A Large Scale Survey of Motivation in Software Development

Idan Amit[a,*], Dror G. Feitelson[a]

[a]*Department of Computer Science*
The Hebrew University, Jerusalem, 91904, Israel

Abstract

Context: Motivation is known to improve performance. In software development, in particular, there has been considerable interest in the motivation of contributors to open-source.
Objective: We would like to predict motivation, in various settings. We identify 11 motivators from the literature (enjoying programming, ownership of code, learning, self-use, etc.), and evaluate their relative effect on motivation using supervised learning.
Method: We conducted a survey with 66 questions on motivation which was completed by 521 developers. Most of the questions used an 11-point scale. We also conducted a follow-up survey, enabling investigation of motivation improvement given improvement in motivators.
Results: Predictive analysis—investigating how diverse motivators influence the probability of high motivation—provided valuable insights. The correlations between the different motivators are low, implying their independence. High values in all 11 motivators predict an increased probability of high motivation. In addition, improvement analysis shows that an increase in most motivators predicts an increase in general motivation.
Conclusions: All 11 motivators indeed support motivation, but only moderately. No single motivator suffices to predict high motivation or motivation improvement, and each motivator sheds light on a different aspect of motivation. Models based on multiple motivators predict *motivation improvement* with up to 94% accuracy, better than any single motivator.

*Corresponding author
Email address: `idan.amit@mail.huji.ac.il` (Idan Amit)

Keywords: Motivation, Software engineering, Open-source development, Survey validity,

1. Introduction

Motivation has a high impact on human performance in many fields [1, 2, 3]. In the context of software development it is especially interesting [4, 5], due to the phenomenon of open-source development [6], where many of the developers are volunteers [7].

We conducted a large-scale survey asking developers about their motivation. Our survey contained questions related to eleven factors that may affect motivation, taken from prior work. Some of these motivators relate to the culture of open-source, such as adhering to its **ideology** [8] and developing software for **self-use** [6]. Other motivators are related to internal reasons, such as **enjoyment** from developing code [9, 10], **learning** from it [4, 9], the feeling of overcoming a **challenge** [11, 12], or gaining **ownership** of a project [3, 13]. Additional motivators extend this to a wider context: being part of a **community** [14, 15], receiving **recognition** for one's work [9, 4] — or, in unpleasant situations, suffering from **hostility** [16, 17]. Finally, there are project-based motivators like a sense of **importance** [18], as well as receiving **payment** for participating in a project [19, 20]. The survey contained 66 questions, covering the 11 motivators. We obtained answers from 1,724 developers, and 521 of them completed the entire survey. A year later we conducted a follow-up survey, answered by 124 of the original participants.

Unlike prior work, we wanted to investigate motivation in the framework of supervised learning, enabling us to predict motivation in many settings. Our research questions are:

- *Research question # 1* For each of the motivators, what is the predictive power of a high reported motivator with respect to high reported motivation?
- *Research question # 2* For each of the motivators, what is the predictive power of reported improvement in the motivator with respect to reported motivation improvement?

Our main goal was to obtain answers to motivation questions by developers whose real activity is public on GitHub. This matching allows to validate labeling functions, heuristics that predict motivation better than a guess. A first example of the use of our data is given in Amit and Feitelson [21], which validated labeling functions like long commit messages and working diverse hours. The labeling functions that were validated on our data predicted well the retention of 151,775 developers in their real natural activity at GitHub.

Then we moved to predict motivation using the motivators. In prior work it was common to focus on hit rate, how common is a motivator [22, 23, 24, 9]. We wanted to extend this analysis and investigate questions like necessary and sufficient conditions for motivation.

The follow-up survey allowed us to further advance from mere motivation prediction to change prediction, a method we are the first to use in motivation. If a motivator increased from the original survey to the follow-up, it improved. We modeled and investigated the predictive power of *motivator improvement* on *motivation improvement.*

Taken together, a motivator that is influential in these separate ways is likely to have a true impact. In general, all the motivators were found to contribute to motivation in the predictive sense (knowing of a high motivator means a higher probability of motivation). All motivators have a positive precision lift, hence developers reporting the motivator are more likely than excepted to report motivation too. Other than challenge, ideology, and hostility, an improvement in the follow-up answers of the motivator also increases the probability of motivation improvement. The precision lift of the other motivators is higher than 50% and even higher than 100% in some cases, indicating high predictability. On the other hand, none of the motivators is sufficient for high motivation—the highest precision is 67% for community. Also, no motivator is necessary—the highest recall of a motivator was 86% for enjoyment. Yet, when used together in a predictive model, one can predict well both high motivation and motivation improvement.

This study makes three main contributions:

- We conducted a large-scale survey on software developers' motivation, covering 11 motivators.
- We make several methodological innovations:
  - We asked participants for their GitHub profiles, which enabled comparing survey answers and actual behavior.
  - This allowed the validation of labeling functions for motivation [21]. The labeling functions introduce a new methodology for mo-

tivation research. They facilitate quantified, reproducible, long-term investigation, based on large-scale data from real projects.

  - We conducted a follow-up survey, asking the same people the same questions again after more than a year. This allowed us to measure the answers' stability and the impact of changes in motivators on changes in motivation.
  - We framed the analysis as a supervised learning problem. We initially considered each single motivator as a classifier for high motivation, moved to full models, and then applied the same methods for motivation improvement.
  - The large scale of the study allowed us to compare the answers of different people in the same project, estimating subjectivity.
  - We used several types of analyses in tandem to investigate the relations between motivators and motivation. Most relations were consistent in most or all the methods, testifying to their validity.

- We analyze the results using a machine-learning framework and reach several conclusions regarding motivators and their influence:
  - We corroborate previous work showing that in general motivators from prior work are indeed correlated with motivation.
  - At the same time we find that none of them alone is enough to guarantee motivation, so developers usually need several reasons to have high motivation. Also, predictive performance is improved by taking multiple motivators into account.
  - We found that although hostility is rare, when it exists it has a negative influence on motivation. Yet, it tends to be unobserved by others in the same project.

2. Related Work

2.1. Motivation

Motivation, in general and in a work context, has been extensively investigated due to its importance. Many theories were suggested. Skinner suggested operant conditioning, learning behavior due to reward and punishments [25]. Maslow's hierarchy of needs sees self-actualization as the top need [26]. McClelland argues that motivation comes from a mixture of affiliation (society based), authority (opportunities to gain it), and achievements (overcoming challenges) [11]. The equity theory claims that motivation might be hurt due to relative comparison and the feeling of not being fairly treated [17].

In the context of work, the Goal Setting Theory claims that challenging yet achievable goals benefit the motivation [12]. Close in spirit is Vroom's Expectancy Theory [27] which claims that one estimates the outcome, the outcome value, and the probability of the value. Given these, the motivation is determined, and one will have more motivation in tasks where an outcome that is valued is likely to be achieved.

Herzbereg et al. suggested the Motivation-Hygiene Theory [28]. According to it, positive motivation is usually due to intrinsic motivators. However, external hygiene factors might lead to the loss of motivation. Hackman and Oldman suggested the Job Characteristics Theory [18]. They claim that the motivation might come from the job itself, due to the significance, autonomy, skill, identity, and feedback related to the job.

These theories are classical and were introduced long ago, and use different approaches (see comparison [29]). Though they were criticized, they are still beneficial [30].

Demarco and Lister [31], and also Frangos [1], claim that the important software problems are human and not technological. So, many have investigated motivation in software engineering [32, 4, 33, 34].

Open-source development is the collaborative development of software that is free to use and further modify. The best-known non-technical equivalent is Wikipedia. A seminal description of the phenomenon is given in Raymond's "The Cathedral and the Bazaar" [6]. Payment is probably the most common way to motivate people to perform a task, though it is extrinsic motivation and therefore its influence is more complex [19, 20]. However, it is common to perform open-source software development as a volunteer, which means that salary is not the motivation, making it startling from an economical point of view at first sight [35]. Therefore, the motivation of open-source developers was investigated as a specific domain, to uncover other motivators [15, 36, 37, 38, 39].

Empirical research also supports the benefit of motivation. Task significance, a motivation cause, was shown to increase productivity [40]. Campbell et al. see performance as a function of motivation but also of knowledge and skill [41].

2.2. Motivators

The research literature has not produced a canonical agreed list of factors that influence motivation. Mayer et al. reviewed 75 years of motivation measures [42]. This showed that many different factors have an effect, but the agreement between them is limited. We therefore needed to select which ones to include in our study.

We based our list of motivators mainly on Beecham et al.'s review of motivation in software engineering [4] and Gerosa et al.'s [9] work on motivation in open-source development. Our default was to include motivators in order to cover more aspects of motivation. The motivators that we chose have a long history going back to Herzberg's Motivation-Hygiene Theory [28], and therefore were thoroughly investigated over the years (e.g., in general [24] and in software development [23]). Note that we excluded some of the motivators that are less relevant to open-source development, like "Job security" and "Company policies". Conversely, we did include "hostility", which is a *de*motivator (it is common to refer to factors of positive influence as motivators and those of negative influence as demotivators) [17]. Since we have only a single demotivator, we use the term "motivator" to refer to both it and the positive motivators.

Attracting developers is important to project sustainability [43]. This includes retaining existing developers and having newcomers. Although we do not identify newcomers, we ask about self-use, enjoyment, learning, and community, indicating common newcomer motivations [44, 45, 46].

In Section 5.3 we list and discuss the 11 motivators used in our study.

3. The Survey Instrument

3.1. Design

Our design aimed to serve some goals. This led to the construction of a relatively long survey with 66 questions (See replication package [47]). The first goal was to obtain labels about quality and productivity in development. The first section of the survey was "Questions regarding yourself" (18 questions). This part included general questions about motivation and verifiable questions about conduct (e.g., the writing of detailed commit messages). We also asked questions about self-rating of skill. After collecting the answers we found out that developers tend to have a very good opinion about their performance so we ended up using these questions only to investigate answer validity problems (See replication package [47]). A few of the questions (e.g., "I enjoy software development very much") were used to indicate the existence of a motivator, later analyzed for influence on general motivation.

