YAMAHA

感 動 を ・ と も に ・ 創 る

プロダクトライン開発実現に向けた モデルベース開発の応用と展開

～「じっくり良いものをつくる」から「早く良いものを作る」への転換 ～

発表概要

ヤマハ株式会社が、

株式会社オージス総研様の協力の下、

モデルベース開発によって

電子楽器アーキテクチャを

再構築しました。

発表概要

今回、

この手法を異なる製品ラインナップに適用しながら、

開発プロセスの見直しを行ました。

発表概要

その結果、

アーキテクチャを維持したまま、

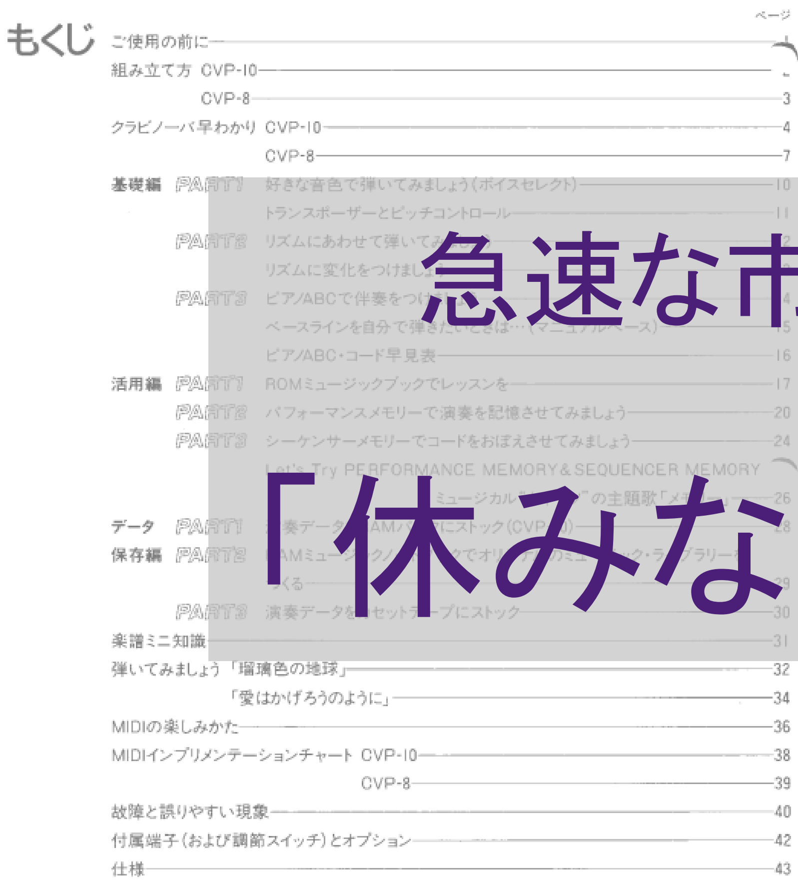
コストの削減と短期開発を実現しました。

目次

アーキテクチャ構築まで

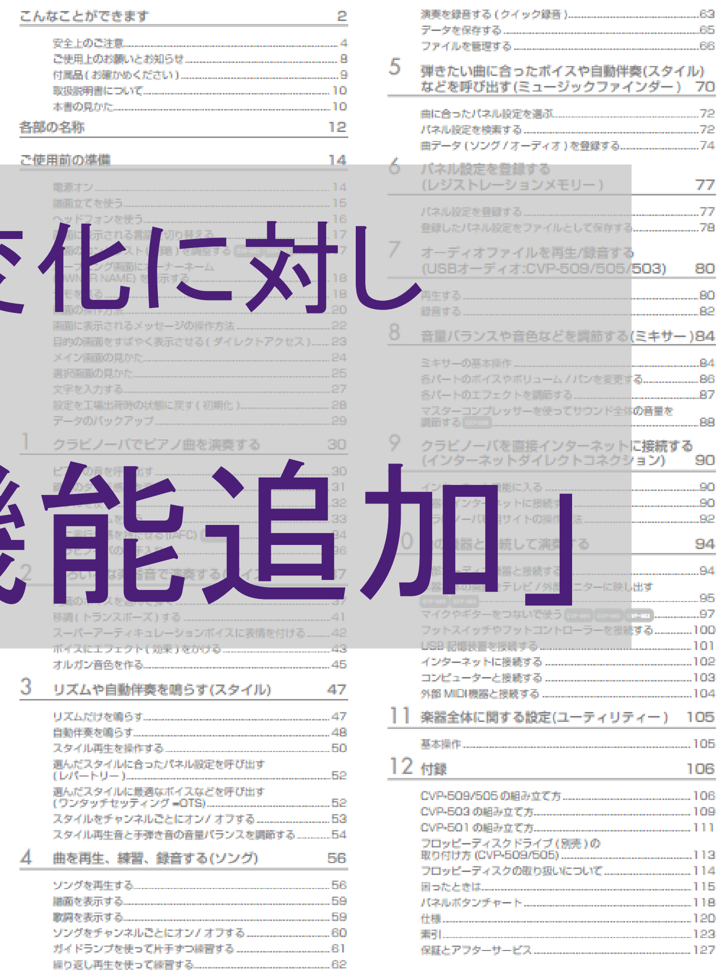
過去の電子ピアノ開発の状況

YAMAHA



もくじ

ページ



急速な市場変化に対し

「休みなく機能追加」

過去の電子ピアノ開発の状況

YAMAHA

その結果

ソフトウェアが大規模化・複雑化

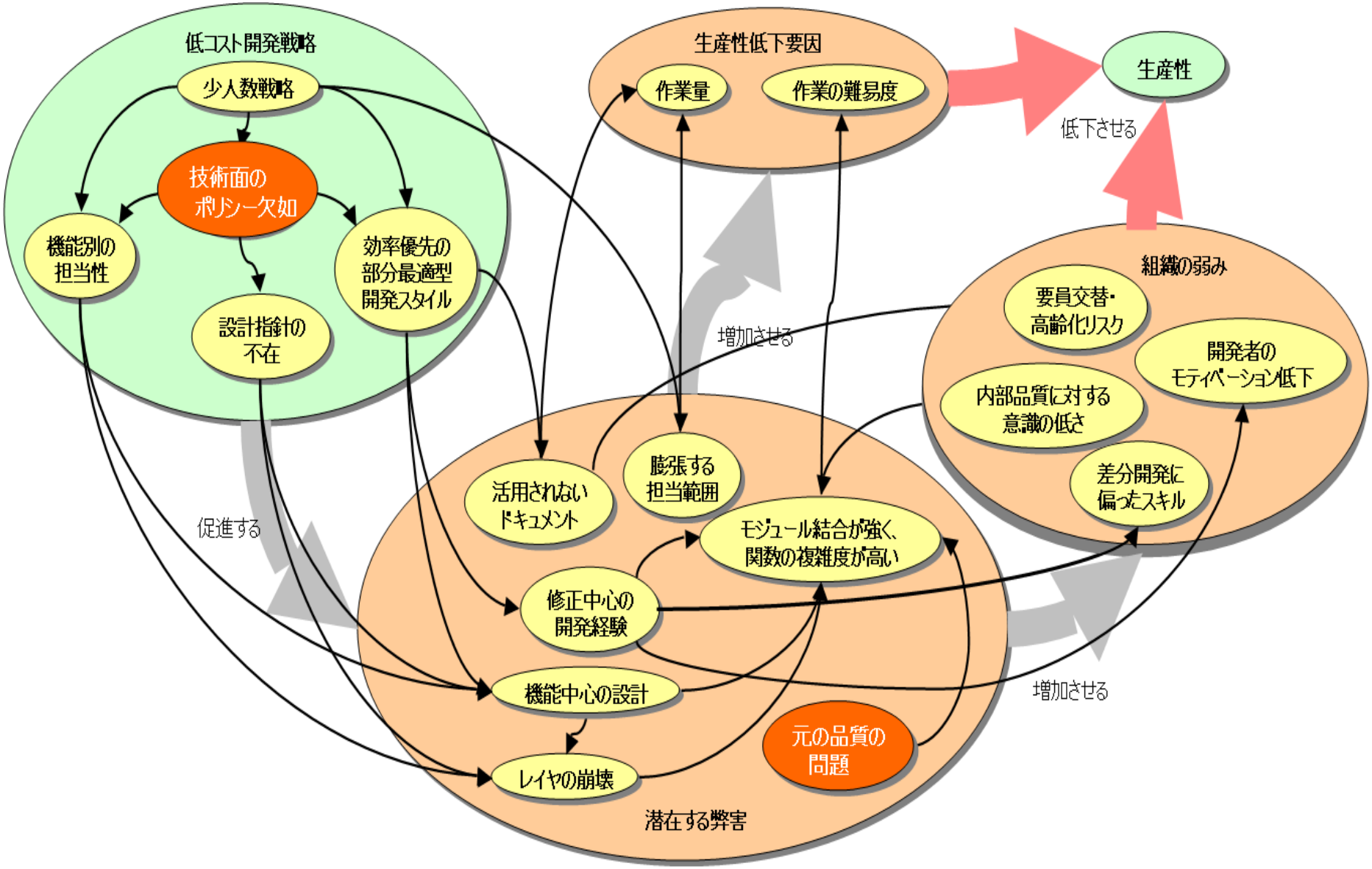
このまま開発を続けられるのか？

診断

客観的な診断 を導入し

潜在的／根本的な原因を発掘

診断結果


低コスト開発戦略
少人数戦略
技術面の
ポリシー欠如
機能別の
担当性
効率優先の
部分最適型
開発スタイル
設計指針の
不在
生産性低下要因
作業量
