Achievement Unlocked: A Case Study on Gamifying DevOps Practices in Industry

Patrick Ayoup
Department of Computer Science and Software Engineering
Concordia University
Montreal, Canada
p_ayoup@encs.concordia.ca

Diego Elias Costa
Department of Computer Science
Université du Québec à Montréal
LATECE Lab
Montreal, Canada
costa.diego@uqam.ca

Emad Shihab
Department of Computer Science and Software Engineering
Concordia University
Montreal, Canada
eshihab@encs.concordia.ca

ABSTRACT

Gamification is the use of game elements such as points, leaderboards, and badges in a non-game context to encourage a desired behavior from individuals interacting with an environment. Recently, gamification has found its way into software engineering contexts as a means to promote certain activities to practitioners. Previous studies investigated the use of gamification to promote the adoption of a variety of tools and practices, however, these studies were either performed in an educational environment or in small to medium-sized teams of developers in the industry.

We performed a large-scale mixed-methods study on the effects of badge-based gamification in promoting the adoption of DevOps practices in a very large company and evaluated how practice adoption is associated with changes in key delivery, quality, and throughput metrics of 333 software projects. We observed an accelerated adoption of some gamified DevOps practices by at least 60%, with increased adoption rates up to 6x. We found mixed results when associating badge adoption and metric changes: teams that earned testing badges showed an increase in bug fixing commits but output fewer commits and pull requests; teams that earned code review and quality tooling badges exhibited faster delivery metrics. Finally, our empirical study was supplemented by a survey with 45 developers where 73% of respondents found badges to be helpful for learning about and adopting new standardized practices. Our results contribute to the rich knowledge on gamification with a unique and important perspective from real industry practitioners.

CCS CONCEPTS

• **Software and its engineering**;

KEYWORDS

gamification, software engineering, devops

ACM Reference Format:
Patrick Ayoup, Diego Elias Costa, and Emad Shihab. 2022. Achievement Unlocked: A Case Study on Gamifying DevOps Practices in Industry. In *Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE '22), November 14–18, 2022, Singapore, Singapore.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3540250.3558948

ESEC/FSE '22, November 14–18, 2022, Singapore, Singapore

ACM ISBN 978-1-4503-9413-0/22/11…$15.00
https://doi.org/10.1145/3540250.3558948

1 INTRODUCTION

The tools, processes, and best practices in software development are constantly evolving as different trends emerge [3]. Although adopting new practices can be very attractive, provoking meaningful change and standardizing a heterogeneous environment at scale is a difficult task [40]. Getting developers, who have been using the same techniques for years, to change their ways is a challenge that needs careful thought and planning.

One creative solution to this problem is to incorporate gamification [15]. Gamification is the inclusion of game elements in non-game context to motivate user activity and improve engagement [10]. Gamification has been reported to show positive results [15, 33, 38], particularly when employed to promote the adoption of new tools and practices in software development [12, 38]. Although these studies showed promising results, the case studies were performed with students [12, 24], or a small to medium-sized teams of developers in industry [15, 17, 30].

Our paper complements the large body of work by performing a large-scale study of gamification and its impact in a real industrial environment. Specifically, we investigate the use of gamification over a year across 333 software development projects at a large company. In our case study, badges associated with DevOps best practices were presented to developers with the aim of improving certain key performance indicators (KPIs).

We conduct a mixed-methods study to evaluate the relationship between gamification and the adoption of new practices. First, we study whether or not gamification is effective in promoting the adoption of new process and practices to see if it can act as an accelerant for changing behavior within an organization. Then, we investigate how the metrics of software development teams shift after making changes to their practices in order to earn badges. Finally, we surveyed practitioners working on these projects to learn how they react to gamification and perceive its impact. The aforementioned questions are of paramount importance to the studied organization (and others, we believe) to understand the effect of their efforts and how to better improve the existing gamification mechanisms.

Our work contributes to practitioners and the research community by:

- Presenting the first large-scale study on the effects of gamification on the adoption of DevOps practices in **industry**. Our study includes 333 projects from a large software development company.
- Evaluating how changes in the DevOps practices encouraged by gamification are associated to changes in Delivery, Quality, and Throughput metrics of software projects.
- Reporting qualitative insights from a survey with 45 industry practitioners about their reactions and perceived impact of gamification.

This study provides a series of implications on gamification as a strategy to change practices in industry. Our case study shows that gamification, if carefully designed, can be a powerful driver of new development practices, even in large and heterogenous industrial contexts. However, measuring the benefits of practice adoption using conventional delivery, quality, and throughput metrics can be difficult. Only some badges showed an association with project metrics change, and our results pointed to some trade-offs between quality metrics and development throughput. In addition, practitioners are driven by the benefits that gamified practices entail, and only to a lesser extent by the competitiveness and achievement provided by games. For instance, practitioners were drawn to deployment and testing practices for the prospect of automation and reducing manual work and improving software quality. Finally, we report on criticism and limitations that need to be addressed by the community to improve the effectiveness of gamification as a catalyst for behavioral change.

2 RELATED WORK

In this section, we describe the fundamentals of gamification and dive in the related works that investigate gamification in SE.

2.1 Gamification and Motivation

In its simplest definition, gamification is the application of game elements and characteristics in a non-game environment [10]. Gamification can manifest itself in many forms by applying game elements such as points, badges, levels, quests, and leader boards to support user engagement and enhance positive patterns [19, 31]. Each game element has the potential to affect user behavior differently [27]. For instance, leaderboards emphasize relative performance and may drive users competitiveness [27], while badges, give the user a sense of self-improvement, and have shown to steer users' long-term behavior towards gamified goals [18].

Several studies have investigated the effects of gamification in a variety of domains [37]. From education [11] and health [23], to marketing and commerce [26], meta-studies have shown benefits of gamification on user engagement and satisfaction [11, 19, 23, 37]. Still, studies have pointed out important limitations of gamification. Not all activities and contexts can be equally and effectively gamified. Users' perception of gamification vary considerably based on age and gender [25], user's receptivity to external rewards [27], and the meaning assigned to gamified elements [6]. Finally, gamification's effectiveness is deeply connected to the design of game elements, and how they interact with the user [27, 35]. A badly designed gamification system can sap user's motivation [20, 28, 46] and steer users to chase metrics instead of encourage behavioral change [27].

As a result, systematic studies unanimously state that more studies are needed to better understand gamification benefits and limitations [11, 19, 29]. Particularly, researchers urge for large-scale studies that assess gamification effectivity on the long-term in the wild to complement studies in a lab environment [19, 29]. Our study contributes to the literature by assessing gamification effectiveness in encouraging practitioners to adopt DevOps practices on a large software development company, over a period of one year.

2.2 Gamification in Software Engineering

Software engineering practitioners are no stranger to gamification. Major code-centric social platforms such as Stack Overflow use badges to evaluate users' commitment, competence and trustworthiness in the platform [2]. Open-source projects in GitHub frequently employ badges to signalize to the community aspects related to the project quality, such as test coverage and build status [41]. Given its prominence, the effects of gamification has been studied in Software Engineering (SE) education [1], and in open-source and industrial software development [9].

Most works that study gamification in SE, focused on educational settings [1, 12, 24, 33, 38]. Alhammad and Moreno performed a systematic study on 21 papers that study gamification in SE education [1]. They found that gamification has reported mostly positive results in improving students engagement and, to a lesser extent, improving students knowledge. Dubois and Tamburrelli [12] reported that students that participated in a gamified course showed better results, motivated by competition with their colleagues. Singer and Schneider, on the other hand, reported mixed results when employing gamification to encourage students to use control version systems more frequently [38]. Code review has also been gamified in a study by Kandelwal et al. [24]. Comments originating from gamified systems were perceived as more useful by users, however the time needed to review the code was longer and uncovered a similar number of bugs in reviews from non-gamified environments.

