



**MERCURY™**

**効率の良いテストシナリオ  
- ソフトウェアテスト・ミーティング -**

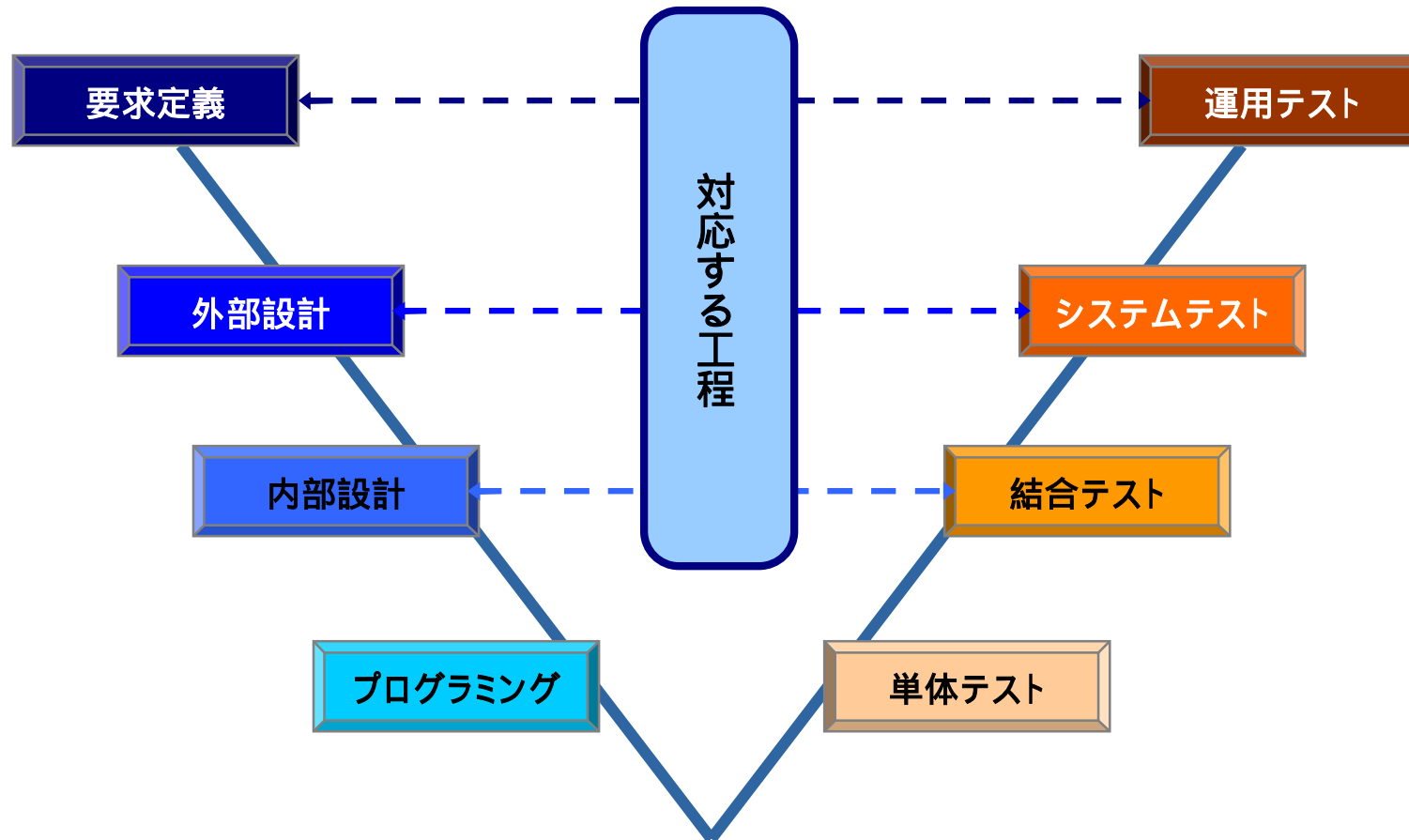
マーキュリー・インタラクティブ・ジャパン（株）

小崎 将弘

# 効率の良いテストシナリオ？

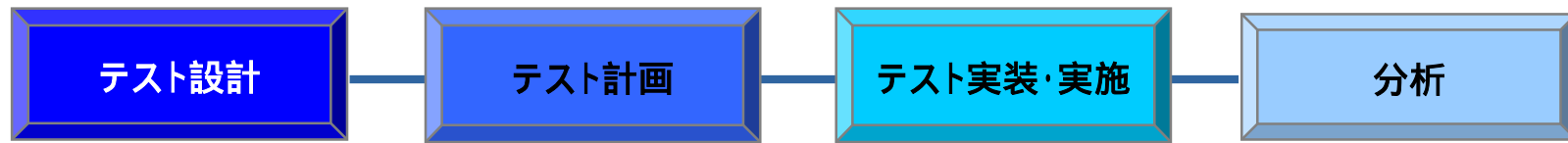
- テストの進め方
  - テストプロセス
- テストの設計
  - より少ないテストケースで
  - より多くのバグを見つける

