

障害修正工数に関する要因分析事例

奈良先端科学技術大学院大学

産学官連携研究員

松村知子

2006年3月16日 エンピリカルソフトウェア工学研究会

キャンパスイノベーションセンター・国際会議室(田町)

背景

- 障害の発生要因の分析は、下流工程における障害発生を防止するプロセス改善などの対策に必要不可欠である
- 従来の障害の原因－結果分析 (Cause - Effect Analysis) は、障害発生数に基いており、プロセス改善による投資効果 (Return on Investment) が明確ではない
 - 仕様変更要求が頻発すると予測される画面デザインなどの場合、変更容易な設計でスケジュール・コストへの影響を小さくしている
- 障害タイプや障害発生工程などの粗い粒度で分析しても、直接的にプロセス改善の有用な情報にはならない
 - 根本原因 (Root Cause) の追及には経験者や専門家による詳細分析が必要で、コストがかかる

本研究の目的

- 修正工数の大小に影響する要因を分析する
- 原因の特定を容易にするため、粒度の細かい障害データを収集し、統計的手法を用いて分析する
- 期待される効果
 - プロセス改善などの方策による投資効果を示すことが可能になり、組織的導入が容易になる
 - プロジェクトに関する知識の少ない作業者でも、容易に要因を分析でき、早期に問題を指摘できる。プロジェクト進行中にリアルタイムフィードバックすることが可能になる

アプローチ

- 修正工数の大小に影響すると思われる障害データ項目と修正工数(人時)を収集する
 - 企業で用いられている障害票やIEEE Standard.1044 Classification for Software Anomaliesなどをベースに, 収集項目を決定
- 統計的手法(箱ひげ図やt検定)を用いて, 修正工数に影響する要因を抽出する
- 要因間の相関を平行座標プロットを用いて, 調査する
- 障害の詳細情報(問題内容, 修正方法など)を精査して, 根本原因を追究する

対象プロジェクト・収集データ項目

- 対象プロジェクト: プローブ情報システム開発
 - ベンダ5社7拠点による分散開発
 - 2005年4月～2005年10月第1フェーズ開発のデータが対象
- 目的変数: 修正工数(人時)
 - 正規分布でないため, 対数変換値を用いる
- 説明変数: (拠点名), 起票分類, 発見箇所(新規, 改造, 再利用), 発見工程, 発見作業, 問題処理機能, 再現度, 重要度, 優先度, 混入工程, 問題原因, 問題を本来発見すべき工程, 問題を発見できなかった要因, 修正工程, 障害存続時間*
- 各説明変数のとりうる値を「カテゴリ」と呼ぶ
 - 重要度のカテゴリは, 「重大」「中度」「軽微」

収集データ概要

- 障害データ数 : 1182件
- 修正工数統計値
 - 修正工数入力データ 1063件
 - 修正工数欠損データ 119件
 - 最小値=0が1件 欠損値扱い
 - 0を除くと, 0.08人時が最小値
 - 欠損値は, 以降分析対象から除く

