THE DOCUMENT COMPANY
FUJI XEROX

第三者テスト部門による
設計品質向上とテスト生産性向上への取り組み

2007年11月1日

富士ゼロックス(株)
品質本部 システム検証部
岡田 俊明　國部 宗紀　杉山 篤志
Toshiaki.okada@fujixerox.co.jp , munenori.kunibe@fujixerox.co.jp , atsushi.sugiyama@fujixerox.cp.jp

開発工程と部門役割概要

開発工程と部門活動時期

主な役割

① 品質目標達成状況の把握と早期警鐘

② システムテストの実施

③ 市場導入可否判断の実施

④ 市場不具合の改善推進

- 開発部門と独立した評価
- ブラックボックステスト
- システム基本仕様と顧客要求への適合検証
- 組込み系SWの場合は、実機システムをテスト

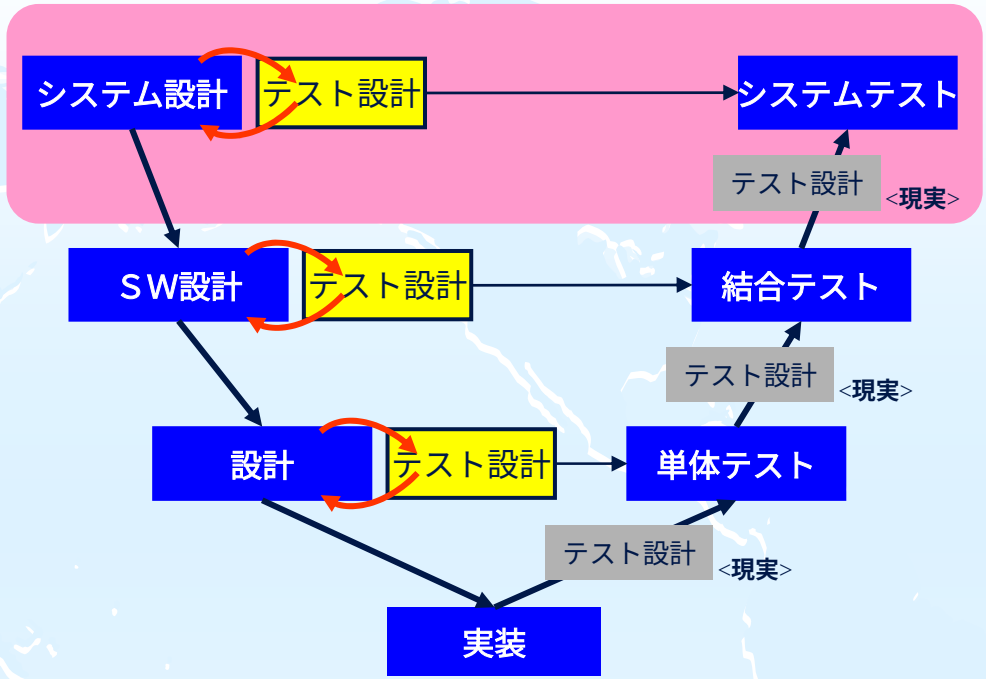
第三者評価部門が陥りやすい状況

	Yes	No
● また設計工程が遅れそうだから、今年も正月/GWは無いな（納期厳しさ：上流＜下流）	□	□
● 仕様はなかなか決まらないし、どうせ仕様変更があるから、テスト直前になってから準備しよう。	□	□
● テストプロセス改善の前に設計工程をどうにかしてよ（自プロセス改善に後ろ向き）	□	□
● 今までのやり方でバグは見つかるからテスト技法は不要。昔やってみたけど現場じゃ使えない。 ・大規模化・複雑化による難解さ（スケールギャップ） ・どこに適用すれば良いかわからない（スキル不足）	□	□
● 市場で不具合が見つかると「テストは何をやっていた！」と言われる。設計に問題があるのに。	□	□
● 本当は開発をやりたかった（若手に多い）	□	□

本日のポイント

◘背景

- ◆設計段階でテストを重視することによって早期に品質を作りこむ重要性/期待の高まり
- ◆現実は、設計品質の悪さや開発遅れによってテスト工程は常に工数逼迫
- ◆経験重視のテスト現場では、技術指向の強いメンバーのモチベーション低下

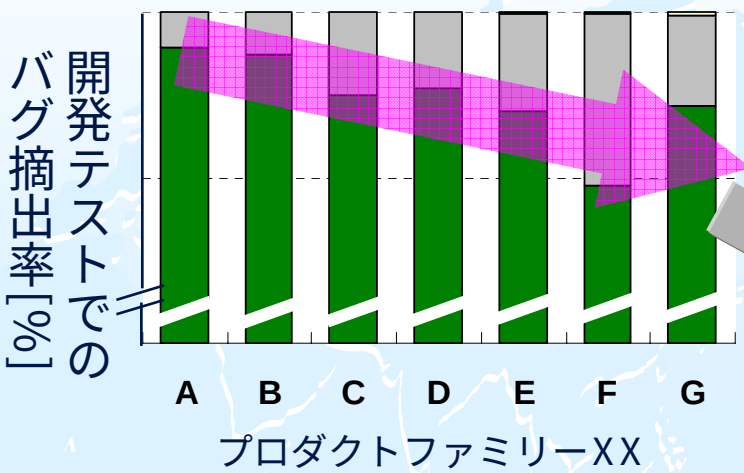


◘ ポイント

本活動に目新しい施策はないが、テスト部門が自部門の技術力を向上させ、その強みを活かして前工程での品質作り込みに取り組む活動が、結果としてテストの生産性向上やテスト技術者のモチベーション向上にも繋がることの実践事例の紹介である。しかし、活動はまだ道半ばである。

