

レビューの壁を破る

国立大学法人 奈良先端科学技術大学院大学 情報科学研究科

森崎 修司

smrs@is.aist-nara.ac.jp

本資料に含まれる研究の一部は文部科学省「次世代IT基盤構築のための研究開発」、同省「eSociety基盤ソフトウェアの総合開発」の委託に基づいて行われたものです。また、本資料に含まれる研究の一部は同省科学研究補助費(若手B:課題番号21700033)の助成を受けました。

本セッションの概要

- メンバの紹介
- 本セッションの主旨、概要説明、注意事項

- 第1部: レビューの課題の列挙・共有
 - 課題(当学で実施したアンケートより)
 - 会場からのご紹介
 - 課題解決に向けた施策検討(できる範囲で)

- 第2部: レビューとテストのバランス
 - レビュー完了基準
 - 事例
 - 優先型レビュー
 - 不具合管理票の分析事例
 - 会場からのご意見とディスカッション

本セッションの背景と主旨

- 背景
 - ソフトウェアレビューの課題や問題に関する公開情報は不十分
 - 有効に機能すると期待できる事例や理由を知っているが、常に有効に働いているわけではない。
- 主旨
 - ソフトウェアレビューに関する課題の共有
(会場からご紹介いたします)
 - 課題の解決にむけた議論
 - 参加者それぞれが何らかの気づきを得ることが目的
- 発展
 - ソフトウェアレビューとソフトウェアテストの関係について問題提起

本セッションでのお願いと注意いただきたいこと

• お願い

- 様々なご意見をお願いします。
- 開発の事情と事情にもとづいたご意見をお願いします。
 - ソフトウェア開発は多岐にわたります。なるべく、ご自身の開発の特徴を明らかにしていただいた上で、うまくいかないこと、うまくいっていること、をご意見ください。
 - 他のプロジェクトでうまくいっている理由がわからなかったり、奇異と感ずることがあっても、すぐには否定しないでください。
- リアルタイム中継はお控えください。

• ご注意いただきたいこと

- ご発言、ご意見は最終的にWebで公開したいと思っております(ご所属、ご氏名は削除し、必要なものは表現をマイルドにします)。
- レビューを良くするための議論をするためのセッションです。
- 参加者の発言は必ずしも組織全体を代表しているわけではありません。

本セッションの概要

- 本セッションの主旨、概要説明、注意事項
- 第1部: レビューの課題の列挙・共有
 - 課題(当学で実施したアンケートより)
 - 会場からのご紹介
 - 課題解決に向けた施策検討(できる範囲で)
- 第2部: レビューとテストのバランス
 - レビュー完了基準
 - 事例
 - 優先型レビュー
 - 不具合管理票の分析事例
 - 会場からのご意見とディスカッション

ソフトウェアレビューの課題(例)

- コミュニケーション・参加者の合意形成
 - レビュー効果の共有が難しい(形骸化)。
 - 指摘方法の難しさ
- コスト・リソース不足
 - 実施工数確保が難しい。
 - スキルの高いレビューアの確保が難しい。
- 技術・制度面
 - 実施方法・手順の未整備
 - 技法の不在
 - 助け合い、レビュー指摘の評価制度がない。

お持ちの課題を紹介ください

- 本資料の課題に追記したいと思います。
- ご自身に関わるプロジェクトや組織の事情と、課題をできるだけ結びつけてご発言ください。
- 他の方の課題、ご意見に対する反論は後からいただきますので、まずは課題の列挙、ご意見をお願いします。
- **ご注意ください。**
 - 必ずしも全てが反映されるわけではありませんのでご了承ください。
 - 個人、組織の情報を削除の上、公開する可能性があります。

課題解決にむけた意見交換

- 特定の条件下のレビューにだけあてはまるもので構いません。
- 躊躇されている方へ
どのような意見であれ、その意見が呼び水となってより優れた意見のきっかけになる可能性があります。
- 例
 - とにかく横道にそれないような仕組みを
 - 潜在的リスクの高い指摘をいかに指摘できたかをレビューアが自慢できる土壌作りを
 - 観点をしぼり、ライトウェイトなレビューを
 - 指摘終了後、指摘できた欠陥をもし見逃していれば修正にどれくらい工数がかかったかを検討
 - 開始前にチームビルディングを
 - よい部分やマネしたい部分も短時間で賞賛
 - レビューア毎のスキルにあわせた観点を設定

