

# Travis Torrent を分析することで得られた 「継続的インテグレーション(CI)ツール」の利用状況

南雲 宙真<sup>†</sup>本田 澄<sup>†</sup>鷺崎 弘宜<sup>†</sup>深澤 良彰<sup>†</sup>早稲田大学 基幹理工学部 情報理工学科<sup>†</sup>

## 1. はじめに

「継続的インテグレーション」(CI)とは開発者の成果をソフトウェアに継続的に統合することでソフトウェアを常に動作可能な最新の状態に保つ習慣を示す。この習慣はアジャイルソフトウェア開発などにおいて強く推奨されており、実際に大規模なソフトウェア開発に採用されるなど注目を集めている[1]。統合を行う際にはソフトウェアの動作を保証するためにビルドやテストをチェックする必要がある。しかし、統合の度にビルドやテストのチェックを手動で行うのは非常にコストがかかる。そこで、これらのビルドやテストを自動で行う「CI ツール」が用いられる。主な CI ツールには Travis CI や Circle CI、Jenkins などが挙げられる。

## 2. 問題定義

### 2.1. 先行研究

先行研究で行われたアンケートにより開発者が CI ツールを利用する主な目的は「ビルドの失敗防止」(87.71%)や「バグの早期発見」(79.61%)であることが分かった。このアンケートは GitHub を用いてオープンソースソフトウェア(OSS)に携わっている開発者 442 人に対して行われた。多くの先行研究が OSS を対象に CI ツールを利用しているプロジェクトと CI ツールを利用していないプロジェクトを比較し、利用しているプロジェクトのビルド・テストの成功率や問題解決の速度がより高いことを明らかにした[1, 2, 3]。ただし、これらの先行研究の大部分は CI ツールを利用しているプロジェクトと CI ツールを利用していないプロジェクトの2分類で比較を行い、CI ツールを利用しているプロジェクトの優位性を示すことで開発者に CI ツール利用を推奨するのみである。

## 2.2 先行研究の課題

これらの先行研究は CI ツールを利用していない開発者に CI ツールを利用するモチベーションを与えることには貢献しているが、既に CI ツールを利用している開発者に対しては有益なデータを示せていない。例を挙げると、開発者が CI ツールを利用している自身のプロジェクトのビルド成功率を評価する際に CI ツールを利用している他のプロジェクトのデータが存在しなければ適切な評価をすることができない。ビルド成功率が極端に高い場合には危険な統合を見逃している可能性があり、逆にビルド成功率が極端に低い場合には厳しすぎる規約が開発を妨げている可能性や開発者のスキルが想定を下回っている可能性がある。CI ツールを利用している開発者にとって有益なデータとは評価基準に成り得るデータなどであり、これらを得るためには CI ツールを利用しているプロジェクトに多角的な分析を行うべきである。

## 3. 実験概要

### 3.1. 実験目的

本稿において我々は CI ツールの利用者にとって有益なデータを得るための最初のステップとしてビルド成功率と問題解決にかかる日数を分析し、それらの評価基準となるデータを得ることができるか実験した。この2つのパラメータは前項で示すように CI ツールを利用する開発者にとって最も関心が高いパラメータである。

### 3.2 実験内容

データセットには先行研究にならい GitHub 上で公開されている OSS の開発データと Travis Torrent で公開されているビルドログを用いた。Travis Torrent は Travis CI のビルドログを収集したデータベースである。Travis CI は GitHub において最も利用率が高い CI ツールである[1]。今回の分析ではこの Travis Torrent から最新の100万ビルド分のログデータをサンプリングして利用した。それぞれのビルドログにはタイムスタンプとビルドされたプロジェクト名、そのビルドの成否が記録されている。サンプリングされた Travis Torrent のビルドログに記録されてい

Usage status of "Continuous integration (CI) tool" obtained by analyzing Travis Torrent

<sup>†</sup>Hiromasa NAGUMO, <sup>†</sup>Kiyoshi HONDA,

<sup>†</sup>Hironori WASHIZAKI, <sup>†</sup>Yoshiaki FUKAZAWA,

<sup>†</sup>School of Fundamental Science and Engineering  
Waseda University

るプロジェクト名から GitHub 上で Travis CI と連携している 740 のプロジェクトを特定した。これらのプロジェクトの中から今回の分析に用いるパラメータを欠損なく収集することができた 614 のプロジェクトを今回の分析対象とした。

