アーキテクト育成の取組み事例

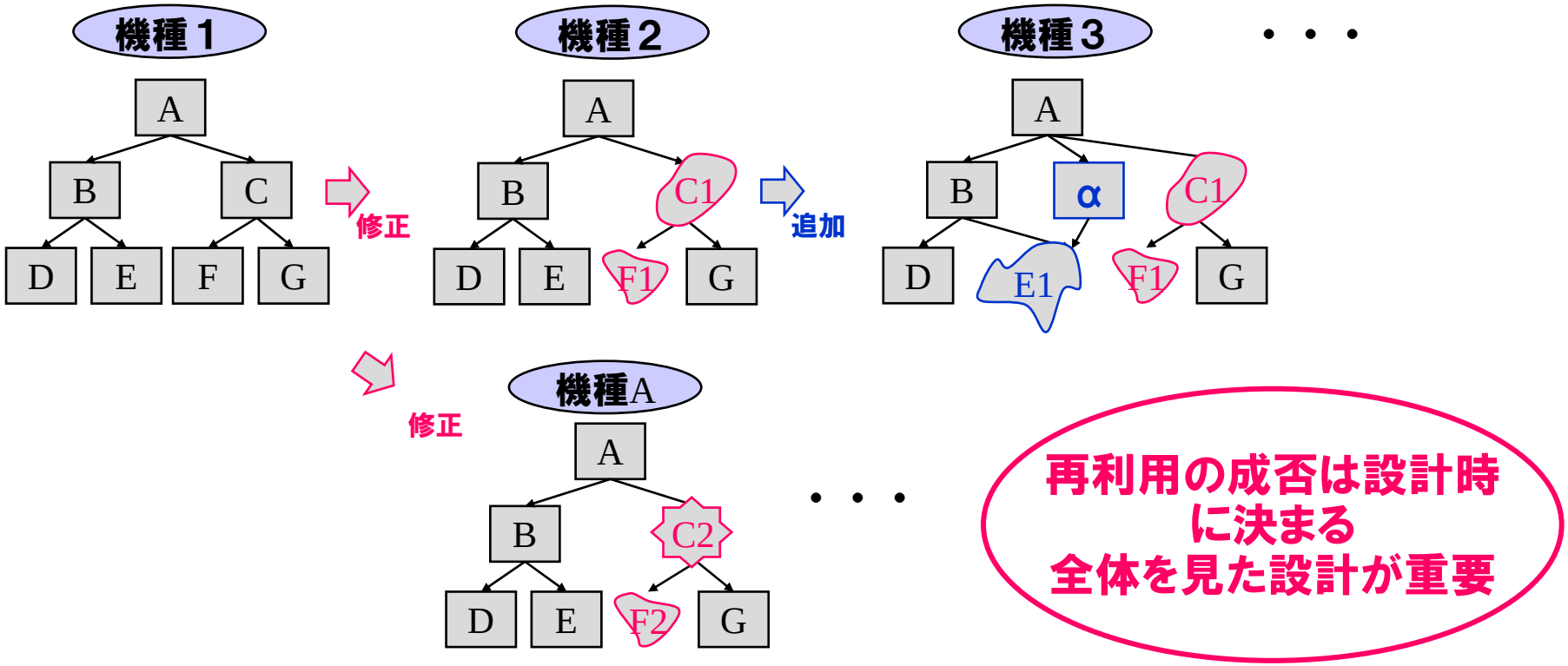
2011年10月18日

パナソニック株式会社
システムエンジニアリングセンター
春名　修介

アーキテクトの必要性

よくある事例：ほんとうに再利用？

■ 作ってからの再利用　⇒　アドホックな~~再利用~~　⇒ 流用



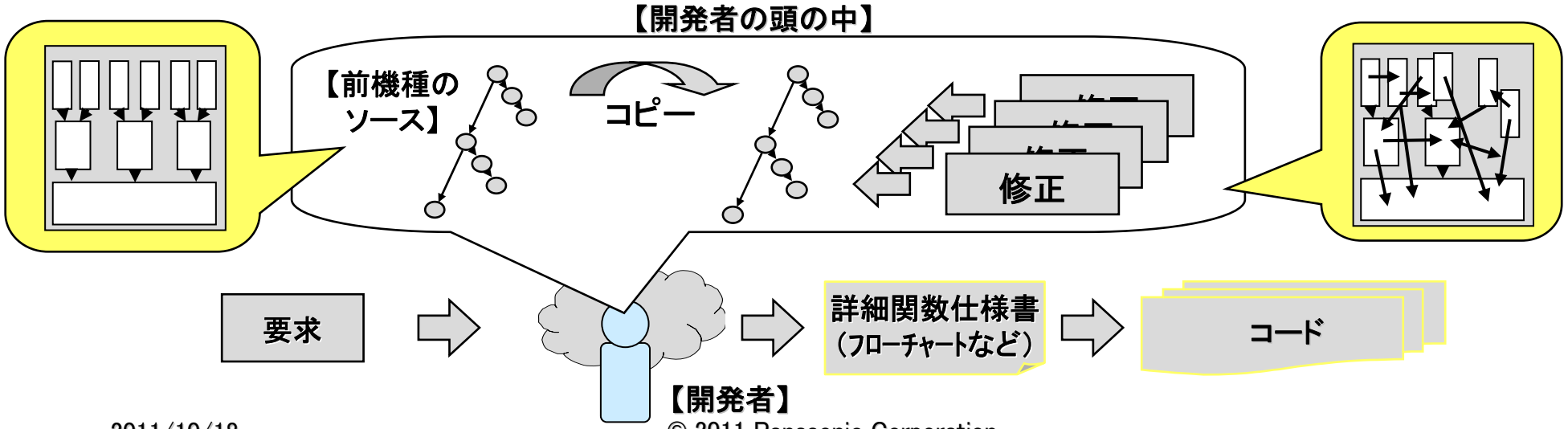
派生部品　C，C1，C2，E，E1，F，F1，F2

⇒　似て非なる部品が山とできる．どれがベースラインなのか不明

また，#ifdefの山となり，開発者以外管理不能

流用開発の弊害

- 以前の機種のソフトウェアをコピーし，必要な部分のみを擦り合わせ的に修正・追加
- 設計意図・全体構造が不明確になり，不適切な修正による構造劣化が進行．規模の拡大共に開発効率・品質が低下
- 大きな機能変更が不可能になり，製品の競争力も低下
- 継続すると，開発者の設計力が低下
  - 特に，全体を見たアーキテクチャ設計ができなくなる

