

JaSST'13 Hokkaido Workshop  
TEF道 Presents

## テストとメトリクスの 素敵なマリアージュ

初心者～中級者向けワークショップ

### このワークショップでは

- ②典型的なメトリクスの使い方や、メリット・注意点を、演習を通じて学んでいただきます。
- ③メトリクスをテストのインプットに活用する演習を通じて、メトリクスの使い方を実践し、その効果を実感していただきます。

JaSST'13Hokkaido ©TEF道

1

### TEF道紹介

正式名称：TEF北海道ソフトウェアテスト勉強会 Testing Engineer's Forum HOKKAIDO  
会費は無料、ほぼすべて弁当で対応。隔週で一回2時間程度の開催。  
たまに公開ワークショップも合宿。じょっちゅう飲み会。他地域勉強会連携など。  
参加メンバーは10名程度で、當時5～6名のメンバーが参加。

【これまでの戦績】  
2006年～2008年 JSTQBテスト勉強会として発足。断続的な活動。  
2009年 JaSST北海道にて成果発表  
2010年 JaSST東京に事例発表、JaSST北海道でワークショップ実施  
2011年 HOTATEワークショップの不定期運営開始  
JaSST北海道にて初心者向けワークショップ実施  
2012年 JaSST東京にて  
TEF道 派生ユニット“チームYuki Da RMA”にて「テスト設計コンテスト」出場  
派生分科会、USDM分科会を発足し、要求仕様に関する勉強を開始。  
2013年 JaSST東京にて “チームYuki Da RMA”「テスト設計コンテスト」準優勝  
書籍「ソフトウェアテスト技術ドリル」を使った勉強会を開催中（ほぼ毎月）！！

JaSST'13Hokkaido ©TEF道



<http://ameblo.jp/tef-do/>  
tefdoblog@gmail.com

2

### メトリクスとは？

- ・ソフトウェア開発で、ソースコードの品質を数値化して定量的に評価することや、その際の評価手法や基準などの体系のこと (IT用語辞典 e-Wordsより)

JaSST'13Hokkaido ©TEF道

3

### 簡単に言うと・・・

- ・「メトリクス」 = 「指標」
  - ・物事を判断したり評価したりするための目じるしとなるもの (デジタル大辞泉より)
- ・「目的」の為に集められる、「観点を持った」データの集計方法と言える
- ・例)
  - ・【目的】システムの品質を確かめたい
  - ・【データ集計方法】バグの数をカウントする
    - ⇒これを「バグ数」という名前のメトリクスと定義する
  - ・【活用方法】
    - ・バグ数の大小で、品質の良し悪しを判断する

JaSST'13Hokkaido ©TEF道

4

### もっと簡単に言うと・・・

- ・プロジェクトやプロダクト状況を「数値化」すること (定量化)
- ・定量化する事により、客観的な状況分析ができる
  - ・「バグがたくさん出ているので大変です！」よりも「バグが〇〇件出ていますので、大変です！」の方が、報告としての価値が高い

JaSST'13Hokkaido ©TEF道

5

## メトリクス例

- 成長オーダー (アルゴリズム解析、O記法など参照)
- ソースコードの行数
- 循環的複雑度
- ファンクションポイント法
- ソースコードの行当たりのバグ数
- コード網羅率
- 顧客要求仕様の行数
- クラスおよびインターフェースの個数
- Robert Cecil Martin のソフトウェアパッケージ測定法
- 凝集度
- 結合度

(Wikipedia「ソフトウェア測定法」より)