The second goal was to obtain information about motivators. This served us for predictive analysis of answers, co-change analysis of motivation improvement, and labeling functions based on behavioural cues in real software development [21]. Due to that we aimed to represent many motivators and use several questions for the main ones. The questions were in the second section "Questions regarding activity in a repository" (28 questions), in which we asked about a specific project and its related behavior. This included our motivation ground truth question: "I regularly have a high level of motivation to contribute to the repository" (based on [48]).

We planned to validate our results with respect to the popular Job Satisfaction Scale questionnaire [49] (10 questions) which was included as is. Analysis and direct feedback indicated confusion, so we ended up using it just for validity. Hence, it served as a source of attention-check items [50]. In questions about the community, we asked people in a single-person project to skip, also serving as attention checking (See replication package for details [47]). Since the mistakes were due to misunderstanding, as some of the participants directly commented, we included participants who answered so and just measured how common these mistakes were.

We also added a "Demography" section (8 questions). Last, we ask an open question requesting comments whose goal is to ensure that we did not miss a significant factor in the structured questions.

By using questions from prior work, we benefit from a previous validation. In the discussion below we note prior work on each motivator (Section 5.3), and in the replication package we identify the specific source of each survey question [47].

An important goal of the survey was to enable us to compare answers regarding motivation and actual behavior (as used in [21]). Therefore, we asked the participants to choose a specific project and provide its name, preferably a public GitHub project. We asked for their GitHub profile to investigate the developer behavior. We also asked for the email from participants who were interested in the research results. Emails and GitHub profiles are personally identifying information, which is usually not collected. We needed them to match the answers to other data related to the same person, but did not include them with the other experimental materials. This was approved by our IRB (study 09032020).

The survey was designed to take about 10 minutes. Most questions used a

Likert scale [51, 52]. Values ranged from 1 to 11, providing closest-to-normal distribution [53]. All participants saw the sections in the same order, yet the order of the questions within the sections was randomized.

3.2. Execution

The target population in software engineering is hard to define in many cases [54]. The GitHub developer population was essential to match answers about motivation and actual behavior in GitHub. However, this population is small compared to all developers. We therefore also reached out to developers in social media, which led to 80% of the participants. We used machine learning to try to distinguish between the populations and did not find a large difference (see supplementary materials).

The survey was conducted using the Qualtrics platform from December 2019 to March 2021. We obtained 1,724 participants, 521 of them completed the survey.

We conducted a pilot until February 2020, including 16 people from our social circles. We received feedback on the focus of the survey and redundant questions, which were deliberate decisions. Few participants alerted us about the principal position of GitHub, feedback we failed to use at this stage. We received wording suggestions that we applied. We checked that the long survey could be completed in 15 minutes, which was OK. After the entire survey was conducted, we found out that 84% of those who finished it did it in this duration. Since the content of the questions was not changed, we included the pilot participants in the survey.

GitHub is a platform for source control and code development used by millions of users [55]. We initially focused on 1,530 active public GitHub projects with 500+ commits during 2018, having 832k developers, described in [56, 57, 58]. About 40,000 developers contributed to these projects that year, of which 9,000 contributed more than 12 commits. We extracted developers' email addresses using the GitHub public email API, fetching the emails of the developers who chose to share them publicly. We sent emails to 3,255 developers with a public email that had enough commits. We also had a gift card lottery, offering $50 to three of the participants. This channel led to 339 participants, which is 20% of the total.

We also recruited participants in social networks by convenience sampling [59, 60], which led to the remaining 80% of participants. We used Reddit, an online discussion site, as an important source of participants. Reddit has numerous subreddits, channels dedicated to discussions on specific topics. It has many channels relevant to programmers such as programming languages, operating-systems, tools, etc. We posted in 46 different subreddits on different dates and hours, to get familiar with the community, as posting at a high rate might be considered as spamming. Each subreddit has different formal and informal rules that should be respected. We found that people showed interest in the survey, which led to 70 comments and discussions (e.g., open-source, software quality) and 212 upvotes. This in turn led to more attention to the survey. We used machine learning to try to distinguish between the GitHub and social-media populations and did not find a large difference (see supplementary materials).

As noted above, we asked participants for the name of their project and their GitHub profile. They provided the names of 484 projects and 303 personal GitHub profiles. After posting the survey on social media, we noted that many participants stop in the "Questions regarding activity in a **repository**" section since they do not contribute to a GitHub repository. This was also accompanied by direct feedback saying that. Since we were interested in answers about motivation in general and needed the GitHub profile only to link to actual behavior, we changed the questions about "GitHub repository" to "any project", avoiding this drop.

The original survey ended in March 2021. A year after the last response, in April 2022, we sent a follow-up survey to the 341 participants who provided their emails in the original survey. We sent them the name of the project on which they initially answered, and asked them to answer on the same project in case they are still active there and on a different project otherwise. The questionnaire was the same as the first one, with the additional validation question "Is it the same project on which you answered last time?". In the follow-up survey, 124 out of the 341 participants we reached out to answered (36.3%).

3.3. Answers Treatment and Validation

Our emails to GitHub developers had 4.4% response rate, close to the 5% reported by [61]. Due to a mistake, we sent multiple emails to developers who worked on multiple suitable projects, and we apologize for that. This was both annoying and mis-estimated the number of developers. After fixing the mistake the response rate was 8.1%, close to the 9% reported by Feitelson which used a similar dataset [62]. We believe that the response rate benefited from the use of the GitHub public email API. Email is used by GitHub in the development, and therefore they are usually updated (our bounce rate was

1%). The API returns the emails of developers that choose to share them publicly, hence more agreeing to cold communication. We believe that many people found the topic, motivation in software development, exciting. We anecdotally noticed it in up-votes and comments on social media, responses to the open questions in the survey, and the 19.8% of the participants that left email wanting to learn the research results.

We checked if participants answered all questions with the same answer (e.g., due to maliciousness or boredom). There was a single such participant, who answered a few "Questions about yourself" and dropped, not included in the analysis anyway. 91% of the participants that did not finish the survey, abandoned in less than 5 minutes. Only 7% answered the full survey in less than 5 minutes, a quick yet possible duration.

Some participants do not answer all questions. In these cases, we used the relevant answers and did not attempt to fill in the missing ones.

Answers have validity problems. The internal coherence, correlation between questions belonging to the same motivator, is moderate. The original survey and the follow-up survey can be treated as a longitudinal study. The surveys agreement is better than independent. We analyzed all the above in-depth and presented them in the replication package [47], showing that despite the noise a signal is found.

However, the strongest validation comes from the ability to predict the behaviour of 151,775 developers in 18,958 GitHub projects. Amit and Feitelson used our dataset to define labeling functions [63, 64, 65] for motivation [21]. The labeling functions are weak classifiers, validated heuristics that predict motivation better than a guess, which are based on behavioural cues. For example, working in more diverse hours and writing longer commit messages are labeling functions. These labeling functions predicted the high answers of our survey participants. When used on the entire GitHub datasets, they predicted retention, and higher activity and output of 151,775 developers. Hence the answers agree with the labeling functions, which agree with the behaviour of a mass of new unseen developers.

All data and code are in the supplementary materials. Personal identifiers were hashed to preserve anonymity yet allow matching.

4. Analysis

Supervised learning is a powerful framework, suitable to our needs. In supervised learning, we try to predict a concept (e.g., high motivation) using a classifier (e.g., decision tree) based on features (e.g., motivators). The versatility of supervised learning allows us to investigate a single motivator, all of them, or temporal changes. One can apply supervised learning to a single motivator for direct evaluation or to all of them, leverage their combined power. Supervised learning also allows to build models, identifying the behavior of the predicted concept.

The concept, which we try to predict, is high motivation. The classifiers, which provide us with predictions, are the motivators (e.g., high ownership). We evaluate how well motivators predict high motivation, using metrics that compare the prediction to the actual motivation. Interesting metrics are the fraction of those with, say, high ownership who indeed are highly motivated (precision), the improvement over just the prevalence in the population (precision lift), and what fraction of the highly motivated have high ownership (recall).

The analysis of each individual motivator with respect to general motivation provides simple basic results yet ignores more complex relations like confounders. We therefore also performed additional, more complex analyses. We analyzed the relationship between motivators to see that this risk is small. We used the follow-up survey to analyze the motivation improvement of each motivator alone. Last, we built combined models utilizing all motivators to avoid the risk of confounding and leverage the power of all motivators.

5. Results

5.1. Demographics

It is difficult to define 'developer' (e.g., by education, profession, minimal activity) and, therefore, to define a representative developer population. However, a possible comparison is provided by the 'Stack Overflow' (SO) annual developer survey. SO is a leading questions-and-answers website for programming, with more than 100 million monthly visitors. The 2021 survey, overlapping our survey period, was answered by over 80 thousand developers [66]. As far as we know, this is the largest survey of developers and therefore an interesting comparison.

Our survey participants come from 74 countries, compared to 180 countries in SO. In our survey, the leading countries are the United States, Germany, Great Britain, India, and France. These are the same leading countries as in SO, but in SO India is in second place and the Western countries have a somewhat lower representation.

4.2% of our participants identified as females, 95.1% as males, and the rest identify as others. This is close to the ratio of participants not identifying as males in the 2021 SO survey, which was 5.3%. It is also close to ratios of 4.8% [67] and 11.2% [68] in other sources.

Figure 1 presents the age distribution. The distributions are rather close with slightly higher ages in our survey.

