：定期审查和重构代码，减少技术债务。

文档更新与维护

版本控制：确保所有文档都在版本控制之下，与软件版本同步更新。

知识库维护：建立和维护一个知识库，收集常见问题和解决方案，方便用户和开发者查询。