Personality Traits of GitHub Maintainers and Their Effects on Project Success

by

Seonghu Yun

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2020

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

The entirety of the thesis was written by myself, including the generation of all figures and tables. Dr. Gema Rodríguez-Pérez identified the maintainers of GitHub repositories and assisted with the modelling, while Dr. Zahra Sheikhbahaee assisted with the clustering analysis. Dr. Rodríguez-Pérez and Dr. Sheikhbahaee are postdoctoral fellows at the David R. Cheriton School of Computer Science University of Waterloo. This thesis also includes content from a publication which I have co-authored:

- Rahul N. Iyer, S. Alex Yun, Meiyappan Nagappan and Jesse Hoey, "Effects of Personality Traits on Pull Request Acceptance," in *IEEE Transactions on Software Engineering*, doi: 10.1109/TSE.2019.2960357.

Sections 1, 2.1 and 5.1 of the paper are partially adopted into sections 1, 2.2 and 5.1 of the thesis.

Abstract

Online collaborative environments have become important virtual workplaces for developers to work on a common problem. GitHub is an example of such environment that hosts a wealth of open source software projects. Questions such as “Who contributes to successful projects?” and “What are the characteristics of lead developers?” require further investigations.

We qualitatively identify 211 maintainers in 25 maintained repositories and 23 unmaintained repositories in GitHub. We measure their Big Five personality traits (Openness, Conscientiousness, Extraversion, Agreeableness, and Neuroticism) as the weighted sum of their Linguistic Inquiry and Word Count (LIWC) dimensions. Our results indicate that maintainers and non-maintainers are significantly different in virtually all personality traits except in Neuroticism. Maintainers in maintained repositories tend to be more open, but less extraverted and less agreeable than maintainers in unmaintained repositories. In addition to Agreeableness being a significant predictor, our analysis suggest that the success of a repository may be explained by the absolute differences in personality traits between maintainers and non-maintainers.

In sum, our work aims to understand the role of a maintainer and the effects of personality traits on project success. Our findings have direct implications such that developers can be more cognizant of their behaviours, as well as their colleagues, which can result in better collaboration. By highlighting personality differences, we show that studying social and psychological constructs can be invaluable in understanding group dynamics during collaborative process.

Acknowledgements

I would like to first thank my supervisor Prof. Jesse Hoey for his support and guidance. Jesse is a true Renaissance man: his studies and expertise spanning from physics, oceanography, computer science (artificial intelligence), and to sociology. His voracious curiosity and interdisciplinary approaches to the scientific endeavour have inspired on a daily basis.

I would also like to thank my readers Prof. Meiyappan Nagappan and Prof. Edith Law for providing valuable feedback.

I am grateful to have worked with amazing collaborators Dr. Gema Rodríguez-Pérez, Dr. Zahra Sheikhbahaee, and Rahul Iyer.

Lastly, I would be remiss not to thank the amazing people at the CHIL, especially Josh Jung, Alex Sachs, and Aaron Li for useful discussions.

Dedication

This is dedicated to my mum and dad who have always encouraged me to dream.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

API application programming interface 16

EM expectation-maximization 14

FFM five-factor model 6

GLMM generalized linear mixed model 23

GMMs gaussian mixture models 13

IPIP International Personality Item Pool 6

KS Kolmogorov–Smirnov 23

KW Kruskal–Wallis 23

LIWC Linguistic Inquiry and Word Count 3

MBTI Myers–Briggs Type Indicator 11

MPPCA mixtures of probabilistic principal component analyzers 14

MWU Mann–Whitney U 23

NEO PI-R Revised NEO Personality Inventory 6

OCE online collaborative environment 1

OSS open source software 1

PCA principal component analysis 13

REST representational state transfer 16

VIF variance inflation factor 25

Chapter 1

Introduction

GitHub[1] is a popular online collaborative environment (OCE) that hosts many open source software (OSS) projects. It has accounted for more than 10 million new developers who contributed to more than 4 million repositories in 2019.[2] When working in OSS projects, hierarchies naturally form and developers start to assume different roles. Aside from being the owner of a repository, developers are divided into two categories: contributors and collaborators. A contributor is someone who does not belong to the core team of a project, but can contribute with limited access. A collaborator, on the other hand, is invited to the core team by the owner and has commit access. By having commit access, they are able to make revisions to a file or set of files directly within the project. Collaborators can have various roles as specified by their permission levels within the project; one of these roles is that of a maintainer who has merge privileges. This means that when a developer submits a contribution in the form of a pull request and gets accepted, the maintainer can merge this set of changes into the project on behalf of the developer who does not have merge privileges.

Given the loosely bound hierarchical organization that forms in GitHub projects, it is reasonable to suspect that OCEs may share resemblance to its offline counterparts—that is, any workplace that occupies a physical space where employees work towards a common set of goals. The fact that maintainers are selectively chosen and hold important responsibilities in a project, their roles appear to be very similar to team leads and/or managers in an office setting. By studying maintainers, we were motivated to examine how leadership affects the success or failure of OSS projects. We believe that effective

[1] https://github.com/

[2] https://octoverse.github.com/

maintainers, particularly those that are competent and virtuous, can lead to successful projects because they:

1. can encourage and lead contributors by example;
2. ensure smooth workflow by organizing and prioritizing tasks;
3. inspire contributors by demonstrating the importance and impact of their work;
4. make final decisions.

In addition to studying leadership in OCEs, we were motivated to understand the role of non-technical factors in OSS projects. Feldt et al. [22], for example, outlined the importance of human factors and champion the collection of psychometric data to gain new insights in the field of software engineering. Tsay, Dabbish, and Herbsleb [66] showed that project managers in GitHub not only use technical factors, but also social factors when evaluating pull requests. This led us to believe that studying personality traits of developers may be invaluable in understanding group dynamics during collaborative process. This is a widely explored topic in the field of industrial and organizational psychology to understand job performance, motivation, and leadership in the workplace. In our recent work, we replicated Tsay et al.'s work, reconfirming the importance of social factors on the pull request evaluation process [29]. Furthermore, we showed that the effects of personality traits on pull request acceptance are not only significant but comparable to the effects of technical factors. In particular, we found that pull requests that were requested by developers that are more open and conscientious, but less extraverted are more likely to be accepted. We also found that pull requests that were closed by developers who are more conscientious, extraverted, and neurotic are more likely to be accepted.

Along with examining individual personality traits, it is useful to examine how one's personality traits may differ from other members in the group. The question "Do birds of a feather flock together?" is a longstanding question in social psychology [10] and has been studied extensively across many different fields. In artificial intelligence and robotics, it was found that users in rehabilitation preferred assistive robots that matched their own personality, which in turn could increase therapeutic goals [62]. Conversely, individuals found robots that have complimentary personalities to be more intelligent and socially present [35]. Studies on pair programming found that variability in personality did not have a great impact on performance in an academic setting [54] or an office setting [26]. Our own findings showed that greater personality difference between the requester and the closer led to more positive effect on pull request acceptance [29]. These contradictory findings suggest that personality alone is often not sufficient to predict human behaviours, but the combination of personality and context provides a better understanding of how one will behave in different situations.

The diversity of GitHub will be reflected by developers exhibiting different behaviours, characterized by varying degrees of personality. We can observe behaviours by extracting their 'digital footprints' or comments, infer personality traits, and examine how they influence the functioning and success of OSS projects. Some contributors and collaborators may have strict and high standards of coding, leading to meaningful contributions. Similarly, some collaborators and particularly maintainers may be encouraging and readily available to assist outside contributors.

1.1 Research Questions and Contributions

To better understand the role of a maintainer and the effects of personality traits on project success, we pose the following research questions:

- RQ0: Do maintainers and non-maintainers show difference in personality traits?
- RQ1: Do maintainers in maintained repositories show difference in personality traits from maintainers in unmaintained repositories?
- RQ2: What is the relationship between maintainers' personality traits and the success of a repository?
- RQ3: What is the relationship between maintainers' personality traits and the popularity of a repository?

While the metric for repository success can be defined in multiple ways, we will consider a repository to be successful if it is maintained and unsuccessful if it is unmaintained or archived. We qualitatively identify 211 maintainers in 25 maintained and 23 unmaintained repositories in GitHub. We measure their 'Big Five' personality traits (Openness, Conscientiousness, Extraversion, Agreeableness, and Neuroticism) as the weighted sum of their Linguistic Inquiry and Word Count (LIWC) dimensions. The primary contributions of this thesis are:

1. Empirical evidence showing the personality difference between maintainers and non-maintainers.

2. Empirical evidence showing the personality difference between maintainers in successful projects and maintainers in unsuccessful projects, and the positive effects of personality difference on project success.

3. Discussions led by evidence from psychological literature.

1.2 Thesis Organization

The rest of the thesis is organized as follows:

- Chapter 2 presents relevant background information and related work for the thesis. We start with an overview of GitHub and theories of personality. After discussing various tools to infer personality traits from text, we review software engineering studies that involve utilizing aspects of personality traits. Lastly, we conclude the chapter by discussing the use of clustering algorithms and its details.

- Chapter 3 outlines detailed methodology on how the dataset was curated and prepared for the data analysis.

- Chapter 4 reports the results of our empirical analyses.

- Chapter 5 describes our findings and its implications, providing supporting evidence from personality and industrial-organizational psychology. We reflect on possible limitations and provide ideas for future research avenues.

- Chapter 6 concludes the thesis by summarizing our work and provides a closing thought.

Chapter 2

Background and Related Work

In this chapter we briefly introduce relevant background information, specifically on the functioning and the workflow of GitHub, as well as theories of personality. Furthermore, we examine tools that have been developed to infer personality traits from text and look at related work, focusing on the study of personality in software engineering.

2.1 GitHub

Using Git as the basis, GitHub not only provides a platform to host OSS projects but it works as a version control system that uses characteristic workflow. Developers can *fork* a repository of their interest, meaning they make a personal copy to their GitHub accounts or local machines. In order to make a contribution, developers typically create a new *branch* and make changes to files and codes in their forked or downstream repository. This contribution can be submitted in the form of a *pull request*, which will notify the changes to the maintainers of the original or upstream repository. Once the pull request is open, it initiates a discussion between the author of the pull request and collaborators, where they can review the pull request and suggest further changes. The maintainers can either accept or reject during this pull request evaluation process. Once accepted, this branch that contains new changes will be merged into the *master* branch of the repository.

In addition to pull requests, another useful feature of GitHub is the use of *issues*. It is the main way for developers to prioritize tasks and keep track of bugs to improve the functioning of projects within the repository. Pull requests can linked to issues to show fellow developers that a progress is being made for a specific issue. Along with technical features, GitHub provides convenient social-networking features. Users can create profiles and can follow other developers on the platform. They can also *watch* a repository and will receive notifications when issues and pull requests are updated. Lastly, users can bookmark a repository by using the *star* feature, which is often used as an indirect measure of the repository's popularity.

2.2 Theories of Personality

What constitutes as personhood depends on one's theoretical perspectives in philosophy and psychology; as such, this holds true with the definition of personality. Nevertheless, personality describes a stable way in which an individual interacts with oneself, with others, and with the world. The study of personality is aimed to understand human nature by examining individual differences in behaviour and in turn using these characteristics to make behavioural predictions [27]. Not surprisingly, personality psychologists are interested in developing theories and models to accurately capture an individual's disposition and their psychological processes, along with developing reliable and valid psychometric measurements.

Modern personality theories are deeply rooted in the idea of the lexical hypothesis, which posits that any behavioural descriptors of personality have been recorded in the human language [3, 15]. Using these descriptors as a lexical database, several research groups performed factor analysis and arrived at five distinctive factors or traits (for a review, see [20]). This forms the basis of the five-factor model (FFM) or the Big Five. To measure these five traits, personality inventories such as the Revised NEO Personality Inventory (NEO PI-R) [18] and International Personality Item Pool (IPIP) [24] are administered. These inventories typically have self-descriptive sentences and are answered on a five-point Likert scale (i.e., the description ranges from very inaccurate to very accurate about oneself):

- Am full of ideas.
- Pay attention to details.
- Don't like to draw attention to myself.
- Am not really interested in others.
- Seldom feel blue.

They can come in a variety of formats, including short and long versions, and self-reported or observer ratings. These inventories are empirically well validated, showing good reliability [67], validity [39] and consistency across cultures [57]. Adopting the definitions from the 10 Aspects scale [19], the main Big Five traits are:

- *Openness to Experience* is a measure of intellect and openness.
  - Individuals who score high on Openness to experience show ingenuity and enjoy having abstract discussions.
  - This trait will be referred to as Openness hereafter.
- *Conscientiousness* is a measure of industriousness and orderliness.
  - Individuals who score high on Conscientiousness are dutiful and detail-oriented.
- *Extraversion* is a measure of assertiveness and enthusiasm.
  - Individuals who score high on Extraversion are gregarious and dominant in social situations.
- *Agreeableness* is a measure of compassion and politeness.
  - Individuals who score high on Agreeableness regard emotional affiliation with others important and are compliant.
- *Neuroticism* is a measure of volatility and withdrawal.
  - Individuals who score high on Neuroticism tend to be sensitive to negative emotions.

