

組み込みシステムの設計者向け 要件定義ガイドラインの提案と評価

| | |
|---------|-----------------------|
| 研究コース 5 | チーム D H C |
| 寺村 幹夫 | (株式会社デンソー) |
| 島田 真也 | (株式会社日立ソリューションズ株式会社) |
| 橋本 達也 | (N T Tコミュニケーションズ株式会社) |

指導員

| | | |
|-----|------|------------|
| 主査 | 栗田太郎 | (ソニー株式会社) |
| 副主査 | 石川冬樹 | (国立情報学研究所) |

目次

0.研究コース 5 マップ

1.はじめに

2.背景

3.現状分析

4.課題提起

5.解決策

6.解決策の評価

7.まとめ

0. 研究コース5 マップ

■ 研究コース5 の各チームの“要求工学上”の研究の位置づけ

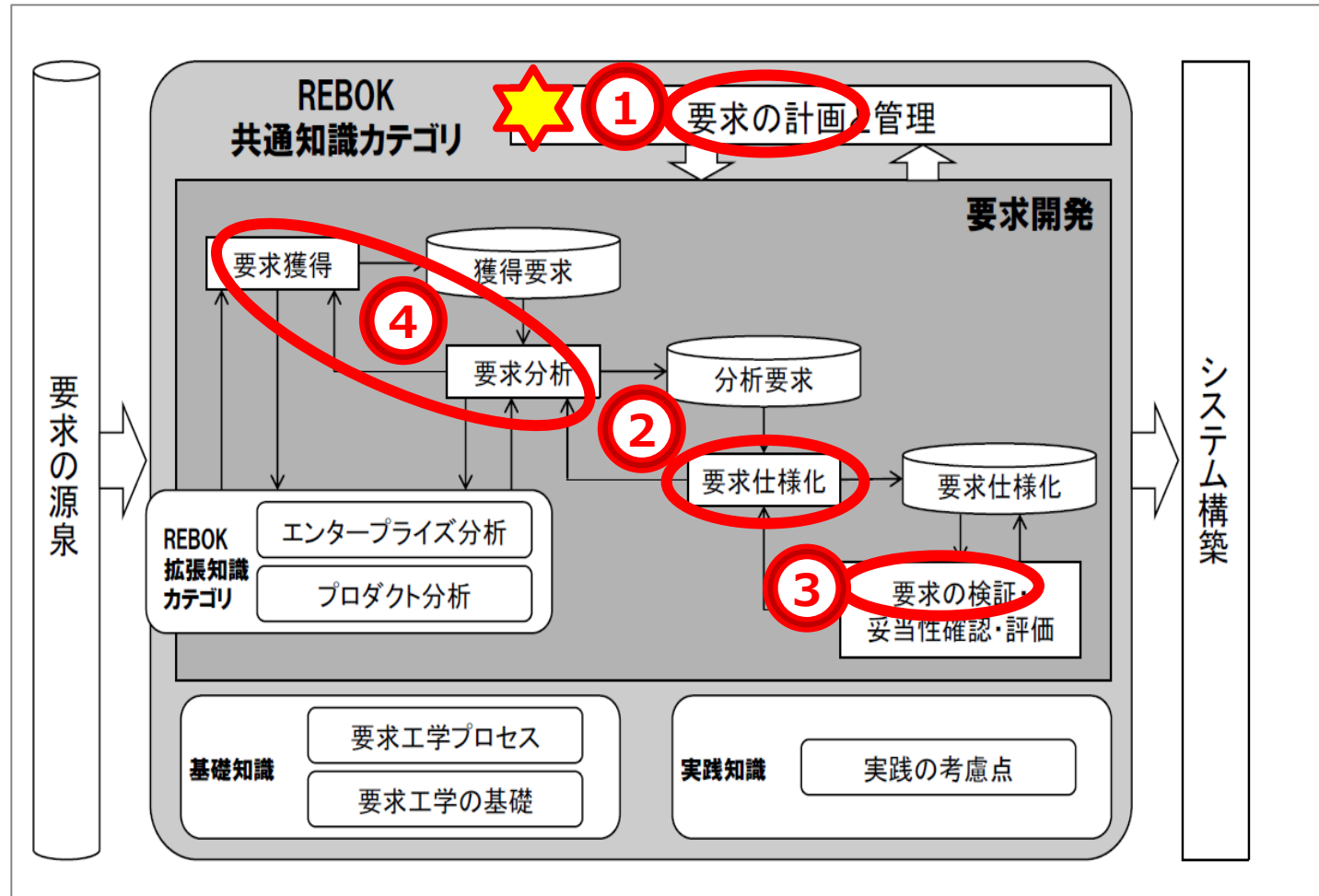
①チームDHC

↑ 本発表

②チームKNT

③チーム形式仕様

④チームGOBIT



出典元：IPA 要求工学知識体系（REBOK）概説

1. はじめに

■ なぜこのテーマなのか？

• 我々 チームDHC の紹介

- ・同じ **ソフトウェア開発**に従事
- ・同じ 超上流工程（システム要求分析～基本設計）に**課題意識**
- ・異なる エンタープライズ系開発、インフラ系開発、組み込み系 の **三者三様**

• 開発の源流となる “要求” が重要

- ・修正コストの例

| 要求 | 設計 | コード実装 | 単体テスト | 出荷後 |
|----|----|-------|-------|-----|
| 1 | 5 | 10 | 20 | 200 |