# 開発工程とソフトウェアテスト



# ソフトウェアテスト

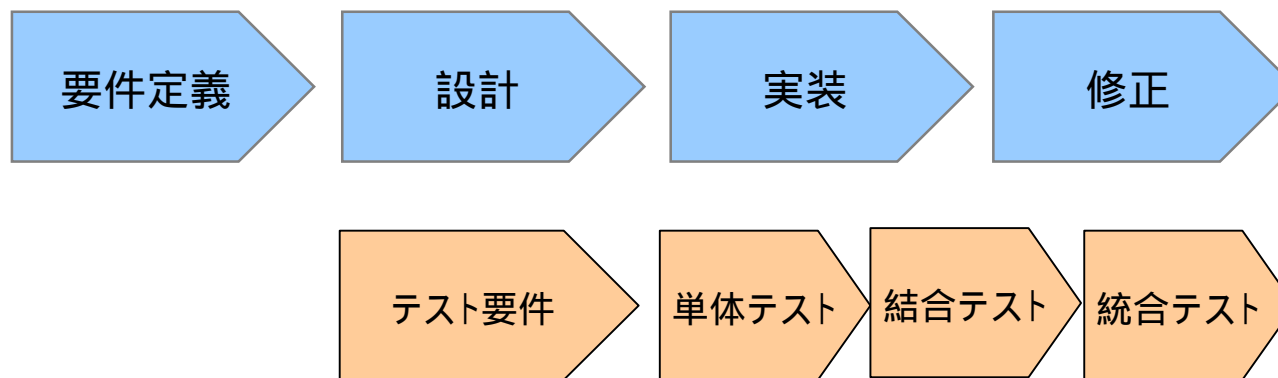
テストも開発プロジェクトとのタイプと同様に、プロジェクトと考えることができる



- テスト項目
- スケジュール
- 担当
- 判断基準
- 優先度

# テスト計画

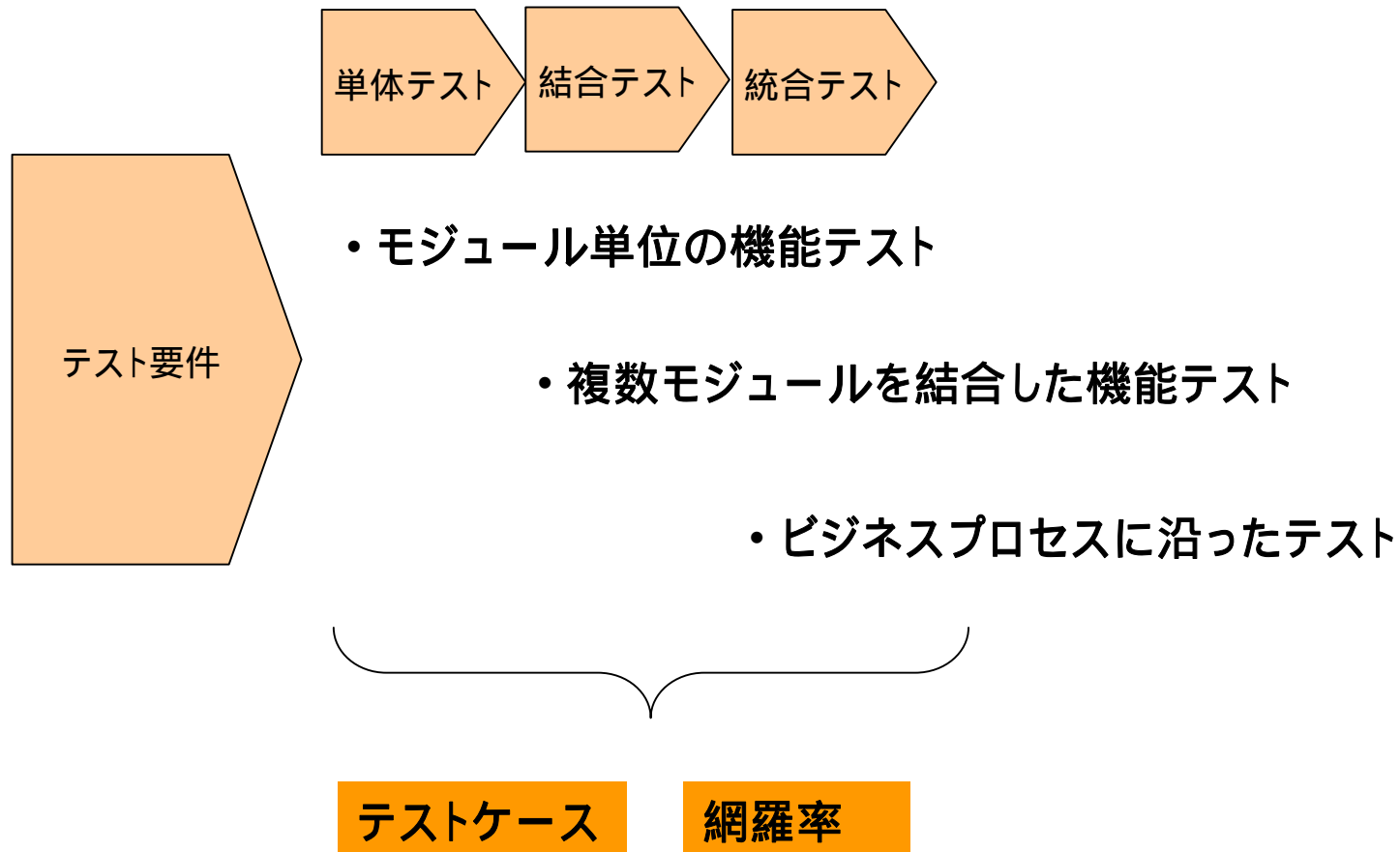
- テストの種類
  - 機能テスト
  - エラーテスト
  - 負荷テスト
  - 性能テスト
  - 回帰テスト
  - システムテスト



# テスト計画

- 設計をレビューしてテスト要件を決定する
- 優先度の高いものからテストケースを作る
- ビジネスの要因から優先度を決める
- ビジネス・リスク：作業量

# テストレベル



# 単体テスト・結合テスト

- 制御パステスト
- 境界値テスト
- 状態遷移パステスト
- 機能組み合わせ表

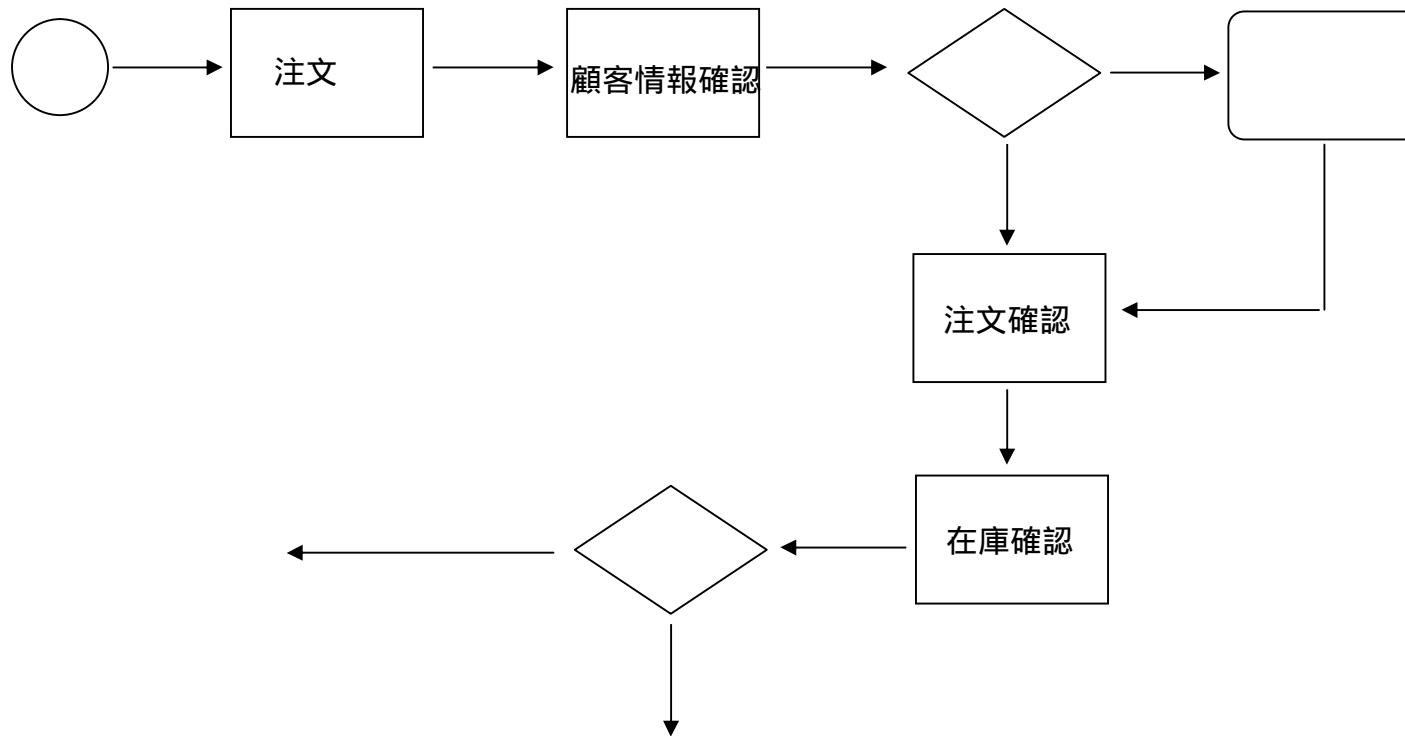
# 機能テスト

- 実装された各機能をテスト
- 各機能ごとにテストケースを作ってテスト
- テストステップの詳細な記述(テスト仕様書)
- 機能一覧
- 機能の組み合わせ

# システムテスト

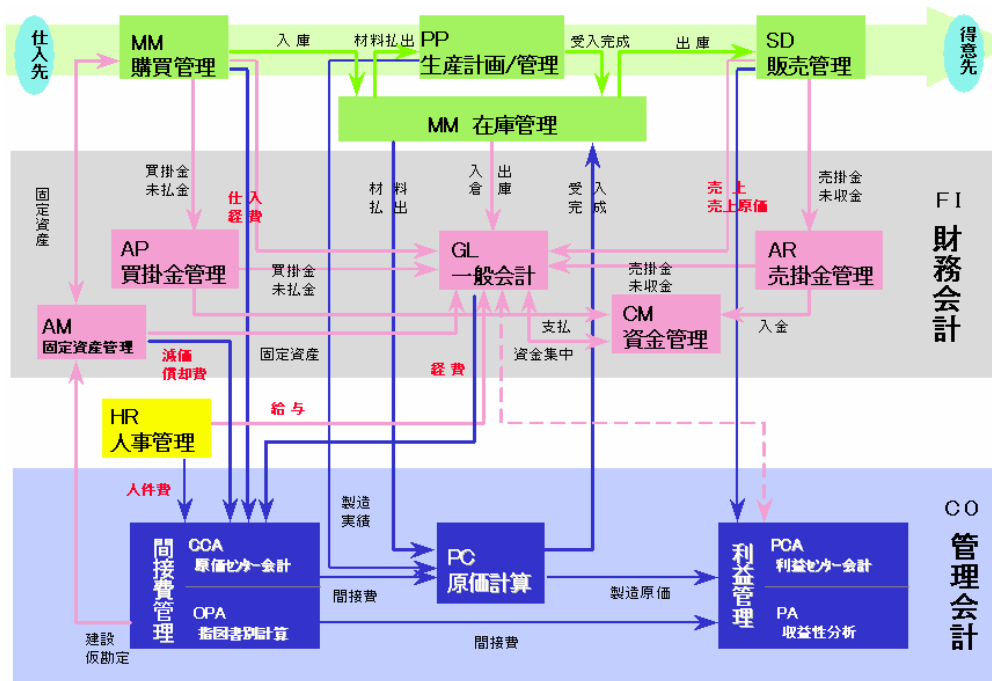
- システムテストでは、業務フローに基づいたテストが必要

新規申し込み



# 業務フロー(ビジネスプロセス)

- ビジネスプロセス(業務連携)をテストする



- ビジネスプロセスの数は膨大
- ビジネスプロセスを構成する単位をビジネスコンポーネントと呼ぶ
- それぞれのビジネスコンポーネントには、“入出力データ”が存在
- それぞれのビジネスプロセス毎に、“入出力データ”の利用パターンは異なる
- 例えば税率が変更された場合、全てのビジネスプロセスは正しいデータ連携を行えるか？

