大規模／多拠点ソフトウェア開発プロジェクトにおける品質改善への取り組み事例

Examples of efforts to improve quality in large / multi-site software development projects

岩崎幸弘
日本電気株式会社 テレコムキャリア品質推進本部

発表要旨：

大規模かつオフショアを含む多拠点によるソフトウェア開発プロジェクトにおいて、品質問題が発生し、納期遅延、開発コスト増などの影響を受けた。

継続開発プロジェクトにおいて同じような問題を発生させないために、品質問題の根本原因を分析したところ、以下が主要因であると判断した。

（品質悪化要因）

① 網羅性/トレーサビリティの問題
② レビュープロセスの問題
③ プロセス理解の問題

これらの問題を改善するために、以下の品質改善施策を検討、実施した。

（品質改善施策）

① トレーサビリティの強化
② テスト項目網羅の見直し
③ 中間レビューの実施
④ オフショア／協力会社への開発プロセス教育展開

これら施策により、継続開発プロジェクトにおいて、大きな成果を得ることができた。

当改善活動を経験事例として発表する事とする。

キーワード：

プロセス改善、レビュー手法、トレーサビリティ、オフショア開発

想定している聴衆

・ソフトウェア品質管理者
・オフショア活用中、もしくは活用に興味のあるプロジェクト管理者

発表者の紹介（全角１００文字）：

日本電気株式会社（ＮＥＣ）勤務。テレコムキャリア品質推進本部所属。

主業務として、ネットワーク管理ソフトウェア（ＥＭＳ／ＮＭＳ）の品質管理／改善、および開発プロセス改善活動に従事。

Orchestrating a brighter world

NEC

[SQiP2017 経験発表]

大規模／多拠点ソフトウェア開発プロジェクトにおける品質改善への取り組み事例

２０１７／９／１５

日本電気株式会社　テレコムキャリア品質推進本部
岩崎幸弘

Orchestrating a brighter world

未来に向かい、人が生きる、豊かに生きるために欠かせないもの。
それは「安全」「安心」「効率」「公平」という価値が実現された社会です。

NECは、ネットワーク技術とコンピューティング技術をあわせ持つ
類のないインテグレーターとしてリーダーシップを発揮し、
卓越した技術とさまざまな知見やアイデアを融合することで、
世界の国々や地域の人々と協奏しながら、
明るく希望に満ちた暮らしと社会を実現し、未来につなげていきます。

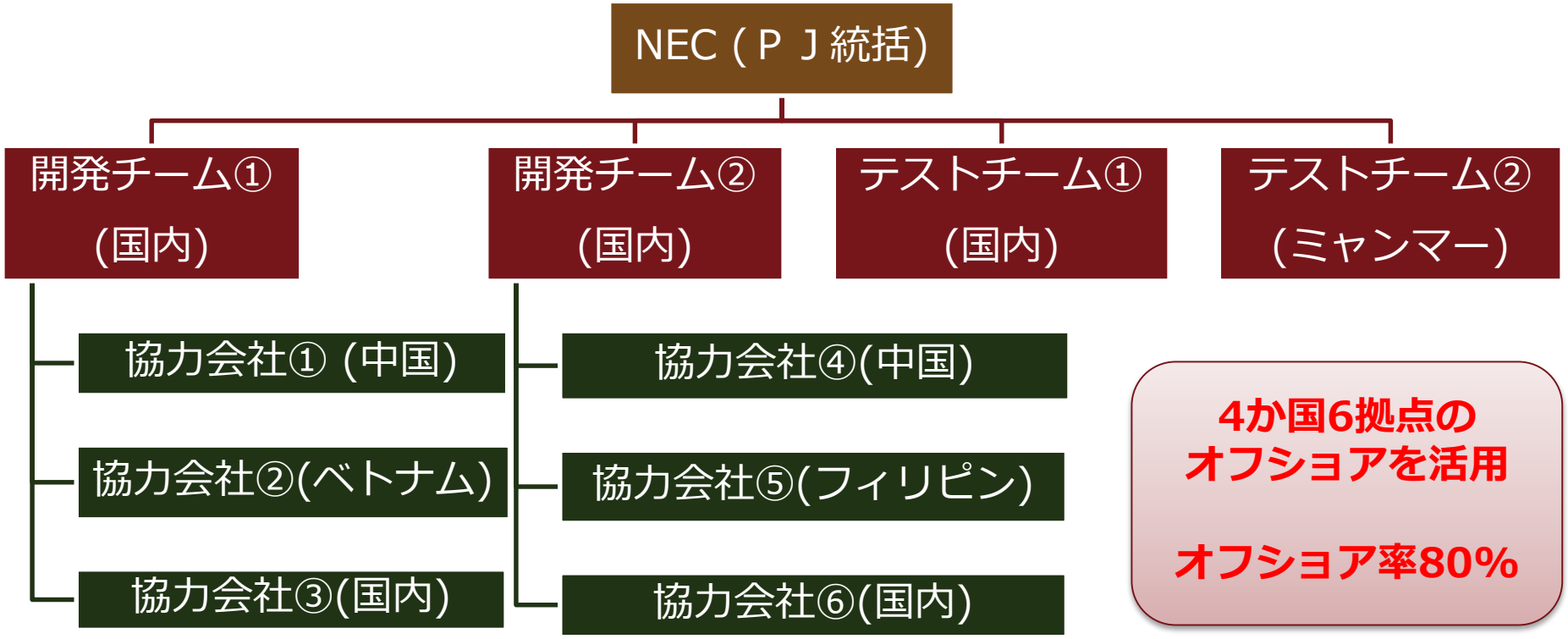
Agenda

- 事例(プロジェクト)概要
- 発生した問題の概要および品質悪化要因の分析
- 品質強化施策の紹介
  - 品質強化施策検討結果
  - 施策①：トレーサビリティの強化
  - 施策②：テスト項目網羅の見直し
  - 施策③：レビュー強化
  - 施策④：オフショア/協力会社への開発プロセス教育展開
  - 品質施策の相互関連性
  - 品質施策の履行確認
  - 施策導入にあたり苦労した点
- 施策効果(定量効果/定性効果)
- まとめ

事例（プロジェクト）概要

事例(プロジェクト)概要(1/2)

- 当プロジェクトは、ネットワーク管理ソフトウェアの新規開発プロジェクトであり、以下のようなオフショアを含む複数拠点での開発が特徴となっている。
- 開発はクライアント／サーバー型開発で進められ、クライアント、サーバーいずれもオフショア拠点がメインに開発を実施。