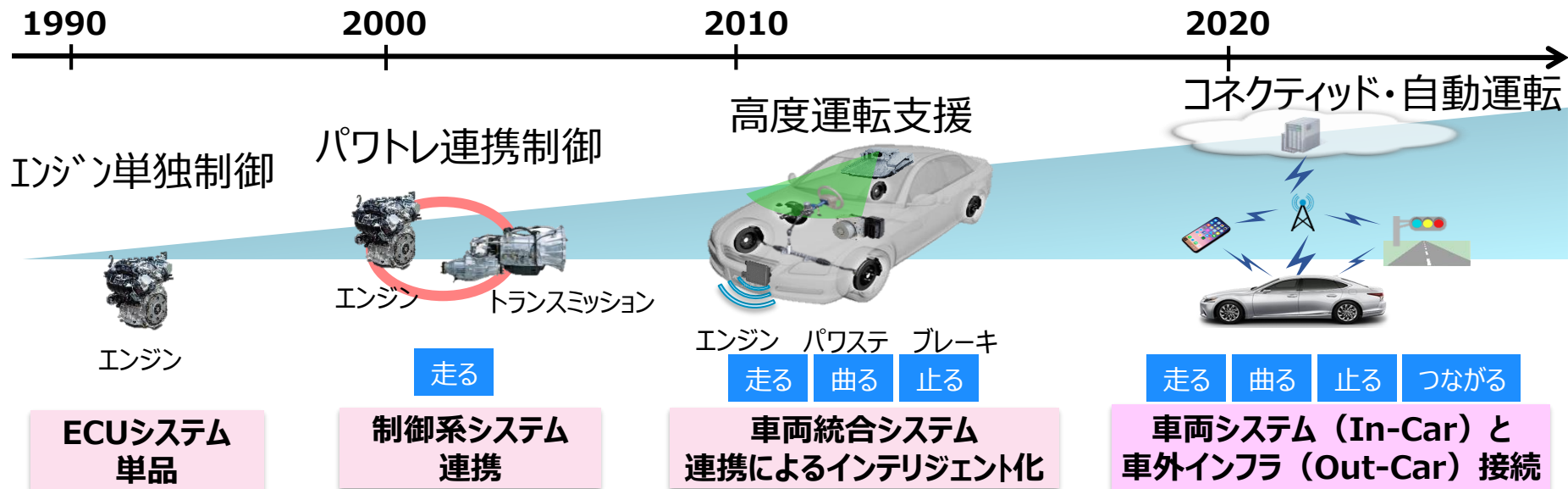
- ・手戻りは全開発費 30%～50%
- ・要求変更の歓迎は顧客対応力向上（Agileマニフェスト）

