AI Agent 课程引入报告

# 1. 为什么需要 Agent

在软件工程和学术研究中，开发者和研究者经常需要查找相关资料（API、库、文档、论文）。  
传统的检索方式依赖关键词搜索，效率低且噪声大。  
AI Agent 可以通过自动化的查询、筛选和总结，大幅提升效率。  
本课程项目示例：输入研究主题，Agent 自动找到相关论文并生成报告。

# 2. 什么是 Agent

- 传统软件系统：基于规则和固定流程。  
- AI Agent：具备自主感知、规划、执行和反馈的能力。  
- 核心组成：  
 - 环境：API、数据库、工具集  
 - 大脑：大语言模型（LLM），负责推理与决策  
 - 执行器：具体调用 API、执行脚本、查询文档  
示例：  
- 文献检索 Agent（课程项目）  
- 代码生成 Agent（Copilot / Codex）  
- Bug 修复 Agent（自动定位并打补丁）

# 3. Agent 在软件工程中的应用

- 知识检索与推荐：类似 Related Paper Agent，扩展到相关代码库检索。  
- 自动化测试：生成测试用例、执行测试并汇总结果。  
- 需求分析：从自然语言需求中提取用户故事、绘制 UML 图。  
- 代码生成与维护：自动生成或调试代码。  
- 项目管理辅助：监控 issue、自动生成周报。  
说明：文献检索 Agent 的方法论可以迁移到软件工程场景。

# 4. 两阶段检索框架与 Agent

在搜索和推荐系统中，常见的架构是“召回 + 重排”：  
- 召回：用向量检索等方法快速找到一批候选项，保证广度。  
- 重排：用更复杂的模型精排，保证精度。  
在文献检索 Agent 中：  
- 召回：调用 OpenAlex、arXiv、Semantic Scholar 等 API 获取候选论文。  
- 重排：本地语义模型（如 BGE-reranker）对候选论文重新排序。  
- Agent：负责 orchestrate（调度 API、整合结果、生成解释）。  
在软件工程中类比：  
- 召回：找相关函数、commit 或 issue。  
- 重排：用上下文模型精筛。

# 5. 展望与课程关联

课程目标：通过构建一个小型 Agent，理解如何将 AI 技术应用于软件工程任务。  
项目示例：输入研究主题，自动输出相关工作报告。  
潜在扩展：  
- 输入需求文档 → 推荐相关代码库。  
- 输入报错日志 → 推荐解决方案。