A Case Study on Gamifying DevOps Practices in Industry

Patrick Ayoup

A Thesis

in

The Department

of

Computer Science and Software Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of

Master of Applied Science (Software Engineering) at

Concordia University

Montréal, Québec, Canada

October 2021

© **Patrick Ayoup, 2021**

Concordia University

School of Graduate Studies

This is to certify that the thesis prepared

By: **Patrick Ayoup**

Entitled: **A Case Study on Gamifying DevOps Practices in Industry**

and submitted in partial fulfillment of the requirements for the degree of

Master of Applied Science (Software Engineering)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the Final Examining Committee:

_____ Chair
Dr. Nikolaos Tsantalis

_____ Examiner
Dr. Nikolaos Tsantalis

_____ Examiner
Dr. Jinqiu Yang

_____ Supervisor
Dr. Emad Shihab

Approved by _____
Lata Narayanan, Chair
Department of Computer Science and Software Engineering

_____ 2021

_____
Mourad Debbabi, Dean
Faculty of Engineering and Computer Science

Abstract

A Case Study on Gamifying DevOps Practices in Industry

Patrick Ayoup

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 teams of developers in the industry.

In this thesis, 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. Our results indicate that gamification accelerated the adoption of some 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.

Related Submitted Publications

The following submitted publications are related to this thesis.

- **Patrick Ayoup**, Diego Elias Costa, and Emad Shihab. "Achievement Unlocked: A Case Study on Gamifying DevOps Practices in Industry". Under Submission to 2022 IEEE/ACM International Conference on Software Engineering (ICSE)

Statement of Originality

I, Patrick Ayoup, hereby declare that I am the sole author of this thesis. All ideas and inventions attributed to others have been properly referenced. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners. I understand that my thesis may be made electronically available to the public.

Dedication

To my parents and brothers

Acknowledgments

First, I would like to thank my parents, brothers, and the rest of my family for their continued inspiration, support, and encouragment in all of my endeavors. Your support is invaluable for helping me reach my goals. None of this would be possible without you.

I express my gratitude to my supervisor Dr. Emad Shihab, and mentor Dr. Diego Costa who shared their experience and knowledge with me and patiently guided me through this challenge from beginning to end. I thank you sincerely for your time and efforts, and for giving me this opportunity to complete this research together.

I would also like to thank my colleagues in the DAS lab for working alongside me while completing this thesis; Rabe, Suhaib, Mohamed, Ahmad, Mahmoud, Abbas, Hassan, Jasmine, Xiaowei, Khaled, Nicholas, Mahsa, Riya, and Farbod.

A generous thank you goes to my colleagues whose support and efforts made this project a success; Bruno, Amber, Abhi, Xintong, Wensheng, Javad, Bill, Dilip, Kamal, and Sofiane. You all made a huge difference in undertaking this project. Thank you!

Finally, I must thank my closest friends who supported me throughout this process from day one through to completion; Alberto and Isabela, who were with me and encouraged me every step of the way, and Jonathan and Allison, who continue to encourage and inspire me with your friendship, advice and incredible dedication to your craft.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The tools, processes, and best practices in software development are constantly evolving as different trends emerge (Cico, Jaccheri, Nguyen-Duc, & Zhang, 2021). Although adopting new practices can be very attractive, provoking meaningful change and standardizing a heterogeneous environment at scale is a difficult task (Toh, Sahibuddin, & Mahrin, 2019). 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 and interesting solution to this problem is to incorporate gamification (Foucault, Blanc, Falleri, & Storey, 2019). Gamification is the inclusion of game elements such as points, badges, leaderboards, levels, and quests in non-game contexts to motivate user activity and improve engagement (Deterding, Dixon, Khaled, & Nacke, 2011). Gamification has been reported to show positive results (Foucault et al., 2019; Prause & Jarke, 2015; Singer & Schneider, 2012), particularly when employed to promote the adoption of new tools and practices in software development (Dubois & Tamburrelli, 2013; Singer & Schneider, 2012). However, if not carefully designed, gamification can also steer developers to unwanted directions (Moldon, Strohmaier, & Wachs, 2021). Although these studies showed promising results, those which performed case studies did so either with students (Dubois & Tamburrelli, 2013; Khandelwal, Sripada, & Reddy, 2017), or a small group of professionals in industry (Foucault et al., 2019; Garca-Mireles & Morales-Trujillo, 2020; Neto, Medeiros, Ibiapina, & da Costa Castro, 2019). There is still a gap and lack of understanding in

industry about the effectiveness of gamifiifaction.

This thesis presents a work that performs a large-scale study of gamification and its impact in a real-world industrial environment. Specifically, we investigate the use of gamification over the span of 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). Each badge targets a specific DevOps practice, and has a target in a related measure which must be met in order to achieve it.

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.

This thesis 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. Practices that had low adoption prior to gamification saw a considerable boost once gamification began, while previously established practices were less influenced by gamification. 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.

1.1 Thesis Overview

Chapter 2 presents the primary related works on Gamification. Chapter 3 sets up the background of the company where this case study took place, and details the gamification techniques which were observed. With the background of this study set in place, the core of this thesis are then spread out across three chapters:

- Chapter 4: The primary reason for implementing a gamification system is to educate and encourage software developers to adopt new practices and procedures. In this chapter, we ask the following research question: **"Is gamification effective as a means to promote the adoption of new practices?"**
- Chapter 5: The gamified practices are each designed to target specific KPIs and help software development teams improve on their metrics. We continue our study by asking **"How does gamification impact the metrics of projects due to earning badges?"**
- Chapter 6: While implementing gamification may have material outcomes in terms of driving adoption and improving software KPIs, it may also have unmeasurable outcomes on the day to day life of the software developers subjected to such a system. In this chapter, we ask **"How do software developers react to gamification and perceive its impact?"** To answer this question, we performed a survey with 75 developers to better understand how they perceive gamification and its impact on their work.

1.2 Thesis Contributions

This thesis 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 on gamification.

Chapter 2

Related Works

In this chapter, we describe the fundamentals of gamification and dive in the related works that investigate gamification in Software Engineering (SE).

2.1 Gamification and Motivation

In its simplest definition, gamification is the application of game elements and characteristics in a non-game environment (Deterding et al., 2011). Gamification can manifest itself in many forms by applying game elements such as points, badges, levels, quests, and leaderboards to support user engagement and enhance positive patterns (Hamari, Koivisto, & Sarsa, 2014; Pedreira, Garcia, Brisaboa, & Piattini, 2014). Each game element has the potential to affect user behavior differently (Mekler, Brhlmann, Tuch, & Opwis, 2017). For instance, leaderboards emphasize relative performance and may drive users' competitiveness (Mekler et al., 2017), while badges, give the user a sense of self-improvement, and have shown to steer users' long-term behavior towards gamified goals (Hamari, 2017).

Several studies have investigated the effects of gamification in a variety of domains (Seaborn & Fels, 2015). From education (Dicheva, Dichev, Agre, & Angelova, 2015) and health (D. Johnson et al., 2016), to marketing and commerce (Meder, Plumbaum, Raczkowski, Jain, & Albayrak, 2018), meta-studies have shown benefits of gamification on user engagement and satisfaction (Dicheva et al., 2015; Hamari et al., 2014; D. Johnson et al., 2016; Seaborn

& Fels, 2015). 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 (Koivisto & Hamari, 2014), users' receptivity to external rewards (Mekler et al., 2017), and the meaning assigned to gamified elements (Cruz, Hanus, & Fox, 2017). Finally, gamification's effectiveness is deeply connected to the design of game elements, and how they interact with the user (Mekler et al., 2017; Sailer, Hense, Mayr, & Mandl, 2017). A badly designed gamification system can sap users' motivation (Hanus & Fox, 2015; Moldon et al., 2021; Yamakami, 2013) and steer users to chase metrics instead of encourage behavioral change (Mekler et al., 2017).

As a result, systematic studies unanimously state that more rigorous studies are needed to better understand gamification benefits and limitations (Dicheva et al., 2015; Hamari et al., 2014; Nacke & Deterding, 2017). Particularly, researchers urge for large-scale studies that assess gamification effectivity on the long-term in the wild, as opposed to a lab environment (Hamari et al., 2014; Nacke & Deterding, 2017). A common criticism of the state of gamification research is that in most studies that include experimentation, sample sizes were very small, focused exclusively on descriptive statistics, and comprised of tests which ran over a short time period (de Paula Porto, de Jesus, Ferrari, & Fabbri, 2021; Hamari et al., 2014). This thesis contributes to the literature by assessing gamification effectiveness in encouraging practitioners to adopt DevOps practices in a large software development company, across 333 projects over a period of one year.

2.2 Gamification in Software Engineering

Recently, gamification has also been explored as an effective measure to encourage behaviors in software engineering education and software development. Major code-centric social platforms such as Stack Overflow use badges to evaluate users' commitment, competence and trustworthiness on the platform (Anderson, Huttenlocher, Kleinberg, & Leskovec,

2013). 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 (Trockman, Zhou, Kstner, & Vasilescu, 2018). Given its prominence, the effects of gamification has been studied in Software Engineering education (Alhammad & Moreno, 2018), and in open-source and industrial software development (de Paula Porto et al., 2021). Pedreira et al. (2014) performed a systematic mapping of gamification in software engineering and found that more than half of the papers reviewed were philisophical with no experimental verification.