有用な要求の仕様化とは何だろう？ をテーマに研究

2. 背景

■ 組み込みシステムを取り巻く環境は？

電子システムの進化 (例) 車載システム



組み込みシステムだけの世界で開発

要求はハードウェア制約で制限されていた

多様な開発と接続

→ 制約が弱まる

組み込みシステムも IoT化の波によって要求の内容が大きく変化

3. 現状分析

■ 組み込み設計者の実状は？

組み込みシステムだけの開発

環境変化：大規模化 短納期 要求の詳細化

取り組み ①分業化 外部委託での人員補強

②抜け漏れのない ドキュメント化

③レビュー強化

※ベテラン設計者の知見でカバー

コード実装中心の開発スタイルは固持

+

多様な開発との接続に向けて

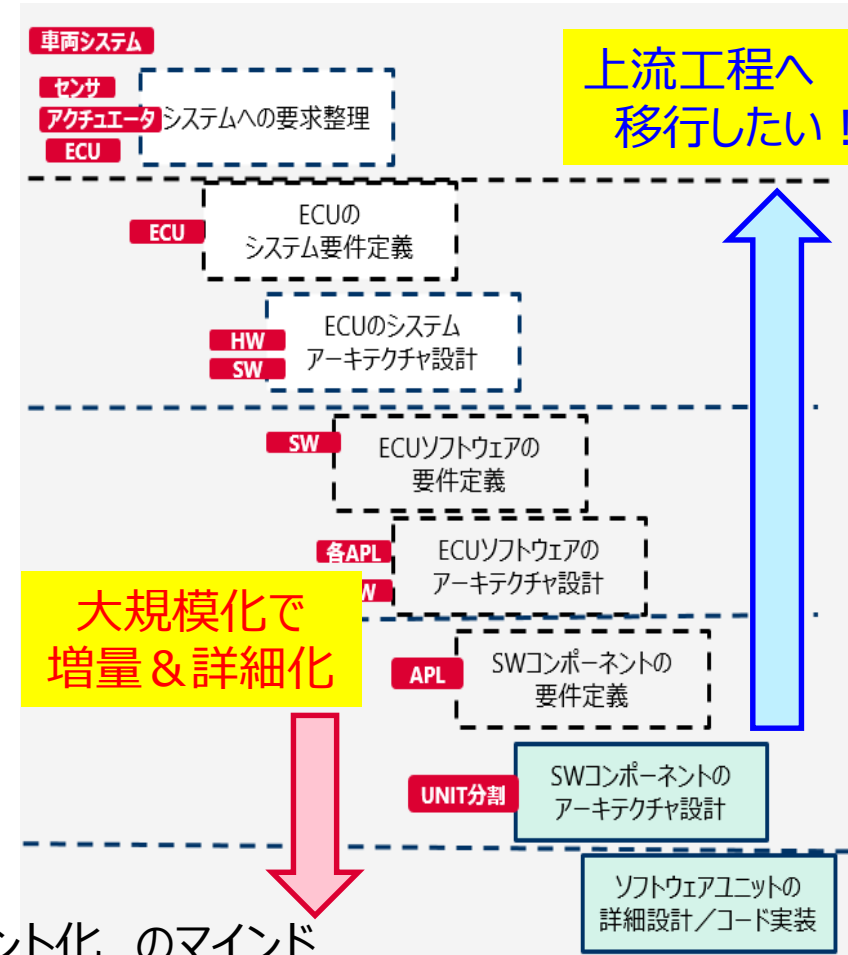
設計中心の開発スタイルへ移行必須

移行に向けた障壁

1) 追加する設計ドキュメントの作成工数

2) 上記、ドキュメント作成スキルの不足

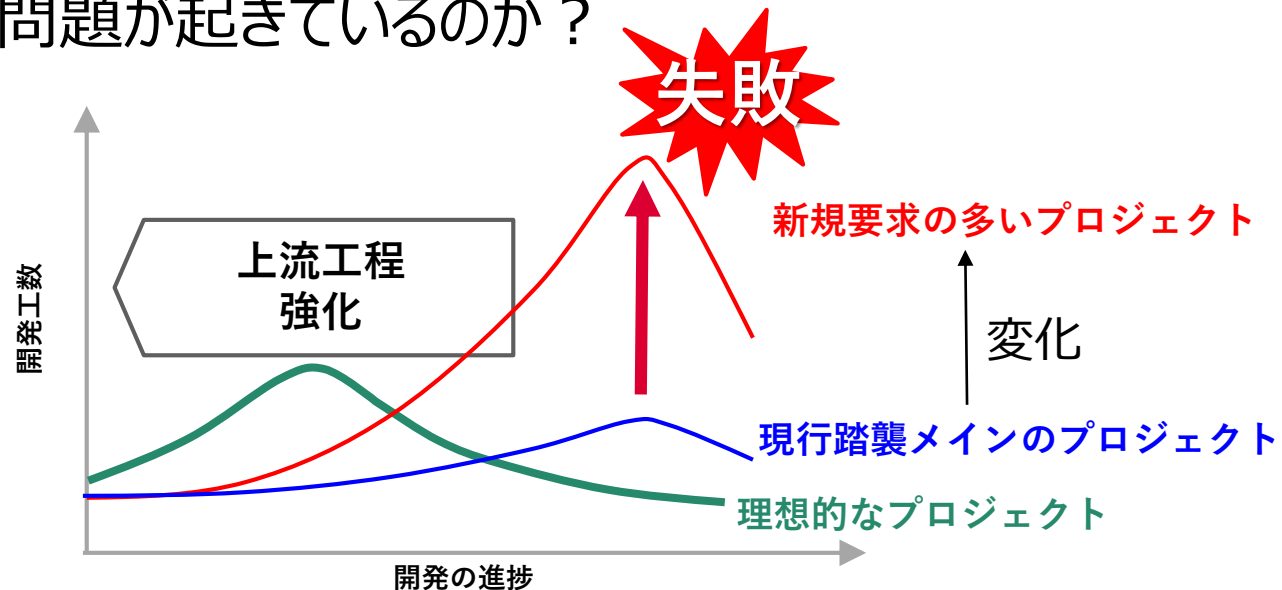
3) 優先が 仕様の詳細化 > 上流のドキュメント化 のマインド



上流工程への課題は理解しつつも 障壁は高くなっている

3. 現状分析

■ どのような問題が起きているのか？



現行踏襲メインのプロジェクト

例) 全体のプロジェクト数 約 80% で 全体工数の 約 20% を占める
手戻り工数も少なく、分業化や外部委託での人員補強でカバー可能

新規要求の多いプロジェクト

例) 全体のプロジェクト数 約 20% で 全体工数の 約 70% を占める
手戻り工数が多い ⇨ 上流のシステム要件分析 や アーキテクチャ設計スキル不足

従来のコード実装中心の開発スタイルでは問題が顕在化

3. 現状分析

■ 具体的な 欠陥事例

| # | 分類 | 作り込み箇所 | 欠陥内容 | 根本原因 |
|---|----------|-------------|-------------------------------|------------------|
| 1 | 機能要件考慮漏れ | SWアーキテクチャ設計 | 複数コンポーネント間で初期値設定が統一されていない | 前提条件/制限制約事項の認識齟齬 |
| 2 | 機能要件考慮漏れ | SWアーキテクチャ設計 | 複数コンポーネント間で停止Duty値設定が統一されていない | 前提条件/制限制約事項の認識齟齬 |
| 3 | 機能要件抽出漏れ | ECUシステム要件 | 他ECUとの協調する機能考慮漏れ | 要件化視点の不足 |
| 4 | 機能要件抽出漏れ | ECUシステム要件 | 強制駆動が70%以上となった場合の駆動モードの機能考慮漏れ | 要件化視点の不足 |
| 5 | 機能要件抽出漏れ | ECUシステム要件 | 通常時と強制駆動時の駆動Duty範囲切替の機能考慮漏れ | 要件化視点の不足 |

