

短期・高品質に向けた 単体テスト作業プロセスの改善

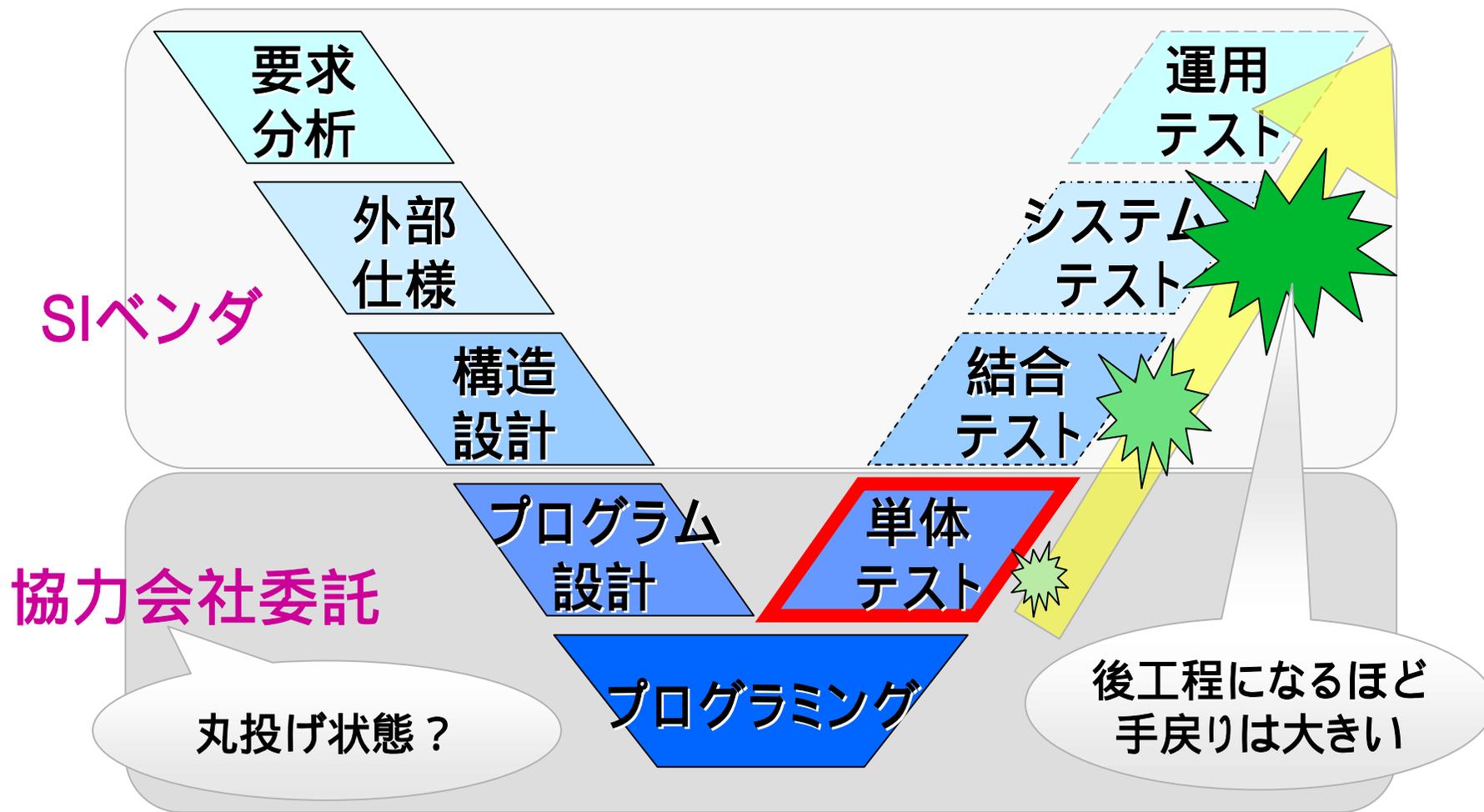
～ ユニットテストフレームワークの開発と
実践評価による効果検証 ～

2007年1月30日
富士通株式会社
田幡 宏之

- 単体テストの実態と課題
- ユニットテストフレームワークの開発
 - テスト仕様書の工夫
 - 自動実行の仕組み
 - 品質評価の方法
- ツールのデモ
- 実践での試行・評価
- まとめ
- 今後の展開

