

通信機器組込み用ソフトウェアの開発 プロセス改善による品質向上の一実施例

～上流工程はXDDPで、下流工程は総合テスト強化
とテスト自動化でプロセス改善～

◎嘉屋 良洋、森中 秀明、村山 知寛、東山 満
アンリツネットワークス株式会社
開発部

2016年10月13日

Contents

1. 会社紹介

2. 背景 – 課題の明確化

3. 原因分析

4. 改善策

5. 導入にあたっての障害と対応

6. 改善による効果

7. 今後の課題

1. 会社紹介

アンリツネットワークス株式会社

ネットワーク機器、監視応用システムの
設計、製造、販売、構築、保守

つなぐ

Being
Smart & Skillful

製品の特徴

- 社会インフラに組み込まれる機器
- 高信頼性を支えるHW/SW

開発の特徴

- ウォーターフォールモデルのソフト開発
- 派生開発が多い



Contents

1. 会社紹介

2. 背景 – 課題の明確化

3. 原因分析

4. 改善策

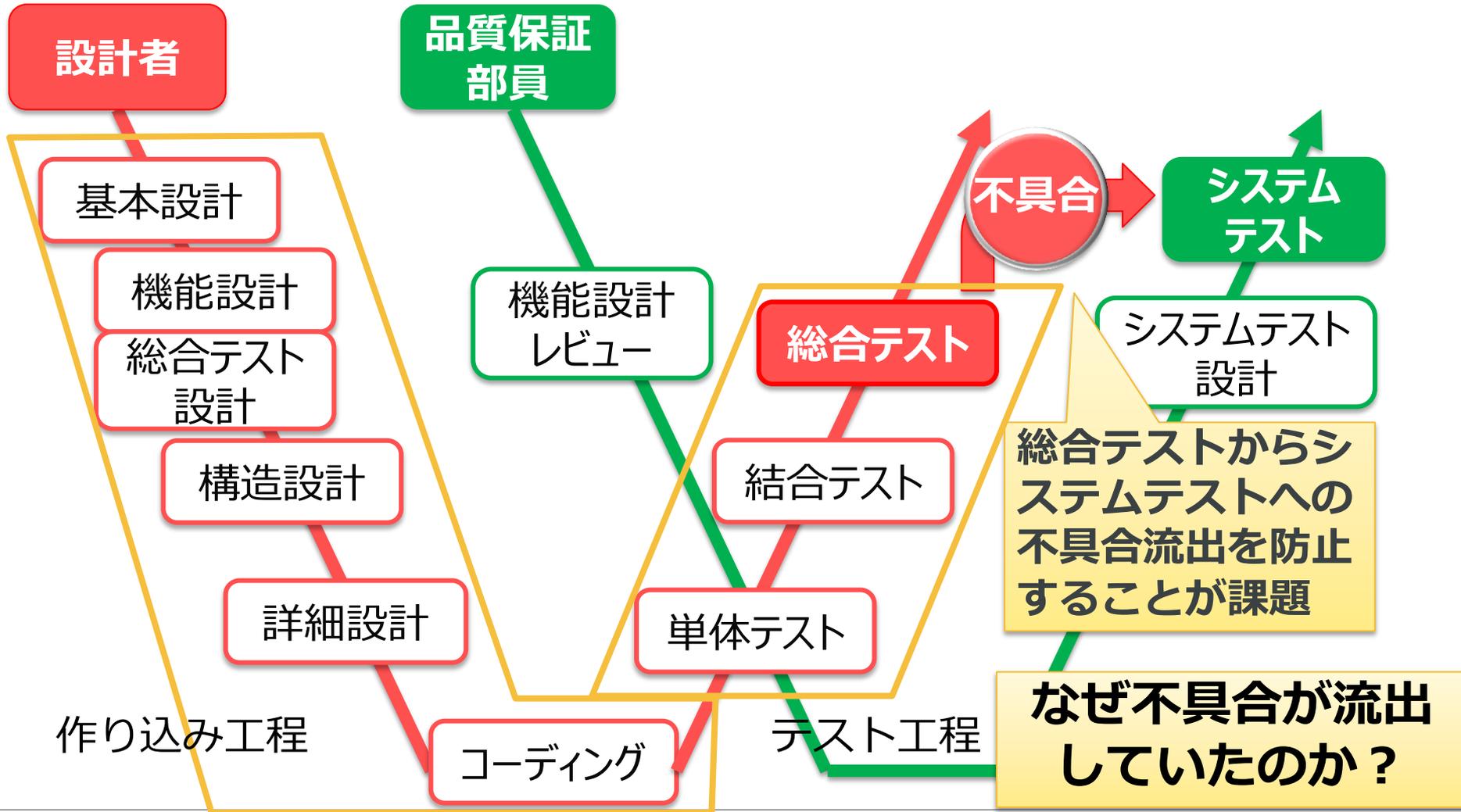
5. 導入にあたっての障害と対応

6. 改善による効果

7. 今後の課題

2. 背景 – 課題の明確化

不具合が設計の最終工程（総合テスト）から後工程（システムテスト）に流出していた。不具合流出を防止することが課題である。



Contents

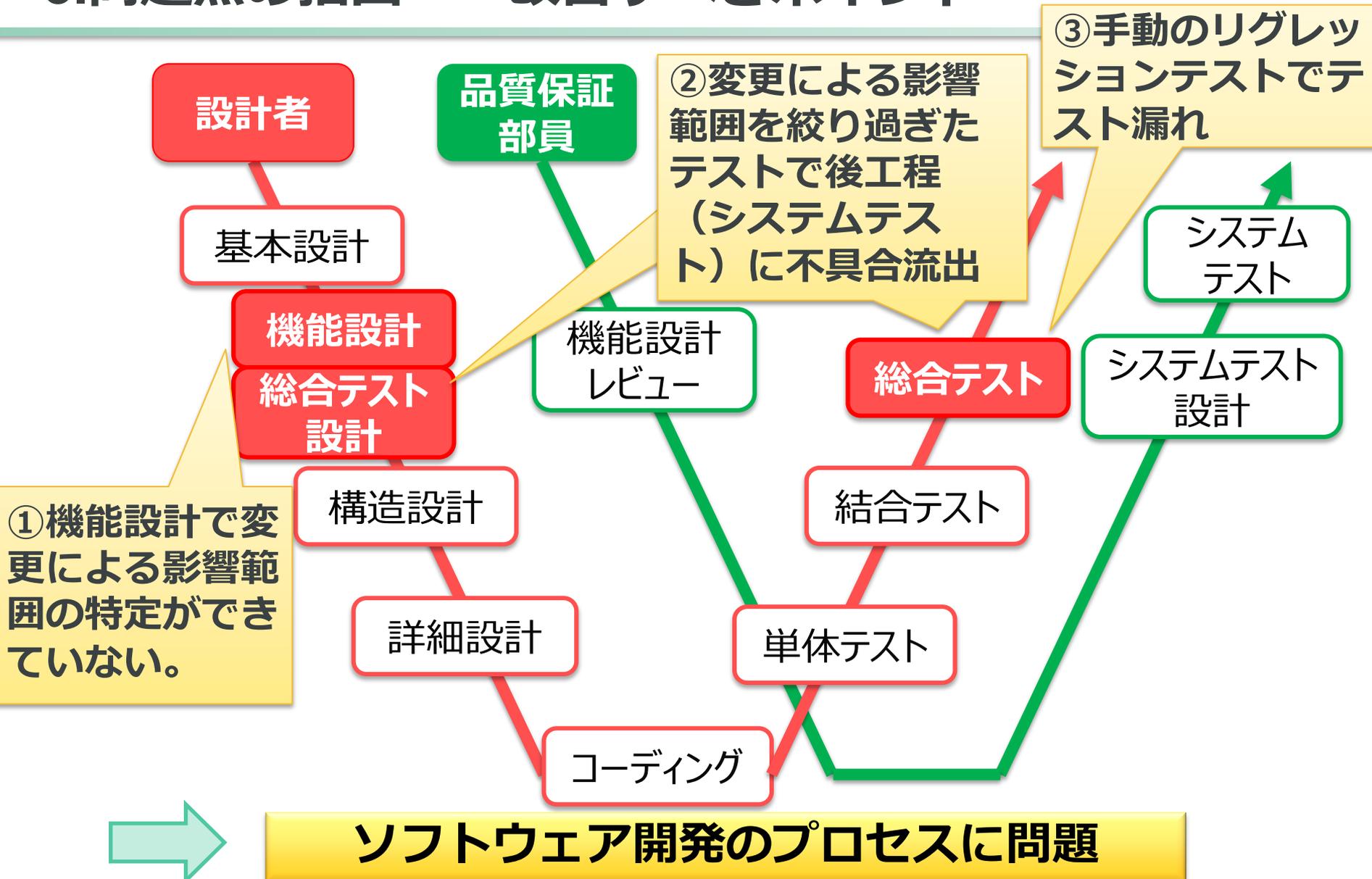
1. 会社紹介
2. 背景 – 課題の明確化
- 3. 原因分析**
4. 改善策
5. 導入にあたっての障害と対応
6. 改善による効果
7. 今後の課題

