



YAMAHA

感動を・ともに・創る

電子楽器開発における
アーキテクチャ・リファクタリングと
反復型開発の実践
—製品適用における問題・対処と
継続への課題—



- 開発の背景
 - プロジェクト診断と課題抽出
- オブジェクト指向によるシステム再構築①
 - プロトタイプ開発～製品適用(開始頃)
- オブジェクト指向によるシステム再構築②
 - 製品適用(完了)
- 総括

● ヤマハ株式会社

- 弊社は「音・音楽」を中心に展開する企業グループです。
 - オートバイは別会社です。
 - 自動車関係の事例を想像されていた方すみませんm(_ _)m
- 紹介するプロジェクト事例は「電子楽器」を対象としております。



開発の背景



● 開発の経緯

● 約10年前にベースシステム確立

● 機能別にモジュール分割されたシステム完成

● 電子楽器としての基本機能に次々と新フィーチャー搭載



● 急速な市場・技術変化に対し

「休みなく機能追加し続けることで」対応

- 従来の開発の問題点
 - モジュールの複雑化
 - システムデザインを無視したパッチワーク的機能追加
 - プログラムのビッグバン
 - 100万step以上のプログラムに仕様が隠蔽
 - 設計文書と実装の乖離
 - メンテナンス不可能な虫食い状態の設計文書

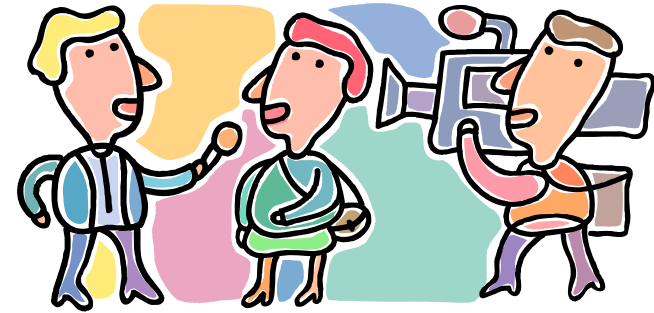
再利用性・開発効率低下が顕著

→機能追加もままならない状況に陥る

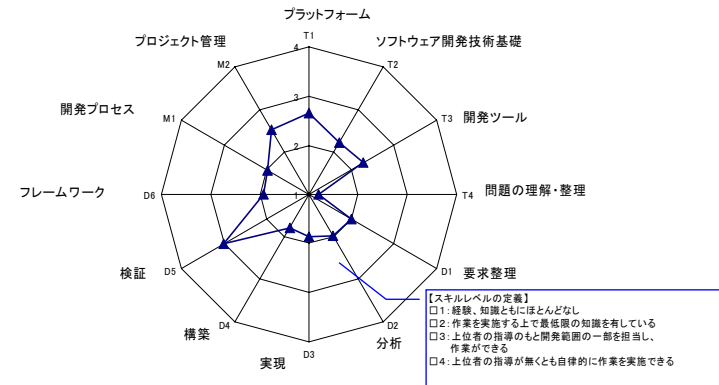
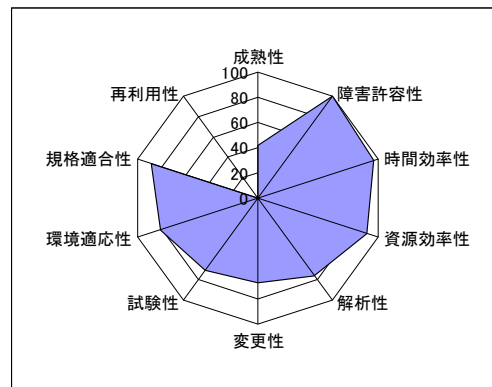
プロジェクト診断と課題抽出



