Information and Software Technology 114 (2019) 1–20

Contents lists available at ScienceDirect

Information and Software Technology

journal homepage: www.elsevier.com/locate/infsof

A large-scale, in-depth analysis of developers' personalities in the Apache ecosystem

Fabio Calefato[a,*], Filippo Lanubile[a], Bogdan Vasilescu[b]

[a] University of Bari, Bari, Italy
[b] Carnegie Mellon University, Pittsburgh, PA, USA

ARTICLE INFO

Keywords:
Personality traits
Large-scale distributed projects
Ecosystems
Apache
Big five
Five-Factor model
Open source software
Human aspects
Psychometric analysis
Computational personality detection

ABSTRACT

Context: Large-scale distributed projects are typically the results of collective efforts performed by multiple developers with heterogeneous personalities.
Objective: We aim to find evidence that personalities can explain developers' behavior in large scale-distributed projects. For example, the propensity to trust others — a critical factor for the success of global software engineering — has been found to influence positively the result of code reviews in distributed projects.
Method: In this paper, we perform a quantitative analysis of ecosystem-level data from the code commits and email messages contributed by the developers working on the Apache Software Foundation (ASF) projects, as representative of large scale-distributed projects.
Results: We find that there are three common types of personality profiles among Apache developers, characterized in particular by their level of Agreeableness and Neuroticism. We also confirm that developers' personality is stable over time. Moreover, personality traits do not vary with their role, membership, and extent of contribution to the projects. We also find evidence that more open developers are more likely to make contributors to Apache projects.
Conclusion: Overall, our findings reinforce the need for future studies on human factors in software engineering to use psychometric tools to control for differences in developers' personalities.

1. Introduction

Personality has been a subject of interest in software engineering since the 1970s when Weinberg [1] first hypothesized that the study of personality could have a substantial impact on the performance of developers. Similarly, in the early 1980s, Shneiderman [2] argued that personality plays a critical role in determining how programmers interact while also complaining about the lack of studies and empirical evidence on the impact of personality factors. Since then, however, there has been *a substantial amount of research* that has investigated the effects of personality in software engineering; e.g., Cruz et al. [3] have identified 90 studies conducted between 1979 and 2014, most of which (~70%) after 2002.

The main reason for the widespread interest in personality-focused research in software engineering is the many practical applications, e.g., the prediction of performance in pair programming [4], work preferences [5], job satisfaction [6], and effective team composition [7]. However, prior research reports *contrasting findings* [3]. One reason for these conflicts lies in the complex and multi-faceted nature of personality. Considering that widespread agreement on the effectiveness of personality frameworks is still under debate in the social sciences (see Section 2.1), it is not surprising that some works on personality in software engineering report clear associations (e.g.,[8]) while others find little to no effects (e.g., [9]).

Another reason for the conflicting findings on the effects of personality in software engineering is the choice of different *psychometric constructs and related instruments* to assess personality. Personality research typically relies on self-assessment questionnaires. There exist many such instruments, but some of them have been heavily criticized for their lack of validity (e.g., the MBTI and KTS, see Section 2.1.2). Still, many studies on personality in software engineering have relied on such dated instruments (e.g., [10,11]). Furthermore, there are some clear problems associated with detecting the personality through questionnaires, such as the extremely low return rates, especially in the software domain [12], and the limited number of occasions (typically only one) to perform data collection [13].

This paper reports on a *large-scale empirical study* of the personality profiles of open source software (OSS) developers from 39 Apache Software Foundation (ASF) projects. OSS projects are an extreme form

* Corresponding author.
E-mail addresses: fabio.calefato@uniba.it (F. Calefato), filippo.lanubile@uniba.it (F. Lanubile), vasilescu@cmu.edu (B. Vasilescu).

https://doi.org/10.1016/j.infsof.2019.05.012
Received 24 September 2018; Received in revised form 11 March 2019; Accepted 28 May 2019
Available online 4 June 2019

of large-scale distributed projects in which no single organization controls the project [14] and, as such, the products developed are typically the results of collective efforts performed by multiple members, each having their different personality [15]. Hence, the study of personalities of OSS developers has the potential of explaining software engineers' behavior in distributed software development in general [16].

In particular, we first mined ecosystem-level data from ASF mailing list emails and code commits contributed by 211 developers over more than a decade (see Section 4.1). Then, using a recent advance in Psycholinguistic research – inferring personality from one's written communication style [17], we extracted the personality profiles of Apache developers and investigate what specific traits are associated with development productivity and the likelihood of becoming a core project contributor – a typical sign of recognition in OSS.

The study is informed by the *Big Five* personality framework (also known as the *Five-Factor model*) [18,19], which has gained a widespread consensus among trait psychologists regarding its validity [20]. Furthermore, we used a psychometric tool developed to automatically detect personality profiles from the wealth of data available from the ASF project repositories (see Section 4.1); this allowed us to perform multiple assessments of contributors' personalities over time.

Our contributions are the following:

- We analyze the output from IBM Personality Insights, a commercial psychometric tool used to automatically detect the personality of developers from their emails.
- Unlike prior studies that rely on questionnaires and collected data only once, we build and publicly release a dataset, consisting of both psychometric and development data, collected from the Apache developers participating in 39 ASF projects.
- We perform an empirical study with multiple statistical analyses to detect common personality profiles among 211 developers and assess the association of personality traits with the likelihood of becoming a project contributor as well as the extent of their contribution.
- Results of the empirical study show that there are three common patterns, or *types*, of personality profiles among Apache developers, characterized in particular by their level of Agreeableness and Neuroticism. Moreover, personality traits do not vary with their role, membership, and extent of contribution to the projects, while also remaining stable over time. Also, we find evidence that developers who exhibit higher levels of Openness and Agreeableness are more likely to make contributors to Apache projects.

The remainder of this paper is organized as follows. In Section 2, we provide an overview on personality and related research, with a specific focus on studies conducted in the software engineering domain. In Section 3, we present the research questions and the analyses performed. In Section 4, we describe the experiment, whose results are reported and discussed, respectively, in Section 5 and 6. Finally, we conclude in Section 7.

2. Background

In this section, we first provide an overview of personality, its concepts and definitions, the instruments used for its measurement, as well as the effect of language and culture (Section 2.1). Then, we review the most recent and relevant literature focusing on personality in the domain of software engineering (Section 2.2).

2.1. Personality theories

Personality is the set of all the attributes – behavioral, temperamental, emotional and mental – which characterize a unique individual [21]. Personality has been conceptualized from a variety of theoretical perspectives and at various level of abstractions. One frequently studied level is *personality traits* [22], a dynamic and organized set of dispositional attributes that create the unique pattern of behaviors, thoughts, and feelings of a person [23]. Accordingly, psychologists have sought descriptive models, or taxonomies, of such traits that would provide a framework that simplifies their efforts to organize, distinguish, and summarize the major individual differences among the myriad existing in human beings.

2.1.1. The big five traits and the five-Factor model

Many personality traits theories and associated instruments have been proposed since the 1930s, although more general acceptance and interest was not achieved until the 1970s.

Despite of the disagreement regarding the number of traits and their precise nature, there is a widespread agreement that the aspects of personality can be organized hierarchically [24]. After decades of research, thanks to the growing and compelling empirical evidence collected, the field has reached a strong consensus on the validity of a general taxonomy of five orthogonal personality traits, called the *Big Five*. The name was first used by Goldberg [25] not to imply that personality can be just reduced to five traits only, but rather to emphasize that five dimensions are sufficient to summarize at the broadest level the main dispositional characteristics and differences of individuals.

Big Five is an expression now considered a synonym with *Five-Factor Model* (FFM). However, the two are slightly different. Big Five is a general term used refer to personality frameworks that consist of five high-level dimensions. These five personality traits have been repeatedly obtained by applying factor analyses to various lists of trait adjectives used in self-descriptions and self-rating questionnaires for personality assessment. These studies have been conducted by psychologists based on the *lexical hypothesis* [26], according to which the most important individual characteristics and differences in personality have been encoded over time as words in the natural language, and the more important the difference, the more likely it is to be expressed as a single word (see [22] for more).

Unlike the Big Five, which only describes the five broad dimensions, the FFM [20,27] is a personality framework that further derives each of the five high-level traits into multiple lower-level facets (see Fig. 1):

- ***Openness*** (inventive/curious vs. consistent/cautious): it refers to the extent to which a person is open to experiencing a variety of activities, proactively seeking and appreciating unfamiliar experiences for its own sake. People low in Openness tend to be more conservative and close-minded.
- ***Conscientiousness*** (efficient/organized vs. easy-going/careless): it refers to people's tendency to plan in advance, act in an organized or thoughtful way and their degree of organization, persistence, and motivation in goal-directed behavior. Low-Conscientiousness individuals tend to be more tolerant and less bound by rules and plans.
- ***Extraversion*** (outgoing/energetic vs. solitary/reserved): it refers to the tendency to seek stimulation in the company of others, thus assessing people's amount of interpersonal interaction, activity level, need for stimulation, and capacity for joy. Those low in Extraversion are reserved and solitary.
- ***Agreeableness*** (friendly/compassionate vs. challenging/detached): it refers to a person's tendency to be compassionate and cooperative toward others, concerning thoughts, feelings, and actions. Low Agreeableness is related to being suspicious, challenging, and antagonistic towards other people.
- ***Neuroticism*** (sensitive/nervous vs. secure/confident): it refers to the extent to which a person's emotions are sensitive to the environment, thus identifying individuals who lack in emotional stability, prone to psychological distress, anxiety, excessive cravings or urges. Those who have a low score in Neuroticism are calmer and more stable.

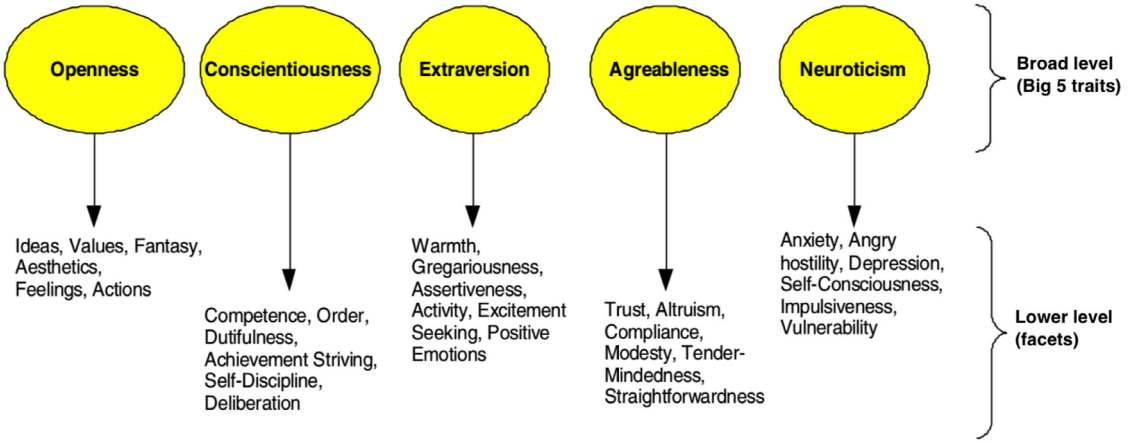

Fig. 1. The Five-Factor Model proposed by Costa & McCrae [27] and used as reference in this study. The Big Five traits are often referred to by the mnemonic OCEAN (image adapted from [28]).

Several independent studies on the FFM (see [29] for more), starting from different taxonomies and questionnaires, have found consistent evidence of the existence of a latent personality structure of individuals, consisting of five main factors. In fact, albeit labeled differently, at the higher level the extracted models showed minor differences and, therefore, they could be generally mapped onto each other. These results confirmed the general ubiquity of five factors across various FFM instruments [24] and, combined with the findings from the studies on the lexical hypotheses, lead trait psychologists to argue that *any* personality model must encompass, at some level, the same Big Five dimensions [25].

Hence, for the sake of simplicity, from now on we will consider Big Five and FFM synonyms and use them interchangeably.

2.1.2. Personality detection from questionnaires

Personality traits have been generally determined using questionnaires, which present a variable number of items (typically tens to hundreds) that describe common situations and behaviors (e.g., "*Do you have frequent mood swings?*"). The subjects taking the test indicate the extent to which each item applies to them using a Likert scale, generally in the range of [1, 5]. Questions are positively or negatively related to a specific trait; based on the answer, a specific value is assigned to each of them. Finally, the trait score is computed by aggregating all the values of its related answers.

One of the first instruments to draw major interest has been the Myers-Briggs Type Indicator (MBTI) [30]. Based on Jung's theories, MBTI allows creating individual profiles along four dimensions through the administration of a 93-item inventory. Despite its popularity, the MBTI instrument has been widely criticized since the late 1980s due to severe psychometric limitations, such as the lack of validity and reliability [31–33].

Another popular personality instrument is the Keirsey Temperament Sorter (KTS) [34,35], a self-assessment questionnaire that classifies individuals according to four distinct profiles. The KTS instrument was inspired by the MBTI and, like the latter, its psychometric validity has been questioned over the years [36].

Given the large consensus gained in the field by the Big Five taxonomy and the validity issues reported of the other personality frameworks, in the remainder of this section we focus our review on the FFM only.

2.1.3. Personality across languages and cultures

Most of the self-report inventories for assessing personality traits are have been translated into numerous languages and used under the assumptions that personality constructs transcend human language and culture. In the last two decades, there have been efforts aimed at showing that such inventories were reliable and showed a consistent structure of Big Five factors across the languages (i.e., upon the translation of inventory items) and cultures of participants.

One of the most comprehensive and popular instruments designed to measure the Big Five traits are the questionnaires developed by McCrae & Costa, that is, the NEO-PI [27], NEO-PI-R [37], and NEO-FFI [37]. McCrae [38,39] reported the high level of internal reliability of the trait scales as well as the robustness of the factorial structure after translating and administrating the NEO-PI-R in more than three dozen countries. These results were useful to show that it is possible to use mean values to capture systematic differences across nations and world regions. In particular, neighboring countries showed similar means of traits compared to regions that are geographically separated [40]. Also, Asian and African regions were characterized by smaller variability than European and American countries, where the heterogeneity of traits was the largest observed [39].

Although the NEO-PI-R is perhaps the most elaborated and widely used instrument for measuring the personality traits related to the Big Five taxonomy, there are other questionnaires belonging to the family of instruments intended to measure the five broadest dimensions of personality. One such instrument is the Big Five Inventory (BFI) [22], which was used by Schmitt et al. [41] to conduct a large study on 56 nations, also arranged in 10 geographical and cultural regions, to make sense of the geographic distribution of the Big Five personality traits. The analysis of the overall responses showed a robust five-factor structure. The same Big Five structure was also congruent with those computed for each of the 10 geographic regions. There was also a high cross-instrument correlation across the BFI and the NEO-PI-R scales. Albeit the distribution of the Big Five traits across nations showed in general small differences, several systematic patterns were evident, especially at the world-region level. Specifically, Schmitt et al. [41] observed that the level of Extraversion was much lower in East Asia (i.e., China, Japan, Korea, and Taiwan) than in the rest of the world. Regarding Agreeableness, Africa scored significantly higher and East Asia scored significantly lower than the other world regions.

Because all the instruments above are proprietary,[1] personality psychologists have developed and validated the International Personality Item Pool (IPIP) and its follow-up IPIP-NEO, as open alternative Big Five inventories that are freely available on the Internet [42].

Given the evidence of the general validity across languages and cultures (as well as instruments), the choice of focusing on the Big Five

[1] The BFI inventory is proprietary but freely available for non-commercial purposes at www.outofservice.com/bigfive.

taxonomy of traits appears even more justified since our study is executed in the context of global software development.

