

JaSST'15 Niigata

ソフトウェアテストの作り方

現場で実施されている特に名前も無い方法

2015年4月24日（金）

富士ゼロックス株式会社

ソリューション・サービス開発本部 秋山浩一

Twitterなど、ご自由にどうぞ
企業名は書かないでね（お願い）
@akiyama924
jasst

自己紹介

- 1985年4月 富士ゼロックス入社
 - ◆ HAYST法のコンサルティング業務に従事（テスト上流系）
- NPO ソフトウェアテスト技術振興協会（ASTER） 理事
 - ◆ JaSST東京実行委員（2003～2014年、今はアドバイザー）
日本最大のテストシンポジウム1600名の動員
 - ◆ JSTQBステアリング委員（2006年～）
テスト技術者資格認定を行う国際組織の日本支部
- 日科技連 SW品質管理（SQiP）研究会 委員長（2011～2013年）
 - ◆ 第5（テスト）分科会主査（08～10年）、副主査（2011年～）
- ISO/IEC JTC 1/SC7 WG26委員（2009年～）
ソフトウェアテストの国際標準「ISO/IEC/IEEE 29119」の策定
- 共著書『ソフトウェアテストHAYST法入門』（2007年）
著書『ソフトウェアテスト技法ドリル』（2010年）
『事例とツールで学ぶHAYST法』（2014年：日科技連出版社）
- 博士（工学）



目次

1. 今日のこだわり
 - 1.1 品質とテスト
 - 1.2 iPhoneのバグから考える
2. 品質と改善
 - 2.1 品質改善活動の見える化と炎上の気配を捉える方法
 - 2.2 品質の改善
3. テストの現場
 - 3.1 機能と仕様
 - 3.2 いつものテストと改良ポイント

1. 今日のこだわり

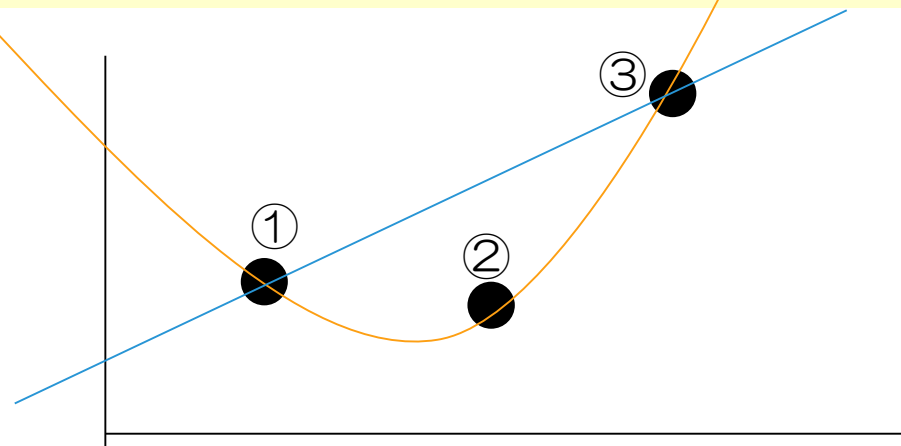
- ◆ 品質とテスト
- ◆ iPhoneのバグから考える

今日のこだわり

今日は、



- 全体（マクロ） - 部分（ミクロ）
 - 全体を網羅 - 部分をピンポイントで狙う
 - 目的 - 手段
 - 品質 - テスト
- ※ 多くの場合、実施時に1つしか選択できない
- 3つ挙げ、1つ選ぶと2つ選ばれない
 - 選択には失敗がつきもの
 - やらなくていいかという気持ち
- だから**重要な2つに絞って説明**したい
- 2つ：方向（大小・上下等）が分かる ★
 - 3つ：直線か曲線かがわかる
 - 1つ：コンサルの場（最良の提示）



にこだわってみたい

品質とテストについて考えてみよう!