問題認識

□ 開発部門の受け渡し品質

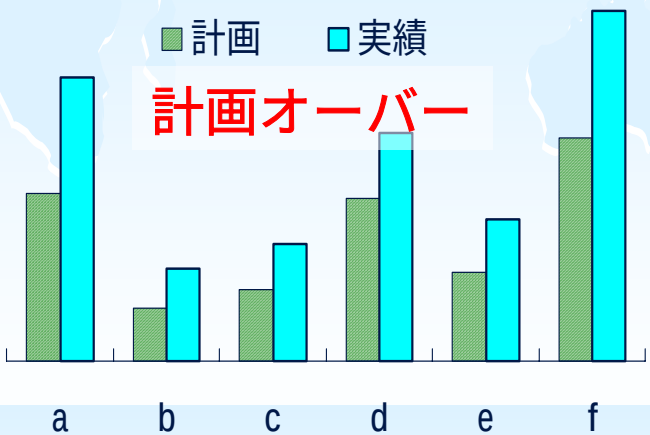


□ 外部与件

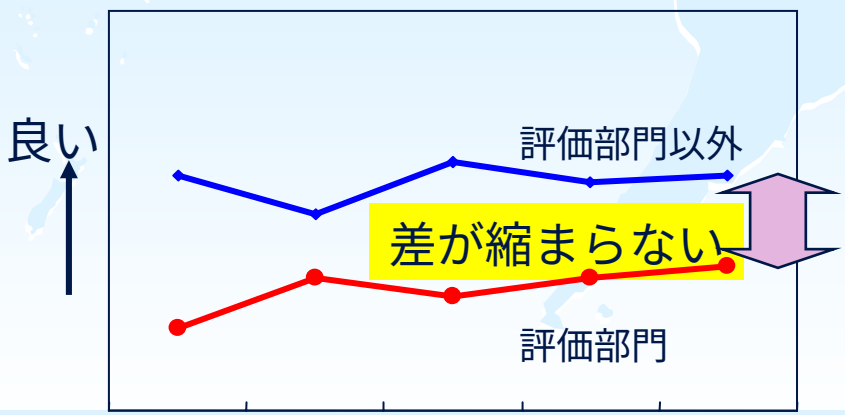
- ◆大規模化
- ◆システム複雑化
- ◆短期開発要求

危機感

□ 第三者テスト工数



□ モチベーション

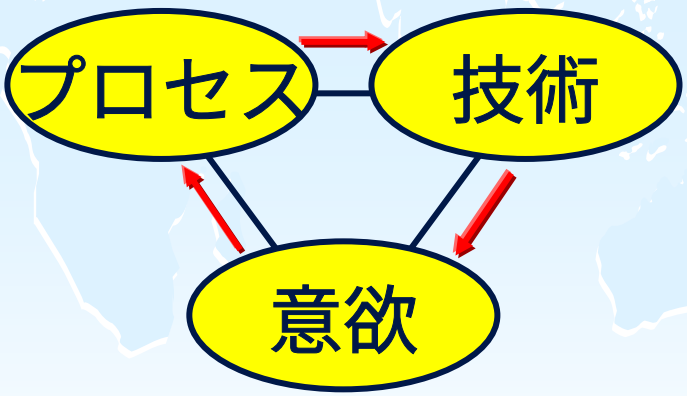


目標達成に向けた３つの視点

目標設定

- 開発からの受け渡し品質向上　バグ摘出率：xx%以上
- 第三者テストの生産性向上　工数/KLOC：xx以下
- 現場モチベーション向上　他部署と同等以上
  （副次効果として期待）

改善を進める上での３つの視点



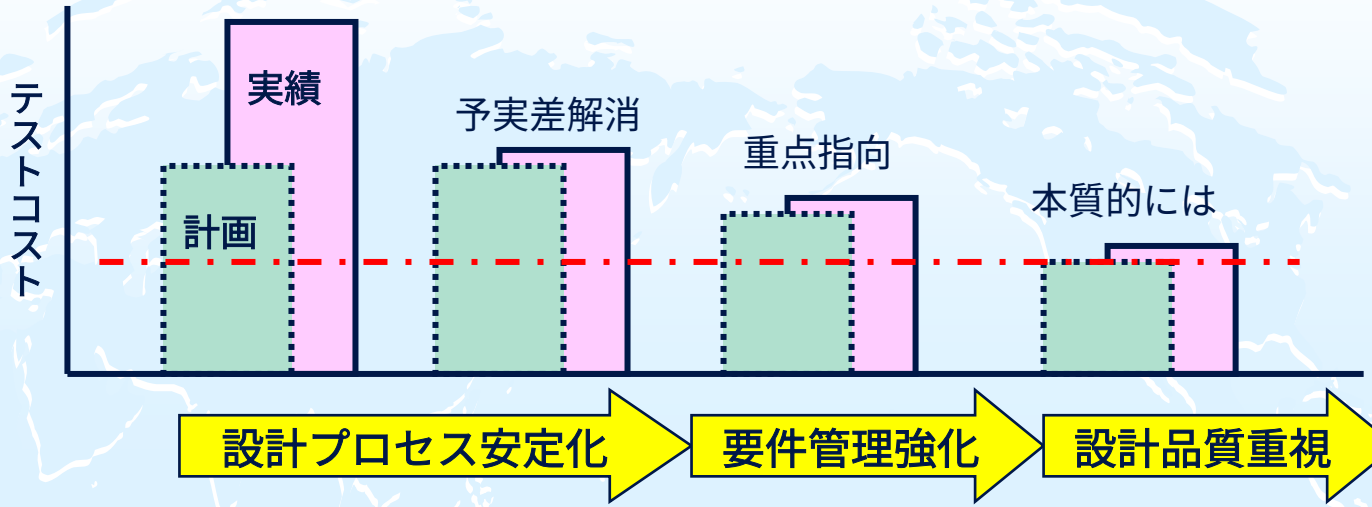
（モチベーション）

- プロセスが安定しないと技術/ツールの導入は進まない。
- 新たなものを作り出す喜びが無ければ技術者としての意欲につながらない。
- 意欲が無ければプロセス改善活動を進めるパワーが出ない。