単体テストの重要性

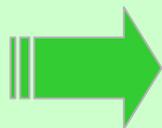
- 基幹系システム開発ではウォーターフォールが主流。
品質を作りこむためにも、最小単位からの検証(単体テスト)が重要。



単体テストの実態と課題

テスト仕様書は？

- ▶ レビューされていない
- ▶ そもそも作成していない



仕様書作成は、手間も時間もかかる。
レビューするのも大変。

テストの実施方法は？

- ▶ 担当者任せ
- ▶ 実施日付のみ記録



どんなデータで、どこまでテストされたのか分からない。
(本当にテストしたのか?)

品質保証は？

- ▶ 特に完了基準はない
- ▶ テスト項目数の指標だけ



どこまでテストすれば十分か。
テスト項目の数え方も曖昧。

単体テスト作業プロセス確立の必要性

属人性を排除

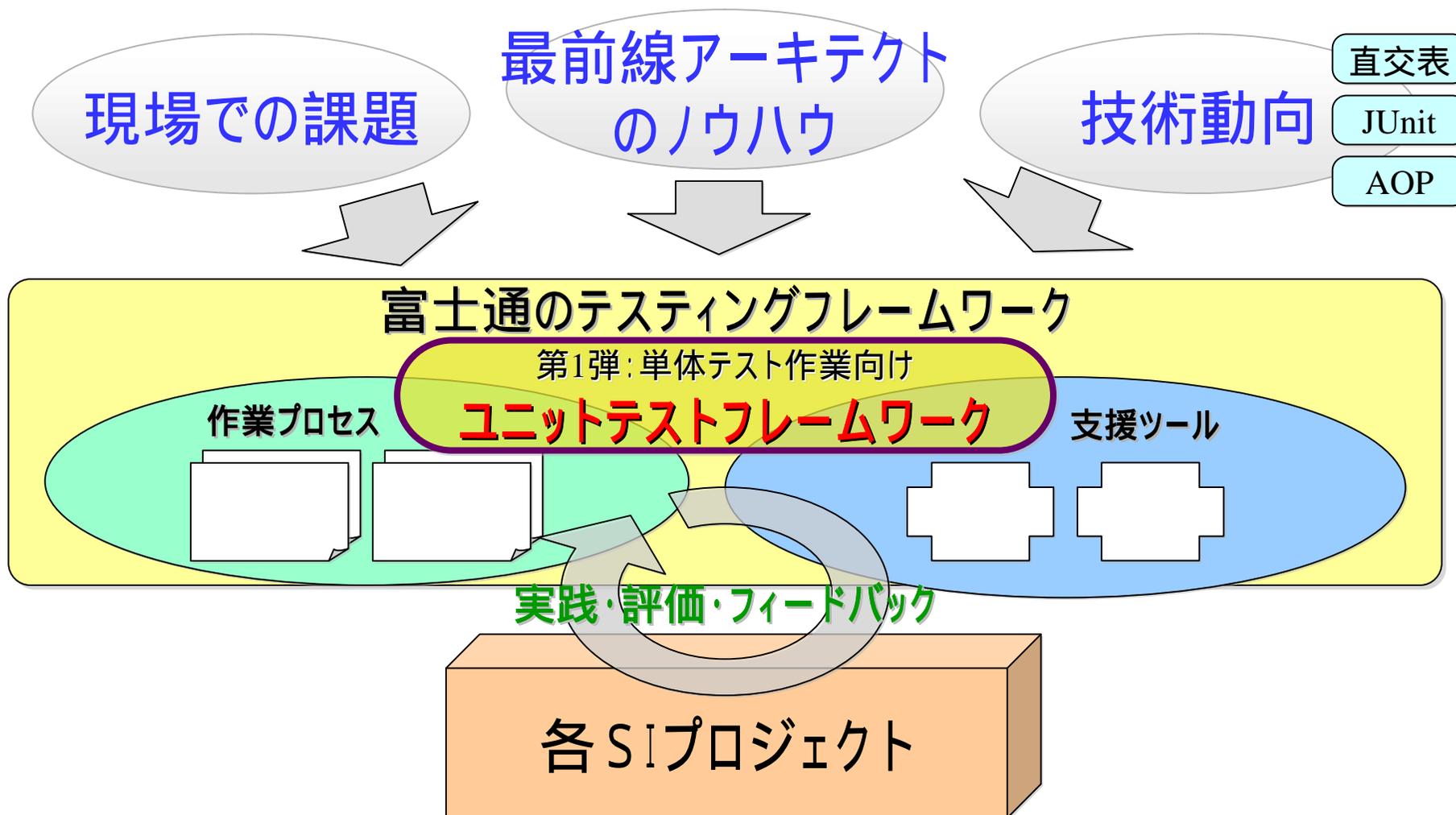
品質の均一化

生産性の確保

広く普及できる

ユニットテストフレームワークの開発

- 単体テストにおける作業プロセスを確立し、作業プロセスに沿ったツールとセットで提供。



テスト仕様書の改善

精度を高くしたマトリクス形式

テスト自動実行

テストデータ生成とドライバ・スタブ汎用化

品質評価方法

カバレッジ評価と明確なテスト指標

ユニットテストフレームワークの特徴 テスト仕様書の改善

- テストケース毎の入力・期待値データを明記することで、テスト項目と実施内容が一体化

従来のテスト仕様書

No.	テスト項目	検証内容	検証日	結果