3. 原因分析

後工程（システムテスト）に流出する原因を分析

No.	原因	問題の工程
1	変更仕様が不明確 なため、変更による影響範囲の考慮漏れ、レビュー不足が発生した。	機能設計
2	設計者自身が「総合テスト設計」と「総合テスト」を担当していると、変更箇所に対する 影響範囲を絞り込み過ぎ ていた。	総合テスト設計 総合テスト
3	手動によるリグレッション・テストでは、費用と時間的制約から テスト範囲が限定 される傾向がありテスト漏れが生じた。	総合テスト

3.問題点の抽出 – 改善すべきポイント



Contents

1. 会社紹介
2. 背景 – 課題の明確化
3. 原因分析
- 4. 改善策**
5. 導入にあたっての障害と対応
6. 改善による効果
7. 今後の課題

4-1. SEPG/TEチームの新設

ソフトウェア開発プロセスに問題があるため改善を実施



新設

SEPG

プロセス改善
(XDDP導入)を
推進

TEチーム

(*1)
総合テスト設計
と総合テストを
担当

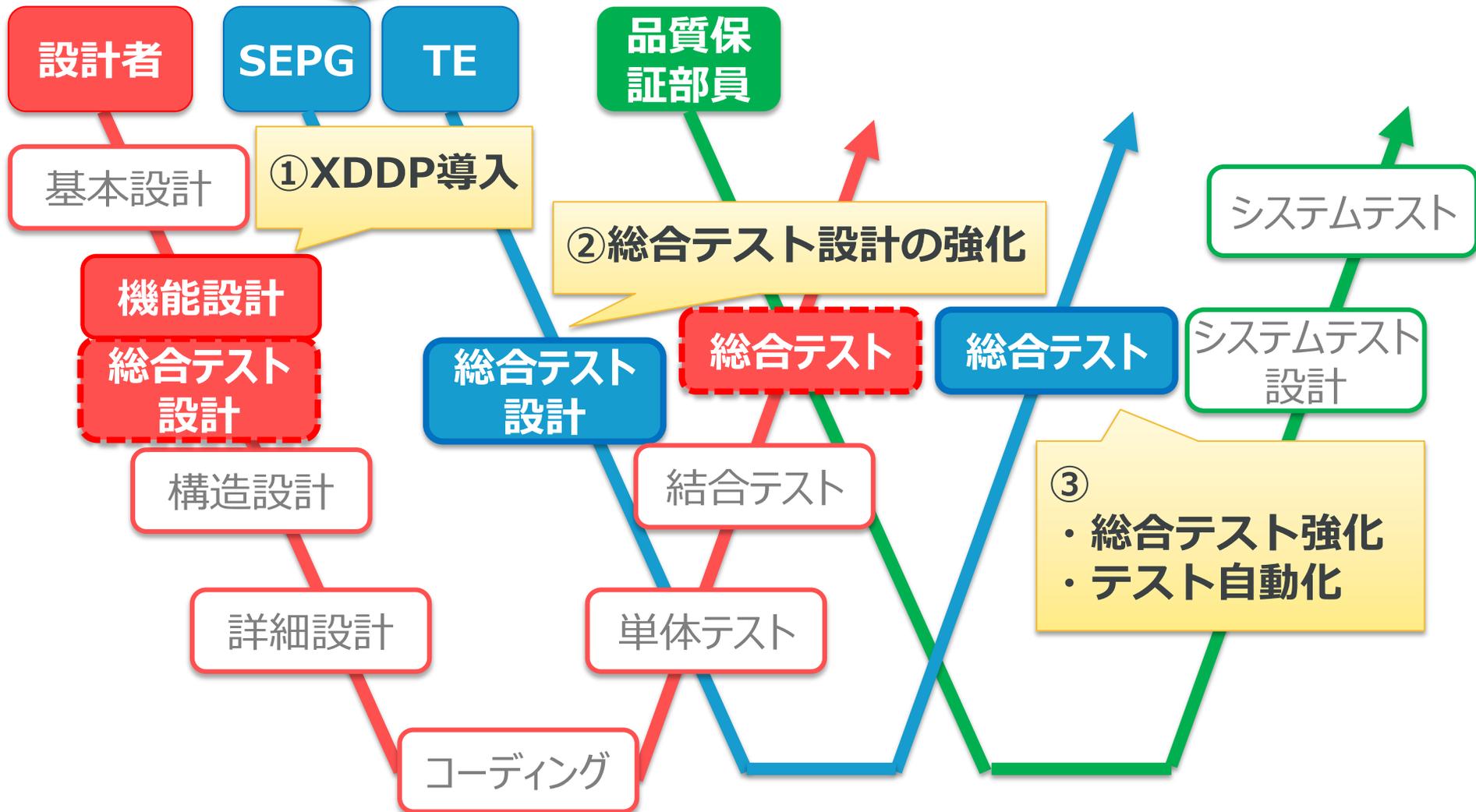
【特徴】

- 1つの組織にSEPGとTEチームの役割を持たせる。
- 設計から分離独立した専任チーム

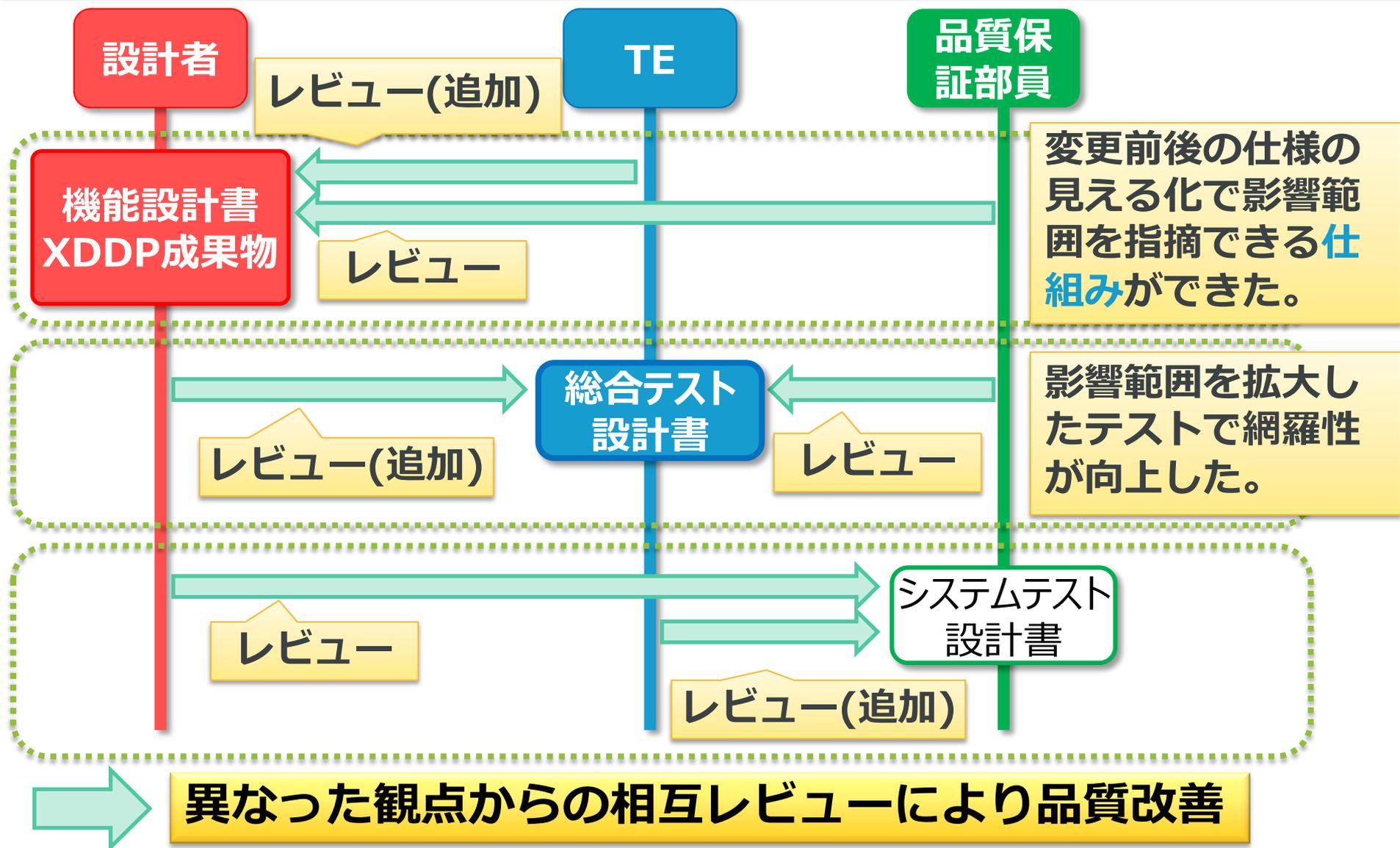
(*1)TEチーム：テストエンジニアチーム
XDDP：eXtreme Derivative Development Process