2.2.1 Software Engineering Education

Most works that study gamification in SE, focused on educational settings (Alhammad & Moreno, 2018; Dubois & Tamburrelli, 2013; Khandelwal et al., 2017; Prause & Jarke, 2015; Singer & Schneider, 2012). Gamification has been evaluated on a variety of modalities in SE education, typically comparing students performance in a gamified versus a non-gamified environment. Alhammad and Moreno (2018) performed a systematic study on 21 papers that study gamification in SE education. They found that gamification has reported mostly positive results in improving students engagement and, to a lesser extent, improving students knowledge. The authors emphasize the need for more studies to draw more conclusive and generalizable remarks for SE education.

Dubois and Tamburrelli (2013) designed and ran an experiment with gamified environments in an undergraduate project course. In this experiement, students were awarded points on a leaderboard for activities such as documenting and testing their code, and adhering to the best practices set forth by SonarQube. Half of the student teams had access to the leaderboard while the other half did not. This study suggests that the teams with access to the leaderboard had better results as they were stimulated by competition.

Singer and Schneider (2012), on the other hand, reported mixed results when employing gamification to encourage students to use control version systems more frequently. The authors performed an experiment with 37 undergraduate students gamifying the use of version control systems. Using a leaderboard with the number of commits, students were

encourage to use version control systems more frequently. Students reported mixed reviews on the gamification system, but feedback showed that it did influence the way the students used the tools.

Prause and Jarke (2015) performed an experiment in a classroom setting using gamification to promote the use of better coding conventions. They implemented reputation scores to encourage students to use better code conventions and reported an improvement in code convention adherence.

Code review has also been gamified in a study by Khandelwal et al. (2017). This study performed an experiment with 183 undergraduate students to study gamified code review environments. 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.

2.2.2 Open Source Software Development

Some works investigated gamification in open source software projects (Moldon et al., 2021; Trockman, Zhou, Kstner, & Vasilescu, 2018; Vasilescu, 2014). Vasilescu (2014) 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. 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. (2018) study showed that badges are mostly reliable as a signal of best practices. However, Moldon et al. (2021) showed that gamification can also steer developers behavior towards unwanted directions. They reported on the significant impact gamification has on the behavior of developers, by studying how developers reacted when GitHub removed the daily activity streak from the platform. The findings show that gamification can steer developers behavior also towards unhealthy and unwanted directions, hence, the gamification system needs to be carefully designed.

2.2.3 Industrial Software Development

A few studies have assessed gamification in industrial settings, most commonly in small and medium sized companies (Foucault et al., 2019; Garca, Pedreira, Piattini, Cerdeira-Pena, & Penabad, 2017; Neto et al., 2019).

Garca et al. (2017) proposed a framework for incorporating gamification into software development tools and performed a case study gamifying project management, requirements management, and software testing at a small company with 19 practitioners. The authors reported seeing a 20% increase in the usage of the requirement and issue tracking tools.

Neto et al. (2019) developed a plugin for Redmine including several gamification elements from role playing games (levels, experience points), badges, and a leaderboard. and evaluated its effectiveness in a case study involving 19 developers from a small company. While many developers felt the gamification had positive effects on their work, the results were inconclusive as to whether or not developer productivity was improved.

Foucault et al. (2019) 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. Changes were categorized as "healthy", "repairing", or "harmful" and points were awarded or subtracted to the author depending on their behaviors. Each developer had an action feed where they can see their scores and view individualized feedback based on their history. Additionally, developers could optionally join "games" which would award levels and badges and compete with colleagues on a leaderboard. Feedback from developers was mostly positive, showing that a sense of competition motivated developers to address static analysis warnings more seriously.

Alhammad and Moreno (2018) "A systematic literature mapping has underscored the difficulty in fully corroborating the above claim because few empirical data are available so far". While past studies included the adoption of tools and practices into their investigation, instead of studying the behavior of a classroom of students, or a few small teams of professional developers, this thesis investigates 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 (ie. version control).

Chapter 3

Study Design

In this chapter, we detail the structure of the company where our case study took place and all necessary background pertaining to the study design.

3.1 Company Structure

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.

3.2 Gamification Timeline

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 3.1, and served as the basis for the badges in the studied gamification system.

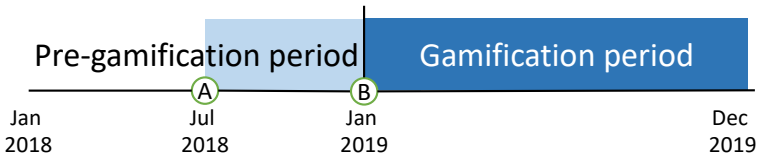

Figure 3.1: Timeline of the Gamification related events

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.

This gamification system was implemented with the aim that users are motivated and self-starting when adopting new practices. No additional incentives such as a monetary or material prize, or a promise of advancement in their careers were provided to users.

3.3 Research Questions

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:

- **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 which are earning badges, this

question investigates how each badge earned is associated with change in their key metrics. More precisely, we want to investigate the associated effects of applying a badge gamification system on a set of key delivery, quality, and throughput 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.

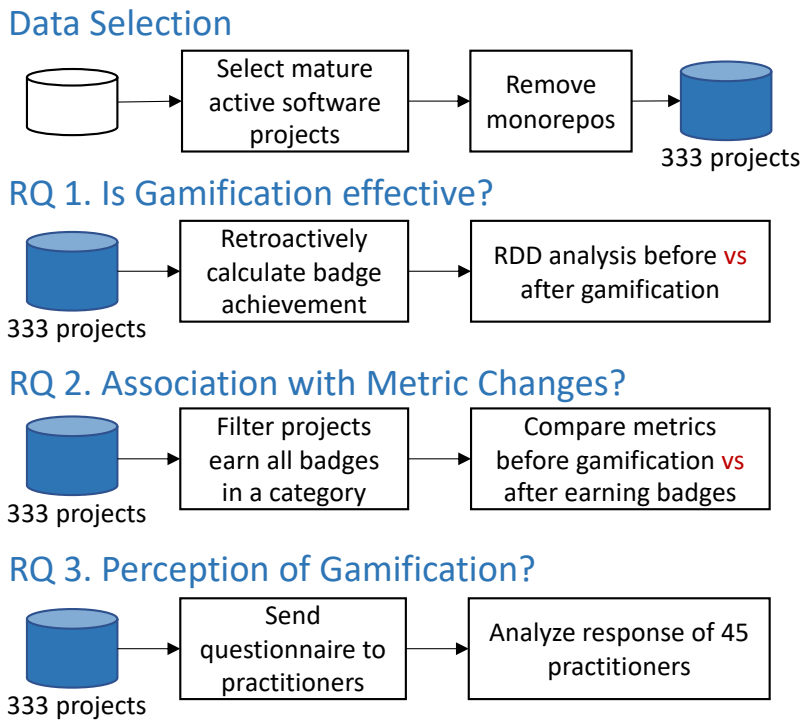

Figure 3.2: Methodology Overview

Figure 3.2 presents a high-level abstraction of our methodology. In the remainder of the section, we describe the datasets used (Section 3.4), the system of badges deployed by the company under study (Section 3.5), and the metrics we select to evaluate the delivery, quality, and throughput of teams before and after gamification (Section 3.6).

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.

3.4 Data Selection

The data used for this case study is extracted from the following three systems: JIRA, Git, and Jenkins. The majority of teams are using JIRA for issue tracking, Git for source control, and Jenkins for continuous integration, however, the way each platform is used varies considerably across teams. Some teams are JIRA power users and have mature and disciplined usage patterns to optimize the accuracy of their issue tracking. On the other hand, there are teams with minimal JIRA usage where the system is essentially used as a todo list with very little additional data describing the issues themselves and the software process. Similarly, with Git, we may see a variety of different branching strategies in place, and on Jenkins, some teams have well optimized CI pipelines including compilation, testing, static analysis and more, while others have very simple verification jobs which simply check that the code compiles. This variance in usage patterns is an important property of our study, as it gives context to understanding the design of the gamification system and the landscape of the development environment before and after it was deployed. A number of selection criterion have been chosen to filter the dataset down to a 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.

3.4.1 Selecting Active Software Projects

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 filtration process by removing projects which 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, have had releases during these years, and are developed by a team of developers. Immature projects are removed for two primary reasons. Firstly, we want

to focus on projects which are stable and have the cycles to focus on addressing the items suggested by the badges, and secondly, new projects generally have an easier time to adapt to best practices as they likely have less technical debt inhibiting them from adoption. Inactive and personal projects are removed as they are not representative of the active software development projects we aim to analyze. In summary, we apply the following filtration criteria:

- Project has minimum 5 distinct authors
- Project is at least 3 years old
- Project has at least 100 commits per year in 2018 and 2019
- Project has at least one build on CI per year in 2018 and 2019
- Project has at least one pull request per year in 2018 and 2019
- Project has at least one release per year in 2018 and 2019
- Project has at least one production deployment per year in 2018 and 2019

One clarification to be made here is around the filter requiring that projects have at least one build on CI per year. While some of the practices which are gamified involve continuous integration practices, this filtration requirement just asks that projects use CI in any capacity, not limited to deployments. At the company under study, projects under development within the past 7 years do use CI to package and release their projects by default. This filter aids in removing legacy projects which are no longer being regularly maintained.

