

安藤 弘毅†, 山本 徹也†, 西岡 竜大†, 片岡 欣夫†, 植木 克彦†, 平山 雅之†

### 1. はじめに

現在のソフトウェア開発においては、開発すべき規模は年々増大し続けており、その一方で開発期間は逆に短縮化している。このような状況でソフトウェアの品質を維持するにはテスト・レビュー等を強化する必要があるが、規模の増加と期間の短縮化から、すべての箇所をレビューすることや、全ての機能を網羅的にテストすることは困難であり、テスト・レビュー等の効率を上げる必要がある。

テスト・レビュー等の効率を上げる為には、不具合が多く発生する箇所を予め同定することが有効である。従来から、ソフトウェアメトリクスと不具合には相関があることが知られており、この相関を用いて不具合が多く発生すると予想される箇所を知ることができる。

しかしながら現在の開発の大部分は、まったく新規に作成するよりも、むしろ既存のプログラムを再利用して変更や機能追加を行う方が多い。このような場合、変更された割合によって、不具合の出方及びメトリクスと不具合の相関が異なることが予想される。今回我々は、変更・追加によって開発されたソフトウェアに対して、変更量を用いてファイルを分類し、メトリクスと不具合との相関について調査した。

### 2. メトリクスと不具合の相関

#### 2.1. 計測対象

計測対象はC言語で記述されたソフトウェアであり、規模は次の通りである。

- ・言語 : C言語
- ・ソースファイル : 約 1200
- ・ステップ数 : 約 50 万行(ELOC)
- ・関数の数 : 約 2 万

このソフトウェアウェアは新規に開発されたものではなく、元となるソースに対して機能を変更・追加して開発されたものである。

#### 2.2 ファイルの分類

計測対象のソフトウェアを元になったソースと比較して変更量を算出し、変更の有無によって対象のソースファイルを以下の三つに分類した。

- ・流用ファイル
  - ・元になったソースをそのまま利用しているファイル
- ・変更ファイル
  - ・元になったソースの一部を変更したファイル
- ・新規ファイル
  - ・元になったソースには存在しない、元になったソースファイルの関数をそのまま利用しているファイル

†株式会社 東芝

図1にそれぞれの分類の全体に対する割合を示す。新規ファイルの総ステップ数は、全体の20%程度を占める程度だが、不具合件数は全体の40%を占めている。

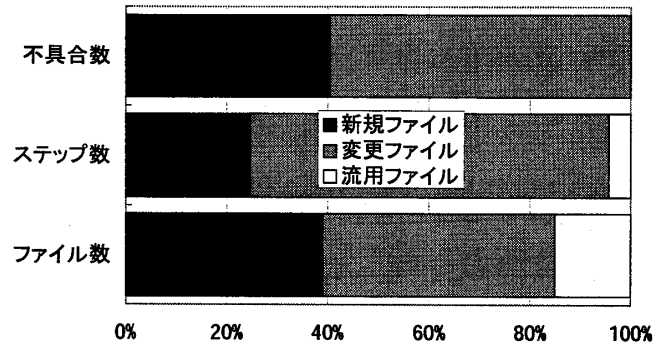


図1

表2に1 KLOCあたりの不具合の発生件数を示す。変更ファイルでは、全体のステップ数からは、新規ファイルの半分程度の割合でしか不具合が発生していないが、変更量で見れば新規ファイルと同程度以上の不具合が発生している。

表1

区分	不具合密度(ステップ数) [/KLOC]	不具合密度(変更量) [/KLOC]
新規ファイル	9.8	9.8
変更ファイル	5.0	11.6

#### 2.3. メトリクスと不具合の相関

分類された新規ファイルと変更ファイルそれぞれについてメトリクスを計測し、不具合発生件数との相関を調べた。今回の対象では、流用ファイルにおいて不具合が発生していないため除外した。なお使用している不具合情報は、結合テスト時に発見された不具合のファイル毎の発生件数である。使用したメトリクスを以下に示す。

##### (1) ステップ数

ファイルのステップ数であり、ファイルの規模を表す。ファイル内に存在する関数のステップ数の総和で算出。このステップ数はコメント、空白行等を除いた実行行数(ELOC)である。

##### (2) 変更量

ファイルの変更されたステップ数。ファイル内の関数を元の関数と行単位で比較し変更されたステップ数を算出、その総計をファイルの変更量とした

##### (3) CC $\geq$ 10 (1)

ファイルの複雑度を表す。ファイル内で複雑度が 10 以上であった関数の総数であり、関数の複雑度は McCabe の複雑度の指標を使用している。McCabe の複雑度の値は、10 以下であることが望ましいとされている。

(4)  $CC \geq 10$  (2)