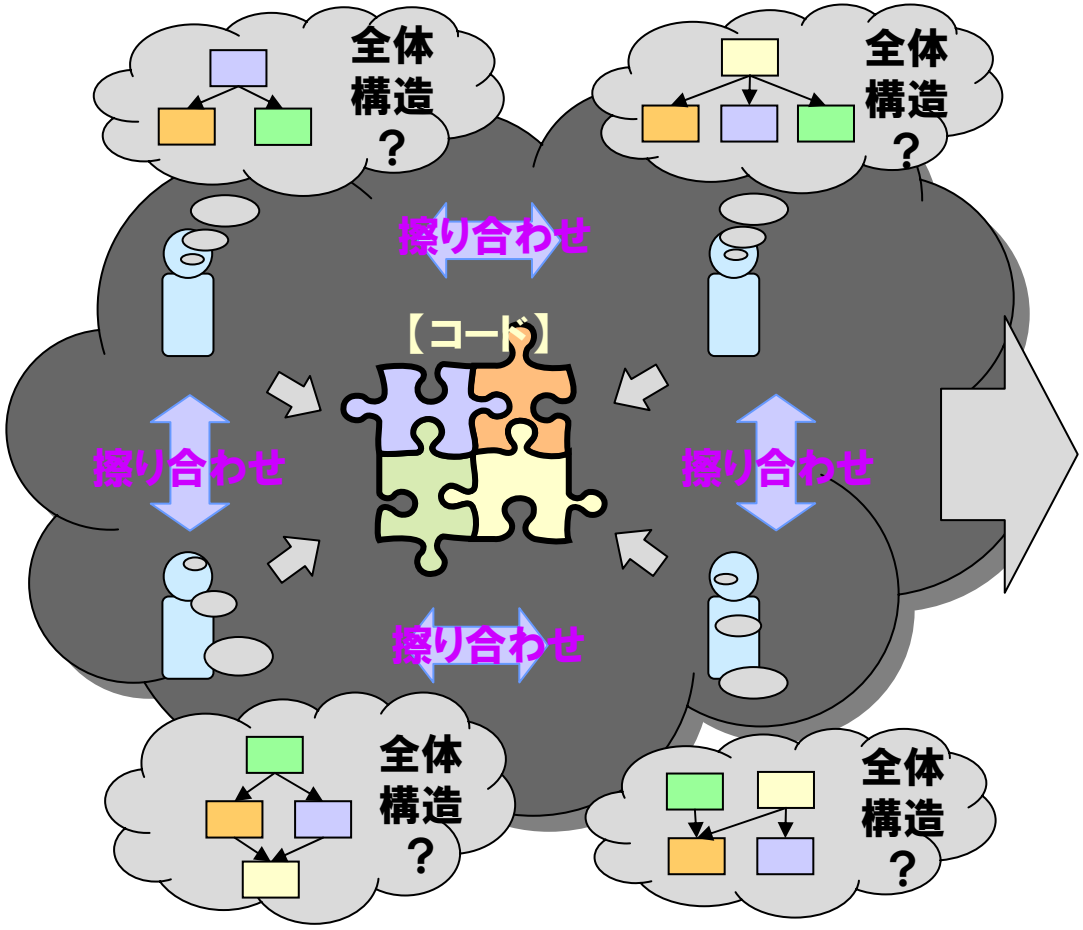


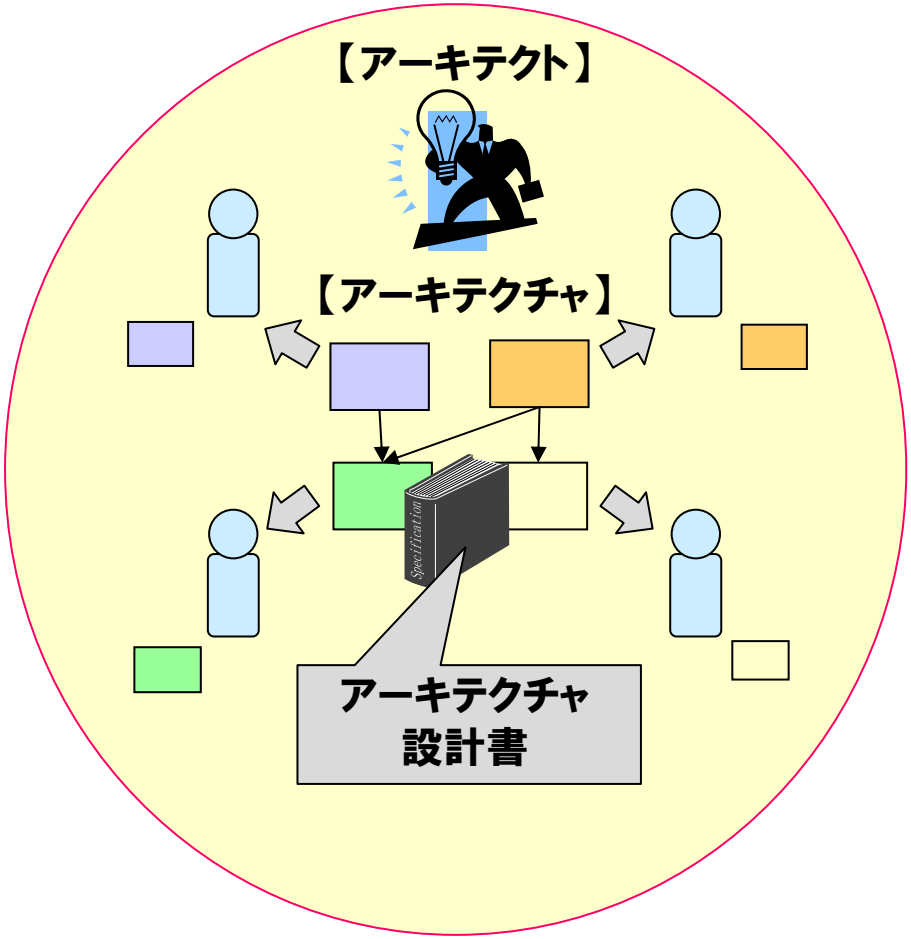
全体が把握できていないことに起因する問題

- **不明確な設計意図**
  - なぜ，現状のコーディングになっているのか，設計方針が分からない
  - コードレベルでの修正で対処してしまうため，構造劣化が進行する
- **不明確な全体構造**
  - 修正の影響範囲が分からず，影響範囲特定のコード解析に時間がかかる
  - どの部分が再利用の範囲か不明確であり，流用は行っているが生産性は低下している
  - インタフェース不整合が多発し，都度打ち合わせで解決している
  - 外部委託時に請負型で委託できない
- **技術が形式化されていない**
  - 他者が理解できるドキュメントが書けない
  - 暗黙の了解が多く，技術・設計情報が組織資産となっていない
  - 新規参加者の即戦力化が難しい