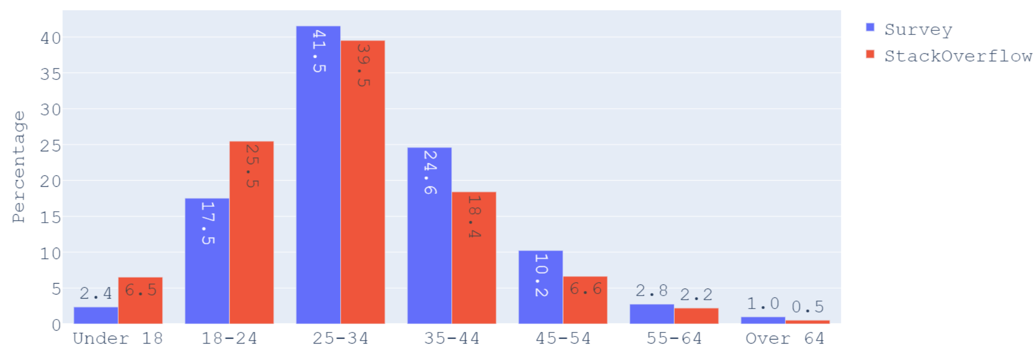

Figure 1: Age Distribution

80.7% of the participants in our survey work as professional programmers, compared to 69.7% in SO. The average years of programming experience in our survey (professional or not) is 11.1, representing very experienced developers. Figure 2 shows that in both surveys about half of the participants have at least 10 years of experience.

Our survey had more participants with high degrees than SO, as shown in Figure 3. The academic domains in our survey were: computer science 46.3%, technology 32.0%, science 9%, business 3.4%, math 2.5%, arts 2.1%, and the rest from other domains.

To conclude, the demographics in our survey are close but not the same as those in Stack Overflow. For example, more people identify as professional developers in our survey. However, it seems that the threat of not representing the developers community is low.

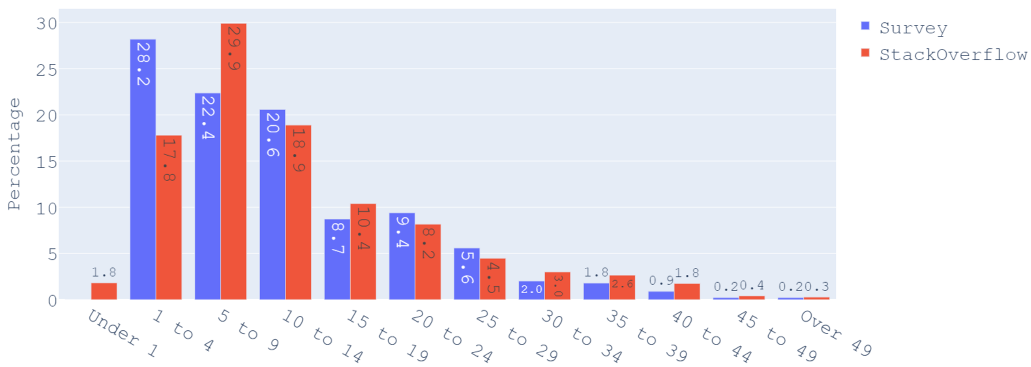

Figure 2: Years of Experience Distribution

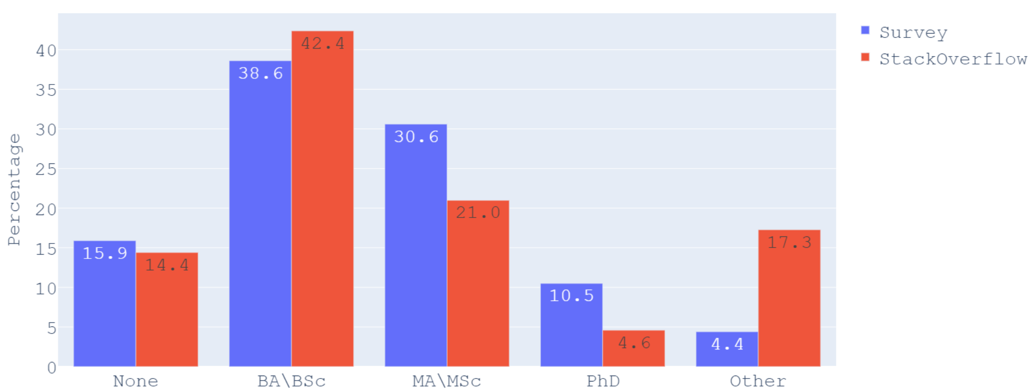

Figure 3: Academic Background

5.2. General Motivation

Motivation might derive from many motivators, from payment to enjoyment. The concept that we would like to investigate is high motivation to contribute to a project, and its relations to these various factors.

We measure general motivation using the question "I **regularly** have a **high** level of motivation to contribute **to the repository**" (pattern is based on [48]). The results show that developers are generally motivated (Figure 4). High motivation (at least 9 = 'somewhat agree' on a scale of 1 to 11) was reported by 52.4% of the participants.

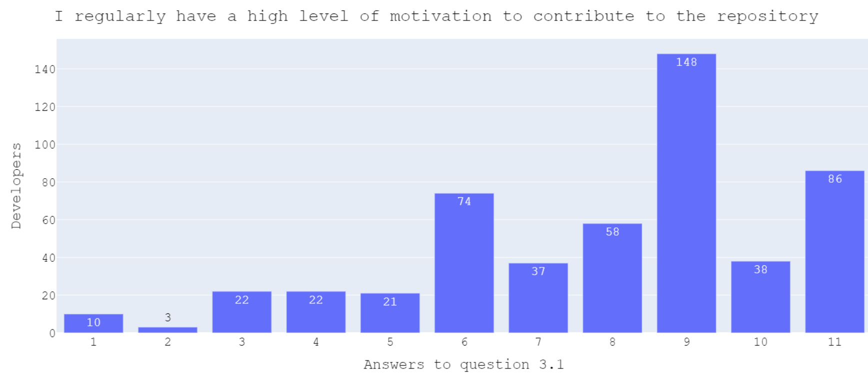

Figure 4: Distribution of answers to general motivation question.

We identified paid people by their answer to a specific question about it (question 3.c). 41% of the participants who answered this question said that they are paid (284 participants). 24% of the participants (377) were identified as contributing to GitHub (by finding a reference to GitHub in the project name). Note that being paid and using GitHub are not mutually exclusive. 38% of participants of GitHub projects that answered the payment question, said that they are paid.

It is generally accepted that motivated workers work longer hours [69]. Showing that our measurement of general motivation exhibits the same relation provides supporting evidence for its validity. And indeed, participants reporting high motivation (at least 9) reported an average of working 19.3 hours a week on the project, compared to 3.4 hours for those reporting low motivation (below 9). This result may be tainted by mixing data about paid developers with data about volunteers, who are common in open-source projects. We checked this by looking at paid workers separately, out of the participants that reported payment, and the influence of payment is indeed large. For unpaid workers the reported average working hours were 10.8 (high motivation) and 4.5 (low), while for paid workers they were 27.9 (high) and 25.8 (low). Thus, both paid and unpaid participants work more hours when motivated yet the average of unmotivated paid developers is higher than that of motivated unpaid developers and therefore we do not mix them.

We planned to also validate the measurement with a similar question from the Job Satisfaction Survey: "Taking everything into consideration, how do you feel about your work?" [49]. However, some of the participants were confused and answered most survey questions on an open-source project to which they contribute, yet answered the Job Satisfaction Survey on their regular job. Despite this confusion, the Pearson correlation between the questions is 0.32. When focusing on paid developers, for which the probability of confusion is lower, the correlation is 0.36.

5.3. Motivators

This section answers *Research question # 1*: For each of the motivators, what is the predictive power of high reported motivator with respect to high reported motivation?

Table 1 summarizes the predictive performance of all the motivators. We discuss each of the motivators in the following subsections. Note that in each row we analyzed participants that answered the motivation question and at least one question about the motivator, so populations are not necessarily identical.

We use common metrics used in machine learning and information retrieval. The concept that we want to predict is high general motivation. This was operationalized by the answer to the question "I regularly have a high level of motivation to contribute to the repository" being 9 'somewhat agree' or above.

Usually in machine learning a classification algorithm (e.g., decision tree) is used to create a model, which is a specific rule providing predictions. In our case the models are just the motivators (e.g., ownership, challenge), also binarized into high and low using 9 as the threshold. Note that in this part the models are pre-defined, and not learned by a classification algorithm, and we only evaluate their predictive performance.

The cases in which the concept is true are called 'positives' and the positive rate is denoted $P(positive)$ (in our case this is 0.52 as noted above). Cases in which the model is true are called 'hits' and the hit rate is $P(hit)$. For example, a high hit rate for ownership means that many participants report ownership, and we want to see whether they are also generally motivated.

Ideally, hits correspond to positives, but usually some of them differ. Precision, defined as $P(positive|hit)$, measures a model's tendency to avoid false positives (FP). But precision might be high simply since the positive rate is high. Precision lift, defined as $\frac{precision}{P(positive)} - 1 = \frac{P(positive|hit) - P(positive)}{P(positive)}$, copes with this difficulty and measures the *additional* probability of having a true positive relative to the base positive rate. Thus, a useless random model will have precision equal to the positive rate, but a precision lift of 0. Recall, defined as $P(hit|positive)$, measures how many of the positives are also hits; in our case, this is how many of the highly motivated participants also report high ownership.

Table 1: High Motivation Predictability by Motivator

Motivator	Hit rate (Fraction ≥ 9)	Performance as predictor of motivation			
		Accuracy	Precision	Prec. lift	Recall
Enjoyment	0.74	0.64	0.62	0.18	0.86
Ownership	0.73	0.59	0.57	0.10	0.81
Learning	0.72	0.59	0.58	0.10	0.80
Importance	0.63	0.61	0.61	0.16	0.73
Challenge	0.62	0.63	0.62	0.20	0.74
Self-use	0.56	0.60	0.61	0.17	0.65
Ideology	0.53	0.57	0.59	0.13	0.60
Recognition	0.48	0.58	0.60	0.18	0.56
Payment	0.45	0.55	0.58	0.10	0.49
Community	0.41	0.63	0.67	0.35	0.53
Hostility	0.07	0.52	0.65	0.30	0.08

We now present our analysis of each individual motivator, from the most to the least prevalent. We show how common high answers (9 and above) are to each motivator, in general, for paid developers, and for open-source developers.

