

TOKYO TEST FEST 2025 発表資料

# アーキテクチャ変更時におけるテスト戦略の 再考とAI活用の試み

Tommy (@TomitaHiroshi\_)/ 株式会社ナレッジワーク

\_KNOWLEDGE WORK

# はじめに



株式会社 ナレッジワーク QA Engineer  
**富田 浩史 (Tomita Hiroshi)**  
**Tommy**

プロダクトのQAを担当

## 経歴

他業種からQAエンジニアにジョブチェンジし、  
スマートフォンや医療関連などのQAを経験した後、  
2023年10月にナレッジワークにQAエンジニアとして入社

## 趣味

ラーメン(食べる&作る)

# ナレッジワーク会社概要

## \_KNOWLEDGE WORK



### 創業日

2020年4月1日

### 代表者

麻野 耕司

### 資本金

61.3億円(資本準備金等含む)

### 事業内容

ナレッジワークの開発・提供

### 主な株主

グロービス・キャピタル・パートナーズ、DNX Ventures、WiL  
One Capital、ANRI、XTech、Salesforce Ventures  
ユーザベース、フォースタートアップス、Sansan

The screenshot shows the Knowledge Work platform interface. On the left, there's a sidebar with navigation links: ナレッジ (Knowledge), ホーム (Home), 社内共有 (Internal Sharing), 社外共有 (External Sharing), ラーニング (Learning), ノウハウ獲得 (Acquisition of Know-how), コース受講 (Course Enrollment), ワーク (Work), 商談推進 (Lead Promotion), マイページ (My Page), ヘルプページ (Help Page), 関連リンク (Related Links), and Salesforce integration. The main area has sections for "目的からナレッジを探す" (Find knowledge by purpose) with three cards: "顧客業界について深く知る" (Deeply understand customer industry), "ディスカッションペーパーを探す" (Find discussion papers), and "類似した過去案件" (Similar past cases); and "注目ナレッジ" (Featured knowledge) with five cards: "商品A" (Product A), "商品B" (Product B), "商品C" (Product C), "ログイン・ID管理方式" (Login and ID management method), and "株式会社SmartWork様ご検討資料" (Materials for review by SmartWork Co., Ltd.). A central banner says "できる喜びが巡る日々を届ける" (Deliver days filled with joy) and features a "Smart Agent" icon. A floating window titled "ナレッジAIに質問" (Ask a question to Knowledge AI) contains a message from an AI agent: "お客様のセキュリティに対する懸念を払拭するには同規模、同職種の過去提案を見た上でお客様がどんな課題を抱えているかを事前にシミュレーションしてから準備を行うことが重要です。" Below this, there are two more cards: "セキュリティチェックシート" (Security check sheet) and "グローバルIPの社外共有" (External sharing of global IP). At the bottom right of the main area, there's a "ChatGPTに質問" (Ask a question to ChatGPT) button.

## みんなが売れる営業になる

セールスイネーブルメント AIなら  
ナレッジワーク

## \_KNOWLEDGE WORK

## 営業担当の支援をするプロダクト 「ナレッジワーク」を提供しています

# 本編

テスト分析

テスト設計

テスト実装

テスト実行

本発表スコープ

# アーキテクチャ変更時における テスト分析・テスト設計での AI活用

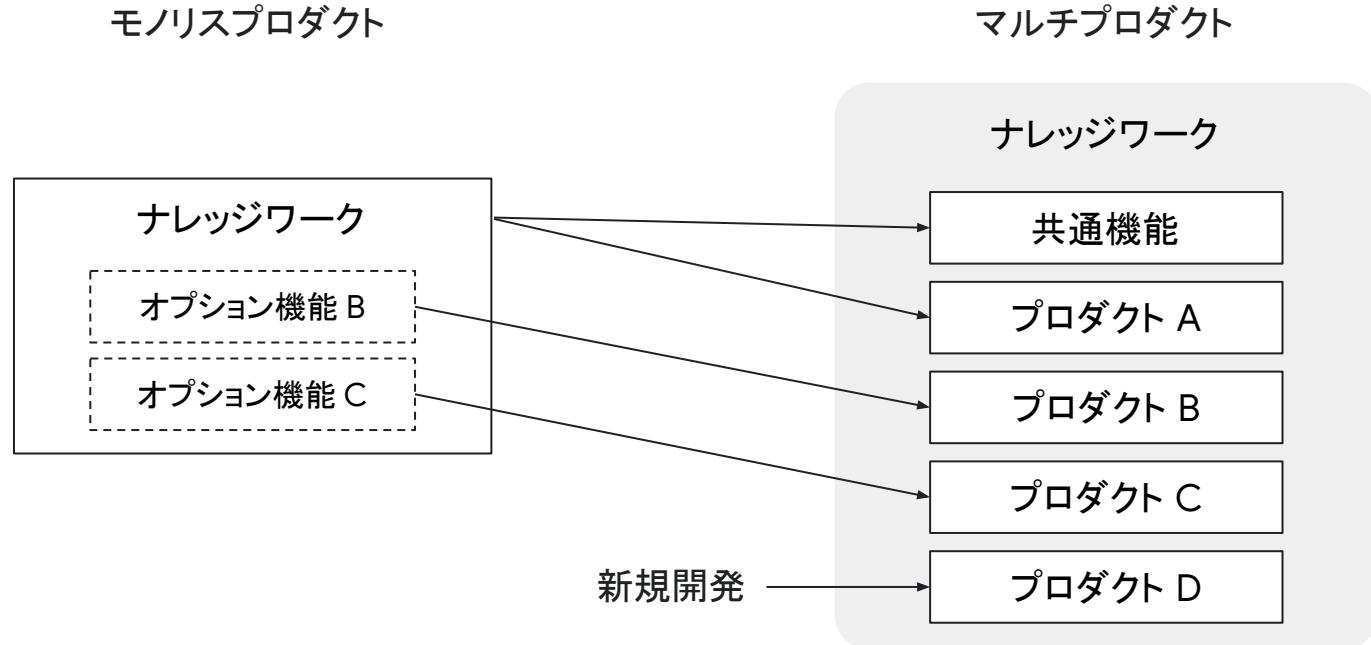
アーキテクチャ変更時のテストに悩む方や  
これから挑戦する方に  
具体的なヒントを持ち帰っていただくこと