2.1.4. Personality detection from text

Self-report inventories are the most popular psychometric instruments to assess personality among researchers and professionals because they are considered reliable and easy-to-use. However, utterances and written text are also known to convey a great deal of information about the speaker and writer in addition to the semantic content. One such type of information consists of cues to individual personality. Psychologists have been able to identify correlations between specific linguistic markers and personality traits [43]. To date, there has been limited but growing amount of work on the automatic detection of personality traits from conversation transcripts and written text [44]. Thanks to the advancements in artificial intelligence (AI) and the widespread diffusion of social media contents, researchers have explored methods for the automatic recognition of various types of pragmatic variation in text and conversations, both short-lived, such as emotion, sentiment, and opinions [45–47], and more long-term, such as personality [48,49].

In Table 1, we review some of the models from research as well as the existing tools to automatically detect personality traits from text. While we cannot claim that the table is complete – the systematic review of this research field is outside the scope of this study – it nonetheless provides an up-to-date overview of the state of the art in field of *automatic personality recognition* [50], often also referred to as *computational personality detection* [51].

Types of solution. Existing solutions for the automatic recognition of personality can be grouped in top-down and bottom-up [63]. A *top-down* solution makes use of external resources (e.g., psycholinguistic databases) and tests their associations with personality traits. A *bottom-up* solution, instead, starts from the data and seeks linguistic cues associated to personality traits. From Table 1, we can observe that, among the tools reviewed here, bottom-up solutions (9) outnumber those top-down (6). In particular, because of the recent advances in the AI field, the tools developed in the last years (e.g., [48,62]) are adopting bottom-up solutions that leverage deep-learning techniques, such as Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs), for processing a number of linguistic cues extracted from large text corpora.

Commercial tools. The most well-known resource, often used as an external psycholinguistic database in other top-down tools, is the Linguistic Inquiry and Word Count (LIWC, pronounced *luke*)[2] [64], a commercial, text-analysis program that counts words in psychologically-meaningful, predetermined categories. Pennebaker & King [43] used it to identify theoretically-predicted associations between linguistic features and each of the Big Five personality traits. Specifically, they used LIWC to count word categories of 2479 essays (i.e., unedited pieces of text) written in a controlled setting by volunteers who had also taken the BFI personality test. In line with the lexical hypothesis, they found that each of the personality traits was significantly associated with LIWC linguistic dimensions, thus providing evidence of the connections between language use and personality [65].

Another commercial tool available for automatic personality recognition is IBM Watson Personality Insights[3] (IBM PI)[61]. Earlier versions of the service (i.e., before December 2016) used the LIWC dictionary along with unspecified machine-learning models. However, the service now uses machine learning with an *open-vocabulary* approach [66] that, as opposed to the closed-vocabulary approach of LIWC, does not rely on any *a priori* word or category judgments. Models based on the open-vocabulary approach have been found to work well also in presence of small amount of text such as tweets [67]. Also, as per IBM release note,[4] the new version of Personality Insights reportedly outperforms the older LIWC-based model.

Research tools. Most of the existing work that adopted a bottom-up solution have employed Support Vector Machines (SVMs) as a supervised learner. The seminal work in this respect is the one by Argamon et al. [52]. They were the first to build SVM-based personality recognition models using several lexical features extracted from a corpus of 2263 essays written by students who also took the NEO-FFI personality questionnaire. However, Argamon et al. focused on the recognition of only Neuroticism and Extraversion.

Oberlander & Nowson [53] built upon the work of Argamon et al. and compared the performance of SVM models against Naïve Bayes networks built using *n*-grams. In this case, the authors analyzed 71 blog posts from volunteers who took a customized version of the IPIP test for the assessment of four of the Big Five traits (i.e., except for Openness). In follow-up work, Nowson & Oberlander [54], repeated the study on a larger dataset of 1672 blog posts written by as many bloggers.

The seminal work for top-down solutions is the Personality Recognizer[5] tool, developed by Mairesse et al. [21] upon conducting a series of experiments where multiple statistical models for personality detection from text were benchmarked. They developed multiple regression models using the same annotated dataset of essays used for the development of LIWC. However, other than using LIWC features, they augmented the models with other dimensions from the Medical Research Council (MRC) psycholinguistic database [68].

The work by Gill et al. [55] is another example of regression models built top-down by leveraging the LIWC database. In this study, the authors analyzed 5042 blog posts from 2393 volunteers who also took a custom, BFI-like personality questionnaire to assess four of the Big Five traits (i.e., except for Openness).

Quercia et al. [56] used regression analysis to predict the Big Five personality traits of 335 Twitter users after analyzing the content of their feeds. These users were selected among those in the myPersonality dataset compiled by Kosinski et al. [69], containing not only their Facebook posts and the answers to the 20-item IPIP questionnaire, but also the links to their public Twitter profiles.

Goldbeck et al. conducted two studies for recognizing personality in Facebook [57] and Twitter [58]. In the first study, they compared regression models built by analyzing 167 Facebook profiles and extracting LIWC features as well as social-network features from their profile and friend network. In the second study, Goldbeck et al. assessed regression models built by analyzing 2,000 tweets from 2000 Twitter users, using both LIWC and MRC psycholinguistic databases, as well as extracting social-network features from their profiles. In both cases, volunteers took the BFI test to establish the ground truth.

Among the reviewed studies, the one by Celli [59] is the only case of unsupervised personality classification model, that is, trained without relying on any personality-annotated dataset to establish ground truth. As such, none of the 156 users who contributed 473 posts on FriendFeed took any personality test. The model was built using the same linguistic features defined in Mairesse et al. [21].

Mohammad & Kiritchenko [60] tested the performance of SVMs on both the LIWC essay and myPersonality Facebook datasets. They extracted the features using an emotion corpus and lexicon, built *ad hoc* from the analysis of the hashtags included in the posts.

While most of existing studies leverage linguistic features for only one language (typically English), Liu et al. [62] developed C2W2S4PT, a multi-language personality classifier built with Recurrent Neural Networks by extracting word and sentence vectors from a corpus of 25,000 tweets, written in English, Spanish, or Italian by 300 volunteers who also took the BFI test.

[2] http://liwc.wpengine.com.

[3] www.ibm.com/watson/services/personality-insights.

[4] https://console.bluemix.net/docs/services/personality-insights/science.html#researchPrecise.

[5] http://s3.amazonaws.com/mairesse/research/personality/recognizer.html.

Table 1
Tools and models for computational personality detection. In column *Solution*, TD stands for Top-Down, BU for Bottom-Up. In column *Task*, C(*n*) stands for Classification with *n* classes, NS for continuous numerical score. The results are reported in column *Performance*, averaged or per trait, in terms of accuracy (ACC), correlation (*r*), Mean Absolute Error (MAE), or Root Mean Square Error (RMSE).

Tool / Model	Reference	License	Solution	Technique	Features	Subjects	Dataset	Validation (ground truth)	Task	Performance
LIWC	Pennebaker & King (1999) [43]	Commercial	BU	Word category frequencies	Closed vocabulary	2479	2479 written essays	BFI	-	word-trait correlations
-	Argamon et al. (2005) [52]	-	BU	SVM	Word category frequencies	>1,100	2263 written essays	NEO-FFI	C(2)	ACC: E=58.0, N=58.2
-	Oberlander & Nowson (2006) [53]	-	BU	SVM, NB	*n*-grams	71	71 blogs posts	Customized IPIP	C(5)	ACC: C=62.0, E=44.7, A=69.8, N=49.3
-	Nowson & Oberlander (2007) [54]	-	BU	SVM, NB	*n*-grams	1672	1672 blog posts	Customized IPIP	C(3)	ACC: C=47.7, E=44.2, A=46.6, N=40.2
Personality Recognizer	Mairesse et al. (2007) [21]	-	TD	Multiple regressions models	LIWC, MRC	2479	2479 written essays	LIWC dataset	C(2)	best ACC: O=62.1, C=55.3,E=55.0, A=55.8, N=57.3
-	Gill et al. (2009) [55]	-	TD	Ordered logistic regression	LIWC	2393	5042 blog posts	Custom BFI-like questionnaire	C(3)	avg $r \approx 0.1$ (except O)
-	Quercia et al. (2011) [56]	-	TD	Regression (M5)	#followers, #followings, listed counts	335	335 Twitter feeds	myPersonality dataset	NS	RMSE: O=.69, C=.76, E=.88, A=.79, N=.85
-	Goldbeck et al. (2011a) [57]	-	TD	Regression (GP, M5)	LIWC, profile info, friend network	167	167 Facebook profiles	BFI	NS	best MAE: O=.099, C.104=, E=.124, A=.109, N=.117
-	Goldbeck et al. (2011b) [58]	-	TD	Regression (GP, ZeroR)	LIWC, MRC, profile info	2000	2000 Tweets per subject	BFI	NS	best MAE: O=.119, C=.146, E=.160, A=.130, N=.182
-	Celli (2012) [59]	-	BU	Unsupervised classification	Mairesse linguistic features	156	473 FriendFeed posts	Custom validity metric	C(3)	avg ACC $\approx$ 63.1
-	Mohammad & Kiritchenko (2014) [60]	-	TD	SVM	hashtag emotion-lexicon, -corpus	2469 / 250	2469 written essays / 10,000 Facebook posts	LIWC dataset / myPersonality dataset	C(5)	best ACC: O=60.7, C=56.7, E=56.4, A=56.0, N=58.3 (LIWC) / O=54.4, C=59.5, E=55.4, A=59.2, N=56.6 (FB)
IBM PI	(2016)[61]	Commercial	BU	Unspecified machine learning model	Open vocabulary, word embedding	-	1,500-2,000 Tweets (validation)	Unspecified Big 5 questionnaire	NS	avg MAE $\approx$ .12 (EN) / avg $r \approx$ .33 (EN)
C2W2S4PT	Liu et al. (2017) [62]	-	BU	RNN	word-, sentence-vectors	152 / 110 / 38	14,166 Tweets (EN) / 9879 (IT) / 3678 (SPA)	BFI	NS	best RMSE: O=.10, C=.09, E=.09, A=.12, N=.14
SenticNet Personality	Majumder et al. (2017) [49]	-	BU	CNN (MLP, SVM)	Word embedding, Mairesse linguistic features	2468	2468 written essays	LIWC dataset	C(5)	best ACC: O=62.7, C=57.3, E=58.1, A=56.7, N=59.4
TwitPersonality	Carducci et al. (2018) [48]	Apache 2.0	BU	SVM (LReg, LASSO)	Word embedding	24 / 250	18,473 Tweets / 9913 Facebook posts	BFI / myPersonality dataset	NS	RMSE: O=,.38 C=.31, E=.30, A=.13, N=.27 (Twitter) / O=.33, C=.53, E=.708, A=.45, N=.56 (FB)

Majumder et al. [49] developed SenticNet Personality,[6] a deep-learning personality-detection model built using Convolutional Neural Networks. Using the LIWC essay dataset as ground truth, they trained different configurations by leveraging word embedding and the linguistic features defined in Mairesse et al.[21].

Finally, Carducci et al. [48] developed TwitPersonality,[7] a personality detection model that uses word vector representations of tweets fed to SVMs. The Twitter histories of 24 volunteers were retrieved along with their Big Five personality traits, measured using the BFI questionnaire. They used the results reported by Quercia et al. [56] as a baseline.

Performance. Overall, despite the growing number of works, the automatic recognition of personality from text is still an extremely complex task, whose performance and quality assessment is also challenging due to the difference in the evaluation procedures and the limited number of existing annotated gold standards, which are costly to produce [63]. Over the last years, a few evaluation campaigns have been organized on computational personality recognition tasks (e.g.,[63,70,71]) and the results drawn from them are no different from the picture obtained from the analysis of the performance of the tools reviewed in this section.

From Table 1, we observe that the personality recognition task is tackled as either a classification task with n={2, 3, 5} classes or as a prediction task for a continuous numeric outcome. In the first case, researchers (e.g.,[21,52–55,60]), discretize the personality scores on n values and classify people accordingly. For example, for n=2, the task becomes a binary classification where people are classified as *high* or *low* (e.g., one standard deviation above or below the mean, or top and bottom quartiles) in each trait. With such an approach, performance is typically assessed in terms of classification accuracy (ACC). Results in Table 1 show that ACC values are in the range of ∼40-70%. In one case, Gill et al.[55] relied on Pearson correlation to assess the accuracy of the ordered logistic regression classification of four personality traits (except Openness), discretized on 3 levels (i.e., {*low, medium, high*}); they found on average a correlation of $r \approx 0.1$ between predicted scores and those obtained from a custom (i.e., not validated) BFI-like questionnaire taken by participants.

According to Schwartz et al.[66], prediction on a continuous numeric scale is a more appropriate task for studies on automatic personality recognition. In such cases, the adopted performance metrics are the Mean Absolute Error (MAE, the average of the absolute value of the difference between the actual and predicted scores), the Root Mean Squared Error (RMSE, the standard deviation of the residuals, that is the prediction errors), and Pearson correlation (r, measured between the predicted and actual trait scores).

Regarding the studies that reported MAE as a performance metric, IBM Personality Insights achieved an average of ∼.12 over the five traits for English, in line with the ∼.15 reported by Goldberg et al.[58] in their study on Twitter and better than the ∼.11 reported by Goldberg et al.[57] the other similar study performed on Facebook.

As regards the three studies that used RMSE, Liu et al.[62] achieved an average of ∼.11 over the five traits, considerably smaller (i.e., better) than the average ∼.79 and ∼.28 - .52, reported respectively by Quercia et al.[56] and Carducci et al.[48].

As for studies reporting Pearson correlation, IBM Personality Insights achieved $r \approx .33$, averaged over the five traits and for English. This finding is entirely consistent with those from literature reviews on personality, showing uniformly that most psychological and behavioral constructs have small to medium effect sizes in the range $.10 - .40$ on a correlational scale [72]. As Meyer et al.[73] noted, achieving correlations above 0.30 in psychology studies is challenging, so much so that even the simple axiom according to which past behavior is predictive of future behavior has been found to produce mere correlations of $r \approx .39$. Accordingly, they argued that, instead of relying on unrealistic benchmarks based on the conventional cut-off points used for interpreting correlation coefficients, researchers who investigate psychological constructs should instead use a baseline in the order of magnitude of correlations independently measured in related work. In other words, both Meyer et al. [73] and Roberts et al. [72] have called for adjusting the norms that researchers hold for what the effect size is in psychology and related fields.

Finally, the work of Celli [59] provides a unique and interesting attempt of using unsupervised classification for recognizing personality traits without previously collecting self-assessments. Celli reported a classification accuracy of ∼63% with 3 classes. Also, he defined a validity metric to measure how stable the traits are across every single post written by an individual. Approaches like this might be useful to investigate large populations of users from whom it is difficult to collect questionnaires.

2.2. Studies on the big five in software engineering

In the following, we briefly review some of the most recent and relevant studies that analyzed personality traits in the domain of software engineering. We review them according to the type of psychometric instruments used, i.e., *questionnaires* vs. *computational personality detection tools*.

2.2.1. Software engineering studies using personality questionnaires

In their systematic literature review (SLR), Cruz et al. [3] lamented difficulties in the meta-analysis due to the contrasting findings reported in the primary studies. One reason was the number of the specific aspects that the studies focus on, such as investigating the effect of the personality of software engineers on job satisfaction and software quality [74], code review [75], and team composition [76]; other studies analyze the personality profiles of software engineers [8] to examine the correlations of personality traits with pair programming performance [9]. Another reason was the variety of personality assessment instruments used. Surprisingly, Cruz et al. found that, combined, about 60% of the primary studies in the SLR had employed either MBTI or KTS, although their validity has been heavily criticized for years. MBTI was also the most used instrument in the primary studies identified in the SLRs conducted by Karimi & Wagner [77] and Karimi et al. [78]. These outdated personality instruments are still used in recent research (e.g., [10]). McDonalds & Edwards [79] reviewed 13 empirical studies in software engineering that used MBTI and found several validity threats with obvious negative impact on the reliability and validity of these studies.

In the rest of this section, we restrict our review to the studies on personality in software engineering that leveraged the Big Five model, for which there is a compelling amount of evidence on its validity. Table 2 lists the most recent and relevant of such prior studies. Despite our choice of focusing studies using the Big Five model, it is still difficult to synthesize the results reported in Table 2. Arguably because of the variety of tests applied and different experimental settings, the results show no clear patterns.

