AiDD
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

# AI Checklist
# QUNAR测试域结合AIGC提效实践

崔宸 去哪儿旅行

# 演讲嘉宾



## 崔宸

去哪儿旅行 高级开发工程师

2022年加入去哪儿旅行基础架构-基础平台团队，主要负责测试域工具的研发。参与过自动化测试、联调平台、写压测等项目，对录制回放场景有深入了解。

熟悉AI大模型通识，23年开始主攻AI大模型应用方向，完成AI在测试域、需求域提效的应用落地。在去哪儿AIGC HACKATHON大赛获得冠军。

# 目录
# CONTENTS

# PART 01
# 背景

# 全流程结合AIGC提效



AI代码可视化

通过AI生成代码流程图

AI生成Checklist

AI生成测试点列表


AI IDE插件

AI chat

代码生成，代码CR

AIOPS

根因分析

工单自动化

# 现有痛点



## 需求沟通效率低

PM/DEV/QA 三方沟通
平均耗时30min-1h

## 自测自发不写case

开发自测可能不充分
机票自测自发比例 86%


## 写checklist耗时

平均耗时：
5pd以下需求1-2h
5pd以上需求3-5h

## 需求文档质量参差不齐

逻辑清晰，沟通效率高
逻辑混乱，沟通效率低
无评估标准，只能凭感觉

# 用大模型生成checklist的好处



**提升QA写checklist的效率**

from 写作业
to 批改作业