1	一般ユーザでの通常権限ログイン	ログイン成功となること		
2	管理者権限でのログイン	ログイン成功となること		
3	有効期限切れユーザIDでログイン	ログイン失敗となること		
4	二重ログイン	ログイン中フラグがtrue		
5	部品呼出しで異常が発生した場合	例外を返す		

日本語記述で曖昧

- ・膨大な量
 - ・レビューしきれない
- 形骸化**

新しいテスト仕様書

入出力	項目名	入力/検証値		テストケース				
		値の種類	値	01	02	03	04	05
ログインMsg	ユーザID	一般ユーザ	0001					
		特殊ユーザ	9001					
	パスワード	一般ユーザ	000A					
		管理ユーザ	900A					
-	-	null						
ユーザマスタ (スタブ復帰値)	ユーザID	一般ユーザ	0001					
		特殊ユーザ	9001					
	パスワード	一般ユーザ	000A					
		特殊ユーザ	900A					
有効期限	期限内	0						
	期限切れ	1						
結果	ユーザ名	一般	鈴木					
	管理者	田中						
(復帰値)	ログイン可否	ログイン成功	true					
	ログイン失敗	false						
ログイン管理 (スタブ引数)	ログインフラグ	ログイン中	1					
	非ログイン中	0						
ユーザID	一般ユーザ	0001						
	特殊ユーザ	9001						
ログイン日時	-	-						
-	-	NullEx						

