



ほかにはない
アンサーを。

三本の矢による シフトレフトアプローチへの転換

オリックス生命保険株式会社

IT品質保証部

中島 輝

akira.nakajima.dd@ins.orix.jp

本発表における発表者の発言内容は必ずしも
当社の公式発表・見解をあらわすものではありません。

1. 自己紹介
2. 背景
3. 目的
4. 三本の矢の施策の特長
5. 工程完了判定
 - ①概要
 - ②計画品質とITサービス品質の評価
 - ③効率的・効果的なレビュー方法
6. 開発とテストの分離
 - ①概要
 - ②適用プロセス
 - ③適用状況
 - ④適用パターン
 - ⑤テスト委託先に対する必須スキルのチェック
 - ⑥社員に対するテストプロフェッショナルの育成
7. 開発プロセスの改善
 - ①概要
 - ②開発プロセスのテーラリング
 - ③標準化の推進
 - ④テスト計画、テスト分析・設計の前倒し
 - ⑤後続工程のタスクの一部前倒し
8. 実施結果
9. 結論と今後の展開

1. 自己紹介

2017年1月オリックス生命に中途入社
現在IT品質保証部にて、以下を推進中

- ①開発・保守の品質保証全般（品質保証マネジャー）
- ②開発とテストの分離
- ③開発・保守に係る標準化

経歴

1999年東証コンピュータシステムに入社
証券システム部に所属し、証券バックシステム（COSMOシステム）の開発・保守に従事
2007年日本ユニシス（現BIPROGY）に中途入社
金融システム部に所属し、証券系大規模プロジェクトのプロジェクトマネジメントに従事

2. 背景

2019年当時、システム投資案件において、結合テスト、システムテストの段階で多数の欠陥を検出した結果、受け入れテストの延期や本番リリース日の変更といった事態が発生した。

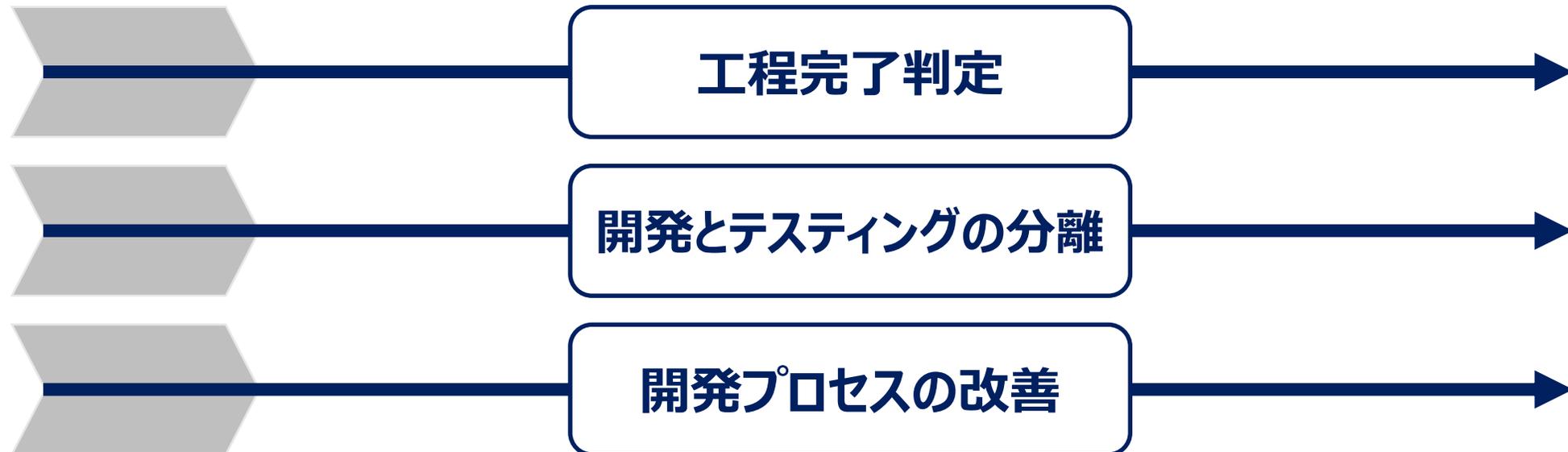
これらの事態の背景に、要件定義工程や外部設計工程で埋め込んだ欠陥をテストの最終段階である結合テストやシステムテストで全て検出できればいいという意識があった。



上流工程からの品質の作りこみができていない

3. 目的

上流工程からの品質の作りこみができていないという課題に対し
三本の矢の施策によりライフサイクルの早い段階での品質の作りこみを強化し、
本番障害、および本番障害における要件定義、外部設計工程の埋め込み比率
を減少させる。



4. 三本の矢の施策の特長

特長①

要件定義工程と外部設計工程の品質を最重要視

特長②

テスト計画、テスト分析・設計の前倒し

特長③

後続工程のタスクの一部前倒し

品質の作りこみ

品質の確認

品質計画

- ✓ 品質に対するユーザの期待値から品質目標を定義
- ✓ ライフサイクル全体に渡って品質を作り込むための戦略を策定
- ✓ テストプロセス全体をドライブする枠組みを定義

