

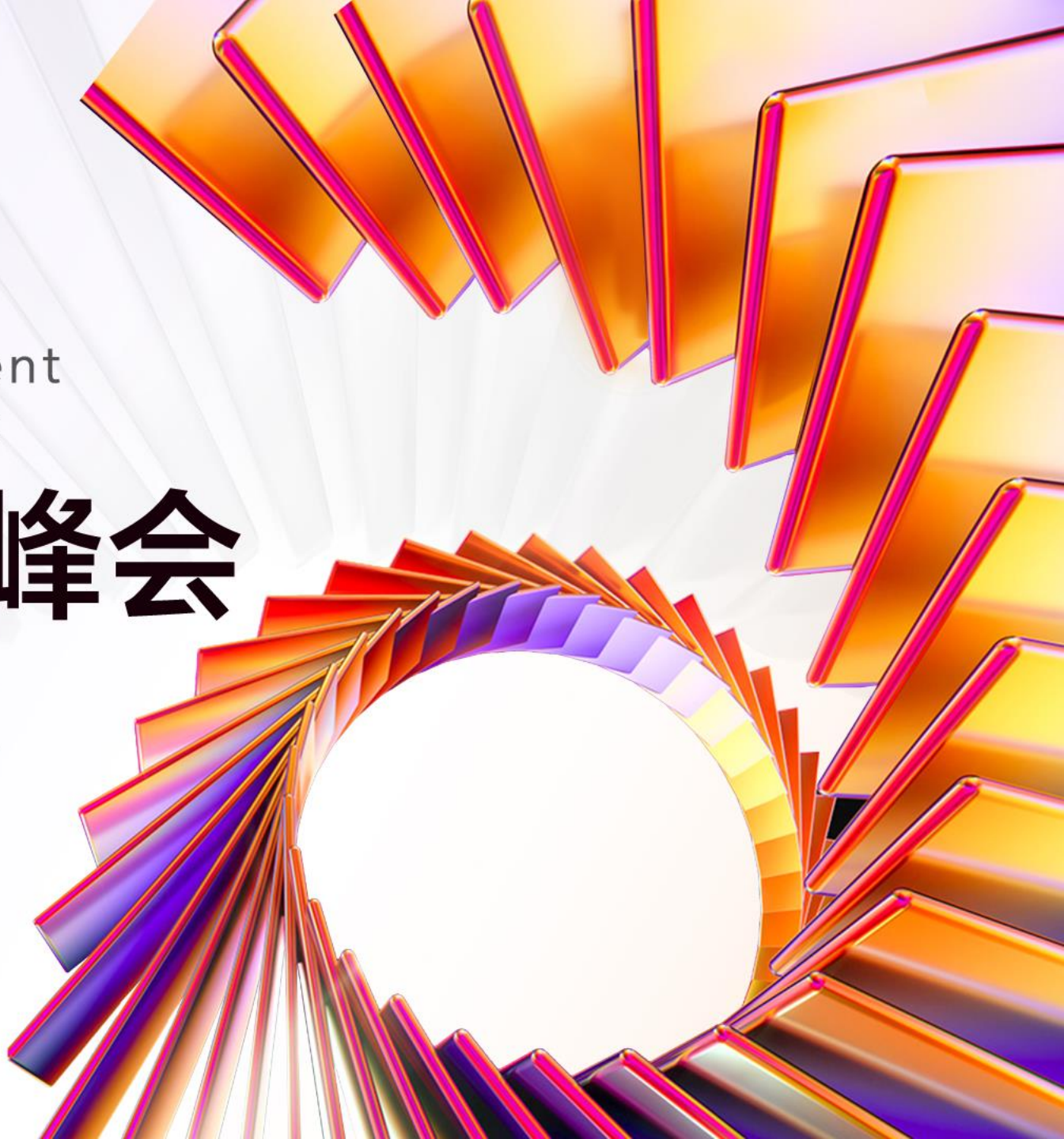


第8届 AI+ Development Digital Summit

AI+研发数字峰会

拥抱AI 重塑研发

11月14-15日 | 深圳





2026 AI+ PRODUCT INNOVATION SUMMIT

01.16-17 · ShangHai

AI+产品创新峰会



Track 1: AI 产品战略与创新设计

从0到1的AI原生产品构建

论坛1: AI时代的用户洞察与需求发现

论坛2: AI原生产品战略与商业模式重构

论坛3: Agentic AI产品创新与交互设计

2-hour Speech: 回归本质



用户洞察的第一性

——2小时思维与方法论工作坊

在数字爆炸、AI迅速发展的时代，
仍然考验“看见”的“同理心”



Track 2: AI 产品开发与工程实践

从1到10的工程化落地实践

论坛1: 面向Agent智能体的产品开发

论坛2: 具身智能与AI硬件产品

论坛3: AI产品出海与本地化开发

Panel 1: 出海前瞻



“出海避坑地图”圆桌对话

——不止于翻译:

AI时代的出海新范式



Track 3: AI 产品运营与智能演化

从10到100的AI产品运营

论坛1: AI赋能产品运营与增长黑客

论坛2: AI产品的数据飞轮与智能演化

论坛3: 行业爆款AI产品案例拆解

Panel 2: 失败复盘



为什么很多AI产品“叫好不叫座”?

——从伪需求到真价值:

