IPA

2019 年度 未踏 IT 人材発掘・育成事業 採択案件評価書

1. 担当 PM

竹迫 良範（株式会社リクルートテクノロジーズ 執行役員）

2. クリエータ氏名

上田 裕己（奈良先端科学技術大学院大学 ソフトウェア工学研究室）

3. 委託金支払額

2,035,600 円

4. テーマ名

開発者が行ったソースコード修正作業を学習し代行するボット

5. 関連 Web サイト

- DevReplay 配布サイト：https://devreplay.github.io/
- DevReplay プロジェクトのソースコード：https://github.com/devreplay
- DevReplay 本体：https://github.com/devreplay/devreplay
  - コード改善パターン生成スクリプト群：
    https://github.com/devreplay/devreplay-pattern-generator
  - ソースコードのトークン化および差分作成スクリプト：
    https://github.com/Ikuyadeu/CodeTokenizer
  - DevReplay Visual Studio Code プラグインおよび Language Server：
    https://github.com/devreplay/vscode-devreplay
  - DevReplay Atom エディタプラグイン：
    https://github.com/devreplay/atom-devreplay
  - DevReplay GitHub アプリケーション：
    https://github.com/devreplay/github-app-devreplay
  - DevReplay Web ページ：
    https://github.com/devreplay/devreplay.github.io

6. テーマ概要

本プロジェクトでは、開発者のソフトウェア修正方法を「学習」し「再現」す

るツール“DevReplay”を開発した。最初に、開発者の開発履歴を「学習」するスクリプト群を開発した。次に「再現」方法をプログラミングエディタ上で提案するアプリケーション、および GitHub で修正方法を提案するボットを開発した。ソフトウェア開発の現場において、デバッギング作業はソフトウェアの信頼性確保に重要であると同時にコストの高い作業でもある。修正作業も含めたデバッギングは開発作業全体で約 50%の時間的コストを開発者が消費しているとの報告もあり、本ツール群を用いることで開発者は定型的な開発作業をツールに任せ、作業コストを減らすことが可能になる。

7. 採択理由

ソフトウェア開発チームに新しいエンジニアが加わるとき、コーディング規約を徹底するために先輩によるコードレビューが非常に重要な時間を占める。本提案の開発支援システムでは、開発者が GitHub 上で過去に修正したソースコードの履歴を元に、自動的にソースコードの修正提案をしてくれる。日本の IT システムの開発現場では、2019 年 5 月の新元号対応や、Python バージョン 2 からバージョン 3 への非互換な移行、Java 8 のサポート終了に伴う EOL（End Of Life）対応などで、ソースコードを一律に修正する機会も発生する。従来のコードレビューでは、人手による作業で多くの時間と精神的コストがかかっており、この開発支援システムがあれば、人件費削減とシステムによるレビュー担保によって、開発者の QoL（Quality of Life）が向上し、結果的にソフトウェアの品質向上が期待できる。文法チェックの lint のように淡々とシステムの bot がコードの修正提案を指摘するようになれば、先輩による厳しいコードレビューで人間関係が荒れてしまうことを心配することがなくなり、開発者の精神衛生も保ちやすくなる。定型的な業務をなくしていき、エンジニアの開発者体験（DX : Developer eXperience）を向上させることは、IT システム開発現場をデジタルトランスフォーメーション（DX : Digital Transformation）につながっていく。実際の開発プロジェクトでレビューを行うことで品質を向上させ、適応対象となるプログラミング言語環境を増やしていき、プロジェクトのコーディング規約の自動作成プログラマのコーディングレベルの推定などに発展することを期待して採択した。

8. 開発目標

本プロジェクトの開発目標はソースコード修正の提案や自動修正を行う統合的な開発環境の実現である。具体的には、GitHub 上で公開されているソフトウェアの開発履歴からソースコードの改善パターンを発見し、不特定多数に共有可能なファイルとして出力する「ソースコード改善パターン発見スクリプト群」と改善パターンに基づいたソースコードの改善を自動的に行う「パターンに基づくソースコード修正、修正推薦ツール」の 2 つを開発目標とした。

このうち、パターンに基づいて自動的にソースコードを修正するツールを利用するユースケースとして以下の 3 つを想定し、それぞれ実装を分けて開発することとした。

- CI を始めとしたコマンド実行ツールによる自動修正
- コード編集中の警告に基づく修正
- オンラインでのコードレビューによる修正提案

9. 進捗概要

本プロジェクトはプログラマ向けの開発ツールであるため、本クリエータが自分でツールを使用しながら設計、実装、修正を行うドッグフーディングという手法を用いて開発を進めた。ソースコード改善パターン発見スクリプト群として、以下の 3 種類のパターンそれぞれを修正するための機能を実装した。