- ◆ 高品質なんていない? (消費者は求めていない……という人がいる)
 - 今は、「安くて」「すぐ手に入る」ものが求められている
 - ◆ デパートよりスーパーより100円ショップ、回転寿司、牛丼屋、マクドナルド
 - ◆ どんなに高い時計でも時刻は同じ、スポーツカーでも日本では100km/h
- ◆ テストは無いほうがいい? (バグのないものを一発で作れ!)
 - もしバグが無ければ、テストは無駄な作業?
 - ◆ 形式手法(フォーマルメソッド)で、数学的に何とかできないの?
 - ◆ すぐに直せば(もしくは直すサービスを受注すれば)そのほうがいいのでは?

「高品質なんていない」「テストは無いほうがよい」と聞いたときに心に引かれたことを2つ書きなさい。(1分間)

-
-

品質とテストの改善は一人では困難（人を動かす必要がある）

◆ 人を動かすために「××さんが言ってた」と言ってみる

■ 「××さんが言ってた」は、思考停止ではあるが……

- ◆ （自分自身が）すばやい判断をして、すばやく行動を起こすために超有効
→ 全部に対して長考する時間は無いから。自分が何かを判断する時に、お父さん、社長、尊敬する人、権威（自己紹介もその一種）に頼るのはアリ
- ◆ 他人を動かせるか？→No! 「はあ？ ××社長が言ってたからなに??」

◆ 自分自身の思いの強さが人を動かす

■ 心に引っかけたものを核にして理論武装し「**思い**」を作って、説得する

- ◆ たまには王様の気分になりたい、大切に扱われたいなあ
- ◆ よいものに囲まれると心が豊かになる
- ◆ 開発者はすごいけど、神様じゃないからたまにはうっかりミスもあるさ（開発者がミスしてもテストでカバーしたらいいやん）
- ◆ 安心材料が欲しい（自分が納得するためのテスト：自信を持つため）
- ◆ 仕様が違っていたら、要求獲得に失敗していたら形式手法はどうなるの？
- ◆ フォーマルメソッドを言い始めて50年経つ。 明確な基準は作れない

iPhoneのバグから考える



- 土曜日の早朝、5:46に25,877歩
- 5:56に26,405歩
→ 10分間で528歩
- カウントアップが止まらない→対処方法を教えてくれ！ ググっても無い！

2. 品質と改善

- ◆ 品質改善活動の見える化と炎上の気配を捉える方法
- ◆ 品質の改善

品質問題の2つのタイプ

◆ 大量テストで大リコール

■ いわゆる大炎上

- ◆ T自動車の大規模リコール（2009年-2010年）→費用総額は1000億円！
- ◆ M証券が株誤発注→東証に107億円賠償命令（M証券の損失は、407億円）
- ◆ K社、LTE通信トラブルの原因と対策を説明、300億円の設備投資

■ 炎上駆動改革（©根本さん：JaSST2015東北参加者募集中：5月29日）

- ◆ 炎上時こそ改革提案のチャンス！（上記300億円の投資を見よ！）
- ◆ 火消しチーム、××対策本部、なぜなぜ分析（コツ： 「××（時間、良い上司）が無い」→その結果起きている問題と自分で解決できることを考える）