5.3.1. Enjoyment

We measure enjoyment [9, 10] by the following questions (numbered by their location in the survey):

2.9 I enjoy software development very much

2.15 I enjoy trying to solve complex problems

3.8 My work on the repository is creative

3.10 I derive satisfaction from working on this repository

An example of the results is shown in Figure 5.

We calculated the average answer to all these questions per participant, and then the average of these averages. This led to an overall average of 9.07. 74% of the participants reported high enjoyment (at least 9 - 'somewhat agree'), more than all other motivators.

76% of the GitHub participants reported high enjoyment and 75% of the paid participants. The correlation of enjoyment with motivation is 0.51, the highest of all motivators.

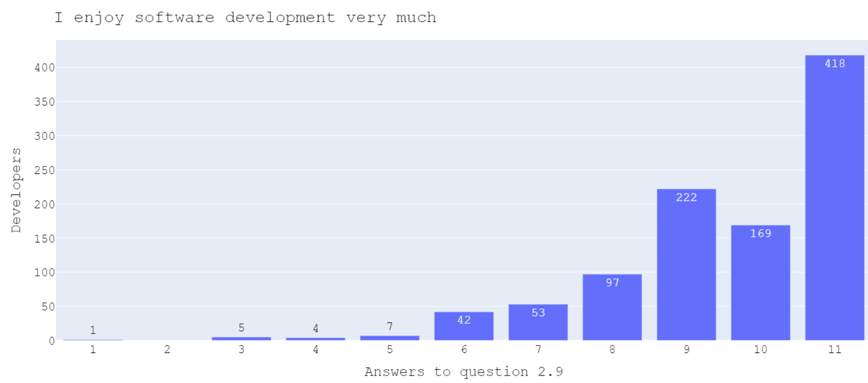

Figure 5: Answers distribution of enjoyment question.

The recall of enjoyment is 0.83, making it very common among people of high motivation, more than both the hit rate and the positive rate. The precision is 0.62 and the precision lift is 0.18, which is positive and significant yet not very high.

Note that the hit rate of 74% for enjoyment is significantly higher than the positive rate of 52%. This has a large influence on the predictive metrics. True positives are the intersection of the hits and the positives, so they are bounded by both. But once the hit rate is higher than the positive rate, the precision is bounded from above. In our case, the model hit rate is 74% and the positive rate is 52%, so the model precision can be at most $\frac{52}{74} = 70\%$. The actual precision of 62% appears to be not high, yet it is 89% of the bound created from the positive rate and the hit rate.

5.3.2. Ownership

We measure ownership [4, 9, 3, 13] by the following questions:

3.2 - I have complete autonomy in contributing to the repository

3.3 - I have significant influence on the repository

3.4 - I feel responsible to the repository success

3.16 - I am a core member of the repository

The average answer for ownership was 9.02. 73% of the participants reported high ownership (9 or above, second highest of all motivators). This was also the percentage for paid participants; with GitHub participants it was 75%. The correlation of ownership with motivation is 0.24. The recall when predicting high motivation based on high ownership was 0.81, higher than the hit rate. However, the precision is 0.57 and the precision lift is only 0.10, partly because ownership is so common.

5.3.3. Learning

Learning [4, 9] is based on the question:

3.17 - I learn from my contributions

The average of the answers was 9.15, the highest among all motivators. 72% of the participants reported high levels of learning, 70% of the GitHub participants and 77% of the paid ones. The correlation of learning with motivation is 0.23. Learning has a recall of 0.80, indicating that it is another very common characteristic of people with high motivation. Its precision is 0.58 and its precision lift is 0.10, which is relatively low.

5.3.4. Importance

Importance [4, 18, 40] is based on the question:

3.11 - The repository is important

The average was 8.62. The correlation of importance with motivation is 0.35. 63% of the participants reported high feelings of importance. The same occurred among GitHub participants, compared to 74% among paid participants. This is rather surprising since we assume that one will have more considerations and constraints in the context of a paid job than in volunteering for open-source projects. So, we would assume that one would have higher freedom to choose by importance when volunteering, leading to a higher rate among GitHub participants, but the data shows the opposite. Importance has a precision of 0.61 and a precision lift of 0.16. It has a recall of 0.73, which is high in absolute terms and relative to its hit rate.

5.3.5. Challenge

Challenge [4, 11, 12, 5] is based on the question:

3.9 - Working on this repository is challenging

The average of challenge answers was 8.41. 62% of participants reported a high sense of challenge, 60% of the GitHub participants and 66% of paid ones. The correlation of challenge with motivation is 0.30. Challenge has precision of 0.62 and precision lift of 0.20. Its recall is 0.74, which is high in absolute terms and relative to its hit rate.

5.3.6. Self-use

Self-use [9, 6] is based on the question:

3.5 - I'm interested in the repository for my own needs

The average of self-use answers was 7.86. 56% of the participants reported high self-use motivation, 61% of GitHub participants, and 43% of paid ones. Note that while usually the probabilities in the entire population, in GitHub, and among paid participants are rather similar, in this case the probabilities are quite different. 'Scratching your own itch' is a well-known motivation in open-source [6] so one would expect a higher probability in GitHub. On the other hand, many companies produce organizational software that does not have personal uses, so 43% of paid participants who self-use may sound rather high. The correlation of self-use with motivation is 0.16. Self-use has a recall of almost two-thirds, 0.65. This seems to be a unique attribute of open-source, enabling people to develop the software that they need. The precision is 0.61 and the precision lift is 0.17. This might be because people see satisfying their needs as a task to complete and not an enjoyable activity. Indeed, self-use and enjoyment have a Pearson correlation of just 0.16.

5.3.7. Ideology

Ideology [9, 8] is based on the question:

2.18 - I contribute to open source due to ideology

53% of the participants reported high ideology-based motivation, rising to 61% of GitHub participants, aligned with the ideological roots of open-source development [8]. 49% of paid participants also gave high answers regarding contribution due to ideology. That can be either due to many people being paid to contribute to open-source, or a common habit of paid developers to contribute to open-source in their free time. Regardless of the reason, the popularity is surprisingly high. But we note that in the answers to the open question participants said that different ideologies (e.g., 'Software should be free' [70], 'Social good') can lead to contribution to open-source software, and that a finer distinction is needed. The average of ideology answers was 7.34. The correlation of ideology with motivation is 0.14, the lowest other than for hostility. The precision of ideology is 0.59, the precision list is 0.13, and the recall 0.60.

5.3.8. Recognition

We measure recognition [4, 9, 20, 6] by the following questions:

2.13 I contribute to open source in order to have an online portfolio

2.14 I try to write high quality code because others will see it

3.15 I get recognition due to my contribution to the repository

3.24 In the past year, members of my GitHub community asked questions that show their understanding of my contributions (based on [71])

3.25 In the past year, members of my GitHub community expressed interest in my contributions (based on [71])

The average of all the questions was 7.33, lower than all other positive motivators. 48% of the participants reported high recognition-based motivation, 49% of the GitHub ones, and 52% of paid ones. The correlation of recognition with motivation is 0.27. The precision is 0.60 and the precision lift 0.18, indicating a boost to motivation. The recall is 0.56 while the recognition hit rate is 0.48.

5.3.9. Payment

Payment [4, 9, 19, 20] is based on the yes/no question:

3.c - I'm being paid for my work in this repository

We note that remuneration in open-source projects may have many facets. Developers may accrue income from donations or lectures. Their work on the project may help them secure future positions or gain access to future consulting contracts. When writing the question we thought of being paid a salary.

41% of the participants that answered the payment question said that they are paid. 38% of participants of GitHub projects that answered the payment question said that they are paid.

Payment has precision of 0.58 and precision lift of 0.10, making it one of the weakest motivators in general and the weakest for its hit rate. Payment is also the only motivator that had a negative precision lift when we run the same analysis on the follow-up survey.

Its recall is 0.49, less than the positive rate of high motivation which is 0.52. The correlation of payment with motivation is 0.15, lowest than all but ideology and hostility.

5.3.10. Community

We measure community [4, 9, 11, 14] by the following questions (which we asked to answer only if you are not the only developer in the project):

3.13 Belonging to the community is motivating my work on the repository

3.14 The community is very professional

3.20 The repository's community of developers is more motivated than that of other repositories

3.24 In the past year, members of my GitHub community asked questions that show their understanding of my contributions (based on [71])

3.25 In the past year, members of my GitHub community expressed interest in my contributions (based on [71])

Note the questions 3.24 and 3.25 are about recognition from the community and therefore appear in both motivators.

The average of community answers was 7.36. 40% of the participants reported high community-based motivation, lower than all positive motivators. This is based on 40% of the GitHub participants and 44% of the paid ones. The correlation of community with motivation is 0.42, the second highest. The precision is 0.67 and the precision lift is 0.35, higher than all other motivators. Hence, though the community motivator is not common, when it exists, the probability of high motivation is higher. The recall is 53%, not very high yet 29% higher than the hit rate.

5.3.11. Hostility

Hostility can be viewed as a community with negative influence. Hostility hurts motivation hence it is a demotivator and not a positive motivator. We measure hostility [16, 17, 28, 72] by the following questions (which we asked to answer only if you are not the only developer in the project):

3.6 We have many heated arguments in the community

3.7 I wish that certain developers in the repository will leave

3.22 In the past year, members of my GitHub community put me down or were condescending to me (based on [72])

3.23 In the past year, members of my GitHub community made demeaning or derogatory remarks about me (based on [72], Figure 6)

The average of hostility answers was 2.80. Only 7% of the participants reported high hostility, 4% of the GitHub participants and 7% of the paid ones. The recall is just 8%, yet it is higher than the hit rate. The correlation of hostility with motivation is 0.01. This is the lowest correlation, close to zero yet not the large negative correlation which is expected.

