

ソースコード上の技術的負債除去を活性化させるゲーミフィケーション環境の開発

Development of Gamification Environment for Self-Admitted Technical Debt Removal

一ノ瀬智浩 † 畑秀明 † 松本健一 †
Tomohiro ICHINOSE Hideaki HATA Kenichi MATSUMOTO

1. はじめに

技術的負債とは、ソフトウェア開発における場当たりの対応やその結果を指す比喩である。技術的負債は、短期的には開発スピードを速める利点がある一方、長期的に見るとソースコードの保守性を低下させ、開発スピードを遅くするという問題がある [1]。開発者が意識的にソースコード上に残す技術的負債は self-admitted technical debt (以降, SATD) と呼ばれる。SATD は TODO, FIXME といったキーワードを含むコメントでその存在が確認できる。Wehaibi らの SATD とソフトウェアの品質に関する研究では、SATD が含まれているファイルに対する変更 (SATD change) と含まれていないファイルに対する変更 (non-SATD change) の 2 種類を比較している。複雑度の比較では、non-SATD change よりも SATD change の方が変更された行数やファイル構造数が多く複雑であるという結果が得られており、SATD はソフトウェアの保守性を低下させているといえる [2]。ソフトウェアの保守性を向上させるには、ソースコード上に残り続けている SATD を除去することが必要だと考えられる。しかし、ソフトウェアの開発期間中は継続的に SATD がソースコード上に混入し、26.25%から 63.45%は取り除かれるが、全体として SATD はソフトウェアに残り続ける傾向があると報告されている [3]。ソースコード上に存在する SATD は、複雑すぎる関数や応急処置的な実装といった、ソフトウェアの動作そのものには影響しないものが大半を占めている [4]。よって、多くの SATD は除去の優先度が低いためにソースコード上に残り続けていると考えられる。そのため、開発者に優先度の低い SATD を除去を促す仕組みがあれば、ソフトウェアの保守性の向上に繋がると考えられる。SATD を除去するためには SATD が存在するソースファイルを知る必要がある。しかし、SATD ソースコード上のコメントを見なければ存在が分からないため、どのファイルに SATD が存在するかを把握しづらいという問題がある。効率良く SATD を除去するには、SATD の存在するファイルが分かりやすく可視化されることが望ましい。

本研究では、ソースコード上に残っている SATD の除去を支援するための、ゲーミフィケーション、およびソースコード可視化を利用したシステムを提案する。提案システムでは、ランキング提示による競争を促すゲーミフィケーションを用いることで、積極的な SATD の除去を開発者に促し、ソフトウェアの保守性の維持向上を目指す。また、提案システムはソースコードのファイル構造を街のように可視化し、SATD が存在するファイルを目立たせることで、除去すべき SATD を分かりやすくユーザに提示する。

2. 関連研究

Potdar らはソフトウェアに存在する SATD の個数を明らかにするため、SATD を示す 62 個のコメントパターンを定義し、4

つの OSS プロジェクトに存在する SATD の個数を調査している。その調査では、SATD がソースファイル上に 2.4%から 31.0%含まれているという結果が得られている [3]。da S. Maldonado らは SATD の種類ごとの個数を明らかにするため、SATD を関連する 5 種類の問題点 (設計・欠陥・文書化・要求物・テスト) ごとに分類している。5 つの OSS プロジェクトのソースコードの調査では、最も割合が高いのは設計に関する SATD であり、42%から 84%を占めているという結果が得られている [4]。設計に関する SATD は複雑すぎる関数や応急処置的な実装といったものであり、ソフトウェアの動作に影響する欠陥や要求物に関する SATD に比べると除去の優先度が低いと考えられる。SATD の除去に対し、何らかのインセンティブを設定することにより、開発者に優先度の低い SATD を除去させられる可能性がある。本研究ではゲーミフィケーションを利用することにより、ソースコード上に残っている、除去優先度の低い SATD の除去を促すシステムを提案する。

ソフトウェアの可視化は、プログラム読解支援や問題検出の結果提示など、様々な目的に対して数多く研究されている [5][6]。Wettel らの提案する CodeCity は、ソースコードのクラスとパッケージの構造を街のように 3D で可視化するシステムである。街の構造は直感的で親しみやすく、ソフトウェアの複雑な構造を単純化しすぎることなく表現するのに適している [7]。Balogh らはコンピュータゲームである Minecraft* を用いてソースコードを街のように可視化する CodeMetropolis を利用し、ソフトウェアテストに関連するメトリクスを可視化するシステムを提案している [8]。Balogh らのシステムでは関連のあるテストケースとソースコードを並べて配置して可視化することで、開発者の理解を支援している。本研究では街の構造を利用した可視化を用い、SATD のあるソースコードに目印を付けることによって、開発者による、SATD があるソースコードの特定作業を支援する。