◆ 改善の停滞

■ 始まった改善活動（CMMI、テスト自動化etc.）はいつまで続けるの？

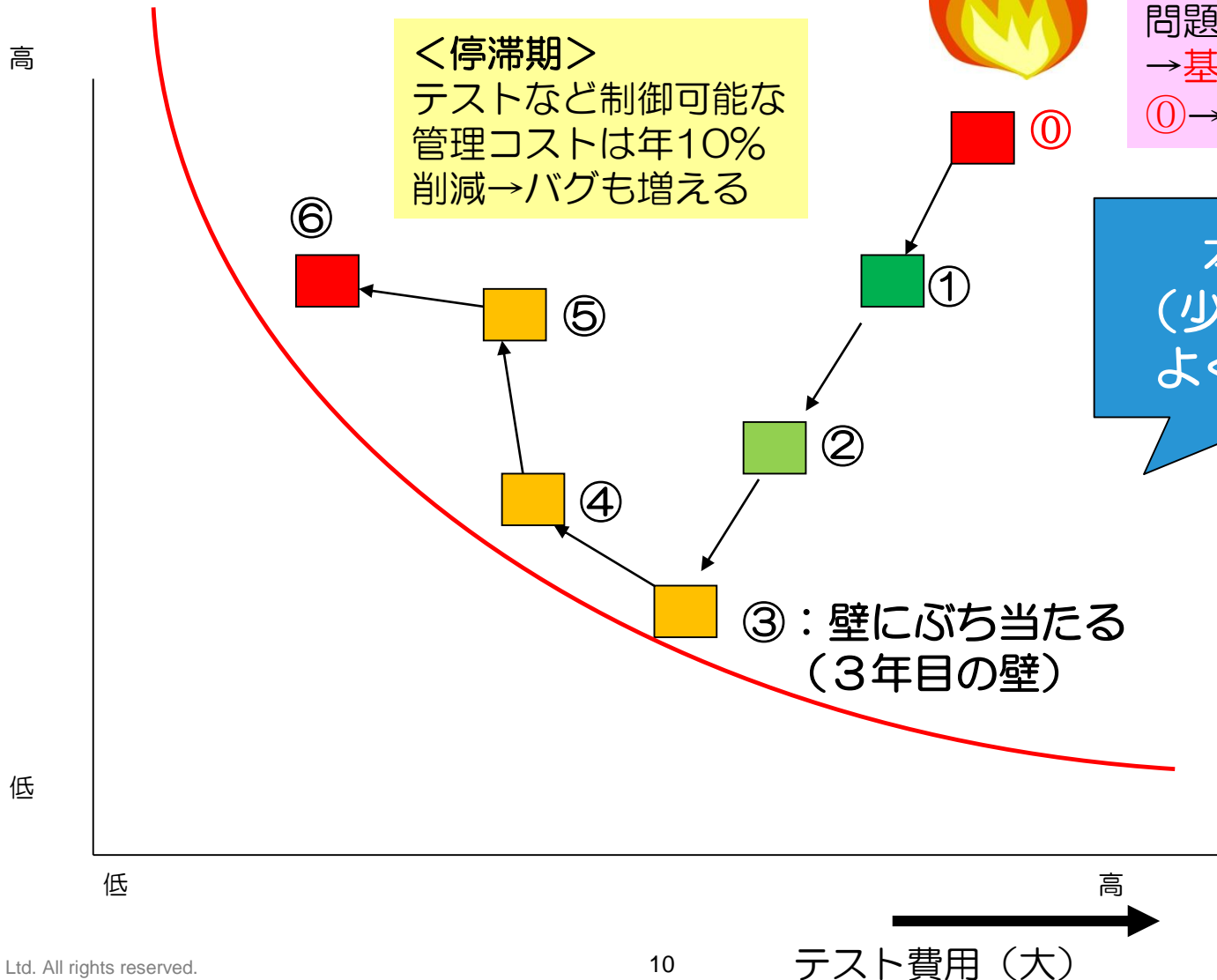
- ◆ 効果があるのに飽きて止めてしまう：『ねばちっこい経営』遠藤 功
 - 改革を継続できる組織的な風土とシステム構築→マクロ的なアプローチ
 - 粘り強い資質を有する人（納豆菌人材）を育てる→ミクロ的なアプローチ
- ◆ 効果が無いのに部門予算は減り続ける→その結果、再び大炎上！（次ページ）

活動状況の見える化と炎上の気配を捉える方法

自分たちの改善活動は成功しているか？

多額な賠償金

品質ロスコスト(大)