基本的考え方：３つの視点を同時に改善することが必須

段階を踏んだ改善アプローチ

- テスト工程の改善と上流工程の改善を対にして進める。



設計プロセス安定化 → 要件管理強化 → 設計品質重視

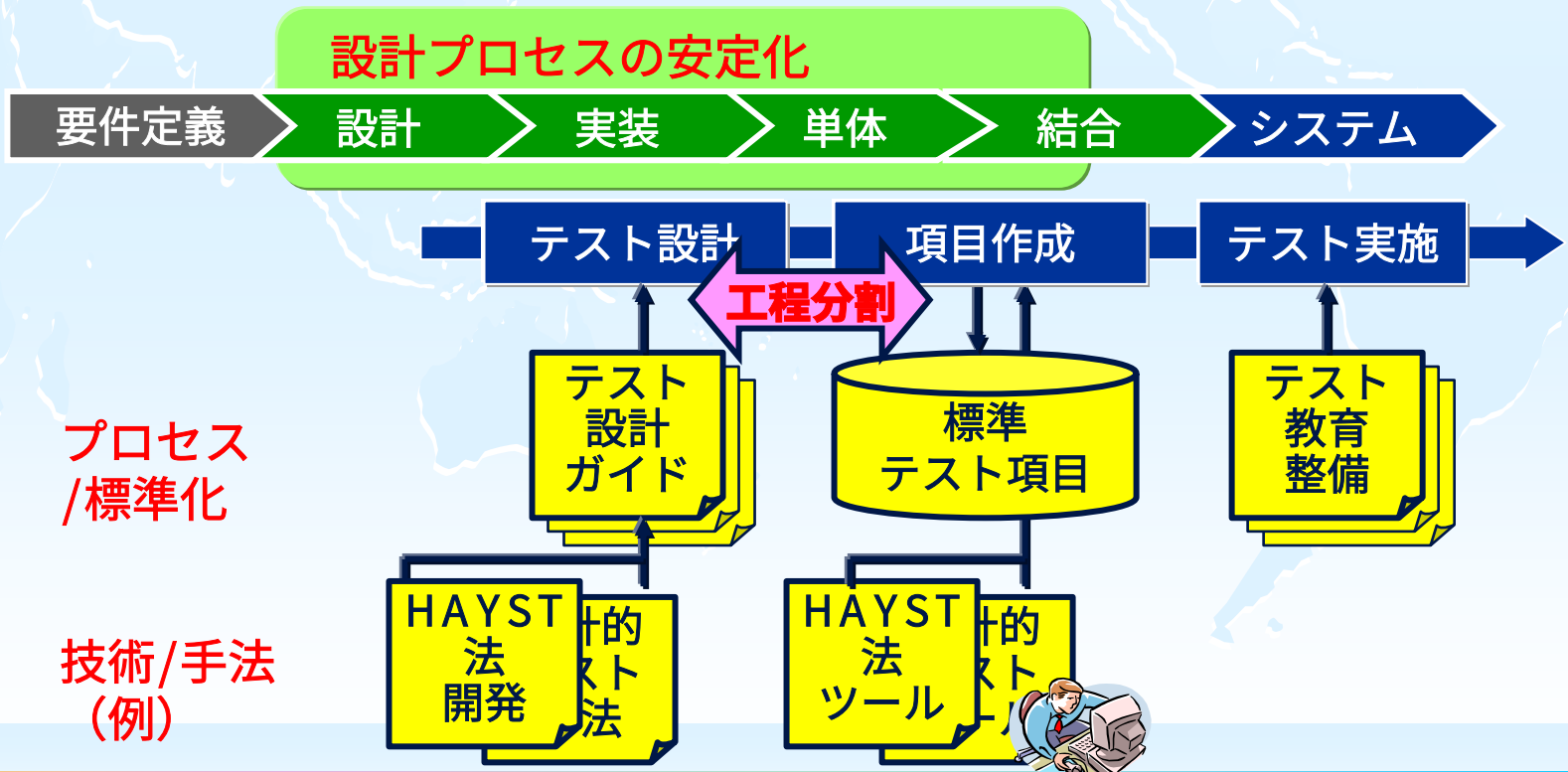
	人海戦術	テスト設計重視	テスト分析重視	上流への取組み
テスト分析	玉石混合となってつき進む	テスト技法導入支援ツール開発	テスト戦略	改善
テスト設計			ツール活用	改善
テスト項目作成		標準化	再利用/改善	再利用/改善
テスト実施		テスト教育	改善	改善

テスト分析/設計の技術を活かす

第１段階：テスト設計重視

- テスト設計と項目作成の工程分離、テスト設計早期着手
- テストカテゴリ化、テスト設計ガイド整備
- ガイドはテスト教育テキストとして活用
- ガイドに基づいたテスト項目を蓄積・流用
- 現場で使えるテスト技法の応用/支援ツール開発

テスト体系化を進める



第１段階：テストの体系化

最低限、以下の条件が必要

テスト設計とテスト項目作成を分離	テスト品質はテスト設計に依存する（個人/経験の依存度を下げたい）
テストの種類分け/カテゴリ分け	システムの各機能をどのテストカテゴリでテストするか対応付けたい
テスト設計ガイド/テスト仕様の作成	効果的にバグを見つけるための設計方法/ノウハウを資産として残したい。

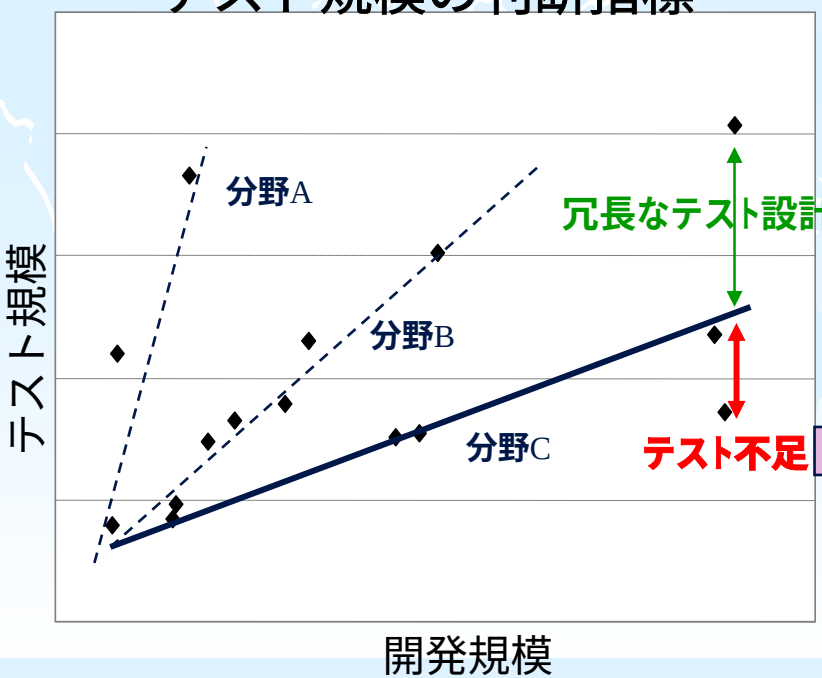
結果として以下のことが可能に

- プロダクトファミリー内ではテスト規模が平準化
  見積もりの精度向上、乖離プロダクトの結果分析が可能
- テストカテゴリに適したテスト技法の導入
  現場で使える技法へ改良/応用、支援ツール開発への投資理解
- ただし、限られたリソースが、各テストカテゴリに適切に配置できているかと言う点に課題は残る。

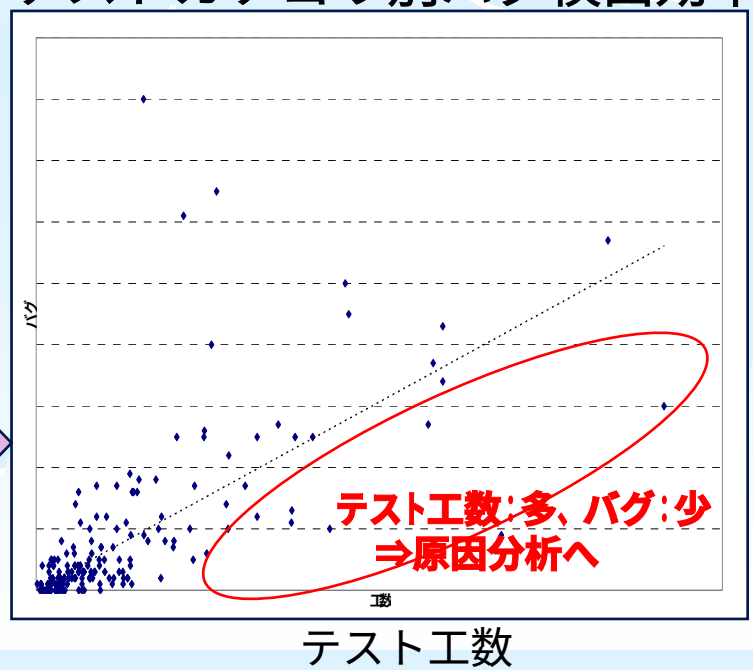
第１段階：結果分析例

- テスト規模のバラツキ縮小→テスト規模の適正さを判断
- 乖離プロダクトの分析
  - テストカテゴリ別のバグ検出効率
  - テストカテゴリ別のテスト消化効率（スピード）
  - テストチームメンバーのスキル/経験度　等々