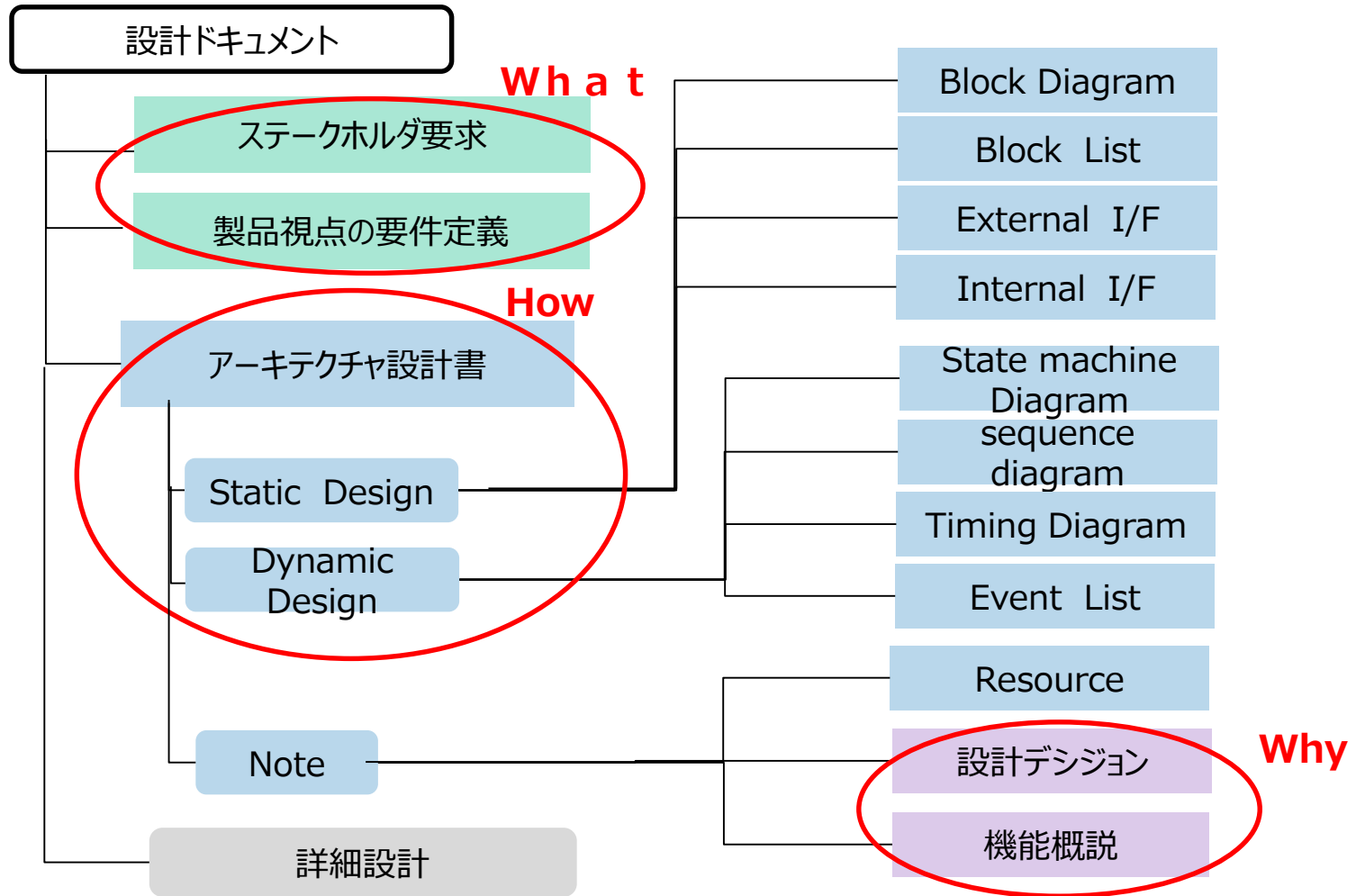
■ 欠陥要因

- ・大規模化 に対する 分業による人員補強 ⇒ 上流工程スキルが弱まる
- ・エンドユーザの上位要求は普遍的なものが続く ⇒ 暗黙知に依存した傾向があった
- ・短納期化で設計ドキュメント化は優先度低 ⇒ 作法無視の “ごった煮”ドキュメント

