



開発を楽しむ ソフトウェアテスト

～ソフトウェアテストは開発の「達成感」に貢献する～

JaSST'13 Tohoku
31 May 2013 13:10 ~14:50

自己紹介



湯本 剛

Consultant at HP Software Japan

ソフトウェアテスト現場で10年ほど経験を積んだ後、テストプロセス改善のコンサルティング、教育に約7年ほど従事

2010年8月よりHP Softwareのテストツール導入支援コンサルタントに転職し、主にテストケース管理ツール、キャプチャリプレイツールの導入支援に従事

外部活動

NPO法人ASTER 理事

- JSTQB 技術委員
 - ISTQB CTAL WPメンバー
 - ソフトウェアテストシンポジウム(JaSST)東京実行委員
- ISO/IEC JTC1 SC7 WG26 エキスパート
日科技連SQiPステアリング委員

書籍、Web記事や雑誌への執筆や翻訳

得意分野

テストマネジメント、テスト分析



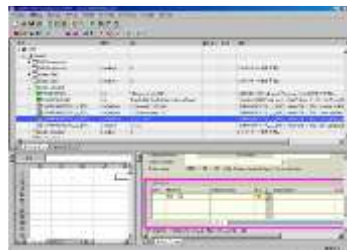
HPのアプリケーション品質管理ツール群

HPのテストツール全体像

キャプチャリプレイツール

Unified Functional Testing(UFT)

- ①アプリケーション操作を記録(スクリプト化)
- ②スクリプトを再生し、自動検証
データ駆動、キーワード駆動テスト



負荷テストツール

PerformanceCenter &LoadRunner (with Diagnostics)

- ①アプリケーション操作を記録(スクリプト化)
- ②別PC上で多重実行
- ③パフォーマンスデータの収集
- ④結果分析



手動テスト支援ツール

Sprinter



スタブツール

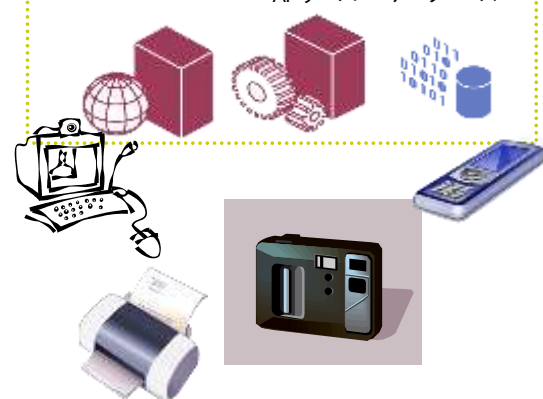
Service Virtualization

外部サービスをシミュレーションして、実際のサービス稼動を想定したテスト環境を作成



さまざまなテスト対象

Webサーバ APサーバ データベース



セキュリティテストツール

WebInspect, Fortify

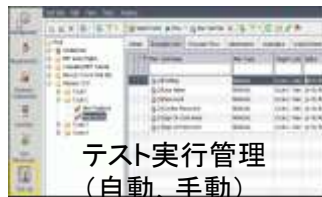
Webアプリケーションに対して様々なアタックをかけ、脆弱性を検証する



テストマネジメントツール

QualityCenter

リリース管理、要件管理
テスト条件、テストケースの管理
不具合管理
テスト実行管理(スケジューラ)
テスト結果収集、進捗参照



テスト実行管理
(自動、手動)



要件カバレッジ



品質リスク分析

アジェンダ

「楽しい」ソフトウェア開発について考えた

- 「達成感」を感じながら仕事ができるか？

ソフトウェアテストは開発にどう貢献しているか

- ソフトウェアテストの役割と価値

ソフトウェアテストの全体像

- ソフトウェアテストに必要な活動を知る＝何をよくしていけばよいかを知る

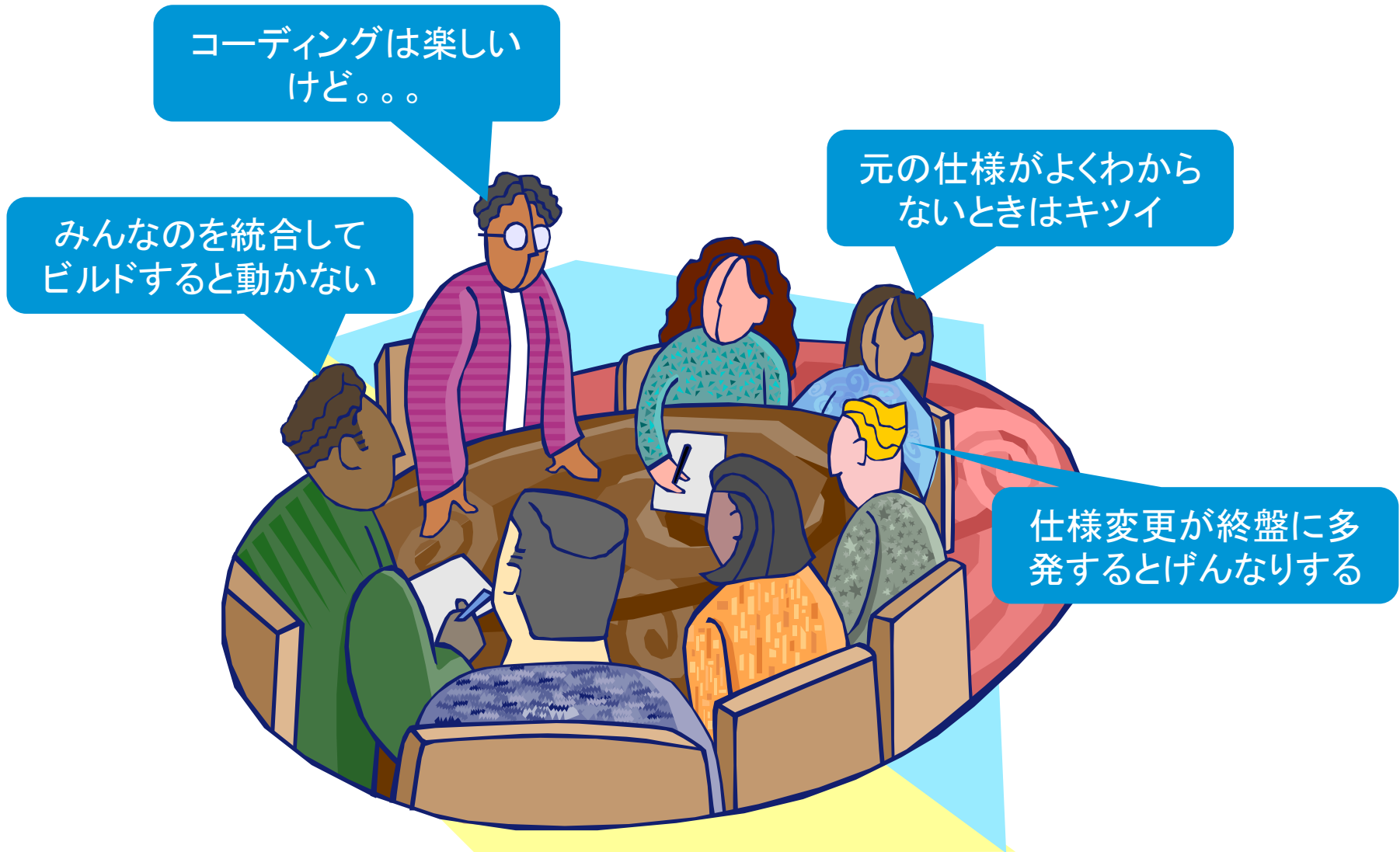
ソフトウェアテストにて押さえておくべきこと

- テスト設計
- テスト分析
- テストツール

「楽しい」ソフトウェア開発について考えた

「達成感」を感じながら仕事ができるか？

ソフトウェア開発は楽しいですか？



ソフトウェアテストは楽しいですか？

もっと効率的
にテストしたい

テストを書いてから実
装しないと気持ち悪い

バグが出続ける夢を見
て夜も寝れません

これからテストってなる
と憂鬱になる

バグを見つけると言わ
れても見つかりません

たとえばこんなこと言われますか？

テスト「あるある」その1



akiyama924 @akiyama924

5月8日

「気分が落ち込んだ時には、『できごと』と『考えたこと』と『気分』に分けて考えてみよう。『できごと』が『気分』を決めていると思いがちだけど、『考えたこと』が『気分』を決めていないかな？」

by 田中 友枝

開く

できごと	考え	気分
「とにかく全部テストしろ！残された時間でできる限りテストして品質を上げろ！」と言われる		
「とりあえずテストしておいて！ちゃんと動けばよいから」って言われる		
「バグ密度が〇〇だから、そこに達成するまでバグを見つけなさい！」と言われる		
「信頼度成長曲線の予測だと後〇件はバグが潜在しているからまだテストしなさい」と言われる		

たとえばこんなことがありますか？

テスト「あるある」その2

できごと	考え	気分
想定外のバグが出て困る		
バグが出ても対応工数がないので困る		
バグがでると怒られる		
バグが出なくて怒られる		

「楽しい」ってどういうことか？

「楽しい」の意味

- 1 満ち足りていて、愉快的な気持ちである
- 2 富裕である。金持ちである
- 3 作物の出来が豊かである

(goo 辞書: <http://dictionary.goo.ne.jp/leaf/jn2/138112/m0u/>)



どんなとき楽しいか？

- お酒を飲んでる
- デートしている
- 音楽をしている(i.e.趣味に没頭している)
- ゲームをしている



ゲーミフィケーション

ゲームデザイン手法や仕組みを用いて問題の解決やユーザー契約などを獲得すること

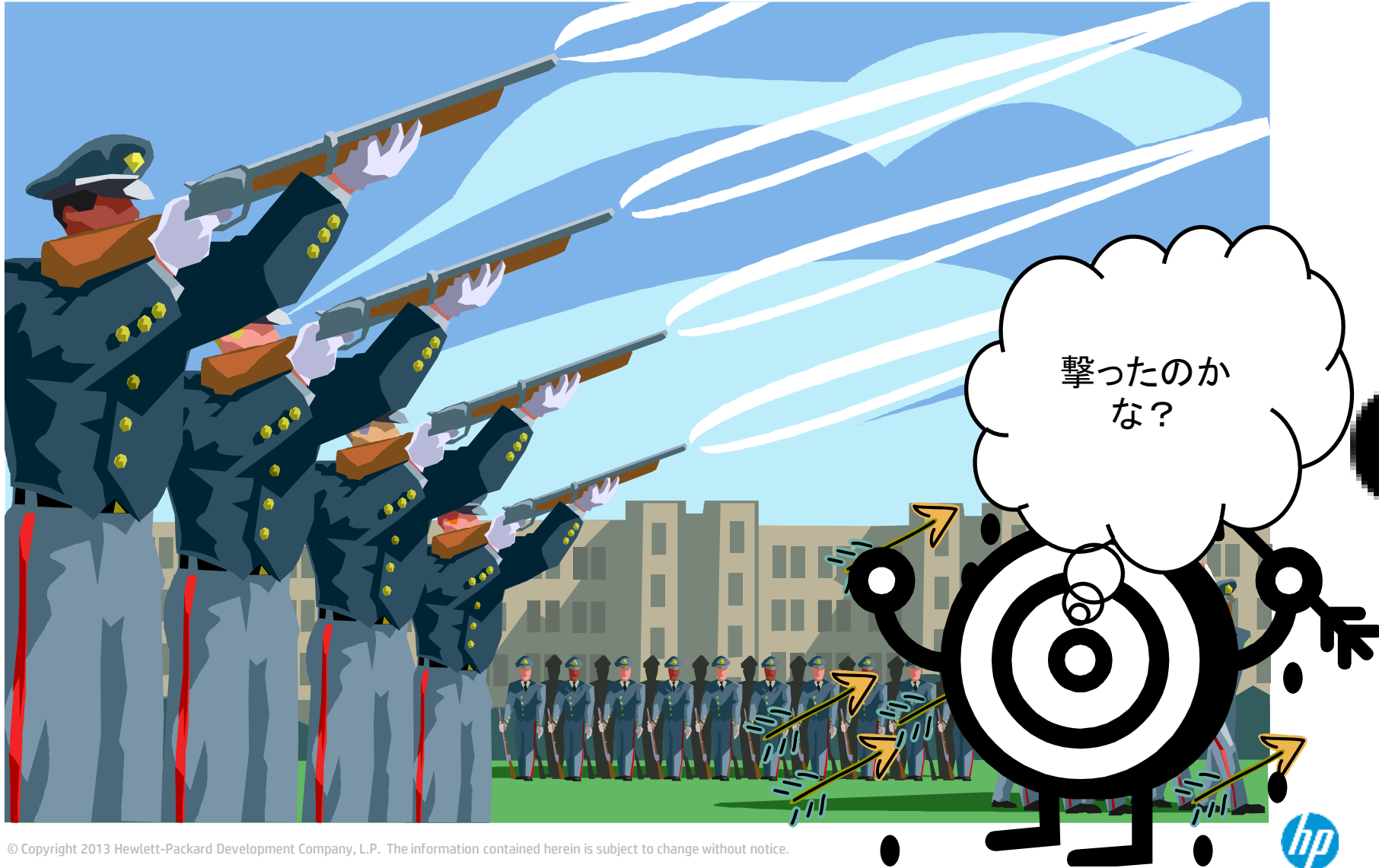
- 単にチャラチャラ面白いだけでは、満足しない...はず(だよね！)
 - 何かを達成するための仕掛け...達成感が物事を楽しくする

<楽しい時の状態>

- 自分の想い/気持ちが通じる/繋がる
- 没頭する
- 時間を忘れる



これは楽しいですか？



「物事を楽しくする仕掛け」がソフトウェア開発にあるか？

ゲーミフィケーションの解説から

「切迫感のある楽観性」(Urgent Optimism)

- 合理的に「成功できる」と信じられると障壁に立ち向かう情熱がわき、自ら「すぐに」行動を起こしたくなる

「ソーシャル構造」(Social Fabric)

- ユーザー同士が共に時間を過ごし、互いを認め合う、そういう何かを一緒にして行くことで、信頼し、快く一緒に何かをしようと思える

「幸せを感じる生産性体験」(Blissful Productivity)

- 人間は、リラックスしているよりも、意味があり(meaningful)、報われる(rewarding)ことを一生懸命やることに幸せを感じるように最適化されている

「叙事詩的な意味合い」(Epic Meaning)

- これはすごいことだ、畏敬の念をいだいてしまうと信じられることに強く動機づけられる

ジェーン・マゴニガル「ゲームで築くより良い世界」 より引用 http://www.ted.com/talks/jane_mcgonigal_gaming_can_make_a_better_world.html



ソフトウェア開発は楽しいですか？

コーディングは楽しい
けど

なぜこの機能作るのが
わからん！

元の仕様がよくわから
ないときはキツイ

調べようにも誰にも聞けばい
いのかね..