解決策

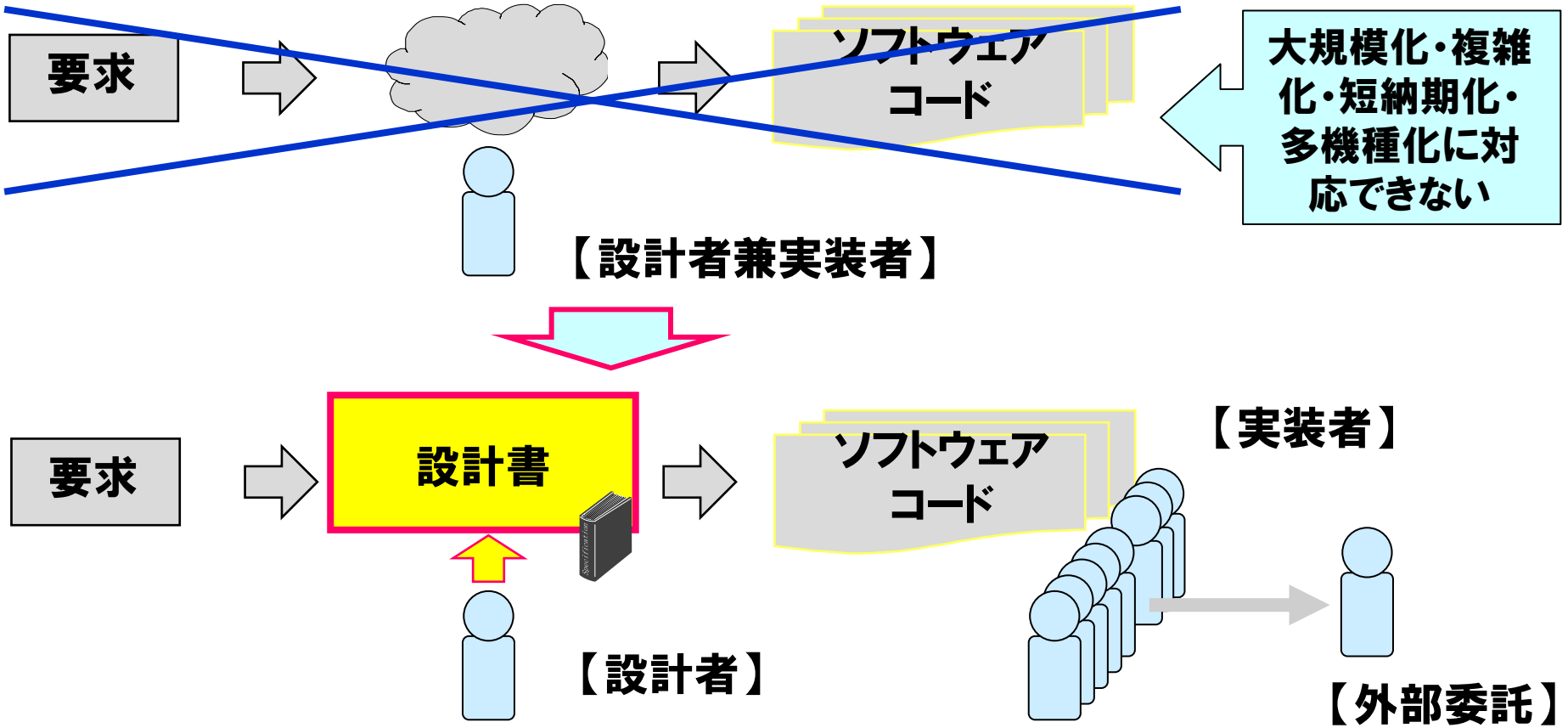
- コードレベルの擦り合わせから，全体構造（アーキテクチャ）を明確にした開発へ





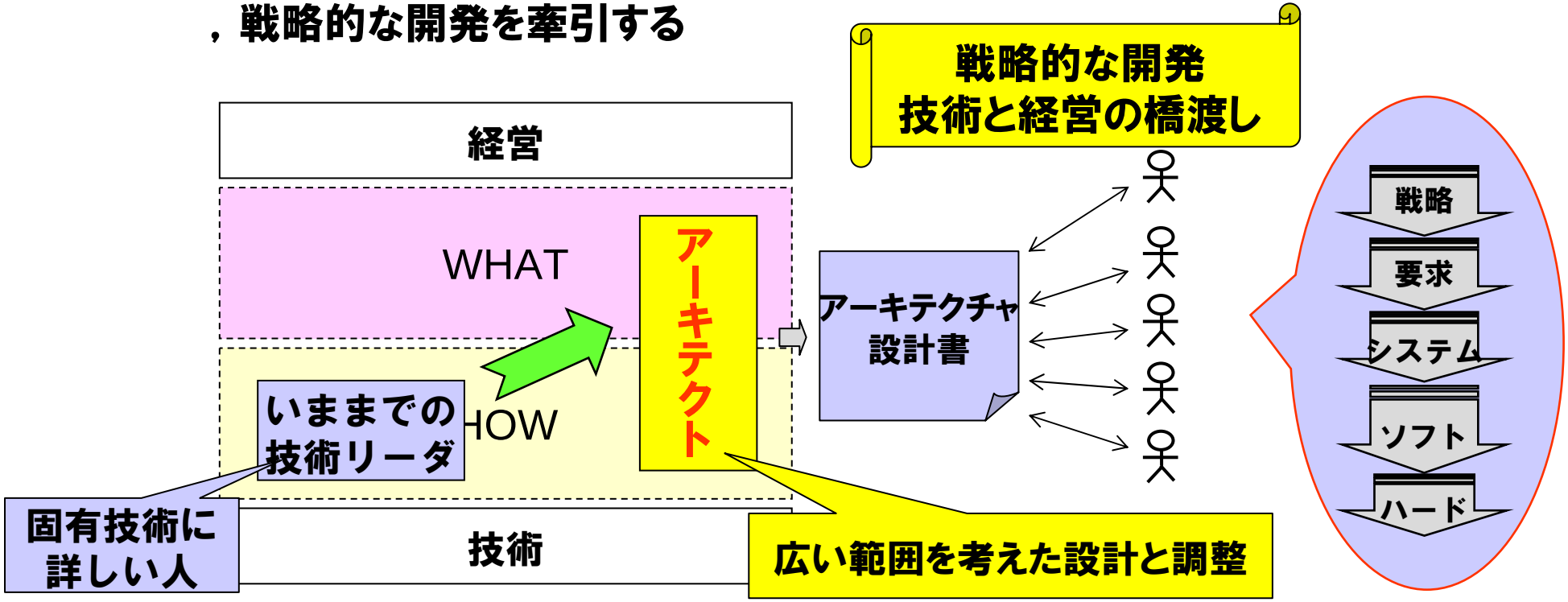
表現することの重要性

- 設計力とドキュメンテーション力は表裏一体
  - コードは大切．しかし，コードだけで設計の意図を伝えることはできない．特に，ソフトウェアの全体を把握することはできない．



アーキテクトの人材像

- 全体の構造設計ができ，設計意図を他者に伝えることができ，
- 『何を，どのように，作るのか』を正しく判断し，
- 広い視点・上位視点を持つ人材
  - アーキテクチャ設計書を共通ツールとして，様々な利害関係者と調整し，戦略的な開発を牽引する