AI产品商业化落地的关键挑战

智能重构产品 数据驱动增长
Reinventing Products with Intelligence, Driven by Data

80+

业界大咖

汇聚产品大咖的真知灼见

40+

创新案例

开拓全新AI+产品视角

10+

分论坛

涵盖产品到商业增长的全链路

800+

听众规模

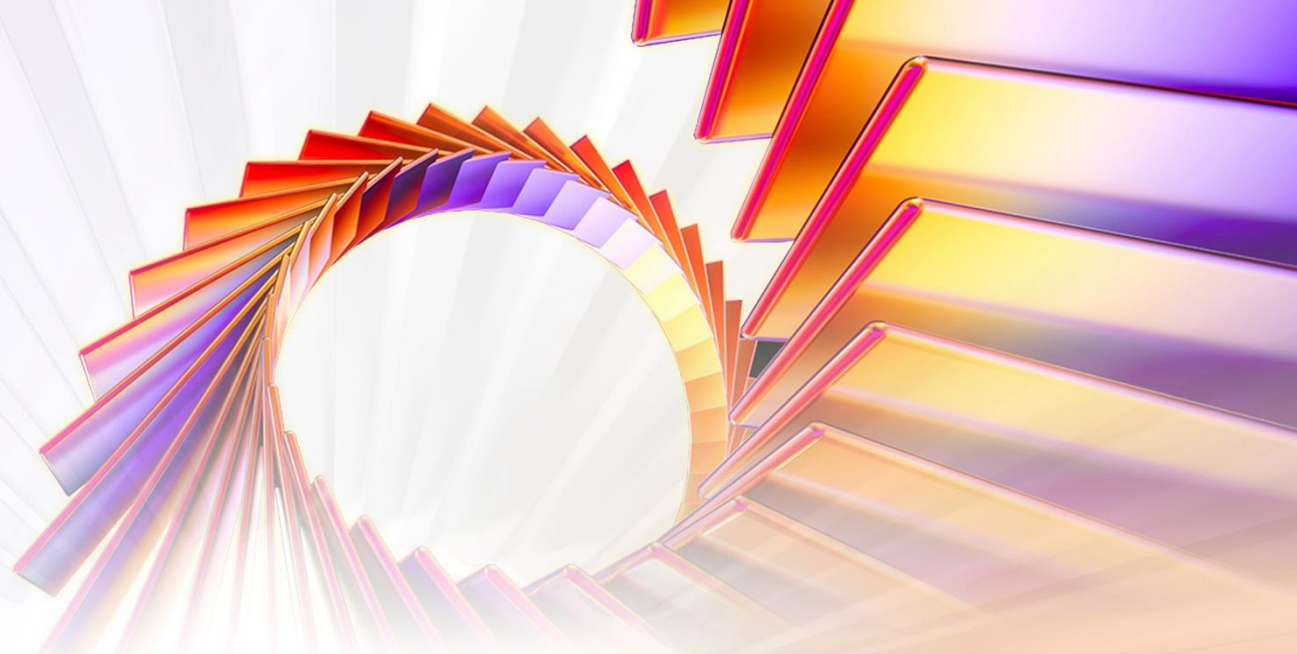
共同探索产品的智能重构



扫码查看会议详情

AI+DD 8th 2025 | 11月14-15日 | 深圳

第8届 AI+ Development
Digital Summit
AI+研发数字峰会
拥抱AI 重塑研发



前端测试新范式

从用例生成与自愈到端到端智能化

王哲 | 百度



王哲

百度资深工程师

2017年毕业加入百度，先后负责百度基础架构、AI 产品等多个核心业务的质量保障工作、并负责百度技术中台群组测试环境治理等工作、目前主要负责 AI 驱动的智能测试解决方案的设计与实施落地。

在测试智能化、自动化测试、测试工具建设以及云原生等方面有着深刻的见解和丰富的实践经验。

目录

CONTENTS

- I. AI时代的前端测试挑战
- II. 用例生成与自愈
- III. 人机协同的端到端测试
- IV. 应用与落地
- V. 总结与未来展望

PART 01

AI时代的前端测试挑战

▶▶ 一个新时代的黎明：AI 正在重塑软件开发



AI 驱动开发

以 Cursor, Comate Zulu 为代表的 AI IDE 已经可以理解整个代码库，并直接基于自然语言的需求实现复杂代码的开发，从编码辅助进化到功能共创。



AI UI/UX 设计

AI 正在设计领域也大放异彩。Galileo AI 则能根据文本在 Figma 中生成专业设计稿，Vercel v0 甚至能将文字或草图直接转换成生产级前端代码。



AI 运维与测试

AI Ops 平台如 Datadog 等正从被动救火转向主动预防。它们能自动诊断根因、预测故障，甚至生成修复代码。同时，AI 也在赋能测试，自动生成单测用例等。

“软件工程的每一个角落，都在被 AI 的浪潮所触及。”



► 关键变量：Computer/Browser Use 能力的范式演进

2024年10月 Claude 3.5 Sonnet 发布 Computer Use 能力

大模型首次内部原理解图像素，可以通过基于截图，自动分析浏览器元素坐标位置并输出操作元素动作和坐标，简化浏览器控制难度。

2025年3月微软发布 PlayWright MCP 服务

微软发布的 PlayWright MCP 将浏览器页面表示为一个简洁且有语义的 DOM 结构树，并为元素增加唯一标识，从而支持纯文本模型可以理解网页状态并实现浏览器控制。

2025年7月 ChatGPT Agent 上线

OpenAI ChatGPT Agent 结合了 Operator 的浏览器操作能力和深度研究模式，在自动操作上网的基础上可以产出详细的分析和报告。

2024年下半年开始，基于大模型操作浏览器的探索开始了

从 2024 开始，模型不仅仅满足于问答等场景，更多的面向 Agent 场景(工具调用等)场景发展，下半年开始，人们已经逐步开始通过截图、DOM 表示等方式尝试让模型理解并操作浏览器了。

2025年1月 OpenAI Operator 发布

OpenAI 正式发布支持操作浏览器来执行任务的 Agent 模式，面向美国 Pro 用户提供服务。

国内模型厂商不甘落后，百花齐放

在模型原理解电脑、浏览器的控制领域上，国内的模型厂商也纷纷发力，包括字节的 UI-TARS 模型、阿里的 Qwen VL 系列模型等纷纷发布相关支持的专用&通用模型。

2025年10月 Gemini 2.5 Pro 推出 Computer Use Model

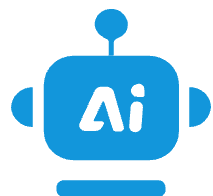
至此为止，Claude, OpenAI, Gemini 3 家世界最领先的模型都针对 Computer Use 场景进行了专项支持。



理想丰满，现实骨感：前端测试的复杂性



▶ 能力与落地的鸿沟：AI Agent 的“最后一公里”