- プロジェクト診断内容
 - プロジェクト概要ヒアリング
 - 関係者インタビュー
 - ソースコード品質診断
 - 開発者スキル診断

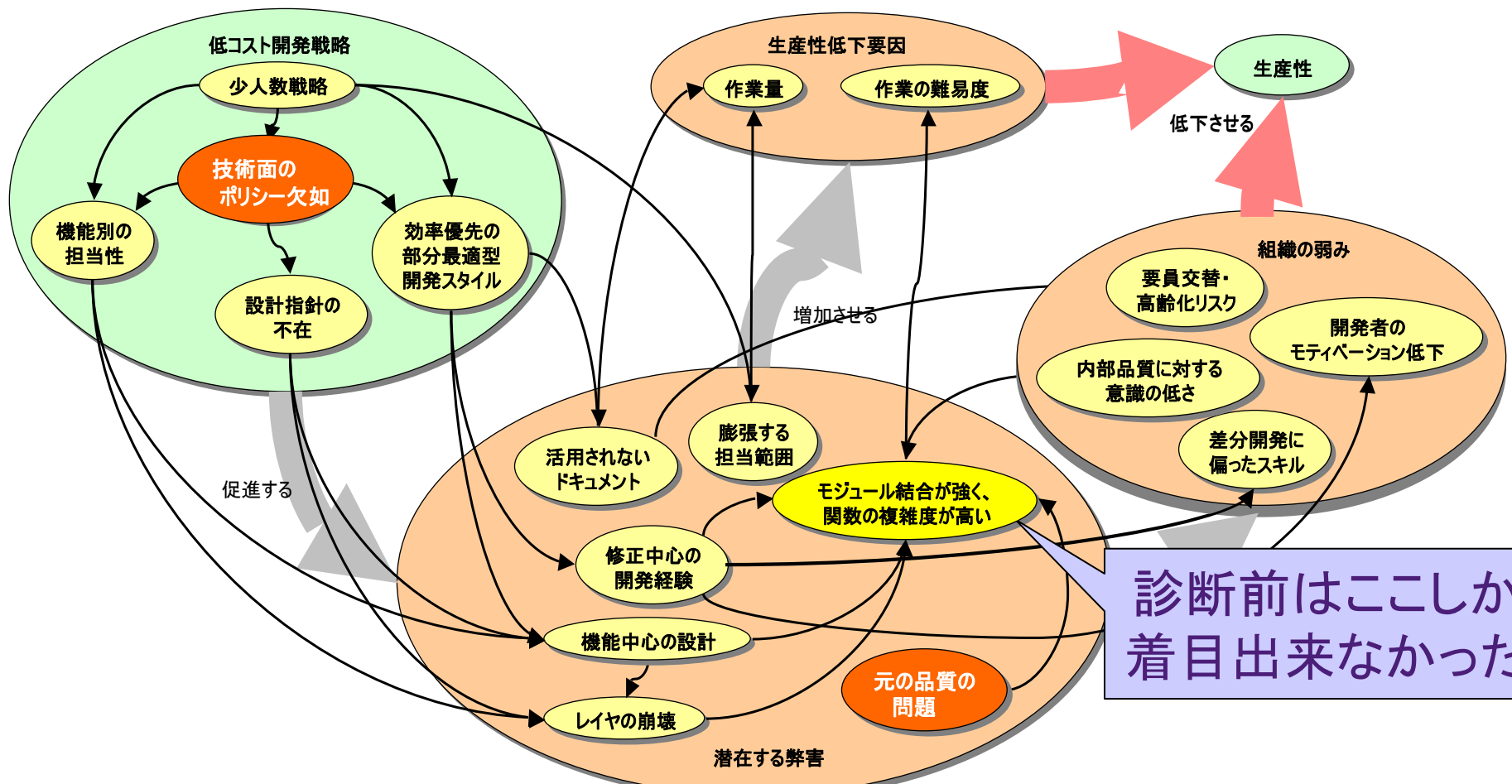


全体のスキルレベル



潜在的な問題原因の発掘

プロジェクト診断/原因把握



根本原因は「アーキテクチャポリシーの欠如」と「元のソフトウェアの品質」

- 技術課題
 - オブジェクト指向で要求を分析、本質を抽出
＝アーキテクチャ再構築
 - 再構築されたアーキテクチャをコア資産として再利用
- 組織課題
 - アーキテクトチームの新設
 - 機能単位から特定の知識領域(ドメイン)単位へ
担当割を移行
- プロセス課題
 - 分析重視の反復型開発
→アーキテクチャ洗練 & 全体構築

