

第 14 讲 · OpenHands 框架 1

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分



AGENTIC AI | LECTURE 14

OpenHands

Frameworks 1 · October 8, 2026

Carnegie Mellon University
Language Technologies Institute

1

智能体式 AI | 第14讲

OpenHands

框架 1 · 2026年10月8日

Akari Asai · 语言技术研究所 · 卡内基梅隆大学 · 11-768: AI 智能体 · 2026年秋季

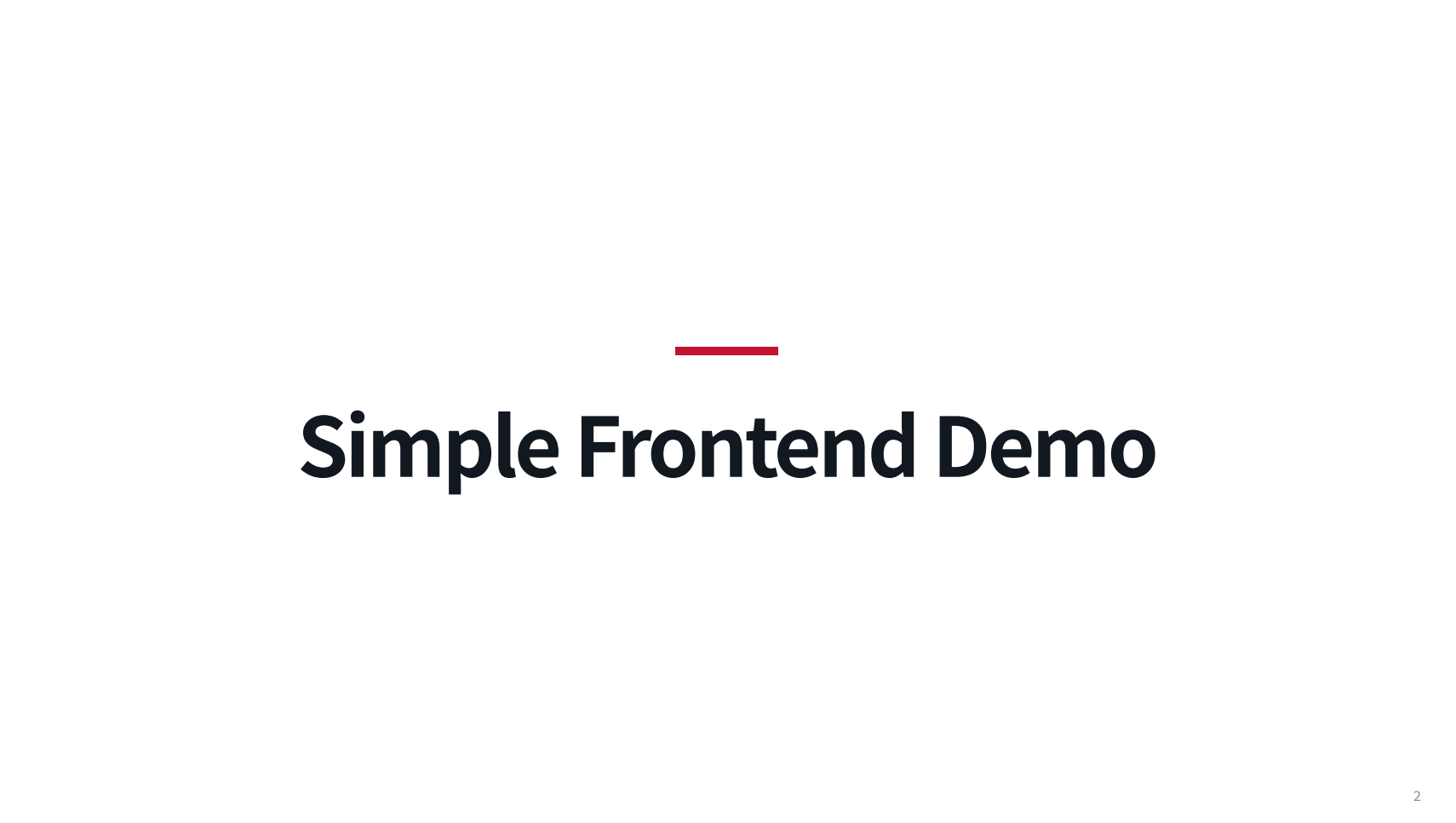
<https://akariasai.github.io/> · aasai@andrew.cmu.edu

OpenHands: 框架 1

第 14 讲 · OpenHands 框架 1

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分



简单前端演示

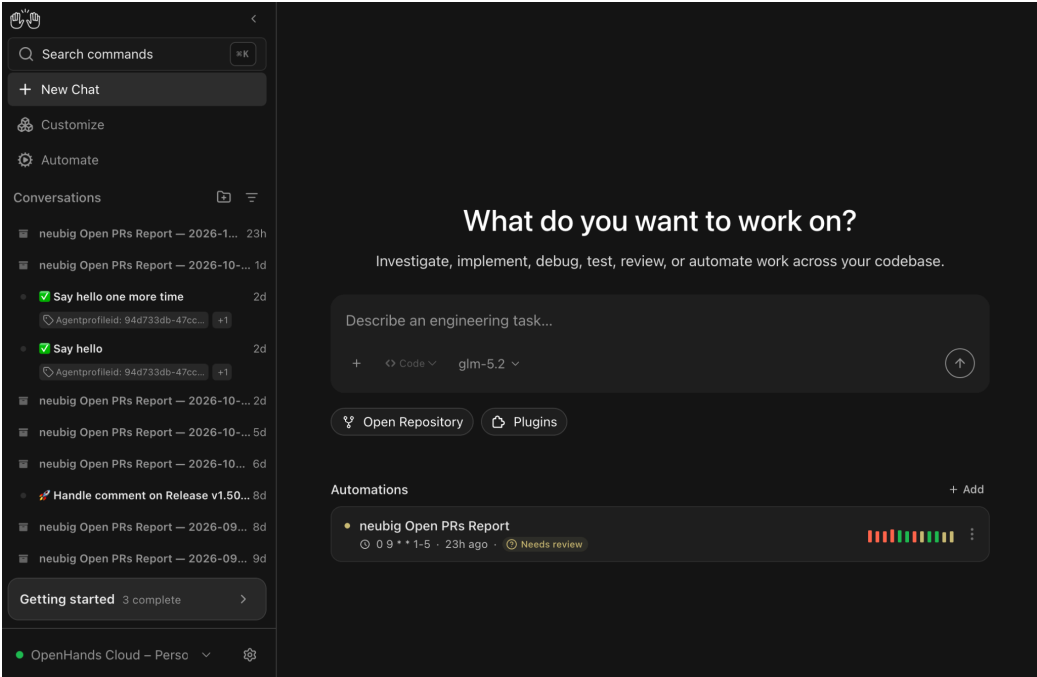
简单前端演示

第 14 讲 · OpenHands 框架 1

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Agent Canvas



OpenHands Agent Canvas · Live demo ↗

3

智能体画布（Agent Canvas）

界面文字：搜索命令；新建聊天；你想处理什么工作？在代码库中调查、实现、调试、测试、审查或自动化工作；打开代码库；自动化。

OpenHands Agent Canvas · 现场演示 ↗

译注 / 与原文翻译分开

界面截图中较小的控件文字未逐项转写。

智能体画布

第 14 讲 · OpenHands 框架 1

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分



发展历程

发展历程

第 14 讲 · OpenHands 框架 1

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Code completion

Through 2023

- Copilot and early Cursor: code suggestions
- HumanEval: single functions, unit tests
- Human-led editing, execution, and checking

