

B5) テストマネジメント ～確実に間違ふよりも漠然と正しく～

データ指向のソフトウェア品質マネジメント ーテスト編ー

東洋大学 野中 誠

ヤマハ株式会社 小池 利和

2014年3月8日

JaSST'14 Tokyo

• 所属

– 東洋大学 経営学部 経営学科 准教授

• 背景

– 工業経営／経営システム工学, ソフトウェア工学, 品質マネジメント

• 主な学外活動

- JaSST'14 Tokyo 実行委員
- 日本科学技術連盟 SQiPソフトウェア品質委員会 運営委員長
- IPA/SEC 高信頼化定量管理部会主査(『ソフトウェア開発データ白書』など)
- 日本SPIコンソーシアム 外部理事
- 組込みシステム技術協会実装品質強化WG メンバー
- 国立情報学研究所 特任准教授 (トッペスイー講座, メトリクス講義担当)

• 主な著書

- 野中・小池・小室著『データ指向のソフトウェア品質マネジメント』日科技連出版社 (2012)
(2013年度日経品質管理文献賞受賞)
- 野中・鷲崎訳『演習で学ぶ ソフトウェアメトリクスの基礎』日経BP社 (2009)
- SQuBOK策定部会編著『ソフトウェア品質知識体系ガイドーSQuBOK Guideー』オーム社 (2007)

• 研究教育

- ソフトウェア品質マネジメント(メトリクスを中心に)
- 経営と情報システムとの関わり



野 中 誠



データ指向のソフトウェア品質マネジメント^①(データ本)

メトリクス分析による「事実にもとづく管理」の実践

2013年度日経品質管理文献賞受賞



2012年11月8日号

p.125 書評

ソフトウェア品質メトリクスで、日本における気鋭の研究者である野中誠氏による解説書。

ソフトウェアの品質データ(メトリクス)は、その収集自体が目的化してはならないが、その分析は**開発現場ではなくてはならない**ものである。

本書はその収集・分析方法、用いるべきモデルなどを、統計学の初歩的解説とともに述べている良書である。



2012年11月号

p.110 書評

ソフトウェアの品質を高めるには、開発の過程で問題をいち早く把握し、適切な対策を打つことが不可欠である。

それには、**問題を可視化**することが大事だ。

本書は、事例を基に、品質データの測定項目や分析手法を解説している。ソフトウェア品質に関わる多様なデータを測定し、分析を繰り返すことは容易ではないが、分析手順を詳しく説明している。

事例を読者自身の開発に当てはめて実践してみれば、必要な知識やスキルを身に付けやすいだろう。

- ・品質メトリクス分析の実践的入門書
- ・**11**の分析事例
- ・データと分析手順の解説をダウンロードできる(同じ分析手順を手元で行える)

主要目次

- 第1章 品質データ分析の基本
- 第2章 品質状況の把握
- 第3章 影響要因の把握
- 第4章 静的予測モデルの構築
- 第5章 動的予測モデルの構築
- 第6章 測定方法
- 付録

野中 誠(東洋大学)

小池 利和(ヤマハ)

小室 睦

(元 日立ソリューションズ、
現 富士フイルムソフトウェア)

2012年9月19日発行

定価 3,780円(税込)

ISBN978-4-8171-9447-3



講演概要

テストに「事実に基づく管理」を導入し、継続的に改善していくには、
テスト状況をデータで把握したり、**改善の着眼点**をデータで示したりすることが求められます。

しかし、このようなデータの活用は必ずしも容易ではありません。

本セッションでは『データ指向のソフトウェア品質マネジメント』の著者である2人が、
「事実に基づく管理」をテストに持ち込もうとして試行錯誤をした過程の一端をご紹介します。

具体的には、

- バグの収束状況をデータで判断するにはどうすればよいか
- 適正なテストリソース計画立案とコントロールのためにデータを如何に活用するか
- 単体テストでの欠陥見逃しの傾向をどのようにデータで示せばよいか
- 第三者テストで検出される欠陥数に影響している要因は何か

野中

小池

という問題について、2人が取り組んだデータ分析・活用事例をお話いただきます。

本セッションが、

「事実に基づくテストマネジメント」を実践する上でのヒントになれば幸いです。

野中がお話しする内容の元ネタ

- **単体テストでの欠陥見逃しの傾向をどのように示せばよいか**
 - － 下村・森(NEC)・野中(東洋大)「ソフトウェア単体テストの十分性評価基準に関する考察」『ウィンターワークショップ2013・イン・那須 講演論文集』情報処理学会(2013)
 - － 下村・森・佐藤(NEC)・野中(東洋大)「ソフトウェアメトリクスを用いた単体テストの品質リスク評価」『ソフトウェア品質シンポジウム2013(SQiP2013) 報文集』(2013)
- **第三者テストで検出される欠陥数に影響している要因は何か**
 - － 野中・小池・小室『データ指向のソフトウェア品質マネジメント』日科技連出版(2012) 4.2節

いずれも、品質管理や技術支援という立場からの分析事例です

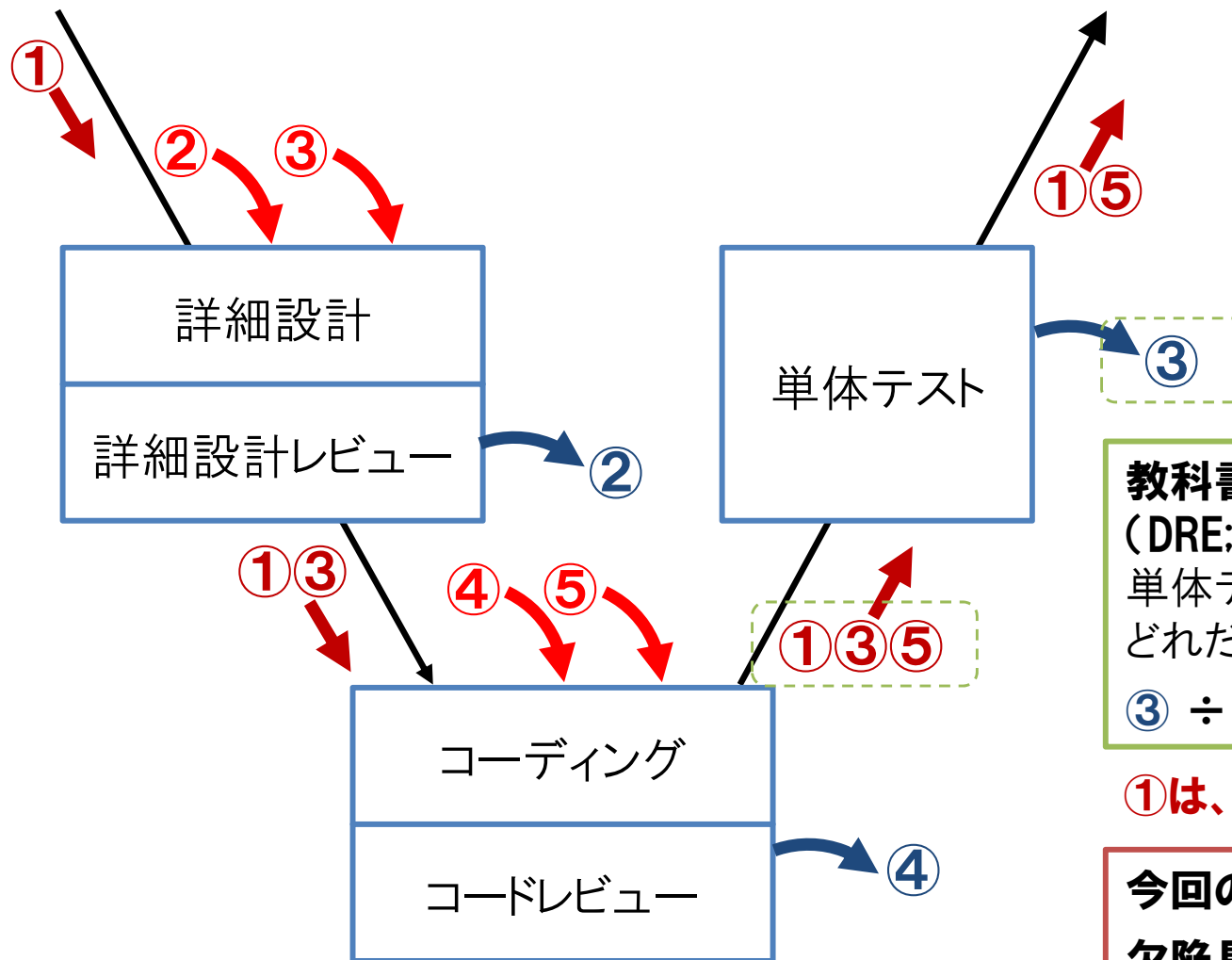
- ・プロジェクトに対してプロセス監査を行うときの評価基準を定める
- ・開発の現場で適用可能な評価基準を定める