3.4.2 Removing Monorepo Projects

In the gamification implementation studied, badges operate on a repository level under the assumption that one repository contains one project. This is problematic for monorepo projects, a repository which contains many independently developed subprojects. A technical limitation of the gamification system under study is that evaluating badge achievement

for these projects will not be accurate as the badges will not reflect the practices of a single development team, but the aggregated subprojects, we therefore remove monorepos from our dataset.

3.4.3 Removing Configuration Projects

The last step of our filtration process removes repositories primarily composed by configuration files. It is common practice to store the configuration for a project seperately in its own repository, but since they are mainly static resources, the activities promoted by the badges do not apply to these projects. Hence, we exclude repositories containing more than 60% of the files as non-code files.

3.4.4 Description of Selected Projects

After our selection process, we identify 333 projects that are candidates for our study. As shown in Table 3.1, projects in our curated dataset have sufficiently long development time (~ 5 years), and are developed by large teams (~ 30 collaborators).

Table 3.1: Descriptive statistics of the 333 selected projects.

	Median	**Min**	**Max**
Project Age (years)	~ 5	> 3	< 20
Total Commits	~ 2000	> 100	$< 70,000$
Authors	~ 30	> 3	< 300
Files	~ 1000	> 40	$< 20,000$

3.5 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. In addition to a category, each badge is assigned a requirement which must be met and

maintained in order for it to be achieved. The badges considered in this study are outlined in Table 3.2 along with the rationale for each badge's design.

3.5.1 Deployment Badges

Deployment badges focus on encouraging users to adopt automated deployment practices. At a base level, this badge category asks that users implement automated deployments by encapsulating their deployment procedures into automated deployment scripts. Once a team has automated their deployment processes, they are suggested to integrate these processes with a continuous integration system so that it can be triggered in response to other events rather than a human at the command line. This is an enabler which allows a team to ultimately adopt continuous delivery practices. Finally, to maintain confidence in the automated processes, software development teams are encouraged to also automated post-deployment verification procedures so that they can rest at ease knowing their unattended deployments were successful.

3.5.2 Git Badges

There are many different approaches to branching and releasing software (Shihab, Bird, & Zimmermann, 2012). Git badges are designed to encourage one particular practice called Trunk Based Development where all changes are merged into the main branch known as the "trunk" in many version control systems and all releases are made from this one branch (Jorgensen, 2001). This strategy encourages an environment where the main branch of a repository is often in a releaseable state, aided by feature flags to manage the configuration of a piece of software, a key requirement for adopting continuous delivery (Hoyos, Abdelkareem, Mujahid, Shihab, & Bedoya, 2021). To encourage software development teams to adopt this practice, a badge which measures that the majority of releases come from the same branch was designed.

Table 3.2: The badges considered in our study.

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	Enables 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	Keeps 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	Keeps delivery pipeline flowing smoothly	–

3.5.3 Quality Tooling Badges

Static analysis tooling such as linters can be helpful in enabling software developers to avoid simple and common mistakes and code smells early in the software development process (Ayewah & Pugh, 2010). They are also useful for encouraging specific conventions and coding styles in a software project to encourage consistancy (B. Johnson, Song, Murphy-Hill, & Bowdidge, 2013). The use of these tools shifts the detection of these issues earlier in the software development lifecycle allowing them to be fixed sooner rather than later. Detecting these issues earlier also saves wasted time in the code review and testing phases as it reduces the number of times a developer needs to iterate on their changes to fix these simple mistakes (B. Johnson et al., 2013). To encourage software development teams to take advantage of these benfits, badges are used to promote these practices.

3.5.4 Review Badges

While static analysis tooling can help automate the detection of simple issues, other semantic issues need human review to detect. It is easy for software development teams to be hasty and merge their code immediately, skipping the code review phase. Code review has the benefit of finding issues before merging and deploying software changes, with the added bonus of increasing the spread of knowledge about a software system among the team as more individuals are involved in the process (Bird & Bacchelli, 2013). In this gamification system, a badge was designed which encourages software developers to comment on pull requests by their peers as evidence that they are participating in some sort of code review process.

3.5.5 Stability Badges

When encouraging the adoption of trunk based development, software development teams will start creating all releases from a single branch branch, requireing it to always be in a healthy state for the delivery pipeline to flow freely (Laukkanen, Itkonen, & Lassenius, 2017). To conteract this effect, the gamification system also provides a badge encouraging

stability of this main branch to ensure that it is always in a releasable state. Stability badges encourage that failed builds on continuous integration are fixed as quickly as possible to not block the release pipeline. Being able to keep the build releasable as often as possible is a key tenant in both Agile and DevOps processes (Rodrguez et al., 2017).

3.5.6 Testing Badges

With a potential increase in release frequency due to the encouraged automation practices, the load on QA and testers increases as more releases are coming their way with less time between them for testing activities. This creates a natural need to automate the testing processes as well. The gamification system provides two badges related to automated testing. Firstly, teams are encouraged to write automated tests and run them as part of their build process in continuous integration. A failure in the tests should trigger a failure in the build thus preventing it from release. This practice also allows software development teams to export their automated test results on every build providing evidence of testing which is useful for audit and traceability purposes. Adding testing to the build pipeline on the other hand could potentially slow it down, especially if there are a lot of tests being run. The second badge encourages software development teams to optimize their tests to run as quickly as possible to mitigate this side-effect.

3.6 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 3.3.

3.6.1 Delivery Metrics

With the delivery metrics, we aim to evaluate how fast teams are delivering software in association with the introduction of the gamification system. Given that the gamification system affects different stages of the development lifecycle, we select a number of delivery metrics that cover each one of the stages we deem may suffer influence. Change Lead Time evaluates the time taken for individual commits to reach production, Cycle Time takes account for the total development time of a task, and Time to First Commit gives a view on the lead time before work is logged against a task. To understand how much time is spent on reviewing activities, Average Review Time is included in our suite of delivery metrics. Finally, Mean Time to Resolution tells us the average life time of a task from end to end. These selected delivery metrics allow us to analyze which activities are getting faster, in addition to the total delivery time. In all cases, a lower value indicates that a software is being delivered more quickly (the desired effect).

3.6.2 Quality Metrics

When software development teams are encouraged to produce more code more quickly, it is also essential to ensure that quality is not negatively affected as a result. We evaluate two key indicators of software quality and how they vary with the introduction of the gamification system. A decrease in Ratio of Bug Fixing Commits is desirable as it indicates the team needs to spend less time on bug fixing activities while a high value for Build Stability is desirable indicating the build is more stable.

3.6.3 Throughput Metrics

In order to measure whether the gamification system is enabling teams to deliver a greater quantity of code, we select several key throughput metrics to understand if the overall throughput of a team is changing in response to the implementation of gamification. Commit, pull request, and release counts normalized by the number of contributing developers in each project were selected for monitoring. We select these metrics because

they provide a view on the quantities produced at both the contributor level (commit and pull request counts) and the product level as a team (release count). In addition to giving visibility on the contributor and product level, commit and pull request counts can be internal measures which give a view of how a team decides to partition their work, and release count gives an exerternal view on how a client or stakeholder would perceive the output of a software development team. In all cases, an increase in these values is desirable as it indicates that a software development team has increased their output in response to the implementation of gamification.

With the relevant background on the study in place, we are prepared to investigate our research questions. In the next chapter, we begin our study by looking at how the implemention of gamification affected the adoption rates of the gamified practices.

Table 3.3: 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	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.
	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 spent 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

Chapter 4

Impact on Adoption of DevOps Practices

A series of badges were designed and presented to users to encourage the adoption of new DevOps practices. In this chapter, we investigate whether or not 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 (Hamari et al., 2014; Hanus & Fox, 2015; Nacke & Deterding, 2017), we have yet to see its effectiveness on large software development companies. Answering this question may shed the light on the benefits and limitations of gamification as a strategy for changing development practices.

4.1 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 for 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) (Thistlethwaite & Campbell, 1960), 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 (Trockman et al., 2018; Zhao, Serebrenik, Zhou, Filkov, & Vasilescu, 2017; Zimmermann & Casanueva Arts, 2019).

First, we conduct an analysis on the average number of badges achieved by all projects, hence our outcome variable Y is the mean number of badges achieved by projects, in which we look for a discontinuity.

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

where T represents **time** in months from the start of the observation period, G is a binary flag indicating the occurrence of **gamification** ($G = 0$) and after gamification ($G = 1$); and A represents the number of months **after** gamification, coded 0 before gamification and incrementally increasing after gamification. In the **control** (C), we include the occurrence of Event A (the DevOps Guidelines document described in Section 3.2), 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)

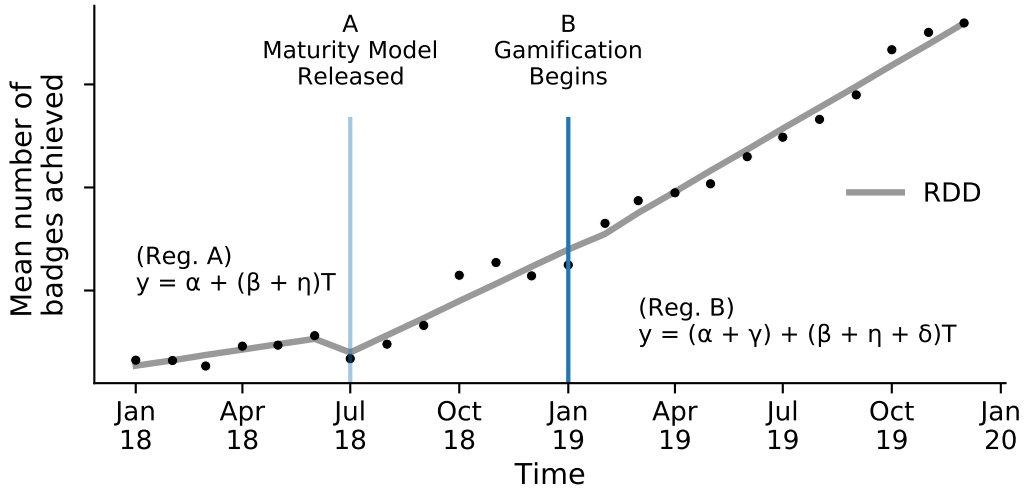

Figure 4.1: The evolution of adopted DevOps practices per month for the 333 studied projects.

and in the slope (δ) of badge adoption once gamification is introduced.