本セッションの概要

- 本セッションの主旨、概要説明、注意事項
- 第1部: レビューの課題の列挙・共有
 - アンケートにもとづく課題
 - 会場からのご意見
 - 課題解決に向けた施策検討(できる範囲で)
- 第2部: レビューとテストのバランス
 - レビューで指摘できる欠陥、問題
 - 事例
 - 優先型レビュー
 - 不具合管理票の分析事例
 - 会場からのご意見とディスカッション

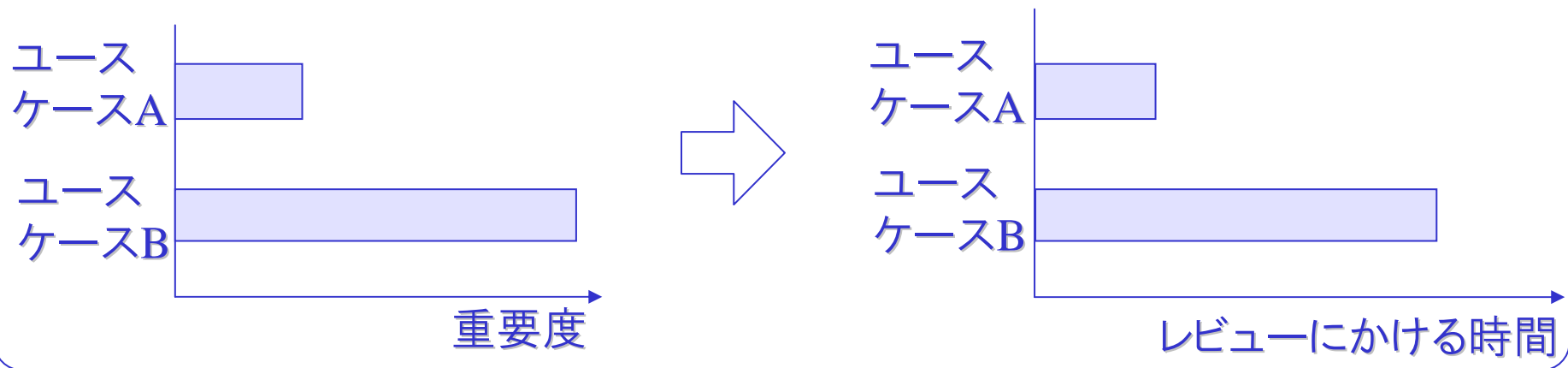
優先度つきレビュー

- レビューで指摘できる欠陥・問題
 - 機能欠陥(正常系、異常系)
 - 性能、セキュリティ、法令等、非機能要件に関する問題
 - 保守、拡張性の問題指摘
- 事例1 時間管理型レビュー
 - ユースケースに沿ってエラー指摘をする。
 - ユースケースの重要度に応じて時間配分(実施順序)を決める。
- 事例2 不具合管理票の修正工数の分析
 - 修正工数が大きいバグや欠陥の優先順位を上げる。
- テストとの連携

Time Controlled Reading

- ユースケースベースドリディングの拡張
 - ユースケースに沿ってエラー指摘を試みる。
 - ユースケースの重要度によって、エラー指摘にかけられる時間を決める。
 - ユースケースの重要度は投票やAHPにより決める。

ユースケースA: カスタマイズした壁紙のダウンロード
 ユースケースB: クレジットカードによる決済



出典: Kai Petersen, Kari Rönkkö, Claes Wohlin, “The impact of time controlled reading on software inspection effectiveness and efficiency: a controlled experiment <http://portal.acm.org/citation.cfm?id=1414004.1414029>
 次にも日本語での簡単な説明があります。森崎 修司, “ソフトウェアインスペクションの動向”, 情報処理学会会誌 Vol. 50, No. 5, pp. 377-384(2009) <http://se.aist-nara.ac.jp/~morisaki/publications/#invitedPaper>