4-2. 3つの施策

$$(V\&V) + V = V3$$



4-3. XDDP適用とレビューの強化



4-4. テスト自動化 – 環境構築プロセス

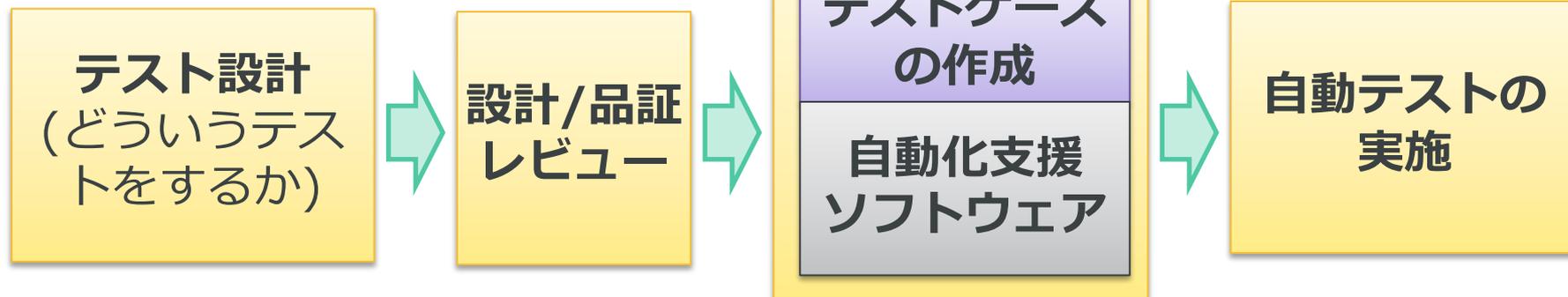
なぜ、今までテスト自動化ができなかったのか？

ネットワーク機器 特有のテストの難しさ
複数の装置の同期を取りながら確認する必要がある。

自動化支援ソフトウェアで解決し、リグレッションテストを自動化



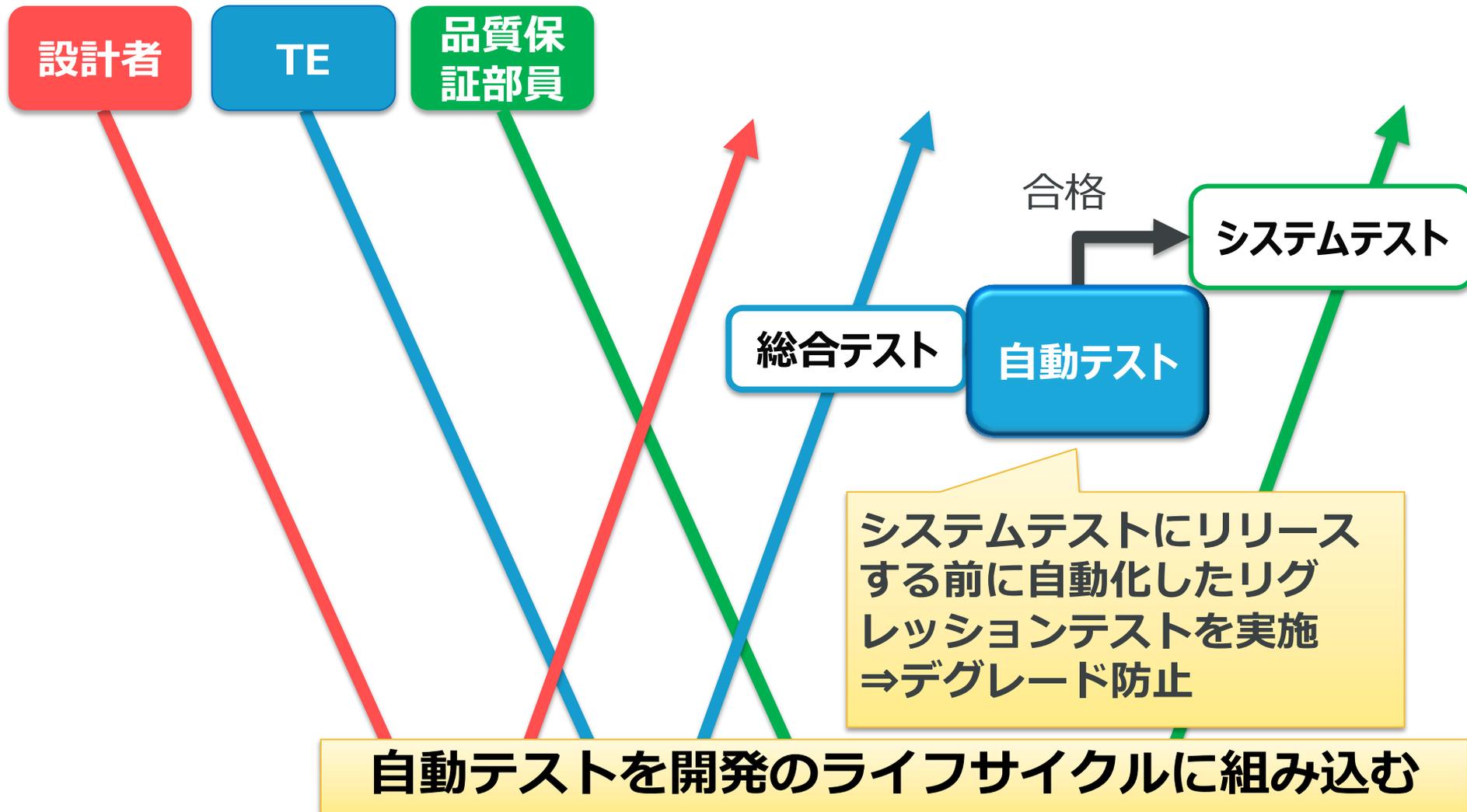
テスト自動化環境の構築プロセス



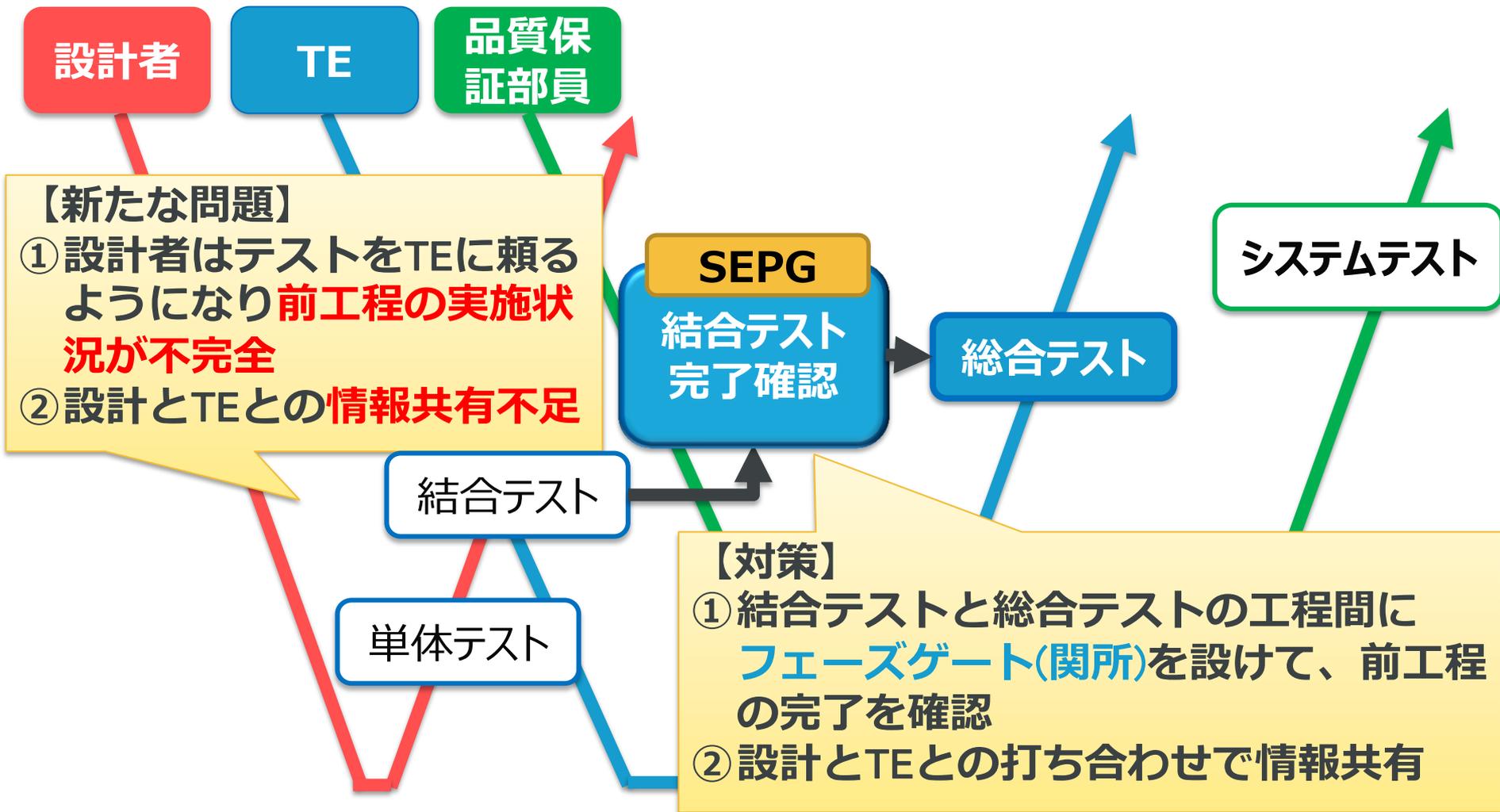
テストケースとは「テストの実行手順」と「結果確認の手順」をまとめたもの

テストケースを「自動化支援ソフト」に組み込む。