テスト規模の判断指標



テストカテゴリ別バグ検出効率



事例：機能組合せテスト設計手法の開発

□ 機能組合せテスト設計の課題

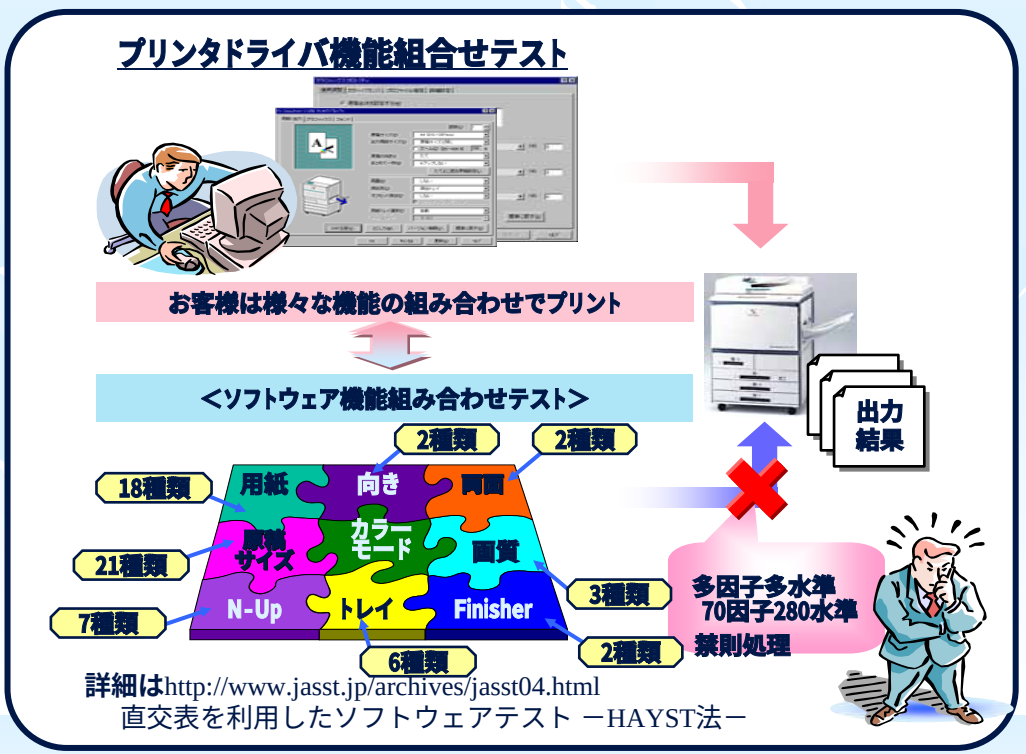
- ◆ 経験則でランダムに組み合わせると網羅率が低下する。
- ◆ 機能組合せを網羅的に行うとテスト項目数が爆発する。

} 相反する課題

組合せ網羅率
経験で作りこみ　約30%

ＨＡＹＳＴ法の開発による網羅率向上とテスト項目数低減の両立

組合せ網羅率
HAYST法　＞80%
（テスト品質指標）



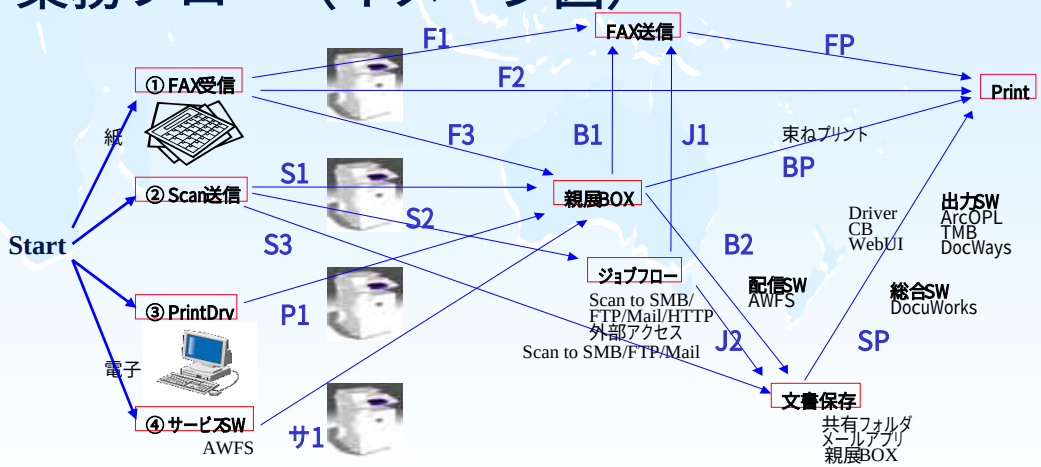
事例：運用テストへの統計的テストの応用

□ 運用テスト設計の課題

- ◆ 顧客業務フローを網羅的にテストしたい一方、良く使うフローを重点的にテストしたい。
- ◆ 運用テストは基本的にランダムテストであり、一般的なテスト技法を適用しにくい面がある。

⇒　統計的テスト手法を運用テストに応用
(パス網羅性の確保＋パス出現率を指定)

