

テスト自動化の落とし穴

日本アイ・ビー・エム株式会社, ソフトウェア事業 Rational事業部
ワールド・ワイド・タイガーチーム 細川 宣啓



テスト現場の苦悩・・・“人だけではできない” “もったいない”作業の連続



実績の集計

膨大なテスト資産を管理
・ケース数、予定、実績
・etc...

可視性

テスト資産の管理

チーム、組織の単位でテストケースを把握し
再利用によるシナリオ定義

生産性

特定スキル依存

高いスキルが求められるテスト
・ソースコードレビュー

保守品質

何度も繰り返し

影響がないことを確認するため
何度も同じテストを実施

品質
生産性

業務テスト1

- 手順 1
- 手順 2
- ...
- 手順 x

- 初回のテスト
- 基盤FW修正
- PKG修正
- ...
- ?

長時間

単純作業を延々と続ける
・6時間トランザクションを処理し続けられるか？

品質
生産性

新たな検査の発生

日々新しく発見されるハッキングなどの
犯罪手口に対する影響検査

新たな問題 1

新たな問題 2

...

検査方法

検査方法

検査方法

脆弱性

単純作業

延々と機械的な入力を行う作業
・1000パターンのデータ入力
・10年分の日付処理
・etc...

生産性

入力パターン	入力金額	出力
1	100	1200
2	101	1300
...		
999999	1999998	ERK

手順 1

手順 2

...

手順 x

大人数

シナリオにしたがった大規模テスト
・10000人規模での同時アクセスできるか？

品質
生産性

役割によって悩みも変わる・・・“どうがんばるか？”

■ 質問：テスト作業で課題になっている点について

課題	合計	割合
テストがうまくいっているかどうか、品質に問題はないかが正確に把握できない	81	51%
テスト状況をレポートするための資料作りが忙しい	66	41%
仕様変更があった場合、どのテストをやり直すべきかすぐに判断できない	60	38%
負荷テストの環境が複雑で構築やテスト実行に工数がかかる	54	34%
テストが予測通りに終わらない	51	32%
最新のテスト計画の内容が分からない	37	23%
テストの前提となる背景(要求)を正確に把握できない	35	22%
仕様変更の内容がチームメンバーに正しく伝わらない	34	21%
色々なテスト手法(手動、ツール等)を利用しているためテスト結果をまとめるのに時間がかかる	24	15%
ほかのチームメンバーがどんな作業をしているか分からない	22	14%
その他	3	2%
Total	160	