Over time since the gamification period began, more badges were introduced to developers. 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 3.2.

4.2 Results

4.2.1 Mean Number of Practices Adopted

The adoption of the recommended practices over time is visualized in Figure 4.1. In this figure, the x-axis presents the timeline on a monthly basis as of the last day of each month. The y-axis represents the mean number of practices adopted across all studied projects, with real values kept anonymized. We then compare the pre-gamification period with the gamification period to see if gamification helped accelerate the adoption of the recommended DevOps practices.

As Figure 4.1 shows, there is no noticable increase in the mean number of recommended practices adopted after the gamification period starts. When applying the RDD analysis on the average badge adoption, γ and δ are not found to be statistically significant in our model ($R^2 = 0.99$), supporting this observation. On the other hand, the occurence of Event A (the release of the DevOps guidelines) showed to be more influential in changing the trend of the mean number of adopted practices. The likely cause for this is an increase in adoption of *Project has Automated Deployment CI Job* following this event as seen in Figure 4.3. While observing the mean number of recommended practices adopted does not show signs of gamification accelerating the adoption, we continue our investigation at the individual badge level to see if gamification has a stronger effect when targeting specific practices.

4.2.2 Individual Practice Adoption

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 Figures 4.2 through 4.6. In these figures, 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. Table 4.1 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.

Deployments are Automated

Before gamification, the practice of *Automated Deployments* (Figure 4.2) had a moderate level of adoption and similarly saw a moderate amount of growth over the gamification period. The RDD analysis showed no significant change in the level but showed a significant increase in the slope of adoption ($\delta = 2.6$), indicating that gamification can help boost this

Table 4.1: Adoption of the five badges achieved before gamification and the increase in adoption after a year of gamification.

Category	**Badge**	**Initial Adoption**	**RDD Analysis**					**% Impr,**
			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%

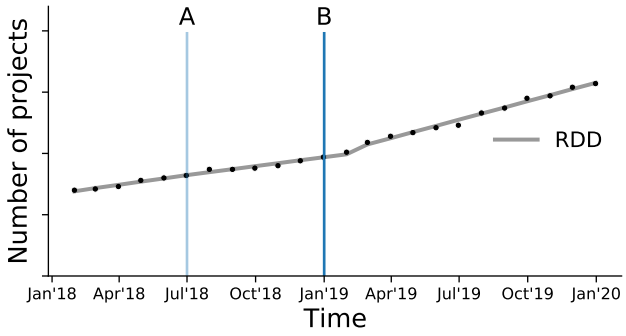

Figure 4.2: Evolution of adoption for Deployments are automated.

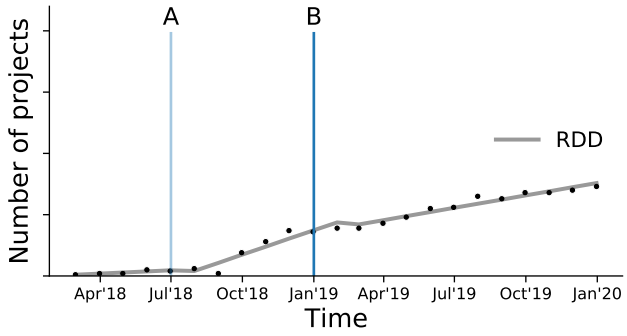

Figure 4.3: Evolution of adoption for Project has deploy CI job.

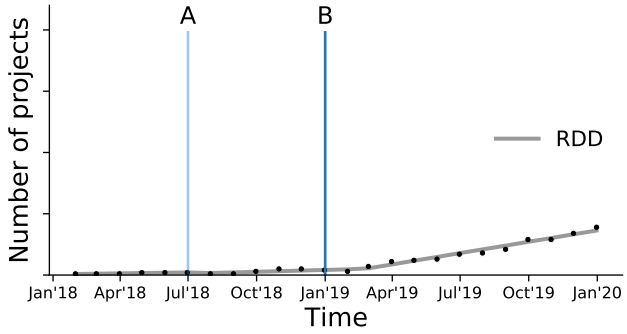

Figure 4.4: Evolution of adoption for Deployment Verification is Automated.

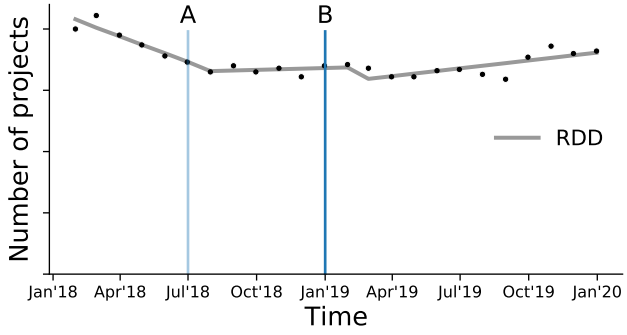

Figure 4.5: Evolution of adoption for Project uses trunk based development.

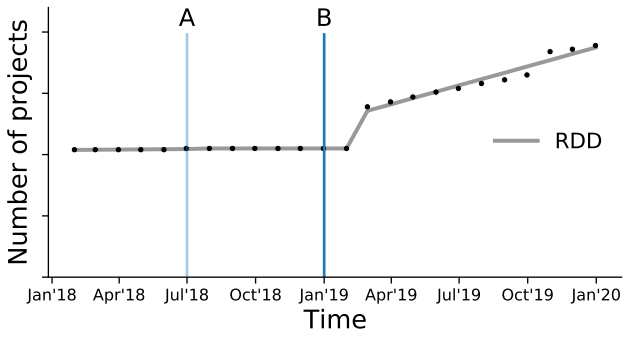

Figure 4.6: Evolution of adoption for Automated Tests are Run on Builds.

practice adoption more gradually, as a continuous and long term change. This increase in slope corresponds to a 2x increase in the rate of adoption after gamification.

Project has Automated Deployment CI Job

Project has Automated Deployment CI Job (Figure 4.3) 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.

Deployment Verification is Automated

Post-Deployment Verification is Automated (Figure 4.4) saw the largest increase in adoption level ($> 650\%$) and displayed a 6x increase in the rate of adoption ($\delta = 2.6$). 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

Project uses Trunk Based Development (Figure 4.5) was the only practice with a high initial level of adoption, and also the only practice which did not see a statistically significant

change in level or slope with gamification. Gamification did not appear to be a strong influencer on the adoption rate of this practice which only showed an approximate 10% increase in adoption level at the end of the gamification period.

Automated Tests are Run on Builds

Finally, *Automated Tests are Run on Builds* (Figure 4.6) 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.

> Gamification accelerated the adoption of practices related to Testing and Deployment, with increases in adoption rates from 60% to 650%. It showed no significant influence on Git practices which were widely adopted before gamification.

As we have observed that gamification has helped accelerate the adoption of these practices, In the next chapter, we will investigate how these practices affect the metrics of the software projects under study.

Chapter 5

Impact on Software Project Metrics

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 have seen in Chapter 4 that gamification has accelerated the adoption of several practices, in this chapter, we examine if the promotion and adoption of these practices is associated with measurable changes of delivery, quality, and throughput metrics on the teams which adopt them.

5.1 Exposure to Gamification

In this section, we evaluate if the exposure of developers to the gamification system may have influenced the metrics related to team's delivery, quality and stability.

5.1.1 Approach

We compute each metric for all 333 projects during the pre-gamification period and compare it with the same metric during the gamification period. Given we analyze a large number of projects, we expect that the influence of the gamification to be the main driver force for any substantial change on the project metrics. In the following, we describe each

of the methodological steps in detail.

First, we compute each metric at the smallest granularity possible. This is done based on the source data and metric. All of the productivity metrics and the build stability metrics do not require this pre-processing step as they simply require counting in the aggregate level in the next step.

- **Change Lead Time:** For each commit, we compute the lead time from the timestamp when it was committed to the time when that commit reaches the production environment.
- **Cycle Time:** For each JIRA issue, we compute the total time spent in an "In Progress" state.
- **Review Time:** For each pull request, we compute the amount of time elapsed from when it was opened to when it was merged.
- **Time to Resolution:** For each JIRA issue, we compute the amount of time elapsed from the time it was opened to when it was resolved.
- **Time to First Commit:** For each JIRA issue, we compute the amount of time elaped from the time it was opened to the timestamp of the first related commit.
- **Ratio of Bug Fixing Commits:** For each commit, we classify it as bug fixing or not based on the type of the related JIRA issue.