**提升自测自发需求质量**

from 不写case
to 自动生成

**可以检查需求文档的质量**

质量好：checklist可接受程度高
质量差：checklist可接受程度低

# 设计关键点



01 准确度提升

02 覆盖度推广

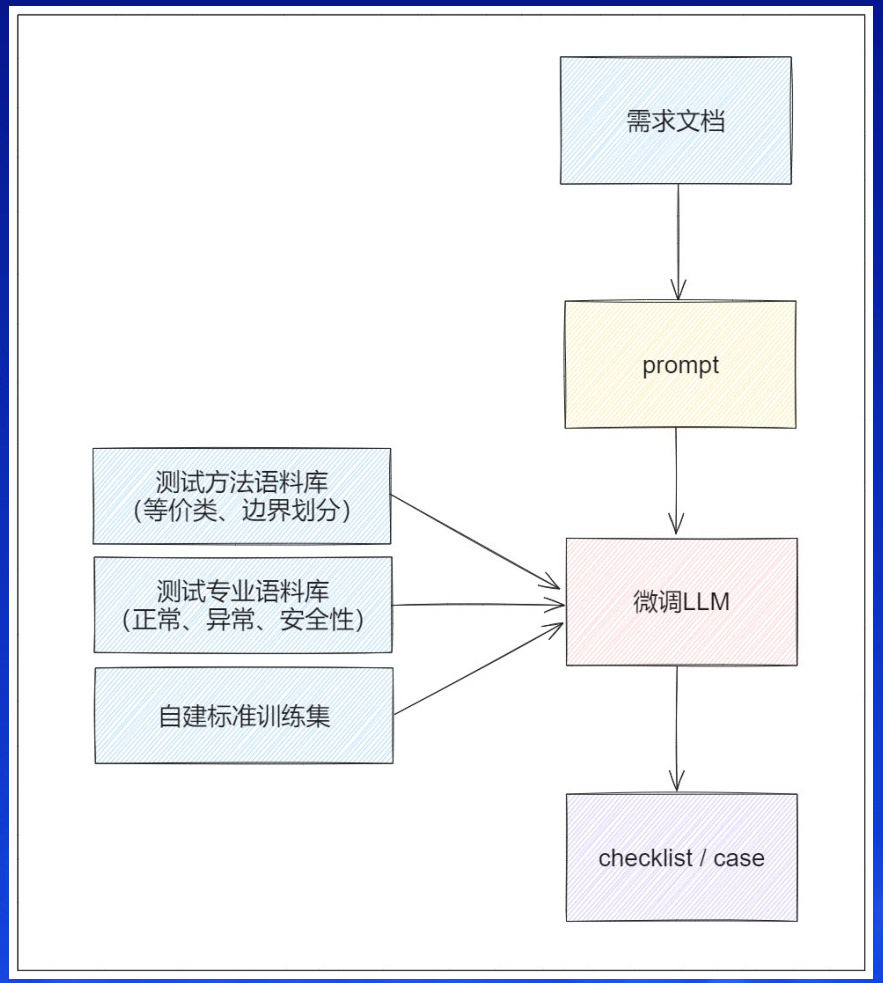
03 效果度量方案

# PART 02
# 设计思路和方案

# 业界参考样例

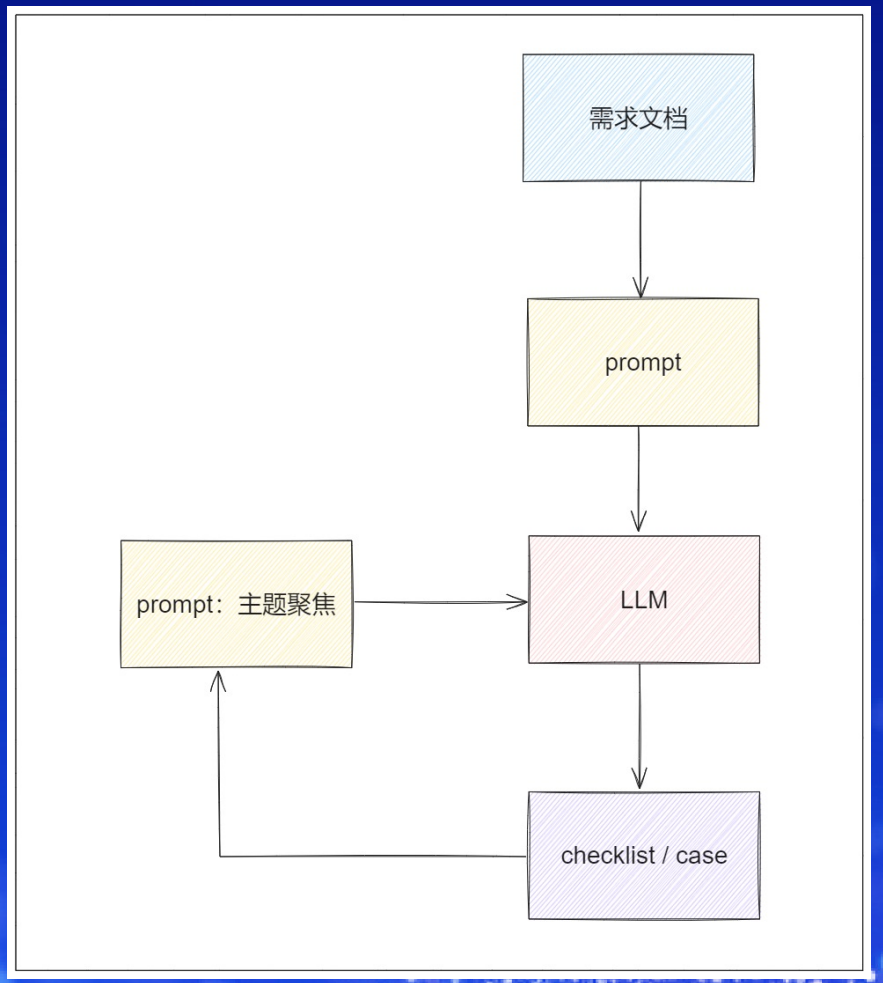


## 基于自有大模型及微调的一键生成方式




门槛高、成本高、需要数据积累

## 基于通用大模型多轮问答的方式

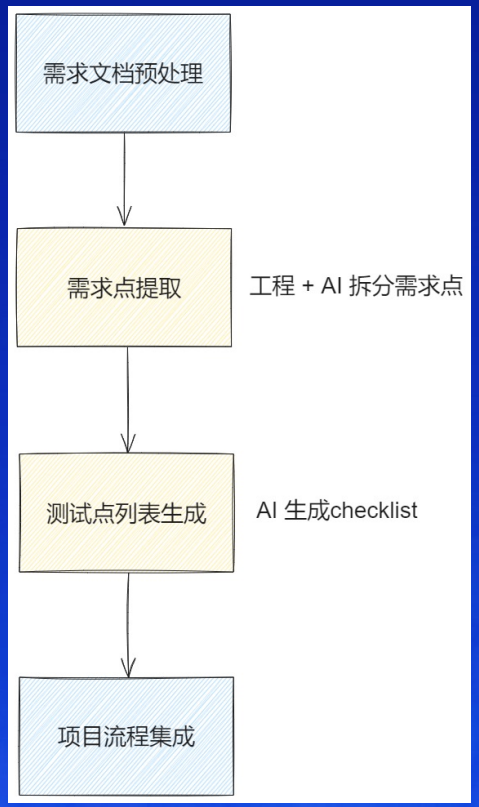



有迁移、整理成本

# Qunar探索的解决方案



主要（便捷）：基于通用大模型+聚焦于需求点+集成项目研发流程中的一键生成方式




辅助（灵活）：智能体多轮问答

# 系统架构

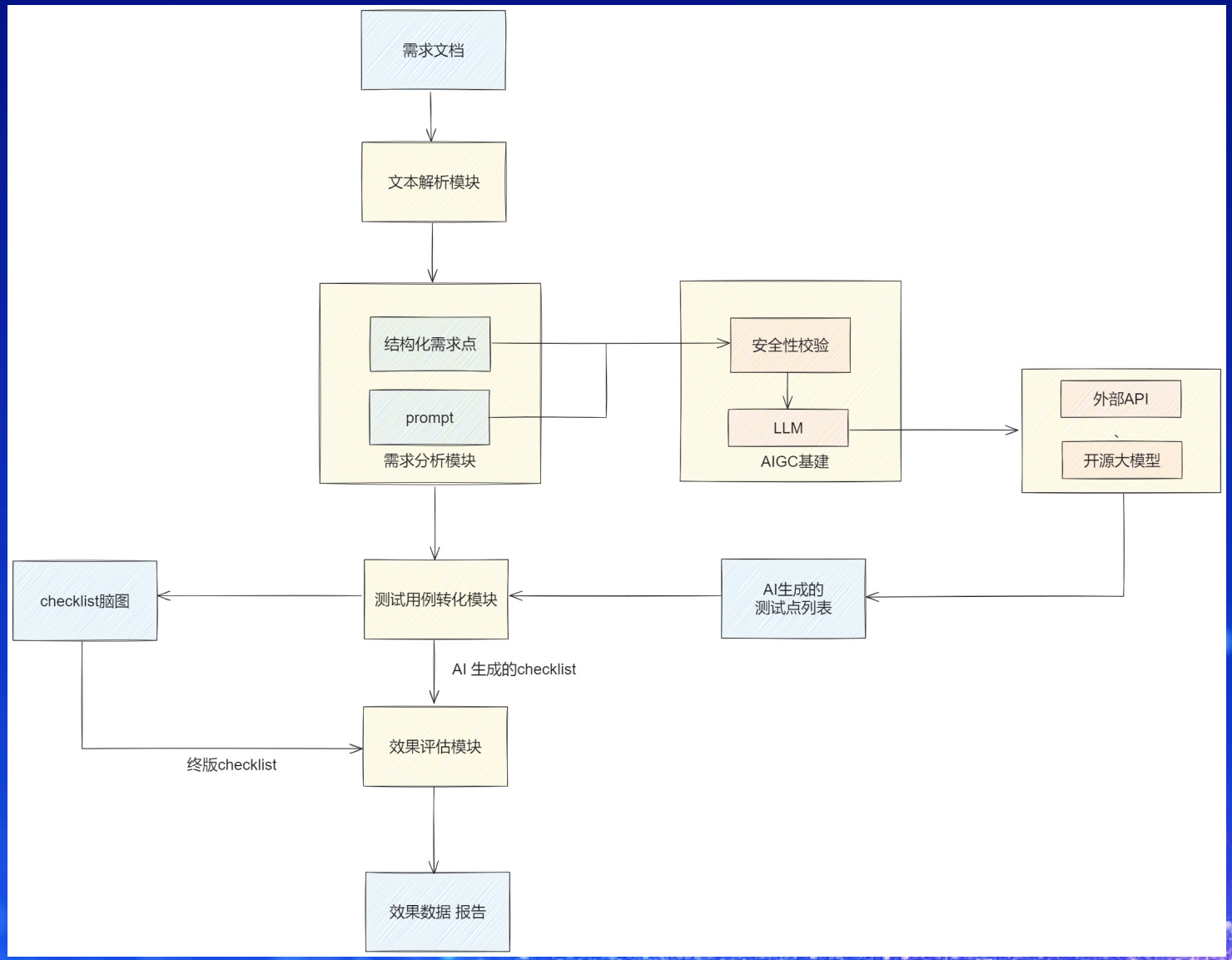



用户交互层
Atlassian JIRA
项目进度管理
PMO
项目流程工具管理
confluence
checklist
case平台
项目流程检查
AI智能体平台
系统功能层
需求文档预处理
需求点拆分
Prompt
checklist生成
思维导图转换
case转换
效果评估
基础组件
chatgpt api
安全脱敏
Embedding
定时Job
数据存储

