TITEL DES EXPOSÉS

Exposé zur Seminararbeit

Johannes Teklote
jotek001@stud.fh-dortmund.de

7091992

18. November 2020

1 Motivation

Laut [Bal09] entfallen 80% des Aufwands in der Softwareentwicklung auf deren Wartung wobei 40% davon benötigt werden, um die zu wartende Software zu verstehen. Somit liegt auch ein großer Teil der Kosten im Bereich der Wartung. Der Wartungsaufwand an sich lässt sich selten reduzieren, da weiterhin Anforderungsänderungen eingereicht werden können, Problembehebungen notwendig werden können oder Abhängigkeiten aktualisiert werden müssen. Allerdings lässt sich der Aufwand für das Verstehen des Codes reduzieren. Dazu ist es notwendig, dass der Code gewisse Qualitätsstandards erfüllt.

Ein Aspekt von qualitativ hochwertiger Software ist laut [ISO11] eine hohe Qualität in Betrieb und Wartung. Dazu gehören die Analysierbarkeit, Erlernbarkeit, Veränderbarkeit, Stabilität und Wartbarkeit der Software. Um Code zu schreiben, der diese Anforderungen erfüllt, gibt es Richtlinien, an die sich alle Entwickler in einem Team halten sollten. Diese Richtlinien besitzen zwar keine Allgemeingültigkeit, aber sie werden meist zu Projektbeginn festgelegt und gelten dann für die gesamte Projektlaufzeit. Zu diesen Regeln gehört beispielsweise, dass Code auf einer Ebene einheitlich eingerückt wird, dass öffnende geschweifte Klammern in der gleichen Zeile stehen wie der Code, der die Klammer erfordert oder aber auch dass Variablennamen sprechend sind und nicht aus einem Buchstaben bestehen, oder dass Parameter auf ihre Gültigkeit überprüft werden, bevor mit ihnen gearbeitet wird.

Die Einhaltung solcher Regeln scheint auf den ersten Blick nur von geringer Bedeutung zu sein, allerdings lenken beispielsweise Unstimmigkeiten in der Formatierung beim Lesen des Codes ab und sie erschweren es, sich auf den eigentlichen Code zu konzentrieren. In [Pra15] und [Spi11] werden diese Formatverstöße als Hintergrundrauschen beschrieben, das vom eigentlichen Code ablenken.

2 Problemstellung

Code, der nach den oben beschriebenen Regeln geschrieben wurde, ist laut [Pra15] leichter verständlich und dadurch auch leichter erlernbar und veränderbar. Um die Teammitglieder dazu zu bringen, sich an diese Regeln zu halten, gibt es zwei Möglichkeiten. Zum einen kann im Reviewprozess des Projekts ein automatischer Test eingebaut werden, der dafür sorgt, dass eigentlich funktionierender Code nicht angenommen wird, wenn er beispielsweise gegen Stylingrichtlinien verstößt oder es potentiell unsichere Variablenzugriffe gibt. Zum anderen können die Teammitglieder durch Belohnungen dazu motiviert werden, selber für die Einhaltung der Regeln zu sorgen und dies zu überprüfen. Um eine solche Motivation zu erreichen, kann Gamification verwendet werden. Hierbei werden Methoden genutzt, die aus Computerspielen bekannt sind. Dazu gehören beispielsweise Elemente wie Levels, Badges oder Leader Boards.

Beide Verfahren führen dazu, dass die Teammitglieder diese Programmierrichtlinien mit der Zeit von sich aus einhalten, ohne immer darauf hingewiesen werden zu müssen, entweder, weil sie gelernt haben, dass der eingereichte Code sonst aus Formgründen abgelehnt wird oder weil sie für besseren Code beispielsweise mehr Erfahrungspunkte bekommen. Es entsteht also ein Lerneffekt, der bewirkt, dass die Teilnehmer auch in Zukunft qualitativ hochwertigen Code einreichen. In beiden Fällen wird die eigene Motivation, guten Code zu schreiben, durch eine extrinsische Motivation verstärkt. Im Rahmen dieser Arbeit liegt der Fokus auf der Motivation durch Gamification.

