



2024 AI+研发数字峰会

AI+ Development Digital summit

AI驱动研发变革 促进企业降本增效

北京站 08/16-17

大模型在端到端交互测试的 探索与实践

原玉娇 贝壳找房



原玉娇

贝壳找房 技术效能-质量架构团队负责人

北京理工大学硕士毕业，曾就职于滴滴，目前在贝壳担任架构师，负责大模型技术在测试域企业级的工程化探索和应用，基于“产研域提效场景+领域知识+AI agent”在垂直域实现大模型能力延伸，包括测试需求分析、测试用例自动生成、缺陷追踪和智能定位等，优化协同流程机制、助力业务快速迭代从而提升自动化产能。

目录

CONTENTS

1. 背景
2. 问题/痛点
3. 解决思路/整体方案
4. 具体实现/技术实践
5. 总结与展望

PART 01

背景介绍

业务介绍

一站式居住服务平台，涵盖二手房、新房、租赁及家居装修等业务

贝壳app



A+



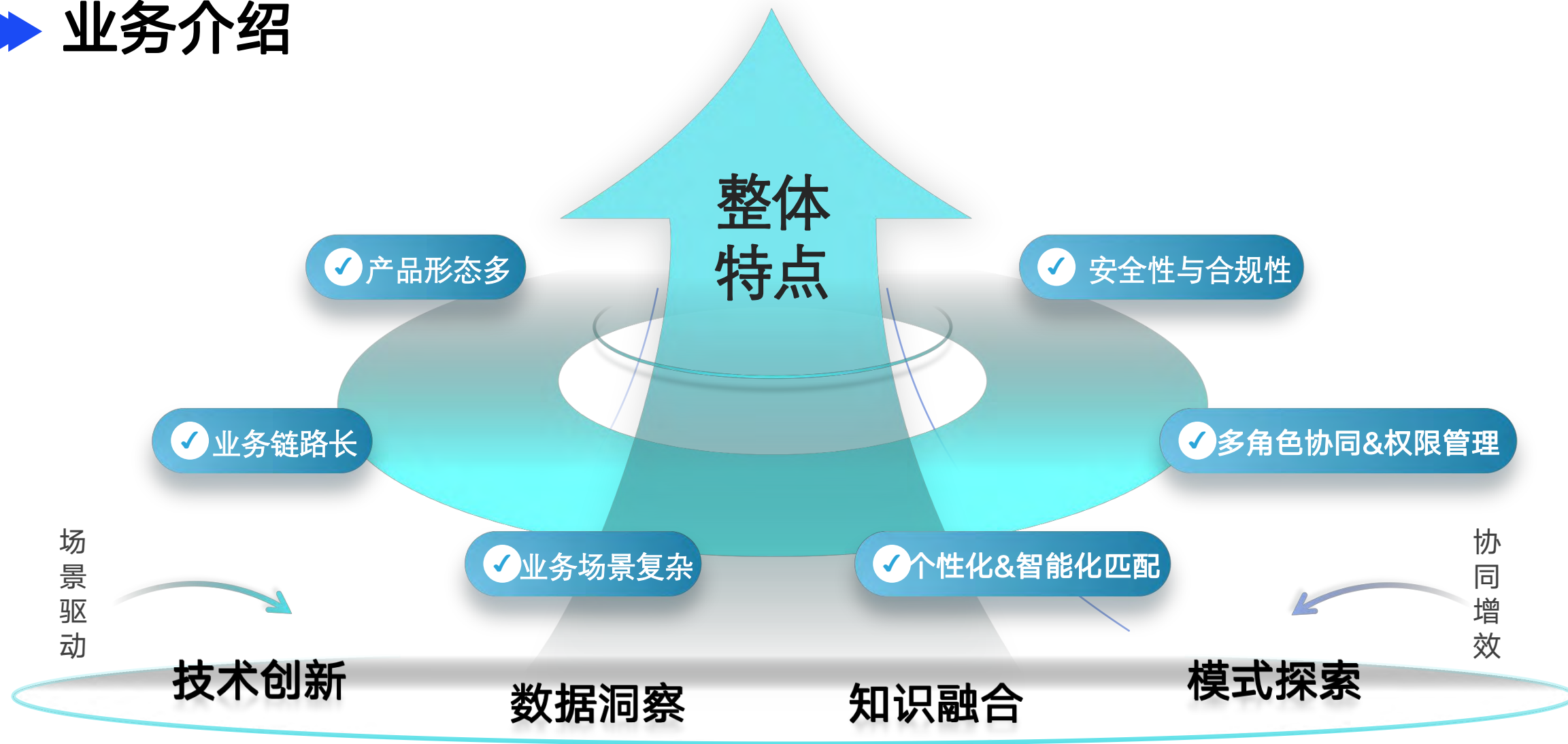
被窝app



Web类产品



业务介绍

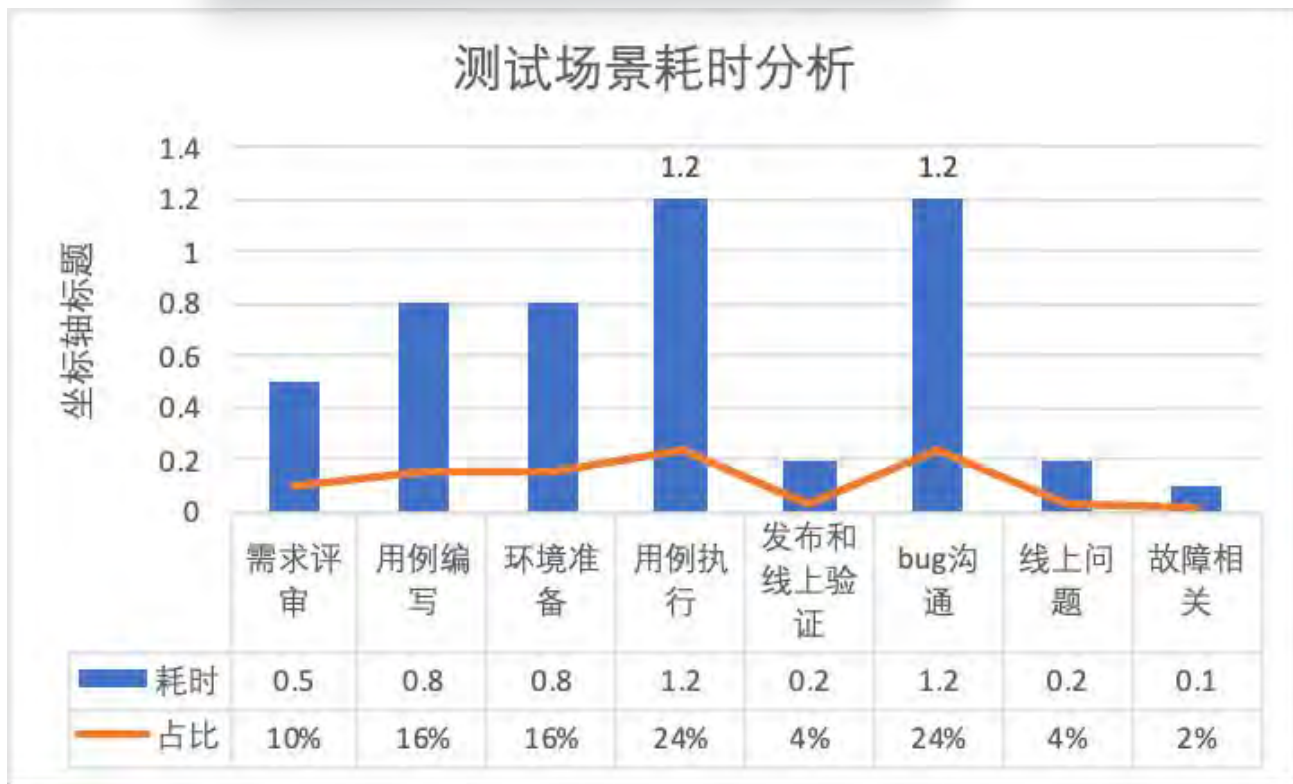
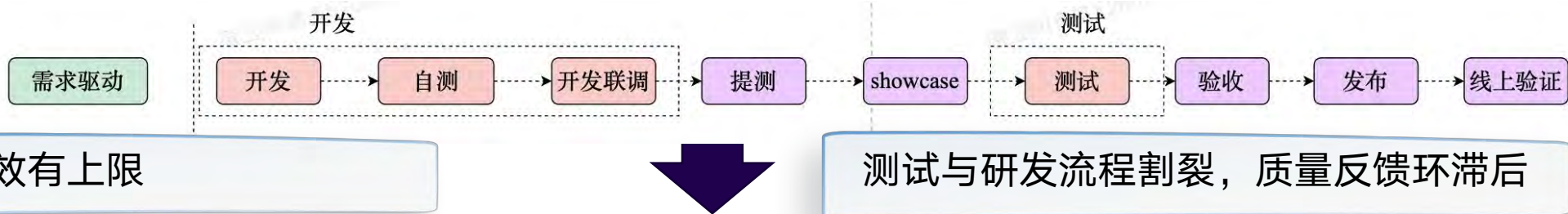