# 执行流程



1. 获取需求文档，对文本进行拆分
2. 结构化文档+prompt向大模型提问
3. 将AI返回的测试点进行转换，渲染为脑图
4. 归档AI生成与手动修改完的case，分析数据

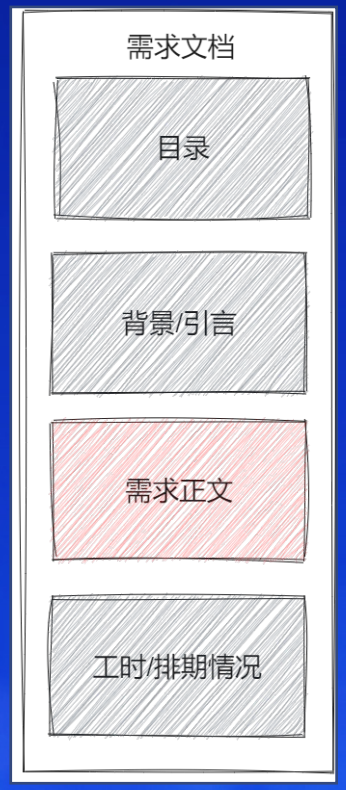


# 产品文档预处理



## 原因分析：

- 需求文档无固定模板，规范性较低
- 需求文档中的无关内容影响生成效果




【需求正文】二级标题

**改动范围**

正文

**需求点1：三级标题**

正文

**需求点2：三级标题**

正文

**需求点3：三级标题**

正文

## 解决方案：

- 需求文档预处理，提取需求正文
- 拆分需求正文，获得结构化需求点

【需求正文】
改动范围
登录页面

需求点1：前端
增加用户名和密码的输入框。
用户名限定格式为大小写字母+数字。
密码最少为6位，格式为大小写字母+数字，并且前端展示为加密字符。
用户名或密码格式不符合的情况提示用户"用户名或密码格式不正确"。

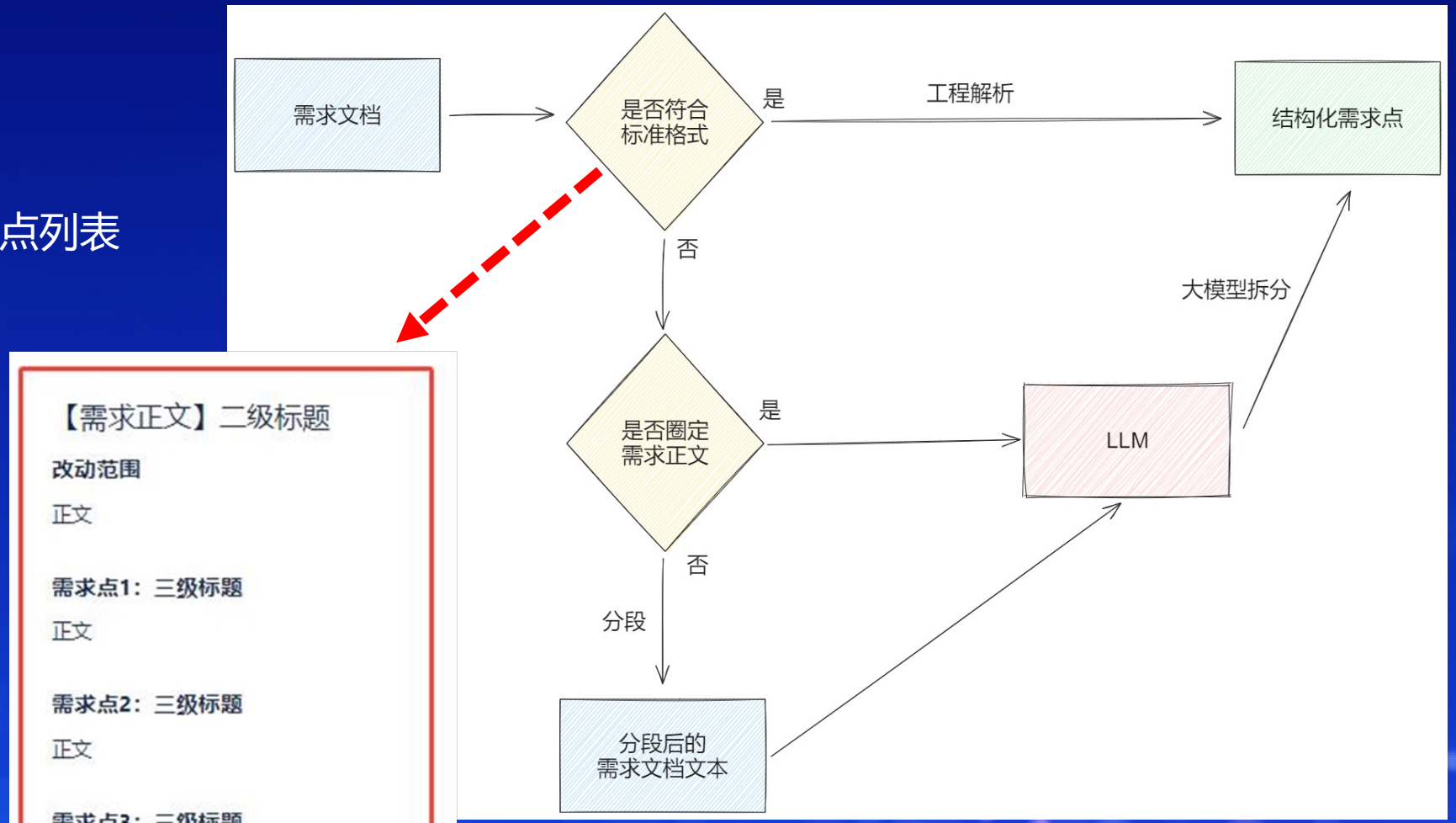
需求点2：后端
用户名入库时校验，格式为大小写字母+数字。
密码入库时校验，格式为大小写字母+数字，最少6位，入库后需要加密。
用户名或密码格式不符合的情况向前端返回"用户名或密码格式不正确"