JaSST'13Hokkaido ©TEF道

6

## 「メトリクス」 クイズ! ①

とあるシステムについての想定バグ数と、実際にテストチームが発見したバグ数は以下でした。この結果から言えることは？

|          |      |
|----------|------|
| 想定 発見バグ数 | 100件 |
| 実績 発見バグ数 | 5件   |

- システムの品質がとても良かった
- テストチームのテスト技術不足
- バグ数の情報だけでは、①とも②とも言える

JaSST'13Hokkaido ©TEF道

7

## 「メトリクス」 クイズ! ②

プログラム品質改善のため、開発仕様書レビュープロセスを導入し、レビュー実施時間とバグ率のメトリクスは以下でした。この結果から言えることは？

| プロジェクト名称 | レビュー時間 | バグ率  |
|----------|--------|------|
| プロジェクトA  | 10時間   | 5.0% |
| プロジェクトB  | 20時間   | 2.5% |
| プロジェクトC  | 50時間   | 1.0% |

- レビュー時間とバグ率には相関関係が無く、レビュー時間を増やしても、バグは減らない
- レビュー時間とバグ率には相関関係があり、レビュー時間を増やすほど、バグは減る。
- レビュー実施とバグ率の直接の因果関係は、このメトリクスだけでは分からない

JaSST'13Hokkaido ©TEF道

8

## メトリクス分析 例題

とあるシステムのバージョンごとの発生バグ件数のグラフがあります。上司はV18～V20にて実施したレビュー改善の効果がでて、バグ件数がV21から少なくなったと言っています。

これは本当でしょうか？本当かどうかを確かめるためには他にどのようなメトリクスが必要でしょうか？



JaSST'13Hokkaido ©TEF道

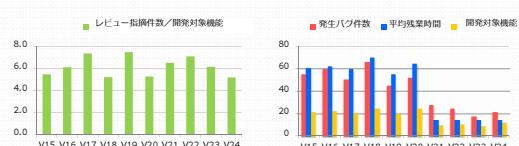
9

## メトリクス分析 例題 回答例

### 必要なメトリクス

レビュー指摘件数 / 開発対象機能数

レビュー指摘重要度の割合 等々



実際は…  
残業規制が掛かっていて、1バージョンあたりの開発対象機能数が半分になっていた。

10

## メトリクスを収集・分析するときの失敗リスク

- 誤ったデータ集計により、誤った判断や報告をしてしまう。
- 誤ったメトリクスの認識により、誤った判断や報告をしてしまう。
- 恣意的なデータ集計により、偏った意図が働いてしまう。
- メトリクスへの過剰な信頼により、プロジェクトが非効率化する。

JaSST'13Hokkaido ©TEF道

11

## メトリクスを収集・分析する メリット

- プロジェクトの正しい情報を得ることができ、プロジェクトの問題を早めに特定して、是正できる
- ステークホルダに、客観的な情報を提供できる
- 重要なトレードオフの客観的判断を下す事ができる
- 判断に対しての客観的正当化が行える

JaSST'13 Hokkaido ©TEF道

12

## b.プロジェクト背景

- 対象製品は、「湯沸しポット」
- 機能テストとシステムテストまでは完了。
- しかし、まだ製品の品質に不安があるため、弱点部分の「品質強化試験」を実施したい。
- システムテストまでのテスト結果・分析結果をもとに弱点を探してピンポイントで突く！
- そのテスト計画を策定中。。

JaSST'13 Hokkaido ©TEF道

14

## JaSST'13 メトリクスWorkshop

- ✓ グループワーク①  
・・・方針のレビュー
- グループワーク②  
・・・テスト設計の検討

JaSST'13 Hokkaido ©TEF道

16

## a.ワークの全体像

①ワークの説明

間違ったメトリクス分析をレビューして、正しい分析とテスト計画へと修正するワークです。

- シチュエーション。
  - システムテスト後の強化試験の計画中
- みなさんに取り組んでいただくこと。
  - ダメ社員のダメな分析結果をレビュー。
  - メトリクスを分析し、テスト計画・分析を修正する。

JaSST'13 Hokkaido ©TEF道

13

## c.「品質強化試験」の方針

①ワークの説明

- システムテストまでの各種メトリクスより、プロダクト品質の弱点とリスクを分析する
- 分析結果より、テスト方針を作成
- 方針を元に、実施済みテストからの【抽出】や、【追加作成】を行う