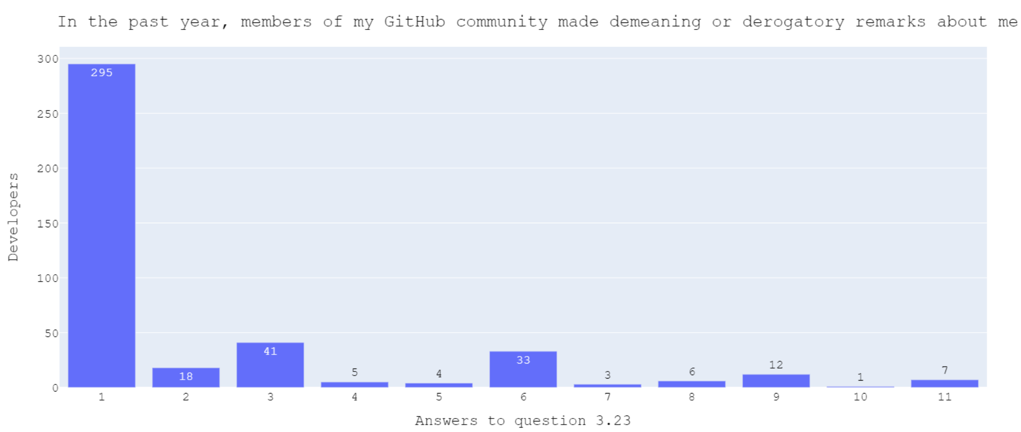

Figure 6: Answers distribution of a hostility question.

Surprisingly, hostility also has a relatively high precision of 0.65 and a high precision lift of 0.30. One would expect that knowing that someone suffers from hostility will *reduce* the probability of high motivation, instead of the increase that we see. But the participants who reported high hostility also reported higher averages for all motivators besides payment. A possible explanation is that those participants kept contributing to the project due to the other motivators; those who suffered from hostility and did not have other reasons to stay probably left. Note, however, that we had only 9 people that reported both high hostility and high motivation, so the analysis is not robust.

We identified 10 pairs of developers which contribute to the same project. This allowed us to evaluate their agreement on hostility. When a person reports heated arguments (question 3.6), the probability that the other participant will agree is just 50%. For the rest of the hostility questions, the other participants never claimed high hostility too. For comparison, in importance and challenge, which also describe aspects of the project, if one participant provided a high half answer (6 or above), the other always agreed. This provides an important indication that hostility might go unnoticed.

5.4. Motivation Improvement Analysis

This section answers *Research question # 2*: For each of the motivators, what is the predictive power of reported improvement in motivator with respect to reported motivation improvement?

An important goal in many data analyses is to uncover causal relations. But causality is hard to define rigorously because it is hard to ascertain that motivator A caused outcome B. The usual approach is to look at correlations between motivators and outcomes in a given dataset, as we did in Section 5.3. We now extend this to look at the *dynamics across time*: the possible correlation between *a change* in a motivator and *a change* in the outcome. For example, we want to see whether an increase in the sense of ownership of a project predicts an increase in motivation.

Such "co-change" analysis [57] is important for the following reason. If causality exists, meaning that in certain contexts motivator A causes outcome B, then a change in A will cause a change in B. But co-change of two motivators does not necessarily imply causality. By identifying instances of co-change, where a change in A correlates with a change in B, we therefore identify cases where causality may be at work.

Note, however, that the actual relationship between motivators and general motivation may be conditioned on other motivators. In this subsection we look at the co-change of motivators and general motivation alone, regardless of context. In the next subsection we consider all the motivators together, to handle cases where the effect of A on B is conditioned on another motivator C.

The change data comes from comparing the original survey and the follow-up survey. In the original survey 341 developers provided their emails. A year after the last response, we reached out and asked them to answer the survey again. We asked them to answer on the same project if they are still active in it. This allowed us to compare the answers of the same person over time. We had 124 follow-up participants in total. 60 of them continued in the same project, and these are the ones we analyze here. For each of them, we look at increases in the motivators and general motivation from one year to the next. Note that if a person reported 3 for ownership in the first survey, and 4 in the follow-up, this is an increase regardless of the values being low.

Table 2: Motivation Improvement Over Time Predictability by Motivator

Motivator	Improvement rate	Prediction of improved motivation			
		Accuracy	Precision	Prec. lift	Recall
Challenge	0.33	0.53	0.10	-0.50	0.17
Ideology	0.30	0.60	0.17	-0.17	0.25
Importance	0.30	0.70	0.33	0.67	0.50
Learning	0.30	0.70	0.33	0.67	0.50
Enjoyment	0.28	0.71	0.34	0.71	0.48
Recognition	0.27	0.72	0.39	0.95	0.47
Self-use	0.22	0.78	0.46	1.31	0.50
Ownership	0.17	0.77	0.45	1.27	0.35
Hostility	0.16	0.70	0.20	-0.01	0.15
Community	0.16	0.75	0.39	0.93	0.28

The probability of improvement in general motivation, i.e. the positive rate, was 20%. Since we included only developers that stayed in the same project for at least a year, there is probably survivorship bias, and the probability in the whole population is probably even lower.

The probability of improvement in the different motivators is at most 33% (Table 2). When the precision lift is positive, it tends to be very high. We could not find out why the lift is negative for challenge and ideology. One could expect a larger negative lift for hostility, which did not materialize. This may be explained by developers having other motivators that offset hostility as explained in Section 5.3.11.

The recall is up to 50% for many motivators and higher than their improvement rate (hit rate). This shows that improvement in importance, learning, enjoyment, recognition, and self-use are common when motivation improves.

Only a single person that was not originally paid received a payment in the follow-up, therefore we did not apply co-change analysis to this motivator.

Co-change analysis can be performed in the downward direction too: given a decrease in a motivator, how common is a decrease in general motivation. Results are quite similar to the upward direction and given in the supplementary materials.

The follow-up survey can also be considered a replication of the original survey. We used all 124 participants that answered the follow-up survey to run the analysis of predicting motivation (as in Table 1). We found a positive precision lift for all motivators besides payment. The agreement supports the results in general. The disagreement in payment indicates that the result is not robust.

5.5. A Combined Motivation Model

This section answers both research question # 1 and # 2, yet by using all motivators together and not one by one.

All the motivators have a positive precision lift of at least 10% (Table 1). This is aligned with the prior work claiming their positive influence. However, the highest precision lift is just 35%, with 67% precision, for the 'Community' motivator. This means that none of the motivators is a sufficient condition for high motivation or close to it. We analyzed the Pearson correlation between motivators (in supplementary materials) and noted that none of the correlations is very high. Hence, each motivator describes a different aspect, and a combination of motivators might help to reach high motivation.

Machine learning allows us to build predictive models for motivation and evaluate their performance. The steps that we followed are describing each case with features, splitting the dataset into train and test, building a model on the train dataset and evaluate it on the test dataset.

The inputs of machine learning models are named "features". So far, we investigated each motivator as a single feature, ignoring all other motivators. This type of analysis suffers from the threat of confounding variables on one hand and does not leverage the full power of the data on the other hand. We therefore built combined models to predict motivation, based on all the motivators as features.

Since exact values tend to change (see replication package [47]) and we are indifferent to the level of low motivation, we prefer classification framework over regression. The predicted concept was high motivation, operationalized as before by answers of 9 ('somewhat agree') or above to "I regularly have a high level of motivation to contribute to the repository". We had 345 participants that answered this question. The positive rate is 52%.

We used the scikit-learn package for classification algorithms [73]. We used low-capacity small models such as decision trees and logistic regression in order to obtain simple interpretable models which are also rather robust to overfitting [74]. We also used models of medium capacity - random forests, k-nearest neighbors, AdaBoost, gradient boosting, support vector machines, stochastic gradient descent, and neural networks to build models of better representation ability and performance.

In order to obtain a reliable performance estimation of models, it is common to train them on one subset of the samples and measure on a different test subset. When splitting samples into train and test, we would like to have enough samples for reliable estimation in the test, and use the rest for train since models perform better with more data. The number of samples we have is very low. Repeating with 10 different seeds we noted that the standard deviation on both the accuracy and the positive rate (which is not influenced from learning) is 0.03. We therefore used 30% for reasonable test estimation and used the rest for training models that are suitable for small datasets.

The performance of all the models was rather close, with accuracy ranging from 62% to 77%. Amusingly, the highest accuracy on the test set (77%) was reached by a single node tree, checking high enjoyment. As Table 1 showed, the accuracy when using enjoyment on the whole dataset is just 64%, so this result is accidental. The model with the second highest accuracy (76%) was a neural network [75], whose capacity is relatively high. The simple model of highest accuracy was a logistic regression model [76] which reached accuracy of 72%. Its intercept was -2.04, indicating a general tendency for low motivation. Hostility had a strong negative coefficient of -0.46. All the other motivators had positive coefficients. The highest were enjoyment with 1.13, self-use with 0.61, and importance with 0.59.

Note that models can assign different weights to false positives and false negatives, and trade off precision and recall. Using this, we could build a precision-favoring decision tree model [77] with precision of 81% and recall of 41%. Conversely, a recall-favoring stochastic gradient descent (SGD) model

[78] reached recall of 96% with precision of 62%.

We also modeled the co-change dataset of Section 5.4, to predict a change in the motivation based on changes in the motivators as features. Such a model is of interest since assuming that motivation is a function, a co-change model can predict the result of a change.