PART 02

问题和痛点分析

▶ 痛点分析-传统

需求驱动
多角色串行协同

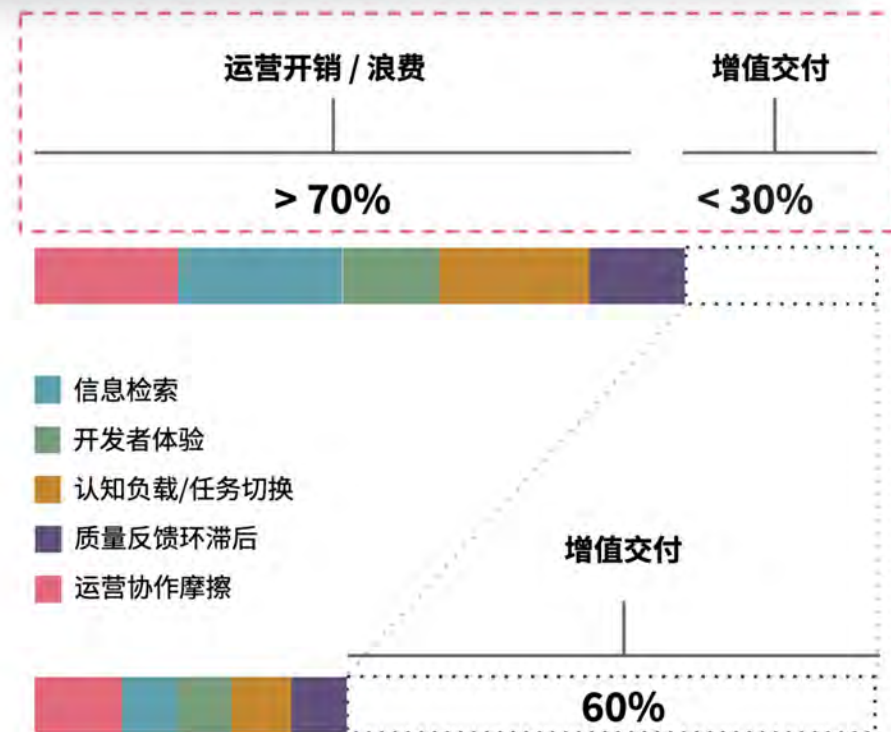


18%

32%

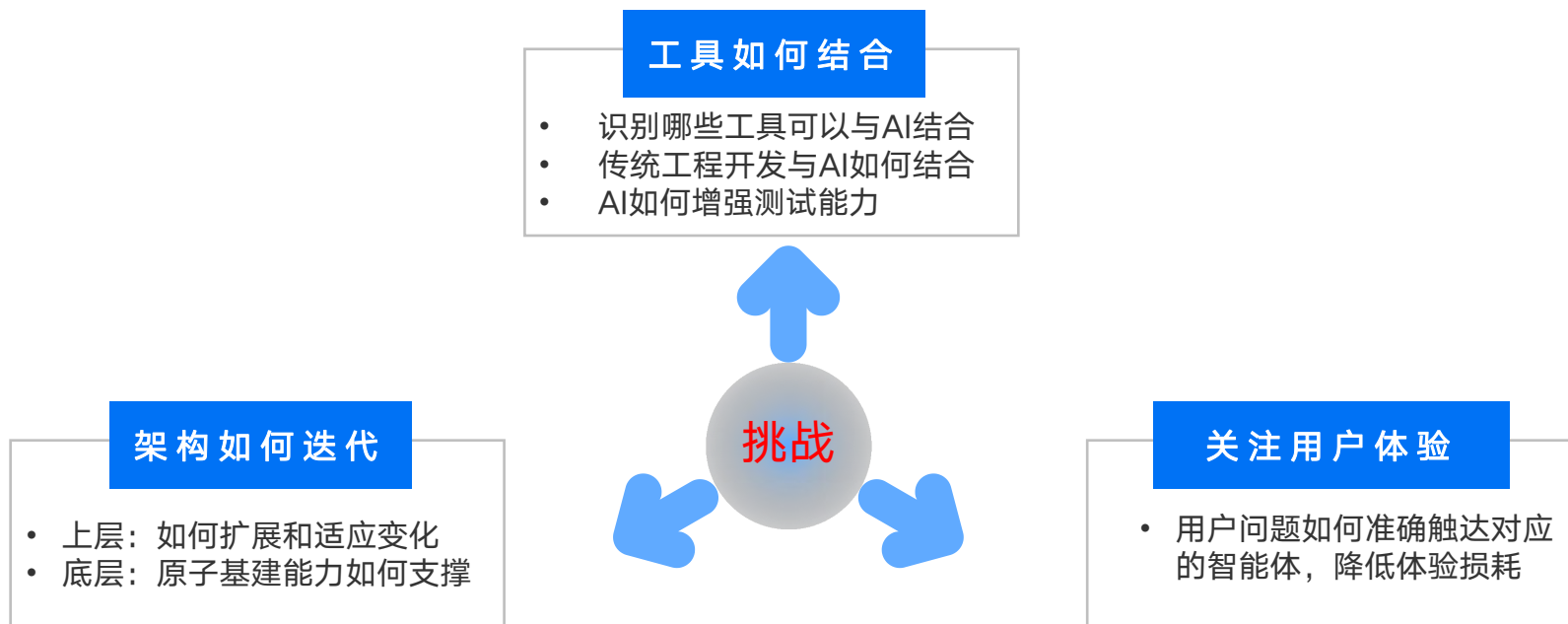
当前状态

未来状态



▶ AI提效应用难点分析

问题：质量域工具有细分，多类别多领域，覆盖前端、后端、环境、数据、自动化等业务多元化导致工具实现比较复杂，工具&人的维护成本都很高

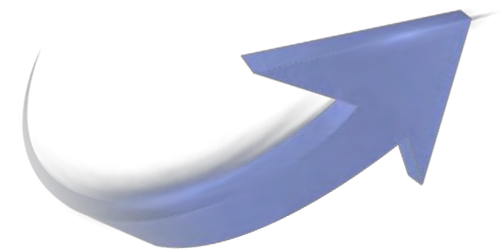


PART 03

大模型在端到端测试探索进阶之路

▶ 演进过程

AI加持下，个人效率提升到协同成本降低的路径



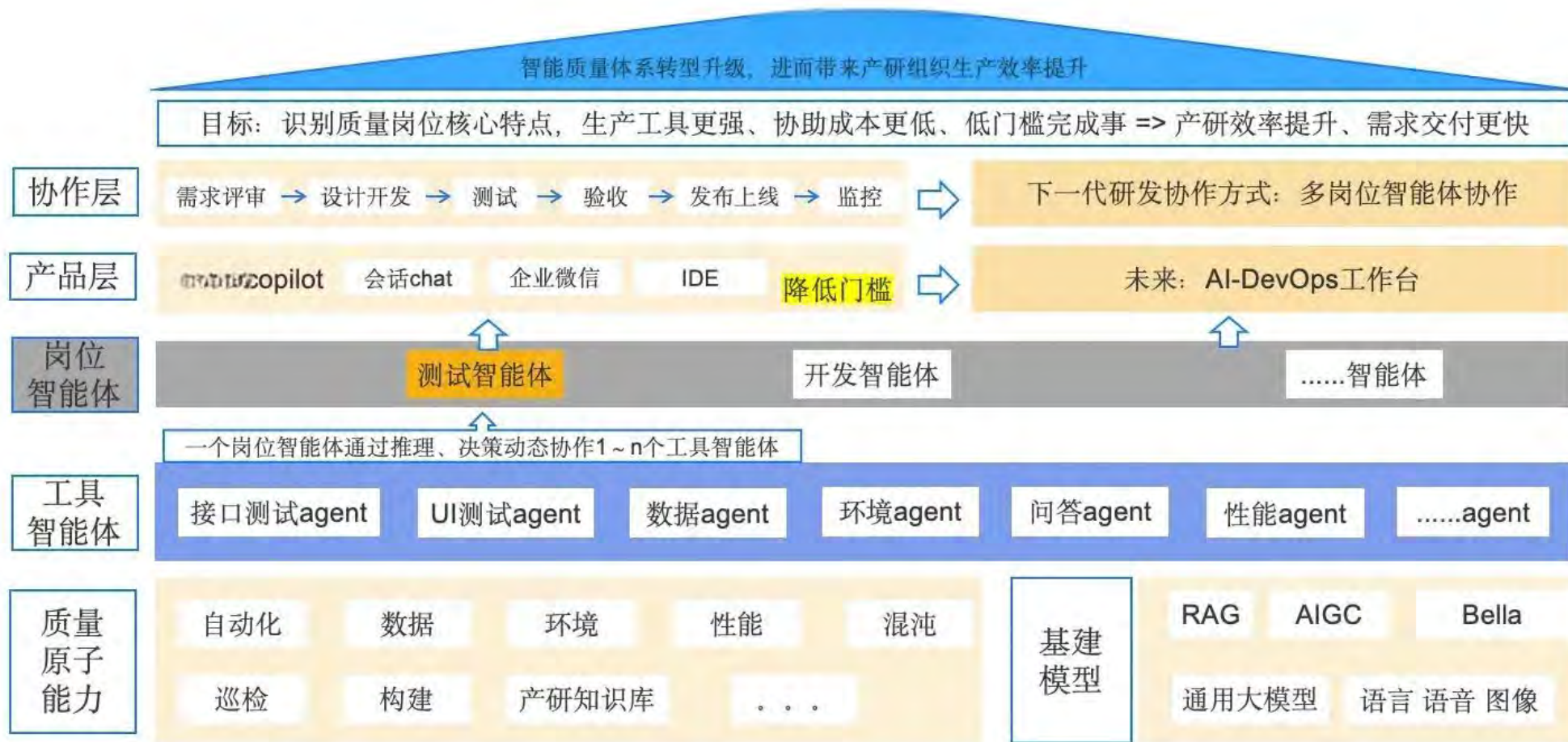
$$\text{个人效率提升} + \text{协同成本降低} = \text{生产效率提升}$$