First, we note that albeit the number (11) of studies on Big Five in software engineering is not large, the papers focus on five different aspects, on which the effect of personality was assessed, namely: team satisfaction (2), individual performance in teamwork (1), profiling personalities of software engineers (5), pair programming performance (2), and programming style (1). The choice of a specific aspect obviously influences the level of analysis, that is, studies on pair programming measured the programming performance of pairs, those focusing on developers' personality profiling focused on individual differences, and finally those focusing on team differences conducted analysis at team level, typically aggregating trait scores by computing the averages and standard deviations. Most of these studies were conducted in an academic context (7 out of 11) rather than professional.

Regarding the instruments, as expected, most studies employed the freely available IPIP instrument (7 out of 10), instead of proprietary

[6] https://github.com/SenticNet/personality-detection.
[7] https://github.com/D2KLab/twitpersonality.

Table 2
Studies on personality in software engineering relying on Big Five questionnaires.

Reference	Focus	Unit of analysis	Context	Questionnaire (#items)	Main findings
Acuña et al. (2009) [74]	Team satisfaction	Team	Academic	NEO-FFI (60)	Satisfaction associated with high AGR and CON, software quality with high EXT
Bell et al. (2010) [80]	Individual performance in team work	Individual	Academic	NEO-FFI (60)	No correlations found
Feldt et al. (2010) [8]	Profiling	Individual	Professional	IPIP (50)	Two clusters, moderate vs. intense (i.e., high on EXT and OPE)
Hannay et al. (2010) [9]	Pair Programming performance	Pair	Professional	Unspecified	Pair performance associated with high EXT
Salleh et al. (2012) [28]	Pair Programming performance	Pair	Academic	IPIP-NEO (120)	Pair performance associated with high OPE
Kosti et al. (2014) [5]	Profiling	Individual	Academic	mini-IPIP (20)	Two clusters, moderate vs. intense (i.e., high on OPE, AGR, and EXT)
Acuña et al. (2015) [6]	Team satisfaction	Team	Academic	NEO-FFI (60)	Satisfaction associated with high AGR, performance with high AGR and EXT
Karimi et al. (2015) [78]	Programming style	Individual	Academic	IPIP (50)	OPE and CON respectively associated with breadth- and depth- first programming styles
Kanij et al. (2015) [81]	Profiling	Individual	Professional	IPIP (50)	Testers are significantly more CON
Kosti et al. (2016) [82]	Profiling	Individual	Academic	mini-IPIP (20)	Four archetypes defined by the high/low levels of EXT and CON
Smith et al. (2016) [83]	Profiling	Individual	Professional	IPIP (50)	Agile devs more EXT and less NEU, managers are more CON and EXT, no differences between devs and testers

alternatives such as the NEO-FFI (3). Different versions of the IPIP tool were used, such as the version with 120 items, the small one with 50, and the minimal version with only 20 items. Considering the low return rate for questionnaires administered to software engineers [12], it is not surprising that researchers prefer the use of free personality instruments with the minimum possible number of items to increase the chance of participation.

An even more varied picture emerges from the analysis of the study findings, which we discuss with respect to their specific focus.

As regards teamwork, Acuña et al. [6,74] found that high Agreeableness is strongly associated with higher levels of job satisfaction.

With respect to pair programming performance, two independent replications, that is, Hannay et al. [9] and Salleh et al. [28], found contrasting results. The former study found no strong connections between personality traits and performance, except for a modest association with Extraversion. The latter, instead, reported a strong direct association between performance and Openness. However, the context was different, as Hannay et al. [9] analyzed professionals, while Salleh et al. [28] analyzed students.

The most recent trend in studying personality in the software engineering field is extracting developers' personality profiles. Feldt et al. [8] and Kosti et al. [5] conducted two replications, the former with professionals and the latter with students. Their findings were consistent, as they were able to identify two clusters of personalities among students/professionals, one called '*intense*' and the other '*moderate*,' characterized by whether individuals exhibit high levels of Extraversion and Openness. In a second follow-up study, Kosti et al. [82] conducted a second replication using a different clustering technique, called Archetypal Analysis, which allowed them to identify four archetypal personality profiles among student subjects, characterized by the combinations of high vs. low levels of Extraversion and Conscientiousness.

Kanij et al. [81] and Smith et al. [83] conducted two studies for characterizing professional developers' personalities based on their role. The former found that testers are significantly associated with higher Conscientiousness, whereas the latter found no difference in that respect. Instead, Smith et al. [83] found managers to be more conscientious and extraverted, agile developers more neurotic and extraverted.

Finally, Bell et al. [80] and Karimi et al. [78] conducted two studies that have not been replicated, thus providing unique results. Bell et al. [80] studied the effect of personality on individual academic performance in teamwork. They reported no correlations. Karimi et al. [78] found that students with higher level of Openness significantly prefer breadth-first programming style, whereas those high on Conscientiousness prefer depth-first.

2.2.2. Software engineering studies using automatic personality recognition

In this section, we review previous studies, listed in Table 3, which investigated the Big Five personality model in the software domain using psychometric tools for automatically extracting personality profiles from communication traces, such as emails and code-review comments. Overall, the findings from these studies show that personalities of developers i) vary with the degree of contribution (e.g., between core and peripheral developers) and ii) reputation, and iii) change over short periods.

Rigby & Hassan [84] studied the Big-Five personality traits of the four top developers of the Apache *httpd* project against a baseline built using LIWC on the entire mailing list corpus. Their preliminary results showed that two of the developers responsible for the major Apache releases have similar personalities, which are also different from the baseline extracted from the email corpus contributed by the other developers.

Bazelli et al. [85] performed a quasi-replication of the previous study using data collected from Stack Overflow instead of a mailing list. They found that the top reputed authors are more extroverted compared to medium and low reputed users, a personality profile consistent to the one observed by Rigby & Hassan [84] for the two top *Apache httpd* developers.

Rastogi & Nagappan [16] analyzed the personality profiles and development activity of about 400 GitHub developers. They found that developers with different levels of contributions have different personality profiles, specifically those with high or low levels of contributions are more neurotic compared to the others. Besides, the personality profiles of most active contributors were found to change across two consecutive years, evolving as more conscientious, more extrovert, and less agreeable.

Calefato et al. [87] and Calefato & Lanubile [86] investigated the relationship between project success and propensity to trust, a facet of the Agreeableness trait in the FFM. To avoid subjectivity in the assessment of project success, they approximated the overall performance of two Apache projects with the history of successful collaborations, i.e., code reviews of pull requests in GitHub. They found preliminary evidence that the propensity to trust of code reviewers (integrators) is an antecedent of pull request integration. They used the previous, LIWC-based version of IBM Personality Insight tool to analyze word

Table 3
Studies on Big Five personality in software engineering using tools for automatic personality recognition.

Reference	Focus	Unit of analysis	Context	Dataset	Tool	Findings
Rigby & Hassan (2007) [84]	Profiling	Individual	4 Apache http developers	~104 K emails from httpd-dev mailing list (1995–2005)	LIW 2007(?)	2 out of 4 top developers responsible of 2005 releases show similar personality profiles, different than overall baseline
Bazelli et al. (2013) [85]	Profiling	Individual	~850K(?) Stack Overflow users	Q&As from Stack Overflow (Aug. 2008 - Aug. 2012)	LIWC 2007(?)	Top reputed users more EXT than others
Rastogi & Nagappan (2016) [16]	Profiling	Individual	423 GitHub developers	Issue, pull-request, and commit comments from selected developers	LIWC 2007	Personality traits of most active developers are different from others, show changes over two consecutive years
Calefato & Lanubile (2017) [86]	Code review	Individual	6 Apache Groovy, 22 Apache Drill core team developers	~5k emails from groovy-dev mailing list (Jan. 2015 – Dec. 2016), ~30 K emails from drill-dev mailing list (Jan. 2012 – Dec. 2016)	IBM PI	Propensity to trust (a facet of AGR) of pull-request reviewers positively associated with the likelihood of code contribution acceptance

usage in pull request comments and automatically extract developers' agreeableness scores.

3. Research questions

The review of prior work on personality revealed several potential factors related to developers' activity and social status, which may affect the automatic detection of personality from the traces left in projects' communication channels and source code repositories. Therefore, to further our understanding of developers' personality profiles, we focus on studying their activities in both the technical part (i.e., code development through commits) and the social part (i.e., communication through emails) of the ASF ecosystem. Building on findings from prior work, in the following we formulate six research questions. Note that RQ2-5 are carried over from the original version of the study reported in [88].

The review of the software engineering Big Five personality studies using questionnaires (see Table 2 in Section 2.2.1) shows that most prior work (5 out of 11 studies) has focused on profiling software developers. Interestingly, all studies have used the IPIP instrument. A similar picture emerges from the analysis of prior work that has relied on tools for the automatic detection of personality from text (see Table 3 in Section 2.2.2), with 3 out of 4 studies relying on the LIWC software. Still, the synthesis of the findings is difficult, thus suggesting that profiling developers' personalities may depend on the context of the analysis. As such, we perform a large-scale analysis to detect developers' profiles within the entire ASF ecosystem, while also seeking subgroups of individuals with similar traits. We ask:

RQ1 — Are there groupings of similar developers according to their personality profile?

OSS project teams consist of different types of contributors, typically organized in a layered structure known as the *onion model* [89]. At the center of this organizational structure are *core contributors*, who are part of the development team and contribute the largest portion of the code base; they also review external code contributions and guide newcomers. *Peripheral contributors*, instead, are not part of the core development team and most of them do not remain involved with the project for long; they are typically involved with contributing bug fixes, adding projects documentation, and code refactoring. According to the findings reported by Rigby & Hassan [84] and Bazelli et al. [85], the personality of top-reputed users in software communities is different from the others. In our experimental scenario, this would suggest potential differences in the personality traits between peripheral and core Apache developers. On this basis, we ask:

RQ2 — Do developers' personality traits vary with the type of contributors (i.e., core vs. peripheral?)

According to the onion model, developers migrate from the edges to the core of OSS projects through a gradual socialization process. These changes in personality observed by Rastogi & Nagappan [16] may be due to the different type of tasks that developers perform and their responsibilities in the community. Therefore, we derive and compare the personality of developers, splitting the corpus of emails before and after they gain write-access to the source code repository (i.e., they become integrators who can accept and merge others' contributions), a sign that they were promoted to the core development team. We ask:

RQ3 — Do developers' personality traits change after becoming a core member of a project development team?

According to Rastogi & Nagappan [16], the personality of developers varies with their degree of code contributions, too. We seek confirming evidence for this finding. We ask:

RQ4 — Do developers' personality traits vary with the degree of development activity?

Calefato et al. [87] and Calefato & Lanubile [86] found initial evidence that the propensity to trust – i.e., the facet of Agreeableness representing the individual disposition to perceive the others as trustworthy – is positively correlated with the chances of successfully accepting contributions in code review tasks. Yet, trust is one the many facets in the Big Five model and previous research did not look at the effects of the personality of developers who author those contributions. Here, we bridge this gap and ask:

RQ5 — What personality traits are associated with the likelihood of becoming a project contributor?

In the onion model of participation in OSS projects, there are also *one-time contributors* (OTCs) who are on the very fringe of the peripheral developers since they have exactly one code contribution accepted to the project repository. The previous two research questions do not consider the number of code commits submitted by those who become contributors, nor possible correlations between development productivity and specific personality traits. Here, as a refinement of RQ4-5, we study whether the personality traits of ASF developers are associated with prolific development activity.

RQ6 — What personality traits are associated with higher amounts of contributions successfully accepted in a project repository?

4. Empirical study

In the following, we first describe the workflow designed for building the experimental dataset; then, we detail the statistical methods chosen to answer each research question.

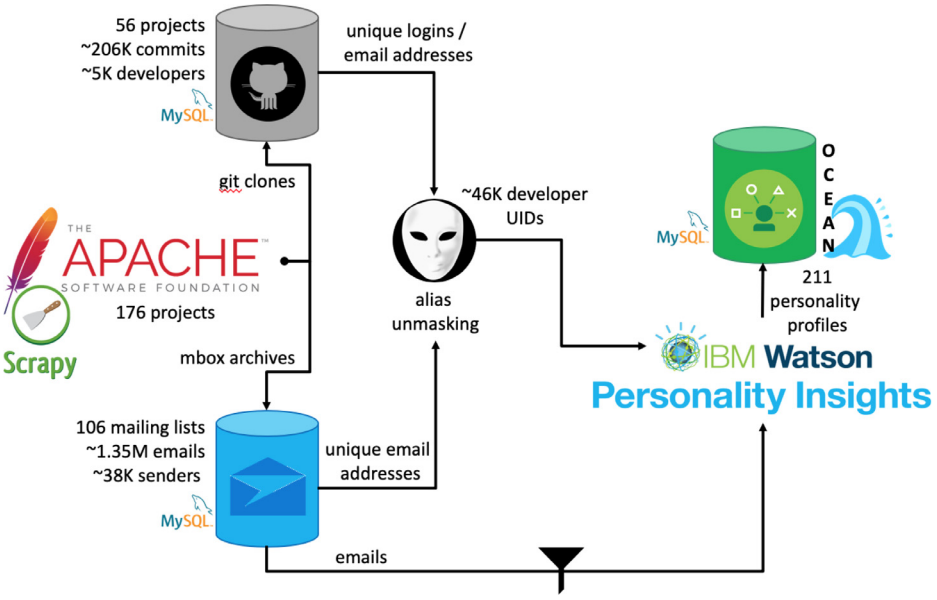

Fig. 2. The workflow designed for building the experimental dataset.

Table 4
The data sources used in our study.

Data source	Data extracted
Web pages	Project name Status (active, incubating, retired) Dev. language Category Repository URI (git, svn) Mailing-list URIs (dev, user) Mailing list name
Email archives	emails (body, subject, sender, recipient, timestamp) Developers' email addresses
GitHub	Repository (id, #commits, timestamp first and last commit) Developer's info (id, email, location) Commit metadata (repository, sha, author id, commiter id, timestamp, commit message, files changed, src files changed, #additions/deletions)

4.1. Dataset

To build our experimental dataset, we mined several data sources. The full list of the metadata extracted from each data source is reported in Table 4. Also, the scripts developed for mining the data source, along with the extracted data, are made available on GitHub[8] for the sake of replicability. The entire workflow for building the dataset is depicted in Fig. 2.

4.1.1. Retrieving projects

The first data source is the *official web pages* of the ASF projects.[9] The list of projects was obtained by developing a custom web scraper, using the Python Scrapy[10] library. Some project metadata were also extracted through the scraper, namely the status of the project (i.e., *active, retired, incubating*), its development language (e.g., *Java, C++*) and category (e.g., *database, web*), the mailing-list archive URIs, and the URI of its code repository. At the end of this stage, a list of 176 ASF projects was retrieved.

4.1.2. Downloading email archives

The second data source is the *mailing list archives*. Through the scraper, we retrieved for each project the URIs of the *dev* mailing list (i.e., containing development-oriented discussion such as bug reports) and *user* mailing list (i.e., containing general purpose discussion such as release announcements) archived in the *mbox* format. Then, we forked, updated, and ran the *mlstats*[11] tool to download the mailing lists to a local MySQL database. At the end of this step, 106 mailing lists were entirely downloaded, for a total of 1.35M emails from ~38,000 senders. The preprocessing and filtering process partially followed the steps described in the work by Shen et al.[17], where the personality of 28 users were automatically detected from a corpus of ~50,000 emails. Specifically, we developed *ad hoc* regular expressions to remove line by line the text (typically starting with '>') copied from previous emails in case of replies or forwards. Then, because the emails contained many lines of codes, we first tried to remove them with further regular expressions. However, the solution did not scale well, due to the variety of programming languages used in the ASF projects. Thus, we resolved on using machine learning. In particular, we used NLoN,[12] an R package that processes text and marks lines containing code [90]. We first used the package out of the box, because its default model has been trained on a corpus including emails from the Mozilla project archives. However, the performance was not satisfying. Then, the first author and a graduate student manually annotated a gold standard to retrain the model. They started with 500 emails, which resulted in an accuracy of about 90%, and then increased the training set up to a 1,000, which ensured accuracy of over 95%.