本稿ではそのプロジェクトの”ビルド成功率”を Travis Torrent からサンプリングしたビルドログの中でそのプロジェクトがビルドされた最新 20 件のビルドの成功率と定義した。次に、そのプロジェクトの”問題解決にかかる日数”を GitHub で公開されているそのプロジェクトの Close された Issue の最新 20 件をサンプリングし、それらの作成から Close までにかかった日数の平均と定義した。今回の実験では作成と Close が同日に行われた Issue は継続的な統合がなされなかったと判断し集計対象外とした。本稿ではこのように定義したパラメータを用いて分析対象である 614 プロジェクトを分析することで以下の 2 つの RQ に答える。

**RQ1. CI ツールを利用しているプロジェクトのビルド成功率を分析することで、プロジェクトのビルド成功率の評価基準となるデータを得ることができるか?**

**RQ2. CI ツールを利用しているプロジェクトが問題解決にかかる日数を分析することで、プロジェクトの問題解決ペースの評価基準となるデータを得ることができるか?**

### 3. 3. 実験結果と考察

#### 3. 4. 1. RQ1 の結果と考察

図 1 は今回の分析で得られたビルド成功率のヒストグラムである。この結果、CI ツールを利用しているプロジェクトのビルド成功率の平均は約 34% であることが判明した。また、多くのプロジェクトはビルド成功率が 15% から 55% の範囲に属している。ビルド成功率がこの程度であるプロジェクトは標準的であると判断できる。ビルド成功率が 10% 以下のプロジェクトは全体の約 1 割であり、ビルド保護を最優先にしているプロジェクトでないならば規約が厳し過ぎるか開発者のスキルが未熟である可能性がある。このように評価基準の参考となる結果が得られた。

#### 3. 4. 2. RQ2 の結果と考察

図 2 は今回の分析で得られた問題解決にかかる日数のヒストグラムである。この結果、CI ツールを利用しているプロジェクトの問題は平均して約 42 日間で解決されていることが判明した。問題解決にかかる日数が 20 日以内のプロジェクトは全体の約 3 割である。この程度のペースで問

題を解決できているプロジェクトは問題に対して素早い対応ができてしていると判断できる。一方、問題解決にかかる日数が 60 日以上プロジェクトは全体の約 2 割であり、この程度のペースで問題を解決しているプロジェクトは高速な問題解決を目指すならば開発を見直すことで改善できる可能性が高いと判断できる。このように評価基準の参考となる結果が得られた。

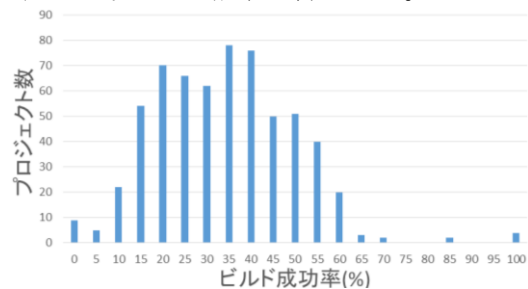


図 1. ビルド成功率のヒストグラム

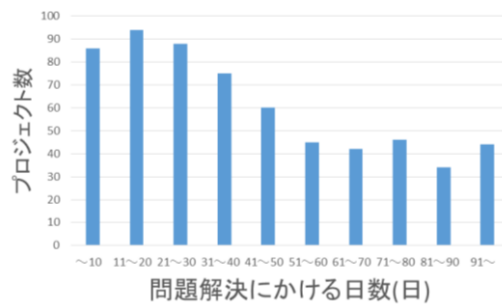


図 2. 問題解決にかかる日数のヒストグラム

### 4. まとめ

本稿で我々は CI ツールを利用しているプロジェクトに多角的な分析を行うことを提案した。更に、その最初のステップとしてビルド成功率と問題解決にかかる日数を分析し開発者が自身のプロジェクトを評価する基準を得ることができた。しかし、今回の結果は CI ツール利用者へ評価基準の一例を示したのみであり、利用者がその評価を維持・改善するために行うべきアクションなどを示すことができていない。今後は更に詳細な分析を続ける必要がある。

### 5. 参考文献

[1] M. Hilton, et al. “Usage, Costs, and Benefits of Continuous Integration in Open Source Projects,” Automated Software Engineering, 2016 pp.423-437  
 [2] M. Meyer, et al. “Continuous Integration and Its Tool,” IEEE Software (Volume: 31, Issue: 3, May-June 2014), pp. 14-16  
 [3] J. Holck, et al. “CONTINUOUS INTEGRATION AND QUALITY ASSURANCE: A CASE STUDY OF TWO OPEN SOURCE PROJECTS,” Australasian Journal of Information Systems Special Issue 2003/2004, pp.41-53