# 文本解析模块



## 符合标准格式：

- 工程化解析需求正文及需求点列表
- 生成准确率高




【需求正文】二级标题

**改动范围**

正文

**需求点1：三级标题**

正文

**需求点2：三级标题**

正文

**需求点3：三级标题**

正文

## 问题：

- 对产品角色要求变高
- QA角色获益
- 推进受阻

# 文本解析模块



## 圈定需求正文：

- 大模型解析需求点列表
- 预处理成本低
- 生成准确率高

**1.背景**

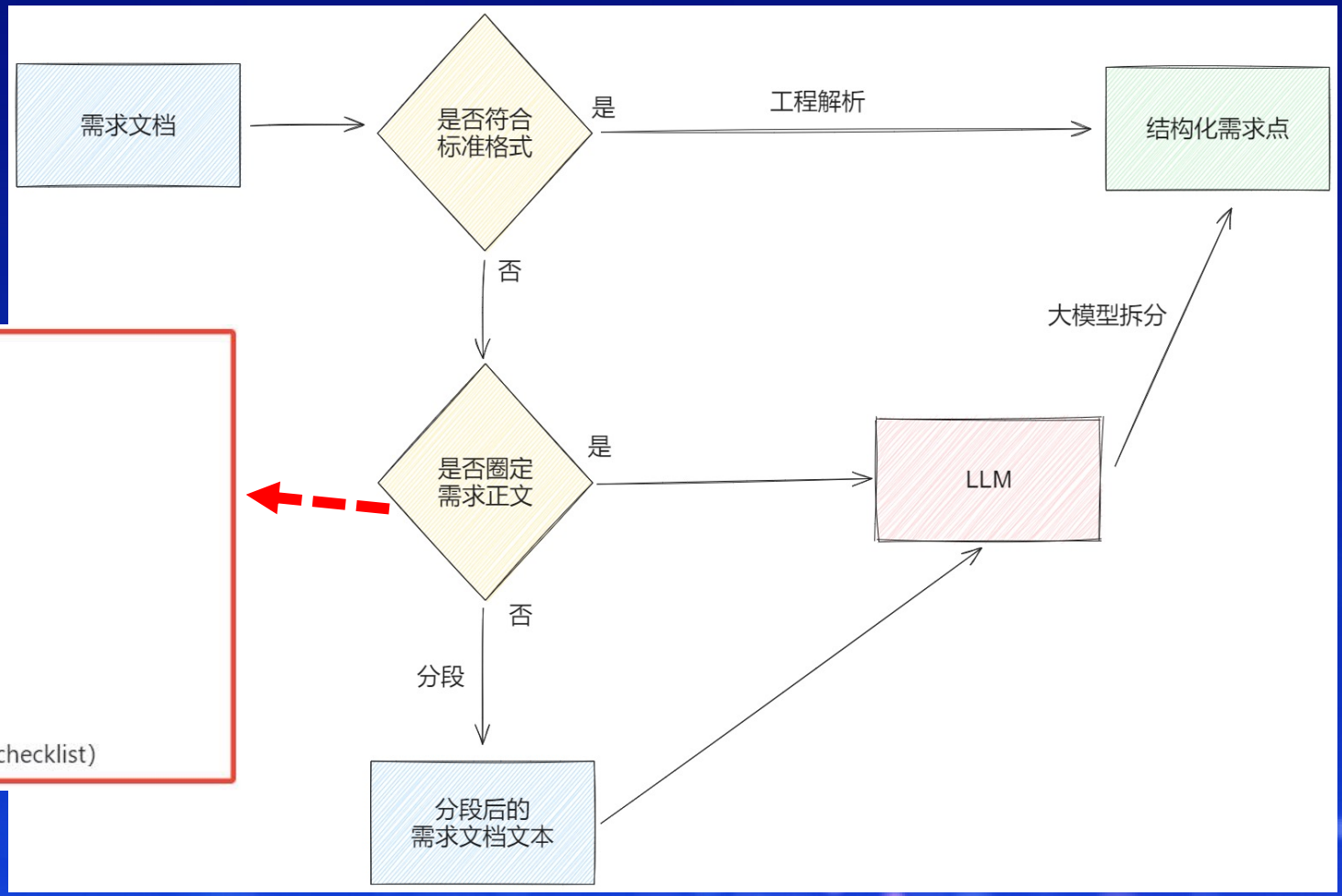
xxxx（忽略）

**2.工时**

xxxx（忽略）

**3.需求方案**

xxxx（根据此部分去生成checklist）

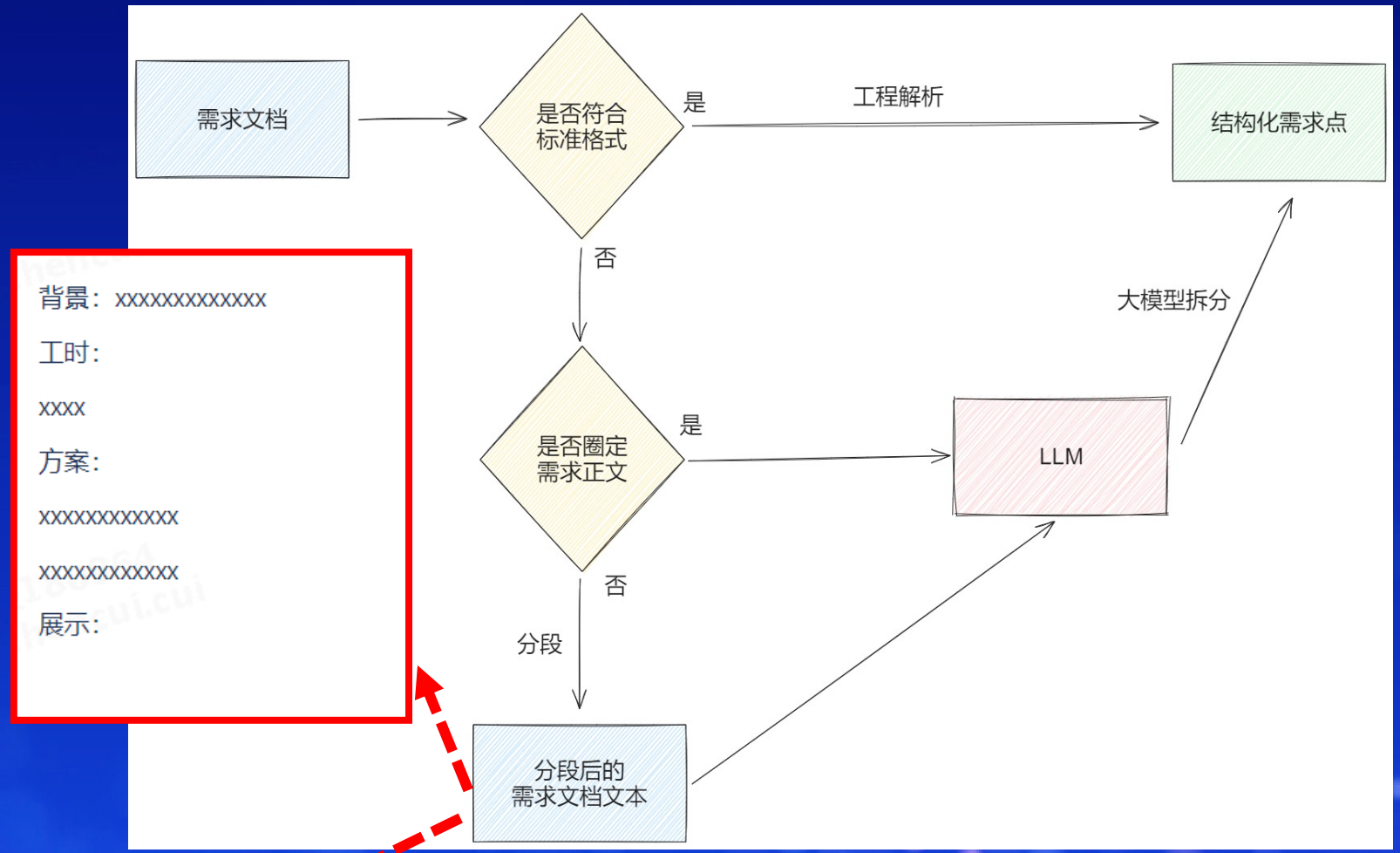


# 文本解析模块



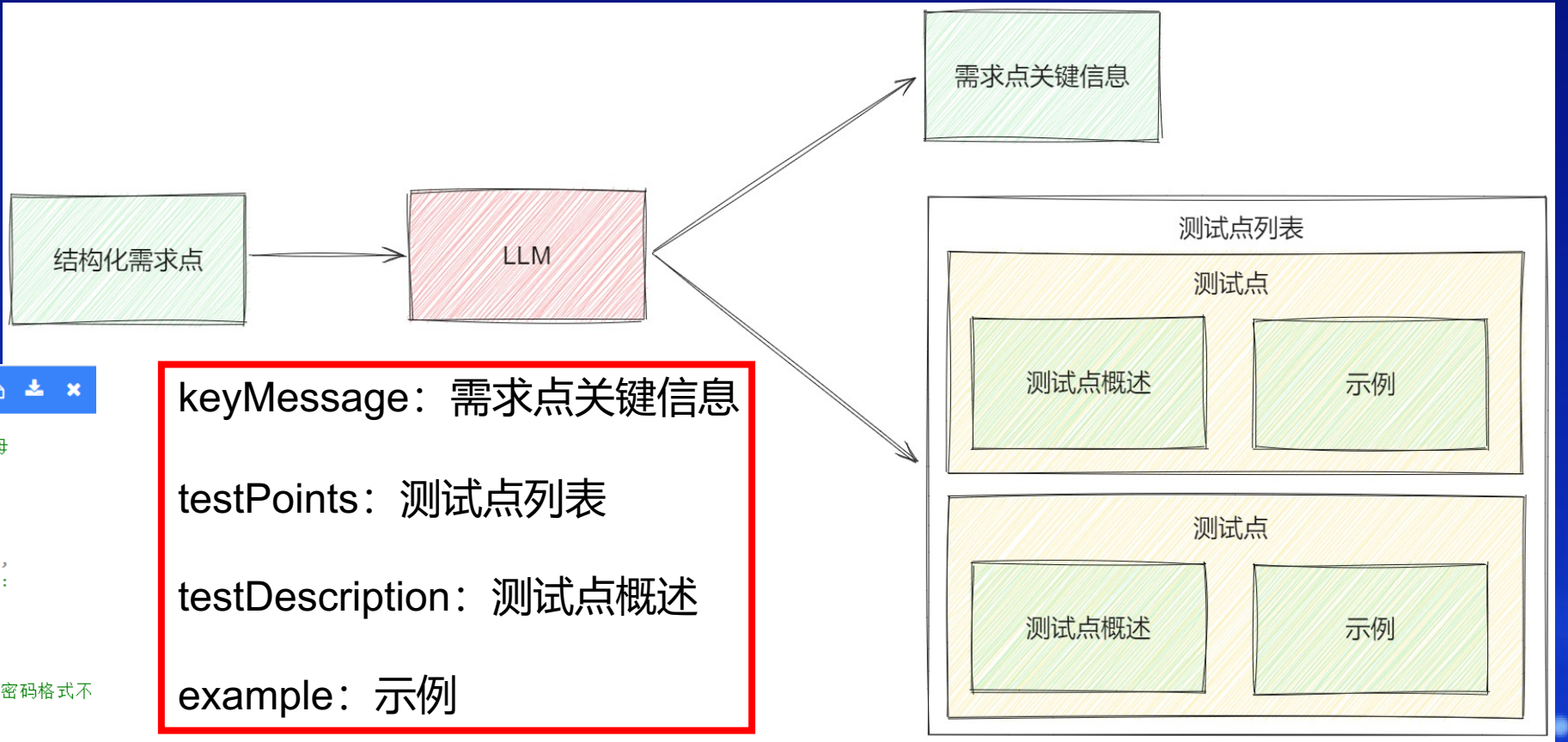
## 完全无格式：

- 大模型解析需求正文
- 大模型解析需求点列表
- 生成准确率取决于文档内容密度



# 需求分析模块






keyMessage：需求点关键信息

testPoints：测试点列表

testDescription：测试点概述

example：示例

reply sample

```
{
  "keyMessage": "用户名和密码格式校验，用户名为大小写字母+数字，密码为大小写字母+数字且最少6位",