上流工程経験 と 設計ドキュメント化 の2つの経験不足が重なる

3. 現状分析

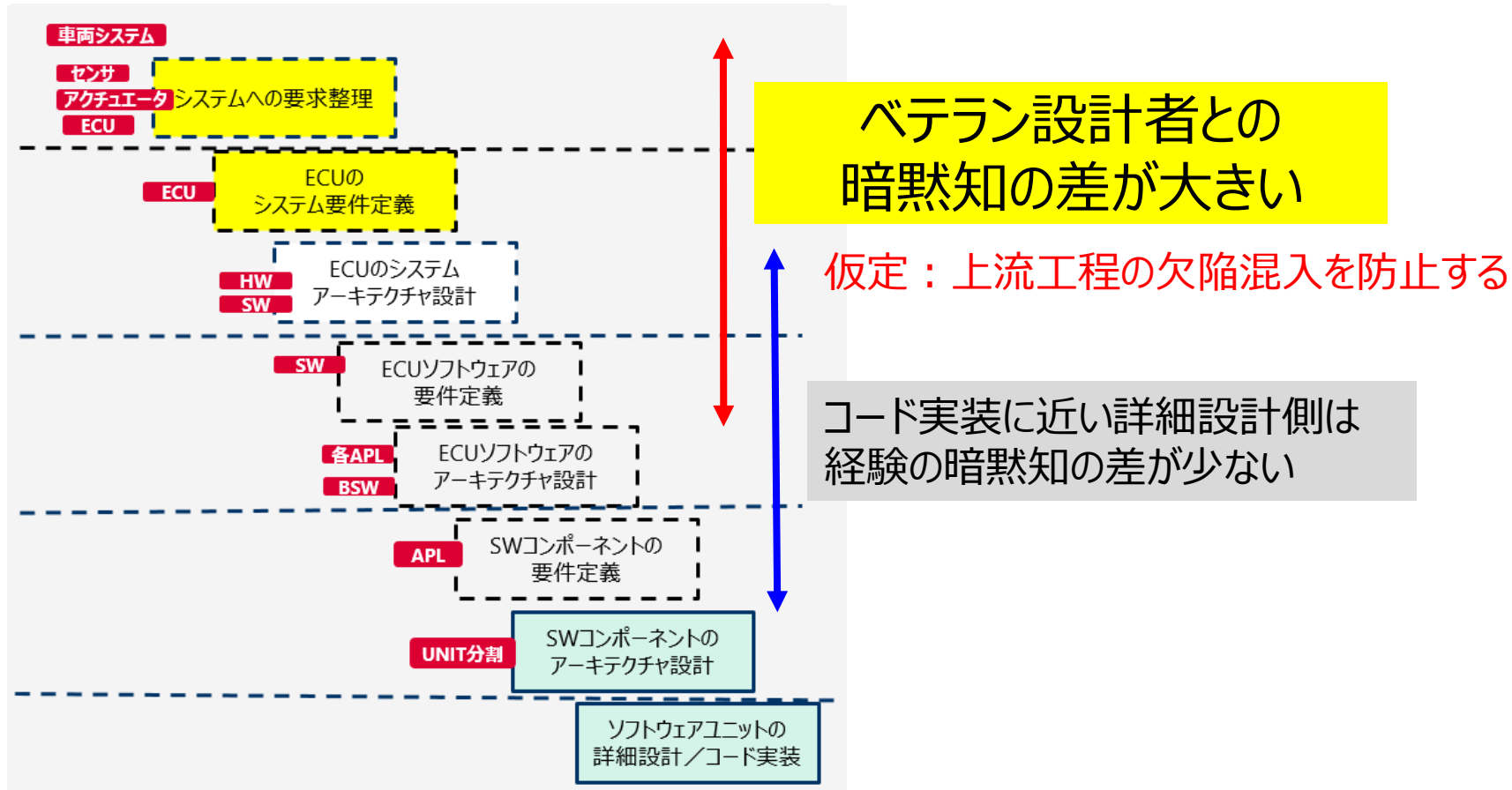
■ 理想とする設計ドキュメント構成



全方位の設計ドキュメント作成が理想だが工数は膨大

4. 課題提起

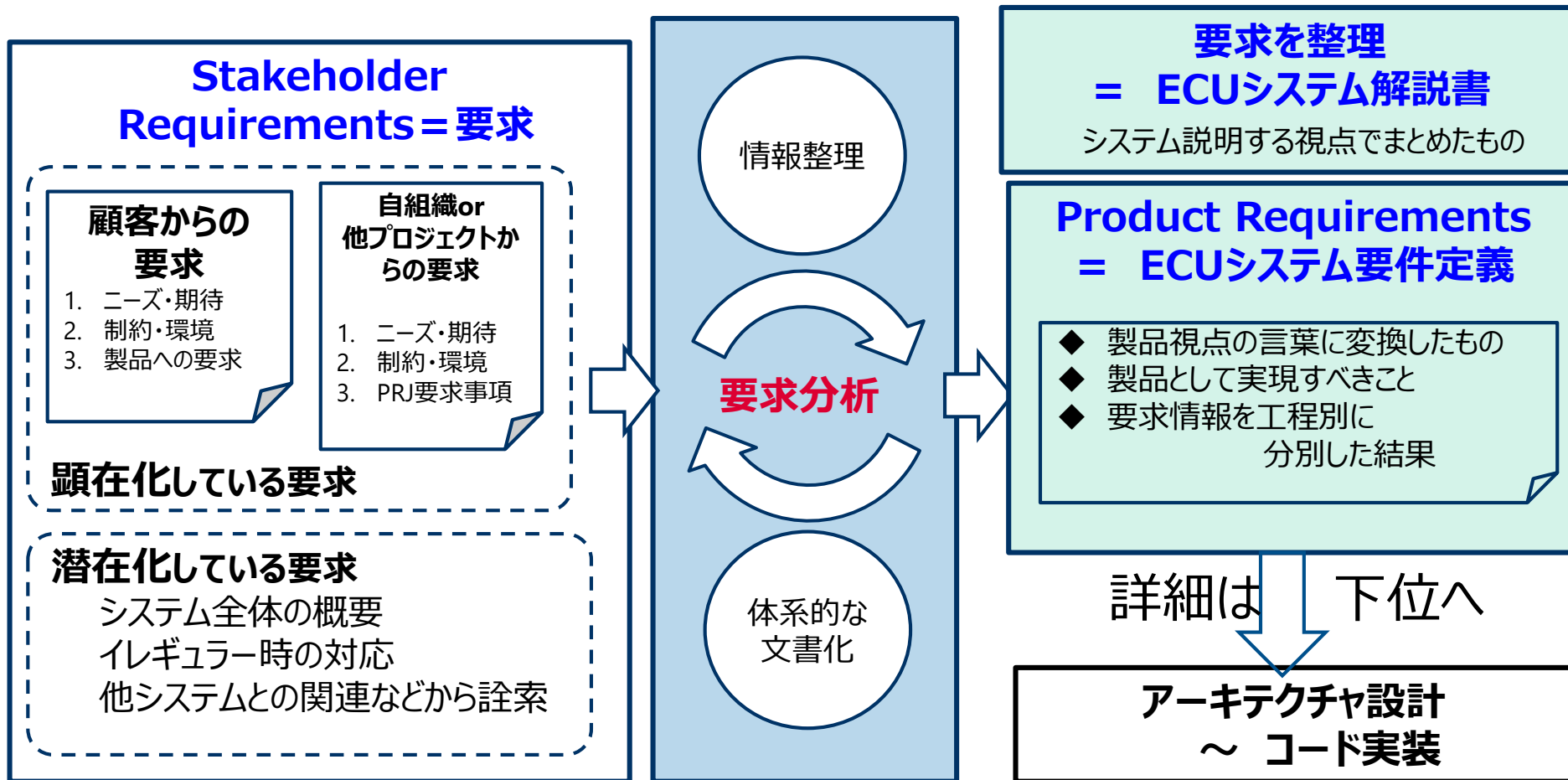
■ 要求？要件？制約？アーキテクチャ設計？ 整備はどこからが良い？



最上位階層の組み込みシステムの要件定義からいこう！

5. 解決策

■ システムへの要求の整理、要件定義はどのようにまとめていく？



システムの概要説明、顧客の要求事項、製品としての要件定義を識別