みんなのを統合して
ビルドすると動かない

いつものこと、あきらめてるよ
困ったもんだ

仕様変更が終盤に多
発するとげんなりする

どんだけ直せばいいの？



ソフトウェアテストは楽しいですか？

もっと効率的
にテストしたい

テストを書いてから実
装しないと気持ち悪い

だって、全部やれ
って言うてもできないよ

それなのにテスト書く時間
が確保されていない！

バグが出続ける夢を
て夜も寝れません

一生バグを直し続けなきゃ
いけない気がする

これからテストってなる
と憂鬱になる

バグが出たってどう直せばいい
かわからないし

バグを見つけろと言わ
れても見つかりません

適当に触っても一向に
バグが出ないし、帰れない



こんな状況で、ソフトウェア開発(テスト)の...

達成感ある？

ソフトウェア開発には「達成感」を得る仕掛けがあるか？

ゲーミフィケーション活用実践のための11の施策

「切迫感のある楽観性」

(Urgent Optimism)

「ソーシャル構造」

(Social Fabric)

「幸せを感じる生産性体験」

(Blissful Productivity)

「叙事詩的な意味合い」

(Epic Meaning)

- 入門教育
 - ・ 無理なくやり方を理解できる
- 状況の可視化
 - ・ 品質、進捗、課題などが常に可視化される
- 自主・自律
 - ・ 強制的でなく自主的にできる
- 難易度コントロール
 - ・ レベルに合わせた仕事が割り振られる
- 協働
 - ・ 協働で成し遂げる
- 競争
 - ・ 仕事において、良い意味でのライバルを持つ
- ダイレクトフィードバック
 - ・ アクションの結果が即座に出る
- ルール
 - ・ 物事の進め方が合理的である
- 称号・成長
 - ・ 成長に応じてレベルアップする
- デザイン・演出
 - ・ 気持ちを掻き立てる仕掛けを取り入れる
- 偉大なる段階的ゴール
 - ・ 「世界一になる」といった類の偉大なゴールが明確に存在し、到達するまでの段階ゴールも明確である

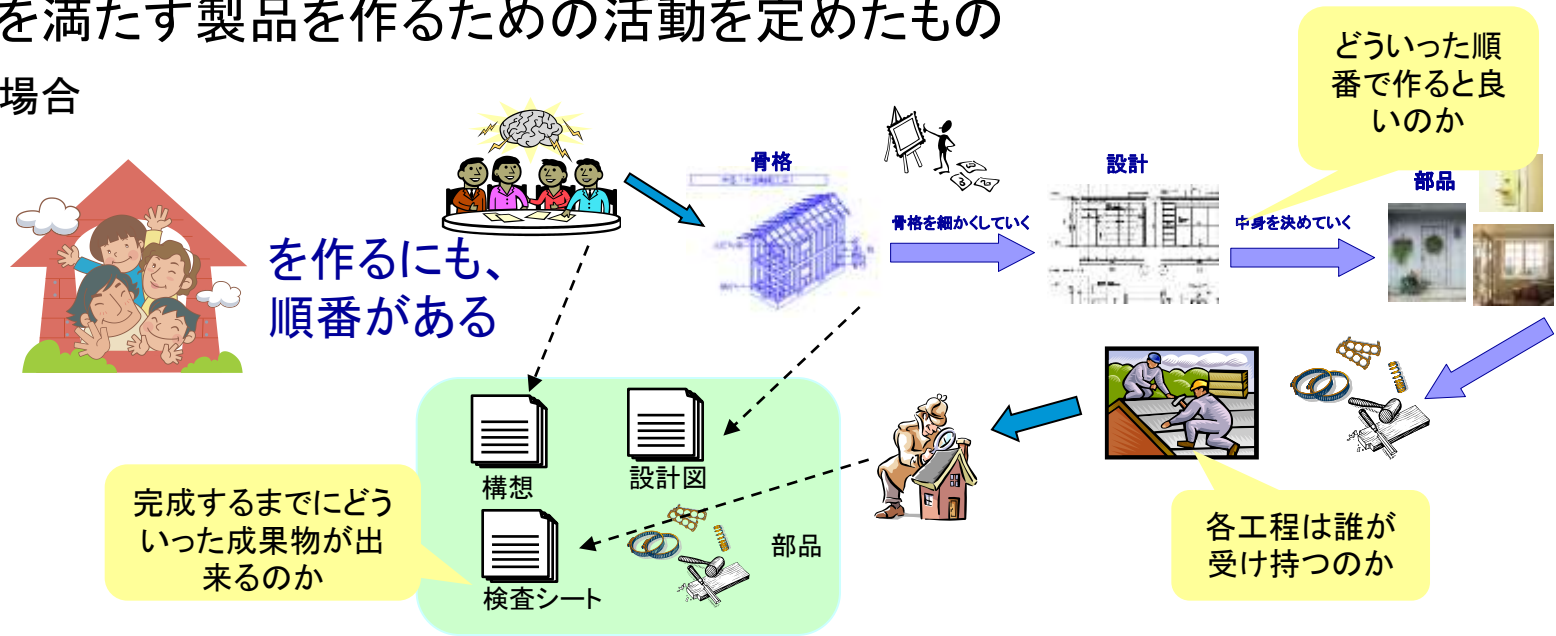
ソフトウェアテストは開発にどう 貢献しているか？

ソフトウェアテストの役割と価値

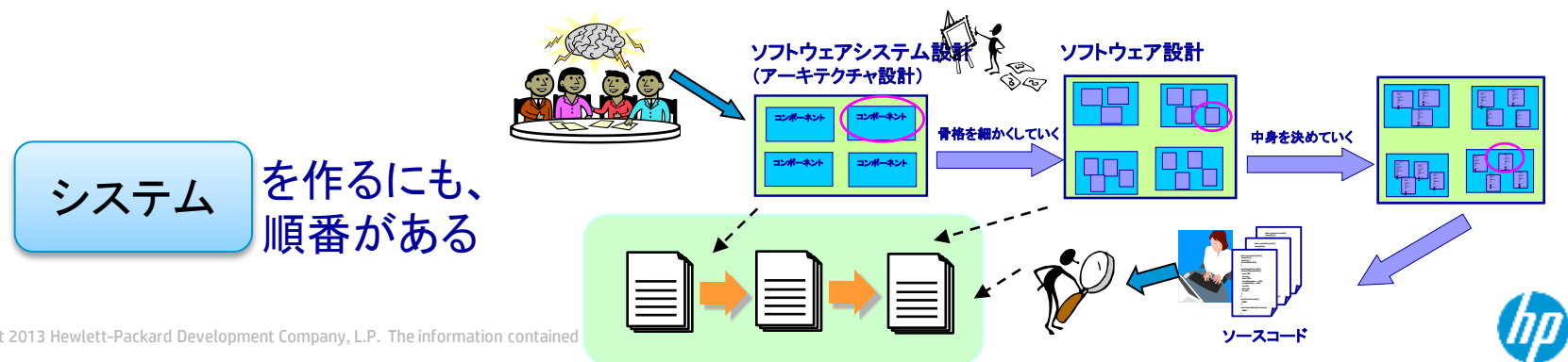
ソフトウェア開発を一軒家購入に例えると...

－ 要求を満たす製品を作るための活動を定めたもの

・ 家の場合

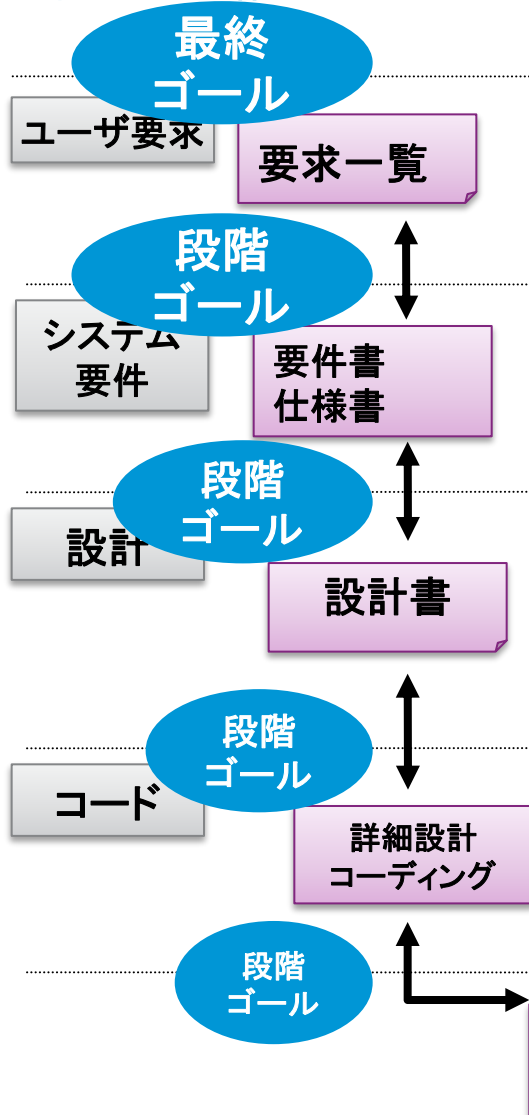


・ システム開発の場合

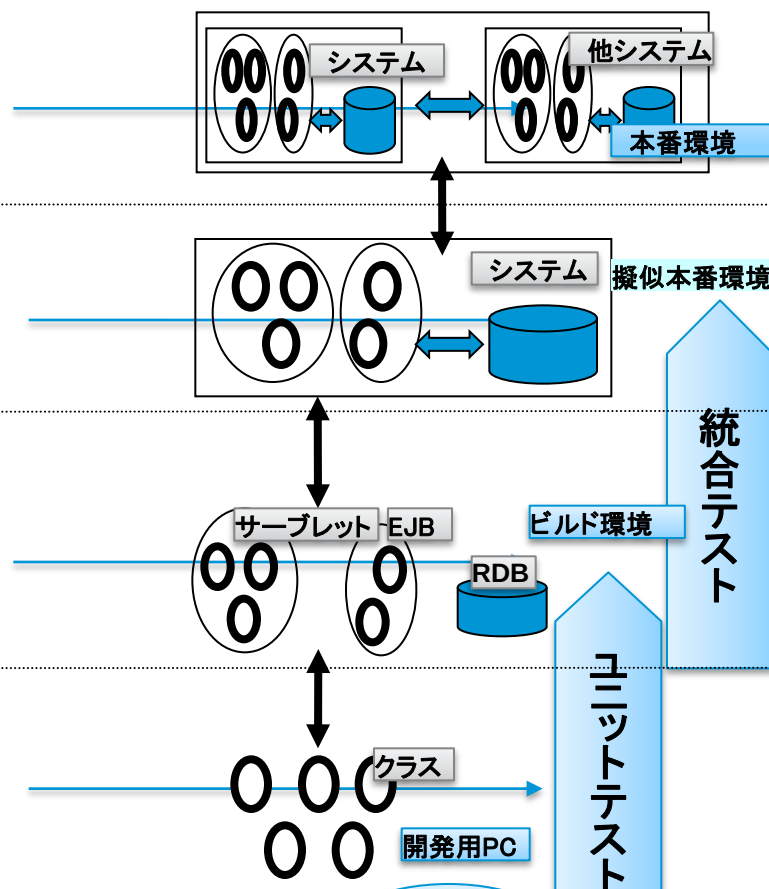


開発の各段階はいつ達成するのか？

偉大なる段階的ゴール



テストレベル



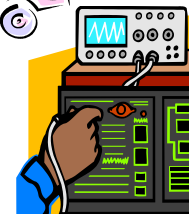
受入テスト



システムテスト



統合テスト



ユニットテスト



ソフトウェアテストをサッカーに例えると...