- **単体テストでの欠陥見逃しの傾向をどのようにデータで示せばよいか**
- 第三者テストで検出される欠陥数に影響している要因は何か
- バグの収束状況をデータで判断するにはどのようにすればよいか
- 適正なテストリソース計画立案とコントロールのためにデータを如何に活用するか

なぜ単体テストでの欠陥見逃しに着目したのか

- 単体テストでの欠陥見逃しに関わる手戻り工数がそれなりに多いため、欠陥見逃しを減らしたい **《重点指向》**
- ツールで測定可能なデータが比較的容易に利用できる **《事実に基づく管理》**
 - ソースコード → 有効行数、重み付きサイクロマチック数、分岐条件数、最大ネスト深さを測定
 - カバレッジ → ステートメント(C0) / デシジョン(C1)を測定
 - テスト項目数 → テストコードのテストメソッド数を測定
 - 見逃し欠陥数 → BTSをもとに測定(混入工程も測定)
- 現場で使える単体テストの十分性評価基準を得たい **《標準化》**
 - usable かつ useful な基準としたい

単体テストでの欠陥見逃し状況をデータで把握するには



**教科書的な、単体テストの欠陥除去比率
(DRE: Defect Removal Efficiency)**

単体テスト開始時に存在した欠陥のうち
どれだけの欠陥を除去できたのかを測定

$$\textcircled{3} \div \textcircled{1}\textcircled{3}\textcircled{5} = 0.33$$

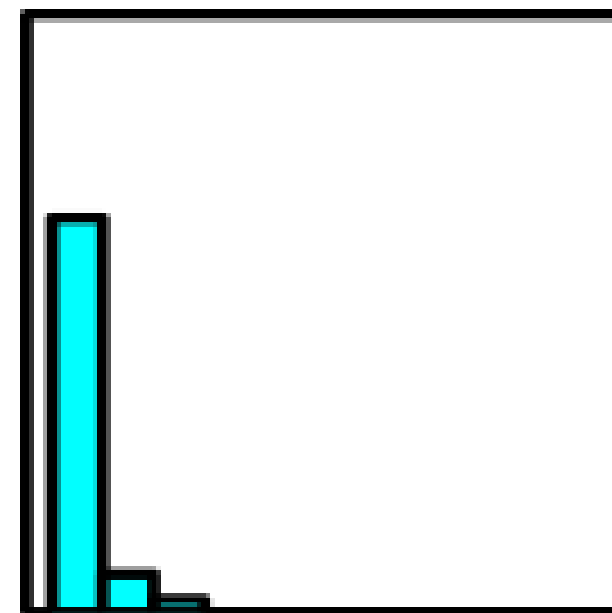
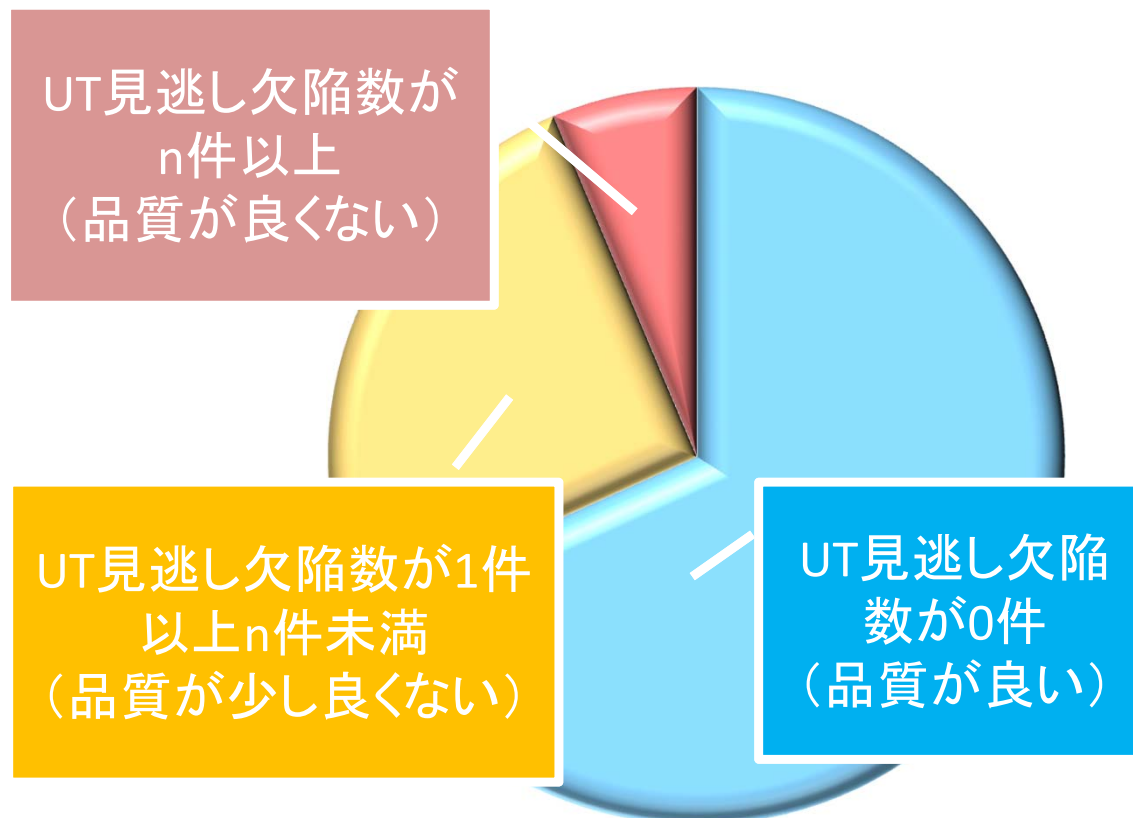
①は、単体テストで除去すべき欠陥か？