- ビジネスコンポーネントテストを繋ぎ合わせて、ビジネスプロセスのテストを作り上げる

# ビジネスプロセスのテスト

業務からビジネスコンポーネントを定義

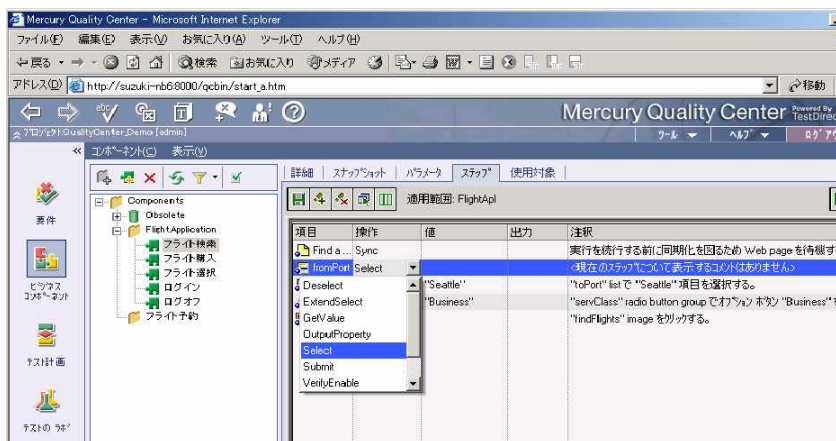
コンポーネントのフローを定義

コンポーネントのステップを定義

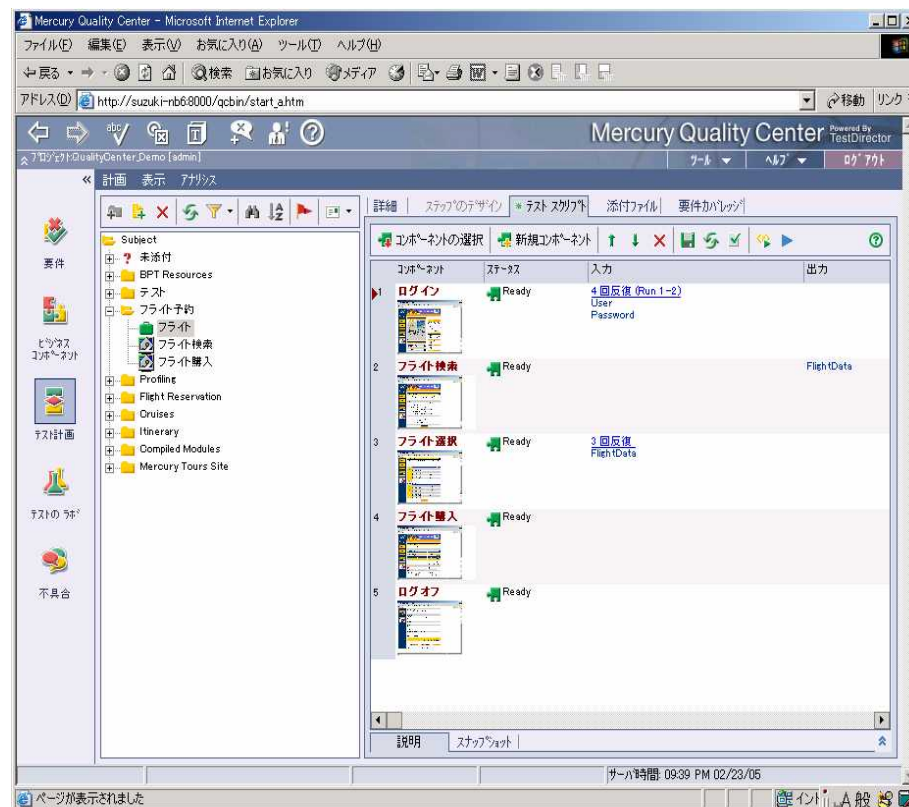
コンポーネントのデータを用意

必要なコンポーネントをつなぎ合わせる

# ビジネス・プロセス・テスト



ビジネスコンポーネントテストの作成  
画面上で、ステップを定義する

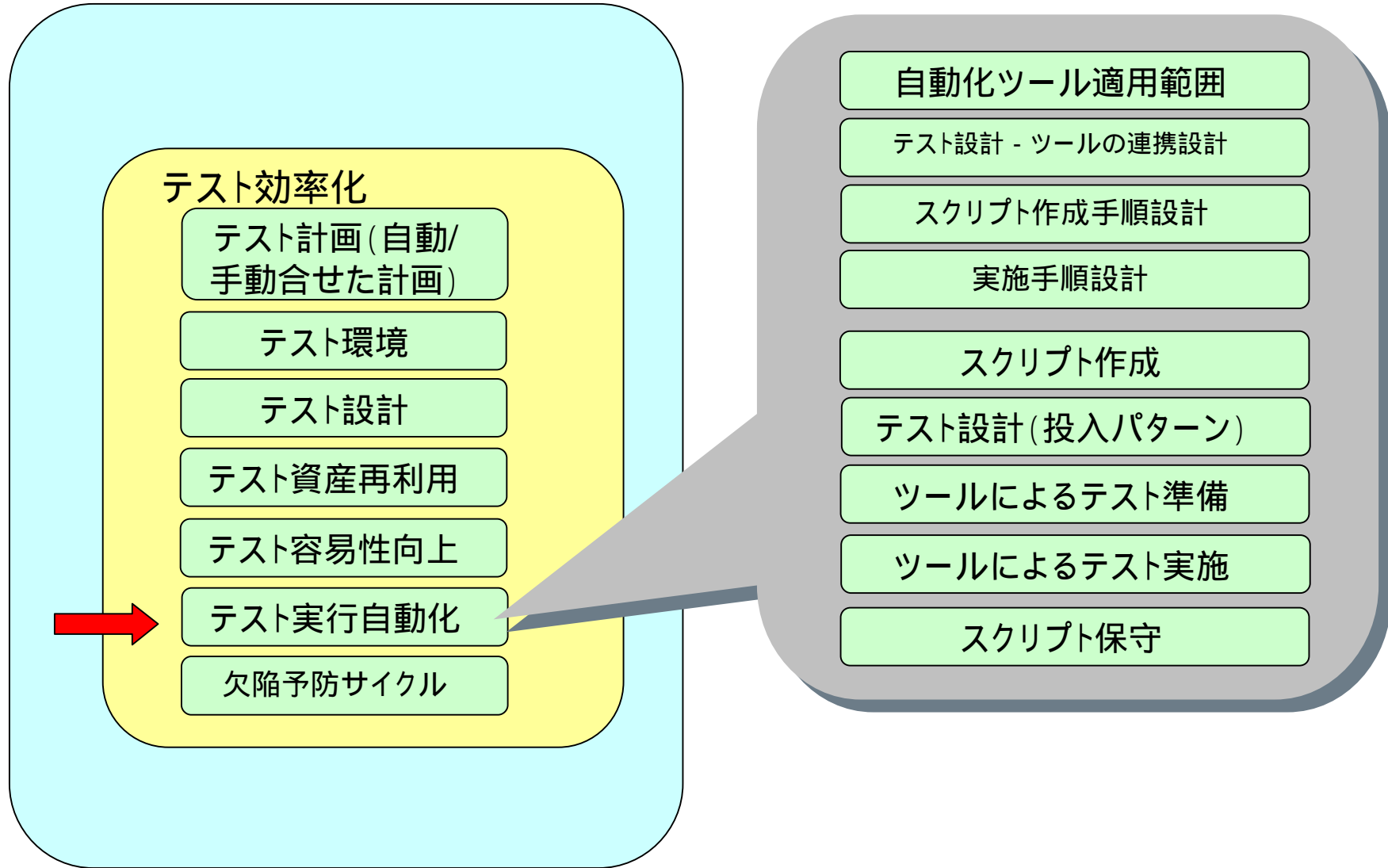


ビジネスプロセステストの作成  
画面上で、コンポーネントテストをつなぎ合わせる

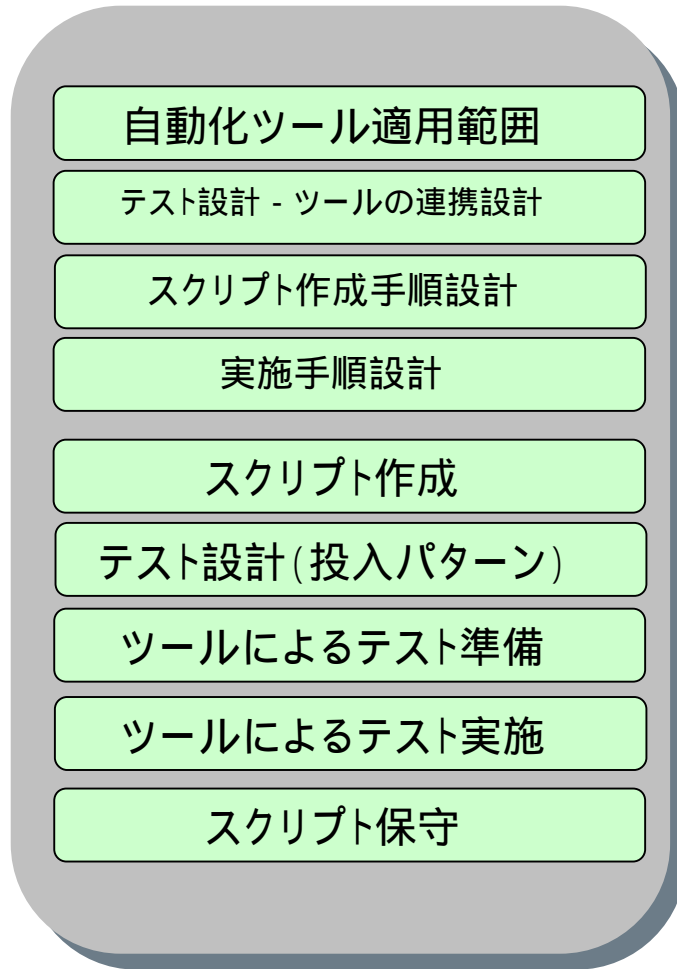
**MERCURY™**



# 対象範囲



# 対象範囲の状況



現行のテスト仕様書から、  
自動化スクリプトへの変換  
できますか

テスト設計担当