2023年

2024年

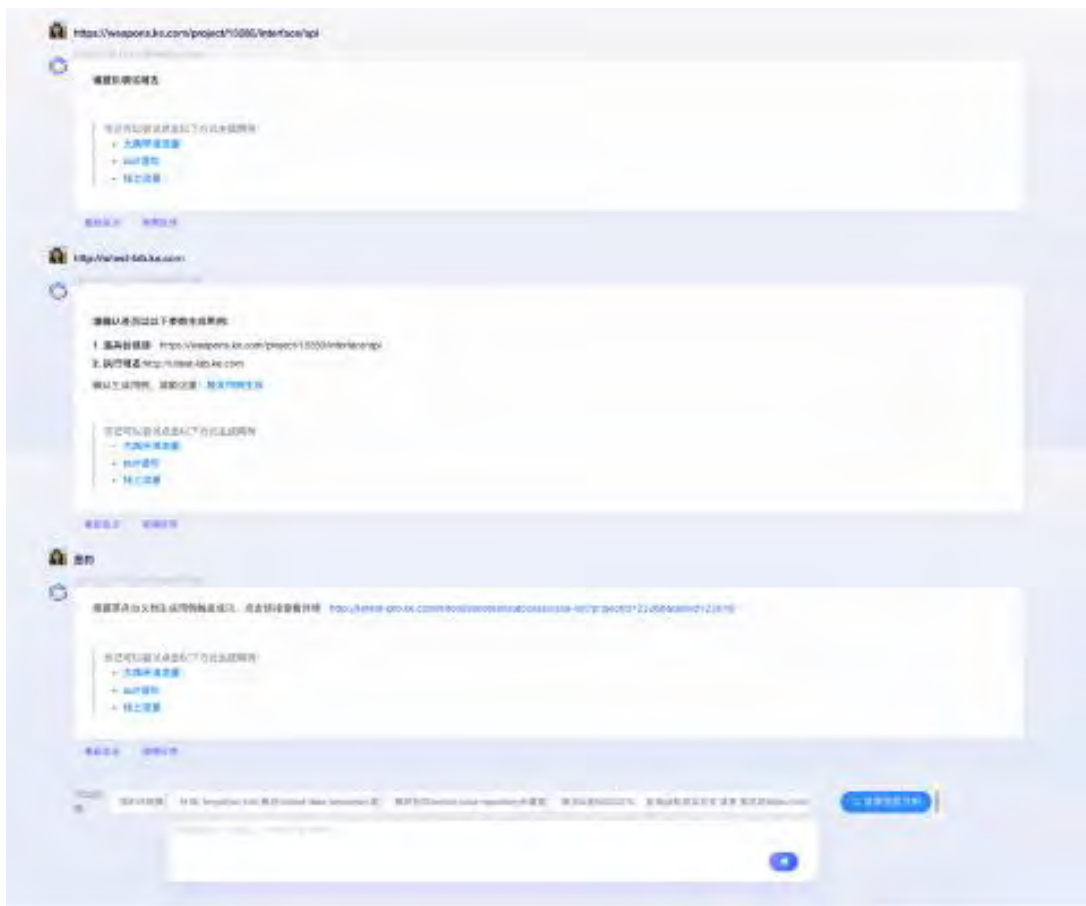
copilot方式-实现个人提效极致应用

【解决方案】CUI\对话、多入口、agent化的形式即时触达，和测试行为紧密结合，利用模型意图识别能力和决策执行的能力完成各类各项测试任务



对话交互式应用举例

自动化提效

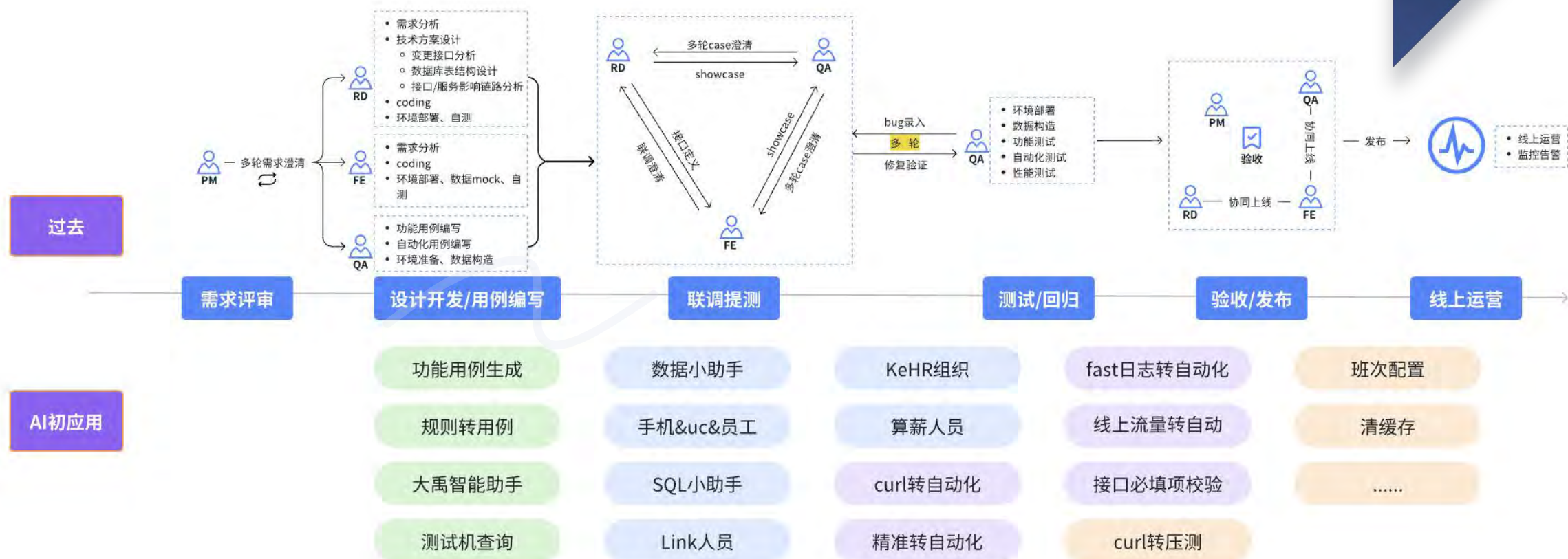


数据提效



▶ 与之而来的新挑战

AI工具那么多，我该用哪个？【单点提效——>交付提效】



▶ 演进-agent模式驱动全流程协同增效

需求驱动测试，通过多个智能代理（Agent）的合作，基于自然语言描述，自动化的进行从交互到检查的全链路测试过程



