Computers in Human Behavior 46 (2015) 94–113

Contents lists available at ScienceDirect

Computers in Human Behavior

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

Forty years of research on personality in software engineering: A mapping study

Shirley Cruz [a], Fabio Q.B. da Silva [a,*], Luiz Fernando Capretz [b]

[a] Centre for Informatics, Federal University of Pernambuco, Av. Jornalista Anibal Fernandes, s/n - Cidade Universitária, Recife, PE 50.740-560, Brazil
[b] Department of Electrical & Computer Engineering, Western University, London, Ontario N6A5B9, Canada

ARTICLE INFO

Article history:
Available online 23 January 2015

Keywords:
Human factors in software engineering
Software psychology
Empirical software engineering
Mapping study
Systematic literature review

ABSTRACT

In this article, we present a systematic mapping study of research on personality in software engineering. The goal is to plot the landscape of current published empirical and theoretical studies that deal with the role of personality in software engineering. We applied the systematic review method to search and select published articles, and to extract and synthesize data from the selected articles that reported studies about personality. Our search retrieved more than 19,000 articles, from which we selected 90 articles published between 1970 and 2010. Nearly 72% of the studies were published after 2002 and 83% of the studies reported empirical research findings. Data extracted from the 90 studies showed that education and pair programming were the most recurring research topics, and that MBTI was the most used test. Research related to pair programming, education, team effectiveness, software process allocation, software engineer personality characteristics, and individual performance concentrated over 88% of the studies, while team process, behavior and preferences, and leadership performance were the topics with the smallest number of studies. We conclude that the number of articles has grown in the last few years, but contradictory evidence was found that might have been caused by differences in context, research method, and versions of the tests used in the studies. While this raises a warning for practitioners that wish to use personality tests in practice, it shows several opportunities for the research community to improve and extend findings in this field.

1. Introduction

The work described in this paper builds on previous work (Cruz, da Silva, Monteiro, & Rossilei, 2011) carried out by the authors that investigates the knowledge produced about the influence of personality in software engineering. The preliminary study showed a considerable amount of conflicting evidence, which suggests that it is an immature research field with many opportunities to be explored by the research community.

When starting the previous study, our initial goal was to collect the largest possible quantity of studies published on the subject. To our knowledge, this was the first attempt to review the literature on personality in software engineering in a systematic way. We found only one study that performed a systematic review regarding the influence of personality on pair programming (Salleh, Mendes, Grundy, & Burch, 2009). This review has been recently updated, and the entire set of studies including one replication can by found in Salleh, Mendes, and Grundy (2014). Other studies have been conducted to review the literature on different aspects of software engineering (Capretz, 2003; Pocius, 1991), but these do not use a systematic approach.

In our previous study, we analyzed 42 primary studies, 38 originating from an automatic search, and 4 from a manual search. Despite the use of a carefully designed and executed systematic review protocol, some known studies were not included in the search results. Our goal in this new review was to increase the sensitivity of the search process. To achieve a higher sensitivity, we changed the search process in two complementary ways. First, we expanded the search string to include synonyms of the search terms. Second, we added a "snowball" search strategy in the second stage of the search process to look for relevant papers in the references of the papers selected in the first stage of the search. We used a set of known relevant papers in the area to calibrate the new search string and increase the sensitivity of the automatic search. The combination of the new string and the "snowball" search resulted in the addition of 48 new papers to the 42 investigated in the first review, bringing the total number of relevant papers analyzed to 90.

* Corresponding author.
E-mail addresses: ssj@cin.ufpe.br (S. Cruz), fabio@cin.ufpe.br (F.Q.B. da Silva), lcapretz@uwo.ca (L.F. Capretz).

http://dx.doi.org/10.1016/j.chb.2014.12.008
0747-5632/

In addition to including new articles, this work adds another research question to the four exiting questions to be answered by extracting data from the primary studies, as detailed in Section 3.1. We also provide more detailed information of the context of the studies, which is necessary to establish a comprehensive understanding of the research area. This updated literature review will help managers, software engineers and interested researchers in the field to determine the current state of research about personality in software engineering.

In this article, we report the results of a systematic review of the studies published between 1970 and 2010 that addressed the problems related to the influence of individual personality in software engineering. We identified and summarized the main topics researched in the studies, as well as the research method (theoretical or empirical), the type of subjects (students or professionals) and, when applicable, the personality tests used. Further, we attempted to integrate the results showing the personality profiles of software engineers and the effects of personality in individual or team performance, although this integration was not always possible due to key differences between the studies.

This article is organized as follows. In Section 2, we present a brief conceptual background about personality theories and related work. In Section 3, we describe the review method. In Section 4, the results of the review are presented, answering our research questions. In Section 5, we discuss the implications of our results for research and practice, and the limitations of this review. Finally, in Section 6, conclusions and directions for future work are presented.

2. Background and related work

There are many definitions of the term *personality* as established by various psychologists; these definitions generally include the basic elements that make up the theoretical conceptualization of the construct. However, it seems that there is no perfect definition of personality, and also no consensus on the issue in the field of psychology. While a deeper debate about nomenclature and conceptual definitions is out of the scope of this paper, we need some definitions in order to guide the review process. In this section, we provide such definitions, briefly describe five related works, and discuss how this article improves on the preliminary results published by Cruz et al. (2011).

2.1. Concepts and definitions

Personality is generally viewed as a dynamic organization, inside the person, of psychophysical systems that create the person's characteristic patterns of behavior, thoughts, and feelings. Ryckman (2004) defined personality as "the dynamic and organized set of characteristics possessed by a person that uniquely influences his or her cognitions, motivations, and behaviors in various situations". We use these definitions because they are general enough to allow the inclusion of studies covering a wide range of personality theories and research methods. The definitions clearly separate personality from other constructs like cognition, motivation, and behavior, which are not the central interests of this review.

The study of personality has been developed over the years to include an abundance of theoretical traditions in the field of psychology. These traditions are organized around seven perspectives on personality, which are frequently labeled as (1) dispositional, (2) biological, (3) psychoanalytic, (4) neoanalytic, (5) learning, (6) phenomenological, and (7) cognitive self-regulation (Carver & Scheier, 1988). The dispositional perspective encompasses the traits and types theory, which is one of the most used theories in organizational psychology (Anderson, Ones, Sinangil, & Viswesvaran, 2002) and in studies on personality in software engineering. The present review focuses on this personality perspective.

The trait and type approach assumes that personality consists of stable inner qualities that differ between individuals and influence behavior. Traits are defined by the American Psychiatric Association as enduring patterns of perceiving, relating to, and thinking about the environment and oneself that are exhibited in a wide range of social and personal contexts. People are assigned a specific personality type based on the classification psychological differences. Types can be distinguished from traits in that the latter can be manifested in different levels or degrees, whereas types are discrete.

Most studies on personality in software engineering use personality tests to identify differences among individuals. In psychology, there are two major categories of personality tests: projective and objective. Projective tests assess individual personality through responses from ambiguous stimuli, with the assumption that personality is unconscious and that an individual's responses will reveal his or her inner characteristics. Objective tests measure personality by self-assessment questionnaires, with the underlying assumption that personality is primarily conscious and can be directly accessed.

The studies included in this review use various forms of objective personality tests. The reason for this is twofold: firstly, objective tests are considered more reliable and valid than projective ones, and secondly, objective tests are easier to administer, thus giving the (false) impression that they can be used by researchers without a deeper background in psychology and psychometrics. While this is true for the initial administration of the test, McDonald and Edwards (2007) warn that interpretation of the results and analysis of their practical implications are not straightforward and require properly trained professionals.

2.2. Related work

We found five studies that review the literature on personality in software engineering. The review presented by McDonald and Edwards (2007) surveyed published articles in software engineering that focus on the application and interpretation of personality tests. The authors reviewed 40 papers published between 1984 and 2004, also conducting an in-depth analysis on 13 distinct empirical studies using personality tests. The aim of this analysis was "to identify whether reliable and valid instruments have been used, whether the test chosen is appropriate for the purpose, and the extent to which the personality testing process used is explicitly reported and discussed" (Mcdonald & Edwards, 2007). The authors placed great emphasis on determining whether the testing process, including interpretation of the results, was carried out directly or in consultation with qualified professionals.

The analysis of the primary studies posed several methodological problems with respect to reliability and validity of the test instruments, and with respect to the incomplete and sometimes incorrect interpretation of the results. The authors conclude the review with several recommendations for potential participants in testing processes, academics conducting tests, and practitioners that wish to interpret results from published work.

The review presented by Hannay, Arisholm, Engvik, and Sjoberg (2010) and Salleh et al. (2009) surveyed published articles that investigated the impact of personality in pair programming, a practice where two programmers work together on the same programming task using one computer and one keyboard. Each of the studies reviewed 10 papers, together totaling 14 distinct articles, as 6 studies were included in both reviews. In general, their findings are quite diverse. While some studies reported that individual personality traits do not significantly affect pair programming performance, others studies reported that pairs of mixed personalities performed better than pairs of same personality.

Pocius (1991) performed a review of 12 published articles that investigated how personality traits relate to programming aptitude and achievement and 6 published articles that describe programmer personality. The author gives evidence that some personality factors, like introversion, contribute to improved programming performance. However, no evidence is given that MBTI categories correlate to programming aptitude and achievement. Other findings reported are that programmers tend to have more introverted, intuitive, and thinking preferences, and that the most frequent MBTI personality types found among them are the ISTJ, INTJ, and INTP.

The review presented by Balijepally, Mahapatra, and Nerur (2006) surveyed 13 distinct articles and found that "Jungian typology, operationalized as Myers-Briggs Type Indicator (MBTI), is the most popular approach for assessing personality profiles in software engineering. Nevertheless, the Five Factor Model (FFM) of personality is currently gaining popularity among personality psychologists."

Although these reviews present important results for the research in the theme, they have one important limitation. There are no explicitly stated search and inclusion/exclusion processes. In fact, we do not consider these studies a systematic review, with the exception of the study by Hannay et al. (2010). The present review has a broader goal of mapping studies about personality in all areas of software engineering.

2.3. Improvements to preliminary results of the mapping study

We presented the preliminary results of this mapping study at the *15th International Conference on Evaluation and Assessment of Software Engineering (EASE'2011)*, and the article has been published in the conference proceedings (Cruz et al., 2011). In the conference paper, we presented the results from our review of 42 articles published between 1978 and 2010.

In the present article, we improve and extend those results in several important ways:

- A new automatic search was performed using a new updated search string that included synonyms of software engineering and personality to increase the sensitivity of the search (coverage); this is further explained in Section 3.3.
- A second stage search was performed to generate a "snowball" effect by consulting the references of relevant papers retrieved in the first stage of the search process; this effectively increased the sensitivity of the search.
- Three primary studies were excluded from the list of initial articles because we found newer or more complete articles with the new search procedures.
- One new research question was added, to guide important discussions about the personality profile of software engineer.
- The presentation of the results and the discussion of the answers to the research questions were largely extended and improved.

The results presented in this current article were based on the set of papers reviewed by Cruz et al. (2011), and extended by another 48 articles found using our improved search strategy. The new information and the resulting discussions represent significant improvements to the preliminary results. We believe this article contributes to a more comprehensive understanding of the landscape of the personality research in software engineering.

3. Review methodology

A *mapping study* (Arksey & O'Malley, 2005) about the influence and role of personality in software engineering was performed. This work is classified as a secondary study, as it is a review of primary studies. The conceptual work on conventional systematic literature reviews (SLR) (Petticrew & Roberts, 2006) and Kitchenham and Charters' (2007) guidelines for performing SLR in a context of software engineering were followed to design and execute this mapping study. Our goal was to collect evidence that could be used to guide research and practice, so we consider this mapping study to be part of the evidence-based software engineering effort (Kitchenham, Dybå, & Jørgensen, 2004).

Cooper (1988) has developed a taxonomy to benefit several audiences in assessing and organizing knowledge synthesis. This taxonomy classifies literature reviews based on six characteristics: focus of attention; goal of the synthesis; perspective on the literature; coverage of the literature; organization of the perspective; and intended audience. According to this taxonomy, our mapping study is classified as shown in Table 1.

3.1. Research questions

In this article, the following central research question guided the search and selection processes of our review:

RQ: What is the current state of academic research on personality in software engineering?

The following specific research questions were used to guide the data extraction, analysis, synthesis and presentation of results:

- *RQ1: What research topics are investigated in the research on personality in software engineering?*
- *RQ2: What research methods are used in the studies, and in what context (academic or industrial) are they applied?*
- *RQ3: What personality tests are administered in the studies, and to what type of participants (professionals or students)?*
- *RQ4: What are the main effects or outcomes of personality on the tasks and processes of software engineering?*
- *RQ5: What are the most common personality types and traits of software engineers?*

We provide the rationale and motivation for each research question below, before presenting the results in Section 4.2.