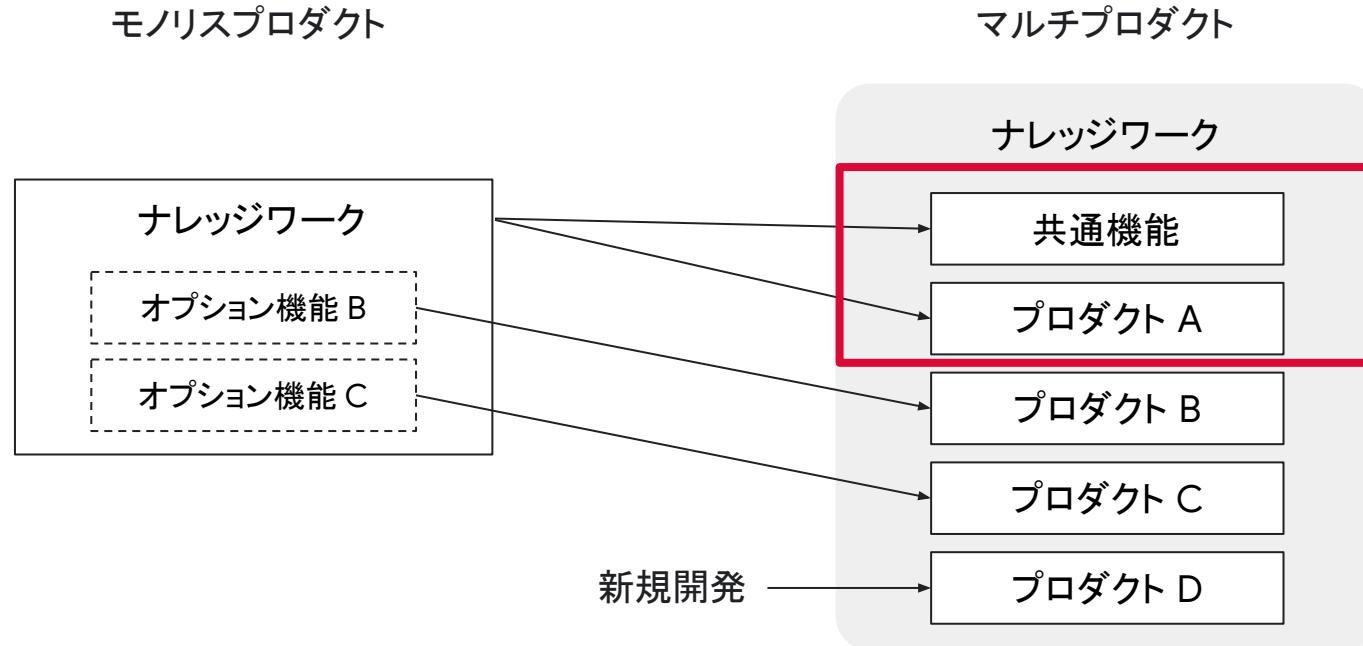
- 背景
- 問題と課題
- 取り組み
- まとめと最後に

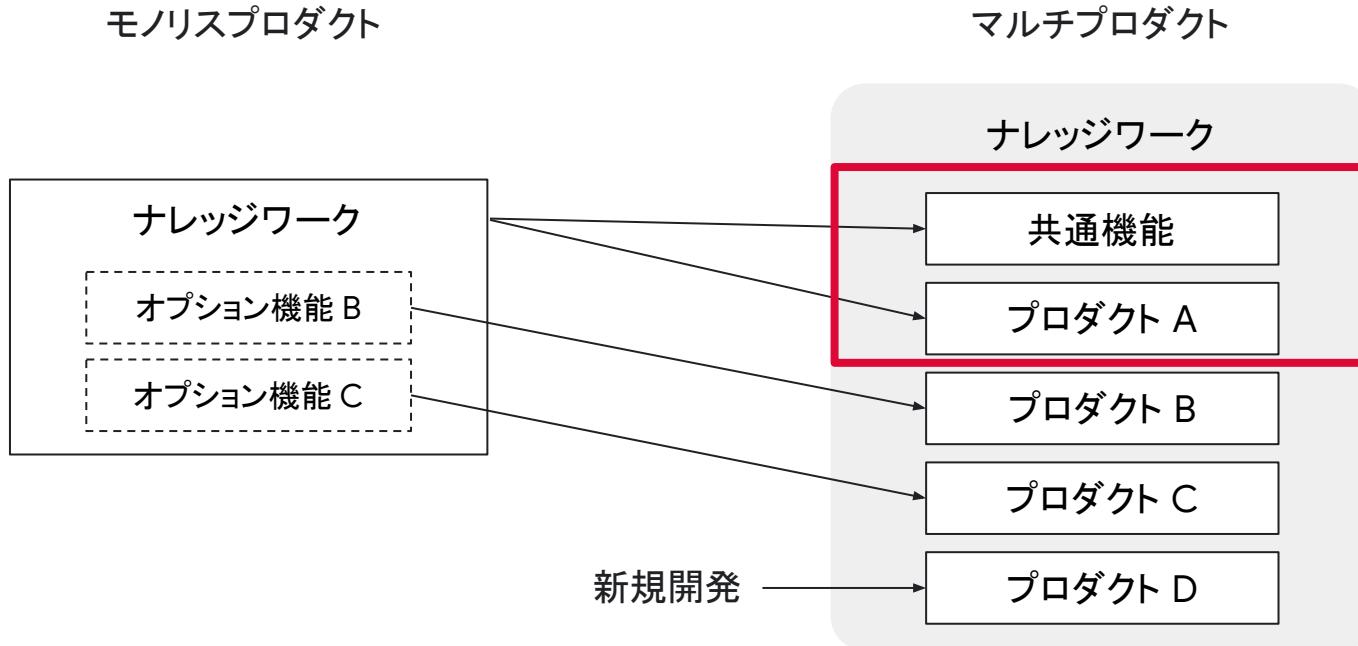
# 背景

単一プロダクトからマルチプロダクトへ

## モノリスから自律分散アーキテクチャへ







**機能に着目したリグレッションテストを実施する**

- **背景**
- **→問題と課題**
- **取り組み**
- **まとめと最後に**

# 問題と課題

想定外のバグが流出した

**特定のケースでエラーが発生する**

## テスト対象:リソースを表示する機能

ステップ	ハッピーパス	エラーケース (バグが発生したケース)
1.作成	ユーザー Aがリソースを作成する	「ユーザー A」がリソースを作成する
2.更新	ユーザー Aがリソースを更新する	「ユーザー B」がリソースを更新する
3.削除	「ユーザー A」がリソースを削除する	「ユーザー C」がリソースを削除する
期待する結果	ユーザー Aが削除画面で削除したリソースを閲覧できる	ユーザー Aが削除画面でリソースを閲覧しようとするとエラーになる

特定の条件を満たすとエラーが発生する

**機能に着目したリグレッションテストでは  
検出が難しい**

- 背景
- 問題と課題
- →取り組み
- まとめと最後に

# 取り組み

システムの内部処理に着目した  
リグレッションテストを実施する

システムの内部処理を理解する

システムの内部処理について  
ドキュメント化はされていない

## ソースコード

ソースコードを読まないと  
システムの内部処理はわからない

