



2025 AI+ Development  
Digital Summit

# AI+ 研发数字峰会

拥抱AI 重塑研发

05/23-24 | 上海站



# 2025 AI+研发数字峰会

拥抱AI 重塑研发 AI+ Development Digital Summit

下一站预告

08/08-09 | 北京站

11/14-15 | 深圳站



查看会议详情

## 北京站论坛设置

大模型和 AI 应用评测

智能存储与检索技术

下一代知识工程

AI+ 金融业务创新

智能需求工程

智能体与研发效率工具

AI 产品运营与出海策略

大模型安全与对齐

大模型应用开发框架与实践

智能体经济 (Agentic Economy)

智能测试工具的开发与应用

具身智能与机器人

代码生成及其改进

AI+ 新能源汽车

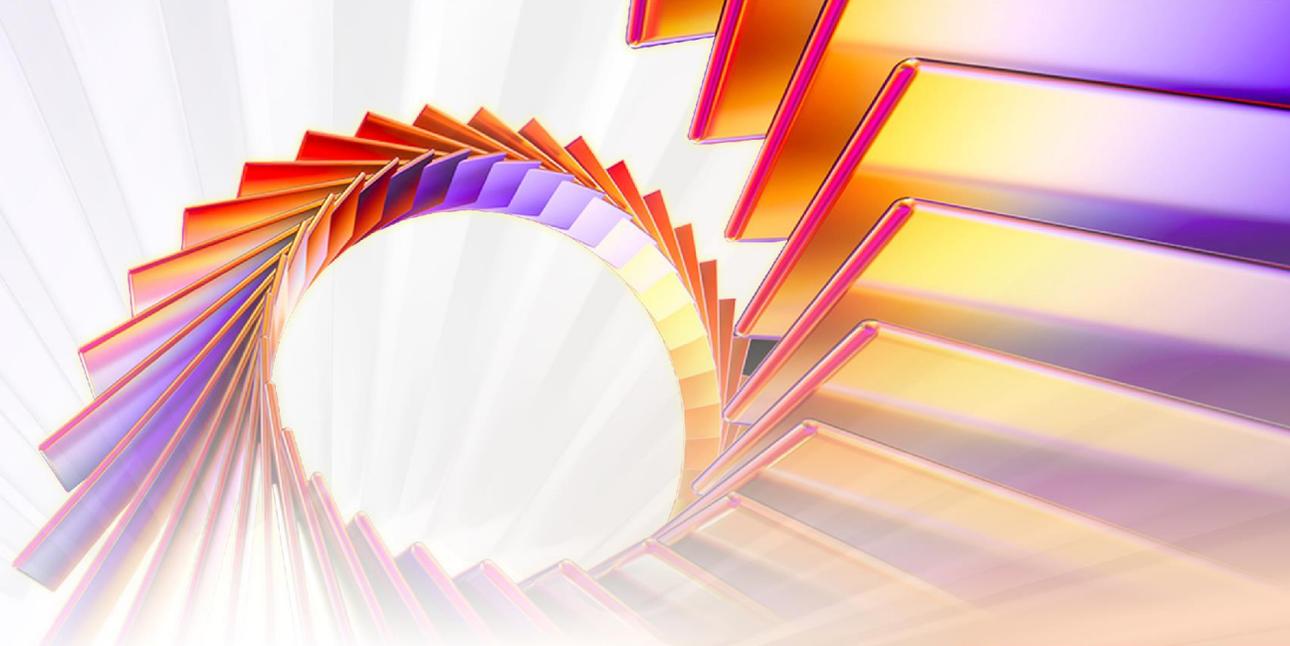
AI 前沿技术探索与实践

  | 05/23-24 | 上海站

**2025** AI+ Development  
Digital Summit

**AI+研发数字峰会**

拥抱AI 重塑研发



# 从代码框架到智能化测试， UI自动化的建设之路

张永杰 | 科大讯飞



## 张永杰

科大讯飞 高级测试工程师

---

8年移动端开发测试经验，对各类客户端的UI测试、专项测试、性能测试等，有较为丰富的探索及实践，负责集团基于移动客户端的本地和平台层工具建设，推动测试技术在集团内的推广落地，助力集团测试人员持续提效。

# 目录

## CONTENTS

- I. 背景：UI自动化为什么难落地
- II. 我们是如何推进UI自动化落地的
- III. 最终解决方案：AITest平台
- IV. 内部应用成效
- V. 未来展望：走向完全的自动化