特長①

要件定義工程と外部設計工程の品質を最重要視

- ✓ ライフサイクルの早い段階から各種計画書、要件・仕様の欠陥を検出・除去し、手戻りを防止
- ✓ テストベースとしての仕様書の記述十分性を検証

後続工程のタスクの一部前倒し

- ✓ 後続工程を先回りして期日からバックキャストする事によるリスク軽減

特長③

テスト計画

- ✓ 品質目標を達成するために必要な「何を」テストするかをテスト計画を立案

テスト分析・設計

- ✓ テスト計画に従い「何を」テストするかを定義するテスト条件の抽出とテストケース設計

特長②

テスト実装・実行

- ✓ テスト実行スケジュールを作成し、テストを実行
- ✓ テスト進捗と品質のモニタリング

終了基準の評価とレポート

- ✓ 各テストレベル完了時に完了基準を満たしているか、評価

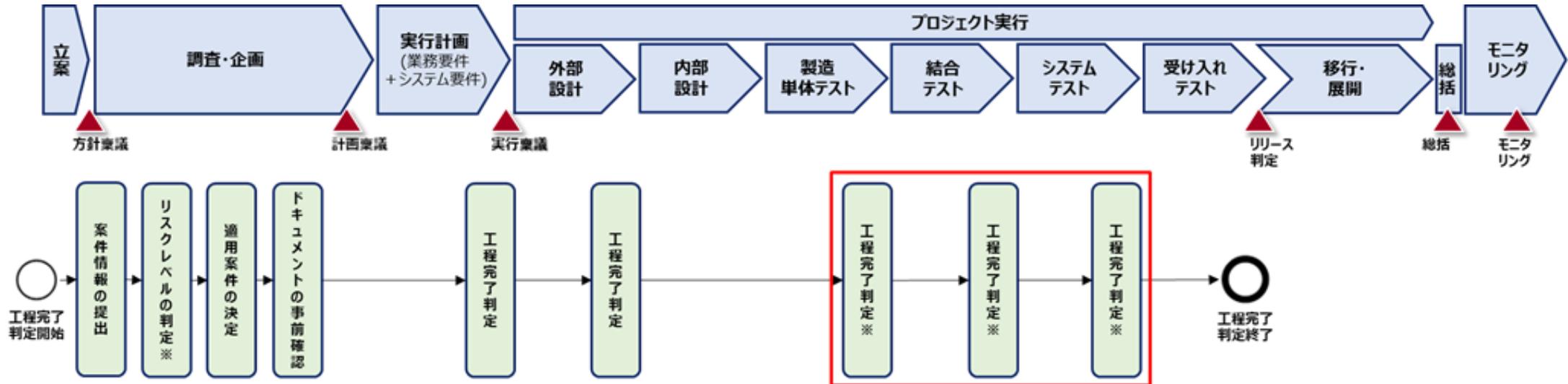
テスト終了作業

- ✓ プロジェクトの振り返りを実行して、プロセスを評価し、改善する領域を発見
- ✓ 横展開可能な作成物のアセット化を実施

5. 工程完了判定

①概要

- あらかじめ定めた工程完了基準に対して、品質保証マネジャーがそれらを満足しているか事実やデータに基づいて評価
- **要件定義と外部設計工程での現物確認（計画品質とITサービス品質の評価）を最重要視**していることが大きな特徴
 計画品質：案件の開発計画の妥当性の品質、ITサービス品質：中間成果物を含めた成果物そのものの品質
 プロセス品質：各作業を決められたプロセスで行っているかという作業品質
- 品質が良いシステムに対しては、テスト工程の工程完了判定を省略することで開発担当者の負荷を軽減



※リスクレベル高：対面で工程完了判定を実施
 リスクレベル中・低：書面で工程完了判定を実施

※投資額が300万以下且つ前年度から今年度にかけて本番障害（投資額が300万以下の案件起因によるもの）を検出していないシステムは対象外

5. 工程完了判定

②計画品質とITサービス品質の評価

■ 計画品質とITサービス品質は以下の観点を重点的に確認

マスターテスト計画書：テスト範囲、前提条件や制約事項、品質リスクと対策、テストレベルとテストタイプ

各レベルテスト計画書：テスト範囲、テストタイプとテスト観点、テストスケジュールと実施内容

移行計画書：移行範囲、移行方針、移行全体スケジュール

ITサービス品質：開発ガイドラインに準拠しているか、ITサービス品質チェックリストの評価基準を満たしているか

	要件定義工程	外部設計工程
計画品質	<ul style="list-style-type: none"> マスターテスト計画書 システムテスト計画書 移行計画書 	<ul style="list-style-type: none"> 外部結合テスト計画書 内部結合テスト計画書
ITサービス品質	<ul style="list-style-type: none"> 要件定義書 システムコンテキスト図 機能要件一覧 非機能要件一覧 トレーサビリティマトリクスなど 	<ul style="list-style-type: none"> 機能一覧 画面遷移図、画面設計書 帳票設計書 外部インターフェース設計書 バッチ設計書 テーブル一覧、テーブル定義書など