・サッカーにたとえば、「守り」がテスト
全員がそれぞれの立場で「守り」を行う



・テストはゴールキーパー
だが守り(テスト)はゴールキーパー一人でやることではない
キーパーが大活躍している試合は見ていてハラハラしないか？

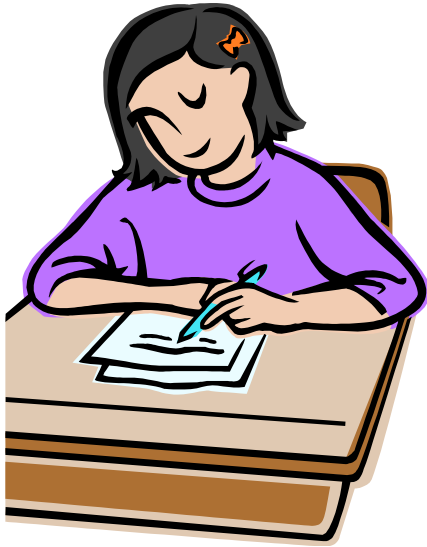
・FWやMFが優秀で攻めも守りも出来ればゴールキーパーは要らないか？



みんながテストについて理解を持って取り組んだほうが上手くいく

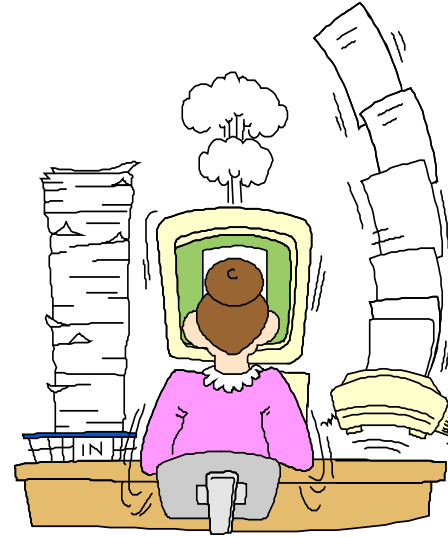


ソフトウェアテストを家計簿に例えると...



家計簿を付ける

=



ソフトウェアをテストする

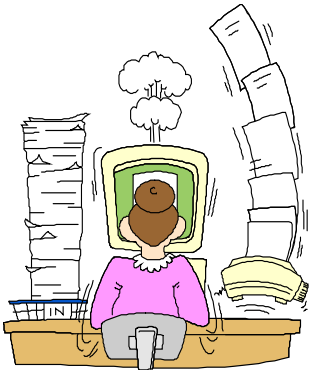
- 家計を決められた金額でまかなえない恐れがあるから家計簿をつける
- 期待通りに動かない恐れがあるからテストをする

実情を確認しているだけ？



•家計簿付けているだけじゃ何も変わらない

- 家計簿の結果をベースに夕飯のおかずを節約するなどのフィードバックをすることに価値がある



•テストを実行するだけじゃ何も変わらない

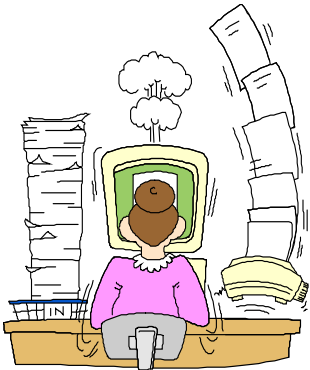
- 期待通りに動作するか確認し、動作しなかったらバグレポートを起票して修正してもらうことに価値がある

余計な手間？



- レシート集めたり、帳簿に転記したり。。

- その手間を惜しむか惜しまないかが目標達成しているか判断できるかどうかの分かれ道



- テストを実施するだけじゃなくって、テストを計画/設計/保守していく事が大事

- テストをエンジニアリングすることがテストの目的を達成し、テストの価値を高める

テストの目的

テストの目的の典型例

欠陥の検出

- 仕様・要件に対する不具合を見つける(当たり前品質の確保)
- 意図的な欠陥識別
 - e.g. ストレス系・評価系のテスト

品質レベルが十分であることを確認

- 仕様・要件に対する合致性の確認
- 技術課題に関連して: 解決策の妥当性確認・検証
- リスクに関連して: リスクの評価、対応策の妥当性確認・検証

判断のための情報を提示

- 信頼度の測定
 - e.g. 運用プロフィールに沿ってランダムに生成した

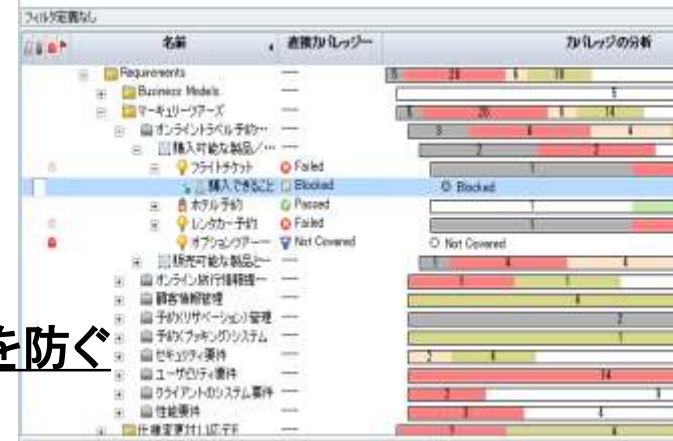
早期にテスト視点で、レビューを行うことで欠陥の作りこみを防ぐ

- 要求に関連して: 要求の妥当性確認、要求の抽出(の促進)
- 要件・設計に関連して: 要件・設計の妥当性確認・検証

ダイレクトフィードバック



状況の可視化



テストの役割と価値

「ゲーミフィケーション活用実践のための11の施策」とのつながり

テストの直接的な効果

欠陥の検出

ダイレクトフィードバック

状況の可視化

品質レベルが十分であることを確認

ダイレクトフィードバック

状況の可視化

偉大なる段階的ゴール

判断のための情報を提示

ダイレクトフィードバック

状況の可視化

偉大なる段階的ゴール

早期にテスト視点で、レビューを行うことで欠陥の作りこみを防ぐ

ダイレクトフィードバック

状況の可視化

テストの間接的な効果

デザイン・演出

ユニットテストツールのグリーンライト

競争

欠陥検出数で競争するなど

ルール

テスト結果という事実を基に進捗を測る(理にかなっている)

入門教育

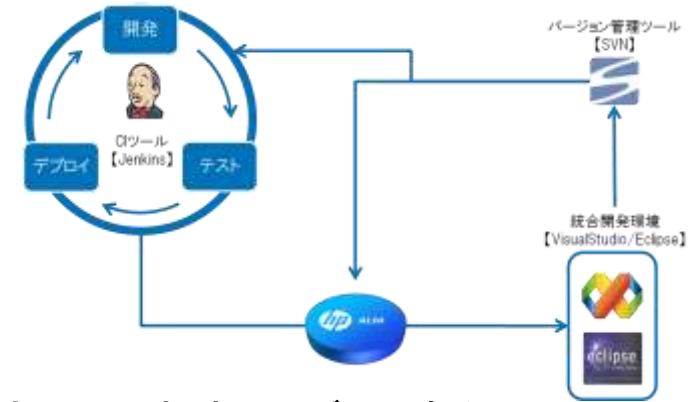
テスト結果、過去のバグをナレッジにして学習をする(リアルな教材)

開発の達成感を共有するための**有益な情報提供**をすることにテストの価値がある！

有益な情報提供とは具体的にいうとどんなものがある？

有益な情報の例

- 欠陥除去
 - テストでバグがでてから修正、再確認までをトレースする
- 今後の改善(欠陥予防)
 - 欠陥原因分析、DREなどのデータ収集を行なう／テスト設計を並行させて設計と相互レビューする
- リリースのための判断材料
 - 要件カバレッジ、テスト実施カバレッジ、バグ修正状況(重要度別)、信頼度成長曲線、バグ管理図などのデータ収集を行なう
- 品質リスクのマネジメント情報
 - 品質リスク(市場でのお客様からのクレーム/運用開始後の機会損失etc..の恐れ)を数値化し、テストすることによりどれだけ軽減されているかを確認する。



言うのは簡単！実際には、何を達成したいか決めて、段取りや仕掛けを入れこまないとね！

これは楽しいですか？



ソフトウェアテストの全体像

テストに必要な活動を知る＝何をすればよいかを知る

そもそもテストとは

テストの定義

ソフトウェアテストとは、通常は無限に大きいと考えられる「プログラムの振る舞いの実行領域」から、最適だと考えられる有限な「テストケースの集合」を選定し、**所期の通りかどうかを実際に動かして検証すること (SWEBOK 2004)**

