Towards Improving Bug Tracking Systems with Game Mechanisms

Rafael Lotufo, Leonardo Passos, Krzysztof Czarnecki
University of Waterloo
{*rlotufo, lpassos, kczarnec*}*@gsd.uwaterloo.ca*

***Abstract*—Low bug report quality and human conflicts pose challenges to keep bug tracking systems productive. This work proposes to address these issues by applying game mechanisms to bug tracking systems. We investigate the use of game mechanisms in Stack Overflow, an online community organized to resolve computer programming related problems, for which the improvements we seek for bug tracking systems also turn out to be relevant. The results of our Stack Overflow investigation show that its game mechanisms could be used to address these issues by motivating contributors to increase contribution frequency and quality, by filtering useful contributions, and by creating an agile and dependable moderation system. We proceed by mapping these mechanisms to open-source bug tracking systems, and find that most benefits are applicable. Additionally, our results motivate tailoring a reward and reputation system and summarizing bug reports as future directions for increasing the benefits of game mechanisms in bug tracking systems.**

I. Introduction

Although efficient bug resolution and contributor recruitment are the main values brought to open-source communities by bug tracking systems [1], the significant amount of *misleading bug reports* and *conflicts among contributors* pose a threat to both these benefits [2], [3], [4], [5]. The inherent uncertainty of failures and contributor inexperience are the main culprits for the existence of misleading bug reports: many reports are found to be irreproducible and to have missing, contradicting, or erroneous claims [2], [3], [6], [7], [8]. Such reports diminish bug resolution productivity and jeopardize a project's timeliness, as developers often base their efforts on false information. As for conflicts, they are often caused by developers asserting their status and superiority over other contributors, effectively discouraging contributor participation and limiting a community's growth [6], [5].

In order to address misleading bug reports and improve resolution productivity, previous works show that bug reports need **quality improvement**, with more complete and concrete descriptions of failures, and less uncertainty [3], [2], [7]. Furthermore, acknowledging that the variety of contributor expertise in a large contributor base will lead to the existence of bug reports of different qualities, a **filtering mechanism** should identify useful contributions [9], [10], [11].

To allow bug tracking systems to take advantage of a large contributor base, open-source projects must have zero-tolerance towards rude behaviour, and counter-productive disputes among members, such as 'holy wars' [12] and 'flaming' [6], should also be discouraged [5]. Bug tracking systems must, therefore, provide a **conflict management system** that can rapidly detect and react to inadequate interactions. In addition to this reactive conflict management approach, the ecosystem would benefit from an explicit **motivation system**, actively stimulating increases in contributor participation.

To address misleading bug reports and conflicts, this work proposes to improve bug report quality, filtering, moderation, and contributor motivation in open-source bug tracking systems through the use of *game mechanisms*. Game mechanisms use reputation and rewards systems to encourage desirable behaviour and have previously been shown to increase trust, motivate participation, and push participants to new levels of achievements in online communities [13], [14], [15], [16]. Our work is also motivated by a recent trend in software development ecosystems to adopt reputation systems. Visual Studio, for example, now has achievements, badges, and leaderboards; Launchpad calculates a reputation score for users based on the quantity of contributions, but does not offer rewards. We are not aware, however, of studies showing the benefits of such reputation systems for software development ecosystems.

To evaluate the effects of game mechanisms on bug tracking systems, we investigate Stack Overflow, a successful ecosystem that uses game mechanisms extensively and, just as bug tracking systems, is organized to be a communication medium for the resolution of software issues: its members post specific programming-related problems, expecting solutions from the community. The improvements we seek for bug tracking systems turn out to be relevant to Stack Overflow: questions, just as bug reports, need to be resolved, therefore high-quality contributions are also important; the large number of contributors and contributions also demand conflict management and capabilities to identify useful contributions; finally, contributor motivation is essential for its success, as contributors are responsible for question resolution. With the results and insights of Stack Overflow, we proceed showing how bug tracking systems, given their unique characteristics, could effectively benefit from game mechanisms.

Our analysis shows that Stack Overflow's game mechanisms implement a formal meritocracy, which is well accepted by its developer community, and is effective: they lead to quality improvement and motivate increases in contribution frequency of up to three times. Rewards also increase the number of competing answers for best contributions by as much as 50%, thereby improving resolution rates. We

also find that developers are highly interested in gaining moderation privileges rather than other privileges. Finally, the moderation system enabled by awarding moderation privileges to the community allows quick detection of inappropriate contributions.

Our comparison between Stack Overflow and bug tracking systems show that, despite their differences, they share the conditions we have identified for the game mechanisms to produce their benefits. Our analysis leads us to conclude that the benefits of game mechanisms can be achieved in bug tracking systems, provided its respective community is open to a formal merit-based reputation system. We also analyze current bug report voting mechanisms that could be used by game mechanisms to identify useful bug reports and find that duplicate bug reports should be used to complement such signal.

This work provides four contributions. To our knowledge, we are the first to propose the use of Stack Overflow's game mechanisms to improve bug tracking systems. Second, this is the first systematic empirical investigation of a developer community in a formal merit-based collaborative ecosystem, showing how such a meritocratic system can not only motivate participation but also improve contributions. Third, we are the first to investigate current voting mechanisms in bug tracking systems and how they can be used with game mechanisms. Finally, motivated by the results of our investigations, we identify two important directions for extracting maximum value from game mechanisms: tailoring a reputation and reward system to a community and summarizing bug reports.

This paper is organized as follows: Section II presents background for our work. Sections III and IV present our methodology and research questions. Section V presents findings for our research questions, Section VI maps these results to bug tracking systems, and Section VII presents important directions for future work. Sections VIII, IX, and X present related work, threats to validity, and conclusions.