5. 工程完了判定

③効率的・効果的なレビュー方法

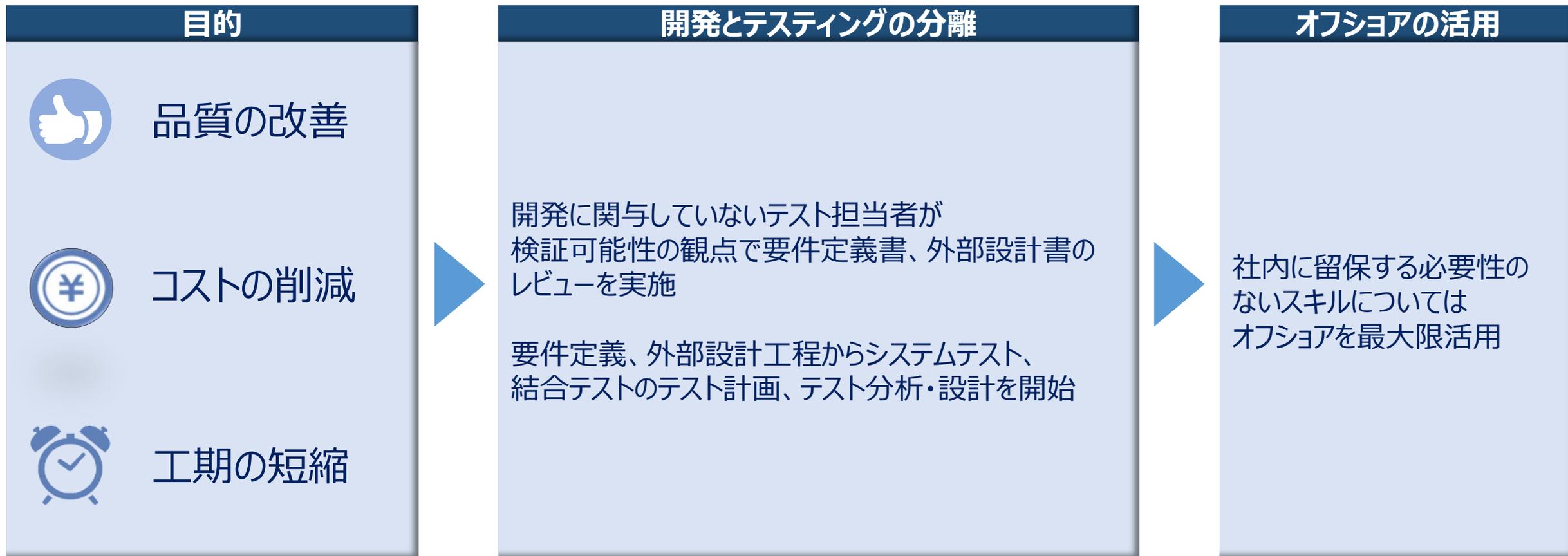
- 品質保証マネジャーが全ての外部設計書の成果物レビューを行うことが理想だが、
全量レビューできない開発規模の場合は、優先順位の高い機能の成果物レビューにリソースを集中させている
- サンプルレビューで検出した勧告事項は、必ず他のドキュメントに対して横展開をすることでレビューの網羅性を担保している

開発規模	レビュー方法の振り分け	レビュー方法
全量レビューできる 開発規模の場合	—	全成果物レビュー
上記以外の場合	以下に該当する機能の場合 <ul style="list-style-type: none"> ■ 外部IFが多く、本番障害時の影響度が大きい機能 ■ 使用頻度の高い共通系の機能 ■ 全設計者の初回成果物 	全成果物レビュー
	上記以外の場合	サンプルレビュー