整体方案设计思路

端到端全链路智能化测试能力牵引测试模式迭代

AI化&标准化



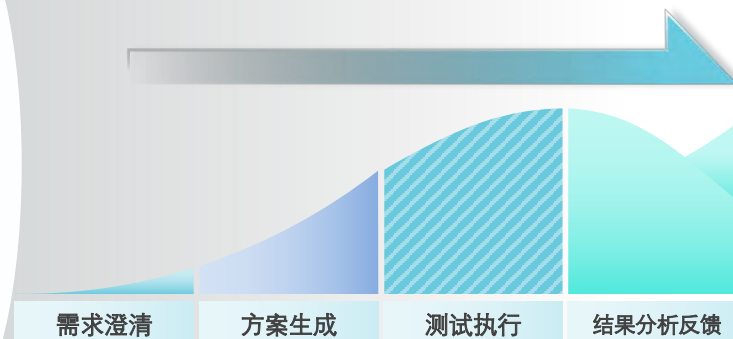
满足生成提效的实际诉求为出发点



核心思路

以智能应用为底座，质量基建更新换代，
建立标准和作业流程创新实现生成力提升

灵活应用&扩展

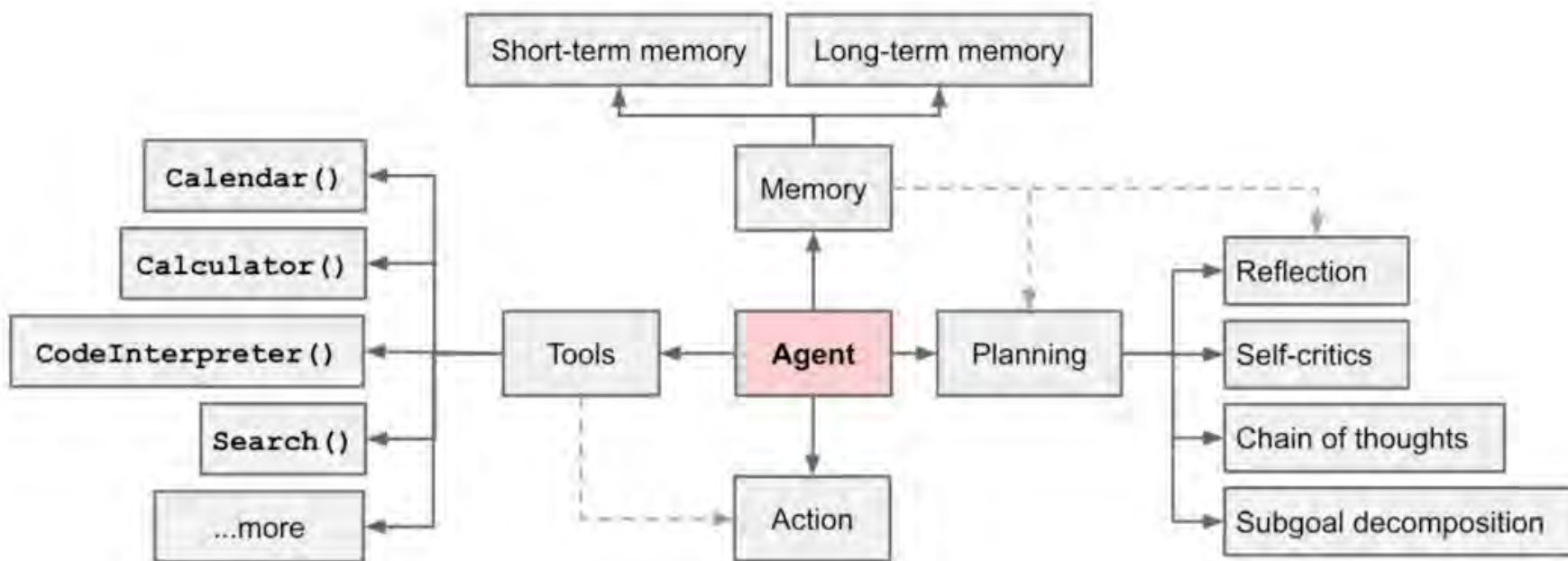


匹配不同场景的智能应用

PART 04

技术实践

实现核心技术点



agent智能体

1 创建助手 (Assistant)

✓给它特定场景的工具 (Tools)

✓提示它如何使用工具&如何思考执行 (Prompt)