テスト実施担当

両者のノウハウを集めて  
設計する必要がある

# 自動化ツールと管理ツール活用

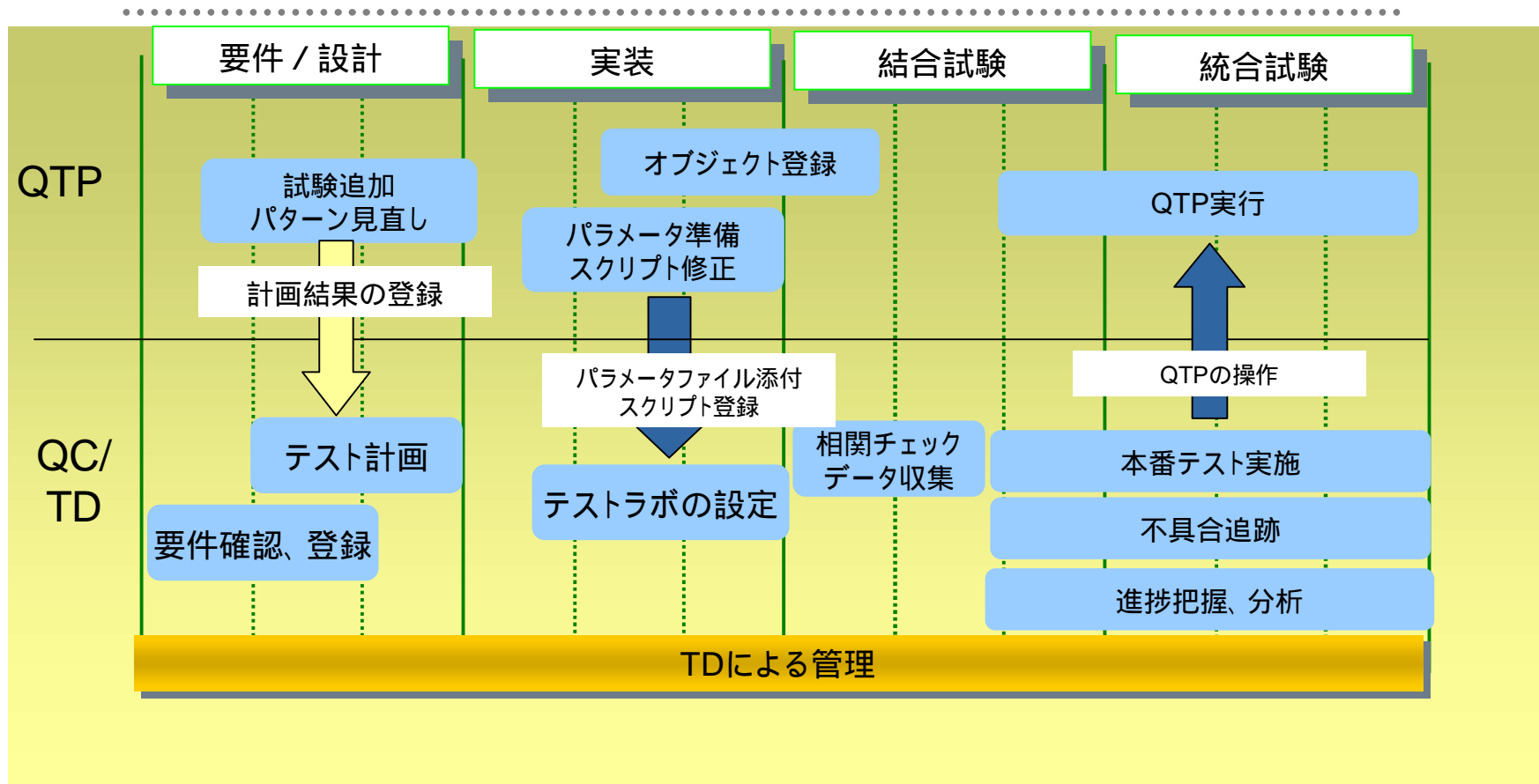
自動化ツールと合わせてテスト管理ツール導入として

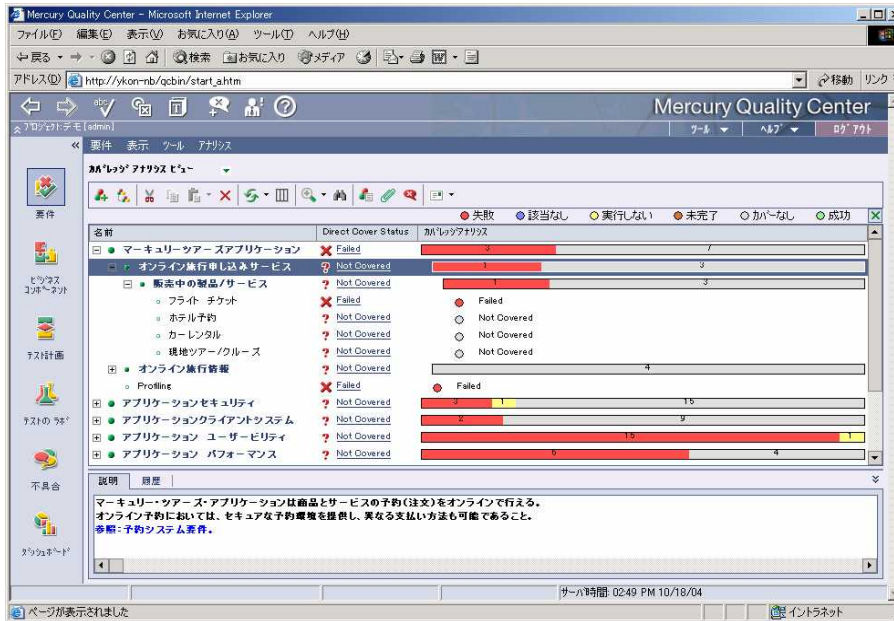
1. テスト実行自動化
2. テスト進捗管理(成功、失敗、未実行)
3. テスト仕様書、投入パターンの文書管理(構成管理)
  - (構成管理は変更履歴、バージョン、他成果物とのトレーサビリティの3つが常に簡単に調べられるようにする)
4. 不具合追跡

# 効果的活用

1. 不具合や進捗のレポート、グラフ機能活用 (傾向分析などテストプロセスの状況把握が可能)
2. テスト資産を一元管理する
  - 手動テストの管理
  - 各テストの管理
3. 不具合情報を開発へフィードバック
  - 開発時のナレッジベースとしてもらい不具合予防につなげる

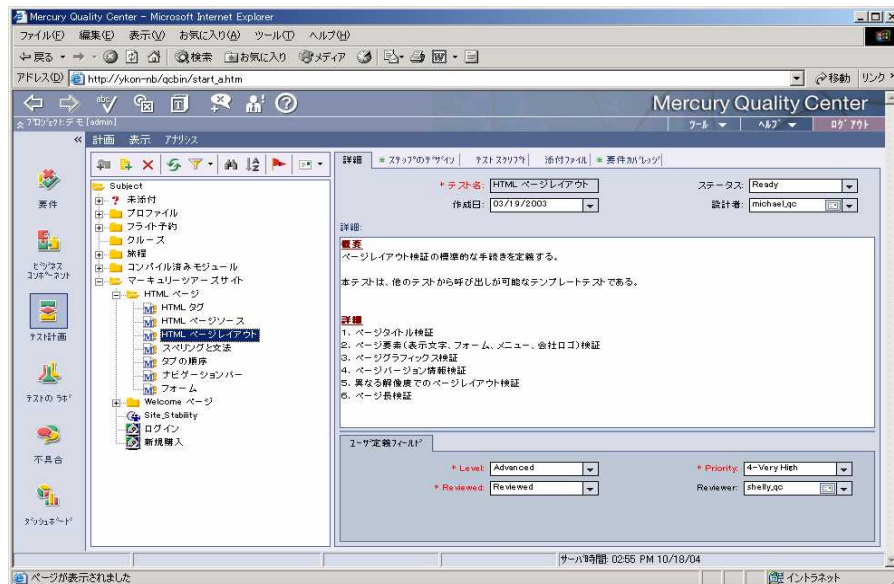
# テストプロセスフロー (例)





