

派生開発カンファレンス2019

**「XDDP導入してから3年経ちました」
～現在どうなったでしょうか～**

三菱電機コントロールソフトウェア株式会社
笹田 朋邦

三菱電機コントロールソフトウェア株式会社 (MCR)

三田事業所

- ・カーマルチメディア
- ・カーエレクトロニクス

伊丹事業所

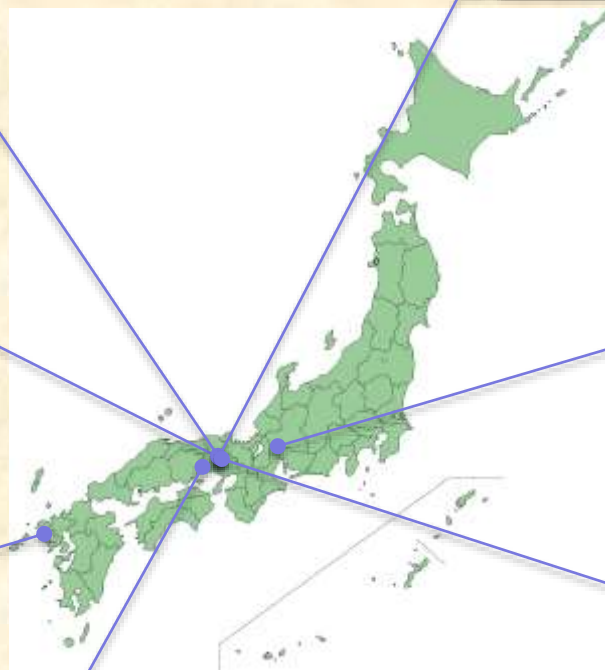
- ・車両情報システム
- ・トレインビジョンシステム
- ・列車走行制御システム

長崎事業所

- ・産業システム
- ・自社製品 (Monity)

姫事事業所

- ・カーエレクトロニクス
- ・情報システムの 構築・販売



トータルソリューション事業所

- ・FAシステム
- ・鉄鋼制御システム
- ・パワーエレクトロニクス

名古屋事業所

- ・FAシステム

神戸事業所

- ・社会・公共・広域分野
- ・電力情報分野
- ・原子力分野
- ・火力・水力発電分野
- ・交通分野

ソフトウェア開発事業

Software Development Business

<http://www.mcr.co.jp/>

社会・公共システム分野

水・道路・ビル管理など快適な生活環境に欠かせない社会のインフラを支えるシステム開発を行っています。

電力システム分野

発電プラントの安全性・安定性の確保、複雑化する電力系統に対応し、電力の安定供給・効率化に貢献しています。

カーエレクトロニクス分野

エンジン・トランスミッション・シャーシなど電子制御用車載コンピュータ装置の開発やカーオーディオ、カーナビゲーションシステムなど、快適で安全なドライブ環境を実現するカーエレクトロニクス製品のソフトウェア開発に取り組んでいます。国内／海外のカーメーカーの案件が有ります。

交通システム分野

交通機関に求められる正確な運行、安全性の確保、サービスの充実など社会的ニーズにこたえています。国内／海外の交通システム案件が有ります。

産業・FAシステム分野

鉄鋼・自動車・食品などの製造工場に対して、制御システム・生産管理システムにより最適なソリューションを提供します。

映像・情報・通信システム分野

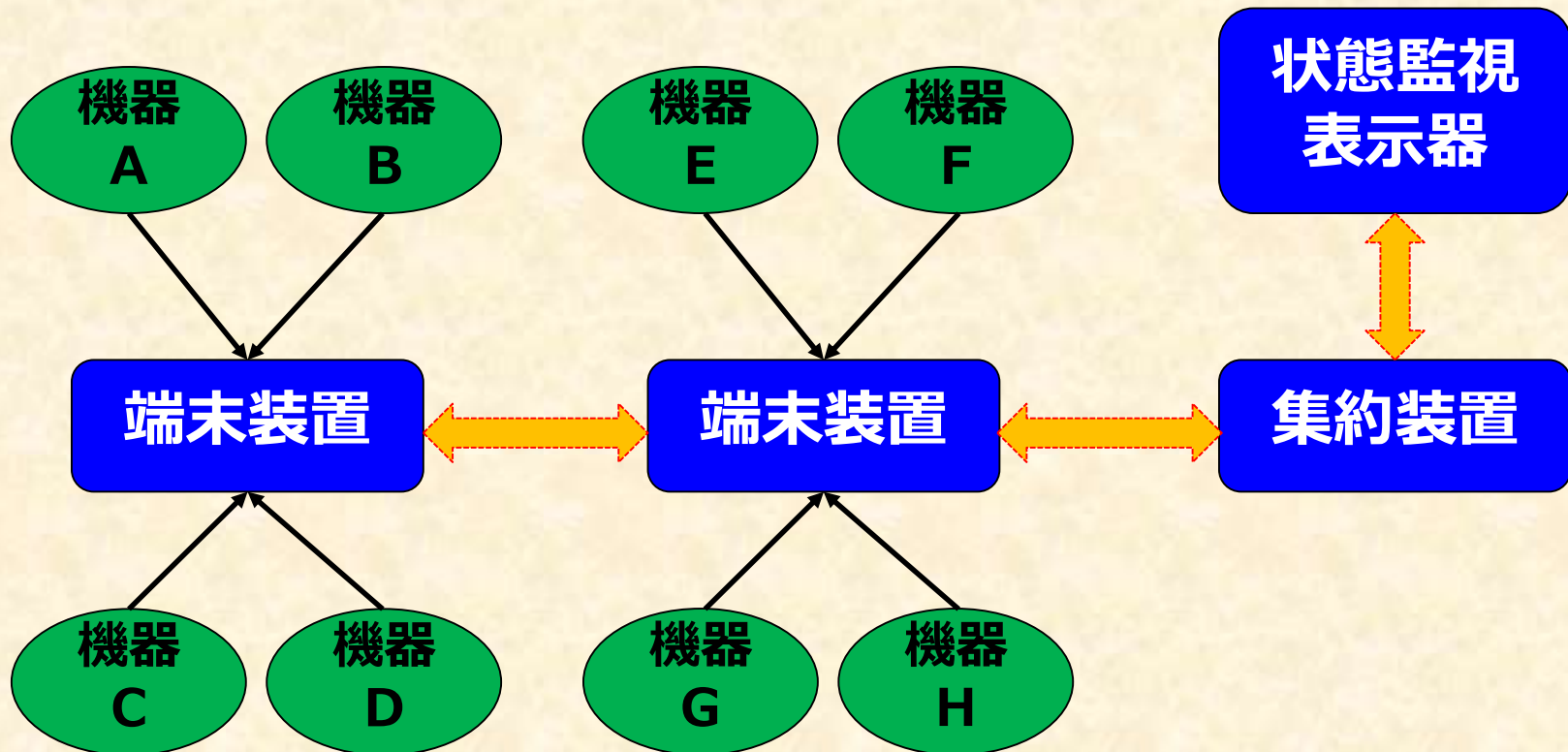
大型映像システムでのダイナミックな映像と多彩な情報表示を実現するシステム。道路上の事象を検出し、道路管理者やドライバーへ情報を提供するシステムを開発しています。

1. 業務紹介

1. 業務紹介

対象機種概要

各機器の異常情報や動作状態を
状態監視表示器で監視するシステム



1. 業務紹介

 : 対象機種受注フェーズ

要求定義

システム設計

ソフトウェア設計

プログラム設計

コーディング

組合せ試験

単体試験

システム統合試験 1

システム統合試験 2
(品質管理部門)

1. 業務紹介

対象機種 1 案件あたりの派生開発規模

全体のソースコードライン数 : 250k line

変更時のソースコードライン数 : 7k~ 10k line

開発期間 : 3~8ヶ月

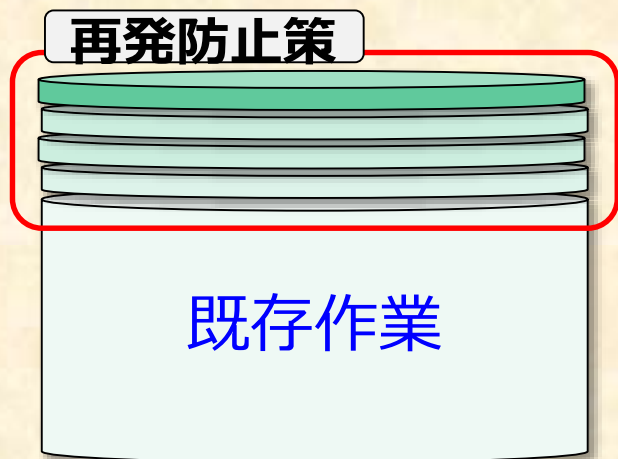
2. 背景

(X D D P 試行～効果の確認)

