Identifying Development-Metrics for Use in a Gamified Mobile Web Application to Support Software Development

Tim Wenzel, Thomas Franke, and Rainer Wasinger
Faculty of Physical Engineering/Computer Sciences,
University of Applied Sciences Zwickau,
08056, Germany
Email: {tim.wenzel.gai, thomas.franke, rainer.wasinger}@fh-zwickau.de

Michael Körner
msg systems ag
Zwickauer Straße 16a,
09112 Chemnitz
Email: michael.koerner@msg.group

***Abstract*—Many software development projects fail before reaching their set goal. This is often due to financial reasons, but psychological aspects, such as motivation can also play a role in the failure of projects. This work outlines a mobile web application prototype that provides first insights into the gamification of software development. The work has two goals, namely: 1. to identify evaluation metrics that lead to successful software development; and 2. to identify from these metrics the criteria suitable for gamification and apply these criteria to a mobile web application prototype. The aim is to provide first insights into what gamification in software development could look like.**

Keywords–Mobile web application; Gamification; Software development metrics.

I. INTRODUCTION

Various studies show that many projects, especially in the field of software development, fail before reaching their set goal [1]. If one considers both the actual failure of a project (i.e., a discontinuation of development), as well as projects that exceed their allocated time and financial costs, then about 55% of IT projects can be considered to be a failure [2]. In addition to technical and economic factors, psychological aspects such as motivation can also play a role in the failure of a project. A literature review of 92 papers relating to motivation in Software Engineering [3] found that there is however still little understanding of how software engineers are motivated.

One possible approach to counteracting the problem of motivation is to use playful means to maintain the motivational drive among project members. This type of motivation is called gamification, and previous studies show that it can be successfully used to increase motivation. For example, in [4], the use of gamification in enterprise collaboration systems is investigated. Based on a 5-point Likert scale ranging from "not at all motivating" to "very strongly motivating", the study reported that 57% of participants [N=35] found gamification to be strongly to very strongly motivating, with an additional 34% of participants finding gamification to provide some motivational benefit, and only 6% finding gamification to be not at all motivating.

The goal of this work is to apply gamification to the field of software development. In particular, criteria for the evaluation of developers are analysed for relevance, and then transferred to a playful mobile web application prototype that is designed to demonstrate first experiences of gamification for use in software development.

II. BACKGROUND

Figure 1 shows the interest in the topic of gamification, as recorded by the online service Google Trends (https://trends.google.com). This service provides information about which search terms were queried by users in a certain period of time and how often. The figure shows that interest in the topic of gamification began to increase significantly around 2011 and has remained high since then.

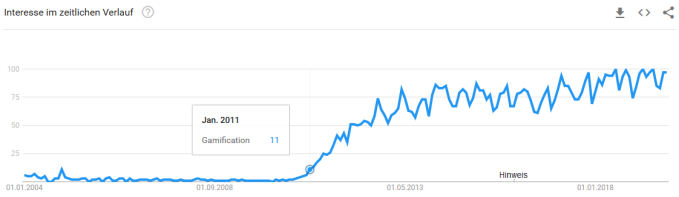

Figure 1. Interest in the topic of gamification based on Google Trends data from 2004 to 2019, retrieved: Nov,2019.

The term Gamification has been defined in the literature in a number of ways. In [5], gamification is defined as 'a process of improving a service by providing a playful experience to the user in order to achieve a better result.' A shorter and more succinct definition is given in [6]: 'The use of game design elements in non-game contexts'. A more specific definition in the context of economic enterprises defines it as 'the integration of playful elements in the work and learning processes of the company' in which 'the users feel a higher motivation to complete tasks through various playful approaches' [7].

Although gamification has been used more generally to increase productivity (e.g., Microsoft Ribbon Hero [8] and the Audi Virtual Training application [9]), its use in software development is still in its infancy.

A. The Use of Gamification in Software Development

Software development typically involves core activities like the planning of software, as well as the actual programming, and testing of code. One example of an application in which gamification is used in the context of software development is Stack Overflow (https://stackoverflow.com). Stack Overflow is a question and answer site to support professional and enthusiast programmers. Some of the gaming elements that are used here include reputation points and badges of varying categories (bronze, silver, and gold). These elements are

 ISBN: 978-1-61208-808-2

designed to achieve a good quality of questions and answers, as well as to promote awareness of the platform and increase new users [10]. A second example is outlined in [11], in which test coverage and code documentation is improved through the use of gamification. In comparison to those works, the focus of this paper is on gamifying the actual programming of code.

B. Gamification Elements

In principle, gamification is intended to increase interest or maintain interest over a certain period of time. The objective is to make tasks more attractive and interesting over the long term. Playing is associated with fun and joy, and the goal is to transfer the joy of playing to a joy of work.