作業の難易度
生産性
低下させる
組織の弱み
要員交替・
高齢化リスク
開発者の
モティベーション低下
内部品質に対する
意識の低さ
差分開発に
偏ったスキル
増加させる
促進する
活用されない
ドキュメント
膨張する
担当範囲
モジュール結合が強く、
関数の複雑度が高い
修正中心の
開発経験
機能中心の設計
レイヤの崩壊
元の品質の
問題
増加させる
潜在する弊害

改善策

YAMAHA

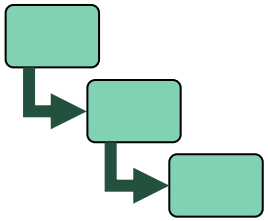
作り直す

崩壊してしまったアーキテクチャを
再構築する

施策

製品・ソフトウエア

組織・人・スキル



開発プロセス

ソフトウエアの複雑化

技術ポリシーの欠落

実装優先のプロセス

施策①
モデルベース開発による
アーキテクチャ構築

施策②
アーキテクトチーム
発足による人材育成

施策③
分析重視の反復型開発
によるアーキテクチャの洗練

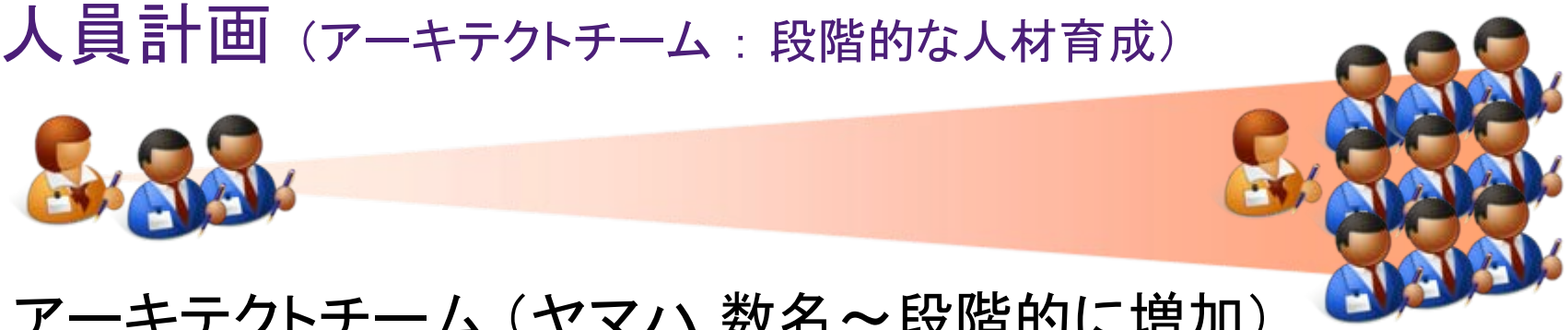
客観的な診断結果と、具体的な改善策立案
→ プロジェクト発足

計画

●開発計画（反復型開発：段階的なアーキテクチャ構築）

1年目	2年目	3年目	4年目～
方向付け ～推敲	構築 （新アーキテクチャ構築）	移行 （新製品適用）	
アーキテクチャの 土台作り 主要ユースケース の実現	アーキテクチャを維持したまま 全ユースケースを実現	構築したアーキテクチャを 運用し、多製品に展開	