4.1.3. Cloning git repositories

The third and last data source is the *project code repositories*. We downloaded to a local machine a clone of the repository for each ASF project using Git. The other projects were discarded. Then, a Python script was written to parse the commit history of each project clone and save to the MySQL database the relevant metadata extracted, such as the IDs of the author and of the integrator, the time stamps, the list of files changed, the number of additions and deletions, etc. (refer to Table 4 for the full list). The number of commits is used as a proxy for project size; likewise, the delta in the years between the first and the last commit is used as a proxy for its longevity. At the end of this step, we selected and

[8] https://github.com/collab-uniba/personality.
[9] https://projects.apache.org/projects.html.
[10] https://scrapy.org.
[11] https://github.com/MetricsGrimoire/MailingListStats.
[12] https://github.com/M3SOulu/NLoN.

cloned the Git repositories of 56 ASF projects, totaling ~206 K commits made by 5080 distinct developers.

4.1.4. Unmasking developer aliases

Looking at the extracted data, we observed that, in many cases, the same sender used multiple email addresses to post messages to project mailing lists. This aliasing issue affected not only the communication but also the project development, as developers often commit code contributions using different email addresses. Therefore, we applied a procedure used in Vasilescu et al. [91] to 'unmask' alias email addresses. First, for each developer/sender stored in our database, an alias set was computed and assigned a unique identifier (UID in the following). Then, we stored a hash map of these UIDs so that, whenever a database entry was processed, the map was used to replace its table ID with the associated unique UID. The map contains the UIDs of 46,304 unique developers who either sent emails or contributed code to the AFS projects. No obvious cases of mislabeling were detected during the manual verification of the unmasking procedure performed on of a significant sub-sample.

4.1.5. Detecting personality

As the final step, we built the experimental dataset by collecting the Big Five scores for each unique developer, using the IBM Personality Insights service.

Personality Insights provides an application programming interface (API) for inferring individuals' intrinsic personality characteristics from digital communications such as email, text messages, tweets, and forum posts. As described earlier, we used the most recent version of the service, which extracts personality characteristics from text by using an open-vocabulary approach (like those proposed in [66,92,93]), which does not limit findings to preconceived relationships between *a priori* fixed sets of words and categories, as done in the closed-vocabulary approach of LIWC [64]. In more detail, the service first tokenizes the input text to develop a representation in an *n*-dimensional space, using an open-source word-embedding technique to obtain a vector representation of the words [94]; then, it feeds these representations to a machine-learning algorithm that infers a personality profile with Big Five characteristics.

Provided with sufficient textual input, the Personality Insights service API returns a JSON document with values in [0, 1] for each of the five personality traits of the writer. As per official documentation,[13] providing fewer than 100 words throws an exception of insufficient input. The precision of the service levels off at around 3000 words. Also, the upper limit is 6000 words, and longer input is truncated. Given these specifications, we developed a Python script that, to ensure sufficient input, retrieves and collates per month all the emails sent to an Apache project mailing lists by each unique developer. To make the script more robust, even if the collated text for a month accounts for fewer than 100 words (remember the NLoN filter used to remove lines of code from emails), it still invokes the service and handles the exception (i.e., skip the month), thus accommodating potential changes to the limits in future releases of the service. Finally, for each developer the Big Five personality profiles are computed as an average of the monthly-based trait scores.

Overall, we extracted the personality profiles of 211 unique developers, of whom 118 contributed both source code changes and emails to the same project, and 93 only sent emails. Project committers who did not participate in discussions over emails, those who made changes exclusively to non-source code resources (e.g., documentation and binary files), and those who contributed overall fewer than 100 words in all their emails were excluded. Each of the 211 developers on average participated in 2 projects, sent over 6900 emails, writing about 15,000 words.

[13] https://console.bluemix.net/docs/services/personality-insights/input.html#sufficient.

4.2. Analysis

We perform several statistical analyses using R version 3.5.2. However, before seeking answers to the research questions defined earlier, we first analyze stability of the automatic personality detection instrument. This preliminary assessment is necessary to ensure that we can safely average the monthly scores into one aggregate personality profile for each developer. Rastogi & Nagappan [16] used LIWC and found that developers' personality profiles extracted from GitHub content change over short-time spans. However, psychology research considers personality traits as rather stable, particularly for working adults [95]. Hence, we perform a Wilcoxon Signed-Rank test to check the stability over time of developers' personality profiles extracted using IBM Personality Insights between the first and second halves of their activity history.

To answer RQ1 (groupings of developers with similar personality), we apply several statistics to reveal the presence of latent structures within our data. First, we run a *Principal Component Analysis* to identify which of the traits may weigh more in differentiating developers' personalities. Then, we execute *Cluster Analysis* on our multivariate dataset to identify homogeneous, mutually exclusive subsets and reveal natural groupings of developers resembling each other while also being different from the others. A similar analysis has been reported in [5,8]. In addition to Cluster Analysis, we also perform *Archetypal Analysis* [96], a statistical method that builds on the idea that any data point in a multidimensional space (i.e., each developer in our dataset), defined by a set of numerical variables (i.e., the vector of the Big Five trait scores), can be represented as a combination of specific points called *archetypes*. In other words, archetypal analysis can identify in our dataset a few archetypal personalities, which can then be used to describe all other developers in terms of the closeness to each archetype. A similar analysis has been applied by Kosti et al. [82].

For RQ2 (variation with project membership), we use the Wilcoxon Signed-Rank test as a non-parametric alternative to *t*-test for paired samples. For RQ3 (variation of personality with the type of contributor) and RQ4 (variation with the degree of development activity), we use the Wilcoxon Rank Sum (or Mann-Whitney U) test, as a non-parametric alternative to *t*-test for unpaired samples. For the analyses above, we use p-values with a significance level of α=0.05 to determine statistical significance. Also, we report p-values adjusted with Bonferroni correction to counteract the problem of inflated type I errors while engaging in multiple pairwise comparisons between subgroups. In case of significant differences, we complement p-values with appropriate effect size measures to quantify the amount of difference between two groups of observations.

For RQ5 (contribution likelihood model), we fit a logistic regression model to our data to assess the likelihood for a developer to become a project contributor, using personality scores as predictive factors. The variables included in the model are detailed below.

Response: `contributor`, a dichotomous yes/no variable indicating whether a developer has authored at least one commit successfully integrated into a project repository.

Main predictors. We include `openness`, `agreeableness`, `neuroticism`, `extraversion`, and `conscientiousness`, that is, one predictor for each of the Big Five personality trait scores.

Controls. Our control variables include `word_count`, a proxy for the extent of communication and social activity of the developer in the community through email messages from which personality traits are extracted, `project_size`, computed as the total number of commits in the projects, and `project_age`, measured in number of years.

The two variables `project_size` and `project_age` are intended to reflect that it may be harder for developers to start contributing to long-running projects that have a large code base. However, because they are highly correlated (Pearson r=0.74) and we only retain `project_age`. Also, the Variance Inflation Factor (VIF) computed on the resulting model reveals no collinearity issues for the predictors (all values <4).

Procedure. We fit the model using the *glm* function in R. Coefficients are considered important when statistically significant at 5% level ($p<0.05$). We evaluated the model fit using McFadded's pseudo-R^2 measure, which describes the proportion of variance in the response variable explained by the model, and AUC, to assess the classification ability of our model compared to random guessing.

Finally, to answer RQ6 (prolific activity model), we perform a regression analysis to evaluate the association between the personality traits of developers and the number of contributions (i.e., pull requests) that they got accepted (i.e., merged) into the Apache projects' repositories.

Response. The dependent variable is `#merged_commits`, which counts the number of commits authored by a developer that have been successfully merged.

Main predictors. We use the same predictors as in the case of the previous research question, i.e., one predictor for each of the Big Five traits.

Controls. We use the same control variables retained as in the case of the previous research question, namely `word_count`, `project_age`, and `project_size`. We know already from RQ6 that `project_age` and `project_size` are highly correlated. Accordingly, we retain the former because it ensures a slightly better fit for the resulting model. Moreover, in this case, we also find a slightly positive correlation between `conscientiousness` and `extraversion`. However, we opt for retaining them because the VIF computed on the fit model shows a value smaller than 4 for both, as well as for the other independent variables.

Procedure. As described above, the dependent variable `#merged_commits` is the count of successfully merged contributions to the source code; therefore it takes non-negative integer values only. Hence, rather than fitting a linear model, we perform a count data regression analysis, which can handle non-negative observations, given that we are intentionally studying the profiles of developers who have had contributions to source code accepted.

There are different count data models that can be used for estimations, whose choice depends on the characteristics of the data. We follow the approach suggested by Greene [97]. The starting point is to consider the Poisson regression model. However, the Poisson distribution has a strong assumption on equidispersion, that is, the equality of mean and variance of the count-dependent variable. If the assumption is rejected, count data can be modeled using the negative binomial distribution, a generalization of the Poisson distribution with an additional parameter to accommodate the overdispersion. Finally, a formal Likelihood Ratio Test (LRT) of overdispersion is executed to ensure that the negative binomial model provides a better fit to the data than the Poisson model, that is, the null hypothesis of equidispersion (Poisson model) against the alternative of overdispersion (negative binomial model) is tested.

5. Results

5.1. Preliminary assessment of stability

To rule out changes in personality over time, we split the dataset by date into two sections. Specifically, for each of the N=211 developers, we assess the time-span between the first and last communication in the dataset; then, we compute the point in time M_T so that approximately half of the observations (i.e., the monthly-based personality scores) are located *before* and *after* it. Then, two aggregate profiles for each developer are created by averaging the trait scores. Finally, for each trait, we perform a Wilcoxon Signed-Rank test to verify the null hypothesis that the median difference between pairs of observations (i.e., for each subject) is not significantly different from zero. Table 5 reports the results from the five paired tests, which show no significant differences between the distributions (all adjusted p-values > 0.05 after Bonferroni correction for multiple tests), thus confirming the stability of personality traits over time.

Table 5
Results of the Wilcoxon Signed-Rank tests for assessing changes in mean personality traits over time (N=211, all p-values >0.05 after Bonferroni correction).

Trait	V	p-value	CI 95%
Openness	6109	0.589	−0.002–0.003
Conscientiousness	5575	0.661	−0.004–0.003
Extraversion	5839	0.964	−0.003–0.003
Agreeableness	5871	0.917	−0.003–0.003
Neuroticism	5915	0.853	−0.003–0.004

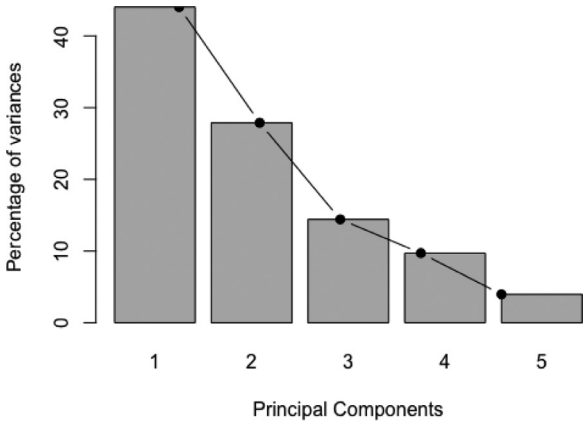

Fig. 3. Percent of variance explained by principal components.

5.2. RQ1 — Personality groupings

Here we report on the results from several techniques used to reveal the presence of natural groupings of personalities within our dataset.

First, we check and find that the distributions of each trait scores do not follow normal distribution (all p-values <0.01).[14] Accordingly, in the following, we use non-parametric statistics, which do not assume normality in the distribution of data. Then, we check for the presence of correlation between trait scores. We use the scale suggested by Hinkle et al. [98] for studies in behavioral sciences. We observe only a moderate positive Pearson correlation between Conscientiousness and Neuroticism (r=0.58). The others are negligible (r<0.3) or low (between 0.3 and 0.4). Finally, we perform a couple of tests to assess the suitability of our data for structure detection. To ensure that there is a sufficient proportion of variance in our variables that might be caused by underlying factors, we first compute the Kaiser-Meyer-Olkin measure, which is equal to 0.5, that is, the minimum acceptable value as suggested in [99]; then, we perform Barlett's test of sphericity, which is significant (χ^2=4088.32, p<0.001). These results suggest that our data is suitable for structure detection.

Principal Component Analysis. We perform Principal Component Analysis (PCA) with varimax rotation, using the *FactoMineR* package. PCA is a statistical procedure that converts a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables, i.e., the principal components. The scree plot in Fig. 3 shows that the first three components out of the five extracted account for most of the variance in the data (86%). However, the analysis of the eigenvalues in Table 6 shows that only the first two have a value over Kaiser's criterion of 1, the cut-off point typically used to retain principal components. Eigenvalues, in fact, correspond to the amount of the variation explained by each principal component. A component with an eigenvalue > 1 indicates that it accounts for more variance than its accounted by one of the original variables in the dataset.

[14] We checked and obtained the same results with both the Pearson χ^2 test of correlation and the Shapiro-Wilk test.

Table 6
Eigenvalues returned by the PCA (only components with eigenvalue >1 are retained).

	Eigenvalue	% of variance
Component 1	2.201	44.023
Component 2	1.394	27.885
Component 3	0.721	14.419
Component 4	0.485	9.705
Component 5	0.198	3.967

Table 7
Standardized loadings for the extracted principal components.

	Component 1	Component 2
Openness	0.79	0.03
Conscientiousness	0.69	0.44
Extraversion	0.27	0.74
Agreeableness	-0.15	0.92
Neuroticism	0.89	0.04

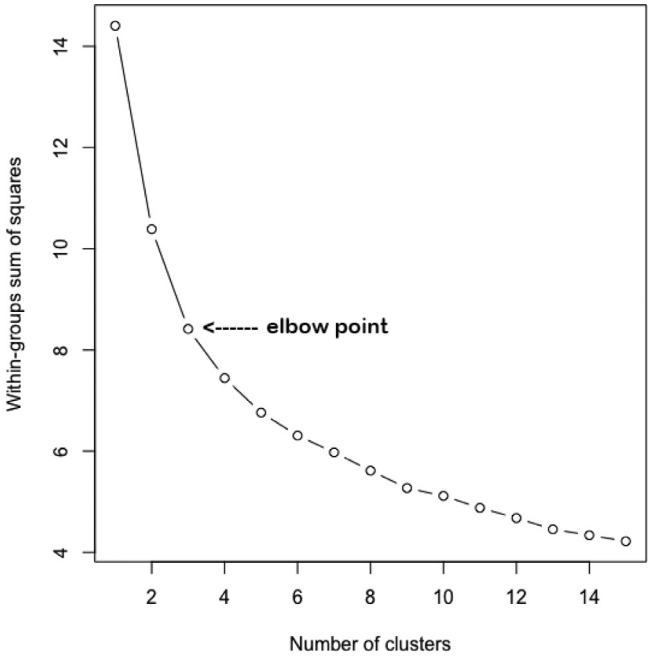

Fig. 4. Plot of within-group heterogeneity against the number of k-means clusters.

Accordingly, we retain the first two components, which account for 72% of the variance. Openness and Neuroticism load on the first component, whereas Conscientiousness, Extraversion, and Agreeableness on the second (see Table 7). The two most strongly-loaded factor for each of the two components are, respectively, Neuroticism (0.89) and Agreeableness (0.92).

Cluster Analysis. Following the approaches presented in [5,8] for extracting clusters of developers' personalities, we apply the *k-means* clustering algorithm using the *stats* package. We use the 'elbow' method to identify the optimal number of cluster from the plot in Fig. 4. The 'elbow' point corresponds to the smallest *k* value (3 in our case) after which we do not observe a large decrease in the within-group heterogeneity, here measured using the sum of squares, with the increase of the number of clusters.

