

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

— 診断を契機にした、改善から改革への
方針転換とその後の活動報告 —



- 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
 - プロトタイプ開発
 - 方向付け
 - 推敲
 - 製品適用
- 総括

弊社紹介



- 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。
 - オートバイは兄弟会社になります。
 - 今回の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



開発の背景



- 開発の経緯

- 約10年前にベースシステム確立
- 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
 - 膨大な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

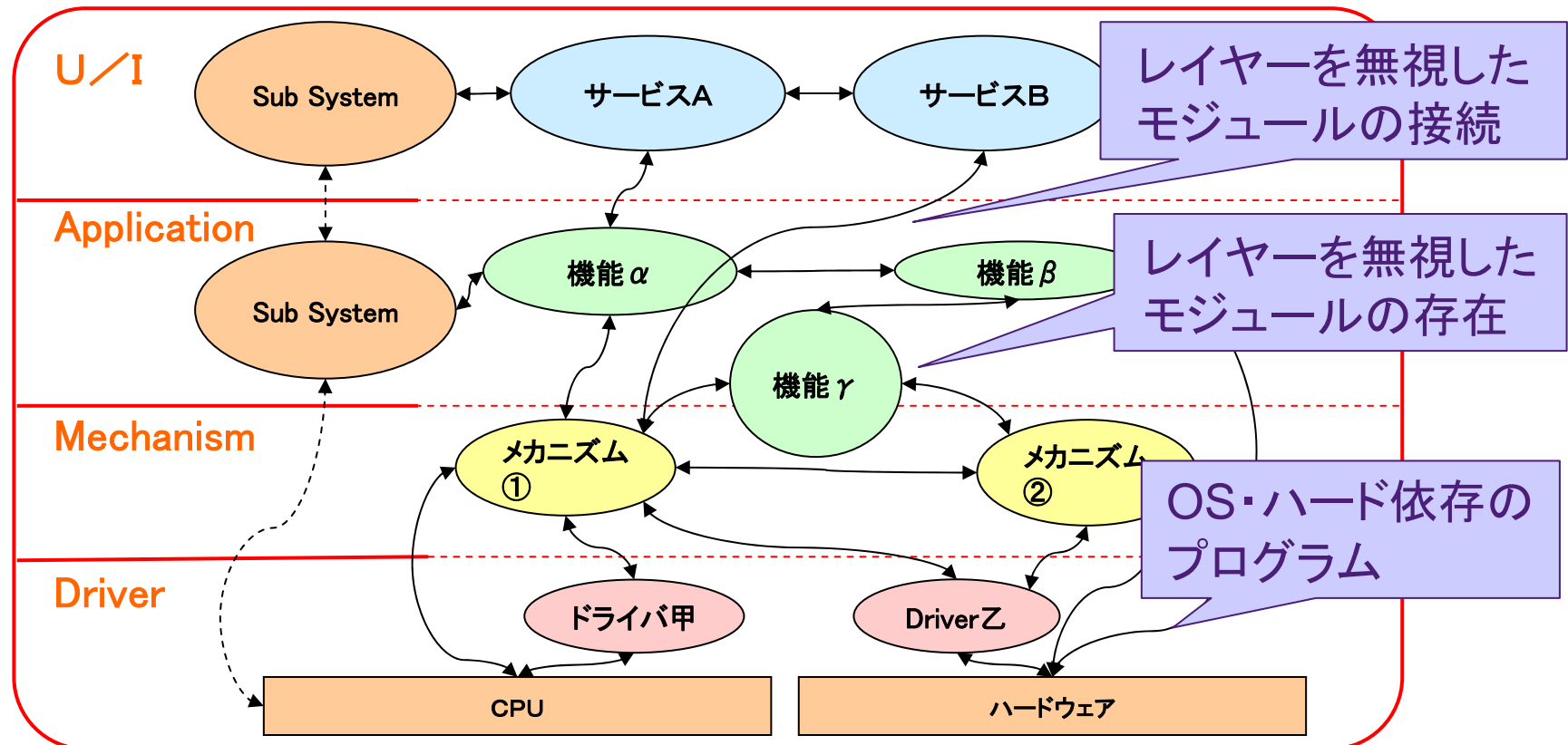
再利用性・開発効率低下が顕著

→機能追加もままならない状況に陥る

開発の背景・フレームワーク再建



- 自社内での問題分析
 - 問題は「分割統治が崩れている」ことにあると認識
 - レイヤー構造を強く打ち出したフレームワーク開発で解決を試みる



- フレームワーク開発
 - 約半年でフレームワーク完成
 - 適用しようとしたが…
 - フレームワークへの不信感
 - システム大変更に抵抗
 - どこから手を付けるべきか悩む