アーキテクト育成研修とスキル認定

ソフトウェア・アーキテクチャ設計のレベル定義

ソフトウェア・アーキテクチャ設計プロセス

事業戦略

目論見

重点

設計方針

設計

静的構造

動的構造

実装構造

周知

ドキュメント

予測

維持

レベル３（上級）

製品開発の場でソフトウェア・アーキテクチャ設計の経験があり，開発を成功に導いた実績がある．技術の将来予測ができ，後進の育成ができる．

レベル２（中級）

設計意図を他人に伝えることができる

製品の目論見の整理

設計方針の立案

優先順位

根拠の明示

文書構成力
視点の広さ

レベル１（初級）

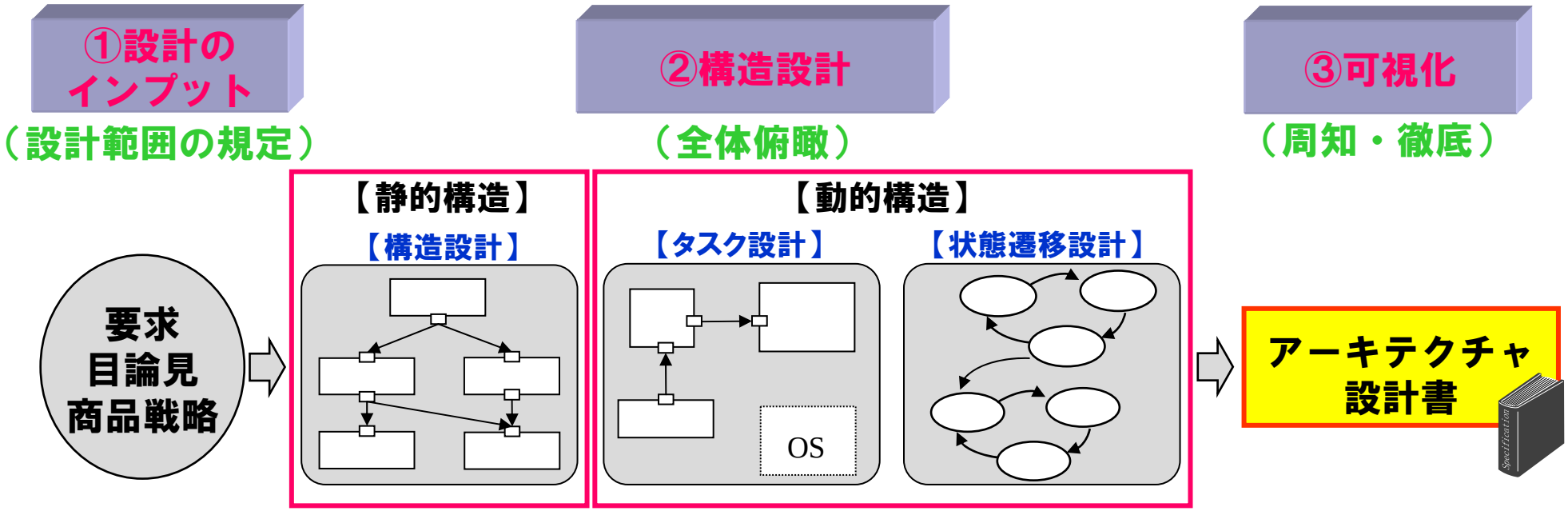
個々の設計手法を組み合わせて構造（コードより抽象度の高い）の設計ができる

構造図，タスク図，状態遷移図が書ける