5. 解決策

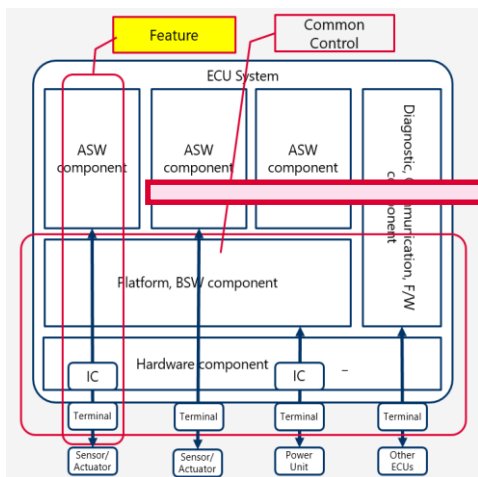
■ 要求の整理方法は？（システム解説書のガイドライン）

状況

・ベテラン設計者は、個々の過去の経験を
知見として継ぎ足し

・組み込み系システム

制約上、複数の機能群となっている



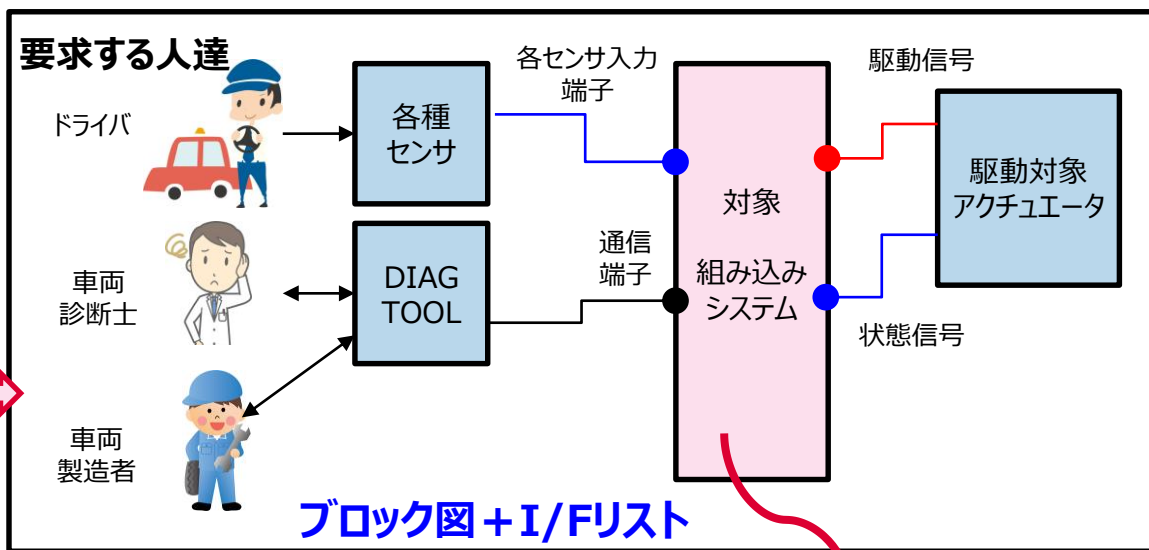
1機能を
提取し

Requirements Specification

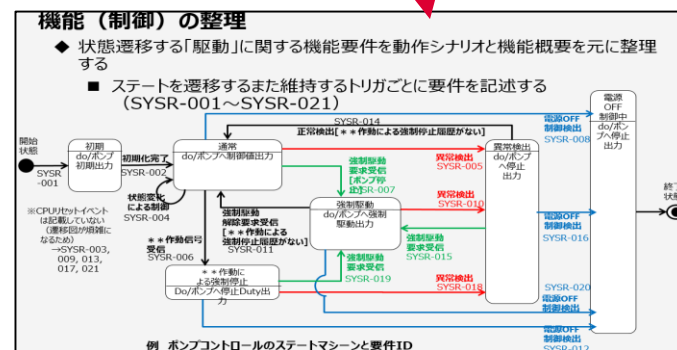
$$= \sum (\text{Feature}) + \sum (\text{Common Control})$$