模型能力

强大的AI基础能力
多模态理解、规划、
推理、生成等

+



业务场景

具体的业务测试场景
真实的业务需求

+



专家经验

领域专家的知识 and 经
验将业务场景痛点与
模型能力有机结合

=



AI Agent

真正解决
业务问题的
测试助手



PART 02

用例生成与自愈

▶ 切入点选择：为何是用例生成与自愈？

大模型的领域优势

代码领域是模型证明其生产价值最多的领域之一

- 结构化程度高，模式识别效果好
- 逻辑清晰，便于AI理解和生成
- 可验证性强，错误容易发现
- 大量开源代码提供训练基础

业务依赖低

自动化用例生成相关场景
业务背景牵扯较小

- 聚焦于交互逻辑，不涉及复杂业务
 - 测试意图相对标准化
 - 减少对业务专家的依赖
 - 更容易实现跨项目复用

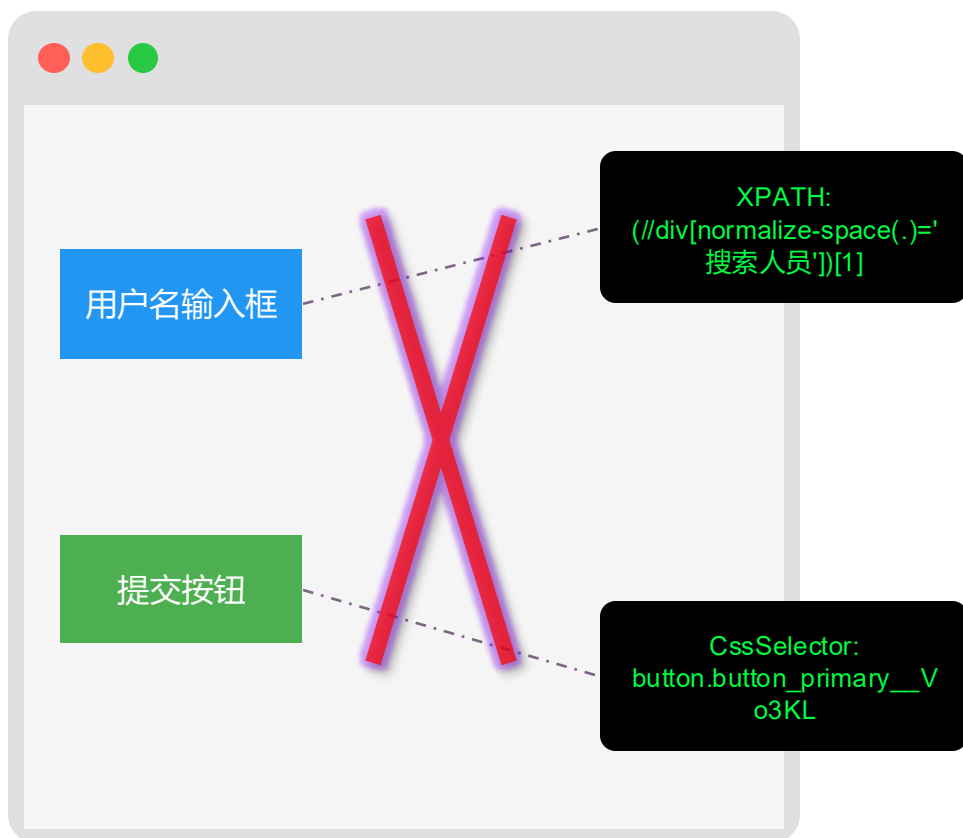
成本痛点突出

前端自动化用例的编写和维护
成本巨大

- 元素定位脆弱，页面变更频繁失效
- 编写复杂，需要深度技术背景
- 维护成本高，投入产出比低
- 测试工程师时间大量消耗在维护上



► 自动化测试的核心症结：复杂又脆弱的元素定位



元素定位表达式是即“硬编码”的坐标

传统自动化代码本质上是给 UI 元素基于当前的特点拍下了一张“硬编码”的地址快照。

UI 页面是“流动的沙丘”

现代前端页面，其 DOM 结构、样式、框架等都在不断迭代，其可用的元素定位表达式往往处于动态变化。

必然的结果

用固定的地址去寻找流动的目标，结果必然是频繁的“查无此人”，导致脚本大规模失效。



► 新范式：从“面向过程”到“面向意图”

面向过程



`driver.find_element(By.XPATH, "/html/body/div[1]/...").click()`

面向意图



点击登录按钮



▶ 眼见为实：视频生成测试

测试过程录制



用户通过电脑屏幕录制，用户操作鼠标进行产品交互

AI 理解操作过程



多模态AI大脑"看"和"理解"用户操作

生成测试步骤描述和代码

```
# 点击"MacBook Pro 14"商品
driver.find_element(By.XPATH,"
//*[@normalize-space()='MacBook Pro 14
英寸']").click()
# 点击购买数量+号
driver.find_element(By.CssSelector,
"svg.lucide-plus").click()
# 点击"加入购物车"按钮
driver.find_element(By.XPATH,
"//button[normalize-space()='加入购物车
']")
# 点击"查看购物车"按钮
driver.find_element(By.XPATH,
"//a[normalize-space()='查看购物车']")
```

输出测试步骤描述以及
自动化测试代码



▶▶ 眼见为实：视频生成测试代码技术架构



▶ 眼见为实：视频生成测试代码 workflow

视频/行为录制

- 通过Web应用内嵌Playwright驱动的真实浏览器环境
- 前后端通过WebSocket实时通信，传递用户操作与浏览器状态
- 高频获取浏览器截图，在前端实时渲染，提供断言、工具调用等交互能力。
- **核心产出：记录每个用户操作 (Action) 及坐标，并快照该瞬间的全路径 XPATH 和完整DOM和截图。**

多模态意图理解

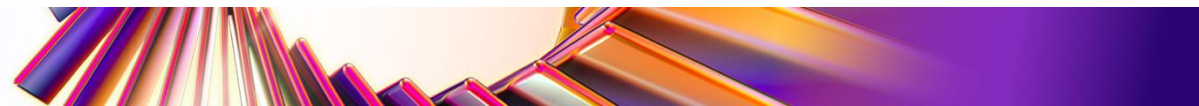
- 将录制的操作截图（组合多张）与截图中操作元素的高亮标识一起分批次提交给多模态大模型 (VLM)。
- 模型理解视觉信息与操作上下文，将机器行为（如点击坐标）转化为人类可读的意图。
- 为保证上下文连贯性，采用批处理（如10个动作）并传入前序动作的自然语言描述。
- **核心产出：每个操作步骤的自然语言描述（如“点击‘登录’按钮”）。**