2.3 Inferring Personality Traits from Text

Given the lexical hypothesis, it is reasonable to assume that one's personality traits can be inferred from their written language. The traditional survey methods may be effective but can be costly and time-consuming. As a natural progression, researchers started developing psycholoinguistic tools that process raw text to obtain personality traits. One of the earliest such tools is the the LIWC developed by the social psychologist Pennebaker [63]. LIWC uses a closed vocabulary approach, featuring a default dictionary of 6,400 words [47]. *Target words* that are contained in the raw text are searched against the LIWC dictionary, categorized, and counted according to their frequencies. These word categories include summary language variables, linguistic dimensions, grammar, and psychological processes. Several studies have shown correlations between LIWC dimensions and Big Five personality traits [48, 41].

LIWC dimensions became a building block in many following psycholinguistic tools. Building upon previous studies that focused on associations between personality and word use, Yarkoni [70] led a large-scale analysis on bloggers. He used a survey-based method and collected personality traits of bloggers by sending out versions of the IPIP measures. He showed a strong association between personality and word use, and proposed a set of equations to infer personality traits from LIWC dimensions. We adopt the slightly modified equations from Calefato et al. [11]:

$$\begin{aligned} Openness = {} & 0.2 \cdot article + 0.17 \cdot prep + 0.15 \cdot death - 0.21 \cdot pronoun - 0.16 \cdot i \\ & - 0.1 \cdot we - 0.12 \cdot you - 0.13 \cdot negate - 0.11 \cdot assent - 0.12 \cdot affect \\ & - 0.15 \cdot posemo - 0.09 \cdot cogproc - 0.12 \cdot discrep - 0.08 \cdot hear \\ & - 0.14 \cdot social - 0.17 \cdot family - 0.22 \cdot time - 0.16 \cdot focuspast \\ & - 0.16 \cdot focuspresent - 0.11 \cdot space - 0.22 \cdot motion \\ & - 0.17 \cdot leisure - 0.2 \cdot home - 0.15 \cdot ingest \end{aligned} \tag{2.1}$$

$$\begin{aligned} Conscientiousness = {} & 0.09 \cdot time + 0.14 \cdot achieve - 0.17 \cdot negate - 0.18 \cdot negemo \\ & - 0.19 \cdot anger - 0.11 \cdot sad - 0.11 \cdot cogproc - 0.12 \cdot cause \\ & - 0.13 \cdot discrep - 0.1 \cdot tentat - 0.1 \cdot certain - 0.12 \cdot hear \\ & - 0.12 \cdot death - 0.14 \cdot swear \end{aligned} \tag{2.2}$$

$$\begin{aligned} Extraversion = {} & 0.11 \cdot we + 0.16 \cdot you + 0.1 \cdot posemo + 0.1 \cdot certain + 0.12 \cdot hear \\ & + 0.15 \cdot social + 0.15 \cdot friend + 0.09 \cdot family + 0.08 \cdot leisure \\ & + 0.11 \cdot relig + 0.1 \cdot body + 0.17 \cdot sexual - 0.12 \cdot number \\ & - 0.09 \cdot cause - 0.11 \cdot tentat - 0.08 \cdot work - 0.09 \cdot achieve \end{aligned} \tag{2.3}$$

$$
\begin{aligned}
Agreeableness = {} & 0.11 \cdot pronoun + 0.18 \cdot we + 0.11 \cdot number + 0.18 \cdot posemo + 0.09 \cdot see \\
& + 0.1 \cdot feel + 0.13 \cdot social + 0.11 \cdot friend + 0.19 \cdot family + 0.12 \cdot time \\
& + 0.1 \cdot focuspast + 0.16 \cdot space + 0.14 \cdot motion + 0.15 \cdot leisure \\
& + 0.19 \cdot home + 0.09 \cdot body + 0.08 \cdot sexual - 0.15 \cdot negemo - 0.23 \cdot anger \\
& - 0.11 \cdot cause - 0.11 \cdot money - 0.13 \cdot death - 0.21 \cdot swear
\end{aligned}
\tag{2.4}
$$

$$
\begin{aligned}
Neuroticism = {} & 0.12 \cdot i + 0.11 \cdot negate + 0.16 \cdot negemo + 0.17 \cdot anx + 0.13 \cdot anger \\
& + 0.1 \cdot sad + 0.13 \cdot cogproc + 0.11 \cdot cause + 0.13 \cdot discrep + 0.12 \cdot tentat \\
& + 0.13 \cdot certain + 0.1 \cdot feel + 0.11 \cdot swear - 0.15 \cdot you - 0.11 \cdot article \\
& - 0.08 \cdot friend - 0.09 \cdot space
\end{aligned}
\tag{2.5}
$$

Table 2.1 presents the variables used in the above equations, corresponding to each LIWC dimension. Yarkoni [70] noted the overwhelming number of negative correlates in Agreeableness, attributing this pattern to fundamental difference in language style rather than content (as cited in [16]). While this method was not developed to directly to infer personality traits of GitHub developers, we are confident in its ability as a general personality recognition tool. Some of its limitations will be discussed in-depth in Chapter 5.2.

Another method called the *Personality Recognizer* by Mairesse et al. [37] was built on top of LIWC dimensions by incorporating additional features from the MRC Psycholinguistic database. They built several models including classification, regression, and ranking models. In addition to all models performing better than the baseline, the ranking model reached the highest accuracy. *IBM Watson Personality Insights* is another service that can infer individual's personality traits, needs, and values from textual information. It uses an open vocabulary approach [58], a combination of GloVe Word Embedding features with Gaussian process regression [4], and unspecified machine learning algorithms.

In our work, we will be using the *LIWC2015* for several reasons. While one can argue that the sophisticated machine learning approaches outperform rule-based models like LIWC, they are resource intensive. It is also worth noting that these models are often built on top of LIWC. By using the simplest, yet effective approach, our intention is to provide a groundwork showing empirical evidence that studying personality traits is meaningful and can provide invaluable information when studying collaborative networks.

Table 2.1: Variables used in equations 2.1–2.5 and their corresponding LIWC dimensions

Variables	**Dimensions**
pronoun	Total pronouns
i	First-person singular
we	First-person plural
you	Second person
article	Articles
prep	Prepositions
negate	Negations
number	Numbers
affect	Affect Words
posemo	Positive emotion
negemo	Negative emotion
anx	Anxiety
anger	Anger
sad	Sadness
social	Social Words
family	Family
friend	Friends
insight	Insight
cause	Cause
discrep	Discrepancies
tentat	Tentativeness
certain	Certainty

Variables	**Dimensions**
see	Seeing
hear	Hearing
feel	Feeling
body	Body
sexual	Sexuality
ingest	Ingesting
achieve	Achievement
focuspast	Past focus
focuspresent	Present focus
motion	Motion
space	Space
time	Time
work	Work
leisure	Leisure
home	Home
money	Money
relig	Religion
death	Death
swear	Swear words
assent	Assent

2.4 Study of Personality in Software Engineering

The study of personality has focused extensively on many aspects of the workplace (e.g., leadership, job performance, job satisfaction, etc.) by industrial-organizational psychologists. The field of software engineering has started to incorporate these measures to examine different characteristics of developers, teams, and projects, both online and offline. Many researchers have utilized the Myers–Briggs Type Indicator (MBTI) [43] that characterizes one's personality into Jungian personality types. Given the many criticisms MBTI has received regarding its reliability and validity [8, 50], we will only focus on studies that utilize the Big Five.

2.4.1 Questionnaires and Inventories

Wang [68] examined a link between the project manager's personality traits and the software project success. They collected personality traits of software development teams by administering the NEO-FFI, a variant of the NEO PI-R. They found that all Big Five personality traits are correlated with project manager's leadership. While it is unclear what constitutes as success in this study, they found that Extraversion is positively correlated with the success of the software development project.

Acuña, Gómez, and Juristo [1] analyzed personality and its relationships to job metrics and qualities in software development teams. Similar to Wang, they used NEO-FFI to determine team member personality. They found that teams who score high on Agreeableness and Conscientiousness show high job satisfaction. Furthermore, they found a positive association between Extraversion and software product quality. Bell et al. [6] employed a similar method but focused on undergraduate computing students and on individual personality. They did not find any significant relationship between personality traits and individual performance within a team environment.

Feldt et al. [21] sought to establish links between personalities of software engineers and their views and attitudes on their professional activities. Engineers' personality traits were evaluated using the 50-item IPIP measure and their views and attitudes were obtained with a simple questionnaire. They found two clusters of personalities: a moderate personality profile and a more intense personality profile, particularly scoring high on Extraversion and Openness. In particular, Extraversion was positively associated with efficient performance when working under a set schedule, while Openness was positively associated with preference towards taking responsibility for the whole project over small parts. Kosti, Feldt, and Angelis [33] replicated Feldt et al.'s study on student population and reconfirmed the existence of two personality clusters. They also showed that extraverted students prefer to work in a team.

Kanij, Merkel, and Grundy [31] specifically focused on the personality traits of software testers, arguing that their tasks are fundamentally different from those involved in designing and programming. They collected the personality data of software practitioners using the 50-item IPIP measure. They showed that software testers are significantly more extraverted than software developers. On the other hand, Smith, Bird, and Zimmermann [59] found no statistical differences between the personality traits of developers and testers. They did, however, find that managers tend to be more conscientious and extraverted than engineers. In addition, they found that extraverted engineers showed preference to Agile software development, while neurotic engineers did not.

Mellblom et al. [42] examined a connection between personality traits and burnout (i.e., reduced professional efficacy and satisfaction) in software developers. Distributing the Mini-IPIP to open source developer mailing lists, they found a strong relationship between Neuroticism and burnout.

It is evident that use of personality questionnaires and inventories lead to wealth of information about developers and how they function in the workplace. All the studies discussed so far involve utilizing the survey method on software engineers and developers working in an office setting. Our study bares some resemblance to Wang's [68] in that we focus on the leadership by examining the maintainers of a given repository. Ours also share some similarities with Smith et al.'s [59] by comparing managers and engineers or rather comparing maintainers and non-maintainers.

2.4.2 Automatic Personality Recognition

With the development of automatic personality recognition tools, researchers quickly adopted this method over the traditional survey method. Rigby and Hassan [52] performed a preliminary analysis on the personality traits of developers from the Apache HTTP Server[1] developer mailing list. By utilizing LIWC on emails, they extracted the Big Five personality traits and found that two developers responsible for major Apache releases share similar personality traits to each other and their personalities differ significantly from the baseline–namely on traits Extraversion and Openness.

Bazelli, Hindle, and Stroulia [5] explored the personality traits of Stack Overflow[2] users. They extracted questions and answers written by users and processed them using LIWC.

[1] https://httpd.apache.org/

[2] https://stackoverflow.com/

After categorizing users into different levels of reputation, they found that top reputed authors tend to be more extraverted than the rest of the users.

Rastogi and Nagappan [51] analyzed the personality traits of GitHub contributors by applying LIWC on comments. Once separating them by their contribution levels, they found that contributors with high-level of contributions are more neurotic than contributors with medium-level of contributions.

Calefato and Lanubile [12] focused on trust (i.e., a facet of Agreeableness and its connection to success of distributed software teams. Using the *IBM Watson Personality Insights* on Apache developers, they found that the propensity to trust is positively correlated with pull request acceptance (a measure of success). This work was further extended by Calefato, Lunabile, and Vasilescu [13], extracting personality traits from code commits and email messages from the Apache Software Foundation. They found three personality types characterized by levels of Agreeableness and Neuroticism. They also found that highly open developers are more likely to become contributors to Apache projects.

These automatic personality recognition methods show promising results and take full advantage of publicly available data. We position ourselves with the aforementioned studies in that we also use an automatic recognition tool in lieu of the survey method. Recall that survey methods require one to administer personality inventories either in person or online and wait for responses. Instead, we extract comments made by individuals in an OCE, process them via LIWC, and automatically extract their personality traits. In this regard, our study is similar to Rastogi and Nagappan's [51] as we focus on understanding the characteristics of GitHub developers and their personality traits.

2.5 Clustering

From exploring the existing literature, one of the noticeable themes that rose was the use of clustering algorithms to explore if personality traits of individuals grouped in meaningful ways. Gerlach et al. [23] reported four distinct personality types by using the gaussian mixture models (GMMs). As an unsupervised clustering algorithm, GMM takes a probabilistic or soft approach—that is, it assigns probability values that a given data point belongs to number of clusters. Higher probability value entails the data point is more likely to belong in the correct cluster. Conversely, Calefato et al. [13] performed dimensionality reduction using principal component analysis (PCA) and then k-means clustering. Unlike soft clustering algorithms, k-means draws hard partitions so that a given data point belongs to a single cluster.

Inspired by these methods, we decided to use the mixtures of probabilistic principal component analyzers (MPPCA). To explain the intricacy of the MPCCA, we start with the simplest definition of the PCA—that is to say, we reduce the p-dimensional vector data projecting into a lower q-dimensional subspace. Tipping and Bishop [65] introduced the probabilistic PCA derived from a Gaussian latent variable model similar to factor analysis. Given the probabilistic formulation, it is able to handle missing values in the data and allows the use of expectation-maximization (EM) algorithm to estimate parameters of the model. Most importantly as a latent variable model, it is capable of representing low-dimensional manifolds embedded in the high-dimensional data, providing a parsimonious explanation of the observation dependencies. Tipping and Bishop [64] further extended their PPCA as a combination of local probabilistic models; hence the mixtures of PPCA or the MPPCA. As a mixture model, it is not limited by linear projections; a given data point in the latent space is represented by the Gaussian posterior distribution, rather than a single vector. In addition, MPPCA performs dimensionality reduction and clustering simultaneously rather than one after another. The parameters are again estimated using the em algorithm, leveraging quick computation.