4-5. テスト自動化 – 運用プロセス



4-6.新たな課題と対策



Contents

1. 会社紹介
2. 背景 – 課題の明確化
3. 原因分析
4. 改善策
- 5. 導入にあたっての障害と対応**
6. 改善による効果
7. 今後の課題

5. 導入にあたっての障害と対応 – ①XDDPの導入

導入にあたっての障害(XDDPの導入)

導入当初、設計者はやるべき作業が増え、設計工数が不足するため、不満の声があり、導入に抵抗があった。(消極的)



対応

- 開発部の組織を挙げてのトップダウンによる推進体制で、日程計画にXDDPを組み込み、標準プロセスとした。
- 上流工程の作業は増えるが、後工程の工数は減るので、トータルとしては作業量が減ることを設計者に納得してもらった。
- 「振り返り」により具体的な効果を部門全員に示した。

5. 導入にあたっての障害と対応 – ②XDDPの導入

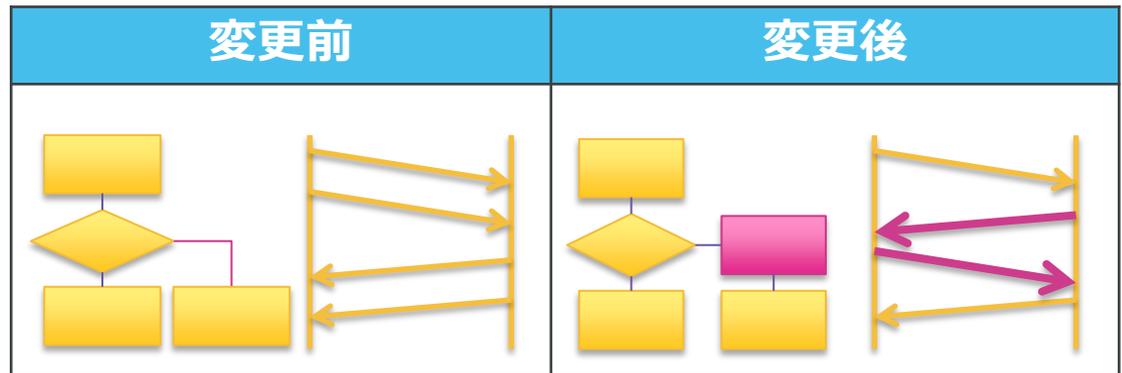
導入にあたっての障害(XDDPの導入)

変更要求仕様書を**文章で表現**することに、こだわり過ぎたため、直感的に理解できず、レビューワーカー間で誤解が発生した。



対応

変更要求仕様書（USDM書式）だけでなく、変更前後の仕様を表す**図表**も組み合わせることで、直感的に理解できるようにした。



5. 導入にあたっての障害と対応 – 総合テスト設計の強化

導入にあたっての障害

新規設立のTEチームは設立当初、**経験とスキル不足**のため、テスト観点の絞り込みが不十分でテスト項目の漏れが発生。



対応

- 経験がありスキルの高い設計者および品質保証部員の**レビューを受ける**ことで、テスト項目漏れを防止した。
- 設計者から技術支援を受けて**スキルアップ**を図った。