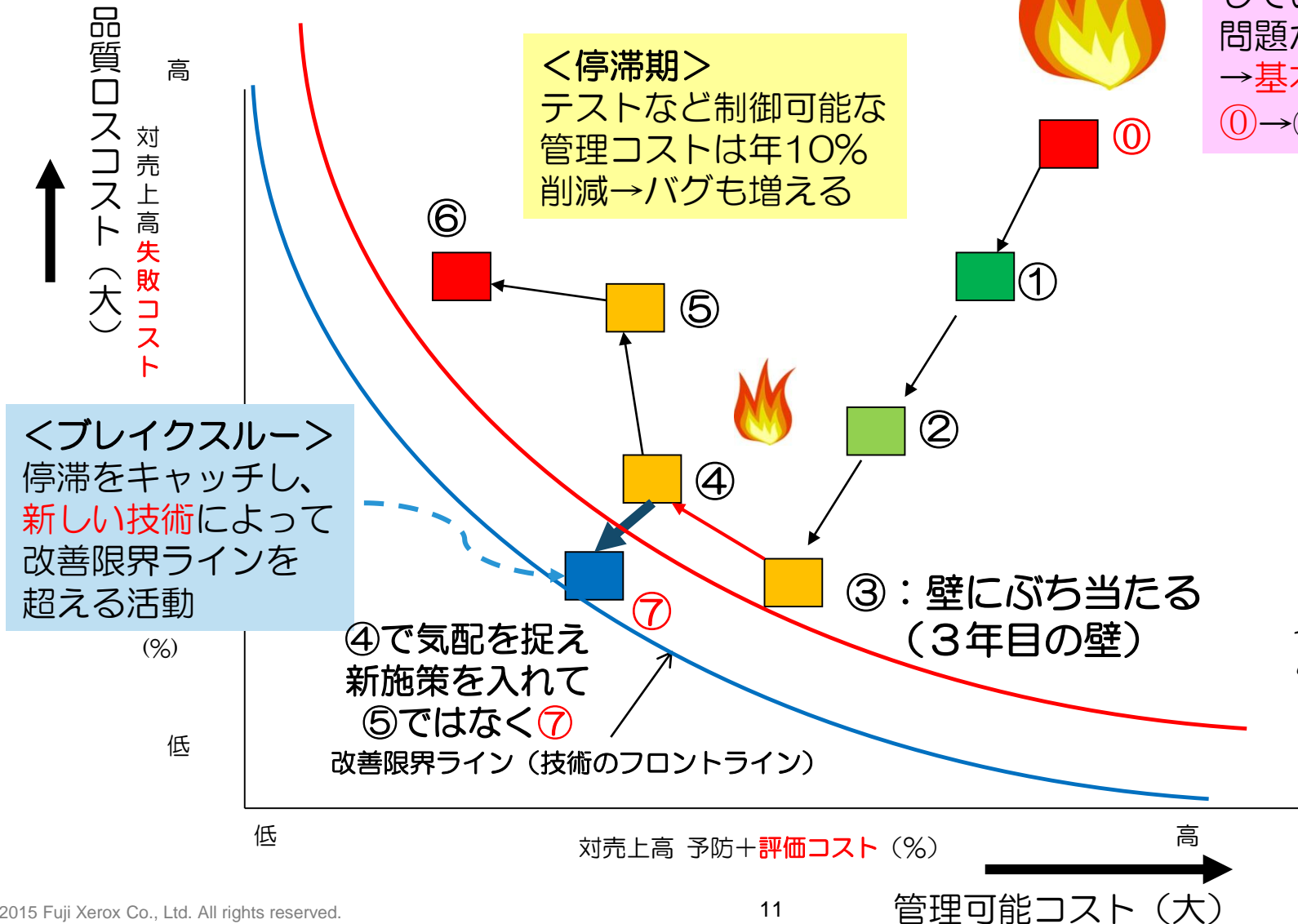
＜停滞期＞
テストなど制御可能な
管理コストは年10%
削減→バグも増える

＜初期＞
テストに莫大な投資を
している割に市場でも
問題が発生している
→基本的な学習により
①→②→③→……

本日の山場
(少し難しいので
よく聞いてね)

活動状況の見える化と炎上の気配を捉える方法

炎上の気配を捉えられているか？
 (管理可能コストの投資で成果は出ているか？)