オブジェクト指向による
システム再構築①
—プロトタイプ開発—



● 進め方

① 少人数でプロトタイプを作り上げ、徐々に組織内への浸透を図る

- 自社内の人員ではオブジェクト指向分析のスキル不足
 - 段階的に分析から詳細に落としていくプロセスの経験がない
- オージス総研様に技術指導を依頼

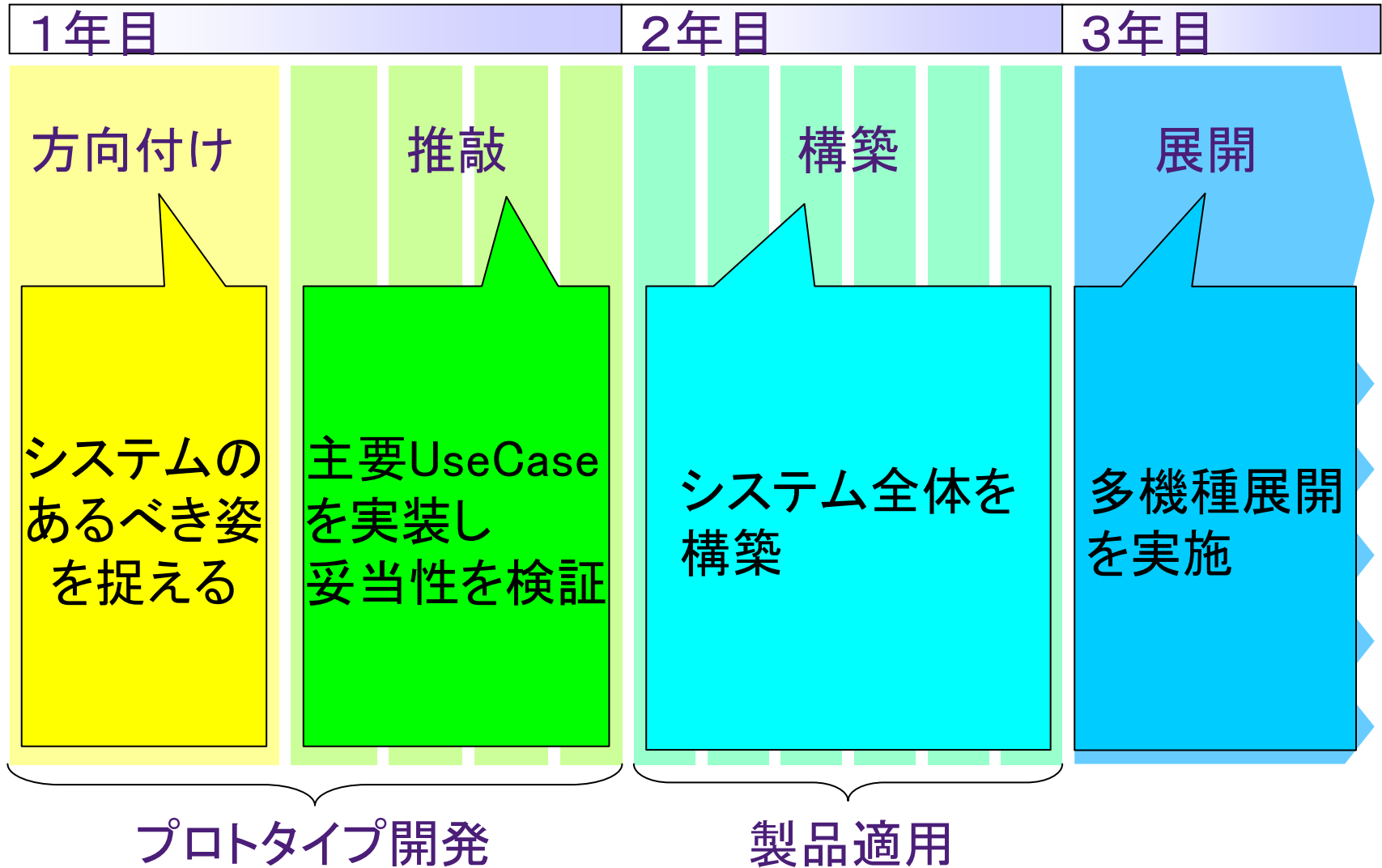
② プロトタイプを元に一製品へ適用

- 徐々に自社での自立開発へ移行

● 人員

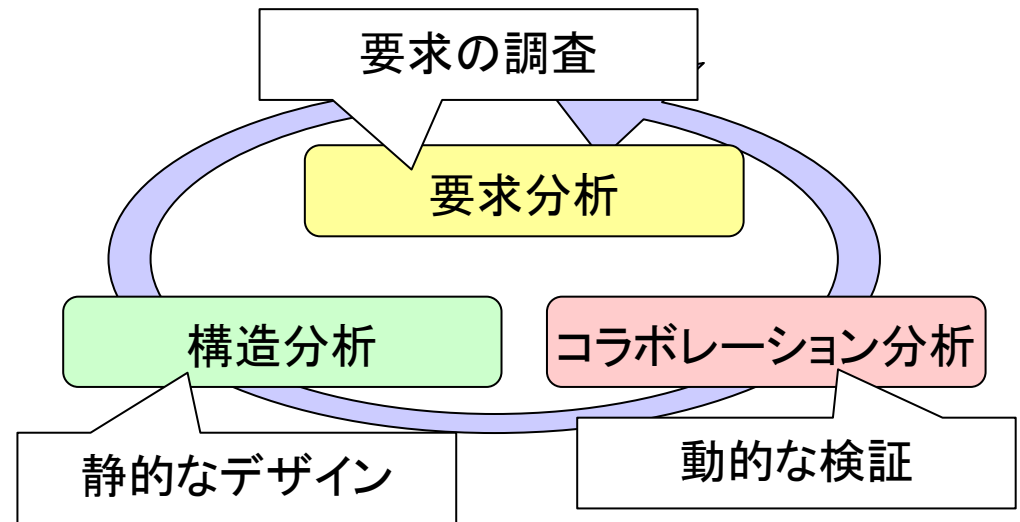
- ヤマハ社員数名～、OGIS総研様1名

活動ロードマップ



- 「電子楽器」の本質的な概念抽出
 - でも分析ってどうやったらいいの？

コンサルタントが
分析の基礎から
成果物まで指導



今までの機能分割から脱皮し
「関心の分離」=ドメイン分割に辿り着く

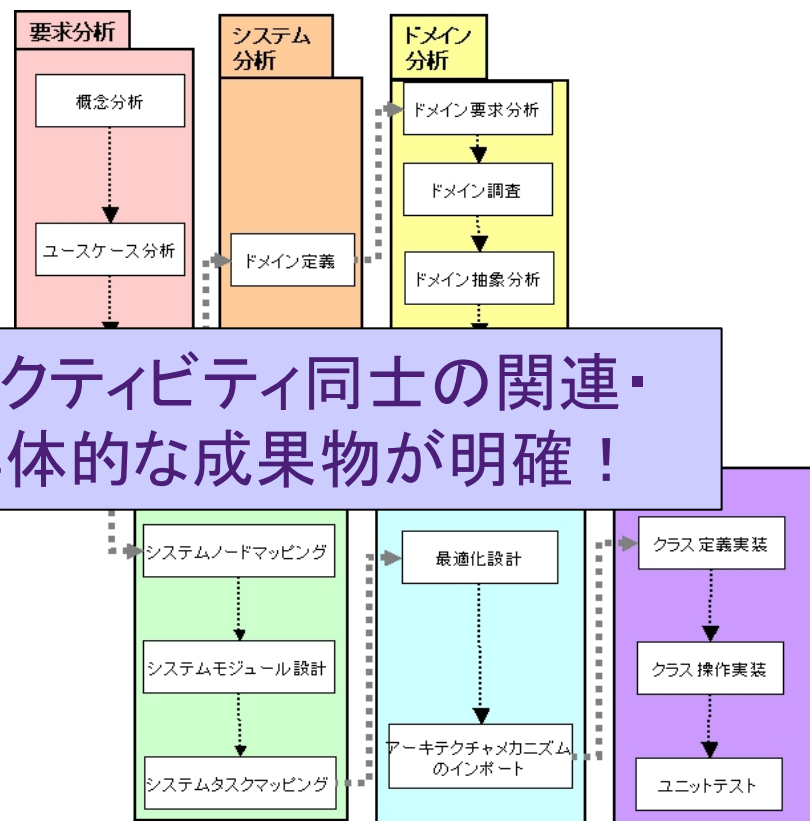
