

ソフトウェアテスト技術の最新動向

JaSST'05 in Osaka

2005/07/15

香川大学工学部

信頼性情報システム工学科

古川善吾

zengo@eng.kagawa-u.ac.jp

目次

- 1. はじめに
- 2. テスト手順
- 3. テスト技術
- 4. テストの種類
- 5. 最近の話題
- 6. おわりに

1. はじめに

- **ソフトウェアテストの位置づけ**
 - Dijkstra: テストによってプログラムの正しさを示すことができない
 - Myers: テストは、誤りを発見する作業である
 - 玉井哲雄: 誤りを発見すると同時に「確からしさを確信する」ための作業
 - 統計的テスト法: ソフトウェア品質を求める
 - テスト駆動開発 – TDD (Test Driven Development)
 - アジャイル開発
- **ソフトウェアテストの効率化、高品質化**
- **→最近のソフトウェアテスト技術の話題**

2. テスト手順

- **テスト設計**
 - テストケースの作成
 - テストデータの作成
- **テスト実施**
 - プログラムの実行
 - テスト結果の判定
- **テスト評価**
 - テストの充分性の評価
 - ソフトウェアの信頼性評価

3. テスト技術

- 静的手法
 - プログラム解析
 - 正当性検証
- 半動的手法
 - 記号実行
 - モデル検査(?)
- 動的手法
 - テスト

4. テストの種類

- 4.1 テストフェーズあるいはテスト目的による分類
- 単体テスト: 処理の流れ
- 統合テスト: モジュール間インターフェイス
- システムテスト: システム全体としての動作
- 受入テスト: システムの機能
- 4.2 プログラムの利用方法
- ホワイトボックステスト
- グレイボックステスト
- ブラックボックステスト

5. 最近の話題

- **統計的テスト**
 - :最近進めている研究。利用モデルの作成、支援ツール
→Static-test
- **TDD (Test Driven Development)**
- **自動テスト:JUnit, CPPUnit**
 - :eXtreme Programming, Unix→XP-Junit
- **ネットワーク対応のテスト:パフォーマンステスト、セキュリティ対策、分散処理のテスト**
- **組込システムのテスト:並行処理**
 - :並行動作の統計的テスト→Concurrent-Statistic
- **国際学会での話題:ISSTA2004, ISSTA2002, ISSTA2000, ICSE2005, ICSE2005, ICSE2003**

ISSTA2004

- 10セッション
- 1. 2ワークショップ：
 - Webサービスのテストと解析、検証
 - ソフトウェアテストの経験的研究
- 2. プログラム解析1：
 - Javaの呼び出し連鎖の静的・動的解析
 - 拡張可能な路依存の値フロー解析を用いたソフトウェア確認
 - JavaのWebサービスに関する頑強性のテスト
- 3. テスト1：
 - malware(virus, torojan horse...)発見器のテスト
 - 複雑な構成空間における効率的な欠陥特性のための被服配列
 - 非決定システムのテストのための最適戦略

ISSTA2004(つづき)

- 4. 経験研究：
 - 開発中の継続テストの経験的評価
 - バグはどこに
- 5. テスト生成：
 - Java PathFinder を用いたテスト入力生成
 - ループ代入フラグの存在における発展的テスト
 - クラスの発展的テスト
 - ツール：モデルに基づくテストのためのAGEDISツール
- 6. テスト2：
 - 限定された徹底的テストによるソフトウェアの正確さ
 - 関数推定を用いた関数正当性の自動化されたブラックボックステスト
 - テスト技法の解析的比較

ISSTA2004(つづき)

- 7. モデルチェック1:
 - 設計レベルでの自動化された交換可能性解析
 - 演算の航空機交通量のモデルと検証
 - 時相論理式の一括最適化コンパイラ
- 8. プログラム解析2:
 - ソフトウェア動作の自動分類のための能動的学習
 - プログラム舵取り(?) によるマルチモードシステムの適用可能性改善
 - ツール:同期反応型プログラムのスライシングツール
- 9. プログラム解析3:
 - 原子性に対する純粹性拡大
 - 部分型を用いた高速制約解決
 - SABER: エラー減衰に基づくスマート解析

ISSTA2004(つづき)

- 10. モデルチェック2：
 - XML操作ソフトウェアのモデルチェック
 - RTCORBAに基づく応用プログラムのモデル化および検証に対する形式的手段
 - パラメタ化された状態機械を用いた検証過程モデルの構築

ISSTA2002

- **Javaプログラムの静的解析：3件**
- **安全性の確認：2件**
- **故障と障害の解析：4件**
- **テスト効率の改善：3件**
- **仕様に基づくテスト：3件**
- **確認 (conformance)と相互操作性テスト：3件**
- **並行プログラムの解析：2件**
- **テストと信頼性の理論：3件**
- **パネル：ISSTA の研究は、産業界に役立っているか？**
- **動的解析：3件**

ISSTA2000

- 静的解析：3件
- オブジェクト指向ソフトウェアとコンポーネントのテスト：4件
- 実時間とプロセス：3件
- 事例報告：4件
 - 回帰テストのためのテストケースの優先付け
 - どのポインター解析法を使用すべきか？
 - 分岐、データフロー、操作テストの導出信頼性比較：ケーススタディ
 - 誤り帰納入力(?)の簡単化

ISSTA2000(つづき)

- **テスト4件**
 - データベース応用ソフトウェアのテストのための枠組み
 - jRapture: 観察ベースのテスト法のためのキャプチャ／リプレイツール
 - テスト可能性、誤りサイズ、定義域一値域比率：永遠の三角形
 - 入出力解析によるブラックボックステストの削減
- **並行性解析：3件**

