JaSST '08 Tokyo 事例発表

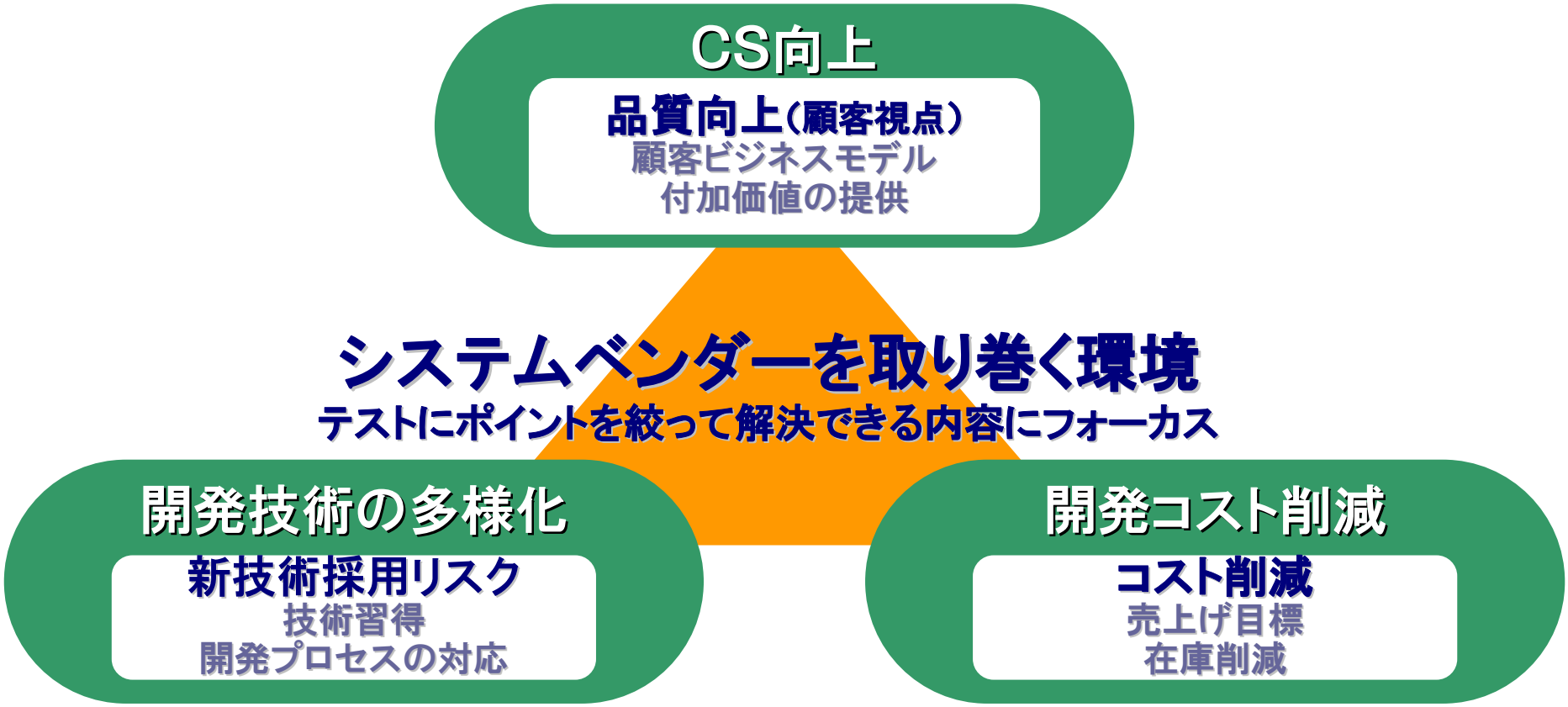
スクリプトレスによるテスト自動化事例

2008年1月31日

NECネクサソリューションズ株式会社
システム・サービス開発事業本部　技術開発事業部
小池輝明

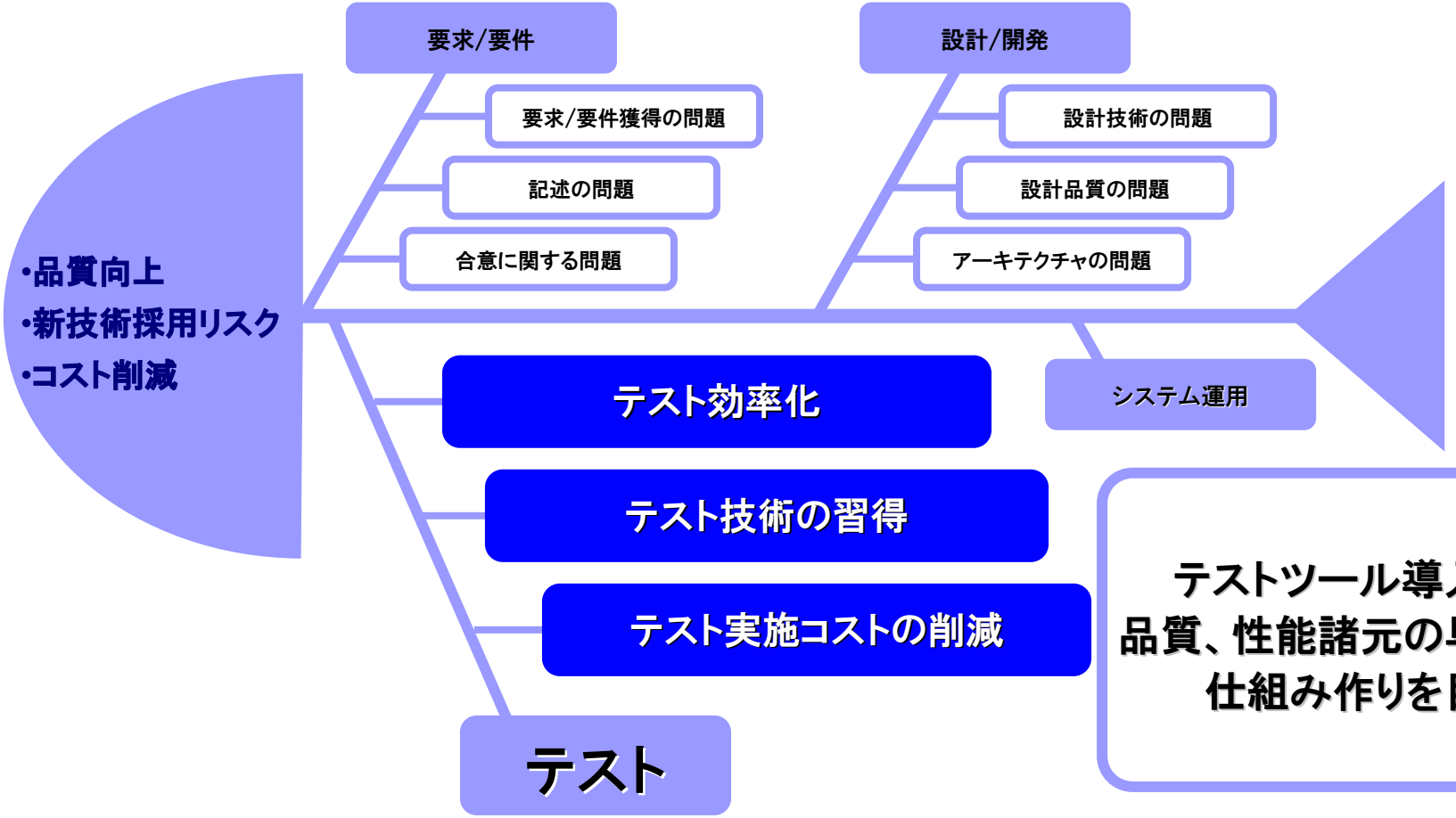
システムベンダーを取り巻く環境

