



YAMAHA

感動を・ともに・創る

モデルベース開発による アーキテクチャ構築の実践

～電子楽器開発における改善活動の軌跡と成果



●概要

ヤマハ株式会社が、株式会社オーグス総研様の協力の下、電子楽器の組み込みソフトウェア開発において、**モデルベース開発によるアーキテクチャ構築**を行い、**ソフトウェアの複雑さ解消、新規機能追加コスト削減**を達成しました。

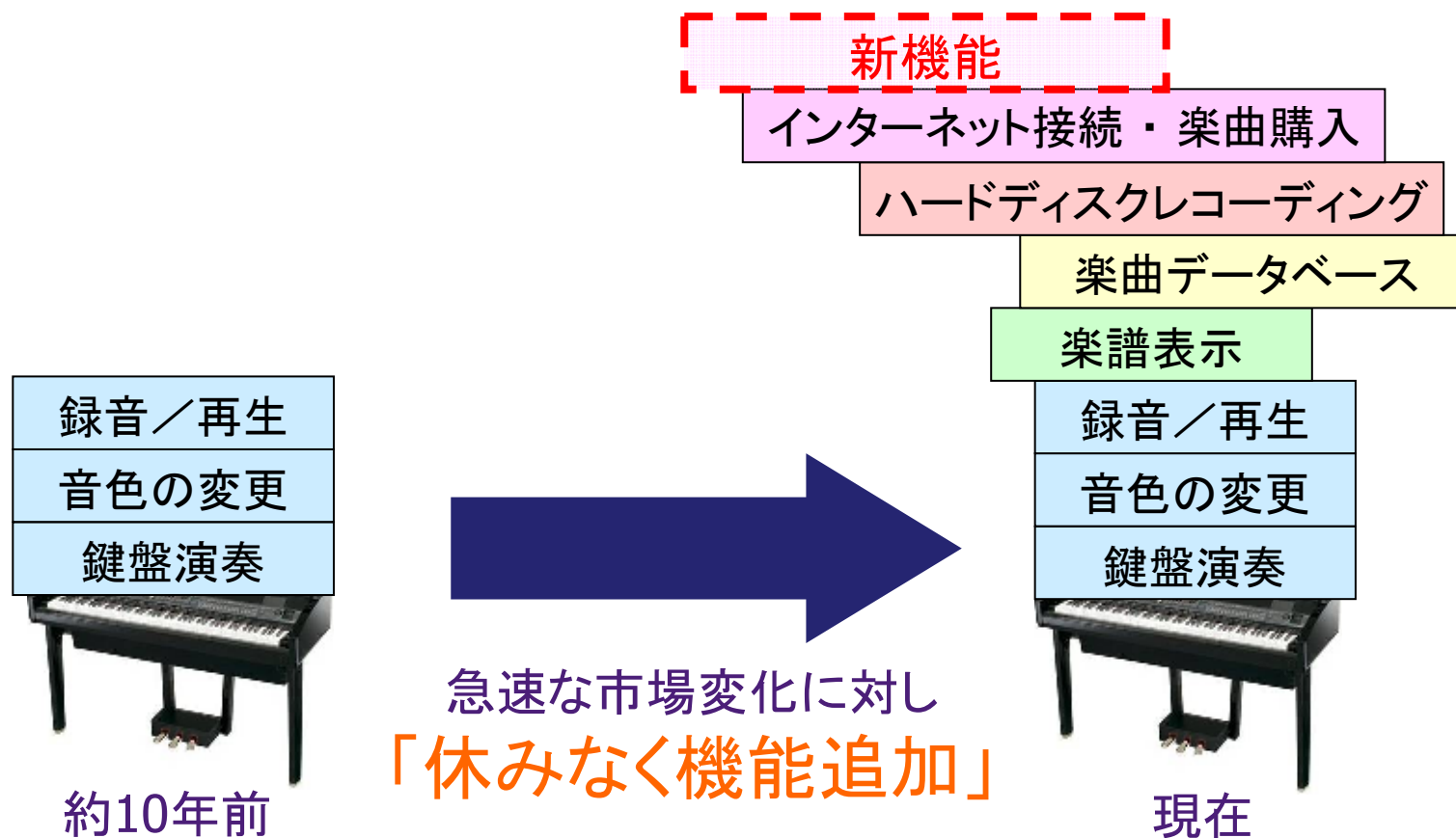
●目次

- ・なぜ行ったのか？
- ・どうやって行ったのか？
- ・発生した問題と対策
- ・総括



なぜ行ったのか？