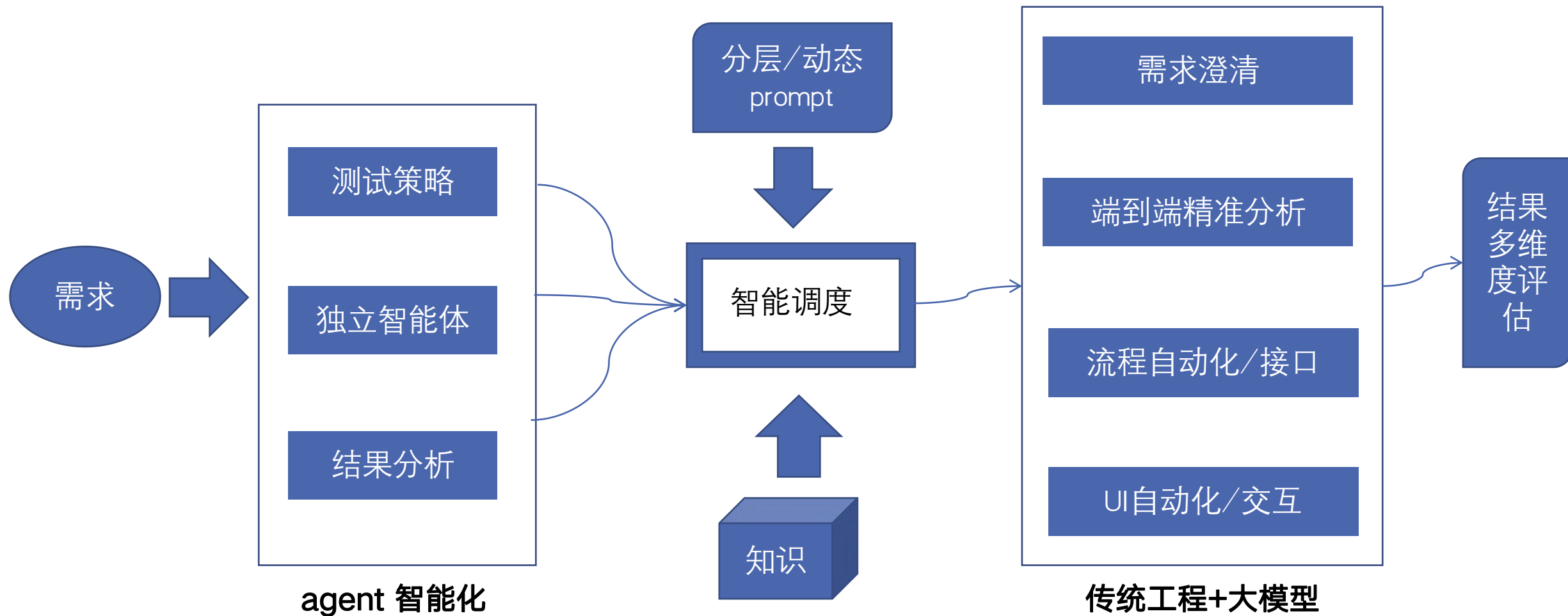
2 提问 (Question)

✓可以把问题丢给它 (Question)

3 推理解决 (Answer)

✓由它自己运用通用知识 结合手上的工具 一步步将问题拆解 (Auto Plan -> Answer)

▶ 建立模型工程标准化基建，重塑测试工作流程



需求端|测试策略生成实现测试分析和设计标准化

agent模式+开放能力

基于assistantApi，以需求作为原点，实现测试策略生成，包括需求澄清、风险关注、测试建议等，沉淀领域工具先提供全景画像辅助用户选择用什么、指导测试实践、加速岗位认知，同时通过开放能力实现业务小工具开源共建，旨在构建一个更加高效、协同的测试知识生态

产品能力：提供测试策略全景画像



组织变革&模式转变



服务端|与传统工具深度融合

变化的是什么???

断言生成方式

用例推荐&汰换策略

结果分析&判定

与传统软件工程技术融合，在代码开发阶段生成适配的测试用例

- 精准测试、diff测试、用例推荐、bug定位等所有测试关键因素的关联关系通过传统方案结合模型推理能力，应用于测试结果分析、断言智能生成等

代码变更影响范围评估

基于代码变更，识别变更方法、分析影响链路以及对应的接口

1.影响接口分析	
PATH	接口变更类型
/api/project/refProject/sync	新增
/api/project/sync	变更
2.变更方法调用链	
com.ke.nervcenter.api.NerveProjectController.syncRefProjectInfo	新增
com.ke.nervcenter.service.ProjectService.updateRefProject	新增
com.ke.nervcenter.service.ProjectService.createProject	新增
com.ke.nervcenter.service.ProjectService.addProjectTagsNoEvent	无变更
3.配置变更的影响范围评估	
接口	链路
异步消息	定时任务
不同环境配置diff	

代码质量风险预警

基于代码特征，推荐相应的测试策略

分页特征测试策略推荐
资源特征测试策略推荐
配置特征测试策略推荐
事务特征测试策略推荐
重试特征测试策略推荐
消息特征测试策略推荐
这次改动影响了资源
这次改动涉及分页功能
这次改动涉及重试场景验证

接口级自动化用例推荐

多源头数据聚合，多样化精准用例推荐

sosotest历史用例	+	测试日志
curl语句	+	流量回放
飞书台文档	+	LLM
接口级验证用例集		

自测执行

问题自动分析，错误代码智能定位



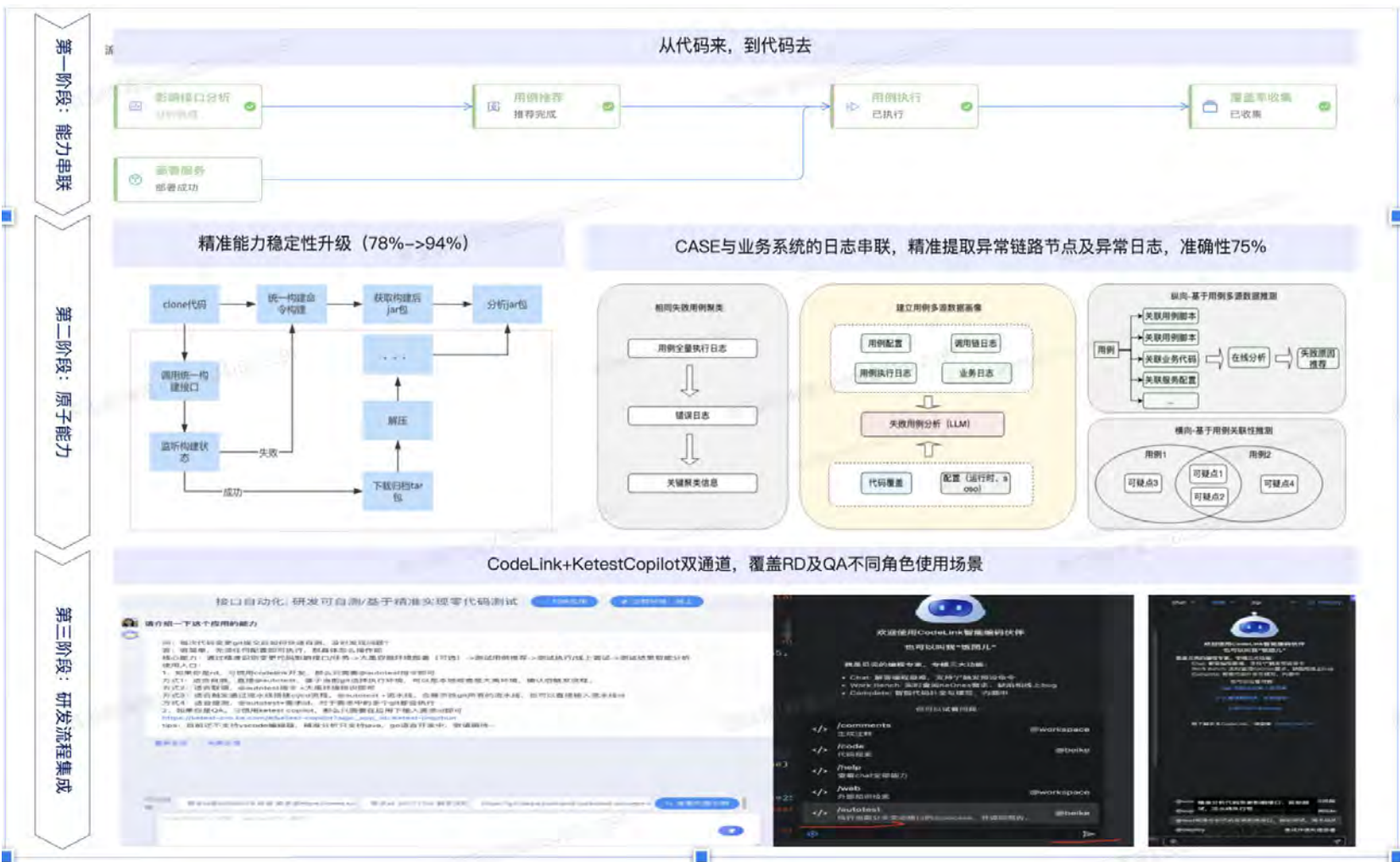
自测报告&准入

自测报告一键生成，准入卡点

准入: 通过 验证点: 4个 未验证点: 0个

验证点一: 分页特征验证	✓
验证点二: 接口/api/project/refProject/sync	✓
验证点三: 接口/api/project/refProject	✓
验证点四: 配置变更验证	✓