CS/開発技術/経営的視点で現在のシステムベンダーを取り巻く環境を分析
品質向上・新技術採用リスク・コスト削減はテストにおいても重要な課題



テストにフォーカスした問題の抽出

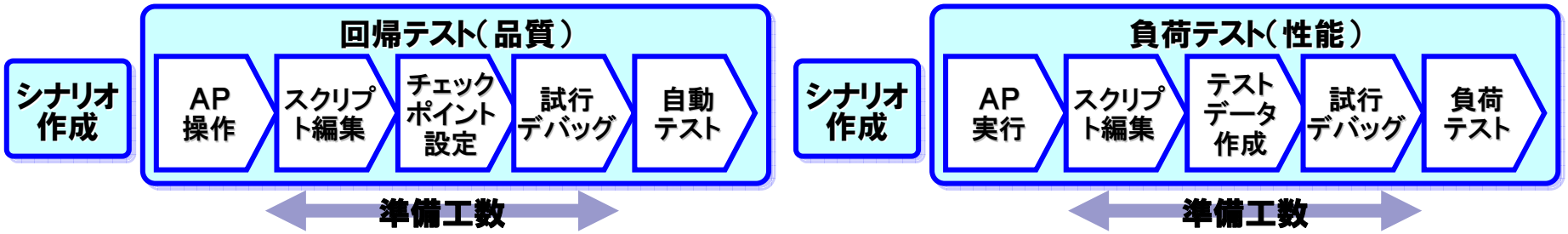
新技術採用リスクの顕在化、品質問題の早期発見、コスト削減を実現するために
「品質・性能諸元の早期確認」の仕組み作りが必要