Some works investigated gamification in open source software projects [28, 42, 43]. Vasilescu studied the engagement and contributions of developers to open source software projects and found that due to the recognition gamification provides, developers are more willing to engage in discussion and contribute more [43]. Open source software projects also commonly use badges to show to the community the adherence to good practices of software development (e.g., test coverage, build status), and Trockman et al. study showed that badges are mostly reliable as a signal of best practices [42]. However, gamification has also been shown to steer developers behavior towards unwanted directions [28], hence, the gamification system needs to be carefully designed.

Finally, a few studies have assessed gamification in industrial settings, most commonly in small and medium sized team of developers [15, 16, 30]. Garcia et al. proposed a framework for incorporating gamification into software development tools and performed a case study at a small company with 19 practitioners [16]. The authors reported seeing a 20% increase in the usage of the requirement and issue tracking tools. Neto et al. [30] developed a plugin for Redmine including several gamification elements and evaluated its effectiveness in a case study involving 19 developers from a small company. While many developers felt the gamification had

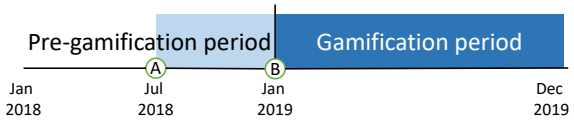


Figure 1: Timeline of the Gamification related events

positive effects on their work, the results were inconclusive as to whether or not developer productivity was improved. Foucault et al. [15] also performed an industrial case study with a system they built to gamify the adoption of good coding practices and the usage of static analysis tools involving 67 participants between two companies. Feedback from developers was mostly positive, showing that a sense of competition motivated developers to address static analysis warnings more seriously.

Our study complements above mentioned works by investigating gamification at scale in industry. Over 300 projects which use a variety of technologies, and solve a number of different business problems are observed in this study. The developers building and maintaining these projects also have a wide range of professional experience levels and backgrounds. Additionally, this study looks at gamifying a variety of practices targeting different phases of the software development lifecycle, while past work mainly focused on one single aspect (i.e., version control).

3 CONTEXT AND TIMELINE

This case study is centered around a large, multi-national company with a particularly large technology division comprised of more than 20,000 practitioners spread across the world. While development teams have the freedom to make decisions on the tools, technologies, and processes they adopt, there are a number of key best practices which should be more widely adopted. The gamification system under study is an initiative towards homogeneizing and promoting the best practices adopted by teams in the company.

In July 2018, an effort began to investigate DevOps best practices which would be beneficial for the development community. The output of this effort is a set of **DevOps Guidelines** suggesting which practices and tools a team is encouraged to prioritize and why they would be beneficial. These guidelines were socialized in July 2018 as marked by event A in Figure 1, and served as the basis for the badges in the studied gamification system. In December 2018, the gamification system was announced and detailed to the development community, and released for general use in January 2019 (event B). Given that each of these events build on each other on a path towards DevOps adoption, it is expected that the events leading to the deployment of the gamification system may influence the adoption of DevOps practices to some degree. Hence, while we aim to study the effect of the gamification system (event B), we include event A in the study to control for eventual effects of the guideline in promoting the adoption of DevOps practices.

4 CASE STUDY DESIGN

The main vision behind the gamification system was to promote the adoption on DevOps practices with the ultimate goal of enabling software development teams to deliver more functionality, more quickly, while maintainining software quality and stability. In this context, the design of our study centers on investigating the impact of gamification of DevOps practices under three main aspects:

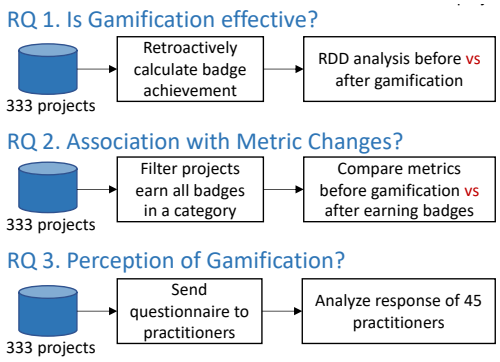


Figure 2: Methodology Overview

RQ1: Is gamification effective as a means to promote the adoption of new practices? We investigate how many projects worked towards earning the badges and what badges were more effective in encouraging the adoption of new practices.

RQ2: How does gamification impact the metrics of projects due to earning badges? Naturally following from RQ1, for projects that are earning badges, this question investigates how each badge earned is associated with change in their key metrics.

RQ3: How do software developers react to gamification and perceive its impact? We conducted a survey with software developers to better understand their motivation to adopt badges and how they perceive their impact on their project metrics.

Figure 2 presents a high-level abstraction of our methodology. In the remainder of the section, we describe the system of badges deployed by the company under study (Section 4.1) and the metrics we select to evaluate the delivery, quality, and throughput of teams before and after gamification (Section 4.2).

As this study was performed at a company in industry, the data used is proprietary and cannot be made available to the community. For this reason, numerical values are expressed in a relative form to properly retain anonymity.

4.1 Badges

The gamification system used in this case study leverages badges which are publicly displayed to the development community on each project's home page. For clarity of organization, badges are grouped into categories according to the following attributes of the software development lifecycle: deployment, git, quality tooling, review, stability, and testing.

A badge is a gamification element that serves as an indicator detailing whether or not the software development team working on a project has adopted a certain practice. Each badge is assigned an achievement requirement which must be met and maintained in order for it to be achieved. In order to encourage developers to adopt a given practice, a badge would be created targeting that practice with a requirement that can be evaluated to determine whether or not that practice has been adopted. For example, to encourage developers to adopt the practice of reviewing pull requests, a badge was created with the achievement requirement that at least 10% of pull requests on a project must have evidence of review in order to achieve that badge.

The primary intention of deploying these badges is to encourage software development teams to learn new practices and to foster

Table 1: The badges considered in our study. The column "RQ1" depicts the badges that could be retroactively calculated for projects before the gamificaton period and are included in our RQ1 analysis. RQ2 and RQ3 analyses include all badges.

Category	Badge	Requirement	Rationale	RQ1
Deployment	Deployments are Automated	Automate deployment procedures	Save time with repetitive activities and avoid human error	✓
	Post-Deployment Verification is Automated	Automate post-deployment verification procedures	Save time with repetitive activities and avoid human error	✓
	Project has Automated Deployment CI Job	Project can be automatically deployed from CI Pipeline	Encourage teams to adopt continuous delivery	✓
Git	Project uses Trunk Based Development	The majority of releases come from the same branch	Simplify development and release workflows. Promote the use of feature flags.	✓
Quality Tooling	Static Analysis / Linters are Used	Run quality tooling as part of automated builds	Identify code smells earlier in the software lifecycle	–
Review	Pull Requests are Reviewed	At least 10% of pull requests have comments by peers	Identify requirement and semantic errors earlier	–
Stability	Failed Builds are Fixed Quickly	Mean time to fix is under 24 hours	Keep target environment stable to enable continuous delivery	–
Testing	Automated Tests are Run on Builds	Run automated tests and persist test results for each build	Produce evidence of testing to improve confidence in more frequent changes	✓
	Unit Tests are Fast	Total unit test runtime is less than 5 minutes	Keep delivery pipeline flowing smoothly	–

a sense of transparency and achievement. When the badges were announced, it was explained what the badges were, that they are awarded to teams, not individuals, and that their adoption was not mandated by upper management or related to employee performance reviews in any way. Developers were informed that they simply serve to be recommendations of best practices and they are there to help if they wish to use them. Alongside the badges, documentation on how to achieve each one was made available to all developers. The badges considered in this study are outlined in Table 1 along with the rationale for each badge's design. As such, all badges are awarded to a team working on a project, and not individuals. Additionally, it was one of the main design philosophies of the badges that they do not single out individual developers, or put teams to compete against eachother.

