PRODUCT ROADSHOW · 产品路演

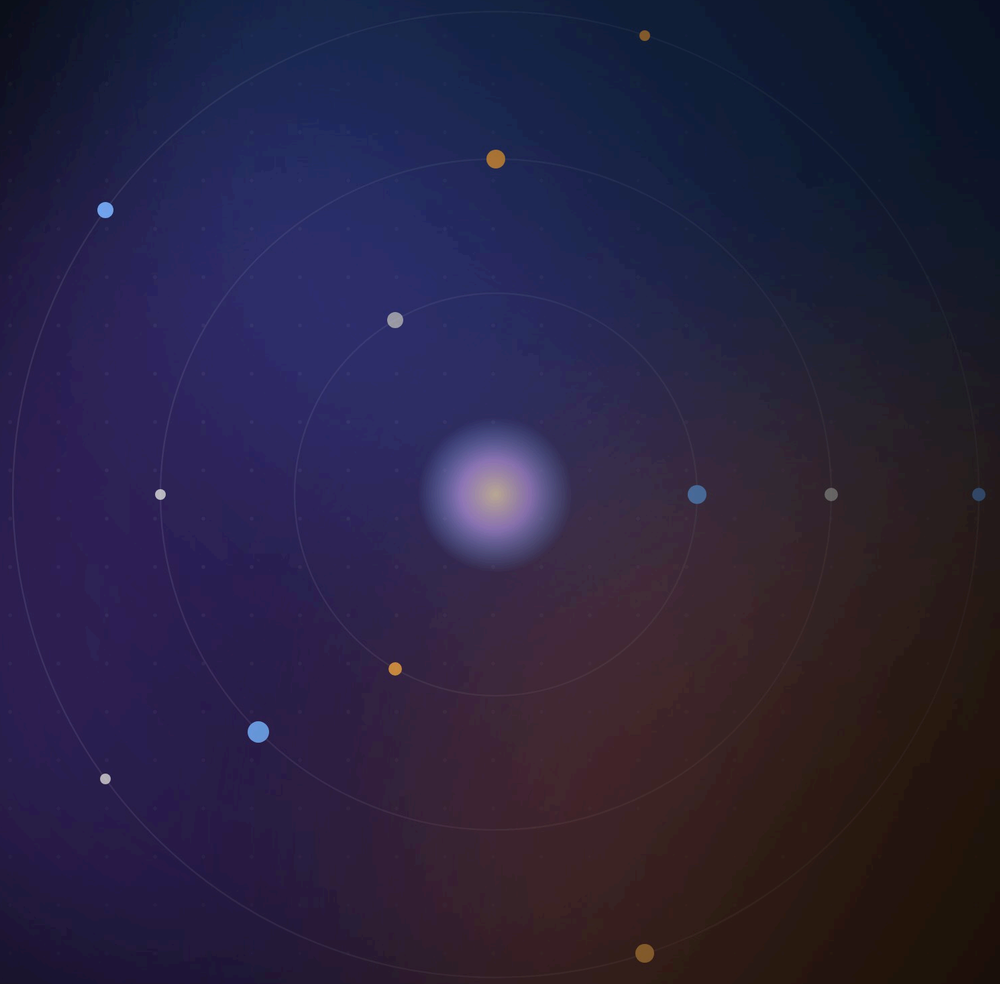
# EAT

EASY AGENT TEAM

# Agent 的企业 IM

人靠 IM 在企业里协作，Agent 也需要一个。

找得到人　拿得到数据　交付得出去



# 能力不再是瓶颈，障碍才是

**AI 已经把能力补上了。**不懂代码，AI 来写；复杂的活，AI 来做。

| 一件事 | 能力 AI 能做到 | 障碍 现实卡在 |
| --- | --- | --- |
| **排查一个用户问题** | **读日志、查数据，定位根因** | 没有日志和数据，只能找研发、等 |
| **分析一份业务数据** | **写查询、跑分析，出结论** | 拿不到数据，找数据同学排队 |
| **分享一个好用的工具** | **做出一个完整的应用** | 没地方上线，只能留在自己电脑里 |
| **搭一个自己的中台** | **贴合需求，随工作迭代** | 数据、权限、上线，处处要找人 |

## AI 补上了能力，补不上障碍。

企业里天然有一道屏障。挡在中间的，是两道墙 **技术鸿沟** **协作税**

# 提出需求之后，还有多少技术问题

需求 × 1

**「我想要一个页面，**
**每天看看哪些客户的用量不正常。」**

需求说完了。

**听不懂的信息，**
**答不了的问题。**

鸿沟不在写代码，在代码之外。

AI 的反问

01 数据从哪里取?