今回の分析対象とした見逃し欠陥: ⑤

欠陥見逃し率は $\textcircled{5} \div \textcircled{3}\textcircled{5} = 0.50$

**教科書的な定義に囚われることなく、目的に合った、運用可能な
測定方法を適用しよう**

単体テストの欠陥見逃し状況



下村・森(NEC)・野中(東洋大)「ソフトウェア単体テストの十分性評価基準に関する考察」『ウィンターワークショップ2013・イン・那須 講演論文集』情報処理学会(2013)

下村・森・佐藤(NEC)・野中(東洋大)「ソフトウェアメトリクスを用いた単体テストの品質リスク評価」『ソフトウェア品質シンポジウム2013(SQIP2013)報文集』(2013)

少数の品質不良の要因を統計解析手法で抽出するのは一般に難しい
でも、安易にあきらめずに、データから読み取れることを考えてみよう

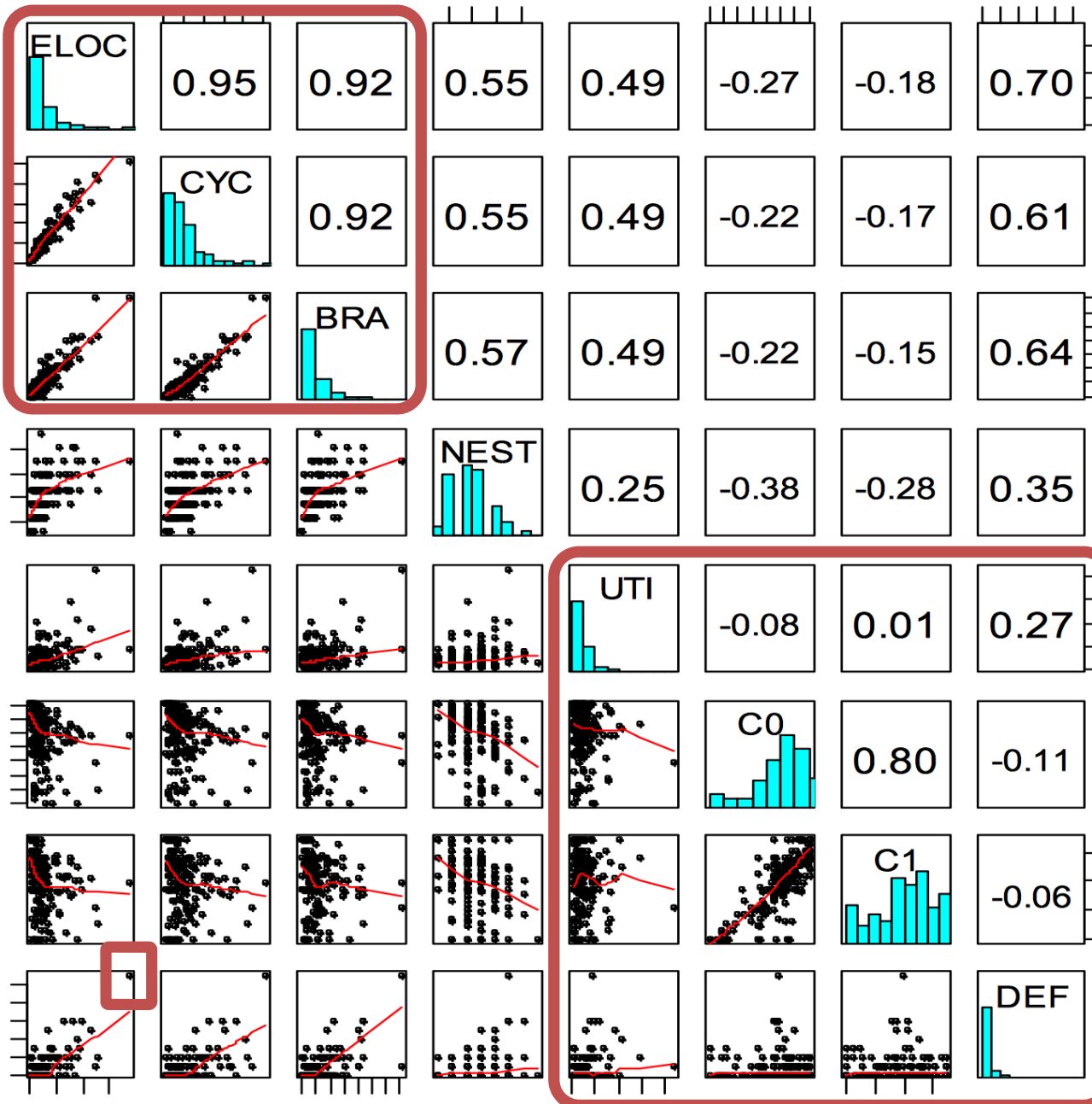
考えられる仮説

- 規模の大きいモジュールは、見逃し欠陥数が多くなる
- 複雑度の高いモジュールは、見逃し欠陥数が多くなる
- 分岐条件数の多いモジュールは、見逃し欠陥数が多くなる
- 最大ネスト深さ数の多いモジュールは、見逃し欠陥数が多くなる
- テスト項目数の少ないモジュールは、見逃し欠陥数が多くなる
- テスト項目密度の低いモジュールは、見逃し欠陥数が多くなる
- カバレッジの低いモジュールは、見逃し欠陥数が多くなる
- ...