入力データ

期待値データ

検証内容も
具体的なデータ値で

テストケース

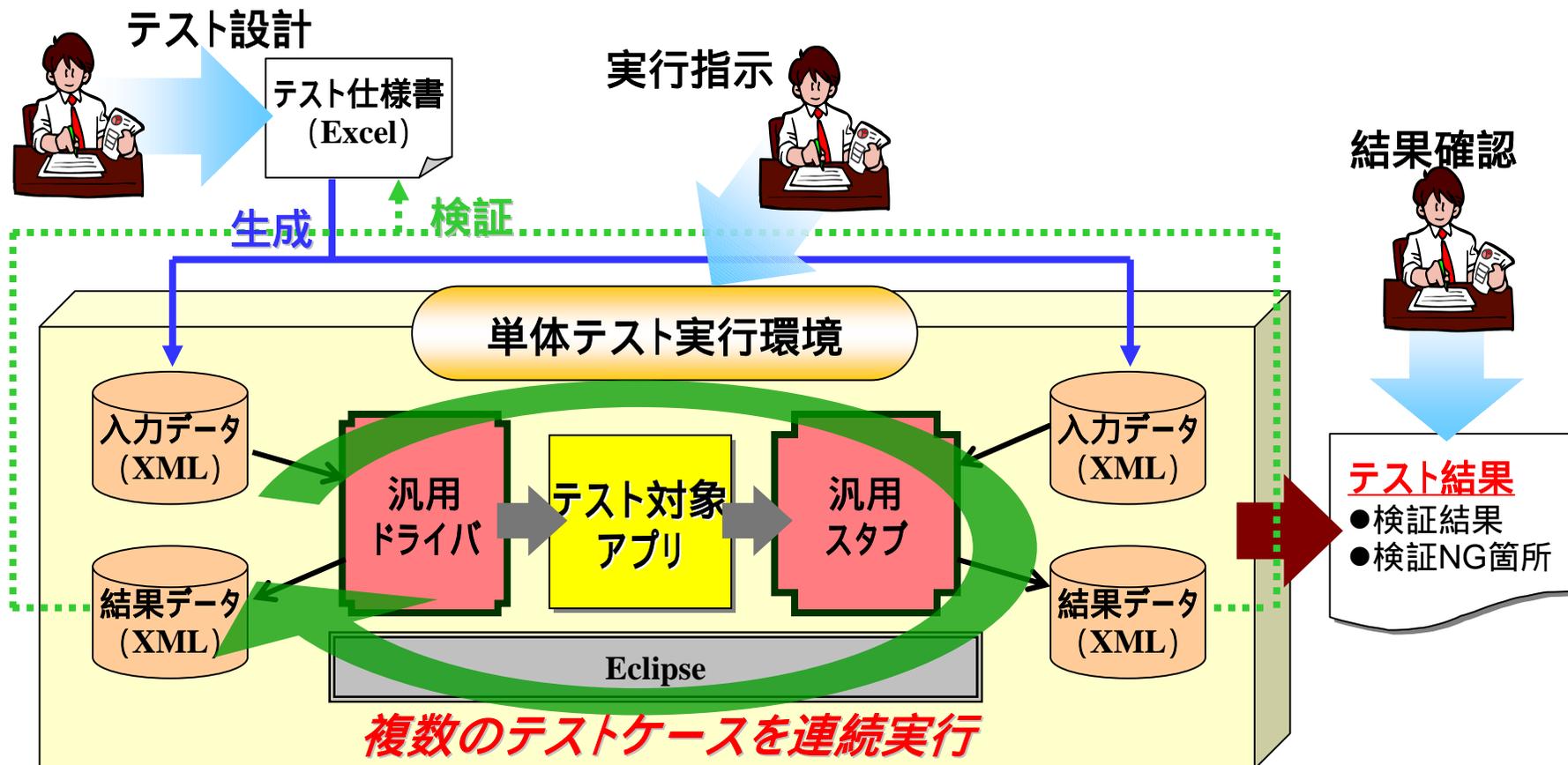
入力データ値
バリエーション列挙

テストケースとデータの
マトリクス

一過性のドキュメントではなく、保守にも活用・継承できる開発資産に！

テスト自動実行の仕組み

- テスト仕様書からテストケース・テストデータを生成し、実行・結果検証は自動化



同じテストの繰り返しや目視検証など、不効率な作業を自動化！

品質評価方法

- 行カバレッジによる品質評価と、テスト実施結果をエビデンスとして自動記録

従来の単体テスト指標

品質指標: 試験密度 (テスト項目数 / ステップ数規模)