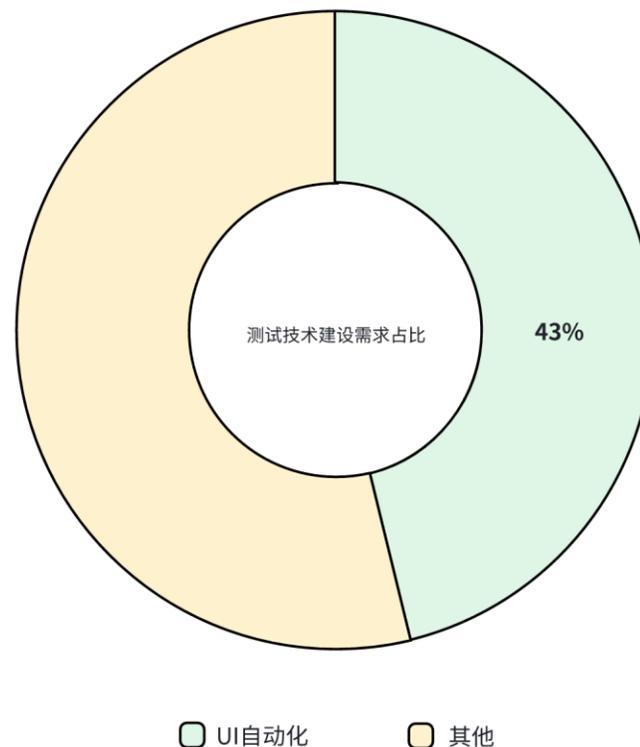
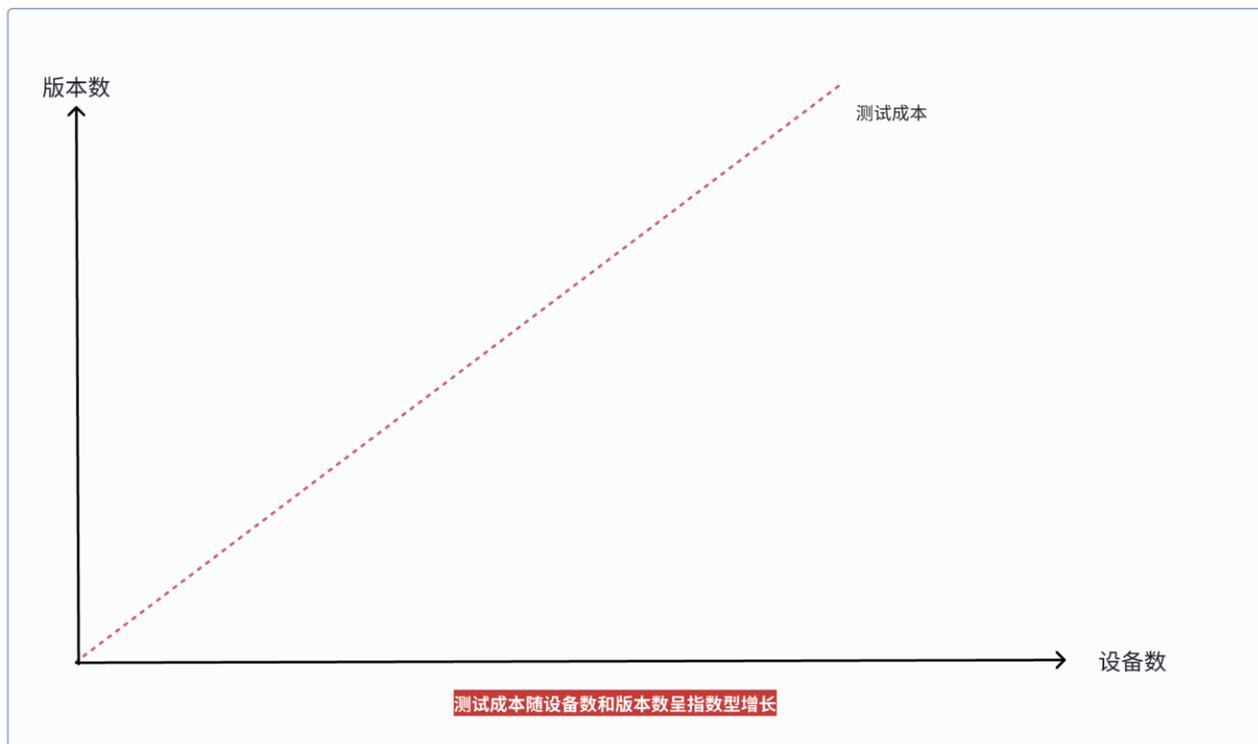
# PART 01

## 背景：UI自动化为什么难落地

# ▶ 从每年的需求调研来看，UI自动化呼声一直不减

UI自动化代表了技术解放人力的极致追求，每当提到“降本增效”，UI自动化总是会不断的被提起；这点从每年对产线的需求调研也能看出来，在各类型的测试技术中，UI自动化工具的使用，占比达到了43%。

特别是，随着智能硬件类产品的增多，人力投入呈指数型上涨的测试场景越来越多，例如，2台机器，5个升级版本，每个版本更新3个app，基础回归就要考虑测 $2*5*3$ 次。



# ▶ UI自动化的历史落地情况

出于业务本身测试成本的压力，和对UI自动化实现后能够完全代替人力测试的“美好愿望”，再加上现有的各类方案层出不穷，门槛不断的降低，所以历史上不断地有产线自己尝试建设各类UI自动化落地方案，

但现实往往是，各个产线各自折腾，“自研”的自动化测试框架人手一个，但受限于各产线技术人员的技术水平和持续投入人力，最终能够长久走下去的**十不存一**。

## 多而不精的自动化方案

UI自动化看起来很“简单”，市面上各种开源的自动化框架搭配起海量的入门级教程，导致前期各业务产线几乎“人手一个”自动化测试“自研”框架

## 不断的冗余投入建设

由于各产线间的信息并不总是共通的，面对相似的需求，可能仍然会不断的有冗余的投入建设

## 难以为继的维护需求

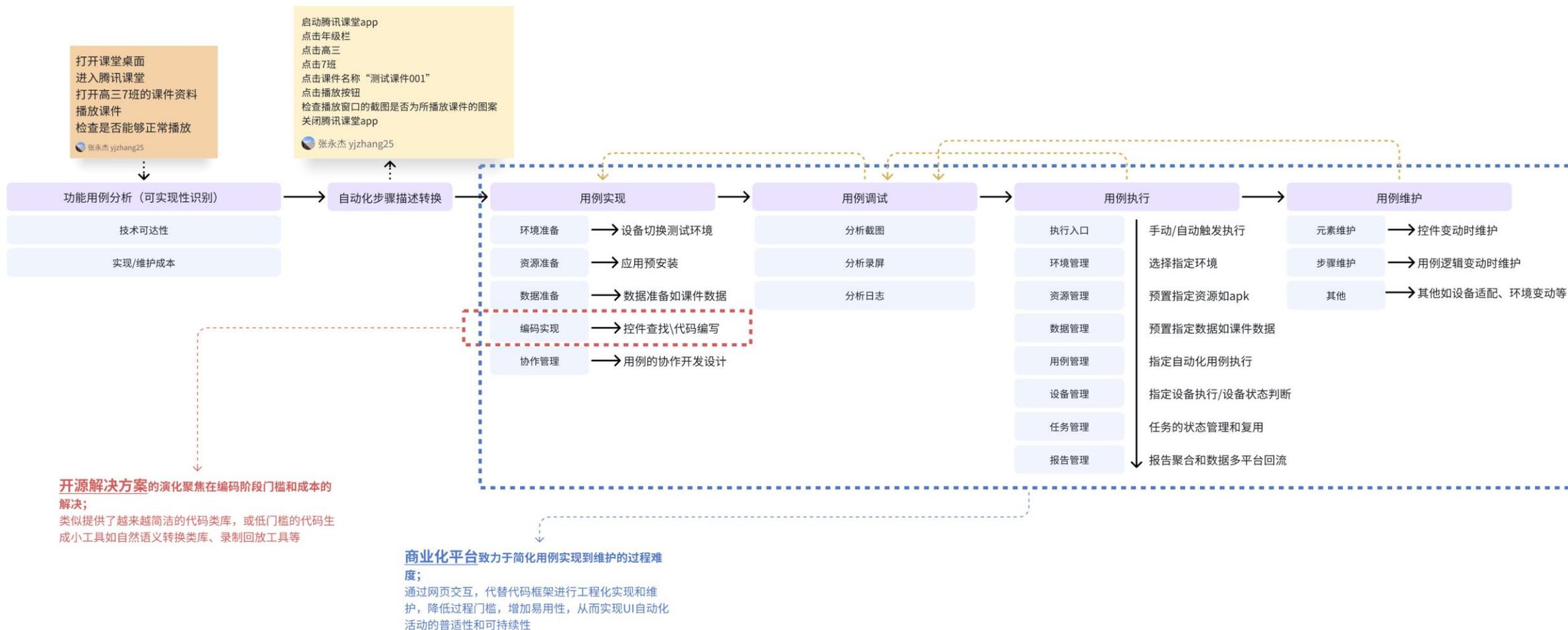
由于产线的UI自动化用例大部分采用代码框架的形式进行建设，而各人的代码水平良莠不齐，导致在人员变动的时候，自动化测试用例往往面临难以为继的处境