「当たり前じゃない？」と思えることも含めて、まずは、広く浅く、仮説候補を挙げてみましょう

ただし、アクションにつながらない(つなげて意味のない)仮説は挙げないようにしましょう

散布図行列で全体を俯瞰する



見逃し欠陥数に外れ値候補あり
⇒ 除外して分析した方がよい
⇒ 規模大のモジュールは要注意

有効行数
重み付きサイクロマチック数
分岐条件数

- ・互いに強い正の相関
- ・見逃し欠陥数との関係が類似

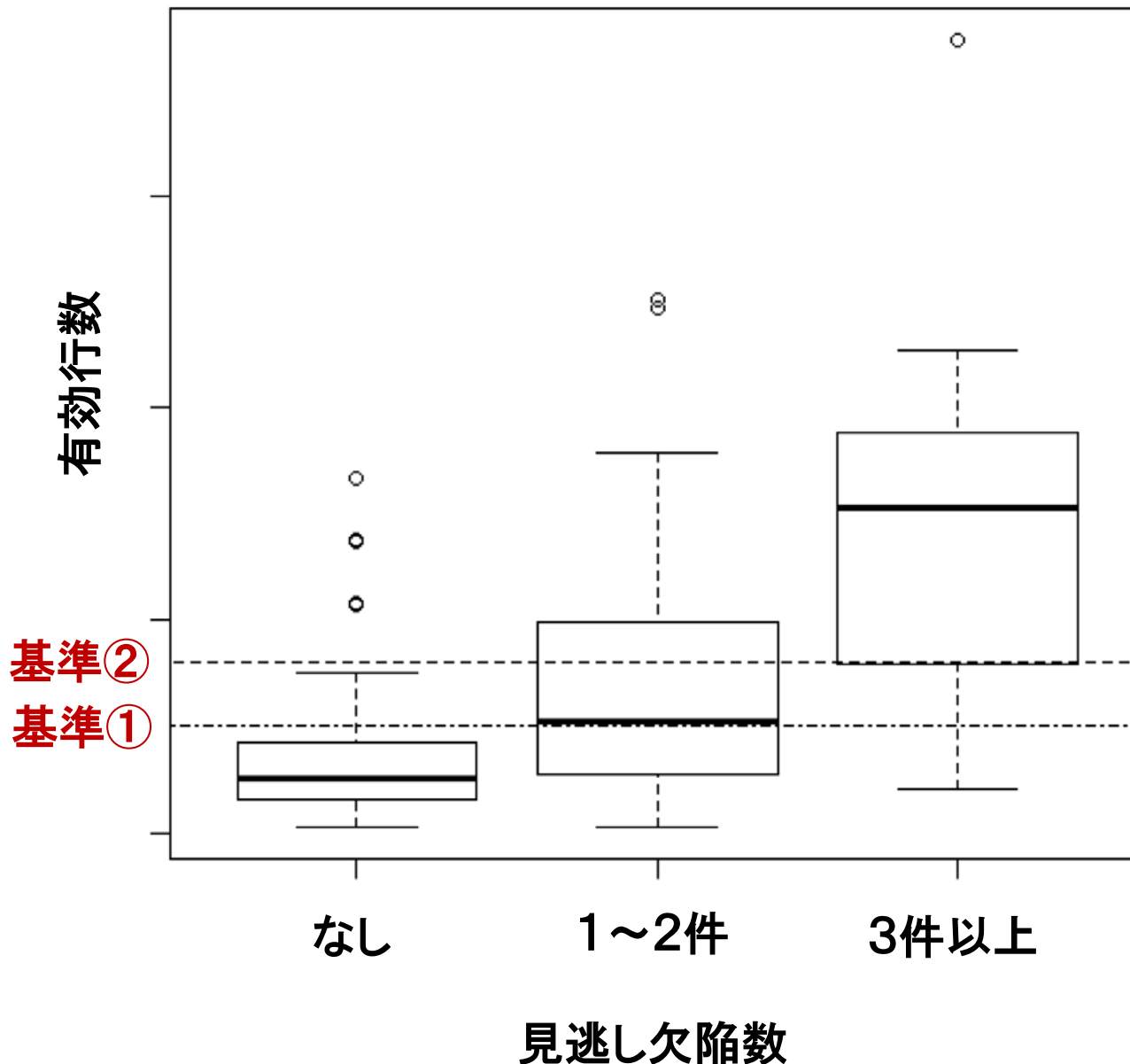
⇒ 有効行数だけ見ておけば十分

単体テスト項目数
ステートメントカバレッジ(C0)
デジジョンカバレッジ(C1)

- ・見逃し欠陥数との相関がない！
- ・単体テスト項目が少なくても、カバレッジが低くても、見逃し欠陥数が少なかったりする

⇒ 仮説と違うことはよくある

見逃し欠陥数で3水準に分けた、有効行数の箱ひげ図



基準②について

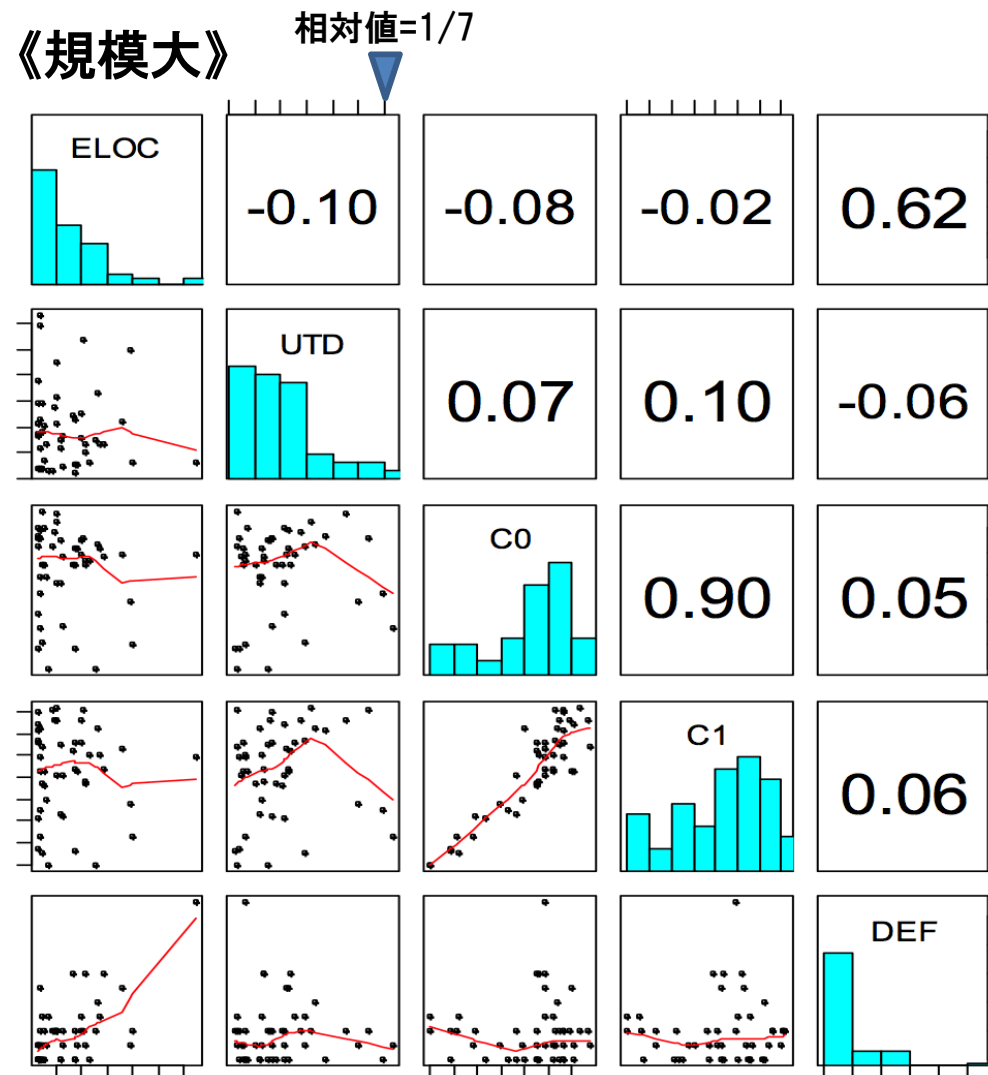
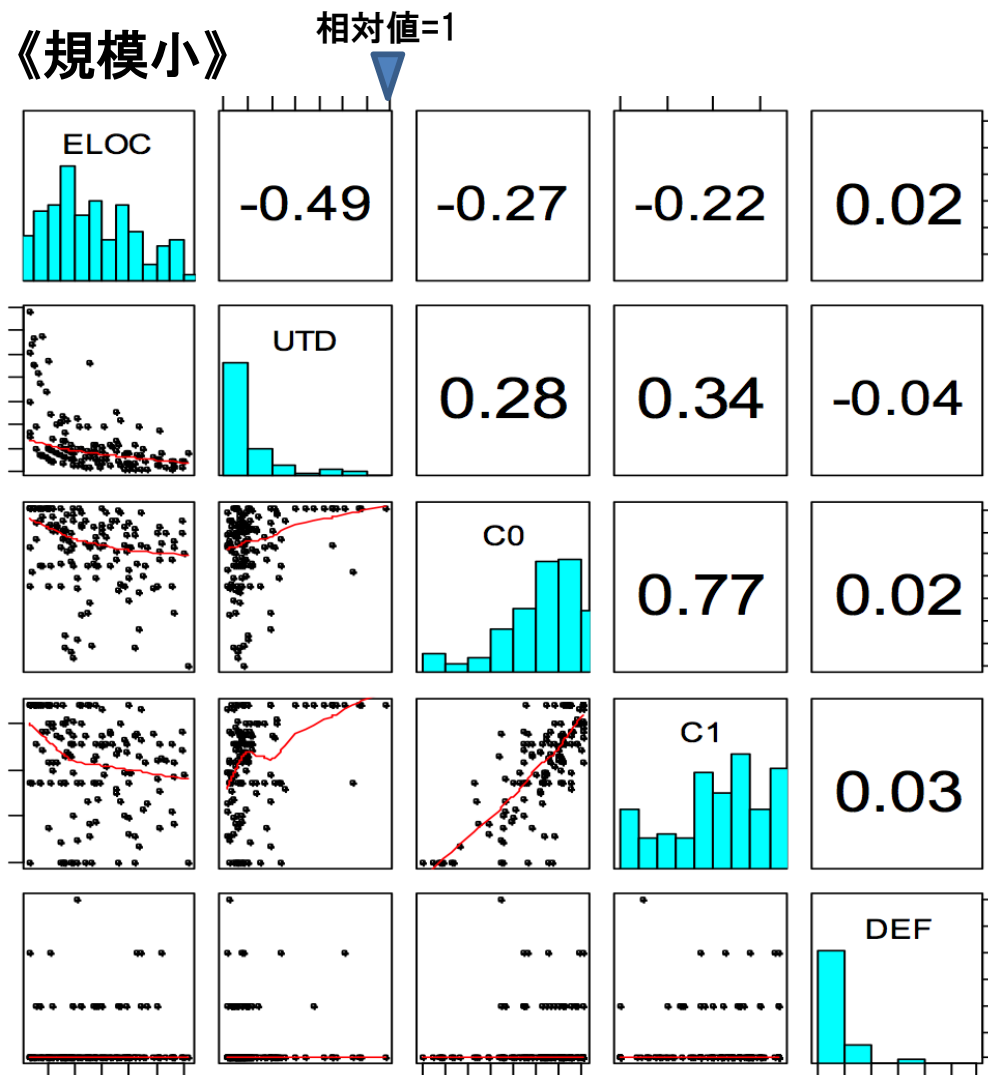
- ・見逃し3件以上のモジュールは、基準②を超えたものが75%
- ・見逃しなしのモジュールは、ほとんどが基準②を下回った