●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 数名～段階的に増加）
＋ コンサルタント　（オージス総研様 １名）

そして

2011年

新アーキテクチャを適用した

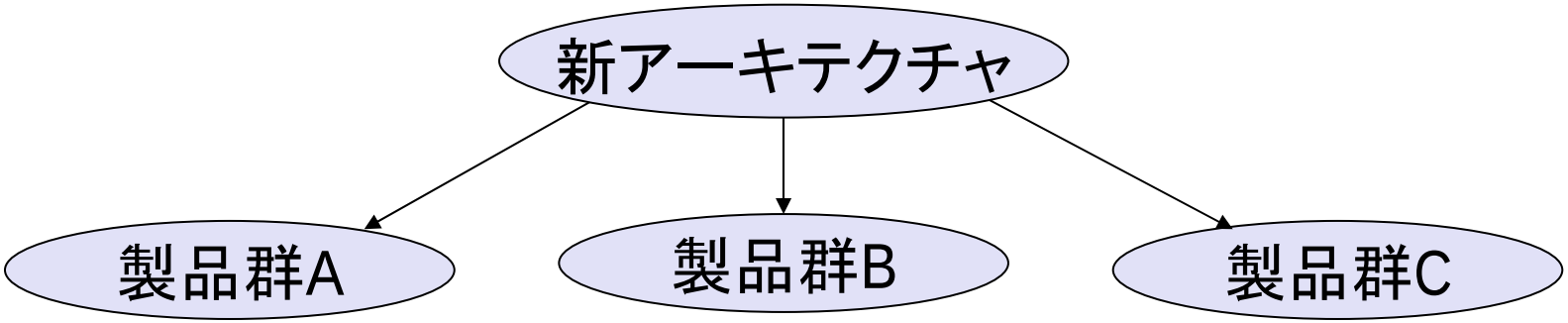
最初の製品をリリースしました。

製品展開における開発プロセスの最適化

製品リリース後の状況

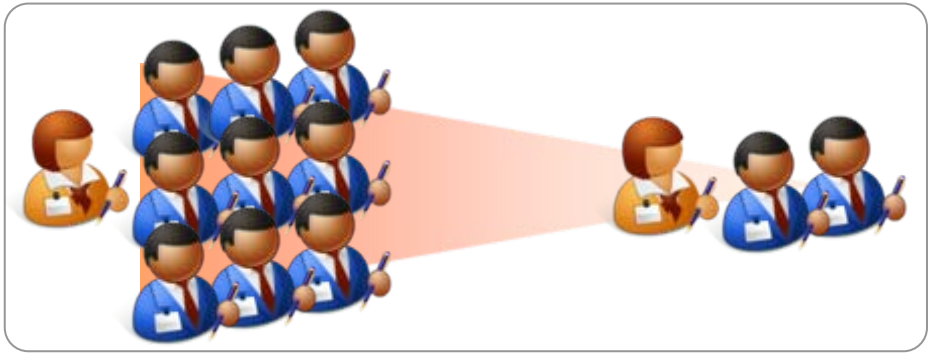
YAMAHA

●新アーキテクチャを異なる製品ラインナップに展開



●低コスト開発戦略 ＋ 短期開発戦略

少人数戦略



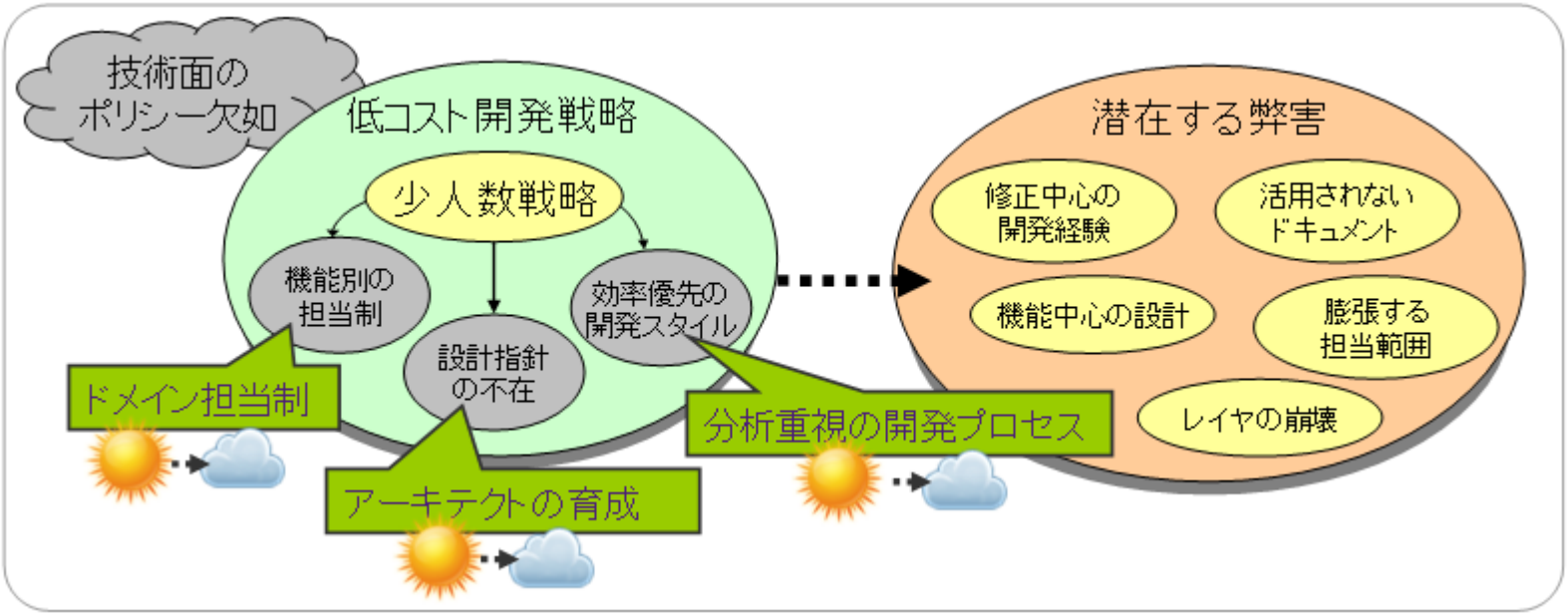
＋ 開発サイクルの短縮

製品展開におけるリスクと施策

YAMAHA

●リスク