開発記録: テスト仕様書兼結果報告書

達成指標には
不十分

自己申告での
実施記録

作業品質の強化

- **行カバレッジ**
ホワイトボックス観点での評価。
- **確実な検証**
期待値比較により、目視検証モレを防ぐ。
- **試験密度**
従来の曖昧なテスト項目数でなく、
テストケース数・検証データ項目数が明確。

テスト作業記録

以下のエビデンスをツールで管理

- 検証結果レポート
- 行カバレッジレポート
- テスト仕様書 (入力・期待値データ)
- テスト結果ログ (出力結果データ)
- テスト実行履歴 (検証NG件数含む)

一定の品質が保証され、エビデンスも確実に蓄積される！

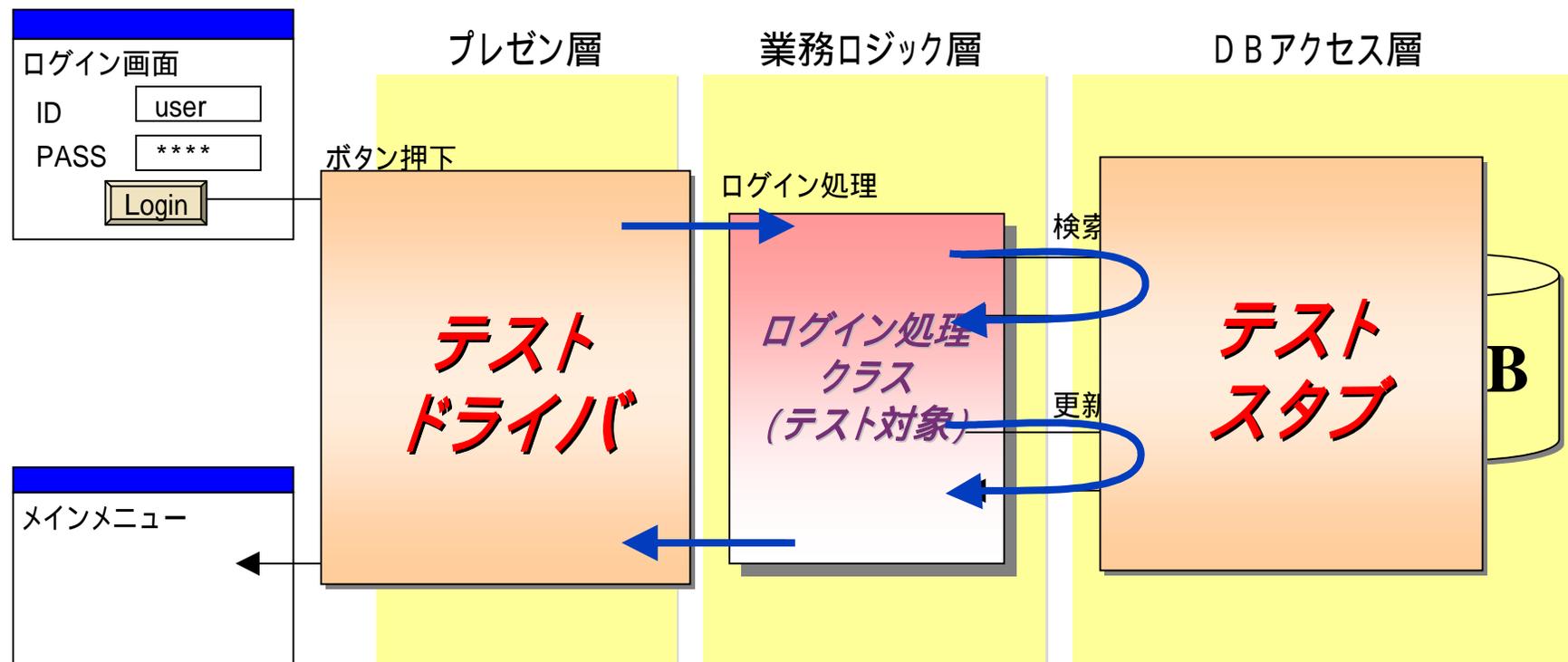
ツールデモ

デモ用アプリの概要

- Webの3層アプリ構成における業務ロジック層アプリの単体テストを実施。プレゼン層・DBアクセス層はドライバ/スタブで代替する。

<ユースケース概要>

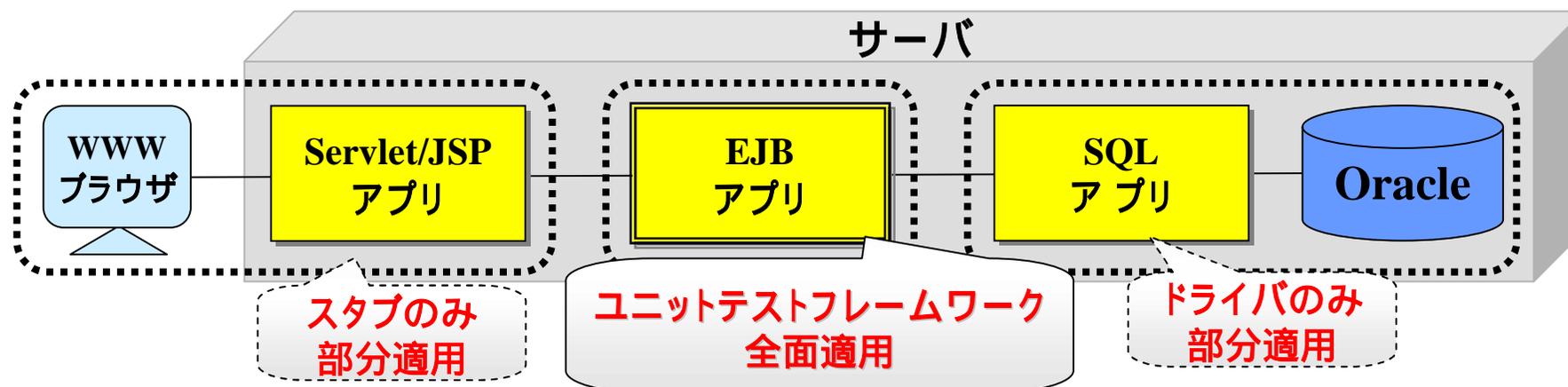
ログイン画面にて、ユーザID/パスワードを入力し、ログインボタンを押下する。
ID/パスワードのチェックを行い、ログイン処理が正常に完了した場合はメニュー画面へ遷移する。