分析データ説明

- 工程分類

- 基本設計, 詳細設計, コーディング/単体試験, (拠点内)結合試験, 統合環境結合試験, 総合試験

- データ前処理

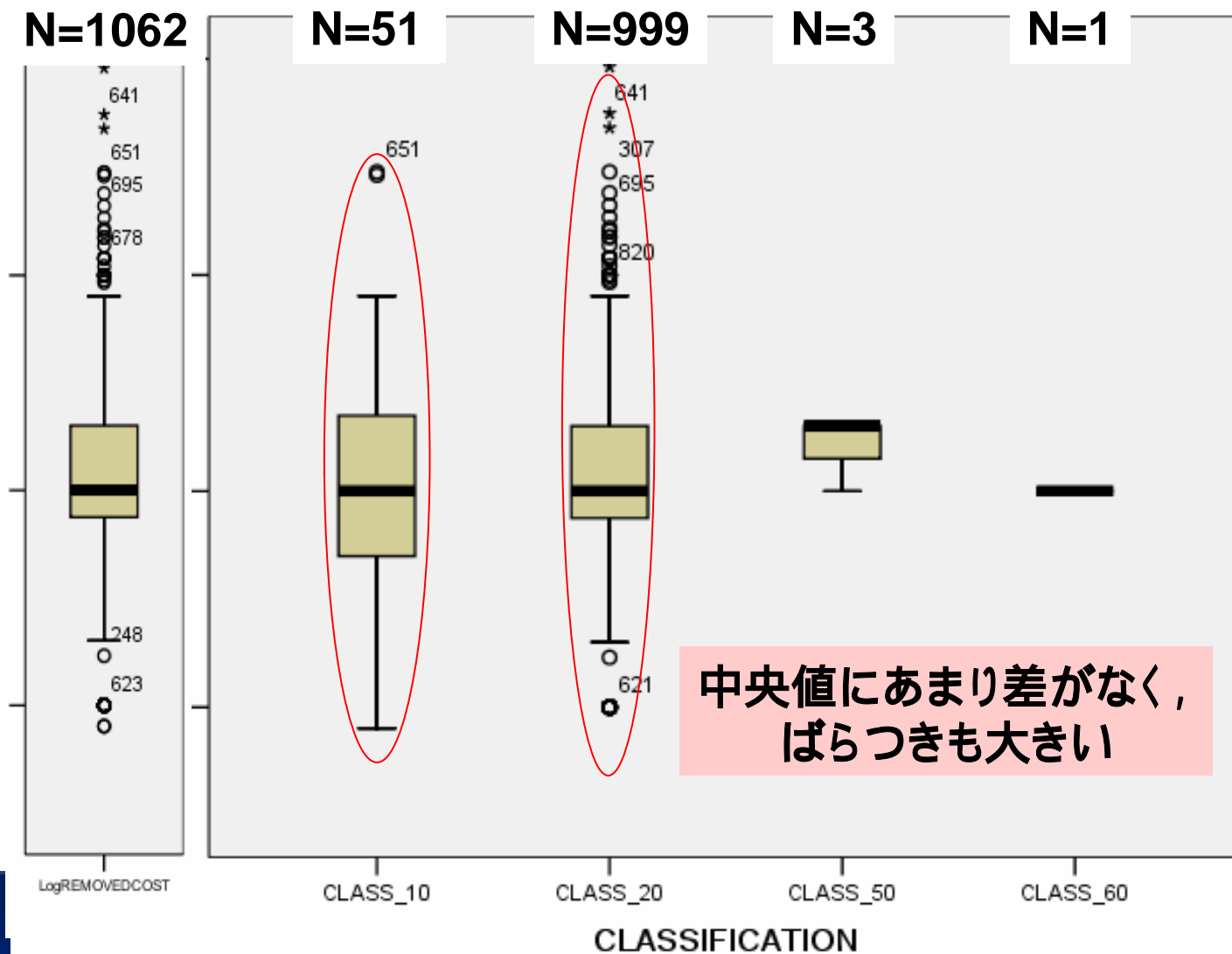
- 統合環境結合試験, 総合試験はデータ数が少ないので, 「総合試験」に統合
- 「問題を本来発見すべき工程」が未入力の場合, 「混入工程」を代用する. 混入工程も未入力の場合, 欠損値とする.
- 「障害存続時間」は, 「発見工程」と「問題を本来発見すべき工程」が同じ場合0, その他は工程差ごとに ± 1 する

統計的分析手順

- 各説明変数のカテゴリ毎の箱ひげ図 (BoxPlot) で、修正工数差のある変数を視覚的に確認する
 - サンプル数10以上で、中央値に差がある、もしくはばらつきの小さいカテゴリが有意差 (意味のある差) が存在すると考えられる
- 各説明変数のカテゴリ毎のt検定で、平均値の差を統計的に確認する
 - 対象カテゴリとそれ以外のデータで、有意水準0.01で検定する
- 平行座標プロット (PCP) で変数間の相関関係をチェックし、本当に修正工数に影響している変数に絞り込む

