以AISE工具链支持金融软件——技术栈翻新

孙洪军 | 软通动力金融事业群CTO

图 3 AI 赋能软件工程各阶段提效年度平均数据对比

- GenAI工具在软件工程似乎遇到了瓶颈天花板
- 下一步突破的方向在哪儿
- 金融行业大型复杂项目的软件工程问题

金融行业老旧IT系统重构的项目困境

特征	老旧系统架构 (Legacy)	新技术架构 (Modern/Target)
架构风格	单体/垂直 (烟囱式)	**微服务**
基础设施	专有主机/小型机，物理服务器，早期虚拟化	**云原生** (容器/K8s)，**混合/多云**，**无服务器**
数据库	集中式 RDBMS，主备	**分布式数据库** (NewSQL, NoSQL, 云原生DB)
集成方式	点对点，**ESB (紧耦合)**，文件交换，DB直连	**API网关**，**事件驱动** (Kafka等)，松耦合
数据处理	**批处理主导** (T+1)	**实时/流处理** (Flink等)，**湖仓一体**
开发模式	瀑布式，开发运维分离	**DevOps**，**持续交付** (CI/CD)，**IaC**
技术栈	COBOL, J2EE,JSP, .NET FX, 重量级中间件	**Spring Boot/Cloud**, Go, Vue.JavCore, **轻量级**
可扩展性	**Scale Up为主**，扩展性差	**弹性伸缩** (Scale Out)，高扩展性
高可用	主备切换 (时间长)	**多活数据中心**，**容错设计**，**混沌工程**
安全性	边界防护为主，内部相对可信	**零信任模型**，**微服务安全**，**全链路加密**
部署运维	手动部署，配置复杂	**自动化部署**，**服务网格** (可观测性)

建设时间长　技术栈过时　无人能完整掌握系统　源代码对应准确文档缺失严重　……

核心诉求：新系统保持和原有系统功能一致

01. 纯人工打造

02. 借助炸药打造

03. 工程化盾构机

传统开发模式 → AI手搓模式 → AI4SE工具链

对比角度	传统开发模式	AI手搓模式	AI4SE自动迁移
核心逻辑保障	⭐ ⭐ ⭐ ⭐ ⭐ (人工深度还原)	⭐ ⭐ ⭐ ⭐ (人机双重校验)	⭐ ⭐ ⭐ (依赖AI可靠性)
实施速度	⭐ 周期长	⭐ ⭐ ⭐ 开发周期有所缩短	⭐ ⭐ ⭐ ⭐ ⭐ 效率提升？倍
成本效益	预算大	成本有所控制	⭐ ⭐ ⭐ ⭐ (长期ROI高)
技术先进性	⭐ ⭐ ⭐ ⭐ (全新架构)	⭐ ⭐ (可能继承技术债)	⭐ ⭐ ⭐ ⭐ ⭐ (AI+)
风险控制	⭐ ⭐ ⭐ ⭐ (可审计性强)	⭐ ⭐ ⭐ (局部不可控)	⭐ ⭐ (解决黑盒风险)
……	……	……	……