ICSE2005

- Research Papers: 15セッション
- Experience Reports: 5セッション
- 研究 3/15
 - テストと解析: 3件
 - テストの経験的評価: 3件
 - 静的・動的解析: *
- 経験 2/5
 - 要求とテスト: 2件
 - 予測と検証: 3件

ICSE2004

- 20セッション 5/20
- テスト1: 3件
- テスト2: 3件
- 検証: 3件
- 静的解析: 3件
- 動的解析: 3件

ICSE2003

- **テスト1：3件**
 - 相互性テストのテストスイートの構成
 - 利用者のセッションデータを用いたWeb アプリケーションテストの改善
 - 操作的抽象化によるテストスイートの改善
- **テスト2：3件**
 - Javaプログラムにおける多様性のテストのためのクラス断片解析
 - コンポーネント配置テストの枠組み
 - モデル検査としてのデータフローテスト
- **型式手法：2セッション、6件**

ICSE2003(つづき)

- **テストと故障修正：3件**
 - IBMにおけるTDDの評価
 - 大規模SDL開発モデルにおける故障修正プロセスの解析
 - 安全性に重要なソフトウェアのテストにおける要求の発見

6. おわりに

- テスト: Webシステムのテスト、仕様に基づくテスト、誤り解析、...
- 形式的手法: 仕様化、形式化→モデルチェック
- テスト (／デバッグ) の経験報告
- 実践的評価



- 産業界における経験データの公開を！！

バグの統計の出典

ソフトウェアテスト技法

1994年2月25日 1版1刷

著者 ボーリス・バイザー
 おのま あきら やまうら つねお
訳者 小野間 彰, 山浦 恒央
発行人 今関 宗夫
発行所 日経BP出版センター

Software Testing Techniques (Second Edition)

by Boris Beizer

Copyright © 1990 by Van Nostrand Reinhold. All Rights Reserved.

Japanese translation rights arranged with Van Nostrand Reinhold through
The English Agency (Japan) Ltd.

Japanese translation copyright © 1994 by Nikkei BP Publishing Center, Inc.

統計の解説

多数の異なる文献からバグの発生頻度を調査した結果、ソースプログラム1000行当たり約2.4件バグ*があった。しかし、サンプルの中に、非実行文を対象外としているので、現実の値はそれより低くなる。このデータの大部分は、設計とは独立したテスト、統合テスト、システムテストで検出したバグである。プログラマがコンポーネントのレベルで検出したバグの件数は明らかではない。表2.1は、バグの絶対的な頻度（つまり、1 k ステップ当たりのバグの件数）ではなく、バグの種類別の相対的な発生頻度を示していることに注意すること。この表のもとになったそれぞれの統計を分析し、自分の場合によく合うように相対発生頻度を調整するとよい。その他の統計データとしては、つぎの文献のものが有用である。AKIY71, BELF79, BOEH75A, BOIE72, DNIE78, ELSH76B, ENDR75, GANN76, GILB77, GOEL78B, HAUG64, HOFF77, ITOH73, LITE76, REIF79A, RUBE75, SCHI75, SCHN75, SCHN79A, SCHW71, SHOO75, WAGO73。

バグの統計－1

サンプルサイズ：6,877,000ステートメント（コメント文を含む）

全バグ数：16,209件,

1k ステップ当たりのバグ件数：2.36

1xxx	要求仕様の不良	1317	8.1%
11xx	要求仕様の誤り	649	4.0
12xx	要求仕様の論理の不良	153	0.9
13xx	要求仕様が不完全	224	1.4
15xx	表現, ドキュメンテーションの不良	13	0.1
16xx	要求仕様の変更	278	1.7
2xxx	概念仕様不良と機能不良	2624	16.2%
21xx	概念仕様／機能の誤り	456	2.8
22xx	概念仕様の誤り	231	1.4
23xx	機能の場合分け誤り	193	1.2
24xx	領域誤り	778	4.8
25xx	ユーザメッセージと診断の不良	857	5.3
26xx	例外条件取扱い不良	79	0.5
29xx	その他の機能不良	30	0.2
3xxx	構造不良	4082	25.2%
31xx	制御フローと構造の不良	2078	12.8
32xx	処理不良	2004	12.4
4xxx	データ不良	3638	22.4%
41xx	データ定義と構造の不良	1805	11.1
42xx	データアクセス取扱い不良	1831	11.3
49xx	その他のデータの不良	2	0.0

バグの統計ー 2

5xxx	コーディング不良	1601	9.9%
51xx	コーディング不良と型不良	322	2.0
52xx	スタイル違反と標準違反	318	2.0
53xx	ドキュメンテーション不良	960	5.9
59xx	その他のコーディング不良	1	0.0
6xxx	統合不良	1455	9.0%
61xx	内部インタフェース不良	859	5.3
62xx	外部インタフェース, タイミング, スループット不良	518	3.2
69xx	その他の統合不良	78	0.5
7xxx	システム, ソフトウェアアーキテクチャ不良	282	1.7%
71xx	O/S 機能の使用誤り	47	0.3
72xx	ソフトウェアアーキテクチャの不良	139	0.9
73xx	回復処理不良	4	0.0
74xx	処理能力不良	64	0.4
75xx	診断不良, 例外処理不良	16	0.1
76xx	パーティション不良, オーバレイ不良	3	0.0
77xx	システムジェネレーション不良, 環境設定不良	9	0.1
8xxx	テストの設計と実行不良	447	2.8%
81xx	テスト設計不良	11	0.1
82xx	テスト実行不良	355	2.2
83xx	テストドキュメンテーションの不良	11	0.1
84xx	テストケース不完全	64	0.4
89xx	その他のテストバグ	6	0.0
9xxx	その他	763	4.7%

表2.1 バグの統計