2. 1 XDDP導入経緯

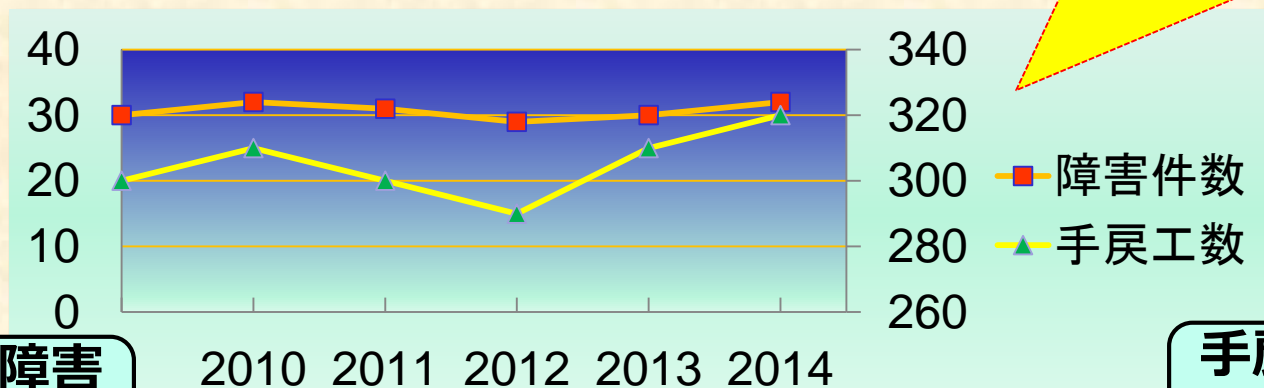
2. 1 XDDP導入前

(1) 品質面の問題



障害の再発防止策を行うたびに工数が肥大化

多数の施策を講じるが障害件数・手戻り工数は横バイ

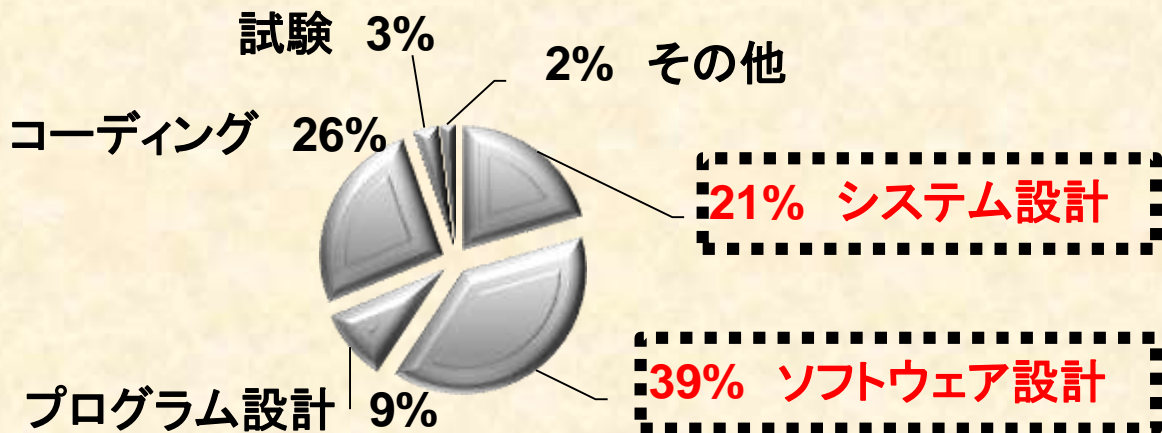


障害
件数

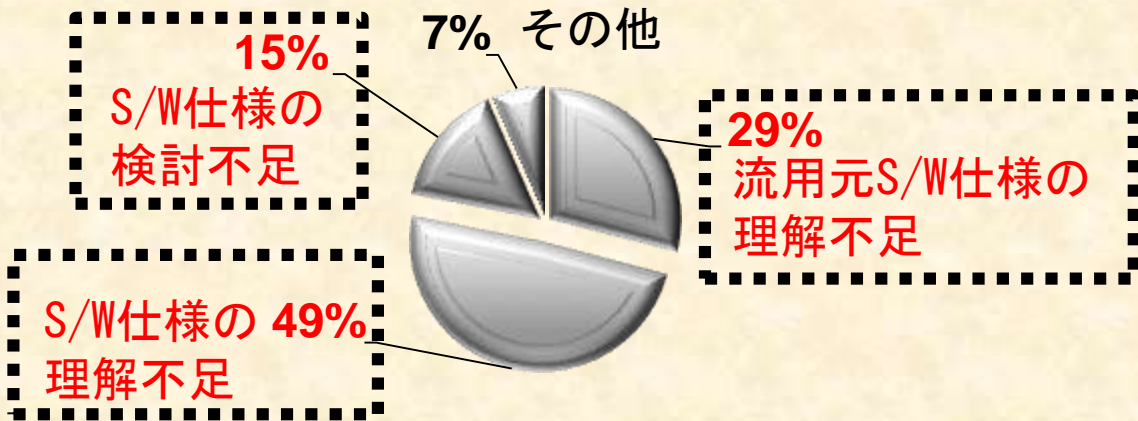
手戻
工数

2. 1 XDDP導入経緯

流出障害
作り込みフェーズ



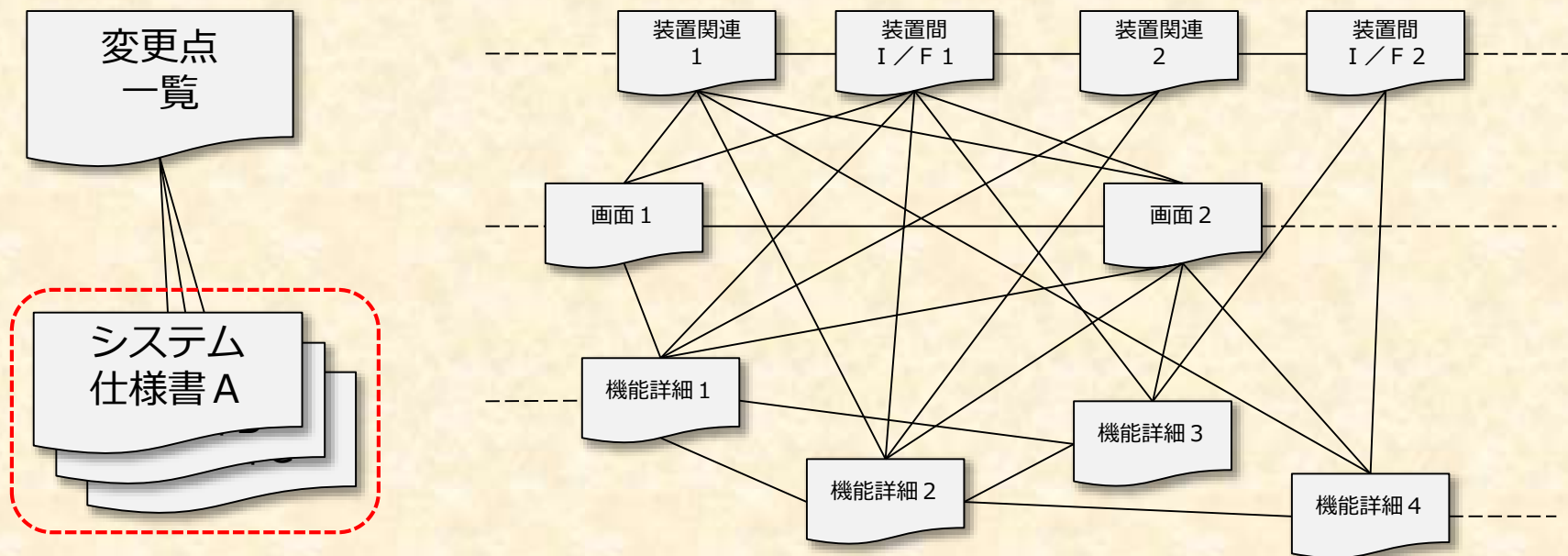
障害原因



対策を行っているが、障害傾向は変わらず

2. 1 XDDP導入経緯

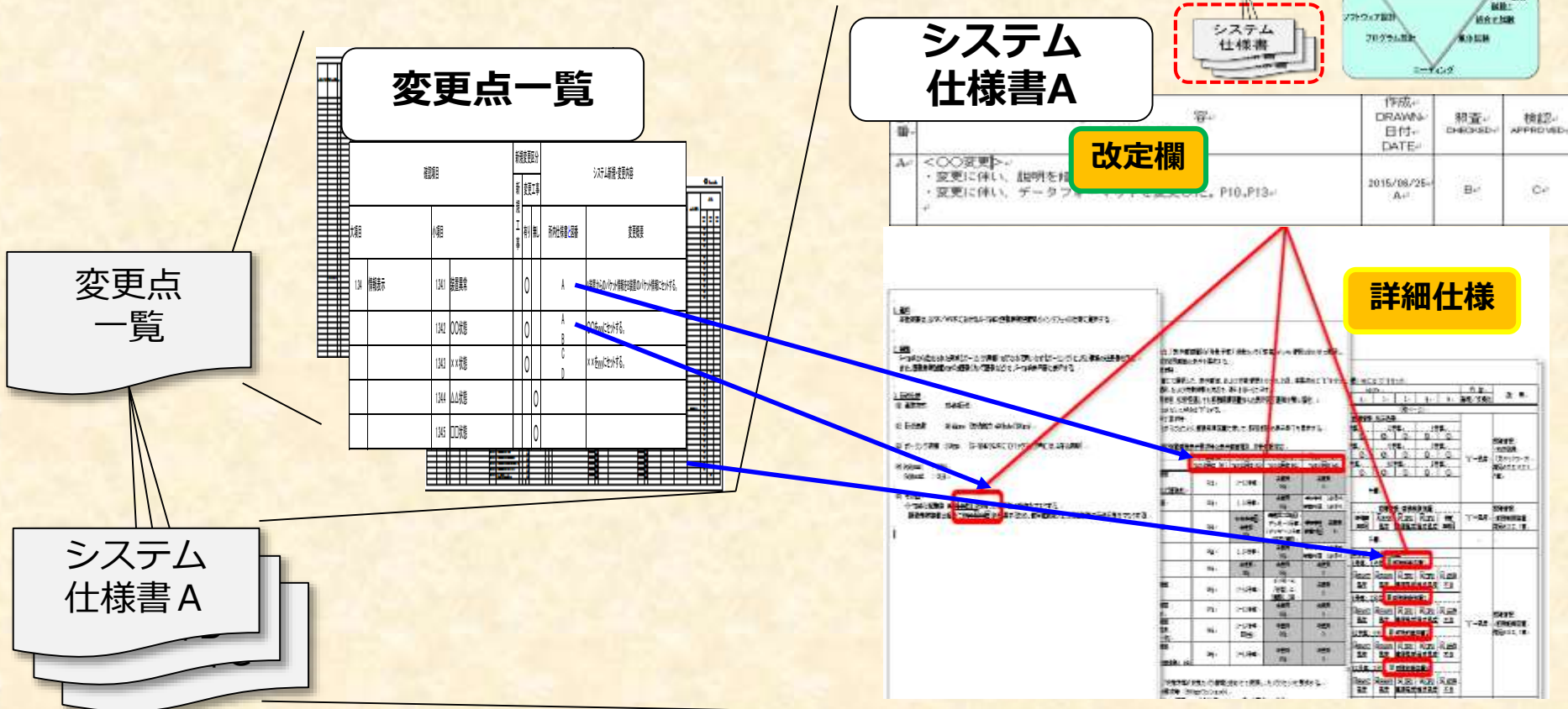
(2) 品質面の問題 システム仕様書体系①



多くの仕様書から関連を整理することに時間がかかる

2. 1 XDDP導入経緯

システム仕様書体系②



- ・ 変更箇所がまとまっておらず、変更モレの確認が困難
- ・ 要求、理由の記載がなく、結果のみの記載となっているため、なぜ変更するのかが不明

2. 1 XDDP導入経緯

(3) 受注フェーズ内の問題

ソフトウェアの全体像が理解できないまま変更仕様を作成



変更対象プログラムの全体像



変更箇所のみ
仕様書化

変更仕様

(プログラム仕様書)