The developers are fairly evenly distributed across the three personality clusters extracted (see Table 8). The table also reports the coordinates of the *centroids*, that is the average position of the elements assigned to a cluster. All the values are *z*-score standardized, with positive (negative) values above (below) the overall means.

Table 8
Size and centers of the three clusters extracted with k-means (N=211, the highest ▲ and lowest ▼ values per trait shown in **bold**).

Cluster (size)	Openness	Conscient.	Extraver.	Agreeabl.	Neurotic.
Cluster 1 (76)	**−0.74▼**	**−0.69▼**	-0.06	0.37	**−0.84▼**
Cluster 2 (55)	**0.90▲**	**0.86▲**	**0.99▲**	**0.45▲**	**0.81▲**
Cluster 3 (80)	0.08	0.07	**−0.62▼**	**−0.67▼**	0.25

Table 9
Results of the Kruskall-Wallis tests for the comparisons of the distributions of each personality trait scores across the three clusters (p-values adjusted with Bonferroni correction).

Trait	Chi-squared	df	p-value	ϵ-squared	CI 95%
Openness	87.836	2	<0.001	0.418	0.297–0.532
Conscientiousness	78.777	2	<0.001	0.375	0.257–0.495
Extraversion	94.554	2	<0.001	0.450	0.354–0.547
Agreeableness	61.248	2	<0.001	0.292	0.197–0.401
Neuroticism	107.560	2	<0.001	0.512	0.410–0.613

Table 10
The three archetypes extracted (the highest ▲ and lowest ▼ standardized values per trait are shown in bold).

Archetype	Openness	Conscient.	Extraver.	Agreeabl.	Neuroticism
Archetype 1	0.51	−0.13	**−0.81▼**	**−1.09▼**	**0.61▲**
Archetype 2	**0.64▲**	**1.06▲**	**1.12▲**	**0.87▲**	0.54
Archetype 3	**−1.15▼**	**−0.93 ▼**	−0.31	0.23	**−1.15▼**

Because the data are not normally distributed, we use the *stats* package to perform five nonparametric Kruskal-Wallis tests to make unpaired comparisons among the three independent score distributions (i.e., the clusters) for each of the five traits. Table 9 shows the results of the Kruskal-Wallis tests, after applying Bonferroni corrections of p-values for repeated tests. Because each p-value is smaller than 0.001 and the ϵ-squared statistic shows a large effect size (≥ 0.26), we conclude that there are significant differences among the distributions of the traits; therefore, they all are important to the formation of the three clusters. To further understand which pairs of clusters are significantly different, we perform the Tukey and Kramer (Nemenyi) *post hoc* test for multiple pairwise comparisons. All the comparisons show significant differences with p-values smaller than 0.001 or 0.01. The only exception is the non-significant difference observed for Agreeableness between Cluster 1 and 2 ($p = 0.99$).

Finally, by comparing the traits values across the threes cluster, we can label Cluster 1 as the subgroup of the '*calm, cautious, and easy-going*' developers who are low in Neuroticism, Openness, and Conscientiousness. Cluster 2 is the subgroup of developers with an '*intense*' personality, given that they exhibit the highest average scores for all the five traits (see the values in bold in Table 8). Regarding Cluster 3, it groups the '*antagonistic introvert*' with low average scores in Extraversion and Agreeableness.

Archetypal analysis. Following the approach presented in [82], we perform Archetypal Analysis using the package *archetype*. We use the 'elbow' criterion again to identify the optimal number of archetypes to extract. From the scree plot in Fig. 5, which shows the fraction of total variance in the data explained by the number of extracted archetypes, we notice that the function plateaus after extracting 3 or 5 archetypes. For the sake of simplicity in characterizing the archetypes, we opt for extracting 3.

Table 10 shows the trait coordinates for each of the three archetypes, standardized for the ease of comparison. We compare the trait values across the three archetypes and obtain results in line with the findings from k-means. In fact, the extracted archetypes can be mapped

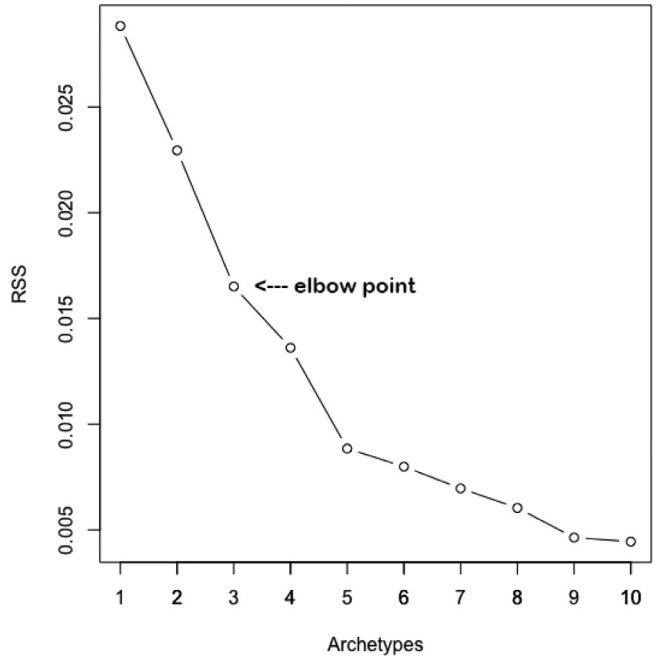

Fig. 5. Scree plot of the residual sum of squares against the number of archetypes.

Table 11
Results of the Wilcoxon Rank Sum tests for the unpaired comparison of median personality trait scores between N=56 core and N=62 peripheral developers (all p-values>0.05 after Bonferroni correction).

Trait	W	p-value	CI 95%
Openness	1583	1.000	−0.009–0.008
Conscientiousness	1625	1.000	−0.010–0.011
Extraversion	1575	1.000	−0.010–0.008
Agreeableness	1273	0.271	−0.017–0.000
Neuroticism	2051	0.063	0.004–0.027

on the three clusters described above. Specifically, the Archetype 2 is similar to Cluster 2 as it models the '*intense*' type of developers (i.e., with high scores on 4 out of 5 traits). The Archetype 1 represents the '*antagonistic introvert*,' as in the case of Cluster 3, who score low Extraversion and Agreeableness. Finally, the Archetype 3 is that of the '*calm, cautious, and easy-going*' developers grouped in Cluster 1, with low scores in Neuroticism, Openness, and Conscientiousness.

5.3. RQ2 — Variation with contributor type

We separate the personality scores of N=118 commit authors in two groups, namely *peripherals* (i.e., those without commit access to the repositories, N=62) and *core* developers (i.e., project members with write access to the source code repository, N=56). For the sake of space, here we omit to report the boxplots. Results of the Wilcoxon Rank Sum tests for unpaired groups comparisons are reported in Table 11, which show no significant differences for any of the five traits (i.e., all adjusted p-values > 0.05, after Bonferroni correction).

5.4. RQ3 — Variation with membership

For each the 56 core developers with write access to source code repositories, we first retrieve the date of the first commit that they review and accept to integrate. We use this date as an approximation of the moment when the developers have become core team members of a project. Then, for any of the projects they gained membership for, we use that date to split the personality trait scores of the developers into two paired groups, i.e., *before* and *after* becoming a project's core team member. Note that in this case we have multiple observations per developer, that is, one for each project of which they are core a member. Fig. 6 shows the differences in the five personality scores between the two groups and Table 12 reports the results of the five Wilcoxon Signed-Rank tests (one per trait). No significant differences are retuned by the tests (all adjusted p-values > 0.05 after Bonferroni correction).

Table 12
Results of the Wilcoxon Signed-Rank tests for the paired comparison of mean personality trait scores of developers before and after becoming members of a project's core-team (N=192, all p-values >0.05 after Bonferroni correction).

Trait	V	p-value	CI 95%
Openness	39	1.000	−0.011 – 0.034
Conscientiousness	40	1.000	−0.008 – 0.031
Extraversion	17	1.000	−0.019 – 0.019
Agreeableness	15	1.000	−0.038 – 0.011
Neuroticism	43	0.654	−0.005 – 0.048

Table 13
Results of the Wilcoxon Rank Sum test for the unpaired comparison of median personality trait scores between developers with high vs. low degree of development activity (adjusted p-values > 0.05 after Bonferroni correction).

	Trait	W	p-value	CI 95%
High vs. low commit authors (peripheral devs)	Openness	476	1.000	−0.004 – 0.021
	Conscientiousness	449	1.000	−0.008 – 0.024
	Extraversion	383	1.000	−0.017 – 0.017
	Agreeableness	341	1.000	−0.018 – 0.009
	Neuroticism	408	1.000	−0.013 – 0.017
High vs. low commit integrators (core devs)	Openness	193	1.000	−0.014 – 0.020
	Conscientiousness	163	1.000	−0.029 – 0.019
	Extraversion	129	1.000	−0.028 – 0.006
	Agreeableness	204	1.000	−0.013 – 0.025
	Neuroticism	151	1.000	−0.040 – 0.017

5.5. RQ4 — Variation with the degree of development activity

We take the core (N=56) and peripheral (N=62) groups created for RQ2, and further split them according to the level of development activity. The level of development activity varies depending on whether they are core or peripheral developers. Hence, we find the mean number of commits authored by developers in the peripheral group and split it into two subsets, *authored-commits_high* (N=17) and *authored-commits_low* (N=45). Similarly, we obtain the subgroups *integrated-commits_high* (N=44) and *integrated-commits_low* (N=8) considering the mean number of *commits integrated* (i.e., accepted) by the core group members. We then perform the unpaired comparisons of the median personality scores between high vs. low-activity developers. Results are in shown in Table 13. The Wilcoxon Rank Sum tests reveal no cases of statistically significant differences between the pairs of trait distributions (i.e., adjusted p-values > 0.05 after Bonferroni correction).

5.6. RQ5 — Contribution likelihood model

In Table 14, we report the results of the logistic model, obtained from the *glm* function of the *stats* package, to study the associations between the personality traits of developers and the likelihood of becoming a project contributor. Therefore, the dependent, dichotomous variable is whether a developer has made a commit to any Apache project. The number of participants involved in this analysis is N=211, where 118 are the developers with at least one commit, and 93 are those with no commits. Because the subset is reasonably balanced, there is no need to deal with the class imbalance problem [100].

We observe that the control variable `project_age` is statistically significant (coeff=-0.42, p<0.001). The only statistically significant

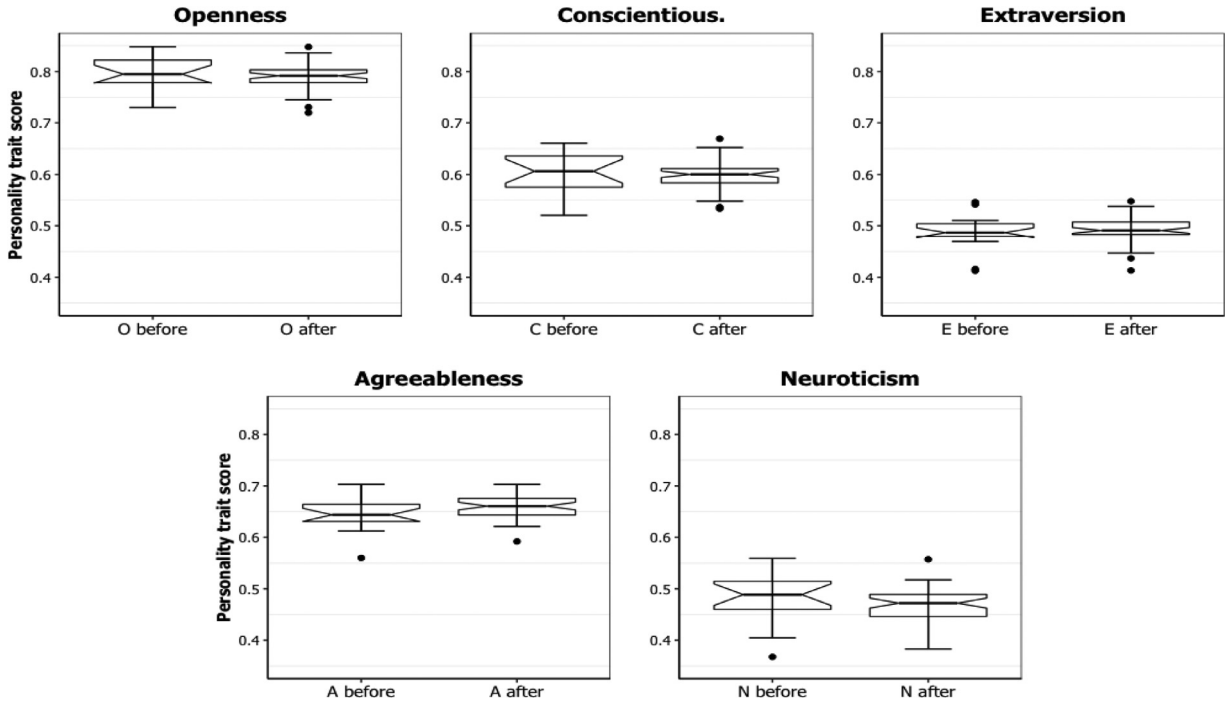

Fig. 6. Differences in the personality traits of the developers before and after becoming core team members.

Table 14
Logistic regression model of the contribution likelihood as explained by personality traits (sig: '***' p<0.001, '**' p<0.01).

	Coef. Estimate	Std. Error	z-value
(*Intercept*)	−29.523	20.175	−1.44
project_age	**−0.420*****	0.113	−3.71
log(word_count)	0.199	0.204	0.98
openness	**54.092****	23.338	2.32
conscientiousness	−18.994	26.623	−0.71
extraversion	−4.652	16.939	−0.27
agreeableness	18.620	22.525	0.83
neuroticism	-19.710	16.939	−1.07

N=211, McFadden Pseudo-R^2=0.397,
AUC=0.89

Table 15
Developers' productivity negative binomial model. The response is the count of commits successfully merged (sig: '***' p<0.001, '**' p<0.01).

	Coef. Estimate	Std. Error	z value
(*Intercept*)	0.807	0.234	3.43
project_age (days)	−0.068	0.044	−1.56
dev_is_integrator=TRUE	**0.648****	0.221	2.93
dev_track_record (days)	**0.544*****	0.033	16.21
log(word_count)	0.003	0.030	0.12
openness	0.036	0.068	0.53
conscientiousness	0.005	0.072	0.08
extraversion	0.046	0.066	0.71
agreeableness	−0.039	0.054	−1.80
neuroticism	0.141	0.078	−1.80

N=471, LogLik=-917, LRT χ^2=514
McFadden Pseudo-R^2=0.115

predictors is openness (coeff=54.09, p<0.01). The significance of the terms is obtained from the Wald test in the *ANOVA*, as implemented in the *car* package.

To evaluate the goodness of fit, we compute McFadden's pseudo-R^2, a statistical measure that represents the percentage of the response variable variation that is explained by the model. The results show that our model it is capable of explaining about 40% of the variability (R^2=0.397).

Furthermore, we measure the performance of the model using the Area Under the ROC curve (AUC). A ROC curve plots the performance of a binary prediction model as the trade-off between its ability to recall the positive instance of the dataset (i.e., the true positive rate, or how many developers predicted as becoming contributors have actually had commits successfully merged) and the false positive rate (i.e., how many developers predicted to become contributors are misclassified). We split the dataset into training (70%) and test (30%) sets, using the stratified sampling function offered by the caret package to maintain the same proportion of dependent variable occurrences across them. The AUC performance of our logistic model is 0.89. Because the AUC performance of a random baseline classifier is 0.5, we conclude that the model performs largely better than random guessing.