6. 開発とテストの分離

①概要

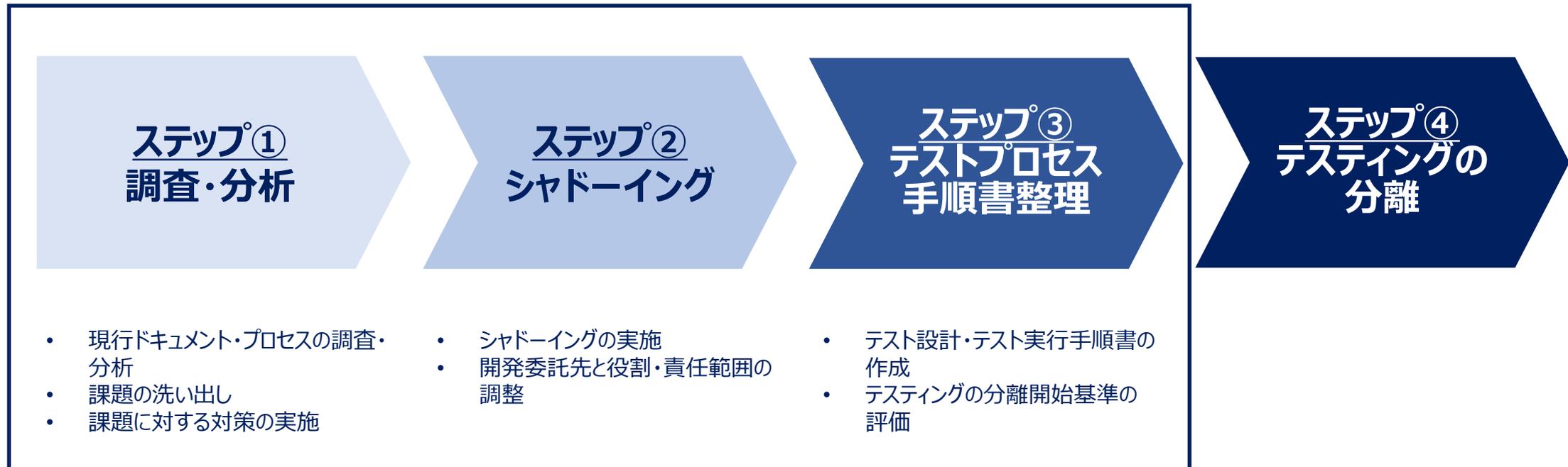
- 開発に関与していないテスト担当者が、ライフサイクルの早い段階から品質の作りこみに関与することで品質の改善、コストの削減、工期の短縮を図る



6. 開発とテストの分離

②適用プロセス

- 現行のドキュメント、テストプロセスを理解しテスト設計・実行手順書を整備し形式知化したうえで、テストの分離を開始
- 発生した問題は別のシステムのテストの分離適用時に必ずフィードバック

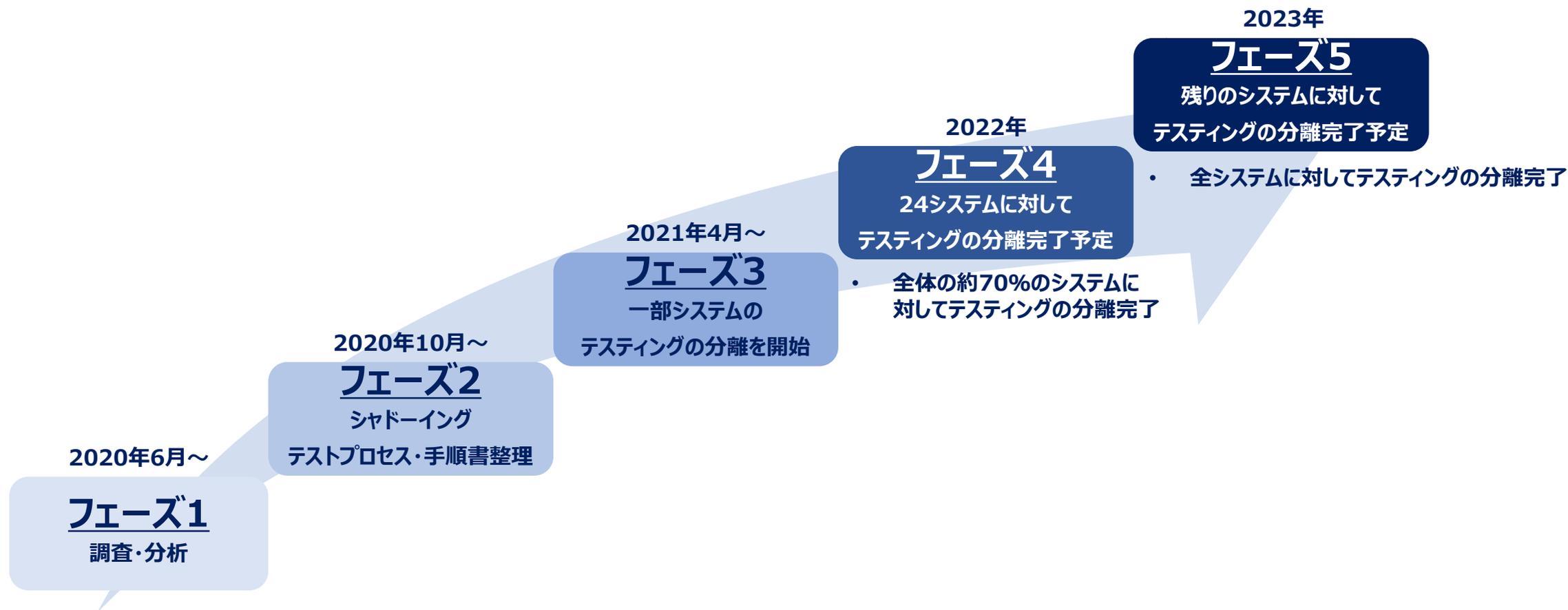


開始直後は半年間の期間を確保、直近は約3か月間の期間を確保

6. 開発とテストの分離

③適用状況

- 今期中に全体の約70%に該当するシステムのテストの分離を完了させ、来期中に全システムの完了を目指す



6. 開発とテストの分離

④適用パターン

- 弊社におけるテストの分離の適用パターンは2通りあり、それぞれのテスト担当者に対して 知識、経験、スキルの観点でJSTQBのテストプロセスを十分に実行可能であることをチェック、支援する仕組みを準備