  "testPoints": [
    {
      "testDescription": "输入符合要求的用户名和密码，入库成功并加密存储",
      "example": "入参：用户名:abc123，密码:Aa1234，输出：入库成功"
    },
    {
      "testDescription": "输入不符合要求的用户名和密码，提示用户“用户名或密码格式不正确”",
      "example": "入参：用户名:@#$%，密码:a1b2c3，输出：用户名或密码格式不正确"
    },
    {
      "testDescription": "输入长度小于6位的密码，提示用户“用户名或密码格式不正确”并拒绝入库操作",
      "example": "入参：用户名：abcd1234，密码：abc，输出：拒绝入库操作"
    }
  ]
}
```

# 需求分析模块——prompt设计



## 角色

测试专家

&

语言分析专家

## 能力

1，提取需求点文本的关键信息

2，给出测试点列表和示例

## 样例

通过history模拟

Few shot的方式来提升回答的准确率

## 规则

1. 关键信息只保留一句精炼的概括信息
2. 每个测试点对应一个示例
3. 以JSON格式输出

# 需求分析模块——LLM选择



## AI基建情况

**外部大模型**

- 安全审核
- 接口统一化

**内部大模型**

- 小参数的开源大模型

**微调**

- 机器显卡有限
- 缺少标准数据集

| 指标\LLM | GPT-4-turbo | GPT-3.5 | chatGLM3-6B |
| --- | --- | --- | --- |
| 采纳率<br>（采纳case数 /生成case数） | 60%-70% | 50%-60% | 30%-40% |
| 召回率<br>（采纳case数 / 终版case数） | 30%-40% | 25%-30% | 20%-30% |
| 成本 | 0.0100$ /<br>1K tokens | 0.0015$ /<br>1K tokens | 0 |

# 项目流程集成



## 触发方式

- 项目管理流程入口触发

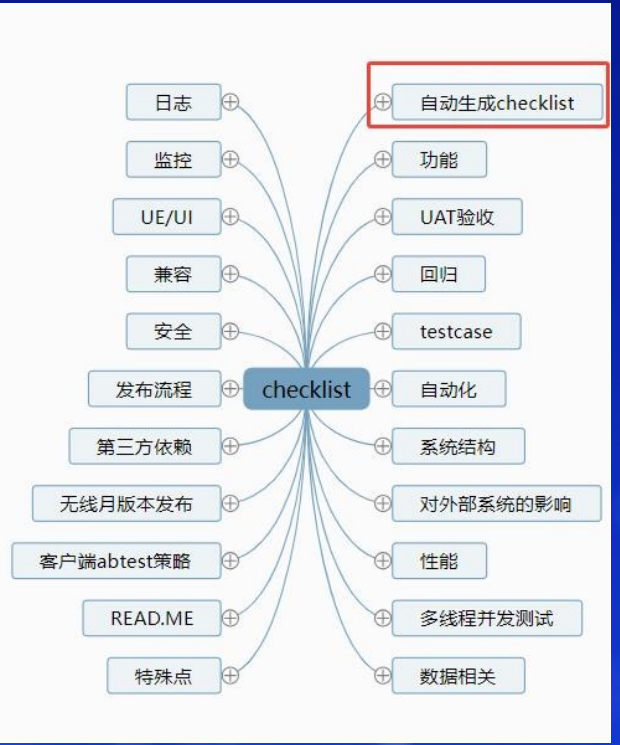

- checklist平台内手动触发


- 定时扫描第二天进入开发中的需求，触发自动生成checklist

qschedule

用户无感知
零成本接入

## 融入原本的通用case模板



# 样例展示



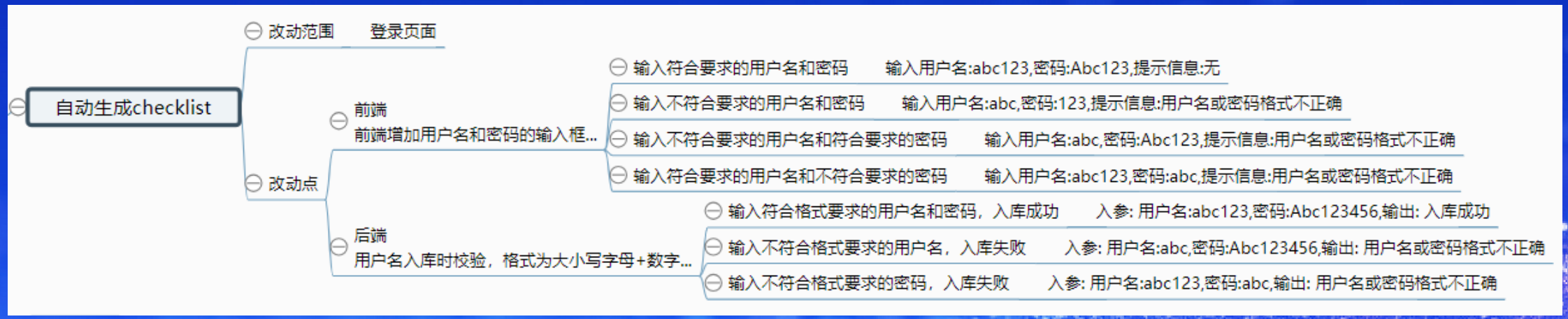
【需求正文】
改动范围
登录页面

需求点1：前端
增加用户名和密码的输入框。
用户名限定格式为大小写字母+数字。
密码最少为6位，格式为大小写字母+数字，并且前端展示为加密字符。
用户名或密码格式不符合的情况提示用户"用户名或密码格式不正确"。