A HumanEval-style task

```
def has_close_elements(  
    numbers, threshold  
):  
    """Nearby number pairs"""  
    # Function completion
```

Input: a function specification
Output: a function implementation

Chen et al. · HumanEval GitHub Copilot · 2021

5

代码补全 · 截至2023年

Copilot 和早期 Cursor：提供代码建议。

HumanEval：单个函数、单元测试。

由人主导编辑、执行和检查。

一个类似 HumanEval 的任务：

```
def has_close_elements(  
    numbers, threshold  
):  
    """Nearby number pairs"""  
    # Function completion
```

输入：函数规格说明。

输出：函数实现。

Chen 等 · HumanEval · GitHub Copilot · 2021

译注 / 与原文翻译分开

代码、标识符及代码注释保留原样。

代码补全

第 14 讲 · OpenHands 框架 1

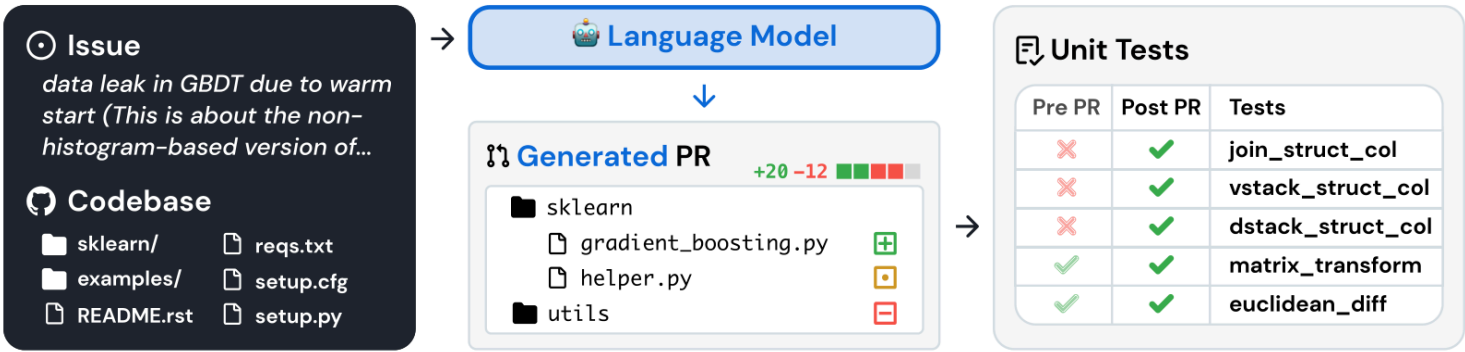
EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Repository-level evaluation

October 2023 · SWE-bench

- Real issue + repository snapshot
- Patch + repository tests; difficult early results



Jimenez et al. · SWE-bench · Figure 1

6

仓库级评测 · 2023年10月 · SWE-bench

真实问题 + 仓库快照。

提交补丁 + 运行仓库测试；早期结果表现困难。

Jimenez 等 · SWE-bench · 图1

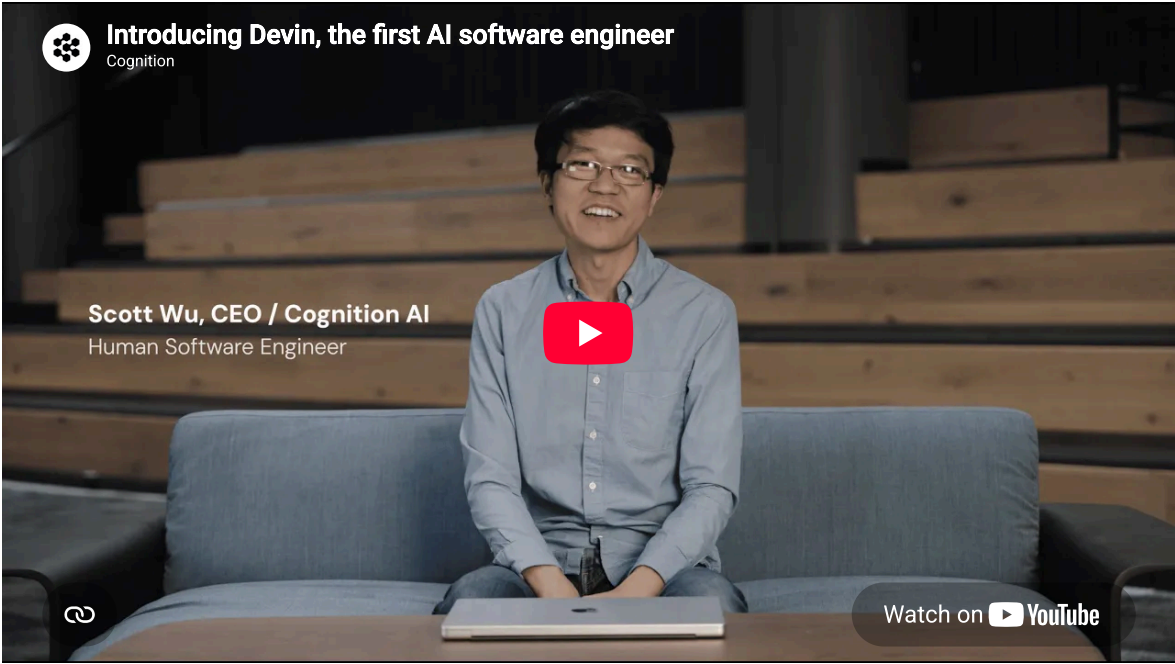
仓库级评测

第 14 讲 · OpenHands 框架 1

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Devin



Cognition · Devin launch demo · March 12, 2024 ↗

7

Devin

推出 Devin：首位 AI 软件工程师。

Cognition

观看

Cognition · Devin 发布演示 · 2024年3月12日 ↗

Devin

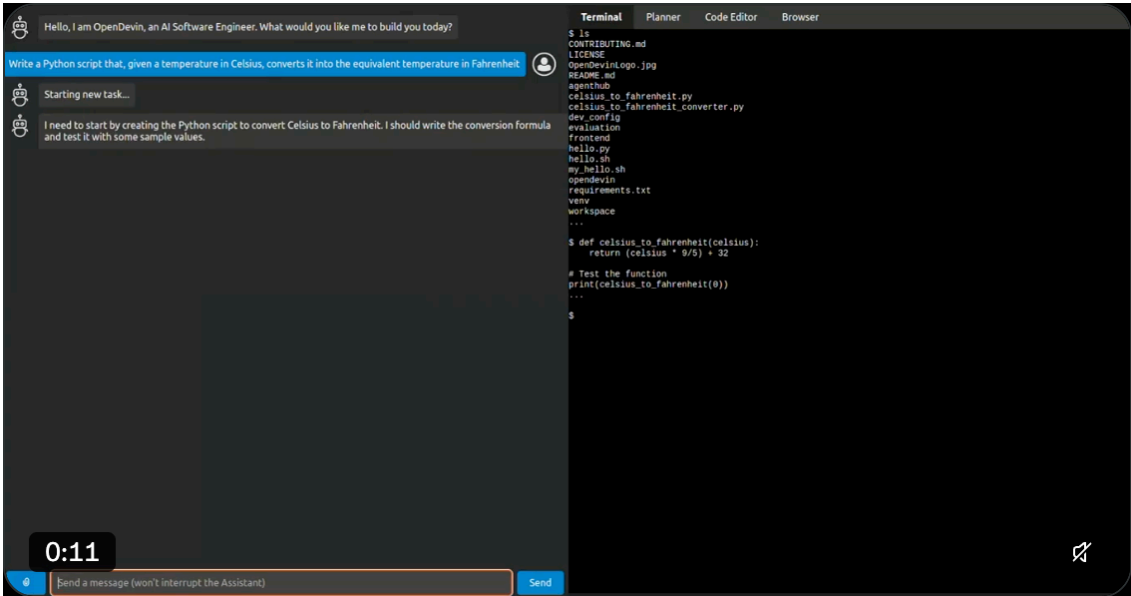
第 14 讲 · OpenHands 框架 1

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

OpenDevin

Frontend + agent + Docker sandbox · March 26, 2024



[Junyang Lin · Original callout ↗](#)