4.2 Metrics

The goal behind the implementation of gamification is to promote new practices that enable teams to deliver software more quickly while optimizing quality and stability. To assess this, we select metrics that cover different aspects of software delivery, quality and throughput. Delivery metrics allow us to evaluate how quickly a team is delivering new functionality, quality metrics give a signal as to how software quality changes with newly adopted practices, and throughput metrics give an image on the quantities produced at both the contributor level (commit and pull request counts) and product level as a team (release count). Each selected metric is described in Table 2.

4.3 Data Selection

The data used for this case study is extracted from the following three systems: JIRA, Git, and Jenkins. A number of selection criterion have been chosen to filter the dataset down to a more homogeneus collection of mature software projects. In the following, we describe in detail the criteria used to select mature software projects which use JIRA, Git, and Jenkins consistently.

We aim to evaluate the effects of gamification on teams that work on active and mature software development projects. To that aim, we start our filtering process by removing projects that are inactive, immature, or are personal projects. Active and mature projects are selected based on the criteria that they have regular activity in JIRA, Git, and Jenkins during 2018 and 2019 (the period of study), have had releases during these years, and are developed by a team of developers. We also exclude monorepos and repositories composed of configuration files as the activities of the badges do not apply to these projects.

After our selection process, we identify 333 projects that are candidates for our study. Projects in our curated dataset have sufficiently long development time (~ 5 years), and are developed by large teams (~ 30 collaborators).

5 RESULTS

5.1 RQ1. Is gamification effective as a means to promote the adoption of new practices?

Motivation: A series of badges were designed and presented to users to encourage the adoption of new DevOps practices. In this RQ, we investigate if these badges have helped promote the adoption of related DevOps practices and which badges had successful outcomes aiming to reach this goal. While gamification has shown to be effective in many contexts [19, 20, 29], we have yet to see its effectiveness on large software development companies. Answering

Table 2: The delivery, quality and throughput metrics considered in the study.

Category	Metric Name	Description	Rationale
Delivery	Change Lead Time	Time elapsed from introducing a commit to its deployment in production [14]	Quantifies the overhead of additional non-coding related activities
	Cycle Time	Total time a JIRA issue is in an "In Progress" state.	Quantifies the amount of development time spent on a JIRA issue. [32]
	Time to First Commit	Time elapsed from the creation of JIRA issue to the first related commit	Quantifies the waiting period before the issue is first addressed
	Mean Time to Resolution	Time elapsed from the creation of the JIRA issue to its resolution	Quantifies the total time an issue takes to be fully completed
	Average Review Time	The average time a pull request takes to be merged	Quantifies how much time is spend on review and reworking of pull requests.
Quality	Ratio of Bug Fixing Commits	Ratio of commits linked to fixing bug issues in JIRA vs all commits.	Quantifies how much work is targeted at fixing bugs vs delivering new features
	Build Stability	Ratio of successful vs unsuccesful builds in continuous integration, including compilation, automated tests and static analysis.	Indication of the overall health of the project.
Throughput	Normalized Commit Count	Total number of commits normalized by the number of contributing developers	Quantifies the output of a team in terms of commits committed
	Normalized Pull Request Count	Total number of pull requests merged normalized by the number of contributing developers	Quantifies the output of a team in terms of pull requests merged
	Normalized Release Count	Total number of releases normalized by the number of contributing developers	Quantifies the output of a team in terms of releases for the client

this question may shed the light on the benefits and limitations of gamification as a strategy for changing development practices.

Approach: Because each badge is associated with a practice, we evaluate whether gamification has helped accelerate adoption of the gamified practices. To investigate the effectiveness of badges in promoting new practices, we looked at the DevOps practices associated with each badge before and after gamification was implemented. To that aim, we calculate the badge achievement status (whether or not a badge is earned by satisfying its requirement) retroactively for each month of the pre-gamification period. With the monthly badge achievement statuses in both periods, we compare the practice adoption in the pre-gamification period against the gamification period.

We compare the adoption of practices (badges) in both periods using data visualization and statistical modeling. We employ the Regression Discontinuity Design (RDD) [39], an analysis that allow us to determine the longitudinal effects of an event on a time-series. RDD is a quasi-experimental analysis that can be used to assess the discontinuity of a function as a result of an intervention, the gamification in our case. This method looks at the difference in a function's level and slope after an intervention with the assumption that without an intervention, the function would remain with the same level and slope. This method has been used in several previous studies to investigate the longitudinal impact of software engineering processes on software metrics [42, 47, 48].

We use RDD to perform an analysis on the adoption of each badge individually where Y is the total number of projects achieving that specific badge. We specify the following linear regression model to estimate the level and slope in Y before and after gamification:

$$Y = \alpha + \beta \cdot T + \gamma \cdot G + \delta \cdot A + \eta \cdot C + \epsilon_i$$

where T represents **time** in months from the start of the observation period, G is a binary flag indicating the period before **gamification** ($G = 0$) and after gamification began ($G = 1$); and A represents the number of months **after** gamification, coded 0 before gamification and incrementally increasing after gamification began. In the **control** (C), we include the occurrence of Event A (the DevOps Guidelines document described on Section 3), to control for effects caused by initiatives prior to the gamification.

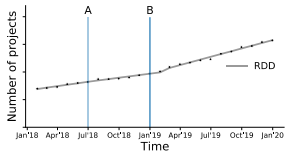
This model is composed by two regressions. Before gamification, the regression line has a $\beta + \eta$ slope, and after gamification the slope changes to $\beta + \eta + \delta$. The change in the regression level is the difference between the two regression values at the gamification starting point, and is given by γ. We are interested in analyzing the change in the level (γ) and in the slope (δ) of badge adoption once gamification is introduced. For this analysis, we consider only those badges that were available at the inception of gamification and were related to practices we could reliably track and extract in both the pre-gamification and gamification periods. Hence, we only conduct the analysis on five of nine badges as shown in column "RQ1" in Table 1.

Results. To investigate which practices are seeing the most adoption, we analyze the adoption of each practice individually using the RDD analysis and the graphics shown in Figure 3. Table 3 presents the adoption of the practices before gamification, the results of our RDD analysis, including the fitness of the model (R^2), the change in the level (γ), and the change in slope (δ) caused by the introduction of gamification. Finally, we also present the improvement in the percent of projects adopting each practice by comparing the adoption level in the last month before gamification against the adoption level one year after the gamification.

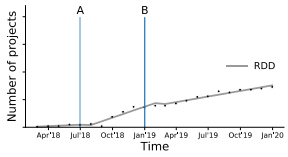
Table 3: Adoption of the five badges achieved before gamification and the increase in adoption after a year of gamification.