II. Background

A. Bug Tracking Systems

Bug tracking systems have been used since 1970 as a collaboration ecosystem to report and resolve bugs. Typically, bug reports are filed by users when they encounter a system failure. When submitted, bug reports generally contain a summary of the failure, the environment settings in which it was triggered, the steps to reproduce it, and possibly other diagnostic information.

Once a bug report is created, the development team will try to diagnose and confirm the failure, only then proceeding with its correction [17]. Bug reports, in general, can have three different resolutions: *fixed*, when the failure has been confirmed and corrected; *won't fix*, when the failure cannot be reproduced or when there is no agreement that it is a relevant or real failure; and *duplicate*, when the failure has already been reported by another unresolved bug report. Unresolved bug reports remain *open*.

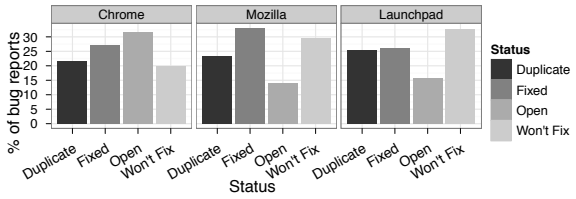

Figure 1: Bug report resolution status

Bug tracking systems are used in both closed and open-source projects. Opposed to most closed-source projects, in open-source they are not only accessible, but are also open to outside participation, and are often used by projects to crowdsource quality assurance efforts for development, alpha, beta, and final releases of their products.

Given the popularity of open-source projects and the market demand for open-source developer skills, there has been an increase in contributors willing to participate in such projects [18], [19]. As bug tracking systems have been found to be the environment in which contributors start learning about projects before moving on to development [20], they are full of new, inexperienced contributors, thus ripe with misleading bug reports and conflicts [2].

1) Misleading Bug Reports: As opposed to useful bug reports, misleading reports are those that after inspection, do not result in changes to the target system—are not fixed—and are often considered diverters of valuable contributor attention. Figure 1 presents the resolution of bug reports found in the Chrome, Mozilla, and Launchpad bug tracking systems, and shows how common misleading bug reports are: in average (mean) only 30% of bug reports are fixed—these findings are also supported by others [21], [22].

Duplicate bug reports also present a challenge to bug tracking systems. While it has been shown that they provide more information about failures and could potentially be used to improve resolution time, many projects ignore duplicate bugs, marking them as closed [22]. Indeed, from the point of view of the developer or bug triager, closing a bug as duplicate is extremely convenient. This causes two problems: developers don't reap the potential benefits of additional information brought by duplicates; and users refrain from submitting new bug reports, as they might feel their contributions are being ignored [22].

Comments are also a crucial component of bug reports, and are used to clarify and correct information in bug reports, and to coordinate bug resolution [8], [21]. Reading through comments, however, is not a trivial task, as each one often introduces more information and uncertainty [21], [6]. In fact, Ko [21] shows that, as relevant information is scattered throughout the conversation decision-making is negatively affected. As a result, highlighting important comments should facilitate bug resolution.

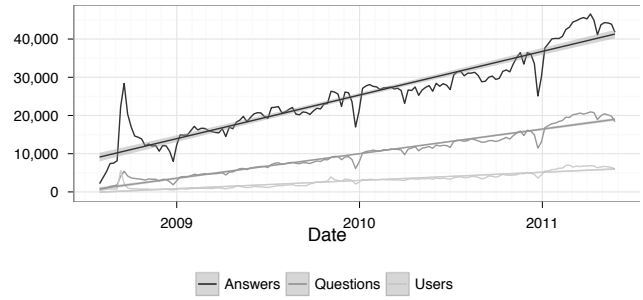

Figure 2: Number of new answers, questions, users per week

2) Conflicts: Bug resolution is ripe with conflicts. Such conflicts are fuelled by the uncertainty of failures, the conflicts of interests between developers and users, power disputes among developers, and the discussion medium itself [21], [23], [6]. In much of these cases, well-known developers often ignore or disregard the contributions of less well-known developers, sending them large amounts of unfair critique. As this can discourage participation from beginners [5], [6], projects seeking to grow its community need to manage such conflicts.

B. Game Mechanisms

McGonigal [13] studies games and finds that competition, great challenges, collaboration, and sense of accomplishment are the factors that make games so compelling. McGonigal argues that it is possible to make tasks more compelling and rewarding by introducing small changes—mechanisms—that add game-like characteristics to the tasks.

Enriching tasks with game mechanims can be used not only for personal a benefit, but also for crowd-sourcing. The Guardian's "Investigate Your MP" game, for instance, successfully motivated thousands of people to examine hundreds of thousands of documents, uncovering serious irregularities in British Parliamentary expense claims.[1]

C. Stack Overflow

Stack Overflow is a web application created for developers that provides a collaborative ecosystem aimed at the resolution of specific computer programming problems. Users post questions about these problems, and the community posts answers attempting to resolve them. It is considered one of the most successful technical question-and-answer applications today [14]. Figure 2 shows how Stack Overflow's usage has grown since its origins in mid-2008. It shows that the number of new answers, questions, and users per week follows an almost linear progression, indicating the total number of answers, questions, and users increases quadratically over time. As of this writing, Stack Overflow had more than 400*k* users, 1.7 *mil.* questions, and over 3.5 *mil.* answers.