5. 導入にあたっての障害と対応 – 総合テストの強化

導入にあたっての障害

TEチームの工数不足

全ての開発プロジェクトでTEチームによる総合テストを実施するようにプロセスを変更したが、TEチームは少人数であったため、工数不足となった。



対応

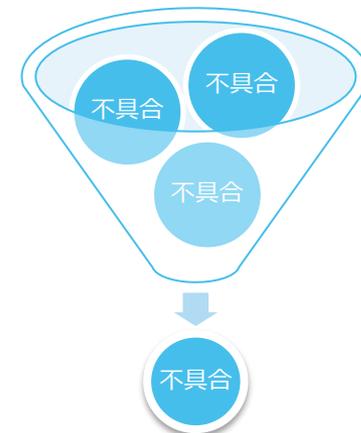
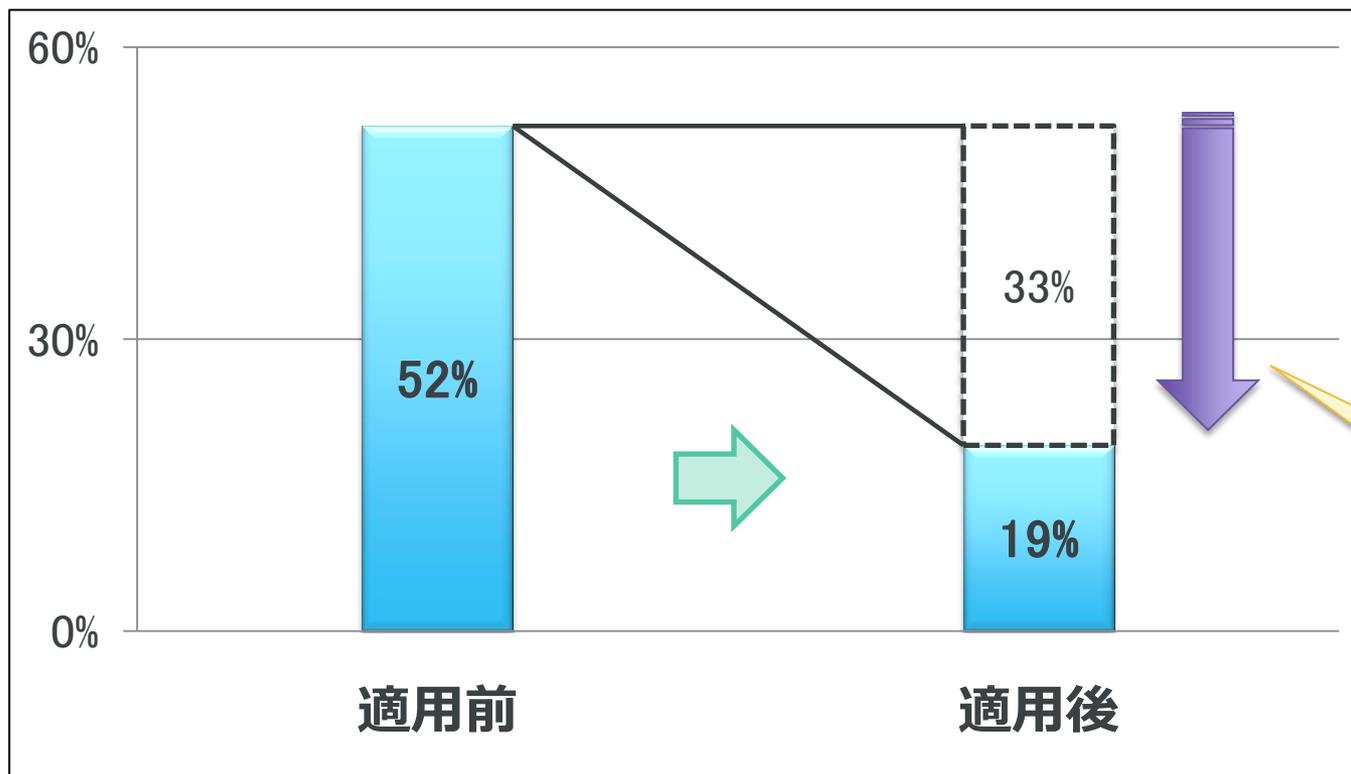
- 成果を示すことで、上位管理者に合意を得てメンバー増員できた。
- 期間を限定して、設計者や品質保証部員の応援や外注により工数を確保した。

Contents

1. 会社紹介
2. 背景 – 課題の明確化
3. 原因分析
4. 改善策
5. 導入にあたっての障害と対応
- 6. 改善による効果**
7. 今後の課題

6. 改善による効果

総合テストから後工程(システムテスト)にどの程度不具合が流出したか（不具合流出率）を今回の施策の適用前と適用後で比較



半分以下
に低減

$$\text{【後工程への不具合流出率(\%)】} = \frac{\text{【システムテスト不具合数】}}{\text{【総合テスト不具合数】} + \text{【システムテスト不具合数】}}$$

Contents

1. 会社紹介
2. 背景 – 課題の明確化
3. 原因分析
4. 改善策
5. 導入にあたっての障害と対応
6. 改善による効果
- 7. 今後の課題**

7. 今後の課題

更に、後工程への不具合流出を防止するには？



- 派生開発だけでなく、**新規開発**の場合も、上流工程の不具合流出防止
- **コーディング工程**の不具合流出防止
(静的解析ツールの適用)
- **単体/結合テスト工程**の不具合流出防止
- **テスト自動化の適用拡大**

ご清聴ありがとうございました。

Anritsu
envision : ensure