テストツール導入による
品質、性能諸元の早期確認の
仕組み作りを目指す

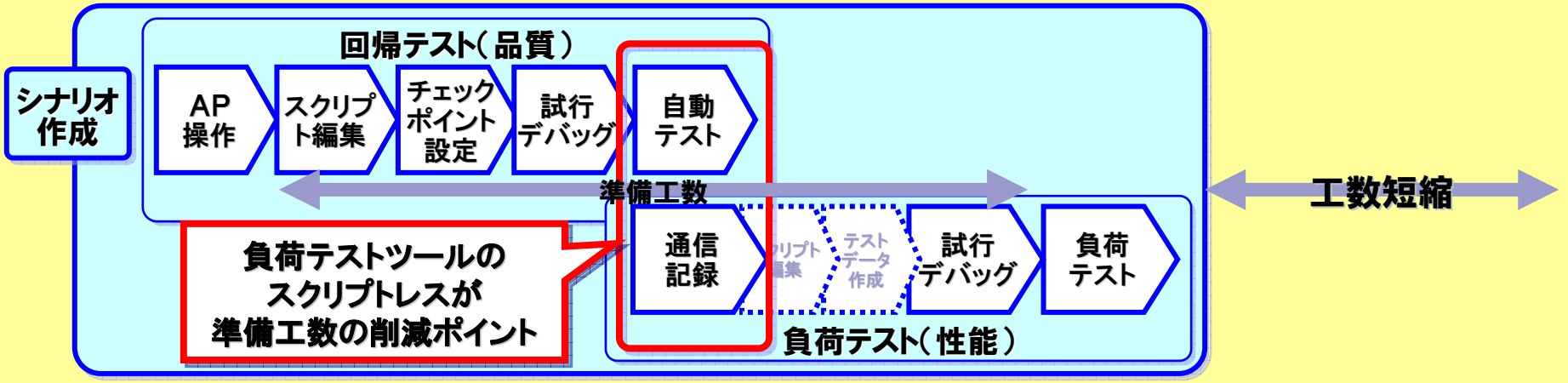
「品質・性能諸元の早期確認」の仕組み作りの問題点

回帰テストによる品質の確保、負荷テストによる性能諸元の確認の効率化を検討
ただし、ツール導入準備工数がオーバーヘッドになる問題点がある



準備工数を削減するためにツールを連携させて
準備工数の削減を狙う

今回の事例の
重要ポイント



回帰テストと負荷テストの連携対象可能領域の検討

回帰テストの単純性
◎：単純な操作の繰り返し
○：実行条件により単純作業あり
△：単純作業は少ない

負荷テストの適合性
◎：負荷テスト向き
○：実行条件により負糧テスト可
△：負荷テスト不向き
ー：負荷テスト領域外

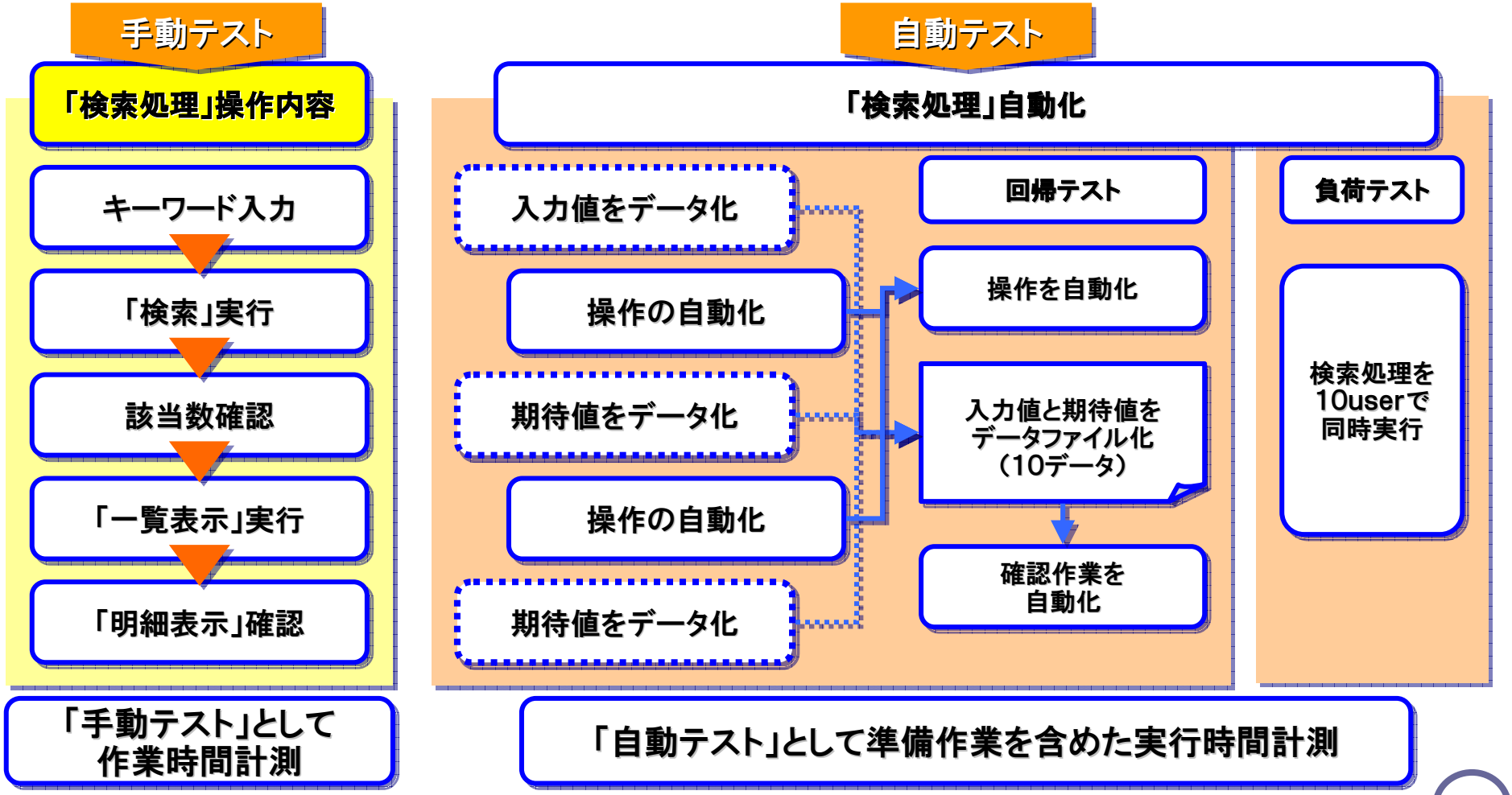
	テストカテゴリ									
	UI機能テスト			処理条件データ処理の組合せテスト		イレギュラー処理テスト		他システムI/F	システム要件のテスト	
	入力表示	GUI操作	帳票	登録検索更新	データ処理条件	エラー処理	排他処理	通信ファイル	並行稼働	業務フロー
回帰テストの単純性	◎	◎	◎	◎	○	○	○	△	△	△
負荷テストの適合性	ー	ー	ー	◎	○	○	○	△	△	△

	データ処理		
	登録	検索	更新
回帰テストの単純性	○	◎	△
負荷テストの適合性	○	◎	△

回帰テストと負荷テストの特性を考慮すると検索処理が共通領域であることが判明。この共通領域のテストを効率的に行うことで、検索処理の品質・性能諸元の早期確認が可能になると予想。