全栈智能，软通动力AI产品及服务架构

AI 咨询引领

顶层设计驱动商业价值转化

AI 技术栈工具

全栈式AI开发赋能平台

AI 工程服务

端到端服务AI应用落地

AI 基础设施

弹性智算可信服务体系

AI 咨询服务

AI战略规划 | AI就绪度评估 | AI场景设计 | AI能力建设规划 | AI实施路径规划 | AI试点验证 | AI规模化应用 | AI监控

行业应用产品

天枢数字孪生平台 | 天坊工业互联平台

终端智能产品

天汇复合机器人 | 天鹤、天锡、天擎人形机器人 | AIPC

天璇MaaS平台

SFT 模型微调 模型评测 模型发布

天璇数据平台

模型数据管理 数据标注 数据资产管理

天璇智能体平台

智能体管理 智能体应用开发

天璇知识库平台

RAG知识库管理 知识检索

模型工程

模型开发 | 模型优化 | 模型安全 | 模型部署

数据集工程

数据采集 | 数据存储及加工 | 数据治理 | 数据服务

智能体工程

多模态感知 | 推理决策 | 控制执行 | 多体协作

天元智算服务平台

算力资源 | 算力调度 | 算力订阅 | 算力监控 | 算力使用分析 | 运维服务

基础硬件

软通华方 AI训练服务器 | 软通华方 AI工作站 | 软通华方 通用服务器

算力服务

AI一体机 | 训推一体化平台 | 智算中心

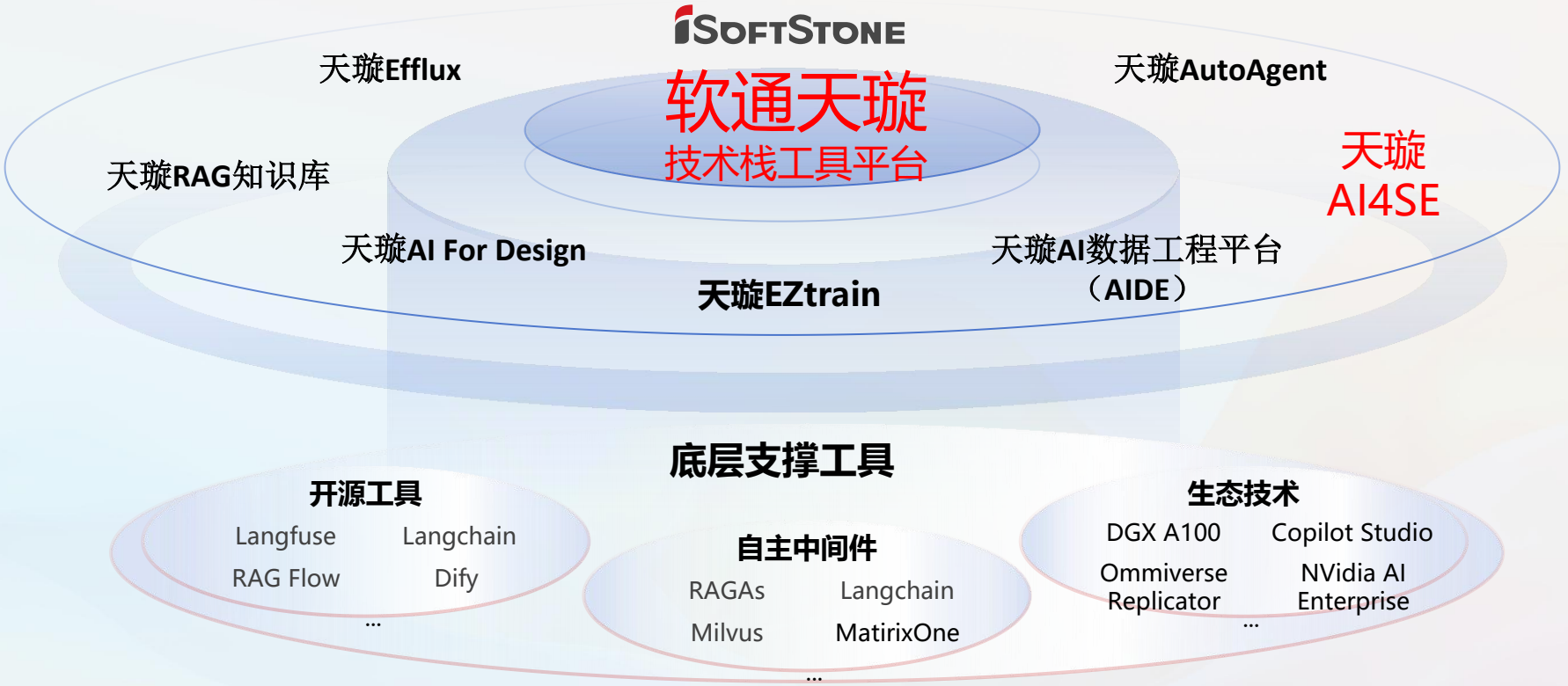