＜初期＞
 テストに莫大な投資を
 している割に市場でも
 問題が発生している
 →基本的な学習により
 ①→②→③→……



「天を見よ
 見えるはずだ
 あの死兆星が！」
 by トキ

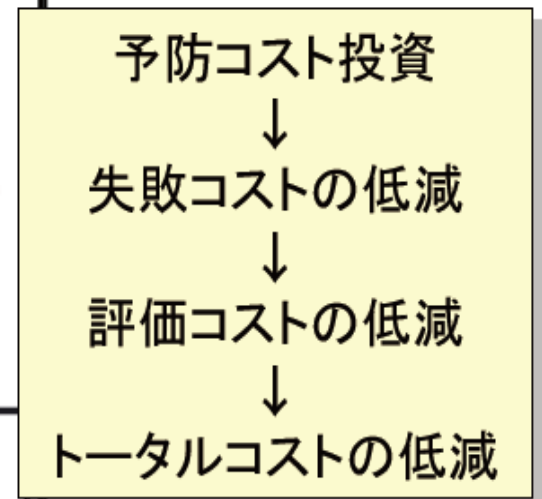
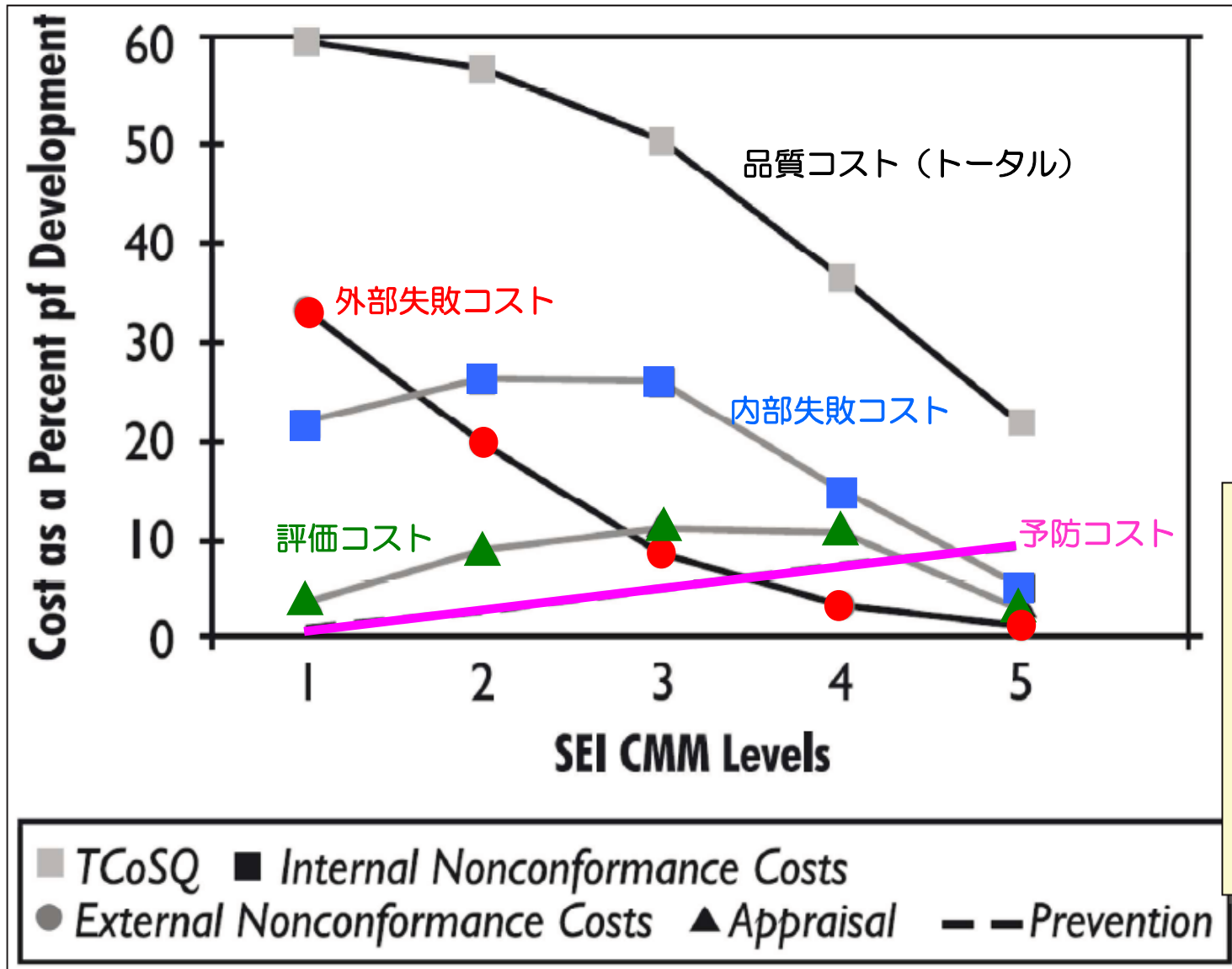
ソフトウェアの品質コスト (Cost of Software Quality)