02 业务模型是什么样?

03 代码在哪里上线?

04 数据存在哪里?

05 密钥怎么拿？能写进代码吗?

06 模型怎么接?

07 出了错去哪看?

Schema Token CI/CD ORM VPC Migration ……

# 找人帮忙，要交多少税

**协作税**：工作每跨过一个人、一个角色，都要交一笔税。



TAX 01

## 转述税

听不懂 AI 在说什么，就讲不清自己卡在哪。信息在路上丢失，语言在路上走样。



TAX 02

## 等待税

权限、数据、答案都握在别人手里。一来一回，时间和精力都耗在等上。



TAX 03

## 放弃税

不是本职工作，就没人愿意一直等下去。好想法，停在了半路。

还没等到 AI 发挥作用，

# 人就先被这些成本劝退了。

# EAT 如何支撑企业中的 AI 工作

**工作在这里发生**
PEOPLE & AGENTS

人 | 电脑本地的 Agent | 个人云端的 Agent | 企业云端的 Agent

**EAT**
**企业协调层**

填平技术鸿沟

**可理解**
规范、知识、代码与文档，过滤后让 Agent 读懂

**可调用**
数据、凭证、模型、部署，按权限取用，按审批放行

吃掉协作税

**可流动**
问题找到对的人，工作在人和 Agent 之间流转

**可积累**
解决过的问题，变成团队下一次的能力

**企业的上下文信息**
CONTEXT & SYSTEMS

代码仓库 | 数据库 | 群聊对话 | 企业文档知识库 | 云资源 | 每个人的电脑

## 员工不用关心底层细节，背后的问题交给 Agent。

EAT 不替代任何 Agent，也不替代任何系统，它是两者之间的那一层。

PRODUCT 能力矩阵

# Agent 干活要的，都在这里

**可理解**
让 Agent 读懂团队

- 数据持续集成
- Skill 规范
- 角色模板
- MCP 配置分发

**可调用**
按权限取用资源

- 环境变量与凭证
- 数据库
- 对象存储
- LLM 网关
- 应用部署
- Folio 页面发布

**可流动**
人与 Agent 间流转

- 求助
- 权限审批
- 飞书通知与卡片审批
- 页面评论 @

**可积累**
解决一次，复用多次

- 经验沉淀
- 经验检索
- 版本与订阅
