

開発者行動を考慮したソフトウェア信頼性モデル

本田 澄^{†1} 鷺崎 弘 宜^{†1} 深澤 良 彰^{†1}

ソフトウェアの欠陥はリポジトリシステムによって管理されることが多くなっており、開発からテストといったソフトウェア開発の全工程を管理することも往々にして行われている。そこで、我々はリポジトリシステムで管理されている欠陥データと開発者の活動履歴から、それぞれの欠陥に対して発見した時間を取得、ソフトウェア信頼度成長曲線に適応しグラフ化を行う。これにより予測される欠陥数を定量的に把握することができ、欠陥数を見積もることや、開発の進捗状況を確認することができる。本研究は Open Source Software について、公開されているリポジトリから欠陥情報と開発者履歴を取得し、開発者の変動を組み込んだモデルを用いて、欠陥数の予測を行った。

Software Reliability Model Considering Developers' Activity

KIYOSHI HONDA,^{†1} HIRONORI WASHIZAKI^{†1}
and YOSHIAKI FUKAZAWA^{†1}

Software development requires repository system to manage the whole developing phases such as coding and testing. We acquire the fault numbers, faults data and developers' activities, which have been recored in a repository system, based on the software reliability growth model. It will be expected that these predicted number of faults can motivate the developers because these predicted data suggest quantitative goals to the developers. In this research, we focused on Open Source Software and predicted the number of faults based on the model which can treat the dynamics of developers, which can be collected from published repository.

1. はじめに

本研究の目的は、ソフトウェア開発において発生する欠陥数を予測するものである。多くのソフトウェアでは開発中に欠陥の有無を調べ、もし欠陥が発見されれば修正されなくてはならない。特に開発中に発見される欠陥の数を見積もることができないことは問題である。そこで我々は開発中に発見された欠陥数を用いて、今後どれだけの欠陥が発見されるかについてソフトウェア信頼度成長曲線に基づき予測した。また、今回用いたソフトウェア信頼度成長曲線については我々が提案した新たなソフトウェア信頼度成長モデル¹⁾²⁾である。このモデルの特徴は開発者の開発時間や開発人数をソフトウェア信頼度成長モデルに反映することができ、加えて開発における様々な不確実な要素を考慮可能なモデルである。また、Open Source Software(OSS)を対象とし、開発者の変動を組み込んだモデルを用いて、そのモデルの精度を求めた。

2. 簡略化した一般化ソフトウェア信頼性モデル

既存研究として様々な提案されているモデルに対して、我々は一つのモデルで、様々なモデルについて適用可能な一般化されたモデルを提案する。まず、我々が仮定することを以下にあげる。

- (1) 単位時間当たりに発見できる欠陥の数は、残存欠陥数と発見できた欠陥数の積に比例する。
- (2) テストに割ける時間は一定ではなく、時間変化を持つ。
- (3) 単位時間当たりに発見できる欠陥の数は、不確定な要素を持つ。

まずはじめに、仮定(1)については、欠陥が発見されないテスト初期の段階と、残る欠陥が少ないテスト終了の段階では、単位時間当たりに発見できる欠陥は少ないことを考慮している。これはテスト初期の段階では、欠陥を見つけずらく、ある程度欠陥が発見された段階で、発見できる欠陥が増えることを反映している。また、テスト終了の段階では残る欠陥が少ないため、発見できる欠陥が少ないことを反映している。

^{†1} 早稲田大学
Waseda University

$$\frac{dN(t)}{dt} = a(N_{max} - N) \quad (1)$$

α は単位時間あたりに発見できる欠陥の数, N は時刻 t における欠陥の数, N_{max} は予測される最大欠陥数を示す. この微分方程式に対して, 仮定 (2)(3) を適用する. 仮定 (2) (3) として, テストに割ける時間が一定ではなく, 時間変化をもつこと, 単位時間あたりに発見できる欠陥の発見数が不確定な要素を持つことを挙げた. 時間に依存するとは, テスト作業やデバッグ作業が実施可能な時間によることを考慮した仮定である. 例として, テストチームが他の業務に時間を割かれ, テストの実施を行えなかったことが考えられる. また, 不確定性についてはテスト作業やデバッグ作業にあたって作業できる数に対して, ある程度のばらつきを考慮した仮定である. 例として, テストにおいて一日に発見できる欠陥数に対して, テストチームの能力により変化し, それは概ねばらつきがあるものと考えられるためである. これを考慮し, テスト実施数について時間的変動と不確定性を含んだ以下の式を導入する.

