

# 亚马逊云科技



## 中国峰会

< : < @ @ < = ㊄ >

η

# 驱动规模化代码现代化与技术债清理

史骥

ИПФ

Xmind

深圳市爱思软件技术有限公司

谷雷

Шџ Т ижса уюн ШЖ

亚马逊云科技

陈超

Шџ Т ижса уюн ШЖ

亚马逊云科技

Жм ащнЩса нтео см Ифтуо м  
ЭЖО驱动的代码现代化工具

# 技术债务：企业创新的阻力

为了短期交付速度而在技术方案上做出的妥协。

$< : m$

IT 预算占比

$< : m$  的  $\infty$  预算用于管理技术债务，而非推进新功能。

ЭЛо ссдтудс Чдтда свз# QHВ

$< = m$

开发者时间占比

开发者平均浪费  $< = m$  的时间来处理技术债务。

ЭПфснало еШпгудм т а н Г Шо еуц а сд

$A : m$

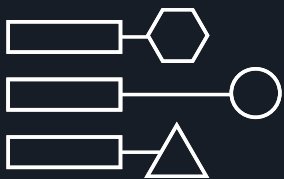
应用改造未达预期

大规模现代化项目未达预期目标

ЭЖввднуфсд

技术债务，为了短期交付速度而在技术方案上做出的妥协

= 本金（过时代码/老旧框架）+ 利息（每次改动的额外成本）+ 违约风险（系统崩溃/安全漏洞/阻碍创新）



## 规模与复杂性

多应用、多语言混合、  
跨系统紧耦合，任何改  
动都可能引发连锁反应



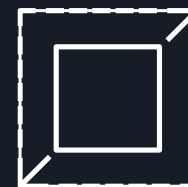
## 负担与惯性

企业优先投资、业务盈利  
而非现代化，直到系统崩  
溃才被迫行动



## 知识黑箱

系统文档不全，业务规则嵌  
入代码中，局限在特定团队  
中，导致跨团队和项目的实  
现不一致和重复工作



## 工具空白

缺乏大规模改造的工具，  
现有工具缺乏大规模应用  
程序的转换需要的规划和  
协调能力

# Жм ащрнЩса нтео см 代码现代化智能体

## 1. 核心能力

**Б ингоц тЛфлЩуавк  
Т огдснщрауион**  
*!УКЦПЩЦСЩсхдс*  
驱动 Б ингоц т工作负载迁移  
解决 Б ингоц т价格挑战

**Ифтуом  
Щса нтео см ауионт**  
*ЖдшЕуоЭКдшТогдл*  
解决技术负债  
加速 Жм ащрн云原生和集集成

**ЫГ ц асд  
Т ижсауион**  
*Assessment + Deployment*  
全面的迁移评估  
一键 ККК 部署

**Т аиңесам д  
Т огдснщрауион**  
*Слжа впуо Тогдсн*  
大型机应用云原生重构

## <! 典型应用

### Б ингоц тЛфлЩуавк 现代化

#### .NET 升级

- 升级到 !УКЦВ Т; : 版本
- 将基于 Б ингоц тЩдсхдс部署转换为 基于 СинфчМсахиуон部署
- 成本节约可达 >:m ð

#### ЩЦСПЩдсхдс现代化

- 将 ЩЦСПЩдсхдс转换为 Жфсо са ТЧЙШ Хо тужсЩЦС
- 关联 !УКЦ应用代码同步做修改
- 成本节约可达 @:m ð
- 关联应用容器化