オージス総研様に
新フレームワーク&プロジェクトの診断を依頼

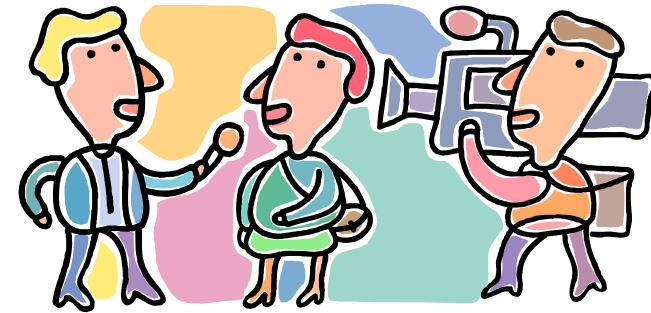
プロジェクト診断と課題抽出



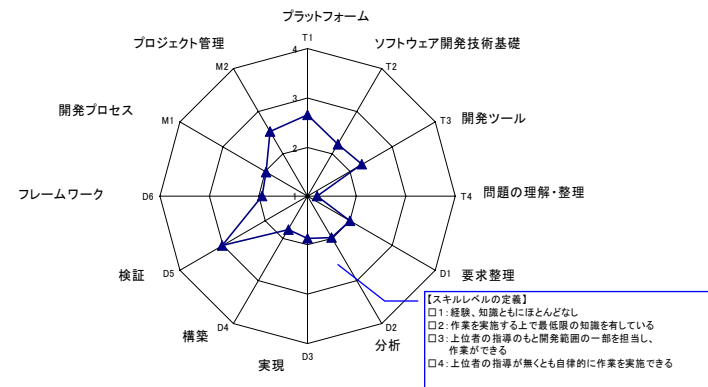
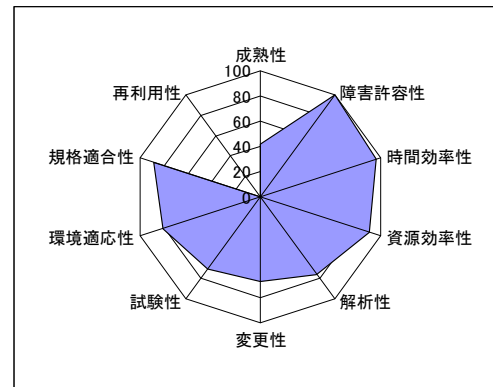
プロジェクト診断