[Graham Neubig · Early frontend ↗](#)

OpenDevin

OpenDevin

前端 + 智能体 + Docker 沙箱 · 2024年3月26日

Junyang Lin · [原始说明 ↗](#) · Graham Neubig · [早期前端 ↗](#)

第 14 讲 · OpenHands 框架 1

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Early coding agents

April 2024

SWE-agent

File search, code edits, fix verification

May 2024

OpenDevin CodeAct 1.0

Code execution, terminal commands, feedback

Agent-computer interfaces as a research topic

[SWE-agent · April 2 release](#)

[CodeAct 1.0 · May 7 announcement](#)

9

早期编码智能体

2024年4月：SWE-agent。文件搜索、代码编辑、修复验证。

2024年5月：OpenDevin CodeAct 1.0。代码执行、终端命令、反馈。

智能体—计算机接口成为研究课题。

SWE-agent · 4月2日发布 · CodeAct 1.0 · 5月7日公告

早期编码智能体

第 14 讲 · OpenHands 框架 1

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Benchmark progress

2025

Better models

Reasoning, code editing, tool use

Better agent systems

Tools, test environments, context, recovery

Benchmark performance: model + agent system

Wang et al. · OpenHands Software Agent SDK · § 5

10

基准进展 · 2025年

更强的模型：推理、代码编辑、工具使用。

更完善的智能体系统：工具、测试环境、上下文、恢复能力。

基准表现 = 模型 + 智能体系统。

Wang 等 · OpenHands Software Agent SDK · §5

基准进展

第 14 讲 · OpenHands 框架 1

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Deployment

2026

- Longer runs, interruptions, parallel tasks
- Progress visibility and result inspection
- Repeatable workflows, scoped access, verification

部署 · 2026年

运行时间更长、会被中断、可并行处理任务。

让进度可见，并可检查结果。

工作流可重复、访问范围受限、结果经过验证。

OpenHands SDK · Neubig · Fujitsu 演讲 · 2026年9月

部署

第 14 讲 · OpenHands 框架 1

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

OpenHands 软件智能体 SDK



OpenHands 软件智能体 SDK

第 14 讲 · OpenHands 框架 1

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Batteries included, but flexible

Ready-to-use agent

Models, tools, state, workspaces

Customizable components

Model choice, new tools,
instructions, deployment

Shared foundation for GUI, CLI, and integrations

[OpenHands Software Agent SDK · Browse the examples ↗](#)

13

开箱即用，仍可灵活定制。

可直接使用的智能体：模型、工具、状态、工作区。

可定制的组件：模型选择、新工具、指令、部署。

为 GUI、CLI 和集成提供共同基础。

OpenHands Software Agent SDK · [浏览示例 ↗](#)

开箱即用，仍可灵活定制

第 14 讲 · OpenHands 框架 1

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Model and tools

```
llm = LLM(  
    model=os.getenv("LLM_MODEL", "gpt-5.5"),  
    api_key=os.getenv("LLM_API_KEY"),  
)  
  
agent = Agent(  
    llm=llm,  
    tools=[  
        Tool(name=TerminalTool.name),  
        Tool(name=FileEditorTool.name),  
        Tool(name=TaskTrackerTool.name),  
    ],  
)
```

[Hello-world example](#) · [Browse the full source](#)

14

- Model choice
- Terminal, editor, task tracker
- Reusable agent configuration

模型与工具

模型与工具

```
llm = LLM(  
    model=os.getenv("LLM_MODEL", "gpt-5.5"),  
    api_key=os.getenv("LLM_API_KEY"),  
)
```

模型选择

```
agent = Agent(  
    llm=llm,  
    tools=[  
        Tool(name=TerminalTool.name),  
        Tool(name=FileEditorTool.name),  
        Tool(name=TaskTrackerTool.name),  
    ],  
)
```

终端、编辑器、任务跟踪器。

可复用的智能体配置。

[Hello-world 示例](#) · [浏览完整源码](#)

译注 / 与原文翻译分开

代码、环境变量名和 API 字段保留原样。

第 14 讲 · OpenHands 框架 1

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Conversation

```
cwd = os.getcwd()
conversation = Conversation(agent=agent, workspace=cwd)

conversation.send_message(
    "3 project facts in FACTS.txt"
)
conversation.run()
```

- Workspace: files and execution
- Conversation: task, actions, observations
- Run: action–observation loop

[Hello-world example](#) · [Browse the full source](#)

15

对话

对话

```
cwd = os.getcwd()
conversation = Conversation(agent=agent, workspace=cwd)
```

```
conversation.send_message(
    "3 project facts in FACTS.txt"
)
conversation.run()
```

工作区：文件与执行环境。

对话：任务、动作、观测。

运行：动作—观测循环。

Hello-world 示例 · [浏览完整源码](#)

译注 / 与原文翻译分开

代码和提示词字符串保留原样。

第 14 讲 · OpenHands 框架 1

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Custom tools

```
class GrepAction(Action):
    pattern: str = Field(
        description="Regex to search for"
    )
    path: str = Field(default=".")

class GrepObservation(Observation):
    matches: list[str] = Field(
        default_factory=list
    )
```

- Action: typed input
- Executor: search implementation
- Observation: model-readable result

[Custom grep tool · Browse the full source ↗](#)

16

自定义工具

自定义工具

```
class GrepAction(Action):
    pattern: str = Field(
        description="Regex to search for"
    )
    path: str = Field(default=".")
```

```
class GrepObservation(Observation):
    matches: list[str] = Field(
        default_factory=list
    )
```

Action：类型化输入。

Executor：搜索实现。

Observation：模型可读的结果。

自定义 grep 工具 · 浏览完整源码 ↗

译注 / 与原文翻译分开

代码、字段名和 API 类型保留原样。

第 14 讲 · OpenHands 框架 1

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Conversation persistence

```
conversation = Conversation(  
    agent=agent,  
    workspace=cwd,  
    persistence_dir="./.conversations",  
    conversation_id=conversation_id,  
)
```

- Saved events and conversation state
- Restoration: same ID and storage
- Workspace persistence: a separate concern

[Persistence example](#) · [Browse the full source](#)

17

对话持久化

对话持久化

```
conversation = Conversation(  
    agent=agent,  
    workspace=cwd,  
    persistence_dir="./.conversations",  
    conversation_id=conversation_id,  
)
```

保存事件与对话状态。

恢复对话：使用相同 ID 和存储位置。

工作区持久化是另一个独立问题。

持久化示例 · [浏览完整源码](#)