### Ифтуом 提供 ;= 大内置功能

#### 版本升级

- 内置 Паха, Хшузонь, Уогдй升级
- 内置 Жм ащрнШЙР еосПаха, Хшузонь, Уогдй升级

#### 框架 语言转换

比如: Паха 转换为 Хшузонь, Шп синж全家桶升级

#### Ж驱动的 Мсахиуон转换

- 内置 Паха 应用的 Мсахиуон转换
- 提供转换前评估

#### 其他

内置静态代码分析

## 3. 独特点

- **专业性** 不仅仅是简单的代码替换工具
- 汇聚了亚马逊云科技专业服务团队多年积累的应用现代化经验深度理解 .NET, Java, Python 等生态系统的智能转换引擎
- 经过千锤百炼的实践凝结为内置功能, 开箱即用
- **规模化** 专为企业级规模设计, 能够应对大型组织的现代化挑战
- 支持从单个项目到数千个应用的批量转换
- 并行处理能力, 大幅缩短整体迁移周期
- 统一的转换标准, 确保跨团队、跨项目的一致性
- 将原本需要数月的人工迁移工作压缩到数天
- ИСО驱动的自动化流程, 可集成到 ИОИИ管道
- **连续性** 一个持续演进的平台, 而非一次性工具
- 持续更新的转换规则和增量迁移支持
- 学习记忆模块, 自我迭代优化

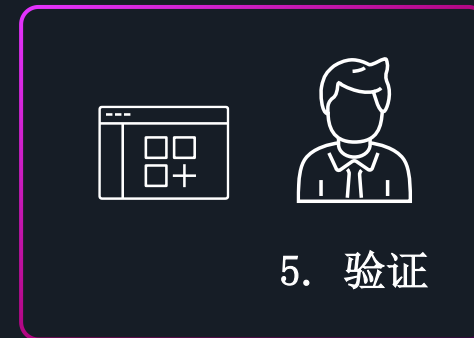
# ЖМ ащрнЩсаңтео см вфтуом 工作流程

集中化团队 (CT)



=! 大规模测试

应用负责人团队 (ЖФ)



应用负责人团队 (ЖФ)



>! 大规模应用

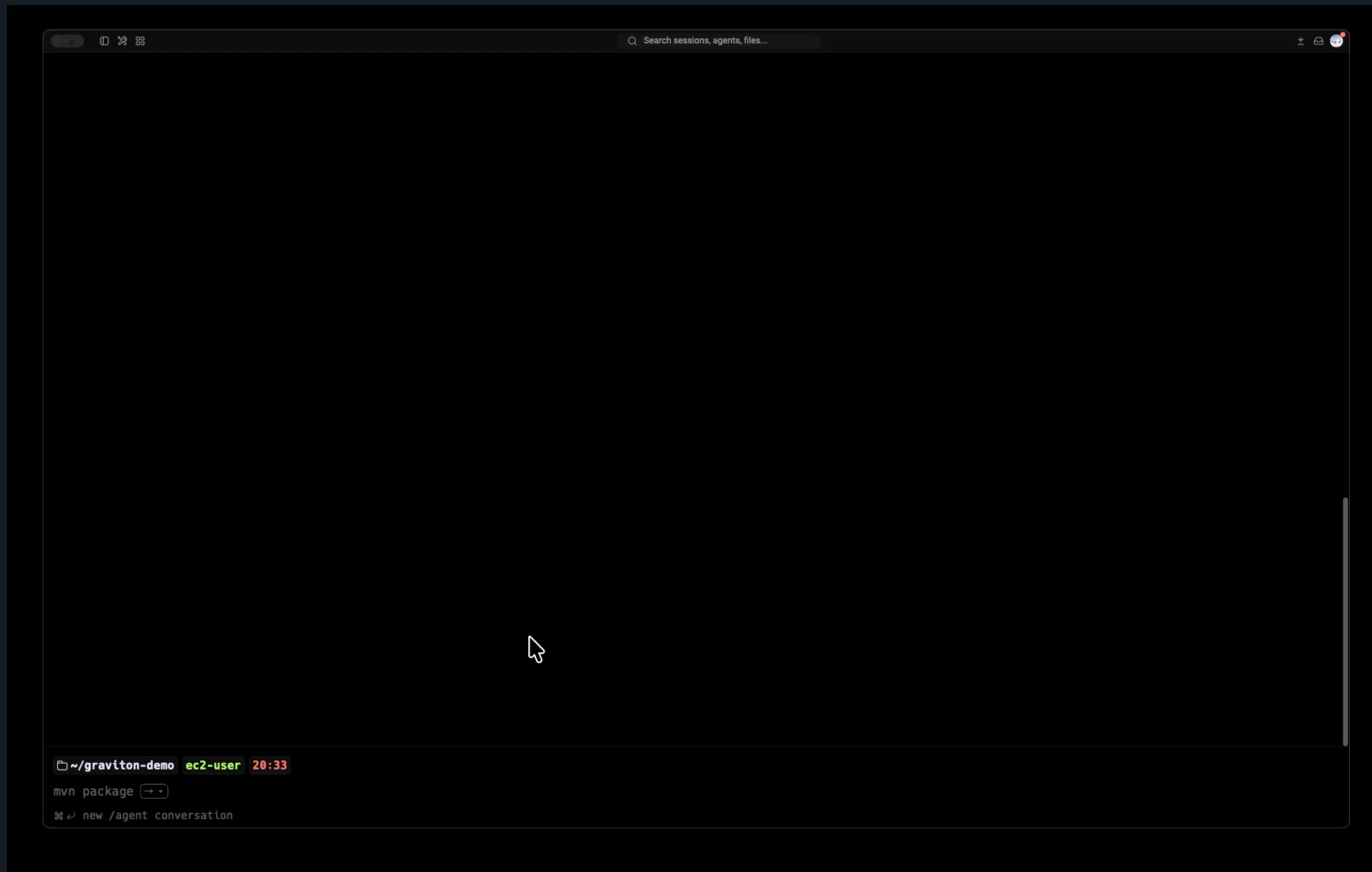
Хфтг# уз софжз тофсвд вонусол пфл# Жпп Фц ндсТ гдх дчдвфудт ин ФКПСО