While choosing a specific clustering algorithm is somewhat subjective, we decided to use the MPPCA for several reasons. We wanted to use a soft clustering approach and this eliminated the use of PCA and/or k-means. PCA, for instance, only considers linear projection of the data, while mixture models are not limited by this assumption. GMM was an obvious candidate; however, there are several advantages of using the MPPCA. First, MPPCA uses less parameters with higher dimensional data. More importantly, it allows the dimensionality of each covariance to be reduced and allows the removal of outliers, while maintaining the prominent information in the data.

Chapter 3

Methodology

In this chapter we provide details on how the data was selected and prepared, and outline the procedure for data analysis.

3.1 Data Selection and Preparation

We referred to the curated repository list created by Coelho et al. [17]. First, we selected 25 unmaintained repositories with at least 100 contributors. A repository was deemed unmaintained if it met at least one of the following criteria:

1. the repository was archived
2. the repository `README` file included any of the following phrases or terms:
   - “no longer maintained”
   - “not maintained”
   - unmaintained
   - deprecated
   - obsolete
3. the last commit was created more than 6 months from the date of data collection (April, 2020)

Table 3.1: Repository-level measures with descriptive statistics

Variables	**Description**	**Mean**	**SD**	**Min**	**Median**	**Max**
age	Age of the repository in days.	2380	1031	203	2416	4357
size	Size of the repository in kilobytes.	103845	155510	1190	52250	961013
collaborators	Number of collaborators.	95	296	1	14	1496
forks	Number of forks.	3197	4595	95	1256	18584
issues_open	Number of open issues.	272	589	0	140	4101
issues_closed	Number of closed issues.	880	3223	4	448	23125
popularity	$\texttt{(stars)} + \texttt{(watchers)} + \texttt{(pull requests)}^2$	36094848	252617628	1184	313064	1786644638
pr_open	Number of open pull requests.	91	258	0	17	1750
pr_closed	Number of closed pull requests.	1272	5670	3	507	40518
stars	Number of stars.	11832	16846	540	4910	88437
watchers	Number of watchers.	559	700	51	275	3185

Then we selected 25 maintained repositories that were similar in terms of the number of contributors. We identified the maintainers of each repository by examining additional data (i.e., GitHub profiles, LinkedIn[1] profiles, and personal websites). As a result, we identified a total of 135 self-recognized maintainers in maintained repositories and 103 self-recognized maintainers in unmaintained repositories. Table 3.1 reports descriptive statistics for repository-level measures obtained in both maintained and unmaintained repositories.

Using the GitHub REST API[2], we extracted all issue comments and pull request review comments made by all developers (i.e., both maintainers and non-maintainers) in each repository. We considered these comments as the main source of textual data for analysis. The discourse that occurs within these comments can often be dry; however, we believe that it is reasonable to assume that these comments reflect the personality of the authors. We also extracted commit metadata, which includes author, merger (i.e., the maintainer responsible for merging the pull request), date, and number of lines added and removed for each repository, using `PyDriller` [61]. Table 3.2 reports descriptive statistics of issues-level

[1] https://linkedin.com/

[2] https://developer.github.com/v3/

Table 3.2: Issues-level measures with descriptive statistics

Variables	**Description**	**Mean**	**SD**	**Min**	**Median**	**Max**
experience	Time between the first and the last commit merged by a maintainer.	1602.1	1137.2	0.0	1411.0	7435.0
mdn_addition	Median lines of code added by a maintainer per commit.	12.3	13.9	1.0	8.0	112.5
mdn_deletion	Median lines of code deleted by a maintainer per commit.	4.2	4.0	1.0	3.0	36.5
num_commits	Number of commits contributed by a maintainer.	1120.1	1966.3	2.0	431.0	14554.0
num_issues	Number of issues created by a maintainer.	301.5	457.8	0.0	120.0	2780.0
num_pr	Number of pull requests created by a maintainer.	187.2	310.4	0.0	75.0	2044.0
num_issue_comments	Number of issue comments made by a maintainer.	1316.8	2377.2	0.0	508.0	18504.0
num_pr_comments	Number of pull request comments made written by a maintainer.	772.5	1537.0	0.0	209.0	10938.0

measures for maintainers in maintained and unmaintained repositories.

We decided to remove bots from each repository as it could introduce noise to our data. This was done by searching all "developers" with their user ids beginning with, including, or ending with: `BOT`, `Bot`, or `bot`. Then we cross-referenced in the corresponding repositories to ensure that these were indeed bots and removed them from our dataset. It is important to note that textual communications in GitHub occurs using markdown format to make comments easily readable. Moreover, these communications often include code blocks in order to convey specific ideas and/or refer to bugs in the project. While these features make communicating to fellow developers more clear, it could introduce unwanted biases as our goal is to infer personality traits from natural language use. As such, we first converted all the extracted comments in the markdown format to html format for easy removal of code blocks, and finally converted to plain text.

Taking plain texts, we used *LIWC2015* [48] to obtain a 'language profile' for each developer in all repositories as seen in Table 3.3. We further removed developers who contributed less than 500 words. This was chosen as a cutoff as it was less than the minimum recommended for the commercially available *IBM Watson Personality Insights* (600 words)[3]. Before the filtering, there were 97,240 developers in 25 maintained repositories and 41,655 developers in 25 unmaintained repositories.

Table 3.3: LIWC output with descriptive statistics

Dimension	**Labels**	**Mean**	**SD**	**Min**	**Median**	**Max**
Word count	WC	3609.17	26104.57	500.00	921.00	2065266.00
Summary Variables						
Analytic thinking	Analytic	80.76	12.26	13.50	83.90	99.00
Clout	Clout	42.94	13.96	1.00	42.17	99.00
Authentic	Authentic	27.98	17.88	1.00	25.70	98.15
Emotional tone	Tone	44.35	20.34	1.00	42.60	99.00
Language Metrics						
Words per sentence	WPS	50.36	167.28	5.91	26.16	6472.00
Words > 6 letters	Sixltr	22.56	4.90	0.17	22.13	81.57
Dictionary words	Dic	62.88	15.03	0.22	67.45	94.17
Function Words	function	37.06	12.45	0.00	40.35	60.51
Total pronouns	pronoun	7.94	3.62	0.00	8.24	23.78
Personal pronouns	ppron	3.73	1.97	0.00	3.66	18.54
First-person singular	i	2.15	1.61	0.00	1.93	13.68
First-person plural	we	0.41	0.55	0.00	0.19	9.26
Second-person	you	0.97	0.95	0.00	0.73	17.00
Third-person singular	shehe	0.02	0.09	0.00	0.00	5.17
Third-person plural	they	0.19	0.25	0.00	0.12	2.75
Impersonal pronouns	ipron	4.20	2.13	0.00	4.34	23.78
Articles	article	5.95	2.68	0.00	6.43	15.88
Prepositions	prep	10.19	3.16	0.00	10.93	20.33
Auxiliary verbs	auxverb	6.35	2.69	0.00	6.69	17.86
Common adverbs	adverb	3.35	1.61	0.00	3.45	13.16
Conjunctions	conj	4.63	1.86	0.00	4.94	14.01
Negations	negate	1.44	0.75	0.00	1.40	14.62
Other Grammar						
Regular verbs	verb	11.93	3.97	0.00	12.56	39.34
Adjectives	adj	3.18	1.37	0.00	3.23	49.67

[3] https://cloud.ibm.com/docs/personality-insights?topic=personality-insights-input

Comparatives	compare	1.78	0.96	0.00	1.78	12.50
Interrogatives	interrog	1.07	0.65	0.00	1.05	8.60
Numbers	number	5.22	6.21	0.00	2.98	99.35
Quantifiers	quant	1.83	0.95	0.00	1.79	10.22
Affect Words	affect	3.16	1.31	0.00	3.09	16.58
Positive emotion	posemo	2.06	1.09	0.00	1.96	15.34
Negative emotion	negemo	1.09	0.77	0.00	0.98	12.62
Anxiety	anx	0.08	0.15	0.00	0.00	3.34
Anger	anger	0.11	0.24	0.00	0.00	9.86
Sadness	sad	0.32	0.34	0.00	0.24	6.04
Social Words	social	3.92	2.01	0.00	3.73	20.89
Family	family	0.02	0.11	0.00	0.00	4.46
Friends	friend	0.05	0.18	0.00	0.00	6.82
Female referents	female	0.02	0.24	0.00	0.00	9.89
Male referents	male	0.04	0.14	0.00	0.00	7.63
Cognitive Processes	cogproc	12.18	3.87	0.00	12.88	29.76
Insight	insight	1.98	1.06	0.00	1.89	14.21
Cause	cause	2.86	1.26	0.00	2.82	16.58
Discrepancies	discrep	1.97	1.04	0.00	1.99	9.38
Tentativeness	tentat	2.70	1.31	0.00	2.74	13.30
Certainty	certain	1.19	0.70	0.00	1.15	8.62
Differentiation	differ	3.57	1.44	0.00	3.69	14.99
Perceptual Processes	percept	1.10	1.01	0.00	0.94	64.74
Seeing	see	0.76	0.88	0.00	0.62	64.74
Hearing	hear	0.17	0.33	0.00	0.09	14.56
Feeling	feel	0.12	0.20	0.00	0.01	6.05
Biological Processes	bio	0.34	0.68	0.00	0.21	66.30
Body	body	0.09	0.35	0.00	0.00	34.61
Health/illness	health	0.17	0.31	0.00	0.07	8.32
Sexuality	sexual	0.03	0.18	0.00	0.00	7.74
Ingesting	ingest	0.08	0.35	0.00	0.00	31.69
Drives and Needs	drives	5.39	1.86	0.00	5.36	23.78
Affiliation	affiliation	0.96	0.83	0.00	0.76	11.02
Achievement	achieve	1.48	0.79	0.00	1.42	13.93
Power	power	2.03	1.25	0.00	1.75	16.40
Reward focus	reward	0.91	0.61	0.00	0.85	13.43
Risk focus	risk	0.67	0.52	0.00	0.58	8.74
Time Orientations						

Past focus	focuspast	2.00	1.04	0.00	1.93	11.89
Present focus	focuspresent	8.67	3.05	0.00	9.15	21.74
Future focus	focusfuture	0.86	0.56	0.00	0.82	5.68
Relativity	relativ	11.01	2.92	0.00	11.14	30.84
Motion	motion	2.46	1.63	0.00	2.15	21.43
Space	space	5.18	1.68	0.00	5.20	19.60
Time	time	3.44	1.53	0.00	3.32	24.50
Personal Concerns						
Work	work	2.80	1.50	0.00	2.55	24.00
Leisure	leisure	0.42	0.63	0.00	0.27	33.15
Home	home	0.20	0.48	0.00	0.00	7.91
Money	money	0.18	0.44	0.00	0.03	9.98
Religion	relig	0.01	0.06	0.00	0.00	3.08
Death	death	0.06	0.25	0.00	0.00	10.48
Informal Speech	informal	1.30	1.33	0.00	1.03	39.11
Swear words	swear	0.01	0.08	0.00	0.00	3.99
Netspeak	netspeak	1.04	1.28	0.00	0.75	39.11
Assent	assent	0.17	0.35	0.00	0.09	14.26
Nonfluencies	nonflu	0.08	0.16	0.00	0.00	4.95
Fillers	filler	0.02	0.07	0.00	0.00	6.21
All Punctuation	AllPunc	36.63	22.26	0.55	28.91	583.48
Periods	Period	7.05	4.78	0.00	6.38	200.29
Commas	Comma	3.38	2.32	0.00	3.21	85.05
Colons	Colon	3.14	4.84	0.00	1.66	200.00
Semicolons	SemiC	0.15	0.70	0.00	0.00	32.24
Question marks	QMark	0.47	0.61	0.00	0.35	27.12
Exclamation marks	Exclam	0.18	0.44	0.00	0.07	24.91
Dashes	Dash	4.96	8.23	0.00	2.60	298.43
Quotation marks	Quote	1.39	3.56	0.00	0.63	104.62
Apostrophes	Apostro	1.71	1.78	0.00	1.38	63.71
Parentheses (pairs)	Parenth	2.67	2.91	0.00	1.92	54.11
Other punctuation	OtherP	11.50	12.82	0.00	7.41	564.57

The final dataset consisted of 25 maintained repositories with 135 maintainers and 14,322 non-maintainers, and 23 unmaintained repositories with 76 maintainers and 3,722 non-maintainers. Finally with the LIWC output we used equations 2.1–2.5 to infer personality traits of each developer. Table 3.4 reports descriptive statistics of personality traits of all developers in both maintained and unmaintained repositories. It is worth noting that these personality traits are not on an absolute scale but rather arbitrary. When obtaining Openness with the equation 2.1, 21 of the 24 variables have negative coefficients. On the other hand, 12 of the 14 variables have negative coefficients when obtaining Conscientiousness with the equation 2.2. As a result, Openness appears to be much more negative than all the other traits, including Conscientiousness, but they are not directly comparable as they occupy different scales.