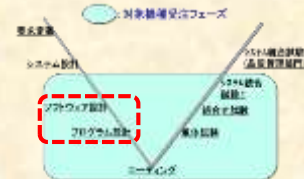
変更前	変更後

ソフトウェアの全体像を理解せずに、変更箇所のみを仕様書化するため、変更モレ・変更方法の誤りが発生

2. 1 XDDP導入経緯

(3) 受注フェーズ内の問題

ソフトウェア設計を下流フェーズで文書化している



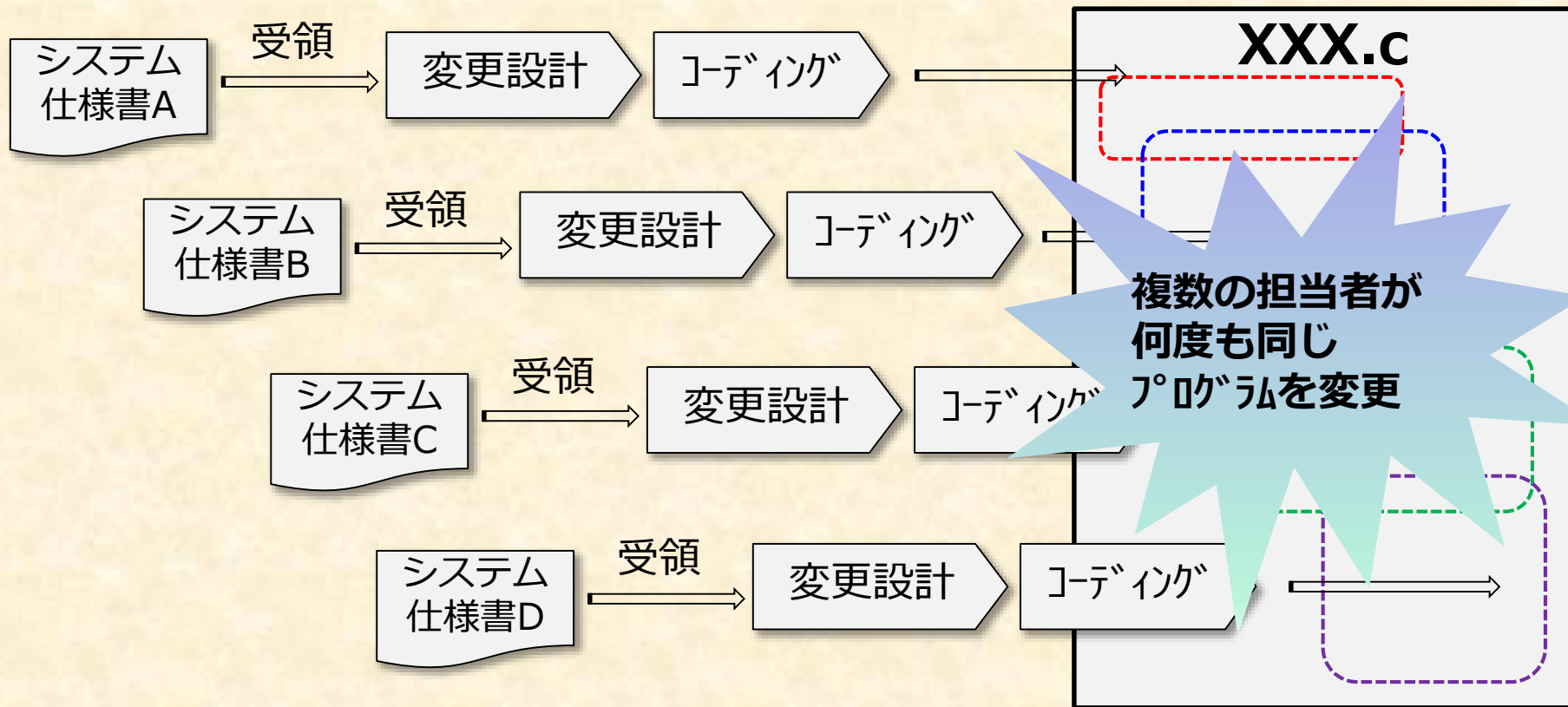
試験と並行して設計書を作成

- ・ DRは開催タイミング、回数ともに不適切
- ・ ソフトウェアの設計検証が不足

2. 1 XDDP導入経緯

(3) 受注フェーズ内の問題

非効率な方法でプログラム変更作業をしている



何度も同じプログラムを変更するのでムダや誤りが発生

2. 2 XDDP 試行 (2015年度)

2. 2 XDDP 試行 (2015年度)

2. 2 XDDP の試行 (2015年度)

試行メンバー：ベテラン2名、中堅1名、若手1名

初試行結果 = **効果あり!**

(FL化率 = 設計・製作工程の摘出誤り件数 / 全工程の摘出誤り件数)

No.	メトリクス	現行プロセス (見積値)	改善プロセス (実績値)
1	障害件数	19件	3件
2	FL化率	66%	94%
3	工数	398hr	390hr
4	コード生産性	20 Line/hr	61 Line/hr
5	ドキュメント生産量	252ページ	228ページ

2015年度は6案件に適用し、同等の効果を確認した。
(2015年度の案件適用率：7%)

3. 課題

3. 課題

(1) XDDPの他案件への展開

(2) 仕様書の改善

(3) 設計の改善

3. 課題

(1) XDDPの他案件への展開

試行6案件から他案件へ拡大できるか

＜展開するにあたって＞

①開発メンバーから

- ・ 今までのやり方や成果物を変えたくない
- ・ コーディングしないと試験に間に合わない
- ・ 記述 (USDM) が多い
- ・ オレには必要ない (自我流)

②管理者から

- ・ プロセスを変える必要があるのか？
- ・ 効果があるのか？
- ・ 他にもっとやることがあるのでは？

3. 課題

(2) 仕様書の改善

- ・ 成果物が次回の開発に生かされていない
⇒作成しっぱなしで、もったいない
- ・ S / W仕様書が整備されていない
⇒差分開発の限界、効率が悪い

(3) 設計の改善

- ・ 派生開発が続きアーキテクチャが崩れている
- ・ S / W構造が複雑
- ・ 複数案件で S / W資産が重複
(クローンソフトなど)

4. 課題についての取組み

4. 1 XDDPの他案件への展開 についての取組み

4. 1 XDDPの他案件への展開の課題についての取組み

(1) XDDPを理解してもらうための説明会の実施

(2) ムダな成果物を廃止

(3) 効果を定量的に示し続ける

(4) マニュアル、規程の整備

(5) 試行実施メンバーによる、他PJへのDRの参画

(6) 品質関連のあらゆる資料に「XDDP」

(7) 社内品質改善発表会 ⇒ 最優秀賞

社内外へ
積極的
アピール

4. 1 XDDPの他案件への展開の課題についての取組み

(1) XDDPを理解してもらうための説明会の実施

①株式会社エクスマーシオン様による説明会

2016年3月 社員向け 約60人出席

(所長、部長、課長は必須)

2016年4月 客先 (システム設計者) 向け 約70人出席

(課長、係長、品証部門、設計者)

<ポイント>

「XDDPとは」だけでなく

- ・実際の開発現場の生の声 (苦労話など)
- ・現状の何が悪いのか?
仕様書体系、開発スタイル etc
- ・今までの障害 (実例) はXDDPで解決できるか?

2017年度の抜本的品質改善ワーキング発足

(1) XDDPを理解してもらうための説明会

今までの障害は、XDDPで防止できるか？

⇒ **できる！（実例で説明）**

例 1) 変更時、仕様書の該当頁の確認のモレによる処理モレ
障害作込み段階：ソフトウェア設計

	現状	問題点	XDDPでは
1	単純なコーディングミスが組合試験で見つかっている	ミスやモレは誰でもあることであるが、それをレビューで取り除けていない	変更設計書で「どのようにコードを修正するか」を記述し、レビューするため、コーディング前に気づきがある

例 2) 入力内容のあいまいな記述
障害作込み段階：システム設計

	現状	問題点	XDDPでは
2	ソフトウェアのモレやミスが、システム試験で見つかっている	インプットに対してどう変更するか確認・チェックができていない	ソフト設計者の視点で「変更要求仕様書」を記述し、変更内容を確認することで、インプットに対するモレ、ミス、勘違いを防ぐ

(2) 品質にあまり効果がない成果物を廃止

工事概要：238KL（新規5.9KL、改造3.2KL）

No	フェーズ	大項目	小項目	担当者（数字はHr）					合計	
				A	B	C	D	E		
1_0	設計	ソフトウェア設計	システム仕様理解	10.0	8.0	8.0	15.5	5.0	46.5	5.5%
1_1			流用元ソースコード解析	8.0	8.0	8.0	32.8	5.0	61.8	7.3%
1_2			ソフトウェア変更点一覧作成	10.0	16.0	8.0	4.0	10.0	48.0	5.7%
1_3			ソフトウェア変更仕様書作成	10.0	54.0	8.0	11.5	15.0	98.5	11.7%
1_4			デザインレビュー	15.0	6.0	2.0	6.0	4.0	33.0	3.9%
1_5		組合試験設計	組合試験仕様書作成	15.0	24.0	8.0	9.0	20.0	76.0	9.0%
1_6		プログラム設計	プログラム修正計画書作成 単体試験設計	25.0	40.0	24.0	63.0	40.0	192.0	22.8%
2_1	コーディング	コーディング	コーディング	18.0	32.0	24.0	39.3	30.0	143.3	17.0%
2_2			差分帳票作成		1.0		10.0		11.0	1.3%
2_3			DIFF作成（含む色塗り）	5.0	11.0	8.0	30.0	10.0	64.0	7.6%
2_4			ステップ数算出	1.0	0.1			2.0	3.1	0.4%
2_5			コードレビュー	20.0	4.0	2.0	8.0	2.0	36.0	4.3%
2_6			整合性チェック	1.0	2.0		15.0	1.0	19.0	2.3%
3_1			静的解析	静的解析	静的解析実施	0.2	0.1	1.0		0.5
3_2	静的解析結果の分析	1.0			2.0	1.0	4.0	1.0	9.0	1.1%



(3) 効果を定量的に示し続ける

**2016年度
XDDP適用工事 (抜粋)**

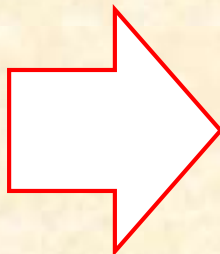
工事名	SW設計～組試まで	FL化率 (%)
	見積り時間に対する実績時間の割合 (%) * 実績時間/見積り時間	
A	110.5	83
B	72.7	93
C	57.1	67
D	72.9	90
E	60.3	90
F	164.4	85
G	96.9	90
H	86.4	85
I	70.0	100
J	70.9	100
K	118.0	100
L	109.0	100
M	113.5	96
N	99.9	86
O	99.1	77
P	78.0	88
Q	88.0	100
R	80.0	90