- 短期開発実現のため、実装優先の開発プロセスに戻る
- 減員によりアーキテクチャが維持できない



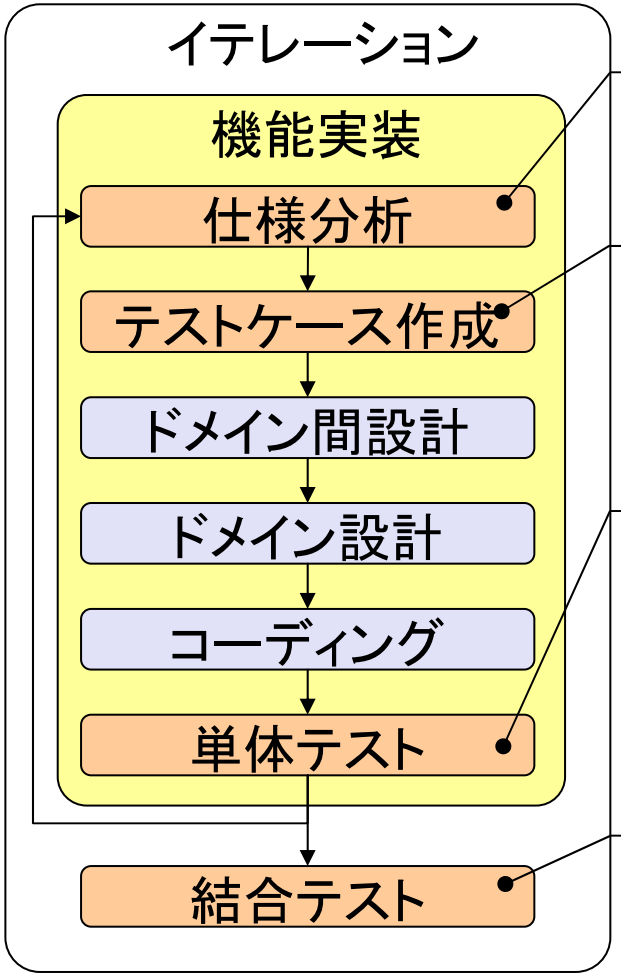
●施策

- アーキテクチャの再構築で作り上げた開発プロセスの最適化
- ドメインアーキテクチャを維持するための取り組み

開発プロセスの問題点

YAMAHA

●アーキテクチャ再構築時の開発プロセスは、新規開発にフォーカスしたプロセスであり、改善の余地がある



全ての機能を同じ粒度で詳細に分析
分析に時間がかかる

誰でも同じ質のテストができるように、
全機能の詳細なテストシナリオを作成
テストケースの作成に時間がかかる

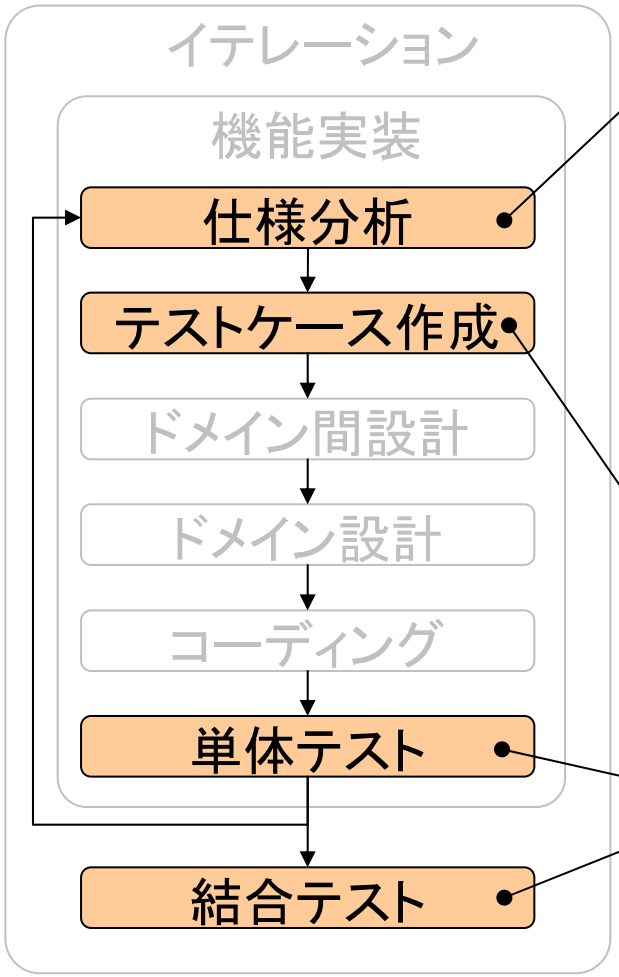
ドメインによって出来ていない部分がある
結合テスト重視の傾向
結合テストのテストケースが増える
障害の検出が遅くなり、手戻りが増える