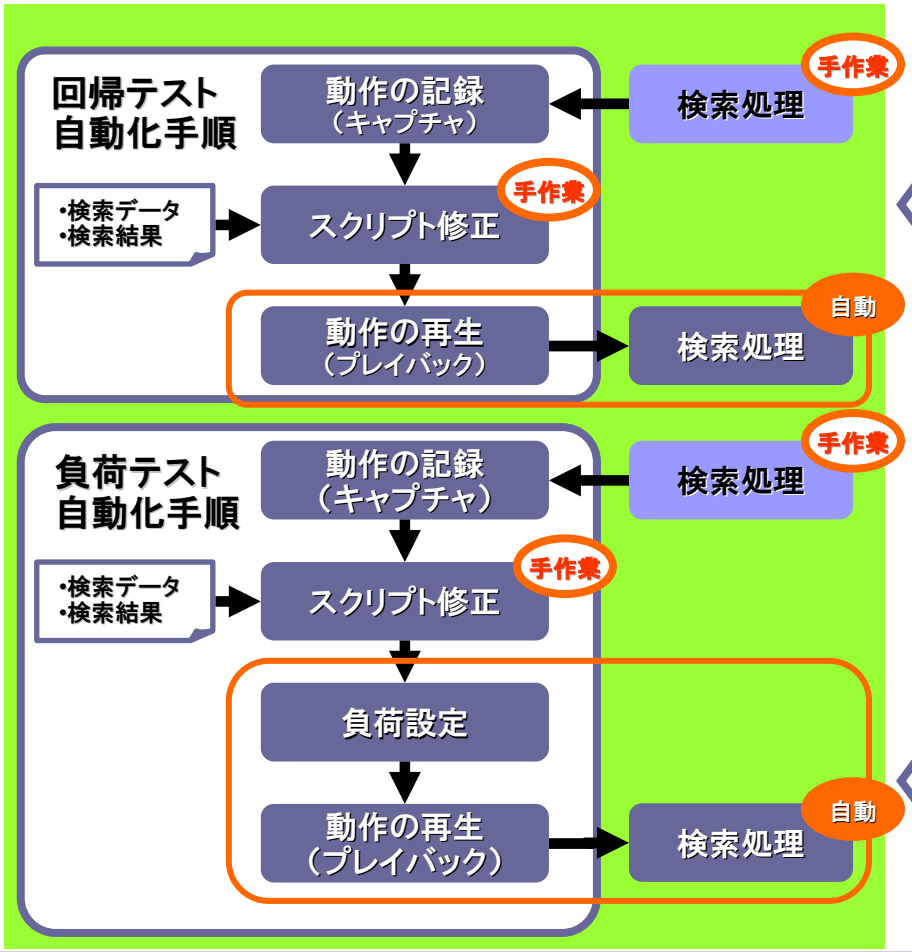
検索処理における自動化の検証方法

手動テストの場合と回帰テスト、負荷テストを含めた自動テストの手順の概要



連携前の回帰テスト自動化手順、負荷テスト自動化手順

回帰テストと負荷テストはツールの導入によって自動化は可能
ただし、ツール毎に導入工数や自動化の準備作業が発生



回帰テストの自動化

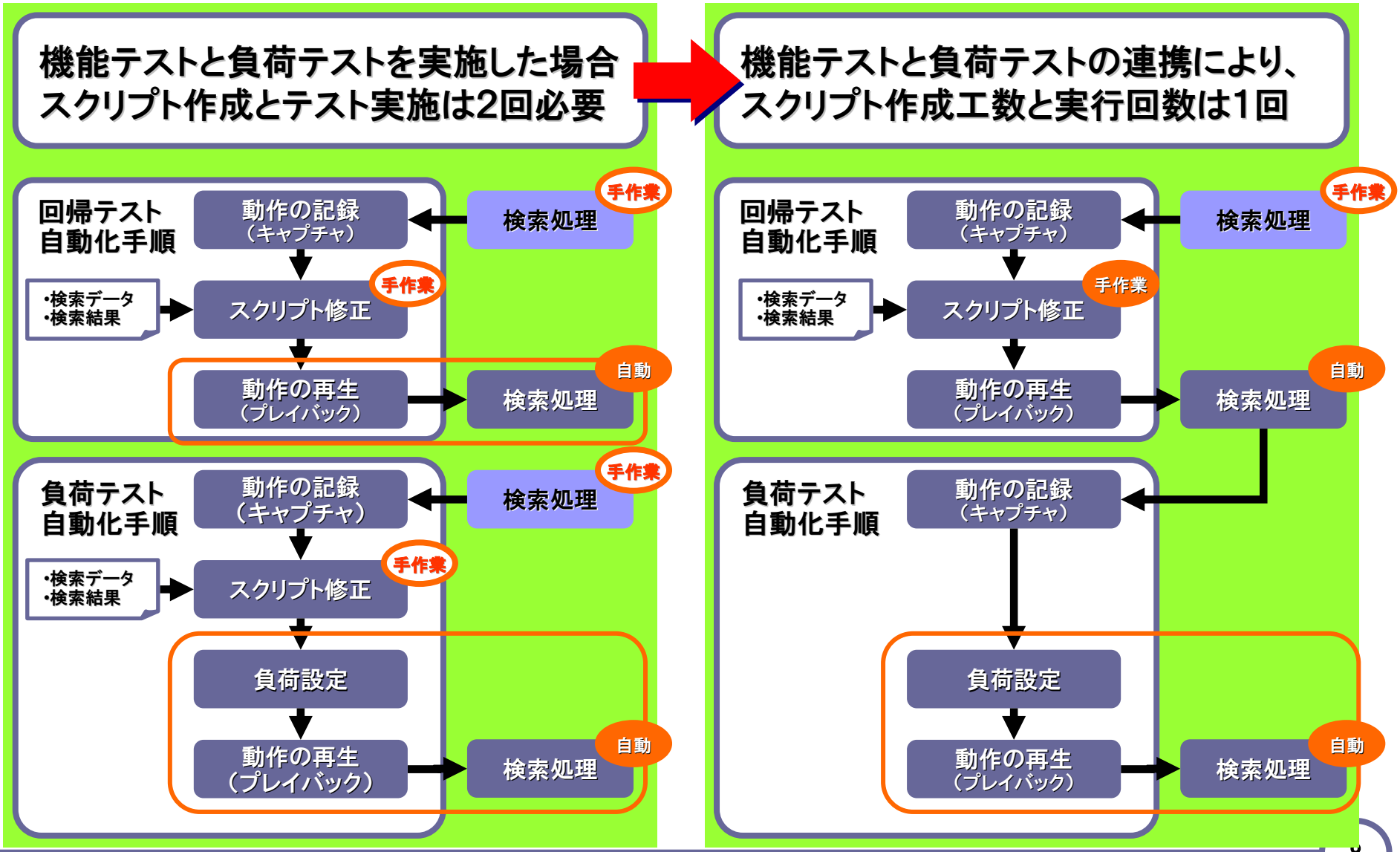
自動化ツールとアプリケーションの適合性評価
スクリプトの試作と実行確認。
自動テストの実施。