3.2. Inclusion and exclusion criteria

We searched the existing literature looking for papers reporting three types of studies (inclusion criteria):

1. Studies that addressed the influence of personality in software engineering.
2. Studies of personality in software engineering that addressed traits or type theories.
3. Empirical studies (based on direct observation or experiments) and theoretical studies (based on an understanding of the theme from experience or reference to other works), industrial experience reports, and literature reviews.

We excluded papers that met any of the following exclusion criteria:

1. Written in a language other than English.
2. Not accessible on the Web.

Table 1
Review classification according to Cooper's taxonomy.

Review classification	
Focus	Research outcomes Research methods Theories Practices or applications
Goal	Integration Identification of central issues
Perspective	Neutral representation
Coverage	Exhaustive with selective citation
Organization	Conceptual
Audience	General scholars Practitioners

Table 2
Manual Sources.

ACM computing surveys
ACM transactions on software engineering and methodology
Communications of the ACM
IEE proceedings software (now IET software)
IEEE software
IEEE transactions on software engineering
Empirical software engineering journal
Software practice and experience
Information and software technology
Journal of systems and software
Int. conference on software engineering
Int. conference on evaluation and assessment of software engineering
Int. symposium on empirical software engineering and measurement

3. Invited papers, keynote speeches, workshop reports, books, theses and dissertations.
4. Incomplete documents, drafts, slides of presentations and extended abstracts.
5. Studies addressing areas of computer science that are clearly not software engineering (e.g., database systems, human–computer interaction, information systems, computer networks, artificial intelligence, etc.).
6. Studies addressing other individual characteristics (e.g., behavior, cognition, competence, abilities, roles, etc.).

3.3. Data sources and search strategy

The search for relevant papers was carried out in three stages that can be summarized as follows. In Stage One, a comprehensive and exhaustive search for relevant papers was performed based on the automated and manual search procedures described below on papers that were published before 2011 (no limit on the start date was imposed). The first stage resulted in a set of selected relevant papers that was used as input to the second stage. In Stage Two, a "snowball" search was performed on all references of each relevant paper selected in Stage One. In Stage Three, we merged the results of the previous stages to consolidate a set of papers to analyze in the present review. Fig. 1 summarizes this search strategy.

A manual search was performed on all relevant journals and conference proceedings (Table 2). We looked at the titles and abstracts of all papers in each source included in Table 2, using the same procedure as that applied to the list of papers returned in the automatic search. The searches were compatible, making the process more easily auditable and replicable. The use of manual search is supported in the literature on systematic reviews to complement and extend the coverage of automatic searches (Kitchenham & Charters, 2007; Petticrew & Roberts, 2006).

In particular, manual searches allow researchers to find published articles that are available in manual sources but have not yet been indexed by the search engines used in the automatic search. In our review, the manual search was responsible for three papers that were not found in the automatic search and would have been missed if the manual procedure had not been used.

The automatic search was performed using five search engines and indexing systems (Table 3). Although there is a high redundancy rate when using multiple search engines (nearly 40% of the total of 19,243 papers were retrieved by more than one search engine), we needed to perform the search on several engines to assure that no important articles were missed. Automatic searches were conducted for the entire test of every paper on all engines but Scopus, which did not perform full-text search. For this engine, the search was performed on the papers' titles and abstracts.

The search string used in the automatic search was constructed from four search terms: software engineering; software engineer; software team; personality. To increase the sensitivity of the search (higher coverage), synonyms for each term were provided (Table 4). We added plurals, as various search engines did not treat wildcards as expected. To build the string, synonyms were joined with OR and the set of synonyms for each term were joined with AND (Fig. 2).

The amount of synonyms added to a search string usually represents a trade-off between sensitivity and precision. A large number of synonyms increases sensitivity but also increases the number of non-relevant papers returned, thus decreasing the precision of the search. In the present review, we opted to increase

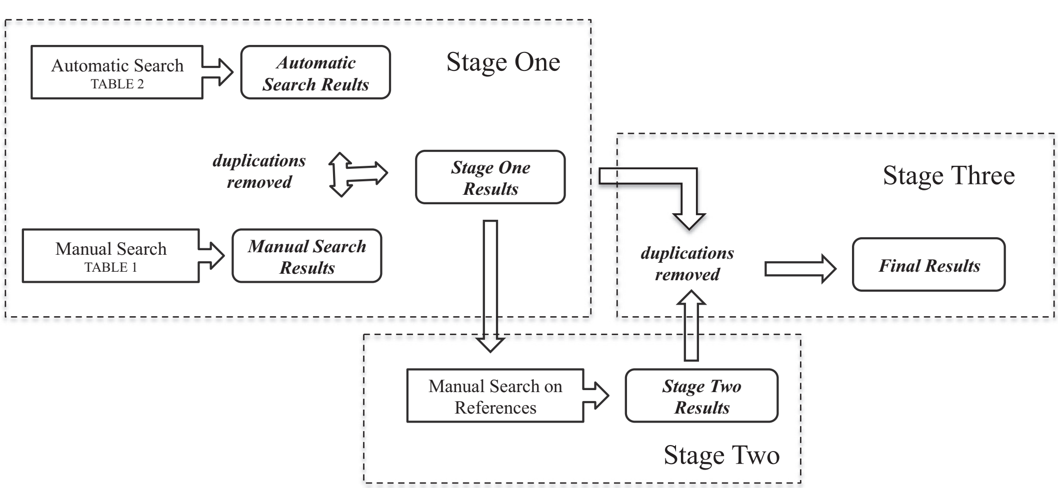

Fig. 1. Stages of the search strategy.

Table 3
Automatic sources.

ACM digital library – http://portal.acm.org
IEEEXplore digital library – http://www.ieeexplore.ieee.org/Xplore
Elsevier ScienceDirect – http://www.sciencedirect.com
Scopus – http://www.scopus.com
El Compendex – http://www.engineeringvillage.com

Table 4
Search string construction.

Keyword	Search terms
Software engineering	Software engineering
	Software development
	Software system
	Software life cycle
	Software process
	Software maintenance
	Software project
	Extreme programming
Software engineer	Software developer
	Software professional
	Software engineer
Software team	SE team
	IS team
	Programming team
	Team project
	Project team
	Pair programming
Personality	Personality
	Psychological typology
	Psychological types
	Temperament type
	Traits

sensitivity at the expense of having to deal with a large number of irrelevant papers.

In the second stage of the search, we used the set of papers resulting from the manual and automatic searches as input and performed a "snowball" search on all selected papers. We performed this search by manually examining the title and abstract of all references in each selected paper, following the same procedure used in the manual and automatic searches.

3.4. Study selection

The selection of relevant papers in each search procedure (manual, automatic, and "snowball") was performed in two steps (Pre-Selection and Selection), as prescribed by the guidelines for systematic reviews (Kitchenham & Charters, 2007). Fig. 3 summarizes the study selection process.

In the pre-selection step, the papers resulting from the search procedure were analyzed by manually looking the title and abstract and excluding all papers that were clearly not relevant to the research questions. This step was performed by four researchers working in pairs on each automatic and manual source, and also on each paper analyzed in the second stage (the "snowball" search). The guidelines recommend that pre-selection should err on the side of caution, i.e., if researchers do not agree, the paper should be included. This recommendation was followed in this step: to avoid rejecting potentially relevant papers, disagreements between members in each pair were solved by adding all selected papers from each member.

In the selection step, inclusion and exclusion criteria were applied to the set of papers resulting from the pre-selection step. Two researchers worked independently on the entire set of papers resulting from each search procedure. The researchers read the abstract, introduction and conclusion of each paper that passed the pre-selection step, then applied the inclusion and exclusion criteria (see Section 3.2) to the potentially relevant papers. Another researcher that did not participate in the original group worked to solve discrepancies; when an agreement was not possible, the differences were solved in a consensus meeting.

3.4.1. Considerations about search quality and efficiency

The quality and efficiency of search strategies used in systematic reviews are important issues in any evidence-based discipline. These concepts are usually operationalized by two indexes: *sensitivity* and *precision*. *Sensitivity* is conceptually defined as the ratio between the number of relevant studies retrieved and the total number of existing relevant studies in the literature. *Precision* is defined as the ratio between the number of relevant studies retrieved and the total number of studies retrieved. An optimal strategy should achieve the highest possible values for both indexes. However, in practice, there is always a trade-off between sensitivity and precision; that is, to achieve high sensitivity (coverage) it is often necessary to sacrifice precision and vice versa.

Sensitivity is usually calculated based on a set of known relevant studies that accurately represent the total relevant studies with respect to the research questions. This set is called the Gold Standard (GS) (Dickersin, Scherer, & Lefebvre, 1994) and is used for two purposes in evidence-based research: firstly, it is assumed to be the *truth* in evaluating the sensitivity of a given search strategy, and secondly, it is used as a training sample to refine the search string (Zhang, Babar, Xu, Li, & Huang, 2011).

We used the set of all papers analyzed in the five reviews described in Section 2.2 as our GS. We tuned our automatic search strategy to achieve 100% sensitivity regarding this set of papers. In particular, we extended the search terms and added synonyms in the search string to achieve this level of sensitivity with respect to the GS, as already discussed above.

However, achieving high sensitivity comes with a price. A high-sensitivity search may retrieve many non-relevant papers, decreasing precision Zhang et al. (2011). Precisions lower than 2% have been reported in several well-known systematic reviews in software engineering Sjøberg et al. (2005), Dyba and Dingsøyr (2008), Beecham et al. (2008), and da Silva et al. (2012). For this reason, our automatic search procedure produced a very low precision rate. This was expected and is consistent with the strategy of achieving the highest possible sensitivity with respect to the GS.

```
("software engineering" OR "software development" OR "software system*" OR "software life
cycle" OR "software process" OR "software maintenance" OR "software project*" OR "extreme
programming") OR ("software developer" OR "software developers" OR "software professional" OR
"software professionals" OR "software engineer" OR "software engineers" OR "pair programmer")
OR ("software team" OR "software teams" OR "SE team" OR "SE teams" OR "IS team" OR "IS
teams" OR "programming team" OR "programming teams" OR "team project" OR "team projects"
OR "project team" OR "project teams" OR "pair programming") AND ("personality" OR
"psychological typology" OR "psychological types" OR "temperament type"OR "temperament
types")
```

Fig. 2. Search string.

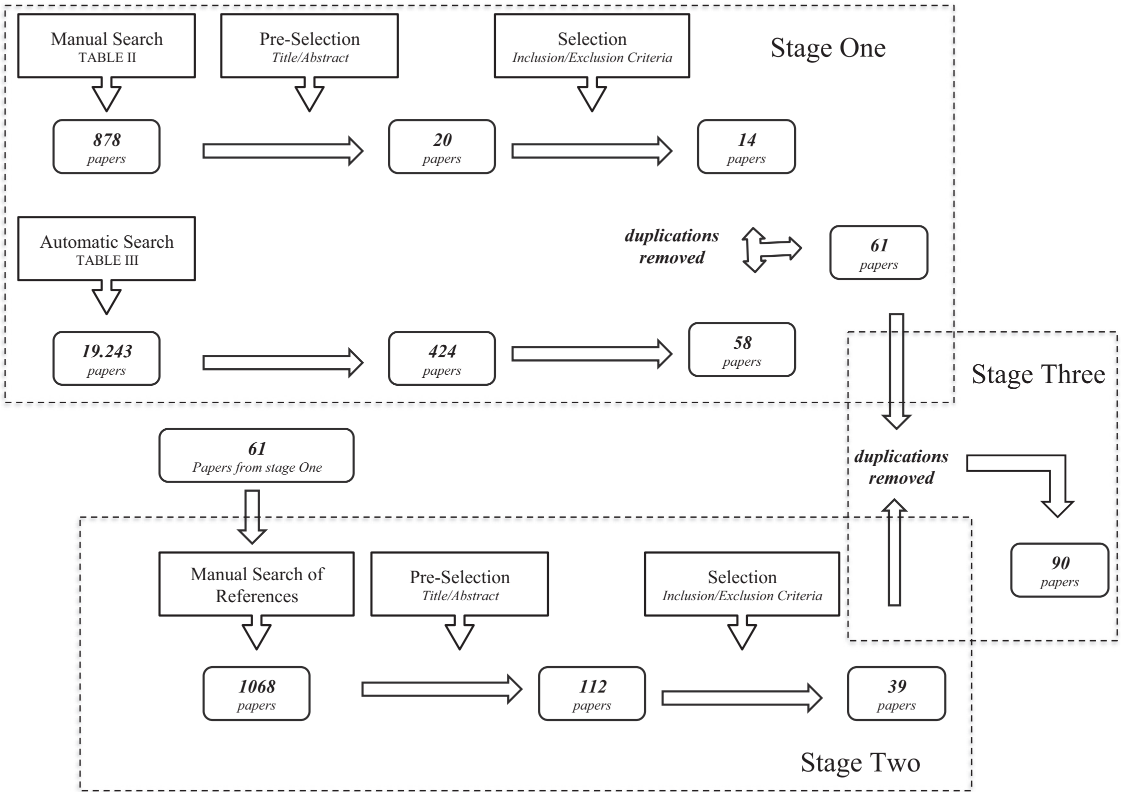

Fig. 3. Detailed description of the selection process.