Next, for each project we aggregate each metric for each period evaluated, the pre-gamification period (2018) and the gamification period (2019). The output of this step is a single value for each project in each period for every metric. For each of the delivery metrics, we group the values produced in the first step by period, then compute the mean value. For ratio of bug fixing commits, we group the classified commits by period, then compute the ratio of bug fixing to non bug fixing commits. For build stability, similarly, we group the CI builds by period, then compute the ratio of passing to failing builds. For all of the productivity metrics, we group the entity being counted (commits, prs, or releases) by period to

Table 5.1: Correlation of exposure to gamification on the delivery, quality and productivity metrics.

Metric Category	Metric	Cliffs Delta	
Delivery	**Change Lead Time**	-0.054 **	Negligible
	Time to First Commit	0.120 ***	Negligible
	Mean Time to Resolution	0.181 ***	Small
Throughput	**Normalized Commit Count**	-0.081 ***	Negligible
	Normalized Release Count	0.161 ***	Small

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

produce a count value for each period and then normalize these count values by the number of contributing authors in the according period. We then combine the metrics of all projects into two distributions, related to the pre-gamification and gamification periods. These two distributions are then tested for significant changes in each of our selected metrics using the Wilcoxon Signed-Rank Test (Wilcoxon, 1945). To quantify the effect size of statistically significant changes, we resort to the Cliff's Delta effect size (Cliff, 1993) and use Romano et al's guide for interpreting the effect size (Romano, Kromrey, Coraggio, Skowronek, & Devine, 2006), similarly to previous works (Costa, Bezemer, Leitner, & Andrzejak, 2021; Wessel et al., 2018).

$$\text{Effect size } d = \begin{cases} negligible(N), & \text{if } |d| \leq 0.147 \\ small(S), & \text{if } 0.147 < |d| \leq 0.33 \\ medium(M), & \text{if } 0.33 < |d| \leq 0.474 \\ large(L), & \text{if } 0.474 < |d| \leq 1 \end{cases}$$

5.1.2 Results

Table 5.1 summarizes the results of measuring the response in metrics to the inception of gamification. We saw negligible to small impact for some delivery and throughput metrics. Quality metrics did not show any impact in this experiment.

Delivery Metrics

The Wilcoxon Signed-Rank Test yielded significant differences in three out of the five delivery metrics, suggesting the exposure to the gamification may have excerted an influence on the mean values of these delivery metrics. The three metrics which saw differences are Change Lead Time, Time to First Commit, and Mean Time to Resolution. While the change of the distributions for these metrics was statistically significant, the magnitude of the change was negligible or small. In the case of Time to First Commit and Mean Time to Resolution, we obtained a positive Cliff's Delta indicating that compared to the pre-gamification period, teams have a slightly longer time to first commit, and time to resolution. This can suggest that teams are spending more time on the development and review activities as seen in the increase of these metrics. For Change Lead Time, however, we obtained a negative Cliff's Delta suggesting that individual commits are being deployed to production more quickly.

Throughput Metrics

As for the throughput metrics, Normalized Commit Count and saw statistically significant differences, while Normalized Pull Request Count appears to be unaffected as a result of exposure to gamification. Normalized Commit Count saw a negative Cliff's Delta, of negligible magnitude, indicating teams are producing fewer commits after gamification began compared to the pre-gamification period. Normalized Release Count on the other hand, saw a small positive change meaning teams are releasing more. Although we are seeing slightly fewer commits after gamification began, it is quite possible that it has enabled teams to deliver more frequently as suggested by a positive Cliff's Delta for Normalized Release Count.

Exposure to the gamification system had mostly negligible and some small effects on some Delivery and Throughput metrics. While Mean Time to Resolutionis slightly longer suggesting JIRA issues are being resolved more slowly, an increase in Normalized Release Count suggests that teams are working in smaller increments and releasing more often.

5.2 Earning Badges

After observing that the introduction of Gamification has had a mostly negligible effect on the metrics of the studied software projects, we continue our study by evaluating only those teams who have earned badges to assess their impact of earned badges on software metrics.

5.2.1 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 complemetary to each other as shown in the categories of badges of Table 3.2. 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 a category on each of the selected metrics. For each badge category, we find the projects which have earned all of the badges in that category during the same month (e.g., all deployment badges). For example, when evaluating the association between the deployment badges and our selected metrics, we only consider projects which have earned both "Deployments are Automated" and "Post-Deployment Verification is Automated" in the same month. 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. As in the previous experiment, both distributions are tested for significant changes in each of our selected metrics using the Wilcoxon Signed-Rank Test, and the effect size of statistically significant changes are measured using Cliff's Delta effect size.

5.2.2 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 a combination of 60 experiments (6 badge categories x 10 metrics), as shown in Table 5.2. 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 5.3 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 have had overall 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

Table 5.2: Comparison of projects' metrics distribution before earning the badges and after earning the badges. We denote "x" on the comparisons that have shown a significant change on the metric and "–" on the comparisons with no significant change (Wilcoxon Signed-Rank Test).

	Metrics	Deployment	Git	Quality Tooling	Review	Stability	Testing
Delivery	Change Lead Time	–	–	–	x	–	–
	Cycle Time	–	–	x	–	–	–
	Time to First Commit	–	–	–	–	–	–
	Mean Time to Resolution	–	–	–	–	–	x
	Average Review Time	–	–	–	–	–	–
Qual.	Ratio of Bug Fixing Commits	–	–	–	–	–	x
	Build Stability	–	–	–	–	–	–
Thr.	Norm. Commit Count	–	–	–	–	–	x
	Norm. Pull Request Count	x	–	–	–	–	x
	Norm. Release Count	–	–	–	–	–	–

Table 5.3: Relationship between earning badges and metrics.

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

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

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.

This result raises the question as to whether or not this reduction in output is due to efforts spent addressing the learning curve of new practices. A future long term study would be of value to better understand this result and observe whether or not this loss in throughput is temporary or not.

> We found significant changes in 7 out of 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.

In this chapter, we have seen how the gamified practices affected the metrics of those projects which adoped them. We now move our investigation towards understanding how software developers feel about gamification and how it has changed their overall development experience, described in the next chapter.

Chapter 6

Developer Perception and Reaction to Gamification

Badges can be expected to increase 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.

6.1 Approach

Chapters 4 and 5 look at the efficiency of gamification for promoting the adoption of new practices and the impact it has as a result on the metrics of projects which earn badges. While these questions look at the impact of gamification quantitatively, we would like to investigate the unmesaurable outcomes and human response to gamification. In this chapter, we designed and distributed 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 (Hoyos et al., 2021). No additional incentive such as a prize or reward was provided for participation in the survey.

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. 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 (Fincher & Tenenberg, 2005). 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.

6.2 Respondent Demographics

We detail the demographical information of survey participants in Table 6.1. 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.

6.3 Results

The results of the survey provide a number of interesting insights covering the benefits and criticisms of gamification by a diverse group of technologists. Table 6.2 summarizes the key themes in each area accompanied by the frequency of appearance. Of these themes, four themes emerge as overarching themes which appear in responses to more than one question in some capacity:

Table 6.1: Results of biographical survey questions.

Role	#	**Experience**	#	**Team Size**	#
Developer	26	$<$ 2 Years	4	1-3 Members	1
Tech Lead	11	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
Other	1				

(1) Reduction of manual overhead in repetitive tasks

(2) Adoption of standardized tooling and processes

(3) Improve software quality and testing practices

(4) Justification to product owners to improve internal processes

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.

Respondents want to reduce manual overhead in their deployment process (18 respondents). The most commonly cited motivating factor for adopting DevOps badges was 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.

Table 6.2: Summary of the common themes observed in the answers to the optional open-ended questions of our survey.

	Theme	Exemplary Quote	Freq.
MOTIVATION	Reduction of manual overhead in deployment process	*"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." "Care free change management, 0 click deployments (once change is approved)"*	18
	Adoption of standardized tooling and processes	*"Standards that enforce every project to do the same."*	11
	Sense of accomplishment at seeing progress	*"[...] compare with other teams and sense of achievement, competition to get to 100% across badges."*	5
	Friendly competition with colleagues	*"[...] peer pressure / friendly competition with other teams and projects of how many badges we have compared to others."*	5
	Improve software quality and testing practices	*"Our driving factor is our drive to reduce risk prone manual process, improve developer efficiency and improve product quality [...]"*	3
	Incentive from management	*"Department targets"*	3
	Justification to product owners to improve internal processes	*"Showing management that some of the devops changes made have a measurable goal [...]"*	2
	Showcasing achievements	*"It's nice to see them all green [...] Achievements always motivate people - whether on devops or XBox"*	2
ADOPTION	Useful to inform about better practices	*"[...] the badge, why it is a recommended practice and how to achieve it are invaluable tools for teams that are onboarding"*	8
	Improving transparency and communication	*"The badges have helped us identify at a repo and more macro levels where we need to invest devops effort."*	7
	Reduction of manual overhead in deployment process	*"I found automated deployment badge to be very helpful as it makes the deployment much easier." "[...] deployment automation badges have pushed us to automate our workflow 100% and now depoyments are much faster and completely hands-free"*	7
	Adoption of standardized tooling and processes	*"It tells me what should I do to adopt standards." "Unit Testing and Automation became part of our developemnt because of badges."*	5
	Setting targets for improvements	*"Deployment badges are key to our plans for automation in 2021"*	2
RESULTS	Reduction of manual overhead in deployment process	*"Smoother deployments, more frequent deployments, easier to release many projects (no difference between releasing 1 project or 20 projects)" "[...] Reduced my time spend with deliveries and reduced the risk of manual mistakes."*	13
	Improve software quality and testing practices	*"Reduced number of defects after code review badges implemented" "Introduced code quality tooling that helped with test coverage and technical dept. Gave up some bad practices of merging PRs without review."*	10
	Adoption of standardized tooling and processes	*"Helping to standardise setup across our repos [...]"*	3
	Improving transparency and communication	*"The main badges such as continuous integration/ PR review seems to keep people honest about their project. It avoids having a project fall on the sidelines."*	3
	Justification to product owners to improve internal processes	*"Deployment frequency increased, automation helps us to "normalise" changes and not make it a big deal [...]"*	2

> *"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*

Respondents want to adopt standardized tooling and processes over custom solutions (11 respondents). Eleven respondents specified that they are encouraged to achieve the badges due to wanting adopt standard tooling 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.