テスト自動化以外の悩みも多すぎて、何を改善すべきか着手すべきか悩む現場が多い



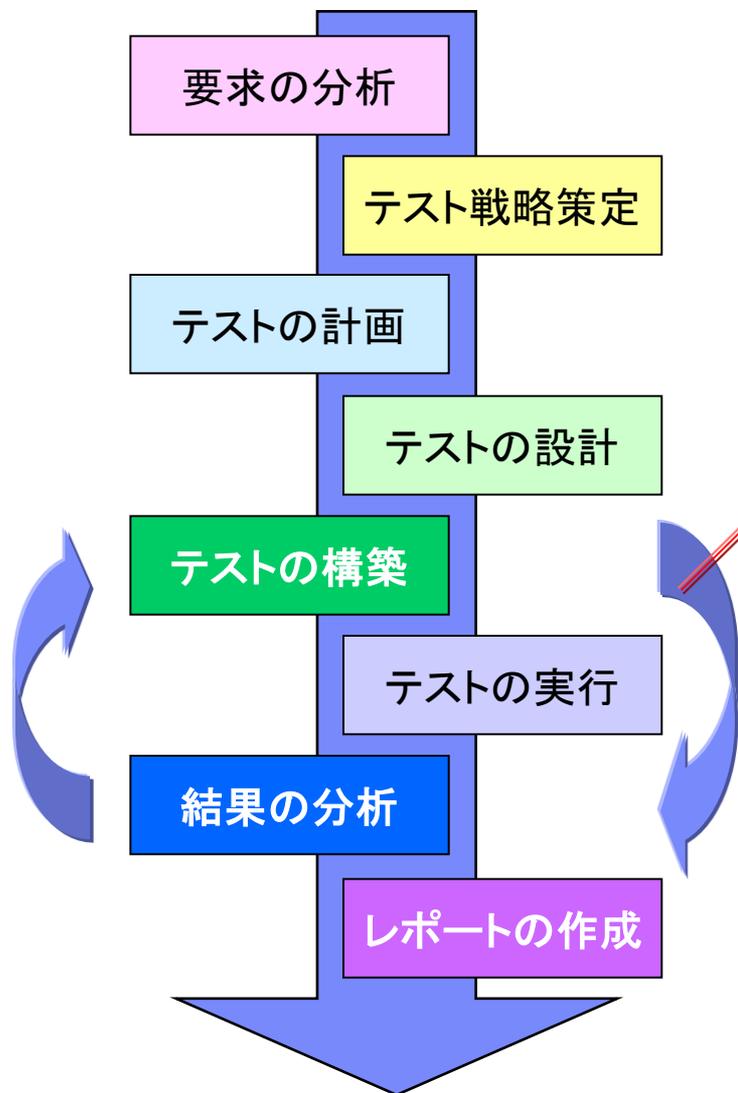
よく言われるテスト自動化ツール導入のメリット

- **作業の自動化（高速化/繰り返し実行）**
 - 人間が行うと手間と時間がかかり、不正確になりがちな作業をツールが担当
- **属人性の排除（正確性/監査性）**
 - ツールによる自動化でプロセスが構築されている場合、個々人の裁量が入る余地が少なくなることによって、無意識によるプロセスからの逸脱を防ぐ
- **時間投資効果による品質向上**
 - これは特にテストプロセスに言えることだが、ツールは夜間でも昼休みでも、何時間でも延々とテストや解析作業を行うので、時間がかかる作業を退社前や休日中に実行させ、これまでテストにかけていた時間を他の作業に割り当てることができる



本当か??

“テストの実行”の作業という視点からテストプロセス全体に目を移すと



繰り返し行うべきテスト作業が自動化されると、一定の効果は見込める

- 新しいビルドができたときのスモークテスト
- 障害が発見され修正後の機能テスト
- 環境移行に伴う無影響確認テスト

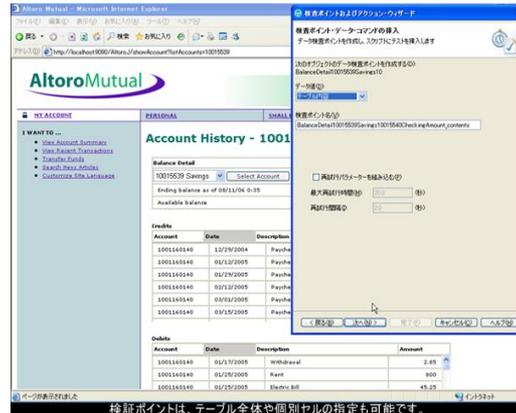
自動化できる部分は「作業」が主流。“ツールが使いにくい”と言われる場合の多くが「思考」に時間がかかる箇所

自動化テストのメリットと「前後の作業」について

「自動化ツールが高速で打鍵テストを実践するため、飛躍的な生産性と品質の向上が実現する」というシナリオは幻想です。安直なツール導入効果は以下のような「前後の付帯作業」によって相殺されます

打鍵テスト自動化

テストツールによる打鍵入力の省力化



落とし穴1: 思ったほど時間が短縮できない

よく耳にするテスト自動化導入後の声として最初にあげられる点は「思ったほど時間が短縮できない」です。その原因を見てみましょう。

- テストケース設計と生成が面倒
- さらにテストスクリプトに**スクリプトの管理が面倒**
- 仕様変更時にテストバリエーションを考慮するのが面倒
- 「意外と時間がかかる」「効果不明」

①テスト設計
 テストケース100%は本当か？

②データ作成
 データ作成工数はツール省力効果より少ない？

打鍵テスト自動化
 テストツールによる打鍵入力の省力化

③結果分析
 結果の分析は人手で行うべき。省力化“難”