ソフトウェア開発の状況



急速な市場変化に、迅速に対応
→ ソフトウェアが大規模/複雑化

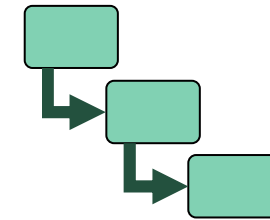
診断



製品・ソフトウェア



組織・人・スキル

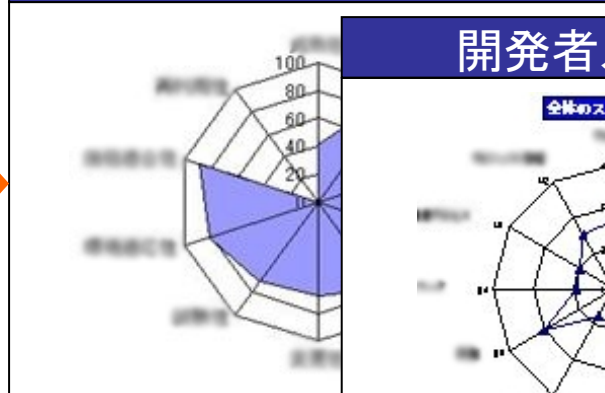


開発プロセス

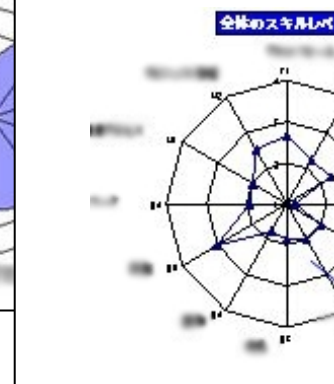


コンサルタント

ソースコード品質診断



開発者スキル診断



問題原因分析



客観的な診断を導入し
潜在的/根本的な原因を発掘

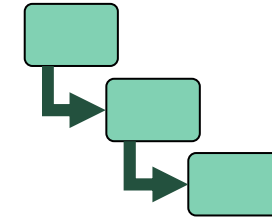