Category	Badges	Adoption Before Gamification	RDD Analysis				% Improvement After Gamification (Dec. 2018 vs Dec. 2019)
			R^2	Level γ		Slope δ	
Git	Project uses Trunk Based Development	High	0.84	-11.5 †		1.6 †	>10%
Deployment	Deployments are Automated	Moderate	0.99	3.1 †		2.6 ***	>60%
	Post-Deployment Verification is Automated	Low	0.98	-2.2 †		2.6 ***	>650%
	Project has Automated Deployment CI Job	Low	0.98	-5 †		-3.2 ***	>95%
Testing	Automated Tests are Run on Builds	Moderate	0.99	25.6 ***		5.2 ***	>75%

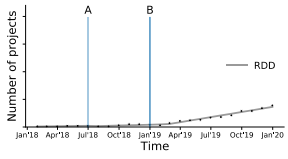
$^{\dagger}p > 0.05$, $^{***}p < 0.001$
Low = adoption lower than than 20%, Moderate = adoption between 20 and 60%, High = adoption higher than 60%



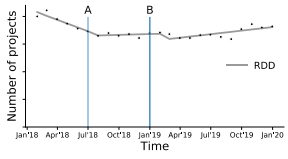
(a) Deployments are automated.



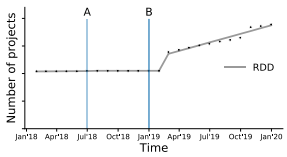
(b) Project has deploy CI job.



(c) Deployment Verification is Automated.



(d) Project uses trunk based development.



(e) Automated Tests are Run on Builds.

Figure 3: Evolution of DevOps practice adoption throughout the years of 2018 and 2019. Event A refers to the release of the DevOps Guidelines and event B shows where Gamification began. While we anonymize the y-axis values, we kept its proportion across badges to make the plots comparable.

Deployments are Automated (Figure 3a) initially had a moderate level of adoption and saw a significant increase in the slope of adoption (δ = 2.6) which corresponds to a 2x increase in the rate of adoption. *Project has Automated Deployment CI Job* (Figure 3b) on the other hand did not experience a significant discontinuity due to gamification. This badge instead saw a short term steep increase in slope after the release of the DevOps guidelines, plateauing right before the gamification period began. While there was a slight increase of slope after gamification, it was not as large in magnitude compared to the effect of the DevOps guidelines. *Post-Deployment Verification is Automated* (Figure 3c) saw the largest increase in adoption level (>650%) and displayed a 6x increase in the rate of adoption (δ = 2.6). This practice in particular was newly implemented in the deployment tooling in the year before gamification, and thus had a very low initial adoption level. As seen in the analysis results, this practice benefited greatly from the education power of gamification. *Project uses Trunk Based Development* (Figure 3d) was the only practice with a high initial level of adoption, and also the only practice that did not see a statistically significant change in level or slope with gamification. Finally, *Automated Tests are Run on Builds* (Figure 3e) saw significant growth in adoption with an increase of >75% from December 2018 to December 2019. While the other practices mainly saw a small rise in level post gamification and a significant increase in slope, this practice saw a substantial increase in level after gamification. This could be due to the fact that this practice is fairly well understood already and is an accessible starting point on the journey to adopting better practices.

> We observed an accelerated adoption of gamified practices related to Testing and Deployment, with increases in adoption rates from 60% to 650%. Gamification showed no significant influence on Git practices, widely adopted before gamification.

5.2 RQ2. How does gamification impact the metrics of projects due to earning badges?

Motivation: The DevOps badges were introduced with the primary aim of promoting the adoption of new DevOps practices and improving the overall software development process. While we observed in RQ1 the acceleration of the adoption of several gamified practices, in this RQ we examine if the adoption of these practices is associated with measurable changes of delivery, quality, and throughput metrics on the teams which adopt them.

Approach: To examine whether there is an association between projects that earn badges and significant metric changes, we compare project metrics before and after badges are earned. Some badges, however, are complementary to each other as shown in the categories of badges of Table 1. For example, *Deployments are Automated* and *Post-Deployment Verification is Automated* are both concerned with the deployment automation process. Hence, it stands to reason that both badges are complementary in promoting a change in the deployment practices which may influence evaluated metrics.

To address potentially confounding effects from closely related badges, we evaluate the observed effect of earning all badges within

Table 4: Relationship between earning badges and metrics. We only present the 7 out of 60 evaluated combinations (badge category x metrics) which showed statistical significant differences.

Category	Metric	Cliffs Delta	Proj.
Deployment	**Normalized Pull Request Count**	-0.400 * M	10
Quality Tooling	**Cycle Time**	-0.322 * S	17
Review	**Change Lead Time**	-0.357 ** M	19
Testing	**Mean Time to Resolution**	0.385 * M	27
	Ratio of Bug Fixing Commits	0.384 ** M	27
	Normalized Commit Count	-0.276 ** S	27
	Normalized Pull Request Count	-0.267 ** S	27

$^{**}p < 0.01$, $^{*}p < 0.05$ on Wilcoxon Signed Rank test.

a category on each of the selected metrics. For each badge category, we find the projects that have earned all of the badges in that category during the same month (e.g., all deployment badges). For each project in this subset, we calculate the mean value of each of our selected metrics over the last six months of the pre-gamification period (July - December 2018), and the first six months after that project earned the badges in the category under study (specific for each project). The result of this process is two distributions, one containing the mean values of a metric pre-gamification and one containing the mean values of a metric post-achievement. These two distributions are then tested for significant changes in each of our selected metrics using the Wilcoxon Signed-Rank Test [45]. To quantify the effect size of statistically significant changes, we resort to the Cliff's Delta effect size [4] and use Romano et al [34] guide for interpreting the effect size, similarly to previous works [5, 44].

Results: We evaluate how badges from 6 different categories (deployment, git, quality tooling, review, stability, testing) affect the 10 metrics related to the aspects of delivery, quality and throughput. Hence, we evaluate 60 combinations total (6 badge categories x 10 metrics). After evaluating each of these combinations, seven of these combinations showed a statistically significant change in the mean value after earning the associated badges. Table 4 summarizes the associated impact of the various badge categories on metrics for the cases where significant change has been observed. Cases where no significant change has been observed are omitted from this table for the sake of brevity.

The results of this analysis show badges related to review, quality tooling, deployment, and testing were overall associated with a small to moderate effect on the selected metrics. We have observed both positive effects suggesting that teams that earn badges had an associated improvement in some metrics while also seeing negative effects for other metrics. These results outline a possible tradeoff which are paid when earning the associated badges.

Delivery Metrics: When considering how earning badges are associated with changes in the delivery metrics of a team, we observed that the most impactful badges are the review and testing related badges. Teams that earn review badges had exhibited an improvement in *Change Lead Time* (negative Cliff's delta), indicating that individual commits are reaching production faster after earning the badges. However, teams that earned the testing badges showed a slow down of the overall resolution time of JIRA issues (*Mean Time to Resolution*), evidenced by the positive Cliff's delta of medium magnitude. Secondary to these badges, teams that earned quality tooling badges exhibited a slightly longer *Cycle Time* to finish their JIRA issues (positive Cliff's delta). This suggests that using code quality tooling may be associated with reducing the total development time alotted to a given JIRA issue.

Quality Metrics: Of the metrics studied, only *Ratio of Bug Fixing Commits* showed any significant change after teams acquired any of the studied badges, i.e., we notice no significant change in the *Build Stability* (our complementary quality metric). Teams that earn testing badges had shown a positive change in *Ratio of Bug Fixing Commits*, of medium effect size (positive Cliff's Delta). This suggests that after achieving the testing badges, software teams have observed a larger proportion of commits are linked with bug issues in JIRA compared to before the gamification period.

Throughput Metrics: Overall, the only categories in which we identify a significant change of throughput metrics after teams acquire the badges has seen only negative effects. Teams who earned the testing badges saw a negative effect for both *Normalized Commit Count* and *Normalized Pull Request Count*, suggesting that they are producing fewer commits and pull requests than before gamification. Additionally, projects earning the deployment badges saw a medium sized negative effect on the *Normalized Pull Request Count* metric, indicating that these teams are outputting fewer pull requests after earning the badge.