Table 3.4: Personality traits with descriptive statistics

Traits	**Mean**	**SD**	**Min**	**Median**	**Max**
openness	-5.73	1.81	-14.85	-5.81	1.38
conscientiousness	-2.33	0.81	-7.24	-2.44	3.21
extraversion	-0.35	0.96	-11.90	-0.20	6.44
agreeableness	3.86	0.78	-1.74	3.85	14.03
neuroticism	2.03	0.83	-1.56	2.07	6.72

3.2 Overview of the Data Analysis

The initial plotting of the developers' personality traits revealed the distributions to be non-Gaussian and this was further confirmed with the Shapiro–Wilk test ($p < 0.001$). We first employed the MPPCA algorithm to see if personality traits of developers cluster in meaningful ways. It is important to note that the number of clusters and the dimension of the subspaces must be given *a priori* when using the MPPCA. Because these parameters are unknown, we ran the algorithm several times with different combinations of clusters and subspaces comparing the log-likelihood values, where a higher value indicates optimality. Figure 3.1 shows that a combination of 9 clusters and 4 low-dimensional subspaces reached the highest log-likelihood value among others. These parameters were chosen and subsequently fixed for the rest of the analysis.

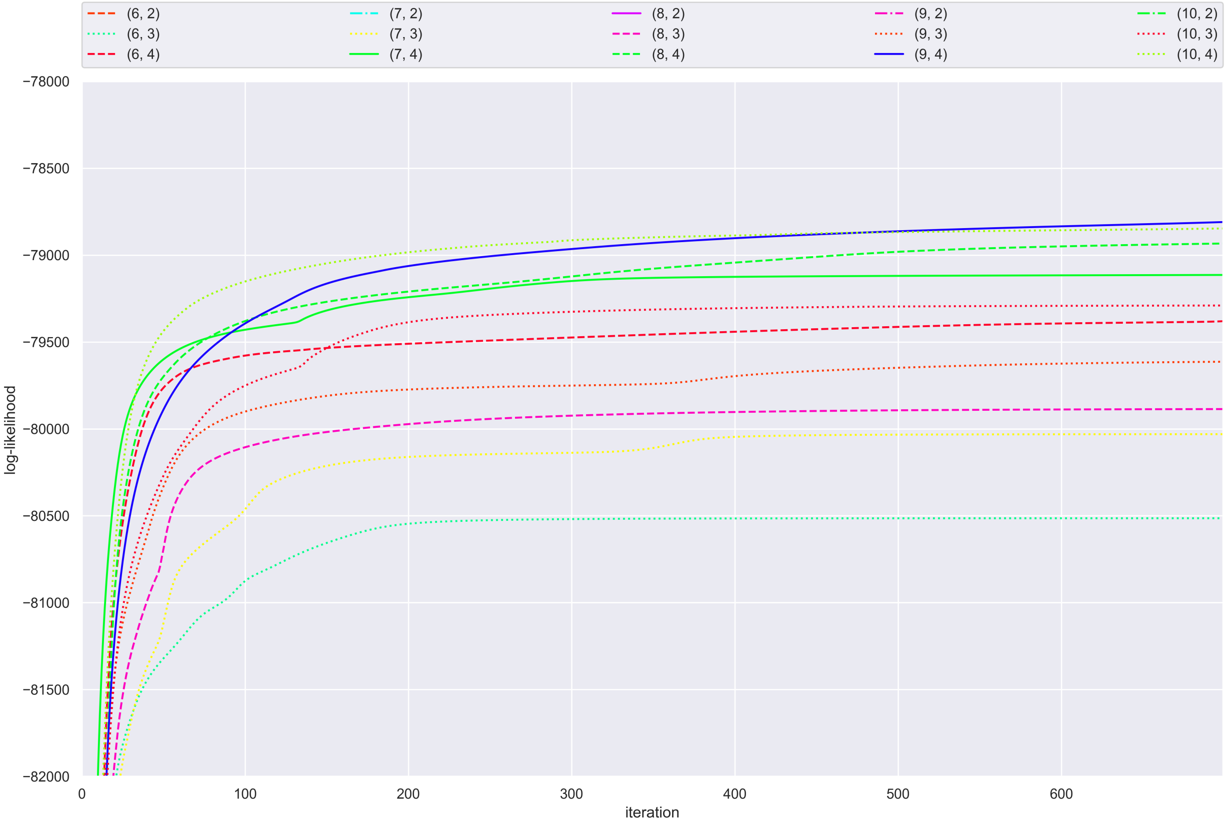

Figure 3.1: Log-likelihood of different combinations of clusters and subspaces of the MP-PCA algorithm

Once we observed the clusters, we used a rank-based nonparametric Kruskal–Wallis (KW) test to verify whether the population medians of the clusters are statistically different. In addition, we used a supplementary nonparametric Kolmogorov–Smirnov (KS) test to illustrate whether two groups were sampled from different cumulative distributions.

RQ1 being the central research question, we further employed the Mann–Whitney U (MWU) test to examine the differences in personality traits of maintainers in maintained repositories and maintainers in unmaintained repositories. The MWU test is similar to the KS test in that it makes two unpaired comparisons. However, it is different as it ranks all the values and it is also less sensitive to the shape and spread of the distributions compared to the KS test. By utilizing similar yet different methods, we can minimize threats to statistical conclusion validity. If the results we obtain from the MWU test is similar or the same as the MPPCA, we can ensure that the observed results are reasonable and it adds more credibility to our findings.

Figure 3.2 illustrates the distributions of personality traits of the maintainers in maintained and unmaintained repositories. The lower whisker and the upper whisker show minimum value and maximum value, respectively. The box itself represents the interquartile range (i.e., the difference between 75th and 25th percentiles). Finally, the notch displays the confidence interval around the median.

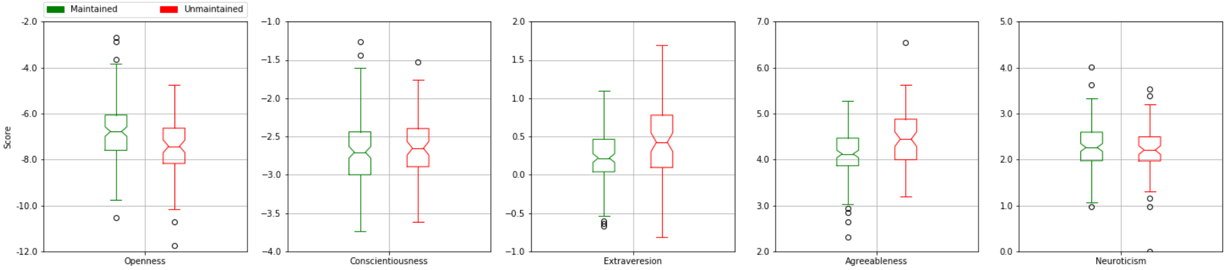

Figure 3.2: Notched boxplot representing personality traits of maintainers in maintained repositories and maintainers in unmaintained repositories

To predict repository success, we used a generalized linear mixed model (GLMM) provided by the `glmer` function of the `lme4` package in *R*. This model was selected as our aim was to capture measurements within the same groups (i.e., same repositories) as random effects. The following variables were considered for the model:

- **Independent Variables:** The Big Five personality traits were considered as independent variables.

- **Dependent Variable:** The response variable was whether a maintainer contributed to a maintained or unmaintained repository.
- **Control Variables:**
  - `experience`
  - `mdn_addition`
  - `mdn_deletion`
  - `num_comments`[4]
  - `num_commits`
  - `num_issues`[5]
  - `age`
  - `size`

The control variables were various software engineering metrics that could have influence on the dependent variable. For example, `experience` and `num_commits` were chosen as they are indirect measures of one's expertise. Technical factors, such as `mdn_addition`, `mdn_deletion`, `num_comments`, `num_issues`, `age`, and `size`, are readily collected when examining pull-based research [25] and they have shown a link with pull request acceptance [66]. Figure 3.3 illustrates the control variables of maintained and unmaintained repositories.

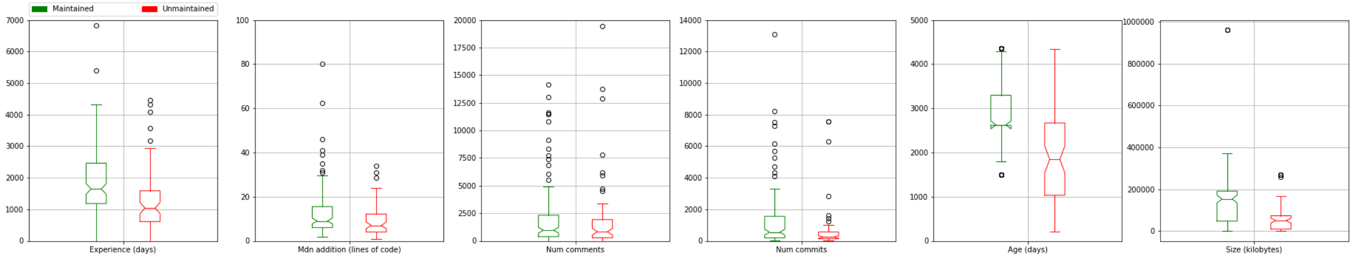

Figure 3.3: Notched boxplot representing control variables of maintained repositories and unmaintained repositories

To reduce any bias when fitting the mixed model, we calculated Spearman's correlation between variables and removed ones with a value higher than 0.7. In particular, we

[4] `num_comments` = `num_issue_comments` + `num_pr_comments`

[5] `num_issues` = `num_issues` + `num_pr`

removed the variables `num_issues` and `mdn_deletion` because they were highly correlated with the variables `num_comments` and `mdn_addition`, respectively. We also calculated the variance inflation factor (VIF) to detect any multicollinearity in the data. VIF provides an index for each independent variable that measures how much the variance of an estimated regression coefficient is increased due to the collinearity. None of our variables suffered from multicollinearity as their VIF values were smaller than 2. Finally, we standardized the values of our variables using *scale* function in *R* before fitting the model. The coefficients were considered pertinent if they were statistically significant ($p < 0.05$).

Chapter 4

Results

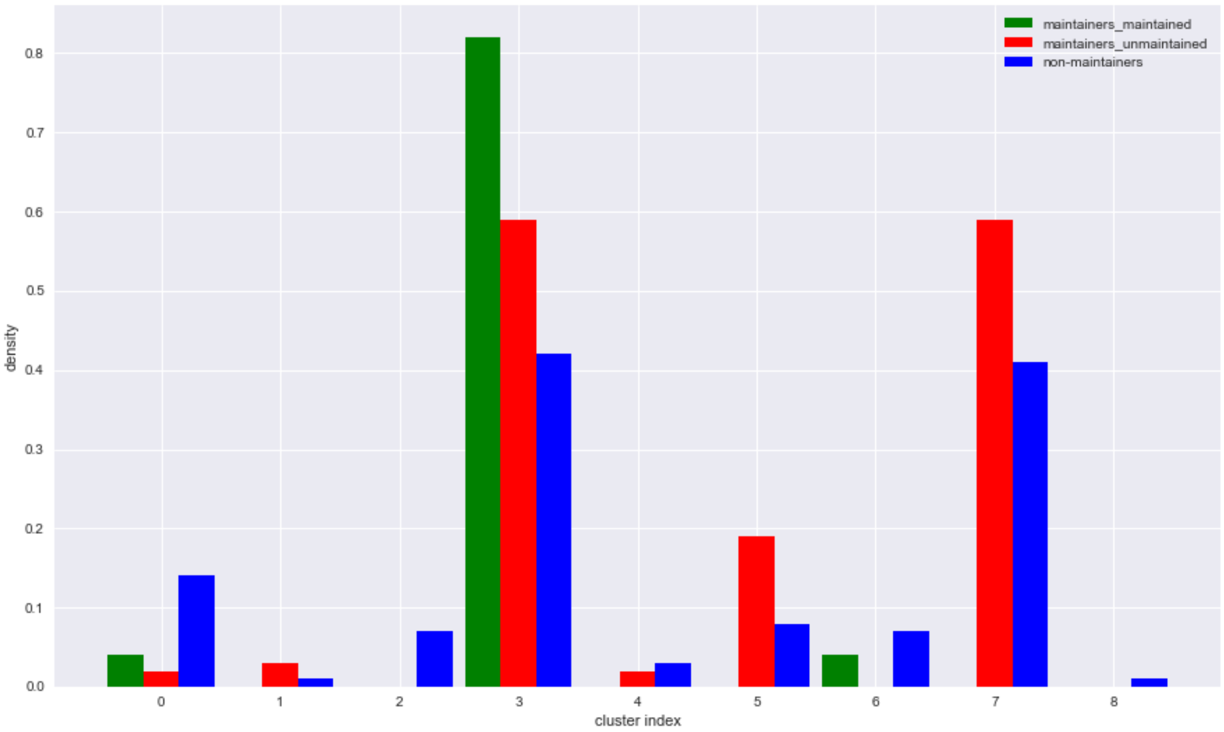

Figure 4.1: Clusters observed with the MPPCA algorithm

Figure 4.1 illustrates the clusters observed with the MPPCA algorithm. To examine the distributions of maintainers and non-maintainers, we picked a cluster or clusters with the highest density. Approximately 80% of maintainers in maintained repositories grouped in a single cluster, while approximately 60% of maintainers in unmaintained repositories were equally distributed across two clusters. Similarly, approximately 40% of non-maintainers were equally distributed across two clusters. We then applied the KW and KS tests on the distributions of these groups.

Tables 4.1 and 4.2 report the results of the MPPCA algorithm. For RQ0, we found that both maintainers and non-maintainers in all repositories are statistically different in most traits but not in Neuroticism. For RQ1, we found that maintainers in maintained repositories and maintainers in unmaintained repositories are statistically different in Openness and Agreeableness.