- 単に「動かして検証する」だけの行為である。

例えるなら

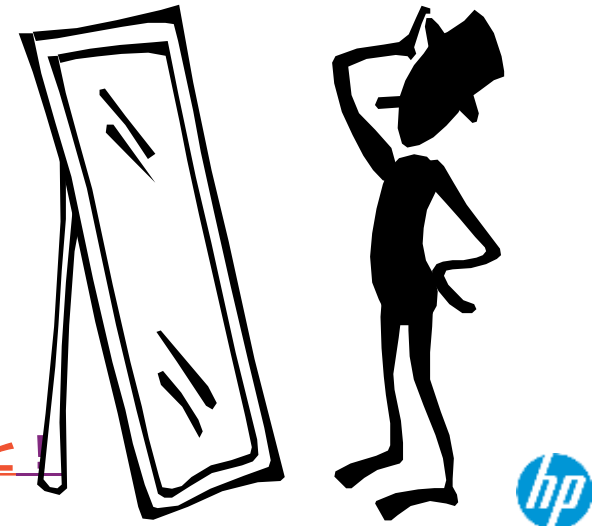
テストは品質を写す「鏡」、テストは鏡を使って知る「行為」

- e.g. コード, コーディング

テストには「何故その行為をするのか？」が必要

- どこを見たいのか？
- 見ることで何を知りたいのか？
- 知ってどうするのか？
- どう見るのか？いつ見るのか？

つまり...何を持って「達成した」とするのか重要だということ



テストは実行だけではない

テスト実行だけでは、「何故その行為をするのか？」が不明のまま
仕組みや仕掛けを入れ込んでおかないと、実行するのもままならない

開発ライフサイクル



テストがテスト実行するだけに見えてしまう

一般的な開発ライフサイクルの表現では、開発が終わるとテストをするという流れになっている

テストは実行だけではない

テスト実行はだけでは、「何故その行為をするのか？」が不明のまま仕組みや仕掛けを入れ込んでおかないと、実行するのもままならない

開発ライフサイクル

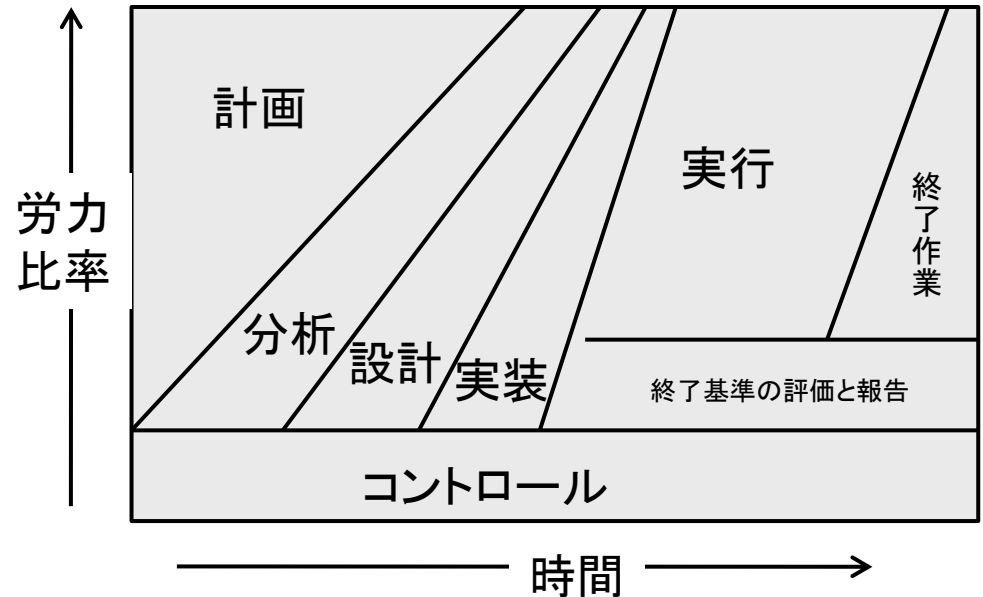


ISTQBが定義するテストプロセス

<http://jstqb.jp/>

ISTQBでは、テストプロセスは、以下の活動に分けて考えるとしている。

- 計画
 - コントロール
 - 分析
 - 設計
 - 実装
 - 実行
 - テスト終了基準の評価と報告
 - テスト終了作業
- テスト開発プロセス



これらの活動はひとつずつ順番に行われるとは限らず、重なったり、並行したり、繰り返されたりする。

テストを開発するとは何か？

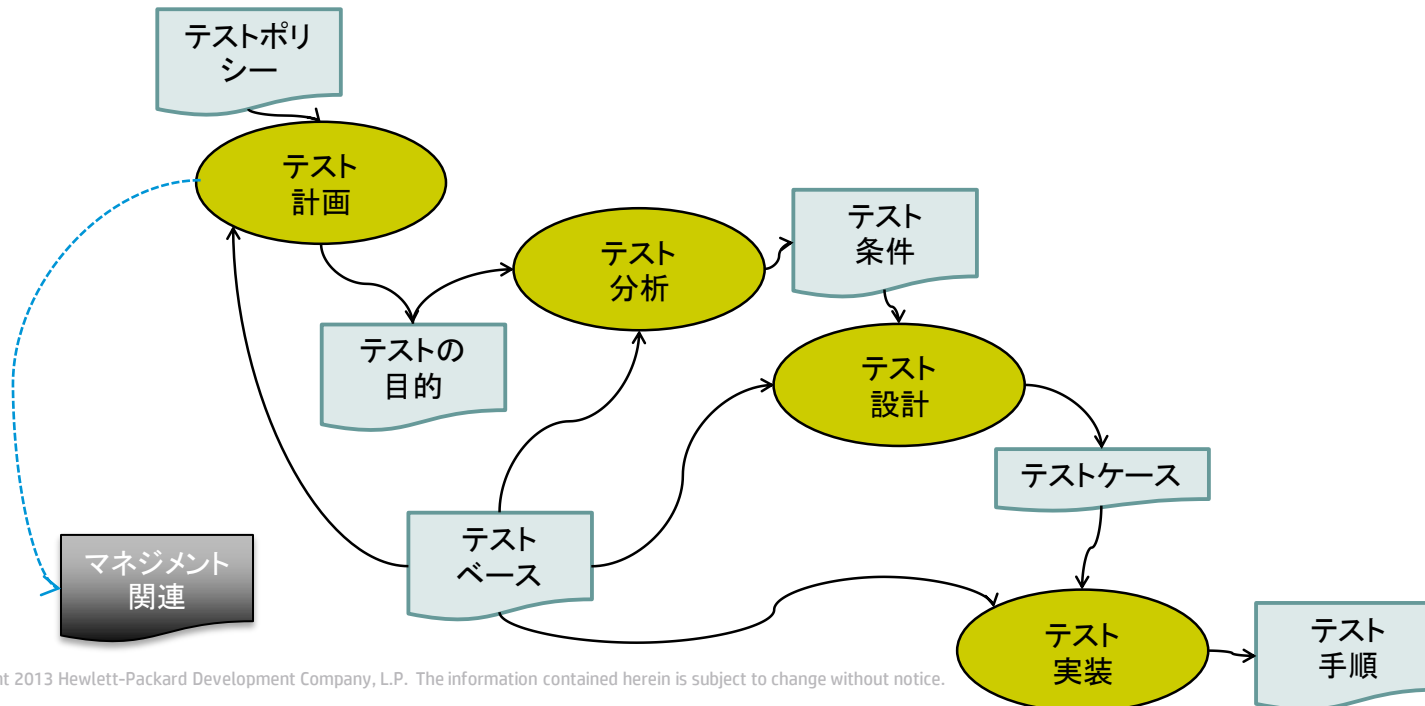
テスト開発プロセス

「何を持ってそれを達成したか？」がわかるテストを実行するための活動

- テストの目的やテスト対象を明らかにする(計画する)
- テストを造れるようにテスト対象を分解する(分析する)
- 目的に沿ったテストを造る(設計する)
- テストを実行できるようにする(実装する)
 - (手動なら手順を考える)(自動ならテストコードを作る)
 - 環境構築、テストデータ作成

目的...「達成」したいこと

段取りや仕掛け



テスト開発プロセスについてさらに知りたい人は...

テスト分析・テスト設計手法の活用

JaSST'09Tokyoパネルディスカッション

「テスト技法からテストメソドロジへの進化を目指して」と題して、日本のテスト業界での第一人者が各自の考え方/実践手法を紹介

秋山 浩一（富士ゼロックス）

- HAYST法

池田 暁（日立情報通信エンジニアリング）

- マインドマップを使ったソフトウェアテスト/TAME

工藤 邦博（ベリサーブ）

- 俺流！（現場で実践している）テスト設計方法論

鈴木 三紀夫（TIS）

- マインドマップを使ったソフトウェアテスト/Tiramis

西 康晴（電気通信大学）

- VSTeP

松尾谷 徹（デバッグ工学研究所）

- CFD法（探索理論/VOP）

湯本 剛（豆蔵...現日本HP）

- 豆蔵流テストングプロセス/ゆもつよメソッド

智美 塾

テスト開発のメタモデルを作り
各自が独自の方法論をつくれ
るようにするための場
(TEFに入ると案内が流れます)

ここで...

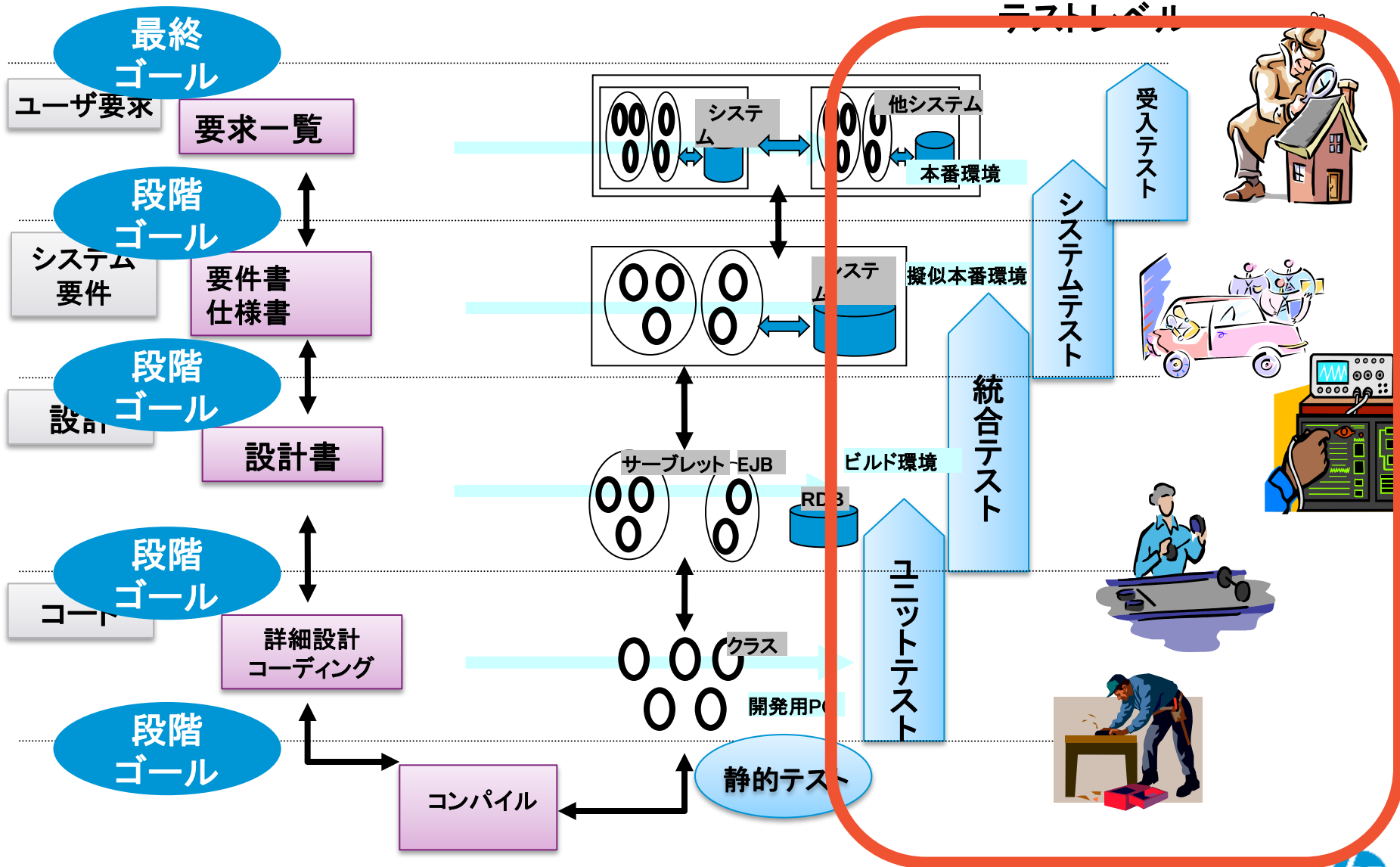
今日説明している ソフトウェアテストの概念の整理



開発のどの詳細レベルをターゲットにテストをするのか？

テストレベル

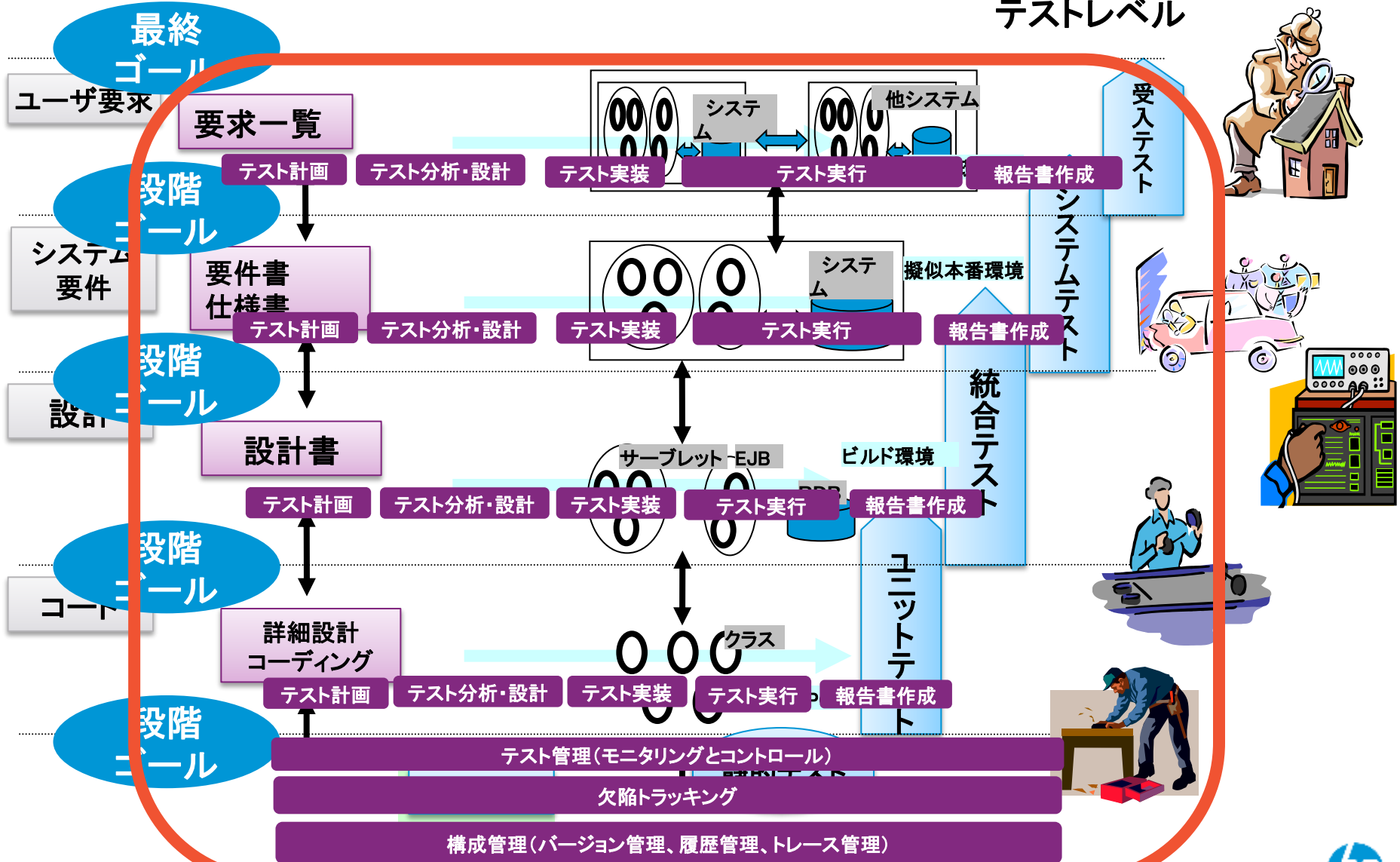
テストレベル



テストを実行するまでどのように進めていくのか？

テストプロセス

テストレベル

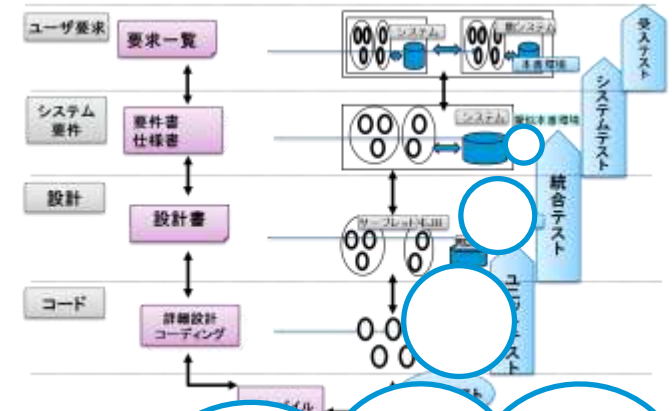


[補足]どのテストの目的に対するテストをするのか？

テストタイプ

品質特性に向けたテスト活動の分類

各テストタイプは、特定のテストの目的にフォーカスしている



テスト目的を満たす事が「十分」であるということは...