修正工数に関する相関ルールを抽出

- (再現性=低) かつ (機能=A) かつ (テスト種別=異常系)
 → 修正工数(平均 18.6人時 標準偏差 3.21)
 出現頻度: 3/8, 全体平均の1.86倍

再現性	機能	テスト種別	修正工数(人時)
低	A	異常系	20
常に	B	正常系	3
高	B	正常系	4
常に	X	異常系	2
低	A	異常系	15
高	X	異常系	10
低	A	異常系	21
低	A	正常系	5

前提部を含む不具合の修正工数の平均: 18.6

修正工数の全体平均: 10.0

基準化平均 = 18.6 / 10.0 = 1.86

もし相関ルールが事実を表してれば、機能Aの異常系のレアケースを重点的にレビューする。

対象不具合管理票

- プロジェクト

- プローブ情報システムのサーバサイドロジック
(自動車から送られる交通情報をもとに最適経路をナビゲーション:経済産業省「先進的ソフトウェア開発プロジェクト」)
- C/C++で330KLOC(流用込み)、約1年
- 複数ベンダによるウォーターフォール型開発

- 不具合

- 発見フェーズ 単体～総合試験
- 記録不具合件数 約1,300
- 組織内でバグと承認されたもの以外は入っていない。

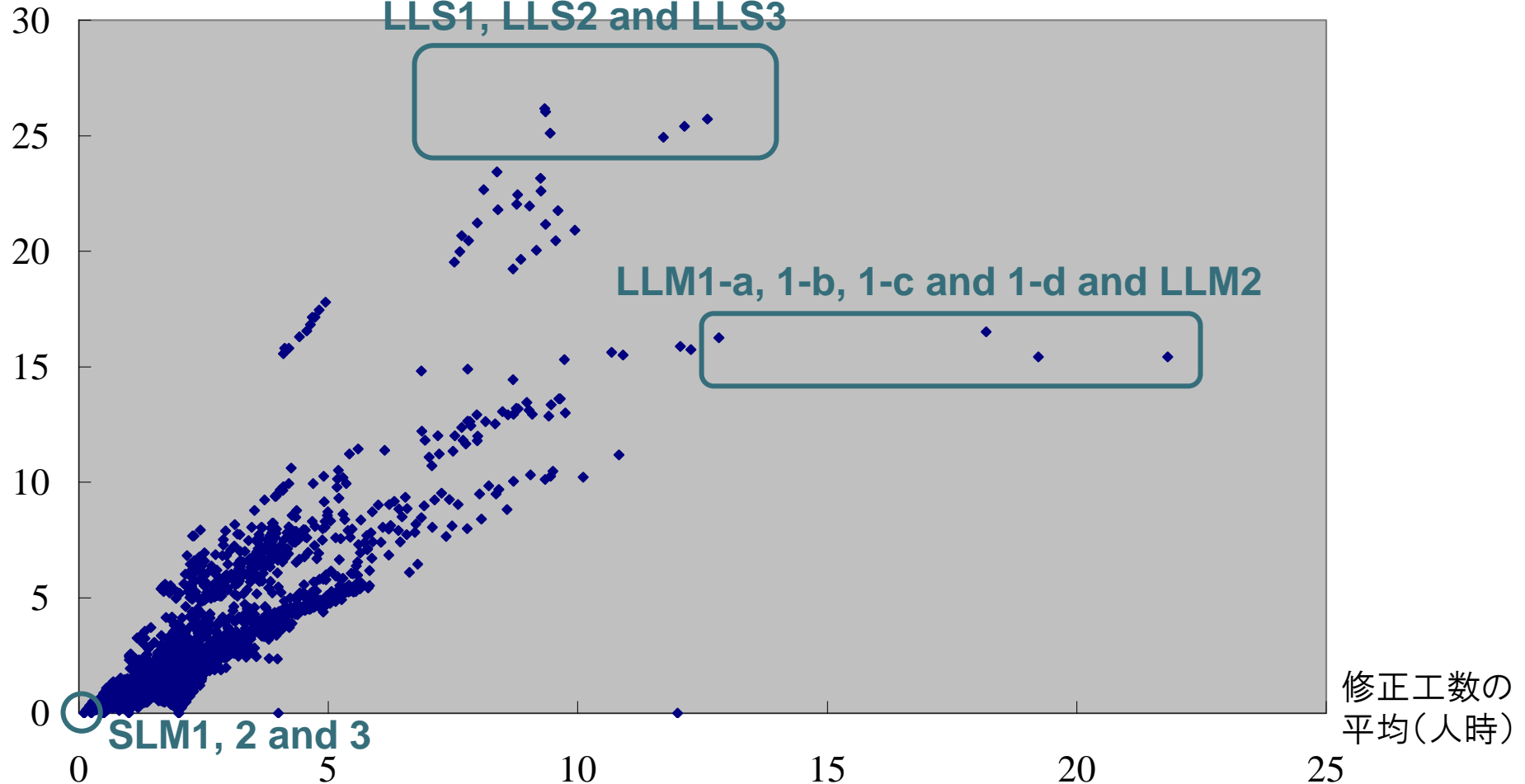
対象とした不具合管理票の項目

変数名	取りうる値
修正工数	修正に必要なとなった工数(人時)
混入工程	混入, 修正, 発見された工程 (基本設計, 詳細設計, 製造/単体試験, 結合試験, 総合試験)
発見工程	
修正工程	
機能	機能名(9機能のいずれか)
発見が遅れた理由	レビュー未実施, レビュー指摘もれ, 修正確認漏れ, 工程間引継ぎ, コミュニケーション不足, 試験項目抽出もれ, テスト計画に含まれていない, 環境が整わずテスト未実施, 結果確認ミス
優先度	高, 中, 低
重要度	高, 中, 低
再現性	常に, たまに, 一度だけ

出典: 森崎、門田、玉田、松村、松本: “Defect Data Analysis Based on Extended Association Rule Mining”, Proceedings of International Workshop on Mining Software Repository, pp.17-24. <http://se.naist.jp/~morisaki/publications/#i-200705>