基準①について

- ・見逃し1~2件のモジュールは、基準①を超えたものが50%
- ・見逃しなしのモジュールは、基準①を下回ったものが75%

結論としては面白くないが(?),
有効行数は、欠陥見逃しの可能性を説明する有用な変数である

規模の大小で分けた、テスト項目密度とカバレッジの散布図行列



テスト項目密度が低く、カバレッジが低くても、コードの見通しが得られているのでテスト項目を積み上げておらず、見逃し欠陥数も少ない
⇒ モジュールのタイプ別に対応を考えたかどうか

規模小のデータに比べて、テスト項目密度が低いところに分布している
⇒ テスト項目密度の底上げが必要
⇒ 見逃し欠陥の分析に基づくテスト設計が必要

見逃し欠陥の修正内容に応じた対策

- ・規模大のファイルに対しては仕様ベースドテストの強化が有効
- ・規模小のファイルに対しては構造ベースドテスト(カバレッジ)の強化が有効

欠陥を摘出すべきテスト区分

修正ファイルの規模	仕様ベースドテストの割合	構造ベースドテストの割合
修正ファイルが全て規模「大」の欠陥	78%	22%
修正ファイルが全て規模「小」の欠陥	0%	100%

テスト区分分類事例

テスト区分	欠陥の原因	実施すべきテスト技法例
構造ベースドテスト	クラス名称誤り、設定値誤り	ステートメントカバレッジ
	判定ロジック評価漏れ	デシジョンカバレッジ
仕様ベースドテスト	状態遷移評価漏れ	状態遷移テスト
	例外処理評価漏れ	ユースケーステスト
	コマンドオプション評価漏れ	デシジョンテーブル
	入力バリエーション評価漏れ	同値分割／境界値テスト

品質管理の基本を適用しよう

- プロセスと結果を監視し、プロセスに対して処置する

- 結果は好ましい状態にあるか？
⇒ **結果**をデータで把握することから始めよう
- プロセスは維持された状態にあるか？
⇒ 結果に影響する要因を考えてみよう



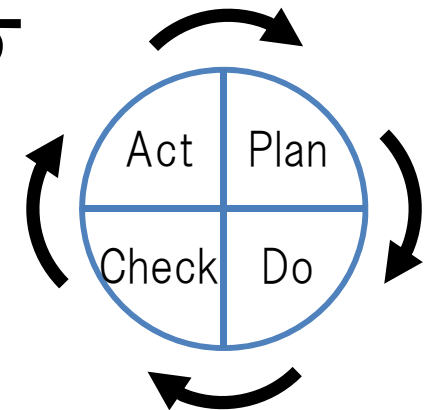
- 重点指向で、必要に応じてプロセスの粒度を細かく管理する

- 単体テストに着目



- それぞれのプロセスについて、PDCAサイクルを回す

- Planには2つの意味がある
 - (1) 目的・目標を決める
 - (2) 目的・目標を達成する方法を決める(**標準**など)
- 標準の正しい意味を知り、正しい標準を定めよう
経験の再利用, 技術・改善・独創性の基盤



- 単体テストでの欠陥見逃しの傾向をどのようにデータで示せばよいか
- **第三者テストで検出される欠陥数に影響している要因は何か**
- バグの収束状況をデータで判断するにはどのようにすればよいか
- 適正なテストリソース計画立案とコントロールのためにデータを如何に活用するか

分析対象のデータ

• 定量データ

- ソフトウェア規模 (KLOC)
- 工数 (上流工程～結合テスト)
- 欠陥数 (開発終了後の独立テストで発見された機能欠陥)

• 定性データ ～ それぞれの要因についての5段階評価 (VL ～ VH)

S: 要求定義プロセス

- S1: 当該工程に関するメンバーの経験
- S2: ユーザ要求文書の品質
- S3: 要求・仕様・テスト仕様レビューの充足度
- S4: 要求レビュープロセスの効率性
- S5: 要求レビューの有効性
- S6: 仕様レビュー指摘欠陥の多さ
- S7: 要求の安定性