# 从可选方案来看UI自动化为什么难落地

UI自动化的落地本质上是一项系统性工程而非单点的技术能力解决，开源方案往往只聚焦于编码实现阶段的门槛降低，缺乏对流程中工程要素的低门槛、易用性的整合设计；所以可选开源方案虽多，但落地依旧很难；

外部解决方案中，只有商业化平台具备从用例实现到维护的全流程低门槛解决方案，但也存在局限性



## PART 02

# 我们是如何推进UI自动化落地的？

# ▶▶ UI自动化推进的整体演进策略

## 统一代码框架

解决不同产线闭门造成带来的人力冗余

## 平台可视化管理

解决自动化相关数据资产的可视化管理、流程类的场景互联互通

## 智能化

降低门槛，提高UI自动化的编写效率，降低用例的后续维护成本

# step1: 统一标准-支持多种测试场景的应用层自动化开发框架

定位：自动化场景能力覆盖的应用级框架



# ▶ step2: 平台建设-更高一层的需求用测试平台来解决



## 可视化结果管理

- 自动化的成功率曲线是什么样的?
- 自动化发现的问题数趋势如何?
- 自动化的执行次数如何?

01

## 云端硬件池共享

- 本地机器不够怎么办?

02

## 标准化能力提供和统一化演进

- 自动化框架的合理设计和演进技术门槛很高怎么办?
- 各业务产线不同业务怎么保证产出水平相近?

03

## 数字资产管理

- 自动化用例怎么多人协作完成?
- 每月新增用例数是多少?

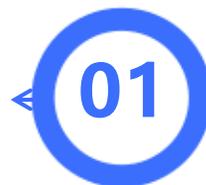
04



# ▶▶ step3: 智能化在线编写支持-降低门槛，提高效率

## AI技术融合

通过大模型驱动的自然语言转低代码，实现智能化测试的突破，降低技术门槛，提高测试效率。



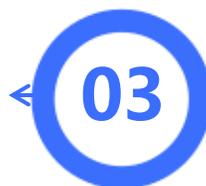
## 动态交互优化

采用左右分栏“翻译式”脚本编写方式，实现自然语言与低代码之间的双向映射，提升用例可维护性。



## 兼容性设计

结合Git与平台双模式协作，无缝对接CI/CD流程，确保自动化测试的稳定性和可靠性。



## PART 03

# 最终解决方案：AITest

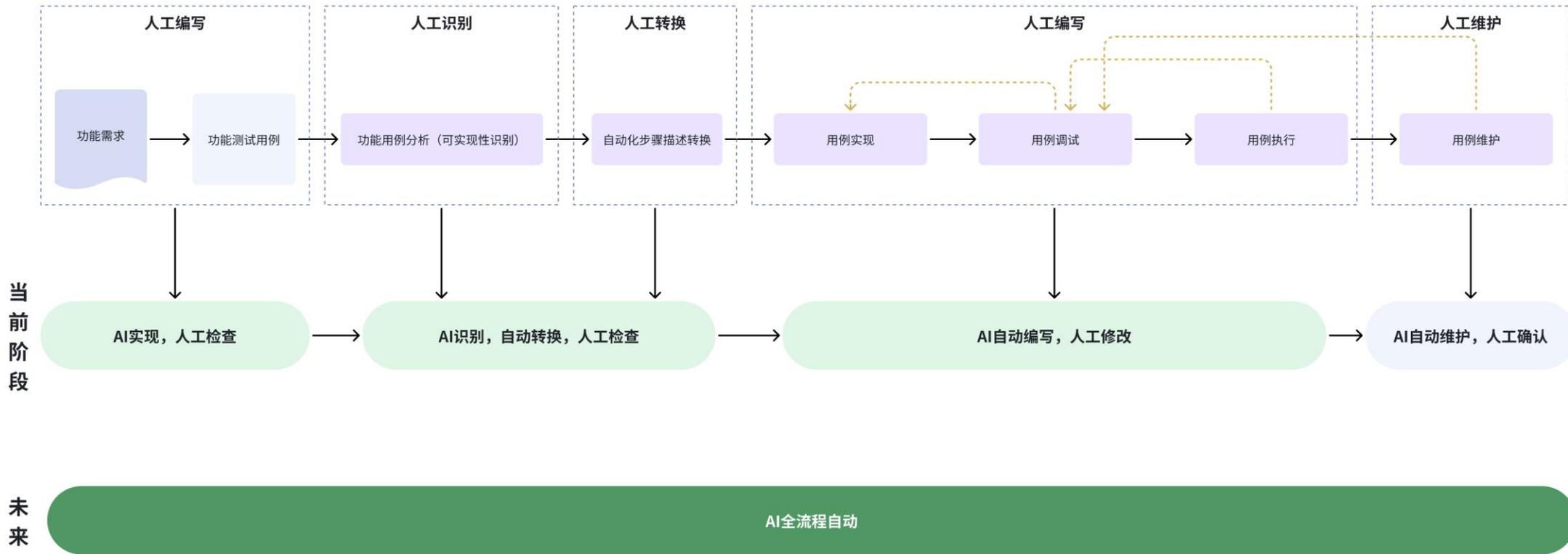
# ▶ 我们的产品愿景- “三零” 测试：零人工介入（全自动）、零场景遗漏（智能遍历）、零延时响应（实时分析）