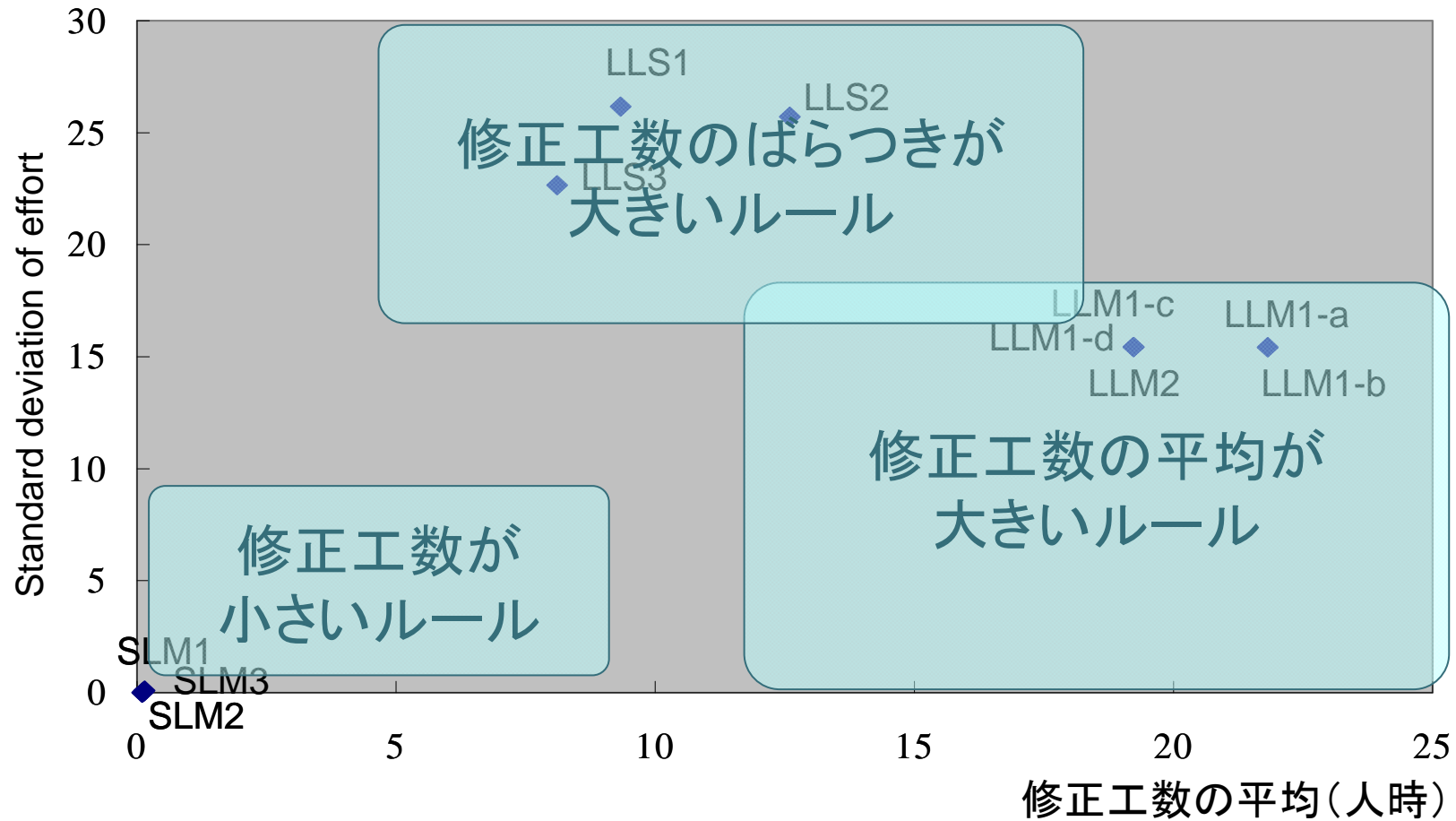
抽出ルール₁の分布(出現頻度 0.5%以上の17,000ルールを抽出)

修正工数の標準偏差



出典: 森崎、門田、玉田、松村、松本: “Defect Data Analysis Based on Extended Association Rule Mining”, Proceedings of International Workshop on Mining Software Repository, pp.17-24. <http://se.naist.jp/~morisaki/publications/#i-200705>

修正工数の標準偏差



レビューとテストのバランスに関する議論

- そもそも相補的に考えることができるのか？
→ できないとしたらどのような場合か？
- 役割分担可能か？
 - レビューでしか修正できないもの
 - レビューでは修正時間がかかりそうなもの、テストでは修正時間が短くてすむもの
- 優先順位付けの方法？
- レビュー開始条件、終了条件？
- レビューでもテストでも発見できない欠陥？

など、ご意見お願いします。

まとめ

- レビューの課題を挙げ、参加者で共有した。
 - 当学で実施したアンケート
 - 会場からの課題、意見
- 課題にむけた改善検討をした。
- レビューとテストのバランス
 - 事例紹介
 - 会場から

- レビューの課題解決の協力をお願い
- 奈良先端科学技術大学院大学発 ソフトウェアレビュー／インスペクションに関する情報(つぶやき)
http://twitter.com/sig_inspection

オンラインでコードレビューの腕試し(今すぐ参加可能)

- 奈良先端科学技術大学院大学では、ソフトウェアレビューの研究結果をフィードバックすることを目的とした特徴分析を実施しています。
- 保守、派生開発のソースコードの差分が持つ特徴と人間の理解に必要なコストとの関係を分析をしようとしています。
- 実務者の方の評価(難易度、理解の所要時間)を募集しています(ネットワーク経由で実施できるJavaソースコードの目視です)。
http://se.aist-nara.ac.jp/html/review/code_review_cost_estimation/html/
- 結果は個人情報となるべく含まない形で回収し、残っている情報は削除の上、公開します。