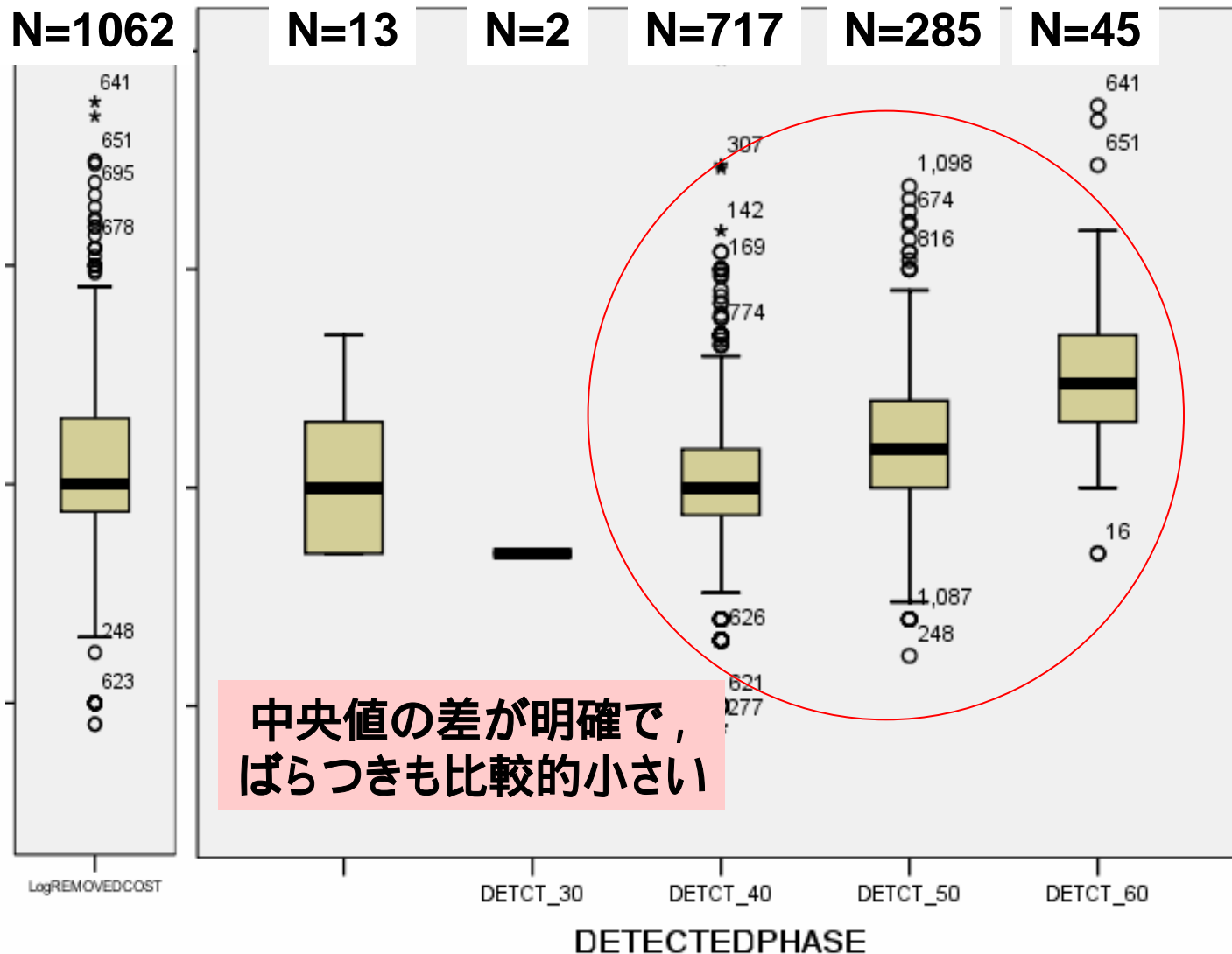
結果例 ~ BoxPlot (起票分類)

サンプル
数(N)



結果例 ~ BoxPlot (発見工程)

サンプル
数(N)



結果例 ~ t検定

- CLASS: 起票分類
 - 10: 仕様不具合 / 20: コード不具合 / 50: ユーザからの仕様変更 / 60: 不具合からの仕様変更
- DETECTED: 発見工程
 - 30: 詳細設計 / 40: コーディング (製造) / 単体試験 / 50: 結合試験 / 60: 総合試験

	N		等分散仮定		t検定結果		
	1	0	P値		P値	P値<0.01	P値<0.05
CLASS_10	59	1,003	0.10497	等分散	0.808151	×	×
CLASS_20	999	63	0.17950	等分散	0.902554	×	×
CLASS_50	3	1,059	0.27400	等分散	0.56017	×	×
CLASS_60	1	1,061		等分散	0.890504	×	×
DETECTED_30	2	1,060	0.11976	等分散	0.229949	×	×
DETECTED_40	717	345	0.23199	等分散	1.7E-24	○	○
DETECTED_50	285	777	0.22939	等分散	3.99E-12	○	○
DETECTED_60	45	1,017	0.70367	等分散	1.89E-17	○	○