Table 5
Data extracted from all papers.

Data	Description
Publication title	Title of the paper
Year of publication	Year of publication of the paper reporting the replication
Authors	Names of all authors
Research	organizations
Affiliations of the authors	
Country	Country where the organization is located
Type of study	Classified as explained in Section 4.1.4 in the following types: Empirical and Theoretical
Research question	Main research question of the paper, or research goal when no explicit question was provided
Personality test	Personality test or inventory used in the study (when applicable)
Study outcomes	Theoretical and/or empirical findings of the study

3.5. Data extraction

Table 5 shows the data extracted from all the selected papers, while Table 6 shows the complementing data extracted from empirical papers only. Each researcher worked independently to extract data from all papers, guided by an extraction form implemented in MS Excel™. In Appendix B, we present a sample extraction form for one paper.

With the support of JabRef and Mendeley, we automatically extracted the following objective data from the papers: publication title, authors' names, year of publication, authors' affiliations, and country. As this information was automatically extracted, no inconsistencies in the extraction were found.

Two researchers worked in the data extraction process for all other data fields to improve the accuracy of the extraction and, consequently, the reliability of the results. A third researcher reviewed disagreements in the extracted data. When an agreement could not be reached, the differences were solved in a consensus meeting.

Table 6
Data extracted from empirical papers.

Research method	According to Easterbrook et al. (2007): experiments or quasi-experiments, case-studies, survey, ethnography, and action research
Subject of investigation	Professional, student or both
Study outcomes related to personality type	The findings related to the personality traits or types of the subjects of investigation of the study

3.6. Synthesis of results

The main goal in synthesizing the results was to group studies into categories representing a common research topic addressed by each paper in the category. The research topic categories were established using a form of thematic analysis (Dixon-Woods & et al., 2005). The thematic analysis was performed by the first author and thoroughly scrutinized by the second author, who is familiar with qualitative synthesis methods. The thematic analysis followed the method described by Dixon-Woods et al. (2005) in the following steps:

- Variable identification: using the research question extracted from the primary study during data extraction, the variables related in the research question were identified and isolated.
- Grouping variables: similar variables, i.e., those expressing the same concept or construct (regardless of operationalization) were grouped in themes.

- Naming groups: after all variables have been grouped, the groups were given names that better represented the research topic.

The remaining data components extracted from each study were integrated using frequency charts. Frequencies of each component in the categories were presented using column charts. Bubble charts were built to relate two or more components, thus providing several combinations of the data.

4. Results

In this section, we present the results of the mapping study along with answers to our research questions. We also provide information mappings to link the answers of the individual research questions. Our results naturally fall into two groups. The results presented in Section 4.1 deal with the descriptive nature of individual studies, while the results related to research questions (RQ1–RQ5) deal with the conceptual aspects of the studies and the interactions between them.

4.1. Descriptive information about the primary studies

In this section, we summarize the descriptive information about the primary studies reviewed in this mapping study. In Appendix A, we present individual information about each paper and the complete references for all 90 publications.

4.1.1. Temporal view of publications

The chart in Fig. 4 depicts the temporal distribution of the primary studies. Just over 72% (68/96) of the primary studies were published after 2002. This indicates that although the human factor in software engineering has been acknowledged and researched since the 1970s, research focusing on personality is much more recent, with the vast majority of the studies developed in the last decade.

The peaks reached from 2006 to 2010 are partly explained by the novel agile software development practice of pair programming. If we look closely at the software engineering (SE) topics described in the papers published from 2006 (Section 4.2.1), we can see that papers addressing agile pair programming practice became increasingly important in studies of personality in software engineering, as depicted in Fig. 5.

The first detected study on pair programming is from the early 90s (Pocius, 1991) and aimed to "discover the relations between personalities and attitudes toward pair programming." From 2006, more studies were published that examined the influence of personality in the context of pair programming.

4.1.2. Researchers and organizations

In the 90 papers reviewed, 153 distinct co-authors were found who were affiliated to 89 distinct organizations. Table 7 presents the co-authors that have published at least two papers and their corresponding organization.

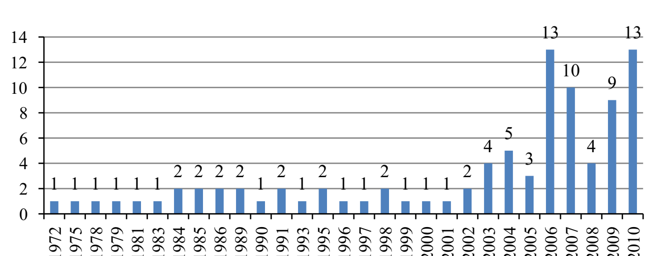

Fig. 4. Temporal distribution of primary studies.

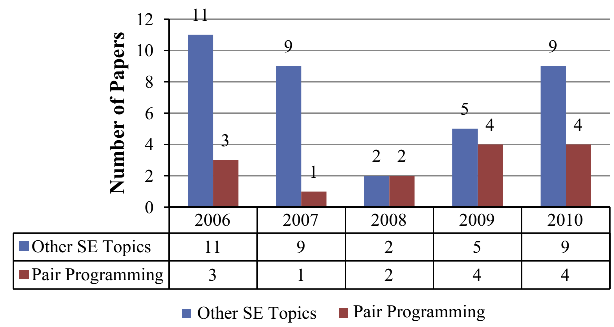

Fig. 5. Temporal analysis of pair programming topic.

4.1.3. Geographic distribution of publications

It is important to note that Fig. 6 shows papers originating from 21 different countries, as it is vital to examine research from different social and organizational cultures. However, just over half of the studies (46/90) were written by at least one researcher from the United States of America (USA). This indicates that more studies from different countries are needed to account for cultural and social differences that may have an effect on the research findings.

4.1.4. Types of studies

We classified the studies according to the following two types:

- Theoretical: a study is theoretical when it proposes models or theories about the effect of personality on different constructs in software engineering without performing an empirical study.
- Empirical: these studies aim to obtain and analyze empirical data about personality and its effect on different aspects of software development.
- Among the selected studies, 83% (75/90) were empirical and 17% (15/90) were theoretical.

4.2. Answers to the research questions

In this section, we present the results of our mapping study that answer the five research questions described in Section 3.1. The results of empirical and theoretical studies are presented separately, to allow a clear distinction between empirically-based evidence and untested propositions and models.

4.2.1. Results from the empirical studies

The answers to the research questions in this section were obtained by extracting and combining data from the 75 empirical papers.

RQ1: What research topics are investigated in the research on personality in software engineering?

The research topics were categorized using the procedure explained in Section 3.4. Since any given study can be related to more than one research question, the sum of the percentages in the chart of Fig. 7 is greater than 100%.

We will briefly describe each research topic with regard to how they are influenced by personality as investigated in the primary studies.

Pair Programming is a practice mainly used associated with agile methodologies in which two programmers work collaboratively on

Table 7
Most frequent researchers involved in the studies and their affiliation.

Author	Institution	Country	# of papers
Capretz, L. F.	Western University	Canada	7
Choi, K. S.	Manhattan College	USA	4
Rodríguez-Díaz, A.	Autonomous U. of Baja California	Mexico	3
Walz, D. B.	U. of Texas	USA	3
Mendes, E.	U. of Auckland	New Zealand	3
Burch, G.	U. of Auckland; U. of Sydney	New Zealand; Australia	3
Licea, G.	Autonomous U. of Baja California	Mexico	3
Karn, J. S.	U. of Sheffield	UK	3
Hannay, J. E.	U. of Oslo	Norway	3
Grundy, J.	Swinburne U. of Technology	Australia	3
Castro, J. R.	Autonomous U. of Baja California	Mexico	3
Wynekoop, J. L.	Florida Gulf Coast U.; U. of Texas	USA	3
Layman, L.	North Carolina State U.	USA	3
Martínez, L. G.	Autonomous U. of Baja California	Mexico	3
Salleh, N.	International Islamic U. Malaysia; U. of Auckland	Malaysia; New Zealand	3
Acuña, S. T.	Autonomous U. of Madrid	Spain	3
Cowling, A. J.	U. of Sheffield	UK	2
Deek, F. P.	New Jersey Institute of Technology	USA	2
Ahmed, F.	United Arab Emirates U.	United Arab Emirates	2
Edwards, H. M.	U. of Sunderland	UK	2
Williams, L.	North Carolina State U.	USA	2
Angelis, L.	Aristotle U. of Thessaloniki	Greece	2
Gallivan, M. J.	Georgia State U.	USA	2
Juristo, N.	Polytechnic U. of Madrid	Spain	2
Rutherfoord, R. H.	Southern Polytechnic State U.	USA	2
Syed-Abdullah, S.	MARA U. of Technology; University of Sheffield	Malaysia; UK	2
McDonald, S.	U. of Sunderland	UK	2

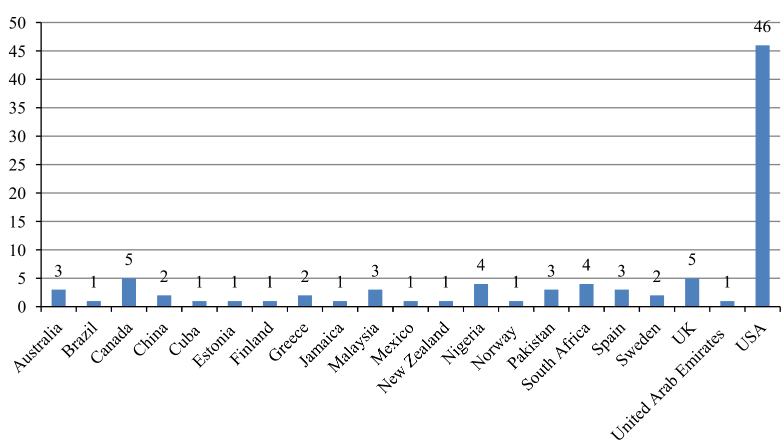

Fig. 6. Geographic distribution of papers.

the same code and sharing the same computer. The potential benefits for software development include defect reduction and improvement in quality and communication. The research on this topic investigates the influence of personality on this practice; including many studies that seek to determine whether pairs of individuals with different personality types perform better then uniform pairs.

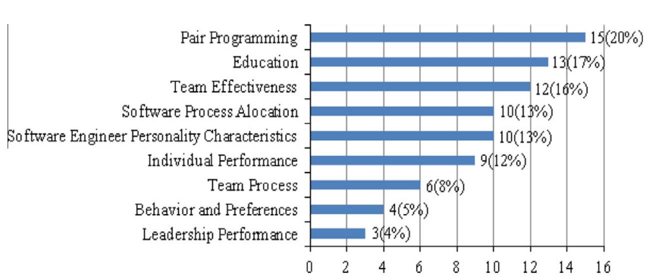

Fig. 7. Research topics.

The influence of personality in the *Education* of a software engineer is seen by some researchers as a key factor for successful learning. Researchers are therefore seeking to understand how teaching practices and styles can be tailored to specific student personalities to improve the efficacy of the learning process.

Team Effectiveness in software engineering is investigated from the standpoint of how it can be affected by personality interactions among team members. These studies examine the impacts of personality factors in the composition of teams, conflicts resolution, job satisfaction and project success in general.

Software Process Allocation considers that software development work is diversified and multidisciplinary, involving tasks such as analysis, design, coding and testing. The assignment of an individual to a role in a software team is therefore seen as a critical factor for project success, as certain personality traits may be better suited to perform certain tasks.