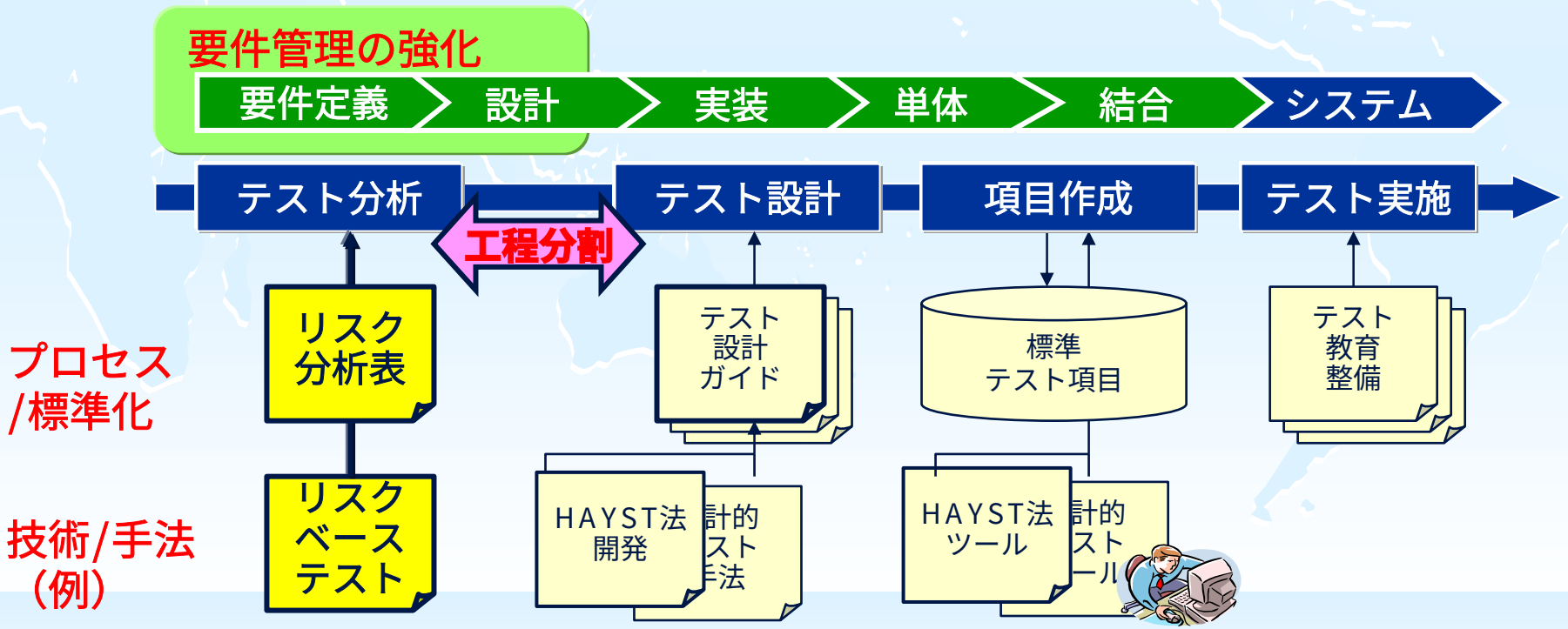
業務フロー（イメージ図）



テスト項目自動生成
- テスト項目数は任意
- パス網羅
- 指定されたパス出現確率

第２段階：テスト分析重視

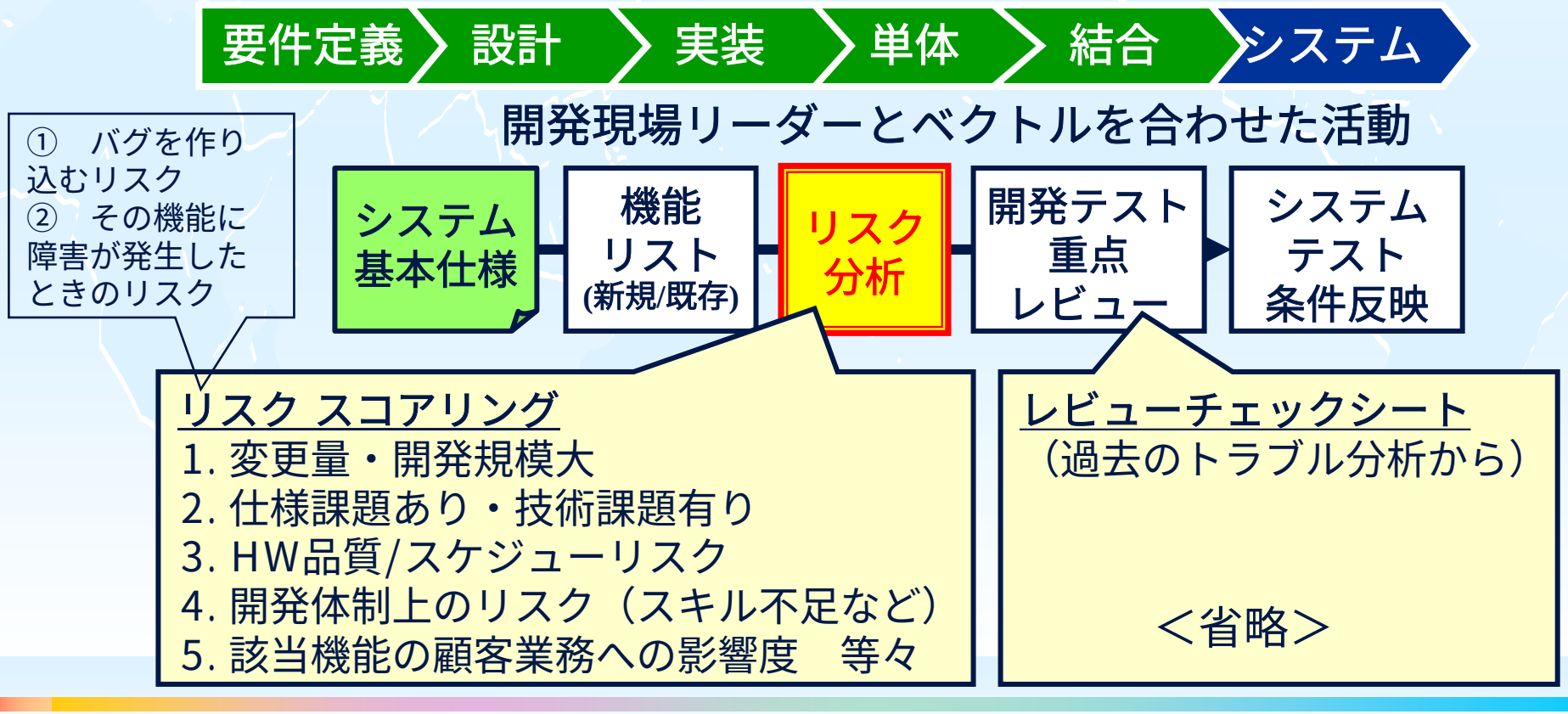
- 限られたリソースをテストカテゴリに適性配置するために、テスト分析とテスト設計を分離する。
  - ・テスト分析：何を重点にテストするか（　戦略）
  - ・テスト設計：どのようにテストするか（　戦術）
- リスクの高い機能を重点的にテストし、低い機能はテストを間引く（リスクベーステストの考え方を導入）