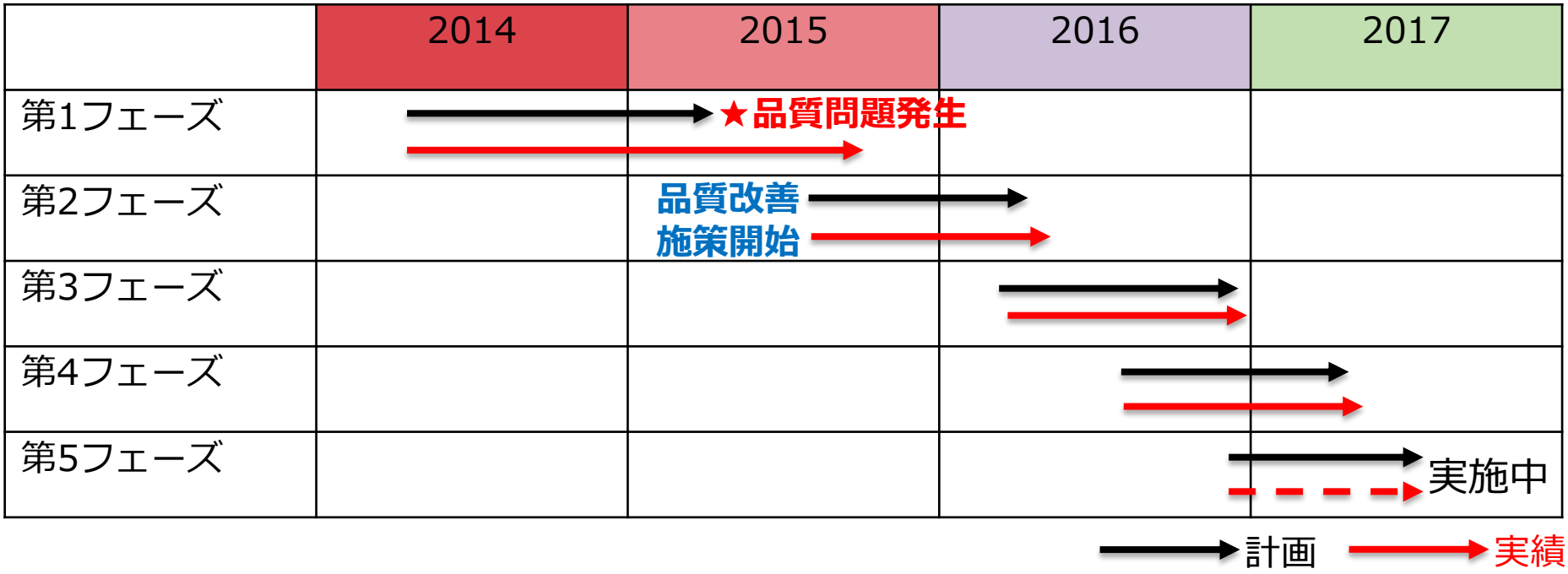
4か国6拠点のオフショアを活用

オフショア率80%

事例(プロジェクト)概要(2/2)

・当プロジェクトは2014年に第1フェーズ開発を開始。
以降、半年サイクルでの開発／リリースを継続中。
(第1フェーズでは数MLOCの開発を実施、第2フェーズ以降も
数百KLOC程度の開発を継続)

→第1フェーズ開発にて品質問題が発生したため、品質強化を検討。
第2フェーズ開発以降に改善施策を適用。



発生した問題の概要および品質悪化要因の分析

プロジェクトで発生した問題の概要

・当プロジェクトのPhase1開発において、設計工程以降、品質問題が発生。
その結果、プロジェクトへの多大な影響が発生した。
→Phase2以降の品質を向上させるため、プロジェクト反省会を開催し
品質悪化の要因のヒアリングを実施

仕様設計問題

・検討漏れ/記述不足が多発
・拠点毎に記載レベルが不統一な為書き直し発生
・レビュー不足、適切なレビューア不参加などの非効率なレビュー

仕様／設計が収束せず

テスト問題

・コードレビューや単体テストで検出すべきバグのシステムテストへの流出
・異常系や複合動作でのバグ多発
・API/インターフェースのバグ多発

テストでバグ頻発

進捗遅れ(25%)

後戻りコスト発生(PJ総費用の30%)

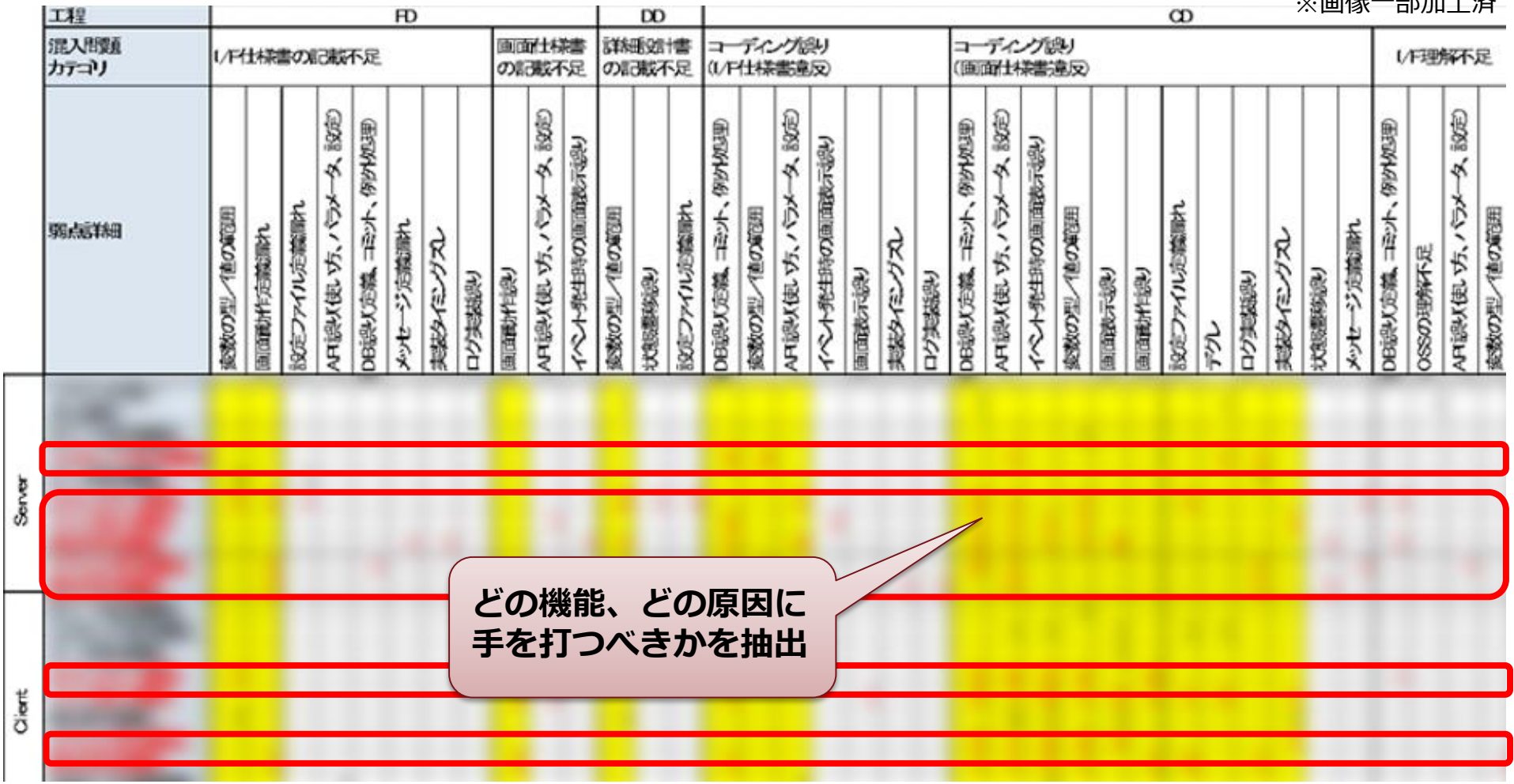
一部機能のリリース延伸

品質悪化要因分析(1/4)