④再検査
 欠陥除去後に再検査しているか？

⑤目的性
 テスト計画や目的に合致した結果が得られたか？

⑥テスト管理
 進捗監視・測定・報告等の管理する作業負荷大



コツ: 「ケース設計の困難さ」と「テストツールの使いやすさ・省力化」は区別して考える

1. テストケース設計とテストツールの自動化を分けてそれぞれを最適化する
2. 「設計時点で試験性を考慮する」+「テスト設計の導入によって負担を軽減」
3. テストケースの生成に関してサポートするツールは導入効果大

落とし穴2: テスト管理がないから、本当に効果を出せているか不明

もうひとつポイントがあります。自動化による導入メリットが、本当に得られているかどうかを測定して管理する仕組みがないため、何となく自動化していると...という現象が多く見られます。

- テスト期間短縮と、テストの質のバランスを考慮する
- 重点テスト箇所はどこか？
- テストを省略できる箇所はどこか？
- テスト管理プロセス・仕組みはうまく回っているか？

①入力データ
 テストケース100%は本当か？

②データ作成
 データ作成工数はツール省力効果より少ない？



③結果分析
 結果の分析は人手で行うべき。省力化“難”

④再検査
 欠陥除去後に再検査しているか？

⑤合目的性
 テスト計画や目的に合致した結果が得られたか？

⑥ **テスト管理** 進捗監視・測定・報告等の管理する作業負担大

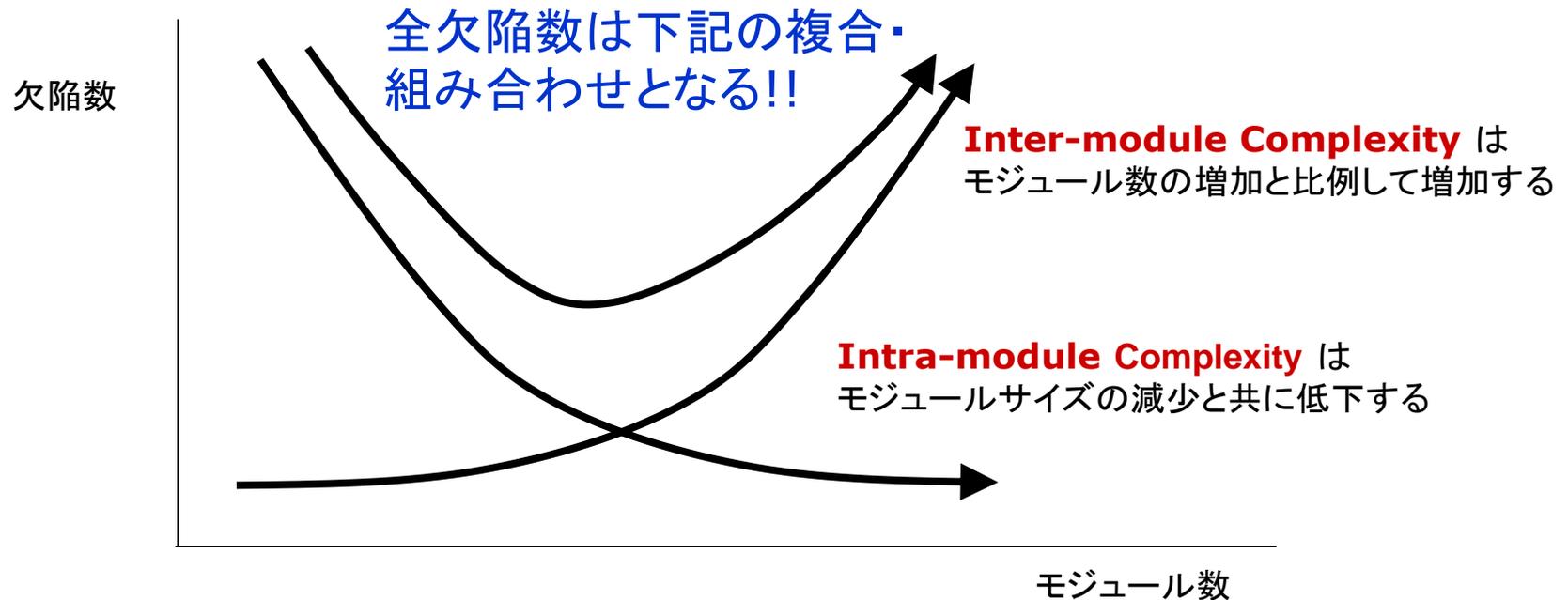