品質コスト = 予防 (P) + 評価 (A) + 失敗 (F) … 品質への投資額

- 予防コスト～ 品質基準の定義、プロジェクトへの介入：横軸
 - 受け入れテスト時の出荷基準定義、関連する品質標準の定義
 - 教育訓練、プロセス改善、メトリクス収集・分析
 - 評価コスト～ 商品のテスト、品質特性の計測・記録：横軸
 - レビュー、インスペクション、テスト、ソフトウェア品質検査
 - 品質監査、出荷判定
- 監査とは、遵守すべき法令や社内規程などの規準に則っているかどうかの証拠を収集し、その証拠に基づいて何らかの評価をおこなうこと
- 内部失敗コスト～ 出荷前に検出された品質問題：縦軸
 - 出荷前の欠陥 (Defect) 管理
 - 手戻り (再現確認、バグ修正、再設計、再レビュー、再テスト/回帰テスト)
 - 外部失敗コスト～ 出荷後に検出された品質問題：縦軸
 - サポート (コールセンター)、調査、公表 (行政届出)、補修更新/部品交換
 - 再現確認、バグ修正、再設計、再レビュー、再現しないこと/回帰テスト

このような分類の手法をPAF アプローチ (Prevention-Appraisal-Failure approach : 予防・評価・失敗アプローチ) という

品質の改善 = 技術とプロセスとマネジメントの改善 ですから



Krasner, H., "Using the Cost of Quality Approach for Software," CrossTalk, Nov. 1998, pp.6-11.

品質の改善の現場でいつもしていること

◆ 致命的なバグ

- ロケットが軌道を外れる、人の財産や時間や命を奪う（火災、金額ミス）
- 新聞などに載ってブランドを傷つけるもの、省庁への報告（消費者の窓）
 - ◆ バグを作らない
 - ・ 静的解析ツール（特にセキュリティ系）と、アーキテクチャ（不正、暴走）
 - ◆ バグを見つける（あるいは、バグのないことを保証する）：次ページ
 - ・ 保証するケースを決める（D-CaseやGQM+Strategieといったゴール指向）
 - ・ テストで網羅する（C1カバレッジ、1-スイッチテスト、ペア網羅）

◆ 致命的ではないバグ

- 回避策がある、軽微で業務（の本筋）を妨げない
 - ◆ 出ても許される。バグが無いことよりも「（報告したバグが）早く直すこと」
 - ◆ ソフトウェアの自動更新によるコスト低減
 - ・ 「2015年3月29日（日）の0時過ぎに、液晶テレビ「x x」で画面が突然消え、数分ごとに再起動と暗転を繰り返す障害が全国規模で発生」というトラブルが発生
 - ・ 「本障害は、オンエアダウンロード放送に当社のものではないデータが影響したときに発生する」と発表（同日17時、120機種、約168万台）

GQM+Strategy

◆ 大きな改善をする

- G：問題の大きさ
- Q：問題のリスト
- M：何で制御するか
- S：対策（技術、プロセス、マネジメント）
 - ◆ GQM+Strategyで整理すると説得しやすい

◆ 改善しない

- したいこと（使う機能）が動けば良い　：　コンテキストドリブン
- する気もないところ（使われないところ）にバグがあってもいいじゃないか
 - ◆ 心の底から無くなって欲しいこと(問題)か？
 - 戦略（戦うところ、省略するところ）
 - 完璧が必要か？

3. テストの現場

- ◆ 機能と仕様
- ◆ いつものテストと改良ポイント

機能と仕様

◆ 機能 (function)

■ モノの働き

- ◆ 人が作り出す機能は必ず目的 (価値) を持つ
(しかし目的が仕様書に書かれるとは限らない)
 - ・ 利用者の「～したい」 (要件) を実現したものがフィーチャー (feature)
 - ・ フィーチャーの中で動きそのものが機能、どのように働くかは非機能

◆ 仕様 (specification)