テスト委託先※による
テストの分離
※現時点で2社
将来的には3社を想定

テスト計画～テスト実行まで
テスト委託先が実施※
原則内部結合テスト～システムテスト
社員は成果物レビューを実施

テスト委託先に対する
必須スキルのチェック
14頁参照

社員による
テストの分離
※内製化できているシステム等

テスト計画～テスト実行まで
社員が実施

社員に対する
テストプロフェッショナルの育成
15頁参照

6. 開発とテストの分離

⑤ テスト委託先に対する必須スキルのチェック

■ テストにおける**ルール毎の必須スキルを明確化**したうえで、テスト委託先のテスト担当者のスキルをチェック

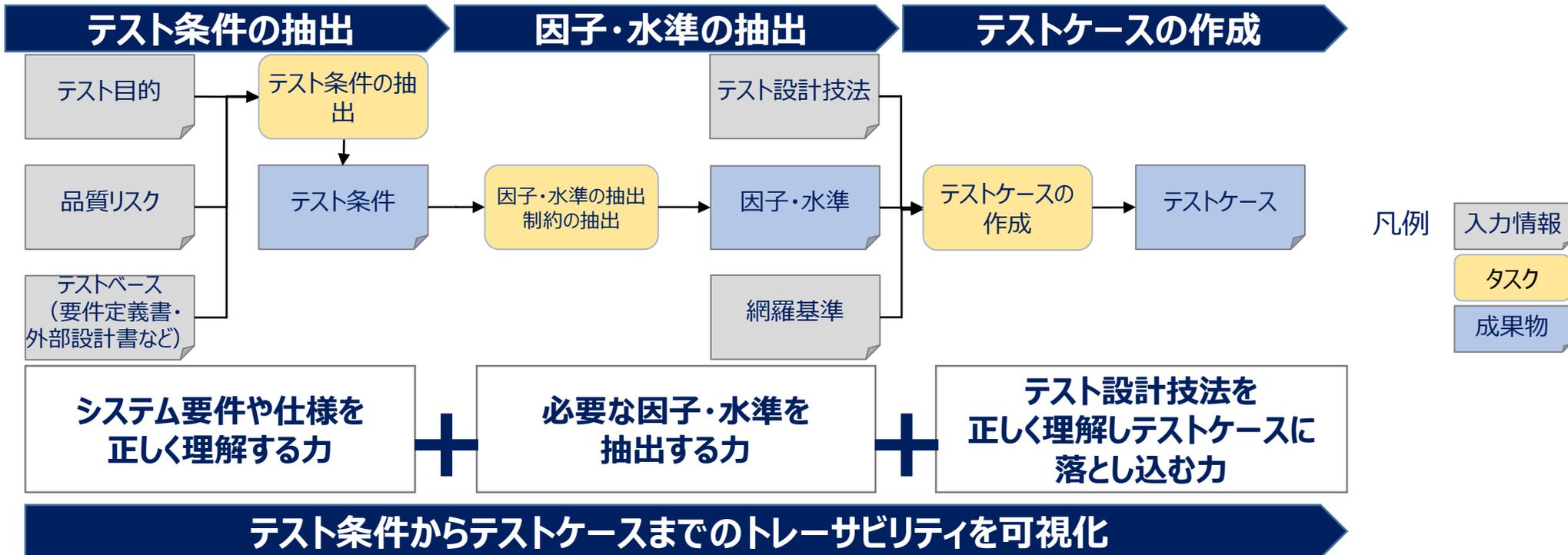
大項目	中項目	小項目	評価基準	ルール		
				テストマネージャ (TM)	テストデザイナー (TD)	テストエグゼキューター (TE)
テストプロセス	テストの計画作業、モニタリング、およびコントロール	スキル	・テスト戦略で特定したテスト目的、テスト範囲、品質リスクに対して必要なテスト活動の特定ができるか？ ・目的の達成の評価に使用する品質メトリクスを収集し追跡する方法を定義できるか？	<input type="checkbox"/> Yes <input type="checkbox"/> No	—	—
		スキル	・テストベースとテスト目的を分析し、テスト条件（テスト観点）を識別（何をテストするのか）することができるか？	<input type="checkbox"/> Yes <input type="checkbox"/> No	<input type="checkbox"/> Yes <input type="checkbox"/> No	—
	テスト設計	知識	・仕様ベースのテスト技法の特徴を理解しているか？ 仕様ベースのテスト技法 ：同値分割法、境界値分析、デジジョンテーブルテスト、状態遷移テスト、組み合わせテスト技法	<input type="checkbox"/> Yes <input type="checkbox"/> No	<input type="checkbox"/> Yes <input type="checkbox"/> No	—
		知識	・経験ベースのテスト技法の特徴を理解しているか？ 経験ベースのテスト技法 ：エラー推測、探索的テスト	<input type="checkbox"/> Yes <input type="checkbox"/> No	<input type="checkbox"/> Yes <input type="checkbox"/> No	—
		知識	・リスクベースドテストのアプローチを理解しているか？ リスクベースドテストのアプローチ ：リスク識別、リスク分析、リスク軽減、テストの優先度付け	<input type="checkbox"/> Yes <input type="checkbox"/> No	—	—
		スキル	・テストベースの分析に基づいて、適切な仕様ベース・経験ベースのテスト技法を選択（どうやってテストするのか）し、最適なテストケースを導き出すことができるか？	<input type="checkbox"/> Yes <input type="checkbox"/> No	<input type="checkbox"/> Yes <input type="checkbox"/> No	—
	テスト実装	経験	・UFT One、Selenium、JMeterなどのテストツールを利用して、テストスクリプトを作成した経験があるか？	<input type="checkbox"/> Yes <input type="checkbox"/> No	<input type="checkbox"/> Yes <input type="checkbox"/> No	<input type="checkbox"/> Yes <input type="checkbox"/> No