译注 / 与原文翻译分开

代码和 API 字段保留原样。

第 14 讲 · OpenHands 框架 1

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Context management

```
condenser = LLMSummarizingCondenser(  
    llm=llm.model_copy(update={"usage_id": "condenser"}),  
    max_size=10,  
    keep_first=2,  
)  
agent = Agent(llm=llm, tools=tools, condenser=condenser)
```

- Older events → summary
- Original instructions + recent work
- Potential detail loss; files and tests as evidence

[Context condenser](#) · [Browse the full source](#)

18

上下文管理

上下文管理

```
condenser = LLMSummarizingCondenser(  
    llm=llm.model_copy(update={"usage_id": "condenser"}),  
    max_size=10,  
    keep_first=2,  
)  
agent = Agent(llm=llm, tools=tools, condenser=condenser)
```

较早的事件 → 摘要。

保留原始指令 + 最近的工作。

可能丢失细节；可用文件和测试作为证据。

[上下文压缩器](#) · [浏览完整源码](#)

译注 / 与原文翻译分开

代码、变量和 API 字段保留原样。

第 14 讲 · OpenHands 框架 1

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Remote execution

```
with DockerWorkspace(  
    server_image=server_image,  
    platform=detect_platform(),  
) as workspace:  
    agent = get_default_agent(llm=llm, cli_mode=True)  
    conversation = Conversation(agent=agent, workspace=workspace)  
    conversation.send_message(  
        "3 project facts in FACTS.txt"  
    )  
    conversation.run()
```

- Agent Server inside Docker
- Same conversation API; remote task and events

[Docker workspace](#) · [Browse the full source](#)

19

远程执行

远程执行

```
with DockerWorkspace(  
    server_image=server_image,  
    platform=detect_platform(),  
) as workspace:  
    agent = get_default_agent(llm=llm, cli_mode=True)  
    conversation = Conversation(agent=agent, workspace=workspace)  
    conversation.send_message(  
        "3 project facts in FACTS.txt"  
    )  
    conversation.run()
```

Docker 内运行 Agent Server。

使用相同的对话 API；任务与事件在远端运行。

Docker 工作区 · [浏览完整源码](#)

译注 / 与原文翻译分开

代码、类名、API 字段和提示词字符串保留原样。

第 14 讲 · OpenHands 框架 1

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Action confirmation

```
conversation.set_confirmation_policy(AlwaysConfirm())
conversation.send_message("Directory listing with ls -la")
conversation.run()
```

- Pause before action execution
- Pending action → human decision
- Rejection → revised plan

Confirmation example · [Browse the full source ↗](#)

20

动作确认

动作确认

```
conversation.set_confirmation_policy(AlwaysConfirm())
conversation.send_message("Directory listing with ls -la")
conversation.run()
```

执行动作前暂停。

待执行动作 → 由人作出决定。

遭到拒绝 → 修改计划。

确认示例 · [浏览完整源码 ↗](#)

译注 / 与原文翻译分开

代码、API 字段和提示词字符串保留原样。

第 14 讲 · OpenHands 框架 1

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

单智能体与多智能体

Single Agent vs. Multi-agent

21

单智能体与多智能体

第 14 讲 · OpenHands 框架 1

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Specialized roles

Early multi-agent workflows: specialized roles

Manager	Issue plan
Reproducer	Bug reproduction
Fault localizer	Bug location
Editor	Code changes
Verifier	Fix verification

Chen et al. · CodeR · § 2 Related: MetaGPT

22

早期多智能体 workflow：专业化角色。

管理者：制定问题计划。

复现者：复现错误。

故障定位者：定位错误。

编辑者：修改代码。

验证者：验证修复。

Chen 等 · CodeR · §2 · 相关工作：MetaGPT

专业化角色

第 14 讲 · OpenHands 框架 1

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

A single-agent baseline

- One agent: reproduce, locate, edit, test
- Shared context across activities
- Separate roles: handoffs and maintenance
- Additional agents for a clear purpose

Neubig · Don't Sleep on Single-agent Systems · 2024

23

一个智能体：复现、定位、编辑、测试。

多个活动共享上下文。

分设角色：需要交接，也增加维护工作。

只有目标明确时才增加智能体。

Neubig · Don't Sleep on Single-agent Systems · 2024

单智能体基线

第 14 讲 · OpenHands 框架 1

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

General-purpose tools

A small tool set, many combinations

Edit files
Source, configuration, documentation

Run commands
Search, dependencies, tests

Browse
Documentation and visual inspection

少量工具，多种组合。

编辑文件：源代码、配置、文档。

运行命令：搜索、依赖项、测试。

浏览：文档与视觉检查。

Neubig · 单智能体设计 · 通用动作空间

通用工具

第 14 讲 · OpenHands 框架 1

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Skills

Additional instructions for the same agent

```
agent_context = AgentContext(  
    skills=[Skill(  
        name="repo.md",  
        content=repo_instructions,  
        trigger=None,  
    )],  
)  
agent = Agent(  
    llm=llm, tools=tools,  
    agent_context=agent_context,  
)
```

[Skill activation example](#) · [Browse the full source](#)

25

- Project conventions and recurring procedures
- Relevant instructions on demand
- Earlier name: microagents

向同一个智能体添加指令。

```
agent_context = AgentContext(  
    skills=[Skill(  
        name="repo.md",  
        content=repo_instructions,  
        trigger=None,  
    )],  
)  
agent = Agent(  
    llm=llm, tools=tools,  
    agent_context=agent_context,  
)
```

项目约定与重复流程。

按需提供相关指令。

早期名称：microagents。

[技能激活示例](#) · [浏览完整源码](#)

译注 / 与原文翻译分开

代码、变量、API 字段和 microagents 名称保留原样。

技能 (Skills)

第 14 讲 · OpenHands 框架 1

EN / 英文官方原页 · 保持原样

MCP tools

Model Context Protocol: external tool connections

```
mcp_config = {
    "repomix": MCPServer(
        command="npx", args=["-y", "repomix@1.4.2", "--mcp"]
    ),
}
agent = Agent(llm=llm, tools=tools, mcp_config=mcp_config)
```

- Server-provided tool descriptions and execution
- Built-in and external tools together
- Server access and trust requirements

[MCP integration example](#) · [Browse the full source](#) ↗

26

MCP 工具

ZH / 中文译文 · 第 1/1 部分

MCP 工具

模型上下文协议（Model Context Protocol）： 外部工具连接。

```
mcp_config = {
    "repomix": MCPServer(
        command="npx", args=["-y", "repomix@1.4.2", "--mcp"]
    ),
}
agent = Agent(llm=llm, tools=tools, mcp_config=mcp_config)
```