⇒ダメ社員が頑張ってやってみたので、上司（皆さん）でレビューをするところからワークが始まります。。

JaSST'13 Hokkaido ©TEF道

15

## グループワーク①-1

グループワーク①

«あなた方は、Aさんの上司です»  
«計画を承認する立場です»

Aさんが作った「メトリクス分析結果」のレビューをお願いします。

JaSST'13 Hokkaido ©TEF道

17

## グループワーク①-2

グループワーク①

- レビュー観点

- 分析が間違っていると思われる点
- 分析が足りないと思われる点
- 他の解釈が可能と思われる分析

JaSST'13Hokkaido ©TEF道

18

## グループワーク①-3

グループワーク①

- レビュー対象

- メトリクス資料 (オープンクローズチャート: バグ数の積み上げ推移／テスト残数の推移)

- 余力があれば、他のメトリクスのレビューも行ってみてください

JaSST'13Hokkaido ©TEF道

19

## ヒント

グループワーク①

- まず分析結果を読む。（いきなりメトリクスを読まない）
- 最初はメトリクスの細かい所を気にしない
- ずっとメトリクス全体を眺めて感覚的に「気になる」箇所にチェックを入れる
  - グラフの形に違和感がある／不自然である
  - 突出した数字がある
  - 「重要な項目」を特定して、注視する
- 「気になった箇所」の詳細を他のメトリクスなどで内容を確かめる

JaSST'13Hokkaido ©TEF道

20

## グループワーク①のまとめ・講評

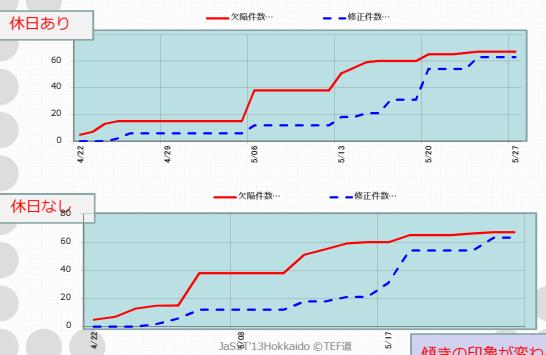
グループワーク①

グループ発表と講評

JaSST'13Hokkaido ©TEF道

21

## 例：休日を抜かしてみると…



傾きの印象が変わる

## メトリクスの分析方法（提案）

グループワーク①

- まずは全体を把握して、「気になる箇所」を特定した後、そこから詳細分析を行う
  - いきなり、アテもなく詳細分析を行うと大変…
  - たとえば、三色ボールペンなどでチェックを行う
- 「詳細分析」を行った結果から「仮説」を立てて、他の手段にて検証を行う
  - たとえば補完する他のメトリクスなど
  - 仮説が正しいかどうか
  - 単一メトリクスで判断するのは危険

JaSST'13Hokkaido ©TEF道

23

## JaSST'13 メトリクスWorkshop

- ✓グループワーク①
  - ・・・方針のレビュー
- ✓グループワーク②
  - ・・・テスト設計の検討

JaSST'13Hokkaido ©TEF道

24

## グループワーク②-1

Aさんはギブアップしてしまったので、みなさんが【グループワーク①】（レビュー結果）の結果と各種メトリクスを利用して、テスト設計を見直してみましょう。

JaSST'13Hokkaido ©TEF道

25

## グループワーク②-1

- 品質強化試験を行う上で、重要と思われる仕様に重要度（◎、○、△、×）を記入してください
- 余力があれば、「機能テスト」をやるのか「システムテスト」をやるのかなどの「テストカテゴリ」や「テスト条件」なども考えてみてください