		スキル	・テストケースの実行を開始するために必要なテストスクリプトの作成、テスト（手動および自動の両方）の実行順序の編成、テストデータを準備することができるか？	<input type="checkbox"/> Yes <input type="checkbox"/> No	<input type="checkbox"/> Yes <input type="checkbox"/> No	<input type="checkbox"/> Yes <input type="checkbox"/> No
	テスト実行	知識	・インシデント、欠陥、欠陥以外の違いを理解しているか？ ・一般的なインシデント管理プロセス・フローを理解しているか？	<input type="checkbox"/> Yes <input type="checkbox"/> No	<input type="checkbox"/> Yes <input type="checkbox"/> No	<input type="checkbox"/> Yes <input type="checkbox"/> No
		スキル	・実行結果と期待結果を比較し、不一致をインシデントとして登録することができるか？ ・インシデントを入念に精査し、インシデントの解決を支援するデータを収集することができるか？	<input type="checkbox"/> Yes <input type="checkbox"/> No	<input type="checkbox"/> Yes <input type="checkbox"/> No	<input type="checkbox"/> Yes <input type="checkbox"/> No
	終了基準の評価とレポート	スキル	・テストサマリレポートで品質を客観的に評価し、強化テストの実施有無、次工程に進んでよいかの判断ができるか？ 品質評価項目例 ：テストケースの十分性・正確性、テスト実行の結果、品質目標に対する実績、欠陥の分析・評価、欠陥の横展開結果の評価、残課題の状況と次工程への引継ぎ事項	<input type="checkbox"/> Yes <input type="checkbox"/> No	<input type="checkbox"/> Yes <input type="checkbox"/> No	—
	共通	経験	・テスト計画、テスト分析、テスト設計、テスト実装、テスト実行、終了基準の評価とレポートまでの一連のテストプロセスの経験があるか？	<input type="checkbox"/> Yes <input type="checkbox"/> No	<input type="checkbox"/> Yes <input type="checkbox"/> No	—
		経験	・開発にかかわっていない第三者の立場での静的テスト、動的テストの経験があるか？	<input type="checkbox"/> Yes <input type="checkbox"/> No	—	—
	資格	JSTQB認定テスト技術者資格 Advanced Level (テストマネージャ)	知識	・当該資格を有しているか？または同等の知識があるか？	<input type="checkbox"/> Yes <input type="checkbox"/> No	—
JSTQB認定テスト技術者資格 Foundation Level		知識	・当該資格を有しているか？または同等の知識があるか？	<input type="checkbox"/> Yes <input type="checkbox"/> No	<input type="checkbox"/> Yes <input type="checkbox"/> No	—

6. 開発とテストの分離

⑥社員に対するテストプロフェッショナルの育成



社員の必須スキル



7. 開発プロセスの改善

①概要

- 標準的な開発プロセスを全投資案件で適用（2018年から開始）し、開発担当者への啓蒙活動を徹底
- 開発プロセスは問題管理、工程完了判定での勧告事項、投資案件で発生した問題点をインプットにして、継続的に改善



7. 開発プロセスの改善

② 開発プロセスのテーラリング

- 開発担当者は開発規模、品質リスクなどを加味して、当該案件で必要なタスク・成果物を決定
- 品質保証マネジャーが、要件定義工程で必ずタスク・成果物の妥当性を確認する。(2022年にプロセス追加)

開発ガイドラインタスクチェック表 (全工程のタスク・成果物が登録済み)