機能群

・分析対象を絞り、ブロック化し要求分析を分析



・対象ブロックに対して
ふるまいを整理



アーキテクチャ設計書の要素を情報整理に活かす

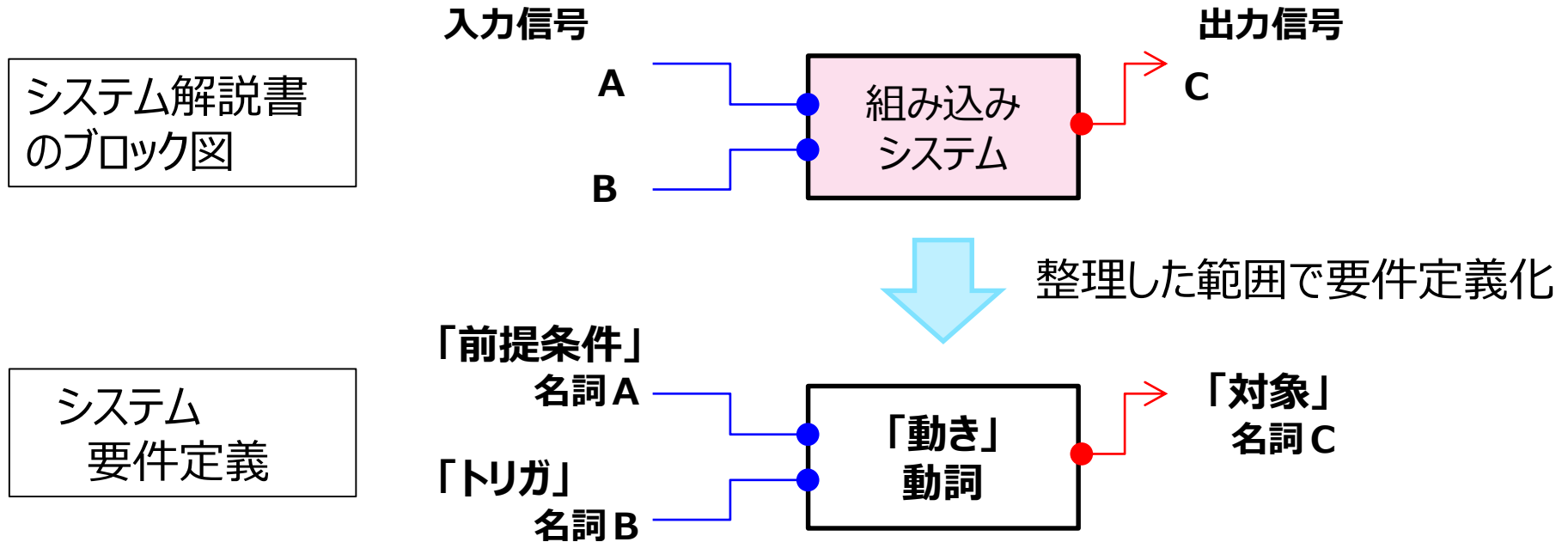
状態遷移図

5. 解決策

■ 要件定義の表記方法は？（要件定義ガイドライン）

システム解説書の活用 + ブラックボックステスト活用 を考慮する

- 1) 要件定義は 原則、単文とする
- 2) 記述形式を「前提条件」+「トリガ」+「対象」+「動き」とする
- 3) 表現をそろえる



ブラックボックスを見立てて用語を限定し、システムの要件に変換

6. 解決策の評価

■ 評価方法

- 模擬的な設計を行い、上流工程の経験値の差が、どれだけで評価する
- 差の指標は、ベテラン設計者（約20年）が レビューで欠陥抽出した件数と一般の設計者（10年前後）が欠陥を防止している件数の差
- 件数の差が小さければ 本提案が効果があったと判断する

サンプル

新規性の高い ある特定機能を模擬的に設計 （コード実装を除く）
詳細設計書で 約250ページ程度

Before

一般の設計者 13人
欠陥抽出できず

差あり

（上流工程
5件差）

ベテラン設計者の
欠陥抽出

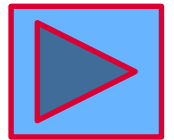
- システム解説書
- システム要件定義

- 上記を作成するための
ガイドライン

After

一般の設計者 x 人が
欠陥防止 できている
5件抽出

差なし



欠陥抽出の差の大小でガイドラインを評価

6. 解決策の評価

■ 評価結果

- 欠陥を抽出し防止した 被験者人数の内訳
(被験者数33人 業務年数平均11.5年 中央値9.7年)

| 欠陥を防止した 件数 | 欠陥を防止した 被験数 | 左記 割合 | 業務年数 平均値 | 業務年数 中央値 |
|---------------|----------------|----------|-------------|-------------|
| 5件 全て | 16人 | 48% | 12.9年 | 11.5年 |
| 4件 | 11人 | 33% | 9.5年 | 11.0年 |
| 3件 | 5人 | 15% | 8.6年 | 11.0年 |
| 2件以下 | 1人 | 3 % | 25.0年 | 12.0年 |

- 33人の被験者は 平均件数 4.3件 欠陥を防止

・経験年数が浅くともベテラン設計者へ近づいた

6. 解決策の評価

■ 評価結果

- 欠陥項目別の 33人の内訳

| 欠陥内容 | 欠陥を防止した被験数 | 業務年数 平均値 | 業務年数 中央値 |
|------------------------|------------|-------------|-------------|
| 特定状態での要件もれ① | 30人 | 11.9年 | 11.5年 |
| 特定状態での要件もれ② | 33人 | 11.5年 | 11.0年 |
| 他SWコンポーネントへの 影響考慮不足 | 29人 | 11.4年 | 11.0年 |
| SWコンポーネント間の 認識齟齬① | 23人 | 11.5年 | 12.0年 |
| SWコンポーネント間の 認識齟齬② | 26人 | 11.7年 | 11.5年 |