Software Engineering Personality Characteristics is a research topic that assigns personality profiles to software engineers according to various personality measures. Some researchers also

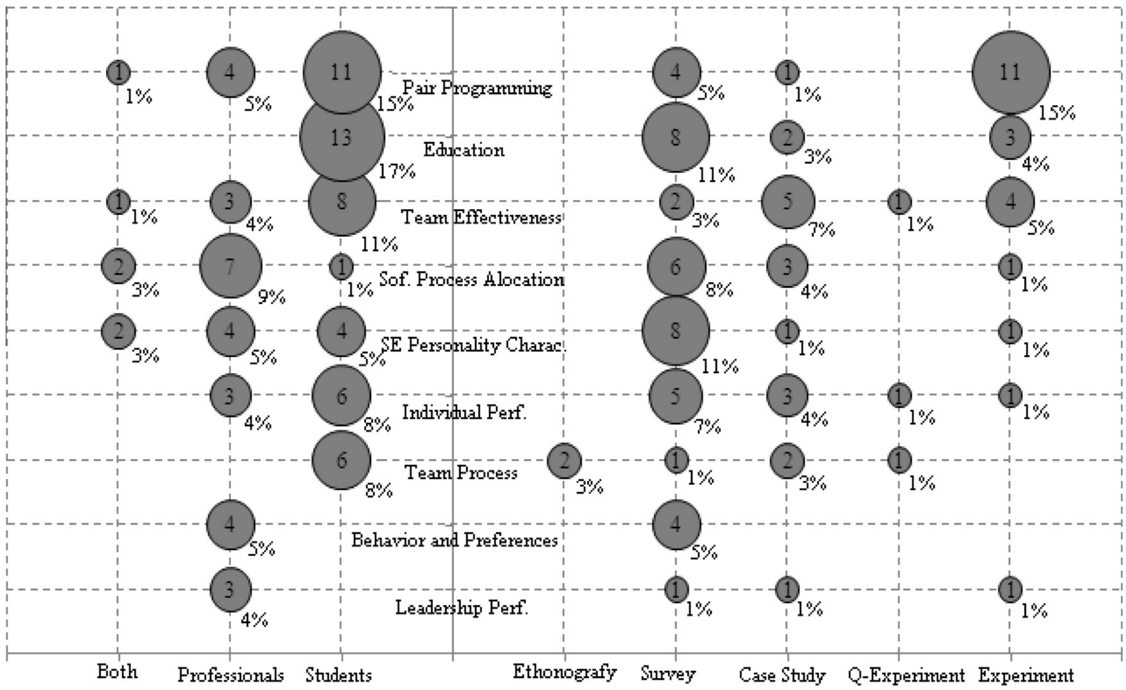

Fig. 8. Mapping of research topics, methods and subjects.

investigate whether the personality characteristics of software engineers differ from those of the general population.

The influence of personality on the *Individual Performance* of a software engineer is investigated to understand which personality traits or types are ideal for the various work tasks involved in software engineering. Researchers that study this topic consider that personality can influence the outcomes of a software project to a greater extent than technology, process or tools.

The research topic of *Team Process* considers the variables of team dynamics like cohesion, conflict resolution and communication. It examines the impacts of personality factors in group behavior variables that are mediators of team performance.

Behavior and Preferences investigates the fact that the attitudes and preferences of software engineers are influenced by their personality traits or types. This research seeks to understand how this relationship is defined, to provide a general comprehension of how these attitudes and preferences explain work styles, habits and preferences for tools and processes.

Leadership Performance is a research topic explored from the perspective of how personality traits or types affect leadership behavior and how this behavior influences individual and team satisfaction, and consequently, the success of the project.

RQ2: What research methods are used in the studies, and in what context (academic or industrial) they are applied?

The question RQ2 aims to identify the research methods used in the primary studies, as well as the contextual setting of each study as defined by the type of participants.

In Fig. 8, we present a unified mapping with two visions: the relationship between research methods and research topics, and the relationship between types of participants (subjects of investigation) and research topics. In the chart, the size of the circle indicates how many articles were identified for each relationship; the number of studies is indicated in the center of the circle. As any given study can apply more than one research method and participant type, the sum of the percentages in the chart is greater than 100%.

In the global context, the distribution between types of participants in the studies is not balanced: 60% (45/75) use students, while only 35% (26/75) use professionals. However, it is important to note that the research topic of *Education* takes place in a completely academic context. When the data from the *Education* topic variable (13 papers) is removed from the total amount, the distribution between types of participants in the studies is more balanced: 52% (32/62) use students and 42% (26/62) use professionals. Furthermore, studies using professionals and those using students are equally distributed among the research topics. The case study and survey studies are well distributed among the research topics, while experimental papers are very focused on pair programming, which may indicate the difficulty of using experiments on other topics.

In Fig. 9, we present a graph illustrating participant types used in the various research methods. While research using case studies and surveys shows a balanced distribution between students and professionals, the experiment-based research is far more centered on using students than professionals, which may be due to the difficulty of using experiment-based research techniques with professionals.

While ethnography studies are not usually applied in academic context, the two primary studies that applied ethnography described their student participant teams working on real software development projects for real clients, which may indicate the use of ethnography in an industrial context.

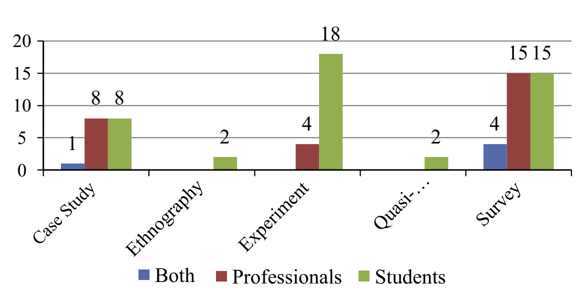

Fig. 9. Relation between participant type and research method.

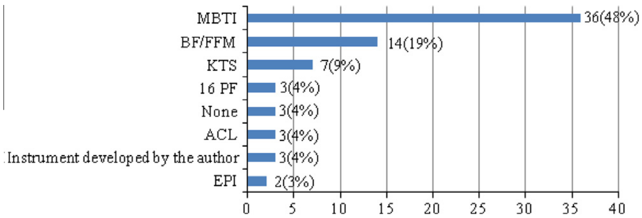

Fig. 10. Personality tests.

RQ3: What personality tests are administered in the studies, and to what type of participants (professionals or students)?

Personality tests used in the primary studies are shown in Fig. 10. Tests that were used in only one study are not shown in the chart; these include the Rotter Internal–External Control Scale (Rotter I–E), Rathus Assertiveness Schedule (RAS), Thurstone Temperament Schedule (TTS), Rosenberg's Self-Esteem Scale (RSES), Judge's Generalized Self-Efficacy Scale (JGSE), Levenson's Locus of Control Scale (LLC), Personality Type A/B, Self-Monitoring of Expressive Behavior (SMEB), Hostility Inventory (HI), Type A Behavior, Minnesota Multiphasic Personality Inventory (MMPI), Personal Resilience Questionnaire (PRQ) and Personality Research Form (PRF).

The Myers-Brigg Type Indicator (MBTI) is used in 48% (36/75) of the studies. Combined with the 9% (7/75) that used the Kersey Temperament Sorter (KTS), 57% (43/75) of the studies used tests based on Jung's Personality Types Theory. Tests based on the Big Five (BF) theory and Five Factor Model (FFM) (Costa & McCrae, 1992), such as the NEO-PI test, were used in 19% (14/75) of the studies. Three studies (4%) did not administer any personality tests. The remaining studies used various other specific tests.

In Fig. 11, we present the combination of the research topics from RQ1 and the participant types from RQ2 with the personality test used in the study. The chart shows that the MBTI has been used in studies that examine all identified research topics, which demonstrates its ubiquity of use. Tests based on the Big Five theory were used in studies on all research topics except for education. While education is the most studied research topic among scholars using the MBTI test, pair programming is the most frequent focus for researchers who favor the Big Five theory. Furthermore, software process allocation is the research topic with the most balanced distribution of the various personality tests. With regard to participant type, the tests are relatively well balanced between students and professionals with the exception of the MBTI, which is the principal test used in education studies and therefore is more frequently administered in an academic context.

RQ4: What are the main effects or outcomes of personality on the tasks and process of software engineering?

To answer this question, we present a brief descriptive synthesis of the results found concerning the influence of personality on software engineering tasks and processes, organized according to research topic (Fig. 11). The reader must be aware that we did not investigate consistency among the operational definitions of the constructs used as outcomes in the studies. In fact, most studies did not characterize these definitions in a way that would allow for such an investigation. Furthermore, we did not check for consistency with regard to the populations used in the studies; again, this was due to a lack of information in most of the studies. Discrepancies among the results presented in the synthesis may therefore be the result of differing operations of the evaluated outcomes, differing contexts or populations, or variability in the version of the tests applied and analysis of the results. To facilitate the understanding of the synthesis, in the subsequent analysis, we put the personality test together with the reference of the study. For example, S2-BF/FFM refers to study S2, which used a test based on BF/FFM theory.

Pair Programming most of the studies on this topic tested whether the composition of the pair influenced the performance of the pair, especially with respect to differences or similarities in personality. Eight studies presented conflicting evidences: three found that pairs with distinct personality types (heterogeneous) perform better than homogeneous pairs (S10-MBTI, S15-KTS, S54-MBTI), four found no relationship between heterogeneous and homogeneous pairs (S3-BF/FFM, S5-BF/FFM, S38-BF/FFM,

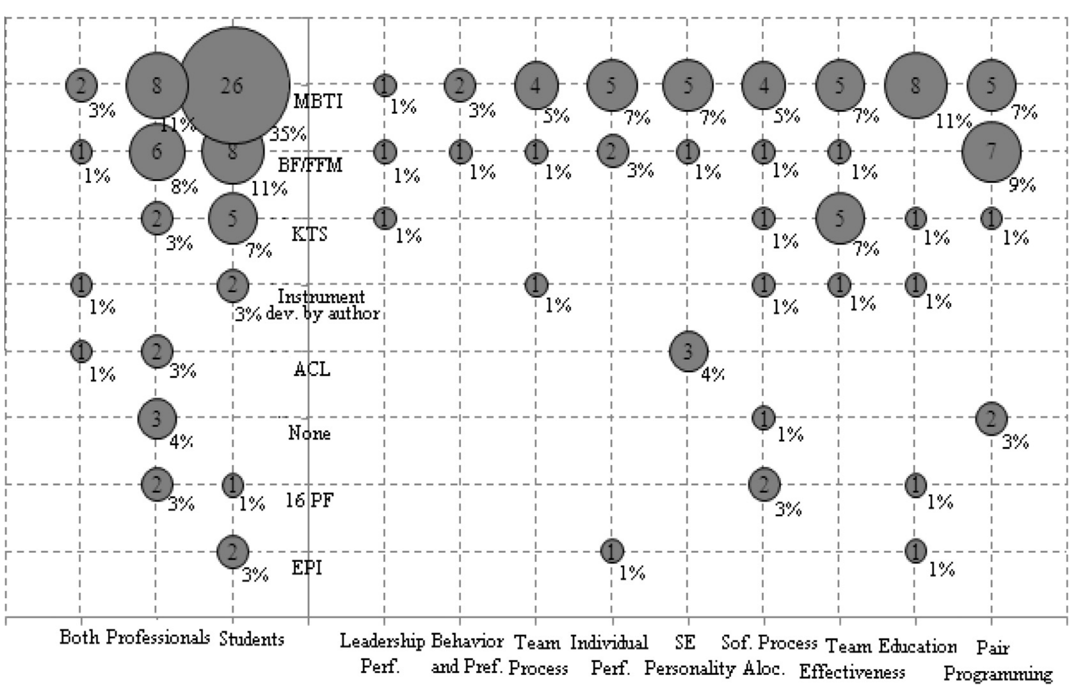

Fig. 11. Mapping of research topics, personality tests and subjects.

S39-BF/FFM), and one study showed that personality clash is one of the most serious problems in pair programming (S17-None), but did not report which personality types cause more clashes. Two studies agree about the influence of personality in pair performance: one study found that "individual personality traits, considered in isolation from partner's personality, does not have significant consequences to pair-programming performance in general" (S74-BF/FFM), while the other study argues that personality has a low predictive capacity for performance of pairs when compared with expertise, task complexity, and country (S2-BF/FFM). In contrast, another study that surveyed a group of professional programmers found that personality is the primary factor influencing pair programming productivity (S83-None). Two studies show that personality has no influence on communication, satisfaction and trust (S9-MBTI), or on pair compatibility (S9-MBTI; S43-MBTI), with exception of one study that "indicated a difference on the sensing-intuition scale can help predict highly compatible pairs" (S54-MBTI). With respect to collaboration between members of a pair, one study found that variability in personality improves collaboration (S4-BF/FFM), while another found that personality type had little effect on the changes in perceptions of the pair collaboration (S20-MBTI).

Education: most of the studies that conducted research on this topic discussed the influence of personality in academic performance. The 10 studies presented conflicting evidences: three found that individual personality increases or helps to predict academic performance (S20-MBTI; S73-TTS; S75-Various), six reported that personality traits were not a significant factor in predicting academic success (S53-MBTI; S42-Instrument developed by the author; S44-MBTI; S55-MBTI and Personality Type A/B; S57-MBTI; S85-16 PF), and one study showed that students with certain traits in a software design course performed significantly better than students possessing other personality traits (S72-MBTI). Other studies provided evidence that adapting educational activities to the personality of a student is an effective way to optimize academic success (S21-MBTI, S27-KTS), and that curricula could be designed to develop understanding of the strengths and weaknesses of various approaches and the problems of interaction between individuals with different personality styles (S66-MBTI).