鲁棒元素定位表达式

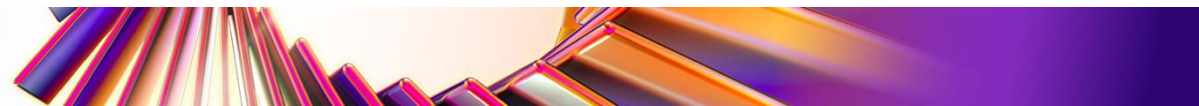
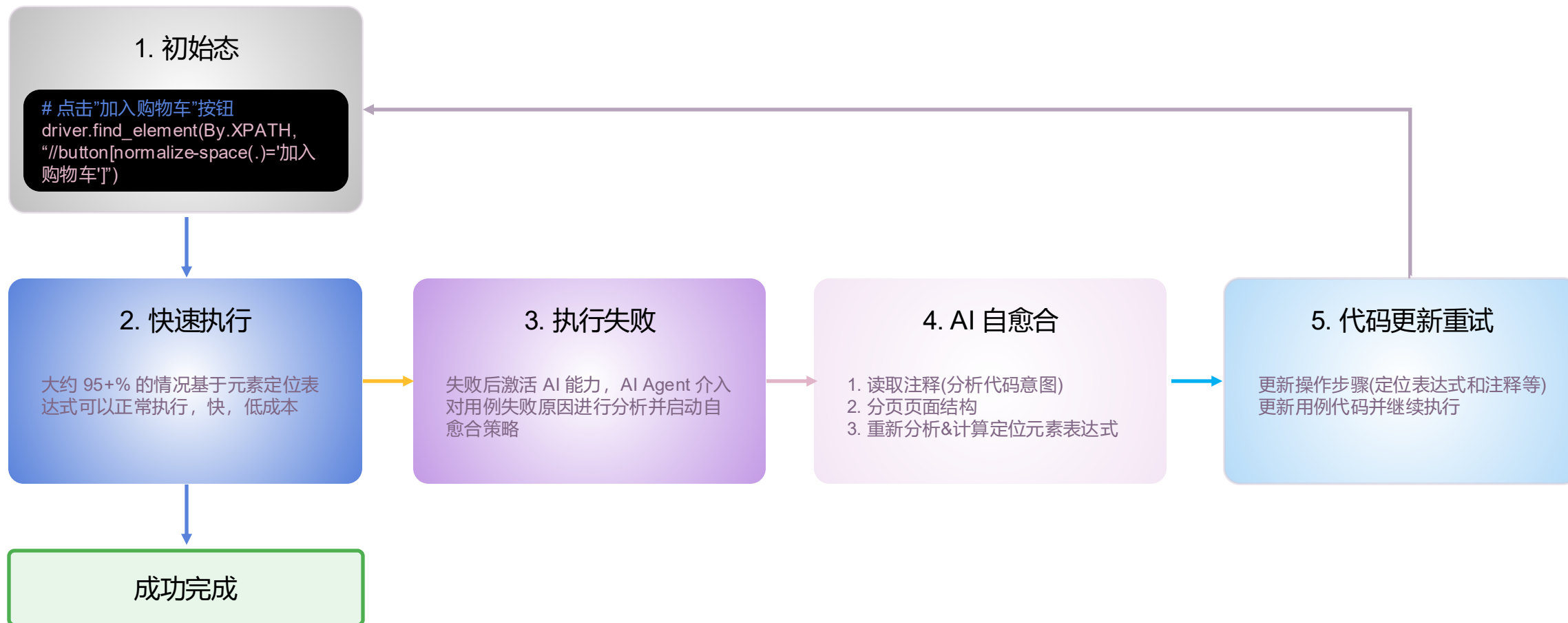
- 结合意图描述、原始XPath和HTML快照，首先通过规则引擎生成一组候选定位表达式。
- 候选集包含文本、ID、CSS类、XPath 等多种健壮性更高的定位策略。
- 将候选集再次提交给模型，由模型根据上下文和最佳实践，选择一个最稳定、最语义化的定位器。
- **核心产出：单一、高质量、可维护的最优元素定位表达式。**

自动化代码生成

- 将前序步骤生成的“动作描述”和“最优定位器”进行组合。
- 通过一个强大的System Prompt，指导大语言模型(LLM)如何将这结构化信息转换为代码。
- Prompt中包含代码风格、基于PlayWright 二次封装的 Lib 库函数定义、错误处理、断言格式等规范。
- **最终产出：一份结构清晰、代码优雅、可直接执行的自动化测试用例代码。**



▶ 代码自愈合：智能的元素定位与修复



用例失败异常捕获

通过封装库，对所有Playwright API调用进行包裹，实现全面、精准的异常捕获。

核心技术：

在 `try...catch` 块中，不仅捕获异常，还能动态分析失败代码的上下文（如变量值、关键注释），为后续分析提供丰富信息。

失败后会立即调用Playwright进行 即时截图，锁定UI现场。



多模态根因分析

将捕获到的“代码意图”（来自失败捕获上下文）和“UI现状”（来自截图）提交给VLM（视觉语言模型）进行智能分诊。

核心技术：

VLM的核心任务是判断用例中的注释的意图与截图是否匹配。其输出是一个结构化的JSON对象，明确指出问题分类和分析依据，为系统决策提供确定性输入。



元素定位自愈

当VLM判断为UI元素变更时，启动自愈流程，自动寻找并生成新的、稳定的元素定位器。

核心技术：

1. Computer Use 模型根据截图和意图，输出目标元素在屏幕上的 (x, y) 坐标和动作。
2. 基于浏览器引擎完成对应的动作并重复获取最新浏览器截图并调用模型判断是否完成对应意图。
3. 基于元素定位表达式生成服务得到新版本的元素定位表达式并完成测试代码的更新。



▶ 代码自愈合：技术方案剖析

智能分析与求助

当VLM判断为产品功能变更时，启动人工求助。

补充说明：

当产品功能迭代后，之前的自动化用例的测试流程往往不适配新的产品功能，此时，需要对测试流程进行相应的调整，此时模型没有足够的输入，需要向业务专家进行求助。

专家接入与指导

系统收集当前的浏览器状态截图、自动化测试代码报错信息、智能根因分析等相关的信息，面向用户提问并等待自然语言的指令。

补充说明：

测试专家需要结合产品的功能变化，重新思考整个测试用例需要调整的测试路径和意图，并通过简单的自然语言的方式进行指令下发。

AI 结合上下文修改代码

利用 Claude Code 的 AI Coding 能力，接入用户自然语言指令和构建的上下文完成测试用例代码的修改。

补充说明：

- Claude Code 可以提供一套成熟、有效的上下文检索、代码改写等能力。
- 对于部分操作依赖元素定位表达式时，会生成自然语言的操作语义代码并参考上一节在执行环节进行元素定位解析。

自动验证与闭环

基于 Claude Code 改写完成的代码，调用用例执行引擎进行代码验证。

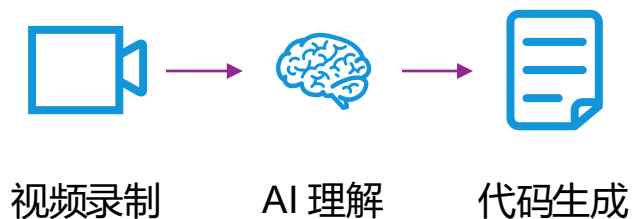
补充说明：

- 验证通过的代码会自动发起代码评审
- 执行过程中，会针对依赖元素定位表达式的自然语言的操作语义代码进行 Computer Use 的模型引导执行并转为对应的元素定位表达式。



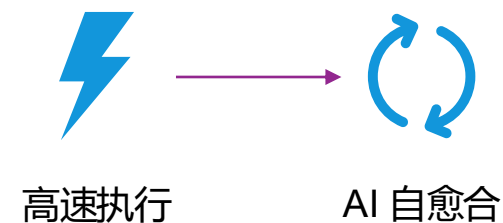
▶▶ 终结回归测试的维护噩梦

用例创建



意图驱动
智能维护

用例执行与维护



核心收益总结

创建成本

趋近于零

操作即用例

维护成本

降低90%

自动愈合

稳定性

大幅提升

意图驱动

执行效率

保持高速

按需 AI



PART 03

人机协同的端到端测试

► 跨越鸿沟：从“回归维护”到“增量创造”