コツ: 闇雲にスクリプトを書いて全範囲を均等にテストする「戦略なし」なテストを避ける

1. テスト管理プロセス・仕組みの導入は、短期的な打鍵自動化と合わせると効果が倍増
2. テスト管理に特效薬ない。古典エンジニアリング技術の地道な実践が必要

コードメトリクスを測りテストの難所を見つける: 複雑度利用

- 複雑度には2つの考え方があります
- **Intra-module complexity** (例: Thomas McCabe's Cyclomatic Complexity)
 - 1モジュール内の複雑度
- **Inter-module complexity** (例: Henry-Kafura's Inter-Module Complexity)
 - 他モジュールとのインタラクションの複雑度



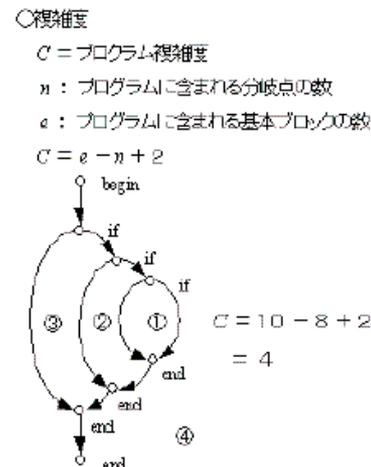
コツ: 測定複雑度の高い箇所 = テスト工数が大きい箇所として計画

古典エンジニアリング技術 + Rational Quality Manager (RQM) によるテスト管理 ～「要求～テスト～障害」までの一連のテストサイクルの管理～

- テストに関する情報を一元化し、チームのコミュニケーションを促進
- 品質に関わる「要求、テスト、障害」までのテストサイクル全体を管理
- ワークアイテム(タスク)を作成し優先度や作業を明確化して追跡可能

