Integrating Technical Debt Management and Software Quality Management Processes: A Framework and Field Tests

Journal-First Extended Abstract[†]

Narayan Ramasubbu
University of Pittsburgh
Pittsburgh, PA
USA
narayanr@pitt.edu

Chris Kemerer
University of Pittsburgh
Pittsburgh, PA
USA
ckemerer@pitt.edu

ABSTRACT

Technical debt, defined as the maintenance obligations arising from shortcuts taken during the design, development, and deployment of software systems, has been shown to significantly impact the reliability and long-term evolution of software systems [1], [2]. Although academic research has moved beyond using technical debt only as a metaphor, and has begun compiling strong empirical evidence on the economic implications of technical debt, industry practitioners continue to find managing technical debt a challenging balancing act [3]. Despite the increasing awareness of the importance of managing technical debt in software product development, systematic processes for implementing technical debt management in software production have not been readily available.

To address this gap, we developed and field tested a normative process framework that systematically incorporates steps for managing technical debt in commercial software production. The framework integrates processes required for technical debt management with existing software quality management processes prescribed by the project management body of knowledge (PMBOK) [4], and organizes the different processes for technical debt management under three steps: (1) make technical debt visible, (2) perform cost-benefit analysis, and (3) control technical debt. To implement the processes, we introduce a new artifact, called the *technical debt register*, which stores, for each software asset, the outstanding principal and the associated interest estimated for the technical debt embedded in the asset. The technical debt register also stores the desired control target for each software asset's technical debt, which is populated and used during the cost-benefit analysis and control target calculations.

There are three main benefits from this integrated approach. First, it enables the uncovering of hidden technical debt embedded in systems. Established quality assurance and control practices can be utilized to effectively associate software defects with specific design and deployment decisions made by programmers. Such associations make technical debt visible to the team and thereby facilitate the quantification of debt-related principal and interest. Second, it helps to bridge the gaps that exist between the technical and economic assessments of technical debt, and aid in formulating actionable policies related to technical debt management. Finally, integrating technical debt management processes with established quality frameworks aids the wider adoption of emerging prescriptions for managing technical debt.

We partnered with three commercial software product development organizations to implement the framework in real-world software production settings. All three organizations, irrespective of their varying software process maturity levels, were able to adopt the proposed framework and integrate the prescribed technical debt management processes with their existing software quality management processes. Our longitudinal data and case-study interviews indicate that the organizations were able to accrue economic benefits from the adoption and use of the integrated framework. And, based on our field study observations, we also identified a set of best practices that support the implementation and use of our framework: facilitating engagement between business and engineering stakeholders, adoption of policies based on a probabilistic analysis framework, and limiting process overheads.

KEYWORDS

Technical debt, software quality, software maintenance, software engineering economics, cost of quality, software product development, software process, case study.

[†] The full version of the paper is forthcoming in *IEEE Transactions on Software Engineering*, https://doi.org/10.1109/TSE.2017.2774832

ICSE '18, May 27-June 3, 2018, Gothenburg, Sweden
© 2018 Copyright is held by the owner/author(s).
ACM ISBN 978-1-4503-5638-1/18/05.
https://doi.org/10.1145/3180155.3182529

REFERENCES

[1] N. Ramasubbu and C.F. Kemerer. 2014. Managing Technical Debt in Enterprise Software Packages. *IEEE Trans. Software Engineering* 40, 8, 758–772.
[2] N. Ramasubbu and C.F. Kemerer. 2016. Technical Debt and the Reliability of Enterprise Software Systems. *Management Science* 62, 5, 1487–1510.
[3] Z. Li, P. Avgeriou, and P. Liang. 2015. A Systematic Mapping Study on Technical Debt and its Management. *Journal of Systems and Software* 101, 3, 193-220.
[4] Project Management Institute. 2017. *A Guide to the Project Management Body of Knowledge*. Sixth edition. ISBN: 9781628251845.