Table 4.1: MPPCA algorithm results on the personality traits of maintainers and non-maintainers in all repositories

Traits	**Kruskal-Wallis**		**Kolmogorov-Smirnov**	
	H statistic	Sig.	D statistic	Sig.
openness	51.76	***	0.34	***
conscientiousness	7.31	**	0.16	**
extraversion	38.28	***	0.30	***
agreeableness	77.61	***	0.42	***
neuroticism	2.64		0.13	

*$p < .05$, **$p < .01$, ***$p < .001$

Table 4.2: MPPCA algorithm results on the personality traits of maintainers in maintained repositories and maintainers in unmaintained repositories

Traits	**Kruskal-Wallis**		**Kolmogorov-Smirnov**	
	H statistic	Sig.	D statistic	Sig.
openness	12.52	***	0.36	***
conscientiousness	1.99		0.19	
extraversion	1.11	***	0.22	***
agreeableness	4.33	***	0.32	***
neuroticism	0.38		0.22	

*$p < .05$, **$p < .01$, ***$p < .001$

We further explored RQ1 with MWU tests. When considering the personality traits of maintainers alone, we found Openness, Extraversion, and Agreeableness to be significant as seen in Table 4.3. In particular, maintainers in maintained repositories are more open, but less extraverted and less agreeable than maintainers in unmaintained repositories. Table 4.4 presents sample comments from maintainers with the aforementioned personality traits. Following up we ran an additional MWU test considering the absolute differences between the personality traits of maintainers and the median personality traits of non-maintainers. We found Agreeableness to be significantly different as seen in Table 4.5. In particular, the absolute difference in Agreeableness between maintainers and non-maintainers in maintained repositories is smaller than in unmaintained repositories.

Table 4.3: Mann–Whitney U test results on the personality traits between maintainers in maintained repositories and maintainers in unmaintained repositories

Traits	**Maintainers' Personality Traits**		
	Median		**Sig.**
	Maintained	**Unmaintained**	
openness	-6.7858	-7.4278	**
conscientiousness	-2.7063	-2.6542	
extraversion	0.2185	0.4254	**
agreeableness	4.1199	4.4421	***
neuroticism	2.3938	2.3316	

*$p < .05$, **$p < .01$, ***$p < .001$

Table 4.4: Sample comments from maintainers who score high on Openness, low on Extraversion, or low on Agreeableness

Traits	**Sample Comments**
openness	Also, what are the error cases for this API? What happens if the CT server is temporarily unavailable? Should the certificate issuance fail, or should the certificate be put in a queue to resubmit to the server?
extraversion	My personal idea is that we don't need to keep that. It don't bring enough benefit. I don't think it would make that implementation much more complicated. There is many errors due to that.
agreeableness	Clearly, ListT [] fails to preserve the associativity monad law. This example violates the requirement given in the documentation that the inner monad has to be commutative. However, all the preceding examples use IO which is neither commutative.

Table 4.5: Mann–Whitney U test results on the absolute differences between maintainers' personality traits and median personality traits of non-maintainers in a repository

Traits	**Absolute Differences in Personality Traits**		
	Median		**Sig.**
	Maintained	**Unmaintained**	
openness	1.0034	1.2342	
conscientiousness	0.3641	0.4032	
extraversion	0.4010	0.4718	
agreeableness	0.3743	0.5580	**
neuroticism	0.4112	0.2889	

$*p < .05$, $**p < .01$, $***p < .001$

Table 4.6 reports the results of the GLMM. In addition to Agreeableness being significant, we found `mdn_addition`, `age`, and `size` to be significant predictors of repository success.

Given the statistical differences we observed in RQ1 with respect to absolute differences, we decided to run the GLMM with the absolute differences between the personality traits of maintainers and the median personality traits of non-maintainers as predictors of repository success. Table 4.7 reports the results of the new GLMM. Similar to the previous model, `mdn_addition`, `age`, and `size` were all significant predictors of repository success. Moreover, the absolute difference in Agreeableness between maintainers and non-maintainers were found to be statistically significant predictors.

Our results suggest that higher `mdn_addition`, `age`, and `size` are associated with higher likelihood of a maintainer contributing to a maintained or successful repository. Moreover, lower levels of Agreeableness is associated with repository success. This was further confirmed when examining the absolute personality differences between the maintainers and non-maintainers. The greater the difference in Agreeableness, the greater the chance of repository success.

Table 4.6: Logistic regression model of the repository success as explained by the personality traits of maintainers

Variables	**Maintainers' Personality Traits**		
	Coef. Estimate	**Std. Error**	**Sig.**
(Intercept)	1.8493	2.8562	
openness	-0.9015	0.5929	
conscientiousness	0.1231	1.4854	
extraversion	-0.2780	1.1220	
agreeableness	-2.7538	0.8792	**
neuroticism	-0.5758	1.5447	
experience	-0.4888	1.6021	
mdn_addition	6.0655	2.2741	**
num_comments	0.3505	1.4397	
num_commits	-0.6721	1.5869	
age	8.1432	1.4588	***
size	12.2806	2.9197	***

$^{*}p < .05$, $^{**}p < .01$, $^{***}p < .001$

Table 4.7: Logistic regression model of the repository success as explained by absolute differences between the personality traits of maintainers and the median personality traits of non-maintainers in a repository

Variables	**Absolute Differences in Personality Traits**		
	Coef. Estimate	**Std. Error**	**Sig.**
(Intercept)	-4.6369	0.9224	***
openness	0.5666	0.4156	
conscientiousness	-1.0435	1.1769	
extraversion	-1.0461	0.9867	
agreeableness	-2.0256	0.7589	**
neuroticism	0.7326	1.1155	
experience	0.1551	1.2643	
mdn_addition	5.9383	2.2193	**
num_comments	0.1161	1.4765	
num_commits	-0.6445	1.7168	
age	8.5358	1.4361	***
size	11.9258	2.9062	***

*$p < .05$, **$p < .01$, ***$p < .001$

Chapter 5

Discussions

In this chapter we discuss our results from the perspectives of personality psychology and industrial-organizational psychology. For each research question we loosely structure the discussion in the following manner: (1) reintroduce the personality trait of interest; (2) state the hypothesis and underlying reasons; (3) rephrase the finding; and (4) provide an interpretation of the finding based on previous research or speculation if there exists no direct support.

RQ0: Do maintainers and non-maintainers show difference in personality traits?

A maintainer is a specialized role that is characterized by increased responsibility. When majority of our research questions revolve around the identity of a maintainer, we make an implicit assumption that maintainers are unique because their roles are fundamentally different from the roles of non-maintainers. By posing RQ0, we wished to test this assumption empirically. We hypothesized that all personality traits would significantly differ between maintainers and non-maintainers in all repositories. Our results indicated this to be true except for trait Neuroticism as seen in Table 4.1. A further investigation is required to understand why there exists no difference in Neuroticism between maintainers and non-maintainers.

RQ1: Do maintainers in maintained repositories show difference in personality traits from maintainers in unmaintained repositories?

Openness is a trait that is closely related to creativity, which has been shown to have a link with leadership effectiveness. By employing intellectual stimulation, transformational leaders foster novel ideas and problem-solving processes [60]. Feldt et al. [21] previously found that software engineers who are highly open prefer to take responsibility for the entire project over individual parts. Given this evidence, we hypothesized that maintainers in successful repositories would be more open than maintainers in unsuccessful repositories. Our results are shown in Tables 4.2 and 4.3 confirming the hypothesis: maintainers in maintained repositories were significantly more open than maintainers in unmaintained repositories. In addition to its relationship to leadership effectiveness, individuals who score high on Openness are welcoming of new ideas and tend to articulate their thoughts clearly [19]. Given the textual communications that occur in GitHub, we believe maintainers with high Openness lead fruitful discussions among contributors, leading the repository to be maintained actively.

Extraversion describes one's enthusiasm and assertiveness in social situations. It has shown an inconsistent relationship with leadership, although more often positive than negative [30]. In software engineering context, Wang's study [68] showed that manager's personality, specifically Extraversion, is positively correlated with their leadership performance and the success of the software development project. We hypothesized that maintainers in successful repositories would be more extraverted than maintainers in unsuccessful repositories. This was surprisingly not true as our results in Table 4.3 showed the opposite: maintainers in maintained repositories were significantly less extraverted than maintainers in unmaintained repositories. This inconsistency may be due to the differences in the study samples; in particular, Wang studied software projects in an office setting while we studied projects that take place online. We speculate that the dominance exhibited by highly extraverted individuals may be ambiguous without additional cues. In an office setting, this dominance along with many non-verbal cues may be interpreted as a sign of good leadership. Conversely, these additional cues are largely absent when communicating textually in an online environment, and thus dominance may be interpreted as aggressive and hostile. A recent study by Kern et al. [32] showed that top developers with high productivity tend to score noticeably lower on Extraversion than other vocations, providing further evidence that the role of Extraversion may be different across software engineering contexts.

Agreeableness characterizes one's propensity for harmonious interpersonal relationships. Its relationship to leadership is context dependent [28]. Calefato and Lanubile [12] showed a positive correlation between Agreeableness and pull request acceptance. In contrary, our previous work showed the likelihood of pull request acceptance is not affected by Agreeableness of requesters or closers [29]. With these conflicting findings, it was unclear how the trait would manifest in maintainers' behaviours. The previously mentioned Kern et al.'s [32] study showed that in addition to Extraversion, top developers scored significantly low on Agreeableness as well. This led us to hypothesize that maintainers in successful repositories would be less agreeable than maintainers in unsuccessful repositories. This was indeed true: maintainers in maintained repositories are significantly less agreeable than maintainers in unmaintained repositories as seen in Table 4.3. Individuals who score low on Agreeableness often appear harsh and less empathic as they are not afraid of confronting others. However, this is a necessary quality of a leader as they need to correct any behaviour or work that is detrimental to the success of the project. We also speculate that this quality of directness may signal fairness to other developers in the repository and perception of fairness has been shown to have an indirect effect on leadership effectiveness [49]. It is worth noting that the effects of Agreeableness on communication is moderated by virtualness [9]. This suggests that while the role of Agreeableness may be significant, its influence in OCEs may be dampened.

RQ2: What is the relationship between maintainers' personality traits and the success of a repository?

In RQ1, we showed that Openness, Extraversion, and Agreeableness are significantly different between maintainers in maintained repositories and maintainers in unmaintained repositories. When considering these traits as predictors of repository success, we found that only Agreeableness is significant as seen in Table 4.6. In particular, lower levels of Agreeableness in maintainers is associated with a higher likelihood of a repository being maintained.

It is interesting to note that this significance persists even when we consider the model with absolute personality differences—that is, personality traits of maintainers in relation to the median personality traits of non-maintainers. As seen in Table 4.7, the greater the absolute difference in Agreeableness, the greater the likelihood of a repository being maintained. This is largely consistent with our previous work where we showed that the absolute difference in Agreeableness between the requester and the closer affect pull request acceptance positively. Moreover, our result is supported by existing evidence that shows team personality diversity has a positive effect on team performance [44].

Taken together, the success of a repository is not only dependent on the maintainers but their relationships with everyone else on the team. In other words, the role of a maintainer is important in guiding developers to move forward and it is also crucial to have competent developers who follow these instructions and bring their talents.

RQ3: What is the relationship between maintainers' personality traits and the popularity of a repository?

We adopted a popularity measure from Aggarwal, Hindle, and Stroulia [2]:

$$Popularity = (stars) + (forks) + (pull\ requests)^2$$

Borges and Valente [7] showed that active promotion on social media has a positive effect on the number of stars of OSS projects. Given this evidence we hypothesized that Extraversion would be a significant predictor of repository success. Our rationale was that extraverted individuals tend to be gregarious and enthusiastic, and thus maintainers with such trait would be able to attract new contributors, thereby increasing the numbers of stars, forks, and pull requests in the repository. This, however, turned out to be false as we saw no significant result. The simplest reason for this may have to do with the formulation of the popularity measure. In social network theory, popularity is often measured by in-degree of the vertex of interest [69]. In other words, popularity of a repository should not only consider stars, forks, and pull requests, but perhaps it should be formulated as a graph problem capturing the dynamics of developers in the repository.

5.1 Implications for Practice

Despite our novel findings, one must be cautious to suggest that there is a single definitive 'personality profile' that can guarantee the success of an OSS project. Personality, by definition, explains one's stable disposition and their patterns of behaviour. Theoretically this entails that one cannot simply change their personality at their own will.

We observed significant personality differences between maintainers and non-maintainers, as well as maintainers in maintained repositories and maintainers in unmaintained repositories. In addition, we saw the importance of absolute differences in personality traits between maintainers and non-maintainers in our findings. It is worth noting that while we considered a repository to be successful if it is actively maintained, other metrics could be used to infer the success of a repository.

That said, there are several implications of our findings that may be useful in practice. First, the owner and the core team members of an OSS project should be cognizant when assigning new maintainers as their behaviours can have a great influence on the success of the project. In particular, highly open developers can make great maintainers by facilitating a creative environment. We are unable to make specific suggestions regarding the traits Extraversion and Neuroticism as our results seem to be the opposite from the trends seen in the existing literature. Thus, further investigations are warranted. Nevertheless, we also believe that maintainers should recruit and encourage diverse team of contributors as the diversity of team personality can lead to successful projects. Lastly, continuous communication, in the form of issues and pull requests, between maintainers and non-maintainers is not only required to improve the quality of work, but can increase the likelihood of project success. This communication process, of course, should be characterized by being civil and respectful, while expressing ideas cogently.