プログラムコードを実行してエラーにならないければよい

- 全ソースコードを一回は動かしてエラーにならなかった
- 全部の分岐を一回は動かしてエラーにならなかった

仕様に書いてある通りに動けばよい

- ●●仕様書通りに動いた
- ●●仕様書に書いてないが本来書くべき事(例外、異常)

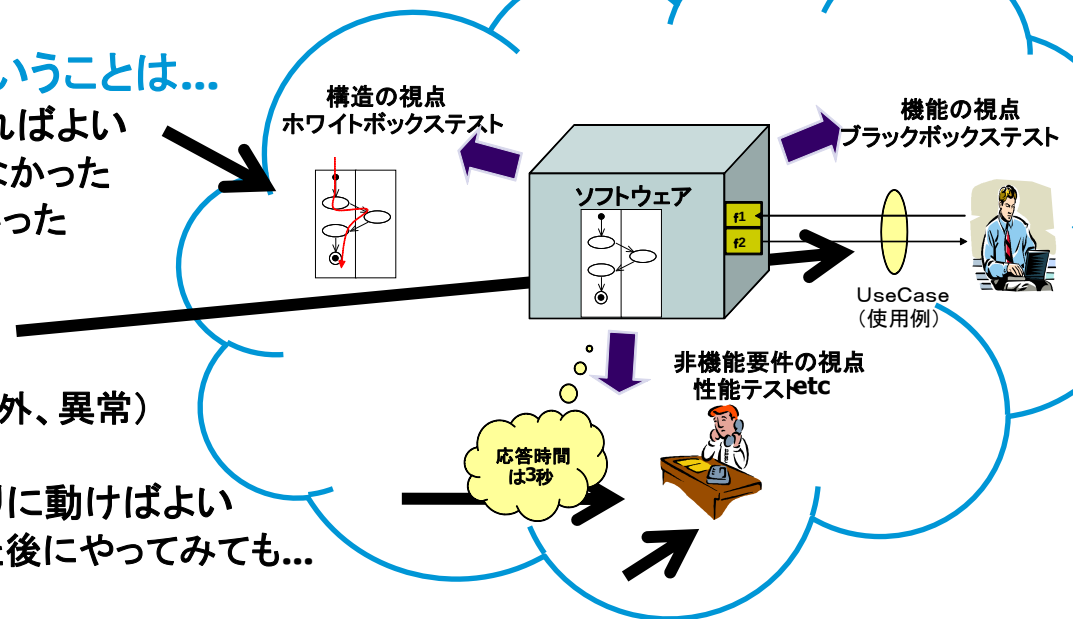
「いつでもどんな時でも」仕様に書いてある通りに動けばよい

- ●●を何回も繰り返しても、一度エラーになった後にやってみても...

「いつでもどんな時でも」「所定の速度で」仕様に書いてある通りに動けばよい

- データ量が膨大でも、CPU負荷が高いときも...

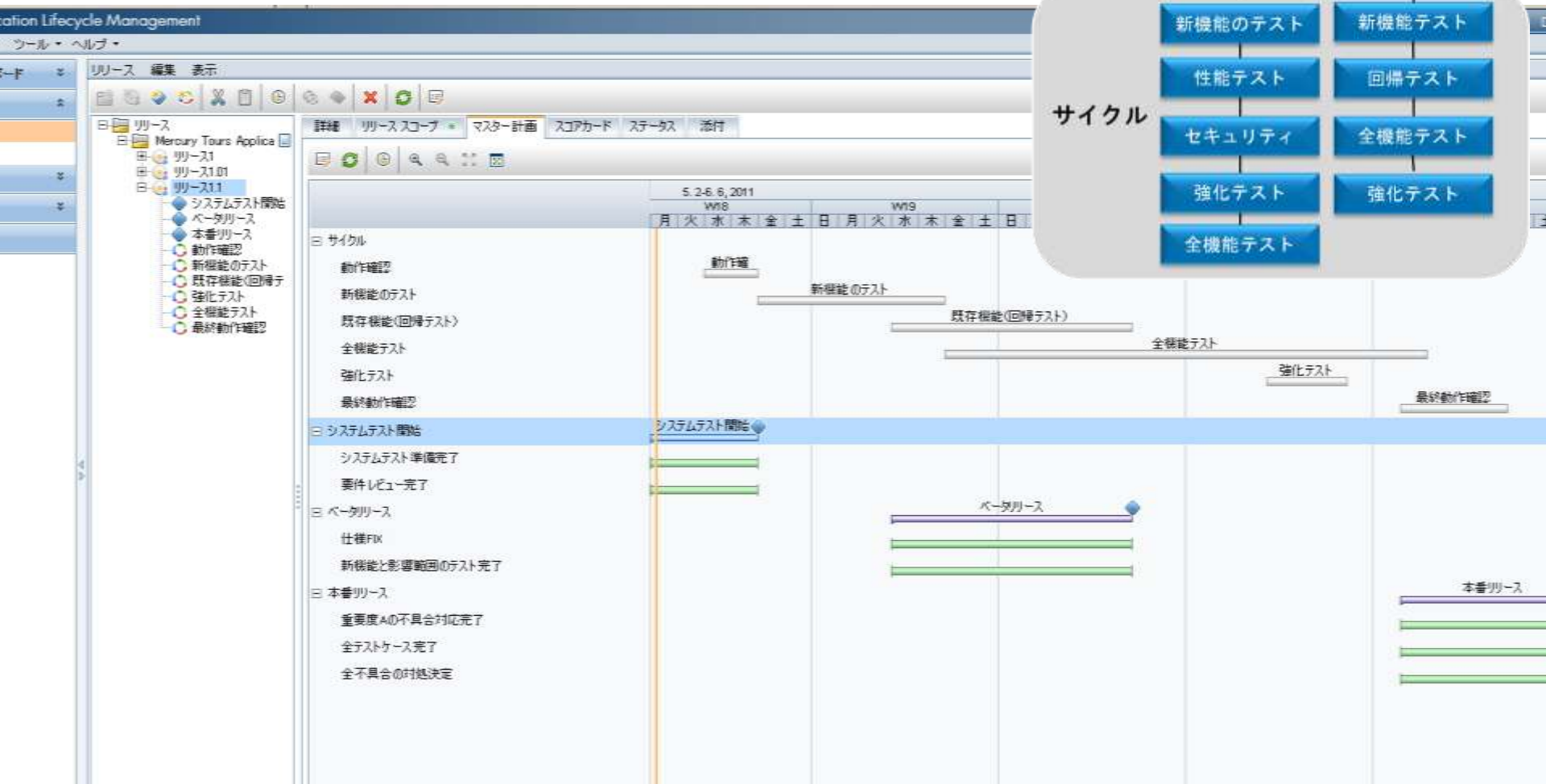
etc...



[補足]テスト実行をどういう順番で進めるか？

テストサイクル (by HP QualityCenter)