Team Effectiveness: the studies showed evidence that personality is related to project success (S30-Instrument developed by the author), to code quality, and to individual satisfaction (S8-BF/FFM). Other studies showed that personality diversity in teams is not directly related to team effectiveness (S24-KTS, S28-MBTI, S30-Instrument developed by the author), while five studies showed the contrary: S22-KTS found that diversity is important in the early stages of the software project and that this importance decreases as the project evolves and the team matures; S36-MBTI stated that heterogeneous teams are "optimum" when solving unstructured tasks, while homogeneous teams are "optimum" when solving structured tasks; the author of S23-KTS strongly believes that using a personality inventory can help establish not only heterogeneous groups, but also create groups who understand and appreciate the strengths and weaknesses of all team members; S40-MBTI found that groups with diverse problem-solving styles received higher grades on their design project; and finally, in S48-MBTI, a technique was applied to identify the effective team personality-types composition, revealing that a balance of personality types resulting in individual team members complementing each other, which increased team performance.

In addition, three studies found evidence of the influence of a team's personality composition on overall team performance. S19-MBTI showed that teams with predominantly introverted members experience lower effectiveness due to communication problems; the author of S62-KTS believes that software development teams can be strengthened by using personality inventories to form groups; and S79-MBTI recommends "organizations that desire to develop effective teams to analyze the personality-type compositions of these groups and help team members understand their own personal attributes as well as appreciate the contribution of the other team members".

Software Process Allocation: the ten empirical studies that research this topic investigated the relationship between personality and performance with respect to technical roles in the software process. From the results of surveys, case studies and experiments of personality types of software engineers, all studies propose how to map personality types to technical roles (S12-MBTI, S24-KTS, S29-16PF, S47-BF/FFM, S49-MBTI, S59-None, S60-16PF, S61-MBTI, S64-Instrument developed by the author, S86-MBTI).

Software Engineer Personality Characteristics: articles researching this topic investigated the personality profiles of software engineers in an attempt to identify possible common personality characteristics. Two studies found that the most common type is ISTJ, the second most frequently reported type is ESTJ and the least common type is INFJ (S12-MBTI, S77-MBTI). These studies also concluded that the type distribution of software engineering students in the United States is different from type distribution found in a general population of that country.

On the other hand, with a Brazilian population, one study found that the most common types are ISFP, INTP and ESTP, while the least common are ENTP, ESTJ and ENTJ (S81-MBTI). The study also concluded that the type distribution of software engineering students in Brazil is different from the type distribution found in a general population of that country.

Another study found that IS professionals differed from adult population norms on 19 of the 24 scales examined (S81-ACL). While two studies found differences in personality between top developers and unexceptional developers (S58-BF/FFM, S69-ACL), one study showed no significance difference between programmers and trainees (S80-MBTI and MMPI). Another study found that engineering programs and software engineering programs have similar type distributions, showing that similar occupations attract particular and similar types (S65-MBTI). S71-ACL presented a methodology for identifying the traits and characteristics of top performing IT personnel. Finally, using the Personality Research Form (PRQ), one study found that it is generally believed that IT professionals possess higher needs for achievement, cognitive structure, and endurance (S78-PRF).

Individual Performance: articles researching this topic investigated the best fit between specific software engineering tasks and personality. Two of the studies identified that a particular personality type is positively related to performance of the task of code review (S25-MBTI) and tester ability in exploratory testing (S51-MBTI). Another concluded that matching the personality of an individual to the demands of the job position results in increased performance (S63-MBTI). Using a combination of the results of several specific tests, one study found that personality exhibits a predictive relationship with object-oriented programming (S37-Various). On the other hand, two studies showed no significant relationship between personality and programming performance (S6-FFM, S50-BF/FFM), and a third study found no significant difference in personality between exceptional (high performance) and unexceptional programmers (S14-MBTI). Finally, it was found that personality predicts programming proficiency (S90-MBTI) and affects individual job satisfaction (S32-PRQ) with regard to challenges at work and feelings of respect and appreciation.

Team Process: The six empirical studies that conducted research on this topic provide evidence that relationships exist between team member personalities and team process, in particular team cohesion (S8-BF/FFM, S19-MBTI, S88-MBTI, S89-MBTI), communication, and conflict (S40-MBTI). On the other hand, no personality factor was significantly correlated with any one component of team process (S30-Instrument developed by the first author).

Behavior and Preferences: The four empirical studies classified in this topic provide evidence that personality influences software engineer attitudes toward judgment and decision-making (S7-FFM); that personality factors affect how individuals react to or prefer techniques, methods and processes (S35-None); that systems analysts tend to be technically oriented (S34-MBTI); and that "there are no differences between users and systems personnel on the extroversion–introversion, thinking–feeling, perception–judgment dimensions, however users tend to be more intuitive than systems personnel" (S46-MBTI).

Leadership Effectiveness: one study examining this topic showed that there is no pattern linking the leadership behavior of the project manager with personality traits (S11-MBTI), while another stated that the personality of the project manager is related to project success (S16-FFM). According to S24-KTS, heterogeneity between project manager and team members regarding some aspects of personality is related to team performance.

RQ5: What are the personality traits of software engineers?

Our answer to this research question is based on the results of the personality tests most frequently used in software engineering: MBTI, BF/FFM and KTS, as presented in Fig. 11. Among the 43 studies that applied MBTI and KTS, only 27 provided the results for the sample of participants (S10, S12, S14, S15, S19, S21, S27, S28, S34, S36, S41, S44, S46, S51, S57, S61, S63, S65, S66, S72, S77, S79, S80, S81, S86, S88, and S89). As for the studies that used tests based on BF/FFM, only 5 out of 14 presented results for a sample of participants (S8, S38, S47, S50, and S74). It is therefore possible to conduct a meta-analysis to integrate these results. However, considering the poor description of and great differences in the research contexts, the lack of details about the statistical treatment of the data, and several methodological problems related to test application and interpretation (as those discussed by McDonald and Edwards (2007) in Section 2.2), a meta-analysis of the data would not provide reliable results.

The results of MBTI types and KTS temperaments are not unanimous among the studies either in a professional context (Table 8) or in an academic context (Table 9). However, the predominant temperaments in KTS tests are SJ (Guardian) and NT (Rational). For MBTI tests, most studies report the prevalence of type Introversion (I) over Extroversion (E); Sensation (S) over Intuition (N), Thinking (T) over Feeling (F), and Judging (J) over Perception (P).

It is not possible to synthesize the results of BF/FFM traits in a uniform manner because of the variety of tests applied and the differences in the presentation of the results among the studies, as presented in Table 10. The "low," "average" and "high" scores are assigned based on a trait's numerical value, this varies with each personality trait and test. It is therefore impossible to identify dominant personality traits based on the evidence of the five factors, since the tests are different and the results show no pattern of the most frequent personality traits either in the professional or in the academic context.

An analysis of the most prevalent personality types among software engineers should consider the various existing roles of the profession, each with different ability and skill requirements. Because software engineering comprises systems analysis, design, programming, testing, and maintenance of software systems, according to Capretz and Ahmed (2010), the theory behind personality types implies that each one is likely to affect some phases of the software life cycle more than others. Therefore, a wide variety of personality types is beneficial for software engineering. However, it is not possible to answer this research question in a way that allows us to analyze each role of software engineering due to the difficulty of interpreting the data provided by primary studies.

Table 8
Top personality types of professional software engineers.

Primary study ID	Sample size	Personality distribution of the subjects		
		Types (most common)	Temperaments (KTS)	Other results
S12	100	ISTJ = 24%	SJ = 45%	I: 57% vs. E: 43%
			NT = 26%	S: 67% vs. N: 33%
			SP = 22%	T: 81% vs. F: 19%
		ESTJ = 15%	NF = 7%	J: 58% vs. P: 42%
S14	20	INTJ = 35%	NT = 55%	I: 90% vs. E: 10%
		ISTJ = 15%	SJ = 20%	N: 65% vs. S: 35%
		ISTP = 15%	SP = 15%	T: 85% vs. F: 15%
			NF = 10%	J: 65% vs. P: 35%
S34	37	ISTJ = 35%	SJ = 76%	I: 60% vs. E: 40%
		ESTJ = 30%	NT = 21%	S: 81% vs. N: 19%
			SP = 12%	T: 89% vs. F: 11%
			NF = 6%	J: 86% vs. P: 14%
S61	38	ENTP = 16%	NT = 34%	E: 61% vs. I: 39%
		INFJ = 13%	NF = 29%	N: 63% vs. S: 37%
		INTJ = 10%	SJ = 21%	T: 63% vs. F: 37%
		ESTJ = 10%	SP = 16%	P: 66% vs. J: 34%
S79	22	INTJ = 23%	NT = 41%	I: 64% vs. E: 36%
		ENTJ = 18%	SJ = 32%	N: 59% vs. S: 41%
		ISTJ = 18%	NF = 18%	T: 64% vs. F: 36%
			SP = 9%	J: 86% vs. P: 14%
S36	20		ST = 55%	
			NT = 25%	
			SF = 10%	
			NF = 10%	
S46	60		NT = 47%	E: 53% vs. I: 47%
			ST = 43%	S: 53% vs. N: 47%
			SF = 8%	T: 87% vs. F: 13%
			NF = 2%	J: 73% vs. P: 27%

The inseparability of the evidence with regard to the various roles of software engineering, as mentioned above, may have caused the divergence of results on software engineer personality; however, other factors may also have caused disparity in the results, such as the variety of applied studies in different countries and cultures, the age and sex of respondents and the version of the personality test, as well as the methodology of application.

4.2.2. Synthesis of theoretical studies

The fifteen theoretical studies are summarized in Table 11. A large majority of the theoretical studies based their model construction or propositions on Jung's Personality Type Theory—in particular, on the MBTI test—although they did not administer the test.

Here we present a brief descriptive synthesis of the proposed theories on the influence of personality on tasks and processes in software engineering, organized according to the research topics (S79-MBTI).

Software Process Allocation: the three theoretical studies that examine this research topic made the following propositions: S1-MBTI proposed a mapping between the job requirements and interpersonal skills (soft skills) required of each role in software development, and the corresponding personality types that most suit these roles. S13-MBTI proposed a model showing the influence of personality variables in the four stages of the software development process. Finally, S56-None attempted to build a theoretical basis to investigate the context of the different tasks required in the systems design process, aligning them with descriptions of the skills and personalities required of the members of design information systems teams.

Table 9
Personality types of students of software engineering.

Primary study ID	Sample size	Personality distribution of the subjects		
		Types (most common)	Temperaments (KTS)	Other results
S10	128	ISTJ = 21% ESTJ = 15% INFP = 7% INTP = 7%	SJ = 47% SP = 20% NT = 17% NF = 16%	I: 55% vs. E: 45% S: 67% vs. N: 33% T: 63% vs. F: 37% J: 57% vs. P: 43%
S19	15	INTJ = 40% INTP = 20% ENTJ = 20%	NT = 80% NF = 20% SJ = 0% SP = 0%	I: 67% vs. E: 33% N: 100% vs. S: 0% T: 80% vs. F: 20% J: 73% vs. P: 27%
S27	85	ESTJ = 19% ISTJ = 18% ISFJ = 13% ESFJ = 8%	SJ = 63% NF = 16% NT = 13% SP = 9%	E: 53% vs. I: 47% S: 69% vs. N: 31% T: 58% vs. F: 42% J: 72% vs. P: 28%
S28	33	ISTJ = 21% ESTJ = 18% ISFJ = 15%	SJ = 63% NT = 19% SP = 5% NF = 0%	I: 59% vs. E: 41% S: 72% vs. N: 28% T: 66% vs. F: 34% J: 78% vs. P: 22%
S49	12	ENTJ = 42% ISTJ = 25%	NT = 58% SJ = 25% SP = 8% NF = 0%	E: 58% vs. I: 42% N: 58% vs. S: 42% T: 92% vs. F: 8% J: 92% vs. P: 8%
S65	68	ISTJ = 19% INTP = 13% ESTP = 12% ESTJ = 12%	SJ = 37% NT = 31% SP = 22% NF = 9%	I: 56% vs. E: 44% S: 59% vs. N: 41% T: 79% vs. F: 21% J: 51% vs. P: 49%
S66	92	ESFP = 20% INFP = 15% ESFP = 15% ENFP = 12%	SP = 47% NF = 35% NT = 12% SJ = 6%	E: 65% vs. I: 35% S: 53% vs. N: 47% F: 74% vs. T: 26% P: 84% vs. J: 16%
S72	85	ISTJ = 28% ENTJ = 18% ISFJ = 9%	SJ = 42% NT = 32% SP = 14% NF = 12%	I: 63% vs. E: 37% S: 61% vs. N: 39% T: 69% vs. F: 31% J: 72% vs. P: 28%
S77	128	ISTJ = 21% ESTJ = 15% INFP = 7% INTP = 7%	SJ = 47% SP = 20% NT = 16% NF = 16%	I: 55% vs. E: 45% S: 68% vs. N: 32% T: 61% vs. F: 39% J: 56% vs. P: 44%
S80	59	INTP = 10% ISTJ = 10% INTP = 8% ESTJ = 8%	NT = 30% SJ = 27% NF = 24% SP = 19%	I: 61% vs. E: 39% N: 54% vs. S: 46% T: 58% vs. F: 42% J: 54% vs. P: 46%
S81	68	ISTJ = 19% INTP = 13% ESTP = 12% ESTJ = 12%	SJ = 37% NT = 32% SP = 22% NF = 9%	I: 56% vs. E: 44% S: 59% vs. N: 41% T: 79% vs. F: 21% J: 51% vs. P: 49%
S88	25	ENTJ = 32% INTJ = 20% INFJ = 16%	NT = 52% NF = 24% SJ = 12% SP = 12%	E: 52% vs. I: 48% N: 76% vs. S: 24% T: 64% vs. F: 36% J: 88% vs. P: 12%
S15	70		SJ = 60% NF = 21% SP = 10% NF = 9%	
S21	67			I: 54% vs. E: 46% N: 82% vs. S: 18% T: 63% vs. F: 37% J: 76% vs. P: 24%
S44	58			I: 63% vs. E: 37% N: 51% vs. S: 49% T: 77% vs. F: 23% J: 54% vs. P: 46%
S51	71			E: 63% vs. I: 37%
S57	88			I: 50% vs. E: 50% S: 68% vs. N: 32% F: 57% vs. T: 43% J: 52% vs. P: 48%
S89	44			I: 50% vs. E: 50% N: 59% vs. S: 41% T: 66% vs. F: 34% J: 75% vs. P: 25%