第２段階：リスクベーステストの導入

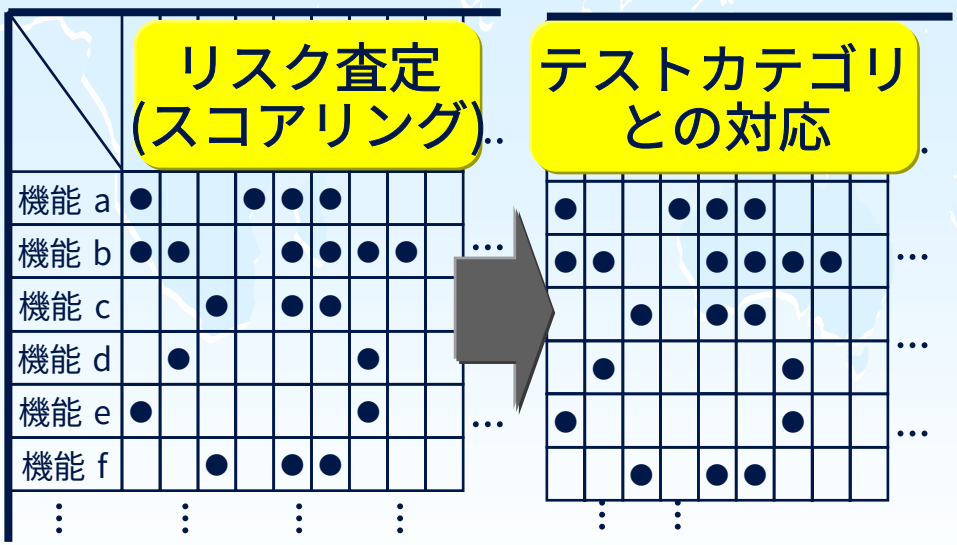
- 開発部門とのリスク機能の共通認識
  - ⇒　開発部門の重点管理対象としてモニタリング
  - ⇒　第三者評価部門による開発テスト計画の重点レビュー
  - ⇒　システムテスト条件への反映（“手厚く”と“間引き”）

- 限られたテスト工数の適正配置

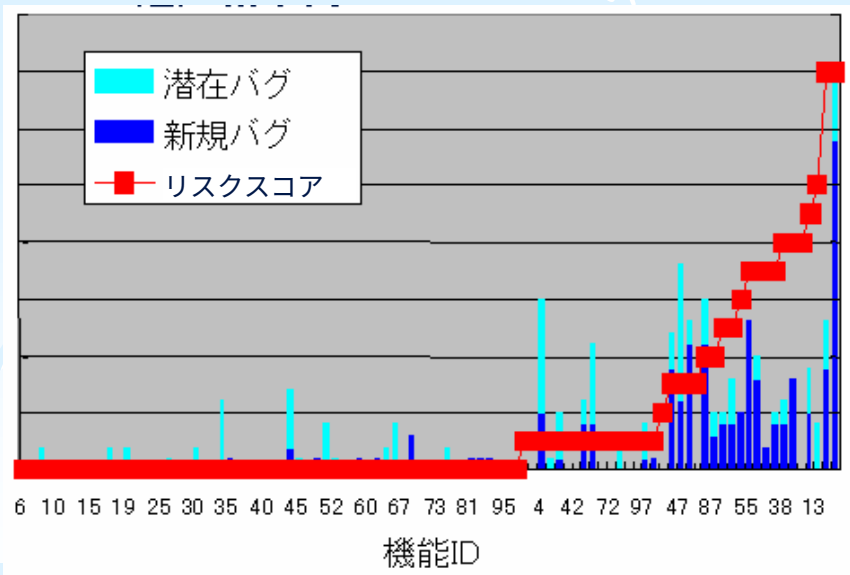


事例：リスクベーステスト適用結果

- リスクスコアに基づいたテストカテゴリへの工数適正配置
  - 機能とテストとの対応明確化
  - 手を抜く機能の明確化（→継続モニタリング）
- リスクスコアとバグ数は相関が見られる。
  しかし、リスク低機能にもバグは発生している。
  → 網羅的なテストを省略できないことが課題。
  機能網羅テストを早めに実施し、その結果からテスト計画を修正