**34案件に適用
(案件適用率：約36%)**

**2017年度
XDDP適用案件 (抜粋)**

案件	組試	システム試験	FL化率
2017-01	1	0	98%
2017-02	9	3	85%
2017-03	3	11	83%
2017-04	0	1	67%
2017-05	1	0	94%
2017-06	13	1	73%
2017-07	10	3	50%
2017-08	0	2	87%
2017-09	0	0	100%
2017-10	11	18	66%
2017-11	0	0	100%
2017-12	0	0	100%
2017-13	0	0	100%
2017-14	0	0	100%
2017-15	1	0	98%
2017-16	5	4	67%
2017-17	0	0	100%
2017-18	2	7	93%
2017-19	0	0	100%
2017-20	1	0	80%
2017-21	1	0	67%
2017-22	0	0	100%
2017-23	0	0	100%
2017-24	1	0	92%

**72案件に適用
(案件適用率：約80%)**



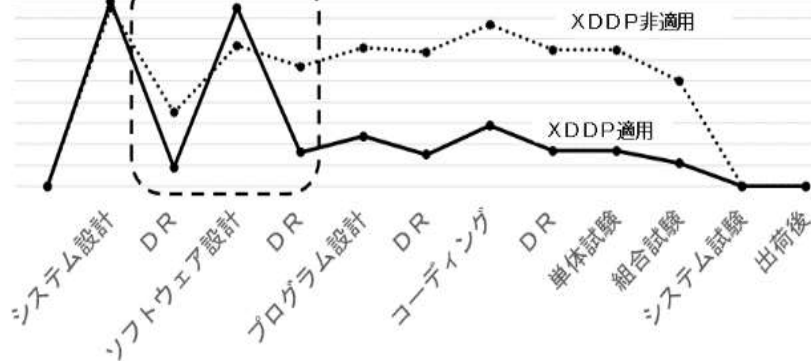
2018年度 XDDP工事適用率98%達成

(3) 効果を定量的に示し続ける

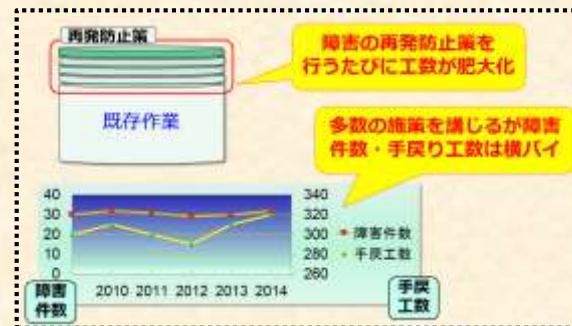
2017年度全工事 XDDP適用/非適用比較

XDDP適用案件では、システム設計及びソフトウェア設計後のDRで、非適用案件より多く障害を刈り取っている

誤り件数の推移



思い起こせば・・・



- ・データに変化が出た
⇒幹部、客先が興味を示した
効果を認め応援してくれた
- ・試行メンバーが自信を持った
⇒言葉に説得力が出てきた
- ・開発メンバーが
効果を実感した&評価された
⇒達成感、次への意欲、期待感

XDDPの他案件への展開（まとめ）

	2015年度	2016年度	2017年度	2018年度
適用案件数	6件	34件	72件	99件
案件適用率	7%	36%	80%	98%

案件適用率



**2018年12月
フォローアップ研修**

管理者向け
現在の位置、目指すべきところ
実務者向け
なぜXDDPなのか

形骸化防止

4. 2 仕様書改善の取組み

4. 2 仕様書改善の取組み

4. 2 仕様書改善の取組み

XDDPを導入することにより、**時間を手に入れた**

- ・品質が劇的に向上
- ・作業工数が劇的に減少



成功体験による改善意欲

新たな課題を見つけた = **仕様書の問題**

(1) 成果物が次回の開発に活かされていない

⇒差分を作成しっぱなし

(2) S/W仕様書が整備されていない

⇒差分開発の限界、効率が悪い

4. 2 仕様書改善の取組み

- (1) 成果物が次回の開発に活かされていない
 ⇒差分を作成しっぱなし (公式文書なし)
 ⇒**公式文書の作成**

公式文書

派生開発ドキュメント

ソフトウェア 機能仕様書

- ・ソフトウェアで実現する機能とその仕様を記載した文書
- ・完成したソースコードに対応して仕様を表す文書
- ・既存の「ソフトウェア仕様書」における機能仕様の位置づけ

変更要求仕様書

ソフトウェア 基本設計書 (方式設計書)

- ・システム設計で求められていることを実現するための、ソフトウェア全体に関わる設計戦略、設計ルール、仕組みを定義した文書

スペックアウト

ソフトウェア 詳細設計書

- ・基本設計の基づいた、各モジュールの詳細仕様を定義した文書

変更設計書

公式文書の作成① ソフトウェア詳細設計書

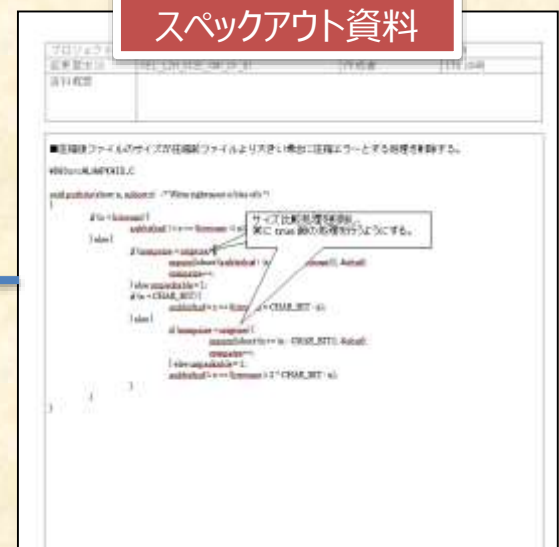
あらかじめ構成を決めておく

- 1 はじめに
 - 1.1 本文書の位置づけ
 - 1.2 対象読者
 - 1.3 本書の構成
 - 1.4 関連文書
- 2 LHAPK詳細設計
 - 2.1 設計制約
 - 2.2 外部仕様
 - 2.2.1 インタフェース設計
 - 2.3 内部設計
 - 2.3.1 設計モジュール構造図
 - 2.3.2 モジュール仕様
 - 2.3.3 相互作用図

改造毎に増やす



before



After



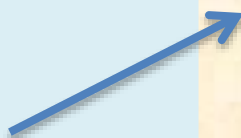
公式文書の作成② ソフトウェア機能仕様書

あらかじめ構成
を決めておく

XDDPで作成した変更要求仕様書
(before/after)

After部分

- 1 はじめに
 - 1.1 目的
 - 1.2 範囲
 - 1.3 定義、略語、短縮形
 - 1.4 参照
 - 1.5 概要
- 2 説明
 - 2.1 背景
 - 2.2 ソフトウェア機能の全体像
- 3 機能仕様
 - 3.1 外部インタフェース
 - 3.1.1 ユーザインタフェース
 - 3.1.2 ハードウェアインタフェース
 - 3.1.3 ソフトウェアインタフェース
 - 3.1.4 通信インタフェース
 - 3.2 機能仕様
 - 3.2.1 機能A
 - 3.2.2 機能B



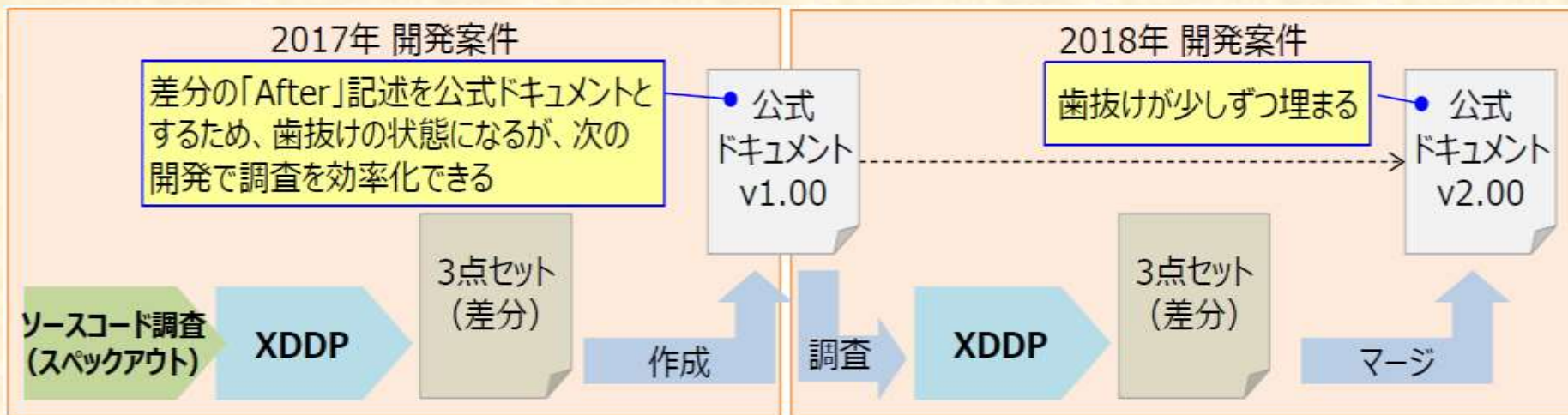
改造毎に増やす

4. 2 仕様書改善の取組み

成果物が次回の開発に活かされていない

⇒公式文書の作成（2017年度から活動）

現在、ソフトウェア詳細設計書の作成を試行中



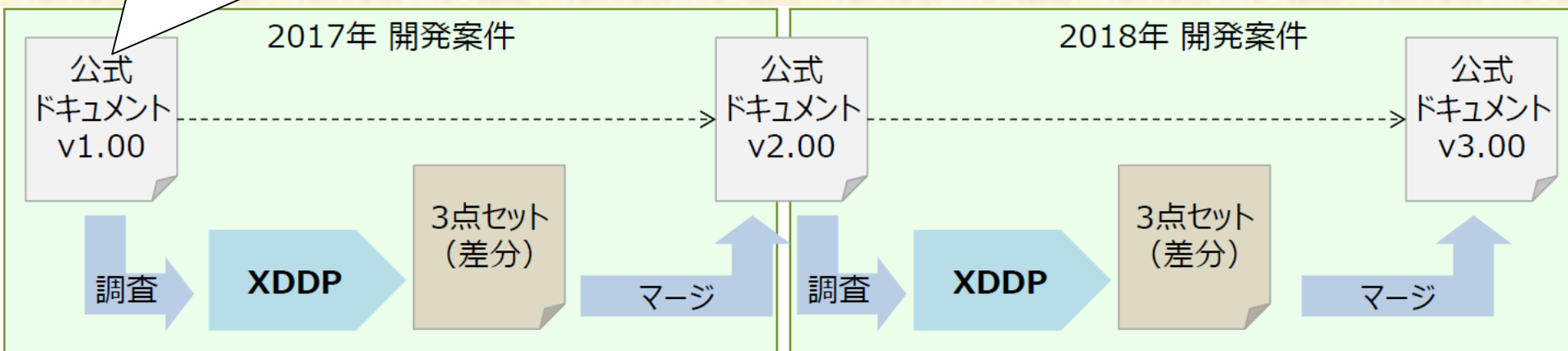
4. 2 仕様書改善の取組み

(2) S/W仕様書が整備されていない
⇒差分開発の限界、効率が悪い

2017年度に新規システムの開発案件が発生
(他案件を流用した開発)