由服务器提供工具说明并执行工具。

内置工具与外部工具可以一起使用。

需要具备服务器访问权限并满足信任要求。

MCP 集成示例 · [浏览完整源码](#) ↗

译注 / 与原文翻译分开

代码、MCP 标识、命令和 API 字段保留原样。

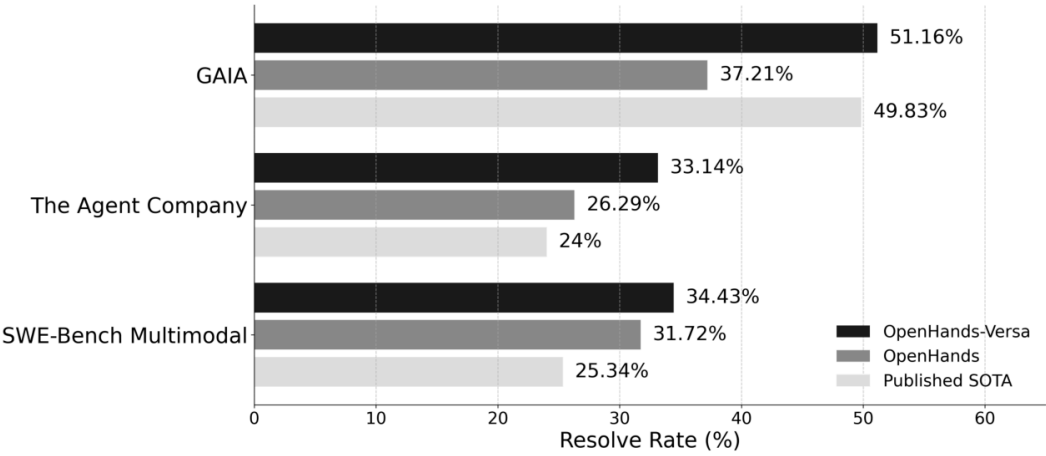
第 14 讲 · OpenHands 框架 1

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Generalist problem solving

OpenHands-Versa: coding + visual browsing + search + file access



Shared tools across coding, research, and office work

Soni et al. · OpenHands-Versa · Figure 1

27

通才式问题求解。

OpenHands-Versa：编码 + 视觉浏览 + 搜索 + 文件访问。

图表：解决率（%），横轴 0、10、20、30、40、50、60。GAIA：OpenHands-Versa 51.16%，OpenHands 37.21%，已发表的 SOTA（当前最佳水平）49.83%。The Agent Company：OpenHands-Versa 33.14%，OpenHands 26.29%，已发表的 SOTA 24%。SWE-Bench Multimodal：OpenHands-Versa 34.43%，OpenHands 31.72%，已发表的 SOTA 25.34%。

编码、研究和办公任务共用工具。

Soni 等 · OpenHands-Versa · 图1

通才式问题求解

第 14 讲 · OpenHands 框架 1

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Division of labor

Parallel work	Independent issues or repositories
Independent review	Patch checking; separate task and permissions
Cost control	Cheaper models for suitable tasks

Task boundaries, shared context, coordination cost

SDK · Delegation example

Fujitsu talk · Pages 11, 13, 16

28

并行工作：处理相互独立的问题或代码库。

独立审查：检查补丁；任务与权限彼此分开。

成本控制：适合的任务使用成本较低的模型。

需要考虑任务边界、共享上下文和协调成本。

SDK · 委派示例 · Fujitsu 演讲 · 第11、13、16页

分工

第 14 讲 · OpenHands 框架 1

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

智能体托管与交互界面

Agent Hosting and Interface

29

智能体托管与交互界面

第 14 讲 · OpenHands 框架 1

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Local and cloud execution

Local computer

Existing files, services, and context
Machine availability during execution

Cloud workspace

Continued work with the laptop off
Remote repository, dependencies, credentials

SDK · Remote workspace examples

30

本地计算机：使用现有文件、服务和上下文；执行期间机器必须保持可用。

云端工作区：即使笔记本电脑关机也能继续工作；使用远程代码库、依赖项和凭据。

SDK · 远程工作区示例

本地与云端执行

第 14 讲 · OpenHands 框架 1

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Interface choices

Execution location × interaction interface

	Local execution	Cloud execution
IDE	Editor workspace	Remote workspace
GUI	Desktop application	Browser application
CLI	Local terminal	Remote session
Integration	Local service	Hosted worker

31

执行位置 × 交互界面。

IDE：本地执行——编辑器工作区；云端执行——远程工作区。

GUI：本地执行——桌面应用；云端执行——浏览器应用。

CLI：本地执行——本地终端；云端执行——远程会话。

集成：本地执行——本地服务；云端执行——托管工作进程。

界面选择

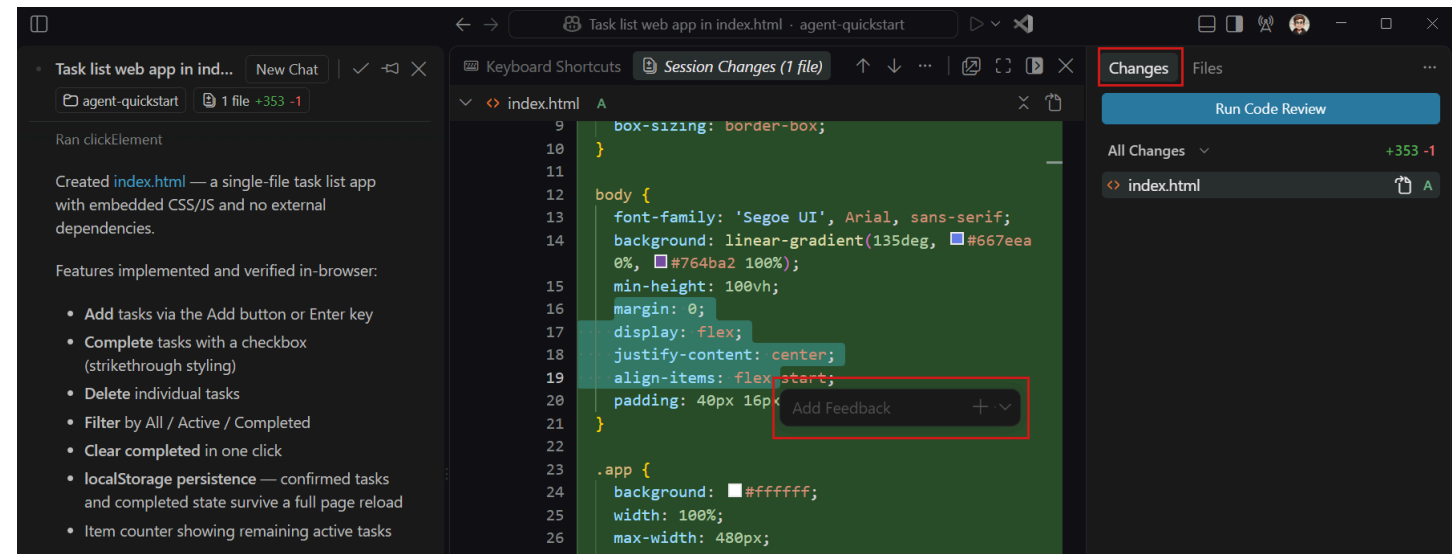
第 14 讲 · OpenHands 框架 1

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

IDE

- Familiar coding workspace
- Files, diffs, and tests
- Task management and review