三方平台/模型

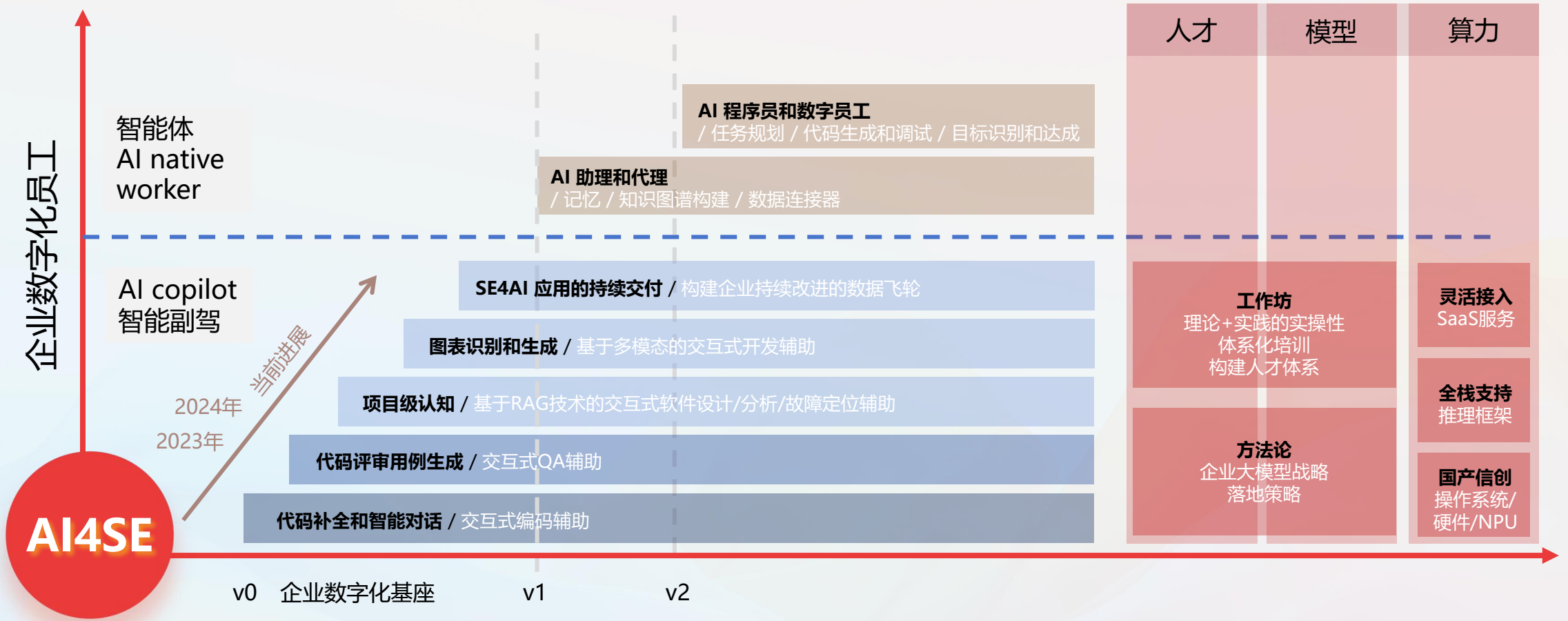
华为云 AI Gallery | 阿里云 通义千问 | 百度云 文心一言 文心千帆 | 腾讯云 腾讯混元 | NVIDIA | Microsoft OpenAI Azure | 科大讯飞 讯飞星火 | 智谱·AI

AI运营及运维

已形成面向模型、智能体、数据及应用“天璇AI工具栈”

- 模型服务平台能力要求框架体系
- AI4SE软件工程工具链
- 可信AI及模型安全体系




企业数字化员工
智能体
AI native
worker
AI copilot
智能副驾
AI4SE
2024年
2023年
当前进展
AI 程序员和数字员工
/ 任务规划 / 代码生成和调试 / 目标识别和达成
AI 助理和代理
/ 记忆 / 知识图谱构建 / 数据连接器
SE4AI 应用的持续交付 / 构建企业持续改进的数据飞轮
图表识别和生成 / 基于多模态的交互式开发辅助
项目级认知 / 基于RAG技术的交互式软件设计/分析/故障定位辅助
代码评审用例生成 / 交互式QA辅助
代码补全和智能对话 / 交互式编码辅助
v0 企业数字化基座
v1
v2
人才
模型
算力
工作坊
理论+实践的实操性
体系化培训
构建人才体系
方法论
企业大模型战略
落地策略
灵活接入
SaaS服务
全栈支持
推理框架
国产信创
操作系统/
硬件/NPU