> We found significant changes in 7 of the 60 metric / badge category combinations. Teams that earned Testing badges showed an increase in the number of bug fixing commits, but output fewer commit and pull requests. Teams that earned Code Review and Quality Tooling badges have exhibited shorter change lead time and cycle time metrics.

5.3 RQ3. How do software developers react to gamification and perceive its impact?

Motivation: Badges are expected to increase the adoption of certain processes and invoke change on a team's key metrics, both which are effects that can be measured directly. However, at its core, badges are designed to invoke change in developer's behavior. Hence, it is important to get quality feedback from developers adopting these practices to understand 1) how they feel about the badges and 2) to get a sense on any unmeasurable outcome the gamification may have in our study case.

Approach: In this RQ, we design and distribute a survey invitation to 600 developers who have contributed to the projects under study and have worked in the company through the inception of gamification on their projects. In order to avoid biased answers and encourage participants to answer truthfully, participants were informed that this was an anonymous survey when invited to participate. We received a total of 45 responses from the invited participants, resulting in a 7.5% response rate, similar to the response rates in other surveys in software engineering research [21].

Our survey is composed of two sections. In the first section, we ask for background information about the respondent such as their role, the amount of experience they have, and the size of their team.

Table 5: Results of biographical survey questions.

Role	#	Experience	#	Team Size	#
Developer	26	< 2 Years	4	1-3 Members	1
Tech Lead	12	2-5 Years	19	4-5 Members	12
Infrastructure Op. Engineer	3	6-10 Years	8	6-10 Members	19
Architect	2	11-15 Years	8	11-15 Members	5
QA	1	16-20 Years	2	16-20 Members	2
Product Owner	1	> 20 Years	4	> 20 Members	6

In the second section, we ask a series of open-ended questions about the participant's perception towards gamification, their motivation for adopting or not adopting badges, and the perceived outcomes on their projects. To detect recurring themes in these responses, the first two authors independently classified them using an open card-sort method [13]. Labels were created while evaluating the responses and new labels were retroactively applied wherever applicable. The annotators then met to discuss their labeling and reach a consensus. This process enabled us to observe which themes are most common across all survey respondents.

Respondent Demographics: Our participants cover a variety of roles in the company, with the majority being developers (26) and tech leads (11). Almost half of the respondents (22) have more than five years of experience in their respective areas, while an additional 19 respondents have 2-5 years of experience. The majority of our respondents are in medium to large sized teams.

*What **motivates** you to achieve DevOps badges?*

The intent of this question is to uncover what motivates the survey respondents to use the badges and adopt their associated practices. The developers surveyed had a wide range of motivations for adopting badges, from the boosted automation of deployment practices to friendly competitive environment.

Reduce manual overhead in their deployment process (18 respondents). The most commonly cited motivating factor for adopting DevOps badges was that developers saw them as a guide to adopting new practices with the hope of reducing manual overhead. In their responses, survey participants detailed that they would like to reduce overhead primarily in the deployment process, but also in their testing processes. Additionally, with the reduction of manual overhead, they also suggest motivation by a reduction in manual error as a result of less manual intervention in these repetitive tasks.

> *"Ease of code development and deployment process. Also, the fact that the deployments can be done with little to no risk. It also takes out any dependency from deployment and development team members" - R14*

Adopt standardized tooling and processes over custom solutions (11 respondents). Eleven respondents specified that they are encouraged to achieve the badges because they make it clear what is the standard tooling to adopt across their projects so that skills learned are reusable and transferrable between projects. This suggests that there is a drive to make these new processes repeatable and more easily supported by adopting tooling and processes which are standardized througout their environment.

Other notable motivations include enjoying a sense of accomplishment from seeing progress and friendly competition with colleagues (5 respondents), and a motivation to earn badges as a means to showcase their achievements to others (2 respondents). Interestingly, only 3 respondents stated their motivation came from a top-down incentive from management, and 2 other participants were motivated to earn badges because they were helpful in justifying the improvement of internal processes to management. This is a particularly encouraging result as it suggests that badges are helpful for encouraging teams to be self-starters and take initiative to make change rather than being asked by their superiors.

> Participants are driven to achieve DevOps badges as they showed a pathway to reducing manual overhead, and standardizing process across teams. Participants also enjoyed seeing accomplishment in adopting practices, and friendly competition with their colleagues.

*Are badges helpful in **adopting** DevOps practices?*

We designed a two-part question to investigate 1) if practitioners found badges helpful for guiding them to try and adopt new practices and 2) an open-ended question to elaborate on why (and why not) badges were deemed helpful.

Overall, 73.3% of the survey respondents answered "yes" when asked whether or not they found badges helpful. When elaborating on why they found badges helpful, we received reponses which apply to the badges in general. These themes are as follows:

Badges are useful for informing developers about better practices (8 respondents). The most popular theme reported is that developers appreciated how badges provide a clear outline of what they should be adopting as best practices and what they should do to adopt them. The educational power of the badges can be very strong. One clear example of this is *Post-Deployment Verification is Automated* . Reviewing Table 3 from RQ1, we can see that after the badge was created, the adoption level grew from very low by a large margin. Furthermore, in their elaboration, 7 respondents explained that the deployment badges were helpful to teach them about automated deployment practices. This feedback from users similarly supports the findings from RQ1 indicating that the associated badges (*Deployments are Automated*, and *Post-Deployment Verification is Automated*) were effective in helping a significant number of teams adopt new practices related to their deployment processes.

> *"The associated posts which describe the badges, why it is a recommended practice and how to achieve it are invaluable tools for teams that are onboarding"* - R39

Badges help improve transparency and communication (7 respondents). While the main intent of badges are to promote the adoption of new practices, survey respondents noted that they are also quite helpful as a dashboard to provide transparency into the status of projects in terms of hygiene of their practices. Having this global view on their project is helpful to determine which practices they should be adopting.

> *"The badges have helped us identify at a repo and more macro levels where we need to invest devops effort."* - R39

Participants also mentioned that badges were helpful for standardizing tooling and processes across teams (5 respondents), and setting concrete targets for improving current DevOps practices (2 respondents).

As for the 26.7% of respondents who answered that they did not find the badges helpful, criticisms which were stated suggest that the badges takes a lot of effort for maintaining a positive appearance to peers, and this may drive the wrong motivations for teams to change their behavior. As a respondent stated:

> *"I'd like to highlight that some may prioritize DevOps achievement in a wrong way which is steering away the focus on the actual KPI - this is a big problem as people are just getting badges for the sake of getting it to show off instead on worrying on the actual outcomes".* - R11

Additionally, some respondents mentioned they had already adopted other practices which were working for them but contradict what the badges promote. This suggests a frustration that their previous efforts may be wasted or not recognised, and they felt a pressure to change their practices:

> *"We had already adopted most of the best practice that the badges are trying to make us adopt. Being forced to try and keep the metrics right is costing us time and forcing us to change our already established practices that were working well"* - R6.

73% of participants find badges helpful explaining that they inform teams about better practices and improve transparency and communication. Approximately 27% of respondents did not find badges helpful, stating it may drive the wrong motivation for teams to change behavior.

*Did you perceive tangible **benefits** of adopting DevOps badges?* The intent of this question is to examine the perceived results by the survey respondents on their projects as a result of adopting the practices associated with the DevOps badges. Of the surveyed participants, 62.2% of respondents noted they observed benefits after earning badges, citing the following reasons:

Reduction of manual overhead in deployment processes and an increase in deployment frequency (13 respondents). The most frequent answer from respondents suggest that automated deployment practices have significantly reduced the complexity, overhead, and stress of deployments and improved quality of life for practitioners, ultimately improving their deployment frequency metrics. Also, there were reports of attitudes towards change management shifting as the badges which promote automated deployment enable more frequent deployments. One respondent reported that their team feels more secure with automation in place and this has changed their outlook on change management.