> *"Standards that enforce every project to do the same" - R9*

Developers enjoy a sense of accomplishment from seeing their progress and a friendly competition with colleagues (5 respondents). Aside from desiring improvements in delivery and efficiency, some respondents indicated that they are simply motivated by seeing their progress as they move through the badges and see them progressively getting achieved. Users also enjoyed using the badges as a means to promote some friendly competition among different teams in their area. As respondent 10 stated:

> *"Compare with other teams and sense of achievement, competition to get to 100% across badges" - R10*

Other notable motivations include wanting 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.

Table 6.3: Badge ranking by users

Badge	Avg Ranking	Users
Short unit test execution time	4.25	36
One branch is used for most releases	4.36	25
Pull requests have evidence of review	4.45	38
Projects use code quality tool	4.55	33
Project has evidence of unit tests	4.55	38
Successful verification of deployments	4.58	38
Failed builds are fixed quickly	4.65	26
Successful automated deployment	4.76	37
Project has deploy job in CI	4.92	38

Participants are driven to achieve DevOps badges for the prospect of reducing manual overhead, standardizing process across teams, sensing accomplishment in adopting practices, and friendly competition with their colleagues.

Are badges helpful in adopting DevOps practices?

First, we presented respondents with a list of nine badges and asked them to rate each badge from most to least helpful. An option to mark a badge as "non-applicable" was also provided if the respondent has no experience with a particular badge. The results of this question are presented in Table 6.3.

According to these results, the participants are quite evenly split across which badges are helpful or not. At first glance, it appears that across almost all of the badges, the mean rating is close to the middle of the scale (4.5). The top three badges with a slightly higher rating are "Short unit test execution time", "One branch is used for most releases", and "Pull requests have evidence of review" Because for most badges, the rating is close to the middle of the scale, this suggests that different teams have very contrasting opinions on the helpfulness of each badge.

To dig deeper, We designed a two-part question to investigate 1) if practitioners found badges to generally be 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 some reponses which apply to the badges in general, and some which are specific about individual badges. 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 is very strong. One clear example of this is *Post-Deployment Verification is Automated* . Reviewing Table 4.1 from RQ1, we can see that after the badge was created, the adoption level grew from very low by a large margin.

> *"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 reduce manual overhead in deployment processes (7 respondents). The next most popular theme with regards to the helpfulness of the badges is shared with one of the main driving factors for adopting badges as detailed earlier. Responses referring to this theme generally suggest that the deployment badges were helpful to drive them to adopt automated deployment practices. Similar to *Automated Tests are Run on Builds* , this feedback from users 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.

> *"Deployment automation badges have pushed us to automate our workflow 100% and now depoyments are much faster and completely hands-free"* - R41

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 to inform teams about better practices, improve transparency and communication, and reduce manual overhead. 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:

Respondents saw a 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. By adopting the practices associated with the deployment badges, some teams were able to reduce their manual overhead by enabling hands-free deployments. 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

Respondents saw an improvement in their 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

Referring back to the analysis on how the adoption of badges impact the key metrics of a project in RQ2, we saw that the adoption of the testing badges had a medium sized positive impact on the bug fixing commit ratio metric indicating that after earning the testing badges, teams were producing more bug fixing commits. This is likely a result of being able to uncover more bugs and issues through their newly adopted testing practices.

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.

Chapter 7

Discussion

In this chapter, we discuss the overarching themes that emerge from the findings reported in Chapters 4, 5, and 6, 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. In their work, de Paula Porto et al. (2021) found that Product Integration (Deployment) and Verification and Validation (Testing) are two of the most frequently cited areas in which gamification exhibited a positive outcome. Our results support this notion as this also holds true in the context of the large 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 (Chapter 4). Teams that earn testing badges are associated with an increase in the number of bug fixing commits (Chapter 5). Related studies have also reported that gamified testing has yielded improvements in defect registration (de Paula Porto et al., 2021). Finally, practitioners frequently cite the reduction of manual overhead in deployment processes as the main motivation for using the badges, and report

perceived improvement in software quality and testing practics (Chapter 6). 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 (de Paula Porto et al., 2021).

While deployment and testing practices have shown to be good candidates for effective gamification, not all practices are naturally well suited for gamification. Practices with a high initial adoption level (Git) saw the least growth and our model did not see gamification to be a significant factor (Chapter 4). Furthermore, Git related badges did not make an apperance in any statistically significant result when evaluating the change in project metrics (Chapter 5). Likewise, these same badges were rated to be second from last in terms of utility and was never mentioned in any of the long form responses (Chapter 6).

Contrasting the success of the deployment and testing badges, with the lack of success of the Git badges, there is a clear lesson learned. Gamification appeared to be effective with little known and new practices. Developers of new tools and techniques should strongly consider a gamification plan to promote the adoption of their product. This can provide potential new users with a simple and effective adoption plan to onboard them with ease.

Only some badges showed 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.

This finding, in conjunction with the results seen in Chapter 4 suggest that gamification may be most effective when applied with a very targeted focus. While a given practice would normally be presecribed to a software development team with the ultimate aim improving specific metrics, the improvement in these metrics may come more long term beyond the scope investigated in this study. As seen in Chapter 4, developers do take action based on being presented the badges. It could be possible to design gamification elements which directly target the improvement of a metric rather than the adoption of a practice. For example, a badge asking for users to keep their defect rate below a specific threshold may accomplish this goal of improving specific metrics.

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. Additional work could also investiage the long term effects of the gamified practices on the metrics of software projects. While our study only looked at the first year after gamification, it could be possible that some benefits are more pronounced after a longer period of practice has passed. It could also be of value to measure not only whether a team has simply adopted a given practice or process, but to investigate to what degree they have done so. For example Pull Requests are Reviewed asks developers to comment on PRs by their peers. Commenting for the sake of commenting may not be of much value, therefore it could be worthwhile to evaluate the quality and utility of the comments themselves.

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, de Paula Porto et al. (2021) also noticed the same problem in four of the studies they reviewed (Dalpiaz et al., 2017; de Melo et al., 2014; Johansson & Ivarsson, 2014; Scherr, Elberzhager, & Holl, 2018), it is difficult to manage motivations and get users to focus on the right things.

Another caution that must be made with gamification systems is that if they are not carefully designed, they may be gamed themselves by users. The target criteria and KPIs should be designed so that they are difficult to fictitiously meet. For example, asking users to comment on pull requests more frequently may result in useless comments such as "Reviewed" or "Approved" simply for the reason of meeting a set of criteria. This kind of response to gamification has no benefit for users and weakens the value of the gamification system. It is also important to design gamification systems in such a way that the software development teams don't feel like it is being used to evaluate or measure them for performance reasons. It should be a positive encouragement tool and not a monitoring tool used to blame or punish developers. For a gamification system to be successful, there should be a basic amount of trust between the software development teams, the gamification designers, and management that the system will be used appropriately.

Chapter 8

Threats to Validity

As with other empirical studies, there are threats to validity which may impact our study. In this section, we discuss the threats to the validity of our findings, broken down by internal, construct, and external validity.

8.1 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 Chapter 4. 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 3.1 to represent the communication of the DevOps Guidelines document. In Chapter 4, this is factored in as a control variable to observe how strongly it impacts our results. Similarly, in Chapter 5, 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 Chapter 6, surveys can be subject to human error and bias. 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. Open ended survey questions were optional, and therefore a smaller number of participants responded to these questions, possibly limiting the variety of responses received. Finally, the question where participants were asked to rank the badges in order of perceived utility, was misunderstood by many participants as a rating question. While the response rate of 7.5% is similar to other studies, the distinguishing factor here is that this particular study was done internally at a company in industry as opposed to an open source environment. One possible reason for a lower participation rate could be that while this survey was advertised as being anonymous, some individuals may still distrust the anonymity in concern for voicing their opinions in the workplace over a survey.

8.2 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. Examples are quality and productivity of a team. 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.

8.3 External Validity

Threats to external validity are related to the generalizability of our findings. 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. Tools and practices which may differ from company to company could possibly affect the effectiveness of gamification when deployed in other environments.

Chapter 9

Conclusion, Contributions, and Future Work

In this chapter, we summarize the conclusions and contributions of this thesis, and provide some possible avenues for future work.