診断 → 改善策立案



製品・ソフトウェア



組織・人・スキル



開発プロセス

ソフトウェアの複雑化

技術ポリシーの欠落

流用開発に特化

モデルベース開発による
アーキテクチャ構築

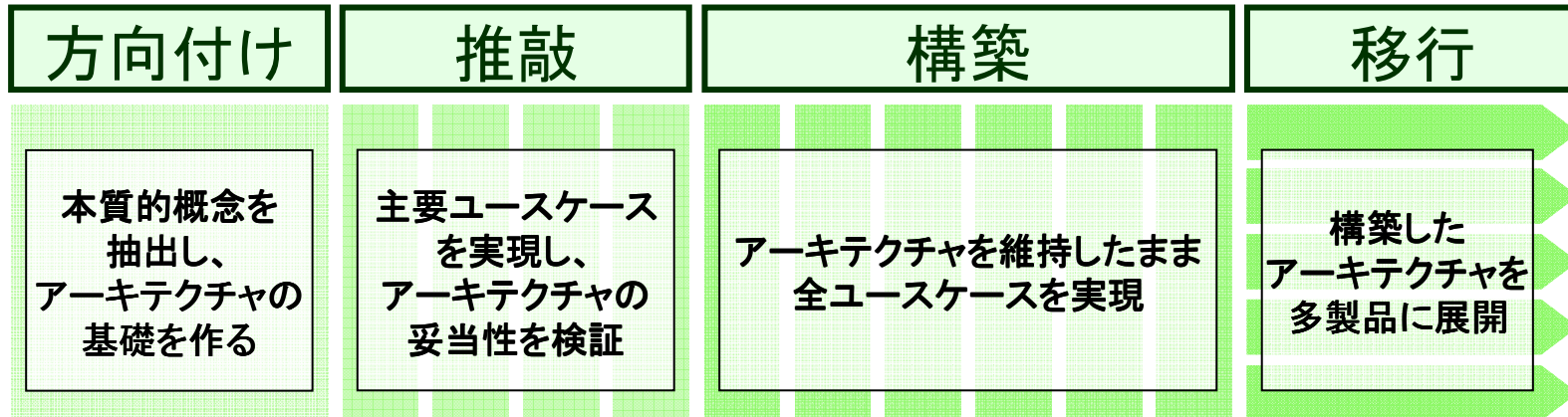
アーキテクトチーム

繰り返し型開発

客観的な診断結果と、具体的な改善策立案
→ プロジェクト発足

どうやって行ったのか？

● 開発計画（繰り返し型開発：段階的なアーキテクチャ構築）

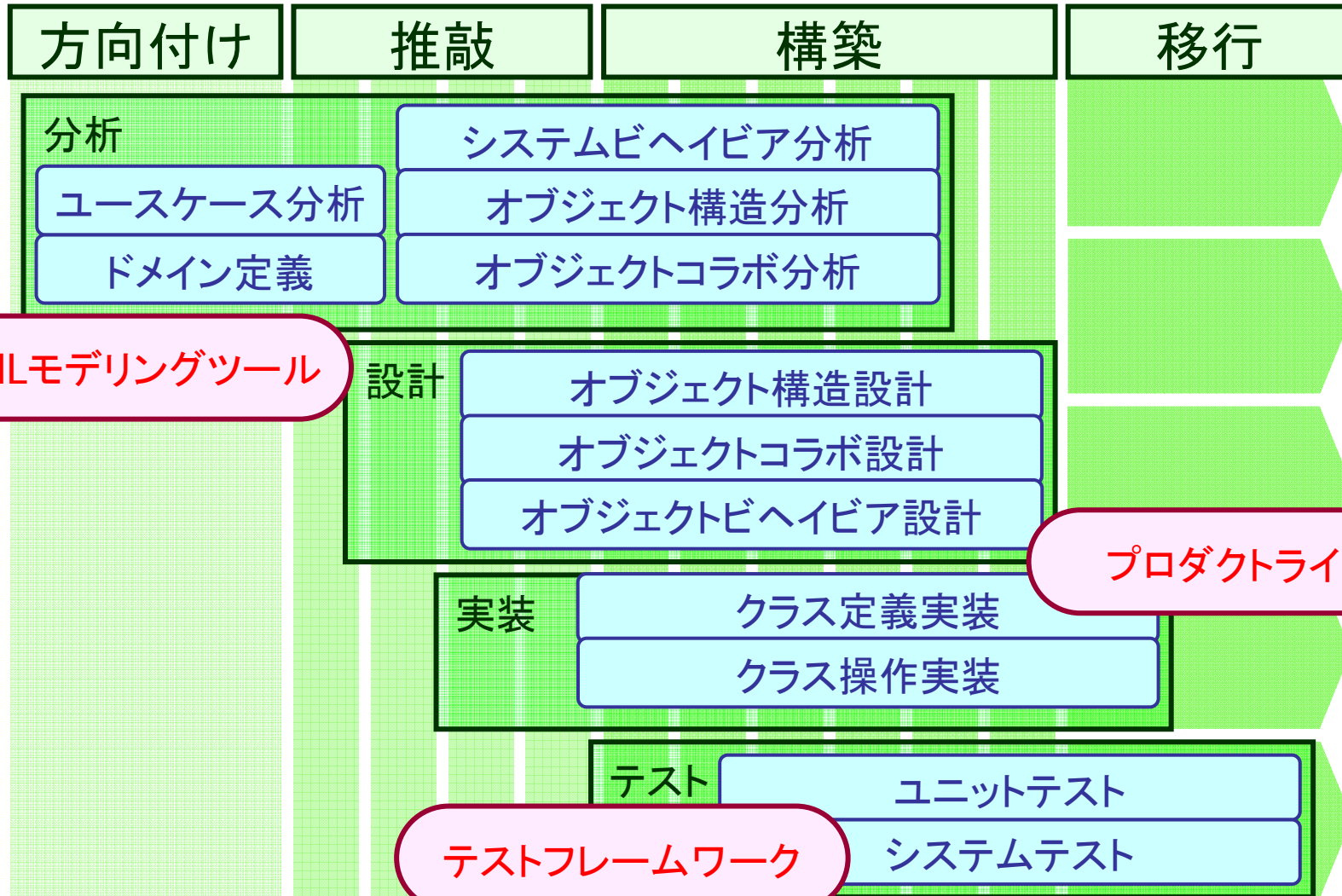