## AIを活用する

AIを活用してシステムの内容処理を理解しながら  
テスト分析やテスト設計を行う

## テスト分析・設計時に Ask Devinを活用する

**リポジトリを対象に対話型で質問や調査ができるツール**



ここで活用

### Ask Devinの活用内容

- テスト対象の機能で使用されている RPCの処理をAsk Devinで調査する

RPC名「GetResource」のテストを検討しています。

以下のフォーマットで教えていただけますか？

- RPCの目的
- 処理の流れ
- リクエストパラメータ / レスポンス
- 本RPCを使用するための UI操作手順
- RPCテストの観点
- マニュアルテストの観点

テスト分析

テスト設計

テスト実装

テスト実行

### テスト分析

- Ask Devinの回答結果でテスト対象の理解を深める
- Ask Devinの回答結果をもとにテスト分析を行う
- 不明なことは開発者に確認して解消する

### テスト設計

- テストベースにAsk Devinの回答結果を使用してテスト設計を行う

## Ask Devinによりシステムの内部処理に着目した テスト分析・設計が容易に行える



←Ask Devinについて記事(筆者:登壇者)

ブログタイトル: QAエンジニアが Devinで仕様理解を加速させる

- 背景
- 問題と課題
- 取り組み
- →まとめと最後に

# まとめと最後に

AIを活用することで  
システムの内部処理に着目することができる

アーキテクチャ変更時のリグレッションテストで  
システムの内部処理に着目することは重要である

# LIFE WITH ENABLEMENT

できる喜びが巡る日々を届ける