[1]Source: http://mps-expenses.guardian.co.uk/

Stack Overflow is appropriate for this investigation, as its community is formed of software developers and its questions have some similarity to bug reports: in both cases, users report problems asking the community to discuss and propose solutions. Stack Overflow's community corroborates this claim, as they consider a "'*language-specific programming problem' ... that exists in code and that can be resolved with correct code*" as the most appropriate type of question the community can help resolve.[2]

Arguably, the game mechanisms used in Stack Overflow are one of the top reasons for its success [14]. Registered users start with 1 reputation point and are able to gain more as their contributions are recognized by the community. 'Recognizable' contributions are primarily those of questions and answers: users recognize useful questions and answers by voting, which rewards five and ten reputation points to the submitters of voted questions and answers. Furthermore, the owner of a question can 'accept' an answer as the best, rewarding 15 reputation points to its owner and two reputation points to themselves for selecting a best answer. By doing so, the owner indicates his satisfaction and considers the question has been *resolved*. Users can also *down*-vote questions and answers they consider to be invalid or incorrect, decreasing users' reputation by two points and their own by one—taking one point from voters attempts to assure users only down-vote contributions they really believe have no value. Users can also comment on questions and answers to suggest improvements or ask for clarification. Furthermore, similar to wiki posts, questions and answers can be edited, allowing other contributors to improve them. Finally, users can offer larger amounts of reputation points—*bounties*—for questions they are not satisfied with the resolution of. Users who offer bounties—which are multiples of 50 reputation points—pay them from their own reputation points.

Not all users can perform all these actions, however: they have to *earn each privilege*. New registered users, for instance, can only post questions and answers. They cannot vote, comment, or edit others' posts. Privileges are earned by accumulating reputation points and achieving *reputation levels*: 15 points to vote up; 50 to comment; 125 to vote down, etc. Users need 2000 points to edit others' posts; 3000 points are needed to vote to close or reopen questions.[3] By awarding contributors with privileges as they show their value to the community, Stack Overflow implements a *meritocracy*.

As can be seen, Stack Overflow uses a reputation and rewards systems that aims to spur competition and a sense of accomplishment. Users are expected to compete for the best answers and questions, while earning reputation and privileges should give users a sense of accomplishment and reward. Finally, having well planned reputation levels should challenge users to reach, for example, the 10000 reputation level.

[2]Source: http://meta.stackoverflow.com/questions/12373

[3]Find full set of privileges at http://stackoverflow.com/privileges

III. Methodology

We select three of Stack Overflow's game mechanisms that can be easily added to bug tracking systems and pose five research questions to assess if they can achieve the goals of increasing participation and contribution quality and of improving filtering and moderation in Stack Overflow. The results of our analysis enable us to identify the *conditions* that allow such mechanisms to achieve the goals in Stack Overflow. Finally, we assess if these conditions are also valid in bug tracking systems, thereby allowing game mechanisms to produce their benefits.

The mechanisms we investigate are: *rewarding reputation points for good contributions* (M1), *reducing a user's reputation points for poor contributions* (M2), and *awarding privileges to users as they reach reputation levels* (M3).

As these mechanisms simply reward and penalize users based on the quality of their contributions as judged by the community, these mechanisms can be easily added to bug tracking systems. For bug tracking systems to implement M1 and M2, they need only to allow users to recognize and down-vote bug reports and comments, the equivalents of Stack Overflow's questions and answers—questions and bug reports describe a software issue for resolution, while answers and comments are posted by contributors to share their work and knowledge and help resolve the issue. In fact, most bug tracking systems already allow users to vote for bug reports they consider important to be fixed. To implement M3, bug tracking systems need only to reward existing privileges to users as they reach certain reputation levels.

A. Data Sets

To answer our research questions, we analyze Stack Overflow over its three-year lifetime, using the StackApps API, provided by Stack Overflow, to retrieve usage information. Due to restrictions imposed by the API on download allowances, we used simple random sampling to download 80% of the 1.7 *mil.* questions and all 3.5 *mil.* answers for those questions. Finally, we randomly sampled $60k$ users and downloaded their entire contribution timeline—all of the questions, answers, comments, and edits they had posted.

Our data sets for bug tracking systems comprise of $12k$ bug reports for Android from Nov-2007 to May-2011, $50k$ bug reports for Chrome from Aug-2008 to Jun-2010, $50k$ for Launchpad from Jan-2008 to May-2011, and $140k$ for Mozilla from Jan-2008 to Jun-2010.

B. Estimating Reputation Over Time

As the StackApps API does not provide the reputation of users at certain points in time, which is a crucial information for many of our questions, we estimate this value using the number of up and down votes users have received for their questions and answers, and the number of times their answers were accepted as the best solution. Given a user u, his reputation at time t is estimated as follows:

$$R_{u,t} = 10V_{\uparrow}(A_{u,t}) + 5V_{\uparrow}(Q_{u,t}) - 2V_{\downarrow}(A_{u,t} \cup Q_{u,t}) + 15V_{\checkmark}(A_{u,t})$$

where $A_{u,t}$ and $Q_{u,t}$ are the sets of all answers and questions posted by user u until time t; functions $V_{\uparrow}$ and $V_{\downarrow}$ return the number of up and down votes of sets of questions and answers; and function $V_{\checkmark}$ returns the number of accepted answers for a set of answers. As estimating a user's reputation requires the data for all questions and answers posted by a user, for questions requiring this estimation, we limit our investigations to users in our timeline data set.

C. Correlations

When comparing certain metrics dependant on reputation, we discretize reputation scores using the reputations levels selected by Stack Overflow to award privileges: 250, 500, 1000, 1500, 2000, 3000, 5000, 10000, 15000 and 20000. When testing for rank correlations, we use the non-parametric Spearman's test; to check if two independent samples contain equally large values, we rely on the non-parametric MannWhitney U test [24]. Although all our statistical tests were significant, for each test we present the p-value to support our claims.

IV. Research Questions

We pose five research questions to assess if game mechanisms in Stack Overflow can be used to address the issues we identified in bug tracking systems.

A. Motivating Participation

In bug tracking systems, community participation is key to the identification and resolution of bugs. This motivates our first question:

RQ 1: Are game mechanisms in Stack Overflow effective in increasing user contribution frequency in a development ecosystem?

B. Conflict Management

Conflict management is an important requirement in open-source discussion forums [5] such as bug tracking systems. As moderation should allow inadequate interactions to be detected as soon as possible and should not depend on the attention of few moderators, we ask:

RQ 2: Does the reputation and rewards system contribute to create an agile and dependable moderation system?