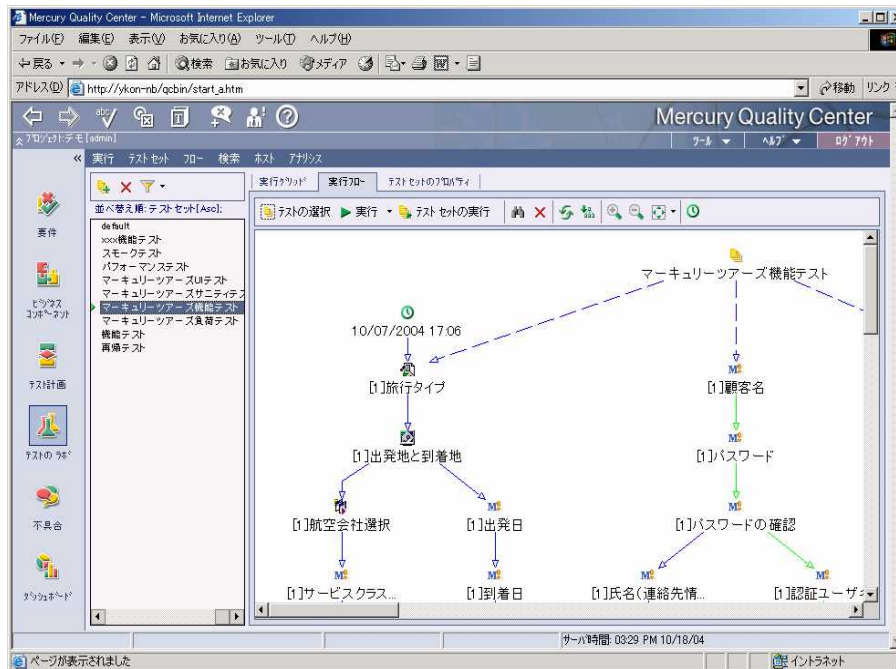
## 要件とのトレース

- 要件を一覧管理する。テスト計画時に各要件に当てはまるテストの関係を設定することで、テストが要件をどれだけ網羅しているかが一目でわかる。
- 各要件に関してどのような不具合が出ているのかも同様にわかる。



## テスト計画

- テスト全体の構成をツリー上に表現できるため、計画の漏れや間違いが少ない。
- 各テスト、手動テストを管理することで、テスト全体の計画が立てられる。

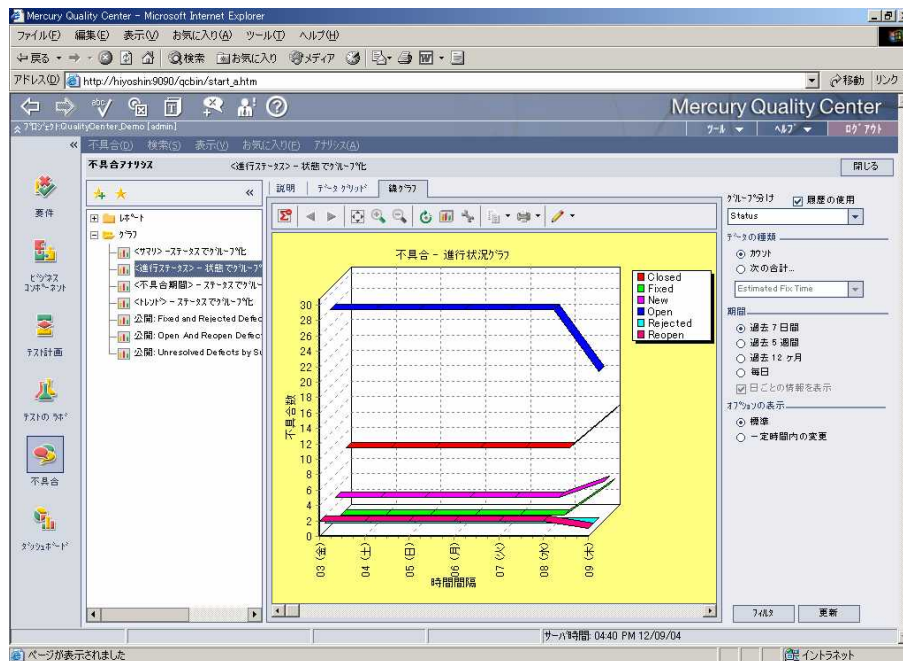


## テスト準備

- テスト仕様書などWord、EXCELファイルの文書管理可能。構成管理機能もあり。
- QTP実行時は添付したパラメータファイルを読み込んでテストを実施。
- 実行漏れを防ぐ

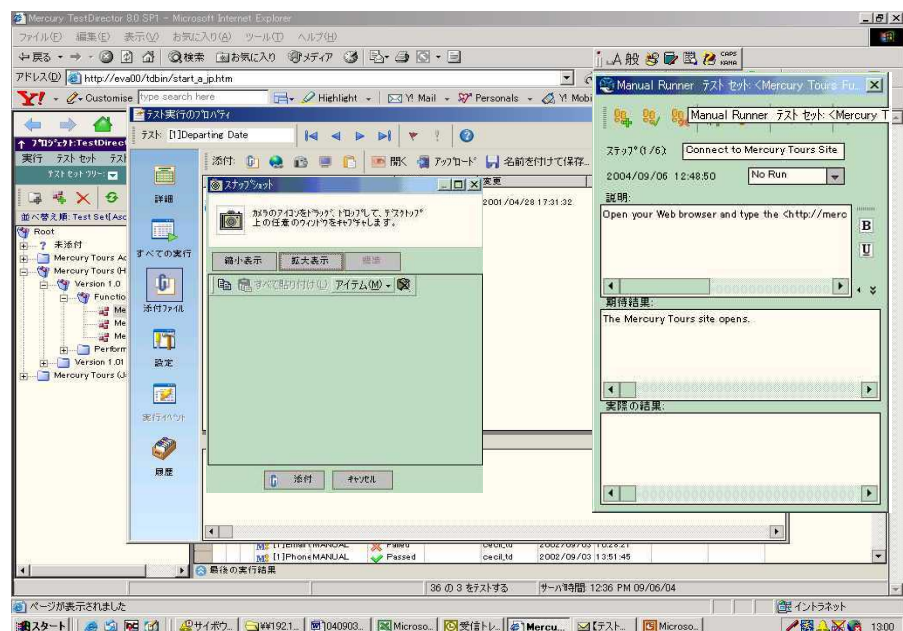
## 自動実行

- 自動テストの集中管理、スケジューラとなり、テスト実績を収集できる。
- 通常のマニュアルテストの実績収集でも便利な機能が多数あり。



## 進捗確認

- 実行済み、成功、失敗のテスト数が簡単に確認可能。
- その他修正済み、未済みの不具合数など状況把握が出来るグラフが多数あり。



## 手動テスト

- テスト結果の画面キャプチャ保存がワンクリックで可能。

不具合の詳細

不具合: 2 出発日と到着日が同じ場合、利用可能なフライトの一覧が表示されない

1 ページ 2 ページ

詳細  
説明  
添付ファイル  
履歴

\* Category: Defect  
\* プロジェクト: Mercury Tours (HT)  
\* Subject: フライトの検索  
\* 再現性: Y  
\* 不具合報告者: slice\_qc  
担当者: james\_qc  
ITG Request Id:

\* ステータス: Reopen  
\* 不具合バージョン: Version 1.0  
\* 不具合発生日: 09/02/2003  
Regression: N  
\* 重要度: 5-Urgent  
優先度: 5-Urgent

計画  
Planned Closing Version: Version 1.0  
修正予想時間: 4 (日)

実際  
Closed in Version:  
実際の修正時間:  
完了日:

実行レポート

OK キャンセル

## 不具合管理

- 不具合文書で管理される項目
  - 内容(再現手順 など)
  - 検知者
  - 優先度
  - 重要度
  - 修正担当者
  - 修正ステータス
  - 修正までの予定日数
  - 修正までの実日数 など
- 障害発生時の画面ショット、障害発生したコンピュータのシステム情報なども添付文書として保存