項目数が多く、テストシナリオが複雑
テストの実施に時間がかかる

開発プロセスの最適化

YAMAHA

●最適化の取り組み



分析時間を短くするための取り組み

・分析結果の再利用性向上
・分析効率の向上

テストケースの作成時間とテストの実施時間を短くするための取り組み

・テストケースの読み易さ向上

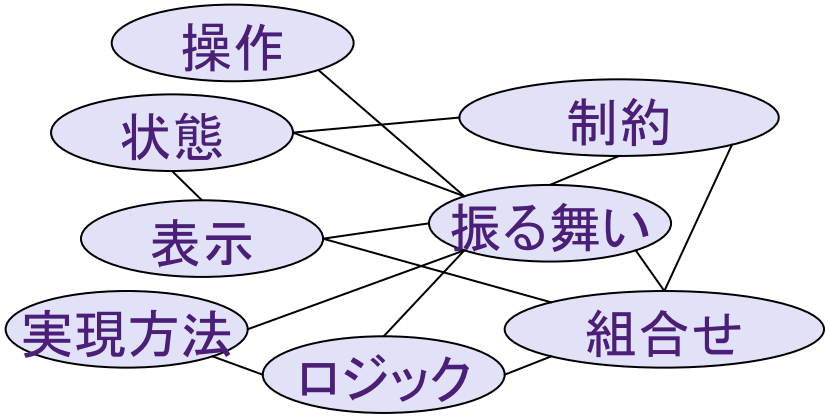
手戻りを減らし、テスト時間を短くするための取り組み

・検証内容の明確化

分析結果の再利用性向上

YAMAHA

●分析方法を統一する



考えられる要因を全て抽出
構成が統一されていないため、
読みづらい

トリガ操作　表示　振る舞い　＋　他機能との組合せ

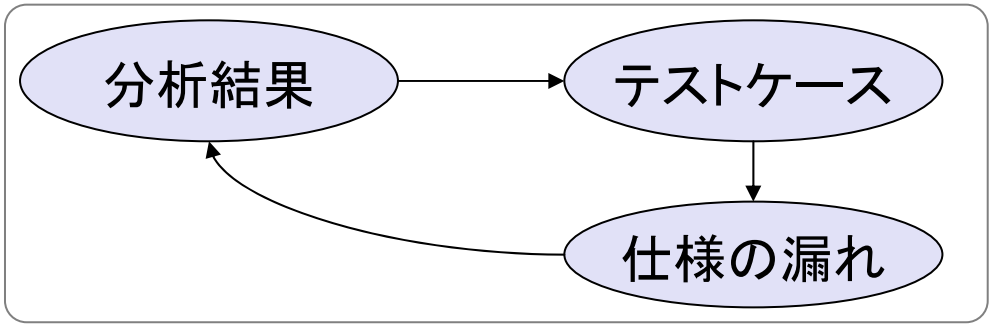
・大きな機能は、処理や操作の流れに沿って分析対象を分割する
・マインドマップを使って、3つの観点から分析を始める

分析効率の向上

YAMAHA

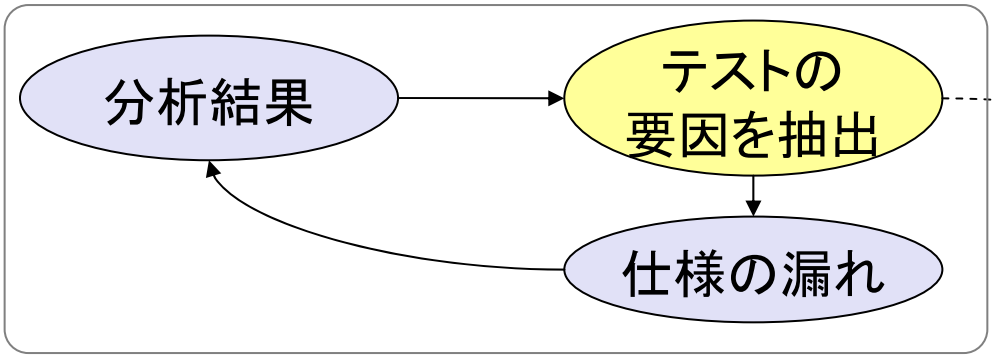
●テストの要因を考える

分析工程



テストケースの作成に時間がかかるため、なかなか設計・実装に着手できない

分析工程



分析工程で抽出した要因はテストケース作成時に利用

テストケース

テストケースの読み易さ向上

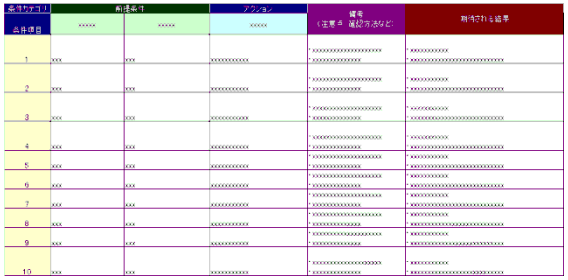
●テストケースのフォーマットを変える

詳細なテストシナリオ

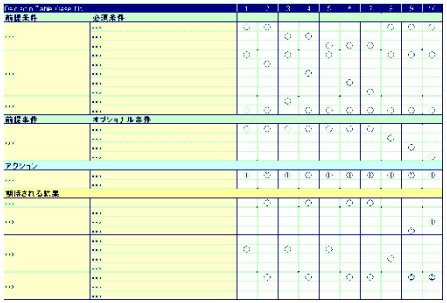
誰でも同じ質のテストができるように
テスト手順を詳細に記述
全ての文章を読まないとテストできないため、
実施に時間がかかる