Two properties of such modeling deserve special attention: accuracy and minimality. We first discuss accuracy. A co-change model allows us to predict *the motivation change* given any *motivator's change*. Perfect accuracy assures us that there are no other external causal variables influencing the samples in our dataset. Assume by contradiction such a variable c, other than the model variables. Hence there is a behavior function g and an assignment of values such that $g(c_1, v_1, ...v_n) \neq g(c_2, v_1, ...v_n)$ where $v_1, ...v_n$ are the values of the model variables. However, since we have perfect prediction given the model variables, it should be that $g(c_1, v_1, ...v_n) = m(v_1, ...v_n) = g(c_2, v_1, ...v_n)$ — a contradiction. Hence such a variable cannot exist. Perfect accuracy is rare, and mostly indicates a problem in the analysis and not capturing all causal variables. However, the accuracy bounds the influence of such external variables.

As for minimality, consider decision trees [77] as models. For each leaf, variables that do not appear in the path to this leaf do not influence the prediction. On the other hand, each variable along the path is necessary, and a change in its value will change the prediction. In this sense, the model is minimal and every variable along the path is required. All the variables that we use here are mutable (can change, in contrast for example to the project creation year). With both perfect accuracy and minimality, the r iable will

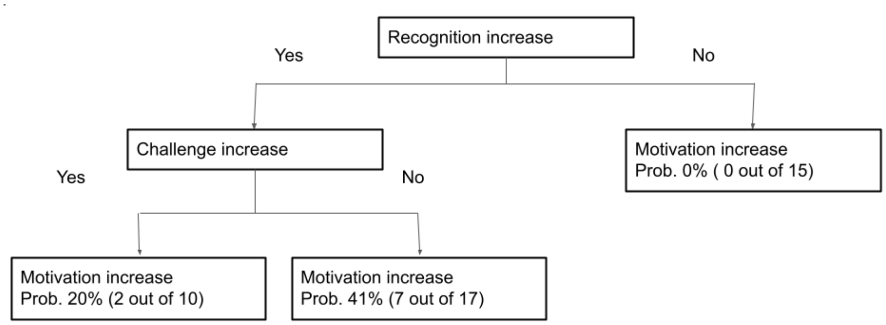

Figure 7: A decision tree predicting motivation improvement

When aiming for high accuracy, we built models based on AdaBoost [79] and Neural Networks [75] which reached accuracy of 94%. Aiming for either precision or recall, 100% were reached. The size of these models is high and far from minimality, which is the price for achieving the high accuracy. Alternatively, we found a small model, presented in Figure 7, whose "recognition increase and challenge decrease" leaf reaches 78% recall with 41% precision. Note, however, that the dataset is a very small dataset with a high VC dimension [80, 81] (due to having many questions and wide scale), and therefore the threat of noise is very high. Also, the dataset probably does not fully represent motivation complexity. A larger dataset will probably better represent human motivation behavior but will require a larger model and have lower performance.

6. Implications

Many people are interested in improving motivation. They include managers, colleagues, open-source project owners, and people interested in increasing their own motivation. Table 1 shows that the community motivator has the highest precision of just 67% . This indicates that none of the motivators alone is enough to predict high motivation. However, in Section 5.5 we show that a model using all motivators achieves a precision of 81%. This indicates the additional predictive power provided by a broader perspective. To increase the motivation of a specific person, one should get familiar with and understand the motivators that the specific person values.

Recognition seems to be a powerful motivator, whose improvement predicts motivation improvement. The single question "In the past year, members of my GitHub community expressed interest in my contributions." has 20% precision lift and even 48% precision lift of motivation improvement. Unfortunately, the recall of the question is only 67%, meaning that a third of the developers report that none of their colleagues expressed interest in their work in the last year. Recognition is a powerful universal motivator that is likely to work for many populations in many contexts. This lack of interest and recognition can be easily filled and will probably lead to a beneficial impact.

An implication for motivation researchers is to demonstrate the complexity of motivation, due to the need to use more than a single motivator to predict it. Our ability to predict the improvement of motivation with an accuracy of 94% indicates that the influence of unconsidered factors is limited.

Other implications are due to our methodological contributions, providing new tools for investigation in motivation and other domains. Any survey will obtain answers that can be aggregated, correlated, and analyzed. No matter how wrong the survey is, one will get some numbers. External evidence provides a simple yet powerful way to validate the survey. We asked the participants for their GitHub profiles, which enable matching their answers with their activity. Amit and Feitelson used our data and this link and showed that the self-report of high motivation predicts retention and more diverse working hours [21]. This provided external validation on 150 thousand developers, which is more than is feasible in any survey.

The extension of co-change analysis [57] into change modeling is an important step from predictive analysis to causality analysis. Co-change is based on observational temporal data, which in many domains is available in much higher volume than intervention data. It is useful in initial exploratory analysis, identifying influential factors and the context in which they are relevant. In Section 5.5, we showed how accuracy and minimality of the model imply causality conclusions.

7. Threats to Validity

Usually, threats to construct validity are due to the measurement method. In the case of motivation, and motivators, there is a problem since the concept themselves are not well defined [42, 82] or over-defined, with 102 definitions [83]. Therefore, it is hard to measure them or evaluate how well a measurement method performs. We cope with this threat using several methods like using questions from prior work, which were already considered to be useful.

The selection of motivators and questions has subjective aspects, and others could be chosen. We based our selection on motivators with massive prior work in motivation in general, in software development, and open-source. We also compared our taxonomy to an automated objective taxonomy, derived from the answers. We found that the taxonomy was, in retrospect, supported by the data.

However, our strongest calming evidence for both construct and external validity, comes from the use of our data to validate motivation labeling functions [21]. Our data agrees with the labeling functions, which agree with the retention of 151 thousands developers working on 18,958 real projects. Hence, the answers agree with the retention of a large number of developers in their natural real work.

Investigating internal validity, some questions have systematic problems. The job satisfaction questions were answered by many participants on their day job. In self-assessment questions developers have a very high perception of themselves, not aligned with their actual performance. We therefore avoided using these answers for motivation analysis.

In order to further reduce the influence of individual questions, we grouped questions by motivators. While we still do the analysis at the question level too (available in the supplementary materials), the aggregation reduces the weight of a specific answer and makes the concepts more robust. However, answers to different questions on the same concepts are only moderately correlated (Section 5.3), so one can argue that our grouping is not correct. Indeed, we grouped questions by subjective judgment of their contents, and in principle a different taxonomy could be used. We compared our content based grouping to the one inferred from correlation (Section 5.3). The match is only partial hence our grouping is supported yet there are also different justifiable groupings.

Surveys are answered by people. Answers of the same person change over time and therefore analysis based on the original survey might not agree with the same analysis on the follow-up survey. Concerning the concept of hostility, different people provided widely different answers about the same project (Section 5.3.11). In this case this does not represent a data-quality problem, since the answers actually represent different experiences, and the difference itself is an important result.

We measured the relations between motivators and motivation in multiple ways: correlation, predictive performance, and co-change. A similar result in all methods (e.g., community increases motivation by 20%) would have been very reliable. However, there are many quantitative and even several qualitative differences in the results. For example, Table 1 shows that all motivators have positive precision lift in high motivation prediction, hence knowing of a positive motivator increases the probability of high motivation. On the other hand, in the follow-up analysis presented in Table 2, three of the eleven motivators have negative precision lift, hence knowing of their increase from the original survey indicates higher probability of motivation reduction. While negative lift is expected for hostility, the results for challenge and ideology disagree with the high motivation prediction.

Reliability was evaluated over time, using the follow-up survey, and with respect to other questions. We also investigated the validity of questions using simple mistakes, biases, and comparison to actual behavior.

Throughout this research we obtained many results. While our number of participants is very high for a survey, we analyzed the answers in many ways. In some scenarios (e.g., developers in the same project, developers answering the follow-up), the numbers are quite small. Statistical learning theory [80, 81] tells us that in such cases several of the empirical results will probably be different from the actual ones. This is an inherited threat from the dataset size and analysis type, which should be resolved by replication studies obtaining more data and supporting the results in different analyses.

Our survey population came from two sources - GitHub developers accessed by email and developers accessed using social media. We used machine learning to see how different these populations are and could not find a model differing them better than the positive rate. This means that there is no obvious big difference between the populations. Similarity between the participants group and the desired population increases the probability of generalization. We provide demographics analysis and a comparison to the Stack Overflow survey (Section 5.1). Though our participants resemble the Stack Overflow survey participants (while being somewhat more professional), it is not clear what the general developers group is (details in the supplementary materials).

8. Discussion and Conclusions

We conducted a large survey of software developers regarding motivation. We grouped the questions by motivators and analyzed their relation to motivation.

Our supervised-learning-based analysis of motivators ranged from using each motivator as a classifier for motivation, through using motivator improvement as a classifier for motivation improvement, to constructing models that use the combined power of multiple motivators for improved prediction. We confirmed that previously suggested motivators do indeed contribute to motivation. At the same time, the influence of each individual motivator is limited, as also noted in prior work [24, 23].

Apparently, the motivation of different developers, working in different contexts, may be influenced by different motivators. No single motivator by itself is sufficient for inducing high motivation. At the same time, none of the motivators is strictly necessary. An analysis of the relations between them indicated that motivators tend to have low correlation. This indicates that one should not look at motivators from the prism of which is the "most important" one; a better description is that each one of them captures a different aspect of motivation [42], and multiple aspects should be satisfied in order to have high motivation.

It is also interesting to notice the relative position of payment. Trying to predict high motivation based on a single motivator, payment has precision lift of 10%, the lowest value of all positive motivators, and the only negative lift on the follow-up survey. Since payment is the common way to promote motivation in businesses, it is important to note that other motivators might lead to a larger effect.

Different people in the same project disagree on hostility, implying that it is not noticed by others. Hence, not noticing hostility is not enough to assure lack of hostility in a project, and therefore actively looking for it might be needed.

Our survey also provided another motivation to investigate motivation. 73% of the participants answered that motivation has more influence on their productivity (answers higher than neutral), and only 9% answered that their skill is more influential. This agrees with prior work on the benefit of motivation for productivity [84].

Participants who reported an improvement in the interest expressed in them had a large tendency for improvement in motivation. Recognition, and specifically expressing interest, is free, applicable in all situations, and influential. Considerate behavior and looking for practical benefits coincide here. Be kind and give recognition, it is likely to pay off.