コツ1) 古典エンジニアリング技術による難所特定

- あらかじめ、テストの難所を設計書/コードから特定
 例) 循環的複雑度・IMCによるテスト工数が多い箇所特定



コツ2) ツールによるテスト管理と追跡

- テスト計画を中心に テスト情報を一元管理

- テスト要件
- テスト環境
- テストケース
- テスト・スクリプト(手動・自動)
- テスト実行結果
- テストで発見された障害

- 手動テストの文書作成と実行の効率化
 一貫性をもった文書作成や実行の管理

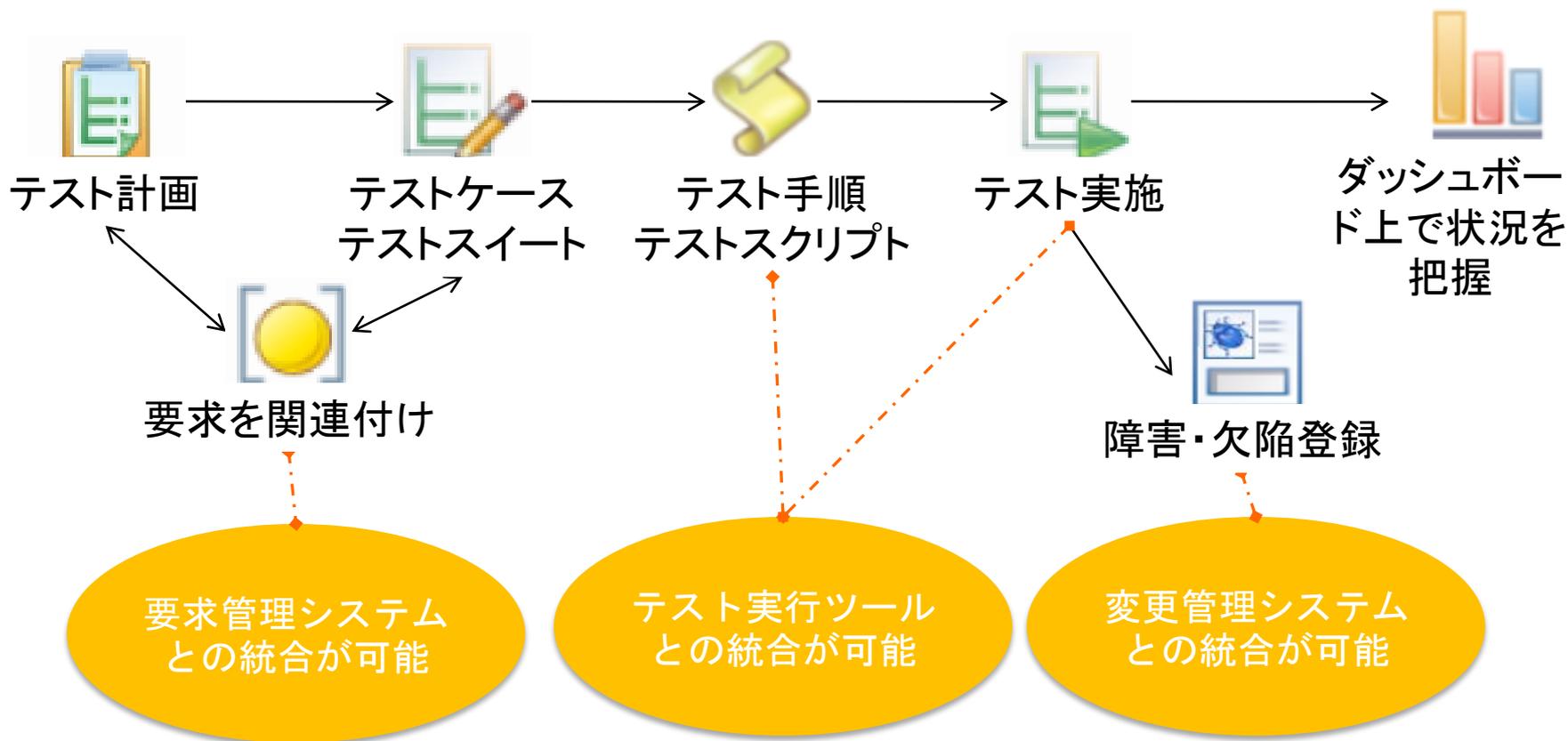
- ダッシュボード・レポート
 全テスト情報を様々な観点で分析





テスト管理ツールの定量効果試算例:

- **\$241,000/障害** ※
リリース後に発生する障害1件あたりの対応コスト
- 再利用により**平均696時間の工数削減** ※
- 重複障害を防ぐことにより**1755時間の工数削減**※



※Source: GBS Test Practices study, 2005-2008, over 846 projects

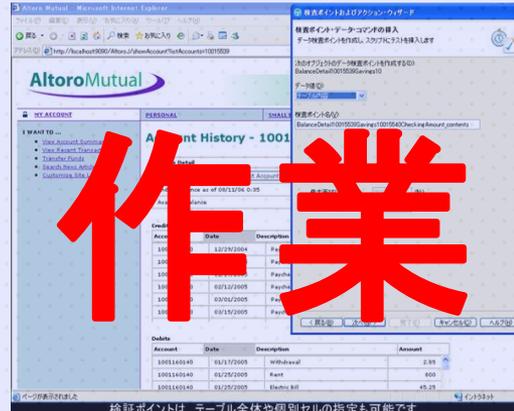
落とし穴3: 作業と思考を区別して考えられない

- ここまでにあげた2つの落とし穴は、総じて「作業と思考の分離」ができていないことに起因します。
- 自動化ツールの前後(前=ケース設計、後=バグ分析)が重要。いずれも「思考」であり、全体最適が必要。
- テストを思考できる専門家(テスト、欠陥、品質の技術者)育成が急務です。専門家を助けるツール導入と考えるべきです

①入力データ
テストケース100%は
本当か？

打鍵テスト自動化

テストツールによる打鍵入力の省力化



③結果分析

結果の分析は人手で行うべき。省力化“難”