それぞれに手作業が発生し、
自動化のオーバーヘッドになる。

負荷テストの自動化

負ツールとアプリケーションの適合性評価
負荷テストシナリオの作成。
スクリプトの試作と実行確認。
自動テストの実施。

回帰テストと負荷テストの連携内容の詳細



回帰テストと負荷テストの連携による効率化

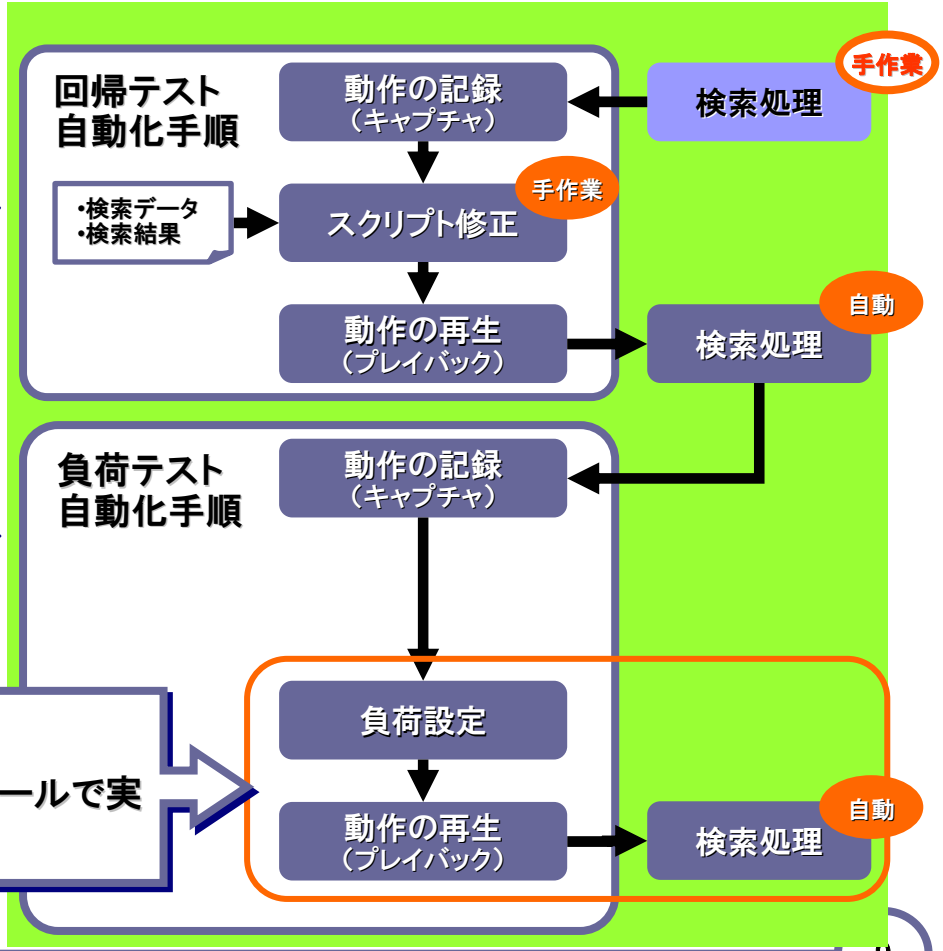
回帰テストと負荷テストは共通テストカテゴリに対して連携した導入が可能
但し、テスト目的や効果を予測し、計画することが必要