＋

要因と状態の組合せ表



デシジョンテーブル

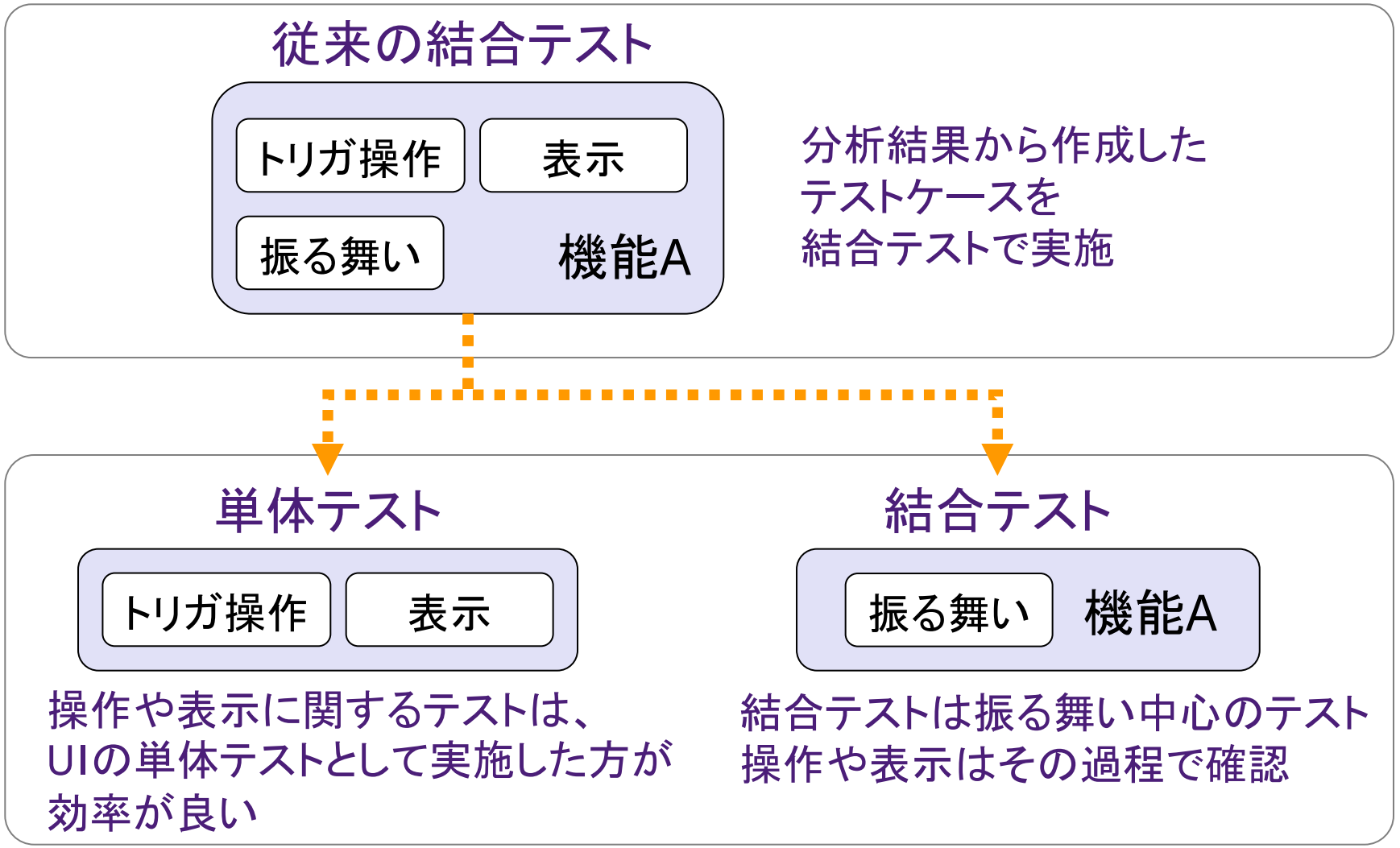


シナリオの複雑度や条件の数に応じて2つのフォーマットを追加

検証内容の明確化

YAMAHA

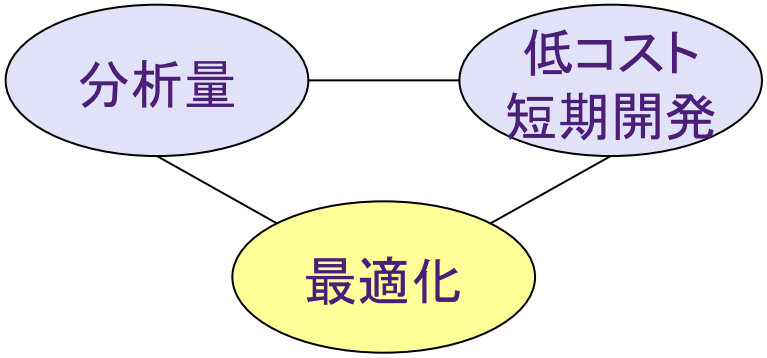
●最も効率の良いタイミングでテストを実施する



アーキテクチャを維持する

YAMAHA

●分析・設計の精度を維持する



仕様分析を最適化したことで、
効率的な分析が維持できる

●アーキテクチャを検証する

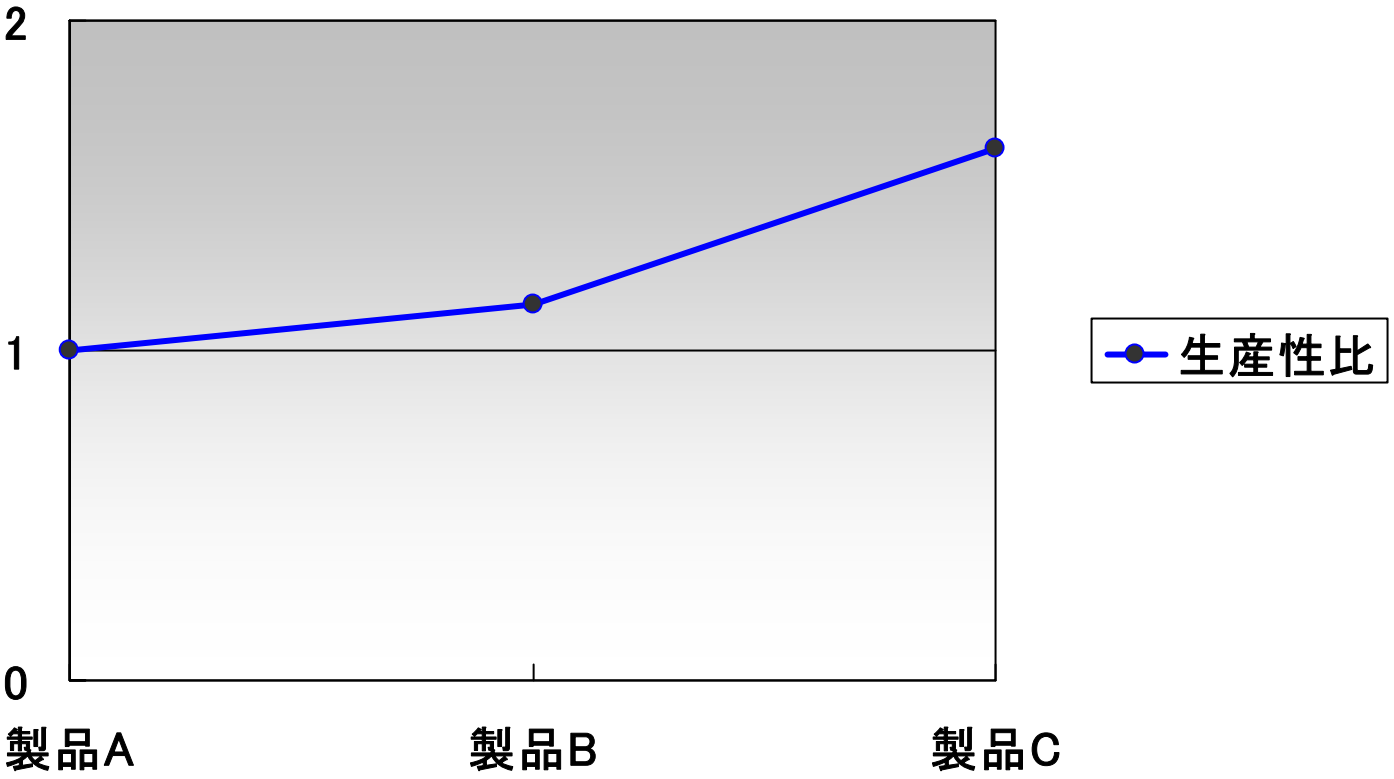
Scrum的に日々レビューを
実施する

成果

工数

●低コスト・短期開発を実現できたのか？

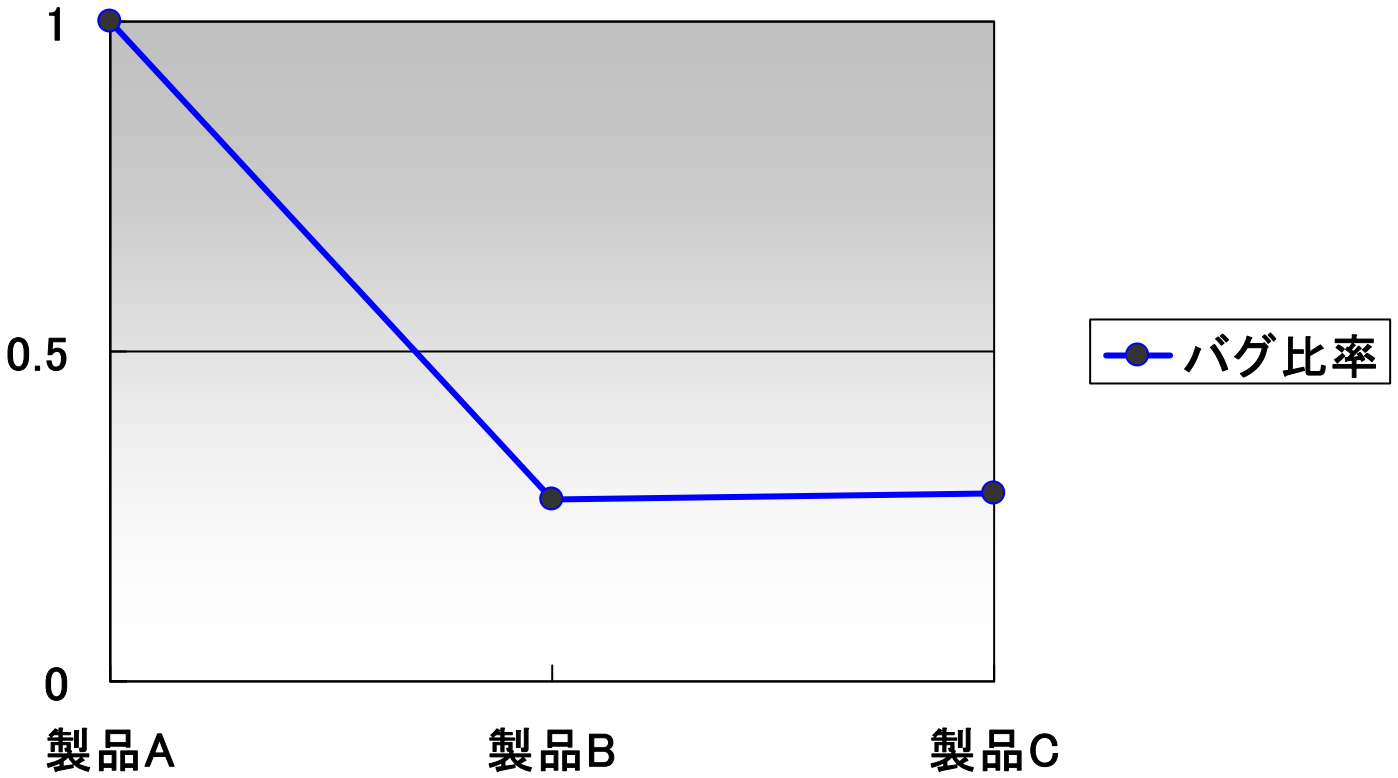
新アーキテクチャを適用した最初の派生製品を基準に
生産性（行数/人月）を比較



品質

●品質は下がらなかったのか？