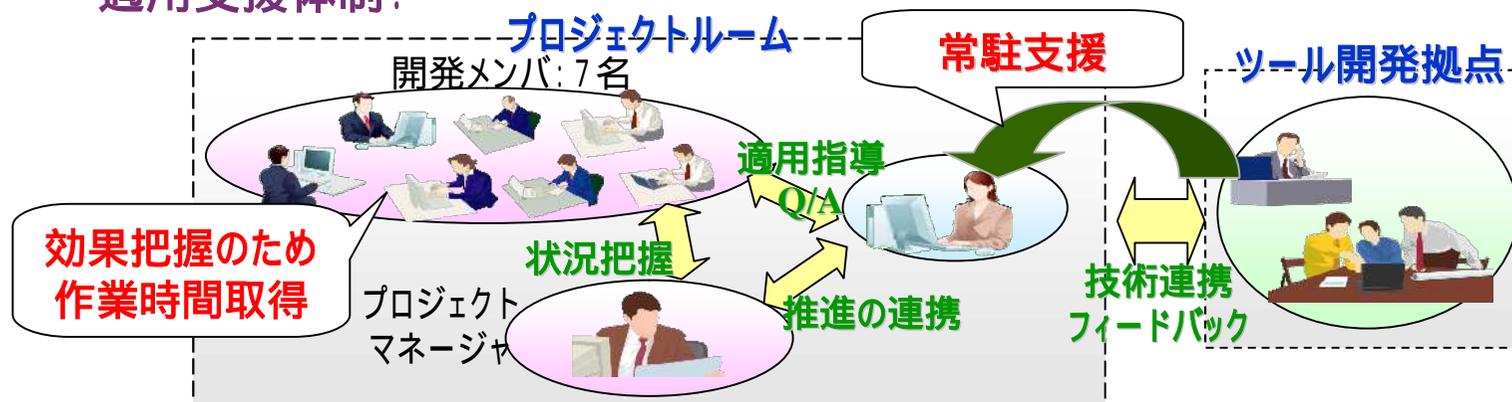
実践評価

実践プロジェクトの概要

プロジェクト名: A証券様 照合・決済業務
開発規模: 約80KStep (Java)
単体テスト期間: 2ヶ月



適用支援体制:



実践結果

- 試行段階のツールでも、品質・生産性ともに十分に効果があることを証明

品質

PT品質 約1.6倍向上

- 規模あたりのテスト項目設定・障害検出率は、従来より高い。

生産性劣化させずにテストの質を向上

- 後工程を分析し、単体テストでの検出漏れ障害は4割減少(品質として1.6倍向上)。

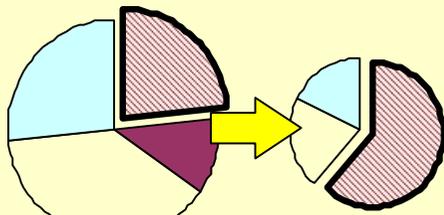
後工程での手戻り作業削減

	従来比
単体テスト試験密度[件/Kstep]	6割増
単体テスト障害検出率[件/Kstep]	4割増
単体テスト不足などによる 後工程障害発生率[件/Kstep]	4割減