存量回归

维护成本
用例稳定性



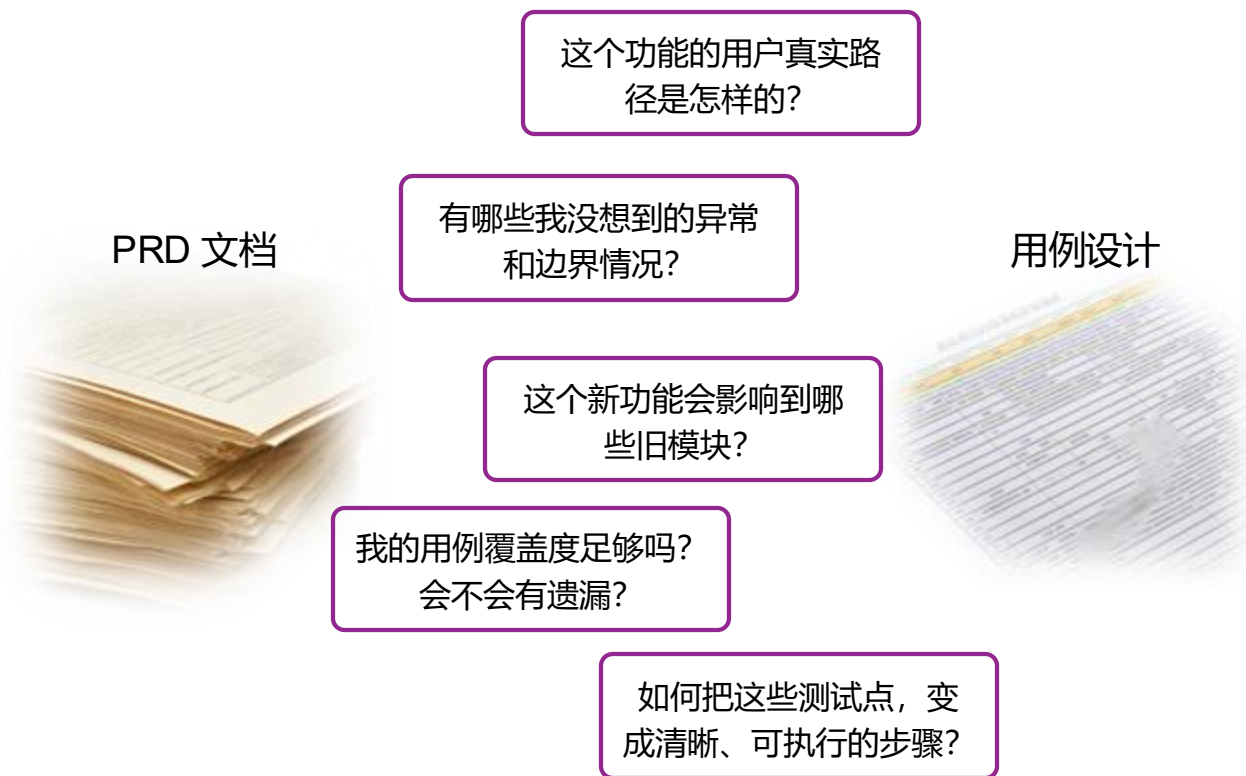
增量测试

需求理解
用例设计
智能执行

在解决如何更聪明的进行回归测试后，我们将目标看向更远的未来，如何更智能的进行新功能测试



困境的起点：测试工程师的“认知迷宫”



🧠 高昂的认知成本

将静态需求文档转化为动态的用户交互场景, 完全依赖人的心智

📈 经验的“天花板”

用例质量严重依赖测试者的个人经验, 难以规模化和标准化

😰 覆盖度的“焦虑”