原架构 =>

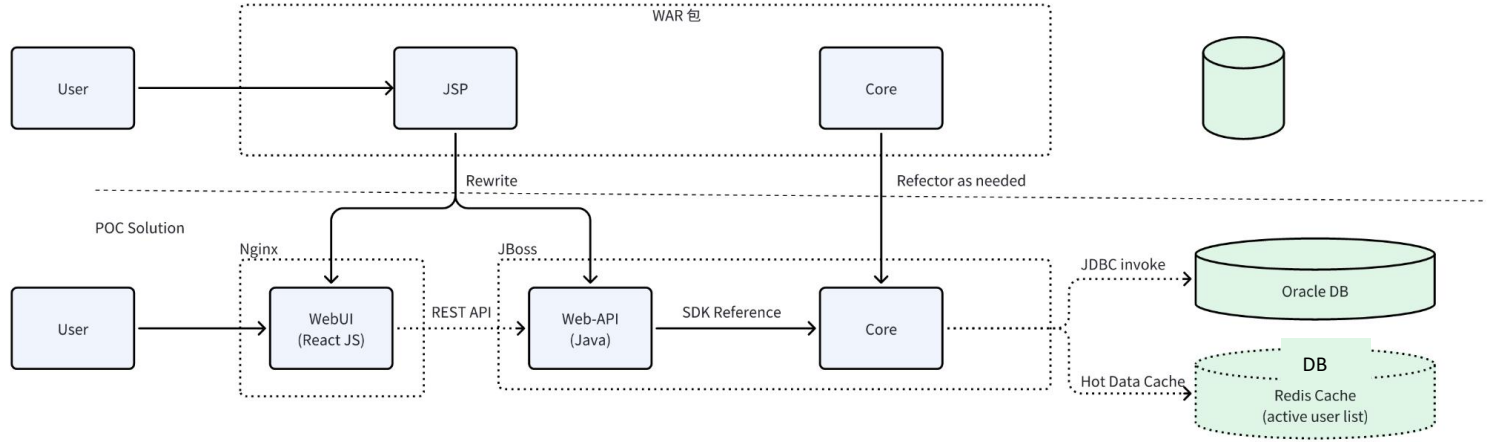
- 前后端一体
- 过时
- 面条代码

AI驱动
Deepseek

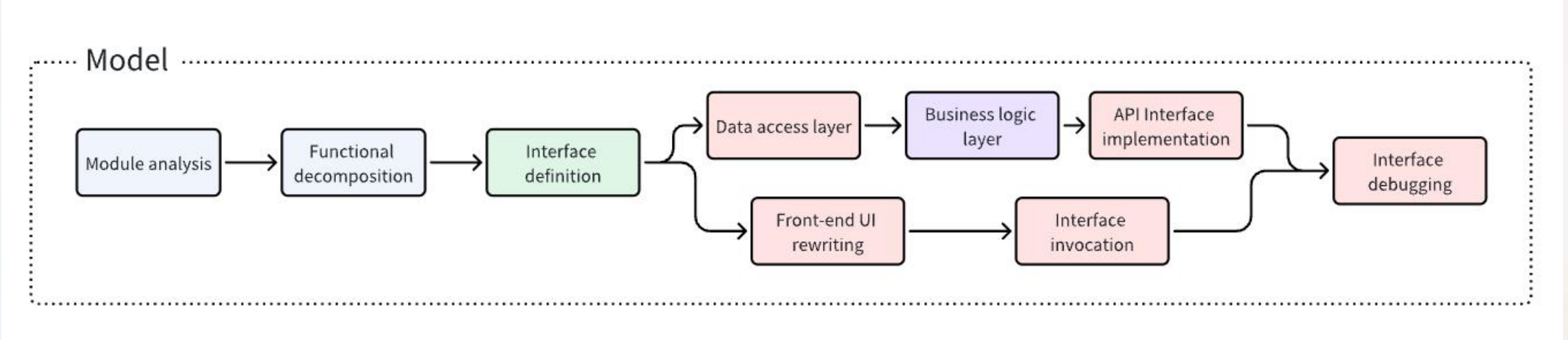
目标架构 =>

- 微服务
- 前后端分离
- 架构清晰

项目核心诉求：保持前端操作和后端业务逻辑不变



模块名称	核心功能	关键特性	集成系统/工具
1. 贷款申请模块	采集申请人数据、贷款类型、金额及支持性文件	多渠道入口（网点/在线/移动/API）、身份核验（KYC）、OCR/eKYC自动填充	CRM、前端门户、文档扫描设备
2. 信用评分与风险评估模块	评估申请人信用资质	对接征信机构（如Equifax）、内部评分卡、风险分类	第三方数据商、ML/AI评分引擎
3. 文档管理模块	管理贷款文件及验证流程	按类型生成清单、文档上传/索引/版本控制、电子签名	DMS、eKYC、电子签名服务商
4. 授信审批模块	自动化或辅助人工审批决策	规则驱动决策引擎、政策检查与人工覆写、多级审批路由	规则引擎（如Drools）、AI/ML预测性审批
5. 反欺诈与合规模块	确保合规并识别可疑行为	AML/OFAC/FATCA检查、模式识别、审计追踪	反欺诈系统、国家级数据库、制裁名单
6. 工作流与流程管理模块	协调贷款处理步骤	可配置工作流、SLA监控、角色任务分配	BPM引擎（如Camunda）、工单工具
7. 抵押品管理模块	追踪贷款关联的担保资产	抵押品估值/留置权追踪、保险与重估调度	估值机构、抵押登记机构