・反省会で挙がった品質悪化要因の裏付け、および原因深堀の為、
Phase1開発の各工程で発生したバグの傾向を分類し特異点を抽出した

バグ分類表

※画像一部加工済



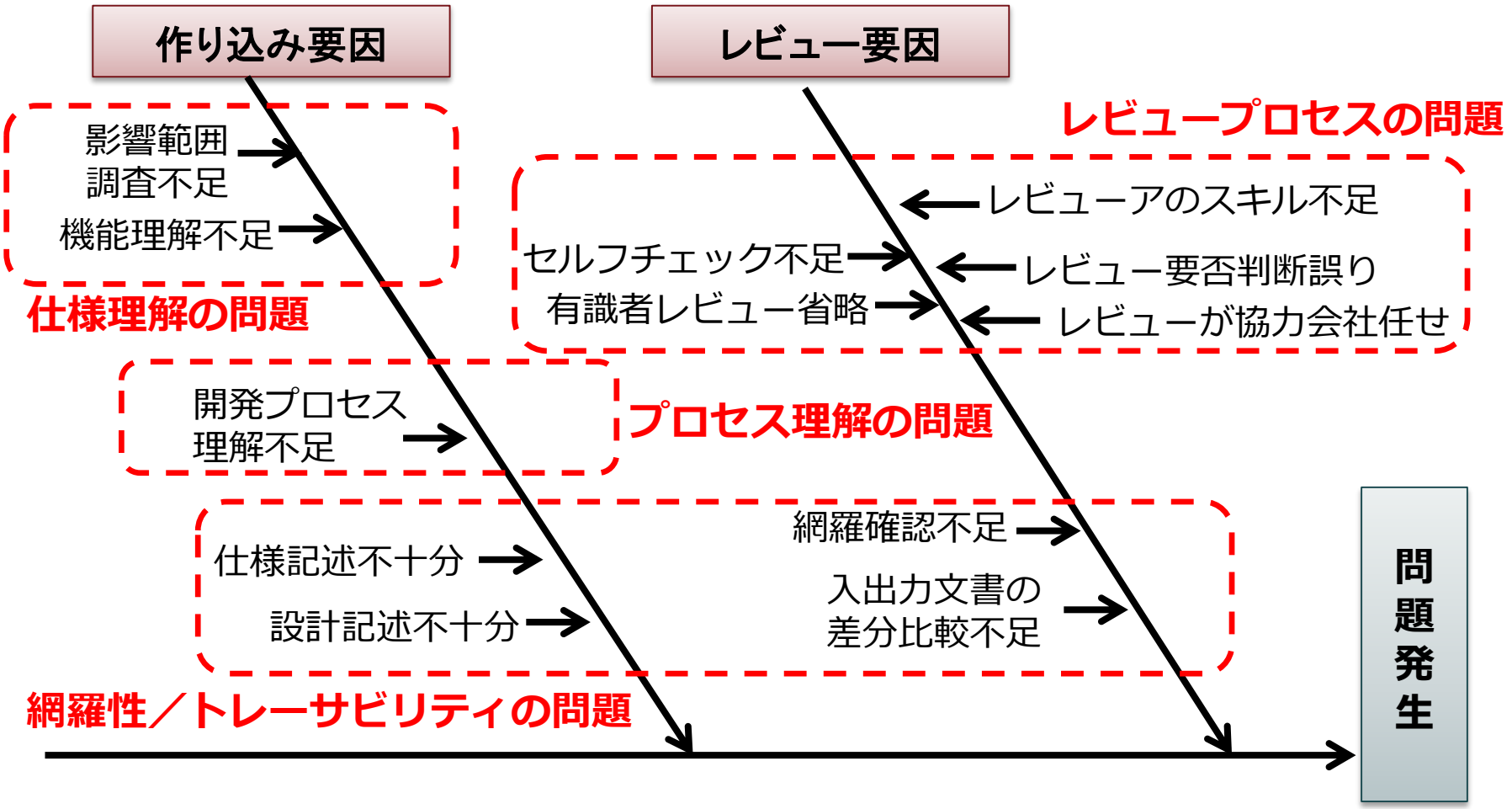
品質悪化要因分析(2/4)

・抽出した特異点について共通的な傾向の整理を実施

発生した問題	共通傾向
・インターフェース、APIの利用方法の誤り ・クライアント／サーバー間の送受信の情報不一致 ・準正常系、異常系、競合動作の定義漏れ ・クラスやオブジェクトの型の不一致 ・データベースの利用方法の誤解 ・画面表示の体裁不統一 ・画面動作定義不足(右クリックメニューなど) ・メッセージ表示漏れ ・ログ出力漏れ／誤り	仕様/設計記述不足
・セルフチェックリストに記述のある項目のチェック漏れ ・単純なコーディング作法の誤り	セルフチェック不足
・有識者のレビュー不参加 ・有識者の設定誤り（知識のないメンバが有識者となっていた）	有識者レビュー不足
・修正漏れの発生 ・誤修正（デグレード）の発生 ・試験項目の定義漏れ（リグレッション含む）	影響範囲調査不足
・開発拠点毎に成果物の体裁が異なる ・開発担当者にて、ドキュメント作成を省略(ルール違反)	開発プロセス理解不足

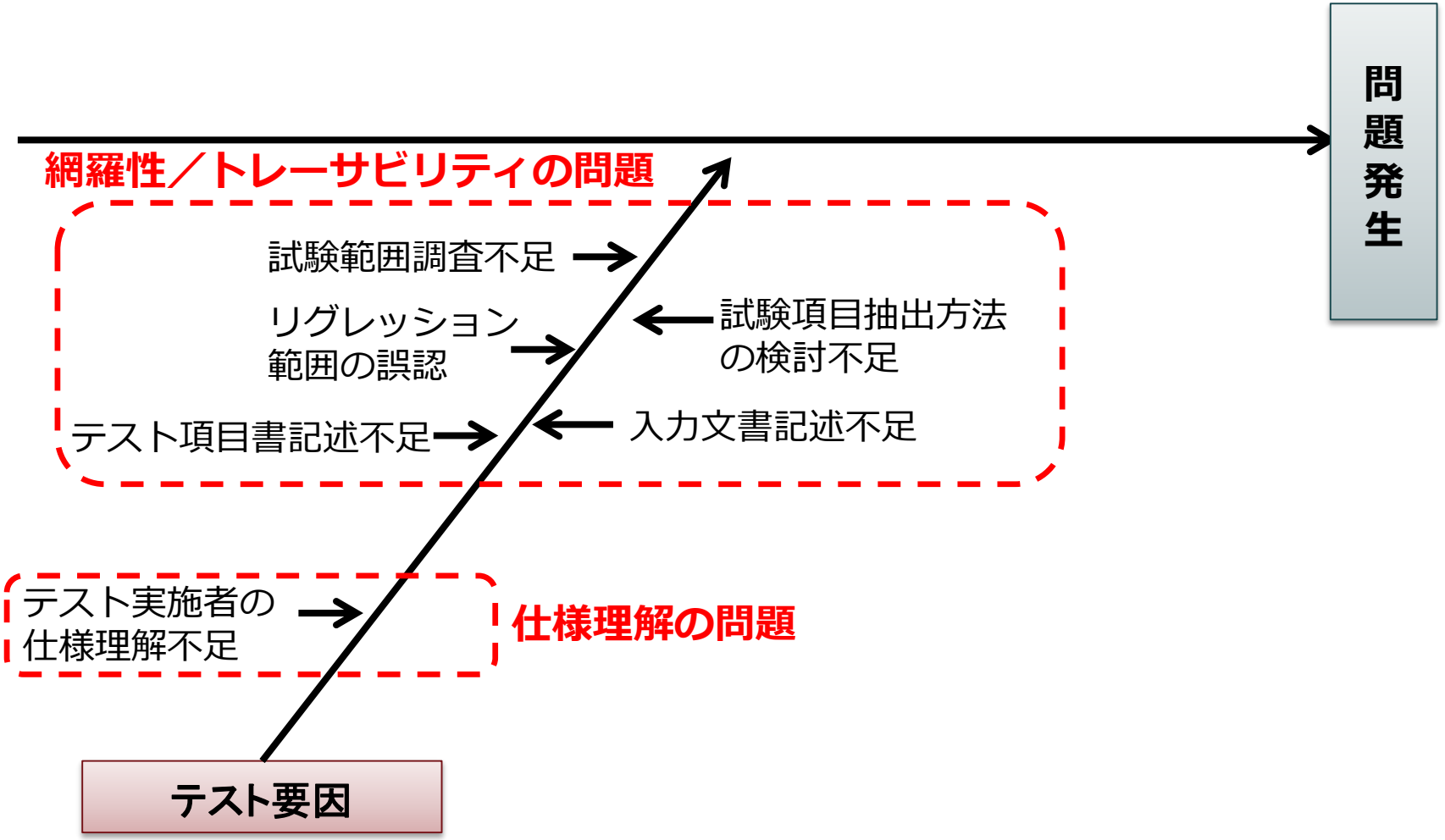
品質悪化要因分析(3/4)