- 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

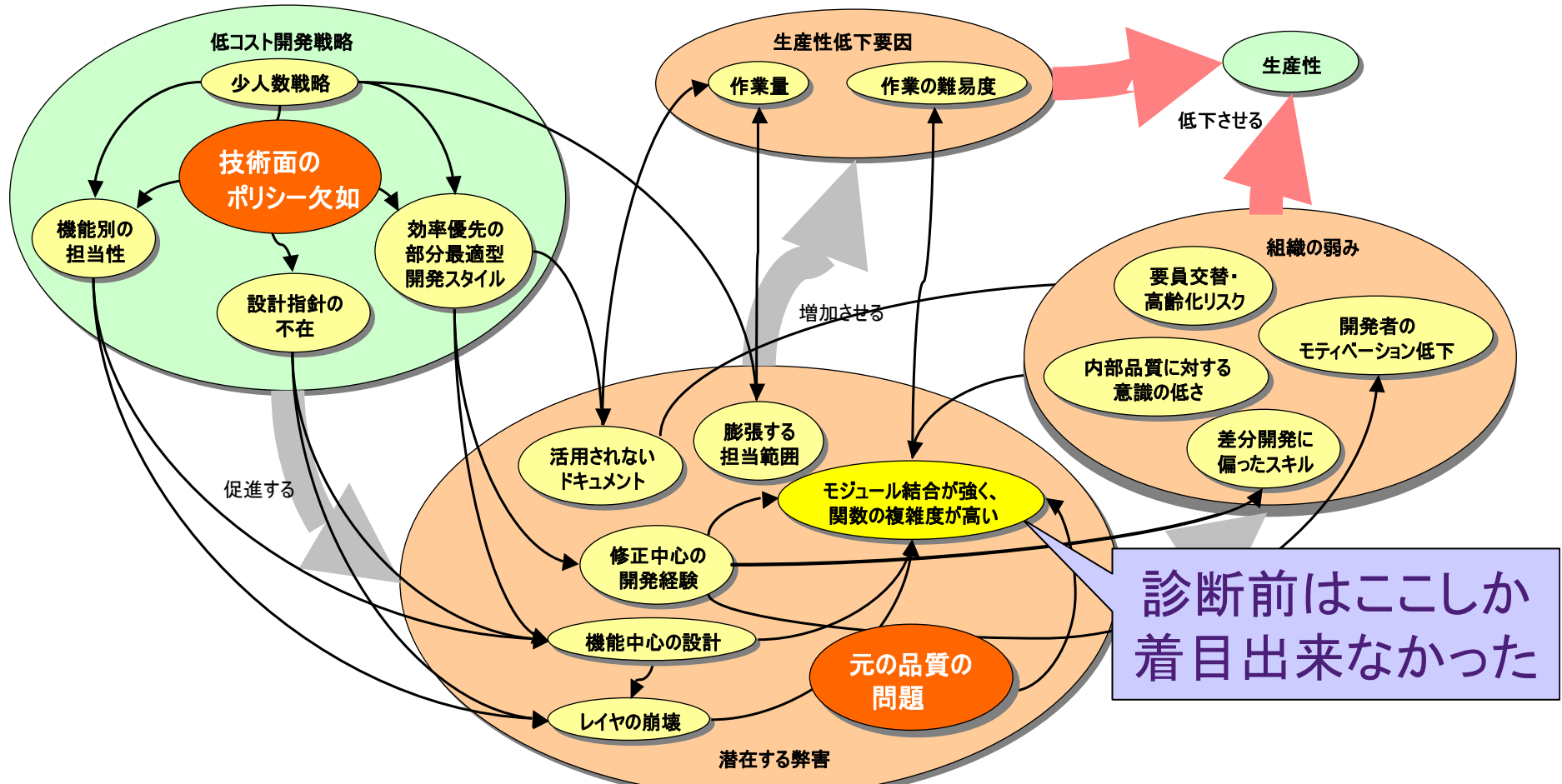


全体のスキルレベル



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」

- 診断結果
 - 潜在する根本原因から解決が必要
 - フレームワーク適用だけでは短期的な改善に留まる
- 「アーキテクチャポリシーの欠如」
 - 流用を重ねシステムアーキテクチャが不鮮明
 - 「電子楽器」特有のシステムアーキテクチャを抽出しアーキテクチャの再定義が必要
- 「元のソフトウェアの品質低下」
 - 設計品質を維持するための「秩序ある開発体制」が必須