Various elements can be used when implementing a corresponding gamification concept. For example, experience points, awards, and virtual currency are used in [12]. Levels and quests are two further elements that can make a game interesting over a longer term, i.e., in which the user is motivated to reach a next level. Some elements do not always work without problems, and a balance must be found between the task at hand and the game concept. If this balance is not struck, tasks can be seen as long-winded and demotivating [13] or too easy [14]. Certain elements have also been previously shown to be counterproductive, e.g., the use of 'time' and 'ranking lists' (see [15] and [16]). Both of these elements were thus excluded in this work.

III. Identification of Evaluation Criteria used in Software Development and Relevant to Gamification

If the concept of gamification is to be used specifically in the field of software development, the corresponding gamified application must be based on certain criteria.

The first goal of this work was to identify criteria that lead to successful software development. This was done by collecting software metrics from the literature [17], from existing software development tools (i.e., GitLab [18] and SonarQube [19]), and from interviews with software development professionals. The interviews were conducted with seven developers and three testers from the software development company msg systems ag (https://www.msg.group). Participants were asked about the software metrics they frequently used in projects. These interviews provided the ability to draw on evaluation criteria from practice, and helped in selecting the final criteria to be used in the mobile web application prototype as described in Section IV.

In this paper, only evaluation criteria specifically concerning *individual* software developers will be discussed. Evaluation criteria for development *teams* are outside the scope of this paper. Similarly, the criteria focus on software *development*. Software *testing* is also considered out of scope.

Quantitative Metrics: GitLab [18] is a web-based tool that provides various project management and bug tracking functionalities based on the version-control system Git, and it provides many different metrics for developers. In this work, GitLab provides the quantitative metrics employed in the mobile web application prototype, i.e., those that are clearly countable. These are described below, with examples of their usage in the mobile web application presented in Table I.

TABLE I. TABULAR OVERVIEW OF THE EXAMINED EVALUATION METRICS, INCLUDING EXAMPLES OF USE

Evaluation Metric	Example Usage within the Mobile Web Application
Lines of Code	Write X lines of code; Write your X^{th} line of code.
Commits	Reach your first commit; Reach X commits.
Issues	Close your first issue; Close X issues.
Merge Requests	Process your first merge request; Process X merge requests.
Merges	X% of your merge requests were without error.
Bugs	Lower your bug count by X; Solve X bugs.
Security	Solve X security issues.
Code Smells	Hurray! Your code has no code smells (reached X times).
Code Duplications	Remove X code duplications.

- **Lines of Code**: Lines of Code describe how many lines of code a developer has effectively written. Neither blank lines nor code lines that have been changed at a later date are included in this statistic.
- **Commits**: A commit adds the latests changes to a source code repository, and it typically relates to multiple changes that are meaningful and address a specific problem.
- **Issues**: An issue is usually a coherent task, which is often limited in time and pursues a clear goal. If this defined goal is achieved, the issue is marked as completed. The number of completed issues is of particular importance for this evaluation metric.
- **Merge requests**: Merge requests are the intention or the request to transfer changes from one development branch to another. This request is processed by other project members by checking the code to be merged for correctness. This can be done by a single person, however more often than not, confirmation from multiple parties is required. If no errors can be found, the code is transmitted by a corresponding confirmation.
- **Merges**: Merges represent the actual acceptance of a merge request, typically by one or more parties.

Qualitative Metrics: The mobile web application also incorporates qualitative metrics through the use of the SonarQube tool. SonarQube [19] is an open-source platform that evaluates the technical quality of program code using statistical methods and assigns numerical values to traditionally qualitative metrics. The qualitative metrics employed in the mobile web application are now described, with example usage also shown in Table I.

- **Bugs**: A software bug is an error or flaw in a computer program. The fewer such bugs occur in a developer's code, the better the code. If a developer's code contains a particularly high number of errors, the evaluation metric will be worse.
- **Security**: Security is also an important evaluation metric, and good knowledge of possible security problems and their solutions is often of paramount importance. Various security risks are outlined in lists like the CWE Top 25 (Common Weakness Enumeration: https://cwe.mitre.org/data/definitions/1200.html).
- **Code Smells and Code Duplications**: A much less security-critical aspect are so-called code smells. A code smell is the term used to describe unclean code passages which, although they function without errors,

should be revised in order to improve the understanding of the written program code. Examples include code duplication, too long methods or classes, too many passed parameters, and mandatory comments.

IV. Prototype Deployment in the form of a Mobile Web Application

In this section, the evaluation criteria identified in Section III are applied to our mobile web application prototype. It should be emphasized that this is not a finished, ready-to-use application. The prototype is only intended to show what gamification can look like and how it feels when used.