フェーズ #	フェーズ名	アクティビティ #	アクティビティ名	タスク 推奨/適宜	#	タスク名	成果物名	開発ガイドライン 成果物テンプレートファイル名 E00 E01 E02 E03 E04 E05 E06 E08 E09 E10 E11 E12 E13	リ ン ク	担当			今回 実施 タスク	←テーラリング(参考)列の該当する開発規模が○(推奨)で 今回実施タスクが○となっていない場合に記入 [推奨]タスクを実施対象外とした場合に、その理由を記入する。
										P M	業 務 責 任 者	I T 責 任 者		
3	業務要件 定義	3.1	新業務の業務要求整理	推奨	3.1.01	キックオフ	・業務要件定義フェーズキックオフ資料	E00_002_キックオフ資料	<input checked="" type="checkbox"/>	◎	△	△		
				推奨	3.1.02	前フェーズの詳細共有	-	-	-	◎	△			
				適宜	3.1.03	パッケージソフトの機能説明	-	-	-	△	◎			
		3.2	新業務の定義	適宜	3.2.01	現行業務フロー作成	・現行業務フロー	-	-	◎	-			
				推奨	3.2.02	業務プロセスフロー・業務要件一覧の作成・更新	・業務プロセスフロー(ver.1) ・業務要件一覧(ver.1) ・アクター一覧(ver.1) ・業務要求一覧(ver.9) ・帳票一覧(ver.1) ・バックログ	E03_001_業務プロセスフロー E03_002_業務要件一覧 E03_003_アクター一覧 E01_002_業務要求一覧 E03_004_帳票一覧	<input checked="" type="checkbox"/>	-	◎	△		
				推奨	4.1.03	機能要求の実現可否の検討	・業務要件一覧(ver.2)	E03_002_業務要件一覧	<input checked="" type="checkbox"/>	-	△	◎		
				推奨	3.2.03	非機能要件一覧の作成(業務)	・非機能要件一覧(ver.5)	E02_003_非機能要件一覧	<input checked="" type="checkbox"/>	-	◎	△		
				推奨	4.1.04	システム開発ソリューションの再検討	-	-	-	-	△	◎		
				適宜	3.2.04	画面・帳票イメージの作成	・画面・帳票イメージ(ver.1) ・業務プロセスフロー(ver.2) ・業務要件一覧(ver.3)	E03_005_画面・帳票イメージ E03_001_業務プロセスフロー E03_002_業務要件一覧	<input checked="" type="checkbox"/>	-	◎	-		
				推奨	3.2.05	要件の評価(業務)	・新業務全体像(ver.3) ・業務要件一覧(ver.4) ・業務プロセスフロー(ver.3) ・非機能要件一覧(ver.6)	E02_001_新業務全体像 E03_002_業務要件一覧 E03_001_業務プロセスフロー E02_003_非機能要件一覧	<input checked="" type="checkbox"/>	-	◎	△		
		3.3	全体計画の見直し	適宜	3.3.01	案件企画書の見直し	・案件企画書(ver.4)	E02_004_案件企画書	<input checked="" type="checkbox"/>	◎	△	△		
				推奨	3.3.02	システム要件定義フェーズのスケジュール・費用見直し	・システム要件定義フェーズスケジュール(ver.2) ・システム要件定義フェーズ費用見積(ver.2)	E00_001_タスク一覧・スケジュール	<input checked="" type="checkbox"/>	◎	△	△		
				適宜	3.3.03	システム要件定義フェーズの要員調達	・契約書	-	-	◎	△	△		

7. 開発プロセスの改善

③標準化の推進

2022年にプロセス追加

2020年にプロセス追加

ステップ①

中途入社時

開発ガイドライン
研修

開発プロセス全体の
理解

目次	
1. 研修の目的とゴール	4-4. タスク一覧
2. 背景・目的	4-5. 成果物関連図
3. スコープ	4-6. 成果物テンプレート
4. ドキュメント	5. タスクのテラリングルールと承認
4-1. ドキュメント構成	6. 開発ガイドラインを利用する際のお願い
4-2. プロセス全体像	7. 開発ガイドラインの格納先
4-3. タスクフロー	8. 開発ガイドラインの問い合わせ先