変更された関数のみを考慮したファイルの複雑度を表す。ファイル内で、新規に追加或いは変更が行われ、且つ複雑度が 10 以上である関数の総数。

2.4. 計測結果

図 2 及び表 2 に各ファイルで発生した不具合の総数と、メトリクスとの相関を示す。

表 2

メトリクス	ステップ数	変更量	$CC \geq 10$ (1)	$CC \geq 10$ (2)
新規ファイル	0.54	0.54	0.53	0.53
変更ファイル	0.49	0.61	0.56	0.63

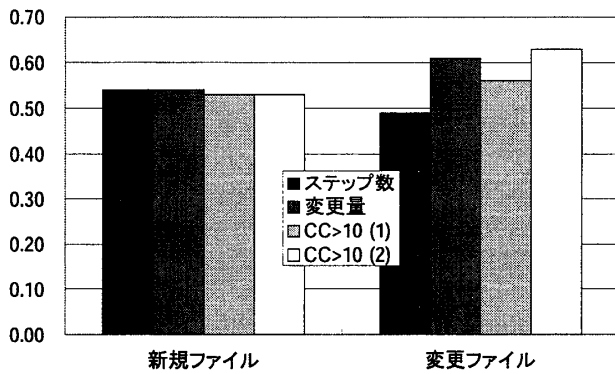


図 2

新規ファイルでの不具合件数と相関は、ステップ数と複雑度で同程度である。変更ファイルでの不具合件数との相関は、複雑度の方が若干高いがその差はわずかである。

変更ファイルの内部には、変更された関数、追加された関数、流用された関数が含まれている。このため、変更ファイルに分類されたファイルには、新規ファイル又は流用ファイルに近いファイルが混ざっている。これらを取り除けば、より特徴がでることが予想される。表 3 は、新規ファイルに近いファイルと流用ファイルに近いファイルを除いて、相関を調べた結果である。また図 3 は、複雑度と不具合件数のグラフである。複雑度との相関がより高くなっている。

表 3

	ステップ数	変更量	$CC \geq 10$ (1)	$CC \geq 10$ (2)
変更ファイル2	0.55	0.61	0.62	0.66

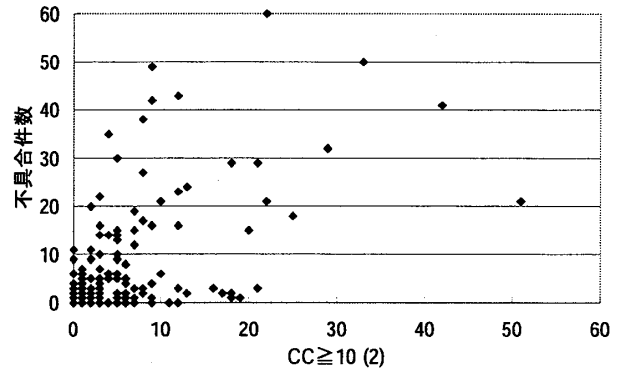


図 3

3. 考察

変更ファイルでは、変更を考慮した複雑度である  $CC \geq 10$  (2) と変更量が、単純なステップ数と複雑度より不具合件数と高い相関を示している。この原因は以下の様に考えられる。

変更部分のソースは新規部分と異なり一から作成せず、元になるソースに変更・追加を加えて開発されたことである。元になったソースは既に一通りのテストを経ており、その際に発見された不具合は既に修正されているため、基本的に新規に作成された箇所より品質は高いと言える。このため、変更部分において不具合が発生するのは、新たに変更・追加された箇所近辺である可能性が高く、全体量を表すステップ数より、変更を行った量の方が不具合との相関が高くなる。

変更や追加によって開発されたソフトウェアに対して、不具合が発生しやすい箇所をメトリクスによって特定するには、そのソース単体を計測しメトリクスを得るだけではなく、元になったソフトウェアのソースと比較し、変更量によってファイルを分類した後、複雑度によって不具合の発生しやすい箇所を特定する

4. まとめ

ソフトウェアのソースファイルを流用ファイル、変更ファイル、新規ファイルの三つに分類し、それぞれメトリクスと不具合との相関を調べた。変更部分については、変更量と複雑度が不具合件数との相関が高いことを示した。変更や追加によって開発されたソフトウェアに対して、メトリクスから不具合を予測する場合は、元になったソフトウェアからの変更量を考慮し、変更箇所では複雑度を用いて予測することが有効だと言える。今回は利用できなかったが、変更頻度等の情報も利用できれば有効であろう。

参考文献

- (1) McCabe, 「A complexity measure」, IEEE Trans. Softw. Eng., 1978
- (2) 篠原他, 「C++ソフトウェアにおけるバグとメトリクスデータとの相関」, 情報処理学会第 59 回全国大会