A. Software Design Issues and Technologies

A technical requirement of the application was that it should be usable on many different devices. This includes devices with different operating systems (i.e., iOS, Android; also PC) as well as different device form-factors (e.g., smartphone, tablet, and laptop). To satisfy this requirement, the prototype was implemented in the form of a web application. A further benefit of this approach is that users do not need to download and install the application for each device. All that is required to use the application is a web browser.

The application is realized with the open source web framework Ionic (https://ionicframework.com). With the Ionic framework, applications can be implemented using HTML5 and CSS. Furthermore, Ionic allows for the creation of so-called Progressive Web Apps (PWAs), i.e., web applications that incorporate features of native apps. They can also be customised to adapt to different screen sizes (e.g., those of smartphones, tablets, and computer screens). Another advantage is that the framework provides the ability to create native iOS and Android applications from a single code-base.

The version-control system Git was used together with GitLab to incorporate many of the qualitative gamification metrics (i.e., commits, issues, merge requests, lines of code, and merges), while SonarQube was used to incorporate the qualitative metrics (i.e., bugs, security vulnerabilities, and code smells). Both GitLab and SonarQube provide REST APIs, through which web requests are made to retrieve the metric evaluation information used by the mobile web application prototype.

B. Mobile Web Application User Interface

Figure 2 shows the user interface for the developed mobile web application prototype. Upon startup, the user must first login with a username and password. For the purpose of this prototype, the user-accounts for the app are manually created by an administrator. After login, the user sees his or her personalised 'Dashboard' as shown in Figure 2A. The dashboard provides each user with an overview of how many rewards he/she have already received and the value of each one. This becomes relevant for a possible competition at a later date. In addition to these functions, the user can store their GitLab key here to allow them to connect to GitLab. These keys need to be created in the user's own personal GitLab account.

If the user switches to the 'Progress' tab (Figure 2B), he/she will first receive an overview of the evaluation criteria. In this view, the criteria are divided into the categories 'quantitative-' and 'qualitative-' metrics. Furthermore, each criterion has a help icon that can be pressed to receive a brief description on how it is to be interpreted. This is especially relevant for new users.

If a user now selects one of the criteria, he/she will be directed to a more detailed page where the exact progress relating to that particular criterion can be viewed. This can be seen for the case of 'commits' in Figure 2C. In addition to textual feedback, progress bars are used here as a gamification element to provide visual support for the user.

When a user reaches the number required for each level, in this case for 'commits', a reward can be claimed by clicking on the corresponding 'Get Reward' button. These rewards are divided into low-, medium-, and high- value items, with the low value items being received for the initial level and higher-value items being received for later (and more complex) levels (see Figure 2A for an example of different rewards). Each reward has a value between zero and one. The higher the value, the more the reward has a positive effect in a competition/challenge. The reward values are defined as low-value (0.1 to 0.4), medium-value (0.3 to 0.7), and high-value (0.6 to 0.9), with a slight overlap in range to allow for 'chance'.

In the 'Challenges' tab (see Figure 2D), users can take part in a short, voluntary competition. Challenges can only be initiated if both players have declared their willingness to compete. Players who have not given their consent will be shown as 'Not Ready' in the overview. The current implementation of challenges is somewhat primitive and assumes that players have been using the application for a similar period of time. Future work would see this feature extended to incorporate the concept of weekly challenges.

If one user challenges another willing user, a dialog opens, in which the winner of the competition is calculated. This is calculated based on the respective rewards of the players. The more rewards, the higher the probability of winning. During the calculation of the result, the respective winning probabilities of the players are displayed. Afterwards the winner of the competition is displayed (see Figure 2E). The calculation of the result, and thus the determination of the winner of a competition, for Player 1 (P_1) when competing against Player 2 (P_2) can be calculated as follows:

$$P_1 = \frac{\sum_{Rewards(P_1)}}{\sum_{Rewards(P_1+P_2)}}.$$

In the final step of determining a winner, a random variable is generated to bias each player's rewards by a slight value of chance. Based on this value and the chances of winning for each player, a winner is determined.

V. Conclusions and Future Work

The goal of this work was to identify evaluation metrics that lead to successful software development, and to identify from these metrics the criteria suitable for gamification and apply these criteria to a mobile web application prototype. This was done in order to provide first insights into what gamification in software development could look like. First discussions of the resulting prototype with professional developers have highlighted that the overall development objective often has a decisive effect on the relevance of a specific gamification criterion (e.g., whether the development is to prioritise speed, security, or error-resistance).

 ISBN: 978-1-61208-808-2

Figure 2. User Interface (UI) of the mobile web application, showing A) the Dashboard and the user's reward items, B) a Progress overview of the evaluation criteria, C) details of the quantitative criterion 'commits', D) the selection of an opponent for a challenge, and E) the outcome of such a challenge.

Future work will now focus on the ability to tune the gamification parameters to individual project needs, as well as to extend the concept of weekly challenges.