Microsoft · VS Code with GitHub Copilot · Change review ↗

32

熟悉的编码工作区。

文件、差异和测试。

任务管理与审查。

Microsoft · VS Code with GitHub Copilot · 变更审查 ↗

译注 / 与原文翻译分开

编辑器截图内的代码和细小控件文字未逐项转写。

IDE

第 14 讲 · OpenHands 框架 1

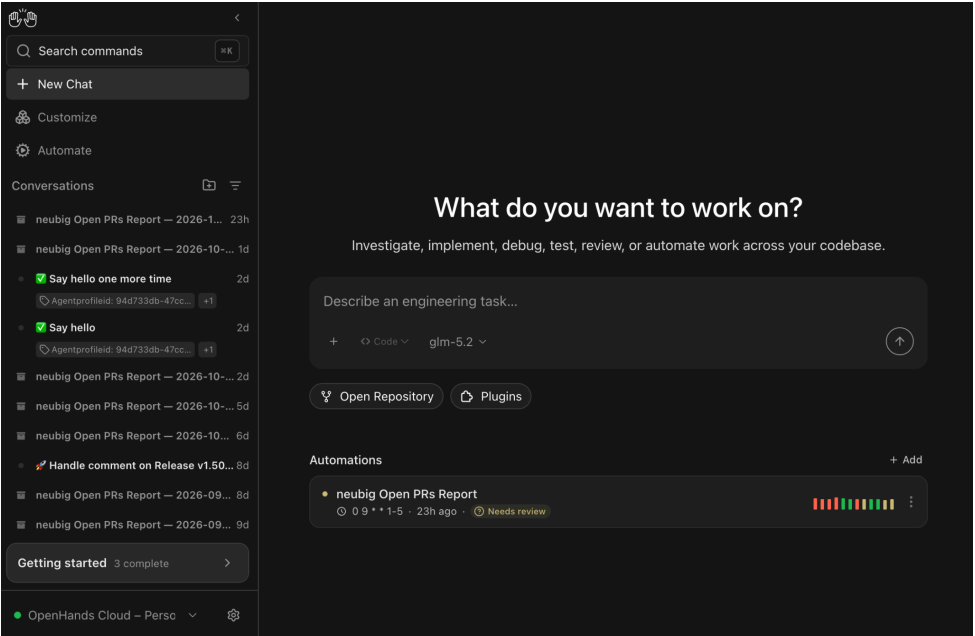
EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Custom GUI

A workspace organized around agent tasks

- Progress and parallel runs
- Previews, diffs, and approvals
- Setup and workspace management



[OpenHands Agent Canvas · Revisit the opening demo ↗](#)

33

围绕智能体任务组织的工作区。

查看进度并并行运行任务。

预览、差异与审批。

设置与工作区管理。

界面文字：开始使用；打开代码库；你想处理什么工作？插件。

OpenHands Agent Canvas · [重看开场演示 ↗](#)

译注 / 与原文翻译分开

界面截图中较小的控件文字未逐项转写。

自定义 GUI

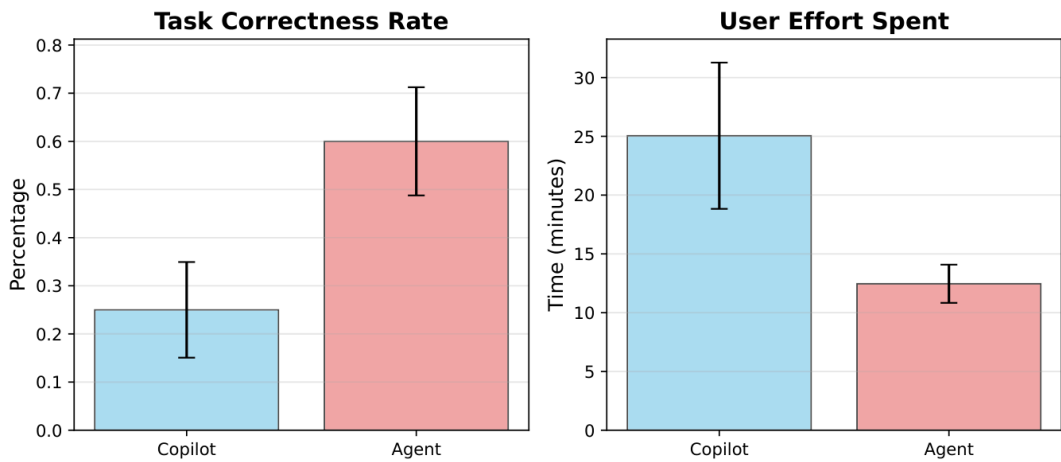
第 14 讲 · OpenHands 框架 1

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Human effort and oversight

20 regular copilot users · Python tasks · Copilot vs. OpenHands



Lower active effort; continued need for understanding and oversight

Chen et al. · Code with Me or for Me? · Figure 4

20名经常使用 Copilot 的用户 · Python 任务 · Copilot 与 OpenHands 对比。
任务正确率图：纵轴为百分比（0.0–0.8）；Copilot 约 0.25，Agent 约 0.60。
用户投入时间图：纵轴为时间（分钟）；Copilot 约25分钟，Agent 约12.5分钟。
投入的主动操作更少；仍需要理解与监督。

Chen 等 · Code with Me or for Me? · 图4

译注 / 与原文翻译分开

柱形为原图目测近似值；误差线表示范围，但图中未给出数值标签。

人的投入与监督

第 14 讲 · OpenHands 框架 1

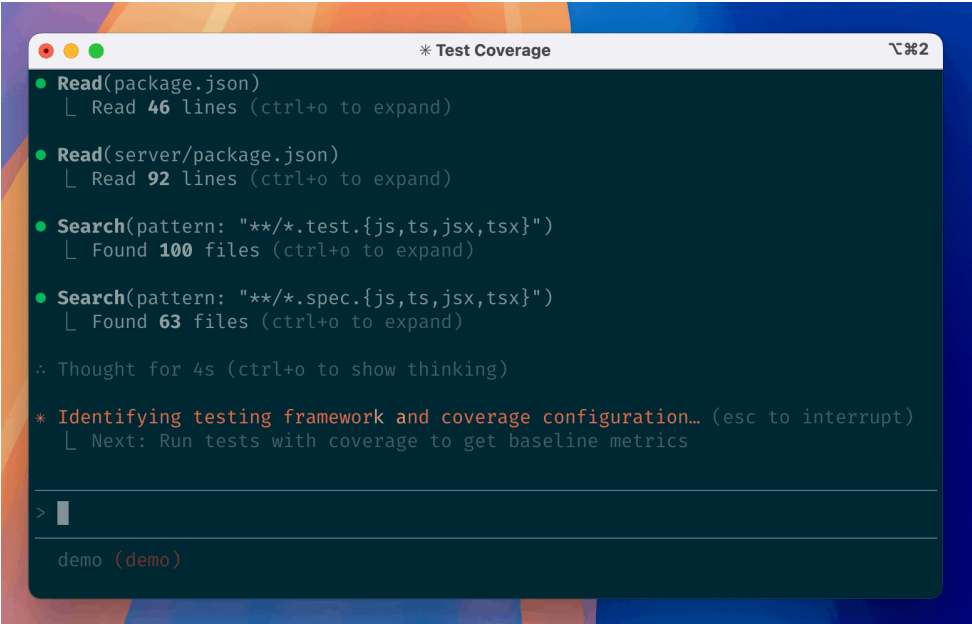
EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

CLI

A familiar interface
across environments

- Local and remote terminals
- Shell commands and scripts
- Separate visual previews