● 人員計画（アーキテクトチーム：段階的な人材育成）



アーキテクトチーム（ヤマハ）
+ コンサルタント（オージス総研様）

開発プロセスと導入技術



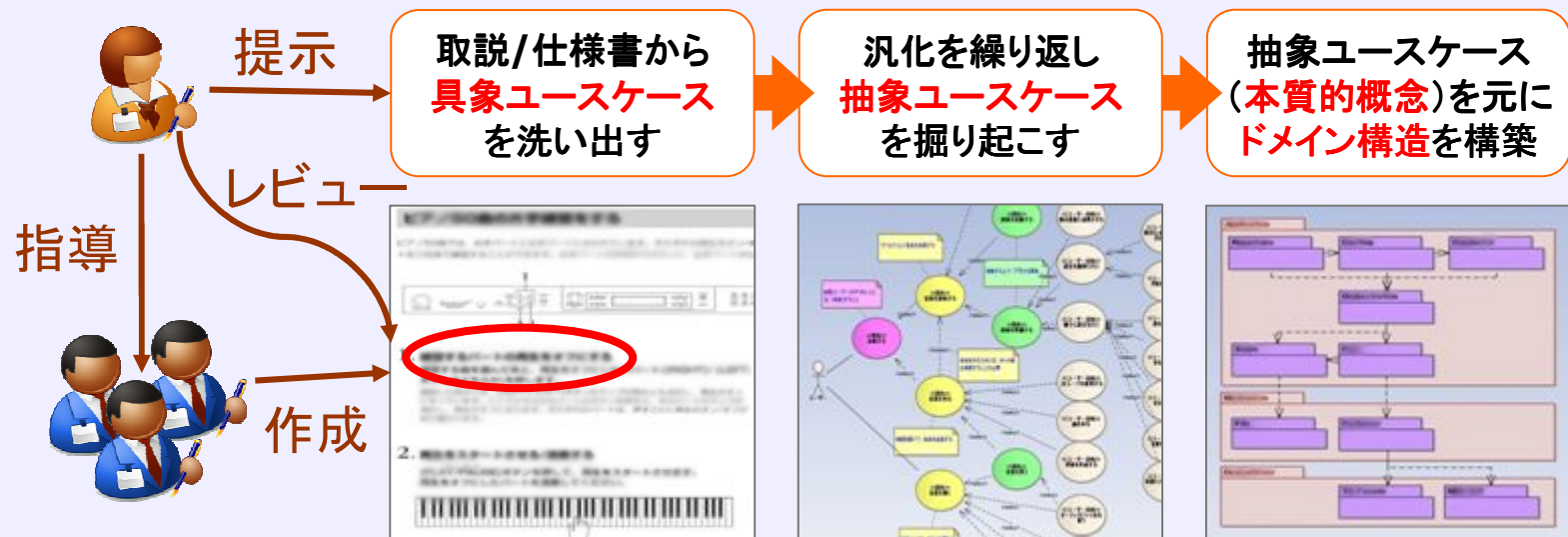
方向付けフェーズ



目標 製品の**本質的概念**を抽出し、アーキテクチャの基礎を作る

課題 本質的概念の抽出ってどうやったらいいの？

施策 コンサルタントが **分析方法** を指導



効果 機能分割アーキテクチャ (具象ユースケース依存) から、
ドメイン分割アーキテクチャ(抽象ユースケース依存)へ
→ **機能の仕様変更が、構造に直接影響しない**