・抽出した傾向について、作り込み、レビュー、テストの観点から
個々になぜなぜ分析を実施したうえで、品質悪化の根本要因を抽出



品質悪化要因分析(4/4)

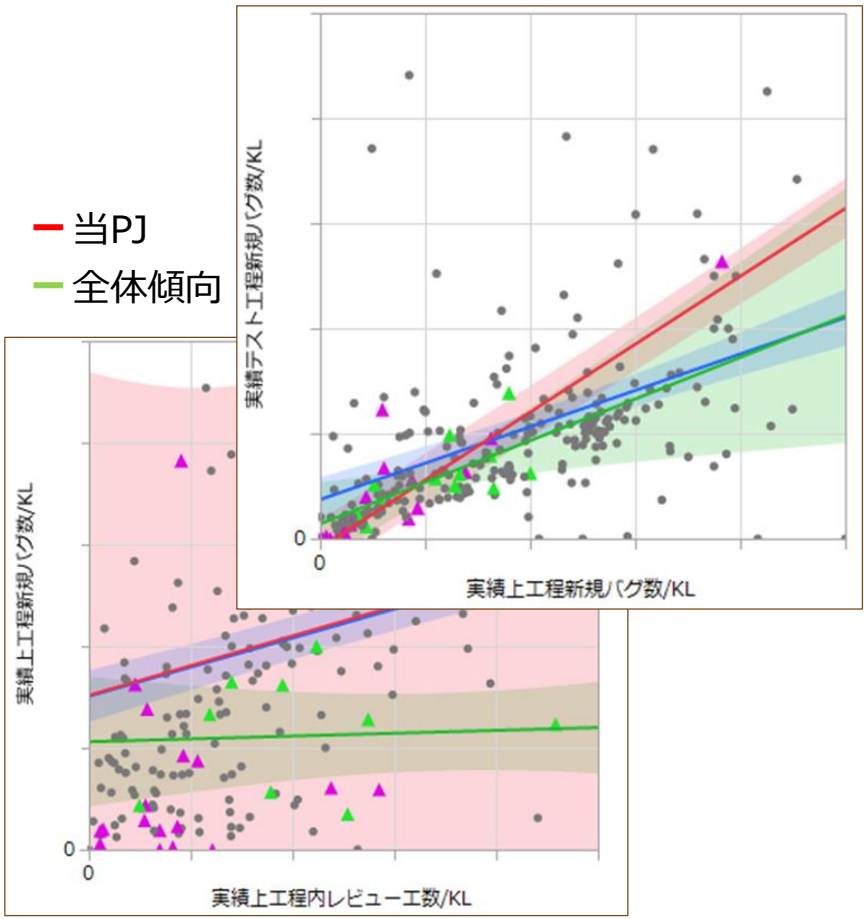
・抽出した傾向について、作り込み、レビュー、テストの観点から
個々になぜなぜ分析を実施したうえで、品質悪化の根本要因を抽出



品質悪化要因分析(定量分析) (1/2)

・品質悪化要因について、定量的な視点からの検証も実施。
　→統計解析を活用し、社内標準データとの比較を実施

分析例(抜粋) 開発機能毎にプロット



以下の観点について全社傾向および自部門内の他プロジェクトとの傾向の差異を比較

- 開発規模－工数
- 工数－開発期間
- 開発規模－全摘出バグ
- レビュー工数－上工程摘出バグ
- テスト項目数－テスト工程摘出バグ
- テスト工数－テスト工程摘出バグ
- 上工程摘出バグ－テスト工程摘出バグ

比較の結果、当プロジェクトについて、
・機能管理単位が大きいほど下流へ流出
・上工程のレビュー効率が悪い
傾向が特に強く見えた。

品質悪化要因分析(定量分析) (1/2)

・前項の全体傾向に加え、各拠点毎／各工程毎の品質データ分析を行い、特異点の深堀を実施
　→オフショアにて詳細設計、製造起因のバグがテストに多く流出。
　　原因としては上位仕様の理解不足、成果物のチェック不足に起因。
　　(統計データでの傾向の主要因がオフショアにあると判断。)

詳細設計工程分析例(抜粋)

成果物作成に費やした作業量

どの工程で作り込んだバグか？

会社名		リリース	平均／開発規模	平均／文書密度	平均／レビュー密度	平均／バグ密度_混入_既存	平均／バグ密度_混入_仕様化	平均／バグ密度_混入_基本設計	平均／バグ密度_混入_詳細設計
[redacted]									

※画像一部加工済

品質悪化要因分析(まとめ)

・当プロジェクトの品質改善の為、以下の取り組みを実施。
①現場に対し、品質悪化要因をヒアリング
②実際に出たバグを分類し、特異点の抽出／整理を実施
③特異点に対し、原因の深堀を実施
④統計的観点からの品質悪化傾向の分析を実施