Education: five theoretical studies were classified as having a focus on this research topic. Three of these (S33-MBTI, S67-MBTI, S68-MBTI) suggest that a form of education with specific activities based on the personality types of the students is more effective to help students develop dexterity in using both familiar and new approaches. Another study found that culture and personality type affect success in computer science training (S45-MBTI). Finally, S71-MBTI suggests that the use of Zig-Zag model, a group problem-solving model based on the Myers-Briggs Type Indicator, in computing courses is beneficial to integrate a collaborative approach throughout the curriculum.

Individual Performance: two theoretical studies were classified as conducting research on this topic. S52-None tries to develop a research framework that allows the software engineering community to assess modeling methods and identify the positive and negative influences of individual personalities. The other study (S76-Various) is a review of studies investigating the relations between personality dimensions and programming aptitude and achievement; this study concludes that there is little relation between Jungian dimensions and programming aptitude and achievement.

Software Engineer Personality Characteristics: the purpose of the two theoretical studies classified to this topic was to present the results of studies that identify the MBTI preferences of the participants. Both studies (S76-Various; S84-MBTI) agree that the most frequent MBTI personality types found among computer programmers are the ISTJ, INTJ, and INTP types. Compared to the general population, these papers conclude that programmers are measured to have increased tendencies toward introversion, intuition, and thinking preferences.

SE Personality Test: two theoretical studies conducted research on this topic. One study (S18-MBTI and 16 PF) presented a critical appraisal of 13 empirical studies that used personality tests in software engineering. The study analyzed the reliability of the test instruments and the evidence of adequate application and interpretation of results, and provided recommendations to potential participants, practitioners, and researchers. The other study (S87-MBTI and BF/FFM) is a comparison between MBTI and FFM and suggests that "FFM not only provides better measures for all factors that are measured by MBTI, but it also allows to assess Neuroticism, an important personality trait that is of interest to researchers studying work groups."

The unique theoretical study in the *Job Retention* research topic suggests that certain personality types influence decision paths in relation to changes in employment in software engineering. The unique theoretical study of *Leadership Performance* suggests a personality type that best fits the role of a project manager.

5. Discussion

Our goal in this review was to plot a general landscape of the body of work on personality in software engineering research. In this section, we also discuss our results and implications for research and practice of software engineering.

Table 10
Personality traits distribution.

Study ID	Sample size	Subject type	BF/FFM test	Trait score (median scores – M)				
				O	C	E	A	N
S8	105	S	NEO FFI	Average (29.43)	Low (29.76)	Average (32.38)	Low (27.97)	High (18.11)
S38	118	S	IPIP NEO	0 < M < 20	20 < M < 40	20 < M < 40	40 < M < 60	40 < M < 60
S47	72	P	Big Five	(52.62)	(61.12)	(50.51)	(51.25)	(62.52)
S50	128	S	NEO FFI	Low (28)	High (32)	Average (30)	Low (29)	Average (18)
S74	50	S	NEO PI	(111)	(107)	(109)	(112)	(83)

O = Openness to experience, C = Conscientiousness, E = Extraversion, A = Agreeableness, N = Neuroticism.

Table 11
Summary of theoretical studies.

ID	Research topic	Personality test
S1	Software process allocation	MBTI
S13	Software process allocation	MBTI
S56	Software process allocation	None
S33	Education	MBTI
S45	Education	None
S67	Education	MBTI
S68	Education	MBTI
S71	Education	MBTI
S52	Individual performance	None
S76	Software engineer personality characteristics and individual performance	Various
S84	Software engineer personality characteristics	MBTI
S87	SE personality test	MBTI and FFM
S18	SE personality test	MBTI and 16 PF
S26	Job retention	MBTI
S31	Leadership performance	MBTI

5.1. Implications for research and practice

As far as we are aware, this is the first attempt to review the literature on personality in software engineering in a broad and systematic way. Our findings show a great concentration of empirical research on the effect of personality in pair programming, as well as on the influence of personality in education, team effectiveness, software process allocation, software engineer personality characteristics and individual performance. The other research topics received much less attention. One of the implications of these findings for the academic community is that this area of research has become open to many research opportunities (Capretz, 2014). However, experiment replications are necessary if we want to consolidate a consistent body of knowledge that can guide future research and influence the practice of software engineering.

In terms of practical uses for software engineering professionals, direct application of the results presented in the studies must be conducted with care. Conflicting evidence suggests that the research field is not mature, and that direct application of the methods and instruments used in the studies may not produce the desired effects. In particular, we agree with McDonald and Edwards (2007) that the interpretation of the results of personality tests and its practical working implications are not straightforward and require properly trained professionals. We also believe that a careful analysis of the context in which the research was conducted is necessary to assess the possibility of generalizing the results to other settings.

5.2. Limitations of this review

The most common limitations in a systematic review are the possible biases introduced in the selection process and inaccuracies in the data extraction. These are also the main possible limitations of this review. The research protocol was based on well-established guidelines incorporating measures to prevent selection bias. Another common limitation to systematic review and mapping is the difficulty in finding all relevant articles. The combination of a multi-engine automatic search, a manual search of relevant publications and a secondary "snowball" search of the primary studies' references improves the coverage of the selection process, reducing possible biases. A multistage selection process was used, and the researchers recorded reasons for inclusion and exclusion of studies at each stage, as recommended by Kitchenham and Charters (2007). Search and selection process in all stages were performed by at least two researchers, and conflicts in the selection process were solved either by careful examination by a third party or in consensus meetings.

Some studies addressed personality trait or personality type using the term cognitive style. Although the distinction between these terms is far from clear (Bishop-Clark, 1995), the majority of authors call the Myers-Briggs Type Indicator (MBTI) a personality measure, while others call it a cognitive style measure (Carland & Carland, 1990; Miller & Yin, 2004). Based on the results of Carver and Scheier (1988), this work considers that cognitive style and personality trait are not the same construct. However, some software engineering papers that use the term cognitive style referring to personality trait may not have been found. Similarly, papers that did not use any of the search terms defined in the protocol may not have been found, e.g. studies that used a personality test but did not explicitly use the word personality or its synonyms.

6. Conclusions

We presented a systematic mapping study of the research literature on the influence of personality in software engineering. From over 19,000 papers found in extensive manual, automatic and secondary searches, we selected 90 relevant papers. These studies investigated nine broad themes with respect to the influence of individual personality in software engineering: Pair Programming, Team Effectiveness, Individual Performance, Software Process Allocation, Behavior and Preference, Education, Project Manager Effectiveness, Personality Test Application, and Job Retention. Only the last two topics were not addressed by empirical studies.

The empirical studies revealed the 7 different personality tests that are most commonly used. While MBTI largely dominates the studies, tests based on FFM, in particular NEO-PI, are becoming more popular. Divergence of the most common personality types among software engineers seems to indicate differences in academic and industrial and cultural contexts, or specialization to one of the many tasks involved in software development.

This article extends the preliminary results of our EASE'2011 workshop paper (Cruz et al., 2011) by adding a new research question, employing a more comprehensive search strategy, and presenting a deeper analysis and discussion about the results.

6.1. Future work

We are currently working on two topics using data extracted from the studies selected in this mapping study:

- Evolution of software engineer personality profiles: data extraction to identify the most common software engineer personality types and traits did not result in type convergence. In the future, we will analyze the evolution of software engineer personality profiles to identify patterns over the years that would explain the changing proportions in software engineer personality types, such as those proposed by Varona, Capretz, Piñero, and Raza (2012).
- Aggregation of findings based on different contexts: our findings were aggregated according to research topic. In the future, we will perform a more comprehensive data aggregation according to the two main types of context: individual vs. team and academic vs. organizational. We hope this investigation will identify patterns that would indicate variation of personality effects in these different contexts.

Furthermore, we believe that performing deeper analyses using meta-ethnography will provide a better understanding of the relationships between research topics. To exercise this method of qualitative synthesis, we performed a meta-ethnography on the six papers in the category of Team Process (da Silva, Cruz, Gouveia, & Capretz, 2013). The results provided evidence that meta-ethnography is an adequate method to synthesize studies that use different research methods.

6.2. Final considerations

Considering the importance of the effect of human factors in many aspects of software engineering, the amount of research on the effects and influences of personality in the field is relatively small. The evidence is weak and in many cases inconclusive. More research is required if we want results that can influence the practice of software development.

This research area is multidisciplinary in nature, combining academic scholarship from both software engineering and humanities, in particular psychology. However, most studies are conducted by researchers from the area of software engineering, without collaboration with psychologists or researchers from related areas (or, if this collaboration exists, it is not explicitly mentioned in the papers). We argue that collaboration is imperative when studying this on this research theme.

Finally, this research should be extended to incorporate deeper analysis and comparison between studies, with particular emphasis on understanding the effect size of the outcomes encountered.

Acknowledgments

Shirley Cruz was a visiting graduate student at Western University, where she received a scholarship from the Government of Canada – Department of Foreign Affairs, Trade and Development, Emerging Leaders in the Americas Program (ELAP). Fabio Q.B. da Silva holds a research grant from the Brazilian National Research Council (CNPq), process #314523/2009-0; this article was written while Prof. Fabio da Silva was on sabbatical leave at the University of Toronto, receiving a CAPES research grant process # 6441/10-6. Luiz Fernando Capretz holds an NSERC Discovery Grant.

Appendix A. Selected papers

In this appendix, we present the summarized information extracted from each paper and the complete list of references to the entire set of 90 selected papers.