- 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



- 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
- オージス総研様に技術指導を依頼

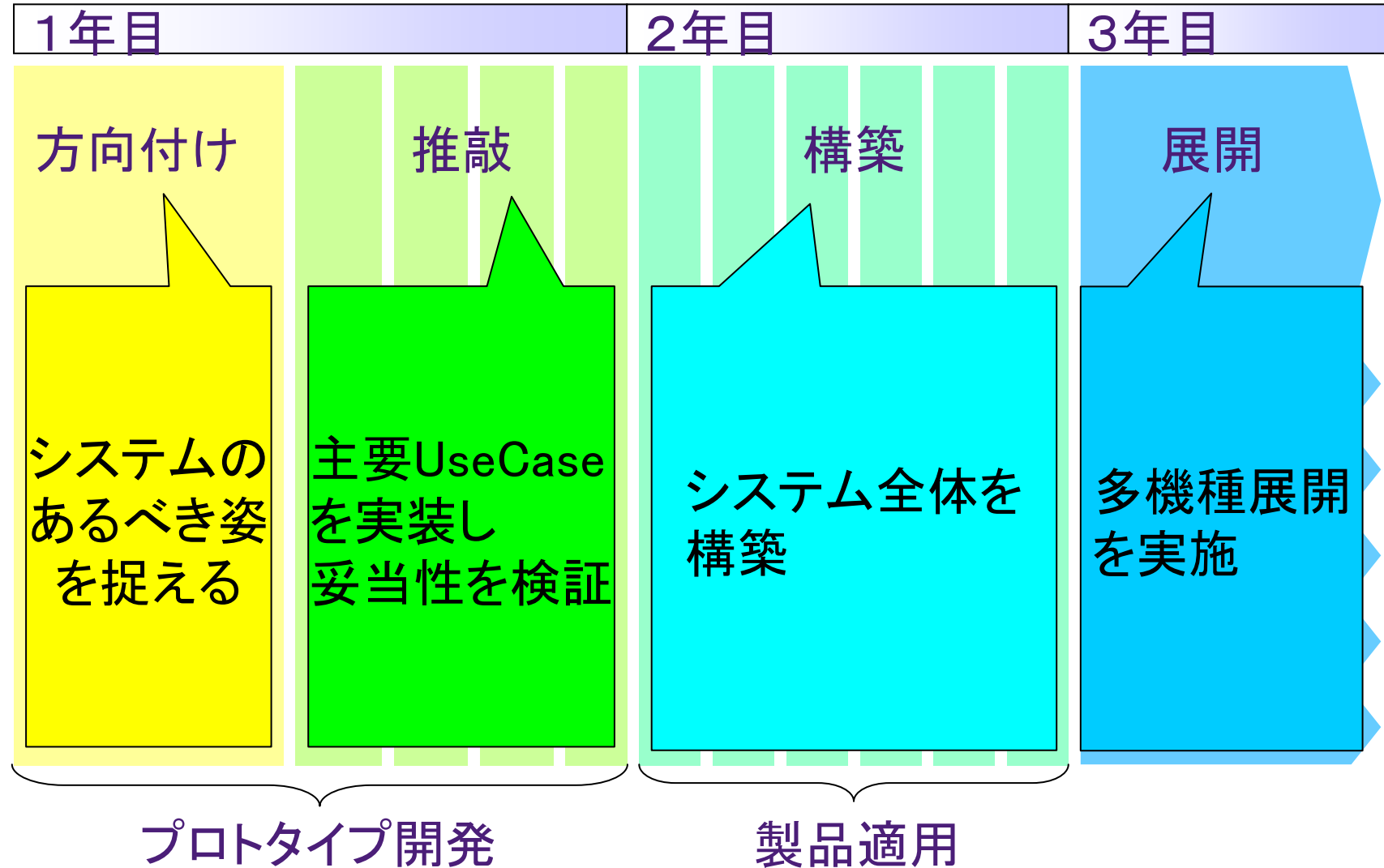
- ② プロトタイプを元に一製品へ適用

- 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

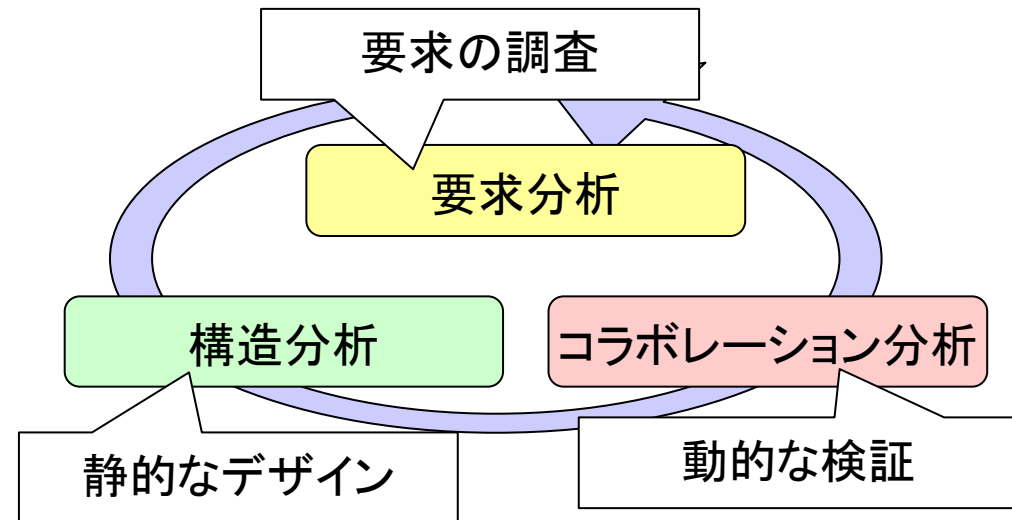


プロトタイプ開発/方向付け



- 「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

プロトタイプ開発/推敲

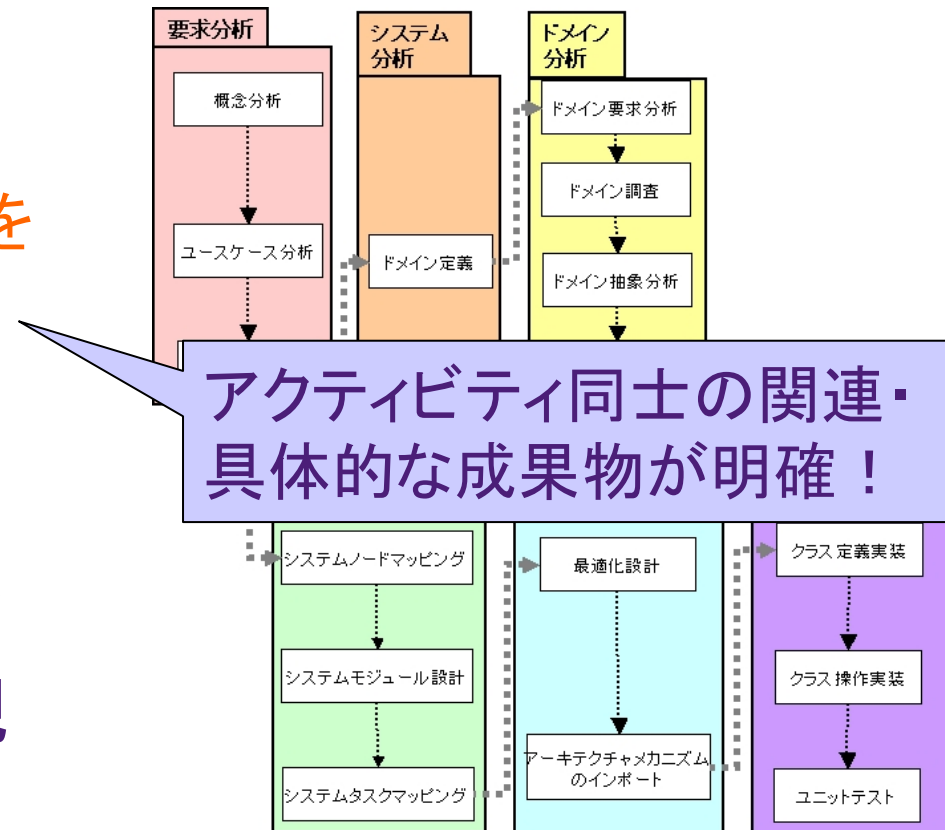


● 「電子楽器」としての基本構成確立

- でも分析モデルをどうやって実装まで落とすの？

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