- 「電子楽器」としての基本構成確立

- でも分析モデルを実装まで落とす
ワークフローが分からない！

コンサルタントが
「e-UML」のプロセスを
テーラリング

モデルから実装まで
一貫した開発を実現

アクティビティ同士の関連・
具体的な成果物が明確！



- 技術面の成果
 - 基本概念に基づくアーキテクチャを確立
 - モデリングツールを利用し、分析～設計～実装が一致
- 組織面の成果
 - アーキテクトチームとドメイン担当制導入
- プロセス面の成果
 - 分析:設計実装=1:1の反復型開発が身に付く
- 課題
 - ドメインごとに更なる分析要素が発覚

- 製品適用のためシステム全体を構築
 - 自社で開発プロセスを回す
 - 自立開発に向かう
- オージス総研様コンサルタントはこの事例を他開発チームへ展開するため異動
 - 定期的にアドバイザーとして参加

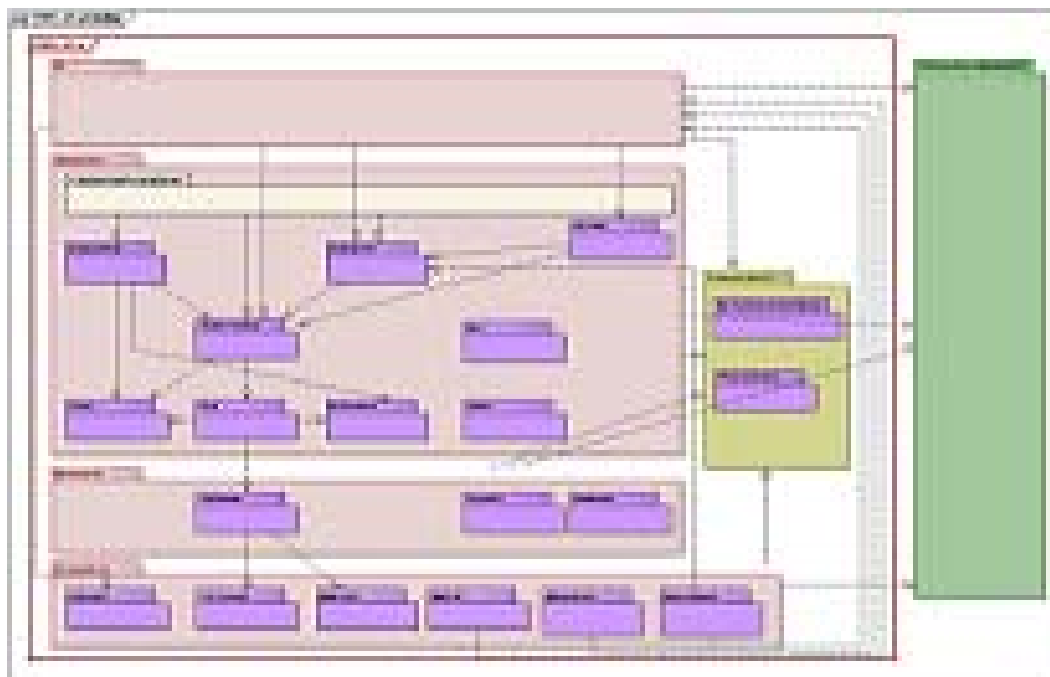
- 技術面の成果
 - アーキテクチャを維持したまま機能拡張を実現
 - 分析～実装の一致により変更対応箇所が把握可能
- 組織面の成果
 - ドメイン単位の担当割
 - 課題対応が局所化し、現場の混乱がなくなる
- プロセス面の成果
 - 開発者の意識がリスク駆動開発へパラダイムシフト
高いリスクを早期に発見・解決
＝自立開発実現

オブジェクト指向による システム再構築②

—製品適用—



- 構造重視を継続
 - ドメインの責務の妥当性に焦点
 - 以前の開発の反省から『構造が崩れないこと』を最重視



× 詳細箇所の完成度が中々高まらない

- 将来の変更を予期しながら、ドメイン構造の検討継続
 - 計画上、かなり先の内容の検討が含まれることも...
- テストケースの粒度が荒い

× 課題の累積

- 期間重視でイテレーションを実施
 - 解決できない課題は次のイテレーションに持ち越し

このまま進めていて大丈夫？

オージャス コンサルタント様に復帰、分析いただくと...