Summary of information about the papers

Study ID	Research topic	Study type	Research method	Test
S1	Software process allocation	Theoretical	–	MBTI
S2	Pair programming	Empirical	Experiment	BFFM-100*
S3	Pair programming	Empirical	Experiment	IPIP-NEO*
S4	Pair programming	Empirical	Case study	BFFM-100*
S5	Pair programming	Empirical	Survey and experiment	IPIP*
S6	Individual performance	Empirical	Quasi-experiment	NEO PI*
S7	Behavior and preferences	Empirical	Survey	IPIP-50*
S8	Team performance and team process	Empirical	Quasi-experiment	NEO FFI*
S9	Pair programming	Empirical	Experiment	MBTI
S10	Pair programming	Empirical	Experiment	MBTI
S11	Leadership performance	Empirical	Case study	MBTI
S12	Software engineer personality characteristics and software process allocation	Empirical	Survey	MBTI
S13	Software process allocation	Theoretical	–	MBTI
S14	Individual performance	Empirical	Survey	MBTI
S15	Pair programming	Empirical	Experiment	KTS
S16	Leadership performance	Empirical	Experiment	NEO-FFI*
S17	Pair programming	Empirical	Survey	None
S18	SE personality test	Theoretical	–	MBTI 16 PF
S19	Team process	Empirical	Ethnography	MBTI
S20	Pair programming and education	Empirical	Survey	MBTI
S21	Education	Empirical	Survey	MBTI
S22	Team performance	Empirical	Experiment	KTS

Selected papers (*continued*)

Study ID	Research topic	Study type	Research method	Test
S23	Team performance	Empirical	Experiment	KTS
S24	Leadership performance; software process allocation and team performance	Empirical	Survey	KTS
S25	Individual performance	Empirical	Case study	MBTI
S26	Job retention	Theoretical	–	MBTI
S27	Education	Empirical	Survey	KTS
S28	Team performance	Empirical	Experiment	MBTI
S29	Software process allocation	Empirical	Experiment	16 PF
S30	Team performance and team process	Empirical	Survey	Instrument developed by the author
S31	Leadership performance	Theoretical	–	MBTI
S32	Individual performance	Empirical	Survey and case study	PRQ
S33	Education	Theoretical	–	MBTI
S34	Behavior and preferences	Empirical	Survey	MBTI
S35	Behavior and preferences	Empirical	Survey	Rotter I–E Rathus assertiveness schedule
S36	Team performance	Empirical	Case study	MBTI
S37	Individual performance	Empirical	Survey	Various[#]
S38	Pair programming	Empirical	Experiment	IPIP-NEO[*]
S39	Pair programming	Empirical	Experiment	IPIP-NEO[*]
S40	Team performance and team process	Empirical	Case study	MBTI
S41	Team performance	Empirical	Case study	KTS
S42	Education	Empirical	Survey	Instrument developed by the author
S43	Pair programming	Empirical	Experiment	MBTI
S44	Education	Empirical	Survey	MBTI
S45	Education	Theoretical	–	None
S46	Behavior and preferences	Empirical	Survey	MBTI
S47	Software process allocation	Empirical	Case study	Big Five[*]

Selected papers (*continued*)

Study ID	Research topic	Study type	Research method	Test
S48	Team performance	Empirical	Case study	MBTI
S49	Software process allocation	Empirical	Case study	MBTI
S50	Individual performance	Empirical	Case study	NEO-FFI[*]
S51	Individual performance	Empirical	Experiment	MBTI
S52	Individual performance	Theoretical	–	None
S53	Education	Empirical	Experiment	MBTI
S54	Pair programming	Empirical	Experiment	MBTI
S55	Education	Empirical	Experiment	MBTI Type A and B personality type questionnaire
S56	Software process allocation	Theoretical	–	None
S57	Education	Empirical	Survey	MBTI
S58	Software engineer personality characteristics	Empirical	Experiment	LIWC[*]
S59	Software process allocation	Empirical	Case study	None
S60	Software process allocation	Empirical	Survey	16 PF
S61	Software process allocation	Empirical	Survey	MBTI
S62	Team performance	Empirical	Experiment	KTS
S63	Individual performance	Empirical	Survey	MBTI
S64	Software process allocation	Empirical	Survey	Instrument developed by the author
S65	Software engineer personality characteristics	Empirical	Survey	MBTI
S66	Education	Empirical	Survey	MBTI
S67	Education	Theoretical	–	MBTI
S68	Education	Theoretical	–	MBTI
S69	Software engineer personality characteristics	Empirical	Survey	Adjective check list
S70	Education	Theoretical	–	MBTI
S71	Software engineer personality characteristics	Empirical	Case study	Adjective Check List
S72	Education	Empirical	Experiment	MBTI

(*continued on next page*)

Selected papers (*continued*)

Study ID	Research topic	Study type	Research method	Test
S73	Education	Empirical	Case study	Thurstone temperament schedule
S74	Pair programming	Empirical	Experiment	NEO PI*
S75	Education	Empirical	Survey	Various&
S76	Software engineer personality characteristics and individual performance	Theoretical	–	Various
S77	Software engineer personality characteristics	Empirical	Survey	MBTI
S78	Software engineer personality characteristics	Empirical	Survey	Personality research form
S79	Team performance	Empirical	Case study	MBTI
S80	Software engineer personality characteristics	Empirical	Survey	MBTI and minnesota multiphasic personality inventory
S81	Software engineer personality characteristics	Empirical	Survey	MBTI
S82	Software engineer personality characteristics	Empirical	Survey	Adjective check list
S83	Pair programming	Empirical	Survey	None
S84	Software engineer personality characteristics	Theoretical	–	MBTI
S85	Education	Empirical	Case study	16 PF
S86	Software process allocation	Empirical	Survey	MBTI
S87	SE personality test	Theoretical	–	MBTI and FFM
S88	Team process	Empirical	Ethnography	MBTI
S89	Team process	Empirical	Case study	MBTI
S90	Individual performance	Empirical	Survey	MBTI

* Personality test based on the Big Five/Five Factor Model theory.
# Rosenberg's Self-Esteem Scale, Judge's Generalized Self-Efficacy Scale, Levenson's Locus of Control Scale and Eysenck's Neuroticism Scale.
& EPI, Crowne-Marlowe Social Desirability Scale, Self-Monitoring of Expressive Behavior, Hostility Inventory, Type A Behavior.

List of studies

S1 – Capretz LF, Ahmed F (2010). Making sense of software development and personality types. IT Professional 12:6–13. http://dx.doi.org/10.1109/MITP.2010.33.

S2 – Hannay J, Arisholm E, Engvik H, Sjoberg DIK (2010). Effects of personality on pair programming. IEEE Transactions on Software Engineering 36:61–80.

S3 – Salleh N, Mendes E, Grundy J, Burch GSJ (2009). An empirical study of the effects of personality in pair programming using the five-factor model. In ESEM'09: Proceedings of the 3rd International Symposium on Empirical Software Engineering and Measurement. IEEE, pp. 214–225.

S4 – Walle T, Hannay JE (2009). Personality and the nature of collaboration in pair programming. In ESEM'09: Proceedings of the 3rd International Symposium on Empirical Software Engineering and Measurement. IEEE, pp. 203–213.

S5 – Chao J, Atli G (2006). Critical personality traits in successful pair programming. In AGILE'06: Proceedings of the Conference on AGILE 2006. IEEE, pp. 89–93.

S6 – Darcy D, Ma MJ (2005). Exploring individual characteristics and programming performance: Implications for programmer selection. In HICSS'05: Proceedings of the 38th Annual Hawaii International Conference on System Sciences. IEEE, pp. 314a.

S7 – Feldt R, Angelis L, Torkar R, Samuelsson M (2010). Links between the personalities, views and attitudes of software engineers. Information and Software Technology 52:611–624.

S8 – Acuna S, Gomez M, Juristo N (2009). How do personality, team processes and task characteristics relate to job satisfaction and software quality? Information and Software Technology 51:627–639. http://dx.doi.org/10.1016/j.infsof.2008.08.006.

S9 – Choi KS, Deek FP, Im I (2009). Pair dynamics in team collaboration. Computers in Human Behavior 25:844–852. http://dx.doi.org/10.1016/j.chb.2008.09.005.

S10 – Choi KS, Deek FP, Im I (2008). Exploring the underlying aspects of pair programming: The impact of personality. Information and Software Technology 50:1114–1126. http://dx.doi.org/10.1016/j.infsof.2007.11.002.

S11 – Strang KD (2007). Examining effective technology project leadership traits and behaviors. Computers in Human Behavior 23:424–462. http://dx.doi.org/10.1016/j.chb.2004.10.041.

S12 – Capretz LF (2003). Personality types in software engineering. International Journal of Human–Computer Studies 58:207–214. http://dx.doi.org/10.1016/S1071-5819(02)00137-4.

S13 – Bishop-Clark C (1995). Cognitive style, personality, and computer programming. Computers in Human Behavior 11:241–260.

S14 – Turley R, Bieman JM (1995). Competencies of exceptional and non-exceptional software engineers. Journal of Systems and Software 28:19–38.

S15 – Sfetsos P, Stamelos I, Angelis L, Deligiannis I (2008). An experimental investigation of personality types impact on pair effectiveness in pair programming. Empirical Software Engineering 14:187–226. http://dx.doi.org/10.1007/s10664-008-9093-5.

S16 – Wang Y, Li F (2009). How does project managers' personality matter? Building the linkage between project managers' personality and the success of software development projects. In OOPSLA'09: Proceedings of the 24th ACM SIGPLAN Conference Companion on Object oriented Programming Systems Languages and Applications. ACM, pp. 867–874.

S17 – Begel A, Nagappan N (2008). Pair programming: what's in it for me? In ESEM 08: Proceedings of the Second ACM/IEEE International Symposium on Empirical Software Engineering and Measurement. ACM, pp. 120–128.

S18 – McDonald S, Edwards HM (2007). Who should test whom? Communications of the ACM 50:66–71.

S19 – Karn J, Cowling T (2006). A follow up study of the effect of personality on the performance of software engineering teams. In ISESE'06: Proceedings of the ACM/IEEE International Symposium on Empirical Software Engineering. ACM, pp. 232–241.

S20 – Layman L (2006). Changing students' perceptions: An analysis of the supplementary benefits of collaborative software development. In CSEET'06: Proceedings of the 19th Conference on Software Engineering Education & Training. IEEE, pp. 159–166.

S21 – Layman L, Cornwell T, Williams L (2006). Personality types, learning styles, and an agile approach to software engineering education. In SIGCSE'06: Proceedings of the 37th SIGCSE Technical Symposium on Computer Science Education. ACM, pp. 428–432.

S22 – Pieterse V, Kourie DG, Sonnekus IP (2006). Software engineering team diversity and performance. In SAICSIT'06: Proceedings of the 2006 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists on IT Research in Developing Countries. South African Institute for Computer Scientists and Information Technologists, Republic of South Africa, pp. 180–186.

S23 – Rutherfoord R (2006). Using personality inventories to form teams for class projects: a case study. In SIGITE'06: Proceedings of the 7th Conference on Information Technology Education. ACM, pp. 9–14.

S24 – Gorla N, Lam YW (2004). Who should work with whom? Communications of the ACM 47:79–82.

S25 – Cunha ADD, Greathead D (2007). Does personality matter?: an analysis of code-review ability. Communications of the ACM 50:109–112.

S26 – Mourmant G, Gallivan M (2007). How personality type influences decision paths in the unfolding model of voluntary job turnover: an application to IS professionals. In SIGMIS CPR'07: Proceedings of the ACM SIGMIS CPR Conference on Computer Personnel Research: The global Information Technology Workforce. ACM, pp. 134–143.

S27 – Galpin V, Sanders I, Chen P-yu (2007). Learning styles and personality types of computer science students at a South African university. In ITiCSE'07: Proceedings of the 12th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education. ACM, pp. 201–205.

S28 – Miller J, Yin Z (2004). A cognitive-based mechanism for constructing software inspection teams. IEEE Transactions on Software Engineering 30:811–825.

S29 – Acuña ST, Juristo N (2004). Assigning people to roles in software projects. Software: Practice and Experience 34:675–696. http://dx.doi.org/10.1002/spe.586.

S30 – Peslak A (2006). The impact of personality on information technology team projects. In SIGMIS CPR'06: Proceedings of the ACM SIGMIS CPR Conference on Computer Personnel Research: Forty Four Years of Computer Personnel Research: Achievements, Challenges & the Future. ACM, pp. 273–279.

S31 – Brewer J (2005). Project managers: can we make them or just make them better? In SIGITE'05: Proceedings of the 6th Conference on Information Technology Education. ACM, pp. 167–173.

S32 – Gallivan MJ (2004). Examining IT Professionals' Adaptation to Technological Change: The Influence of Gender and Personal Attributes. ACM SIGMIS Database 35:28–49.

S33 – Capretz L (2002). Implications of MBTI in software engineering education. ACM SIGCSE Bulletin 34:134–137. http://dx.doi.org/10.1145/820127.820185.

S34 – Smith D (1989). The personality of the systems analyst: an investigation. ACM SIGCPR Computer Personnel 12:12–14.

S35 – Lee J, Shneiderman B (1978). Personality and programming: Time-sharing vs. batch preference. In ACM'78: Proceedings of the ACM Annual Conference 2. ACM, pp. 561–569.

S36 – White K (1984). A preliminary investigation of information systems team structures. Information & Management 7:331–335.

S37 – Cegielski CG, Hall DJ (2006). What makes a good programmer? Communications of the ACM 49:73–75.

S38 – Salleh N, Mendes E, Grundy J, Burch GSJ (2010). The effects of neuroticism on pair programming: An empirical study in the higher education context. In ESEM'10: Proceedings of the ACM/IEEE International Symposium on Empirical Software Engineering and Measurement.

S39 – Salleh N, Mendes E, Grundy J, Burch GSJ (2010). An empirical study of the effects of conscientiousness in pair programming using the five-factor personality model. In ICSE'10: Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering. ACM, pp. 577–586.

S40 – Lewis T, Smith WJ (2008). Creating high performing software engineering teams: the impact of problem solving style dominance on group conflict and performance. Journal of Computing Sciences in Colleges 24:121–129.

S41 – Linberg KR (1999). Software developer perceptions about software project failure: a case study. Journal of Systems and Software 49:177–192. http://dx.doi.org/10.1016/S0164-1212(99)00094-1.

