

論文内容の要旨

申請者氏名 XIAO TAO

Technical Debt (TD) is a metaphor used to describe situations where long-term software artifact quality is sacrificed for short-term goals. An explicit representation of such a phenomenon is Self-Admitted Technical Debt (SATD), which refers to debt intentionally introduced and documented as comments by developers. Although significant work has been done on SATD in source code, little is known about SATD in build systems. To address this, this thesis tackles to unveil the implied cost as known as SATD in build systems.

This thesis first characterizes SATD in build systems through a qualitative analysis of 500 SATD comments in the Maven build system of 291 projects. The findings reveal that limitations in tools and libraries and complexities of dependency management are the most frequent causes, accounting for 50% and 24% of the comments, respectively. Developers often document SATD as issues to be fixed later. Additionally, classifiers were trained to detect the two most frequently occurring reasons and four most frequently occurring purposes of SATD, achieving an F1-score of 0.71–0.79. The results also indicate that raising awareness of SATD can lead to its resolution, as evidenced by the 16 identified 'ready-to-be-addressed' SATD instances.

This thesis second investigates the prevalence and characteristics of SATD clones in build systems, examining 50,608 SATD comments from Autotools, CMake, Maven, and Ant build systems. The results show that SATD clones are more prevalent in build systems than in source code, with rates ranging from 62% to 95%. Furthermore, 76% of SATD clone occurrences have high similarity scores, and a quarter of these clones are introduced by the original author. The study identifies external factors as the most frequent locations and limitations in tools and libraries as the most frequent causes of cloned SATD comments.

Together, this thesis also builds a framework for actionable practices of SATD in build systems. This thesis provides a comprehensive understanding of SATD in build systems, highlighting the need for future research on tool support to track and manage SATD backlogs and the design of automated recommendation systems for repaying SATD effectively based on resolved clones.

論文審査結果の要旨

申請者氏名 XIAO TAO

本論文は、ソフトウェア開発のビルド作業における技術的負債(Technical Debt)の実態を明らかにするとともに、その中でも特に、「開発者自身がその存在を自覚していることを注釈文等に記載している自己申告型技術的負債(Self-Admitted Technical Debt: SATD)」を対象に、効率よく負債を償還するためのフレームワークを提案するものである。

ソフトウェア開発における「ビルド」とは、ソースコードをユーザーに提供できる形のソフトウェア製品とするための一連の作業を指し、ソースコードのコンパイル、ライブラリのリンク、パッケージングなどが含まれる。大規模なソフトウェアや複数のプラットフォームで稼働するソフトウェアの開発においては特に重要とされ、作業手順を記したスクリプトを用いた自動化システム(ビルドシステム)が普及している。一方、技術的負債とは、理想とされる、あるいは、より優れているとされるが実現に時間のかかる方法を選択する代わりに、実現は容易であるが手直しが将来必要となる方法で実装されているソフトウェアプロダクトの状態、あるいは、実装されているプロダクト部分のことである。ソースコードにおける技術的負債に関する研究は数多く行われているが、ビルドに用いられるスクリプトを対象とした研究はごく限られている。

本論文では、まず、広く用いられているビルドシステムの一つであるMavenを対象に、291のソフトウェア開発プロジェクトにおける500のSATDを詳細に分析し、ソフトウェア開発に用いられるツールやライブラリの制限や依存関係の複雑さに起因するSATDが最も多いことなどを明らかにした。また、対象を4つのビルドシステムAutotools、CMake、Maven、Antに拡張し、50,608のSATDについて、重複する記述(クローン)に注目し、より詳細な分析を行った。その結果、SATDにおけるクローンは、ソースコードにおける62%よりも格段に多い95%に達し、クローンとしての類似度も高いことなどが明らかになった。以上の結果に基づき、ソフトウェア開発のビルド作業における自己申告型技術的負債を追跡・管理するためのフレームワークを提案した。提案するフレームワークでは、償還された負債とのクローン化率の高い負債を提示するなどにより、負債償還の効率化の可能にしている。

以上の通り、本論文は、ソフトウェア開発のビルド作業における自己申告型技術的負債とそのクローン化に着目することで、個々のソフトウェアについてはもちろんのこと、ソフトウェアエコシステム群に波及する開発・保守リスクを低減・回避するための、これまでにない具体的なフレームワークを実現した。開発・保守リスクの系統的な低減・回避につながる知見は、広くソフトウェア開発とソフトウェアエコシステムの高度化、そして、ソフトウェア工学研究の発展に大きく貢献することから、博士(工学)論文として価値あるものと認める。