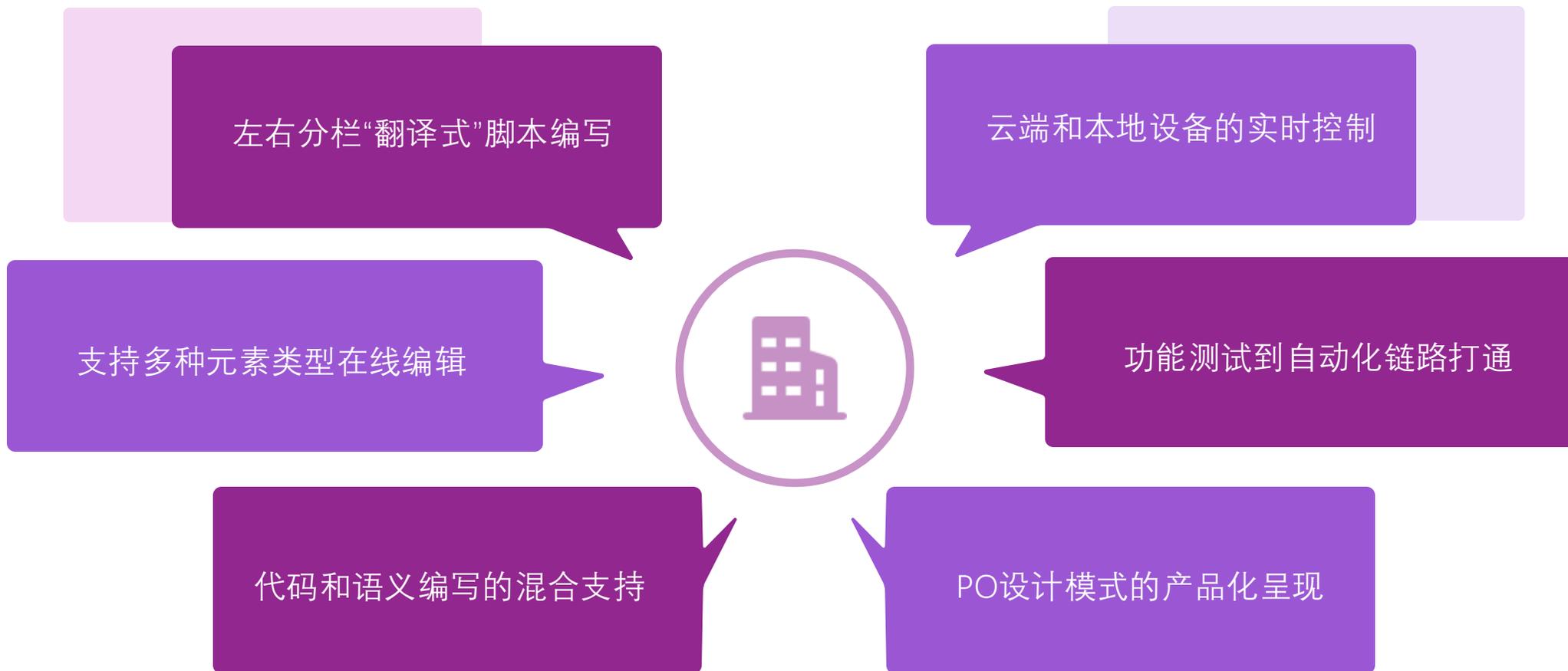
市面的解决方案，大多聚焦在自动化用例实现和维护阶段的低门槛化上；但是对更前一步的链路，即，从功能需求（测试的源头）到功能测试用例再到自动化的实现上，几乎没有涉及，这产生了几问题：