8. 贷款决策与批准模块	集中化审批逻辑管理	决策矩阵配置、多层级审批流程	通知系统、审批仪表盘
9. 放款模块	处理资金发放	资金划转（RTGS/NEFT/ACH）、分期放款	核心银行系统（CBS）、支付网关
10. 审计、报表与分析模块	提供可视化、合规与性能追踪	监管/内部报告（MIS）、用户日志、组合分析	BI工具（Power BI/Tableau）、报表引擎
11. 客户沟通模块	管理通知与进度更新	集成短信/邮件/WhatsApp、状态追踪、聊天机器人	实时客服系统、通信API
12. 集成网关/API管理模块	确保系统间无缝通信	API网关管理、安全协议（OAuth2/JWT）、REST/SOAP支持	第三方服务、核心系统接口



模块分析　后端代码转换

功能分解　前端代码转换

接口定义　接口联调

灵活应用AI重构策略和AI重写策略

解决了：

- 代码混合度高：JSP文件中通常混杂了HTML、Java代码（Scriptlet）、JSTL标签和JavaScript，难以清晰分离。
- 接口边界模糊：业务逻辑分散在JSP、Action类和Service层，真正的接口定义不明确。
- 状态管理混乱：Session和Request属性在前后端之间随意传递，缺乏明确的契约。
- 前端依赖后端模板：前端展示逻辑依赖JSP的渲染能力，难以独立开发和测试。
- JSP中前后端逻辑混合，难以分离
- 从无类型JSP到TypeScript的转换
- 多次文件替换失败，路径或内容不匹配
- ……

开发环境及工具

Development Tools

VSCode

IDEA

AISE

AISE CodeBuddy

aimp(GIT)