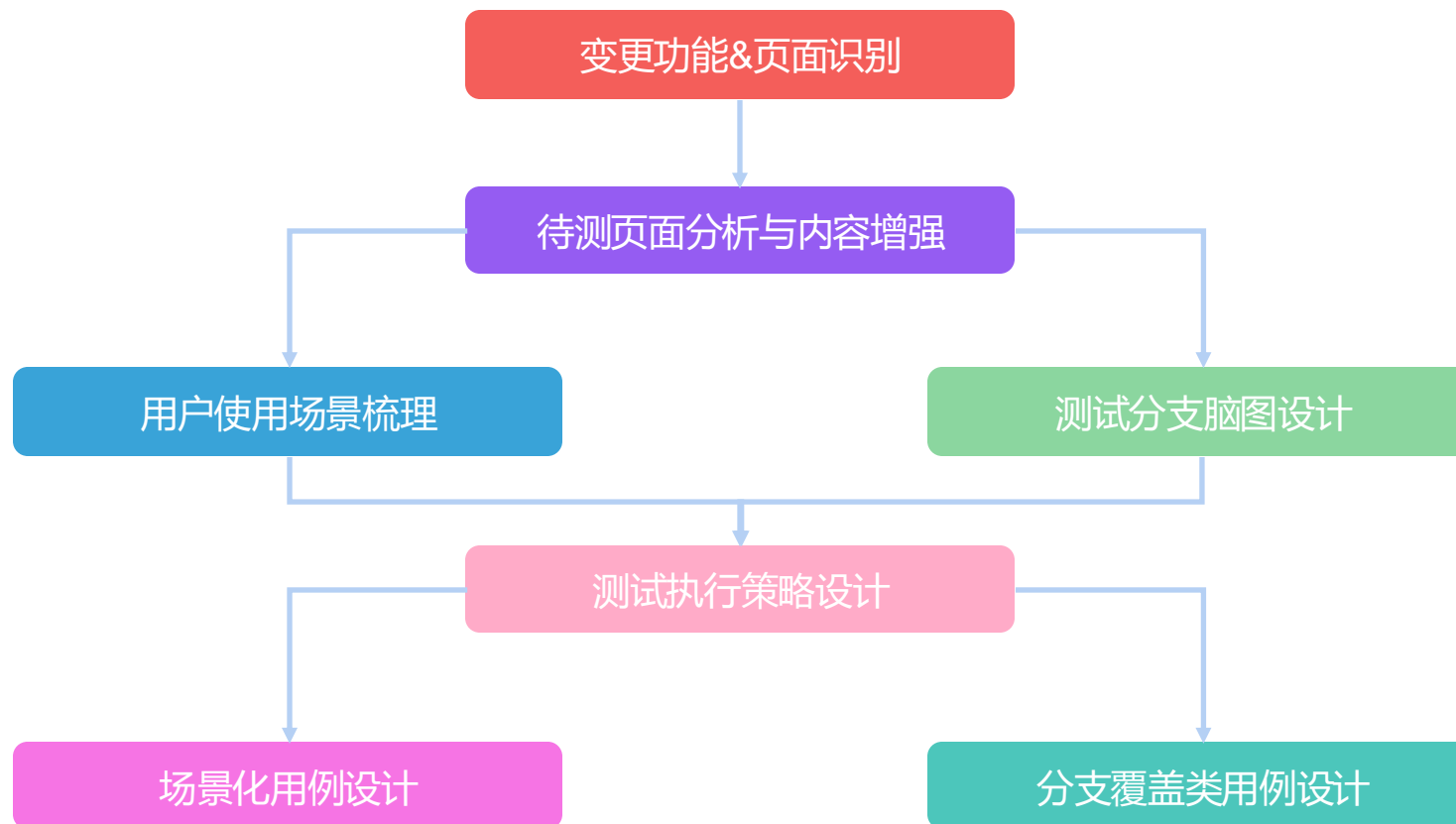
永远不确定是否已经覆盖全面, 存在未知的盲点

🕒 低效的脑力劳动

大量时间消耗在重复性的用例设计思考上

► 破局之路：智能测试设计的“七步成诗”

系统性、工程化的智能测试设计解决方案



► 解构 “七步法”：从理解到规划，再到生成



深度理解

变更功能&页面识别

待测页面分析与内容增强

AI 首先基于需求文档分析出本次设计的**核心功能变动**以及相关的**前端页面**，然后通过访问待测页面找到相关功能的**页面截图**，进一步了解前端功能实际行为。



智能规划

用户使用场景梳理

测试分支脑图设计

测试执行策略设计

这是 AI 的核心“思考”过程。它模拟真实用户，梳理出所有可能的**业务流程**，并构建逻辑清晰、覆盖完备的**测试分支脑图**。
自动应用**等价类**、**边界值**等专业测试策略，规划“测什么”和“怎么测”。



精准生成

场景化用例设计

分支覆盖类用例设计

AI 将复杂的规划，输出为两种可以直接使用的、自然语言描述的用例。一种是**宏观的用户使用流程**，一种是**精细化的条件分支步骤**，使得用尽可能少，尽可能贴近用户使用习惯的方式实现更多的测试覆盖。





传统用例设计

用例冗余重复

每个用例测试点相对零散，在多个用例之间为了完成对应的测试点，有大量的冗余、重复操作步骤

覆盖凭感觉

容易遗漏测试点交叉关联场景和各类复杂的边界条件情况

步骤描述模糊

步骤描述简单，对用例执行人有一定要求，需要理解一定的业务背景，否则无法有效执行用例



杂乱、数量众多的测试用例，用例无法独立执行

VS



AI七步设计法

更少的执行步骤

AI 自动优化执行策略，尽可能在每条用例中覆盖尽可能多的测试点，最终通过更少的执行步骤实现完备覆盖

更全的测试覆盖

通过测试分支的脑图设计展开尽可能的保证了测试覆盖完备度，分支条件、边界值等都考虑到

更明确的操作步骤描述

生成的自然语言步骤描述是详尽的，准确描述浏览器操作验证流程，专为 AI 执行而设计



覆盖所有测试分支的简洁用例组合，执行路径优

指令式协同，专家的“点石成金”

批量扩展 · 测试维度的瞬间丰富

一个基础的登录流程，默认流程中仅通过几条用例分别验证了手机号登录、邮箱登录等几种不同登录方式的正常登录流程。

为所有登录场景，统一补充密码错误、账号不存在和网络超时这三种异常场景测试用例

测试场景瞬间丰富，测试覆盖变的足够完整，同时考虑：

1. 多种登录模式
2. 每种登录模式下正常登录情况
3. (新增)每种登录模式下的多种登录异常情况

批量修改 · 领域知识的精准注入

在一个绑定银行卡的银行卡绑定相关的测试中，由于 AI 不了解我们的系统测试数据，导致测试步骤比较模糊，没有明确的银行卡类型和对应银行卡号，无法具备 AI 执行能力。

所有银行卡的类型修改为'招商银行信用卡'，并设置银行卡号为 4312-xxxx-xxxx。

测试用例步骤瞬间完善，用例执行操作十分明确，AI Agent 可以自动执行一系列相关用例，无需人工点点点。

人机协同的完美融合

AI 负责规模化执行

将人的高阶意图，转化为精细、无误的批量操作

人负责战略性决策

聚焦于业务逻辑、风险洞察和领域知识，实现智慧的“一键部署”

最终价值

将数小时甚至数天的人工维护工作，压缩至**几分钟**的自然语言交互



执行的奥秘: AI 任何像人一样操作浏览器



意图理解与规划层

基于实际状态拆解测试步骤，并规划下一步操作

输入

自然语言操作步骤描述 & 当前页面信息
点击登录按钮进入登录页面，用户名输入xxx，密码输入xxx，点击登录按钮

LLM 意图解析与规划

分析上一步行为的效果并规划下一步操作
上一步点击登录按钮后已经成功进入了登录页面，下一步应该在用户名输入框输入 xxx



多模态感知与元素定位层

AI 的“眼睛”，综合分析当前页面截图、页面 HTML 结构树等信息找出对应需要操作的元素信息

输入

规划的下一步操作和当前浏览器信息，包括浏览器截图和 HTML 结构树

元素识别和动作参数提取

基于下一步参数的描述，基于浏览器截图找出对应元素在页面中的坐标，同时从 HTML 结构树找出对应元素的 ref-id，双重保证来提升元素识别的准确率，并构造对应浏览器操作可能需要的参数。