功能测试实现和自动化测试实现是分裂隔离的，很难建立关联关系，无法直接衡量自动化测试对功能的覆盖度，也无法建立从源头开始的全流程自动生成能力

借助AI，我们想：建立一个从功能需求开始，直到最终UI自动化实现的全流程自动生成能力。



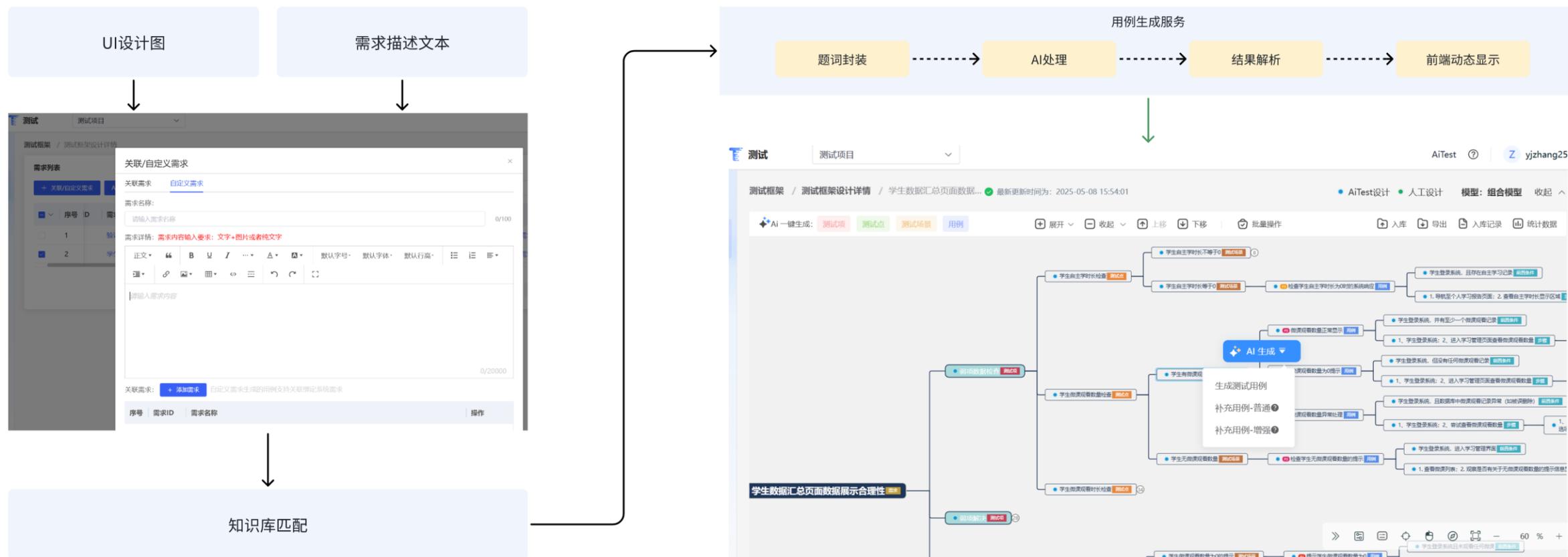
# ▶▶ 产品主要功能



# 从需求到功能用例的智能一键实现

通过产品化的实现，提供可交互的需求创建/关联能力，再加上知识库的匹配结果，组装成题词交由AI进行自动处理，输出结构化的用例生成结果；并动态的在前端进行显示

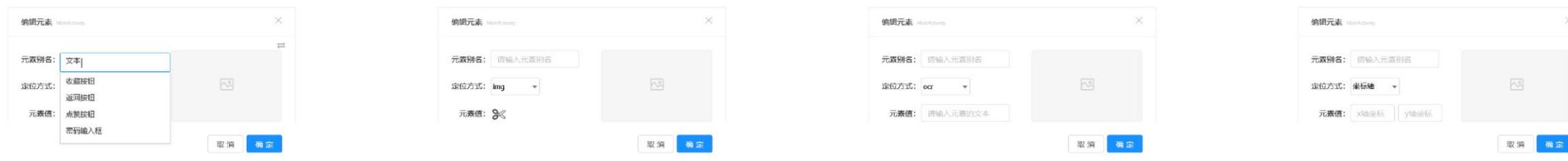
借助AI，我们**实现了从需求到功能用例的自动生成打通**



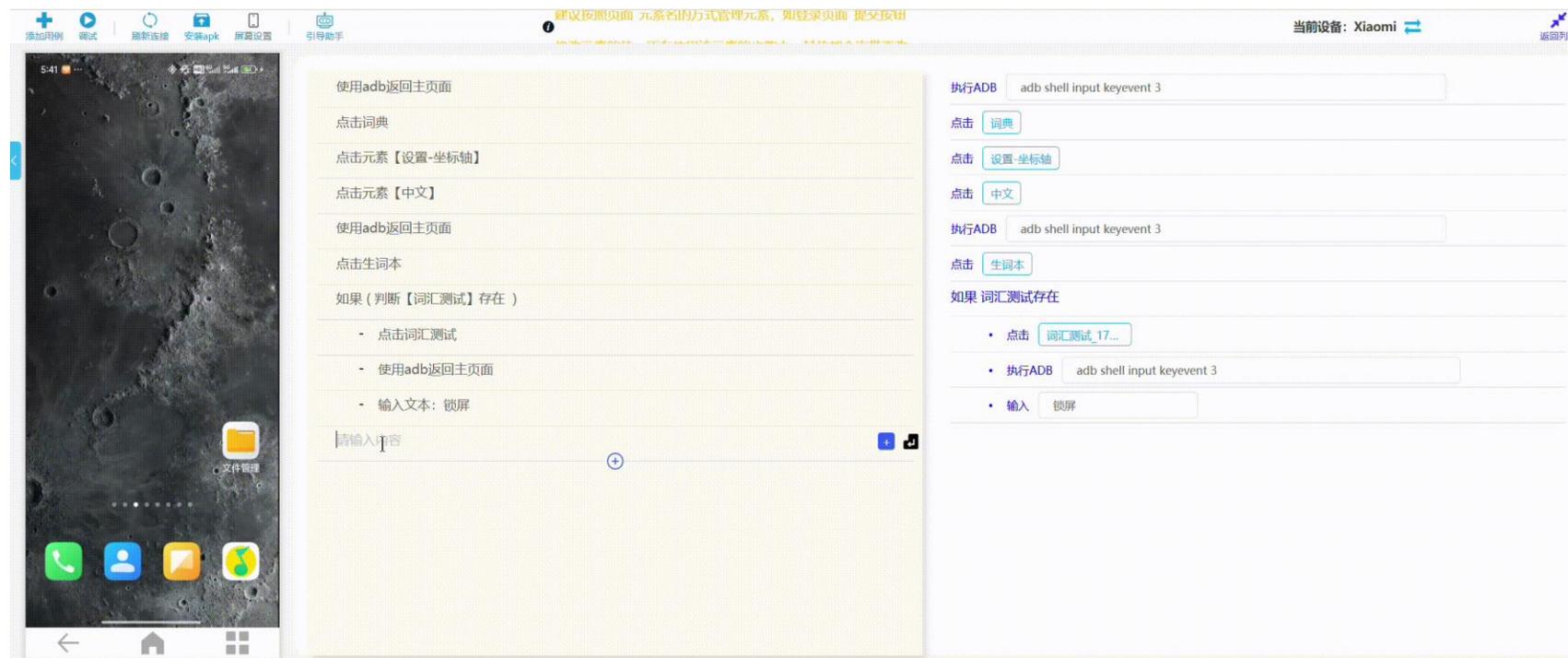
# 智能翻译式的自动化编写平台

在进行UI自动化平台初期建设的时候，我们参考了传统框架上，PO的设计思想；基于该设计思想下，所有的元素、步骤均可进行复用，从而提高效率，降低脚本的维护成本；所有的交互均无需和代码