9.1 Conclusion

In this thesis, we conducted a mixed-methods study on the effects of badge-based gamification at a large company.

In Chapter 4, we asked our first research question; **"Is gamification effective as a means to promote the adoption of new practices?"**. To answer this question, we used statistical modeling techniques to investigate how badges can accelerate the adoption of new DevOps practices. We saw that gamification can be effective in promoting the adoption of new practices and processes with practice adoption increasing at least 60% in most practices.

We continued this study in Chapter 5 asking **"How does gamification impact the metrics of projects due to earning badges?"**. In this chapter, we studied the association between gamification and software metrics. First, we looked at whether or not the presence of a gamification system impacted the metrics of the projects under study and

saw very modest results suggesting that teams are taking slightly longer to resolve JIRA issues while they are working in smaller increments and releasing more often. Next, we investigated whether or not earning badges had a more pronounced effect. We found that teams who earned testing badges showed an increase in the number of bug fixing commits, but output fewer commits and pull requests. Teams that earned Code Review and Quality Tooling badges have shown shorter change lead time and cycle time metrics.

Finally, in Chapter 6, we ask **"How do software developers react to gamification and perceive its impact?**. We conducted a survey with practitioners to understand how developers react to gamification and perceive its impact. 73% 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. 62% of the survey participants perceived tangible benefits resulting from adopting these practices. While we heard from developers what they found useful about gamification, we also learned about their criticisms and that gamification systems need to be carefully designed to avoid driving the wrong motivations for change and wasting the time of developers who already have other solutions which are working well for them.

9.2 Contributions

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 on gamification.

9.3 Future Work

While this thesis makes a new contribution to the current literature on gamification in software engineering, there are more avenues which can be pursued to continue on this thread of research.

9.3.1 Evaluate other Gamification Mechanisms

While this thesis focused exclusively on badge-based gamification, there are many more gamification mechanisms available for gamification system designers to use such as points, leaderboards, quests and levels. One avenue of interesting future work could be a comparative study of these different mechanisms to evaluate the effectiveness of each technique.

9.3.2 Evaluate the Long Term Effects of Gamification

This thesis took place over the course of two years, comparing one year of behavior in a non-gamified environment with a full year in a gamified environment. There could be a learning curve associated with learning new processes and practices which may result in the benefits of the associated practices manifesting later on once a software development team becomes more comfortable with it. One of the survey participants in this study suggested that it was too early to evaluate the benefits at the time of the study. A longer term study would help uncover whether or not teams stick to their adopted practices long term and what the long term effects on their metrics would be.

9.3.3 Establish and/or Confirm a link between Practitioners' Perception and the Result in their KPIs

As mentioned in Chapter 7, many participants of our survey mentioned that deployment badges were a game change in terms of utility. While users were very enthusiastic about this aspect in particular, we did not observe any positive outcomes in the evaluated metrics related to this item in particular. The community could benefit from a more in depth study about the relationship between perception and reality with participants in a gamified

system. A different set of KPIs, such as developer productivity KPIs could be of benefit to better capture the effects on the developers themselves to possibly cover the gap where our selected metrics were unable to detect changes.

9.3.4 Evaluate the Effects of Gamification on Individual Contributors

In our investigation, we saw the effects of gamification on team behavior and software development projects. Gamification could also potentially be useful for driving desired behaviors in individuals. A future study could design a gamification system which targets practices which individual contributors should be adopting and evaluate how the adoption of these practices affect their performance and productivity.

9.3.5 Apply Gamification to New Employee Training and Onboarding

We saw in this study that gamification has a powerful educational effect as it was effective in getting software development teams to adopt new behaviors. Another future study could look at using gamification in other aspects of software development such as employee onboarding and training. While a new joiner may go through an initial training period, a gamification system may help with their long term integration and development in their new work environment.

References

Alhammad, M. M., & Moreno, A. M. (2018). Gamification in software engineering education: A systematic mapping. *Journal of Systems and Software*, *141*, 131-150. Retrieved from https://www.sciencedirect.com/science/article/pii/S0164121218300645 doi: https://doi.org/10.1016/j.jss.2018.03.065

Anderson, A., Huttenlocher, D., Kleinberg, J., & Leskovec, J. (2013). Steering user behavior with badges. In *Proceedings of the 22nd international conference on world wide web* (p. 95106). New York, NY, USA: Association for Computing Machinery. Retrieved from https://doi.org/10.1145/2488388.2488398 doi: 10.1145/2488388.2488398

Ayewah, N., & Pugh, W. (2010). The google findbugs fixit. In *Proceedings of the 19th international symposium on software testing and analysis* (p. 241252). New York, NY, USA: Association for Computing Machinery. Retrieved from https://doi.org/10.1145/1831708.1831738 doi: 10.1145/1831708.1831738

Bird, C., & Bacchelli, A. (2013, May). Expectations, outcomes, and challenges of modern code review. In *Proceedings of the international conference on software engineering* (Proceedings of the International Conference on Software Engineering ed.). IEEE. Retrieved from https://www.microsoft.com/en-us/research/publication/expectations-outcomes-and-challenges-of-modern-code-review/

Cico, O., Jaccheri, L., Nguyen-Duc, A., & Zhang, H. (2021). Exploring the intersection between software industry and software engineering education - a systematic mapping of software engineering trends. *Journal of Systems and Software*, *172*, 110736. Retrieved from https://www.sciencedirect.com/science/article/pii/

S0164121220301667 doi: https://doi.org/10.1016/j.jss.2020.110736

Cliff, N. (1993, 11). Dominance statistics: Ordinal analyses to answer ordinal questions. *Psychological Bulletin*, *114*, 494-509. doi: 10.1037/0033-2909.114.3.494

Costa, D., Bezemer, C.-P., Leitner, P., & Andrzejak, A. (2021). What's wrong with my benchmark results? studying bad practices in jmh benchmarks. *IEEE Transactions on Software Engineering*, *47*(7), 1452-1467. doi: 10.1109/TSE.2019.2925345

Cruz, C., Hanus, M. D., & Fox, J. (2017). The need to achieve: Players' perceptions and uses of extrinsic meta-game reward systems for video game consoles. *Computers in Human Behavior*, *71*, 516-524. Retrieved from https://www.sciencedirect.com/science/article/pii/S0747563215300960 doi: https://doi.org/10.1016/j.chb.2015.08.017

Dalpiaz, F., Snijders, R., Brinkkemper, S., Hosseini, M., Shahri, A., & Ali, R. (2017). Engaging the crowd of stakeholders in requirements engineering via gamification. In S. Stieglitz, C. Lattemann, S. Robra-Bissantz, R. Zarnekow, & T. Brockmann (Eds.), *Gamification: Using game elements in serious contexts* (pp. 123–135). Cham: Springer International Publishing. Retrieved from https://doi.org/10.1007/978-3-319-45557-0_9 doi: 10.1007/978-3-319-45557-0_9

de Melo, A. A., Hinz, M., Scheibel, G., Diacui Medeiros Berkenbrock, C., Gasparini, I., & Baldo, F. (2014). Version control system gamification: A proposal to encourage the engagement of developers to collaborate in software projects. In G. Meiselwitz (Ed.), *Social computing and social media* (pp. 550–558). Cham: Springer International Publishing.

de Paula Porto, D., de Jesus, G. M., Ferrari, F. C., & Fabbri, S. C. P. F. (2021). Initiatives and challenges of using gamification in software engineering: A systematic mapping. *Journal of Systems and Software*, *173*, 110870. Retrieved from https://www.sciencedirect.com/science/article/pii/S0164121220302600 doi: https://doi.org/10.1016/j.jss.2020.110870

Deterding, S., Dixon, D., Khaled, R., & Nacke, L. (2011). From game design elements to gamefulness: Defining gamification. In *Proceedings of the 15th international academic mindtrek conference: Envisioning future media environments* (p. 915). New York,

NY, USA: Association for Computing Machinery. Retrieved from https://doi.org/10.1145/2181037.2181040 doi: 10.1145/2181037.2181040

Dicheva, D., Dichev, C., Agre, G., & Angelova, G. (2015, 07). Gamification in education: A systematic mapping study. *Educational Technology & Society*, *18*, 75-88.

Dubois, D. J., & Tamburrelli, G. (2013). Understanding gamification mechanisms for software development. In *Proceedings of the 2013 9th joint meeting on foundations of software engineering* (p. 659662). New York, NY, USA: Association for Computing Machinery. Retrieved from https://doi.org/10.1145/2491411.2494589 doi: 10.1145/2491411.2494589

Fincher, S., & Tenenberg, J. (2005, 07). Making sense of card sorting data. *Expert Systems*, *22*, 89-93. doi: 10.1111/j.1468-0394.2005.00299.x

Foucault, M., Blanc, X., Falleri, J.-R., & Storey, M.-A. (2019, 12). Fostering good coding practices through individual feedback and gamification: an industrial case study. *Empirical Software Engineering*, *24*. doi: 10.1007/s10664-019-09719-4

Garca, F., Pedreira, O., Piattini, M., Cerdeira-Pena, A., & Penabad, M. (2017). A framework for gamification in software engineering. *Journal of Systems and Software*, *132*, 21-40. Retrieved from https://www.sciencedirect.com/science/article/pii/S0164121217301218 doi: https://doi.org/10.1016/j.jss.2017.06.021

Garca-Mireles, G. A., & Morales-Trujillo, M. E. (2020). Gamification in software engineering: A tertiary study. In J. Mejia, M. Muoz, . Rocha, & J. A. Calvo-Manzano (Eds.), *Trends and applications in software engineering* (pp. 116–128). Cham: Springer International Publishing.