# テストプロセスの改善

管理ツールの効果的活用、また目的に対する効果を達成するためには、全体の考慮が必要

1. 効果的なテスト設計
  - テスト設計技法の使い方、テスト計画、ドキュメント類見直し
2. 不具合追跡～完了までの自動化、可視化
  - リリース前、リリース後の不具合追跡を確実に行うための仕組み
3. テストプロセスの見直し
  - モデルに基づいたテストプロセス改善
4. 不具合傾向のフィードバックと対策実施によるPDCA
  - 不具合除去から不具合予防へ

# テスト設計

テスト項目	テスト項目の条件	投入パターン	テスト結果の確認項目・確認内容
		番号	
ABC契約者の契約状態がホールド状態のときに強制終了ができること。	データ投入パターン参照	XXXXX	契約者IDを検索して、強制終了が選択できること。 登録したIDを検索したときに、契約状態に「ホールド」と表示されること。
	データ投入パターン参照	YYYYY	契約者IDを検索して、強制終了が選択できること。 登録したIDを検索したときに、契約状態に「ホールド」と表示されること。

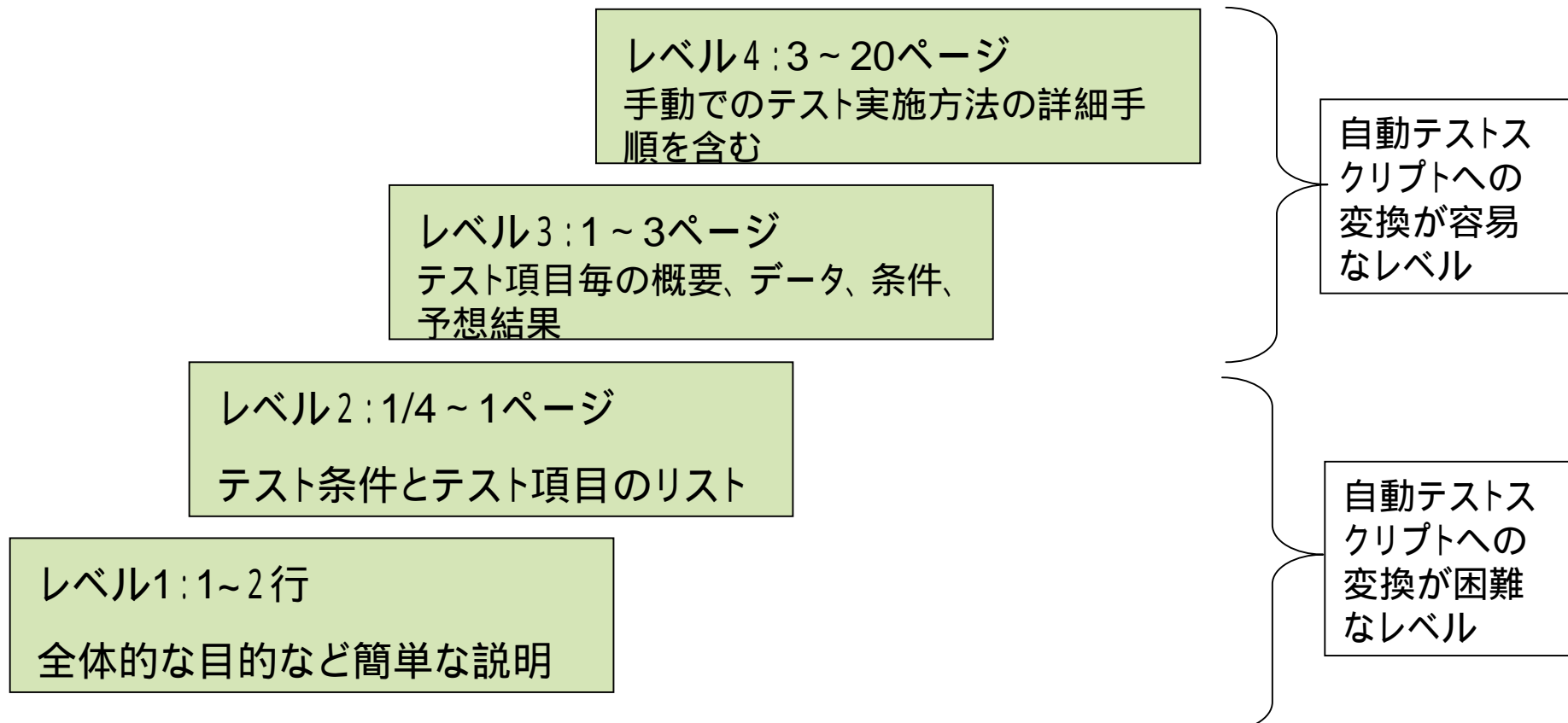
分類が投入パターン中心になっていないため、実施用に再変換が必要。

パターンの観点が(条件の違い)がわからない。

- 投入するデータは不具合を見つけやすいデータ(境界値)にすべきだが、どのようなデータを投入しているかわからない。
- 投入パターンの網羅が出来ているかわからない。(テストの漏れ、重複があるかもしれない)
- 本テストの目的(ロジックの不具合検出、コンポーネント組み合わせの不具合検出、機能の間違い検出、ユーザ観点での評価など)がわからない。

# テストケース設計の粒度レベル

- スクリプト作成(パラメータ設定)工数は、現状の手動テストケースの粒度に依存する。



RexBlack 「基本から学ぶテストプロセス管理」から

# テストケース粒度の例

## あるシステムの検索機能のテストケース

1. ABCシステムで検索機能が正常に動作することを確認
    - 1.1 検索パラメータを入力せずに検索ボタンをクリックする
    - 1.2 無効な検索キーを入力してみる
    - 1.3 一意の検索キー全部または一部を入力、一致するレコードが表示
      - 1.3.1 苗字全体、その他のパラメータはブランク
- 説明: 苗字のみの検索キーで検索DB内の一致するレコードを検索する  
初期条件: Win2KでABCシステムの検索画面が開いている状態  
入力: DBに存在する苗字入力後にEnter押下  
予想結果: 検索結果ウインドウに苗字と検索キーが一致するクライアントが表示

レベル4は、初期条件にてOS、対象システムバージョン、ウインドウ、カーソルの状態やデータベースID(データも決まったものが入っている)を定義して、入力作業の操作手順、入力文字、予想結果の動作手順、表示結果件数や文字まで明記するレベルをさしている。

A photograph of a conference room with a long, light-colored table and several chairs. One chair is highlighted in a dark red color. The Mercury logo is overlaid on the left side of the image.