浏览器执行层

基于实际状态拆解测试步骤，并规划下一步操作

输入

浏览器操作动作类型、操作对象和参数
例如: 点击 ref id 为 e32 的元素，坐标为 (197,300)。

浏览器引擎执行

Playwright 框架
基于元素特征找到对应的元素，然后通过底层浏览器引擎完成对应操作，并获取操作后的浏览器状态信息

输出

浏览器操作后的浏览器状态，包括截图、HTML 结构树等信息

智能反馈循环

动态调整策略

异常智能处理

实时操作反馈

操作后智能等待

单次多步推理

异常实时检测

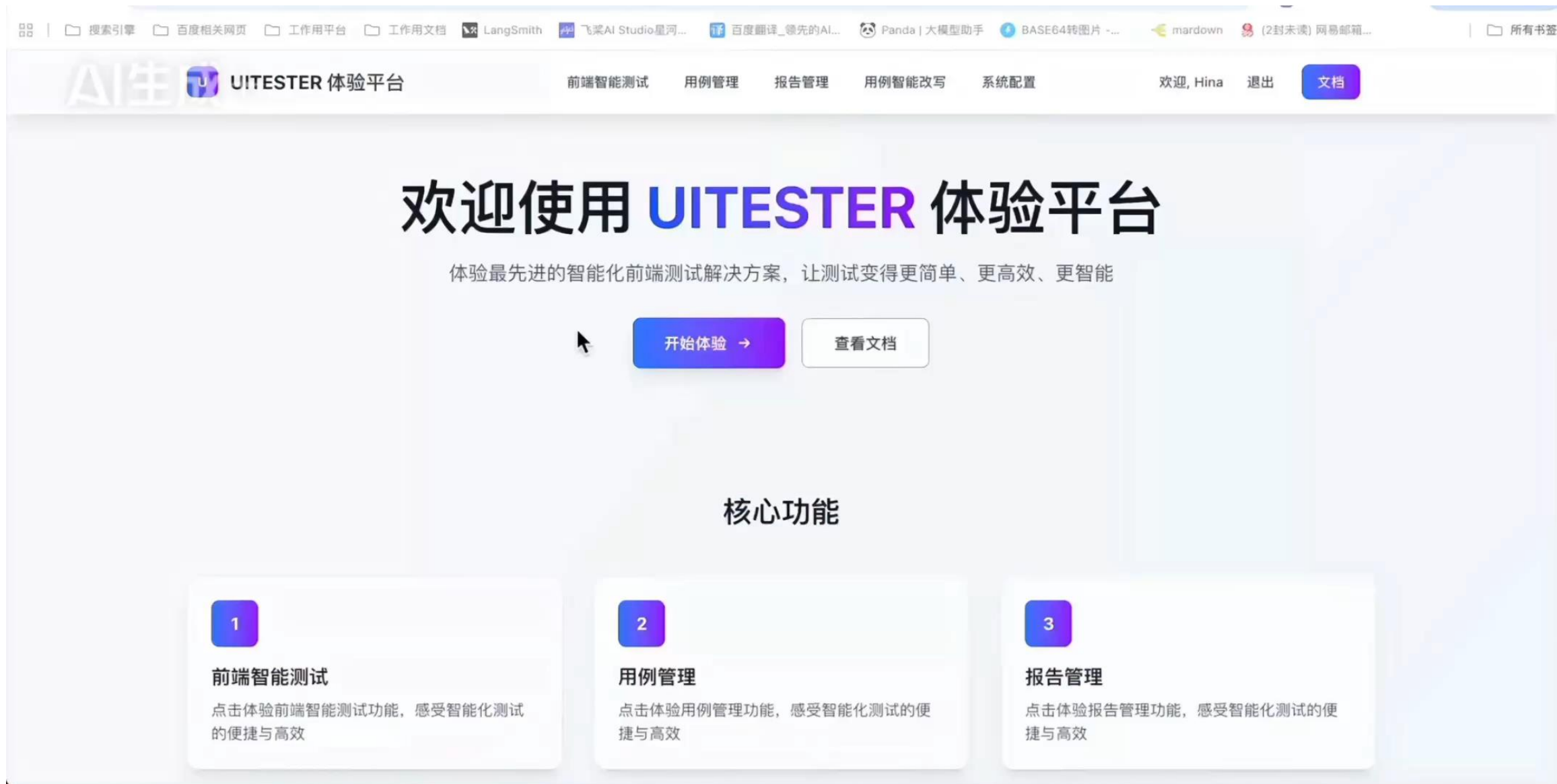


▶ 终极闭环: 一次测试, 永久沉淀



PART 04

应用与落地



实践检验：效率、质量与模式的全面飞跃

效率革命



将以天为单位的测试设计、执行与脚本编写工作，压缩至数量级变化的的人机协同和AI执行，实现工作模式的根本性变革。

数量级提升

即时响应

质量深化

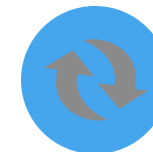


AI的系统性分析能力，显著提升了测试覆盖率，尤其是对复杂逻辑和边界场景的发现能力，有效辅助团队完善测试策略，弥补传统测试方法的盲点。

有效完善

消除盲点

模式颠覆



实现了"一次测试，永久沉淀"的理想模式。新功能验证的过程，就是回归资产自动生成的过程，彻底消除了"测试债"。

低成本沉淀

价值飞轮

核心价值

不仅仅是效率的提升，更是工作模式的根本性变革，让测试从成本中心转变为价值创造引擎，推动软件质量保障进入智能化新时代



▶ 开放体验：即刻开启您的 AI 测试之旅



PixelUI体验平台

<https://uitester.baidu.com/experience>

注意事项

1. 体验环境为展示版本，功能有所限制
2. 请勿上传业务真实资料或敏感机密文件等





“播种-浇灌-共建” 三步法



策略

精心选择1-2个有意愿、有痛点的“种子团队”。投入核心资源重点支持，尽可能保证100%成功。

目标

打造一个有显著变化的成功范例，形成口碑效应。



策略

将标杆项目中的能力沉淀为标准化的平台能力和详尽的文档、教程。组织分享和培训，降低接入成本。

目标

让更多的团队可以“自助式”接入，实现从点到面的扩张。



策略

鼓励有能力的团队参与共建、分享最佳实践。建立内部开源机制，从中央厨房演变为生态共建。

目标

激发整个组织的活力，让平台持续演进满足不同业务的场景需求，最终形成茂密的森林。



PART 05

总结与未来展望

模型成本与延迟

规模化应用的经济账。大规模部署AI测试工具面临着高昂的计算成本和响应延迟，需要在效果与成本之间找到平衡点。

私有化与数据安全

在核心业务中的信任问题。企业核心业务数据的安全性要求，使得AI工具的私有化部署和数据隐私保护成为关键障碍。

挑战

模型能力局限

目前我们的实践来看用例执行的有效成功率在90%，离生产环境的可靠性要求仍有差距。

业务背景理解

需要较多的人工协同完善。AI对复杂业务逻辑和个性化领域知识的理解缺失，需要大量人工介入来补充上下文信息。



成本&效率优化策略

通过模型压缩和策略优化等模式降低模型推理成本，例如场景专用的小模型等；通过用例执行策略的优化实现用例有效执行前提下的最大并发度提升执行效率等。

私有化模型的发展

越来越多的开源模型（UI-TARS、Qwen VL 等）支持浏览器控制等能力，随着模型的迭代的发展，开源模型将会具备闭源模型的替代能力，从而实现安全的私有化部署。

应对

准确度提升计划

通过多策略协同，进一步提升准确率，包括：模型能力本身的不断迭代提升；多模型的集成决策和结果验证机制等。

知识增强能力

上下文业务知识的增强能力的重要性不断提升，随着 AI 应用整体的发展，业务知识图谱将逐步成型，同时，Claude Code 等基于代码理解的能力逐步提升，都将会提供更多更有效的知识背景。