Hamari, J. (2017). Do badges increase user activity? a field experiment on the effects of gamification. *Computers in Human Behavior*, *71*, 469-478. Retrieved from https://www.sciencedirect.com/science/article/pii/S0747563215002265 doi: https://doi.org/10.1016/j.chb.2015.03.036

Hamari, J., Koivisto, J., & Sarsa, H. (2014). Does gamification work? – a literature review of empirical studies on gamification. In *2014 47th hawaii international conference on system sciences* (p. 3025-3034).

Hanus, M. D., & Fox, J. (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*, 152-161. Retrieved from https://www.sciencedirect.com/science/article/pii/S0360131514002000 doi: https://doi.org/10.1016/j.compedu.2014.08.019

Hoyos, J., Abdelkareem, R., Mujahid, S., Shihab, E., & Bedoya, A. E. (2021). On the removal of feature toggles. *Empirical Software Engineering (EMSE)*, *26*, 1-26.

Johansson, M., & Ivarsson, E. (2014). *An experiment on the effectiveness of unit testing when introducing gamification* (Unpublished master's thesis).

Johnson, B., Song, Y., Murphy-Hill, E., & Bowdidge, R. (2013). Why don't software developers use static analysis tools to find bugs? In *2013 35th international conference on software engineering (icse)* (p. 672-681). doi: 10.1109/ICSE.2013.6606613

Johnson, D., Deterding, S., Kuhn, K.-A., Staneva, A., Stoyanov, S., & Hides, L. (2016). Gamification for health and wellbeing: A systematic review of the literature. *Internet Interventions*, *6*, 89-106. Retrieved from https://www.sciencedirect.com/science/article/pii/S2214782916300380 doi: https://doi.org/10.1016/j.invent.2016.10.002

Jorgensen, N. (2001). Putting it all in the trunk: incremental software development in the freebsd open source project. *Information Systems Journal*, *11*(4), 321-336. Retrieved from https://onlinelibrary.wiley.com/doi/abs/10.1046/j.1365-2575.2001.00113.x doi: https://doi.org/10.1046/j.1365-2575.2001.00113.x

Khandelwal, S., Sripada, S. K., & Reddy, Y. R. (2017). Impact of gamification on code review process: An experimental study. In *Proceedings of the 10th innovations in software engineering conference* (p. 122126). New York, NY, USA: Association for Computing Machinery. Retrieved from https://doi.org/10.1145/3021460.3021474 doi: 10.1145/3021460.3021474

Koivisto, J., & Hamari, J. (2014). Demographic differences in perceived benefits from gamification. *Computers in Human Behavior*, *35*, 179-188. Retrieved from https://www.sciencedirect.com/science/article/pii/S0747563214001289 doi: https://

doi.org/10.1016/j.chb.2014.03.007

Laukkanen, E., Itkonen, J., & Lassenius, C. (2017). Problems, causes and solutions when adopting continuous deliverya systematic literature review. *Information and Software Technology*, *82*, 55-79. Retrieved from https://www.sciencedirect.com/science/article/pii/S0950584916302324 doi: https://doi.org/10.1016/j.infsof.2016.10.001

Meder, M., Plumbaum, T., Raczkowski, A., Jain, B., & Albayrak, S. (2018). Gamification in e-commerce: Tangible vs. intangible rewards. In *Proceedings of the 22nd international academic mindtrek conference* (p. 1119). New York, NY, USA: Association for Computing Machinery. Retrieved from https://doi.org/10.1145/3275116.3275126 doi: 10.1145/3275116.3275126

Mekler, E. D., Brhlmann, F., Tuch, A. N., & Opwis, K. (2017). Towards understanding the effects of individual gamification elements on intrinsic motivation and performance. *Computers in Human Behavior*, *71*, 525-534. Retrieved from https://www.sciencedirect.com/science/article/pii/S0747563215301229 doi: https://doi.org/10.1016/j.chb.2015.08.048

Moldon, L., Strohmaier, M., & Wachs, J. (2021, may). How gamification affects software developers: Cautionary evidence from a natural experiment on github. In *2021 ieee/acm 43rd international conference on software engineering (icse)* (p. 549-561). Los Alamitos, CA, USA: IEEE Computer Society. Retrieved from https://doi.ieeecomputersociety.org/10.1109/ICSE43902.2021.00058 doi: 10.1109/ICSE43902.2021.00058

Nacke, L., & Deterding, S. (2017, 01). The maturing of gamification research. *Computers in Human Behavior*, *71*, 450-454. doi: 10.1016/j.chb.2016.11.062

Neto, P. S., Medeiros, D. B., Ibiapina, I., & da Costa Castro, O. C. (2019). Case study of the introduction of game design techniques in software development. *IET Software*, *13*(2), 129-143. Retrieved from https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/iet-sen.2018.5149 doi: https://doi.org/10.1049/iet-sen.2018.5149

Pedreira, O., Garcia, F., Brisaboa, N., & Piattini, M. (2014, 01). Gamification in software

engineering a systematic mapping. *Information and Software Technology*, *57*. doi: 10.1016/j.infsof.2014.08.007

Prause, C. R., & Jarke, M. (2015). Gamification for enforcing coding conventions. In *Proceedings of the 2015 10th joint meeting on foundations of software engineering* (p. 649660). New York, NY, USA: Association for Computing Machinery. Retrieved from https://doi.org/10.1145/2786805.2786806 doi: 10.1145/2786805.2786806

Rodrguez, P., Haghighatkhah, A., Lwakatare, L. E., Teppola, S., Suomalainen, T., Eskeli, J., ... Oivo, M. (2017). Continuous deployment of software intensive products and services: A systematic mapping study. *Journal of Systems and Software*, *123*, 263-291. Retrieved from https://www.sciencedirect.com/science/article/pii/S0164121215002812 doi: https://doi.org/10.1016/j.jss.2015.12.015

Romano, J., Kromrey, J. D., Coraggio, J., Skowronek, J., & Devine, L. (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.*

Sailer, M., Hense, J. U., Mayr, S. K., & Mandl, H. (2017). How gamification motivates: An experimental study of the effects of specific game design elements on psychological need satisfaction. *Computers in Human Behavior*, *69*, 371-380. Retrieved from https://www.sciencedirect.com/science/article/pii/S074756321630855X doi: https://doi.org/10.1016/j.chb.2016.12.033

Scherr, S. A., Elberzhager, F., & Holl, K. (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)* (p. 247-251).

Seaborn, K., & Fels, D. I. (2015). Gamification in theory and action: A survey. *International Journal of Human-Computer Studies*, *74*, 14-31. Retrieved from https://www.sciencedirect.com/science/article/pii/S1071581914001256 doi: https://doi.org/10.1016/j.ijhcs.2014.09.006

Shihab, E., Bird, C., & Zimmermann, T. (2012). The effect of branching strategies on

software quality. In *Proceedings of the acm-ieee international symposium on empirical software engineering and measurement* (p. 301310). New York, NY, USA: Association for Computing Machinery. Retrieved from https://doi.org/10.1145/2372251.2372305 doi: 10.1145/2372251.2372305

Singer, L., & Schneider, K. (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)* (p. 5-8). doi: 10.1109/GAS.2012.6225927

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

Toh, M. Z., Sahibuddin, S., & Mahrin, M. N. (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* (p. 173177). New York, NY, USA: Association for Computing Machinery. Retrieved from https://doi.org/10.1145/3316615.3316619 doi: 10.1145/3316615.3316619

Trockman, A., Zhou, S., Kstner, C., & Vasilescu, B. (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)* (p. 511-522). doi: 10.1145/3180155.3180209

Trockman, A., Zhou, S., Kstner, C., & Vasilescu, B. (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)* (p. 511-522).

Vasilescu, B. (2014). Human aspects, gamification, and social media in collaborative software engineering. In *Companion proceedings of the 36th international conference on software engineering* (p. 646649). New York, NY, USA: Association for Computing Machinery. Retrieved from https://doi.org/10.1145/2591062.2591091 doi: 10.1145/2591062.2591091

Wessel, M., de Souza, B. M., Steinmacher, I., Wiese, I. S., Polato, I., Chaves, A. P., &

Gerosa, M. A. (2018, November). The power of bots: Characterizing and understanding bots in oss projects. *Proc. ACM Hum.-Comput. Interact.*, *2*(CSCW). Retrieved from https://doi.org/10.1145/3274451 doi: 10.1145/3274451

Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics Bulletin*, *1*(6), 80–83. Retrieved from http://www.jstor.org/stable/3001968

Yamakami, T. (2013). Gamification literacy: Emerging needs for identifying bad gamification. In J. J. J. H. Park, J. K.-Y. Ng, H.-Y. Jeong, & B. Waluyo (Eds.), *Multimedia and ubiquitous engineering* (pp. 395–403). Dordrecht: Springer Netherlands.

Zhao, Y., Serebrenik, A., Zhou, Y., Filkov, V., & Vasilescu, B. (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* (p. 6071). IEEE Press.

Zimmermann, T., & Casanueva Arts, A. (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)* (p. 13-23).