CIツールとリポジトリシステムを用いた欠陥数予測

Prediction Of Faults Using CI tool and Repository System

本田 澄, 中井 秀矩, 鷲崎 弘宜, 深澤 良彰

khonda@ruri.waseda.jp, hide-and-seek@toki.waseda.jp, {washizaki, fukazawa}@waseda.jp
早稲田大学 基幹理工学部 情報理工学科

森 牧, 小川 健太郎, 高橋 一貴

{makimori, keogawa, takakaha}@yahoo-corp.jp

ヤフー株式会社

発表要旨:

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、リポジトリシステムと CI(continuous integration)ツールを連動させ、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 CI ツールへのプラグインとして、リポジトリシステムで管理されている欠陥データから、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、今後 発生すると予測される欠陥数についてグラフ化を行い、CI ツールにて確認できる環境の構築を行った.

我々が開発したプラグインを用いることで、予測される欠陥数を定量的に把握することができ、開発者やマネージャが悩んでいた欠陥数を見積もることができ、開発の進捗状況を日々確認することができる。加えて、残りの欠陥について後どれだけの欠陥を発見すればよいか定量的な目標ができ開発者のモチベーションの向上や維持につながることが期待される。本研究は共同研究企業であるヤフー株式会社とともに有効性を検証している。

キーワード:

欠陥予測, CI ツール, リポジトリシステム, ソフトウェア信頼度成長曲線

想定している聴衆

開発者・マネージャ・品質部門

発表者の紹介(全角100文字):

早稲田大学 基幹理工学研究科情報理工学専攻 博士課程 本田 澄

国際会議 Profes2013 にて"A Generalized Software Reliability Model Considering Uncertainty and Dynamics in Development"を発表.

^{*} 副題は不要であれば行ごと削除してください

CIツールとリポジトリシステムを用いた 欠陥数予測

早稲田大学 基幹理工学部 情報理工学科

〇本田 澄, 中井 秀矩, 鷲崎 弘宜, 深澤 良彰

e-mail: khonda@ruri.waseda.jp

ヤフ一株式会社

森 牧, 小川 健太郎, 高橋 一貴

CI(継続的インテグレーション)ツール

- □ CI(継続的インテグレーション)とは
 - プログラマーが開発の際に品質改善や納期短縮のために行う習 慣および方法
 - 例:自動ビルド,単体テストの自動実行
- Clツール



http://jenkins-ci.org/

Jenkins



- https://circleci.com/
- Circle Cl

Travis CI



https://travis-ci.org/

- □ プラグイン
 - CIツールを拡張できる
 - Cobertura Plugin:テストカバレッジレポート
 - FindBugs Plugin:潜在的な不具合やパフォーマンスに影響する コードをレポート