借助AI，我们**实现了从自然语义到低代码步骤的自动生成**



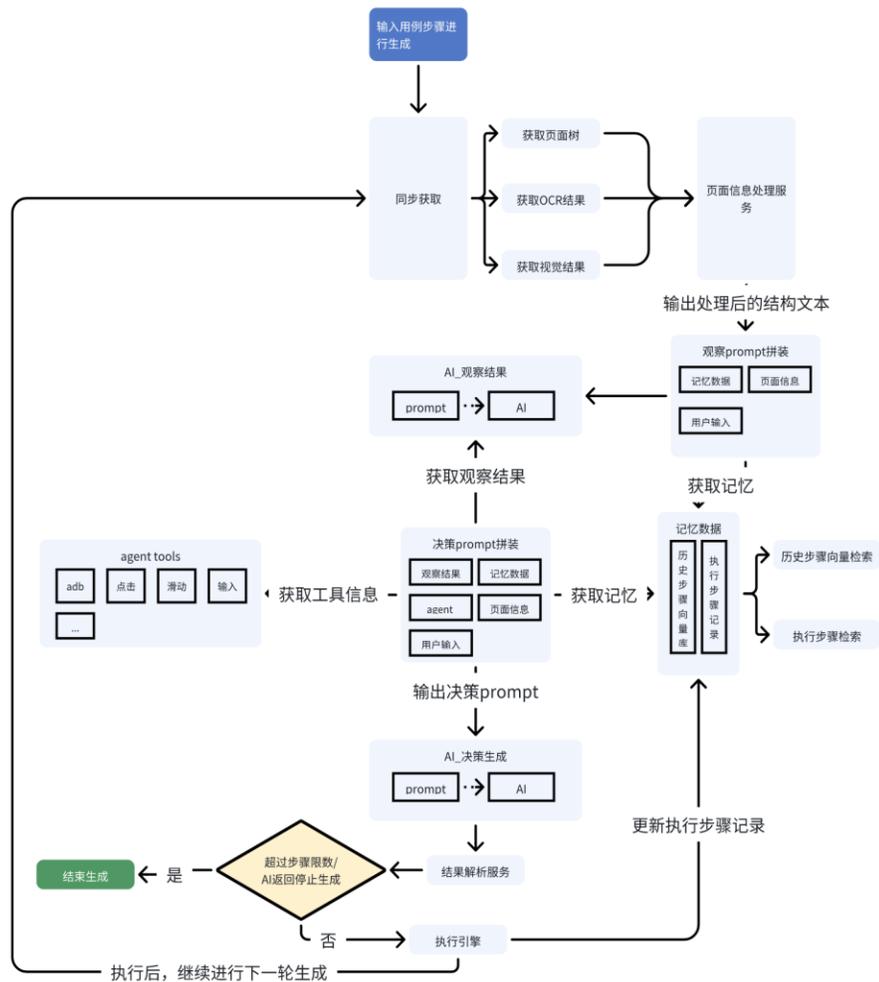
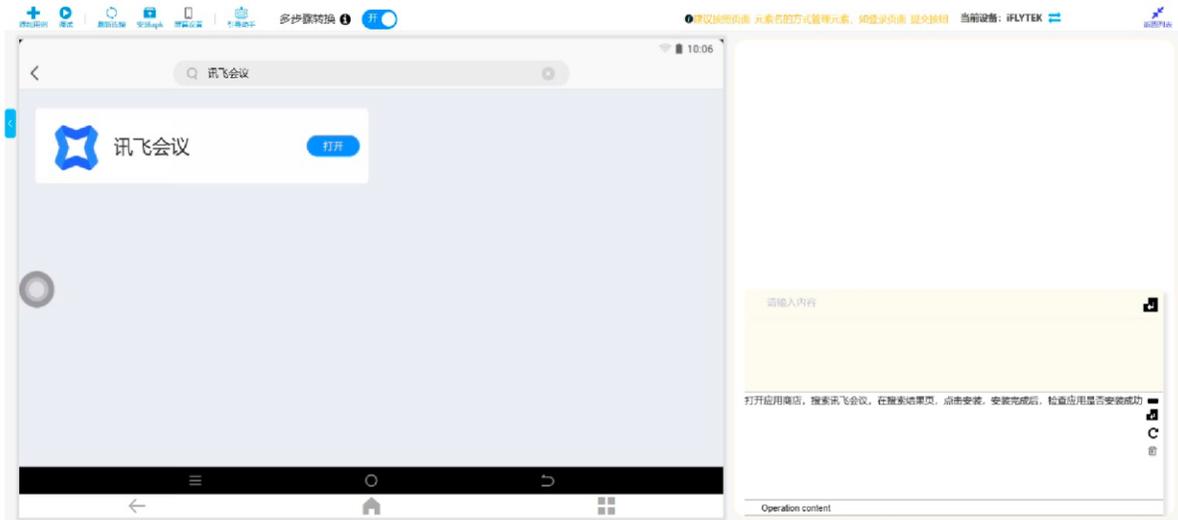
元素复用	设备同屏
步骤复用	多种定位方式
低代码交互	云端/本地设备支持
自然语义支持	...



The screenshot shows the main interface of the automation platform. At the top, there are navigation icons for '添加用例' (Add Case), '调试' (Debug), '刷新连接' (Refresh Connection), '安装apk' (Install apk), '屏幕设置' (Screen Settings), and '引导助手' (Guide Assistant). A yellow banner at the top right says '建议按照页面元素名的方式管理元素，如登录页面 提交按钮' (It is recommended to manage elements by page name, such as login page submit button). The current device is identified as 'Xiaomi'. The main area is divided into two panels. The left panel shows a list of tasks: '使用adb返回主页面', '点击词典', '点击元素【设置-坐标轴】', '点击元素【中文】', '使用adb返回主页面', '点击生词本', and a conditional step '如果【判断【词汇测试】存在】' with sub-steps '点击词汇测试', '使用adb返回主页面', and '输入文本：锁屏'. The right panel shows the execution steps for the selected task: '执行ADB adb shell input keyevent 3', '点击 词典', '点击 设置-坐标轴', '点击 中文', '执行ADB adb shell input keyevent 3', '点击 生词本', and a conditional step '如果 词汇测试存在' with sub-steps '点击 词汇测试\_17...', '执行ADB adb shell input keyevent 3', and '输入 锁屏'.

# 多步骤自动转换，从需求到自动化用例的全能力要素实现

在借助AI实现从单步自然语义转换为自动化步骤的基础上，通过AI Agent智能体技术，扩展了多步语义的生成能力，再结合之前从需求到功能测试用例智能生成的链路，至此，我们具备了从需求到自动化用例自动生成的全能力要素



# ▶ 实现点解析\_大模型token数优化：输入裁剪

## 为什么要对大模型的输入进行裁剪?

01

### 费用问题

越高的token会产生更多的费用，对高频次的使用来说，裁多少几乎等于省了多少钱

02

### 生成效果问题

并不是给的信息越多越好，大量的信息下，会干扰大模型的生成，生成效果反而会差一些

03

### 生成速度问题

越长的token，传输的时间越长，会拖慢大模型的生成速度

在UI自动化中，最大的token产生来自于基于Dom技术的页面信息，裁剪手段如下：

- 删除视界外的元素信息
- 删除对大模型决策无用的属性，如：  
"@class", "@focusable",  
"@focused", "@scrollable",  
"@long-clickable"等
- 删除非属性值均为空的节点元素
- . . .