④再検査

欠陥除去後に
再検査しているのか？

5.目的性
テスト計画や目的に
合致した結果が
得られたか？



思考

作業

思考

⑥ テスト管理

進捗監視・測定・報告等の管理する作業負荷大

コツ: テスト専門家を育て、全体最適のために“思考”と“作業”の分離を行う



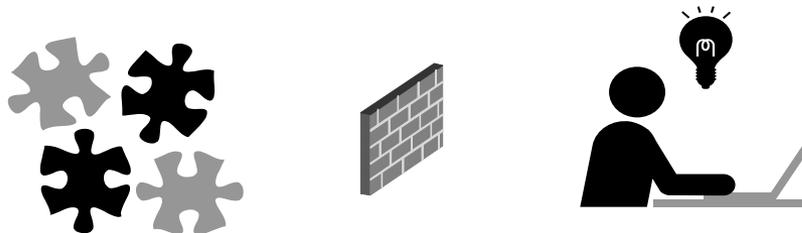
テストプロセスの自動化の考慮点

- 出来る限りシンプルに考える = **KISS** (Keep It Stupid Simple!!)
- ツールやアプローチを小さいセットにまとめる(分割統治が基本)
- 既存のソリューションを探す + 再利用できる範囲を明確化する
- 出来る限り標準に従う
- 試験性 (=テストのし易さ) を考慮した設計を実施する
 - 「事前に考慮されていない試験性を、後から得ることはできない」

まとめ：成功の鍵



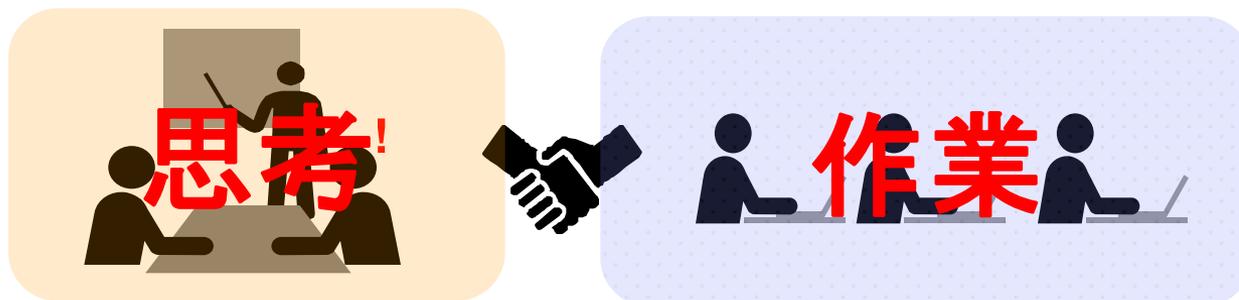
コツ: 「ケース設計の困難さ」と「テストツールの使いやすさ・省力化」は区別して考えよう