Anthropic · Claude Code · Terminal demo ↗

35

跨环境使用熟悉的界面。

本地与远程终端。

Shell 命令与脚本。

使用独立的可视化预览。

Anthropic · Claude Code · 终端演示 ↗

译注 / 与原文翻译分开

截图底部可辨文字为“测试覆盖率”和“正在识别测试框架及覆盖率配置.....”。

CLI

第 14 讲 · OpenHands 框架 1

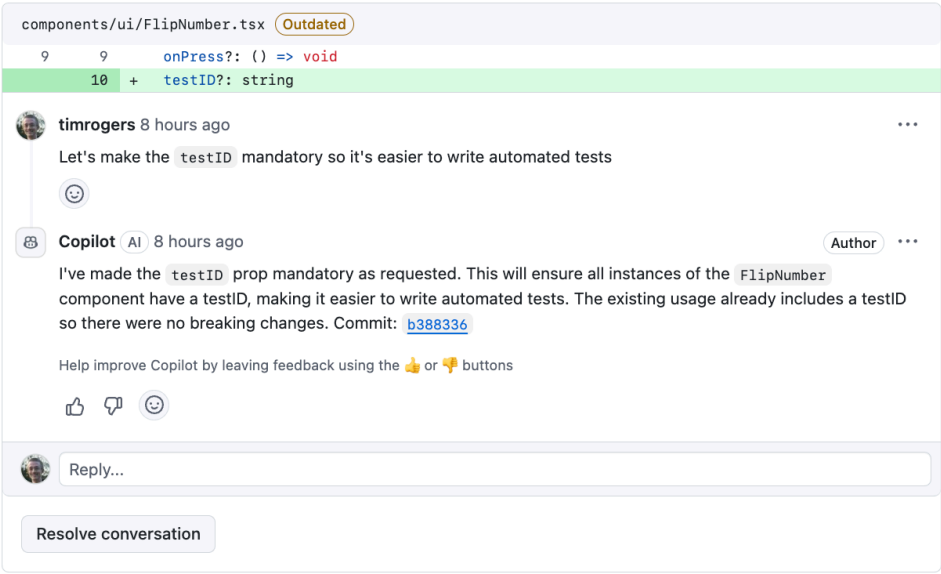
EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Integrations

Agents in existing work channels

- Issues: task context
- PRs: diffs, tests, review
- Chat: requests and results



GitHub · Copilot coding agent · PR review ↗

在现有工作渠道中使用智能体。

Issues：任务上下文。

PR：差异、测试、审查。

聊天：请求与结果。

GitHub · Copilot coding agent · PR 审查 ↗

译注 / 与原文翻译分开

示例 PR 截图含一条要求将 testID 属性设为必填的评论，以及 Copilot 回复称已将该属性设为必填、现有用法已包含 testID，因此没有破坏性变更；代码文件和控件文字未逐项转写。

集成

第 14 讲 · OpenHands 框架 1

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

软件开发自动化

Automations for Software Development

37

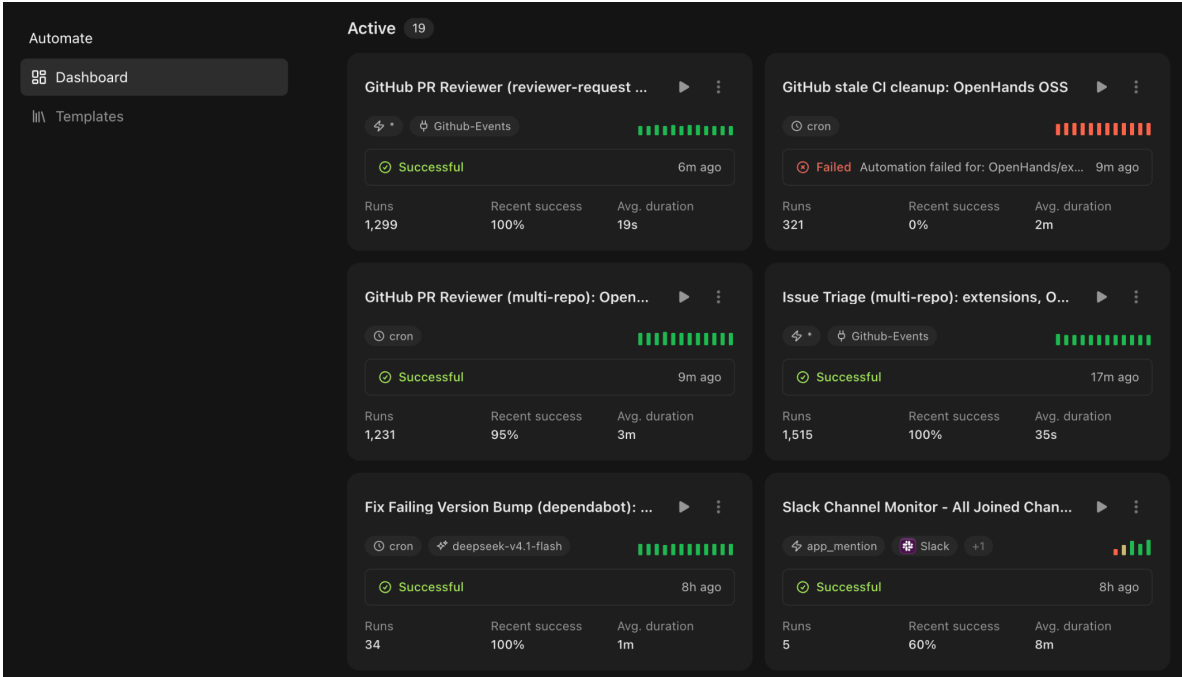
软件开发自动化

第 14 讲 · OpenHands 框架 1

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Automation dashboard



OpenHands OSS automation demo ↗

38

自动化仪表板。侧栏：自动化、仪表板、模板。活动自动化：19个。

GitHub PR Reviewer (reviewer-request...) ：成功，6分钟前；运行 1,299 次；近期成功率 100%；平均时长 19秒。

GitHub stale CI cleanup: OpenHands OSS：失败，9分钟前；运行 321 次；近期成功率 0%；平均时长 2分钟。

GitHub PR Reviewer (multi-repo) ：成功，9分钟前；运行 1,231 次；近期成功率 95%；平均时长 3分钟。

Issue Triage (multi-repo) ：成功，17分钟前；运行 1,515 次；近期成功率 100%；平均时长 35秒。

Fix Failing Version Bump (dependabot) ：成功，8小时前；运行 34 次；近期成功率 100%；平均时长 1分钟。

Slack Channel Monitor - All Joined Channels：成功，8小时前；运行5次；近期成功率60%；平均时长8分钟。

OpenHands OSS 自动化演示 ↗

译注 / 与原文翻译分开

卡片标题在截图中被截断的部分以省略号表示；仅转录可辨识内容。

自动化仪表板

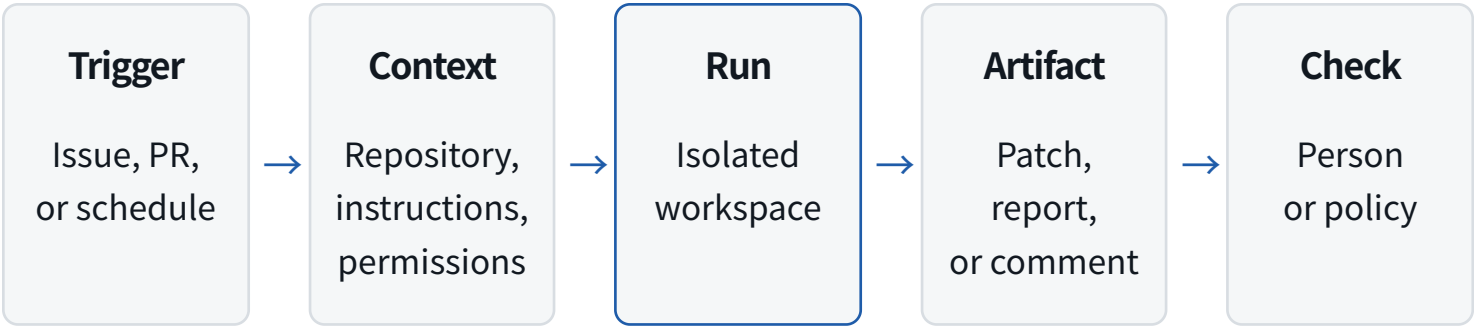
第 14 讲 · OpenHands 框架 1

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

An automation run

Event → bounded task → inspectable result



Adapted from Neubig · Fujitsu talk · Page 8

39

事件 → 有界任务 → 可检查的结果。

触发条件：Issue、PR 或定时计划。

上下文：代码库、指令、权限。

运行：隔离的工作区。

产物：补丁、报告或评论。

检查：由人员或策略完成。

改编自 Neubig · Fujitsu 演讲 · 第8页

一次自动化运行

第 14 讲 · OpenHands 框架 1

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Automation demo

Existing issue-triage or PR-review run

- Trigger and instructions
- Repository and permissions
- Input event → run → output
- Output quality against the original task