JaSST'13Hokkaido ©TEF道

26

## ヒント

- 欠陥重要度ランクAのバグが出た機能に起こりえることは・・・？
- 修正のオーブン日数が大きいと何が考えられる・・・？
- プロジェクト状況を考えて、エンジニアに何が起こっているかを想像してください
- システムテスト突入の判断は正しかったのか？
- そもそも、品質強化試験開始が果たして出来る状況なのか？

JaSST'13Hokkaido ©TEF道

27

## グループ発表

グループワーク②

### グループ発表と講評

JaSST'13Hokkaido ©TEF道

28

## たとえばこんな感じ

- 欠陥重要度ランクAのバグが出た機能に起こりえることは・・・？
  - ・デグレードのリスクがある
- 修正のオーブン日数が大きいと何が考えられる・・・？
  - ・修正に時間がかかる⇒難易度が高い⇒デグレードリスクが・・・
- プロジェクト状況を考えて、エンジニアに何が起こっているかを想像してください
  - ・急激に慌てて修正されている⇒ちゃんと修正されていない？
- システムテスト突入の判断は正しかったのか？
  - ・安全性試験と負荷試験は、再度実施が必要？

JaSST'13Hokkaido ©TEF道

29

## グループワーク②のまとめ

- メトリクスを正しく収集分析すると、プロジェクトの意思決定に有用な情報となる
  - たとえば、品質状況の把握やテスト方針決定
  - 前工程の状況や、前回プロジェクト状況や、類似プロジェクトとの状況が参考になる
  - プロダクトメトリクスとプロジェクトメトリクスの両面から考える
  - 意外なメトリクスも参考になる場合がある

JaSST'13Hokkaido ©TEF道

30

## 全体まとめ

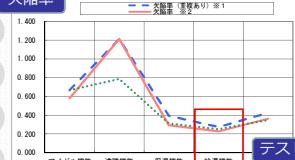
- メトリクス分析により、「気になる箇所」の特定が素早く行える
- 正しいメトリクス分析により、プロジェクトへの有用な情報を提供できる
- 単一メトリクスだけではなく、複数のメトリクスで補完して分析しよう
- 仮説を立てる→メトリクスにて検証の流れ
- メトリクスに振り回されない（メトリクス収集を目的としない）

JaSST'13Hokkaido ©TEF道

31

## 複数のメトリクスで分かること

欠陥率



給湯機能のバグ率が低く、品質が良いように思えるが……

無駄なテストケース数が増えると、今回の定義の「欠陥率」では、欠陥率がどんどん下がっていく！

テスト密度が突出している⇒冗長なテストを実施しているのではないか？



## よくよく調べてみると・・・

仕様に対して、テストケース数が多すぎる？（全組合せを実施している）

| 8組<br>給湯<br>機能 | 2.6給湯<br>ボタン | 給湯ボタン<br>異常時   |
|----------------|--------------|--|
|                | 260-11       | ■熱湯によるやけど・空焚き・電源異常時の使用による事故を防止するため<br><ボタンを作動できる場合>以下の条件を全て満たすとき、給湯ボタンを作動して1回、ポンプを作動する<br>→(1)ポンプを作動する<br>(2)各条件を満たしていない場合は給湯できない。<br>・ロック解除状態であること<br>・水温が適正であること<br>・蓋センサがonであること<br>・ヒーター用電源がonであること<br>・温度エラーを検知していないこと<br>・保温行為中であること |

JaSST'13Hokkaido ©TEF道

33

## 今回の分析で足りないもの

- 今回のリスク分析は「品質リスク」のみです。（対象システム自体の品質リスク）
- 「ユーザーにとって不利益な問題が起きてはいけない」という「ユーザーリスク」の分析も合わせて考慮してください
  - たとえば、ユーザーへの影響（直接的危険性／経済的損失）が大きい機能は重要

JaSST'13Hokkaido ©TEF道

34

ありがとうございました

資料に対する問い合わせはこちら  
tefdoblog@gmail.com

JaSST'13Hokkaido ©TEF道

35