C. Improving Contribution Quality

To improve the chances of having bugs fixed, bug tracking systems must receive high-quality contributions. To investigate whether rewards are effective in increasing contribution quality, we ask:

RQ 3: Does the reward system drive contributors to post better answers?

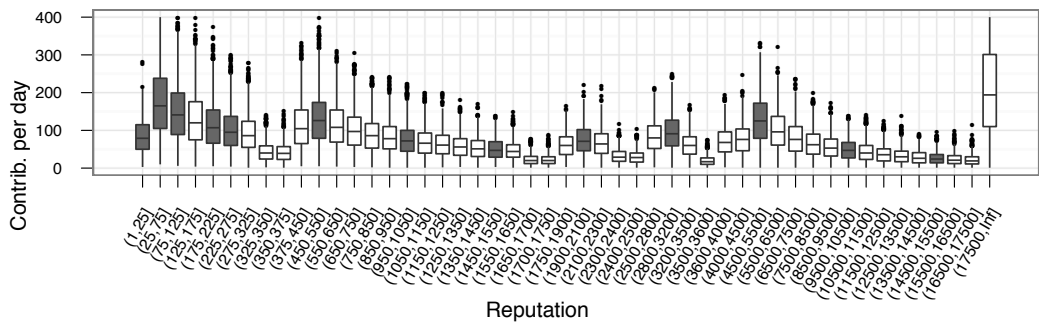

Figure 3: Contribution frequency by reputation

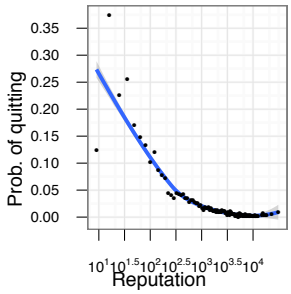

Figure 4: Quit probability

Peer-reviewing can also bring further improvements to bug reports. We therefore ask:

RQ 4: Does the reward system increase peer-reviewing frequency?

D. Filtering Contributions

Filtering useful contributions in bug reports is important so that users are not distracted from looking at misleading information. We seek, therefore, a signal that accurately identifies useful contributions. Two candidate signals are the reputation of the contribution's owner and the number of recognitions a contribution receives. This leads us to our final research question:

RQ 5: Are reputation or recognition good signals for filtering useful contributions?

V. Results

RQ 1: Our first research question investigates if rewards are a good motivator for user participation. In order to answer this question, we look for usage patterns indicating an increase in participation just before or after reputation levels that award privileges. Figure 3 presents the number of answers and questions submitted to Stack Overflow per day, by user reputation, and shows increases in contribution frequency just before reputation scores that award privileges—reputation ranges in which privileges are awarded are coloured in grey. For example, contribution frequency is tripled—from 40 to 120—to earn privileges rewarded at 500 points. Our data, however, does not allow us to infer if users increase contribution frequency as a sprint to gain privileges, or if users are motivated to increase contribution frequency as a result of receiving such privileges. Nevertheless, this correlation indicates that users are interested in gaining privileges, and as a result, participation frequency increases.

Besides showing strong indications of the influence of rewards in increasing contribution frequency, these results also show that not all reputation levels awarding privileges results in increases in contribution frequency. This suggests the importance of setting up reputation levels at intervals and with privileges that are compelling to the community. Curiously, all reputation levels that had an increase in contribution frequency reward privileges related to reviewing and moderating other user's contributions: commenting on other's posts; retagging questions; editing other's posts, voting to approve editions; voting to close or reopen questions; voting to approve tag wiki edits. Reputation levels awarding privileges not related to moderating or reviewing did not show an increase in contribution frequency—for example, reduced advertising, voting to close one's own questions, creating new tags. This correlates with Bergquist's claims that contributors in open-source often seek reputation in order to assert relationships of power over lower-reputation users [6].

Besides the influence of privilege rewards on contribution frequency, we also find users with more reputation and privileges are less likely to quit than those with fewer privileges. We consider a user has quit, if he has not posted a question, answer, comment, or edit since the previous 60 days from this investigation. Figure 4 shows how the probability of users quitting decreases with their reputation. We calculate the probability of users quitting at reputation r as the number of users that quit at that reputation, divided by the number of users that ever reached reputation r. As Figure 4 shows, there is a steep decrease in the probability of quitting from 10 to 300 reputation points, and then a stabilization of this probability below 5% after 300 points. Interestingly, there is a slight increase in the probability of quitting for users with more than 10000 points, suggesting a decrease in user motivation after achieving such high reputation and privileges.

These results suggest that users feel more committed to the community as they gain experience, privileges, and reputation; however the correlation between reputation and decrease in probability of quitting does not indicate that such a decrease is *caused by* reputation or rewards. Figure 3 indicates that, **as users are interested in gaining privileges, rewarding users with reputation and privileges increases contribution frequency**. Without rewards, user contributions would likely decrease over time, without future increases.

RQ 2: For our next question, we investigate if the reputation and reward system contribute to create an agile and dependable moderation system—qualities we have defined

in Section IV-B. Stack Overflow's community is advised to identify and moderate inappropriate contributions: contributions that do not adhere to the community's principles, are offensive, are not the type of questions or answers the community expects, or is a duplicate question or answer.

Stack Overflow's moderation system is tightly tied to its rewards mechanism. As higher reputation offers users more moderation privileges, there is naturally a large number of users with low moderation privileges, and then a progressively smaller number of users with higher moderation privileges—a *population pyramid* of privileges. The first level of moderation is the flagging of contributions, an action that every user with 15 points or more can perform, and brings a flagged contribution to the attention of high reputation moderators. A second moderation level is the down-vote, a judgement that every user with more than 125 reputation points can make. The down-vote is effective because it discourages users from posting low-quality contributions. The next moderation level is available to users with more than 3000 points, allowing them to vote to *close*, *reopen* or *migrate* questions to other Stack Exchange sites. These are the users who will be notified when the first level moderators flag contributions for attention. Next, users with 10000 points can vote to *delete* questions and access other moderation tools.