Jenkins

JUnit

sonarqube

VERACODE

DeepSeek

AI Tools

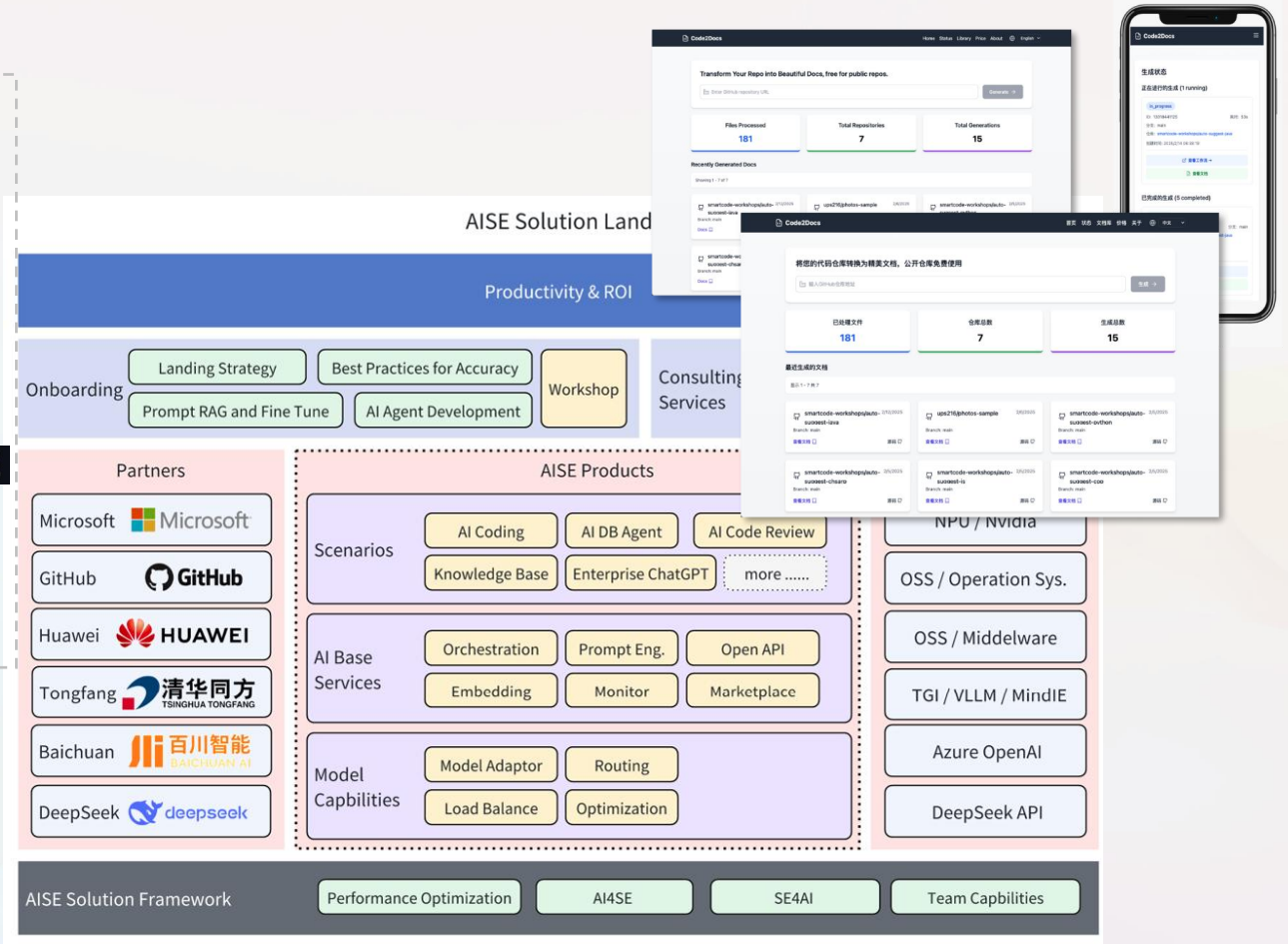
docker

JBoss

NGINX

Deployment

面向技术栈翻新的AI工程化组件及流程化链接



技术栈效率提升结果

项目分析 3倍

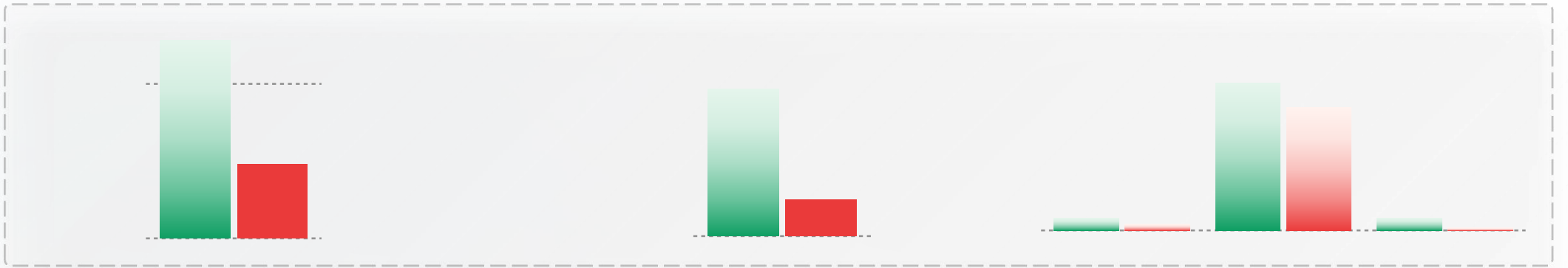
通过软通AI工具自动生成结构化项目文档，并整合AI智能分析工具对模块代码进行功能解析与语义提炼，可高效实现项目的全景式技术解析与精细化模块把控