組込みソフトウェアにおけるアーキテクチャ設計

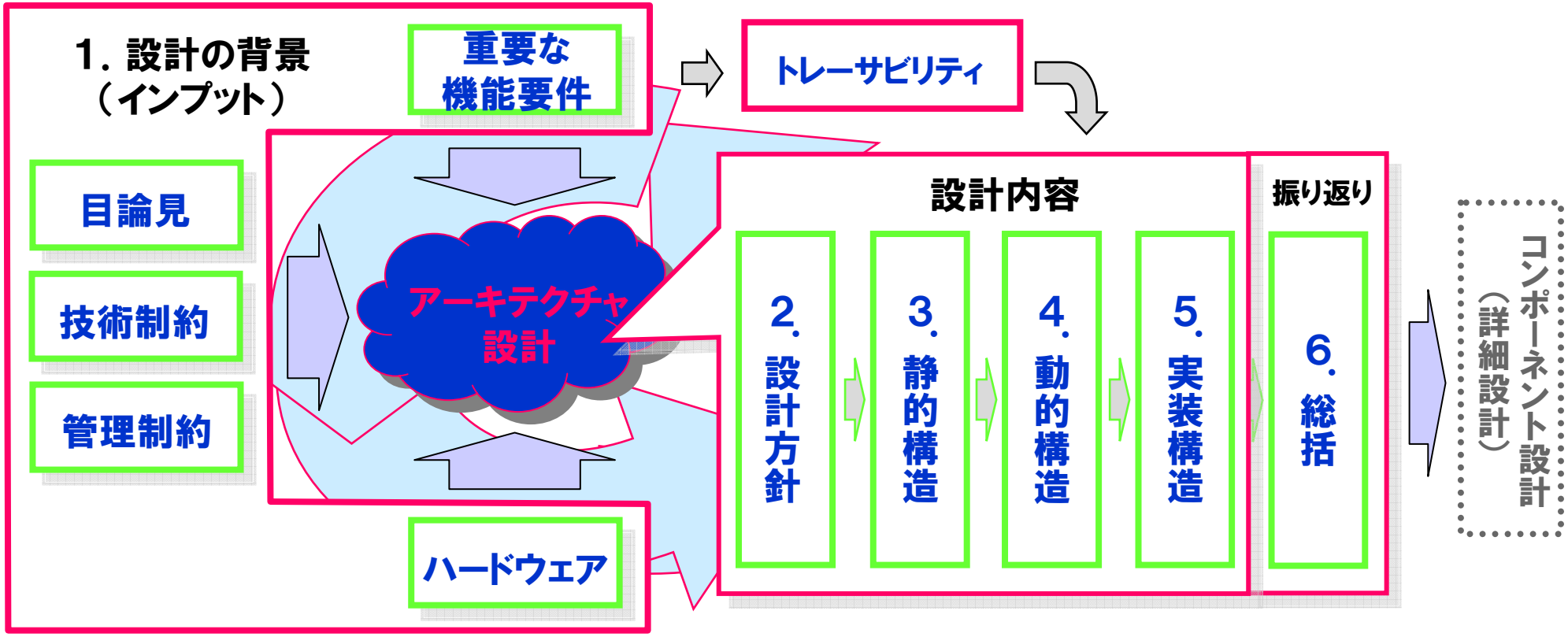
- 静的視点と動的視点の両立が重要
  - 静的視点：構造化設計，OO設計など
  - 動的視点：タスク設計，状態遷移設計など
    - » 静的構造についての方法論は多い．タスク設計・状態遷移設計については属人的で明確にはなっていない．技術伝承が途絶えている？

➢ **組込みソフトウェアは，静的構造設計だけでは動かない．**



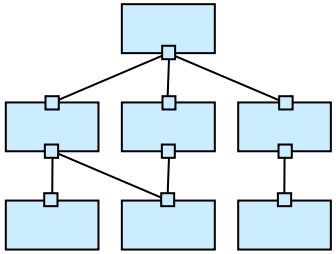
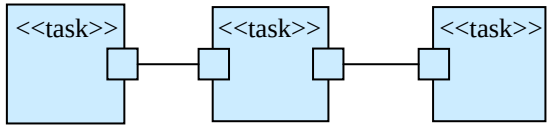
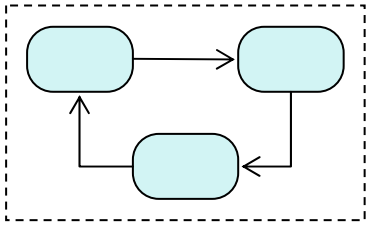
アーキテクチャ設計書記載項目の規定

