

TITEL DES EXPOSÉS

Exposé zur Seminararbeit

Johannes Teklote

jotek001@stud.fh-dortmund.de

7091992

18. November 2020

1 Motivation

Laut [Bal09] fallen 80% des Aufwands in der Softwareentwicklung während der Wartung an, wobei 40% davon zum Verstehen der Software benötigt werden. Somit liegt auch ein großer Teil der Kosten in diesem Bereich. Der Wartungsaufwand an sich lässt sich an der Stelle selten reduzieren, da weiterhin Anforderungsänderungen eingereicht werden können, Problembehebungen notwendig werden können oder Abhängigkeiten aktualisiert werden müssen. Allerdings lässt sich der Aufwand, der zum Verstehen des Codes notwendig ist, reduzieren. Dazu ist es notwendig, dass der Code gewisse Qualitätsstandards erfüllt.

Ein Aspekt von qualitativer Software ist laut [ISO11] die Qualität in Betrieb und Wartung. Dazu gehören die Analysierbarkeit, Lernbarkeit, Veränderbarkeit, Stabilität und Wartbarkeit. Um Code zu schreiben, der diese Anforderungen erfüllt, gibt es Richtlinien, an die sich alle Entwickler in einem Team halten sollten. Zu diesen Regeln gehört beispielsweise, dass Code auf einer Ebene einheitlich eingerückt wird, dass öffnende geschweifte Klammern in der gleichen Zeile sind, 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.

Der Einfluss dieser Regeln scheint auf den ersten Blick sehr gering, 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, die vom eigentlichen Code ablenken.

2 Problemstellung

Code, der sich entsprechend an die Regeln hält, ist laut [Pra15] leichter verständlich und dadurch auch leichter erlernbar und veränderbar. Um die Teammitglieder dazu zu bringen, sich entsprechend an die 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 verwendet, die aus Computerspielen bekannt sind. Hierzu gehören beispielsweise Levels, Badges oder Leader Boards.

Beide Verfahren führen dazu, dass die Teammitglieder diese Richtlinien mit der Zeit von sich aus einhalten, ohne immer darauf hingewiesen zu werden, entweder, weil sie gelernt haben, dass der eingereichte Code immer 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 qualitativen Code einzureichen. 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 die Gamification.

3 Zielsetzung

Ziel dieser Arbeit ist die Erweiterung der Codeanalyseplattform coderadar um Gamification-Elemente zur Steigerung der Code-Qualität. Dazu werden zunächst die Gamification-Elemente in coderadar implementiert. Anschließend wird eine Balancierungsmatrix für eine gleichmäßige Verteilung der Punkte erarbeitet. Abschließend wird eine 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 zum Einen die Erläuterung, was Gamification ist und in wie weit Gamification-Elemente in dieser Arbeit Verwendung finden. Außerdem 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. Darüber hinaus 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 und entsprechend die Punkte verteilt werden. Des Weiteren werden die Spielmechanismen, die Spielregeln und das Spielziel erläutert. Außerdem wird eine Balancierungsmatrix ausgearbeitet, die für eine gleichmäßige Punkteverteilung sorgt.

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

Anschließend erfolgt die Evaluierung. Dazu wird zunächst ein Projekt 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 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 als auch bei 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