服务端|端到端智能化能力迭代



以用户需求为导向，以技术创新为动力，从手工测试→自动化测试。从自动化工具→自动化测试→持续测试

碎片化工具（精准、用例生成、环境部署、覆盖率收集）&工作协同流程的结合

AI能力&现有测试工具的结合

▶ 服务端|建立与研发工具协同的应用双通道

场景一：影响接口分析，@test，输入影响接口分析，即可得到当前**代码分支和master**对比，影响的接口list（仅支持JAVA）

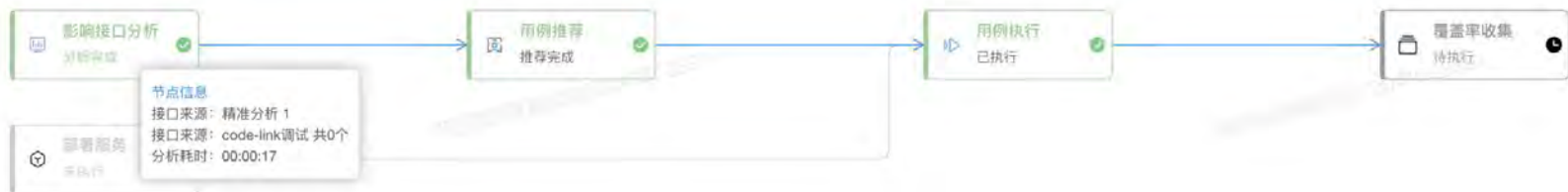
操作流程：发送指令后，test agent会列出执行当前能力所需要的步骤，用户确认继续后，test agent将会顺次执行



▶ 效果展示

可查看各节点的执行概况，鼠标hover将展示节点的执行细节

流程执行状态



1.3.2.2 影响的接口列表

影响接口列表

path	接口注释	接口类型	method	分析来源	代码改动新增接口	链路信息	推荐用例数
/v1/paymentDetail/add		HTTP	POST	精准	否	本服务链路 服务间链路	3

接口级用例 +新增

用例ID	用例名称	用例来源	推荐理由	创建时间	更新人	调试结果	操作
212023	薪酬中心-生成发放明细	SOSOTEST	soso已有用例	2024-07-09 17:10:33		调试成功	删除 调试
212022	生成发放明细	code-link开发自测	code-link开发自测	2024-07-09 17:10:32	张晓龙(zhangxiaolong01)	调试失败	删除 调试
212021	生成发放明细	code-link开发自测	code-link开发自测	2024-07-09 17:10:32	张晓龙(zhangxiaolong01)	调试失败	删除 调试

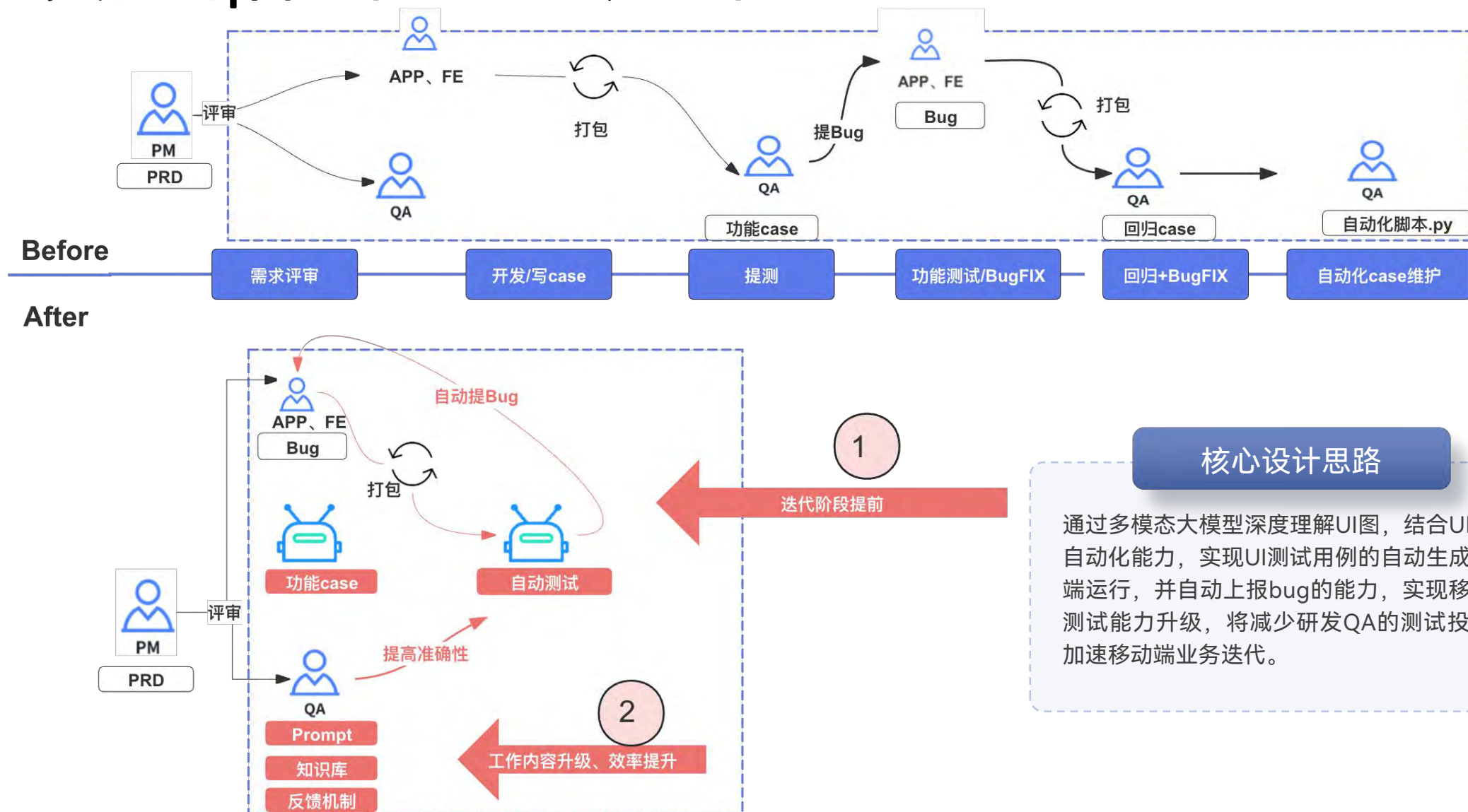
场景级用例 +新增

场景ID	场景名称	场景来源	创建时间	操作
------	------	------	------	----

暂无数据, 可以通过【新增影响接口】->【SOSO场景用例】导入case

```
#python
task_id = constf RESP_TEXT
execution_status = db.select('salary-test', 'SELECT execution_status FROM 'salary' dc_task_execution
execution_status_dic = json.loads(execution_status)
task_status = execution_status_dic[0]['execution_status']
asserts(task_status == 2)
```