Data Availability

All experimental materials (except for identifying data such as emails and GitHub profiles) is available at [47].

References

[1] S. A. Frangos, Motivated humans for reliable software products, in: D. Gritzalis (Ed.), Reliability, Quality and Safety of Software-Intensive Systems: IFIP TC5 WG5.4 3rd International Conference on Reliability, Quality and Safety of Software-Intensive Systems, Springer US, Boston, MA, 1997, pp. 83–91. `doi:10.1007/978-0-387-35097-4\_7`. URL `https://doi.org/10.1007/978-0-387-35097-4_7`

[2] D. Ariely, E. Kamenica, D. Prelec, Man's search for meaning: The case of legos, Journal of Economic Behavior & Organization 67 (3-4) (2008) 671–677. doi:10.1016/j.jebo.2008.01.004.

[3] C. F. Lam, S. T. Gurland, Self-determined work motivation predicts job outcomes, but what predicts self-determined work motivation?, Journal of Research in Personality 42 (4) (2008) 1109–1115. doi:10.1016/j.jrp.2008.02.002.
URL http://www.sciencedirect.com/science/article/pii/S0092656608000214

[4] S. Beecham, N. Baddoo, T. Hall, H. Robinson, H. Sharp, Motivation in software engineering: A systematic literature review, Information & Software Technology 50 (9-10) (2008) 860–878. doi:10.1016/j.infsof.2007.09.004.
URL https://doi.org/10.1016/j.infsof.2007.09.004

[5] H. Sharp, T. Hall, An initial investigation of software practitioners' motivation, in: Proceedings of the ICSE Workshop on Cooperative and Human Aspects on Software Engineering, 2009, pp. 84–91. doi:10.1109/CHASE.2009.5071418.
URL http://dx.doi.org/10.1109/CHASE.2009.5071418

[6] E. S. Raymond, The Cathedral and the Bazaar, 1st Edition, O'Reilly & Associates, Inc., Sebastopol, CA, USA, 1999.

[7] H. Sharp, N. Baddoo, S. Beecham, T. Hall, H. Robinson, Models of motivation in software engineering, Information & Software Technology 51 (1) (2009) 219–233. doi:10.1016/j.infsof.2008.05.009.
URL https://doi.org/10.1016/j.infsof.2008.05.009

[8] S. Belenzon, M. Schankerman, Motivation and Sorting in Open Source Software Innovation, CEP Discussion Papers dp0893, Centre for Economic Performance, LSE (Oct. 2008).
URL https://ideas.repec.org/p/cep/cepdps/dp0893.html

[9] M. Gerosa, I. Wiese, B. Trinkenreich, G. Link, G. Robles, C. Treude, I. Steinmacher, A. Sarma, The shifting sands of motivation: Revisiting what drives contributors in open source, in: Proceedings of the 43rd International Conference on Software Engineering, 2021, pp. 1046–1058.

doi:10.1109/ICSE43902.2021.00098.
URL https://doi.org/10.1109/ICSE43902.2021.00098

[10] M. Argyle, Do happy workers work harder? the effect of job satisfaction on job performance, in: R. Veenhoven (Ed.), How harmful is happiness? Consequences of enjoying life or not, Universitaire Pers, Rotterdam, The Netherlands, 1989, pp. 94–105.

[11] D. C. McClelland, The Achieving Society, Van Nostrand, Princeton, NJ, 1961.

[12] E. A. Locke, Toward a theory of task motivation and incentives, Organizational Behavior and Human Performance 3 (2) (1968) 157–189. doi:10.1016/0030-5073(68)90004-4.
URL http://www.sciencedirect.com/science/article/pii/0030507368900044

[13] N. Baddoo, T. Hall, Motivators of software process improvement: an analysis of practitioners' views, Journal of Systems and Software 62 (2) (2002) 85–96. doi:10.1016/S0164-1212(01)00125-X.
URL http://www.sciencedirect.com/science/article/pii/S016412120100125X

[14] G. Asproni, Motivation, teamwork, and agile development, Agile Times 4 (Mar 2004).

[15] Y. Ye, K. Kishida, Toward an understanding of the motivation open source software developers, in: Proceedings of the 25th International Conference on Software Engineering, 2003, pp. 419–429. doi:10.1109/ICSE.2003.1201220.

[16] N. Raman, M. Cao, Y. Tsvetkov, C. Kästner, B. Vasilescu, Stress and burnout in open source: Toward finding, understanding, and mitigating unhealthy interactions, in: Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: New Ideas and Emerging Results, 2020, pp. 57–60. doi:10.1145/3377816.3381732.
URL https://doi.org/10.1145/3377816.3381732

[17] J. S. Adams, Toward an understanding of inequity, Journal of Abnormal Psychology 67 (5) (1963) 422–436. doi:10.1037/h0040968.

[18] J. R. Hackman, G. R. Oldman, Motivation Through the Design of Work: Test of a Theory, Academic Press, New York, 1976.

[19] R. Bénabou, J. Tirole, Intrinsic and extrinsic motivation, The Review of Economic Studies 70 (3) (2003) 489–520.

[20] D. Riehle, The economic motivation of open source software: Stakeholder perspectives, Computer 40 (4) (2007) 25–32. doi:10.1109/MC.2007.147.

[21] I. Amit, D. G. Feitelson, Motivation research using labeling functions, in: Proceedings of the 28th International Conference on Evaluation and Assessment in Software Engineering, EASE '24, Association for Computing Machinery, New York, NY, USA, 2024, p. 222–231. doi:10.1145/3661167.3661224.
URL https://doi.org/10.1145/3661167.3661224

[22] J. Fitz-Enz, Who is the dp professional, Datamation 24 (9) (1978) 125–128.

[23] J. D. Couger, Motivators vs. demotivators in the is environment, Journal of Systems Management 39 (6) (1988) 36.

[24] F. Herzberg, One more time: How do you motivate employees?, New York: The Leader Manager (1986) 433–448.

[25] B. F. Skinner, The Behavior of Organisms: An Experimental Analysis, New York: Appleton-Century-Crofts, 1938.

[26] A. H. Maslow, A theory of human motivation, Psychological Review 50 (1943) 370–396.

[27] V. H. Vroom, Work and Motivation, Wiley, New York, 1964.

[28] F. Herzberg, B. Mausner, B. B. Snyderman, Motivation to Work, Wiley, New York, 1959.

[29] J. Irvine, A framework for comparing theories related to motivation in education., Research in Higher Education Journal 35 (2018).

[30] N. Bassett-Jones, G. C. Lloyd, Does herzberg's motivation theory have staying power?, Journal of Management Development 24 (2005) 929–943. doi:10.1108/02621710510627064.

[31] T. DeMarco, T. Lister, Peopleware: Productive Projects and Teams, Dorset House, 1987.

[32] P. Lenberg, R. Feldt, L. G. Wallgren, Behavioral software engineering: A definition and systematic literature review, Journal of Systems and Software 107 (2015) 15–37. doi:10.1016/j.jss.2015.04.084.
URL http://www.sciencedirect.com/science/article/pii/S0164121215000989

[33] C. França, F. Q. B. da Silva, H. Sharp, Motivation and satisfaction of software engineers, IEEE Transactions on Software Engineering 46 (2) (2020) 118–140. doi:10.1109/TSE.2018.2842201.

[34] S. A. K. Ghayyur, S. Ahmed, S. Ullah, W. Ahmed, The impact of motivator and demotivator factors on agile software development, International Journal of Advanced Computer Science and Applications 9 (7) (2018). doi:10.14569/IJACSA.2018.090712.
URL http://dx.doi.org/10.14569/IJACSA.2018.090712

[35] J. Lerner, J. Tirole, Some simple economics of open-source, Journal of Industrial Economics 50 (2002) 197–234.
URL http://www.people.hbs.edu/jlerner/simple.pdf

[36] A. B. Cristina, C. Rossi, Altruistic individuals, selfish firms? the structure of motivation in open source software, First Monday 9 (2004) 9. doi:10.5210/fm.v9i1.1113.

[37] Y. Li, C.-H. Tan, H. Teo, T. Mattar, Motivating open source software developers: Influence of transformational and transactional leaderships, in: Proceedings of the 2006 ACM SIGMIS CPR Conference on Computer Personnel Research: Forty Four Years of Computer Personnel Research: Achievements, Challenges & the Future, 2006, pp. 34–43. doi:10.1145/1125170.1125182.

[38] G. Hertel, S. Niedner, S. Herrmann, Motivation of software developers in open source projects: an internet-based survey of contributors to

the linux kernel, Research Policy 32 (7) (2003) 1159–1177, open Source Software Development. doi:10.1016/S0048-7333(03)00047-7.
URL http://www.sciencedirect.com/science/article/pii/S0048733303000477

[39] J. A. Roberts, I.-H. Hann, S. A. Slaughter, Understanding the motivations, participation, and performance of open source software developers: A longitudinal study of the apache projects, Manage. Sci. 52 (7) (2006) 984–999. doi:10.1287/mnsc.1060.0554.
URL https://doi.org/10.1287/mnsc.1060.0554

[40] A. Grant, The significance of task significance: Job performance effects, relational mechanisms, and boundary conditions, The Journal of Applied Psychology 93 (2008) 108–24. doi:10.1037/0021-9010.93.1.108.

[41] J. P. Campbell, R. A. McCloy, S. H. Oppler, C. E. Sager, A theory of performance, in: N. Schmitt, W. C. Borman, Associates (Eds.), Personnel Selection in Organizations, Jossey-Bass Pub., 1993, pp. 35–70.

[42] J. D. Mayer, M. A. Faber, X. Xu, Seventy-five years of motivation measures (1930–2005): A descriptive analysis, Motivation and Emotion 31 (2007) 83–103.

[43] I. Chengalur-Smith, A. Sidorova, S. L. Daniel, Sustainability of free/libre open source projects: A longitudinal study, Journal of the Association for Information Systems 11 (11) (2010) 5.