5.2 Threats to Validity

Using LIWC to infer developers' personality traits may naturally raise a question regarding its construct validity. It is possible that the personality traits we obtain from LIWC may not actually represent the true personality of the developers. We claim that the personality traits that have been extracted are reasonably true given the wealth of GitHub comments used. In addition to studies showing correlations between LIWC dimensions and the Big Five traits [48, 41], many studies have utilized this method and were successful [52, 5, 13]. Furthermore, there is evidence showing strong correlations between self-reported personality measures and observer rating personality measures [39], suggesting that one does not have to depend solely on self-reported measures as a valid method of obtaining personality traits.

Another concern revolves around the fact that the actual content of GitHub comments is both technical and software engineering specific. We believe this issue can be mitigated by removing the code blocks and focusing on natural language only. It is also important to note that LIWC places an emphasis on style or function words over content words [63]. Content words refer to words that have lexical meaning, such as nouns, verbs, adjectives, and adverbs. On the other hand, style words often have little or ambiguous meaning, but provide functional purposes as in the case of conjunctions. Given that there is evidence suggesting style words are more closely related to one's social and psychological words [16], we believe that our ability to extract true personality traits is not impeded by the technical content in GitHub comments.

It is worth noting that there is no strict definition or simple metric that shows whether a repository is maintained or not. Recall that we devised a set of criteria to select unmaintained repositories. In the case where a repository was archived or it explicitly stated that it was no longer maintained, its status as an unmaintained repository was objectively clear. However, there were many repositories that did not have explicit statements about its status, but were presented with diminished activities. To combat this ambiguity, we introduced an additional criterion that states that a repository is unmaintained if the last commit was created more than 6 months from the date of data collection. We believe that this is a reasonable criterion to pose to distinguish repositories that are maintained from unmaintained ones.

The relatively low sample size of maintainers in maintained and unmaintained repositories may also be a threat to validity. While there exists a possibility of not identifying all the maintainers during the manual process, we believe that this data is trustworthy as all the maintainers were indeed self-recognized as maintainers of the chosen projects. In addition, it may be hard to generalize our results since the number of repositories chosen and contributors might be not representative of all OSS projects. We believe that the current dataset is still large enough to perform empirical analyses and serves as a good starting point. A replication study with a wider range of repositories will confirm our findings and provide further statistical power.

5.3 Future Work

The first step in future directions would be to create a new dataset with much larger sample size and to replicate the current findings. A larger dataset would not only provide more generalizability and statistical power, but also more credibility to our findings. We also acknowledge that the current definition of *success* of a repository may be incomplete. Recall that we considered a repository to be successful if it is maintained and unsuccessful if it is unmaintained or archived. What constitutes as success does not have to depend solely on the status or activity level of a repository. Like the proverb "All good things must come to an end.", successful repositories may end up being archived because all relevant tasks have been accomplished and it ran its course. As such, it may be important to devise a more comprehensive definition and/or metric of success in OSS projects. In alignment with this sentiment, it may be worth develop definitions or metrics of impact and innovation in OSS projects. Impact, for instance, could be defined as simple as *code reusability* to gauge how influential a repository is to the open source community. On the other hand, quantifying innovation may be more challenging. If one were to define innovation as solving a problem in a meaningful way, how could this be quantified? It is evident that there exists many OSS projects that accomplishes interesting tasks, but it does not have to be innovative. Nevertheless, these are interesting avenues worth exploring.

In addition to the replication study and improvements on metrics, another future direction is to create a new automatic personality recognition tool. The increased computational power has allowed researchers to develop different methods to infer personality traits, using deep learning. Using the Pennabaker and King dataset [48], Majumder et al. [38] used a combination of convolutional neural network, multilayer perceptron, and support vector machine, training on word embeddings and features outlined in Mairesse et al. [37]. This model was then tested against the LIWC dataset. Liu, Perez, and Nowson [36] created a new model using a recurrent neural network on Twitter[1] corpus. They administered personality inventories on participants and tested the accuracy of their model. Twitter appears to be a promising source of data as another study by Carducci et al. [14] utilized tweets to develop a new supervised learning method that automatically computes personality traits. As such, there are many inspirations to create a new model; we would be interested in comparing the accuracy of our newly developed model against existing ones.

Along with creating a new personality tool, it would be beneficial to conduct a case study in an office setting. First by conducting a case study, it would allow us to administer personality inventories directly to the developers working in the office. This would provide a ground truth and would provide an opportunity to validate our new tool. In this paper, we made implicit assumptions that OCEs are similar to offline environments. Observing people in the workplace and how their personalities manifest in the OCEs, we would be able to make direct comparisons and make conclusions about whether OCEs are similar to its offline counterparts.

Last but not least, there are several research questions that could be pursued relating to the evolution of maintainers:

- How do developers become maintainers in OSS projects?
- How long does it take developers to become maintainers?
- Once developers become maintainers, what kind of contributions do they make?
- Why do maintainers quit?

These questions would involve analyzing both qualitative and time series data, such as the types of contributions, first/last issue and pull request created by developers and maintainers.

[1] https://twitter.com/

Chapter 6

Conclusion

We presented an empirical analysis of 50 GitHub repositories with more than 200 self-recognized maintainers to understand the role of a maintainer in OSS projects and the effects of their personality traits on project success. As a specialized role, maintainers hold important responsibilities, directing and delegating tasks to contributors, and leading the overall direction of the project.

Our results showed that there are significant differences between the personality traits of maintainers and non-maintainers—namely, Openness, Conscientiousness, Extraversion, and Agreeableness. Once establishing personality differences between maintainers and non-maintainers, we observed significant differences between maintainers in maintained repositories and maintainers in unmaintained repositories. In particular, maintainers in maintained repositories are more open, less extraverted, and less agreeable than maintainers in unmaintained repositories. When examining repository success, Agreeableness was found to be a significant predictor. Specifically, we noted the absolute personality differences between maintainers and non-maintainers result in positive effects on repository success. This suggests OSS projects can benefit from having a diverse group of developers in terms of their personalities.

By highlighting personality differences within online teams in the GitHub ecosystem, our work provides a compelling argument that studying social factors and psychological constructs can bring new insights on the mechanisms of online collaboration. Taking our core ideas we can conduct further experiments, utilizing various methods like qualitative interviews that can enrich our understanding of social and group dynamics in OCEs.

References

[1] Silvia T Acuña, Marta Gómez, and Natalia Juristo. How do personality, team processes and task characteristics relate to job satisfaction and software quality? *Information and Software Technology*, 51(3):627–639, 2009.

[2] Karan Aggarwal, Abram Hindle, and Eleni Stroulia. Co-evolution of project documentation and popularity within github. In *Proceedings of the 11th Working Conference on Mining Software Repositories*, pages 360–363, 2014.

[3] Gordon W Allport and Henry S Odbert. Trait-names: A psycho-lexical study. *Psychological monographs*, 47(1):i, 1936.

[4] Pierre-Hadrien Arnoux, Anbang Xu, Neil Boyette, Jalal Mahmud, Rama Akkiraju, and Vibha Sinha. 25 tweets to know you: A new model to predict personality with social media. In *Eleventh International AAAI Conference on Web and Social Media*, 2017.

[5] Blerina Bazelli, Abram Hindle, and Eleni Stroulia. On the personality traits of stackoverflow users. In *2013 IEEE international conference on software maintenance*, pages 460–463. IEEE, 2013.

[6] David Bell, Tracy Hall, Jo Erskine Hannay, Dietmar Pfahl, and Silvia Teresita Acuna. Software engineering group work: personality, patterns and performance. In *Proceedings of the 2010 Special Interest Group on Management Information System's 48th annual conference on Computer personnel research on Computer personnel research*, pages 43–47, 2010.

[7] Hudson Borges and Marco Tulio Valente. What's in a github star? understanding repository starring practices in a social coding platform. *Journal of Systems and Software*, 146:112–129, 2018.

[8] Gregory J Boyle. Myers-briggs type indicator (mbti): some psychometric limitations. *Australian Psychologist*, 30(1):71–74, 1995.

[9] Bret H Bradley, John E Baur, Christopher G Banford, and Bennett E Postlethwaite. Team players and collective performance: How agreeableness affects team performance over time. *Small Group Research*, 44(6):680–711, 2013.

[10] Donn Byrne and William Griffitt. Similarity and awareness of similarity of personality characteristics as determinants of attraction. *Journal of Experimental Research in Personality*, 1969.

[11] Fabio Calefato, Giuseppe Iaffaldano, Filippo Lanubile, and Bogdan Vasilescu. On developers' personality in large-scale distributed projects: the case of the apache ecosystem. In *2018 IEEE/ACM 13th International Conference on Global Software Engineering (ICGSE)*, pages 87–96. IEEE, 2018.

[12] Fabio Calefato and Filippo Lanubile. Establishing personal trust-based connections in distributed teams. *Internet Technology Letters*, 1(4):e6, 2017.

[13] Fabio Calefato, Filippo Lanubile, and Bogdan Vasilescu. A large-scale, in-depth analysis of developers' personalities in the apache ecosystem. *Information and Software Technology*, 114:1–20, 2019.

[14] Giulio Carducci, Giuseppe Rizzo, Diego Monti, Enrico Palumbo, and Maurizio Morisio. Twitpersonality: Computing personality traits from tweets using word embeddings and supervised learning. *Information*, 9(5):127, 2018.

[15] Raymond B Cattell. The description of personality: Basic traits resolved into clusters. *The journal of abnormal and social psychology*, 38(4):476, 1943.

[16] Cindy Chung and James W Pennebaker. The psychological functions of function words. *Social communication*, 1:343–359, 2007.

[17] Jailton Coelho, Marco Tulio Valente, Luciana L Silva, and Emad Shihab. Identifying unmaintained projects in github. In *Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, pages 1–10, 2018.

[18] Paul T Costa Jr and Robert R McCrae. *The Revised NEO Personality Inventory (NEO-PI-R).* Sage Publications, Inc, 2008.

[19] Colin G DeYoung, Lena C Quilty, and Jordan B Peterson. Between facets and domains: 10 aspects of the big five. *Journal of personality and social psychology*, 93(5):880, 2007.

[20] John M Digman. Personality structure: Emergence of the five-factor model. *Annual review of psychology*, 41(1):417–440, 1990.

[21] Robert Feldt, Lefteris Angelis, Richard Torkar, and Maria Samuelsson. Links between the personalities, views and attitudes of software engineers. *Information and Software Technology*, 52(6):611–624, 2010.

[22] Robert Feldt, Richard Torkar, Lefteris Angelis, and Maria Samuelsson. Towards individualized software engineering: empirical studies should collect psychometrics. In *Proceedings of the 2008 international workshop on Cooperative and human aspects of software engineering*, pages 49–52, 2008.

[23] Martin Gerlach, Beatrice Farb, William Revelle, and Luís A Nunes Amaral. A robust data-driven approach identifies four personality types across four large data sets. *Nature human behaviour*, 2(10):735–742, 2018.

[24] Lewis R Goldberg et al. A broad-bandwidth, public domain, personality inventory measuring the lower-level facets of several five-factor models. *Personality psychology in Europe*, 7(1):7–28, 1999.

[25] Georgios Gousios and Andy Zaidman. A dataset for pull-based development research. In *Proceedings of the 11th Working Conference on Mining Software Repositories*, pages 368–371, 2014.

[26] Jo E Hannay, Erik Arisholm, Harald Engvik, and Dag IK Sjoberg. Effects of personality on pair programming. *IEEE Transactions on Software Engineering*, 36(1):61–80, 2010.

[27] Robert Hogan. What is personality psychology? *Psychological Inquiry*, 9(2):152–153, 1998.

[28] Gregory M Hurtz and John J Donovan. Personality and job performance: The big five revisited. *Journal of applied psychology*, 85(6):869, 2000.

[29] Rahul N Iyer, S Alex Yun, Meiyappan Nagappan, and Jesse Hoey. Effects of personality traits on pull request acceptance. *IEEE Transactions on Software Engineering*, 2019.

[30] Timothy A Judge, Joyce E Bono, Remus Ilies, and Megan W Gerhardt. Personality and leadership: a qualitative and quantitative review. *Journal of applied psychology*, 87(4):765, 2002.

[31] Tanjila Kanij, Robert Merkel, and John Grundy. An empirical investigation of personality traits of software testers. In *2015 IEEE/ACM 8th International Workshop on Cooperative and Human Aspects of Software Engineering*, pages 1–7. IEEE, 2015.

[32] Margaret L Kern, Paul X McCarthy, Deepanjan Chakrabarty, and Marian-Andrei Rizoiu. Social media-predicted personality traits and values can help match people to their ideal jobs. *Proceedings of the National Academy of Sciences*, 116(52):26459–26464, 2019.

[33] Makrina Viola Kosti, Robert Feldt, and Lefteris Angelis. Personality, emotional intelligence and work preferences in software engineering: An empirical study. *Information and Software Technology*, 56(8):973–990, 2014.

[34] Makrina Viola Kosti, Robert Feldt, and Lefteris Angelis. Archetypal personalities of software engineers and their work preferences: a new perspective for empirical studies. *Empirical Software Engineering*, 21(4):1509–1532, 2016.