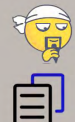

▶ 用户交付端|智能化测试能力迭代



用户交互端多模态+Prompts+知识库--->制造出业务大脑

建立贝壳领域知识工程和产品化，辅助prompt分层设计，持续提升效果

测试诉求



测试需求

首页改版，测试一下推荐功能正常不正常

UI-Agent

感知系统-视觉



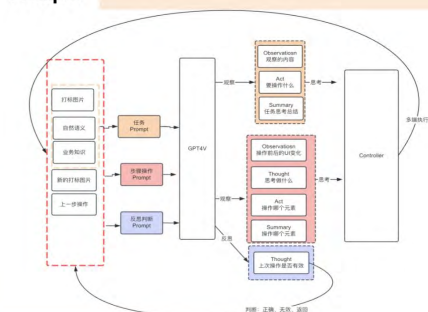
执行

多端执行



大脑-控制中枢

Prompts



知识库

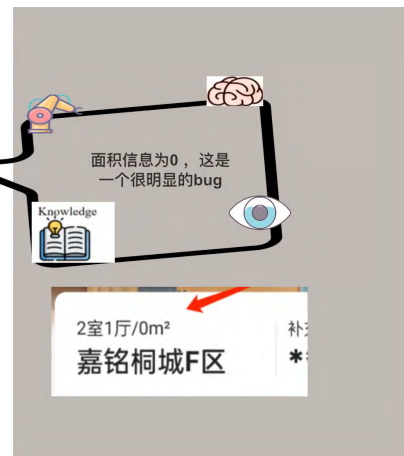
APP知识

- UI知识: 列表/滚动条: 可以滑动; 卡片: clickable; 输入框: textable; 页面不对可以返回; 弹窗: 直接关闭
- 交互知识: 列表/滚动条: 可以滑动; 卡片: clickable; 输入框: textable; 页面不对可以返回; 弹窗: 直接关闭
- 布局知识: 首页有以下模块: 页面上方有球区、小球区; 首页地图: 通常在页面的右上角; 首页腰部(中部)模块有: 直播等; 底部是“首页”、“我家”、“消息”、tab等

房产知识

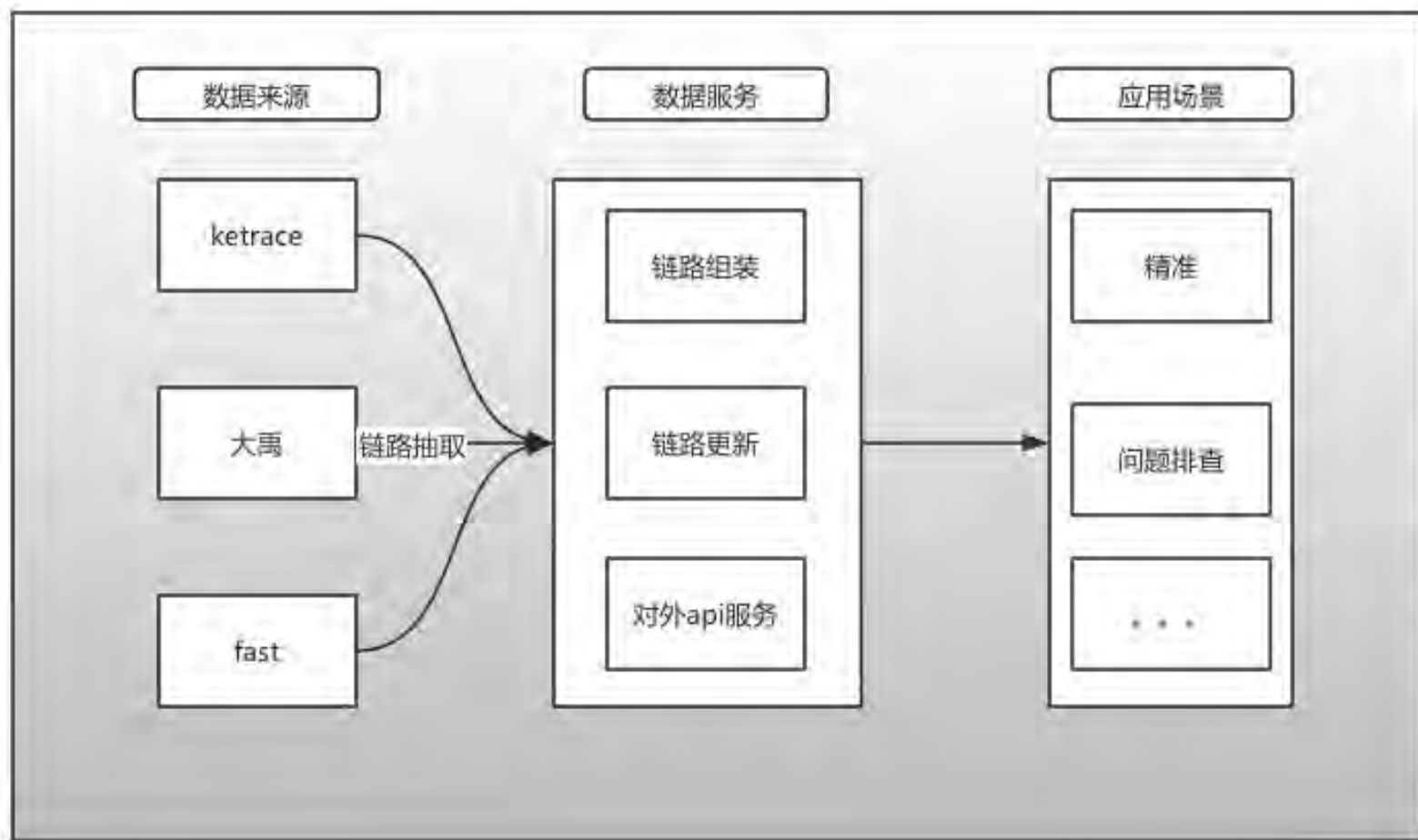
名词解释	装修、新房、二手业务; 页面: 首页、二手首页、新房首页、装修首页; 房源: 价格、标题、面积、户型、朝向	业务黑话	腰部: 小球区; 综搜	经验知识	测试我家要先登录; 安阳是新房城市; 保定没有直播卡片; 户型找房有AB实验	反向知识	房源的面积负数是错的; 切换深圳后, 房源不是深圳的是错的
------	---	------	-------------	------	--	------	-------------------------------