これらの分析の結果、当プロジェクトの主な失敗要因は
以下であると判断。
- 網羅性/トレーサビリティの問題
- レビュープロセスの問題
- プロセス理解の問題

この３つの観点に基づき、品質強化施策を実施する事とした。

品質改善施策紹介

品質改善施策検討結果

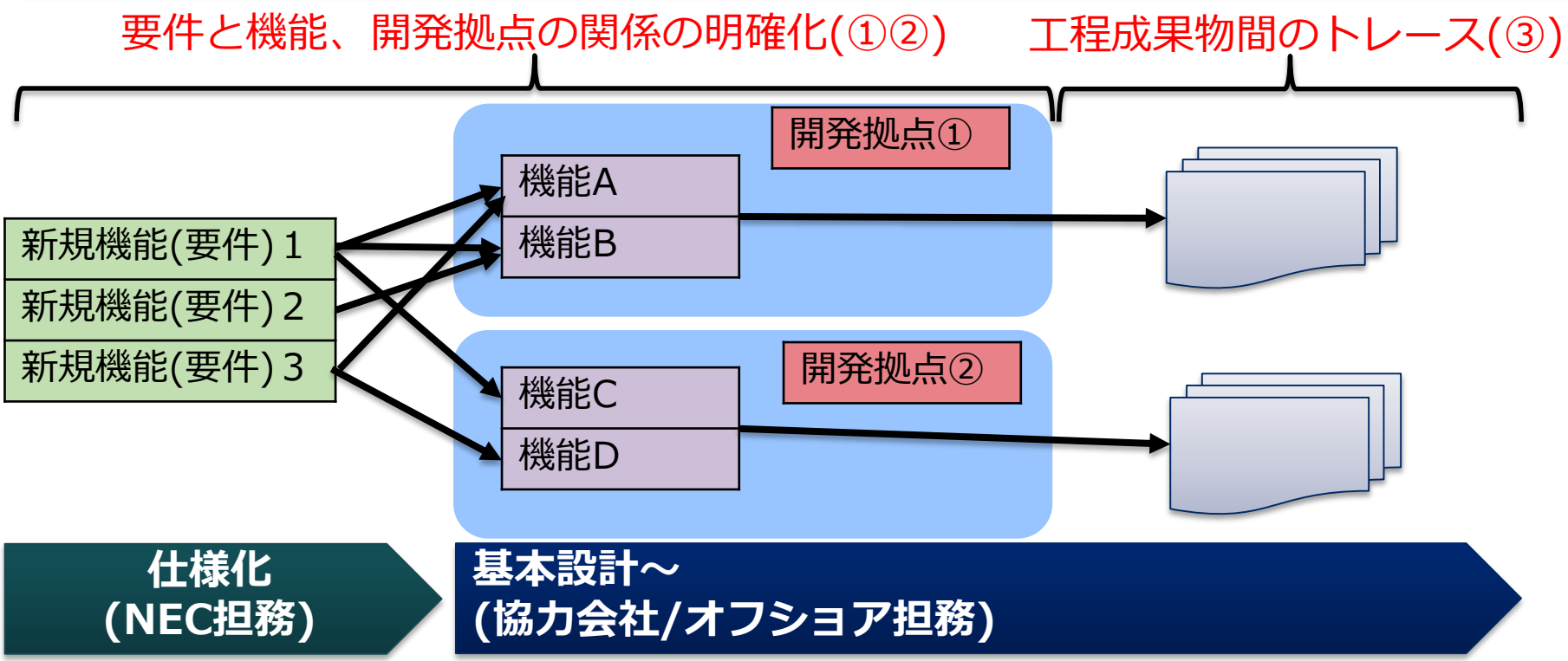
・これまでの分析を基に、以下の施策を実施する事とした。

今回実施した主要改善施策

課題	施策	施策概要
網羅性／トレーサビリティの問題	①トレーサビリティの強化	・要件漏れ防止の為、トレーサビリティ担保を強化。 ・影響範囲分析の漏れ防止の為、要件とモジュールの対応マトリクスを作成し、影響範囲の見える化を行う。 ・管理可能な単位での開発規模分割を行う。
	②テスト項目網羅の見直し	・評価漏れを防ぎ、且つ、無駄のない評価を行う為、開発影響範囲と機能を突合せ、効果的、効率的に評価項目を抽出する。
レビュープロセスの問題	③中間レビューの実施	・成果物完成後の作業後戻りを減らすために、工程中間段階で作業内容のチェックを行う。 ・各レビューにおいて、参加すべきレビューアを明確にし、レビュー参加状況をチェックする。
プロセス理解の問題	④オフショア/協力会社への開発プロセスの教育展開	・工程開始前にプロセスおよび作業内容の周知を行う事で、作業の後戻りを防止すると共に、品質の均一化を図る。

施策①：トレーサビリティの強化(1/3)

・トレーサビリティ強化について、拠点間/機能間/工程間の漏れ防止の為、
①機能と開発拠点のトレーサビリティマトリクス
②機能と影響範囲のトレーサビリティマトリクス
③要件トレーサビリティマトリクス
の3種類のマトリクスを作成し、トレーサビリティを確保。
(今回、①②は１つの表にマージして作成)



施策①：トレーサビリティの強化(2/3)

・開発要件と実装機能(モジュール)との関係表を作成。(前頁の①②)
どの拠点／どの部位で開発が発生するかを一覧化した。
→それぞれの要件の責任者と開発責任者を見える化することで、
開発要件および開発の責任者が誰であるかが明確になり、
拠点間の開発内容の意識整合が容易となった。
→開発規模の見える化により、管理可能な規模への分割が容易となった。

※画像一部加工済

機能マトリクス(抜粋)

開発リーダー

要件に対する有識者

機能責任者	サービス分類	サービス名	概要	モジュール責任者(開発チーム/協力会社名/担当者名)	チーム② 会社①	チーム② 会社①	チーム① 会社②	チーム① 会社③	チーム① 会社②	チーム① 会社②	チーム① 会社④	
				モジュール名(管理単位)	A	B	C	D		E		F	
					A-1	B-1	C-1	D-1	D-2	E-1	E-2	F-1	F-2
						2.1				1.5			
						0.9				0.3			
						2.1				1.5	3.5		
								0.3	0.4	0.6	3.5		
						2.1				0.3	3.6		
						1.8	0.6	0.4	0.9		2.1		
						0.6		0.3	0.2	1.3	2.0		
								0.2		0.5	4.2		
						2.1						1.0	
												0.3	
													4.7
											1.0		0.7
													1.0
													2.3
													0.0
									0.2				1.3
													3.3

開発箇所/規模を開発リーダーが提示

施策①：トレーサビリティの強化(3/3)

- 要件トレーサビリティマトリクスの作成をルール化し、各工程完了時に品質担当者によるトレーサビリティチェック結果の確認を徹底。(前頁の③)

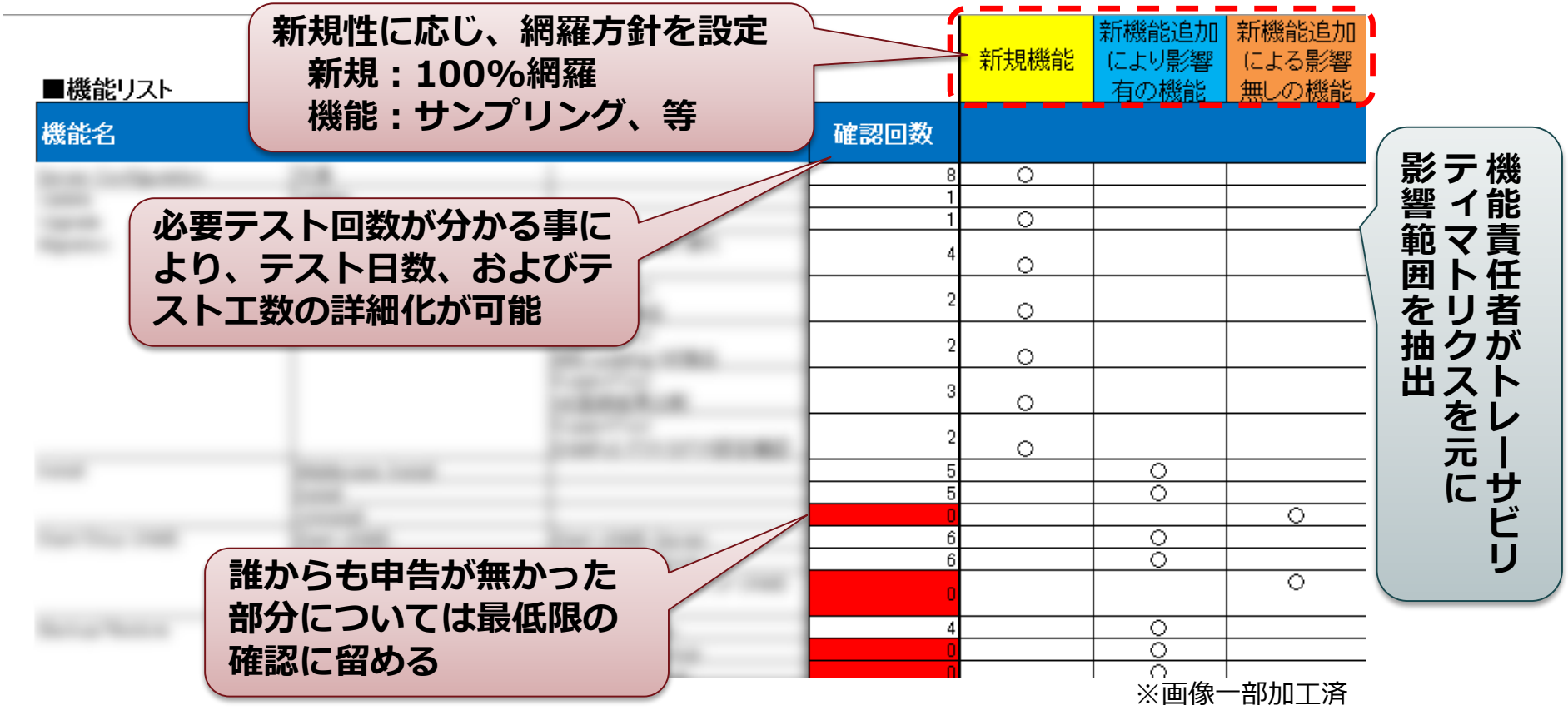
トレーサビリティマトリクス例

※画像一部加工済

要求					BD			FD			DD			CD	
文書	章	項	内容	機能名	文書	章	要求番号/図表番号/内容	文書	章	要求番号/図表番号/内容	文書	章	要求番号/図表番号/内容	アプリケーション名	ソースファイル