リポジトリシステム

- □ リポジトリシステムとは
 - 開発プロジェクトに関連するデータを一元的に保存,活用するシステム
- □ リポジトリシステム
 - GitHub

GitHub

https://github.com/

Subversion

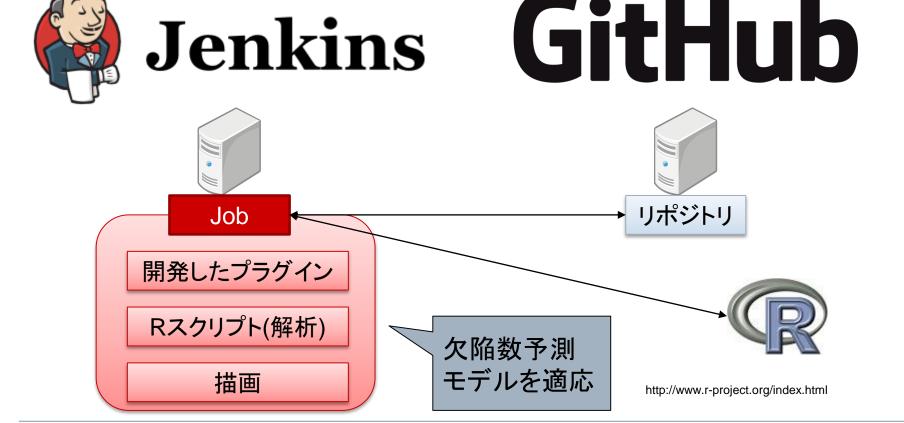
SUBVERSION°

http://subversion.apache.org/

- □ 利用例
 - CIツールとの連携
 - □ テスト自動化等
 - 欠陥の管理

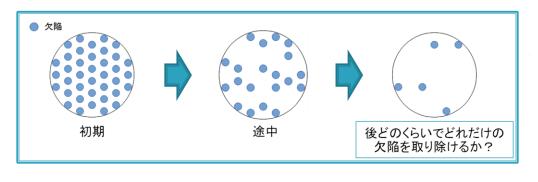
欠陥数予測の仕組み

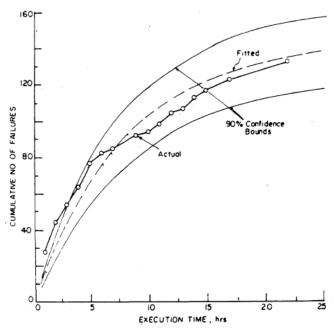
- JenkinsとGithubを用いて、自動的に欠陥数を取得し欠陥数予測モデルを適応
 - Jenkinsで利用できるプラグインを開発し自動的に欠陥数を取得



欠陥数と信頼性

- □ 1980年頃から行われた研究
- □ 欠陥数の予測について
 - ソフトウェアの"信頼性"に対して、欠陥を十分に発見できたかどう かを評価する基準
 - (発見された)欠陥数の成長を モデル化し、今後どれだけの 欠陥が発見されるかを予測

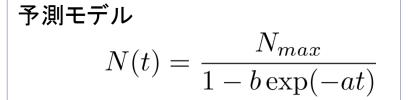


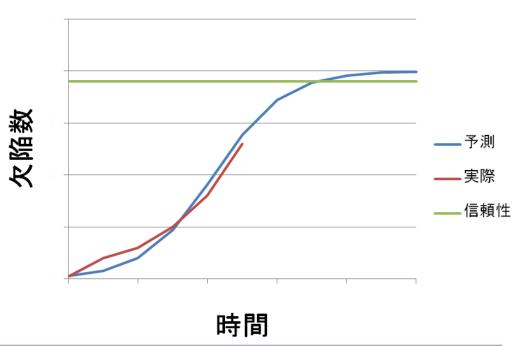


AMRIT L. GOEL, "Software Reliability Models: Assumptions, Limitations, and Applicability" IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. SE-11, NO. 12, 1985

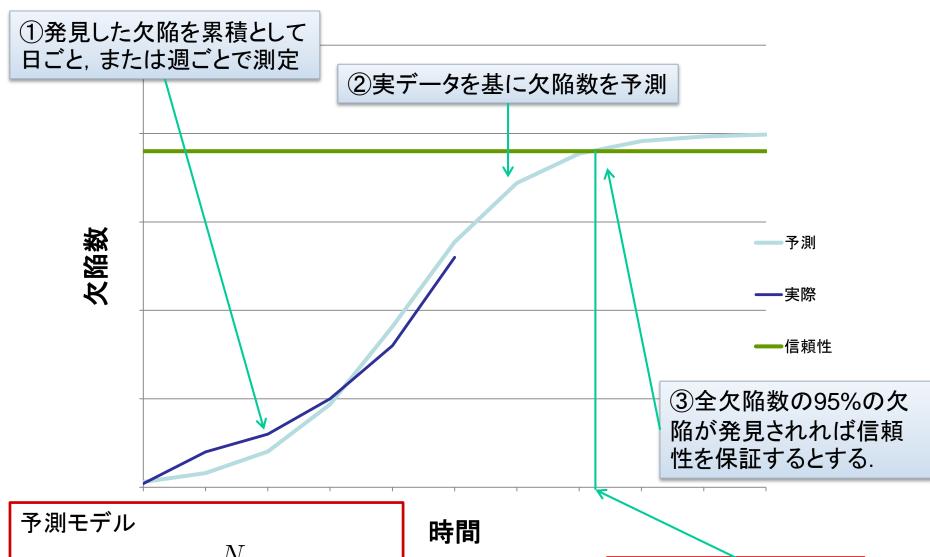
欠陥数予測

- □ 欠陥数を予測し、取り除くべき欠陥数を定める
 - 後どれだけテスト工数に時間を割けばよいか
- ソフトウェア信頼性モデル
 - 発見した欠陥数とかかった時間から、全欠陥数を予測
 - どれだけテストを行えばよいか, リリース時期の指標となる.
 - □ 例)予測した欠陥数の95%の欠陥が取り除かれればリリース