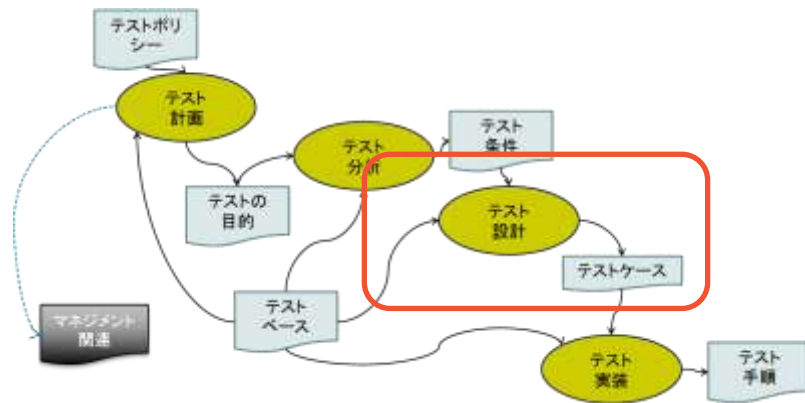
リリーススケジュールに基づいて共通の目標を達成するために行われる一連テスト作業



ソフトウェアテストにて押さえておくべきこと

テスト設計
テスト分析
テストツール

テスト設計

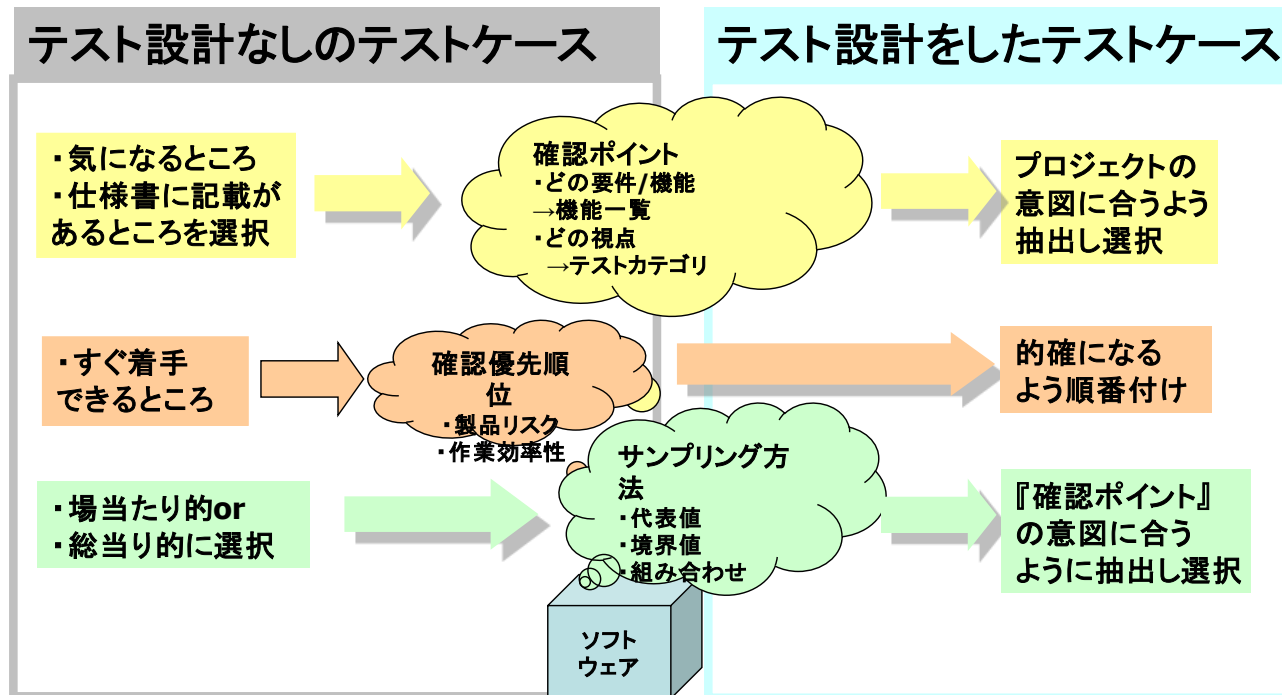


テスト設計...テスト目的を実現するための「手段」

テスト目的とテスト対象を決めたらテストが実行できるか？

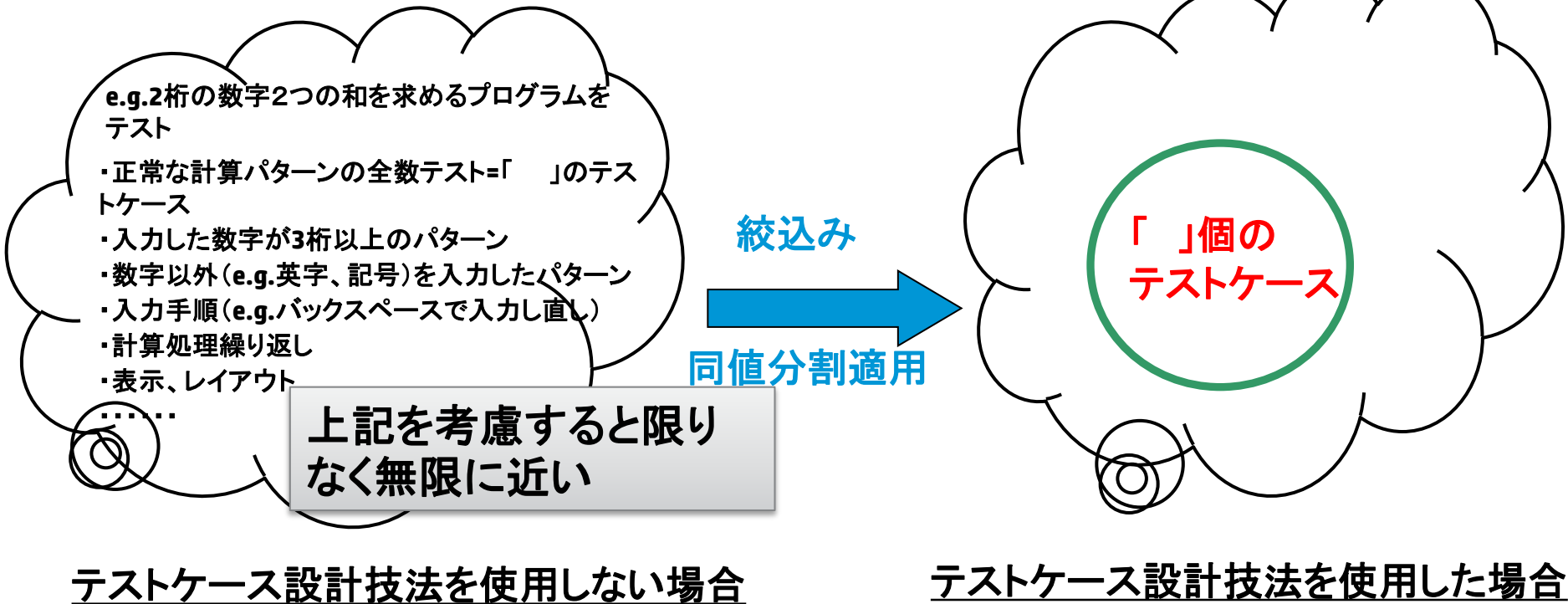
- テスト対象に関することは全て仕様書に書かれているか？
- 仕様書に書かれた事全てテストしたほうがよいか？
 - 時間が足りませんか？(工夫して適切な量に収めることが必要)
 - 十分ですか？(テストの目的に沿ってないといけない)

テストを設計すること＝テストの目的を満たすテストケースを造ること



一般的なテスト設計の説明

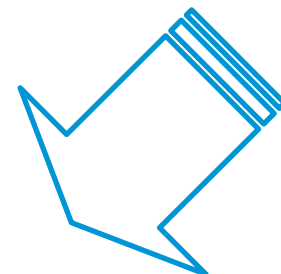
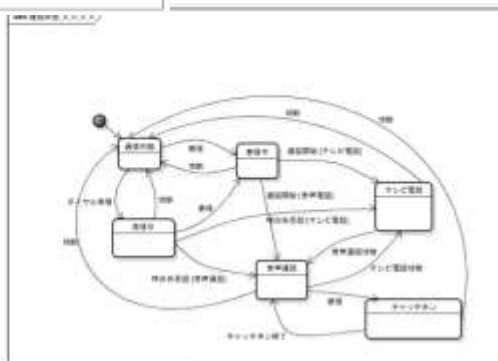
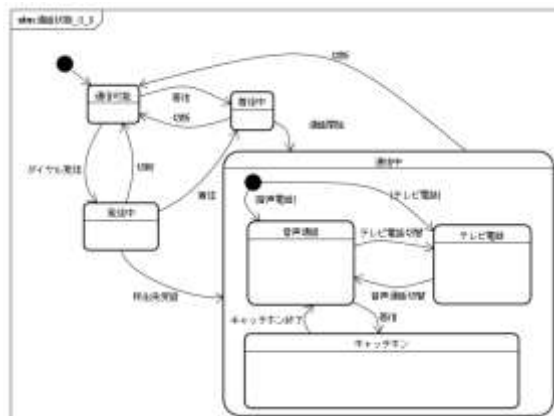
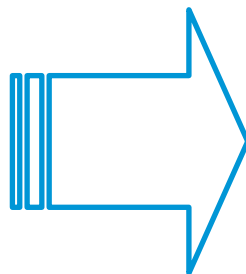
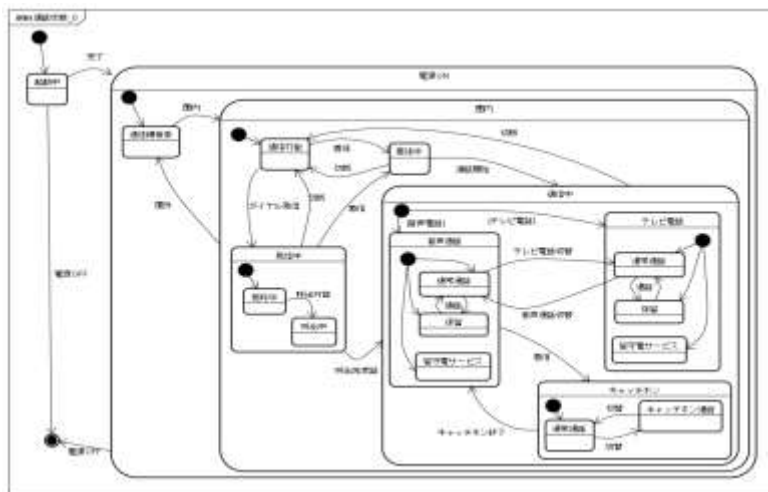
テストケース設計技法の使用例



テストケース設計技法の例

- ・ 同値分割、デシジョン・テーブル...

状態遷移モデルを使ったテスト設計



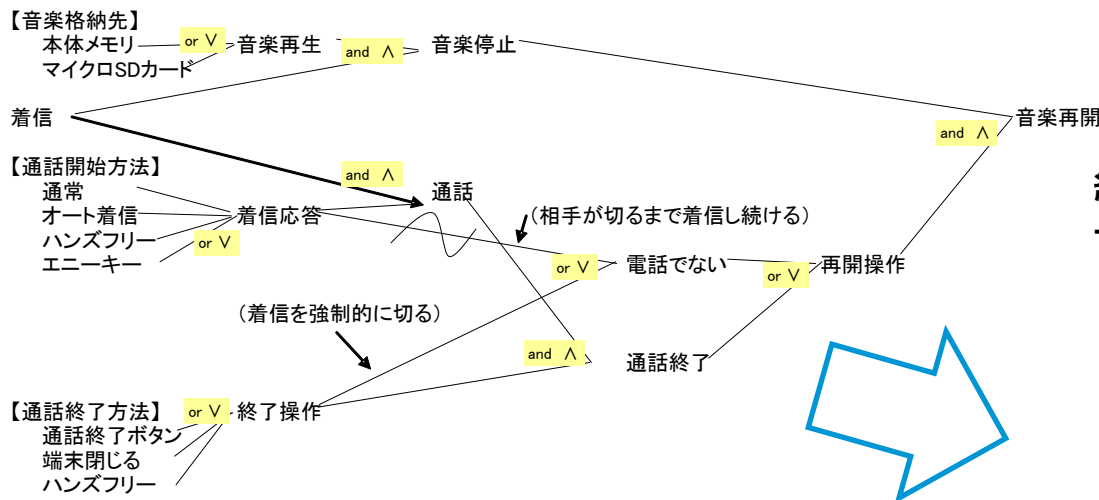
番号	状態	遷移	状態
1	待機	ダイヤル	通話開始
2	通話中	通話終了	通話終了
3	通話中	通話開始	通話開始
4	通話中	通話終了	通話終了
5	通話中	通話開始	通話開始
6	通話中	通話終了	通話終了
7	通話中	通話開始	通話開始
8	通話中	通話終了	通話終了
9	通話中	通話開始	通話開始
10	通話中	通話終了	通話終了
11	通話中	通話開始	通話開始
12	通話中	通話終了	通話終了
13	通話中	通話開始	通話開始
14	通話中	通話終了	通話終了
15	通話中	通話開始	通話開始

Astah*(<http://astah.change-vision.com/ja/product/astah-professional.html>)で作成

原因結果グラフを使ったテスト設計

音楽再生時に着信すると一時停止した場合、着信が終了すると再生する

上記仕様から原因結果グラフ
(ブルグラフとも呼ぶ)を作成する



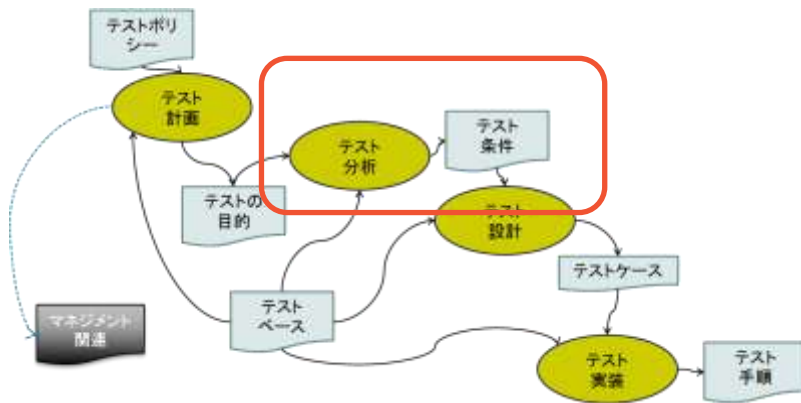
結果を設定する組合せを見つけて
デシジョンテーブルを作成する

	ノード名	#1	#2	#3	#4	#5	#6	#7	#8	#9
原因	本体メモリ	T	F	T	T	T	T	F	T	T
原因	マイクロSD	F	T	F	F	F	F	T	F	F
原因	着信	T	T	F	T	T	T	T	T	T
原因	通常	T	F	F	F	T	T	T	F	T
原因	オート着信	F	T	F	F	F	F	F	F	F
原因	ハンズフリー	F	F	T	F	F	F	F	F	F
原因	エニーキー	F	F	F	F	F	F	F	T	F
原因	通話終了ボタン	T	F	T	F	T	T	T	T	F
原因	端末閉じる	F	T	F	F	F	F	F	F	F
原因	ハンズフリー	F	F	F	F	F	F	F	F	T
中間	音楽再生	T	F	T	T	T	T	T	T	T
中間	音楽停止	T	F	F	T	T	T	T	T	T
中間	着信応答	T	T	T	F	T	T	T	T	T
中間	通話	T	T	F	F	T	T	T	T	T
中間	通話終了操作	T	T	T	F	T	T	T	T	T
中間	通話終了	T	T	F	F	T	F	T	T	T
中間	電話でない	T	T	T	T	F	F	T	T	T
中間	再開操作	T	T	T	T	T	F	T	T	T
結果	音楽再開	T	F	F	T	T	F	T	T	T

単純に考えれば1024パターンとなる
デシジョンテーブルが9パターンに絞り込める

<http://softest.cocolog-nifty.com/labo/CEGTest/>で作成

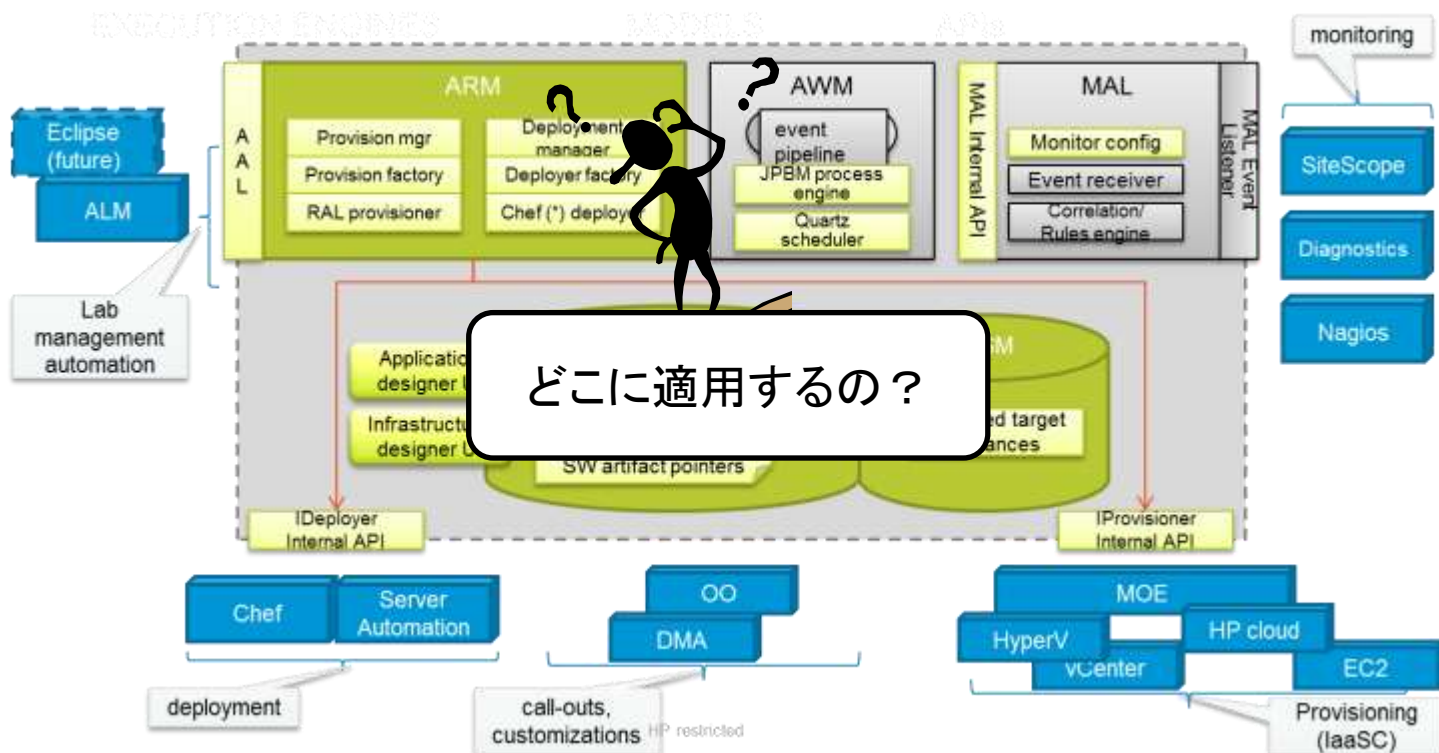
テスト分析



現実の問題

典型的なテストケース設計技法の教科書では...