F: 新規開発機能

- F1: 新規開発機能の複雑度
- F2: 新規開発機能の規模
- F3: ベース規模に対する新規規模の大きさ

D: 開発プロセス

- D1: 当該工程に関するメンバーの経験
- D2: プログラマの能力
- D3: 設計レビュープロセスの効率性
- D4: 開発メンバーのモチベーション

T: テスト、手戻り

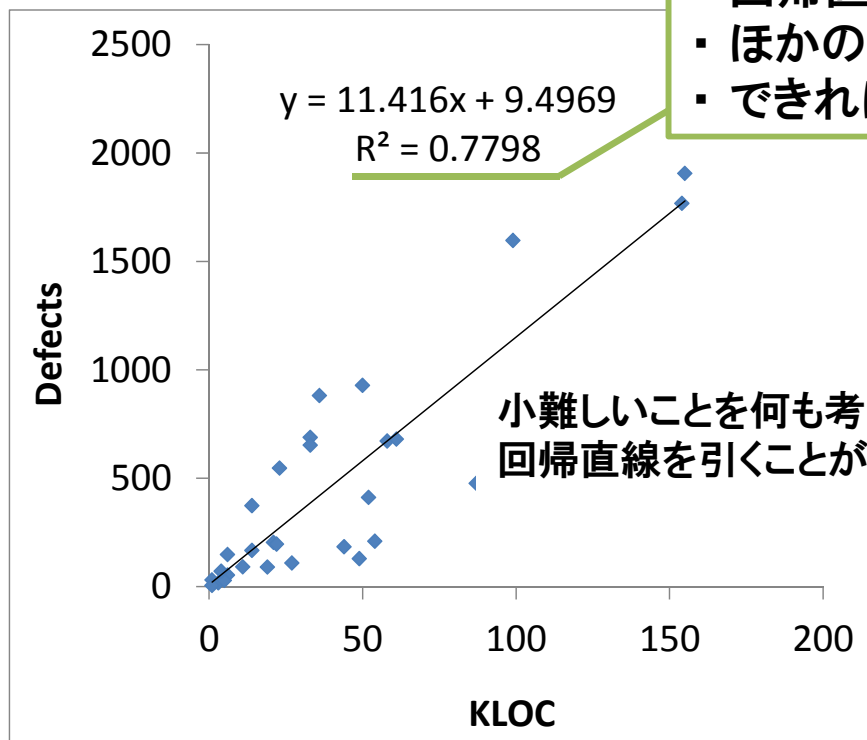
- T1: テストプロセスの効率性
- T2: 結合テストに関するメンバーの経験
- T3: 統合テストに関するメンバーの経験
- T4: テストケースの質

P: プロジェクト管理

- P1: 開発メンバーのトレーニングの質
- P2: 構成管理プロセスの効率性
- P3: プロジェクト計画の適切さ
- P4: 開発拠点/グループの多さ
- P5: 主要なステークホルダの関与度
- P6: 顧客の関与度
- P7: 協力会社マネジメントの有効性
- P8: チーム内コミュニケーション
- P9: プロセス成熟度

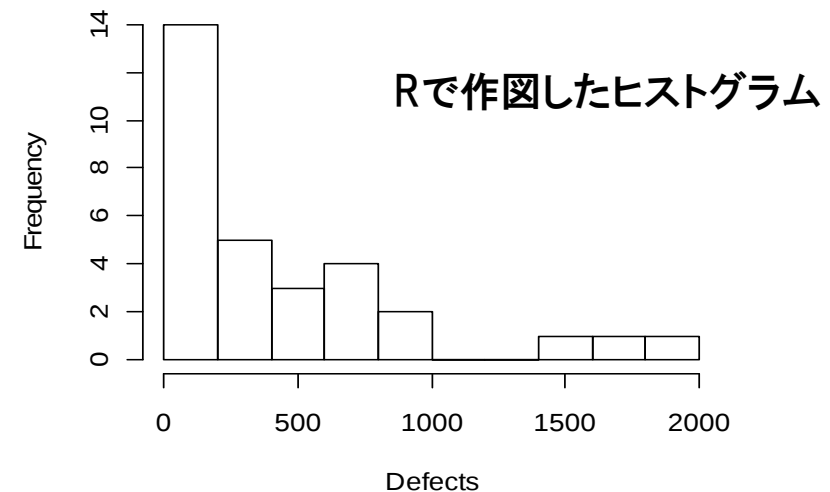
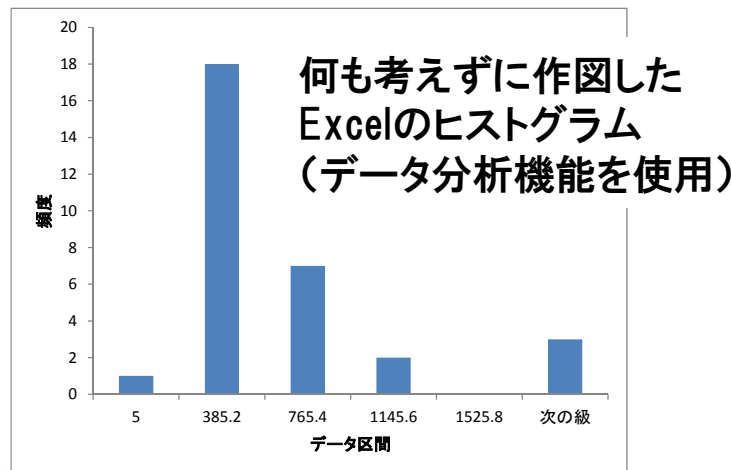
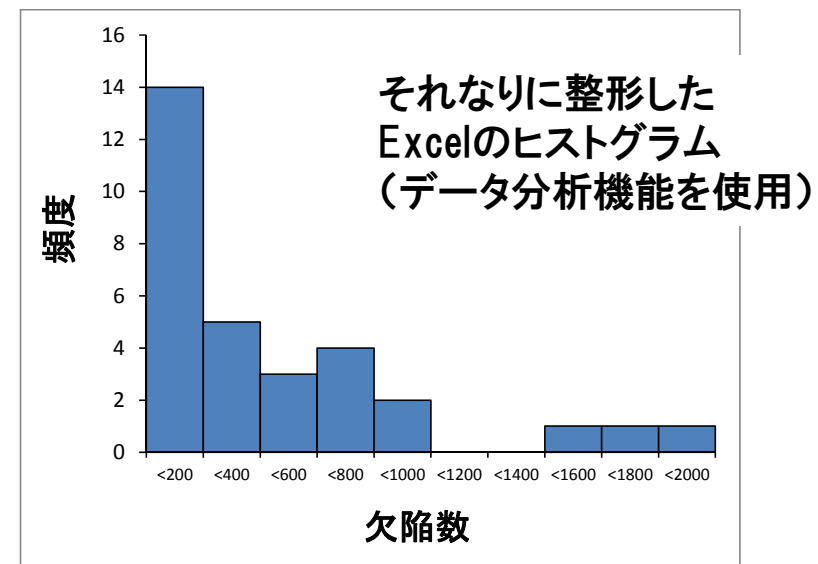
Fenton, N., et al. *Project Data Incorporating Qualitative Facts for Improved Software Defect Prediction*. in *Predictor Models in Software Engineering*, 2007.
PROMISE'07: ICSE Workshops 2007. *International Workshop on*. 2007.