(1) 開発チームや個人の中で閉じた暗黙知として運用されているパターン

現在、開発グループの中で暗黙知となっているパターンを発見するために、ソースコードの変更履歴から修正方法をパターンとして取り出すスクリプトを開発した。また、スクリプトの利用では対象とする開発者、期間を指定する機能を実装した。これを用いてチーム内で無意識に行ってきた習慣だけでなく、例えば著名なプログラマの癖なども明文化することができる。

(2) 複数のソフトウェアにまたがり、文献もしくはソースコードとして伝わっているパターン

上記のスクリプトには複数のプロジェクトを指定する機能を実装している。これにより複数のプロジェクトの共通点および相違点をパターンとして明文化することができる。DevReplay が生成するパターンファイルは JSON 形式であり、ソースコードを切り貼りするだけで編集、作成が可能である。この特徴を用いて生成したパターンファイルを手作業で修正、または新たにパターンファイルを作成し、知識としてのノウハウを実践可能な形式で配布することができる。

(3) 既存のコーディング規約で定義および自動修正が実施されているパターン

開発者はプログラミング言語ごとに異なるツールを探し、インストールする必要がある。また、既存ツールの欠点として、対象となる言語およびそれに基づくコーディング規約の策定に対応するには、ツール開発者による更新を待つ必要がある点が挙げられる。

DevReplay 単体で複数プログラミング言語のサポートを行い、ルール策定とツール対応の間にある時間差を埋めるために、有名な言語の標準的なパターンを DevReplay に内蔵し、ユーザによる編集を可能にしている。このパターンに

より、ユーザは DevReplay をインストールするだけでソースコードを自動的に修正できる。

パターンに基づくソースコード修正、修正推薦ツールとして開発した DevReplay の自動実行を行う VS Code のプラグイン「vscode-devreplay」の利用画面を図 1 に示す。また、DevReplay の自動実行を行う GitHub アプリケーション「Github-app-devreplay」の利用画面を図 2 に示す。

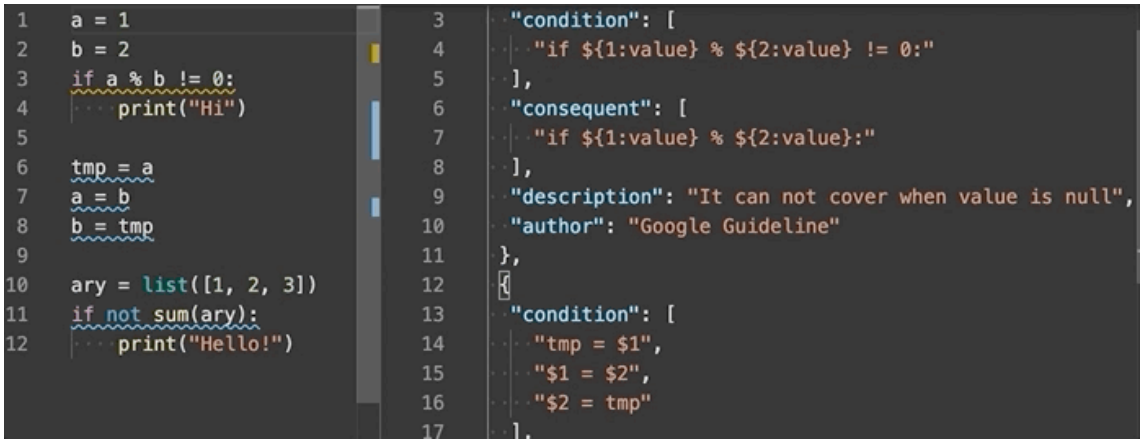

図 1：VS Code における DevReplay 利用画面

図 2：GitHub における DevReplay 利用画面

10. プロジェクト評価

機械学習の結果については各現場で正解かどうかの判定を行う形になるが、本プロジェクト期間では、ツールのプロダクト品質と機能を高める部分に注力した。最終的に C, C++, Java, Dart, JavaScript, Python, Go, TypeScript, COBOL, Ruby, PHP, R の 12 言語をサポートし、様々な開発者に使ってもらえるプロダクトに仕上がった。VSCode 拡張の他に LSP をサポートしたことにより、VSCode のみならず、Vim など様々なエディタを使っている開発者にも利用されることが期待される。

11. 今後の課題

現状の課題として、数多くのプログラミング言語、環境に対応した結果、各プログラミング言語に特化した既存ツールと比較して、特定の目的へ特化した精度の高い修正方法を提示できないことが挙げられる。また、本プロジェクトを構成するツールは 6 つあり、クリエーター人が学業と並行して今後のメンテナンスおよび機能の拡張を続けるのは困難であり、似たような開発ツールを作成している企業へメンテナンス権限の移譲や、オープンソースによる開発者の拡大が必要である。