× 構築フェーズになっていない

- メンバーのマインドが推敲フェーズの延長線上
 - 構築フェーズとはどうあるべきか？の知識不足
 - 推敲フェーズと異なり、ドメイン毎進度はマチマチ

× 品質保証が弱い

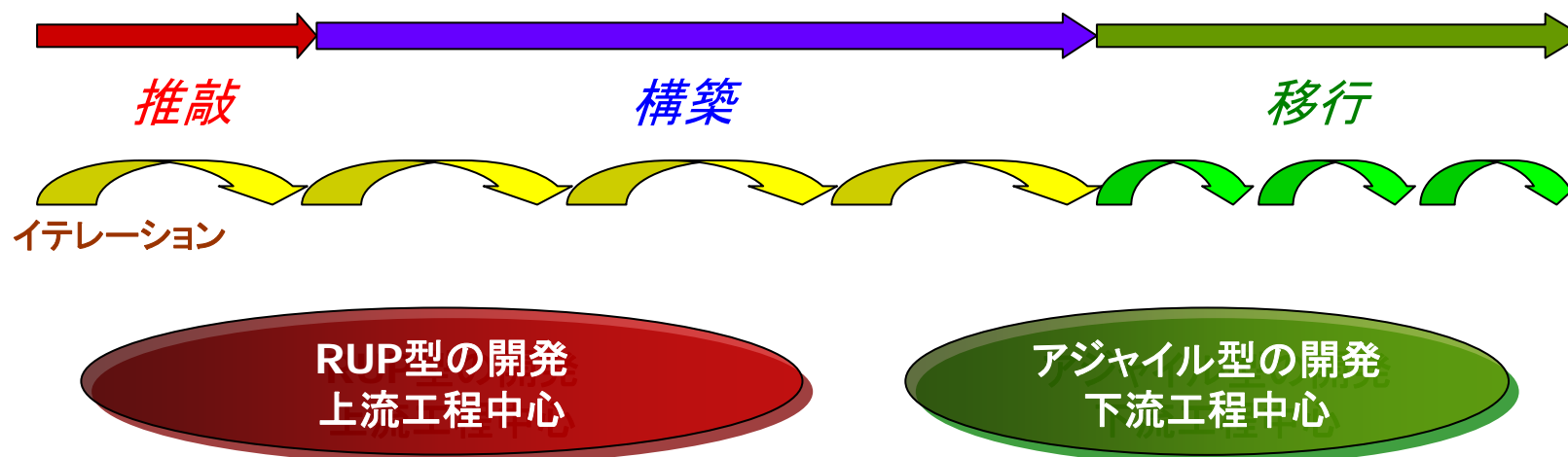
- 期間優先のためテスト時間が不足
- 単体テスト環境が未整備
- 結合テストケースの粒度が荒い

✓ タスクの状況に応じたプロセスを適用

上流工程中心のアプローチは、アーキテクチャの構築段階で有効

下流工程中心のアプローチは、詳細仕様やパフォーマンス仕様を作りこんでいく段階で有効

両方を適材適所に使い分け



✓ 品質向上施策の実施

- 単体テスト環境整備
 - UnitTest
 - スタブ作成ツール準備
- テストケース合理化
 - All Pair法などを参考に
- 回帰テスト実施
 - 品質低下を早期に検出

✓ 再計画

- 累積課題の解消に特化したイテレーションを追加実施

✓ 品質保証アクティビティの明確化



- 品質
 - 別部隊の従来開発と同等の品質を確保
- パフォーマンス
 - 通常使用では問題無し
 - 一部に改善の余地はある
 - パフォーマンスチューニング箇所の局所化
 - データ中心設計による
 - メモリ確保・解放関連のチューニングも重要

総括



- 技術面、組織面
 - 機種適用開始前と比べ、表立った問題は無し
- プロセス面
 - イテレーションの方向性に応じたプロセスの調整
 - 計画時点の認識が必要
 - 品質保証
 - 改善効果を計測可能とする指標の模索
 - 自動実行などのテスト効率化

プロセス面の継続的な改善が今後の課題

御清聴ありがとうございました





YAMAHA

感動を・ともに・創る