> *"Smoother deployments, more frequent deployments, easier to release many projects (no difference between releasing 1 project or 20 projects)"* - R16

Improve testing practices and software quality (10 respondents). Aside from gains derived from automating deployments, developers also noted that they observed earning testing badges had positive outcomes on software quality. Specific outcomes quoted include repayment of technical debt, an increase in unit test coverage, and a perceived increase in software quality.

> *"Introduced code quality tooling that helped with test coverage and technical debt. Gave up some bad practices of merging PRs without review."* - R20

Other tangible benefits mentioned by participants were the standardization of tooling and processes (3 respondents) and improving transparency of processes and communication in a team (3 respondents). Participants mention, once again, that the badges have helped convince management to improve internal processes (2 respondents).

From the survey participants, 37.8% reported not identifying tangible benefits from adopting badges. Only one of these participants elaborated on this by stating it was too early for them to tell whether or not there are any tangible benefits. Participants also provided other valuable feedback. One respondent mentioned that changing their practices negatively impacted their productivity because their current practices were already working well for their team. Similarly, respondents suggested that changing behaviors made their developers nervous about doing things which would cause them to lose a badge, an unintended consequence of gamification. For example, given the badge Unit Tests are Fast , which requires tests to run in less than 5 minutes, a participant stated:

> *"Some tests take time to run, how do we make sure we run all the tests and [the] metric does not get affected?"* - P13

The majority of participants (62%) reported perceived tangible benefits of adopting DevOps badges. Gamified DevOps practices have reduced manual overhead in deployment and improved software quality and test practices. Still, 38% report not identifying tangible benefits, with some complains of lower productivity and unintended consequences.

6 DISCUSSION

In this section, we discuss four overarching themes that emerge from the findings of our three RQs, which serve as implications to practitioners and researchers on the effectiveness of gamification.

Deployment and testing practices are good candidates for effective gamification. The results of our study indicate that deployment and test practices exhibited the best outcome of the gamified practices in the company under study. Of the badges we evaluated in this study, testing and deployment badges have shown to yield the highest growth in adoption following the implementation of gamification (RQ1). Teams that earn testing badges are associated with an increase in the number of bug fixing commits (RQ2). Related studies have also reported that gamified testing has yielded improvements in defect registration [9]. Finally, practitioners frequently cite the reduction of manual overhead in deployment processes as the main motivation for using the badges (RQ3), and report perceived improvement in software quality and testing practics (RQ3). Our findings are also corroborated by related work, which cites Product Integration (Deployment) and Verification and Validation (Testing) as most frequently cited areas in which gamification exhibited a positive outcome [9].

Not all badges show an association with project metrics change. When considering how metrics change with the implementation of gamification, in RQ2 we saw a small fraction of badge category / metric combinations showing any significant change after teams aquired the relevant badges. Not all of these observed associations are positive. While an association between adopting the testing practices and a higher bug fixing commit ratio was seen, testing badges were also associated with a reduction in throughput metrics. As such, when encouraging new practices, there may be trade-offs between the KPIs associated to the practices adopted by a team. It is important to note that analysing changes on a large heterogeneous set of projects is a complex task, and confounding factors could interplay. Additionally, these associations observed do not suggest causation.

Benefits of badges are not easily measurable. Interestingly, when comparing the results from RQ2 with the survey responses in RQ3, we observed a contrast between developer perception and the measured metrics. While many participants have mentioned the deployment badges to be a game changer, we did not observe any positive outcomes in the evaluated metrics. Whether or not the badges produce concrete change in the evaluated metrics, they may impact the perception of developers on their processes and improve their quality of life in their work. In the future, more studies should be conducted to establish and/or confirm a link between the practitioners' perception and the result in their KPIs.

Gamification systems need to be carefully designed. Although the survey participants had a lot of positive feedback about gamification, there were also a number of critics. One survey respondent expressed that they fear gamification can potentially drive the wrong motivations for change. Gamification may drive some developers to adopt the badges solely to check off all of the boxes and show off without being mindful of the underlying KPIs which are meant to be optimized by the badges. Another respondant also expressed fear that developers will waste too much time to maintain their badges, even if they are not actually deriving any real benefit, simply for the sake of vanity. In their study, Porto et al [9] also noticed the same problem in four of the studies they reviewed [7, 8, 22, 36], it is difficult to manage motivations and get users to focus on the right things.

7 THREATS TO VALIDITY

In this section, we discuss the threats to the validity of our findings, broken down by internal, construct, and external validity.

Internal Validity. Threats to internal validity are related to experimenter bias and errors. First, analysing data from a large set of projects from a real world enterprise in a heterogenous environment was very challenging and errors in this process could affect our results. We mitigate this risk by including only the badges related to practices we could reliably track and extract from studied projects, leading us to remove four badges in our analysis in RQ1. Second, many factors could influence software developers to adopt DevOps practices, other than gamification, such as seeing examples of positive outcomes from other teams and companies. For this reason, we detailed event A in Figure 1 to represent the communication of the DevOps Guidelines document. In RQ1, this is factored in as a control variable to observe how strongly it impacts our results. Similarly, in RQ2, practices which target the same KPI (ie. deployment related practices) may have confounding associated effects. In order to address this, we focused on the effects of groups of practices and observed how the targeted KPIs change. Even with our mitigation, we were careful to describe the metrics change as an association (not causation) with gamification, as there could be many other reasons explaining the change of a KPI metric. Finally, in RQ3, surveys can be subject to human error and bias. There is a possibility that some survey respondants may be overly positive about their experience with gamification to make the gamification designers feel good about their work. We mitigate this risk by submitting our survey to a large sample group from different areas of the company in attempt to get a full viewpoint of how individuals in different working situations view gamification.

Construct Validity. Our study uses a number of metrics to help assess the changes projects go through after practices are adopted and badges are earned. Some of these metrics, however, attempt to measure constructs of a software project which are not easy to measure (e.g., software quality). For instance, the quality metric *Ratio of Bug Fixing Commits* can be viewed in two contradictory manners. Having a high ratio of bug fixing commits can be viewed as a project having a lot of bugs, but it can also be viewed as a team being very active in improving the quality of their system. For this reason, the power of this metric in isolation is relatively low and it is best used in combination with other metrics to help better explain the state of a project. As for the throughput metrics, these metrics in isolation do not give the full picture of productivity as teams can have a variety of habits delivering functionality. More frequent releases could be more desirable, however, it is not safe to generalize that a team with this practice is more productive than a team which does larger, less frequent releases. The release size metric could help complement our analysis, but the data was unavailable for this study.

External Validity. This study took place in a large company with a distinct software development culture and approach. Other companies may not operate in the same way, and therefore the findings of this study may not be generalizable to all companies.

8 CONCLUSION

In this paper, we conducted a mixed-methods study on the effects of badge-based gamification at a large company. We investigated how badges can accelerate the adoption of new practices and their associations with a set of key delivery, quality, and throughput metrics. We also conducted a survey with practitioners to understand how developers react to gamification and perceive its impact. Our findings showed that gamification can be effective in promoting the adoption of new practices, with practice adoption increasing at least 60% in most practices. Teams that earned badges related to code review and code quality tooling saw a small to moderate reduction in their cycle time and change lead time metrics. Additionally, teams which earned testing badges saw an increase in their bug fixing commits but output fewer commits and pull requests. Finally, 74% of these survey participants found badges to be useful for learning new practices and were motivated by badges which demonstrate the prospect of reducing manual overhead and standardizing processes across teams and projects.

