

未来に向かって

ビッグデータ時代の ソフトウェア・アナリティクス



亀井靖高 (九州大学) 島垣潤二 (ソニーモバイルコミュニケーションズ(株)) 野中 誠 (東洋大学)

意思決定のための ソフトウェアデータの解析

ソフトウェア開発では、多くの意思決定が求められる。限られた資源とスケジュールの中でソフトウェアが予定通りに構築できるように、たとえば、品質・コスト・進捗等の管理に基づいて、出荷・購入・導入等の意思決定を適切に行うことが求められる。

ソフトウェアが社会のすべてを定義する SDx (Software-Defined Everything または Software-Defined Anything) 時代に向かっている今日、ソフトウェア開発データが大規模なスケールで利用可能である。そのため、蓄積されたデータを利活用し意思決定を実施するための手助けとして、ソフトウェア・アナリティクスが注目を集めている。ソフトウェア・アナリティクスは、IEEE Software の 2013 年 7 月・8 月の特集号によると「*Software analytics is analytics on software data for managers and software engineers with the aim of empowering software development individuals and teams to gain and share insight from their data to make better decisions.*」と定義されており¹⁾、マネージャや開発者にとってより良い意思決定のために、知見の発掘・共有を促すデータ分析ということが言える。

⇒ Goto still considered harmful? 大規模データによる実証的評価

Dijkstra 博士の“*Letters to the editor : go to statement considered harmful*”が 1968 年 3 月の the Communications of the ACM にて発表された。構造化プログラミングの普及に伴い、goto 文の不適切な使用によるコード可読性への影響が少なく

なっている。しかしながら、goto 文の使用が控えられていることは経験則としては知られているものの、実際に harmful なのか? といった実証的な分析結果はほとんどなかった。

Nagappan ら²⁾ は、GitHub (ソフトウェア開発プロジェクトのオンライン共有ストレージ) のソースコードリポジトリを解析し、C 言語で記述された約 2,100,000 のファイルのうち 11% 程度に goto 文が含まれていたと報告している。また、ランダムにサンプルしたファイルを対象に、目視により goto 文の使用のされ方を調べた。その結果、全体の約 80% はエラーハンドリングで用いられており、適切に利用されていたと報告している。さらにリリース後のバグ修正のコミットを調べた結果、goto 文が修正されたり削除されたりすることが稀なため、適切な goto 文の使用は harmful ではないと結論づけている。

Nagappan らの知見は、たとえば、(1) プログラム教育において goto 文の実用的な用途を教えるか否か、(2) プログラム言語の設計において goto 文が適切に利用されていた用途 (例: エラーハンドリングや終了処理) を扱えるようにするか否かの判断材料に用いられる可能性がある。

実プロジェクトにおける ソフトウェア・アナリティクス

本章では、企業におけるソフトウェア・アナリティクスの事例を 2 つ紹介する。

⇒ コードレビュー品質

コードレビューの効果を調べるためのソフトウェ

名称	概要	URL
GHTorrent	GitHub API を経由して得られた GitHub のミラーデータセット (GitHub API の利用回数制限を気にせず利用できる)。	http://ghtorrent.org/
TravisTorrent	GitHub 内にて実行された Travis CI のビルド結果が蓄積されたデータセット。	https://travistorrent.testroots.org/
Tera Promise	過去のソフトウェア工学研究で用いられた再利用可能なデータセット群 (工数見積りやソースコード品質)。	http://openscience.us/repo/
MSR Data showcase	実務者・研究者によって査読され、一定の評価が得られたデータセットが公開されている。	http://2016.msrconf.org/#/data
StackExchange	Stack Overflow (プログラミング技術に特化した Q&A サイト) 等における開発者の Q&A に関するデータセットが公開されている。	https://archive.org/details/stackexchange

表-1 ソフトウェア・アナリティクスに利用可能な公開データセット

ア・アナリティクスが、ソニーモバイルコミュニケーションズ (株) (以降、ソニーモバイル) の開発データを用いて行われた³⁾。ソニーモバイルでは、スマートフォンで動作する Android をベースにした組み込み系システムの開発が行われている。その開発では、Web ブラウザ上でコードの校閲を可能にする Gerrit (オンラインのコードレビューシステム) を用いてコードレビューが実施され、原則として開発者の承認が得られたコミットのみが版管理システムに組み込まれる。

当該論文では、ソニーモバイル内の Gerrit に蓄積された約 20,000 件のコードレビューを、リリース後の品質 (発見された欠陥の数) と対応づけて分析している。分析の結果、たとえば、レビューがされたか否かはリリース後の欠陥に大きな影響を与えてはいない一方で、コードレビューをせずに外部 (Android) コードを内部コードに取り込んだ場合、品質低下に繋がっていることが分かった。

分析結果に対して開発者にインタビューを行った結果、コードレビューを重点的に行ういくつかの改善案が導かれた。データに基づいた分析結果を使い開発者とうまくコミュニケーションを行うことで、開発プロセスの改善につながる知見が得られた事例である。

➔ 信頼性とコストの影響要因

「開発の早い段階から品質をコントロールすればソフトウェアの信頼性を高めることができる」や「信頼性要求の高いソフトウェアの開発には相応の品質保証工数を必要とする」などの「こうあってほしい」と思うことを、特定の一組織だけでなく、複数企業

の実プロジェクトデータを用いて実証することは意外に難しい。このようないわば「当たり前」のことをきちんとデータで示そうと、(独) 情報処理推進機構技術本部ソフトウェア高信頼化センター (IPA/SEC) では国内約 10 社の実プロジェクトデータを用いて実証した⁴⁾。