@! 持续学习





# Практика миграции ИИ-агентов в облако: опыт из практики



# 案例分享

CTO

Xmind.com



## 一款全功能思维导图和头脑风暴软件 思维导图领域的独特先行者

Xmind 提供多种思维导图结构，颜值还高到离谱。秉承高效有趣、专业会玩的理念，我们死磕思维导图的 20 年，每天 965 按时下班喊着放过自己，且常年排名 Mac/iOS App Store 效率榜单前列，多次获得 App Store 推荐。



# 项目背景与迁移挑战

## 客户与场景

ЭМ ИГ 后端业务系统

改造范围”> 个 Хшүз оН< 老项目（总代码量约 ;< 万行），难度中等到较高

当前进展”第一个项目（= 万行代码）已完成改造并上线生产环境，运行稳定

后续计划”第二个应用（ü= 万行）δ 第三个应用（α@ 万行）待改

## 迁移复杂度

+我们以为只是把 Хшүз оН< 的代码跑在 Хшүз оН= 上。结果发现，这是一场涉及语言语义、依赖生态、数据库驱动、加密认证、容器构建的全栈战役。XII

涉及改造点类型：语法、字符串、依赖、加密、数据库、容器、第三方库、环境配置。



# 项目改造阶段 – Definition 定义转换

自然语言描述需求 提供验证命令 提供参考文档

## 经验教训

### ✔ 做对了”

- α提供 提供验证命令
- α先修复 再跑转换
- α选择中等难度项目作为起点

### ⚠ 可以改进”

- α初始未提供验证命令导致首次 遗漏依赖替换
- α未提前提供数据库 文件

## ЭТ ИГ 的做法

- α使用 交互式会话描述转换需求
- α提供项目代码仓库作为
- α指定目标”

提供验证命令

- 💡 先手动修复 再跑转换
- 确保 有验证正确性的手段

# 项目改造阶段 ЭХЛАН 规划转换

ЖЖДНУ自动分析代码 生成转换计划 人工审查计划完整性

## 💡 经验与教训

🐛 依赖替换规划不完整”

α ауч规划 Т< Иашпуо вашпуожсапзш

但只替换了 сдрфисдм днут!пип

未规划代码中 им по су的同步更新

αХЛАН只关注依赖文件

未扫描代码引用链路

🐛 除法 Т ТТ替换有遗漏”

α ауч替换了大部分但个别位置遗漏

α无法通过 жсдп区分

+该换X和 +不该换+

🔑 ХЛАН不是完美的，后续阶段验证补充

## 📄 ЭТ ИГ 的做法

ЖЖДНУ自动生成转换计划”