■ 規定された条件で働くこと

- ◆ 仕様には、動作条件や入力値の範囲といった機能が働く条件が書かれる
- ◆ テストのコツは、モデルルーム見学会のコツと同じ

■ 関係者がレビューして「コレを作ればイケる」と同意したもの

- ◆ 何をもとに仕様を作ったか確認すること (本当に要求?) 詳細は2ページ後
- ◆ 誰がレビューしたか確認すること (ステークホルダー、ペルソナ)

いつものテストと改良ポイント（機能のテスト）

◆ 機能仕様に対するテストを作成（function specification）

- お客様が欲しいものを一つずつ確認しているので基本的にOK
 - ◆ 仕様書に赤ペンで確認ポイントを書きこんでテスターに渡す
 - ・ テスターは探索的テスト（妥当性確認、BMWの間欠ワイパー）
- ※ 問題となるのは、テスターのスキルに依存することとテストの再現性

◆ 改良ポイント

- テストを上手く分割する
 - ◆ テスト対象を小さく分解する（Test object → Test item）
 - ・ 目的機能/ユーザーストーリー/featureで区切る
 - ・ INVEST（Independent/独立, Negotiable/交渉可, Valuable/価値がある, Estimable/見積可能, Small/適切な大きさ, Testable/テスト可能）
 - ◆ テスト目的を小さく分解する（何でもシナリオテスト？）詳細は2ページ後
 - ・ 品質特性（ISO 9126, 25010標準の読み変え。JIS X 25010）
 - ・ ユーザーの声をよく聞く（自分で使う、良いものと悪いものを知る）

いつものテストと改良ポイント（機能のテスト続き）

◆ 機能仕様に対するテストを作成（function specification）

- お客様が欲しいものを一つずつ確認しているので基本的にOK
 - ◆ 仕様書に赤ペンで確認ポイントを書きこんでテスターに渡す
 - ・ テスターは探索的テスト（妥当性確認、BMWの間欠ワイパー）
- ※ 問題となるのは、テスターのスキルに依存することとテストの再現性

◆ 改良ポイント

- 仕様から要求を考え、他に仕様がないか考える
 - ◆ 仕様から要求を復元する（要求→妥当性確認ができる！）
 - ・ 要求といいながら仕様が書かれている（その通りに作るから仕様が漏れる）
 - ・ 一度、仕様から要求を考える
 - ◆ 復元した要求から仕様を考える
 - ・ テストでは要求を正しく認識し、テストする
 - ・ そのための分析が大切

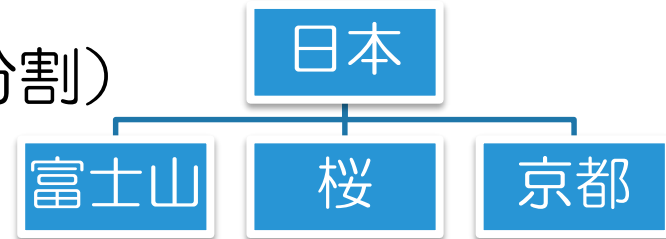
いつものテストと改良ポイント（分析）

◆ MECE？（仕様書の目次などでなんとなく分割）

■ テスト観点をマインドマップで分解する

◆ 日本について教えてください

- Please tell us about Japan
- 请您介绍一下日本（中国）
- Tafadhali tueleze kuhusu Japan（スワヒリ）
- 일본은 무엇입니까?（韓国語）



◆ 改良ポイント（テストの切り口で分割→構造・特性→関係性を考える）

■ 日本の分解（関係には、因果関係と単純相関、トレードオフ関係がある）

- ◆ 日本と言っても広い→構造を分ける（地域：4つの島、地方、県、市、、、）
- ◆ 日本の何を知りたい→属性（特性）を分ける（歴史、気候、文化）←重要

- Wikipediaで国の説明は、地理（地形、気候）、歴史、政治、経済（農業、工業、商業）、軍事、交通（道路、鉄道、海運、空運）、国民性（言語、宗教、教育、結婚）、文化（文学、音楽、美術、映画）、スポーツ（野球、サッカー、バスケット）、科学技術（物理、化学、医療）などに分解

FUJI Xerox