ACKNOWLEDGEMENT

The authors would like to thank msg systems ag for their support and supervision of this work.

REFERENCES

[1] J. C. Schopp and M. Goeken, "Erfolgsfaktoren und Misserfolgsfaktoren im Projektmanagement- ein Systematischer Review [Success and failure factors in project management]," in Projektmanagement und Vorgehensmodelle. Gesell. für Informatik, 2018, pp. 51–61.

[2] C. Becker and E. Huber, "Die Bilanz des (Miss)-Erfolges in IT-Projekten [The balance of (mis)-success in IT projects]," Pentaeder, Tech. Rep., 2008. [Online]. Available: https://www.pentaeder.de/projekte/2008/12/10/ [retrieved: Aug,2020].

[3] S. Beecham, N. Baddoo, T. Hall, H. Robinson, and H. Sharp, "Motivation in Software Engineering: A systematic literature review," Information and Software Technology, vol. 50, no. 9, 2008, pp. 860–878.

[4] P. Schubert, J. Hager, and L. Paulsen, "Auswirkungen von Gamification in Enterprise Collaboration Systems [Effects of gamification in Enterprise Collaboration Systems]," in Mensch und Computer 2014 - Tagungsband, A. Butz, M. Koch, and J. Schlichter, Eds. Berlin: De Gruyter Oldenbourg, 2014, pp. 3–14.

[5] K. Huotari and J. Hamari, "Defining Gamification: A Service Marketing Perspective," in Proceeding of the 16th International Academic MindTrek Conference. New York, NY, USA: ACM, 2012, pp. 17–22.

[6] S. Deterding, D. Dixon, R. Khaled, and L. Nacke, "From Game Design Elements to Gamefulness: Defining "Gamification"," in Proceedings of the 15th International Academic MindTrek Conference. New York, NY, USA: ACM, 2011, pp. 9–15.

[7] S. Stieglitz, "Gamification – Vorgehen und Anwendung [Gamification - Procedure and Application]," HMD Praxis der Wirtschaftsinformatik, vol. 52, no. 6, 2015, pp. 816–825.

[8] J. Lopez, "Microsoft Unveils Ribbon Hero 2.0," 2011. [Online]. Available: https://www.gamification.co/2011/04/26/microsoft-ribbon/ [retrieved: Aug,2020].

[9] S. Killian, "Audi Virtual Training: New gamification learning concept in the digital car dealership," 2018. [Online]. Available: https://www.audi.com/en/career/working-world/gamification-audi-virtual-training.html [retrieved: Aug,2020].

[10] D. Kavaler and V. Filkov, "Determinants of quality, latency, and amount of Stack Overflow answers about recent Android APIs," PLOS ONE, vol. 13, no. 3, 03 2018, pp. 1–29.

[11] D. J. Dubois and G. Tamburrelli, "Understanding Gamification Mechanisms for Software Development," in Proceedings of the 9th Joint Meeting on Foundations of Software Engineering. New York, NY, USA: ACM, 2013, pp. 659–662.

[12] S. Strahringer and C. Leyh, Gamification und Serious Games - Grundlagen, Vorgehen und Anwendungen. Wiesbaden: Springer Vieweg, 2017.

[13] H. S. Durinkowitz, Crash-Kurs für Verkaufsleiter: Vom Start weg auf der Gewinnerseite [Crash course for sales managers: On the winning side right from the start]. Wiesbaden: Gabler Verlag, 2013.

[14] T. Dehghan, "Welche konkreten Ziele verfolgen Sie mit der Messeteilnahme? [What concrete goals are you pursuing with your participation in the trade fair?]," in Der Messecoach. Wiesbaden: Springer Fachmedien, 2018, ch. 2, pp. 9–25.

[15] H. Müller, "Zur Ethik von Rankings im Hochschulwesen," University of Münster, Tech. Rep., 2013. [Online]. Available: https://ideas.repec.org/p/zbw/ciwdps/12013.html [retrieved: Aug,2020].

[16] M. Nake, "Motivationsstrategien in der Gamification Handbuchliteratur," Bachelor's Thesis, Leuphana Universität Lüneburg, 2014.

[17] G. Gousios, E. Kalliamvakou, and D. Spinellis, "Measuring Developer Contribution from Software Repository Data," in Proceedings of the 2008 International Working Conference on Mining Software Repositories. New York, NY, USA: ACM, 2008, pp. 129–132.

[18] GitLab, "GitLab Documentation Version 12.7," 2020. [Online]. Available: https://docs.gitlab.com/ee/api/ [retrieved: Aug,2020].

[19] SonarQube, "SonarQube Documentation Version 8.1," 2020. [Online]. Available: https://docs.sonarqube.org/latest/ [retrieved: Aug,2020].

 ISBN: 978-1-61208-808-2