信頼性モデル

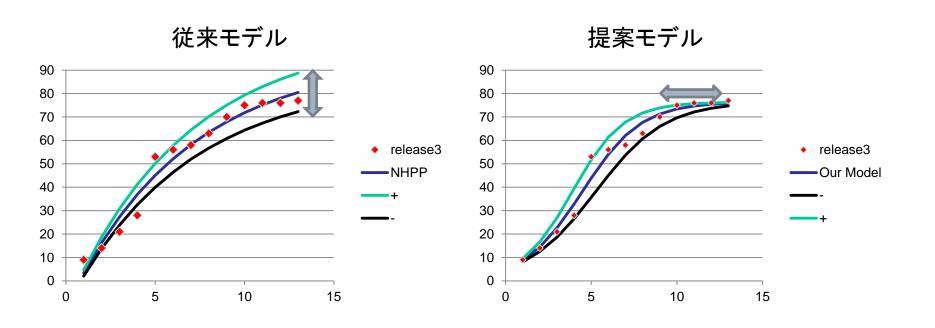


 $N(t) = \frac{N_{max}}{1 - \exp(-at)}$

おおよそ, 95%の欠陥 が発見できる.

開発における問題

- □ いつまでテストを行えばよいのか?
 - 欠陥がなくなるまでテストを繰り返す?
 - →期間やコストは限られている
 - 欠陥の検出件数の計画を決め、その値(計画値)とのかい離を評価 しなければならない.

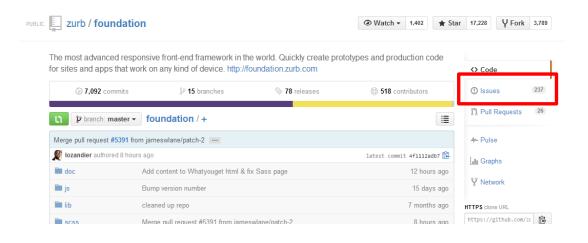


開発における問題

- □ 欠陥を十分に発見できているか
 - 予測される欠陥数に対して今どれだけ欠陥を見つけたか.
 - 例)80%発見できた
 - □ リリースまで残り15%の欠陥を見つける必要がある
- □ リリースに間に合うか判断できない
 - 開発のスコープの決定を支援
 - □ このモジュールの開発はスコープから外す
 - テスト工数,計画の決定を支援
 - □ このモジュールは十分に欠陥を発見できているので他のモジュールのテスト工数を増やす

測定対象と出力

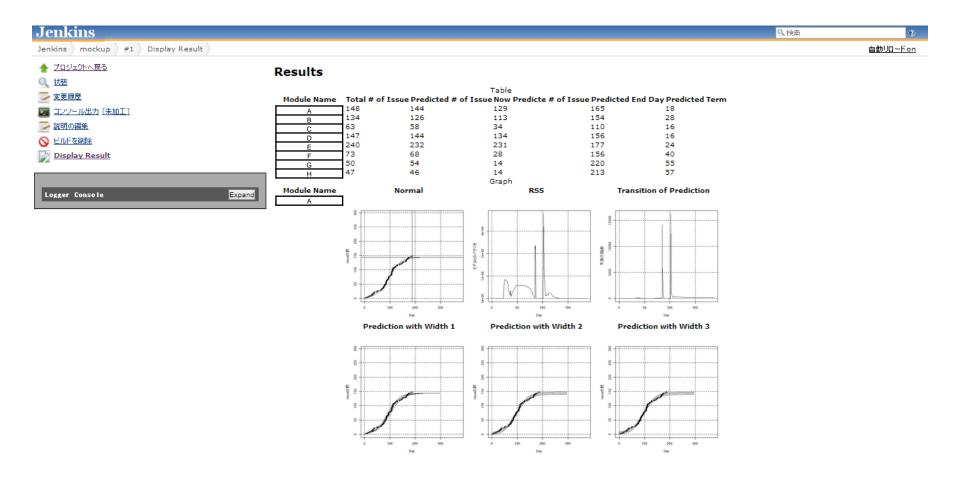
- → 対象のプロジェクトから欠陥数と欠陥の発見時間を抽出
 - 例: https://github.com/zurb/foundation



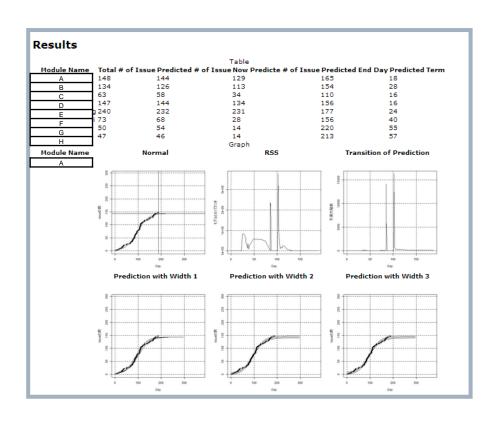
- □ 欠陥の累積数と発見時間をRへ
- Rを用いて予測モデルとのフィットを計測
- □ 欠陥数,予測欠陥数,現時点での予測欠陥数,終了予測日,予測終了期間を数字で表示
- 予測モデルをグラフで表示