The results above tell us that higher openness traits scores are associated with higher chances for developers to become project contributors. To provide a more quantitative interpretation, we note that the mean openness value in the dataset is 0.79. Given the logistic model in Table 14, the probability of becoming a contributor for those developers with openness scores below averages is 64%, compared to 87% for developers with scores equal to or above averages (+36%).

5.7. RQ6 — Merged commits count data model

Table 15 shows the results of the count data regression in which the number of successfully merged code commits is measured with the personality traits variables. The number of developers with at least one commit is 118, who have made contributions to about 2 projects on average. The sample used in this analysis contains N=471 observations (commits data).

Before reporting the regression results, we briefly comment on the model choice. First, the likelihood ratio test (LRT) of overdispersion shows a test statistic (χ^2=514, p<0.001) that leads to reject the null hypothesis of equidispersion and, therefore, the negative binomial model (LogLik=−917) is preferred to the Poisson model (LogLik=−1174).

Accordingly, in Table 15 we report the results only for the negative binomial model.

To ease the assessment of the relative importance of the continuous predictors, we z-transform them so that the mean of each measure is 0 and the standard deviation is 1.

We observe that none of the five predictors related to personality has a significant effect. Instead, regarding the control variables, we observe that the authors' track record (i.e., the number of days between their first and last successful contribution) has a positive and significant association (coefficient=0.544) with the number of their merged contributions ($p<0.001$). Similarly, we find a positive and significant association between the response variable and the fact that a developer is a core team member who has integrated external contributions (coefficient=0.648, $p<0.01$). However, the model fits the data marginally (Pseudo-R^2=0.115).

6. Discussion

The results reported in the previous section add to the body of existing evidence about mining the personality traits of developers from software-related repositories.

6.1. Ecological validity of digital cues from emails

In this study, trait observations have been averaged by month, resulting in one aggregate personality profile for each developer. Albeit personality is considered stable [95], especially in working adult, depending on how it is measured and aggregated (e.g., days vs. weeks), personality can also be observed as variable [101]. Thus, because of the large time scale of data analyzed in our study – with email archives spanning ~15 years – we deemed necessary to confirm the ecological validity of the digital cues fed to personality tool by verifying the stability of traits over the years before carrying out further analyses. Our results (see Section 5.1) are in line with those from prior research in Psychology, which found personality to be stable, especially in working adults, over multiple years [95] and even decades [18].

On the contrary, Rastogi & Nagappan [16] found that GitHub developers' personality change over short periods (i.e., two or three consecutive years), evolving as more conscientious and extrovert, and less agreeable. While further investigations are needed to explain this difference, we note that Rastogi & Nagappan made these claims despite the negligible to small effect sizes calculated for their paired t-tests (i.e., Cliff's δ [102] values as low as 0.04).

6.2. Personality types (RQ1)

Regarding the first research questions (RQ1 — *Are there groupings of similar developers according to their personality profile?*), our results strengthen prior evidence that software developers differ significantly in their personality profiles.

First, we performed Principal Component Analysis, which helped us uncover that Neuroticism (i.e., emotional stability vs. lack thereof) and Agreeableness (i.e., being cooperative vs. antagonistic) are the two most important traits in differentiating developers by personality type. Previous research on OSS has found evidence that conversations over emails among developers often deteriorate into conflicts (or *flame wars*) [103,104]. Considering that personalities profiles here have been extracted from a corpus of emails, Agreeableness and Neuroticism levels may reflect developers' general behavior during discussions. In other words, developers high in Neuroticism may be those who tend to use negative polarity lexicon because they tend to get involved in such heated discussions, and vice versa for those high in Agreeableness. As future work, we will employ software engineering-specific sentiment analysis toolkits, such as EMTk [46,105,106], to analyze the extent and influence of such flaming behaviors on developers' lexicon.

We also used two different techniques, the k-means clustering algorithm and Archetypal Analysis, which gave us consistent results about the existence of three subgroups of personalities. We informally labeled these types as '*intense*', '*antagonistic introvert*', and "*calm, cautious, and easy-going.*" Similar research involving software engineering students [5] and professionals [8] found two types of personalities among, the *intense* and the *moderate*. Our findings may have further refined their results. However, it is arduous to claim that these are the main types existing among software engineers – prior work has found contrasting evidence as to whether software engineers represent a homogeneous group [107] – or among OSS developers – as the ASF ecosystem has a carefully-defined Code of Conduct[15] whose policies are likely to influence how developers behave over email [108].

Overall, our findings reinforce the need for future studies on human factors in software engineering to use psychometric tools to control for potential, personality-related confound factors [109,110].

6.3. Personality and context (RQ2, RQ3)

A recent trend in psychology [101] is that personality effects interact with the environment, i.e., individual personality has certain main effects that need to be seen as a contextualized behavior. In other words, researchers assume that there is variability in how different individuals respond to the same situation, whereas there is presumed to be be stability in how the same individuals behave across similar situations and variability across dissimilar situations.

To assess the interplay between context and personality, we first checked the interaction of personality with the type of contributor (RQ2 —*Do developers*" personality traits vary with the type of contributors, i.e., core vs. peripheral?)), given that core and peripheral developers have different tasks to perform and responsibilities to uphold. Our findings (see Section 5.3) show no significant differences between core and peripheral developers' personality traits. Then (see Section 5.4), we consistently found that the personality of developers does not change after becoming core project members (RQ3 — *Do developers' personality traits change after becoming a core member of a project development team?*).

Interestingly, our results contrast with the findings of Rigby & Hassan [84] and Bazelli et al. [85], who found that top developers have different personality traits from the others. However, Rigby & Hassan [84] analyzed data from four developers only. The contrast with Bazelli et al. [85], instead, is arguably explained by the different experimental domains. In fact, they analyzed posts and question-answering activity of developers within Stack Overflow, while we are looking at emails and source code development in the Apache ecosystem.

6.4. Personality and extent of contribution (RQ4, RQ5, RQ6)

With RQ4 (*Do developers' personality traits vary with the degree of development activity?*), we checked whether developers who contribute more source code changes exhibit different median trait scores compared to the others. We found no differences between developers when grouped by their level of activity. Instead, Rastogi & Nagappan [16] found that developers who contribute more score high on Openness, Conscientiousness, Extraversion, and Neuroticism, and low on Agreeableness. As in the case of RQ2-3, further investigations are needed to explain the contrasting results.

While the previous research question showed no differences in personlity between developers with different levels of activity, it did not allow us to uncover associations between personality traits and contributing source code changes. Accordingly, we performed statistical analysis using Generalized Linear Models (GLM) to establish associations of personality traits specifically with the likelihood of becoming a contributor (RQ5 — *What personality traits are associated*

[15] https://www.apache.org/foundation/policies/conduct.html.

with the likelihood of becoming a contributor?) and the number of accepted contributions (RQ6 — *What personality traits are associated with the number of code contributions successfully accepted in a project repository?*).

Regarding RQ5, the logistic model developed showed that, as expected, the control variable `project_age` has a significant negative effect on the chances of becoming an Apache project contributor (i.e., developers' onboarding is harder for projects with a long history and a large code base). Instead, the control variable `word_count` (i.e., the proxy for the amount of social activity in a project community) is not statistically significant. This means that the amount of communication that a developer exchanges in the ASF communities is not associated with the likelihood of becoming a contributor. However, previous research (e.g., [111]) has found that contributions coming from submitters who are known to the core development team have higher chances of being accepted. Combined, these findings indicate that the quality of the messages and their recipients are important to become a contributor, rather than the overall amount of communication exchanged.

Furthermore, the results of the logistic regression show that more open developers are more likely (+36%) to contribute commits that are successfully integrated into a project repository. This finding complements the results of our previous work [86,87], where we found that more agreeable integrators are more likely to accept the pull requests during code review sessions. Agreeableness, in fact, is associated with the propensity to trust other, being empathetic, and avoiding harsh confrontations — facets of personality that are 'helpful' during cooperative tasks such as code reviews, where more open/agreeable contributors and integrators are likely to collaborate with less friction. Previous research on OSS projects has highlighted that newcomers face several entry barriers, not only technical but also social, when placing their first contribution, leading in many cases to dropouts [112,113]. Hence, overall, our findings suggest that more open/agreeable core members may be better suited to shepherd newcomers during their immigration phase (i.e., on-boarding and first contributions) [114,115]. In previous work, Canfora et al. [116] successfully tested an approach to recommend the 'right mentors' among core team members to guide OSS project newcomers. Their recommendations were based on discovering previous interactions through emails on topics of shared interest. A recent trend is to use bots in collaborative development environments, such as GitHub, to automatically assign code-review tasks to those project members who have made the largest and most recent contributions to the changed files [117]. Our findings suggest that such bots could be augmented with psychometric capabilities so that they could automatically mine personality profiles from the developers communication traces left in the project repositories and recommend the 'best-fitting' reviewers both technically and socially (i.e., more open and agreeable). More in general, finding the 'right mix' of personalities has potential implications regarding team-building not only for OSS projects but also for commercial ones, especially if distributed. In previous research, Yang et al. [118] found that agreeableness helped teammates coordinate through the development of shared mental models, thereby enhancing software team performance. In a laboratory experiment, Karn et al. [119] found that software teams reported higher cohesion and performance in cases of both homogeneity in personality type and some mixtures of types.

As regards RQ6, the count-data model developed fits the data marginally (~0.11% of variability explained, see Table 15). Looking at the estimates, we note that none of the personality trait predictors is significant. The only significant predictors are the control variables `dev_track_record=TRUE` and `dev_is_integrator=TRUE`, which indicate that, respectively, long-time contributors and core-members who integrate external contributions are associated with higher numbers of accepted commits. Hence, there does not seem to be *one* personality type associated with higher productiveness. On the one hand, these results are not surprising; in fact, they are in line with the results of both RQ4 (i.e., no differences in mean personality traits score among developers when grouped by activity level) and prior work that uncovered the technical antecedents of accepted contributions in OSS projects (e.g., [120,121]). On the other hand, combined with the findings from our previous work on trust [86,87] and RQ5 (i.e., more open developers are more likely to contribute), these results suggest that personality may have an impact on development activities that entail direct communication with others, as in code review tasks. Still, given the marginal fit and the cross-sectional nature of the data fed into the regression models, here we can only hint at possible causal relations, which we reserve to investigate in future work.

6.5. Limitations

There are many open challenges for research to increase the validity of results.

Lack of gold standards. Because automatic personality recognition approaches are inherently data-driven, the availability of experimental datasets plays a crucial role. With the withdrawal of myPersonality,[16] only a few are available as of this writing, such as the Essay dataset [43], the EAR dataset [122], and the benchmarks used for the evaluation campaigns in the two editions of the *Workshop on Computational Personality Recognition* [63,70]. We believe that the collection and diffusion of standard benchmarks will help to improve both the validity and performance of tools by allowing more rigorous comparisons. In particular, to date, personality datasets from the software engineering (SE) domain are completely missing.

Trait rating accuracy. The present is one of the very few studies existing on personality computation in the SE domain. The results reported in these studies (see Section 2.2.2) obliviously depend on the accuracy of the automatically measured trait scores as compared to the actual personality of the subjects involved.

In this study, we relied on the IBM Personality Insights tool, which was trained using the Big Five personality scores from surveys conducted among thousands of volunteers who also shared their Twitter feed content in different languages (i.e., English, Spanish, Japanese, Korean, and Arabic). The language-specific models were developed independently of user demographics such as age, gender, or culture. To understand the accuracy of the service in inferring personality profiles, IBM conducted a validation study by collecting tweets from 1,500-2,000 participants who also took the 50-item IPIP test to establish ground truth. As reported earlier, the comparison[17] between the inferred and actual personality scores showed an average MAE ≈ 0.12 over the five traits and an average correlation $r \approx 0.33$ close to the upper limit of the correlation range between 0.1 and 0.4, suggested as practical benchmark in previous personality studies [66] and meta-analyses [72,73].

While individual self-ratings are typically used as gold standards to set ground truth, it must be pointed out that psychology research now considers the definition of a 'true' personality profile out of reach for both self and external raters [101]. Despite extensive evidence supporting their validity (see Section 2.1), self-assessment questionnaires are subject to ratings being biased towards social desirability, with individuals potentially projecting how they would like to be perceived rather than how they actually are [123]. Furthermore, previous research has shown that, albeit tendentially highly correlated, there are differences between personality constructs based on self-reports and those based on external observers' ratings [124].

Therefore, when evaluating the performance of automatic personality recognition tools, researchers must keep in mind that personality is an elusive concept whose assessment makes it an activity that is complex for any rater, whether self, external observer or computer.

Trait observability in context. Funder [125] introduced a framework of factors that can affect the accuracy of the rating of traits by human observers, such as *relevance* (i.e., the context must allow a person to

[16] https://sites.google.com/michalkosinski.com/mypersonality.
[17] https://console.bluemix.net/docs/services/personality-insights/science.html.

express the trait) and *availability* (i.e., the trait must be perceptible to others). Arguably, such factors also hamper the ability of computers to 'perceive' personality traits.

As regards relevance, some traits are naturally more 'external' than others and, therefore, more likely to be perceived by other judges, including computers [101]. It is therefore not surprising that Vinciarelli & Mohammadi [50] found in their survey that the reviewed studies consistently reported larger effect sizes for Extraversion, one of the most interpersonal traits of the five, which emerges from overt behavior towards others.

As for availability, currently there seem to be still a large gap between abstract, nuanced information like personality traits, and the cues that AI services can observe from the analysis of digital artifacts. In this perspective, it is not surprising that research on personality computing has so far privileged trait models like the Big Five, which are particularly suitable for processing because of the representation of personality as continuous numeric scores. Nonetheless, even in this case, research required further simplification of the richness of the theory by limiting the analysis to the first level of the hierarchy, while discarding the lower-level facets.

We argue that future research on personality computing in the SE domain should pay close attention to assessing what information is actually relevant and available in the specific context of the study. Context can be modeled at different levels of granularity. For example, context can be broadly considered at project level, to see if there are differences in the personality profiles of developers across the projects they participated in. However, analysis at a finer granularity, such as task level, may make it easier to contextualize the relevance and availability of traits in digital traces left by developers. For example, in code reviews, developers performing the inspection of external contributions are likely to behave in ways that make Agreeableness (i.e., cooperation with others) and Conscientiousness (i.e., thoroughness of the inspection) emerge from their comments, as supported by initial evidence reported in our previous work on trust [86,87].

Lack of self-reported data. One of the main limitations of the study revolves around the use of the Personality Insights service, which enabled the automated assessment of the personalities of a large number of developers from their emails, without having to rely on self-reported data. By exploiting a large number of communication messages archived in these software-related repositories – i.e., the toolset belonging to the social-programmer ecosystem [126] – more and more recent studies like ours have started to employ natural language processing (NLP) instruments for the automatic analysis of content. Still, many of these tools have not been designed or trained for handling the technical content typical of the software domain [127]. For instance, Jongeling et al. [128] have compared various sentiment analysis tools used in previous studies in software engineering and found that they can disagree with the manual labeling of corpora performed by individuals as well as with each other. Therefore, we advocate caution when drawing conclusions from NLP tools not specifically trained for the specific purpose and lexicon, and we acknowledge this as a potential threat to instrumentation validity. Still, prior research (e.g., [17]) found evidence that personality traits can be successfully derived from the analysis of written texts such as emails [129]. We also stress that we employed the Personality Insights service on emails only after parsing them to remove (most of) the technical content therein. In addition, Wang & Redmiles [130] used the LIWC 2007 tool to compute the baseline trust of developers parsing the content of their emails. The authors compared the results obtained using LIWC against those obtained using another linguistic resource (i.e., the NRC lexicon) and found them to converge. Finally, we note that while individuals may vary in how their personality traits manifest in email communication, potentially reducing the reliability of the automated inference technique we use, the large size of the sample that we study implies a reduction to the mean in terms of individual traits. In this sense, we expect that by averaging over hundreds of observed developers in the regression models, the inferred personality scores can still reflect the intensity and directionality of underlying associations with the response variables. We leave a detailed comparison of our findings obtained with Personality Insights API to LIWC and other similar tools as future work.