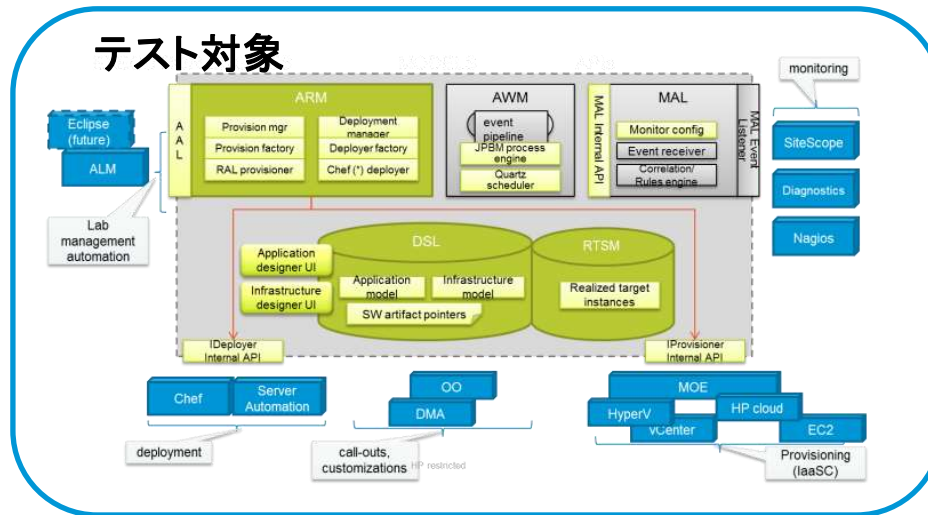
- 小さい問題を扱っている
 - プログラムレベルの問題(理解しやすい)
 - 現実にテストする対象のごく一部しか扱うことができない



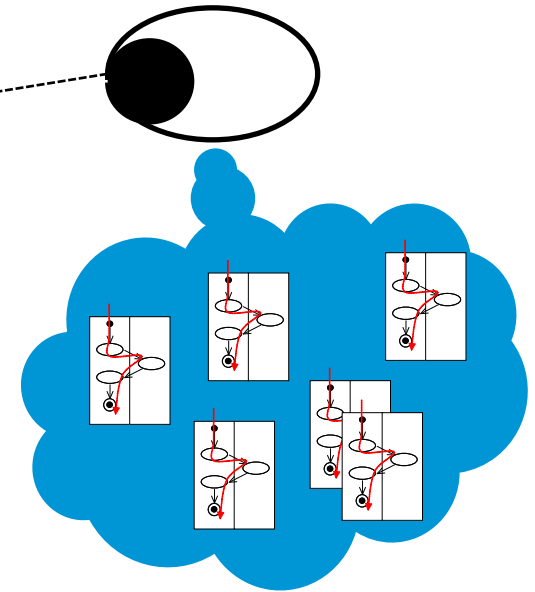
分析してますか？

テストケースを作るためにはテスト対象をテスト技法が適用できるようなサイズに(論理的に)分解しなければならない→これが分析

- 分解の切り口＝観点



観点



6モジュールはテストが必要！

たとえばこんな分解で詳細化した場合

テストケースを書くときに「大項目」「中項目」「小項目」といった分解、詳細化をすることが多い

- 以下の2つの書き方は何が問題でしょうか？

大項目	中項目	小項目
正常系確認	他機器との接続	デジカメ×プリンタ
ピクトブリッジ	接続パターン	正常系
性能テスト	XMLパーシング	ピクトブリッジ

大項目	中項目	小項目1	小項目2	小項目3	テストケース
印刷	設定	印刷部数	---	---	100部印刷した場合
設定	プリント設定	一般	異常系	エラーメッセージ	「印刷部数が99部超えました」と表示されること

分解の考え方

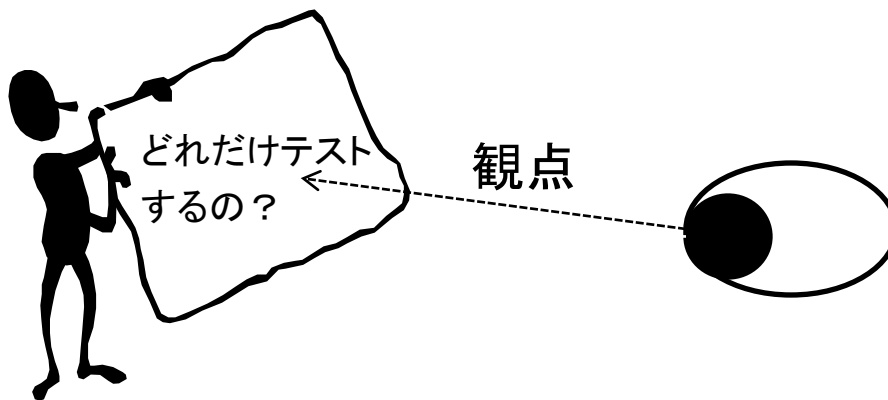
テスト目的を2つの「観点」で具体的にする。

見方(網羅的、ピンポイント的)...目的のブレイクダウン

見る位置(外観、中身など)...テスト対象のブレイクダウン

• テスト対象

- システム...要件定義書
- 機能...外部仕様(画面仕様)
- モジュール...内部仕様(プログラム仕様)
- etc



• テスト目的

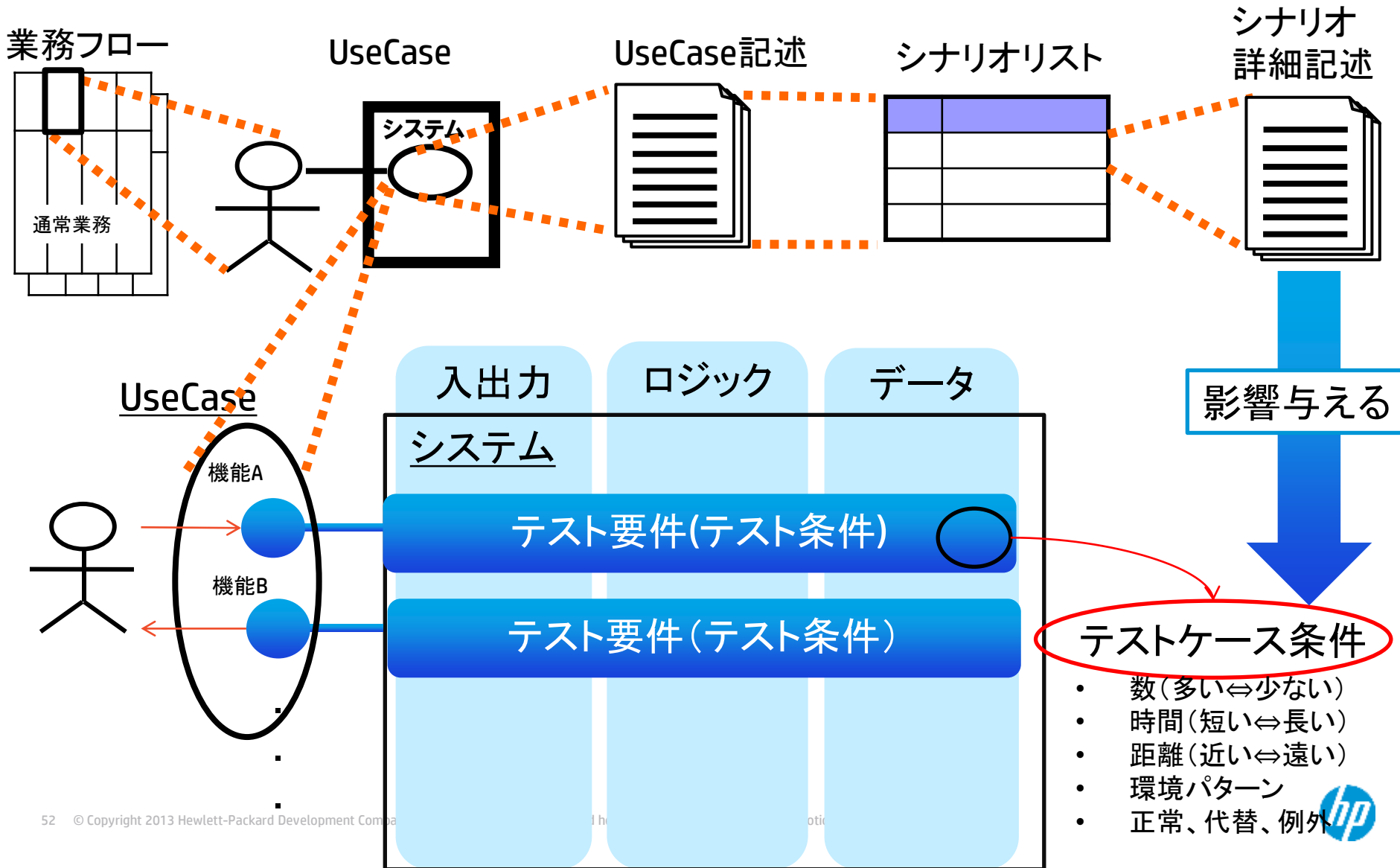
<網羅>

- 構造を網羅(ホワイトボックス)
- 機能を網羅(ブラックボックス)
- 非機能を網羅

<ピンポイント>

- バグが出そう
- バグになると大問題

ブラックボックステストでのテスト対象に対するテスト観点



テストケースの構造

本来のテストケースは以下の構造となり、段階的に成果物を作っていく
(つまり、プロセスがある)

画面

テスト対象機能項目

仕様項目

テスト分析
(機能一覧、テストカテゴリ)

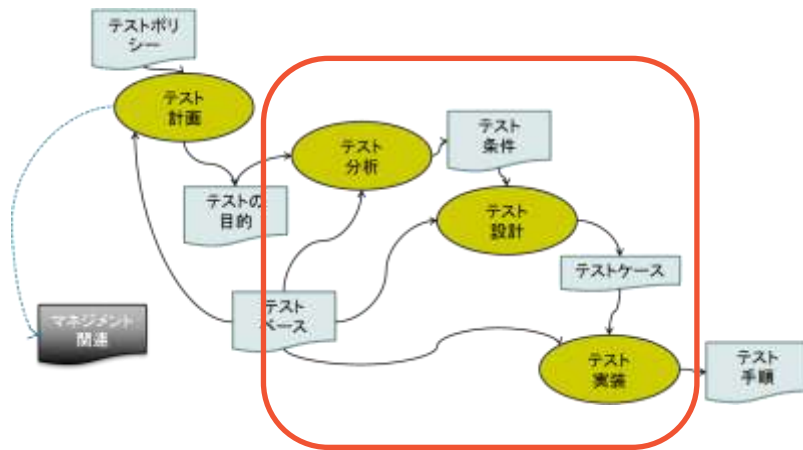
テスト条件(テストすべき事柄)

テストケース

テスト設計
(マトリクス、図)