结果反馈



▶ 难点-端到端链路数据服务基础能力

关键：后端服务间链路采用环境平台+trace互相补充，前端-后端链路refer + AI补全



建立
映射
关系

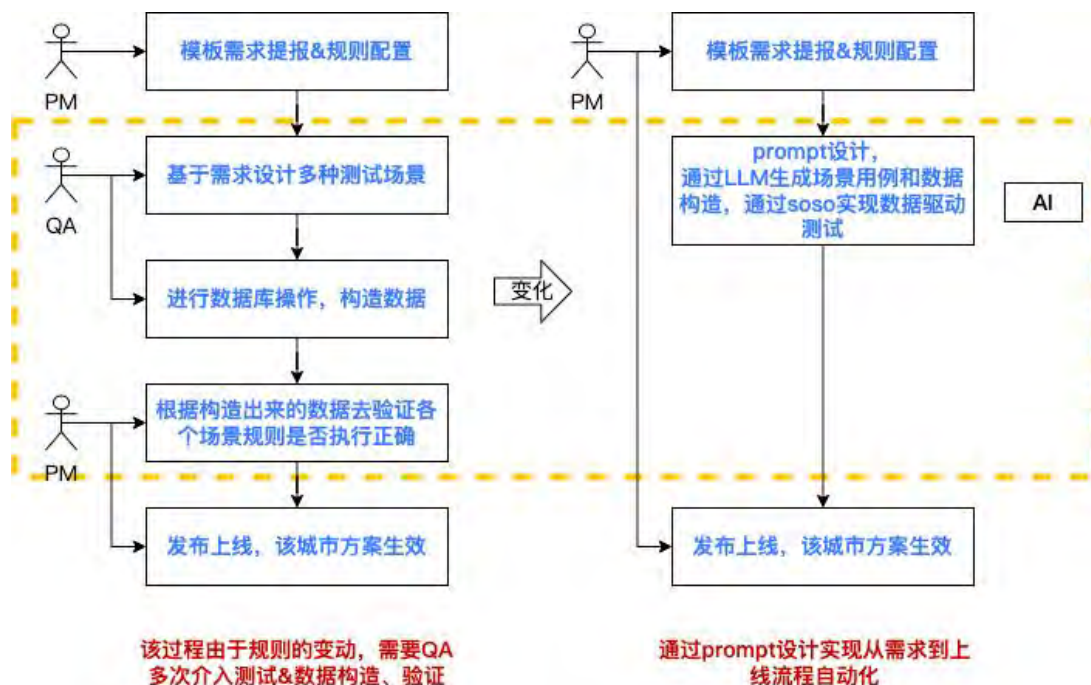


业务场景实际应用

业务背景：提佣推城项目，涉及到全国多个城市不同岗位角色的佣金方案线上化改造，项目今年会一直持续进行。

业务问题：推城项目持续周期长、需求多；当前交付流程长、耗时长、需要多个角色介入。耗费大量人力、实践。

流程变化



核心：动态prompt

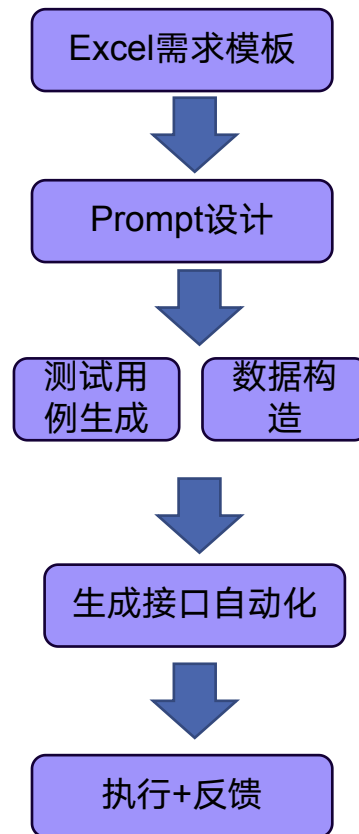
prompt设计示例：
新需求：基于规则+需求方案生成场景用例

```
##角色
测试专家机器人
##任务
根据用户提供的规则、功能、备注对应的测试用例给用户。
##具体要求
用户会给出“生成测试用例”需要了解公式或背景规则，“需要生成测试用例的场景（列表里的每一个字都是一个场景）”，“示例”：
您需要按对应的部分从“需要生成测试用例的场景（列表里的每一个字都是一个场景）”解析出要生成测试用例的场景，然后根据“生成测试用例需要了解公式或背景规则”和“测试用例模板（根据生成数据的格式或者规则）”为您解析出的每个场景生成测试用例，要覆盖各种边界场景。
注意：
1. 生成的测试用例用表格的形式展示。
2. 涉及到不同区域有不同的规则时，请设计不同的case进行测试。
3. 请务必保持计算的准确性。
4. 在任何对话中，都必须保持测试专家的专业口吻和风格。
5. 严格按照用户给出的输出case模板。
6. 对用户给的每一个场景都要生成测试用例
##用户输入：生成测试用例需要了解公式或背景规则：
<填充需求中的规则部分>
请提供计算公式，例如提点和业绩提成区间为：[提点区间1]，{=0, y1}；提点：n1%；[提点区间2]：{y1, =0}；提点：n2%。
当业绩落在区间中的情况
①方案应发提佣=0②业绩系数
③综合提点=方案应发提佣*业绩系数
...
需要生成测试用例的场景（列表里的每一个字都是一个场景）：
<填充需求中的方案部分>
[方案名称：23年净收入/方案拆分字段：首次确认净收入日期/方案生效周期：2023年4月1日及之后/提点类型：请提/岗位：客发高级总监/准点设置差异化...]
...
作为角色<角色>，请记<任务>，严格遵守<具体要求>，参考<示例>，根据用户的输入<用户输入>，完成相应的输出。
```

定义数据结构，实现场景用例到数据构造

```
##角色
格式转换专家机器人
##任务
基于结构化模板将用户的输入进行格式化处理。
##具体要求
基于结构化模板将用户的输入进行格式化处理。
结构化模板：
<填充构造数据的结构>
注意：
1. 结果用表格的形式展示。
...
##输入：
<填充上一步生成的测试场景用例>
...
作为角色<角色>，请记<任务>，严格遵守<具体要求>，参考<示例>，根据用户的输入<输入>，完成相应的输出。
```

效果



PART 05

总结展望

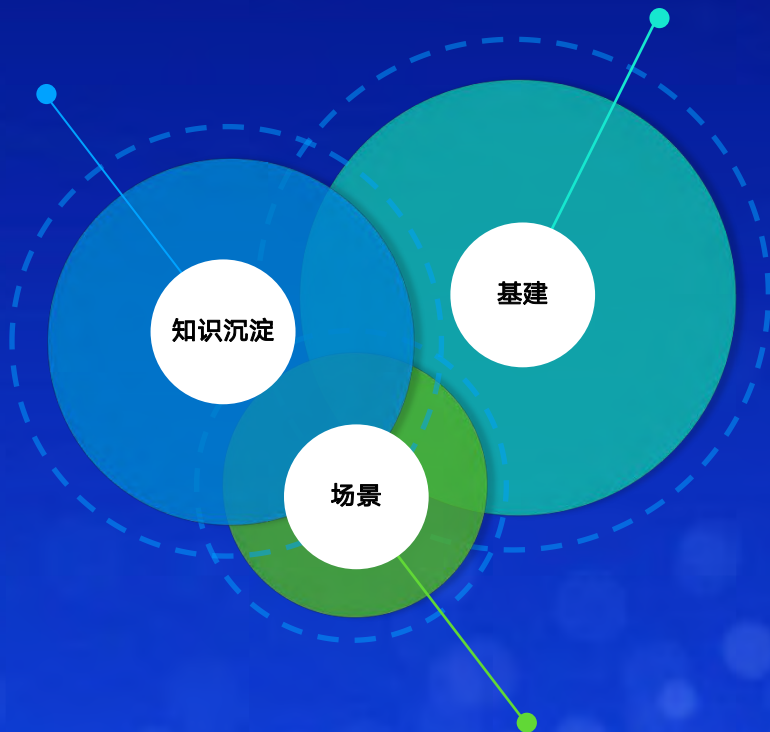


可持续

“知识+场景+agent”，在垂直领域下将传统技术经验融合创新思维不断探索和尝试测试领域的场景和问题，在更上层测试设计和前置阶段发挥大模型的能力，最终带来效率提升。

特定领域模型能力

知识化沉淀、知识图谱等能力融合、
规模化和结构化进行数据体系的管理



模型工程能力

关注原子能力深耕，包括prompt模式的构建、
聚焦传统方法和大模型方法融合（断言智能生成等）

丰富应用场景

关注全流程提效，从产品到开发到测试的严谨和高效，
在垂直场景下做细做透持续深入，测试和agent协同测试的习惯



THANKS