Language. Another potential issue related to the use of a tool to mine personality from text is related to the use of English as *lingua franca* in emails, i.e., some developers did not communicate using their native language. A limited vocabulary may have arguably prevented some lexical cues related to their personality from emerging from their written communication, as argued in the lexical hypothesis. Research in personality Psychology has validated psychometric questionnaire across nations after translating the question items [41]. Furthermore, previous studies on global software engineering have shown that language disparity and the use of English as *lingua franca* do affect development activities [131–133].

Lack of demographic data. Previous research on personality has found that lexicon and personality vary with age, gender, and nationality [41,66]. We acknowledge that our personality dataset does not include these pieces of information about developers. However, we note that this kind of information is usually unavailable in public project repositories due to privacy concerns.

External validity. Since the Apache ecosystem may not be representative of all types of large, distributed projects, especially commercial, we acknowledge the need to gather further evidence. Yet, independent replications are also welcome, as we have made all the code and the entire dataset available online.[18]

7. Conclusions

In this paper, we presented a quantitative analysis of the personality traits of the developers working in the Apache ecosystem. Developers' personalities were extracted from the projects' mailing list archives and modeled on the Big Five personality framework, using the IBM Personality Insights service.

We found there are three common types of personality profiles among developers, characterized in particular by their level of Agreeableness and Neuroticism. We also confirmed that developers' personalities traits assessed automatically are stable over time. Moreover, personality traits do not vary with their role, membership, and the level of contribution to the projects. Furthermore, we developed a couple of regression models and found that the developers who are more open are more likely to make projects contributors. This finding has practical implications in recommending the right mentors to project newcomers as well as for building new teams by considering the analysis of personalities for the prospect team members.

Part of our findings is in contrast with previous work on the personality of developers, thus calling for further replications. Nonetheless, overall, our results reinforce the need for future studies on human factors in software engineering to use psychometric tools to control for differences in developers' personalities.

We are currently collecting self-assessments from OSS developers, which, paired with a text corpus extracted from a large amount of communication traces available from public OSS project repositories, will provide us with an experimental dataset to train our own SE-specific tool for automatic personality recognition. This effort is still ongoing as obtaining a sufficient amount of self-assessments is a slow and challenging process due to the typical low return rate of web surveys in SE research [134].

Conflict of interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

[18] https://github.com/collab-uniba/personality.

Acknowledgements

We thank IBM for providing free access to the Personality Insights API. The computational work has been executed on the IT resources made available by two projects, ReCaS and PRISMA, funded by MIUR under the program "PON R&C 2007-2013." We are also grateful to Marco Iannotta for his help in the data extraction process.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.infsof.2019.05.012.

References

[1] G.M. Weinberg, The psychology of computer programming, 29, Van Nostrand Reinhold New York, 1971.
[2] B. Shneiderman, Software psychology: human factors in computer and information systems, Winthrop Publishers California, 1980.
[3] S. Cruz, F.Q. da Silva, L.F. Capretz, Forty years of research on personality in software engineering: a mapping study, Comput. Human Behav. 46 (2015) 94–113, doi:10.1016/j.chb.2014.12.008.
[4] N. Salleh, E. Mendes, J. Grundy, G.S.J. Burch, An empirical study of the effects of conscientiousness in pair programming using the five-factor personality model, in: Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering - ICSE '10, 1, ACM Press, New York, New York, USA, 2010, p. 577, doi:10.1145/1806799.1806883.
[5] M.V. Kosti, R. Feldt, L. Angelis, Personality, emotional intelligence and work preferences in software engineering: an empirical study, Inf. Softw. Technol. 56 (8) (2014) 973–990, doi:10.1016/j.infsof.2014.03.004.
[6] S.T. Acuña, M.N. Gómez, J.E. Hannay, N. Juristo, D. Pfahl, Are team personality and climate related to satisfaction and software quality? aggregating results from a twice replicated experiment, Inf. Softw. Technol. 57 (2015) 141–156, doi:10.1016/j.infsof.2014.09.002.
[7] F.Q. da Silva, A.C.C. França, M. Suassuna, L.M. de Sousa Mariz, I. Rossiley, R.C. de Miranda, T.B. Gouveia, C.V. Monteiro, E. Lucena, E.S. Cardozo, E. Espindola, Team building criteria in software projects: a mix-method replicated study, Inf. Softw. Technol. 55 (7) (2013) 1316–1340, doi:10.1016/j.infsof.2012.11.006.
[8] R. Feldt, L. Angelis, R. Torkar, M. Samuelsson, Links between the personalities, views and attitudes of software engineers, Inf. Softw. Technol. 52 (6) (2010) 611–624, doi:10.1016/j.infsof.2010.01.001.
[9] J.E. Hannay, E. Arisholm, H. Engvik, D.I.K. Sjoberg, Effects of personality on pair programming, IEEE Trans. Softw. Eng. 36 (1) (2010) 61–80, doi:10.1109/TSE.2009.41.
[10] A.R. Gilal, J. Jaafar, M. Omar, S. Basri, A. Waqas, A rule-based model for software development team composition: team leader role with personality types and gender classification, Inf. Softw. Technol. 74 (2016) 105–113, doi:10.1016/j.infsof.2016.02.007.
[11] A. Raza, L.F. Capretz, Z. Ul-Mustafa, Personality profiles of software engineers and their software quality preferences, Int. J. Inf. Syst. Soc. Change 5 (3) (2014) 77–86, doi:10.4018/ijissc.2014070106.
[12] E. Smith, R. Loftin, E. Murphy-Hill, C. Bird, T. Zimmermann, Improving developer participation rates in surveys, in: Cooperative and Human Aspects of Software Engineering (CHASE), 2013 6th International Workshop on, 2013, pp. 89–92.
[13] T. Yarkoni, Personality in 100,000 words: a large-scale analysis of personality and word use among bloggers, J. Res. Pers. 44 (3) (2010) 363–373, doi:10.1016/j.jrp.2010.04.001.
[14] J.R. Erenkrantz, R.N. Taylor, Supporting distributed and decentralized projects: drawing lessons from the open source community, TUM (2003) 21.
[15] X. Xia, D. Lo, L. Bao, A. Sharma, S. Li, Personality and project success: insights from a large-scale study with professionals, in: 2017 IEEE International Conference on Software Maintenance and Evolution (ICSME), IEEE, 2017, pp. 318–328, doi:10.1109/ICSME.2017.50.
[16] A. Rastogi, N. Nagappan, On the personality traits of GitHub contributors, in: 2016 IEEE 27th International Symposium on Software Reliability Engineering (ISSRE), IEEE, 2016, pp. 77–86, doi:10.1109/ISSRE.2016.43.
[17] J. Shen, O. Brdiczka, J. Liu, Understanding email writers: personality prediction from email messages, Lect. Note. Comput. Sci. (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 7899 LNCS (2013) 318–330, doi:10.1007/978-3-642-38844-6_29.
[18] R.R. McCrae, O.P. John, An introduction to the five-factor model and its applications, J.Pers. 60 (2) (1992) 175–215, doi:10.1111/j.1467-6494.1992.tb00970.x.
[19] B.D. Raad, The Big Five Personality Factors: The Psycholexical Approach to Personality, Hogrefe & Huber, 2000.
[20] R.R. McCrae, P.T. Costa, Validation of the five-factor model of personality across instruments and observers, J. Pers. Soc. Psychol. 52 (1) (1987) 81–90, doi:10.1037/0022-3514.52.1.81.
[21] F. Mairesse, M.A. Walker, M.R. Mehl, R.K. Moore, Using linguistic cues for the automatic recognition of personality in conversation and text, J.Artif.Intell.Res. 30 (2007) 457–500.
[22] O.P. John, S. Srivastava, Chap. 4 - The big five trait taxonomy: history, measurement, and theoretical perspectives, in: L.A. Pervin, O.P. John (Eds.), Handbook of personality: Theory and research (2nd ed.), Guilford, New York, 1999.
[23] R.M. Ryckman, Theories of Personality, 10th ed., Cengage Learning, Boston, 2012.
[24] L.R. Goldberg, The structure of phenotypic personality traits, Am.Psychol. 48 (1) (1993) 26–34, doi:10.1037/0003-066X.48.1.26.
[25] L.R. Goldberg, Language and individual differences: the search for universals in personality lexicons, Rev. Pers. Soc. Psychol. 2 (1) (1981) 141–165.
[26] G.W. Allport, H.S. Odbert, Trait-names: a psycho-lexical study, Psychol. Monograph. 47 (1) (1936) i.
[27] P.T. Costa, R.R. McCrae, The NEO-PI Personality Inventory, Psychological Assessment Resources, Odessa, FL, 1985.
[28] N. Salleh, E. Mendes, J. Grundy, Investigating the effects of personality traits on pair programming in a higher education setting through a family of experiments, Empir. Softw. Eng. 19 (3) (2014) 714–752, doi:10.1007/s10664-012-9238-4.
[29] J.M. Digman, Personality structure: emergence of the five-Factor model, Annual Review of Psychology 41 (1) (1990) 417–440, doi:10.1146/annurev.ps.41.020190.002221.
[30] I.B. Myers, M.H. McCaulley, N.L. Quenk, A.L. Hammer, MBTI manual: aguide to the development and use of the Myers-Briggs type indicator, 3, Consulting Psychologists Press Palo Alto, CA, 1998.
[31] R.R. McCrae, P.T.C. Jr, Reinterpreting the Myers-Briggs type indicator from the perspective of the five-factor model of personality, J. Pers. 57 (1) (1989) 17–40.
[32] J.B. Murray, Review of research on the Myers-Briggs type indicator, Percept. Mot. Skills 70 (3_suppl) (1990) 1187–1202.
[33] L.E. Hicks, Conceptual and empirical analysis of some assumptions of an explicitly typological theory, J Personal. Social Psychol. 46 (5) (1984) 1118–1131, doi:10.1037/0022-3514.46.5.1118.
[34] D. Keirsey, M.M. Bates, Please Understand Me: Character & Temperament Types, Prometheus Nemesis Book Company Del Mar, CA, 1984.
[35] D. Keirsey, Please understand me 2, Prometheus Nemesis Book Company, 1998.
[36] N.R. Abramson, Internal reliability of the Keirsey temperament sorter II: cross-national application to american, canadian, and korean samples, J. Psychol. Type 70 (2) (2010) 19–30.
[37] P.T. Costa, R.R. McCrae, I.P.A. Resources, Revised NEO Personality Inventory (NEO PI-R) and NEO Five-Factor Inventory (NEO-FFI), Psychological Assessment Resources, 1992.
[38] R.R. McCrae, Trait psychology and culture: exploring intercultural comparisons, J. Pers. 69 (6) (2001) 819–846.
[39] R.R. McCrae, NEO-PI-R data from 36 cultures: Further intercultural comparisons, in: R.R. McCrae, J. Allik (Eds.), International and cultural psychology series. The Five-Factor model of personality across cultures, Kluwer Academic/Plenum, New York, NY, US, 2002, pp. 105–125, doi:10.1007/978-1-4615-0763-5_6.
[40] J. Allik, R.R. McCrae, Toward a geography of personality traits: patterns of profiles across 36 cultures, J. Cross-Cult. Psychol. 35 (1) (2004) 13–28.
[41] D.P. Schmitt, J. Allik, R.R. McCrae, V. Benet-Martínez, et al., The geographic distribution of big five personality traits: patterns and profiles of human self-description across 56 nations, J. Cross-Cult. Psychol. 38 (2) (2007) 173–212, doi:10.1177/0022022106297299.
[42] L.R. Goldberg, J.A. Johnson, H.W. Eber, R. Hogan, M.C. Ashton, C.R. Cloninger, H.G. Gough, The international personality item pool and the future of public-domain personality measures, J. Res. Pers. 40 (1) (2006) 84–96, doi:10.1016/j.jrp.2005.08.007.
[43] J.W. Pennebaker, L.A. King, Linguistic styles: language use as an individual difference., J. Pers. Soc. Psychol. 77 (6) (1999) 1296.
[44] V. Kaushal, M. Patwardhan, Emerging trends in personality identification using online social networks—a literature survey, ACM Trans. Knowl. Discov. Data 12 (2) (2018) 1–30, doi:10.1145/3070645.
[45] N. Novielli, F. Calefato, F. Lanubile, Towards discovering the role of emotions in stack overflow, in: Proceedings of the 6th International Workshop on Social Software Engineering, ACM, New York, NY, USA, 2014, pp. 33–36, doi:10.1145/2661685.2661689.
[46] F. Calefato, F. Lanubile, F. Maiorano, N. Novielli, Sentiment polarity detection for software development, Empir. Softw. Eng. 23 (3) (2018) 1352–1382, doi:10.1007/s10664-017-9546-9.
[47] K.R. Scherer, Vocal communication of emotion: a review of research paradigms, Speech Commun. 40 (1–2) (2003) 227–256.
[48] G. Carducci, G. Rizzo, D. Monti, E. Palumbo, M. Morisio, Twitpersonality: computing personality traits from tweets using word embeddings and supervised learning, Information 9 (5) (2018), doi:10.3390/info9050127.
[49] N. Majumder, S. Poria, A. Gelbukh, E. Cambria, Deep learning-based document modeling for personality detection from text, IEEE Intell. Syst. 32 (2) (2017) 74–79, doi:10.1109/MIS.2017.23.
[50] A. Vinciarelli, G. Mohammadi, A survey of personality computing, IEEE Trans. Affect. Comput. 5 (3) (2014) 273–291, doi:10.1109/TAFFC.2014.2330816.
[51] G. Farnadi, G. Sitaraman, S. Sushmita, F. Celli, M. Kosinski, D. Stillwell, S. Davalos, M.F. Moens, M. De Cock, Computational personality recognition in social media, User Modell. User-Adapt. Interact. 26 (2–3) (2016), doi:10.1007/s11257-016-9171-0.
[52] S. Argamon, S. Dhawle, M. Koppel, J. Pennebaker, Lexical predictors of personality type, in: Proceedings of Classification Society of North America, Annul Meeting St. Louis MI, June, 2005.
[53] J. Oberlander, S. Nowson, Whose thumb is it anyway?: classifying author personality from weblog text, in: Proceedings of the COLING/ACL on Main Conference Poster Sessions, 2006, pp. 627–634.