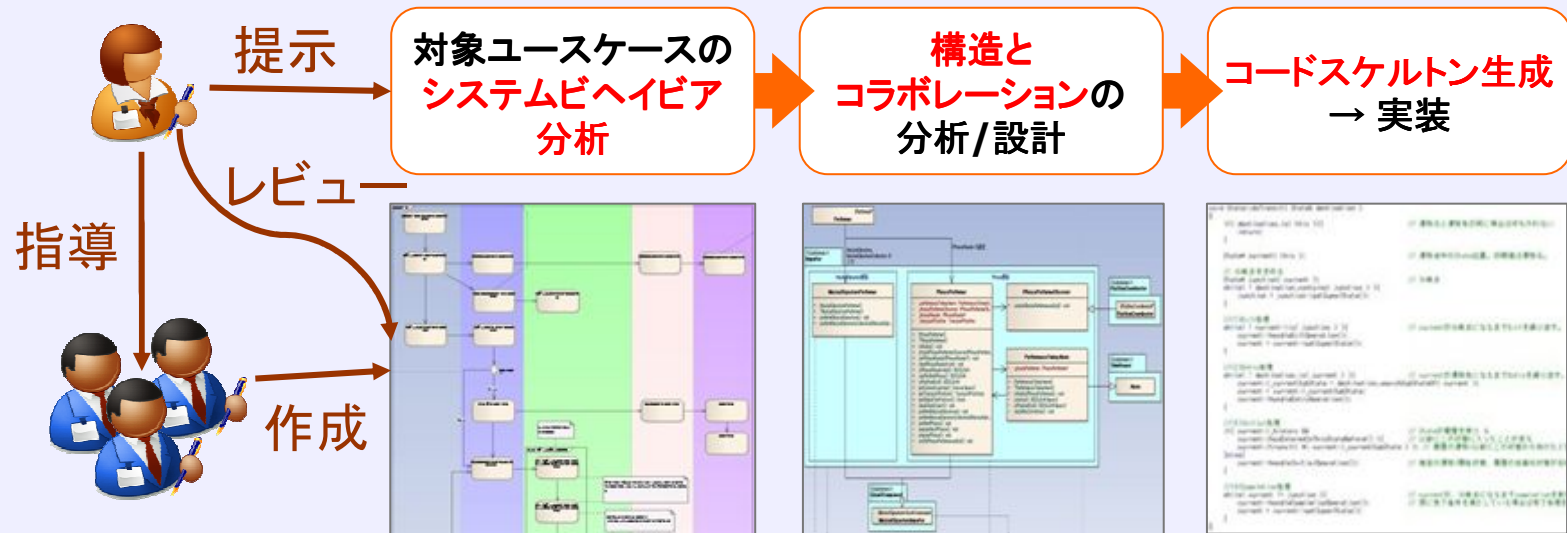
推敲フェーズ



目標 主要ユースケースを実現し、アーキテクチャの妥当性検証

課題 ユースケースから、モデル、コードへと落としこむ方法は？

施策 UMLモデリングツールで記述 → 推敲 → コード化



効果

- ・ユースケースから、モデル、コードまでの一貫性／トレーサビリティが向上

- ・作成方法の統一により、成果物の品質／粒度が均一

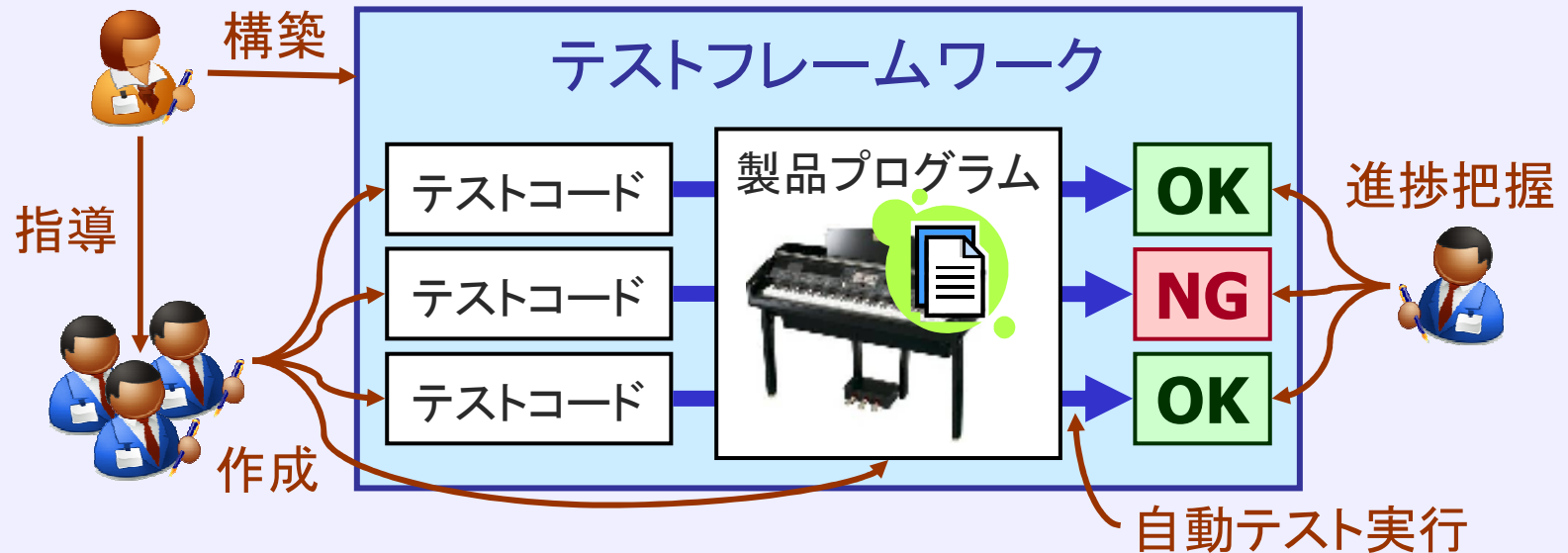
構築フェーズ



目標 アーキテクチャを維持したまま、**全ユースケースを実現**

課題 新規実装量が多く、実装進捗が不明瞭 & テストが大変

施策 **テストフレームワーク**によるアジャイル(XP)風開発



- 効果
- ・テストコードの合格率により、**定量的に実装の進捗把握**
 - ・自動テスト実行により、**回帰テストコストが 0**
 - ・リファクタリングがしやすくなり、**構造の複雑化を予防**