- 审计留痕

**三个入口，一套权限** ONE IDENTITY · ONE PERMISSION MODEL

**控制台**
给人：管理、审批、查看

**eat CLI**
给电脑本地的 Agent

**MCP**
给云端的 Agent



# 从业务需求到可用应用

现在 需求 → 成员自己的 Agent → **EAT** → 一个能打开的应用

01

## 规范先到

团队预先定好的规范，直接植入每个成员的 Agent。人不懂规范没关系，他的 Agent 懂。

02

## 资源自取

要数据、要权限，Agent 自己申请。通过 EAT 连接企业数据，员工不用弄懂数据在哪里、数据定义是什么。

03

## 交付到底

建库、上线、发布页面，由 Agent 通过平台完成。最后拿到的，是一个能打开、能分享的地址。

**成员只对需求负责，技术由平台兜底。**

# 让 Agent 去问知道的人

**成员的 Agent**
卡在产品细节，或技术问题

求助 · 带着上下文 →

**知道的人**
产品问产品，技术问研发
由本人，或他的 Agent 回答

答案 →

**回到原来的工作**
接着往下做

01
## 说得准
Agent 的提问精炼，直奔问题，不用听一堆铺垫。

少交 转述税

02
## 背景全
人懒得交代的来龙去脉，Agent 一次带齐。

少交 转述税

03
## 有秘书
对方的 Agent 先接住：分类整理，琐碎的问题直接代答。

少交 等待税

04
## 只答一次
答案沉淀成经验，下一个 Agent 遇到，直接复用。

同一笔税 不交第二次

## EAT 吃掉协作税，让 AI 真正在团队里干活。

IN PRACTICE 真实使用

# 让业务想法，变成可用的应用

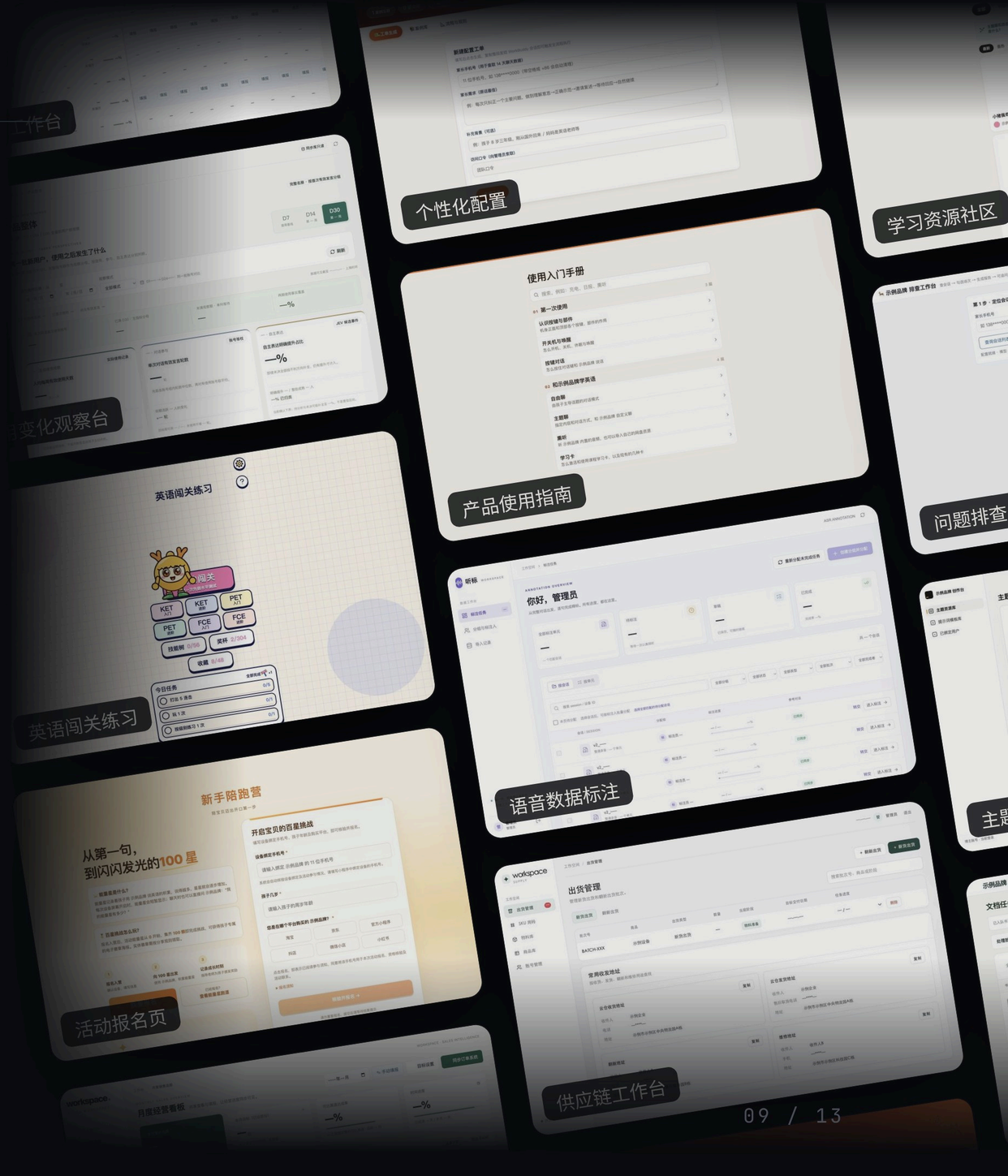
**30**

## 个应用与页面 上线 61 天

- 营销与内容生产
- 销售与客户管理
- 客户服务与排查
- 经营分析与决策
- 财务分析
- 供应链与库存
- 个人中台
- 组织管理与知识协作

**每一个，都出自**
**非技术同事之手。**

EAT · EASY AGENT TEAM

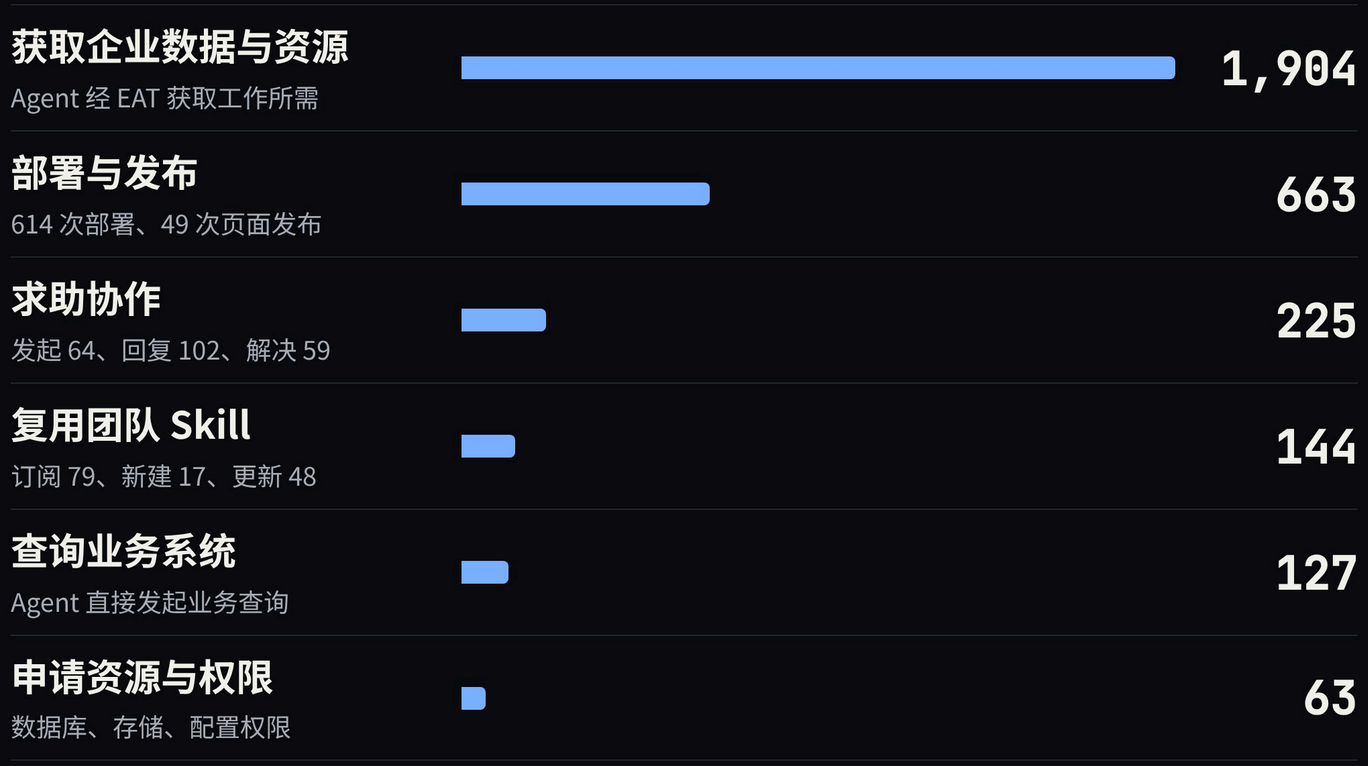