調査の結果、ソフトウェア開発の上流工程におけるレビューでの不具合検出比率が高いと信頼性が高い傾向にあることや、ソフトウェア規模あたりのテスト不具合検出数が少ないと信頼性が高い傾向にあることが示された。当たり前のことに思えるかもしれないが、データが示す事実の力には一定のインパクトがあり、こうした知見の積み重ねが改善を進める原動力になる。

ただし、文献 4) は各企業に閉じた分析結果の要約であり、オープンなデータにはなっていない。

オープンなデータセット

ソフトウェア・アナリティクスにとって分析技術と同様に重要となるのが、分析に用いるデータの準備である。たとえば、現行プロジェクトを他プロジェクトと比較するためには、これまでのプロジェクトに関する開発データの蓄積が必要である。

同じデータを収集する手間の削減や価値ある知見の発掘に対する期待等からさまざまなデータが公開されている (表-1 に一部を掲載)。たとえば、GHTorrent を用いることで、現行プロジェクトに導入しようと検討しているオープンソースの候補の中から、コミット頻度が高くバグ修正が頻繁であるプロ



図-1 IROP キーボード

ジェクトはどれかといった比較ができる。GHTorrentにTravisTorrentとStack Overflowのデータセットを加えることで、ソースコードの自動進化(自動バグ修正)などの実現の可能性も秘めている。

IROP キーボードに学ぶソフトウェア・アナリティクスの落とし穴

オープンなデータセットが整備されつつある。RやWekaといった統計解析のツールも近年、充実している。そのため、ソフトウェア・アナリティクスを始めることの障壁は小さくなっている。

しかしながら、障壁が小さくなっているからこそ、注意も必要である。Zellerらは、ソフトウェア・アナリティクスを実施する際に犯してしまいがちな典型的な失敗をあえて追従することで、これまで／これからのソフトウェア・アナリティクス研究に対して警鐘を鳴らしている⁵⁾。Eclipseプロジェクトに含まれるソースコードの文字を解析し、どの文字が含まれるソースコードに欠陥の混入が多いかを調査している。その結果、I, R, O, Pが含まれるソースコードは、そうでないものと比べてバグの混入率が高いことを示した上で、**図-1**に示すようなIROPキーボードを提案している。

著者らの警鐘が6章にまとめられている。IROPキーボードに関するものとしては、(1) 相関関係(Correlation)は必ずしも因果関係(Causation)ではないという点、(2) アクショナブルな結果を求める必要(キーボードからIROPキーを除くことに意味があるのか?)があるという点が挙げられる。ソフトウェア・アナリティクスの初学者だけでなく、実務者・研究者も文献⁵⁾は一読する価値がある。

未来に向かって

本稿では、SDx到来を見据えた「ビッグデータ時代のソフトウェア・アナリティクス」として、企業で蓄積された開発データを用いたソフトウェア・アナリティクスの事例を2つ述べた。今後、より細粒度のデータ(たとえば、プライバシーを配慮した上でのバイオメトリクス)が蓄積・公開されれば、属人性を包含したアナリティクスが大きく飛躍するかもしれない。

ソフトウェア開発企業では、蓄積されているにもかかわらず、予算的・時間的制約からいまだ分析されていないデータが存在する。その一方で、分析技術はあるもののデータが蓄積されておらず評価できていない技術も存在する。今後、産学連携がさらに強まり、より多種多様なデータの蓄積と技術の深化が繰り返され、ソフトウェア・アナリティクスのより良い循環が加速することを願う。

参考文献

- 1) Menzies, T. and Zimmermann, T. : Software Analytics : So What?, IEEE Software, Vol.30, Issue.4, pp.31-37 (2013).
- 2) Nagappan, M., Robbes, R., Kamei, Y., Tanter, É., McIntosh, S., Mockus, A. and Hassan, A. E. : An Empirical Study of goto in C Code from GitHub Repositories, FSE, pp.404-414 (2015).
- 3) Shimagaki, J., Kamei, Y., McIntosh, S., Hassan, A. E. and Ubayashi, N. : A Study of the Quality-Impacting Practices of Modern Code Review at Sony Mobile, ICSE, Software Engineering in Practice Track, pp.212-221 (2016).
- 4) (独) 情報処理推進機構技術本部ソフトウェア高信頼化センター(IPA/SEC) : 横断的アプローチによるソフトウェア開発データの分析～高信頼性定量化部会 信頼性メトリクスWG検討報告書～(2015), http://www.ipa.go.jp/sec/reports/20150416_1.html
- 5) Zeller, A., Zimmermann, T. and Bird, C. : Failure is a Four-Letter Word, PROMISE, pp.5:1-5:7 (2011).

(2017年4月28日受付)

亀井靖高 (正会員) kamei@ait.kyushu-u.ac.jp
 2009年奈良先端科学技術大学院大学情報科学研究科博士後期課程修了。日本学術振興会特別研究員(PD)、カナダQueen's大学博士研究員、九州大学大学院システム情報科学研究院助教を経て、2015年より同大学院准教授。博士(工学)。

島垣潤二 Junji.Shimagaki@sonymobile.com
 2007年北海道大学工学部学士課程、2009年スイス連邦工科大学(ETH)物理学部修士課程修了。同年よりソニーモバイル(株)でソフトウェア開発者、ソフトウェア工学リサーチャーとして活動。

野中 誠 (正会員) nonaka-m@toyo.jp
 1995年早稲田大学理工学部工業経営学科卒業。同大学院博士後期課程単位取得退学。2003年より東洋大学経営学部専任講師。2014年より同大教授。IPA/SEC 高信頼性定量化部会 主査、日本科学技術連盟ソフトウェア品質委員会(SQiP)運営委員長など、産学の交わる場で活動。