Although Stack Overflow does not provide data on contributions that suffered moderation, thereby not allowing us to assess its performance, we argue that **its rewards mechanisms, by building a population pyramid of privileges, enable a dependable moderation system**. The system can rely on the attention of the large user base with low moderation privileges to take the first moderation actions on inappropriate contributions, only then relying on a smaller base of high-reputation moderators to take further actions. We also argue that **it is agile, since flagging is available to the large majority of users, increasing the probability of users quickly detecting improper interactions.**

RQ 3: Our next question asks if rewards drive contributors to post better answers. We find that questions offering higher rewards receive more answers and have a higher likelihood of being resolved. Questions offering bounties—large amounts reputation—receive 50% more answers than non-bounty questions—bounty questions receive an average of 3.34 answers (median of 3), while non-bounty questions receive only 2.53 (median of 2)—and bounty questions are 10% more likely to be resolved than non-bounty questions—all claims are substantiated by the Mann-Whitney U test, with p-values $< 2.2 \times 10^{-16}$.

We also find that questions with higher reward-potential also receive more answers and are more likely to be resolved, even when not offering bounties. Considering a question's reward is the sum of the number of recognitions its answers has received, a correlation rank (ρ) of 0.52 between this value and a question's number of recognitions shows that a question's number of recognitions is a good indication of its reward

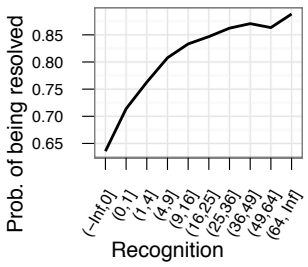

Figure 5: Resolution probability

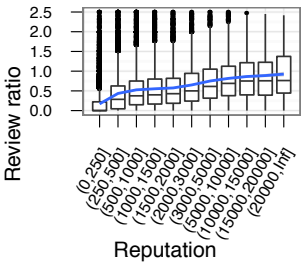

Figure 6: Review ratio

potential. Then, similar to bounty questions, non-bounty questions with more reward-potential receive more answers ($\rho = 0.34$), and are also more likely to be resolved ($\rho = 0.13$)—in all cases, the p-values were $< 2.2 \times 10^{-16}$. Figure 5 shows how the probability of a question being resolved increases with its reward-potential and confirms these findings.

These results indicate that resolution likelihood increases as a result of rewards attracting more participation. Increased participation, however, may spur a large number of low-quality answers. Instead, we find, that users restrain themselves from submitting low-quality answers to avoid receiving down-votes: there is a 63% chance of users deleting their own low-quality contributions that received 3 or more down-votes. Consequently, **as users are motivated to earn reputation points, rewarding good contributions and penalizing bad contributions is an effective means to improve contribution quality**.

RQ 4: Our next question asks if the reward system increases not only contribution but also peer-reviewing frequency. By peer-review, we consider contributions that are the result of an evaluation of answers and questions, with suggestions for improvement. In Stack Overflow, we consider all edits to questions and answers to be improvements. Comments are also used as an instrument of peer-reviewing: a sample of 400 comments chosen randomly shows that 52% of them are improvements to others' questions and answers, bringing alternative solutions, additional information, and explanations of why answers are incorrect. We therefore consider that, when comparing frequencies of comments and edits, more comments and edits implies more peer-reviewing.

We first find that users perform more peer-reviewing as they gain reputation. Here, we consider as peer-reviews all comments and edits made to answers or questions that a contributor did not ask or answer, and therefore has no stake in. We then calculate the *review ratio* as the number of reviews each user has posted in a week, divided by the number of answers and questions they posted in that same week, thereby effectively comparing the number of reviews to the number of answers and questions.

Figure 6 presents boxplots of the review ratio of users by

reputation, along with a LOESS regression, and shows that it increases with higher reputation. Similar assessment on individual users also finds that this correlation holds for over 85% of users. These results match Maslow's hierarchy of needs [25], suggesting that contributors with low reputation are interested in gaining more reputation: their efforts are focused on posting questions and answers. As users' reputations increase, they are ever more likely to review questions and answers they have no stake in, suggesting that they are more concerned in maintaining the system and evaluating other contributors' posts, instead of publishing their own.

More importantly, performing a similar analysis as in RQ 1 to look for increases in peer-reviewing frequency at reputation levels, we have found similar results to those shown in Figure 3, with increases in peer-reviewing frequencies for the exact same reputation levels in which we found increases in contribution frequencies in RQ 1, showing **that the rewards system also motivates increases in peer-reviewing frequency**.

The increases in peer-reviewing and contribution frequencies for the same reputation levels suggest that **peer-reviewing occurs as a result of the contribution process**. A high correlation between contribution and peer-reviewing rates—Spearman's $\rho = 0.69$, p-value $< 2.2 \times 10^{-16}$—supports this and indicates a dependency between the two rates: as users contribute, they evaluate their peer's contributions, offering suggestions for improvements and corrections. Users who do not evaluate previous contributions before posting their own and submit inappropriate contributions, such as duplicates, suffer the risk of being down-voted or flagged.

RQ 5: Our next question asks whether reputation or recognition can be used to filter useful contributions—questions and answers. In Stack Overflow, a useful question is one that is not misleading and will be resolved. A useful answer is one that contributes to resolving the question.