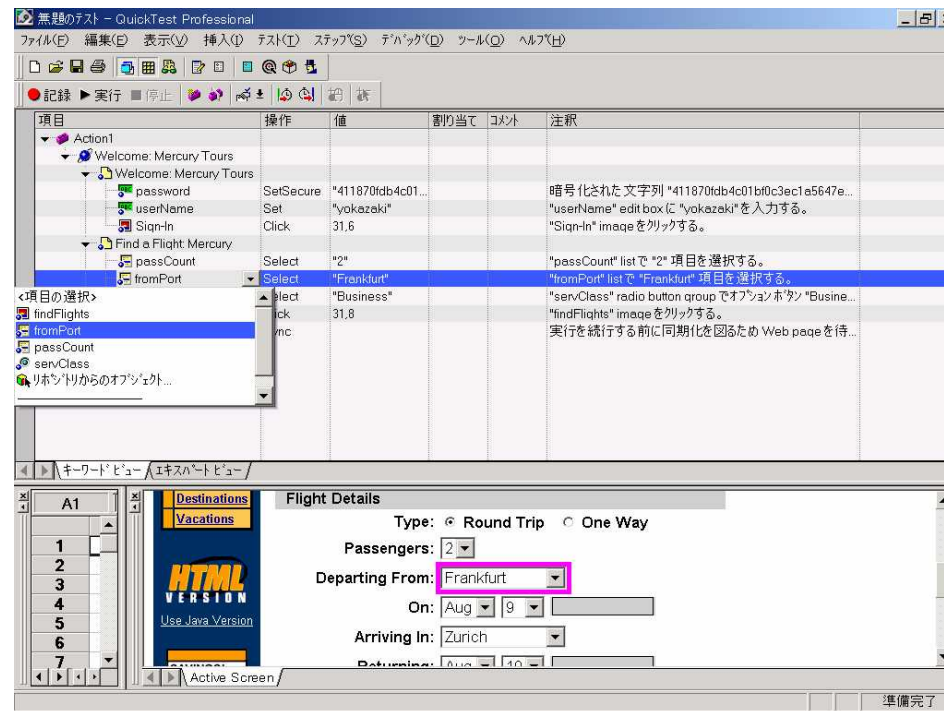
**MERCURY™**

**自動テスト**

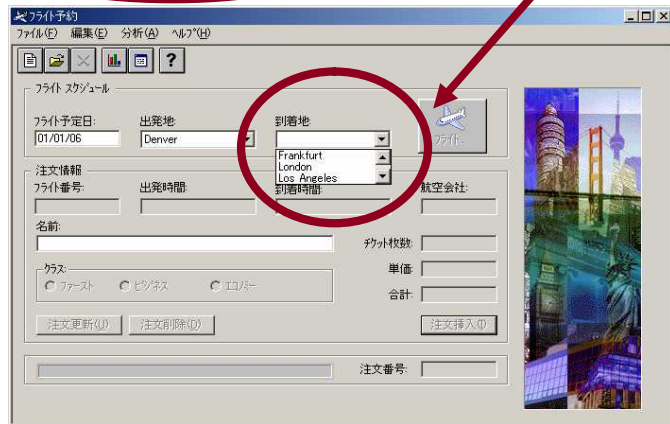
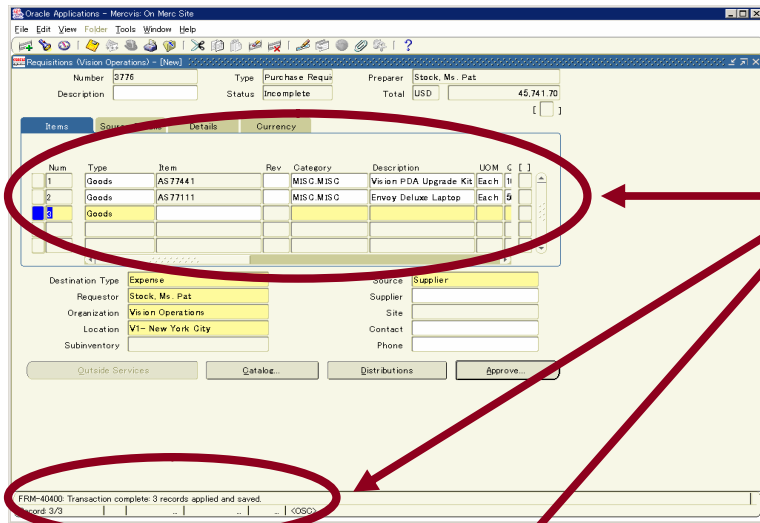
# 自動テストの成熟度

- 1 . レコード&リプレイ
- 2 . プログラミング
- 3 . フレームワーク
- 4 . キーワード

# 自動化の3ステップ



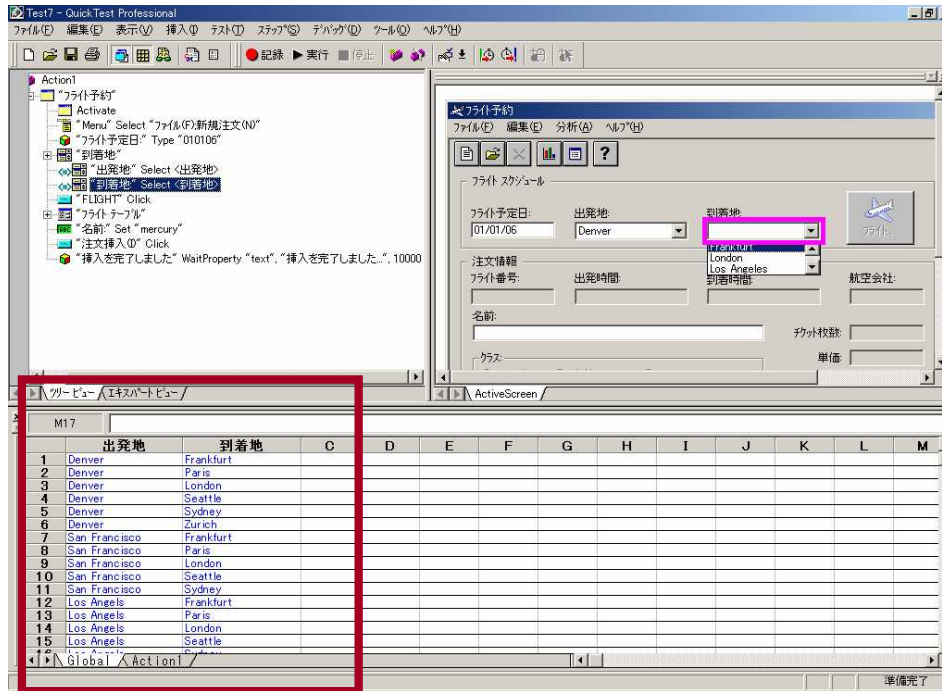
# スクリプトのカスタマイズ(検証項目)



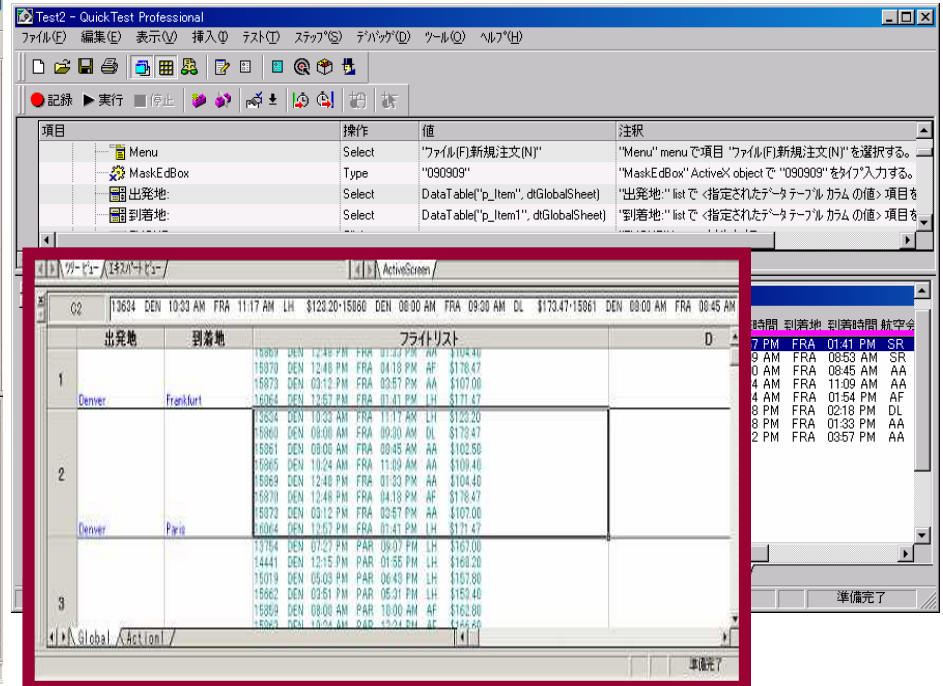
自動的に検証するためのチェックポイントのスクリプトに設定

- テーブル各セルの表示文字列
- ステータスバーの表示文字列
- リストボックスのコンテンツ
- オブジェクトのプロパティ値
- ビットマップ
- データベースの値
- .....

# パラメータ化

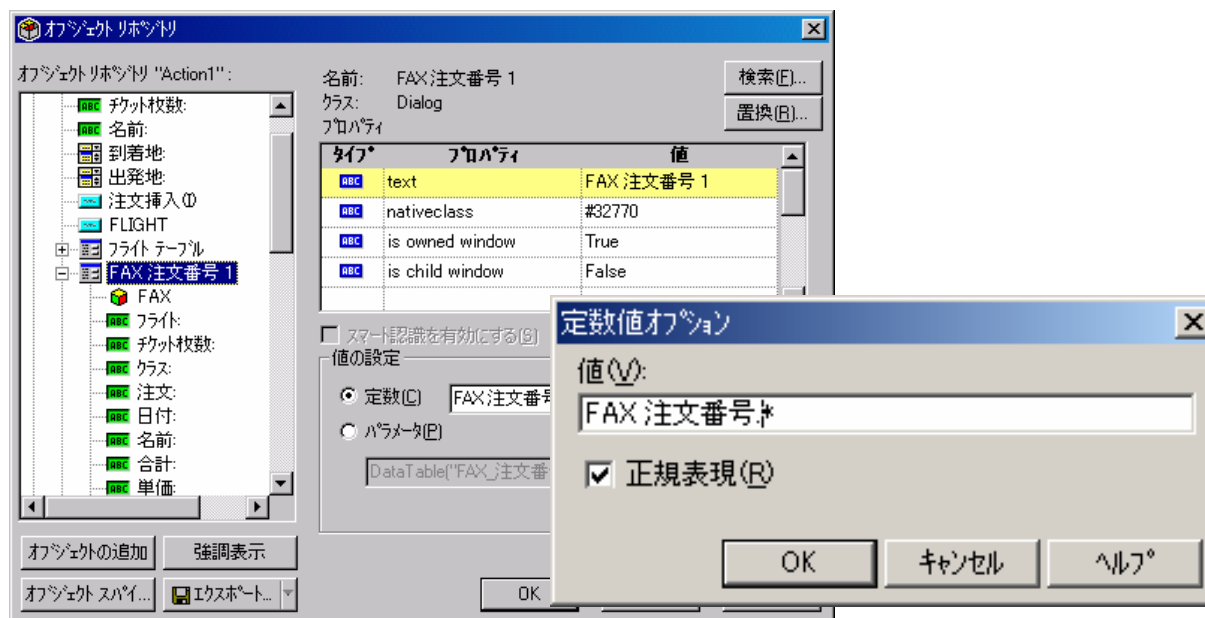


入力値のパラメータ化(テストケース)