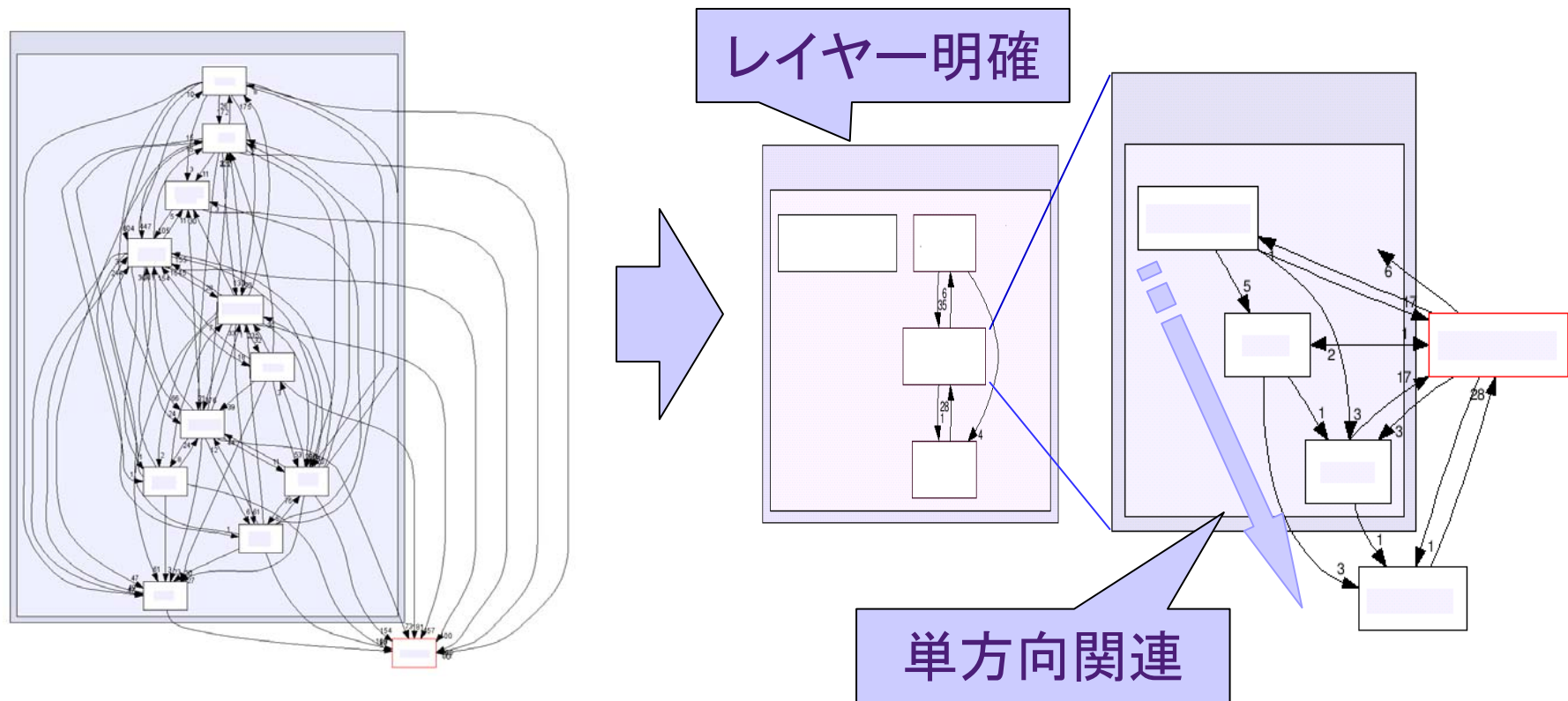
- 技術面の成果
 - 基本概念に基づくアーキテクチャを確立
 - モデリングツールを利用し、分析～設計～実装が一致
- 組織面の成果
 - アーキテクトチームとドメイン担当制導入
- プロセス面の成果
 - 分析:設計実装=1:1の反復型開発が身に付く
- 課題
 - ドメインごとに更なる分析要素が発覚

- 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

構造面の効果

- 改善前と改善後の構造解析結果
 - DoxygenによるDirectory Dependencies出力結果



品質面の効果



- 改善前と改善後の品質測定結果
 - 静的解析ツールQACにてメトリクス集計

ケース	改善前	改善後
A社	71.19	80.42
E社	67.69	85.20
G社	71.10	-
J社	53.47	-
YAMAHA	70.46	84.27

総括



あの時診断を受けなかったら・・・



