オープンソース開発におけるソフトウェアテストの実態調査

プロジェクトマネジメントコース　矢吹研究室　1040060　清水 竜吾

1. 研究背景

これまでのソフトウェア開発の現場では，主にウォーターフォール型開発手法が採用されていた．それに代わってアジャイル型開発手法が普及してきている．ウォーターフォール型開発手法は仕様を最初にすべて決めてしまうため，スケジュール管理は容易だが，開発途中の仕様変更は難しい．その一方で，アジャイル型開発手法は簡単なウォーターフォール型開発手法をサイクルして開発を行うため，開発途中の仕様変更が容易である．

アジャイル型開発手法では，テスト駆動開発手法（Test Driven Development）がよく採用される．これは，先にプログラムに必要な機能について初めにテストを考案し，そのテストが動作する必要最低限な実装を行い，コードを洗練させ，短い工程で計画・開発を繰り返し行う手法である．

実際の製品として販売・配布されている多くのソフトウェアはテストを含めて，開発プロセスに関する情報は一般には知ることができなかったが，近年では開発プロセスに関する情報を一般でも見ることができるオープンソースソフトウェア（OSS）の開発が盛んである．OSS開発には，OSSホスティングサービスを利用しての開発されることが多い．ホスティングサービスとはソースコードやドキュメントのバージョン管理するシステムやバグ・テスト情報を記録できるシステムなどの複合システムである．

最もよく利用されているホスティングサービスの一つがGitHubである．実際，GitHubの登録リポジトリ数は，サービスを開始した2008年には38,423件だったのに対して，2012年には4,614,306件と大幅に増加した．ユーザ数も同期間に41,157人から2,763,437人へと大幅に増加した．

このように多くのプロジェクトをホストするGitHubのプロジェクトを調査・解析することによって，近年のソフトウェア開発傾向を調べることができると思われる．

1. 研究の目的

本研究では，OSSのソフトウェア開発工程であるテスト工程に着目してソースコードの成長の様子を調査し，ソフトウェア開発プロセスを明らかにすることを目的とする．

1. 研究方法

GitHubにホスティングされているソフトウェアのソースコードをメインコードとテストコードに分け，それぞれの成長の様子を調べる。時系列として、コミットごとにテストコードとメインコードの行数を計測することによって，時系列データを作成する．取得した時系列データで折れ線グラフを作成することによって、データの可視化をすることができる。

1. 結果・考察

調査した結果，以下のようなグラフの結果が得られた．

**③**

**②**

**①**

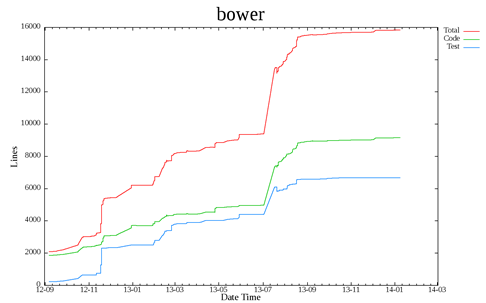
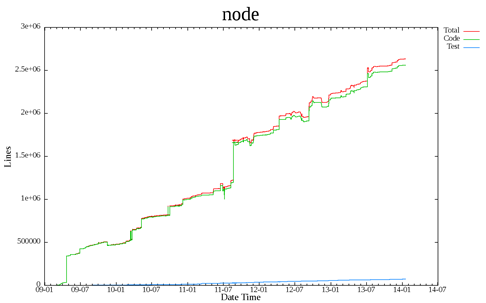
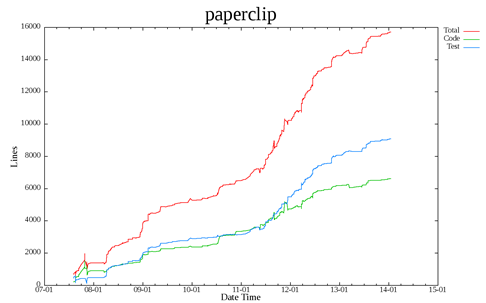


図 1　コードとテストコード相違グラフ

以上のグラフからは以下の三つの傾向がみられる．

1. テストのほうが多いものがある．
2. テストと本体が一緒に成長しているものがある．
3. 人気があるものでも、テストが殆ど無いものがある．

参考文献

[1] Mint（経営情報研究会). 図解でわかる ソフトウェア開発のすべて. 日本実業出版社, 2000, 327p.

[2] 佐藤聖規, 和田貴久, 河村雅人, 米沢弘樹, 山岸啓, 川口耕介. Jenkins実践入門. 技術評論社, 2011, 336p.

[3] The GitHub Blog. The Octoverse in 2012. 2012-12-19. https://github.com/blog/1359-the-octoverse-in-2012