To evaluate whether reputation or recognition are good signals for identifying useful answers, we rely on an information-retrieval approach to evaluate the precision rates of using reputation and recognition for selecting the best answers. We calculate $P@n$ [26], as shown in Equation (1), to measure the percentage of questions q whose best answer can be found by looking at its top n answers, ranked by σ (reputation or recognition). As does Stack Overflow, we consider the answer accepted by the question's owner as the best answer.

$$P@n = \frac{|\{q \in \textit{questions} : \text{best\_answer}(q) \in \text{top}(n, q, \sigma)\}|}{|\textit{questions}|} \quad (1)$$

Figure 7 shows $P@n$ for questions which have an accepted answer. It compares the precision when ranking answers by number of recognitions and by user reputation at the time the question was posted. As shown, recognition has considerably higher precision rates compared to reputation: the answer with most votes is the best answer in around 70% of questions, compared to 50% for user reputation. This indicates that **both reputation and recognitions are good predictors of useful contributions, with recognitions still having as much as 20% higher precision**, considering the top answer.

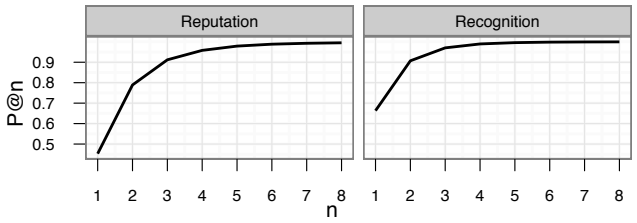

Figure 7: $P@n$ for reputation and votes

A. Summary

Our investigations have shown that in Stack Overflow, game mechanisms achieve the goals of increasing participation and quality and improving filtering and moderation. Our results have also identified the conditions that allow these game mechanisms to achieve these goals. Figure 8 summarizes these findings, with arrows linking each goal to the game mechanisms and conditions required to achieve it.

For the goal of increasing participation, RQ 1 shows that *rewarding users with privileges* (M3) increases user participation, provided the community is interested in *resolving software programming issues* (C1), is *motivated to earn reputation and privileges* (C2), and is *rewarded as users evaluate and recognize their contributions* (C3).

For quality improvement, as found in RQ 3, the mechanisms of *rewarding users for good contributions* (M1) and *penalizing them for poor contributions* (M2) increases contribution quality. This occurs because *users are motivated to earn reputation points* (C2) and will compete to send high-quality answers to gain recognition. Furthermore, in order to maintain users' interests in contributing, it is important that *the community evaluates and recognizes good contributions* (C3) so that users are rewarded. RQ 4 has also shown that, as *peer-reviewing occurs as a by-product of contributing* (C4), increases in participation frequency will trigger increases in the frequency of users improving contributions by commenting and editing.

As for identifying useful contributions, RQ 5 shows that, as *users routinely evaluate and recognize contributions they find to be useful* (C3), a question or answer's number of recognition offers a very good signal to judge its usefulness.

Finally, as seen in RQ 2, as *users evaluate and recognize good contributions* (C3), the well-crafted rewards system *creates a population pyramid of privileges* (M3) that enables an agile and dependable moderation system.

VI. Mapping back to bug tracking systems

We now evaluate if, once Stack Overflow's game mechanisms are applied to bug tracking systems, as described

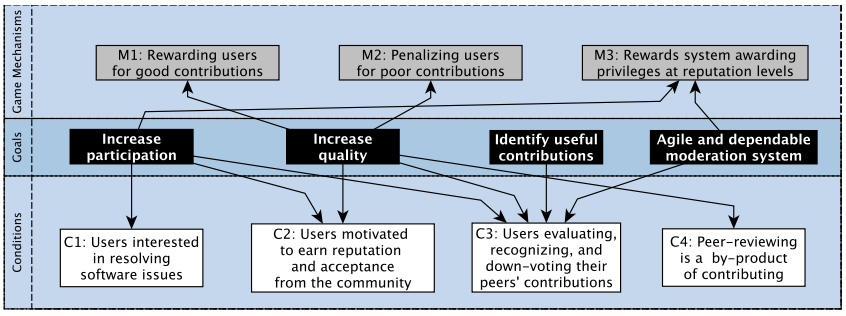

Figure 8: Conditions for game mechanisms to achieve goals

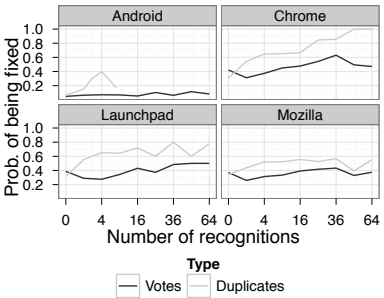

Figure 9: Fix-probability by recognitions

in Section III, they will produce similar results as in Stack Overflow, and increase participation and quality and improve filtering and moderation. To do so, we investigate if the conditions to achieve the goals (conditions C1 to C4) are valid for bug tracking systems.

A. Asserting Conditions for Bug Tracking Systems

First, users of bug tracking systems are likely to be *interested in solving software issues* (C1). To assert the remaining conditions, we analyze a random set of 100 bug reports from each project in our data sets. We find, similar to Breu [8], that at least 27% of comments in bug reports result from the *evaluation of other comments* (C3), asking for clarification and additional information. Evermore, the number of peer-reviews—users correcting and improving their peer's comments—increases with the number of comments—$\rho = 0.63$, p-value < 0.003—indicating that *peer-reviewing occurs as a by-product of contributing* (C4). Furthermore, users *evaluate and recognize both bug reports and comments* (C3): most bug tracking systems allow users to recognize, by voting, a bug report they consider important to be fixed; as for comments, we find that an average of 22% of comments recognize the validity and usefulness of other comments.

While conditions C1, C3, and C4 are valid for most bug tracking systems, condition C2 depends on a project and its community to accept a formal merit-based reputation system, and to be *motivated to earn reputation and privileges* (C2). We acknowledge that some projects and contributors might not be interested in such. Small teams, for example, in which members know each other well, might not be motivated to strive to earn reputation to differentiate themselves from others. The example of Stack Overflow, however, suggests that some developer communities should be interested.