期待値のパラメータ化

# オブジェクトリポジトリの編集



オブジェクト情報はリポジトリで一元的に管理される。  
属性が動的に変化する場合も、正規表現を使うなど  
することで保守効率が向上する。

# 例

- 基本方針

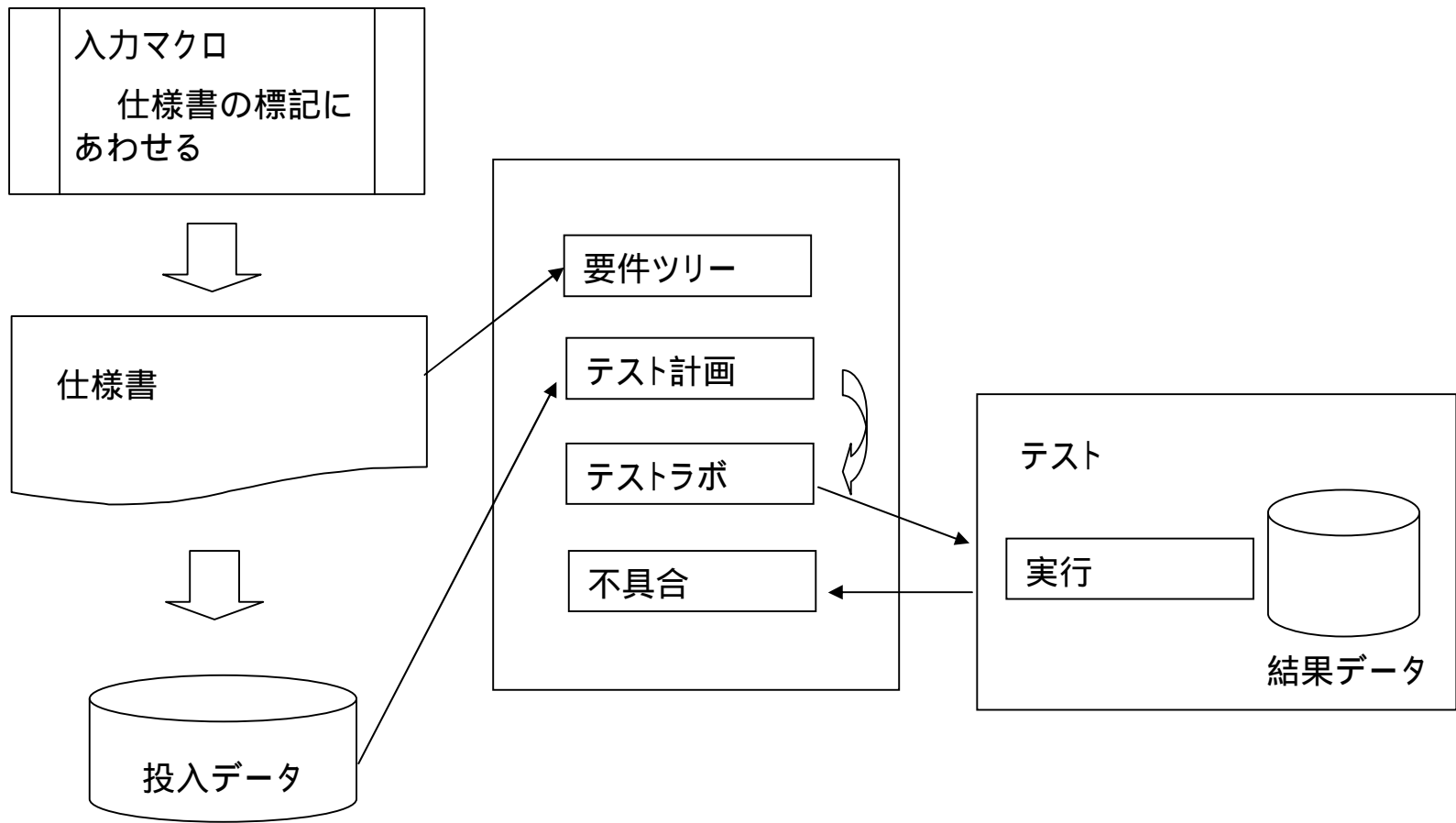
- 投入データ・パターンの管理は Excel 上で行う
- 投入データ・パターンでテストケースをあらわす

テンプレート	Global	Local
契約種別	種別選択	契約の種類を選択
顧客管理	顧客管理	
会計処理	科目選択	勘定科目の選択

Global							
Iteration	契約種別	顧客	住所	...	...	...	...
	1						

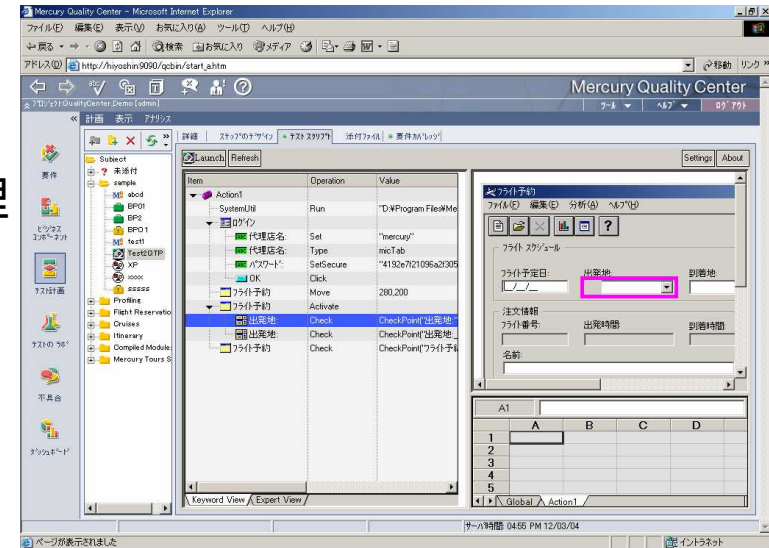
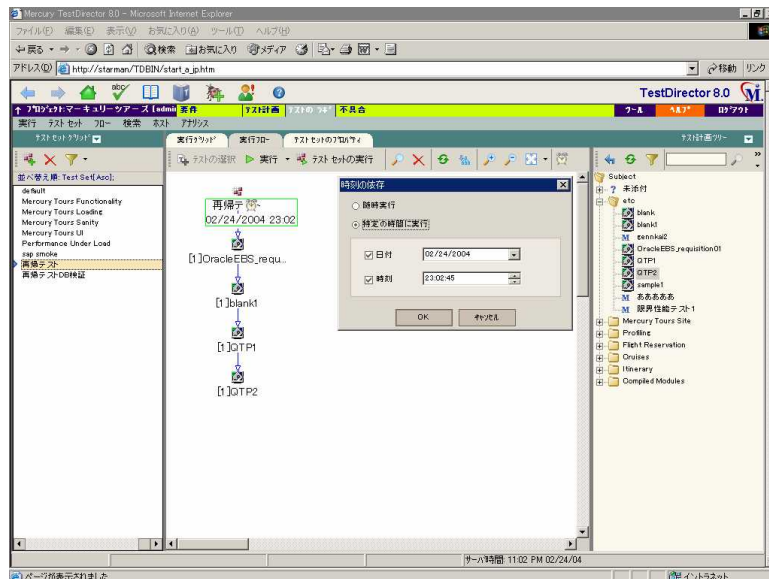
契約種別	
Iteration	種別選択
	1 xxxx
	2
	3

住所			
Iteration	住所1	住所2	番地
	1 東京都	港区西新橋	2
	2		
	3		



# 連携利用の効果

- スクリプトの一元管理
  - スクリプトは再利用可能なテスト資産として管理
- スクリプトのレビュー



- テストセットのスケジューリング
  - 日時、実行PCを指定し、複数のスクリプトを自動実行させる



# MERCURY™

## お問い合わせ

マーキュリー・インタラクティブ・ジャパン株式会社

〒105-0003 東京都港区西新橋2-38-5 西新橋MFビル7階

電話 03-5402-9300 FAX 03-5425-2288

ホームページ <http://www.mercury.co.jp> e-mail [info\\_jp@mercury.com](mailto:info_jp@mercury.com)