前端 4倍

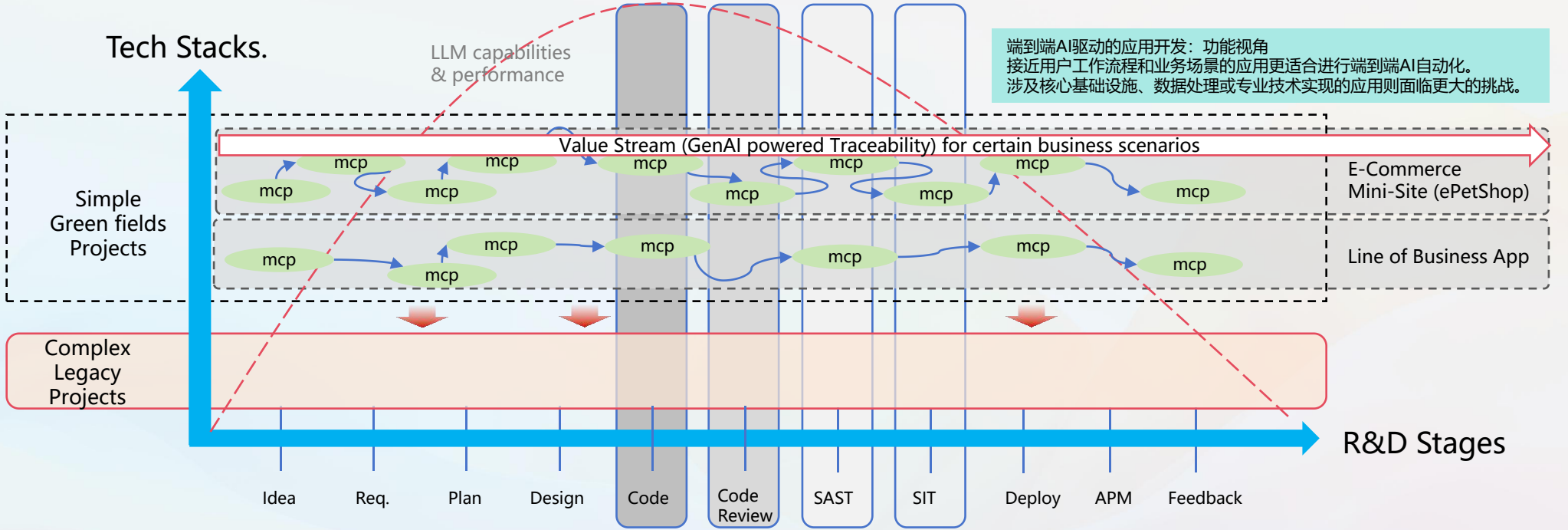
通过 软通AI工具 自动输出符合项目规范的前端UI代码，可显著提升开发效率。降低开发门槛

后端 5倍

数据访问层：通过 软通AI工具 完成生成全部代码，准确率高达95%以上



- 横向整合：逐步在研发阶段实施MCP驱动的代理，以增强认知和执行能力，最终实现端到端自动化。
  纵向整合：专注于高影响力、标准化的领域，如编码、代码审查、静态应用安全测试（SAST）和质量保证（QA）
- 现有的数字研发系统（DevOps平台）将帮助LLM全面了解企业研发工作流程的原子能力
- MCP的兴起建立了一个统一的集成标准来解决NxN集成问题，将现有功能连接起来，形成端到端的生产力



谢谢观看