B. Stack Overflow and Bug Tracking Systems Differences

Although Stack Overflow and bug tracking systems share many characteristics, there are two important differences that affect, but do not invalidate, the effects of game mechanisms.

Identifying Useful Bug Reports: There is a fundamental difference in how issues are resolved. In Stack Overflow, it is the community that decides the resolution of a question; in bug tracking systems, it is ultimately the most influential contributors—the minority of developers—that decide if a bug will be fixed, as it is they who accept changes into the code base. Consequently, developers might disagree with the high number of recognitions the community has given to particular bugs, rendering the number of recognitions a suboptimal signal for developers to identify important bug reports.

To verify this, we investigate if, similar to our findings for Stack Overflow shown in Figure 5, the probability of a bug being fixed increases with its number of recognitions. We evaluate two forms of bug recognition: votes and duplicate bug reports. Voting is a mechanism available in most bug tracking systems and is designed to allow users to vote for bug reports they consider important to be fixed. Duplicate bug reports can also be considered a form or recognition, as they imply that the bug has been triggered by more than one user. We find that, as shown in Figure 9, for all projects, except Android, the probability of a bug being fixed increases from 2 to 36 recognitions for both votes and duplicates—it seems Android is an outlier, perhaps due to its somewhat closed nature.[4]. Furthermore, the probability for a bug being fixed is higher for duplicates than for votes, suggesting that duplicates bring more evidence of a bug's relevance than a simple vote and casting more doubt on the harmfulness of duplicates [22].

This confirms that votes and duplicates can be used to identify useful bug reports. It also confirms, however, that developers have different opinions from users, since even bug reports with as much as 64 votes have only a 50% chance of being resolved. While this reduces the utility of such signal for developers, Launchpad has managed to successfully use vote recognition to help identify useful bug reports: when a bug report receives a certain minimal number of votes, it automatically changes status from 'unconfirmed' to 'confirmed', alerting contributors that a bug should be further investigated. Our findings show that, in addition to votes, duplicate bug reports should also be used to identify important bug reports.

Identifying Useful Comments: Another important difference lies in how knowledge is organized in questions and bug reports. In questions, knowledge is organized as independent

[4]http://www.theregister.co.uk/2011/04/12/google_says_android_both_open_and_closed/

answers, allowing users to read only the most useful ones. In bug tracking systems, contributions are organized as a conversation. As a result, although allowing comments to be recognized will identify good comments, users might still need to read previous comments to understand the context of a highly recognized one [14].

The impact these differences bring to the use of game mechanisms in bug tracking systems is limited: bug report recognition might not be an optimal signal for developers, but can still be used by the community, as is done in Launchpad; comment recognition will identify useful comments, but might not reduce the need to read all comments.

VII. Future Directions

In addition to mapping Stack Overflow's game mechanisms to bug tracking systems, we identify, through our findings, two directions for future work: tailoring a rewards system and summarizing bug reports.

Tailoring a Rewards system: As we have seen, condition C2 is dependent on a project and its community, implying that some communities might not accept such a merit-based reputation system. Android, for instance, as seen in Section VI-B, shows very little response to votes compared to Chrome, Launchpad and Mozilla, suggesting that current contributors are not so open to input from outside contributors. Addressing this requires an understanding of the unique motivations of different open-source contributors—payed, non-payed, enthusiast, beginner, developer, non-developer—in different projects, of different sizes, and guidelines on applying and tailoring a reputation and reward system according to contributor's profiles.

These guidelines should also consider that, to make game mechanisms effective, privileges offered as rewards need not only be compelling, but also useful in addressing misleading contributions and conflicts. Stack Overflow, as shown in RQ 1, offers privileges aimed exclusively at increasing participation, such as reduced advertising, and privileges exclusively aimed at increasing contribution quality, such as editing others' posts. Curiously, our results show that privileges that increase contribution quality are also the ones more capable of increasing contribution frequency.

Summarizing Bug Reports: As shown by Ko [21], the linearity of comments in bug tracking systems and the difficulty of finding important information in them negatively affects decision-making. As decisions require reliable information, misleading comments are another negative influence.

Bug reports providing summaries of important, reliable information about a failure would ease locating and reasoning about such information. As we have discussed, however, simply identifying useful comments is not enough, as readers will loose context. A future direction of work could experiment in using non-conversation forms of contributions, similar to answers, to facilitate summarization. For instance, comments such as "this bug also occurs in version 2.10", or "rebooting the computer solves the problem", could be promoted into a list of important diagnostic information and possible solutions according to how they are recognized by the community, effectively summarizing the bug.

VIII. Related Work

Mamykina et al. [14] qualitatively and quantitatively characterize Stack Overflow. Their quantitative analysis focuses on showing that questions are answered very quickly. Through interviews with Stack Overflow's design team and users, they find that Stack Overflow was created with the intention of creating productive competition to increase participation and quality. Our work complements Mamykina in providing strong statistical evidence that such mechanisms succeeded in increasing participation and quality and other improvements. More importantly, we focus on studying the effects of Stack Overflow's game mechanisms on a *developer community*, and find that they prefer privileges that allow them to review or moderate their peers' contributions and that the more reputation users have, the more they will review their peers' contributions. Finally, we provide an analysis of how these mechanisms, applied to bug tracking systems, should resolve many of their current issues.

Guo [9] and Hooimeijer [11], study the characteristics of bug reports that are chosen to be fixed by developers. They find that developers choose to fix bugs opened by well-known contributors, and that severity is an important factor. Their prediction model achieves an accuracy rate of around 60%. We find that recognition as votes and duplicates can be used as an additional signal to detect important bug reports.

Previous work in improving bug report quality [3], [27] uses natural language processing, machine learning, and heuristics, to detect the lack of steps to reproduce, traces and attachments. This automated approach, however, is not able to perform a deeper analysis, such as can be done by human evaluation. In contrast, our work accepts the inevitability of poor contributions, and addresses the problem through the use of game mechanisms to encourage contribution improvement.