$$a = \alpha(t) + \sigma dw(t) \quad (2)$$

$\alpha(t)$ は時間に依存した項である. σ は不確定性に起因するばらつきである. $dw(t)$ についてはばらつきが従う確率分布である正規分布 (ガウシアンホワイトノイズ) である. 本研究では開発者数の変動を考慮するため, 簡略化のため, $\sigma dw(t) = 0$ として以下の式を用いる.

$$dN(t) = \alpha(t)(N_{max} - N)dt \quad (3)$$

3. 開発者数の変動

特に OSS では開発者の変動が大きく, 初めて開発に加わる開発者や一度開発に加わったが離脱する開発者といった様々な種類がある. 図 1 に GitHub で公開されている OSS の一つである Web フレームワークの foundation についてその開発者数の変動を月単位で示す. また, 開発者の変動のみでなく Issue の数についても示す. 図 1 Foundation における開発者数の変動と Issue の増加が示すように, 特に version 4 では開発者数がそれまでの version より増加し, Issue の増加がそれまでの version より大きくなっていることが見て取れる.

4. 開発者の変動を考慮した GSRM

図 1 Foundation における開発者数の変動と Issue の増加からわかるように, Issue と開発者数には関係性があると考えられるため, 我々が提案した GSRM へ開発者数の要素を用いてモデル化を行った. 結果

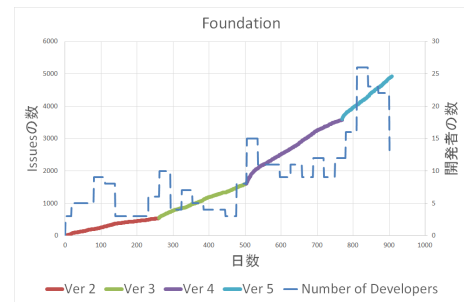


図 1 foundation における開発者数の変動と Issue の増加

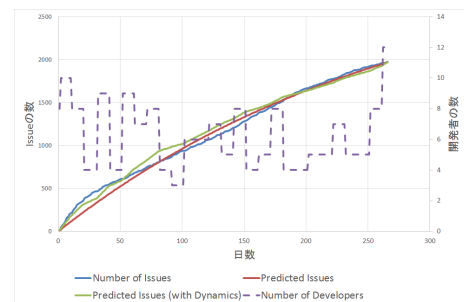


図 2 foundation に対する開発者数の変動を考慮した GSRM

として, 開発者数の変動を考慮した GSRM については開発者数の変動を考慮しない GSRM より, 実データより外れたモデルであった. 評価方法としては残差平方和を用いた. 開発者数の変動を考慮しない場合, RSS は 9.77×10^5 開発者数の変動を考慮した場合, RSS は 1.33×10^5 であり, 考慮した場合のモデルの方が実際のデータにフィットしていないことがわかった.

5. 考 察

今回は開発者数を GitHub から収集したが, その際にある期間内で一行だけコミットされた開発者を一人と数えたため, より多くのコミットを行った開発者と同等に扱っている. そのため, 開発者の変動をモデルに組み込んだ際にずれが生じたと考えられる. 今後の展望として, 頻度の少ない開発者を無視, もしくは別として取扱いモデルに組み込む必要がある.

参 考 文 献

- 1) Honda, K., et al. "A Generalized Software Reliability Model Considering Uncertainty and Dynamics in Development", PROFES 2013 (2013)
- 2) Honda, K., et al. "Predicting Time Range of Development Based on Generalized Software Reliability Model" APSEC 2014 (2014)