単体テストで検出すべきもの

生産性

従来と同等以上



PT作業の時間配分

- サンプルングではPT生産性向上。
- 時間配分では、PT仕様書作成が作業の中心に。

適用性

幾つか課題あり

今後へフィードバック

< 現場からの指摘事項 >

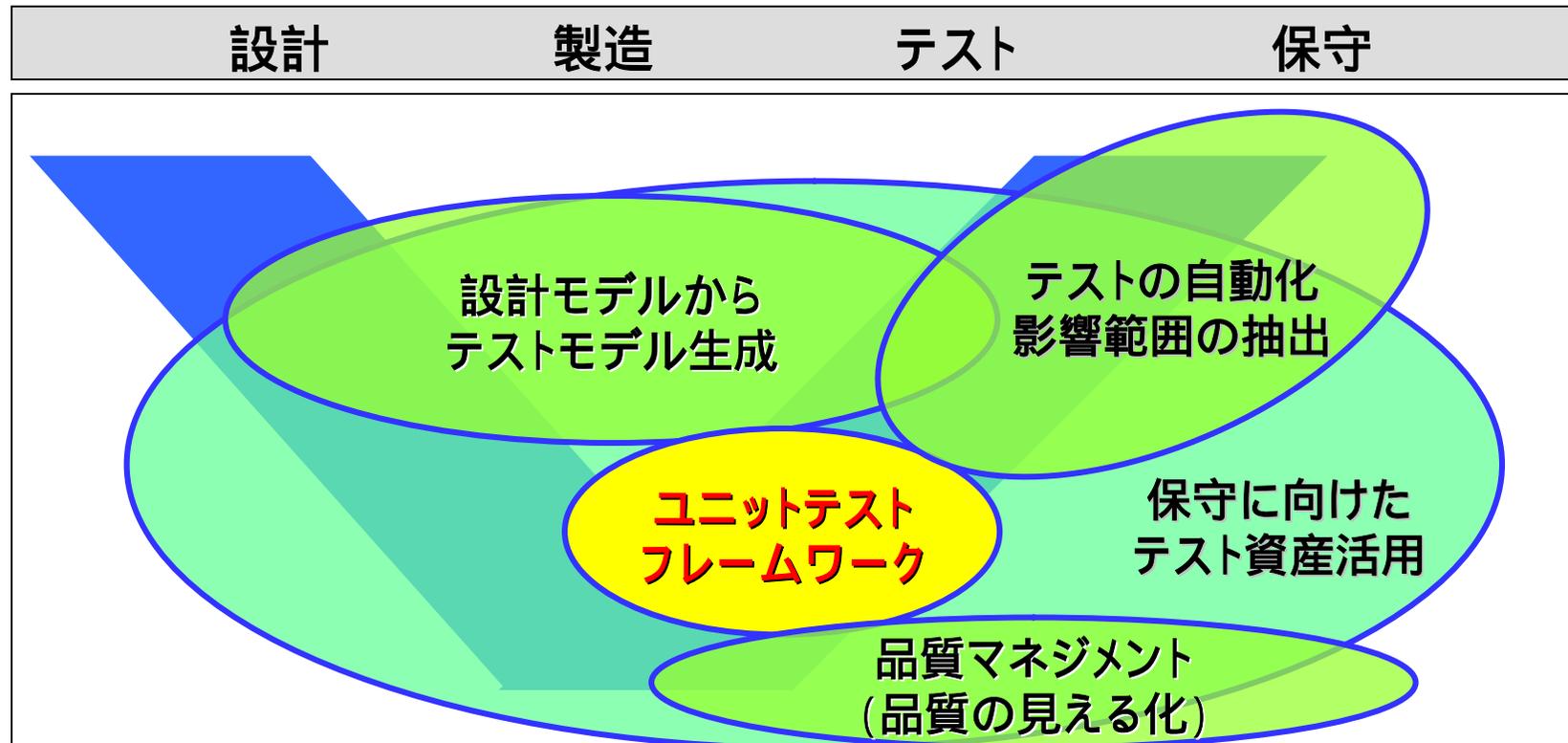
- PT仕様書が肥大化した時の可視性
- 保守に活かすためのPT資産管理方法
- ツール操作性やドキュメントへの改善

etc.

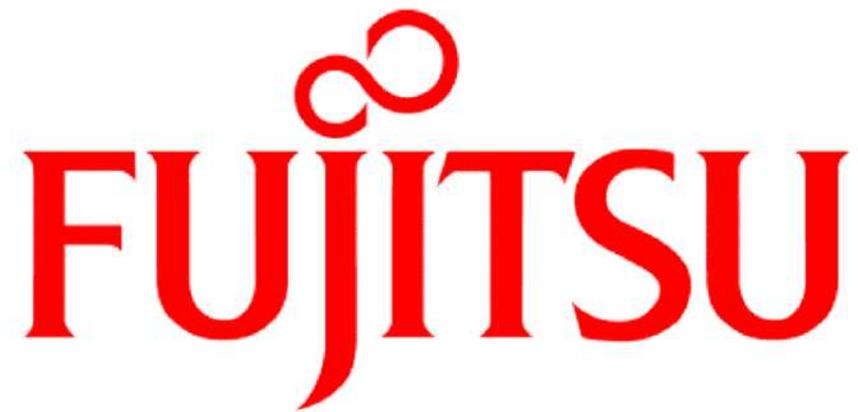
- **現場の問題点・要因・背景を把握し、あるべき作業プロセスを再定義**
理想像や流行り手法をはめるだけではダメ。
- **既存の技法・ツールの考え方をうまく組み合わせる**
そのまま適用するのは難。作業プロセスとして成り立つよう応用する。
- **煩雑な手作業を徹底して自動化**
その分、品質向上に向けた作業に集中化させる。

今後の展開

結合テスト・システムテストにも取組み、設計・保守を含めたシステムライフサイクル全体をカバーした品質向上施策へ。



業務システム開発における
テストフレームワークとして確立していく！



THE POSSIBILITIES ARE INFINITE