冲击

哪些“重复性”工作正在被代替？

UI脚本的自动生成: 基于测试场景描述或执行记录即可自动生成对应的自动化代码，大幅减少手工编写工作。

无休止的手工测试: 未来 AI 可以7*24小时不间断执行，效率和覆盖范围远超人工模式。

脆弱的脚本维护: 前端用例自愈能力可自动适配前端变化，显著降低脚本维护成本。

如果你的核心价值主要体现在这些“体力活”上，那么变革的压力已经到来。

机遇

我们的新价值在哪里？

成为AI的“指挥官”: 从执行者到指挥官，你负责制定测试策略，评审测试用例，Review 测试结果，AI 负责高效执行。

成为质量数据“分析师”: 从找Bug到防风险，从测试数据分析质量短板，优化测试模式将工作焦点前移。

成为复杂场景的“探索家”: 结合对业务的深入理解，设计 AI 想不到的、高价值的测试场景。

将重复性工作交给AI，把人的智慧聚焦在策略、分析和创造这三件更有价值的事情上。



▶ 能力重塑：新一代测试专家的四大核心能力

技术驱动力

善于充分利用技术工具，尤其是AI技术工具，充分提升测试效率、扩展测试深度，从中高效发现潜在风险的能力。

AI赋能

效率提升

战略与探索性思维

具备全局的产品视野，进行风险驱动的策略设计，运用探索性测试思维，具备探索系统边界和未知风险的能力。

全局视野

风险驱动

流程优化与全局影响力

将质量工作结合 AI 能力实现测试前置并融合整个研发流程，共同构建高质量产品的能力。

流程优化

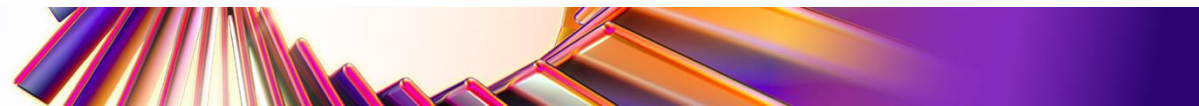
影响力

根基能力: QA 专业素养

不可动摇的立身之本

包括: 业务理解能力、测试理论基础、用户同理心

这是指挥AI、训练AI、解读结果、判断质量好坏的标尺和源泉



▶ 转型路径：如何拥抱AI，重塑专家能力

第一阶段：思维转变

从排斥到共生：将 AI 视为高效的合作伙伴

从“执行者”到“指挥官”

你的价值不再是亲手执行设计每个用例、执行每一个步骤和编写每一行代码，而是定义目标、规划策略、审查结果，让AI为你冲锋陷阵。

从“发现者”到“定义者”

你的核心工作不再是寻找单个缺陷，而是定义“什么是高质量”，并教会AI如何去设计和验证。

第二阶段：能力融合

人机协同：用智慧驾驭并增强 AI

AI辅助探索性思维

将已知的、重复的路径交给AI覆盖，从而解放你的精力，去探索未知的、需要人类直觉的复杂场景。

从“使用者”到“共建者”

结合业务独特性，通过插件、知识增强、模型微调等方式扩展和增强AI，使其成为解决特定问题的“定制武器”。

第三阶段：价值创造

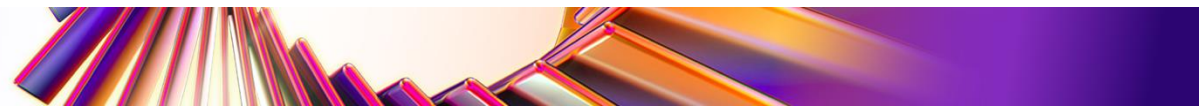
超越执行：成为不可或缺的质量策略家

驱动预测性质量保证

利用人机结合的洞察力，从“事后发现缺陷”转向“事前预测风险”，将质量工作提升到新的战略高度。

成为团队的“质量顾问”

凭借数据洞察和战略视野，你不再是流程的终点，而是贯穿始终的质量赋能者和团队信赖的专家顾问。



最终图景：一个为“持续创新”而生的研发范式

之前：质量债务的拖累



未来：创新飞轮驱动



测试工程师 → 质量策略师

专注于定义复杂业务场景、设计测试策略，
将个人经验转化为可规模化的智能

开发者获得“即时质量反馈”

AI成为永不疲倦的“质量结对伙伴”，在编码
阶段提供实时反馈，让大量Bug在诞生前就
被消灭

企业实现“高置信度交付”

发布不再是赌博，以更高频率、更强信心交
付，将精力聚焦于业务创新本身



科技生态圈峰会 + 深度研习

——1000+ 技术团队的选择



NiDD 峰会

NiDD 峰会 上海站

AI+研发数字峰会

时间: 2026.05.22-23

NiDD 峰会 北京站

AI+研发数字峰会

时间: 2026.08.21-22

NiDD 峰会 深圳站

AI+研发数字峰会

时间: 2026.11.20-21



AiDD峰会详情

NiDD × AI+产品峰会

NiDD × 上海站

AI+产品创新峰会

时间: 2026.01.16-17

NiDD × 北京站

AI+产品创新峰会

时间: 2026.08.23-24



产品峰会详情



2026 AI+ PRODUCT INNOVATION SUMMIT

01.16-17 · ShangHai

AI+产品创新峰会



Track 1: AI 产品战略与创新设计

从0到1的AI原生产品构建

论坛1: AI时代的用户洞察与需求发现

论坛2: AI原生产品战略与商业模式重构

论坛3: Agentic AI产品创新与交互设计

2-hour Speech: 回归本质



用户洞察的第一性

——2小时思维与方法论工作坊

在数字爆炸、AI迅速发展的时代，
仍然考验“看见”的“同理心”



Track 2: AI 产品开发与工程实践

从1到10的工程化落地实践

论坛1: 面向Agent智能体的产品开发

论坛2: 具身智能与AI硬件产品

论坛3: AI产品出海与本地化开发

Panel 1: 出海前瞻



“出海避坑地图”圆桌对话

——不止于翻译:

AI时代的出海新范式



Track 3: AI 产品运营与智能演化

从10到100的AI产品运营

论坛1: AI赋能产品运营与增长黑客

论坛2: AI产品的数据飞轮与智能演化

论坛3: 行业爆款AI产品案例拆解

Panel 2: 失败复盘



为什么很多AI产品“叫好不叫座”?

——从伪需求到真价值:

AI产品商业化落地的关键挑战

智能重构产品 数据驱动增长
Reinventing Products with Intelligence, Driven by Data

80+

业界大咖

汇聚产品大咖的真知灼见

40+

创新案例

开拓全新AI+产品视角

10+

分论坛

涵盖产品到商业增长的全链路

800+

听众规模

共同探索产品的智能重构



扫码查看会议详情



第8届 AI+ 研发数字峰会
AI+ Development Digital Summit

感谢聆听!

扫码领取会议PPT资料