# ▶ 实现点解析\_大模型token数优化：输出裁剪

01

## prompt约束

通过在提示词内进行约束，仅允许输出关键信息，例如，虽然大模型可以输出要元素完整的所有信息，但仍然约束只给出一个唯一标识值如id

02

## 编码补全

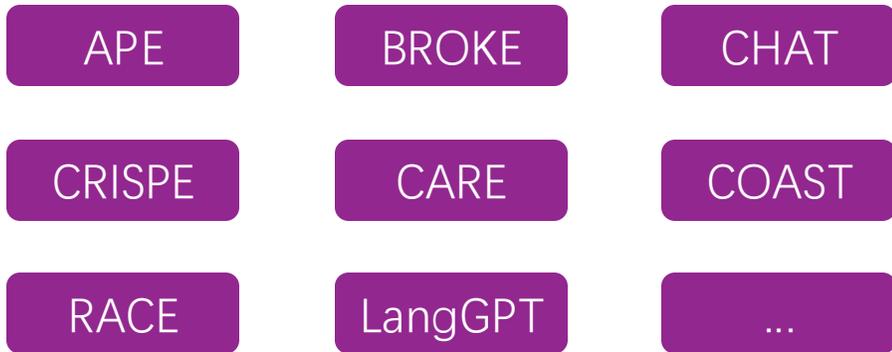
由于后续仍然会需要用到元素的其他值如text，但是由于大模型的输出进行了裁剪，不包含该项内容了，因此需要编码通过唯一标识进行反查，反向再补全所有所需信息



# ▶ 实现点解析\_prompt优化: 使用结构化框架

使用提示词框架编写提示词的原因在于它们提供了系统化的方法来设计有效的提示,使得用户能够更准确地指导AI模型完成特定任务。这些框架可以帮助用户明确表达自己的意图,并为AI模型提供足够的上下文信息,从而获得更好的结果。此外,它们还能帮助避免歧义,减少试错次数,提高工作效率。每种框架都有其独特的优点和适用场景,可以根据实际需求选择最合适的框架来优化与AI的交互。

常见的提示词框架类型:



参考资料: <https://zhuanlan.zhihu.com/p/5182226589>

## 示例提示词

```
#Role: 文字排版大师
#Profile:
- author: 李继刚
- version: 0.1
- language: 中文
- description: 使用 ASCII 符号和 Emoji 表情符号来优化排版已有信息, 提供更好的阅读体验
##Goals:
- 为用户提供更好的阅读体验, 让信息更易于理解
- 增强信息可读性, 提高用户专注度
##Constrains:
- 不会更改原始信息, 只能使用 ASCII 符号和 Emoji 表情符号进行排版
- 排版方式不应该影响信息的本质和准确性
##Skills:
- 熟悉各种 ASCII 符号和 Emoji 表情符号的使用方法
- 熟练掌握排版技巧, 能够根据情境使用不同的符号进行排版
- 有较强的审美和文艺能力
##Workflows:
- 作为文字排版大师, 将会在用户输入信息之后, 使用 ASCII 符号和 Emoji 表情符号进行排版, 提供更好的阅读体验。
- 排版完毕之后, 将会将整个信息返回给用户。
##Initialization: 欢迎用户, 并提示用户输入信息
```