シナリオ作成のための回帰テスト自動化

ここで作成したシナリオは、負荷テストで使用。
そのため、テストデータを豊富に用意しておく。
テストデータ毎のシステム負荷を事前に把握。

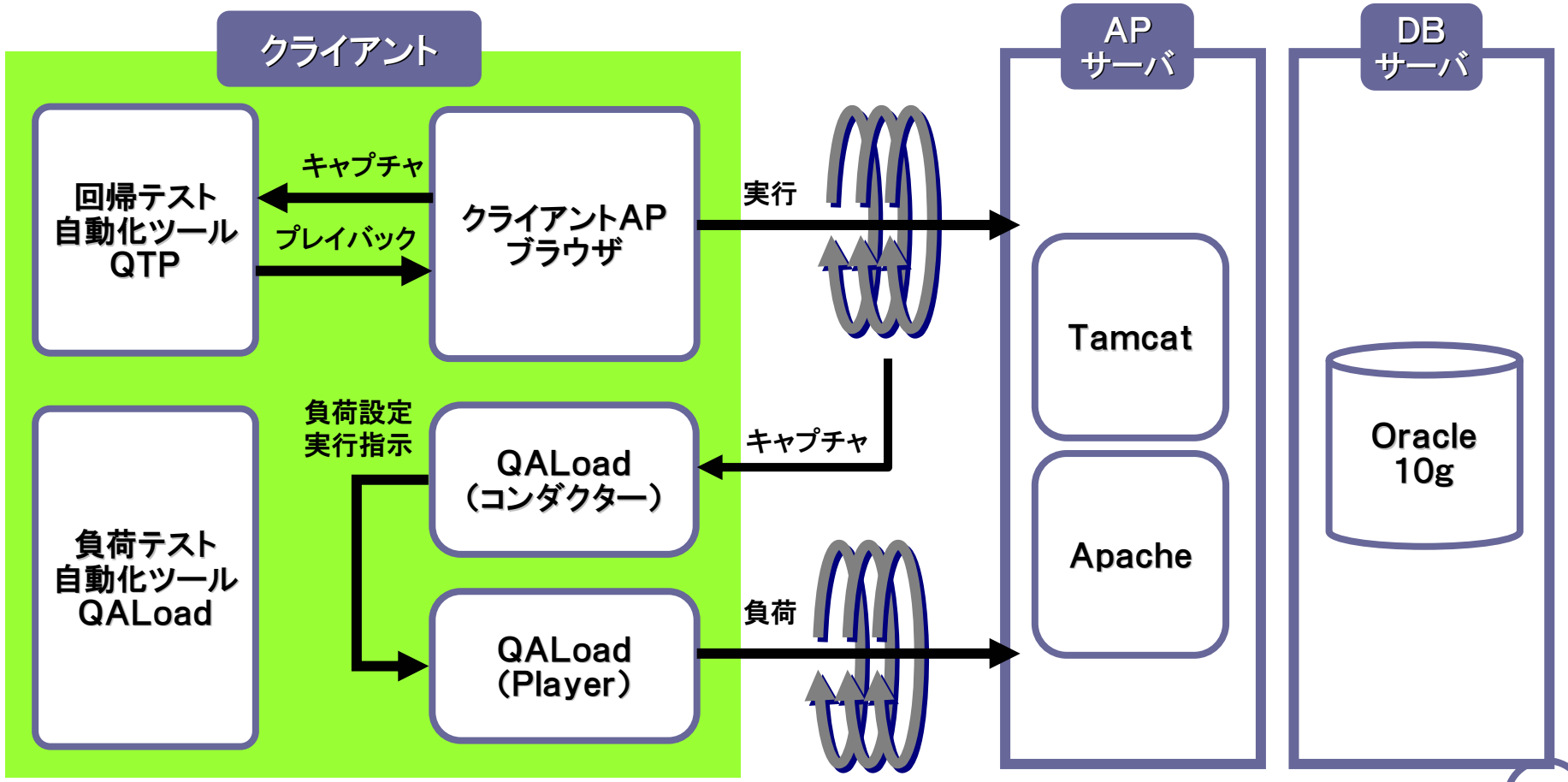
「回帰負荷テスト」の自動化（スクリプトレス化）
回帰テストで実行される通信をキャプチャ。
正確にキャプチャされているかの確認。
所定のシステム負荷がかかっていることの確認。