; !语法层面转换规划

псину псинуЭСТ дчвдпу语法 Т除法

< !依赖替换清单

识别需升级的第三方库及目标版本

= !类型适配策略

бщудтТтус分离方案

> !人工审查

确认计划完整性，识别遗漏

# 项目改造阶段 ЭОМ ПЛДМ ДНУ 执行转换

ЖЖДНУ按计划自动执行代码转换 完成约 80% 的机械性工作

## 💡 经验与教训

🐛 ХИПЪ Ц; !А < В Эсдакинж Из анжд'

α йц ъ днво г дР返回 тус Ёц ат бшудтΣ

α影响 ЖПЛД登录 可能认证崩溃

α ауч升版本号但未更新调用代码

🐛 不必要的 !г дво г дР+фусЕХИ Σ

α КЛДМ ДНУЩсдд уо тусинж本应返回 бшудт

α ауч加了 г дво г д 最终 сдхдсу

🐛 пип ЕЖЛО б а.Ео пуюн已废弃”

α未来 пип升级 Йовкдс构建会失败

🔑 ЖС倾向于 X统一成 тухИИ和 XI升最新ИИ 两者不总是正确的

## 📄 ЭТ ИИГ 的做法

Agent 自动执行 (~80% 工作量):

- `print "x" → print( "x " )`
- `except Exception, e → as e`
- `dict.has_key(k) → k in dict`
- `dict.iteritems() → .items()`
- `/ → //` (整除替换)
- `unicode → str`
- `requirements.pip` 依赖版本升级
- `import` 语句调整

人工: 审查变更, 关注 Breaking, Changes 和业务意图判断

# 项目改造阶段 调试修复

自动循环 人工排查环境差异和运行时间问题

## 经验与教训

陷阱 最阴险

只在特定  
环境变量开启时执行

本地  路径未执行

线上  环境变量存在 崩溃

必须是

影响内部服务鉴权 数字签名

返回值变化

第三方库静默改变返回类型

解决方案

给 提供

按字段类型强制转换

## XMind 的做法

自动循环

执行验证命令 失败 分析

修复 重新验证 !!! 直到通过

能自动修复的

构建失败 错误等

测试失败 有测试用例时

需要人工排查的

环境差异 环境变量

第三方库运行时行为变化

字段类型问题

提供 文件给

# 项目改造阶段 验证确认

四层递进验证 确保转换质量达到生产标准

## 💡 经验与教训

上线试运行后发现的问题”

α 遗漏语法问题 快速修复

α 路径崩溃

需确保验证环境接近生产

α 人工审查 发现

人工审查 发现

+ 没有验证命令的

就像蒙着眼睛开车+

## 📄 XMind 的做法

四层递进验证”

Сашдс; ”自动验证

ГОВКДСВОМ ПОТД ФПО МАКДУТУ

Сашдс< ”手动接口验证 开发者

逐个 调用 业务流程走通

Сашдс= ”团队

重点审查 和依赖升级

Сашдс> ”生产灰度 上线

测试环境 灰度发布 监控

# 项目收获与总结

; 人T; 周

总人工

δ<?

总费用

= 万行

代码量

αB:m

Жждн自动

α<:m

人工判断

生产

已上线

## 🎯 核心收获

; !Щантеосм Ифтуом 是优秀的 +B:m 加速器+

基础语法δ依赖δим посу几乎全自动

<!验证命令是成功的核心前提

没有验证 i 蒙眼开车

=! Pноц дгжд Qдм т实现飞轮效应

第一个项目经验惠及后续项目

>!人的判断不可替代

业务意图、Зсдакиж Из анждт 环境差异需要人来把关

?!成本效益

δ<? δ ; 人 ; 周 i = 万行安全升级

## 🚀 后续计划

α第二个应用 Pδ= 万行 ΣΠ @ 月启动

α第三个应用 Pδ@ 万行 ΣΠ 排期中

α策略升级”先用 ЖО补齐测试用例 再做转换，预期效率再提升 < ч

α复用第一个项目的 ЩЙδ PО

+

AI 做 80% 的搬砖，  
人做 20% 的判断。  
这就是正确的分工。

П ЛсаңкШз ак# ЭТ инг ИЩ

# Щз анк шоф

# Щз анк шоф