⇒この機会にS/W仕様書の整備とプロダクトライン開発

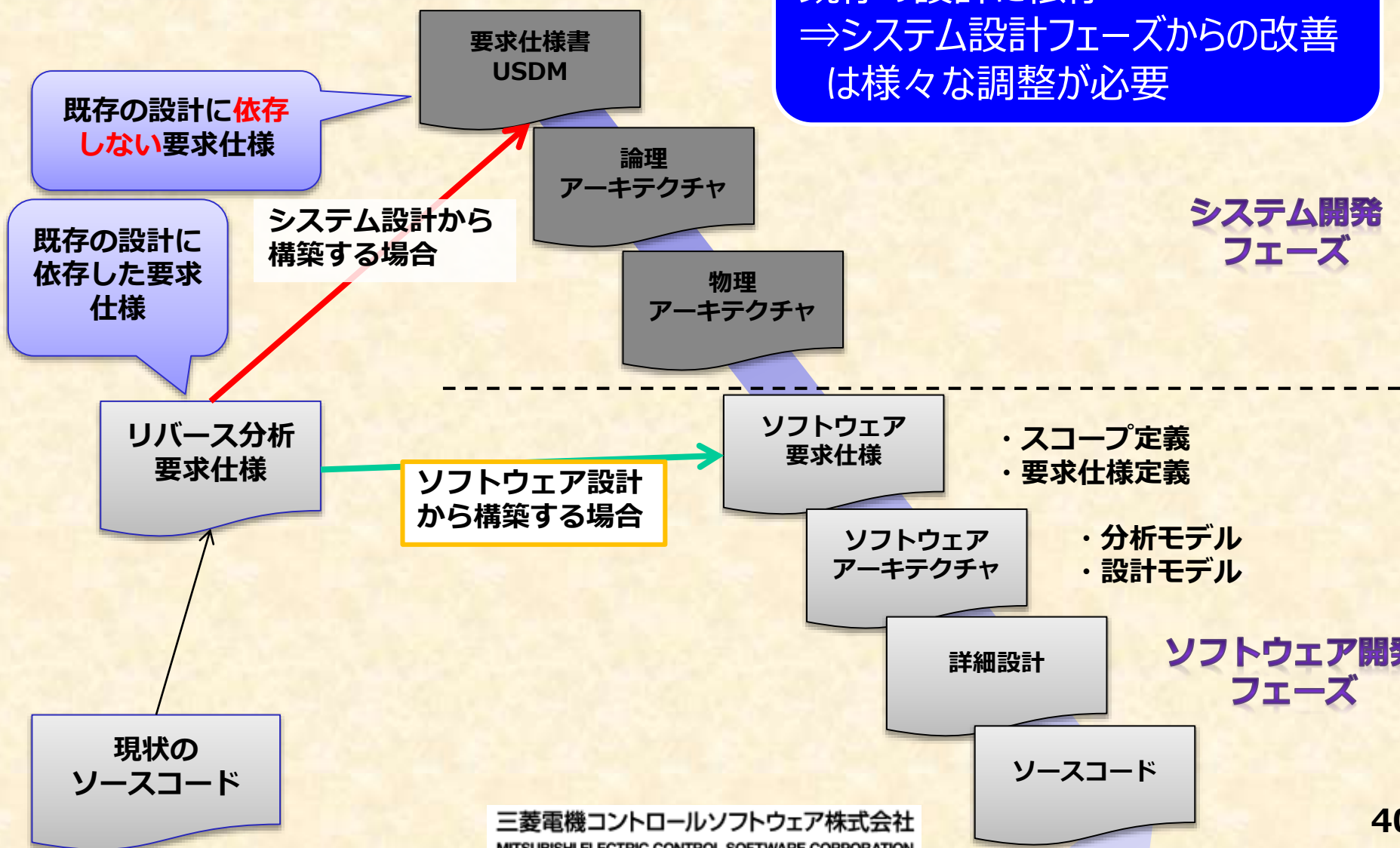
もし公式ドキュメントが
あった場合



4. 2 仕様書改善の取組み

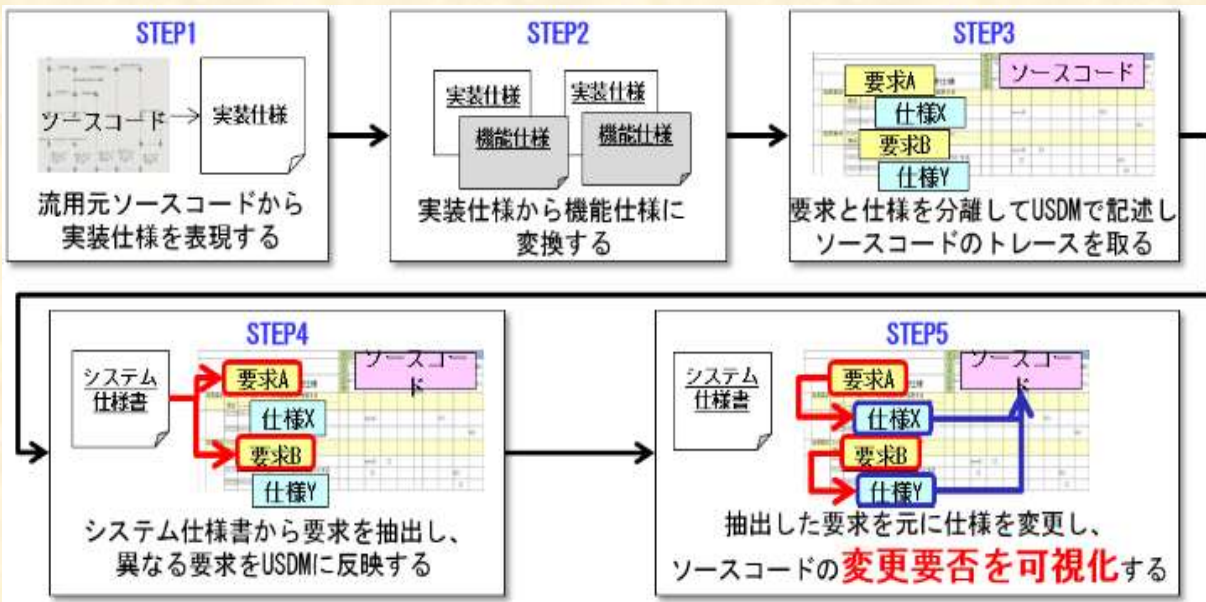
S / W仕様書の整備

システム設計は受注範囲外
 既存の設計に依存
 ⇒システム設計フェーズからの改善
 は様々な調整が必要



4. 2 仕様書改善の取組み

プロダクトライン開発を見据えたS/W仕様書の整備

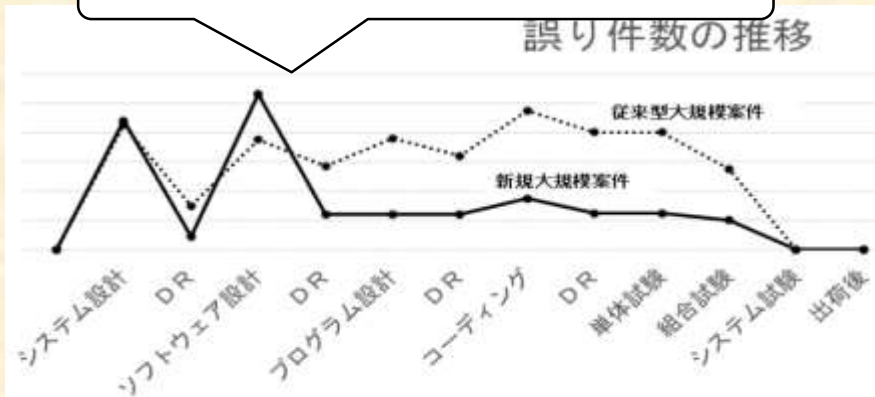


開発規模

新規： 17.71KL
 改造： 88.17KL
 流用： 213.38KL

**全211機能
 について完成**

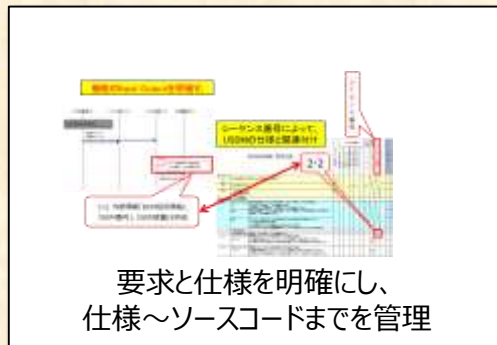
上流フェーズのDR活性化



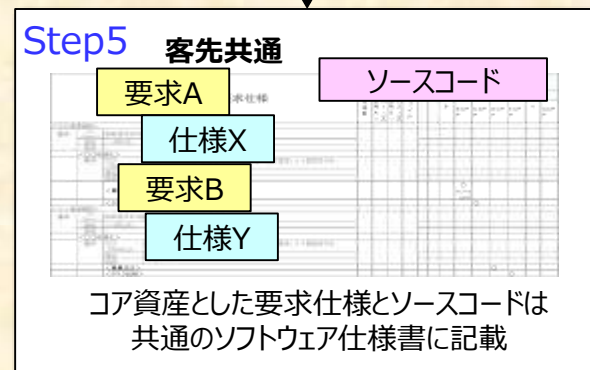
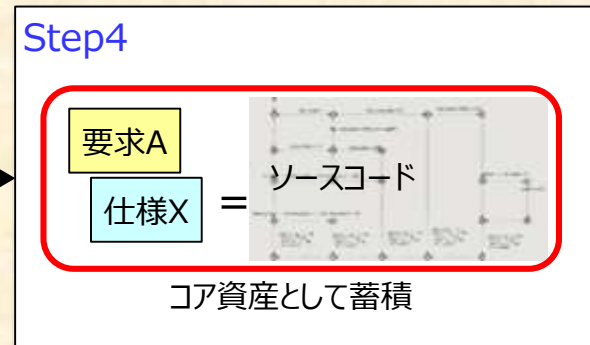
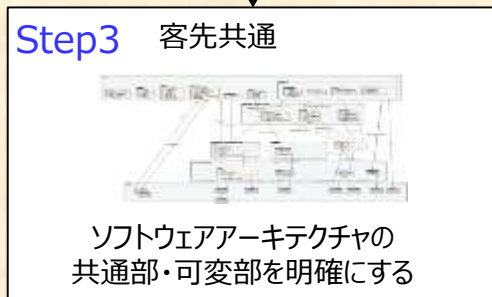
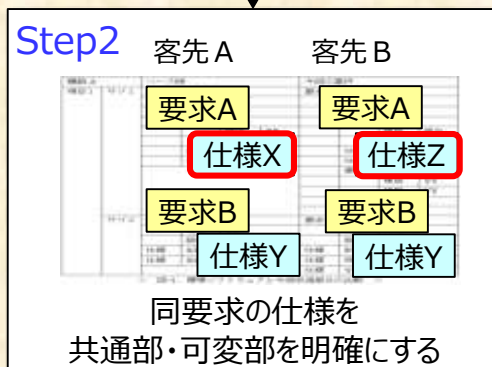
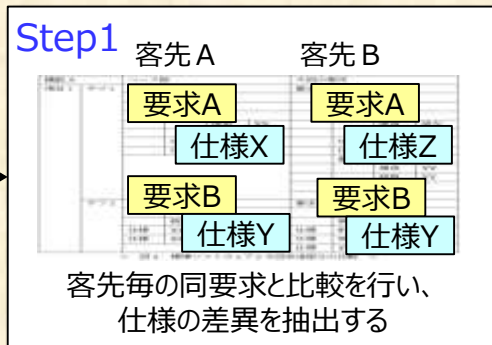
**品質向上
 手戻り削減**