データの傾向を把握する: まずはヒストグラムと散布図



- ・ 回帰直線によって、欠陥数のばらつきの78%を説明できている？
- ・ ほかの定性的なデータも使って、予測精度を高めたい
- ・ できれば、少ない変数で欠陥数を予測したい

小難しいことを何も考えなくても
回帰直線を引くことができる



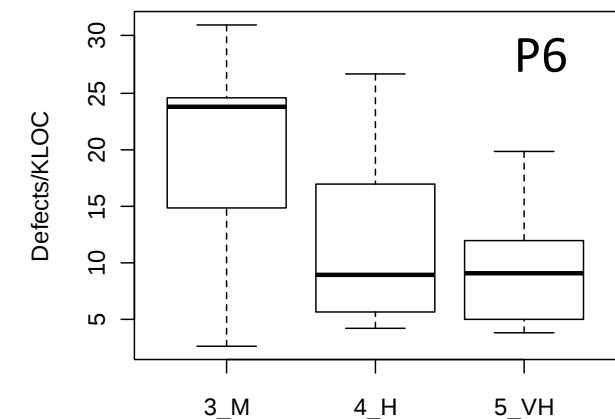
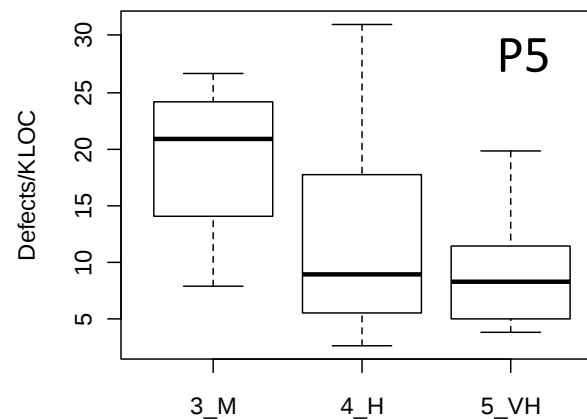
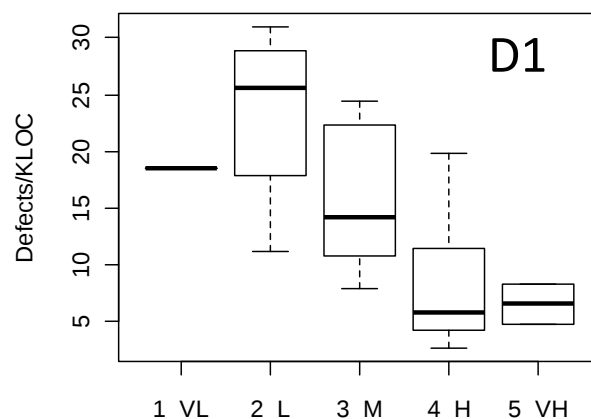
そのほかの定性的なデータも使って予測精度を高めたい

相関係数を使って、有用そうな変数にあたりをつける

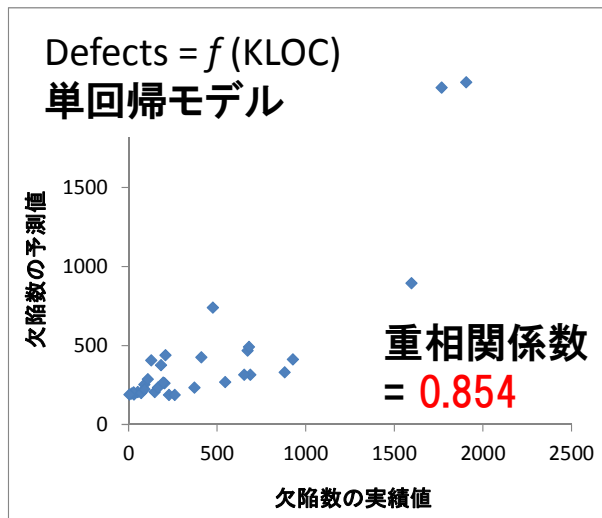
データ項目	欠陥密度	不具合数
D1:開発スタッフ経験	-0.693	-0.408
P5:ステークホルダ関与度	-0.435	-0.254
P6:顧客関与度	-0.374	-0.137

※ スピアマンの順位相関係数を使っています

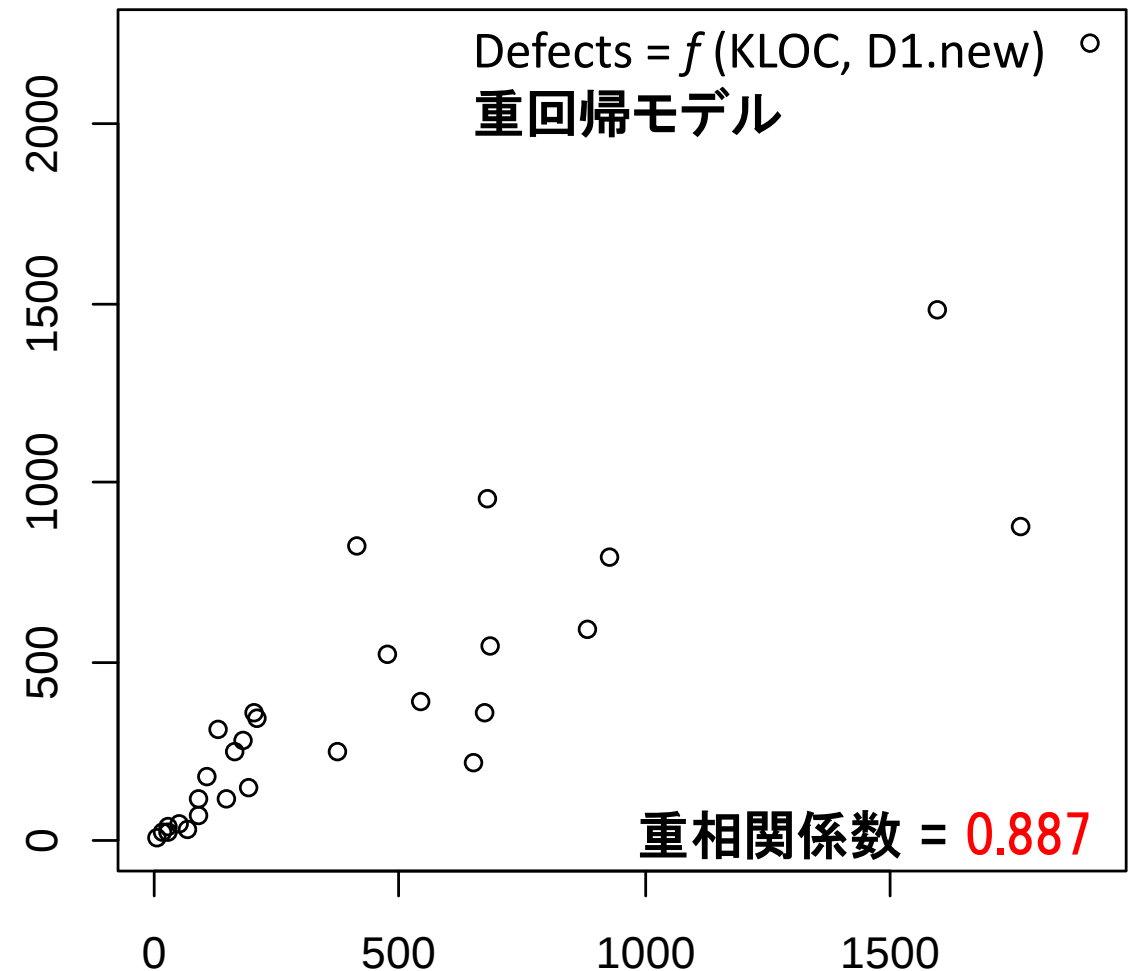
箱ひげ図を描いて、欠陥数(欠陥密度)に影響する要因を探る
もちろん、要因と効果の因果関係も検討しなければならない