需求点2：后端
用户名入库时校验，格式为大小写字母+数字。
密码入库时校验，格式为大小写字母+数字，最少6位，入库后需要加密。
用户名或密码格式不符合的情况向前端返回"用户名或密码格式不正确"

reply　sample

```
{
  "keyMessage": "用户名和密码格式校验，用户名为大小写字母+数字，密码为大小写字母+数字且最少6位",
  "testPoints": [
    {
      "testDescription": "输入符合要求的用户名和密码，入库成功并加密存储",
      "example": "入参：用户名:abc123，密码:Aa1234，输出：入库成功"
    },
    {
      "testDescription": "输入不符合要求的用户名和密码，提示用户"用户名或密码格式不正确"",
      "example": "入参：用户名:@#$%，密码:a1b2c3，输出：用户名或密码格式不正确"
    },
    {
      "testDescription": "输入长度小于6位的密码，提示用户"用户名或密码格式不正确"并拒绝入库操作",
      "example": "入参：用户名：abcd1234，密码：abc，输出：拒绝入库操作"
    }
  ]
}
```



# 自测自发case前后对比



## 使用AI Checklist前



## 使用AI Checklist后

# PART 03
# 效果评估方案

# 效果评估模块




## 覆盖率

按照项目维度统计用户使用情况

项目覆盖率：使用的项目数/全部项目数

T | 完全可采纳节点数

P | 部分可采纳节点数

F | 完全不可采纳节点数

## 采纳率

原始生成结果中用户选取自动生成节点的概率

采纳率：(T+0.5*P)/A0

A0 | 自动生成总节点数

A1 | 终版checklist节点数

## 召回率

用户进行修改之后采用的自动生成节点与总结点数的比率

召回率：(T+0.5*P)/A1

# 效果评估模块

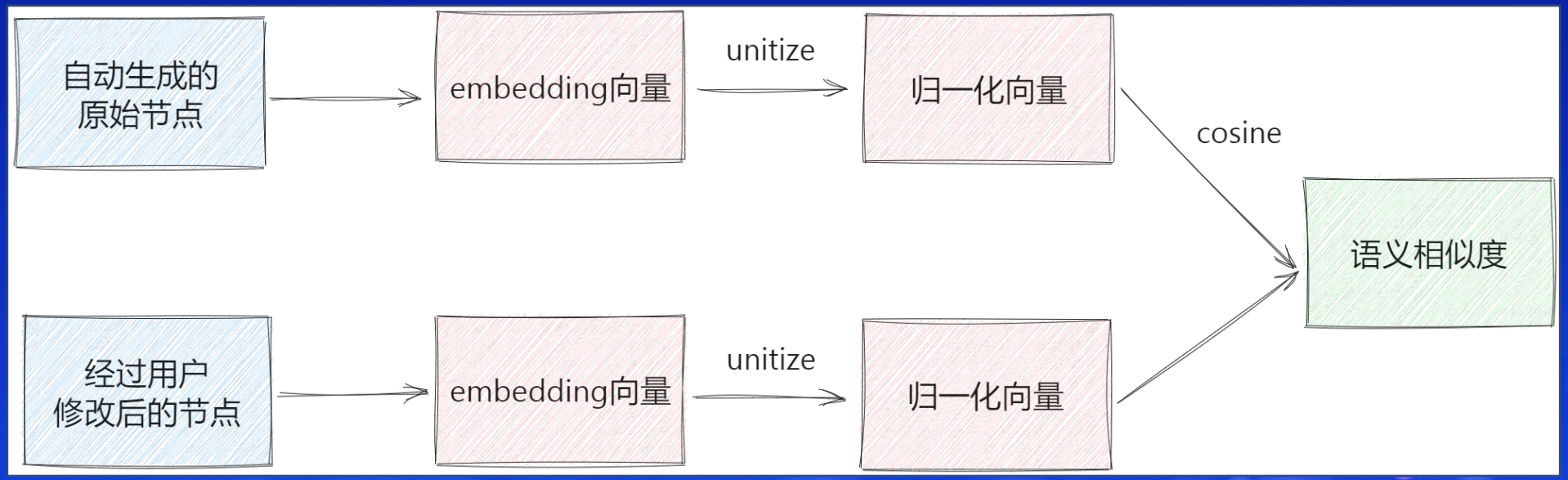


| 方案 | 优点 | 缺陷 | 备注 |
| --- | --- | --- | --- |
| 用户点击反馈 | 实现简单，无需另外设计 | 会增加额外的流程，提高用户使用成本。 | • 可能影响项目覆盖率<br>• 可能影响统计效果 |
| 字符串匹配 | 用户无感知，对流程无侵入。只需要有AIGC源数据和用户终版数据即可触发效果评估。 | 传统数学方式的匹配，不符合人修改checklist的使用习惯，统计效果会较差。 | Case1: 密码正确时弹窗<br>Case2: 密码不正确时弹窗<br>字符串匹配：87.5% |
| 基于Embedding模型匹配 | 同上。并且巧妙利用了用户修改checklist的使用习惯，结合embedding向量做语义相似度匹配。 | 前期需要人工评估一些case，建立一套合理的阈值和权值模型。如果使用外部的embedding模型接口会有一些花费。 | Case1: 弹窗不能关闭<br>Case2: 弹窗不可以关闭 |