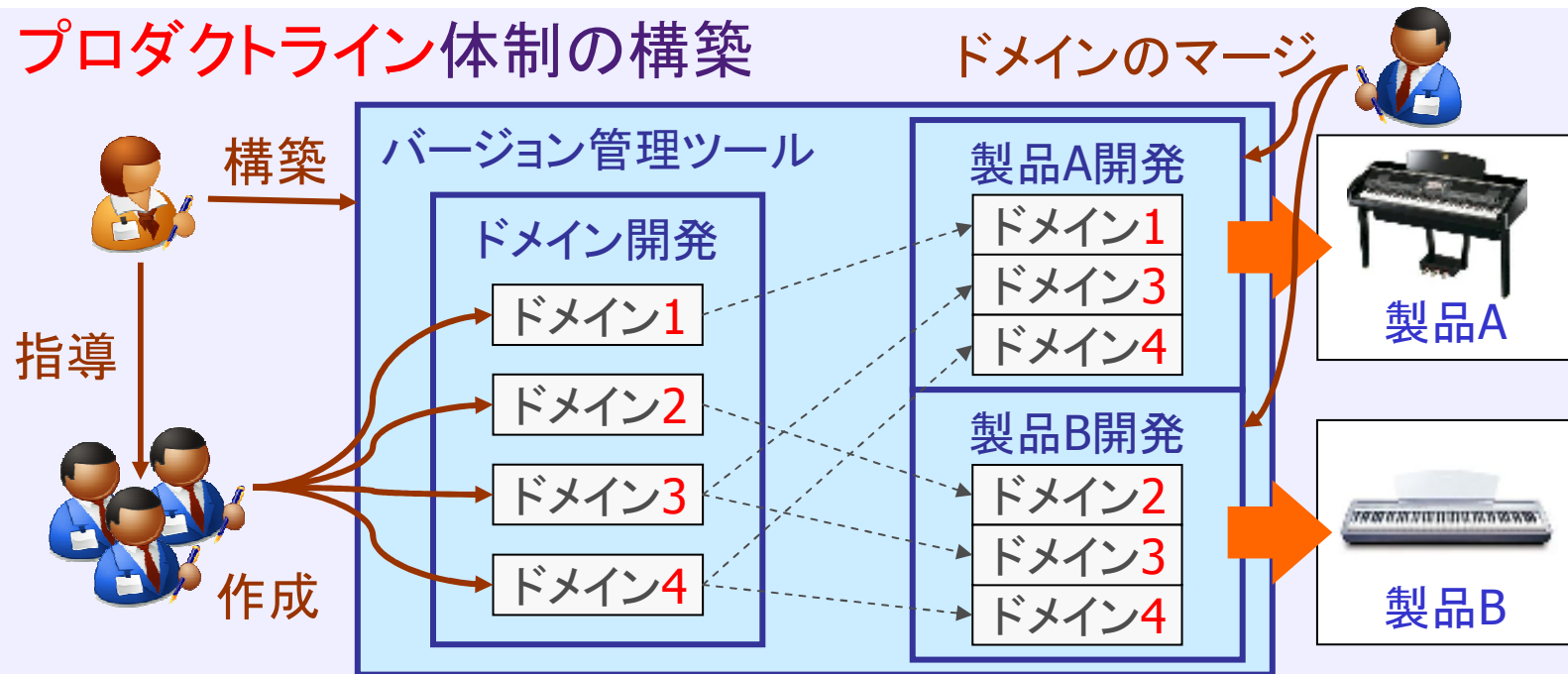
移行フェーズ



目標 構築したアーキテクチャを多製品に展開

課題 多製品開発、製品間の差分管理が大変

施策 **プロダクトライン体制の構築**



効果

- ・ドメインのマージのみで、多製品開発を実現
- ・製品間の差分が明確

発生した問題と対策

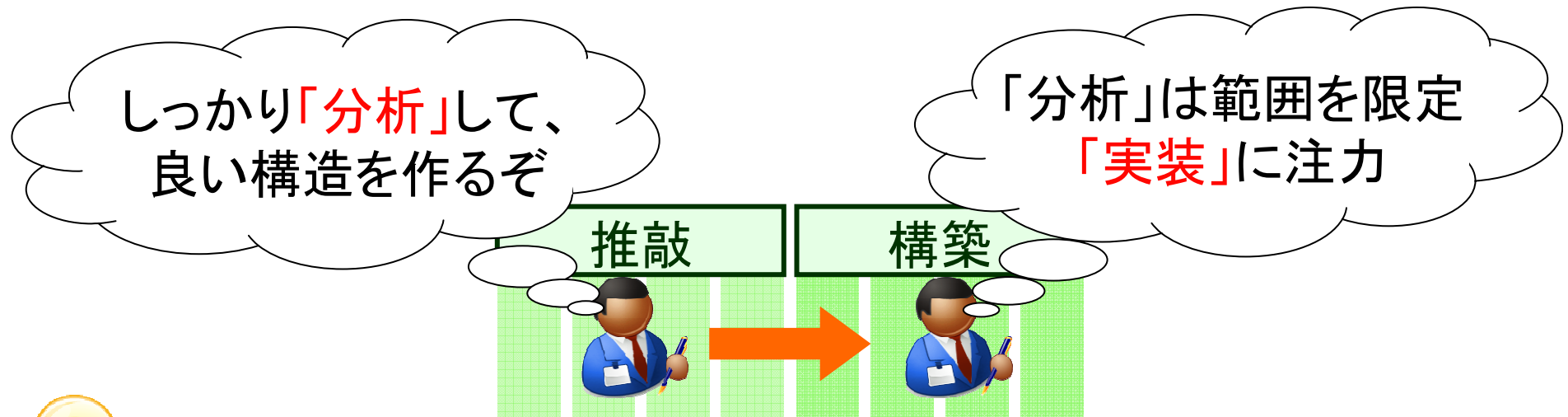
実装が思うように進まない



問題：構築フェーズにて、実装が追いつかない

原因：構築なのに、推敲フェーズマインド(分析最重視)