[44] I. Steinmacher, M. A. G. Silva, M. A. Gerosa, D. F. Redmiles, A systematic literature review on the barriers faced by newcomers to open source software projects, Information and Software Technology 59 (2015) 67–85.

[45] C. Hannebauer, V. Gruhn, On the relationship between newcomer motivations and contribution barriers in open source projects, in: Proceedings of the 13th International Symposium on Open Collaboration, 2017, pp. 1–10.

[46] C. Hannebauer, V. Gruhn, Motivation of newcomers to floss projects, in: Proceedings of the 12th International Symposium on Open Collaboration, 2016, pp. 1–10.

[47] I. Amit, D. G. Feitelson, Replication package https://github.com/evidencebp/motivation-survey (Dec 2023).

[48] E. Murphy-Hill, C. Jaspan, C. Sadowski, D. Shepherd, M. Phillips, C. Winter, A. Knight, E. Smith, M. Jorde, What predicts software developers' productivity?, IEEE Transactions on Software Engineering 47 (3) (2021) 582–594. `doi:10.1109/TSE.2019.2900308`.

[49] D. Hills, C. Joyce, J. Humphreys, Validation of a job satisfaction scale in the australian clinical medical workforce, Evaluation & the Health Professions 35 (2011) 47–76. `doi:10.1177/0163278710397339`.
URL `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.916.1150&rep=rep1&type=pdf`

[50] H. Shamon, C. Berning, Attention check items and instructions in online surveys with incentivized and non-incentivized samples: Boon or bane for data quality?, Shamon, H., & Berning, CC (2020). Attention Check Items and Instructions in Online Surveys with Incentivized and Non-Incentivized Samples: Boon or Bane for Data Quality (2019) 55–77.

[51] R. Likert, A technique for the measurement of attitudes., Archives of Psychology 22 (1932) 55.

[52] A. Joshi, S. Kale, S. Chandel, D. K. Pal, Likert scale: Explored and explained, British Journal of Applied Science & Technology 7 (4) (2015) 396. `doi:10.9734/BJAST/2015/14975`.

[53] S.-O. Leung, A comparison of psychometric properties and normality in 4-, 5-, 6-, and 11-point likert scales, Journal of social service research 37 (4) (2011) 412–421.

[54] S. Baltes, P. Ralph, Sampling in software engineering research: A critical review and guidelines, Empirical Software Engineering 27 (4) (2022) 94.

[55] Github user search showing 37,446,292 available users, accessed: 2019-05-23.
URL `https://github.com/search?q=type:user&type=Users`

[56] I. Amit, D. G. Feitelson, Which refactoring reduces bug rate?, in: Proceedings of the 15th International Conference on Predictive Models

and Data Analytics in Software Engineering, PROMISE'19, Association for Computing Machinery, New York, NY, USA, 2019, pp. 12–15. doi:10.1145/3345629.3345631.
URL https://doi.org/10.1145/3345629.3345631

[57] I. Amit, D. G. Feitelson, Corrective commit probability: A measure of the effort invested in bug fixing, Software Quality Journal 29 (4) (2021) 817–861. doi:10.1007/s11219-021-09564-z.
URL https://doi.org/10.1007/s11219-021-09564-z

[58] I. Amit, End to end software engineering research (2021). arXiv:2112.11858.

[59] I. Etikan, S. A. Musa, R. S. Alkassim, et al., Comparison of convenience sampling and purposive sampling, American Journal of Theoretical and Applied Statistics 5 (1) (2016) 1–4. doi:10.11648/j.ajtas.20160501.11.

[60] A. S. Acharya, A. Prakash, P. Saxena, A. Nigam, Sampling: Why and how of it, Indian Journal of Medical Specialties 4 (2) (2013) 330–333. doi:10.7713/ijms.2013.0032.

[61] Y. Huang, D. Ford, T. Zimmermann, Leaving my fingerprints: Motivations and challenges of contributing to oss for social good.

[62] D. G. Feitelson, "we do not appreciate being experimented on": Developer and researcher views on the ethics of experiments on open-source projects, arXiv 2112.13217 [cs.SE] (2021). arXiv:2112.13217.

[63] A. Blum, T. Mitchell, Combining labeled and unlabeled data with co-training, in: Proceedings of the Eleventh Annual Conference on Computational Learning Theory, COLT' 98, ACM, New York, NY, USA, 1998, pp. 92–100. doi:10.1145/279943.279962.
URL http://doi.acm.org/10.1145/279943.279962

[64] A. J. Ratner, C. M. De Sa, S. Wu, D. Selsam, C. Ré, Data programming: Creating large training sets, quickly, in: D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, R. Garnett (Eds.), Advances in Neural Information Processing Systems 29, Curran Associates, Inc., 2016, pp. 3567–3575.
URL http://papers.nips.cc/paper/6523-data-programming-creating-large-training
pdf

[65] I. Amit, E. Firstenberg, Y. Meshi, Framework for semi-supervised learning when no labeled data is given, U.S. patent #US11468358B2 (2017). URL https://patents.google.com/patent/US11468358B2

[66] Stackoverflow, 2021 annual developer survey (2021). URL https://insights.stackoverflow.com/survey/2021

[67] J. Terrell, A. Kofink, J. Middleton, C. Rainear, E. Murphy-Hill, C. Parnin, J. Stallings, Gender differences and bias in open source: Pull request acceptance of women versus men, PeerJ Computer Science 3 (2017). doi:10.7717/peerj-cs.111.

[68] G. Robles, L. Arjona Reina, A. Serebrenik, B. Vasilescu, J. M. González-Barahona, Floss 2013: A survey dataset about free software contributors: Challenges for curating, sharing, and combining, in: Proceedings of the 11th Working Conference on Mining Software Repositories, 2014, pp. 396–399. doi:10.1145/2597073.2597129. URL https://doi.org/10.1145/2597073.2597129

[69] D. G. Beckers, D. van der Linden, P. G. Smulders, M. A. Kompier, M. J. van Veldhoven, N. W. van Yperen, Working overtime hours: relations with fatigue, work motivation, and the quality of work, Journal of Occupational and Environmental Medicine (2004) 1282–1289.

[70] R. Stallman, Why software should be free (2007). URL https://www.gnu.org/philosophy/shouldbefree.en.html

[71] A. Kluger, O. Bouskila-Yam, Facilitating listening scale (FLS), in: D. L. Worthington, G. D. Bodie (Eds.), The Sourcebook of Listening Research: Methodology and Measures, Wiley Online Library, 2017, pp. 272–280. doi:10.1002/9781119102991.ch25.

[72] L. Cortina, V. Magley, J. Williams, R. Langhout, Incivility in the workplace: Incidence and impact, Journal of Occupational Health Psychology 6 (2001) 64–80. doi:10.1037/1076-8998.6.1.64.

[73] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, Journal of

Machine Learning Research 12 (2011) 2825–2830.
URL http://www.jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf

[74] D. Arpit, S. Jastrzębski, N. Ballas, D. Krueger, E. Bengio, M. S. Kanwal, T. Maharaj, A. Fischer, A. Courville, Y. Bengio, S. Lacoste-Julien, A closer look at memorization in deep networks, in: 34th International Conference on Machine Learning, PMLR, 2017, pp. 233–242.

[75] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444. doi:10.1038/nature14539.

[76] P. F. Verhulst, Resherches mathematiques sur la loi d'accroissement de la population, Nouveaux Memoires de l'Academie Royale des Sciences 18 (1845) 1–41.

[77] J. R. Quinlan, C4. 5: programs for machine learning, Elsevier, 2014.

[78] B. Zadrozny, C. Elkan, Transforming classifier scores into accurate multiclass probability estimates, in: Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2002, pp. 694–699. doi:10.1145/775047.775151.
URL https://doi.org/10.1145/775047.775151

[79] Y. Freund, R. E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, Journal of Computer and System Sciences 55 (1) (1997) 119–139. doi:10.1006/jcss.1997.1504.
URL https://www.sciencedirect.com/science/article/pii/S002200009791504X

[80] V. Vapnik, A. Chervonenkis, On the uniform convergence of relative frequencies of events to their probabilities, Theory of Probability & Its Applications 16 (2) (1971) 264–280. doi:10.1137/1116025.
URL https://doi.org/10.1137/1116025

[81] V. Vapnik, The Nature of Statistical Learning Theory, Springer science & business media, 2013.

[82] K. Toode, P. Routasalo, T. Suominen, Work motivation of nurses: A literature review, International Journal of Nursing Studies 48 (2) (2011) 246–257. doi:https://doi.org/10.1016/j.ijnurstu.2010.09.013.

URL https://www.sciencedirect.com/science/article/pii/S0020748910003147

[83] P. R. Kleinginna Jr, A. M. Kleinginna, A categorized list of motivation definitions, with a suggestion for a consensual definition, Motivation and emotion 5 (3) (1981) 263–291.

[84] J. A. Shepperd, Productivity loss in performance groups: A motivation analysis., Psychological Bulletin 113 (1) (1993) 67. doi:10.1037/0033-2909.113.1.67.

Acknowledgments

First, we would like to thank our participants. Other than just answering the survey, they alerted on problems and suggested many ideas. Many of them left their email to receive the research results which is heartwarming. We could not conduct this research without you. You are our partners. Thank you!

We thank the IRB committee for their feedback and approval of the study (09032020).

This research was supported by the ISRAEL SCIENCE FOUNDATION (grant no. 832/18).

We thank David Amit for the support, interest, and many wise ideas and questions. We thank Avraham Kluger who introduced to us the importance of listening and helped in many ways, and to Dan Ariely who guided us into the domain of motivation. We also thank Tali Kleiman, Itamar Gati, Zafrir Buchler, Doody Parizada, Eyal Zaidman, Yaniv Mama, Asaf Korem, and Nili Ben Ezra for the discussions, insights, and help.