[54] S. Nowson, J. Oberlander, Identifying more bloggers, Proc. Int. Conf. Weblogs Soc. (ICWSM'07) (2007).
[55] A.J. Gill, S. Nowson, J. Oberlander, What are they blogging about? personality, topic and motivation in blogs, in: Third International AAAI Conference on Weblogs and Social Media, 2009.
[56] D. Quercia, M. Kosinski, D. Stillwell, J. Crowcroft, Our twitter profiles, our selves: predicting personality with twitter, in: Proceedings - 2011 IEEE International Conference on Privacy, Security, Risk and Trust and IEEE International Conference on Social Computing, PASSAT/SocialCom 2011, IEEE, 2011, pp. 180–185, doi:10.1109/PASSAT/SocialCom.2011.26.
[57] J. Golbeck, C. Robles, K. Turner, Predicting personality with social media, in: CHI '11 Extended Abstracts on Human Factors in Computing Systems, in: CHI EA '11, ACM, New York, NY, USA, 2011, pp. 253–262, doi:10.1145/1979742.1979614.
[58] J. Golbeck, C. Robles, M. Edmondson, K. Turner, Predicting personality from twitter, in: 2011 IEEE Third International Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third International Conference on Social Computing, 2011, pp. 149–156, doi:10.1109/PASSAT/SocialCom.2011.33.
[59] F. Celli, Unsupervised personality recognition for social network sites, in: Proc. of sixth international conference on digital society, 2012.
[60] S.M. Mohammad, S. Kiritchenko, Using hashtags to capture fine emotion categories from tweets, Comput. Intell. 31 (2) (2015) 301–326. doi: 10.1111/coin.12024.
[61] Ibm personality insights, the science behind the service, 2016. https://console.bluemix.net/docs/services/personality-insights/science.html?locale=en#science.
[62] F. Liu, J. Perez, S. Nowson, A language-independent and compositional model for personality trait recognition from short texts, in: Proc. 15th Conference ofthe European Chapter of the Association for Computational Linguistics (EACL'17): Volume 1, 2017, pp. 754–764.
[63] F. Celli, F. Pianesi, D. Stillwell, M. Kosinski, Workshop on computational personality recognition: shared task, in: Proceedings of the Workshop on Personality Recognition, 2006, 2013, pp. 2–5.
[64] J.W. Pennebaker, R.L. Boyd, K. Jordan, K. Blackburn, The development and psychometric properties of LIWC 2015, 2015.
[65] Y.R. Tausczik, J.W. Pennebaker, The psychological meaning of words: LIWC and computerized text analysis methods„ J. Lang. Soc. Psychol. 29 (1) (2010) 24–54.
[66] H.A. Schwartz, J.C. Eichstaedt, M.L. Kern, L. Dziurzynski, S.M. Ramones, M. Agrawal, A. Shah, M. Kosinski, D. Stillwell, M.E.P. Seligman, L.H. Ungar, Personality, gender, and age in the language of social media: the open-Vocabulary approach, PLoS ONE 8 (9) (2013) e73791.
[67] P.-H. Arnoux, A. Xu, N. Boyette, J. Mahmud, R. Akkiraju, V. Sinha, 25 tweets to know you: anew model to predict personality with social media, in: Eleventh International AAAI Conference on Web and Social Media, 2017, pp. 472–475.
[68] M. Coltheart, The MRC psycholinguistic database, Q. J. Exper. Psychol. 33 (4) (1981) 497–505.
[69] M. Kosinski, D. Stillwell, T. Graepel, Private traits and attributes are predictable from digital records of human behavior, Proc. Natl. Acad. Sci. (2013) 201218772.
[70] F. Celli, B. Lepri, J.-I. Biel, D. Gatica-Perez, G. Riccardi, F. Pianesi, The workshop on computational personality recognition 2014, in: Proceedings of the ACM International Conference on Multimedia - MM '14, 2014, pp. 1245–1246, doi:10.1145/2647868.2647870.
[71] F. Rangel, F. González, F. Restrepo, M. Montes, P. Rosso, Pan@fire: overview of the pr-soco track on personality recognition in source code, in: P. Majumder, M. Mitra, P. Mehta, J. Sankhavara (Eds.), Text Processing, Springer International Publishing, Cham, 2018, pp. 1–19.
[72] B.W. Roberts, N.R. Kuncel, R. Shiner, A. Caspi, L.R. Goldberg, The power of personality: the comparative validity of personality traits, socioeconomic status, and cognitive ability for predicting important life outcomes, Perspect. Psychol. Sci. 2 (4) (2007) 313–345.
[73] G.J. Meyer, S.E. Finn, L.D. Eyde, G.G. Kay, K.L. Moreland, R.R. Dies, E.J. Eisman, T.W. Kubiszyn, G.M. Reed, Psychological testing and psychological assessment: a review of evidence and issues, Am. Psychol. 56 (2) (2001) 128.
[74] S.T. Acuña, M. Gómez, N. Juristo, How do personality, team processes and task characteristics relate to job satisfaction and software quality? Inf. Softw. Technol. 51 (3) (2009) 627–639, doi:10.1016/j.infsof.2008.08.006.
[75] A.D.D. Cunha, D. Greathead, Does personality matter? Commun. ACM 50 (5) (2007) 109–112, doi:10.1145/1230819.1241672.
[76] N. Gorla, Y.W. Lam, Who should work with whom? Commun. ACM 47 (6) (2004) 79–82, doi:10.1145/990680.990684.
[77] Z. Karimi, S. Wagner, The influence of personality on computer programming: asummary of a systematic literature review, Fakultät Informatik, Elektrotechnik und Informationstechnik. Institut für Softwaretechnologie, 2014.
[78] Z. Karimi, A. Baraani-Dastjerdi, N. Ghasem-Aghaee, S. Wagner, Links between the personalities, styles and performance in computer programming, J. Syst. Softw. 111 (C) (2016) 228–241, doi:10.1016/j.jss.2015.09.011.
[79] S. McDonald, H.M. Edwards, Who should test whom? Commun. ACM 50 (1) (2007) 66–71, doi:10.1145/1188913.1188919.
[80] D. Bell, T. Hall, J.E. Hannay, D. Pfahl, S.T. Acuna, Software engineering group work, in: Proceedings of the 2010 Special Interest Group on Management Information System's 48th annual conference on Computer personnel research on Computer personnel research - SIGMIS-CPR '10, ACM Press, New York, New York, USA, 2010, p. 43, doi:10.1145/1796900.1796921.
[81] T. Kanij, R. Merkel, J. 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, IEEE, 2015, pp. 1–7, doi:10.1109/CHASE.2015.7.
[82] M.V. Kosti, R. Feldt, L. Angelis, Archetypal personalities of software engineers and their work preferences: a new perspective for empirical studies, Empir. Softw. Eng. 21 (4) (2016) 1509–1532, doi:10.1007/s10664-015-9395-3.
[83] E.K. Smith, C. Bird, T. Zimmermann, Beliefs, practices, and personalities of software engineers, in: Proceedings of the 9th International Workshop on Cooperative and Human Aspects of Software Engineering - CHASE '16, ACM Press, New York, New York, USA, 2016, pp. 15–18, doi:10.1145/2897586.2897596.
[84] P.C. Rigby, A.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), IEEE, 2007, p. 23, doi:10.1109/MSR.2007.35.
[85] B. Bazelli, A. Hindle, E. Stroulia, On the personality traits of StackOverflow users, in: 2013 IEEE International Conference on Software Maintenance, IEEE, 2013, pp. 460–463, doi:10.1109/ICSM.2013.72.
[86] F. Calefato, F. Lanubile, Establishing personal trust-based connections in distributed teams, Internet Technol. Lett. 1 (4) (2017) e6, doi:10.1002/itl2.6.
[87] F. Calefato, F. Lanubile, N. Novielli, A preliminary analysis on the effects of propensity to trust in distributed software development, in: 2017 IEEE 12th International Conference on Global Software Engineering (ICGSE), IEEE, 2017, pp. 56–60, doi:10.1109/ICGSE.2017.1.
[88] F. Calefato, G. Iaffaldano, F. Lanubile, B. Vasilescu, On developers' personality in large-scale distributed projects: the case of the apache ecosystem, in: Proceedings of the 13th Conference on Global Software Engineering, in: ICGSE '18, ACM, New York, NY, USA, 2018, pp. 92–101, doi:10.1145/3196369.3196372.
[89] Y. Ye, K. Kishida, Toward an understanding of the motivation open source software developers, in: Proceedings of the 25th International Conference on Software Engineering, 2003, pp. 419–429.
[90] M.V. Mäntylä, F. Calefato, M. Claes, Natural Language or Not (NLON): apackage for software engineering text analysis pipeline, in: Proceedings of the 15th International Conference on Mining Software Repositories, ACM, New York, NY, USA, 2018, pp. 387–391, doi:10.1145/3196398.3196444.
[91] B. Vasilescu, A. Serebrenik, V. Filkov, A data set for social diversity studies of GitHub teams, in: Proceedings of the 12th Working Conference on Mining Software Repositories, IEEE Press, Piscataway, NJ, USA, 2015, pp. 514–517.
[92] B. Plank, D. Hovy, Personality traits on twitter–or–how to get 1,500 personality tests in a week, in: Proceedings of the 6th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, 2015, pp. 92–98.
[93] P.-H. Arnoux, A. Xu, N. Boyette, J. Mahmud, R. Akkiraju, V. 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.
[94] J. Pennington, R. Socher, C.D. Manning, Glove: Global vectors for word representation, in: Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1532–1543.
[95] D.A. Cobb-Clark, S. Schurer, The stability of big-five personality traits, Econ. Lett. 115 (1) (2012) 11–15.
[96] C. Adele, B. Leo, Archetypal analysis, Technometrics 36 (4) (1994) 338–347, doi:10.1080/00401706.1994.10485840.
[97] W.H. Greene, Econometric Analysis, 8th Edition, Prentice Hall, 2018.
[98] D.E. Hinkle, W. Wiersma, S.G. Jurs, et al., Applied statistics for the behavioral sciences, 2nd ed., Houghton Mifflin, Boston, 1988.
[99] A. Field, J. Miles, Z. Field, Discovering Statistics Using R, Sage, 2012.
[100] H. He, E.A. Garcia, Learning from imbalanced data, IEEE Trans. Knowl. Data Eng. 21 (9) (2009) 1263–1284, doi:10.1109/TKDE.2008.239.
[101] A.G.C. Wright, Current directions in personality science and the potential for advances through computing, IEEE Trans. Affect. Comput. 5 (3) (2014) 292–296, doi:10.1109/TAFFC.2014.2332331.
[102] N. Cliff, Dominance statistics: ordinal analyses to answer ordinal questions., Psychol.Bull. 114 (3) (1993) 494.
[103] M.S. Elliott, W. Scacchi, Free software developers as an occupational community: resolving conflicts and fostering collaboration, in: Proceedings of the 2003 International ACM SIGGROUP Conference on Supporting Group Work, 2003, pp. 21–30.
[104] C. Jensen, S. King, V. Kuechler, Joining free/open source software communities: an analysis of newbies' first interactions on project mailing lists, in: System Sciences (HICSS), 2011 44th Hawaii International Conference on, 2011, pp. 1–10.
[105] F. Calefato, F. Lanubile, N. Novielli, EmoTxt: atoolkit for emotion recognition from text, in: Proc. of 7th Int'l Conf. on Affective Computing and Intelligent Interaction Workshops and Demos, 2017, pp. 79–80, doi:10.1109/ACIIW.2017.8272591.
[106] N. Novielli, F. Calefato, F. Lanubile, A gold standard for emotions annotation in stack overflow, in: Proc. of 15th Int'l Conf. on Mining Software Repositories, 2018, doi:10.1145/3196398.3196453.
[107] S. Beecham, N. Baddoo, T. Hall, H. Robinson, H. Sharp, Motivation in software engineering: a systematic literature review, Inf. Softw. Technol. 50 (9–10) (2008) 860–878.
[108] P. Tourani, B. Adams, A. Serebrenik, Code of conduct in open source projects, in: Software Analysis, Evolution and Reengineering (SANER), 2017 IEEE 24th International Conference on, IEEE, 2017, pp. 24–33.
[109] R. Feldt, L. Angelis, M. 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, ACM Press, New York, New York, USA, 2008, pp. 49–52, doi:10.1145/1370114.1370127.
[110] P. Lenberg, R. Feldt, L.G. Wallgren, Behavioral software engineering: a definition and systematic literature review, J. Syst. Softw. 107 (2015) 15–37, doi:10.1016/j.jss.2015.04.084.
[111] N. Ducheneaut, Socialization in an open source software community: a socio-technical analysis, Comput. Support. Coop. Work (CSCW) 14 (4) (2005) 323–368.

[112] I. Steinmacher, I. Wiese, A.P. Chaves, M.A. Gerosa, Why do newcomers abandon open source software projects? in: Cooperative and Human Aspects of Software Engineering (CHASE), 2013 6th International Workshop on, 2013, pp. 25–32.
[113] I. Steinmacher, T. Conte, M.A. Gerosa, D. Redmiles, Social barriers faced by newcomers placing their first contribution in open source software projects, in: Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing, 2015, pp. 1379–1392.
[114] C. Bird, A. Gourley, P. Devanbu, A. Swaminathan, G. Hsu, Open borders? immigration in open source projects, in: Proceedings of the Fourth International Workshop on Mining Software Repositories, 2007, p. 6.
[115] I. Steinmacher, M.A. Gerosa, D. Redmiles, Attracting, onboarding, and retaining newcomer developers in open source software projects, Workshop on Global Software Development in a CSCW Perspective, 2014.
[116] G. Canfora, M.D. Penta, R. Oliveto, S. Panichella, Who is going to mentor newcomers in open source projects? in: Proceedings of the ACM SIGSOFT 20th International Symposium on the Foundations of Software Engineering, 2012, p. 44.
[117] C. Lebeuf, M.-A. Storey, A. Zagalsky, Software bots, IEEE Softw. (1) (2018) 18–23.
[118] H.-D. Yang, H.-R. Kang, R.M. Mason, An exploratory study on meta skills in software development teams: antecedent cooperation skills and personality for shared mental models, Eur. J. Inf. Syst. 17 (1) (2008) 47–61.
[119] J.S. Karn, S. Syed-Abdullah, A.J. Cowling, M. Holcombe, A study into the effects of personality type and methodology on cohesion in software engineering teams, Behav. Inf. Technol. 26 (2) (2007) 99–111.
[120] J. Tsay, L. Dabbish, J. Herbsleb, Influence of social and technical factors for evaluating contribution in github, in: Proceedings of the 36th International Conference on Software Engineering, ACM, 2014, pp. 356–366.
[121] G. Gousios, M. Pinzger, A.v. Deursen, An exploratory study of the pull-based software development model, in: Proceedings of the 36th International Conference on Software Engineering, ACM, 2014, pp. 345–355.
[122] M.R. Mehl, J.W. Pennebaker, D.M. Crow, J. Dabbs, J.H. Price, The electronically activated recorder (ear): a device for sampling naturalistic daily activities and conversations, Behav. Res. Method. Instrum.Comput. 33 (4) (2001) 517–523, doi:10.3758/BF03195410.
[123] G. Boyle, E. Helmes, Methods of personality assessment, Cambridge Handbook Pers. Psychol. (2009) 110.
[124] M.K. Mount, M.R. Barrick, J.P. Strauss, Validity of observer ratings of the big five personality factors, J. Appl. Psychol. 79 (2) (1994) 272.
[125] D.C. Funder, On the accuracy of personality judgment: a realistic approach, Psychol. Rev. 102 (4) (1995) 652.
[126] L. Singer, F. Figueira Filho, B. Cleary, C. Treude, M.-A. Storey, K. Schneider, Mutual assessment in the social programmer ecosystem: an empirical investigation of developer profile aggregators, in: Proceedings of the 2013 Conference on Computer Supported Cooperative Work, ACM, 2013, pp. 103–116.
[127] N. Novielli, F. Calefato, F. Lanubile, The challenges of sentiment detection in the social programmer ecosystem, in: Proceedings of the 7th International Workshop on Social Software Engineering, ACM, 2015, pp. 33–40.
[128] R. Jongeling, P. Sarkar, S. Datta, A. Serebrenik, On negative results when using sentiment analysis tools for software engineering research, Empir. Softw. Eng. 22 (5) (2017) 2543–2584.
[129] J. Shen, O. Brdiczka, J. Liu, Understanding email writers: personality prediction from email messages, Lect. Note. Comput. Sci. (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 7899 LNCS (2013) 318–330, doi:10.1007/978-3-642-38844-6_29.
[130] Y. Wang, D. Redmiles, Cheap talk, cooperation, and trust in global software engineering, Empir. Softw. Eng. 21 (6) (2016) 2233–2267.
[131] F. Calefato, F. Lanubile, T. Conte, R. Prikladnicki, Assessing the impact of real-time machine translation on multilingual meetings in global software projects, Empir. Softw. Eng. 21 (3) (2016) 1002–1034.
[132] F. Calefato, F. Lanubile, R. Prikladnicki, A controlled experiment on the effects of machine translation in multilingual requirements meetings, in: 2011 Sixth IEEE International Conference on Global Software Engineering, 2011, pp. 94–102.
[133] Y. Wang, Language matters, in: 2015 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM), IEEE, 2015, pp. 1–10.
[134] E. Smith, R. Loftin, E. Murphy-Hill, C. Bird, T. Zimmermann, Improving developer participation rates in surveys, in: 2013 6th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE), IEEE, 2013, pp. 89–92.