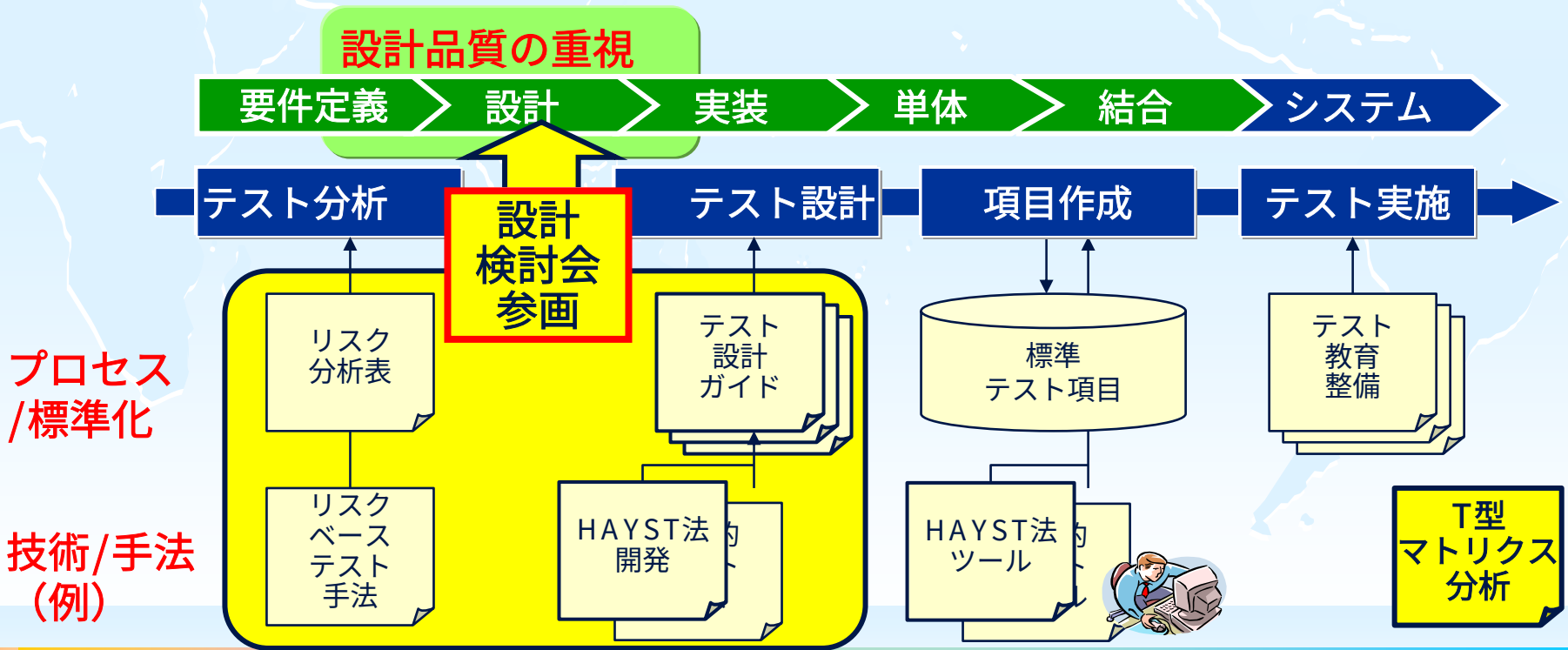


適用結果例



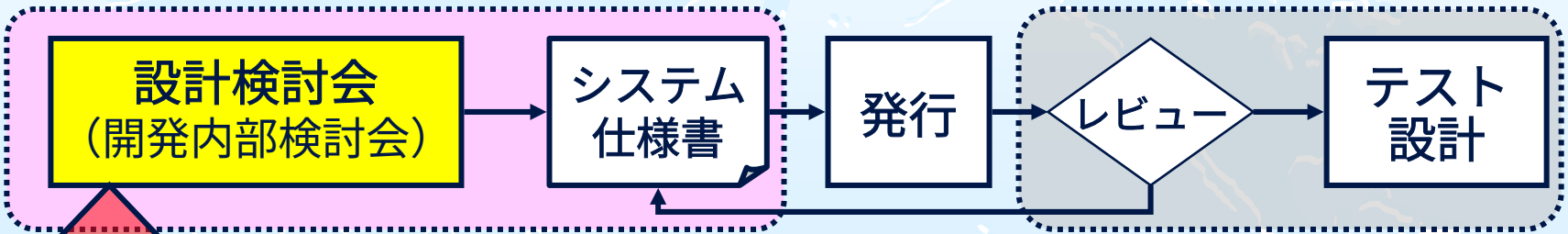
第３段階：設計品質向上への取組み

- システムテストで外部仕様問題を検出しても遅い
  - 基本設計の修正によるスケジュールインパクト
  - 暫定対策もしくは制限事項になりやすい
- システム仕様書/外部仕様書の発行時期が遅い。
  - 個々の機能はWF開発だが全体では刺身状開発
  - 十分に内部設計を詰めてから発行したい（開発者）



第３段階：仕様書を作る前にレビューする

- 仕様書が発行されてから指摘するのではなく、仕様書を発行する前に問題を指摘する。
  そのために、開発部門の設計検討会へ参画。



仕様を作る過程でレビュー　　出来上がった仕様をレビュー

**システム
テスト設計
の視点**

**市場不具合
再発防止
の視点**

＜例＞

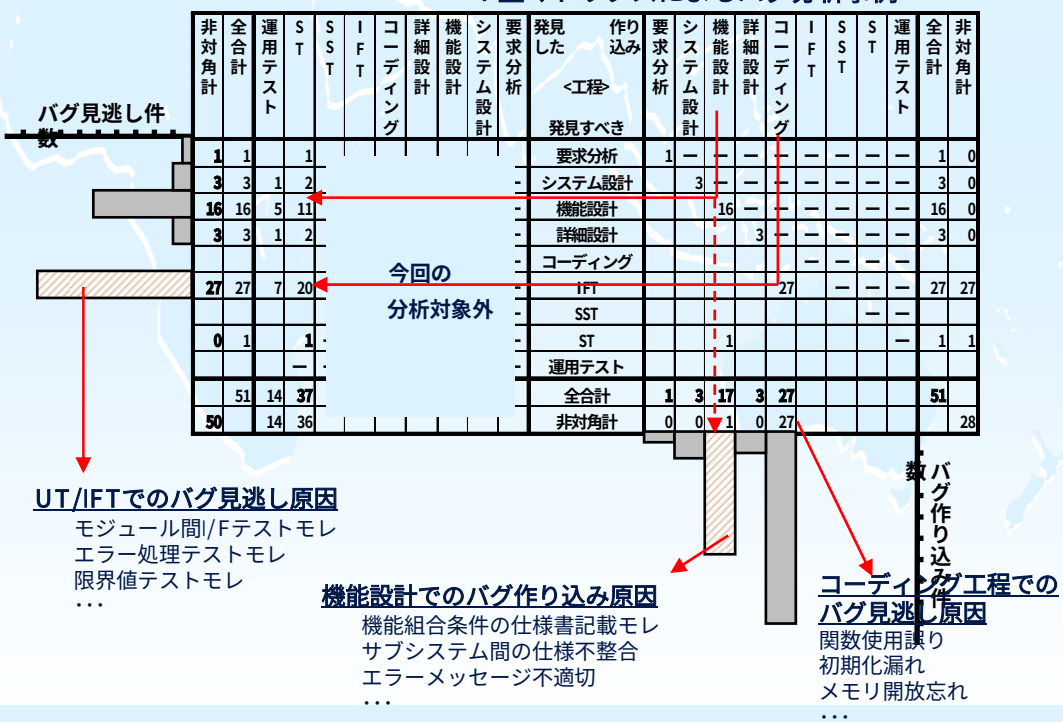
- 機能組合テスト設計：
  禁則条件は考慮できているか。
- 状態遷移テスト設計（動作モード）：
  遷移条件に他機器からのアクセスは考慮できているか。他機器の開発者と仕様をすり合わせたか。
- エラー/例外処理テスト設計：
  連携動作する他機器との異常処理を考慮できているか（他の機器がエラー、過負荷でレスポンス不可のときなど）

第３段階：市場不具合の徹底分析

- 市場不具合を１件１件地道に分析
  - 開発担当者へのインタビュー
  - 「人」ではなく「プロセス」の原因抽出
  - 現場が効果があると認める対策抽出
    ⇒　標準開発プロセスへ反映

時間はかかる

T型マトリクスによるバグ分析事例



非対角計	全合計	運用テスト	ST	SST	IFT	コーディング	詳細設計	機能設計	システム設計	要求分析	発見した 作り込み <工程> 発見すべき	要求分析	システム設計	機能設計	詳細設計	コーディング	IFT	SST	ST	運用テスト	全合計	非対角計
1	1		1								要求分析	1	ー	ー	ー	ー	ー	ー	ー		1	0
3	3	1	2								システム設計		3	ー	ー	ー	ー	ー	ー		3	0
16	16	5	11								機能設計			16	ー	ー	ー	ー	ー		16	0
3	3	1	2								詳細設計				3	ー	ー	ー	ー		3	0
											コーディング						ー	ー	ー			
27	27	7	20								IFT					27		ー	ー		27	27
											SST							ー	ー			
0	1		1								ST			1					ー		1	1
			ー								運用テスト											
	51	14	37								全合計	1	3	17	3	27					51	
50		14	36								非対角計	0	0	1	0	27						28

開発者へのインタビューシート

1. 不具合内容
① 障害現象と本来の仕様
② 間違ったアルゴリズム、正しいアルゴリズム

2. 作り込んだ原因
① どこの工程で作り込んだか？
② 開発者は仕様を正しく認識していたか？
③ 正しく認識していない場合、それはなぜか？

3. 検出できなかった原因
① レビューの対象になっているか？
② レビューしたのになぜ検出できなかったか？
なぜレビュー対象にならなかったのか？
② どんなレビューをすれば検出できるか？
③ なぜテストで検出できなかったのか？
④ どうすればテストで検出できるか？