4. 2 仕様書改善の取組み

プロダクトライン開発の取組へ (現在進行中)



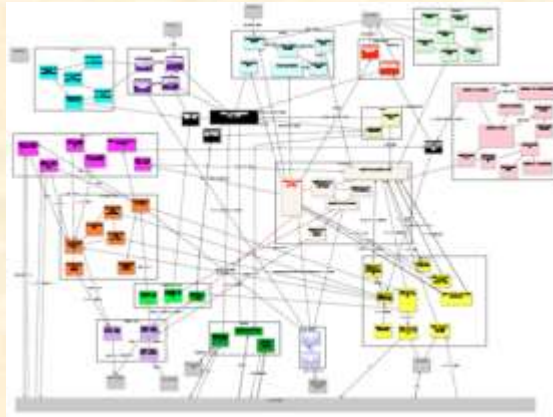
**プロダクトライン
開発の
道が拓けた**



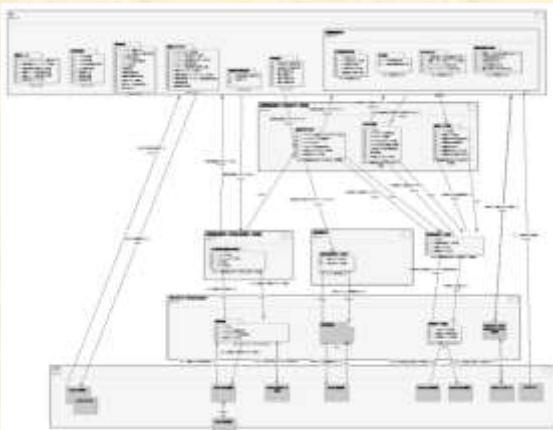
コア資産として順次蓄積

4. 2 仕様書改善の取組み

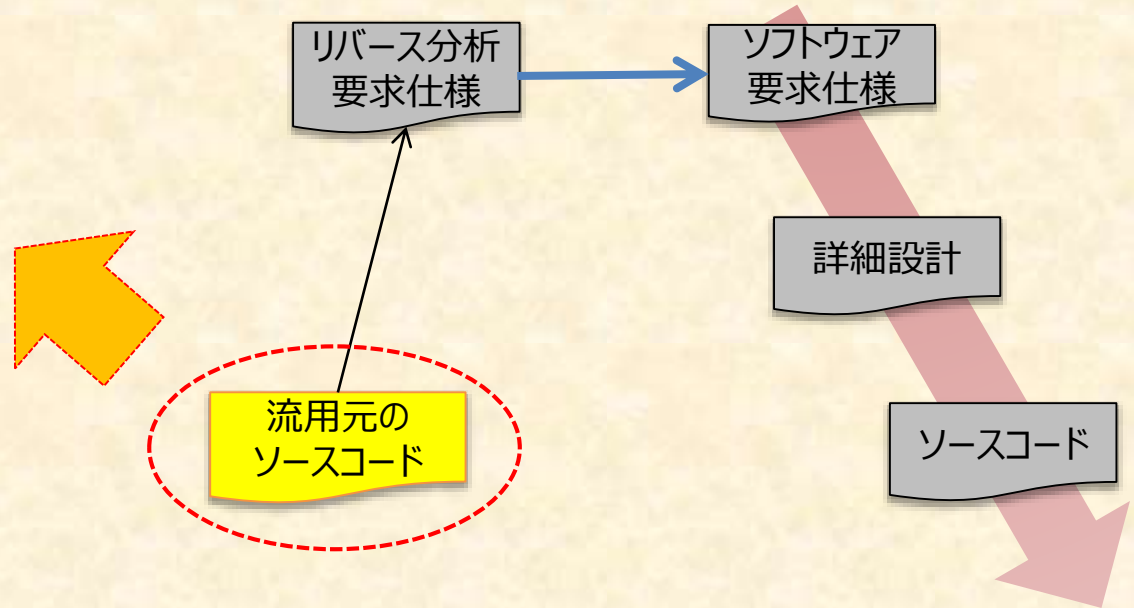
S/W仕様書の整備をしていてわかったこと



現状のアーキテクチャ？



あるべきアーキテクチャ



要求・仕様の整理により、
現状のソフトウェアアーキテクチャの
複雑さが、問題であることが分かった。

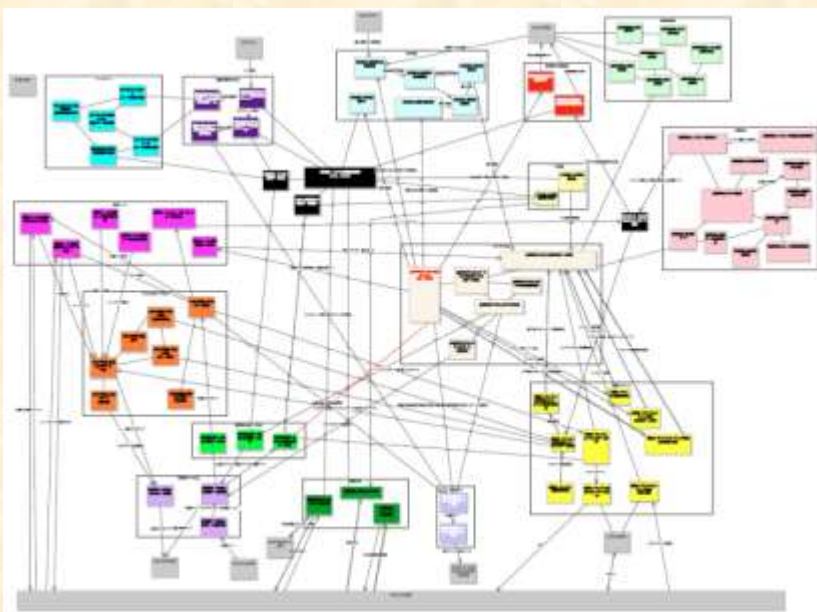
4. 3 設計改善の取組み

4. 3 設計改善の取組み

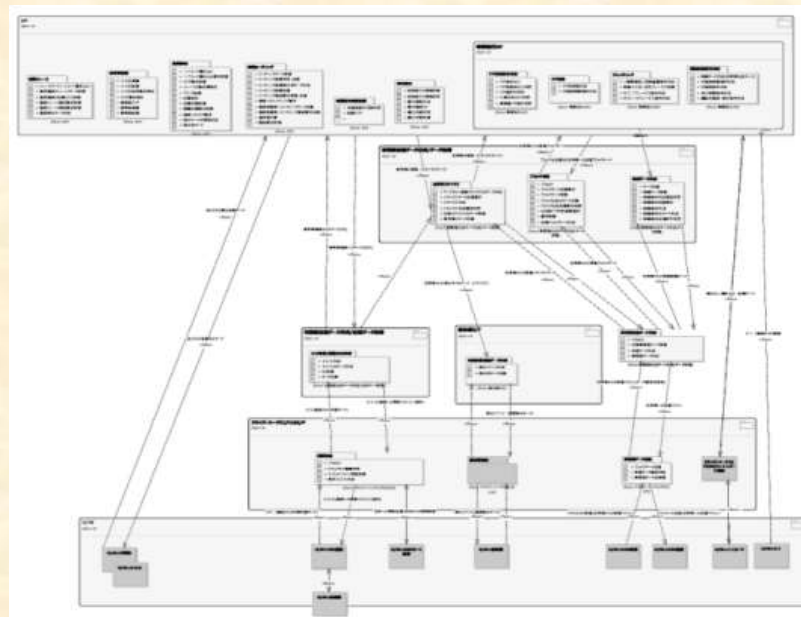
派生開発が続きアーキテクチャが崩れている

- ・ S / W構造が複雑
- ・ 複数案件で S / W資産が重複（クローンソフトなど）

リバース分析した時の結果

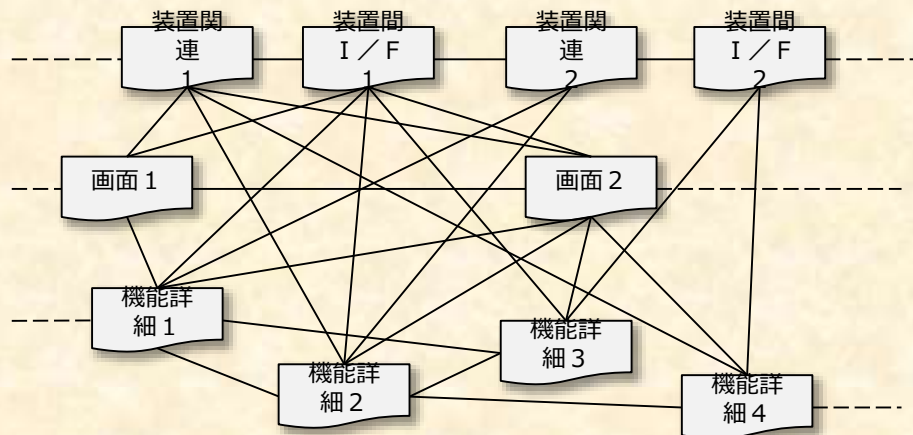


理想形



4. 3 設計改善の取組み

インプットの問題

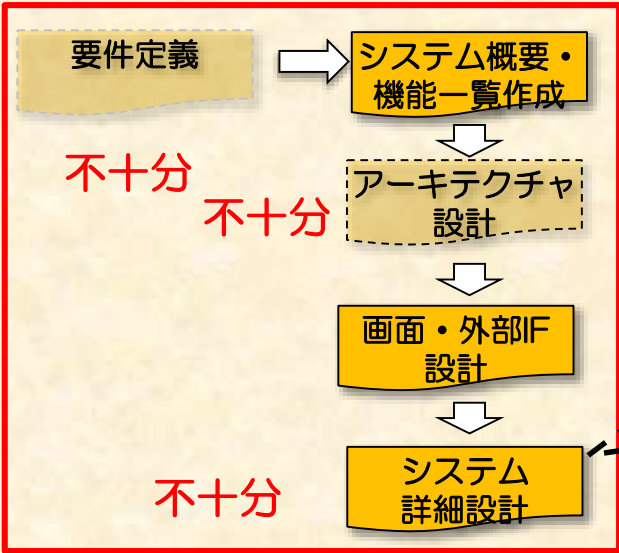
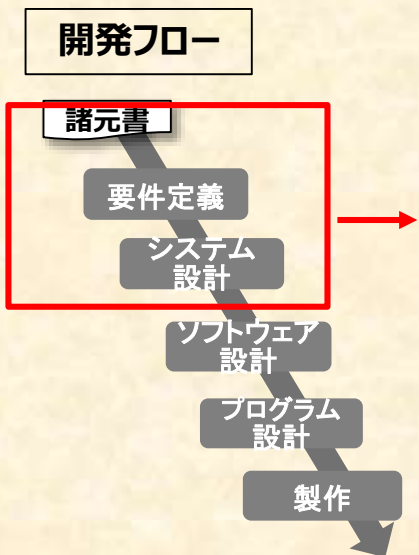