- アーキテクチャ設計への理解不足，設計書への記載内容の認識も各自バラバラな状況を改善するために，記載項目を規定
  - 構造設計内容の記載に加えて，設計のインプット情報の記載を重視
  - 各項目間のトレーサビリティ確保の方法を具体化



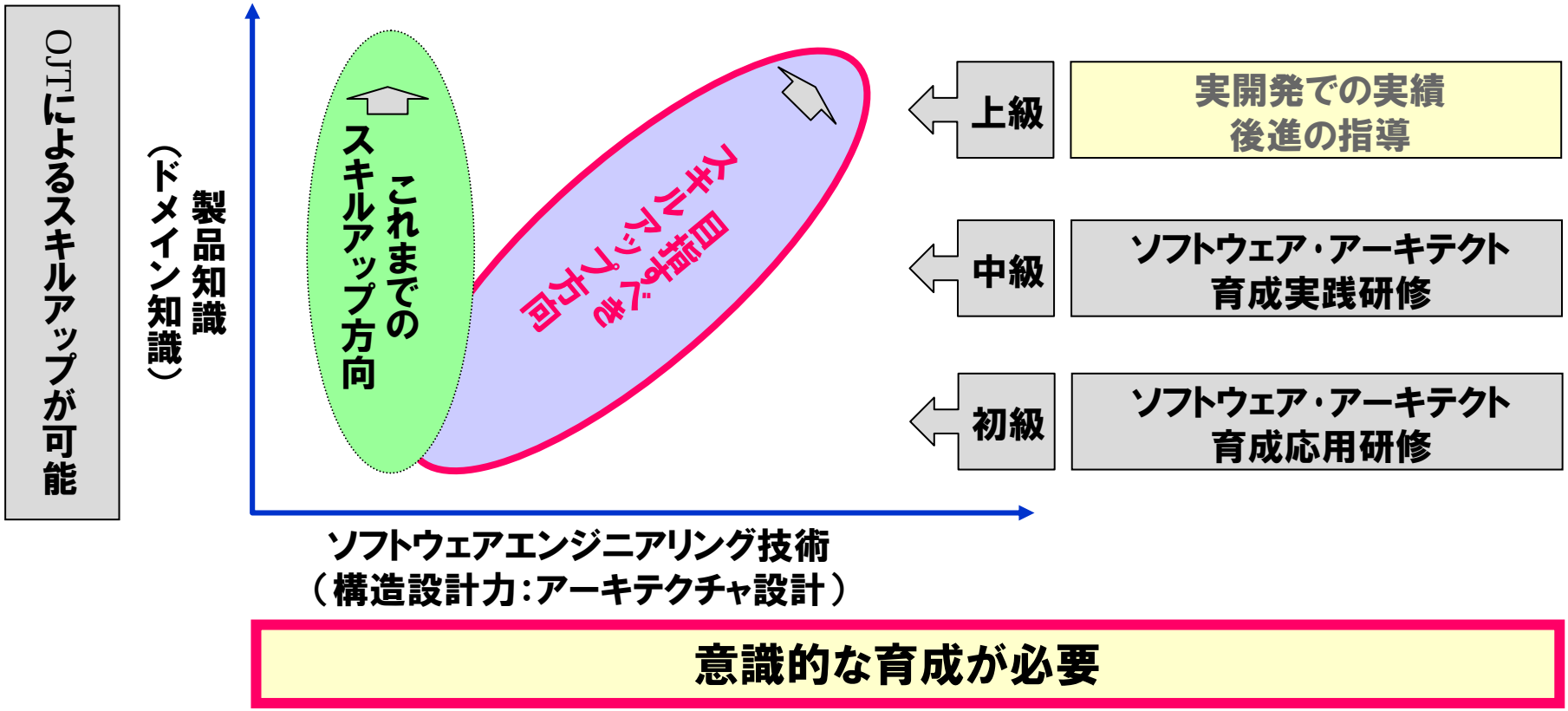
設計ビューと表記法の統一

- 構造設計の局面と表記法を3つに限定
  - モデル駆動開発への移行の容易性も視野に
- 表記法とUMLとの対応，ツールとの対応を明確化
- 表記方法に関する規約の策定

静的構造	動的構造	
静的構造図	タスク関連図	状態遷移図
コンポジット構造図（UML2.0） または SysMLのブロック図  機能ブロック（コンポーネント）間の関係を記述	コンポジット構造図（UML2.0） （task，interrupt，memoryなどのステレオタイプ付き）  タスク，割込み，共有メモリなどの関係を記述	状態遷移図（UML2.0） または，状態遷移表  静的構造の単位で記述