4. 再発防止策
① どのような対策防止策が考えられるか？
② 他部門にお願いする対策があるか？

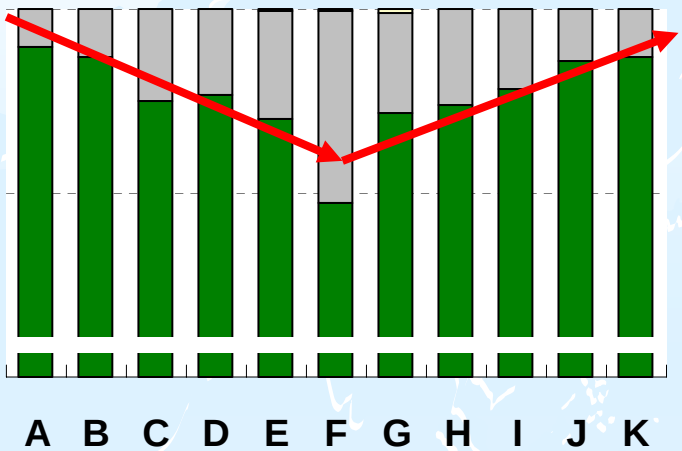
施策のまとめ

- 第１段階：テスト設計重視
  - テスト設計とテスト項目作成の工程分離
  - テストカテゴリ分け
  - テスト技法やツールの開発/導入
- 第２段階：テスト分析重視
  - テスト分析とテスト設計の工程分離
  - リスクスコアによる重点テスト戦略
- 第３段階：設計品質向上への取組み
  - 仕様書を発行する前にレビュー
  - レビューの視点はテスト設計と市場不具合の再発防止

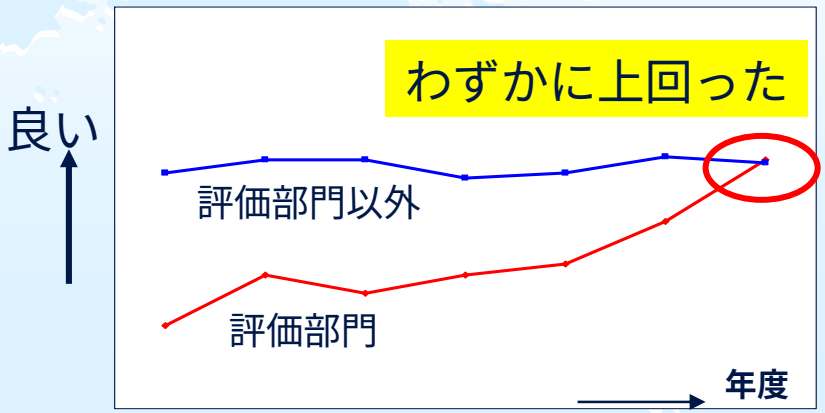
目標達成状況

施策の直接的な効果測定はできていないが、結果は以下の通り。

□ 開発部門の受け渡し品質

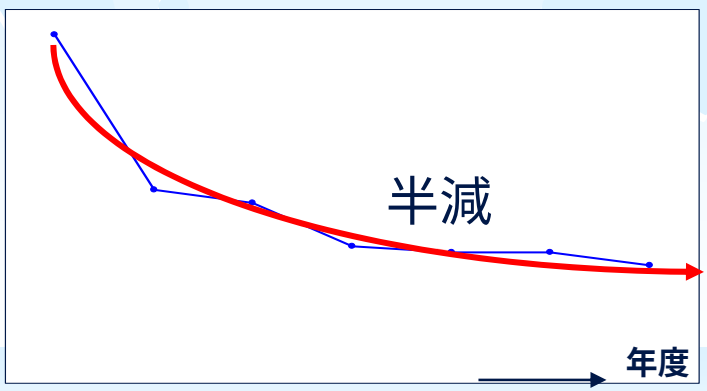


□ モチベーション

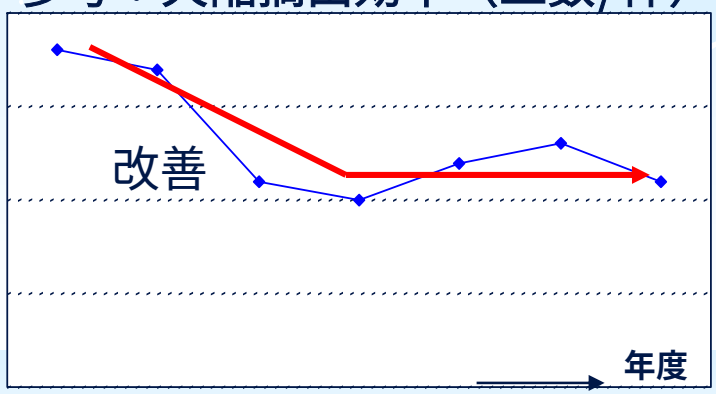


□ 第三者テスト工数

評価工数（KLOC当たり）



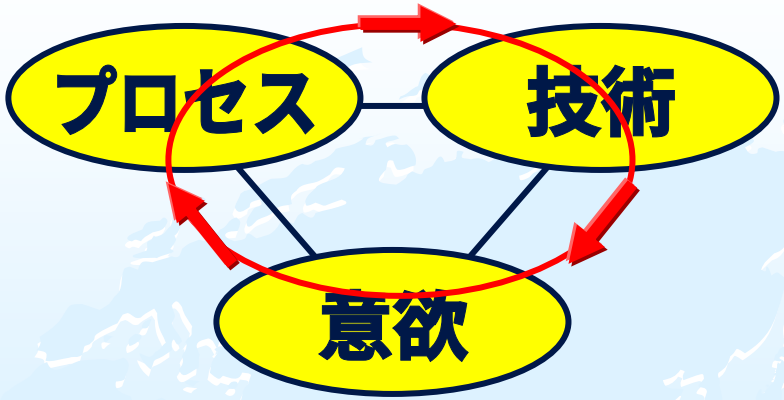
参考：欠陥摘出効率（工数/件）



第三者評価部門が陥りやすい状況

	Yes	No
● また設計工程が遅れそうだから、今年も正月/GWは無いな（納期厳しさ：上流＜下流）	□	☑
● 仕様はなかなか決まらないし、どうせ仕様変更があるから、テスト直前になってから準備しよう。	□	☑
● テストプロセス改善の前に設計工程をどうにかしてよ（自プロセス改善に後ろ向き）	□	☑
● 今までのやり方でバグは見つかるからテスト技法は不要。昔やってみたけど現場じゃ使えない。 ・大規模化・複雑化による難解さ（スケールギャップ） ・どこに適用すれば良いかわからない（スキル不足）	□	☑
● 市場で不具合が見つかると「テストは何をやっていた！」と言われる。設計に問題があるのに。	☑	□
● 本当は開発をやりたかった（若手に多い）	□	☑

活動のポイント



プロセス	技術/技法	意欲
・設計とテストは両輪。開発部門と対等の立場で活動を進める。 ・改善は少しづつ段階を踏んで。	・現場で使える技法を導入する。汎用品が無ければ自分達で作る。 ・ソフトウエア分野の手法に限定せずに、使えるものは何でも使う。	・改善は創造的な活動・前向きな活動・自主的な活動（啓蒙） ・プロセス改善活動への取組みを評価する。

ご清聴ありがとうございました。