# 效果评估模块



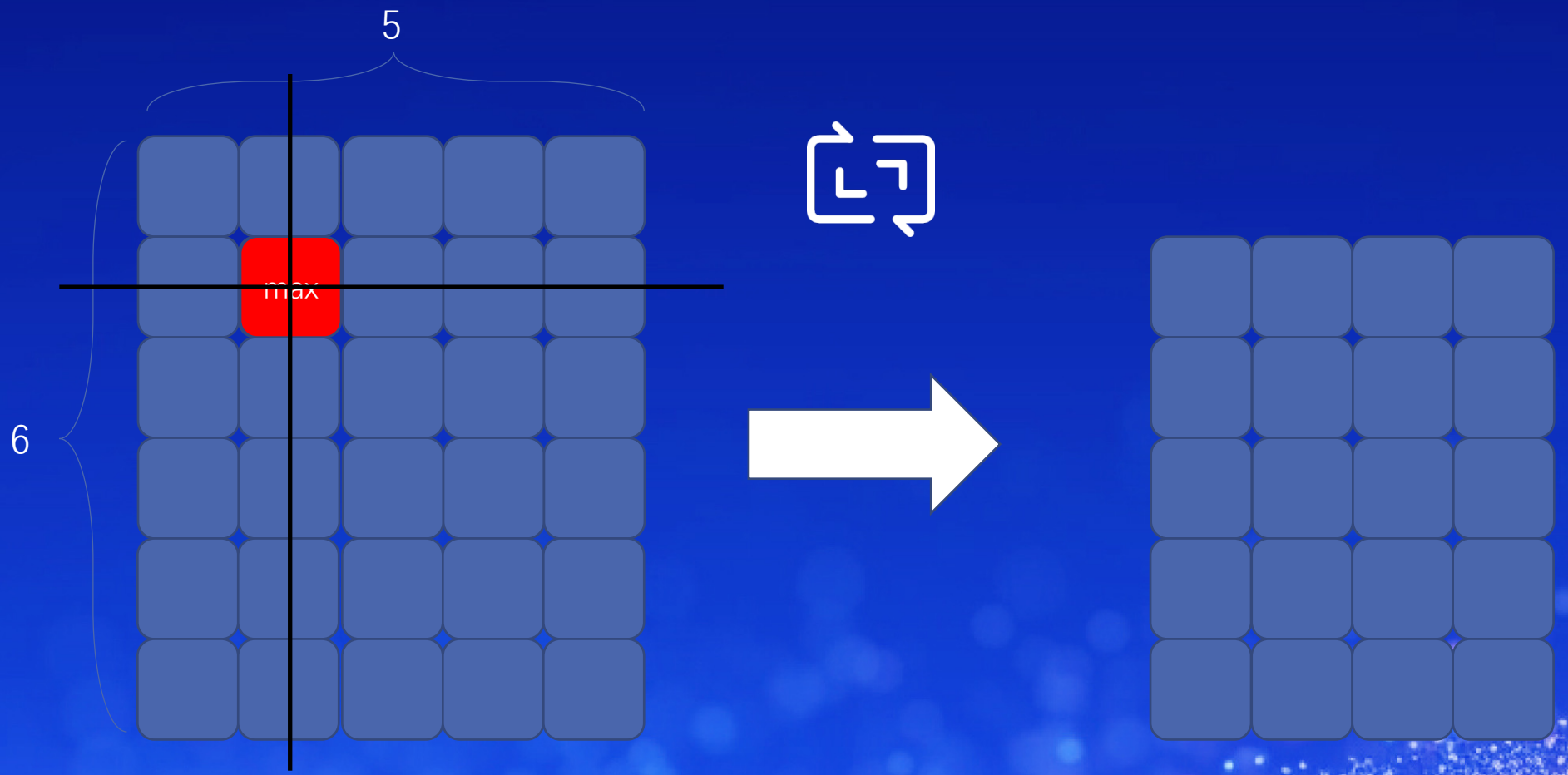
中文文本embedding模型是一种将文本转换为向量表示的技术，它能够捕捉文本的语义和语法信息，并将其转换为连续的向量空间中的点。这种表示方式在自然语言处理领域被广泛应用于各种任务，如文本分类、情感分析、命名实体识别等。



# 效果评估模块



自动生成节点数：5
终版checklist节点数：6



# 效果评估模块



| 相似度阈值 | 含义 | 接受率权值 |
| --- | --- | --- |
| 98% | (T) 完全可采纳 | 100% |
| 90% | (P) 部分可采纳 | 50% |

**A0** 自动生成总节点数 = 5

**A1** 终版checklist节点数 = 6


采纳率：(2*100%+1*50%)/5 = 50.0%

召回率：(2*100%+1*50%)/6 = 41.7%

# PART 04
# 成果及未来计划

# 目前效果



## 准确率

60%-70%

采纳率∝

需求文档逻辑清晰程度

## 召回率

30%-40%

召回率∝

需求文档需求点拆分细致程度

## 落地范围

每月500+个项目使用

产品需求覆盖率60%-70%

## 提效成果

- 5pd及以下需求，每个需求节省0.1pd
- 5pd以上需求，每个需求节省0.2pd
- 年化可节省约200pd
- 填补自测自发不写checklist的缺口

# 未来计划



## 01 内部大模型微调

涉及核心私密数据的需求，可以走内部大模型生成

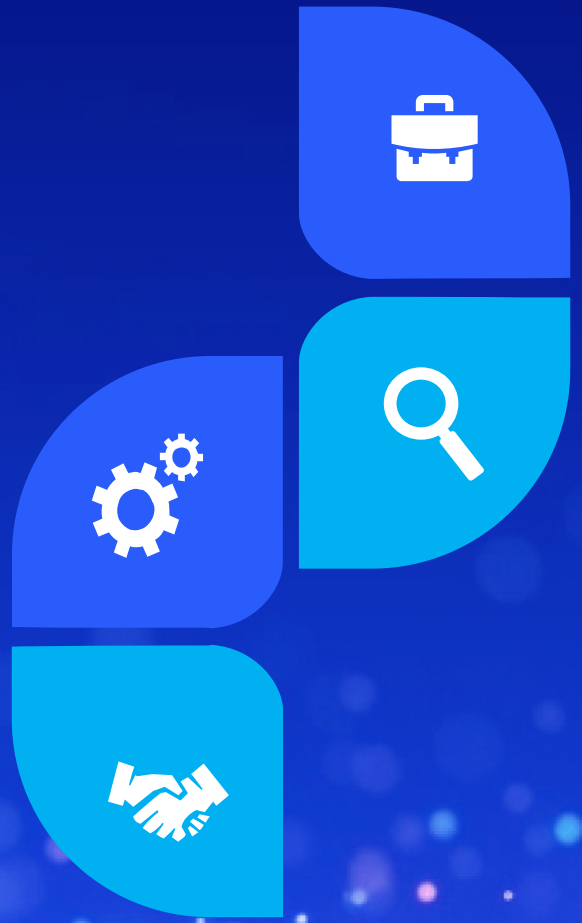
## 02 接入内部知识库

业务知识库：公司内部概念，黑话，历史资料等

技术知识库：系统调用关系，业务代码资料等

## 03 结合多模态

支持解析PRD中存在的流程图、UI图信息



# THANKS