ステップ②

案件開始時

開発ガイドライン
成果物説明

成果物の
位置づけと内容の理解

テストレベル	テストタイプ	テスト目的	テストロードの方針
システム	システムテスト	開発チームが作成したシステムが、開発環境で動作することを確認する。 開発環境での動作確認は、開発環境での動作確認と同等の環境で行う。 開発環境での動作確認は、開発環境での動作確認と同等の環境で行う。 開発環境での動作確認は、開発環境での動作確認と同等の環境で行う。	
システム	統合テスト	開発チームが作成したシステムが、開発環境で動作することを確認する。 開発環境での動作確認は、開発環境での動作確認と同等の環境で行う。 開発環境での動作確認は、開発環境での動作確認と同等の環境で行う。 開発環境での動作確認は、開発環境での動作確認と同等の環境で行う。	
システム	モジュールテスト	開発チームが作成したシステムが、開発環境で動作することを確認する。 開発環境での動作確認は、開発環境での動作確認と同等の環境で行う。 開発環境での動作確認は、開発環境での動作確認と同等の環境で行う。 開発環境での動作確認は、開発環境での動作確認と同等の環境で行う。	
システム	単体テスト	開発チームが作成したシステムが、開発環境で動作することを確認する。 開発環境での動作確認は、開発環境での動作確認と同等の環境で行う。 開発環境での動作確認は、開発環境での動作確認と同等の環境で行う。 開発環境での動作確認は、開発環境での動作確認と同等の環境で行う。	
システム	検証テスト	開発チームが作成したシステムが、開発環境で動作することを確認する。 開発環境での動作確認は、開発環境での動作確認と同等の環境で行う。 開発環境での動作確認は、開発環境での動作確認と同等の環境で行う。 開発環境での動作確認は、開発環境での動作確認と同等の環境で行う。	

ステップ③

案件実行中

成果物の
第三者チェック※
※工程完了判定

開発ガイドラインに対する
準拠度合のチェック

7. 開発プロセスの改善

④テスト計画、テスト分析・設計の前倒し

- テスト計画、テスト分析・設計に該当するテストアクティビティとその成果物を開発工程と並行して実行・作成
- 開発担当者とは別のテスト担当者が上流工程から要件定義書・外部設計書を検証可能性の観点でレビューしたうえでテスト計画書、テスト仕様書を作成

テストレベル	テストプロセス	改善前 (V字モデル)	改善後 (W字モデル)
全体	マスターテスト計画	要件定義 外部設計	要件定義 外部設計
内部結合テスト	テスト計画	内部設計	外部設計
	テスト分析 テスト設計	内部設計	内部設計 開発
外部結合テスト	テスト計画	単体テスト	外部設計
	テスト分析 テスト設計	単体テスト	内部設計
システムテスト	テスト計画	単体テスト	要件定義 外部設計
	テスト分析 テスト設計	単体テスト	外部設計 内部設計

7. 開発プロセスの改善

⑤ 後続工程のタスクの一部前倒し

移行計画書は外部設計工程から作成着手する開発プロセスとなっている。

確定見積り（規模、生産性、工数、期間、コストなど）を行うには、移行方針の検討が必要であり、

移行計画書の中で、要件定義で決めるべき項目を明確化



改善前の
開発プロセス

移行計画書の作成着手

改善後の
開発プロセス

移行計画書の作成

移行方針の検討 移行計画詳細の検討

移行計画書

#	項目	新規追加	要件定義	外部設計	具体的な検討内容
1	本書の目的	-	○	-	
2	本書の位置づけ	-	○	-	
3	移行方針	-	○	-	前提条件、制約事項の洗い出し 移行タイミング、データ移行方法（一括移行、初期移行+差分移行、段階移行など） 並行稼働実施要否、プロジェクト全体での移行、システム毎の移行 移行ツールの作成有無、移行確認テストの実施有無と実施タイミング 移行リハーサルの実施有無とリハーサルの回数
4	移行対象範囲	-	○	-	
5	移行全体スケジュール	-	○	-	
6	移行リハーサルの方針	○	○	-	
7	成果物	○	○	○	
8	本番移行スケジュール	-	-	○	
9	本番移行日の業務制限/制約事項	○	-	○	
10	移行実施内容	-	-	○	
11	移行実施体制・役割	-	-	○	
12	本番移行連絡フロー	○	-	○	
13	本番移行連絡ルール	-	-	○	

一部タスクを前倒し・現物確認することによる後続工程での手戻りの極小化・リスクの軽減

8. 実施結果

年度	本番障害に対する 要件定義工程での 埋め込み比率 (要件定義工程で埋め込んだ 件数/本番障害件数)	本番障害に対する 外部設計工程での 埋め込み比率 (外部設計工程で埋め込んだ 件数/本番障害件数)	合計
2018年度	30%	31%	61%
2019年度	13%	35%	48%
2020年度	24%	23%	47%
2021年度	17%	15%	32%

**2019年度
三本の矢の施策開始**

上流工程での埋め込み比率
が徐々に減少し約半分に

本番障害検出率（本番障害
件数/移行数）も約半分に

9. 結論と今後の展開

要件定義、外部設計工程での品質の作りこみそのものが品質の改善につながり、**本番障害検出率、および本番障害における要件定義、外部設計工程の埋め込み比率を約半分にまで減少**させることができた。

機能要件に関する品質の作りこみは大きく改善されたが、非機能要件の検討不十分による本番障害が相変わらず発生している。今後は**非機能要件における品質をどのように作りこんでいくか**検討していく必要がある。

ご清聴ありがとうございました



本発表に関するお問い合わせ先

オリックス生命保険株式会社

IT品質保証部

中島 輝

akira.nakajima.dd@ins.orix.jp