# 过去要找人的事，现在直接办

**3,126**

**次平台能力交互**

上线 61 天



# 从售后发起，到研发修复上线

售后发起，Agent 借助 EAT 索取所需上下文信息，**7 分钟完成排查定位**。

一条客诉的完整交接

- 20:14 售后发起问题排查：「设备报错误码 1」
- 20:21 **7 分钟** Agent 借助 **EAT**，索取所需上下文信息，完成排查定位
- 20:33 Agent 通过 **EAT 向研发上报 BUG**，附完整分析
- 次日 12:03 研发的 Agent 完成修复，经 EAT 回复**「已修复上线」**

次日 12:03 **研发完成修复，求助单已解决**



# 每个人，都有一个 Personal Agent

协作和上下文打通后，大量的**对齐**和**试错**，就可以交给它。

01 · 对齐交给 Agent

## 信息透明

~~做到哪了？~~ ~~为什么这么做？~~ ~~怎么还没做完？~~

**你的 Personal Agent** 去问 · 拿回 **各处的 Agent**

| 进度 | 依据 | 卡点 |
| --- | --- | --- |
| 联调中，周五上线 | 评审定了方案 B | 在等数据权限审批 |

**透明的是事，不是人。**

收掉 转述税 · 等待税

02 · 试错交给 Agent

## 先做再验

**一个想法**
告诉你的 Agent → **你的 Personal Agent**
拉上相关同事的 Agent

相关同事的 Agent · 站在团队的角度

| 产品 | 研发 | 销售 | 财务 |
| --- | --- | --- | --- |
| 用户要不要 | 做起来多重 | 客户买不买 | 账算不算得过 |

**演练一遍**
跑数据，做推演 → **一个结论**
用真实数据说话 → **你拍板**
人只做决策

**视角各自带，立场是团队的。**

收掉 放弃税

# 好想法，不再停在半路。

EAT · AGENT 的企业 IM

# 让需求更直接地成为成果，
# 让协作更顺畅地推进工作。

可理解 · 可调用 · 可流动 · 可积累

OPEN SOURCE

**扫码查看源码**

github.com/othorizon/easy-agent-team

**谢谢** THANK YOU