開発ツール

■ Jenkins Pluginとしてツールを開発



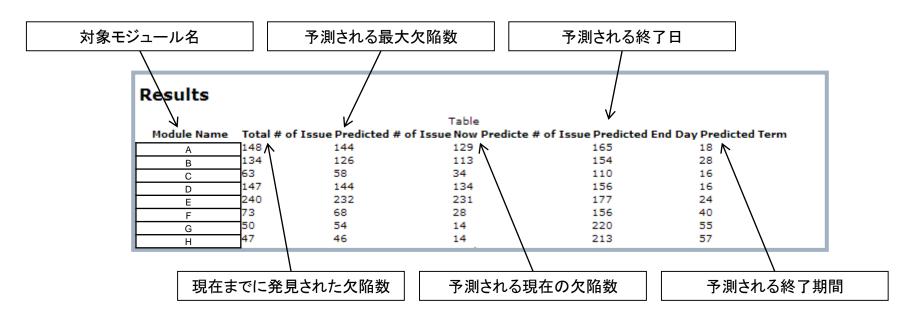
表示内容



- → 上部に表形式で欠陥数, 予測欠陥数,現時点での 予測欠陥数,終了予測 日,予測終了期間を数字 で表示する.
- 下部にグラフで日数-欠陥 数(Normal), 日数-予測モ デルのズレ(RSS), 日数-予測欠陥数(Transition of Prediction), 予測モデル Predictión with Width 予測モデル Prediction with Width 予測モデル Prediction with Width を表示する.

表示内容(表)

□ 表が示す意味



対象モジュール名: 対象とするプロジェクトに属するモジュール名を示す。

• 予測される最大欠陥数: 予測モデルから求まる最大の予測欠陥数を示す.

• 予測される終了日: 開発開始から最大欠陥数の95%の欠陥を発見するまでの日数を示す.

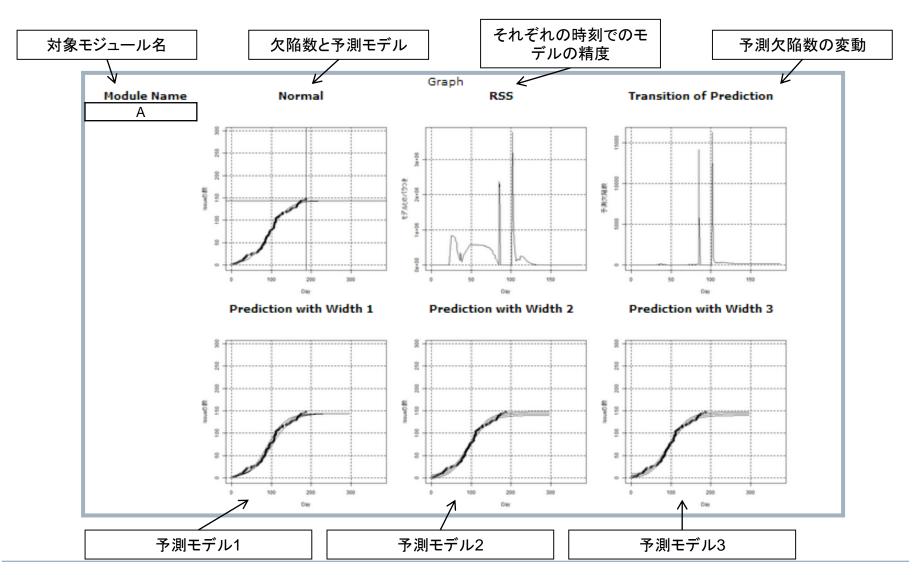
現在までに発見された欠陥数: Github上に記録されたIssueの数を示す。

• 予測される現在の欠陥数: 予測モデルを適応し,現時点までの予測欠陥数を示す.

• 予測される終了期間: 予測される終了日からどれだけの日数のズレがあるかを示す.