施策②テスト項目網羅の見直し

・前述のマトリクスをベースに各機能の責任者が影響範囲を抽出
→各機能毎に必要なテスト箇所が明確になると共に、テスト項目の漏れを防ぎ、かつ効率的に実施するために必要なテストパターン、および必要なテスト回数が明確となった。



※画像一部加工済

施策③レビュー強化(1/2)

・従来、工程完了時にのみフォーマルレビューを実施していたが、工程の前半／中間でのレビューを導入する事で、成果物に対する漏れや誤りの早期発見／早期是正が可能となった。
→バグ流出減による後戻りコスト減／日程遅延減の効果
→総レビュー時間は変わらず（フォーマルレビューでの指摘減の為）
※特にオフショアに対して高い効果が得られた

工程進捗	観点	レビュー観点詳細(例)
10% (目次作成後)	ルール/作法の確認	・参照すべき規定、ルール、テンプレート類の確認 ・インプットとなるドキュメントの整理、確認 ・前工程の担当者による機能説明 ・仕様／設計書の目次構成の整合 ・コーディング作法の確認 ・テスト手法、テスト観点の整理
30% (ドラフト段階)	記述方針/方向性の確認	・仕様設計の記述方針、概要レベルでの妥当性チェック ・不明瞭な個所に対する補足および図表等の追加指示 ・コーディングルールのサンプルチェック ・テスト観点の漏れや誤りのチェック
100% (最終レビュー)	最終チェック	・最終成果物に対する詳細チェック ・トレーサビリティベースでの網羅確認

施策③レビュー強化(2/2)

・レビューを実施するにあたり、機能／ドキュメント単位で、適任者をあらかじめ責任者（キーレビューア）として定義。
（適任者の抽出は前述の機能マトリクスを元に定義）
→レビュー漏れ防止／キーレビューアの意識向上の効果があった。

工程	構成管理/レビューア確認表					
	構成管理表アイテム名	作成分担		レビューア	キーレビューアによる確認日	キーレビューアのレビュー確認結果
		管理責任部門	(所属)作成者	(所属)氏名 (★:キーレビューア)		
FD(機能設計)					11/4	○
FD(機能設計)					11/4	○
FD(機能設計)					11/4	○
FD(機能設計)					11/4	○
FD(機能設計)					11/4	○
FD(機能設計)					11/4	○
FD(機能設計)					11/4	○
FD(機能設計)					11/4	○
FD(機能設計)					11/8	○
FD(機能設計)					11/8	○
FD(機能設計)					11/8	○
FD(機能設計)					11/8	○
FD(機能設計)					11/15	○
FD(機能設計)						○

※画像一部加工済

施策④オフショア/協力会社への開発プロセスの教育展開

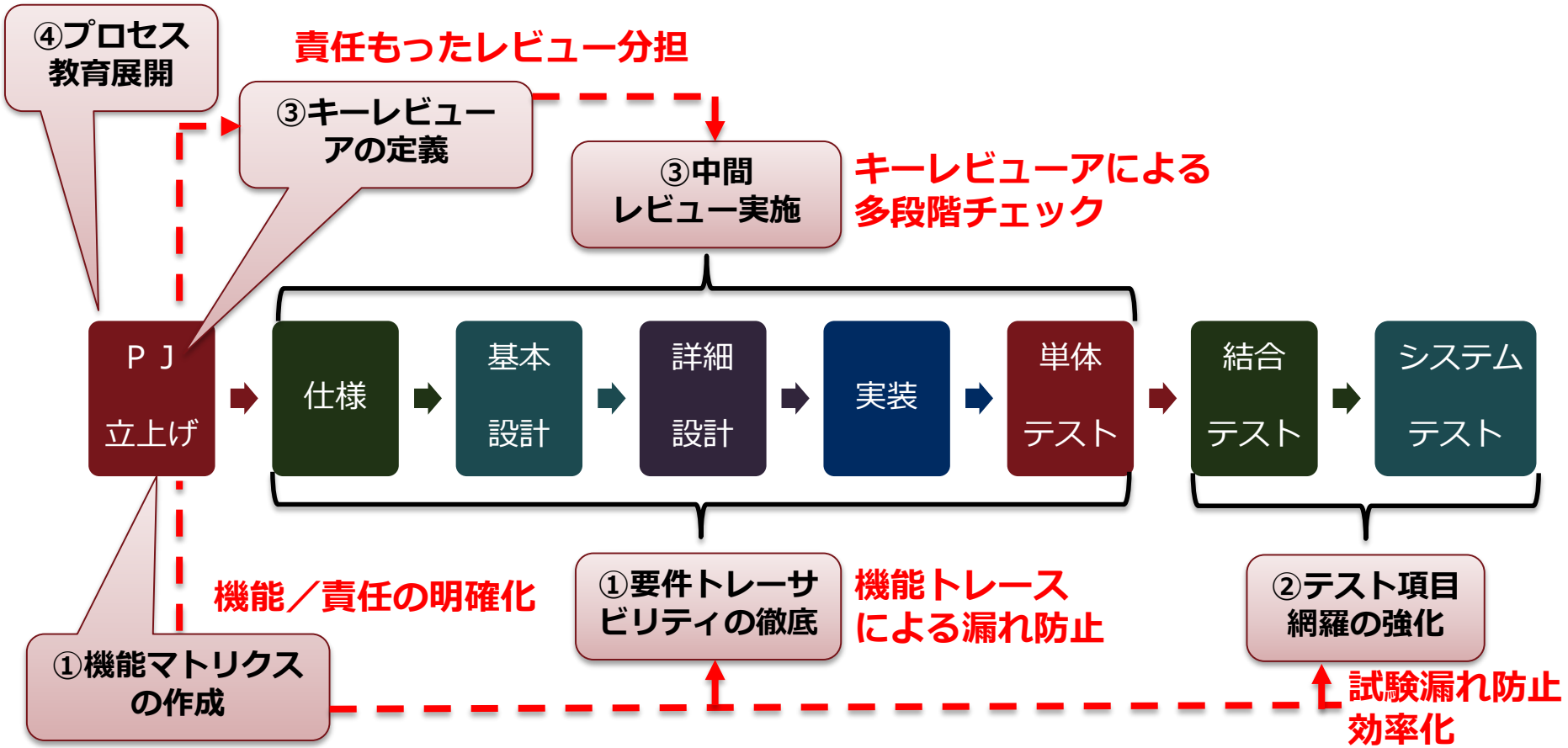
・プロジェクト立上げ段階で全メンバに対してプロセス教育／周知を図る
→各工程でのプロセス逸脱やルール違反が大幅に減少し、成果物の品質均一化に繋がった。そのことによりバグ発見しやすくなった。

教育計画表(抜粋)　※画像一部加工済

作業項目	対象	形式	教材	計画 開始日	計画 終了日	実績 開始日	実績 終了日
■一般開発知識教育							
セキュリティ	全員	講座	社内Web教育素材 資料番号:[1]	2/27	2/27	2/27	2/27
品質教育	全員	講座	社内教育資料 資料番号:[2]-[01]	2/27	2/27	2/27	2/27
開発プロセス	全員	講座	社内教育資料 資料番号:[2]-[02]	2/27	2/27	2/27	2/27
■製品知識教育							
HW装置の紹介	新入メンバー	講座	・・・・・・・・・・.pdf 資料番号:[3]-[01]	2/28	2/28	2/28	2/28
NMSの紹介		講座	・・・・・・・・・・.pdf 資料番号:[3]-[02] 資料番号:[3]-[04]	2/28	2/28	2/28	2/28
開発SWの紹介							
- SWアーキテクチャ紹介		講座	アーキテクチャ仕様書 資料番号:[3]-[05]-[01]	2/28	2/28	2/28	2/28
- 開発環境構築		手順書で環境構築	インストールマニュアル 資料番号:[3]-[05]-[02]	2/28	2/28	2/28	2/28
関連SWの紹介 （Optional）							
- 製品紹介		自習	・・・・・・・・・・・・.pdf 資料番号:[3]-[03]-[01] 資料番号:[3]-[03]-[03]	2/28	2/28	2/28	2/28