結果 ~ 有意差のある変数・カテゴリ

- 有意差のあるカテゴリで，平均修正工数差が1.0人時以上
 - 問題原因: 指摘ミス(仕様どおり)，その他
 - 発見工程: コーディング/単体試験，総合試験
 - 障害存続時間: 0，2，3
 - 混入工程: 詳細設計
 - 発見すべき工程: 詳細設計
 - 修正工程: 総合試験
 - 再現度: 再現頻度少，再現なし
 - 重要度: 軽微
 - 発見できなかった要因: 試験項目抽出漏れ，環境上の問題で後工程に持っていった，結果確認ミス，その他
 - 発見作業: レビュー

グレーは分析対象外
青字は修正工数が小さいもの

変数間相関の仮説

- 仮説1: 再現度 & 発見作業
 - 発見作業が「レビュー」の場合, システムの動作検証なしで修正するため「再現なし」ではないか?
- 仮説2: 問題を発見できなかった要因 & 発見工程
 - 問題を発見できなかった要因はテスト工程の問題で後工程にずれ込んだものが多い. 発見工程が遅いことが修正工数の大きい原因ではないか?
- 仮説3: 問題を本来発見すべき工程 & 発見工程 & 障害存続時間
 - 発見工程が遅いほど修正工数が高い, 設計工程が原因の障害は修正工数が高い, など一般に言われているが, 最も影響するのは障害存続時間ではないか?

変数間相関調査結果(1)

- PCPデモ : GNATS_hipo1_PCP.csv
 - 修正工数(対数) - 再現度-発見作業
- 仮説1
 - 再現なしはレビュー発見障害が混入したため、確認テストなしに目視確認で対応完了している。
 - レビュー以外の再現なし障害3件は、確認結果、誤入力と判定
- 仮説2
 - 試験項目漏れはほとんど結合試験発見の障害だが、修正工数はばらついている
 - 結果確認ミス・環境上の問題で後工程に持っていったテストの障害は、結合・総合試験発見に分散しているが、修正工数は大きい方に偏っている

変数間相関調査結果(2)

● 仮説3

- 総合試験で発見された障害は, 発見すべき工程・障害存続時間によって修正工数に大きな偏りはない
- 結合試験で発見された障害は, 発見すべき工程が詳細設計の場合修正工数が大きい方に偏る
- t検定による確認

発見工程 = 結合試験			
説明変数	カテゴリ	サンプル数	P値
発見すべき工程	詳細設計	33	5.16E-05
	コーディング/単体テスト工程	171	0.016676
	結合テスト工程	50	0.160007
発見工程 = 総合試験			
説明変数	カテゴリ	サンプル数	P値
発見すべき工程	詳細設計	5	0.7974
	コーディング/単体テスト工程	21	0.143952
	結合テスト工程	6	0.388991

有意差あり

考察

- 結果確認ミスや環境上の問題で後工程に持って行ったテストの障害は、発見工程に関わらず、修正工数が多い
 - 障害原因が複雑で、発見や確認が困難(環境設定やI/Fの確認を含む)
- 発見工程が総合試験では、発見すべき工程(混入工程)に関係なく修正工数が高い。発見工程が結合試験では、詳細設計工程で混入した障害が修正工数が高い。
 - 総合試験から統合環境であるためと思われる。

今後の課題

- 修正工数の細分化
 - 修正工数を「分析・解析工数」「修正工数」「確認工数」に分類して、工数に関する原因－結果関係をより明確にする
- 修正工数の大きい障害の予測
 - 修正工数の大きい障害を予測することによって、テスト工程中のコスト超過・スケジュール遅延リスクを検出する
 - 予測手法：判別分析，アソシエーション分析，マハラノビスタグチ法など

謝辞

- データ収集・分析, 及びインタビューに多大なご支援・ご協力いただいた関係者の皆様に心から感謝いたします
 - ソフトウェアエンジニアリング研究組合 (COSE) 参加企業の皆様
 - プローブ情報システム開発関係者の皆様
 - ソフトウェアエンジニアリングセンター (SEC) の皆様