近年では、シリアスゲームや Games with a purpose といったゲームを問題解決に利用する手法への関心が高まっている [9][10]。その中でもゲーミフィケーションはゲームそのものではなく、ゲームの要素を社会活動や、活動を支援するシステムに取り入れる事で意欲を維持向上させる手法である [11]。ゲーミフィケーションによって作業を支援するシステムはこれまでに複数提案されている [12][13][14]。また、ソフトウェア開発への適用も注目されている [15][16]。既存研究で提案されたシステムでは、ゲームの要素として競争がよく用いられている。競争はカイヨワの提案する楽しさを生む 4 つの要因 [17] や、Steven が定義した人を動機づける 16 個の欲求 [18] などにおいても見られるため、ゲーミフィケーションに用いる要素として適しているといえる。本研究では、システム内で SATD の除去数をランキング形式で提示し、開発者同士で競わせて動機づけることで SATD の積極的な除去を支援する。

† 奈良先端科学技術大学院大学, Nara Institute of Science and Technology

*Minecraft, <http://minecraft.net/>

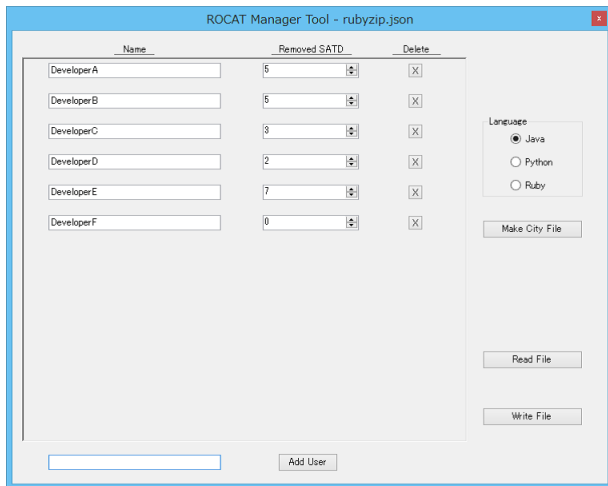


図 1: 管理者用ツール

3. 提案システム

提案システムは Git* で管理されているリポジトリを対象とする。Python プログラムでリポジトリを解析し、街の情報を生成する。各開発者の SATD の除去数情報はプロジェクト管理者用ツール上で編集し、街の情報に付与する。街はゲームエンジンの一つである Unity† 上で可視化する。

3.1 リポジトリ解析

リポジトリ解析は Python ライブラリの一つである GitPython‡ を利用した、約 180 行の Python プログラムで実施する。解析プログラムは Git リポジトリに存在するソースファイルを探索し、街の情報としてソースファイル名や存在するディレクトリ名、コード行数、コメント行数、SATD が存在する行番号を取得する。対応しているプログラミング言語は Java, Ruby, Python の 3 種類である。STAD は Potdar らの定義する 62 個のキーワードから判断する [3]。解析結果は Json ファイルとして出力される。

3.2 管理者用ツール

開発者によるゲームの不正行為を防ぐため、街の情報や SATD の除去数の生成・更新は、街やランキングを提示するシステムとは別の、プロジェクト管理者用のツール上で実施する。管理者用ツールは C# で実装された約 650 行の GUI プログラムである。管理者用ツールの画面を図 1 に示す。

“Make City File” ボタンをクリックするとディレクトリ選択画面が表示される。Git リポジトリを選択すると、ラジオボタンで選択しているプログラミング言語を対象にリポジトリを解析する。“Read File” ボタンをクリックするとファイル選択画面が表示される。街の情報を記録した Json ファイルを選択すると、Json ファイル上に記録されている開発者名と、各開発者の SATD の除去数の一覧が表示される。開発者は除去数は “Add User” ボタンや、“×” ボタンで追加・削除することができる。SATD の除去数は数値入力コントロールで編集することができる。“Write File” ボタンをクリックすると、クリックした時点で管理者用ツールに表示されている開発者と SATD の除去数の情報が、街の情報を記録した Json ファイルに追加・上書きされる。