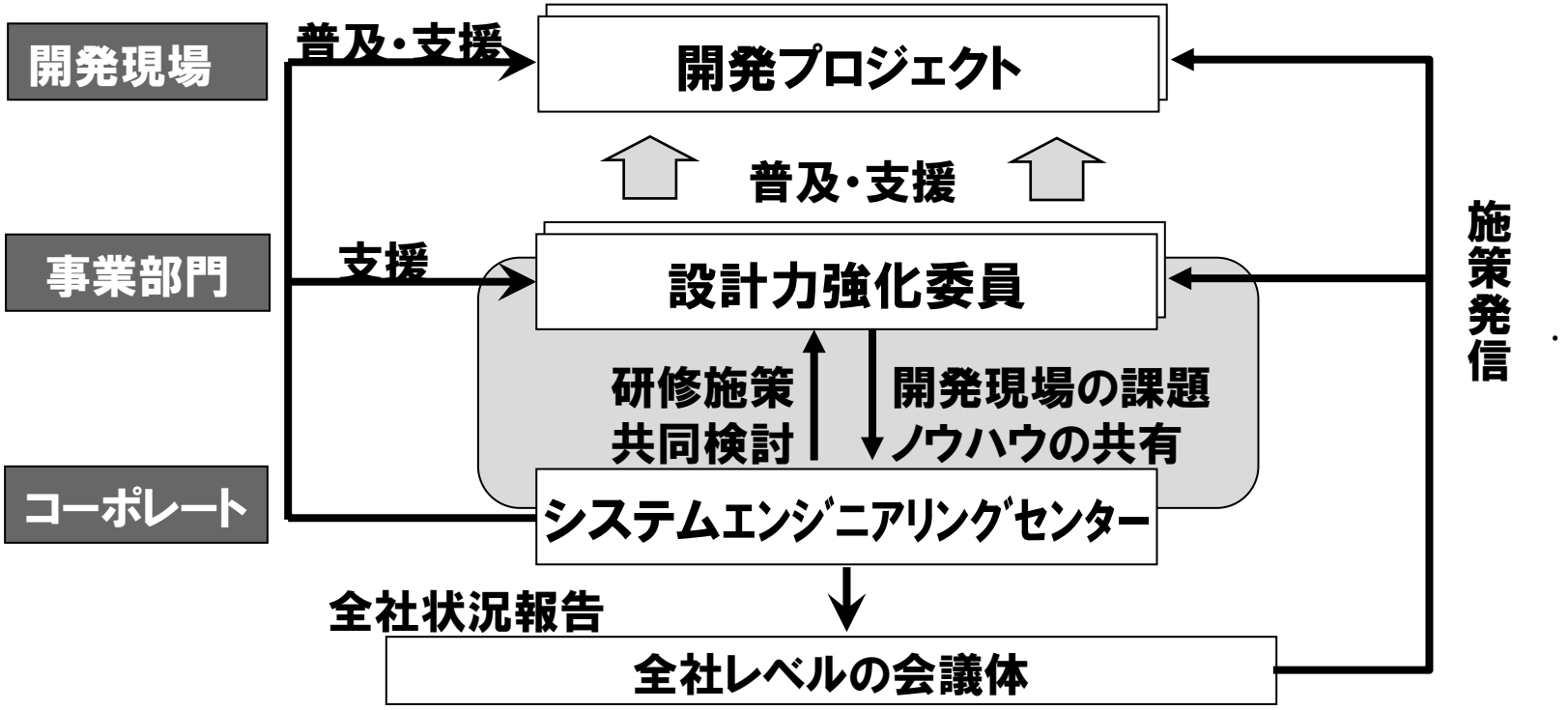
スキルレベルに応じた研修とスキル認定

- 目的：アーキテクチャ設計力を持つ人材を全社施策として意識的に育成
  - 社内の有識者を組織し，外部コンサルタントの協力のもと，社内の実態に合わせた研修コンテンツを開発．スキル底上げのため，全社的に実施



全社推進体制

- 研修試行が高評価・好結果であったことを受け，
- 全社施策の立案・発信の仕組みの中で人材育成施策を承認
- 事業計画と連動した人材育成計画の策定，
- 研修の受講及びスキル認定を全社施策（トップダウン）として実施



施策普及のポイント

- **現世ご利益を明確にすること**
  - アーキテクチャ設計は抽象的な考え方の要素が強いため，直ぐに理解できない人も多いことは事実
  - 現実のどのような問題を解決してくれるのかを納得してもらうことが重要
- **本質的な理解を促すこと**
  - 方法論を普及させるのではない．どのような方法論でもよい．上流で考えることができれば，目的は達成（手段と目的が逆転しないように！！）
- **簡単であること**
  - 開発現場のやり方と乖離が無い方法で，上流設計の勘所を掴んでもらうことが先決．
    - » 組込みソフトウェアの特性を考慮し，設計局面（ビュー）を限定
    - » 過去の組込みソフト開発で当たり前であったことこそ重要．先人の技術伝承
- **モティベーションを高めること**
  - 時にはトップダウンも必要，仲間意識，危機感，選ばれた存在 などなど．．

ご清聴ありがとうございました