コツ: 闇雲にスクリプトを書いて全範囲を均等にテストする「戦略なし」なテストを避けよう



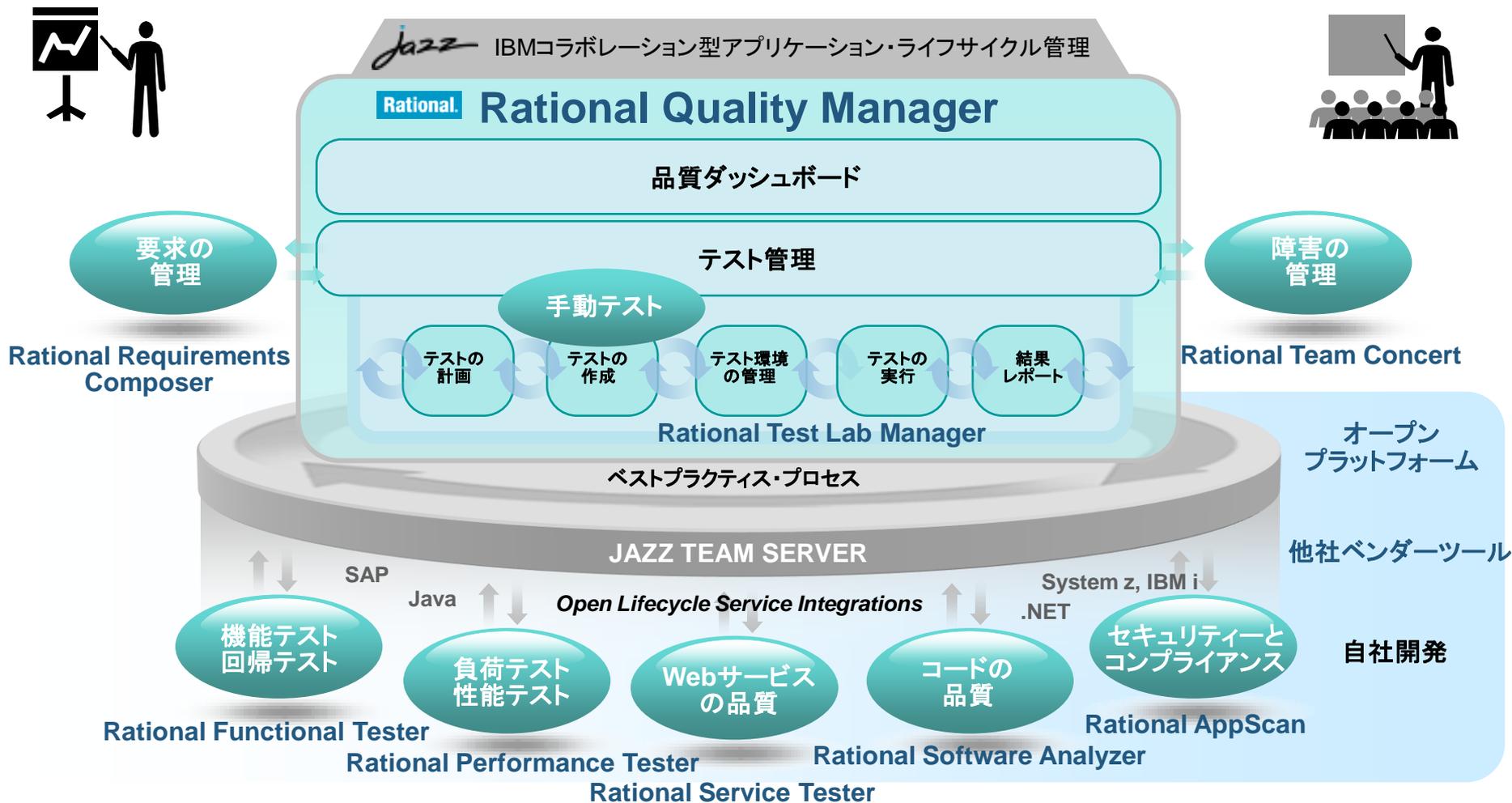
コツ: テスト専門家を育て、全体最適のために“思考”と“作業”の分離を行おう



IBM Rational テスト・ソリューションの全体像

あらゆる種類のテストとテスト・プラットフォームを

ライフサイクル全体にわたって品質とテスト管理することがプロジェクトの成功へ



<http://ibm.com/software/jp/rational/>



Learn more at:

- [IBM Rational software](#)
- [IBM Rational Software Delivery Platform](#)
- [Process and portfolio management](#)
- [Change and release management](#)
- [Quality management](#)
- [Architecture management](#)
- [Rational trial downloads](#)
- [Leading Innovation Web site](#)
- [developerWorks Rational](#)
- [IBM Rational TV](#)
- [IBM Business Partners](#)
- [IBM Rational Case Studies](#)

© Copyright IBM Corporation 2011. All rights reserved. The information contained in these materials is provided for informational purposes only, and is provided AS IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, these materials. Nothing contained in these materials is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software. References in these materials to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. Product release dates and/or capabilities referenced in these materials may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way. IBM, the IBM logo, Rational, the Rational logo, and other IBM products and services are trademarks of the International Business Machines Corporation, in the United States, other countries or both. Other company, product, or service names may be trademarks or service marks of others.