[35] Kwan Min Lee, Wei Peng, Seung-A Jin, and Chang Yan. Can robots manifest personality?: An empirical test of personality recognition, social responses, and social presence in human–robot interaction. *Journal of communication*, 56(4):754–772, 2006.

[36] Fei Liu, Julien Perez, and Scott Nowson. A language-independent and compositional model for personality trait recognition from short texts. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 754–764, Valencia, Spain, April 2017. Association for Computational Linguistics.

[37] François Mairesse, Marilyn A Walker, Matthias R Mehl, and Roger K Moore. Using linguistic cues for the automatic recognition of personality in conversation and text. *Journal of artificial intelligence research*, 30:457–500, 2007.

[38] Navonil Majumder, Soujanya Poria, Alexander Gelbukh, and Erik Cambria. Deep learning-based document modeling for personality detection from text. *IEEE Intelligent Systems*, 32(2):74–79, 2017.

[39] Robert R McCrae and Paul T Costa. Validation of the five-factor model of personality across instruments and observers. *Journal of personality and social psychology*, 52(1):81, 1987.

[40] RR McCrae, PT Costa, GJ Boyle, G Matthews, and DH Saklofske. *Sage handbook of personality theory and assessment*. Boyle, 2008.

[41] Matthias R Mehl, Samuel D Gosling, and James W Pennebaker. Personality in its natural habitat: manifestations and implicit folk theories of personality in daily life. *Journal of personality and social psychology*, 90(5):862, 2006.

[42] Emanuel Mellblom, Isar Arason, Lucas Gren, and Richard Torkar. The connection between burnout and personality types in software developers. *arXiv preprint arXiv:1906.09463*, 2019.

[43] Isabel Briggs Myers, Mary H McCaulley, and Robert Most. *Manual, a guide to the development and use of the Myers-Briggs type indicator*. consulting psychologists press, 1985.

[44] George A Neuman, Stephen H Wagner, and Neil D Christiansen. The relationship between work-team personality composition and the job performance of teams. *Group & Organization Management*, 24(1):28–45, 1999.

[45] Warren T Norman. Toward an adequate taxonomy of personality attributes: Replicated factor structure in peer nomination personality ratings. *The Journal of Abnormal and Social Psychology*, 66(6):574, 1963.

[46] Oscar Hernán Paruma Pabón, Fabio A González, Jairo Aponte, Jorge E Camargo, and Felipe Restrepo-Calle. Finding relationships between socio-technical aspects and personality traits by mining developer e-mails. In *2016 IEEE/ACM Cooperative and Human Aspects of Software Engineering (CHASE)*, pages 8–14. IEEE, 2016.

[47] James W Pennebaker, Ryan L Boyd, Kayla Jordan, and Kate Blackburn. The development and psychometric properties of liwc2015. Technical report, 2015.

[48] James W Pennebaker and Laura A King. Linguistic styles: Language use as an individual difference. *Journal of personality and social psychology*, 77(6):1296, 1999.

[49] Rajnandini Pillai, Chester A Schriesheim, and Eric S Williams. Fairness perceptions and trust as mediators for transformational and transactional leadership: A two-sample study. *Journal of management*, 25(6):897–933, 1999.

[50] David J Pittenger. Cautionary comments regarding the myers-briggs type indicator. *Consulting Psychology Journal: Practice and Research*, 57(3):210, 2005.

[51] Ayushi Rastogi and Nachiappan Nagappan. On the personality traits of github contributors. In *2016 IEEE 27th International Symposium on Software Reliability Engineering (ISSRE)*, pages 77–86. IEEE, 2016.

[52] Peter C Rigby and Ahmed E Hassan. What can oss mailing lists tell us? a preliminary psychometric text analysis of the apache developer mailing list. In *Fourth International Workshop on Mining Software Repositories (MSR'07: ICSE Workshops 2007)*, pages 23–23. IEEE, 2007.

[53] Norsaremah Salleh, Emilia Mendes, and John Grundy. Investigating the effects of personality traits on pair programming in a higher education setting through a family of experiments. *Empirical Software Engineering*, 19(3):714–752, 2014.

[54] Norsaremah Salleh, Emilia Mendes, John Grundy, and Giles St J Burch. An empirical study of the effects of personality in pair programming using the five-factor model. In *2009 3rd International Symposium on Empirical Software Engineering and Measurement*, pages 214–225. IEEE, 2009.

[55] Norsaremah Salleh, Emilia Mendes, John Grundy, and Giles St J Burch. The effects of neuroticism on pair programming: an empirical study in the higher education context. In *Proceedings of the 2010 ACM-IEEE international symposium on empirical software engineering and measurement*, pages 1–10, 2010.

[56] Norsaremah Salleh, Emilia Mendes, John Grundy, and Giles St J Burch. An empirical study of the effects of conscientiousness in pair programming using the five-factor personality model. In *2010 ACM/IEEE 32nd International Conference on Software Engineering*, volume 1, pages 577–586. IEEE, 2010.

[57] David P Schmitt, Jüri Allik, Robert R McCrae, and Verónica Benet-Martínez. The geographic distribution of big five personality traits: Patterns and profiles of human self-description across 56 nations. *Journal of cross-cultural psychology*, 38(2):173–212, 2007.

[58] H Andrew Schwartz, Johannes C Eichstaedt, Margaret L Kern, Lukasz Dziurzynski, Stephanie M Ramones, Megha Agrawal, Achal Shah, Michal Kosinski, David Stillwell, Martin EP Seligman, et al. Personality, gender, and age in the language of social media: The open-vocabulary approach. *PloS one*, 8(9):e73791, 2013.

[59] Edward K Smith, Christian Bird, and Thomas Zimmermann. Beliefs, practices, and personalities of software engineers: a survey in a large software company. In *Proceedings of the 9th International Workshop on Cooperative and Human Aspects of Software Engineering*, pages 15–18, 2016.

[60] John J Sosik, Surinder S Kahai, and Bruce J Avolio. Transformational leadership and dimensions of creativity: Motivating idea generation in computer-mediated groups. *Creativity Research Journal*, 11(2):111–121, 1998.

[61] Davide Spadini, Maurício Aniche, and Alberto Bacchelli. PyDriller: Python framework for mining software repositories. In *Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering - ESEC/FSE 2018*, pages 908–911, New York, New York, USA, 2018. ACM Press.

[62] Adriana Tapus and Maja J Mataric. Socially assistive robots: The link between personality, empathy, physiological signals, and task performance. In *AAAI spring symposium: emotion, personality, and social behavior*, pages 133–140, 2008.

[63] Yla R Tausczik and James W Pennebaker. The psychological meaning of words: Liwc and computerized text analysis methods. *Journal of language and social psychology*, 29(1):24–54, 2010.

[64] Michael E Tipping and Christopher M Bishop. Mixtures of probabilistic principal component analyzers. *Neural computation*, 11(2):443–482, 1999.

[65] Michael E Tipping and Christopher M Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3):611–622, 1999.

[66] Jason Tsay, Laura Dabbish, and James Herbsleb. Influence of social and technical factors for evaluating contribution in github. In *Proceedings of the 36th international conference on Software engineering*, pages 356–366, 2014.

[67] Chockalingam Viswesvaran and Deniz S Ones. Measurement error in "big five factors" personality assessment: Reliability generalization across studies and measures. *Educational and Psychological Measurement*, 60(2):224–235, 2000.

[68] Yi Wang. Building the linkage between project managers' personality and success of software projects. In *2009 3rd International Symposium on Empirical Software Engineering and Measurement*, pages 410–413. IEEE, 2009.

[69] Stanley Wasserman, Katherine Faust, et al. *Social network analysis: Methods and applications*, volume 8. Cambridge university press, 1994.

[70] Tal Yarkoni. Personality in 100,000 words: A large-scale analysis of personality and word use among bloggers. *Journal of research in personality*, 44(3):363–373, 2010.

APPENDICES

Appendix A

Descriptive Statistics

Table A.1: Repository-level measures of maintained repositories with descriptive statistics

Variables	**Description**	**Mean**	**SD**	**Min**	**Median**	**Max**
age	Age of the repository in days.	2874	757	1489	2696	4357
size	Size of the repository in kilobytes.	142841	199898	2271	68015	961013
collaborators	Number of collaborators.	99	295	3	22	1496
forks	Number of forks.	4696	5660	680	1784	18584
issues_open	Number of open issues.	448	792	0	284	4101
issues_closed	Number of closed issues.	1365	4541	25	480	23125
popularity	$\texttt{(stars)} + \texttt{(watchers)} + \texttt{(pull requests)}^2$	71883941	357241900	22653	428599	1786644638
pr_open	Number of open pull requests.	167	350	7	46	1750
pr_closed	Number of closed pull requests.	2103	8007	3	563	40518
stars	Number of stars.	17111	21091	2802	8012	88437
watchers	Number of watchers.	829	872	139	530	3185

Table A.2: Repository-level measures of unmaintained repositories with descriptive statistics

Variables	Description	Mean	SD	Min	Median	Max
age	Age of the repository in days.	1887	1043	203	1884	4347
size	Size of the repository in kilobytes.	64848	79037	1190	30536	267939
collaborators	Number of collaborators.	91	302	1	10	1495
forks	Number of forks.	1698	2528	95	912	12333
issues_open	Number of open issues.	96	131	0	20	430
issues_closed	Number of closed issues.	396	311	4	407	962
popularity	$(\texttt{stars}) + (\texttt{watchers}) + (\texttt{pull requests})^2$	305755	293378	1184	230899	977854
pr_open	Number of open pull requests.	15	34	0	5	168
pulls_closed	Number of closed pull requests.	442	302	4	454	986
stars	Number of stars.	6553	8749	540	2556	31560
watchers	Number of watchers.	289	300	51	194	1282

Table A.3: Issues-level measures of maintained repositories with descriptive statistics

Variables	**Description**	**Mean**	**SD**	**Min**	**Median**	**Max**
experience	Time between the first and the last commit merged by a maintainer.	1798.1	1179.6	0.0	1568.0	7435.0
mdn_addition	Median lines of code added by a maintainer per commit.	13.2	13.8	2.0	9.0	108.5
mdn_deletion	Median lines of code deleted by a maintainer per commit.	4.4	3.5	1.0	3.0	20.0
num_commits	Number of commits contributed by a maintainer.	1252.5	1843.9	2.0	550.0	13102.0
num_issues	Number of issues created by a maintainer.	309.5	435.9	0.0	145.0	2363.0
num_pr	Number of pull requests created by a maintainer.	206.0	334.7	0.0	92.0	2044.0
num_issue_comments	Number of issue comments made by a maintainer.	1296.0	2288.8	0.0	495.0	13036.0
num_pr_comments	Number of pull request comments made by a maintainer.	867.7	1577.3	0.0	213.0	10727.0

Table A.4: Issues-level measures of unmaintained repositories with descriptive statistics

Variables	**Description**	**Mean**	**SD**	**Min**	**Median**	**Max**
experience	Time between the first and the last commit merged by a maintainer.	1223.7	948.7	117.0	1032.5	4225.0
mdn_addition	Median lines of code added by a maintainer per commit.	10.4	14.1	1.0	7.0	112.5
mdn_deletion	Median lines of code deleted by a maintainer per commit.	3.9	4.9	1.0	2.8	36.5
num_commits	Number of commits contributed by a maintainer.	864.6	2174.1	10.0	238.0	14554.0
num_issues	Number of issues created by a maintainer.	285.9	500.3	0.0	109.0	2780.0
num_pr	Number of pull requests created by a maintainer.	150.9	255.5	0.0	51.5	1284.0
num_issue_comments	Number of issue comments made by a maintainer.	1356.9	2555.2	0.0	552.0	18504.0
num_pr_comments	Number of pull request comments made by a maintainer.	588.6	1449.1	0.0	159.5	10938.0

Table A.5: Personality traits of developers in maintained repositories with descriptive statistics

Traits	**Mean**	**SD**	**Min**	**Median**	**Max**
openness	-5.72	1.73	-14.85	-5.78	1.38
conscientiousness	-2.39	0.79	-7.24	-2.49	3.21
extraversion	-0.33	0.92	-10.29	-0.19	3.63
agreeableness	3.85	0.74	-1.74	3.84	9.62
neuroticism	2.05	0.81	-1.56	2.09	6.72

Table A.6: Personality traits of developers in unmaintained repositories with descriptive statistics

Traits	**Mean**	**SD**	**Min**	**Median**	**Max**
openness	-5.77	2.10	-14.36	-5.95	0.28
conscientiousness	-2.14	0.83	-4.72	-2.24	1.18
extraversion	-0.39	1.09	-11.90	-0.25	6.44
agreeableness	3.89	0.92	-1.16	3.91	14.03
neuroticism	1.94	0.90	-1.34	2.01	5.14