3 Zielsetzung

Ziel dieser Arbeit ist die Erweiterung der Codeanalyseplattform coderadar um Gamification-Elemente zur Steigerung der Code-Qualität. Dazu werden zunächst ausgewählte Gamification-Elemente in coderadar implementiert. Anschließend wird eine Balancierungsmatrix für eine gleichmäßige Verteilung der Punkte, auf denen Levels, Leader Boards und ähnliches basieren, erarbeitet. Abschließend wird ein Feldversuch durchgeführt, um festzustellen, in wie weit die vorgestellten Maßnahmen die Softwarequalität nachhaltig verbessern konnten.

4 Vorgehensweise

Zunächst werden die Grundlagen dieser Arbeit vorgestellt. Dazu gehört die Erläuterung, was Gamification ist und in wie weit Gamification-Elemente in dieser Arbeit Verwendung finden. Daneben wird im Rahmen der Vorstellung verwandter Arbeiten darauf eingegangen, welche Implementierungen und Implementierungsansätze es bereits für Gamification-Elemente in Zusammenhang mit der Code-Qualität gibt. Außerdem wird

die Codeanalyseplattform coderadar vorgestellt, die als Basis für die im Rahmen dieser Arbeit erfolgende Implementierung dienen wird.

Anschließend wird das Spielkonzept vorgestellt. Dazu gehört zunächst die Auswahl der Metriken, anhand derer die Codequalität bewertet. Entsprechend diesen Metriken wird der konkrete Code dann untersucht. Anhand des Ergebnisses dieser Untersuchung werden anschließend die Punkte verteilt, auf denen später Levels oder Leader Boards beruhen. Des Weiteren werden die Spielmechanismen, die Spielregeln und das Spielziel erläutert. Außerdem wird eine Balancierungsmatrix ausgearbeitet, die für eine gleichmäßig gewichtete Punkteverteilung für unterschiedliche Aktionen sorgt.

Danach wird auf die Implementierung eingegangen. Zunächst wird die Modellierung der erforderlichen Änderungen an der coderadar-Plattform vorgenommen. Auf Basis dieser Modellierung wird eine Herangehensweise ausgearbeitet. Dann erfolgt die eigentliche Implementierung.

Anschließend erfolgt die Evaluierung. Dazu wird zunächst ein Projekt zur Untersuchung in coderadar eingecheckt und anhand der ausgewählten Metriken bewertet. Diese Bewertung soll als Basis für den Feldversuch dienen. Über eine gewisse Zeit wird das Projekt dann verfolgt und es wird beobachtet, in wie weit sich die Bewertung des untersuchten Projektes verändert. Anschließend sollen die Projektteammitglieder hinsichtlich der Effizienz der Gamification-Elemente in coderadar befragt werden.

Abschließend wird das Ergebnis der Arbeit zusammengefasst. Auf Basis der Bewertungsveränderung und der Ergebnisse der Umfrage wird ein Fazit zu der vorgestellten Implementierung von Gamification-Elementen und ihrem Einfluss auf nachhaltige Code-Qualität gezogen. Außerdem wird reflektiert, welche Herausforderungen es bei der Implementierung und der Durchführung des Feldversuchs gab. Zum Schluss wird ein Ausblick auf die weitere Arbeit gegeben.

Literatur

[Bal09] Balzert, Helmut: *Lehrbuch der Softwaretechnik: Basiskonzepte und Requirements Engineering*. Heidelberg : Spektrum Akademischer Verlag, 2009. http://dx.doi.org/10.1007/978-3-8274-2247-7_3. http://dx.doi.org/10.1007/978-3-8274-2247-7_3. – ISBN 978–3–8274–2247–7

[ISO11] International Organization for Standardization: System und Software-Engineering – Qualitätskriterien und Bewertung von System und Softwareprodukten (SQuaRE) – Qualitätsmodell und Leitlinien. 2011. – Standard

[Pra15] Prause, Matthias Christian und J. Christian und Jarke: Gamification for enforcing coding conventions, 2015, S. 649–660

[Spi11] Spinellis, Diomidis: elytS edoC. In: *IEEE Software* 28 (2011), März/April, Nr. 2, S. 104–103