実績値と予測値の散布図



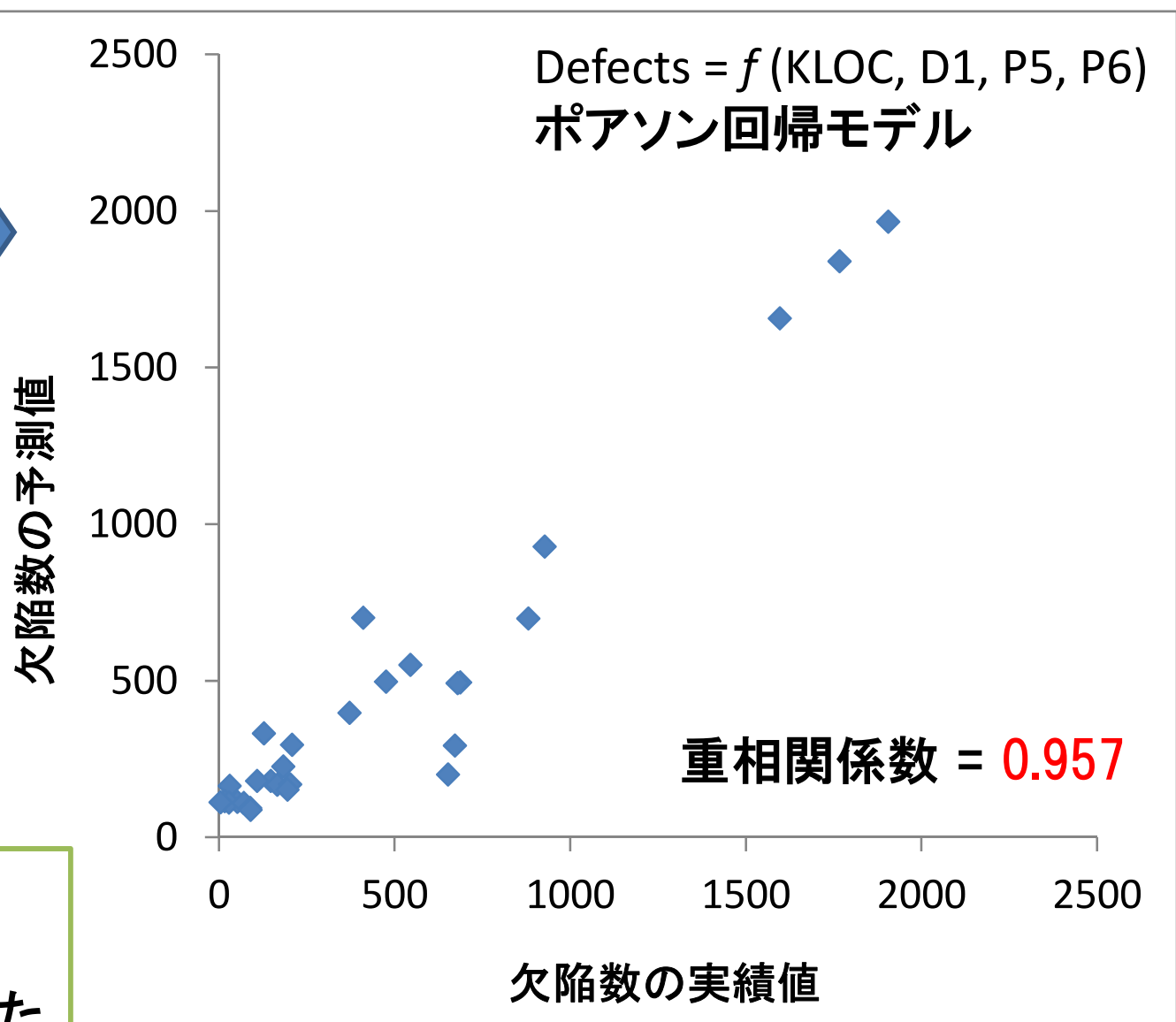
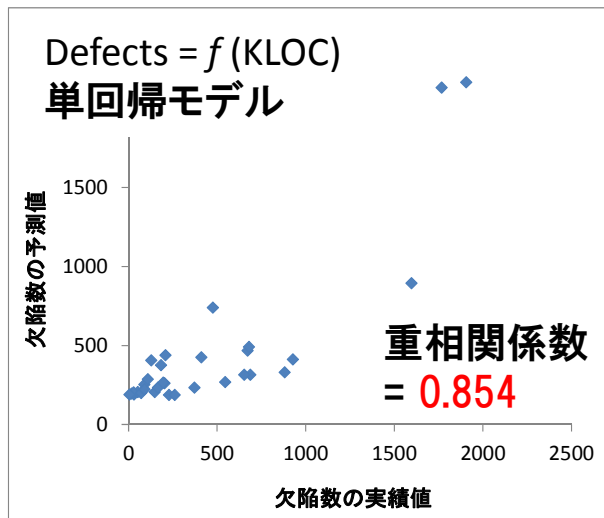
不具合検出数の実績値(件)



不具合検出数の予測値(件)

変数を少し増やしたことにより、
僅かだが予測精度の高いモデルが得られた

実績値と予測値の散布図



ポアソン回帰モデルという
手法を用いたことで、
予測精度を高めることができた

品質データ分析は改善の推進力

- 「事実に基づく管理」は品質管理の基本原則である
- ソフトウェア開発における「事実にもとづく管理」は容易でない
 - データの測定・収集プロセスが、**人に大きく依存**している
 - 目的と整合性のあるメトリクス分析をし続けることが難しい
- ソフトウェア開発において「事実に基づく管理」を進めることが、中・長期的には品質向上の有効な施策である
 - **品質向上(または悪化)のメカニズム**を解明する
 - データが示す**客観的事実の力**を活用し、改善の推進力とする
 - データを手がかりに原因を特定し、アクションをとり、効果を確認する
そして、
 - 再発防止のためにプロセスを改善し、失敗プロジェクトを撲滅する
 - 顧客価値の向上につながる活動に注力し、顧客の満足度を高める

*“I’d rather be **vaguely right** than **precisely wrong**.”*

精密に誤るよりも、漠然と正しくありたい

– John Maynard Keynes