アーキテクチャ	単独/分散/統合	オージス総研様コンサルティング 本質を抽出した知識領域分割
トレーサビリティ	コン え	要求から実装まで一貫
仕様可視性	eUMLをテーラリングした 詳細なアクティビティと	詳細仕様まで掘り起こし文書化
組織構造	成果物指導	複数のアーキテクトが各ドメイン を分担するメッシュ型
プロセス	PM 診断による組織的な 問題への気づき	アーキテクチャが共有されており 代替が利く
品質	ウ 終 益 スマ ア	反復型開発でリスクの積み残し 最小限
品質	品 ま	定期的な品質診断で 高水準維持
スキルアップ	全体がゆっくり持ち上がる	反復型開発指導 アーキテクトがエバンジェリスト となり一気に持ち上げる

- 技術面
 - オブジェクト指向開発の実践
 - UML/OOによるモデルベース開発を修得・分析能力向上
 - 要求から実装までのトレーサビリティ確保
 - 仕様変更への対応スムーズ
- 組織面
 - ドメイン担当制によるアーキテクチャ&品質維持
 - 人的な分割統治でアーキテクチャの崩壊を防ぐ
- プロセス面
 - 反復型開発によるリスク早期解決
 - リスクの早期顕在化とフィードバックループ

複数の施策の相乗効果を実感