品質施策の相互関連性

・今回実施した施策は、それぞれに関係性があり、個々に実施するよりもセットで実施した方が効果的だと分かった。
→特にＰＪ立上げ段階で要件/機能と責任分担を明確にすることが重要。



品質施策の履行確認

・今回実施した施策を確実に履行するためにプロジェクト診断として、各工程でのプロセス履行状況のフォローを実施する仕組みを確立。

プロジェクト診断リスト(抜粋)

番号	チェック項目	PJ計画	仕様化	方式設計	基本設計	詳細設計	診断結
品質データ分析(品質会計)							
5	バグ密度予実(工程毎)	予(9/26) ○(9/26)	予(10/13) ○(10/13)	予(10/31) ○(10/31)	予(11/18) ○(11/11) ○(11/25)	予(12/9) △→○(12/14)	△
6	バグ密度予実(累計)		予(10/13) ○(10/13)	予(10/31) ○(10/31)	予(11/18) ○(11/11) ○(11/25)	予(12/9) △→○(12/14)	予(1/6) △→○(1/17)
7	バグ密度傾向				予(11/18)		
成果物分析(品質)							
17	構成管理						
18	要件トレーサビリティ						
PJ管理状態							
21	プロセス適用状況						

リスクありの場合、ステークホルダーへエスカレーションを実施

No(大)	診断項目	対象工程	頻度(タイミング)	No(中)	チェック項目	No(小)	診断実施単位	チェック方法
18	要件トレーサビリティ	仕様化	工程完了時	18-1	機能要件の要求仕様化	18-1-1	全体	機能要件(RFP)から、実現する要求機能が、トレーサビリティマトリックスを使って洗い出し実施されていることを確認する
			工程完了時	18-2	非機能要件の要求仕様化	18-2-1	全体	非機能要件(RFP)についても、実現する要求機能として、トレーサビリティマトリックスを使って洗い出し実施しされていることを確認する
		方式設計	工程完了時	18-3	順方向トレーサビリティ	18-3-1	全体	HWとSWを開発する場合、 ・要求仕様書の各機能から、SW要素とHW要素への洗い出し作業が、トレーサビリティマトリックスを使って実施されていることを確認する ・すべての上位ドキュメント毎にトレーサビリティマトリックスがあることを確認する
		方式設計	工程完了時			18-3-2	全体	・要求仕様書の各機能から、各FBで実現する機能への洗い出し作業が、トレーサビリティマトリックスを使って実施されていることを確認する ・すべての上位ドキュメント毎にトレーサビリティマトリックスがあることを確認する

※画像一部加工済

施策導入にあたり苦労した点

施策導入にあたり、以下のような課題があったが、都度改善を図り、施策遂行を推進した。

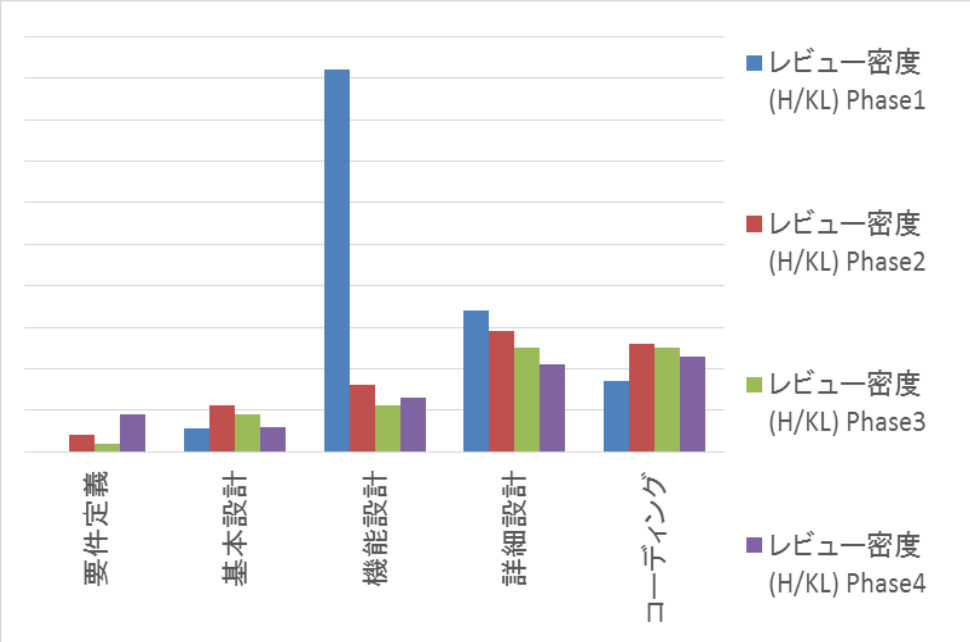
苦労した点／課題	解決策
・機能マトリクスの精度向上	・仕様／設計が曖昧な開発初期段階でマトリクス整備する事は困難であった為、初期段階では影響範囲に対しての○×表のレベルでOKとし、設計詳細化の進度に合わせ、段階的に記述を詳細化するよう推進した。
・オフショアへのプロセス展開の効率化	・オフショア担当工程よりも前の工程の段階で、オフショアリーダーに来日してもらい、トランスファー作業を実施。有識者として育成されたリーダーにて現場に合わせた具体的な教育展開を検討してもらうようにした。
・キーレビューアによる、他チームへのレビュー参加	・各拠点リーダー間による、横通しでの機能整合や、リーダー参加の進捗確認等、リーダーの意思疎通の場をあらかじめ増やしておくことで、コミュニケーションに対する抵抗感を減らした。
・オフショアからの中間レビュー準備状況報告の遅延	・デイリーでオフショアの進捗確認会議(電話会議)を実施し、リアルタイムでの状況把握に努めた。

施策効果

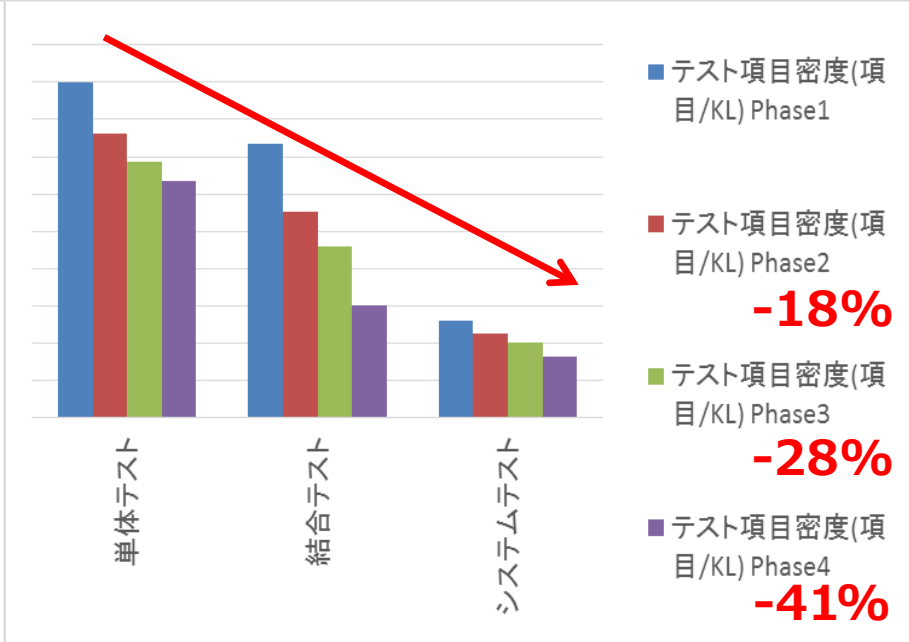
施策効果 (1/3) (定量効果①)