- 多くの仕様書から関連を整理することに時間がかかる
- 要件定義や理由が曖昧
- **QA件数：約200件/工事**

課題

機能仕様を理解しやすい
システム仕様書の整備

プロセスの問題



複雑なS/W構造の一因

課題

最適なアーキテクチャ設計

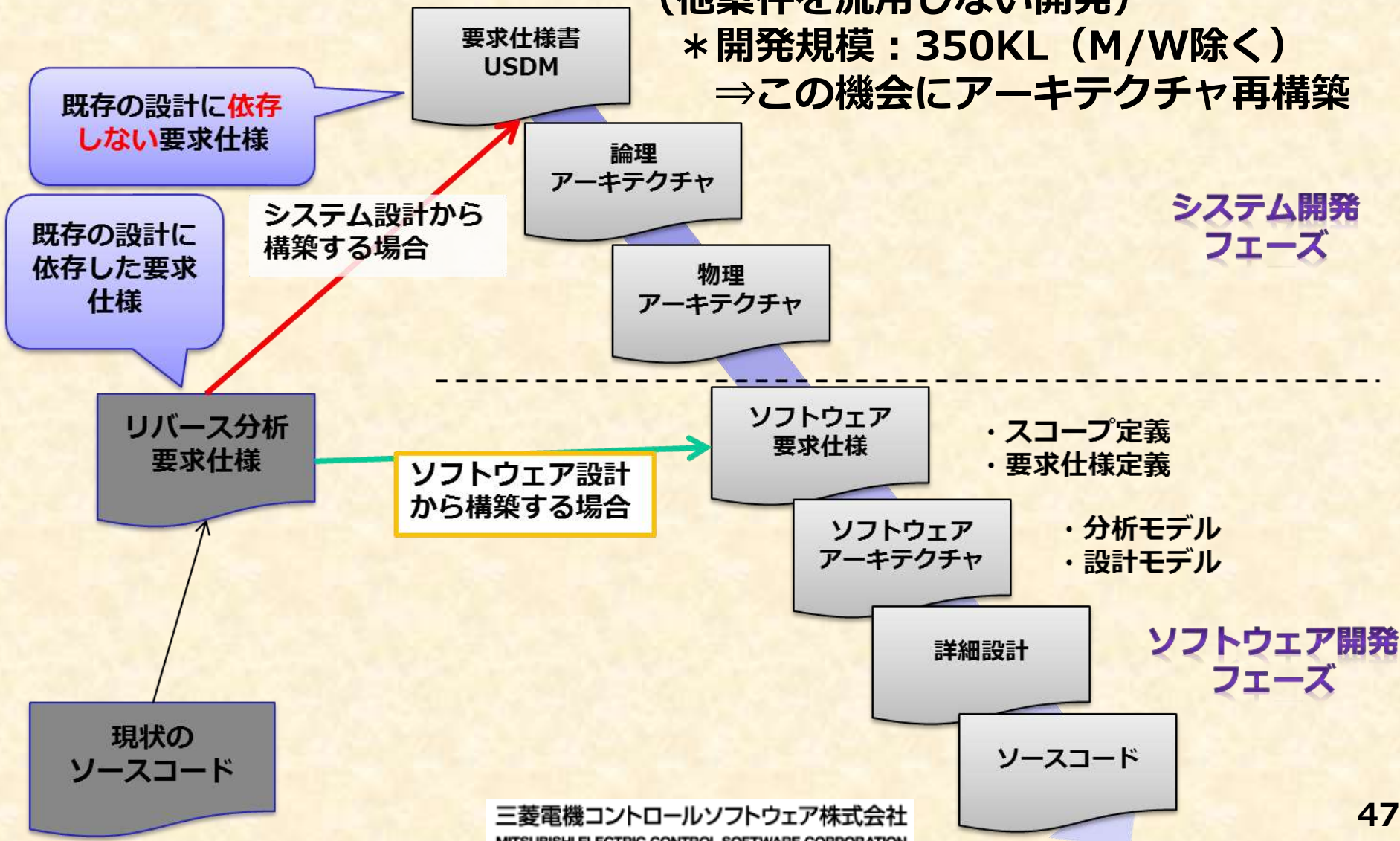
S/W構造設計が不十分なままシステム詳細設計を実施

4. 3 設計改善の取組み

2018年度新規システムの開発案件が発生
(他案件を流用しない開発)

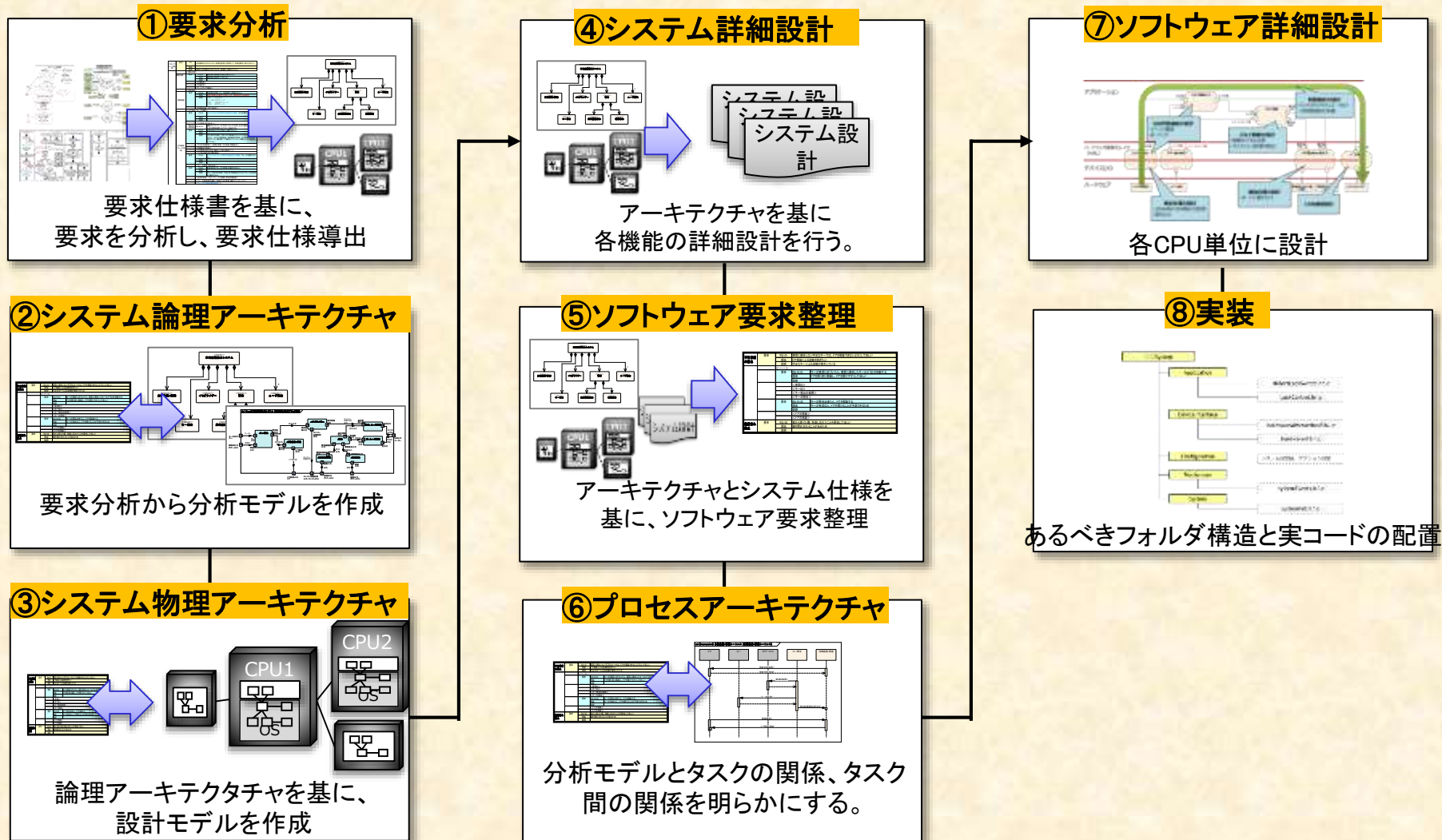
* 開発規模：350KL (M/W除く)