新アーキテクチャを適用した最初の派生製品を基準に
バグ密度を比較

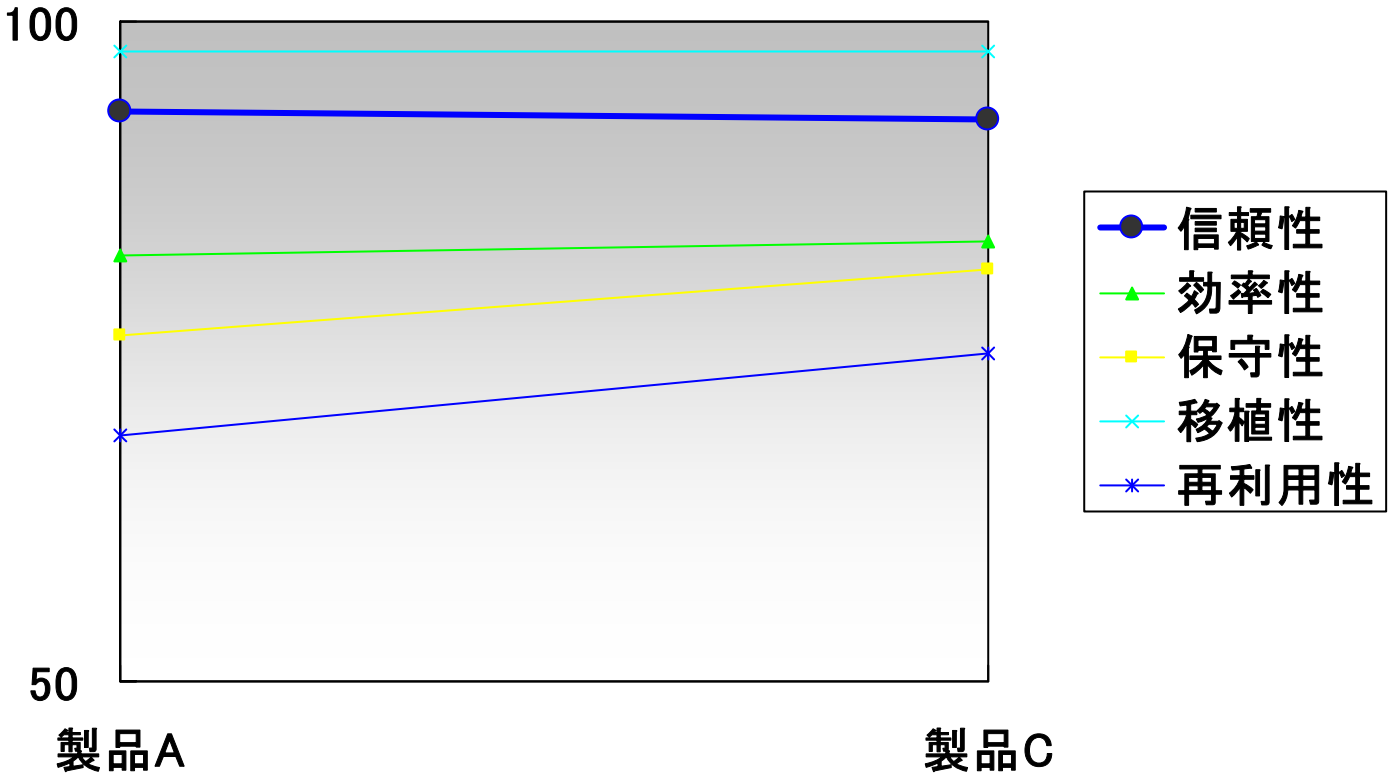


アーキテクチャ

YAMAHA

●アーキテクチャを維持できたのか？

ソースコード品質評価ツール Adqua を使ってソースコードの品質を定量的に評価



Adqua：http://www.ogis-ri.co.jp/product/b-08-000001A6.html

何よりも

YAMAHA

●PDCAを回し続けられた

製品展開毎に開発プロセスを改善

1製品目

↓ 品質は良いが、工数が予想よりも膨らんだ

2製品目 ― 結合テストの最適化

↓ 品質は良いが、更なる生産性の改善が必要

3製品目 ― 単体テストの最適化
結合テストの最適化

↓ 品質は良いが、更なる生産性の改善が必要

4製品目 ― 仕様分析の最適化
単体テストの最適化
結合テストの最適化

今後の課題

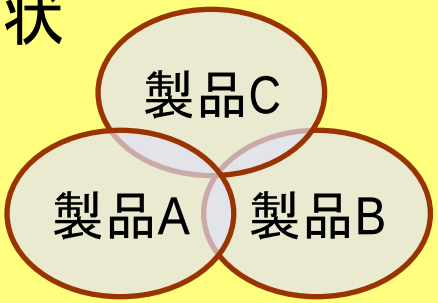
新しい価値の提案に注力する