3.3 街とランキングの提示

街の可視化及びランキングの提示は Unity で実装された約 1000 行の GUI プログラムで実施し、管理者ツールで生成・更新された Json ファイルを Unity で読み込み、可視化する。可視化した街の画面を図 2 に示す。使用したリポジトリは (a) が rubyzip§, (b) が javaparser¶ である。各ディレクトリは白い土台で、各ソースファイルは水色のビルで可視化される。ビルの高さはコード行数に、横幅はコメント数に対応している。マウスポインタをビルに合わせるとファイル名が、土台に合わせるとディレクトリ名が表示される。ファイル名とディレクトリ名が表示される様子を図 3 および図 4 に示す。図 2 に示すように、SATD が存在するファイルのビルの上には模様が付いた球体が目印として表示される。ビルはソースファイルが存在するディレクトリの土台上に作られるため、開発者はどのソースファイルがどのディレクトリに存在するか、また、どのソースファイルに SATD が存在しているかを素早く確認することができる。

図 5 にランキング部分を拡大したものを示す。図 5 のように、街と共に各開発者の SATD の除去数のランキングが表示される。開発者は自分の順位や、自分や他の開発者が何個の SATD を除去したかを知ることができる。除去数をランキング化することで、開発者の競争心が刺激され、より多くの SATD が除去されると期待される。

4. まとめと展望

本研究では、ソースコードの可視化およびゲーミフィケーションにより、SATD の除去を支援するシステムを提案した。提案システムを開発現場に適用することにより、開発者の SATD の除去作業に対する意欲が向上してより多くの SATD が除去されることが期待でき、ソフトウェアの保守性の向上につながると考えられる。ゲーミフィケーションは金銭的なインセンティブを利用することなく作業者を動機づけることができる手法である。そのため、提案システムはボランティアで開発が行われる OSS プロジェクトに特に適していると考えられる。

現在、提案システムを Web ブラウザ上で利用できるようにするための実装を進めている。今後、提案システムを実際に被験者に利用してもらい、より SATD を除去しようと思ったか、また、提案システムを導入することでより多くの SATD が除去されたかを評価する予定である。

参考文献

- [1] Nico Zazworka, Michele A. Shaw, Forrest Shull, and Carolyn Seaman, “Investigating the Impact of Design Debt on Software Quality,” Proceedings of the 2nd Workshop on Managing Technical Debt, pp.17–23, 2011.
- [2] Sultan Wehaibi, Emad Shihab, Latifa Guerrouj, “Examining the Impact of Self-admitted Technical Debt on Software Quality,” IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering, pp.179-188, 2016.
- [3] Aniket Potdar and Emad Shihab, “An Exploratory Study on Self-Admitted Technical Debt,” 30th IEEE International Conference on Software Maintenance and Evolution, pp.91-100, 2014.

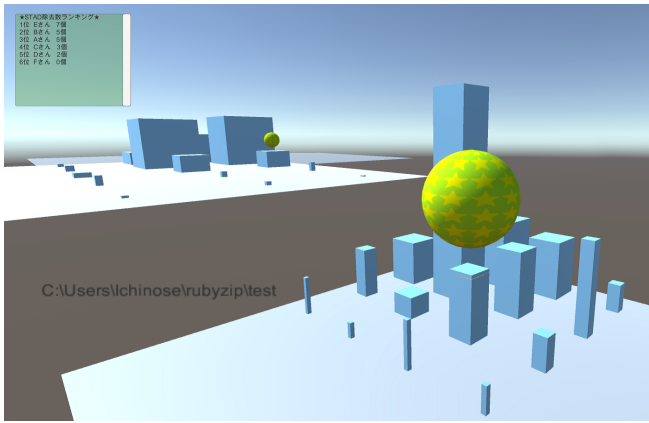
*Git, <https://git-scm.com/>

†Unity, <http://unity3d.com/>

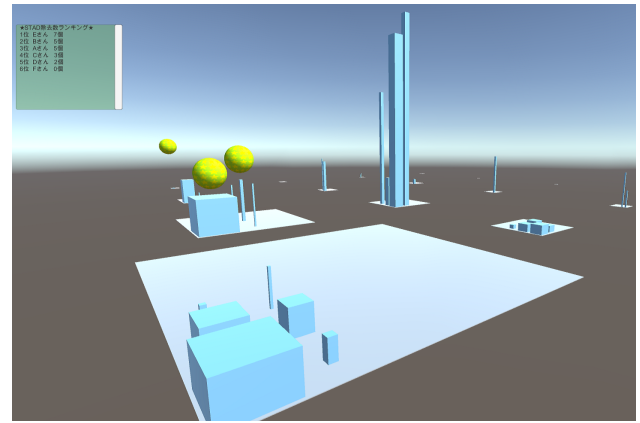
‡GitPython, <http://gitpython.readthedocs.io/en/stable/>