IX. Threats to Validity

Internal Validity: Our model for estimating a user's reputation assumes that all votes for a post occur within the first 24 hours of its creation. This is corroborated by our finding that 98% of posts receive all but two of its votes within its first 24 hours. Still, Stack Overflow does not provide information on reputation points awarded by bounties or edits, nor on reputation points lost by down-votes. Nevertheless, as bounties are uncommon (only 2% of questions have offered bounties), edits award only two points, and down-votes cost only one point, we consider our estimation of reputation accurate enough for the purposes of this work.

Although we claim that rewards are the cause of increased participation and competition, we recognize that we have nothing but strong correlations and indications of such

causation. Other, unknown factors, might alternatively cause increases in participation and higher number of answers for questions with higher reward potential.

External Validity: We do not perform an experimental evaluation of using game mechanisms on bug tracking systems. Considering the challenges of setting up a valid experiment to perform an extensive evaluation, we consider Stack Overflow and its open data to currently be the best existing surogate for a bug tracking system with game mechanisms, as it is an organic ecosystem of thousands of software developers participating due to their real needs, and focused on resolving real software issues. We have carefully identified Stack Overflow's and bug tracking systems' similarities and differences and shown that the differences do not invalidate the applicability of game mechanisms to bug tracking systems.

X. Conclusion

We investigate if Stack Overflow's game mechanisms are effective in increasing contribution quality and participation, in filtering contributions, and creating an agile and dependable moderation system. This investigation shows that the game mechanisms are effective in such capabilities. When mapping these mechanisms to bug tracking systems, we find that, despite their differences, by applying a formal reputation and rewards system to current open-source bug tracking systems, the benefits of increasing contribution frequency, of improving contribution quality, and of moderation should be readily accessible.

Our work also motivates directions for future work to further improve bug tracking systems and maximize the effects of game mechanisms: by tailoring a reputation and reward system and summarizing bug reports using recognitions.

References

[1] A. J. Ko and P. K. Chilana, "How power users help and hinder open bug reporting," *CHI*, 2010.

[2] J. Sun, "Why are bug reports invalid?" *ICST*, 2011.

[3] T. Zimmermann, R. Premraj, N. Bettenburg, S. Just, A. Schrter, and C. Weiss, "What makes a good bug report?" *TSE*, 2008.

[4] C. Reis, "An overview of the software engineering process and tools in the Mozilla project," *Workshop on Open Source Software*, 2002.

[5] K. Fogel, *Producing Open Source Software – How to Run a Successfull Free Software Project*. O'Reilly Media, 2010.

[6] M. Bergquist and J. Ljungberg, "The power of gifts: organizing social relationships in open source communities," *ISJ*, 2001.

[7] P. Schugerl, J. Rilling, and P. Charland, "Mining Bug Repositories—A Quality Assessment," *CIMCA*, 2008.

[8] S. Breu, R. Premraj, and J. Sillito, "Information needs in bug reports: improving cooperation between developers and users," *CSCW*, 2010.

[9] P. J. Guo, T. Zimmermann, N. Nagappan, and B. Murphy, "Characterizing and predicting which bugs get fixed," *ICSE*, 2010.

[10] C. Weiss, R. Premraj, T. Zimmermann, and A. Zeller, "How Long Will It Take to Fix This Bug?" *MSR*, 2007.

[11] P. Hooimeijer and W. Weimer, "Modeling bug report quality," *ASE*, 2007.

[12] D. Cohen, "On holy wars and a plea for peace," *IEEE Computer*, 1981.

[13] J. McGonigal, *Reality Is Broken: Why Games Make Us Better and How They Can Change the World*. Penguin Press, 2011.

[14] L. Mamykina, B. Manoim, M. Mittal, G. Hripcsak, and B. Hartmann, "Design lessons from the fastest q&a site in the west," *CHI*, 2011.

[15] C. Dellarocas, M. Fan, and C. A. Wood, "Self-Interest, Reciprocity, and Participation in Online Reputation Systems," *Social Science Research*, 2004.

[16] N. Ducheneaut, N. Yee, E. Nickell, and R. Moore, "Alone together?: exploring the social dynamics of massively multiplayer online games," *CHI*, 2006.

[17] J. Aranda and G. Venolia, "The secret life of bugs: Going past the errors and omissions in software repositories," *ICSE*, 2009.

[18] J. Lerner and J. Tirole, "The Simple Economics of Open Source," *SSRN Electronic Journal*, 2000.

[19] C. Wu, J. Gerlach, and C. Young, "An empirical analysis of open source software developers motivations and continuance intentions," *Information & Management*, 2007.

[20] V. S. Sinha, S. Mani, and S. Sinha, "Entering the Circle of Trust : Developer Initiation as Committers in Open-Source Projects," *MSR*, 2011.

[21] A. Ko and P. Chilana, "Design, discussion, and dissent in open bug reports," *iConference*, 2011.

[22] N. Bettenburg, R. Premraj, T. Zimmermann, and S. Kim, "Duplicate bug reports considered harmful...really?" *ICSM*, 2008.

[23] S. Hiltz and K. Johnson, "Experiments in Group Decision Making: Communication Process and Outcome in Face-to-Face Versus Computerized Conferences," *Human Communication*, 1986.

[24] A. Acuri and L. Briand, "A practical guide for using statistical tests to assess randomized algorithms in software engineering," *ICSE*, 2011.

[25] A. H. Maslow, "A theory of human motivation." *Psychological review*, 1943.

[26] S. Büttcher, C. L. A. Clarke, and G. V. Cormack, *Information Retrieval: Implementing and Evaluating Search Engines*, 2010.

[27] N. Bettenburg, R. Premraj, T. Zimmermann, and S. Kim, "Extracting structural information from bug reports," *MSR*, 2008.