[OpenHands OSS automation demo ↗](#)

40

运行现有的 Issue 分类或 PR 审查自动化。

触发条件与指令。

代码库与权限。

输入事件 → 运行 → 输出。

依据原始任务评估输出质量。

OpenHands OSS 自动化演示 ↗

自动化演示

第 14 讲 · OpenHands 框架 1

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Software factory

Recurring tasks, explicit handoffs, checks

Issue triage	Request and acceptance criteria
Implementation	Patch, tests, pull request
Code review	Independent verification
Merge check	Passing checks and approval on the current commit

重复性任务、明确的交接、检查。

Issue 分类：请求与验收标准。

实现：补丁、测试、拉取请求。

代码审查：独立验证。

合并检查：针对当前提交，通过检查并获得批准。

改编自 Fujitsu 演讲 · 第13页 · OpenHands · 软件工厂示例

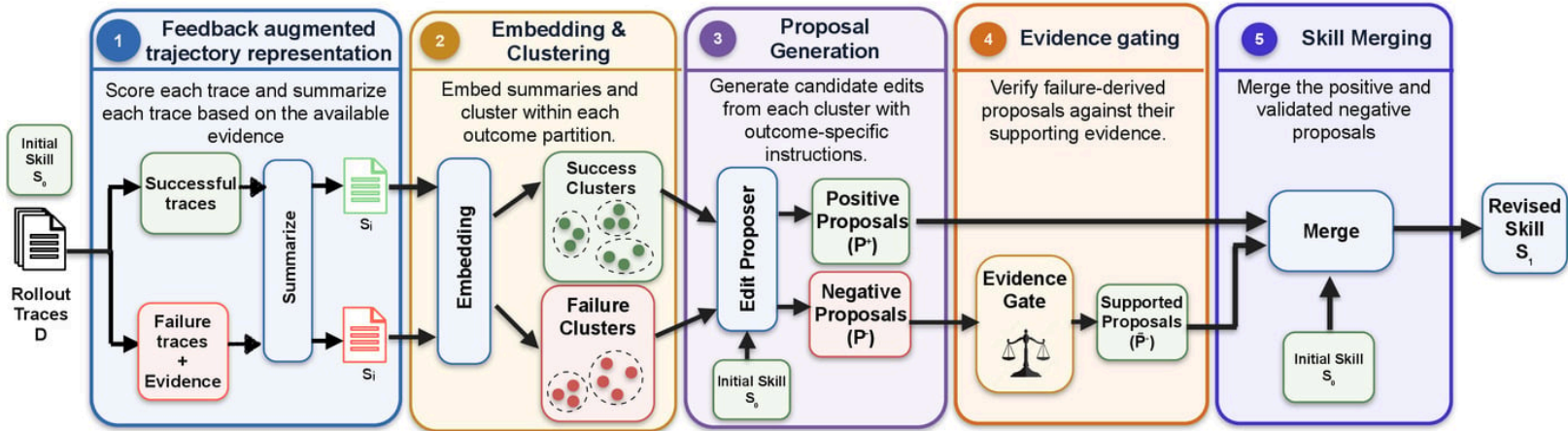
软件工厂

第 14 讲 · OpenHands 框架 1

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Learning from past runs



Khatry et al. · SkillRefiner · Figure 1

42

从过往运行中学习。

输入：轨迹采样 D 。

- 1. 由反馈增强的轨迹表示：根据可用证据为每条轨迹评分并生成摘要；将成功轨迹与失败轨迹 + 证据分别摘要，生成摘要 S_i 。
- 2. 嵌入与聚类：嵌入摘要，并在每个结果分区内聚类；形成成功簇与失败簇。
- 3. 提案生成：依据各簇生成候选编辑，并使用针对结果的专门指令；编辑提案器基于初始技能 S_0 ，生成正向提案 (P^+) 和负向提案 (P^-)。
- 4. 证据门控：根据支持证据核验由失败案例产生的提案；筛出有证据支持的提案 (P^-_s)。
- 5. 技能合并：合并正向提案与经过验证的负向提案，并结合初始技能 S_0 ，得到修订后的技能 S_1 。

Khatry 等 · SkillRefiner · 图1

译注 / 与原文翻译分开

图中 “ P^-_s ” 的下标在原图较小，按可辨字形转写。

从过往运行中学习

第 14 讲 · OpenHands 框架 1

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Skill refinement results

Held-out tasks · Accuracy (%) and PR-review F1

Task	Init.	Model	Initial	GEPA	T2S	SKILLREFINER
SpreadsheetBench	LLM	GPT-5.4-mini	77.5	78.5	79.0	84.0 [†] _*
	LLM	MiniMax-M2.7	68.0	75.0	69.5	76.0 _*
	Human	GPT-5.4-mini	86.0	86.0	86.5	86.5
	Human	MiniMax-M2.7	79.0	73.0	81.5	84.5 [†]
DAPO-Math	LLM	GPT-5.4-mini	90.0	90.0	91.0	91.0
	LLM	MiniMax-M2.7	53.5	53.5	61.0	64.0 [†]
PR Review	Human	GPT-5.4-mini	0.16	—	0.13	0.18 _*
	Human	MiniMax-M2.7	0.11	—	0.13	0.15

Higher scores than initial skills across both agent models

Khatry et al. · SkillRefiner · Table 1

技能改进结果

留出任务 · 准确率 (%) 与 PR 审查 F1。

表头：任务 | 初始化对象 | 模型 | 初始值 | GEPA | T2S | SkillRefiner。

SpreadsheetBench：LLM, GPT-5.4-mini: 77.5、78.5、79.0、84.0[†]; LLM, MiniMax-M2.7: 68.0、75.0、69.5、76.0_{*}; Human, GPT-5.4-mini: 86.0、86.0、86.5、86.5; Human, MiniMax-M2.7: 79.0、73.0、81.5、84.5[†]。

DAPO-Math：LLM, GPT-5.4-mini: 90.0、90.0、91.0、91.0; LLM, MiniMax-M2.7: 53.5、53.5、61.0、64.0[†]。

PR Review：Human, GPT-5.4-mini: 0.16、—、0.13、0.18_{*}; Human, MiniMax-M2.7: 0.11、—、0.13、0.15。

两种智能体模型的分数都高于初始技能。

Khatry 等 · SkillRefiner · 表1

译注 / 与原文翻译分开

[†] 和 _{*} 为表中原有标记；本页图中未提供其说明。

第 14 讲 · OpenHands 框架 1

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分



Questions?

Next Class: LangGraph

44

问题与答疑?

下节课: LangGraph

问题与答疑