



Title	教育環境におけるエラーデータ収集ツールの試作
Author(s)	柏本, 隆志; 神谷, 年洋; 楠本, 真二 他
Citation	電子情報通信学会ソサイエティ大会講演論文集. 1997, 1997, p. 19-19
Version Type	VoR
URL	https://hdl.handle.net/11094/51099
rights	copyright©1997 IEICE
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

<https://ir.library.osaka-u.ac.jp/>

The University of Osaka

D-3-5

教育環境におけるエラーデータ収集ツールの試作

Fault data collection tool on educational environment

柏本 隆志[†] 神谷 年洋[†] 楠本 真二[†] 井上 克郎[†] 毛利 幸雄[‡]
 Takashi KASIMOTO[†] Toshihiro KAMIYA[†] Shinji KUSUMOTO[†] Katsuro INOUE[†] Yukio MOHRI[‡]

[†]大阪大学大学院基礎工学研究科情報数理系[‡]日本ユニシス株式会社[†]Graduate School of Engineering Science, Osaka University [‡]Nihon Unisys Corporation

1 はじめに

近年、ソフトウェアは大規模・複雑化する一方で、納期の短縮化、コスト削減、品質の向上が求められている[3]。ソフトウェアの品質や開発の生産性を向上させるための一つの手法としてエラー分析がある。本研究では、エラー分析の支援を目的として、信頼性の高いエラーデータを収集するためにデータ入力を半自動化したエラーデータ収集ツールを試作する。

2 エラー分析

エラー分析とは、ソフトウェアの開発段階や利用段階で発見されたエラーについて、その原因や混入過程の分析を行ない、その結果に基づいて開発環境を改善することにより、エラーの再発を防止する手法である[1]。エラー分析を行なうには、信頼性の高いデータの収集が必要となる。

しかし、実際にはエラーデータの収集には、多くのコストがかかることや収集したデータが正確でないことが多いという問題点が指摘されている。この原因としては、データ収集の経験が少ない開発者がエラー収集を行なうということやデータの収集が基本的に手作業で行なわれるということを考えられる。データ収集ができるだけ自動化したツールの開発が必要となってきている。

3 エラーデータ収集ツール

3.1 収集データ

エラーの発見、修正に関連するレビュー作業、テスト作業、デバッグ作業の各段階における作業時間の情報、発見/修正されたエラーの原因、エラーの場所、エラーの内容、修正内容等を収集対象とする。

3.2 ツールの概要

Windows95 上で動作するツールを MFC(クラスライブラリ)を用いて、C++で実装した。プログラムのサイズは約7000行である。ツールの実行画面の一例を図1に示す。この例では、デバック作業中に、エラーを検出した時の情報の入力が行われている。

4 適用例

ツールを日本ユニシス株式会社の新人研修における C++ プログラム開発演習に適用し、エラー分析を行った[2]。この演習では、8 チームがそれぞれ、酒屋問題プログラムを Visual C++ 開発環境を用いて作成した。エラーの分類結果

(個数、修正時間、各エラー毎の平均修正時間)を表1に示す。表1から、インターフェースエラーが多く作り込まれたことがわかる。また、個数は少ないが、クラス階層(の設計誤り)についてのエラーの修正に多くの時間を要していることが確認できる。

5まとめ

エラーデータ収集ツールを試作した。今後の課題としては、収集したデータを分析するための機能(例えば、統計処理)を充実させること、チームによるソフトウェア開発に対応できるようにすることなどがある。

参考文献

- [1] 日野: “ソフトウェア障害の分析と予防(高信頼性運動として)”, 第4回ソフトウェア生産における品質管理シンポジウム論文集, pp.91-98(1985).
- [2] 毛利他: “新人研修におけるチーム作業の実験的評価-プログラム完成度の評価と要因分析”, 第13回ソフトウェアシンポジウム論文集, pp.57-62(1992).
- [3] 山田, 高橋: “ソフトウェアマネジメントモデル入門”, 共立出版(1993).

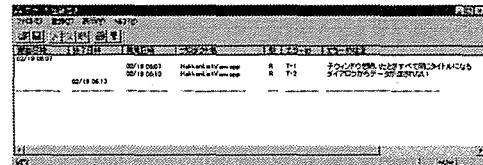


図1: エラーデータ収集ツールの画面例

表1: エラー分析例

タイプ	個数	修正時間	平均修正時間
クラス階層	2	520	260
メソッド不足	8	450	56
インターフェイス	22	2000	91
メンバ関数	13	1355	104
制御	3	722	241
場合分け	3	36	12
初期化	6	396	66
誤解	5	470	94
その他	2	55	28