REFERENCES

[1] Manal M. Alhammad and Ana M. Moreno. 2018. Gamification in software engineering education: A systematic mapping. *Journal of Systems and Software* 141 (2018), 131–150. https://doi.org/10.1016/j.jss.2018.03.065

[2] Ashton Anderson, Daniel Huttenlocher, Jon Kleinberg, and Jure Leskovec. 2013. Steering User Behavior with Badges. In *Proceedings of the 22nd International Conference on World Wide Web* (Rio de Janeiro, Brazil) *(WWW '13)*. Association for Computing Machinery, New York, NY, USA, 95–106. https://doi.org/10.1145/2488388.2488398

[3] Orges Cico, Letizia Jaccheri, Anh Nguyen-Duc, and He Zhang. 2021. Exploring the intersection between software industry and Software Engineering education - A systematic mapping of Software Engineering Trends. *Journal of Systems and Software* 172 (2021), 110736. https://doi.org/10.1016/j.jss.2020.110736

[4] Norman Cliff. 1993. Dominance Statistics: Ordinal Analyses to Answer Ordinal Questions. *Psychological Bulletin* 114 (11 1993), 494–509. https://doi.org/10.1037/0033-2909.114.3.494

[5] Diego Costa, Cor-Paul Bezemer, Philipp Leitner, and Artur Andrzejak. 2021. What's Wrong with My Benchmark Results? Studying Bad Practices in JMH Benchmarks. *IEEE Transactions on Software Engineering* 47, 7 (2021), 1452–1467. https://doi.org/10.1109/TSE.2019.2925345

[6] Carlos Cruz, Michael D. Hanus, and Jesse Fox. 2017. The need to achieve: Players' perceptions and uses of extrinsic meta-game reward systems for video game consoles. *Computers in Human Behavior* 71 (2017), 516–524. https://doi.org/10.1016/j.chb.2015.08.017

[7] Fabiano Dalpiaz, Remco Snijders, Sjaak Brinkkemper, Mahmood Hosseini, Alimohammad Shahri, and Raian Ali. 2017. *Engaging the Crowd of Stakeholders in Requirements Engineering via Gamification.* Springer International Publishing, Cham, 123–135. https://doi.org/10.1007/978-3-319-45557-0_9

[8] Alexandre Altair de Melo, Mauro Hinz, Glaucio Scheibel, Carla Diacui Medeiros Berkenbrock, Isabela Gasparini, and Fabiano Baldo. 2014. Version Control System Gamification: A Proposal to Encourage the Engagement of Developers to Collaborate in Software Projects. In *Social Computing and Social Media*, Gabriele Meiselwitz (Ed.). Springer International Publishing, Cham, 550–558.

[9] Daniel de Paula Porto, Gabriela Martins de Jesus, Fabiano Cutigi Ferrari, and Sandra Camargo Pinto Ferraz Fabbri. 2021. Initiatives and challenges of using gamification in software engineering: A Systematic Mapping. *Journal of Systems and Software* 173 (2021), 110870. https://doi.org/10.1016/j.jss.2020.110870

[10] Sebastian Deterding, Dan Dixon, Rilla Khaled, and Lennart Nacke. 2011. From Game Design Elements to Gamefulness: Defining "Gamification". In *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments* (Tampere, Finland) *(MindTrek '11)*. Association for Computing Machinery, New York, NY, USA, 9–15. https://doi.org/10.1145/2181037.2181040

[11] Darina Dicheva, Christo Dichev, Gennady Agre, and Galia Angelova. 2015. Gamification in Education: A Systematic Mapping Study. *Educational Technology & Society* 18 (07 2015), 75–88.

[12] Daniel J. Dubois and Giordano Tamburrelli. 2013. Understanding Gamification Mechanisms for Software Development. In *Proceedings of the 2013 9th Joint Meeting on Foundations of Software Engineering* (Saint Petersburg, Russia) *(ESEC/FSE 2013)*. Association for Computing Machinery, New York, NY, USA, 659–662. https://doi.org/10.1145/2491411.2494589

[13] Sally Fincher and Josh Tenenberg. 2005. Making sense of card sorting data. *Expert Systems* 22 (07 2005), 89–93. https://doi.org/10.1111/j.1468-0394.2005.00299.x

[14] Nicole Forsgren, Dustin Smith, Jez Humble, and Jessie Frazelle. 2019. *2019 Accelerate State of DevOps Report.* Technical Report. 15–16 pages. http://cloud.google.com/devops/state-of-devops/

[15] Matthieu Foucault, Xavier Blanc, Jean-Rémy Falleri, and Margaret-Anne Storey. 2019. Fostering good coding practices through individual feedback and gamification: an industrial case study. *Empirical Software Engineering* 24 (12 2019). https://doi.org/10.1007/s10664-019-09719-4

[16] Félix García, Oscar Pedreira, Mario Piattini, Ana Cerdeira-Pena, and Miguel Penabad. 2017. A framework for gamification in software engineering. *Journal of Systems and Software* 132 (2017), 21–40. https://doi.org/10.1016/j.jss.2017.06.021

[17] Gabriel Alberto García-Mireles and Miguel Ehécatl Morales-Trujillo. 2020. Gamification in Software Engineering: A Tertiary Study. In *Trends and Applications in Software Engineering*, Jezreel Mejia, Mirna Muñoz, Álvaro Rocha, and Jose A. Calvo-Manzano (Eds.). Springer International Publishing, Cham, 116–128.

[18] Juho Hamari. 2017. Do badges increase user activity? A field experiment on the effects of gamification. *Computers in Human Behavior* 71 (2017), 469–478. https://doi.org/10.1016/j.chb.2015.03.036

[19] J. Hamari, J. Koivisto, and H. Sarsa. 2014. Does Gamification Work? – A Literature Review of Empirical Studies on Gamification. In *2014 47th Hawaii International Conference on System Sciences*. 3025–3034.

[20] Michael D. Hanus and Jesse Fox. 2015. Assessing the effects of gamification in the classroom: A longitudinal study on intrinsic motivation, social comparison, satisfaction, effort, and academic performance. *Computers & Education* 80 (2015), 152–161. https://doi.org/10.1016/j.compedu.2014.08.019

[21] Juan Hoyos, Rabe Abdelkareem, Suhaib Mujahid, Emad Shihab, and Albeiro Espinosa Bedoya. 2021. On the Removal of Feature Toggles. *Empirical Software Engineering (EMSE)* 26 (2021), 1–26. Issue 2.

[22] Marcus Johansson and Erik Ivarsson. 2014. *An experiment on the effectiveness of unit testing when introducing gamification.* Master's thesis.