§rubyzip, <https://github.com/rubyzip/rubyzip>

¶javaparser, <https://github.com/javaparser/javaparser>



(a) rubyzip



(b) javaparser

図 2: 街とランキング

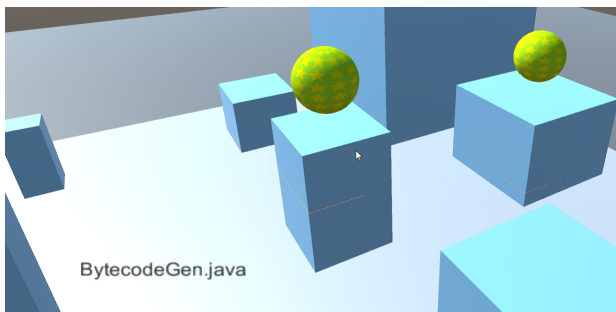


図 3: ファイル名表示

★STAD除去数ランキング★		
1位	Eさん	7個
2位	Bさん	5個
3位	Aさん	5個
4位	Cさん	3個
5位	Dさん	2個
6位	Fさん	0個

図 5: ランキング

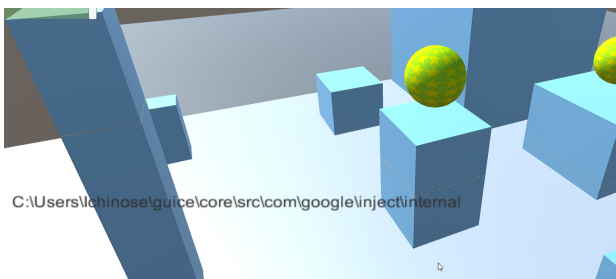


図 4: ディレクトリ名表示

[4] Everton da S. Maldonado and Emad Shihab, “Detecting and Quantifying Different Types of Self-Admitted Technical Debt,” IEEE 7th International Workshop on Managing Technical Debt, pp.9-15, 2015.

[5] 喜多義弘, 片山徹郎, 富田重幸, “Java プログラム読解支援のためのプログラム自動可視化ツール Avis の実装と評価,” 電子情報通信学会論文誌, Vol.95, No.4, pp.855-869, 2012.

[6] 肥後芳樹, リビエリシモネ, 松下誠, 井上克郎, “大規模ソースコードを対象としたコードクローンの検出と可視化,” 情報処理学会論文誌, Vol.48, No.11, pp.3510-3519, 2007.

[7] Wettel Richard and Lanza Michele, “CodeCity: 3D Visualization of Large-scale Software,” Companion of the 30th International Conference on Software Engineering, pp.921-922, 2008.

[8] Gergő Balogh, Tamás Gergely, Árpád Beszédés and Tibor Gyimóthy, “Using the City Metaphor for Visualizing Test-

Related Metrics,” IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering, pp.17-20, 2016.

[9] 白鳥和人, 星野准一, “シリアスゲーム,” 人工知能学会誌, Vol. 23, No. 1, pp. 79-84, 2008.

[10] L. von Ahn, “Games with a purpose,” Computer , vol.39, no.6, pp.92-94, 2006

[11] 神馬豪, 石田宏実, 木下裕司, “顧客を生み出すビジネス新戦略ゲーミフィケーション,” 大和出版, 2012.

[12] 倉本到, 柏木一将, 植村友美, 渋谷雄, 辻野嘉宏, “Weekend Battle : エンタテインメント性の作業環境への提供により作業意欲を維持向上させるシステム,” ヒューマンインタフェース学会論文誌, Vol. 8, No. 3, pp. 331-342, 2006.

[13] M. S. El-Nasr, L. Andres, T. Lavender, N. Funk, N. Jahangiri, and M. Sun, “Igniteplay: Encouraging and Sustaining Healthy Living through Social Games,” In Proc. International Games Innovation Conference 2011, pp. 23-25, 2011.

[14] 狩野翔, 福島拓, 吉野孝, “用例評価のモチベーション維持支援システム「用例の森」の開発と評価,” 情報処理学会論文誌, Vol. 53, No. 1, pp. 138-148, 2012.

[15] Daniel J. Dubois and Giordano Tamburrelli, “Understanding Gamification Mechanisms for Software Development,” ESEC/FSE 2013 Proceedings of the 2013 9th Joint Meeting on Foundations of Software Engineering, pp.659-662, 2013.

- [16] L. Singer and K. Schneider, "It Was a Bit of a Race: Gamification of Version Control," In Proc. 2nd International Workshop on Games and Software Engineering, pp.5-8, 2012.
- [17] ロジエ・カイヨワ, "遊びと人間," 講談社, 1990.
- [18] S. Reiss, "Who Am I?: The 16 Basic Desires That Motivate Our Behavior and Define Our Personality," Berkley Trade, 2002.