FG Software Engineering
Leibniz Universität Hannover
Welfengarten 1
3. Stock, Flur G

Master-Arbeit

- zu vergeben -

Vermeidung von technischen Schulden in Startups beim Einkauf von Software

Hintergrund

Vielen Startups fehlt die nötige Expertise, um Software intern zu entwickeln. Daher beauftragen Startups oft externe Dienstleister, die die Softwareentwicklung übernehmen. Dabei werden oft technische Schulden in Kauf genommen, die aufgrund von fehlendem Wissen seitens der Auftraggeber (Startups) meist erst nach erfolgter Abnahme entdeckt werden. Oft sind diese technischen Schulden jedoch so gravierend, dass eine Weiterentwicklung des Prototypen bzw der Software nicht möglich ist. Eine Neuentwicklung kostet jedoch Geld, das Startups meist nicht haben. Aus diesem Grund ist es wichtig, das Bewusstsein für technische Schulden bei den Startups zu schärfen und gleichzeitig darauf aufmerksam zu machen, welche technische Schulden akzeptabel sind und welche nicht.

Aufgabe

Im Rahmen dieser Arbeit soll eine Checkliste erstellt werden, die es Startups ermöglicht, technische Schulden (TD) zu erkennen und zuverlässig zu vermeiden. Dazu wird zunächst im Rahmen einer Literaturstudie untersucht, mit welchen technischen Schulden Startups häufig Probleme haben. In einem nächsten Schritt werden die identifizierten Probleme den unterschiedlichen TD-Typen (z.B. Code Debt, Architecture Debt, Design Debt, Test Debt, Security Debt, Update Debt) zugeordnet, die wiederum den Phasen eines Entwicklungsprozesses zugeordnet werden können. Aus diesen Zuordnungen wird dann die Checkliste erstellt, die möglichst konkrete Maßnahmen beinhaltet, wie die unterschiedlichen Arten von Problemen frühzeitig identifiziert und verhindert werden können (z.B. durch statische Codeanalysen, Architekturreviews von Experten und ähnlichem). Dabei soll es anhand der Checkliste auch möglich sein, die einzelnen technischen Schulden anhand ihrer Relevanz und ihres Risikopotentials zu priorisieren (Code kann man ausbauen, aber späte Änderungen an der Architektur sind sehr kostspielig).
Im letzten Schritt soll das Entrepreneurial Software Engineering Model anhand der Checkliste erweitert werden, indem aufgezeigt wird, an welchen Punkten im Modell TD entstehen kann und an welchen Stellen das Modell erweitert werden muss, um TD vorzubeugen. Dies erlaubt einen Abgleich zwischen dem „typischen“ Vorgehen von Startups (wird im Modell dargestellt) und dem zur Vermeidung von technischen Schulden empfehlenswerten Vorgehen.

Dazu sind folgende Schritte durchzuführen:

1) Systematische Suche nach Literatur zu technischen Schulden in Softwarestartups
2) Mapping der gefundenen technischen Schulden zu bekannten TD-Typen und zu den Phasen in einem Softwareprozess
3) Ableiten einer Checkliste zur Identifikation von technischen Schulden und möglichen Gegenmaßnahmen
4) Erweiterung des Entrepreneurial Software Engineering Models zur Vermeidung von technischen Schulden
5) Verschriftlichung und Diskussion der Ergebnisse

Im Sinne des guten wissenschaftlichen Arbeitens sind alle Schritte nachvollziehbar und gründlich zu dokumentieren, sodass diese bei Bedarf erneut ausgeführt werden können. Für eine sehr gute Note ist eigenes Engagement und das Einbringen eigener Ideen notwendig. Darüber hinaus haben der Umfang der Checkliste und die Erweiterung des Modells maßgeblichen Einfluss auf die Note.