Table A.7: LIWC output in maintained repositories with descriptive statistics

Dimension	**Labels**	**Mean**	**SD**	**Min**	**Median**	**Max**
Word count	WC	3751.83	28282.20	500.00	916.00	2065266.00
Summary Variables						
Analytic thinking	Analytic	81.51	11.87	13.50	84.62	99.00
Clout	Clout	43.44	13.92	1.00	42.75	96.79
Authentic	Authentic	28.68	17.86	1.00	26.32	98.15
Emotional tone	Tone	43.85	20.21	1.00	42.04	99.00
Language Metrics						
Words per sentence	WPS	50.50	160.99	5.91	26.44	5411.00
Words > 6 letters	Sixltr	23.01	4.90	0.90	22.64	60.64
Dictionary words	Dic	63.32	14.39	0.37	67.49	91.34
Function Words	function	37.15	11.95	0.00	40.03	60.51
Total pronouns	pronoun	7.80	3.48	0.00	8.02	23.78
Personal pronouns	ppron	3.66	1.89	0.00	3.57	18.54
First-person singular	i	2.05	1.58	0.00	1.81	13.68
First-person plural	we	0.41	0.55	0.00	0.20	9.26
Second-person	you	1.00	0.93	0.00	0.76	17.00
Third-person singular	shehe	0.01	0.09	0.00	0.00	5.17
Third-person plural	they	0.18	0.24	0.00	0.11	2.75
Impersonal pronouns	ipron	4.14	2.07	0.00	4.22	23.78
Articles	article	6.09	2.65	0.00	6.55	15.88
Prepositions	prep	10.25	3.05	0.00	10.92	20.33
Auxiliary verbs	auxverb	6.29	2.61	0.00	6.55	17.86
Common adverbs	adverb	3.32	1.56	0.00	3.40	13.16
Conjunctions	conj	4.70	1.82	0.00	4.97	14.01
Negations	negate	1.42	0.73	0.00	1.38	14.62
Other Grammar						
Regular verbs	verb	11.91	3.81	0.00	12.43	25.12
Adjectives	adj	3.20	1.31	0.00	3.24	13.33
Comparatives	compare	1.77	0.95	0.00	1.75	12.50
Interrogatives	interrog	1.09	0.65	0.00	1.08	8.60
Numbers	number	5.06	6.00	0.00	2.90	85.96
Quantifiers	quant	1.84	0.95	0.00	1.79	10.22
Affect Words	affect	3.18	1.29	0.00	3.09	16.58

Positive emotion	posemo	2.05	1.07	0.00	1.95	15.34
Negative emotion	negemo	1.11	0.77	0.00	1.00	12.62
Anxiety	anx	0.08	0.15	0.00	0.00	3.34
Anger	anger	0.11	0.24	0.00	0.00	9.86
Sadness	sad	0.32	0.34	0.00	0.25	6.04
Social Words	social	3.97	1.92	0.00	3.81	17.19
Family	family	0.02	0.12	0.00	0.00	4.46
Friends	friend	0.05	0.15	0.00	0.00	3.51
Female referents	female	0.03	0.25	0.00	0.00	9.89
Male referents	male	0.03	0.12	0.00	0.00	5.17
Cognitive Processes	cogproc	12.43	3.78	0.08	13.09	29.76
Insight	insight	2.02	1.08	0.00	1.92	14.21
Cause	cause	2.96	1.26	0.00	2.93	16.58
Discrepancies	discrep	2.04	1.03	0.00	2.07	9.38
Tentativeness	tentat	2.73	1.30	0.00	2.76	13.30
Certainty	certain	1.21	0.70	0.00	1.16	8.62
Differentiation	differ	3.65	1.42	0.00	3.76	14.99
Perceptual Processes	percept	1.07	0.83	0.00	0.92	14.56
Seeing	see	0.75	0.68	0.00	0.61	12.20
Hearing	hear	0.15	0.30	0.00	0.09	14.56
Feeling	feel	0.11	0.20	0.00	0.00	6.05
Biological Processes	bio	0.32	0.44	0.00	0.20	8.32
Body	body	0.08	0.21	0.00	0.00	5.53
Health/illness	health	0.17	0.31	0.00	0.07	8.32
Sexuality	sexual	0.03	0.19	0.00	0.00	7.74
Ingesting	ingest	0.07	0.24	0.00	0.00	7.28
Drives and Needs	drives	5.49	1.84	0.00	5.45	23.78
Affiliation	affiliation	0.97	0.80	0.00	0.79	11.02
Achievement	achieve	1.49	0.80	0.00	1.42	13.93
Power	power	2.12	1.29	0.00	1.84	12.73
Reward focus	reward	0.90	0.60	0.00	0.83	13.43
Risk focus	risk	0.68	0.51	0.00	0.60	7.98
Time Orientations						
Past focus	focuspast	1.99	1.03	0.00	1.91	11.89
Present focus	focuspresent	8.61	2.95	0.00	9.04	21.74
Future focus	focusfuture	0.87	0.55	0.00	0.83	5.68
Relativity	relativ	11.25	2.87	0.00	11.36	30.84
Motion	motion	2.65	1.69	0.00	2.33	21.43

Space	space	5.22	1.66	0.00	5.25	18.51
Time	time	3.45	1.50	0.00	3.32	24.50
Personal Concerns						
Work	work	2.84	1.50	0.00	2.58	24.00
Leisure	leisure	0.40	0.53	0.00	0.27	12.62
Home	home	0.21	0.48	0.00	0.00	7.91
Money	money	0.16	0.36	0.00	0.02	5.73
Religion	relig	0.01	0.05	0.00	0.00	1.91
Death	death	0.06	0.28	0.00	0.00	10.48
Informal Speech	informal	1.22	1.17	0.00	0.99	24.07
Swear words	swear	0.01	0.07	0.00	0.00	2.45
Netspeak	netspeak	0.98	1.14	0.00	0.73	24.07
Assent	assent	0.15	0.29	0.00	0.07	14.26
Nonfluencies	nonflu	0.07	0.14	0.00	0.00	3.76
Fillers	filler	0.02	0.05	0.00	0.00	0.88
All Punctuation	AllPunc	36.13	21.32	0.55	28.90	400.98
Periods	Period	6.86	4.60	0.00	6.25	200.29
Commas	Comma	3.47	2.39	0.00	3.27	85.05
Colons	Colon	3.07	4.18	0.00	1.68	76.49
Semicolons	SemiC	0.15	0.63	0.00	0.00	25.57
Question marks	QMark	0.43	0.56	0.00	0.33	27.12
Exclamation marks	Exclam	0.16	0.32	0.00	0.06	11.06
Dashes	Dash	5.09	8.79	0.00	2.54	298.43
Quotation marks	Quote	1.51	3.82	0.00	0.69	104.62
Apostrophes	Apostro	1.61	1.74	0.00	1.27	63.71
Parentheses (pairs)	Parenth	2.81	3.05	0.00	1.98	54.11
Other punctuation	OtherP	10.98	11.32	0.00	7.41	181.08

Table A.8: LIWC output in unmaintained repositories with descriptive statistics

Dimension	**Labels**	**Mean**	**SD**	**Min**	**Median**	**Max**
Word count	WC	3062.61	15081.28	500.00	940.00	661298.00
Summary Variables						
Analytic thinking	Analytic	77.89	13.26	18.62	80.17	99.00
Clout	Clout	41.00	13.93	6.93	39.84	99.00
Authentic	Authentic	25.27	17.72	1.00	23.08	93.53
Emotional tone	Tone	46.23	20.74	1.00	44.78	99.00
Language Metrics						
Words per sentence	WPS	49.79	189.49	7.43	25.18	6472.00
Words > 6 letters	Sixltr	20.80	4.49	0.17	20.49	81.57
Dictionary words	Dic	61.18	17.14	0.22	67.27	94.17
Function Words	function	36.75	14.19	0.00	42.03	57.44
Total pronouns	pronoun	8.48	4.06	0.00	9.24	20.23
Personal pronouns	ppron	4.02	2.22	0.00	4.08	14.92
First-person singular	i	2.54	1.66	0.00	2.35	11.51
First-person plural	we	0.38	0.57	0.00	0.16	6.52
Second-person	you	0.85	1.00	0.00	0.61	12.54
Third-person singular	shehe	0.02	0.08	0.00	0.00	1.61
Third-person plural	they	0.23	0.30	0.00	0.15	2.40
Impersonal pronouns	ipron	4.45	2.33	0.00	4.87	11.42
Articles	article	5.41	2.76	0.00	5.95	13.88
Prepositions	prep	9.97	3.54	0.00	10.97	16.96
Auxiliary verbs	auxverb	6.57	2.98	0.00	7.26	15.86
Common adverbs	adverb	3.46	1.77	0.00	3.70	9.92
Conjunctions	conj	4.37	1.99	0.00	4.77	10.24
Negations	negate	1.51	0.81	0.00	1.49	7.33
Other Grammar						
Regular verbs	verb	12.04	4.53	0.00	13.24	39.34
Adjectives	adj	3.07	1.58	0.00	3.18	49.67
Comparatives	compare	1.82	1.01	0.00	1.89	6.59
Interrogatives	interrog	1.00	0.64	0.00	0.97	4.91
Numbers	number	5.79	6.90	0.00	3.28	99.35
Quantifiers	quant	1.78	0.96	0.00	1.79	8.36
Affect Words	affect	3.10	1.37	0.00	3.05	12.23

Positive emotion	posemo	2.08	1.16	0.00	2.01	12.06
Negative emotion	negemo	1.00	0.73	0.00	0.88	10.98
Anxiety	anx	0.07	0.12	0.00	0.00	1.64
Anger	anger	0.11	0.23	0.00	0.00	4.18
Sadness	sad	0.30	0.32	0.00	0.22	4.39
Social Words	social	3.74	2.28	0.00	3.37	20.89
Family	family	0.02	0.10	0.00	0.00	2.12
Friends	friend	0.08	0.26	0.00	0.00	6.82
Female referents	female	0.02	0.19	0.00	0.00	6.82
Male referents	male	0.06	0.21	0.00	0.00	7.63
Cognitive Processes	cogproc	11.24	4.08	0.00	12.00	23.90
Insight	insight	1.83	0.99	0.00	1.78	9.17
Cause	cause	2.49	1.18	0.00	2.47	12.65
Discrepancies	discrep	1.71	1.04	0.00	1.72	8.02
Tentativeness	tentat	2.60	1.33	0.00	2.67	9.16
Certainty	certain	1.09	0.66	0.00	1.08	8.11
Differentiation	differ	3.26	1.46	0.00	3.43	8.72
Perceptual Processes	percept	1.20	1.51	0.00	1.01	64.74
Seeing	see	0.80	1.39	0.00	0.62	64.74
Hearing	hear	0.21	0.41	0.00	0.11	8.82
Feeling	feel	0.14	0.21	0.00	0.08	3.02
Biological Processes	bio	0.41	1.20	0.00	0.27	66.30
Body	body	0.16	0.65	0.00	0.04	34.61
Health/illness	health	0.14	0.29	0.00	0.06	5.87
Sexuality	sexual	0.03	0.15	0.00	0.00	5.87
Ingesting	ingest	0.09	0.60	0.00	0.00	31.69
Drives and Needs	drives	5.00	1.90	0.00	4.99	16.64
Affiliation	affiliation	0.92	0.90	0.00	0.68	10.72
Achievement	achieve	1.47	0.77	0.00	1.41	7.02
Power	power	1.66	0.99	0.00	1.52	16.40
Reward focus	reward	0.95	0.65	0.00	0.92	9.91
Risk focus	risk	0.62	0.56	0.00	0.51	8.74
Time Orientations						
Past focus	focuspast	2.05	1.08	0.00	2.01	10.45
Present focus	focuspresent	8.90	3.41	0.00	9.68	18.58
Future focus	focusfuture	0.82	0.58	0.00	0.77	5.60
Relativity	relativ	10.08	2.93	0.00	10.34	24.52
Motion	motion	1.71	1.09	0.00	1.61	9.24

Space	space	5.01	1.78	0.00	5.01	19.60
Time	time	3.39	1.62	0.00	3.33	15.70
Personal Concerns						
Work	work	2.67	1.50	0.00	2.41	22.43
Leisure	leisure	0.51	0.93	0.00	0.26	33.15
Home	home	0.16	0.48	0.00	0.00	7.04
Money	money	0.25	0.67	0.00	0.07	9.98
Religion	relig	0.02	0.10	0.00	0.00	3.08
Death	death	0.03	0.13	0.00	0.00	3.76
Informal Speech	informal	1.61	1.78	0.00	1.25	39.11
Swear words	swear	0.02	0.11	0.00	0.00	3.99
Netspeak	netspeak	1.25	1.70	0.00	0.86	39.11
Assent	assent	0.23	0.51	0.00	0.14	9.46
Nonfluencies	nonflu	0.12	0.22	0.00	0.05	4.95
Fillers	filler	0.03	0.13	0.00	0.00	6.21
All Punctuation	AllPunc	38.52	25.47	8.79	29.03	583.48
Periods	Period	7.79	5.35	0.00	6.89	117.12
Commas	Comma	3.06	2.01	0.00	2.95	39.57
Colons	Colon	3.42	6.82	0.00	1.55	200.00
Semicolons	SemiC	0.17	0.91	0.00	0.00	32.24
Question marks	QMark	0.64	0.74	0.00	0.46	9.46
Exclamation marks	Exclam	0.25	0.72	0.00	0.11	24.91
Dashes	Dash	4.50	5.58	0.00	2.83	57.32
Quotation marks	Quote	0.95	2.25	0.00	0.44	79.47
Apostrophes	Apostro	2.08	1.88	0.00	1.83	39.56
Parentheses (pairs)	Parenth	2.17	2.23	0.00	1.71	32.74
Other punctuation	OtherP	13.49	17.25	0.00	7.41	564.57

Glossary

Agile software development An iterative method of organization and collaboration that aims to prioritize tasks, reduce delivery time, and promote incremental feature delivery. 12

Git Git is an open source distributed version control system that tracks any changes during software development. 5

repository A repository is a storage for projects, documentation, and metadata; it keeps track of file revision history. 1