- 下3項目はアーキテクチャ設計工程

下位のアーキテクチャ設計にも好影響を与えている

7. まとめ

■ 研究の成果

環境変化→問題の顕在化

新規要求のプロジェクト増加と共に
手戻り工数が増加し抑制したい

問題分析 → 仮定

上流工程にて 欠陥が混入している
ベテランの**上流工程経験を活かす**必要あり
設計ドキュメント化も工数がかかり、選択要

解決策

- ①要求整理のガイドライン、②システム解説書
- ③②から要件定義化への変換ガイドライン

成果物

④最上位のシステム要件定義

想定する効果

手戻り工数割合が全プロジェクト工数の25%以下に抑制

検証

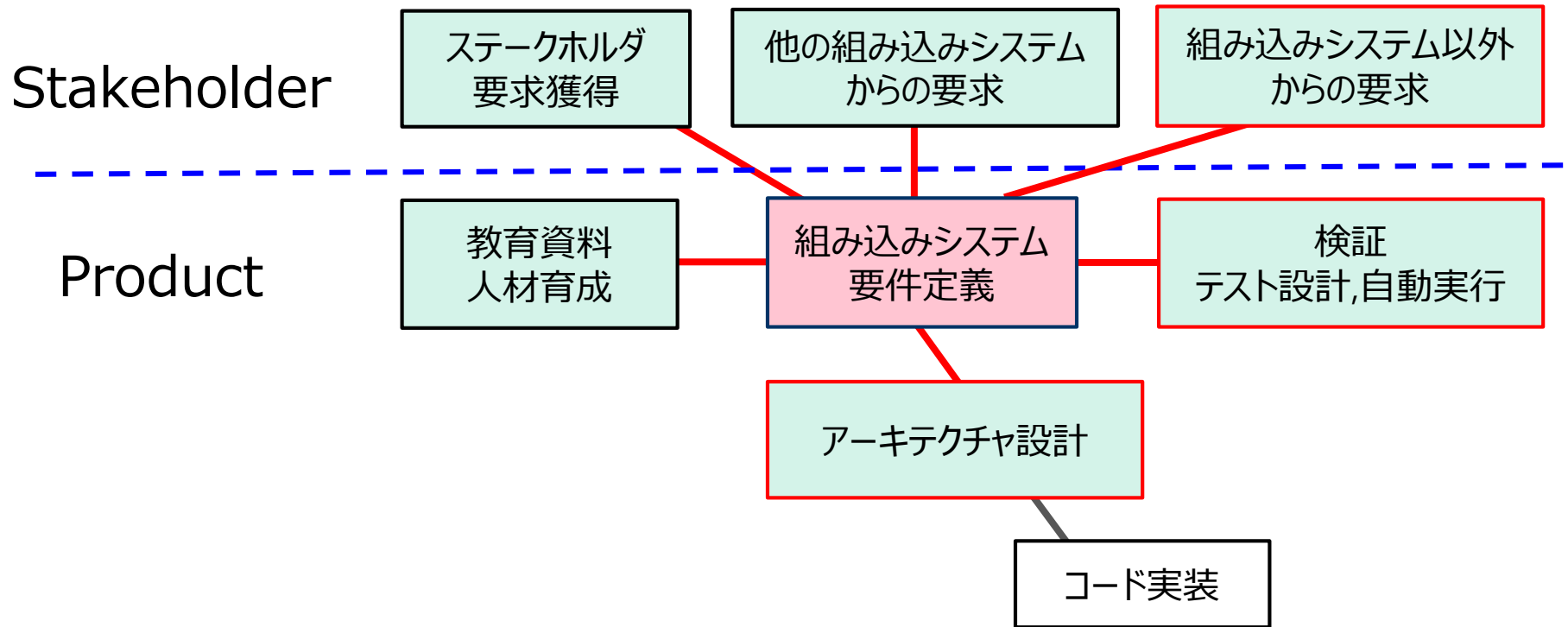
上流工程の欠陥の混入を阻止している
ことを実証

属人性排除 + 下位工程への波及効果

最上位のシステム要件定義が有効であると考える

7. まとめ

■ 今後の取り組み



1. 要件ベースの自動テストからの・・・現行踏襲アーキテクチャの再構築障壁の下げる
2. エンタープライズ系システム と 組み込みシステムとの接続に有用な要求の仕様化

本テーマを起点に全方位で新たな課題設定が可能

ご清聴ありがとうございました

研究コース5 チーム D H C

参考文献

- [1] 山本修一郎, 「要求」の40年 1979—2019 ～過去・現在・未来～ ASDoQ 2019 基調講演
- [2] 山本修一郎, デジタルトランスフォーメーションに向けた要求管理知識, 第20回知識流通ネットワーク研究会, SIG-KSN-020-01
- [3] 福嶋愼一, 組み込み系ソフトウェア開発の課題分析と提言 2008年度JEITAソフトウェア事業委員会セミナー
- [4] 組み込み系ソフトウェア向け設計ガイドESDR [事例編], 独立法人情報処理推進機構 (IPA)
- [5] 貝瀬康利, 開発要件の完成度を高めるアプローチの検討, JASA技術本部セミナー, JG-5
- [6] 要求工学知識体系(REBOK) とユーザ指向要求工学の調査研究, 社団法人 情報サービス産業協会, 20-J007
- [7] 要求開発・管理ベストプラクティスとその体系化の調査研究, 情報サービス産業協会, 19-J004
- [8] 要求開発ベストプラクティスが示す成功パターンの調査研究, 情報サービス産業協会, 18-J008
- [9] 山田大介, ソフトウェア疲労をアーキテクチャ設計で ソフトウェア品質管理研究会2019年度第8回特別講義
- [10] IEEE, Requirement Boilerplates: Transition from Manually – enforced to Automatically-Verifiable Natural Language Patterns