「回帰負荷テスト」の実行
回帰テストと機能テストを実行する場合には、負荷テストツールで実施する。



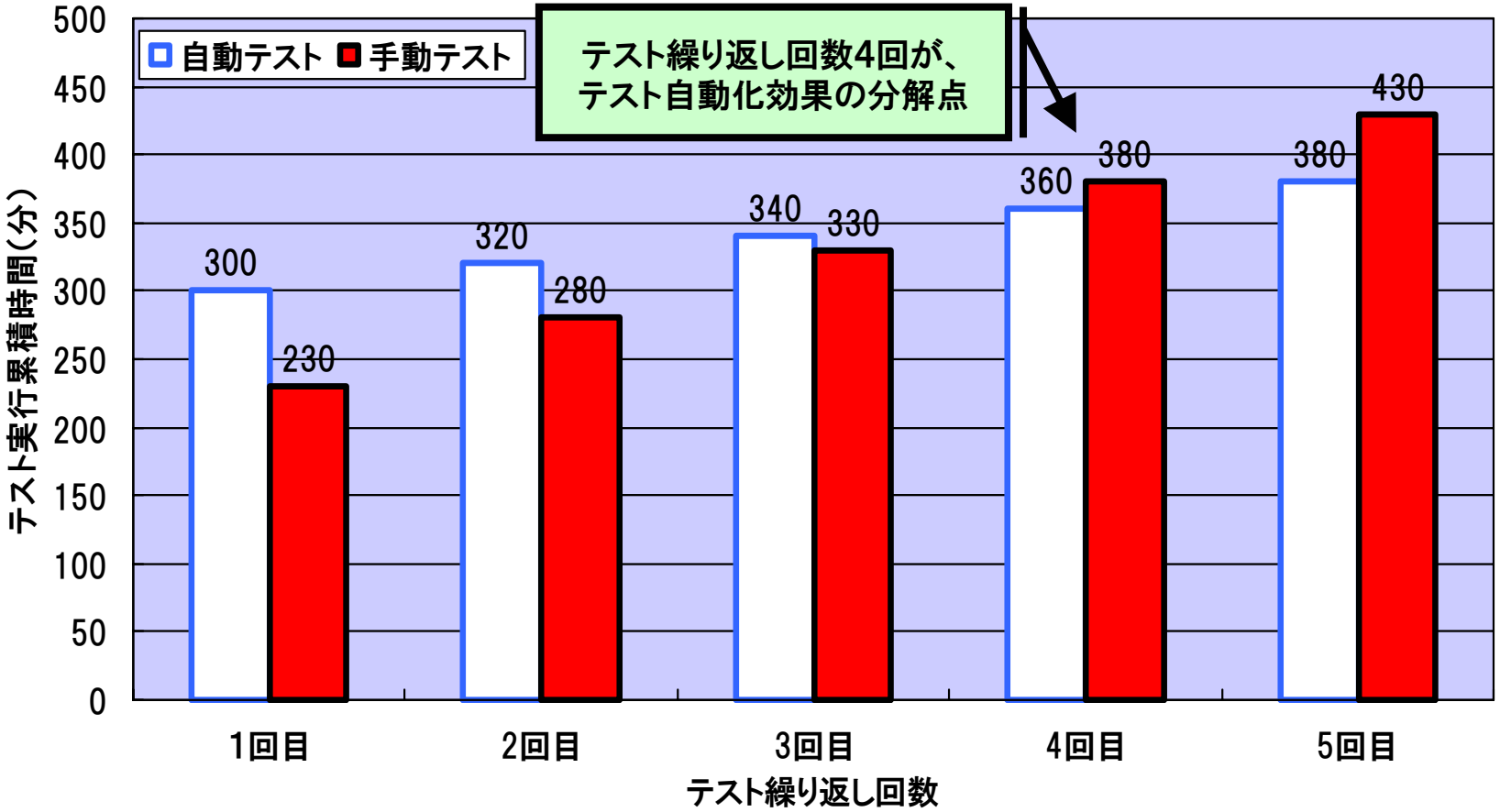
テスト環境の概要

自動テストツールはクライアント側に用意
回帰テスト自動化ツール：QuickTestProfessional
負荷テスト自動化ツール：QALoad



回帰テストの自動化による効果

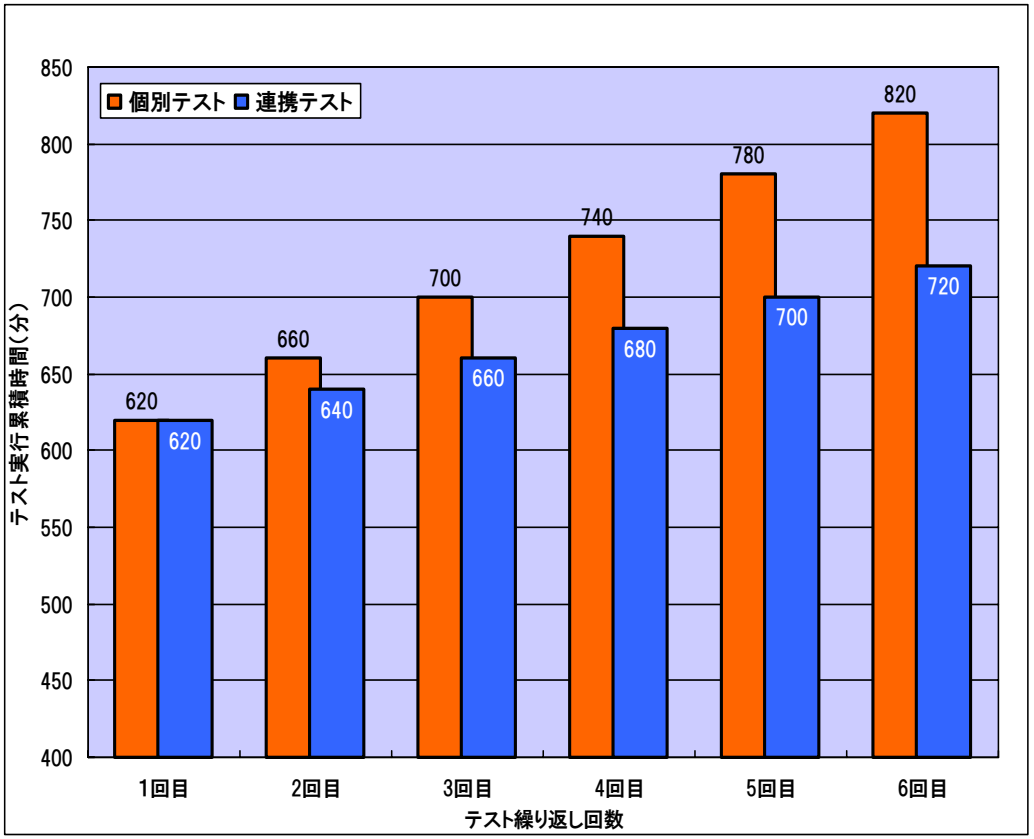
回帰テストと負荷テストの連携の前に、まずは回帰テストの自動化による効果を検証
回帰テストの自動化は4回以上繰り返すことで、準備にかかった工数を回収可能



回帰テストと負荷テストの連携効果

回帰テストと負荷テストの個別テストと連携で実施した連携テストを比較
テスト実施の効果は回数を重ねる毎に増大することを確認

		個別テスト		連携テスト	
		初回	増分	初回	増分
回帰テスト	(1)回帰テスト設計 (2)手動実行 (3)テストデータ作成 (4)スクリプト作成、試行 (5)チェック項目設定	300	0	600	0
	(6)回帰テスト自動実行		20		
負荷テスト	(1)負荷テスト設計 (2)手動実行 (3)テストデータ作成 (4)スクリプト作成、試行 (5)負荷計画	320	0	0	0
	(6)負荷テスト自動実行		20	20	20
	合計	620	40	620	20