表示内容(グラフ)

□ グラフが示す意味



表示内容(グラフ)

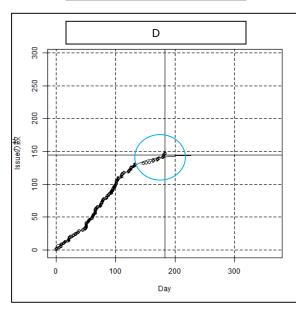
- □ 欠陥数と予測モデル
 - 日数ごとにそれまでに発生したIssueの数をまとめ、ドットで数を示す.また、Issueの最終発生日に予測モデルを適応したものを曲線で示す.
- □ それぞれの時刻でのモデルの精度
 - Issueの発生日それぞれで適応したモデルと実際のIssueとのズレを曲線で示す.
- □ 予測欠陥数の変動
 - Issueの発生日それぞれで適応したモデルが示す予測欠陥数を曲線で示す.
- □ 予測モデル1
 - 開発の終了に近付くにつれて不確実な要因が大きくなるモデルを曲線で示す。この要因が大きく働く場合と小さく働く場合を上下の曲線で示す。
- □ 予測モデル2
 - 開発のすべてにおいて一定の不確実な要因が働くモデルを曲線で示す。 この要因が大きく働く場合と小さく働く場合を上下の曲線で示す。
- □ 予測モデル3
 - 開発の初期段階において不確実な要因が大きくなるモデルを曲線で示す。この要因が大きく働く場合と小さく働く場合を上下の曲線で示す。

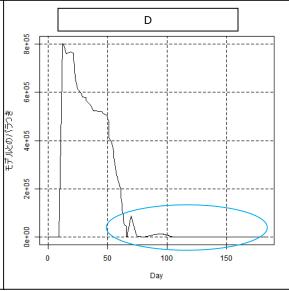
利用方法(例)

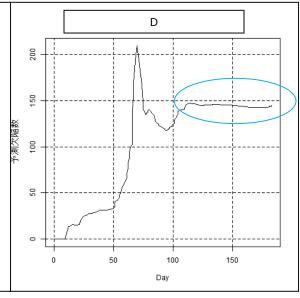
欠陥数と予測モデル

それぞれの時刻でのモ デルの精度

予測欠陥数の変動

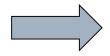






- 1. 現状で発見した欠陥の数がモデルが示す予測欠陥数より多い.
- →予測より多く発見できており, テストが 十分に行えている.
- 2. これまでのモデルと発見した欠陥の数とのバラつきに変化が見られない.
- →発見した欠陥と現在のモデルが精度 よく求まっている.
- 3. これまでのモデルが示す予測欠陥数に変化が見られない.
- →予測欠陥数が収束しており、今後新 たな欠陥が発見されることは少ない.

Module Name	Total # of Issue	Predicted # of Issue	Now Predicte # of Issue	Predicted End Day	Predicted Term
D	147	144	134	156	16

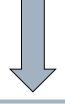


テストによる欠陥の発見は十分に行えており, テストの労力をデバッグに割り当てよう.

計測結果

	予測欠陥数	発見欠陥数	。 逐正済み欠陥数	潜在欠陥数	終了予測残り日数(95%発見)	区間
А	173	117	93	80	57	17.0051
В	96	107	97	-1	1	-
С	53	52	48	5	1	-
D	145	124	103	42	43	72.41165
E	230	198	186	44	25	58.13167
F	97	58	57	40	76	14.61955
G	49	41	38	11	60	86.91578
Н	33	35	25	8	-	-

※2013年8月30日時点



発見欠陥数が予測欠陥数を超えており、十分なテストが行えたと考えられる.

	予測欠陥数	発見欠陥数	多正済み欠陥数	潜在欠陥数	終了予測残り日数(95%発見)	区間
Α	144.09	148	115	29.09		-
В	126.72	134	126	0.72	-	-
С	58.5	63	58	5	1	-
D	144.6	147	128	16.6	-	-
E	232.88	240	228	4.88	-	-
F	68.08	73	66	2.08		-
G	54.63	50	45	5		-
Н	46.47	47	34	12.47	-	_

※2013年11月26日時点

まとめと今後の予定

□ まとめ

- CIツールとリポジトリシステムを利用し、欠陥予測を行うツールを開発
- 開発者からのフィードバッグを受け予測が概ね正しいことを確認

□ 今後の予定

- ドメイン(特定領域)の異なるプロジェクトでは開発環境や要求の内容に違いがあると考えられ、不具合の発生やその挙動に違いがあることが考えられる.
 - □ 例:事務処理プログラムと制御系プログラムでは対象とするプログラムの仕様や要求は異なる
- ドメイン特有の不具合の発生や挙動を特定することで不具合予測の精度を高める。