[23] Daniel Johnson, Sebastian Deterding, Kerri-Ann Kuhn, Aleksandra Staneva, Stoyan Stoyanov, and Leanne Hides. 2016. Gamification for health and wellbeing: A systematic review of the literature. *Internet Interventions* 6 (2016), 89–106. https://doi.org/10.1016/j.invent.2016.10.002

[24] Shivam Khandelwal, Sai Krishna Sripada, and Y. Raghu Reddy. 2017. Impact of Gamification on Code Review Process: An Experimental Study. In *Proceedings of the 10th Innovations in Software Engineering Conference* (Jaipur, India) *(ISEC '17)*. Association for Computing Machinery, New York, NY, USA, 122–126. https://doi.org/10.1145/3021460.3021474

[25] Jonna Koivisto and Juho Hamari. 2014. Demographic differences in perceived benefits from gamification. *Computers in Human Behavior* 35 (2014), 179–188. https://doi.org/10.1016/j.chb.2014.03.007

[26] Michael Meder, Till Plumbaum, Aleksander Raczkowski, Brijnesh Jain, and Sahin Albayrak. 2018. Gamification in E-Commerce: Tangible vs. Intangible Rewards. In *Proceedings of the 22nd International Academic Mindtrek Conference* (Tampere, Finland) *(Mindtrek '18)*. Association for Computing Machinery, New York, NY, USA, 11–19. https://doi.org/10.1145/3275116.3275126

[27] Elisa D. Mekler, Florian Brühlmann, Alexandre N. Tuch, and Klaus Opwis. 2017. Towards understanding the effects of individual gamification elements on intrinsic motivation and performance. *Computers in Human Behavior* 71 (2017), 525–534. https://doi.org/10.1016/j.chb.2015.08.048

[28] L. Moldon, M. Strohmaier, and J. Wachs. 2021. How Gamification Affects Software Developers: Cautionary Evidence from a Natural Experiment on GitHub. In *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*. IEEE Computer Society, Los Alamitos, CA, USA, 549–561. https://doi.org/10.1109/ICSE43902.2021.00058

[29] Lennart Nacke and Sebastian Deterding. 2017. The maturing of gamification research. *Computers in Human Behavior* 71 (01 2017), 450–454. https://doi.org/10.1016/j.chb.2016.11.062

[30] Pedro Santos Neto, Danilo Batista Medeiros, Irvayne Ibiapina, and Otávio Cury da Costa Castro. 2019. Case study of the introduction of game design techniques in software development. *IET Software* 13, 2 (2019), 129–143. https://doi.org/10.1049/iet-sen.2018.5149 arXiv:https://ietresearch.onlinelibrary.wiley.com/doi/pdf/10.1049/iet-sen.2018.5149

[31] Oscar Pedreira, Felix Garcia, Nieves Brisaboa, and Mario Piattini. 2014. Gamification in software engineering – A systematic mapping. *Information and Software Technology* 57 (01 2014). https://doi.org/10.1016/j.infsof.2014.08.007

[32] K. Power and K. Conboy. 2015. A Metric-Based Approach to Managing Architecture-Related Impediments in Product Development Flow: An Industry Case Study from Cisco. In *2015 IEEE/ACM 2nd International Workshop on Software Architecture and Metrics*. 15–21.

[33] Christian R. Prause and Matthias Jarke. 2015. Gamification for Enforcing Coding Conventions. In *Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering* (Bergamo, Italy) *(ESEC/FSE 2015)*. Association for Computing Machinery, New York, NY, USA, 649–660. https://doi.org/10.1145/2786805.2786806

[34] Jeanine Romano, Jeffrey D Kromrey, Jesse Coraggio, Jeff Skowronek, and Linda Devine. 2006. Exploring methods for evaluating group differences on the NSSE and other surveys: Are the t-test and Cohen's d indices the most appropriate choices. In *Annual meeting of the Southern Association for Institutional Research.*

[35] Michael Sailer, Jan Ulrich Hense, Sarah Katharina Mayr, and Heinz Mandl. 2017. How gamification motivates: An experimental study of the effects of specific game design elements on psychological need satisfaction. *Computers in Human Behavior* 69 (2017), 371–380. https://doi.org/10.1016/j.chb.2016.12.033

[36] Simon André Scherr, Frank Elberzhager, and Konstantin Holl. 2018. Acceptance Testing of Mobile Applications - Automated Emotion Tracking for Large User Groups. In *2018 IEEE/ACM 5th International Conference on Mobile Software Engineering and Systems (MOBILESoft)*. 247–251.

[37] Katie Seaborn and Deborah I. Fels. 2015. Gamification in theory and action: A survey. *International Journal of Human-Computer Studies* 74 (2015), 14–31. https://doi.org/10.1016/j.ijhcs.2014.09.006

[38] Leif Singer and Kurt Schneider. 2012. It was a bit of a race: Gamification of version control. In *2012 Second International Workshop on Games and Software Engineering: Realizing User Engagement with Game Engineering Techniques (GAS)*. 5–8. https://doi.org/10.1109/GAS.2012.6225927

[39] D. Thistlethwaite and D. Campbell. 1960. Regression-discontinuity analysis: An alternative to the ex post facto experiment. *Journal of Educational Psychology* 51 (1960), 309–317.

[40] M. Zulfahmi Toh, Shamsul Sahibuddin, and Mohd Naz'ri Mahrin. 2019. Adoption Issues in DevOps from the Perspective of Continuous Delivery Pipeline. In *Proceedings of the 2019 8th International Conference on Software and Computer Applications* (Penang, Malaysia) *(ICSCA '19)*. Association for Computing Machinery,

New York, NY, USA, 173–177. https://doi.org/10.1145/3316615.3316619

[41] Asher Trockman, Shurui Zhou, Christian Kästner, and Bogdan Vasilescu. 2018. Adding Sparkle to Social Coding: An Empirical Study of Repository Badges in the npm Ecosystem. In *2018 IEEE/ACM 40th International Conference on Software Engineering (ICSE)*. 511–522. https://doi.org/10.1145/3180155.3180209

[42] A. Trockman, S. Zhou, C. Kästner, and B. Vasilescu. 2018. Adding Sparkle to Social Coding: An Empirical Study of Repository Badges in the npm Ecosystem. In *2018 IEEE/ACM 40th International Conference on Software Engineering (ICSE)*. 511–522.

[43] Bogdan Vasilescu. 2014. Human Aspects, Gamification, and Social Media in Collaborative Software Engineering. In *Companion Proceedings of the 36th International Conference on Software Engineering* (Hyderabad, India) *(ICSE Companion 2014)*. Association for Computing Machinery, New York, NY, USA, 646–649. https://doi.org/10.1145/2591062.2591091

[44] Mairieli Wessel, Bruno Mendes de Souza, Igor Steinmacher, Igor S. Wiese, Ivanilton Polato, Ana Paula Chaves, and Marco A. Gerosa. 2018. The Power of Bots: Characterizing and Understanding Bots in OSS Projects. *Proc. ACM Hum.-Comput. Interact.* 2, CSCW, Article 182 (Nov. 2018), 19 pages. https://doi.org/10.1145/3274451

[45] Frank Wilcoxon. 1945. Individual Comparisons by Ranking Methods. *Biometrics Bulletin* 1, 6 (1945), 80–83. http://www.jstor.org/stable/3001968

[46] Toshihiko Yamakami. 2013. Gamification Literacy: Emerging Needs for Identifying Bad Gamification. In *Multimedia and Ubiquitous Engineering*, James J. (Jong Hyuk) Park, Joseph Kee-Yin Ng, Hwa-Young Jeong, and Borgy Waluyo (Eds.). Springer Netherlands, Dordrecht, 395–403.

[47] Yangyang Zhao, Alexander Serebrenik, Yuming Zhou, Vladimir Filkov, and Bogdan Vasilescu. 2017. The Impact of Continuous Integration on Other Software Development Practices: A Large-Scale Empirical Study. In *Proceedings of the 32nd IEEE/ACM International Conference on Automated Software Engineering* (Urbana-Champaign, IL, USA) *(ASE 2017)*. IEEE Press, 60–71.

[48] T. Zimmermann and A. Casanueva Artís. 2019. Impact of Switching Bug Trackers: A Case Study on a Medium-Sized Open Source Project. In *2019 IEEE International Conference on Software Maintenance and Evolution (ICSME)*. 13–23.