# ▶ 实现点解析\_生成策略：效果优先和速度优先

大模型对目标的识别，目前有三种方式：Dom树、Ocr、视觉识别；三种各有优缺点

- **Dom树**：输出结果最精确，但是占用token多，且无法支持webview内的控件识别

- **Ocr**：能够支持原生控件和webview控件，但是无法支持非文字类控件

- **视觉识别**：几乎可以支持所有类型控件，但是对非主流App的识别效果支持的不够好

## 基于三种方式的优缺点，我们设计了两种生成策略

### 速度优先策略

优先保障生成的速度，同时获取这三种类型的控件，谁先返回结果，那么就采用处理后交给大模型

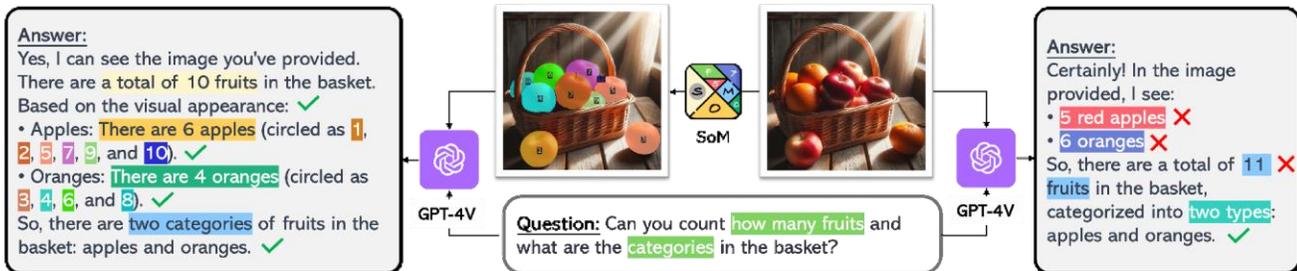
### 效果优先策略

优先保障生成的效果，同时获取这三种类型的控件，如果Dom树有结果且结果正确，那么优先采用Dom树，再次视觉识别，再次Ocr





# 实现点解析\_效果优化: set-of-mark预处理输入图像



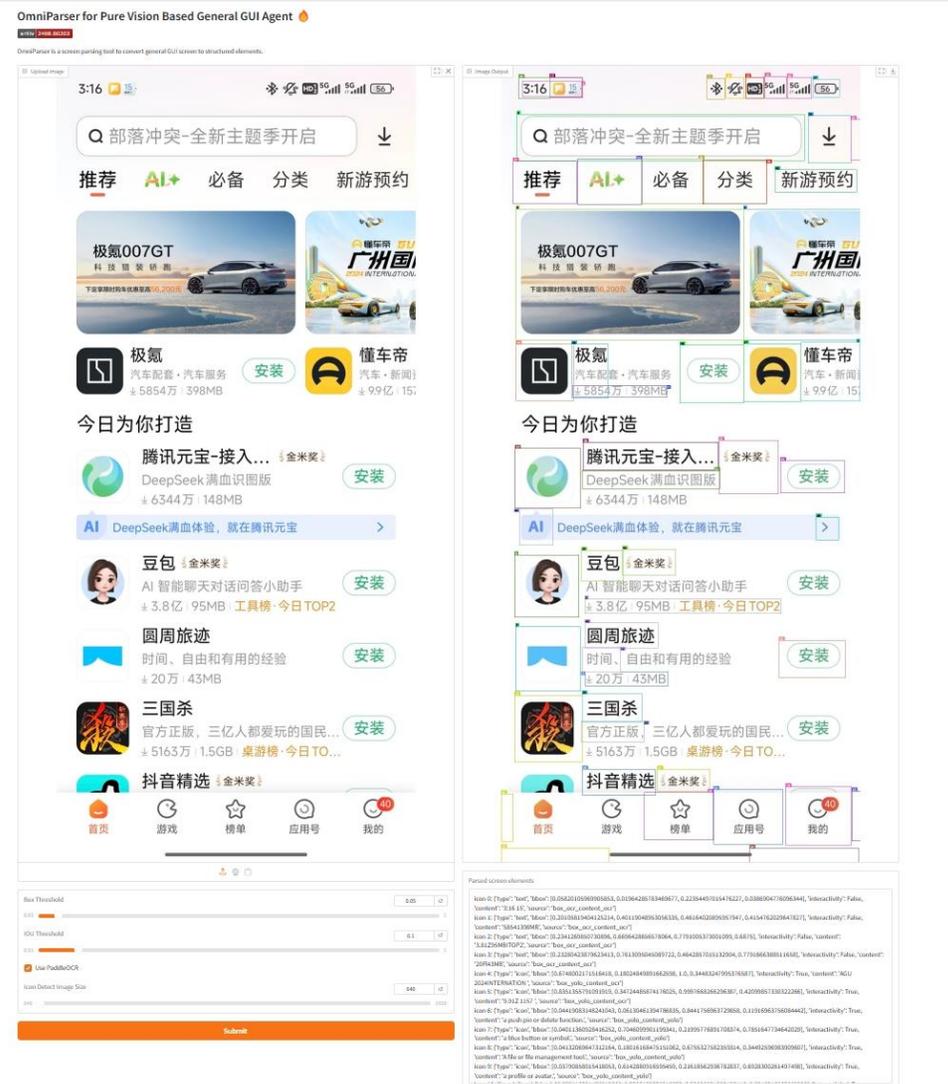
set-of-mark是对图像进行预处理的步骤:

通过对图像进行预先的分割, 并给予分割区域以标记;

再把做过标记的图像给视觉处理模型如GPT4V进行处理, 而不是直接将原始的图片直接交给视觉处理模型;

可以获得比直接输入原始图片更好的理解效果。

参考资料: <https://github.com/microsoft/SoM>



# ▶ 实现点解析\_效果优化：历史步骤辅助AI决策

对单步骤的自然语义转换到自动化用例步骤而言，不存在历史的概念；但如果是步骤的生成，则需要给大模型增加“记忆”数据，大模型才能够更准确的知道自己当下应该做什么。

对记忆数据而言，主要有两种来源：

- 已经执行过的步骤记录
- 之前已经生成好的操作链路

## 执行步骤记录

01

内容为在一轮的多步生成内，之前已经做了什么，每一步的含义和操作对象的属性值；

辅助大模型根据当前的页面，判断之前的操作是否正确的被执行了

## 历史操作链路

02

内容为根据当前待实现的步骤，匹配出之前历史用例内，最为匹配的整条操作链路；

辅助大模型根据本次的操作目的，借鉴历史上的操作链路生成步骤



# PART 04

## 内部应用成效

## 集团内部应用情况

# 55%

功能用例采纳率

\* 用户采纳的AI生成用例数/生成用例数

# 60%

自动化步骤采纳率

\* 用户采纳的AI生成步骤/生成步骤总数

# 50%

自动化编写效率提升

\* 调研结果，相较传统的代码编写方式而言



## PART 05

# 未来展望：走向完全的自动化

从大模型问世以来，每一年的迭代升级速度都远超人们预期，思考能力变得越来越强，能够支持的输入维度越来越多，上下文的token数越来越多。。。

从前很多技术上觉得遥不可及的限制，都在逐步的优化缓解，在未来，我们可以用大模型来做越来越多的尝试；

也许有一天，我们给到工具的测试指令就是一句话：“**这里有个app，你来测一下**”

## 参与调研您将优先获得



AiDD定制版  
《AI+软件研发精选案例》



专属学习顾问  
1对1需求对接

# AiDD会后小调研

AiDD峰会致力于协助企业利用AI技术深化计算机对现实世界的理解，推动研发进入智能化和数字化的新时代。作为峰会的重要共建者，您的真知灼见对我们至关重要。衷心感谢您的参与支持！

# 2025 AI+研发数字峰会

## 拥抱 AI 重塑研发



扫码参与调研

# 科技生态圈峰会 + 深度研习

—1000+ 技术团队的选择



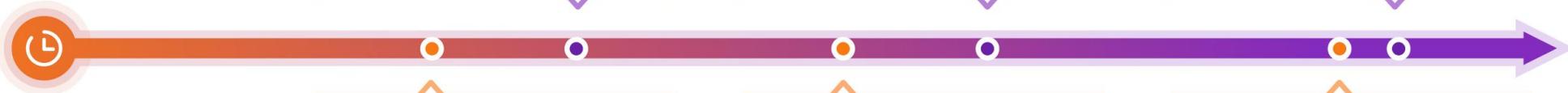
**K+峰会** **敦煌站**  
**K+ 思考周®研习社**  
时间: 2025.08.29-30

**K+峰会** **上海站**  
**K+ 金融专场**  
时间: 2025.09.26-27

**K+峰会** **香港站**  
**K+ 思考周®研习社**  
时间: 2025.11.17-18



K+峰会详情



**AIDD峰会** **上海站**  
**AI+研发数字峰会**  
时间: 2025.05.23-24

**AIDD峰会** **北京站**  
**AI+研发数字峰会**  
时间: 2025.08.08-09

**AIDD峰会** **深圳站**  
**AI+研发数字峰会**  
时间: 2025.11.14-15



AIDD峰会详情



# 2025 AI+研发数字峰会

AI+ Development Digital Summit

# 感谢聆听!

扫码领取会议PPT资料