施策効果について、フェーズ１からフェーズ４の開発における
レビュー/試験項目/工程別バグ/総バグ実績をKLOCあたりの密度にて比較

レビューの質の向上による時間増無し
（レビュー効率化）



試験項目の効率化

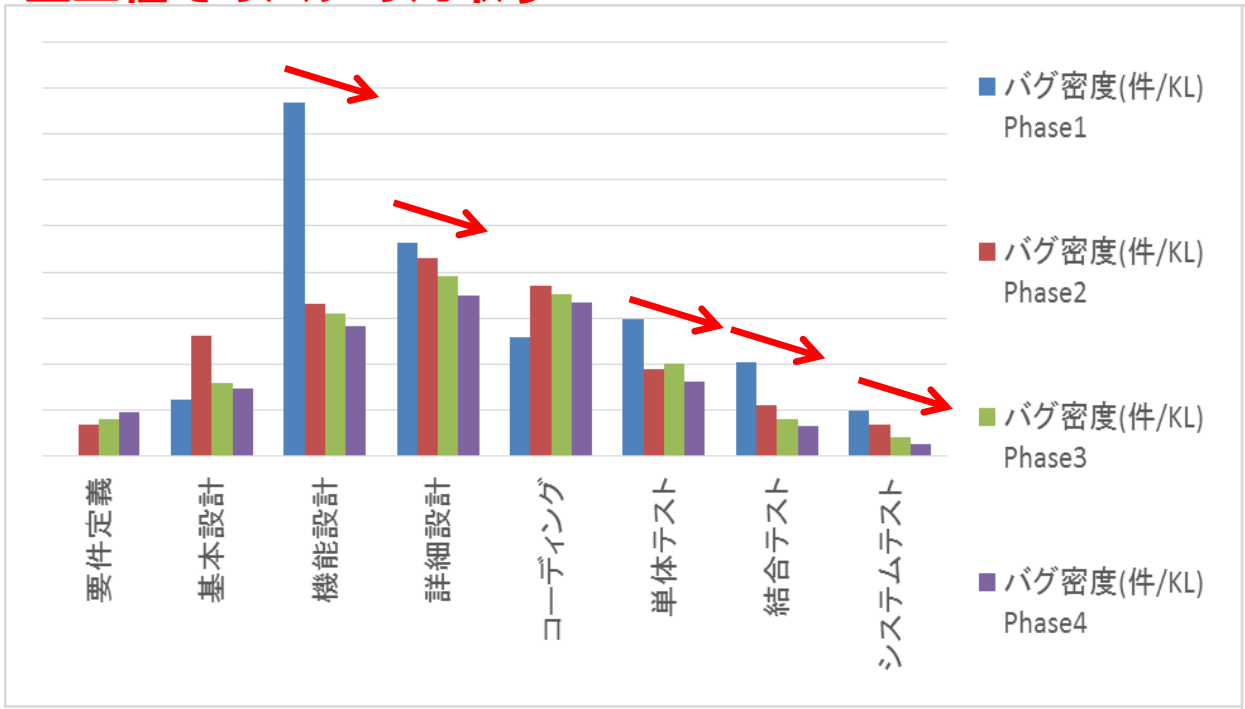


※各数値はフェーズ1に対する改善度合い

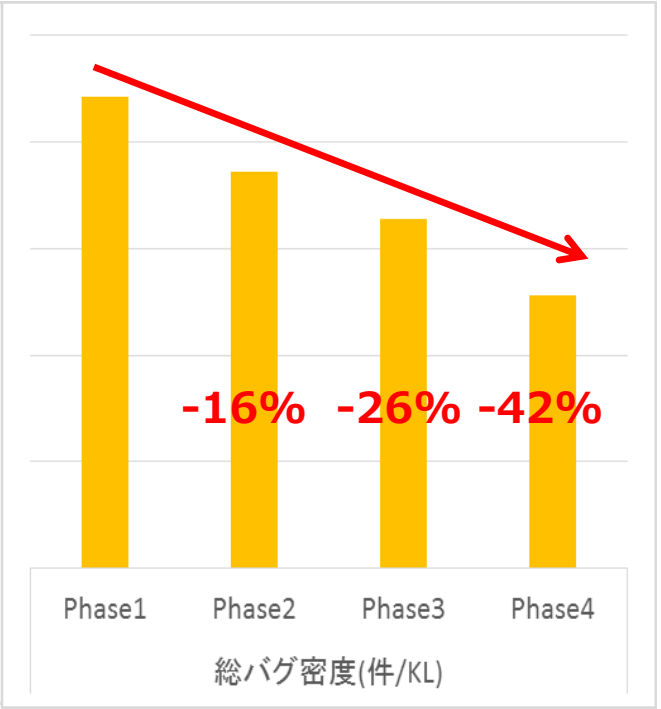
施策効果 (1/3) (定量効果①)

施策効果について、フェーズ１からフェーズ４の開発における
レビュー/試験項目/工程別バグ/総バグ実績をKLOCあたりの密度にて比較

各工程におけるバグの減少
上工程でのバグの刈取り



総バグの改善



※各数値はフェーズ1に対する改善度合い

施策効果 (3/3) (定性効果)

定量的な効果に加え、開発チームにて以下のような効果を実感

施策	施策に対する主なコメント
①トレーサビリティの強化	・責任範囲の明確化により、各機能毎の作業内容の理解が深まる事で、チーム間のコミュニケーションが取りやすくなった。 ・機能分担が明確になる事で、進捗／品質上のボトルネックが明確になり、リスクを回避しやすくなった。
②テスト項目網羅の見直し	・テストシナリオを論理的に立案／検証できるようになり、テスト項目作成日程の短縮になった。 ・テスト項目の組み合わせのバリエーション作成が容易となった。
③中間レビューの実施	・スキルの高いレビューアによるレビューにより、レビュー1回あたりの指摘が増え、総レビュー回数／時間の削減ができた。 ・精度の高いレビューにより、レビュー漏れが減少した。
④オフショア/協力会社への開発プロセス教育展開	・設計書に何を書くべきか／どうコーディングすべきか等のルールの理解が深まることで、記載漏れが減少し、作業の手戻りが減少した。 ・プロセスの全体像を理解する事で、自己分析および品質改善活動のPDCAをオフショアが自主的に行えるようになった。

まとめ

まとめ (1/2)

総括

- 多拠点による新規大規模開発、特にオフショア開発では、開発分担が曖昧となり、抜けや漏れが発生しやすい状況となる。

- 今回取り組んだ施策はいずれも拠点間の認識を統一する効果があるものであり、チーム間での共通認識を持つことで、プロジェクト全体の品質向上に繋がったと考える。

- また、オフショア対しても、体系的な施策を展開する事でオフショア内でのプロセスの成熟の効果があり、自主的な品質改善に取り組む風土の醸成が出来たと考える。

まとめ (2/2)

今後の展望

- 今回の施策は、新規大規模開発において、非常に効果的な施策であると考えるが、開発新規性が低いプロジェクトや中／小規模のプロジェクトにおいては、施策実施による作業が負担(オーバーヘッド)になる可能性がある。

- これまで取り組んだ施策について、効果は維持しつつも、より効率化した方法を模索し、これらプロジェクトの負担にならない形での品質改善を検討していきたいと考える。

御清聴ありがとうございました

Orchestrating a brighter world

NEC