NAIST-IS-MT1551010

修士論文

ゲーミフィケーションを用いたソースコード上の技術的負債除去

一ノ瀬 智浩

2016 年 2 月 2 日

奈良先端科学技術大学院大学
情報科学研究科

本論文は奈良先端科学技術大学院大学情報科学研究科に
修士 (工学) 授与の要件として提出した修士論文である。

一ノ瀬 智浩

審査委員：

松本 健一 教授　（主指導教員）
安本 慶一 教授　（副指導教員）
畑 秀明 助教　（副指導教員）

ゲーミフィケーションを用いたソースコード上の技術的負債除去*

一ノ瀬 智浩

内容梗概

人類がこの地上に現われて以来、π の計算には多くの関心が払われてきた。

本論文では、太陽と月を利用して π を低速に計算するための画期的なアルゴリズムを与える。

ここには内容梗概を書く。ここには内容梗概を書く。ここには内容梗概を書く。ここには内容梗概を書く。ここには内容梗概を書く。ここには内容梗概を書く。ここには内容梗概を書く。ここには内容梗概を書く。ここには内容梗概を書く。ここには内容梗概を書く。ここには内容梗概を書く。ここには内容梗概を書く。ここには内容梗概を書く。ここには内容梗概を書く。

ここには内容梗概を書く。ここには内容梗概を書く。ここには内容梗概を書く。ここには内容梗概を書く。ここには内容梗概を書く。ここには内容梗概を書く。ここには内容梗概を書く。ここには内容梗概を書く。ここには内容梗概を書く。ここには内容梗概を書く。ここには内容梗概を書く。ここには内容梗概を書く。ここには内容梗概を書く。

キーワード

π, 天文学, 数学, 計算機, アルゴリズム

*奈良先端科学技術大学院大学 情報科学研究科 修士論文, NAIST-IS-MT1551010, 2016 年 2 月 2 日.

英語タイトルが入る *

Tomohiro Ichinose

Abstract

The calculation of π has been paid much attention since human beings appeared on the earth.

This thesis presents novel low-speed algorithms to calculate π utilizing the sun and the moon.

This is a sample abstract. This is a sample abstract. This is a sample abstract. This is a sample abstract. This is a sample abstract. This is a sample abstract. This is a sample abstract. This is a sample abstract. This is a sample abstract. This is a sample abstract.

This is a sample abstract. This is a sample abstract. This is a sample abstract. This is a sample abstract. This is a sample abstract. This is a sample abstract. This is a sample abstract. This is a sample abstract. This is a sample abstract. This is a sample abstract.

Keywords:

π, astronomy, mathematics, computer, algorithm

*Master's Thesis, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-MT1551010, February 2, 2016.

目 次

図 目 次

表 目 次

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的な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 tech-nical 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 SATDchange 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，除去するべき 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．

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 システム構成

サーバ側含めたシステム全体の構成説明.

3.2 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．

3.3 Web システム

Web システムの説明.

3.3.1 Rocat

（ACIT の論文引いて）Rocat を説明.

3.3.2 ソースコードと SATD の情報提示

元の Rocat からの改良点他，HEIZO での可視化部分の説明.

3.3.3 ランキング

ランキング部分の説明.

4. 実験

本章では，提案システムの評価実験について述べる.（以下，実験概要の説明. 何のためにどんな実験をするのか）

4.1 実験対象

実験対象のプロジェクト（リポジトリ）についての説明.

4.2 実験方法

システムにリンクするバッジを作ってReadMeに〜あたりの説明とバッジを貼ってからの流れを説明.

4.3 評価

定量的評価と定性的評価の2種類をする.

4.3.1 SATD除去数の評価

定量的評価の説明.

4.3.2 使いやすさの評価

定性的評価の説明.

5. 結果と考察

本章では，提案システムの評価実験の結果，およびその考察を述べる．

5.1 SATD 除去数

定量的評価の結果と考察．

5.2 使いやすさ

定性的評価の結果と考察．

6. 妥当性の脅威

妥当性の脅威の説明. [1]

7. 終わりに

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．

謝辞

Thank you. Thank you.

参考文献

[1] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. B. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *CoRR*, abs/1408.5093, 2014.

これはおまけの図です。

図 1　おまけの図

付録

A．おまけその１

　これはおまけです。これはおまけです。これはおまけです。これはおまけです。これはおまけです。これはおまけです。これはおまけです。これはおまけです。これはおまけです。これはおまけです。これはおまけです。これはおまけです。これはおまけです。これはおまけです。これはおまけです。これはおまけです。これはおまけです。これはおまけです。

B．おまけその２

　これもおまけです。これもおまけです。これもおまけです。これもおまけです。これもおまけです。これもおまけです。これもおまけです。これもおまけです。これもおまけです。これもおまけです。これもおまけです。これもおまけです。これもおまけです。これもおまけです。これもおまけです。これもおまけです。これもおまけです。これもおまけです。