テストケース条件	値	操作	期待結果
- 入力データ - 事前状態 - 環境 - タイミング			- 出力結果 - 事後状態

テストツール

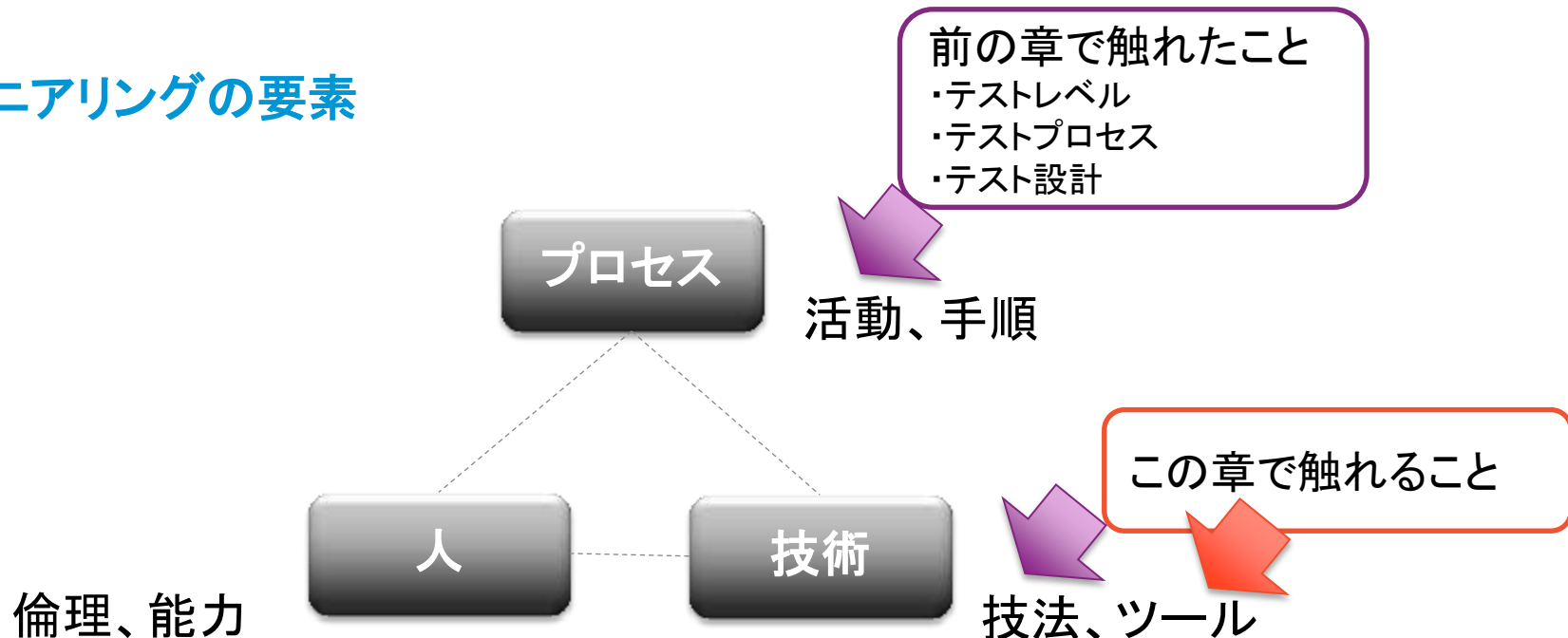


ソフトウェアエンジニアリング

エンジニアリングの定義(IEEE610)

“The application of a systematic, disciplined, quantifiable approach to structures, machines, products, systems, or processes” (構造、機械、製品、システムまたはプロセスへ体系的で、訓練されていて、定量的なやり方を適用すること)

エンジニアリングの要素

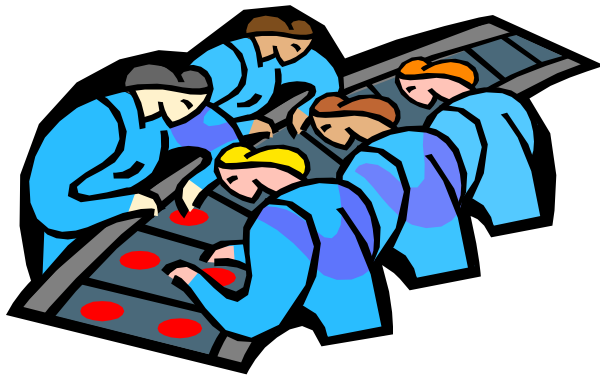


開発ツールはエンジニアリングの一要素

エンジニアリングの他の要素を支援するのが**道具（ツール）**

- ・ 支援する相手がエンジニアリングとして成立してないと成果が出ない
 - 「本来やるべきこと」を効率化する
 - 「やっていないもの」はツールでは効率化できない（やってないことをやると逆に工数が増える）
- ・ 人間が手でやるときには考えられなかったことが実現可能になる
 - 人間がやるならば**100人は必要だ**と思うことでも**道具を使えば1人で出来て**しまう

人間が束になってこなすことを



より「速く」「長く」「正確に」行ってくれる



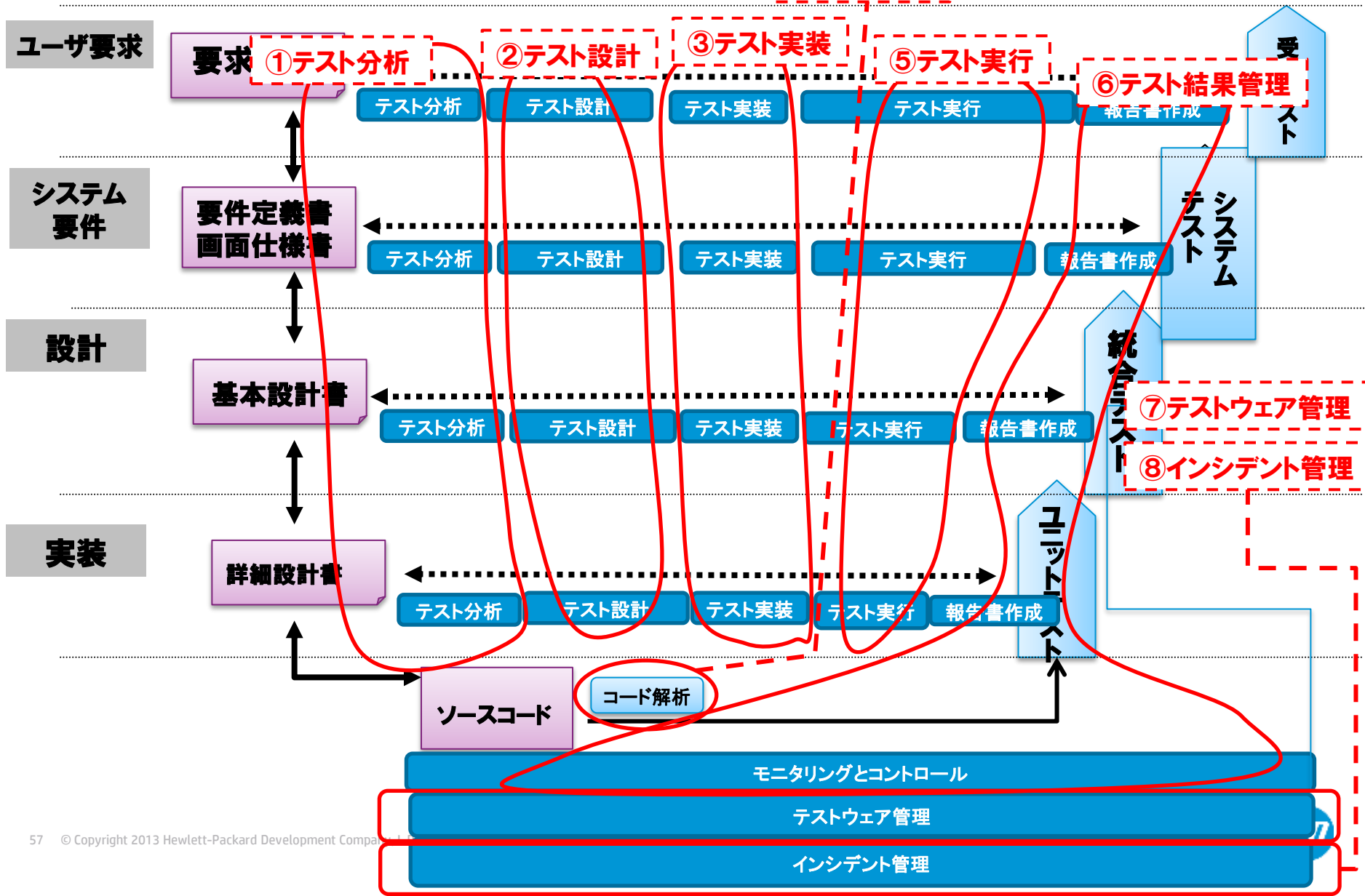
テストプロセスとテストツールの関係

開発の作業

開発の成果物

テストの作業(プロセス)

④コード解析



テストの作業	作業を支援するツールの種類
テスト分析	要件管理ツール
テスト設計	状態遷移テストツール 組合せテスト支援ツール(直交表、オールペア) 原因結果グラフツール 動的解析ツール カバレッジ計測ツール その他のテスト設計支援ツール
テスト実装	スタブツール シミュレータ ラボイメージツール テストデータジェネレータ テストケース管理ツール
コード解析	静的解析ツール 構造解析ツール
テスト(自動)実行	ユニットテストツール キャプチャ/リプレイツール 性能テストツール セキュリティテストツール テスト自動実行支援ツール
テストウェア管理	構成管理ツール
テスト結果管理	テスト結果管理/テスト結果レポートツール
インシデント管理	インシデント管理ツール

「テストツールまるわかりガイド」

NPO法人ASTER(ソフトウェアテスト技術振興協会)の研究事業のひとつである「テストツールWG」の活動の成果として発表

テストツールまるわかりガイド(入門編)

Version 1.0.1

ASTER テストツール WG 著



発行 NPO法人 ソフトウェアテスト技術振興協会

本冊子の目的

- ・日本のソフトウェア開発の現場におけるテストツールの普及と導入推進
- ・開発現場の若手から管理者まで幅広い層に対してテストツールを活用するための基礎的な情報を提供

本冊子の特徴

- ・有償／無償ツールを分け隔てることなく、現場で活用できる多様なテストツールを紹介
- ・オープンソースのテストツールについては、ユーザの立場である策定メンバーが代表的なツールを1種ずつ選定
- ・市販テストツールについては、企業間の垣根を越えて多様なツールを知っていただくために、国内の14社のテストツールベンダ／販社が協力して執筆

この冊子で伝えたいポイント

道具(ツール)ありきではなくテストの作業ありき
何を効率化したいのか？

作業がそもそもちゃんとできないのに効率化は出来ない
家計簿をつけたことがない人が家計簿ソフトを買っても何も効率化しない

道具(ツール)の導入は「改善」がセット

己を知る

- 健康診断チェックリスト

道具(ツール)を使いながら作業もちゃんとやるようにしていく

- 道具を使う場面(作業)を知る(三章)
- ツールの効能を知る/使用上の注意を知る(四章)

「ハイコンセプト」なエンジニアになるための一助に！

- あなたの仕事は他の国よりこれを安く出来る？
- あなたの仕事はコンピュータを活用するともっと上手く早くできる？
- あなたの仕事はこれから需要がある？

道具(ツール)活用が必須の時代であるということ！

まとめ



まとめ

「楽しい」ソフトウェア開発について考えた

- 「達成感」を感じながら仕事ができるか？

ソフトウェアテストは開発にどう貢献しているか

- ソフトウェアテストの役割と価値

ソフトウェアテストの全体像

- ソフトウェアテストに必要な活動を知る＝何をよくしていけばよいかを知る

ソフトウェアテストにて押さえておくべきこと

- テスト設計
- テスト分析
- テストツール

参考文献

湯本剛「日経ソフトウェア 連載 基礎から学ぶソフトウェアテスト」

(日経BP 2005) <http://itpro.nikkeibp.co.jp/article/COLUMN/20070222/262867/?ST=selfup>

高橋寿一,湯本剛「現場の仕事がバリバリ進むソフトウェアテスト手法」

(技術評論社)2006

湯本剛「ソフトウェアテストPressVol.10 特集1 今こそ聞きたいテストの上流設計」

(技術評論社)2010

豆蔵「ソフトウェアエンジニアリング講座 テスティング基本編」

(豆蔵)

IEEE Computer Society「SWEBOKソフトウェアエンジニアリング知識体系」

(オーム社)2005

ISTQB「ISTQBテスト技術者資格制度Foundation Levelシラバス日本語版 Version2011.J01」

(NPO法人ASTER,JSTQB)2011

ASTER テストツールWG「テストツール丸わかりガイド入門編1.00」

(NPO法人ASTER)2012

ジェーン・マゴニガル「ゲームで築くより良い世界」

http://www.ted.com/talks/jane_mcgonigal_gaming_can_make_a_better_world.html

安部 一真「ゲーミフィケーションをマネジメントに活用する方法」～ロールプレイングゲームに学ぶ「部下を魅了する」リーダーシップ～ (PMAJ 2013) <http://www.pmaj.or.jp/library/regular.html>

参考文献ではないですが、、奈良さん、みっきーさんには会話の中でインスピレーションをいただきました！

Thank you
ありがとうございました