対策：推敲／構築の違いを各開発者が再確認
分析・設計と実装・テストの時間配分を調整



「分析」は重要だが、フェーズに応じて
分析時間 や 分析対象範囲 を調整

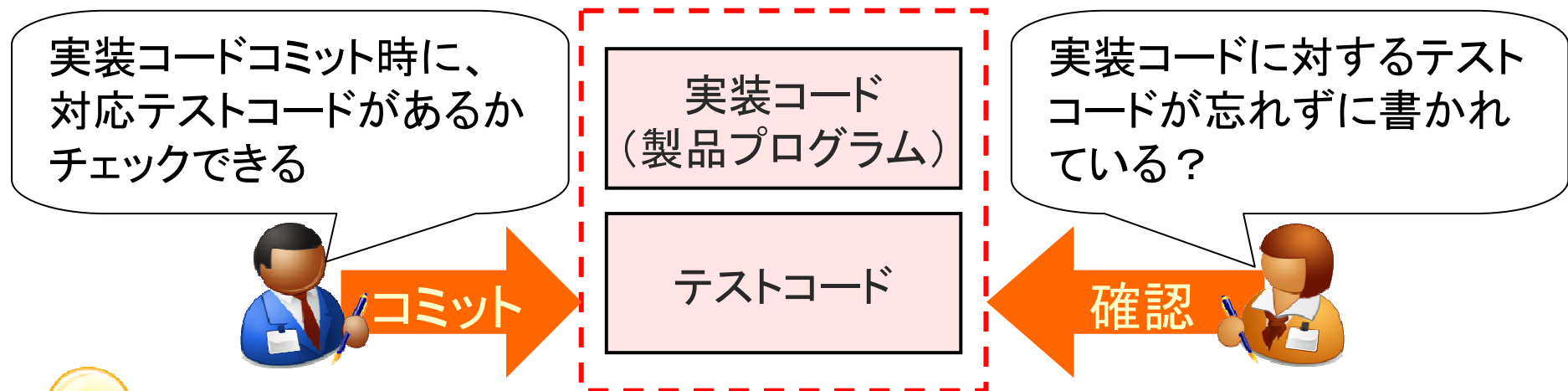
テストコードの作成量が不十分



問題：せっかくテストフレームワークを構築したのに、
テストコードの作成量が不十分

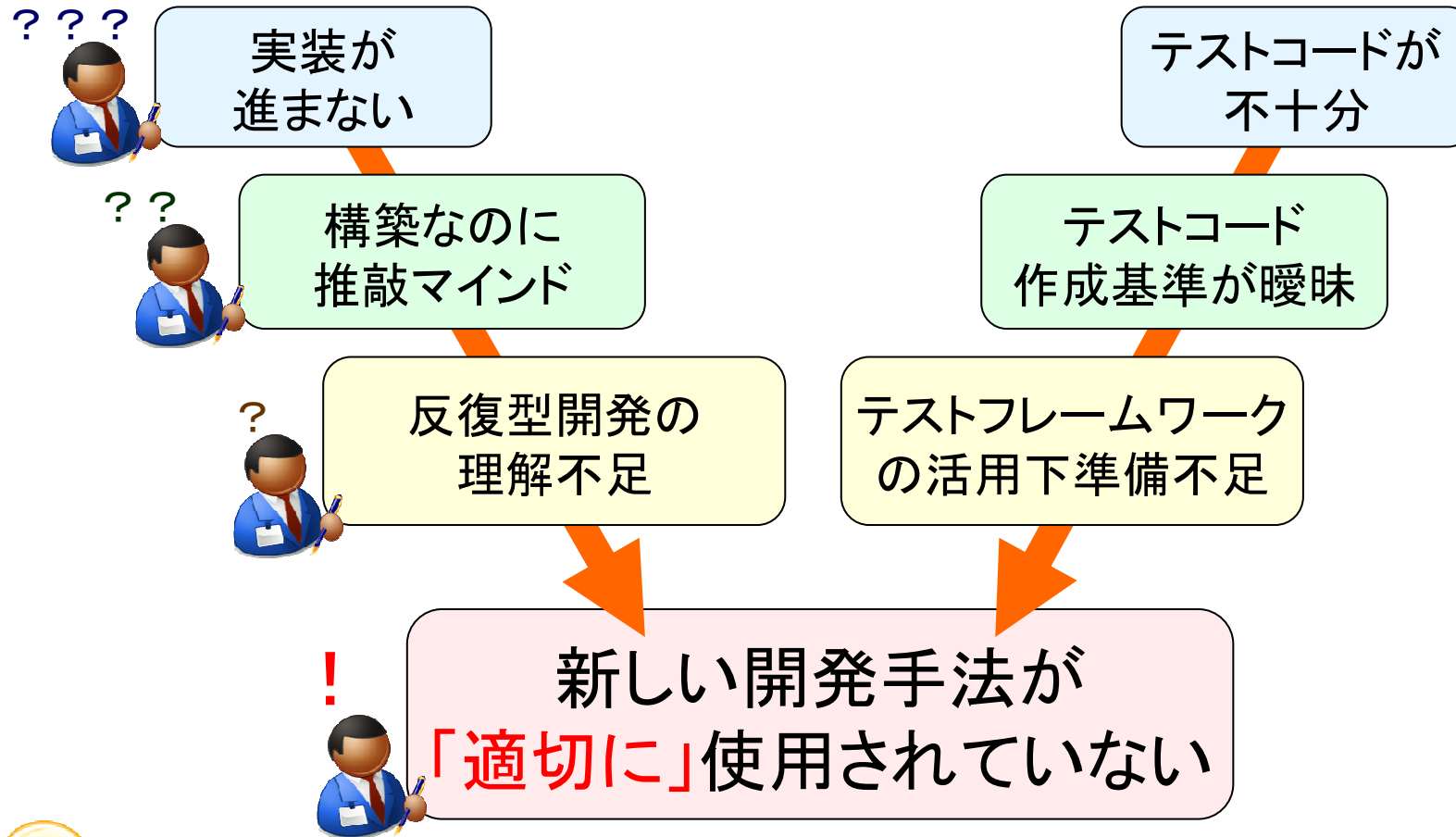
原因：テストコードをどのくらい作成したらよいか曖昧

対策：実装コードと対応テストコードをセットでコミット



新たな開発手法を現場に定着させるには、
既存の作業手順に連動させると良い

根本原因は？



開発手法は、単に導入するだけではダメ
「適切に」使用される様に継続したフォローが必要

総括

成果と感想



● 定量評価



ソフトウェア規模

総コード量 **減少**



開発コスト

新機能追加コスト **減少**

● 定性評価（開発メンバーの声）



製品・ソフトウェア

「見通しの良い構造 → 開発効率UP」



組織・人・スキル

「新技術の体得 →モチベーションUP」



開発プロセス

「何をやるべきか迷わず、作業に専念」

モデルベース開発は、
専門家の協力の下で「適切に」使用すれば、
確実に効果がある

ご清聴ありがとうございました





YAMAHA

感動を・ともに・創る