AI Engineering Employee Platform

1. 项目目标

构建一个能够自主完成：

- 项目审核
- 需求分析
- 代码开发
- 自动测试
- Bug修复
- 代码Review
- 报告生成
- 多Agent协同

的 AI 工程员工平台。

最终目标：

```
产品经理提出需求
        ↓
AI Supervisor
        ↓
任务拆解
        ↓
多个专业Agent并行工作
        ↓
自动测试
        ↓
自动Review
        ↓
生成PR
        ↓
人工确认
        ↓
部署上线
```

---

2. 当前版本架构

v1

已经实现：

```
User
 ↓
LangGraph
 ↓
Sandbox
 ↓
Claude/DeepSeek
 ↓
Test
 ↓
Report
```

工作流：

```
prepare_sandbox
 ↓
review_project
 ↓
execute_code
 ↓
run_tests
 ↓
generate_report
```

---

3. 当前目录结构

```
ai-engineering-employee/

├──── .env
├──── .gitignore
├──── requirements.txt

├──── app/

│
├──── main.py
│
├──── graph.py
│
├──── state.py
│
├──── executors/
│     ├──── __init__.py
│     └──── claude_executor.py
│
├──── tools/
```

```
│   ├── __init__.py
│   └── sandbox.py
│
├── reports/
│
└── sandboxes/
```

---

4. 核心模块设计

State

EngineeringState

负责：

任务状态管理

保存：

```
user_request
repo_path
sandbox_path
plan
code_result
test_result
report
status
attempts
```

Graph

StateGraph

负责：

流程编排

例如：

```
成功
 ↓
测试

失败
 ↓
报告
```

未来：

```
失败
 ↓
重试
 ↓
修复
 ↓
再次测试
```

Sandbox

负责：

```
复制项目
```

结构：

```
sandboxes/

task_001
task_002
task_003
```

每个任务独立。

避免：

```
AI直接修改生产代码
```

Executor

负责：

```
调用具体Agent
```

当前：

```
Claude + DeepSeek
```

未来：

```
Claude
Codex
OpenHands
Gemini
Qwen
本地模型
```

统一接口：

```
run_task()
```

---

5. 未来目录结构

```
app/

executors/

 ├──── claude_executor.py
 ├──── codex_executor.py
 ├──── openhands_executor.py
 ├──── gemini_executor.py

agents/

 ├──── supervisor.py
 ├──── architect.py
 ├──── developer.py
 ├──── tester.py
 ├──── reviewer.py
 ├──── reporter.py

tools/

 ├──── sandbox.py
 ├──── git_manager.py
```

```
├── docker_manager.py
├── report_generator.py

workflows/

├── engineering_graph.py
├── review_graph.py
├── bugfix_graph.py
├── release_graph.py
```

---

6. 多Agent架构

Supervisor

负责：

```
接收需求
拆分任务
分配Agent
```

例如：

```
开发登录系统
```

Supervisor拆分：

```
1 登录页面
2 用户接口
3 JWT认证
4 单元测试
5 文档
```

Architect

负责：

```
设计方案
```

输出：

```
目录结构
数据库设计
接口设计
```

Developer

负责：

```
写代码
```

推荐：

```
Claude Code
Codex
```

Tester

负责：

```
自动测试
```

执行：

```
pytest
npm test
go test
cargo test
```

Reviewer

负责：

```
Code Review
```

检查：

```
Bug
安全问题
性能问题
代码规范
```

Reporter

负责：

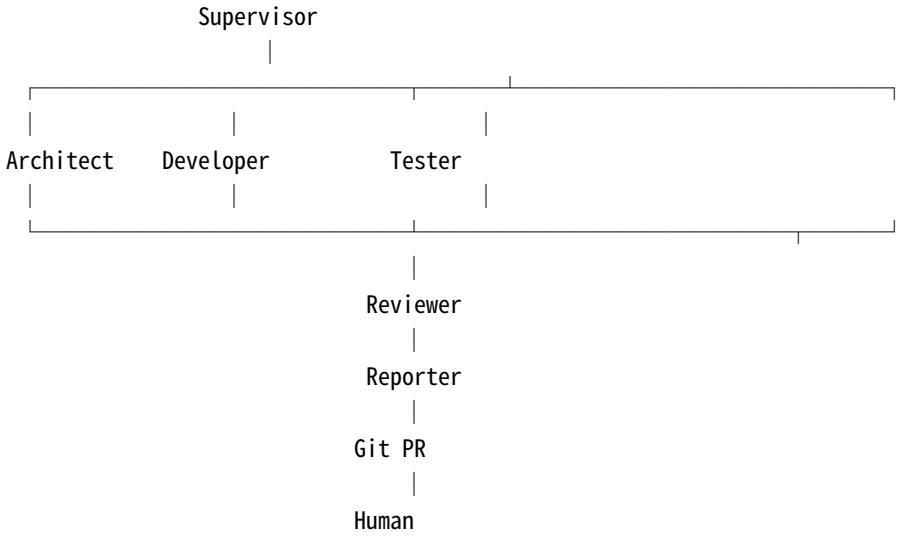
```
生成报告
```

输出：

```
修改文件
测试结果
风险分析
下一步建议
```

---

7. 最终生产架构



---

8. 开发路线图

Phase 1（已完成）

- LangGraph
- DeepSeek
- Claude Code
- Sandbox
- Report

Phase 2（下一步）

- Diff生成
- Git集成
- 自动Commit
- 自动PR

Phase 3

- Codex Agent
- OpenHands Agent
- 多Agent协同

Phase 4

- Web管理后台
- FastAPI
- Redis Queue
- PostgreSQL

Phase 5

- AI研发团队

```
1 Supervisor
3 Developer
2 Tester
1 Reviewer
```

完全自动化软件开发流水线。