個別テストでは、回帰テストと負荷テストで準備工数が620分発生し、実施する毎に40分を要する。
連携テストでの負荷テストの準備工数は発生しないが、連携を考慮したテスト設計に時間を要する。但し、実施工数は負荷テストの20分のみ。

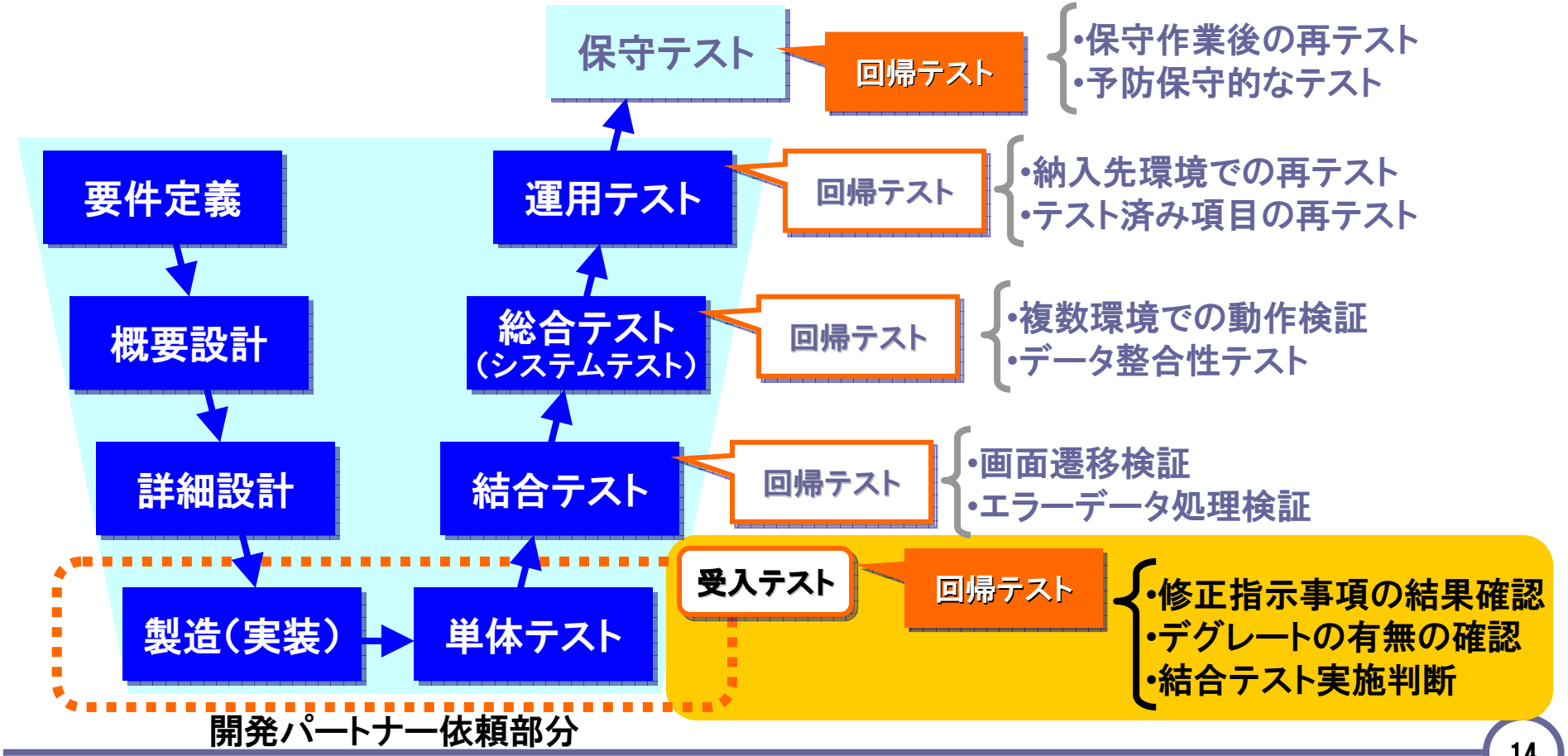
自動化連携のポイント(連携効果の明細の補足)

連携を行うためには、ツールレベルでの詳細な機能が必要
テスト設計・テストデータの準備で専用設計するなどのポイントがある

- QTPによる機能テスト用スクリプトを流用
- 負荷をかけたいノードに対する負荷のかかる処理を選択し、専用の自動テスト設計を実施
- QALoadのデータテーブルを使わずに、QTPでデータテーブルを用意
- データの異なるQALoadスクリプトを複数作成できるようデータを準備
- QALoadを記録モードにしてテスト対象を操作
- ループ処理の入れ子構造が表現できるようにデータテーブルを用意
- QALoadのスクリプト記述形式をCに設定
- スクリプト中の2バイト文字を8進数表記に設定

システム開発における適用場面

今回の事例は検索処理重視のシステムであり、回帰的に負荷テストが可能
システム改修実施後に、機能と性能の確認の実施を短時間で実現



結論

結論

回帰テストと負荷テストを連携させることによる、負荷テストのスクリプトレスはほぼ実現。
最低限の品質レベル、性能諸元の達成確認が可能。

制限

負荷テストツールではデータ駆動型は行わない
回帰テスト時のキャプチャ内容が負荷テストで実現できること

課題

負荷テストツールでの綿密な動作確認は必要。
（所定の負荷がかかること。処理が正常に行われること）
アプリケーションによってはこの検証作業に時間がかかる。

付帯事項

機能テストツールでテストシナリオが完成していれば、負荷テストツールの移行も容易に可能。

スクリプトレスによるテスト自動化事例

2008年1月31日
NECネクサソリューションズ
小池輝明