⇒この機会にアーキテクチャ再構築



4. 3 設計改善の取組み

設計の課題 ⇒ アーキテクチャ再構築（現在進行中）

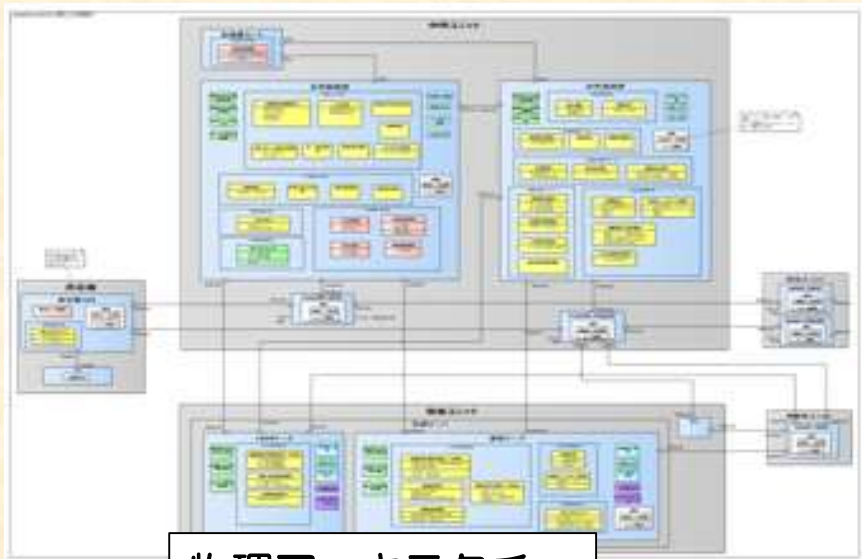
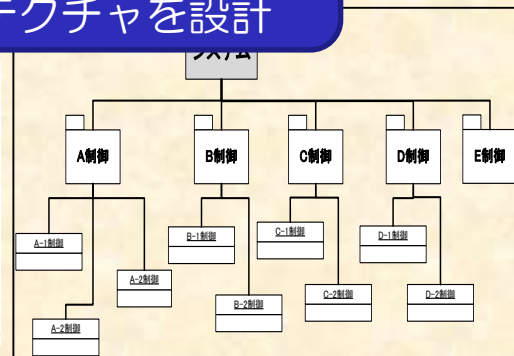
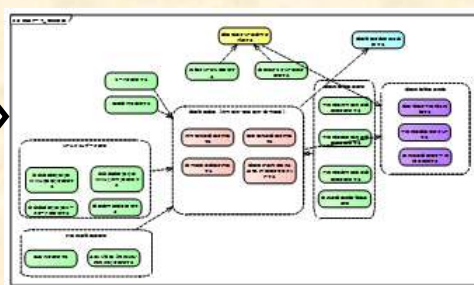
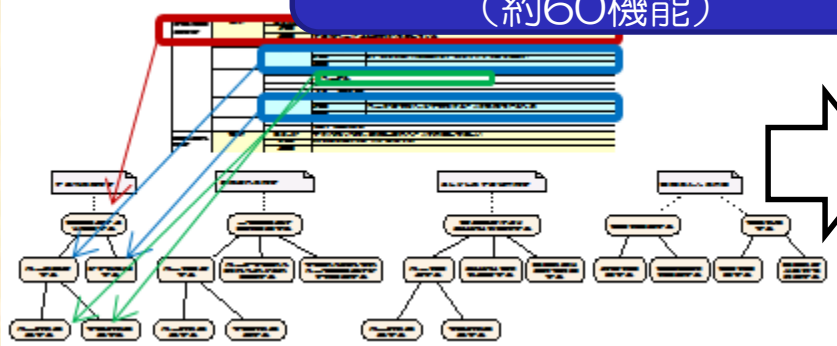


4. 3 設計改善の取組み

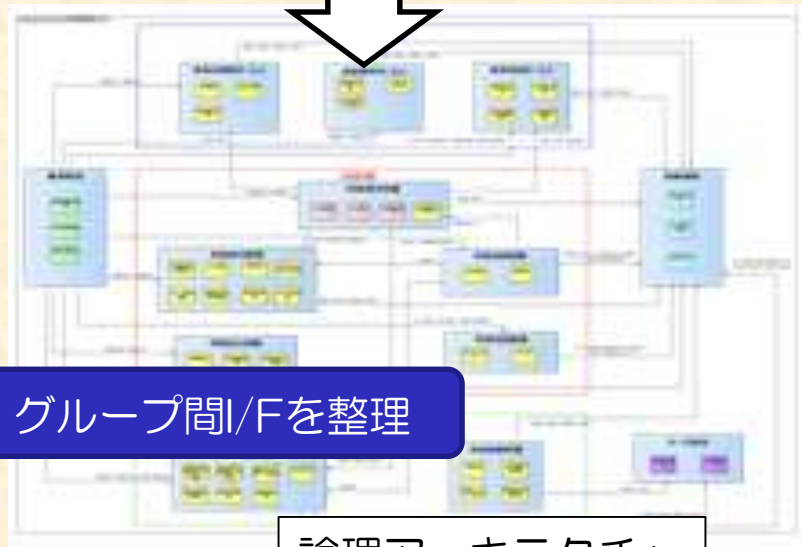
設計の課題 ⇒ アーキテクチャ再構築（現在進行中）

客先要求書から要求分析し
USDM形式で機能を導出
(約60機能)

役割・責務で機能を分割・統合
⇒14の機能グループに整理し
論理アーキテクチャを設計



物理アーキテクチャ



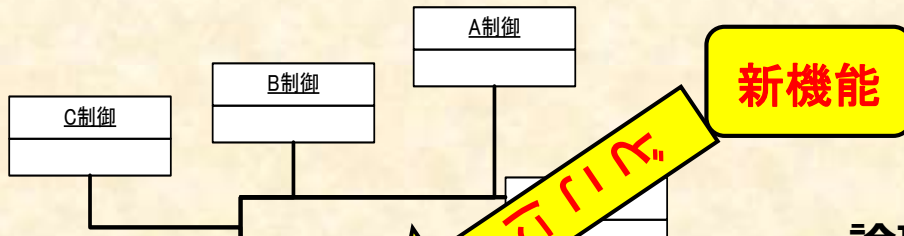
グループ間/Fを整理

論理アーキテクチャ

4. 3 設計改善の取組み

設計の課題 ⇒ アーキテクチャ再構築（現在進行中）

従来



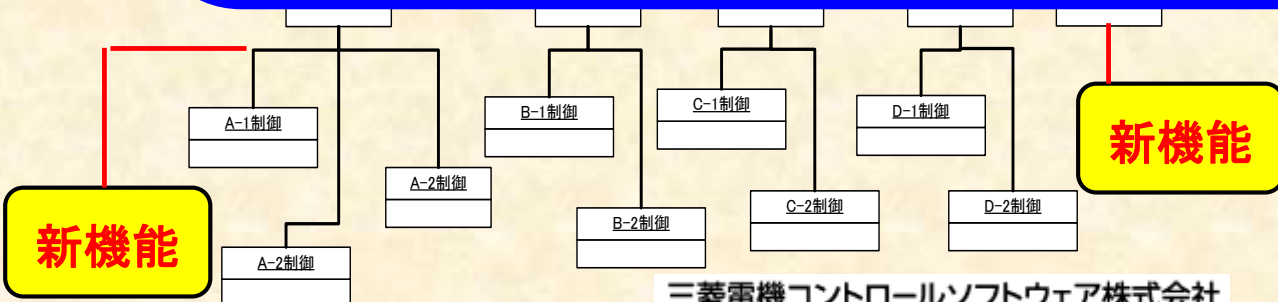
論理的なコンポーネント構成がなく、
モジュール配置

<アーキテクチャ再構築効果>

- ・従来よりもS/W構造が最適化
16人が5チームに分かれ、アーキテクチャ設計をもとに、
S/W設計製作を実施し、統一されたS/W構造が構築できた
- ・可読性が向上し、コードレビューが効率的になった

取組

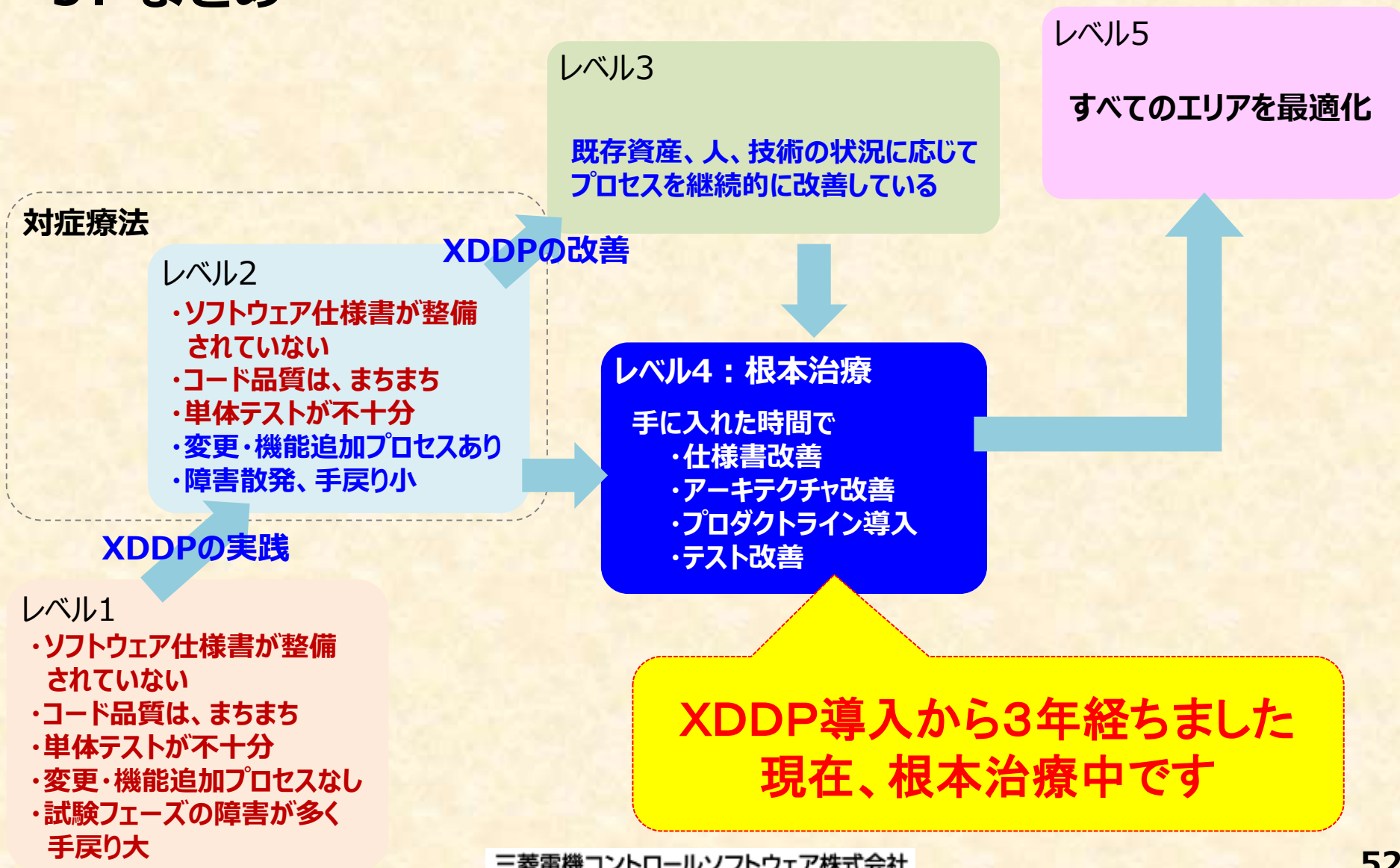
（ス
ない）



5. まとめ

5. まとめ

5. まとめ



ご清聴ありがとうございました