YAMAHA

●仕様の整理

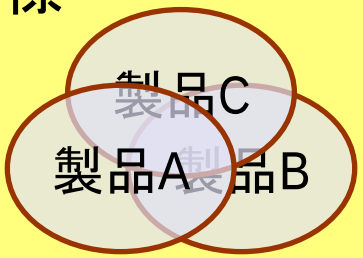
・細かな仕様の違いが多すぎる

現状



仕様を整理して、
本当に必要な仕様を
見極める。

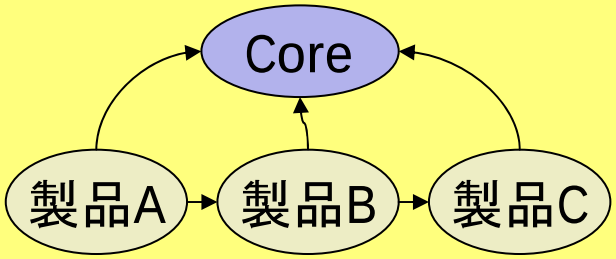
目標



●Core資産を構築する

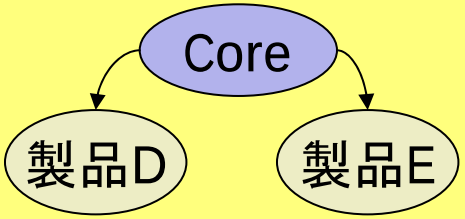
・仕様の違いが簡単に実現できるシステムにリファクタリング

現状



Core資産から
製品を作る

目標



総括

まとめ

YAMAHA

●定量評価


生産性　約1.6倍


品質　バグ密度が約1/3に


アーキテクチャ　保守性・再利用性が5%アップ

●定性評価


製品・ソフトウエア　アーキテクチャを維持できた


組織・人・スキル　他ドメインの知識を習得できた


開発プロセス　改善を繰り返すことができた

ご清聴ありがとうございました

YAMAHA

YAMAHA

感 動 を ・ と も に ・ 創 る