S42 – Lutes K, Alka H, Purdum J (2009). Do introverts perform better in computer programming courses? In Proceedings of the American Society for Engineering Education Conference. pp. 12255–12263.

S43 – Sennett J, Sherriff M (2010). Compatibility of partnered students in computer science education. In SIGCSE'10: Proceedings of the 41st ACM Technical Symposium on Computer Science Education. ACM, pp. 244–248.

S44 – Werth L (1986). Predicting student performance in a beginning computer science class. In SIGCSE'86: Proceedings of the 7th Technical Symposium on Computer Science Education. ACM, pp. 138–143.

S45 – Carter L, Jernejcic L, Lim N (2007). Success in CS: Is culture a factor? FIE'07: Frontiers In Education Conference – Global Engineering: Knowledge Without Borders, Opportunities Without Passports. IEEE, pp. T3A-16–T3A-21.

S46 – Bostrom R, Kaiser KM (1981). Personality differences within systems project teams: Implications for designing solving centers. In SIGCPR'81: Proceedings of the 8th Annual Computer Personnel Research Conference. ACM, pp. 248–285.

S47 – Martínez LG, Rodríguez-Díaz A, Licea G, Castro JR (2010). Big five patterns for software engineering roles using an ANFIS learning approach with RAMSET. In MICAI'10: Proceedings of the 9th Mexican International Conference on Artificial Intelligence – Part II. Lecture Notes in Artificial Intelligence – LNAI 6438, Springer-Verlag, Heidelberg, pp. 428–439.

S48 – Omar M, Syed-Abdullah S-L (2010). Identifying effective software engineering (SE) team personality types composition using rough set approach. In 2010 International Symposium in Information Technology (ITSim) 3:1499–1503.

S49 – Martínez LG, Licea G, Rodríguez-Díaz A, Castro JR (2010). Experiences in software engineering courses using psychometrics with RAMSET. In ITiCSE'10: Proceedings of the 15th Annual Conference on Innovation and Technology in Computer Science Education. ACM, pp. 244–248.

S50 – Bell D, Hall T, Hannay J, Pfahl D, Acuna ST (2010). Software engineering group work: personality, patterns and performance. In SIGMIS-CPR'10: Proceedings of the Special Interest Group on Management Information Systems – 48th Annual Conference on Computer Personnel Research. ACM, pp. 43–47.

S51 – Shoaib L, Nadeem A, Akbar A (2009). An empirical evaluation of the influence of human personality on exploratory software testing. In INMIC'09: IEEE 13th International Multitopic Conference. IEEE, pp. 1–6.

S52 – Cunha A, Canen A, Capretz MAM (2009). Personalities, cultures and software modeling: Questions, scenarios and

research directions. In CHASE'09: Proceedings of the ICSE Workshop on Cooperative and Human Aspects on Software Engineering. IEEE, pp. 23–31.
S53 – Golding P, Facey-Shaw L, Tennant V (2006). Effects of peer tutoring, attitude and personality on academic performance of first year introductory programming students. In 36th Annual Frontiers in Education Conference. IEEE, pp. 7–12.
S54 – Williams L, Layman L, Osborne J (2006). Examining the compatibility of student pair programmers. In AGILE'06: Proceedings of the Conference on AGILE 2006. IEEE, pp. 411–420.
S55 – Corman L (1986). Cognitive style, personality type, and learning ability as factors in predicting the success of the beginning programming student. ACM SIGCSE Bulletin 18:80–89.
S56 – Leifer R, White KB (1985). Designing information system task teams. In SIGCPR'85: Proceedings of the 21th Annual Conference on Computer Personnel Research. ACM, pp. 112–117.
S57 – Whipkey K (1984). Identifying predictors of programming skill. ACM SIGCSE Bulletin 16:36–42.
S58 – Rigby PC, Hassan AE (2007). What can OSS mailing lists tell us? A preliminary psychometric text analysis of the apache developer mailing list. In MSR'07: Proceedings of the 4th International Workshop on Mining Software Repositories. IEEE, p. 23.
S59 – Young S, Edwards H, McDonald S, Thompson JB (2005). Personality characteristics in an XP team: a repertory grid study. In HSSE'05: Proceedings of the Workshop on Human and Social Factors of Software Engineering. ACM, pp. 1–7.
S60 – Moore J (1991). Personality characteristics of information systems professionals. In SIGCPR '91 Proceedings of the Conference on SIGCPR. ACM, pp. 140–155.
S61 – Teague J (1998). Personality type, career preference and implications for computer science recruitment and teaching. In ACSE'98: Proceedings of the 3rd Australasian Conference on Computer Science Education. ACM, pp. 155–163.
S62 – Rutherfoord R (2001). Using personality inventories to help form teams for software engineering class projects. In ITiCSE'01: Proceedings of the 6th Annual Conference on Innovation and Technology in Computer Science Education. ACM, pp. 73–76.
S63 – Ketler K, Smith RD (1993). Placement, performance and turnover of information systems professionals: implications for HRIS. In SIGCPR'93: Proceedings of the Conference on Computer Personnel Research. ACM, pp. 131–138.
S64 – Sodiya A, Longe H, Onashoga S, Awodele O (2007). An improved assessment of personality traits in software engineering. Interdisciplinary Journal of Information, Knowledge, and Management 2:163–177.
S65 – Capretz L (2002). Are software engineers really engineers? World Transactions on Engineering and Technology Education 1:2–4.
S66 – Carland J, Carland JAC (1990). Cognitive styles and the education of computer information systems students. Journal of Research on Computing in Education 23:1–12.
S67 – Capretz L (2006). Clues on Software Engineers' learning styles. International Journal of Computing & Information Sciences 4:46–49.
S68 – Capretz L (2003). Connecting to engineering students in the classroom. World Transactions on Engineering and Technology 2:13–18.
S69 – Clark J, Walz D (2003). Identifying exceptional application software developers: A comparison of students and professionals. Communications of the Association for Information Systems 11:137–154.
S70 – Daigle R, Doran M, Pardue JH (1996). Integrating collaborative problem solving throughout the curriculum. In SIGCSE'96: Proceedings of the 27th SIGCSE Technical Symposium on Computer Science Education. ACM, pp. 237–241.
S71 – Wynekoop JL, Walz DB (2000). Investigating traits of top performing software developers. Information Technology & People 13:186–195. http://dx.doi.org/10.1108/09593840010377626.
S72 – Ahmed F, Campbell P, Jaffar A (2010). Learning & Personality Types: A Case Study of a Software Design Course. Journal of Information Technology Education 9:16.
S73 – Alspaugh C (1972). Identification of some components of computer programming aptitude. Journal for Research in Mathematics Education 3:89–98.
S74 – Heiberg S, Puus U, Salumaa P, Seeba A (2003). Pair-programming effect on developers productivity. In XP'03: Proceedings of the 4th international Conference on Extreme Programming and Agile Processes in Software Engineering. Springer-Verlag, pp. 215–224.
S75 – Kagan DM, Douthat JM (1985). Personality and learning FORTRAN. International Journal of Man–Machine Studies 22:395–402. http://dx.doi.org/10.1016/S0020-7373(85)80046-8.
S76 – Pocius KE (1991). Personality factors in human–computer interaction: A review of the literature. Computers in Human Behavior 7:103–135. http://dx.doi.org/10.1016/0747-5632(91)90002-I.
S77 – Choi K (2006). An Analysis of Computing Major Students' Myers-Briggs Type Indicator Distribution. In Proceedings of the Information Systems Education Conference 23:1–8.
S78 – Woodruff C (1979). Personality profiles of male and female data processing personnel. ACM SIGCPR Computer Personnel 8:10–14.
S79 – Bradley J, Hebert FJ (1997). The effect of personality type on team performance. Journal of Management Development 16:337–353.
S80 – Barnes P (1975). Programmer paranoia revisited. In SIGCPR '75: Proceedings of the 13th Annual SIGCPR Conference. ACM, pp. 114–131.
S81 – Capretz L (2008). Psychological types of Brazilian software engineering students. Journal of Psychological Type 68:37–42.
S82 – Wynekoop J, Walz DB (1998). Revisiting the perennial question: are IS people different? ACM SIGMIS Database 29:62–72. http://dx.doi.org/10.1145/298752.298759.
S83 – Choi K (2007). Team programming influencing factors: A field survey. Journal of Information Technology Management 18:135–147.
S84 – Sach R, Petre M, Sharp H (2010). The use of MBTI in software engineering. 22nd Annual Psychology of Programming Interest Group. pp. 19–22.
S85 – Hostetler T (1983). Predicting student success in an introductory programming course. ACM SIGCSE Bulletin 15:40–43.
S86 – Martínez LG, Castro JR, Licea G, Rodríguez-Díaz A (2011). Towards a Fuzzy Model for RAMSET: Role Assignment Methodology for Software Engineering Teams. Soft Computing for Intelligent Control and Mobile Robotics. pp 23–41.
S87 – Balijepally V, Mahapatra R, Nerur SP (2006). Assessing personality profiles of software developers in agile development teams. Communications of the Association for Information Systems, vol. 18, article 4, pp. 54–75.
S88 – Karn JS, Syed-Abdullah S, Cowling AJ, Holcombe M (2007). A study into the effects of personality type and methodology on cohesion in software engineering teams. Behaviour & Information Technology 26:99–111. http://dx.doi.org/10.1080/01449290500102110.
S89 – Karn J, Cowling AJ (2001). An initial study of the effect of personality on group cohesion in software engineering projects. In Proceedings of the 8th International Conference on Empirical Assessment in Software Engineering. pp 155–165.

S90 – Evans G (1989). What best predicts computer proficiency? Communications of the ACM 32:1322–1327. http://dx.doi.org/10.1145/68814.68817.

Appendix B. Sample extraction form

ID	S2
Year of publication	2010
Research Question (RQ)	This paper reports on a study of the impact of the Big Five personality traits on the performance of pair programmers, also analyzing the impact of expertise and task complexity
RQ components – dependent variables	Performance of pair programmers
Components measures – indicators	Task programming (correctness, duration, methodology, extensibility, cost effectiveness, redesign, regression grade)
Research topic	PP performance
Research topic – consolidated	Pair programming
Study type	Empirical
Research method	Experiment
Subject type	Professionals
Sample size	196
Test	BFFM-100
Test-consolidated	BF/FFM
Outcomes	We found no strong indications that personality affects pair programming performance or pair gain in a consistent manner, especially when including predictors pertaining to expertise, task complexity, and country
Software engineer personality	Our sample of programmers score lower ($p < 0.0001$) on extraversion, lower on emotional stability ($p = 0.0065$), and higher on openness to experience ($p < 0.0001$) than the reference group
Obs	Job performance

References

Anderson, N., Ones, D. S., Sinangil, H. K., & Viswesvaran, C. (2002). Handbook of industrial, work and organizational psychology (Vol. 1, Personnel Psychology). Thousand Oaks, CA: Sage Publications Ltd.

Arksey, H., & O'Malley, L. (2005). Scoping studies: Towards a methodological framework. *International Journal of Social Research Methodology, 8*, 19–32.

Balijepally, V., Mahapatra, R., & Nerur, S. P. (2006). Assessing personality profiles of software developers in agile development teams. *Communications of the Association for Information Systems, 18*, 54–75. Article 4.

Beecham, S., Baddoo, N., Hall, T., Robinson, H., & Sharp, H. (2008). Motivation in software engineering: A systematic literature review. *Information and Software Technology, 50*, 860–878.

Bishop-Clark, C. (1995). Cognitive style, personality, and computer programming. *Computers in Human Behavior, 11*, 241–260.

Capretz, L. F. (2003). Personality types in software engineering. *International Journal of Human–Computer Studies, 58*, 207–214. http://dx.doi.org/10.1016/S1071-5819(02)00137-4.

Capretz, L. F. (2014). Bringing the human factor to software engineering. *IEEE Software, 31*(2), 102–104. http://dx.doi.org/10.1109/MS.2014.30.

Capretz, L. F., & Ahmed, F. (2010). Making sense of software development and personality types. *IEEE IT Professional, 12*, 6–13. http://dx.doi.org/10.1109/MITP.2010.33.

Carland, J., & Carland, J. A. C. (1990). Cognitive styles and the education of computer information systems students. *Journal of Research on Computing in Education, 23*, 1–12.

Carver, C. S., & Scheier, M. F. (1988). *Perspectives on personality*. Boston: Allyn & Bacon. doi:10.1037/030893.

Cooper, H. M. (1988). The structure of knowledge synthesis. *Knowledge in Society, 1*, 104–126.

Costa, P. T., & McCrae, R. R. (1992). *NEO PI-R professional manual*. Odessa, FL: Psychological Assessment Inc.

Cruz, S. S. J. O., da Silva, F. Q. B., Monteiro, C. V. F., & Rossilei, I. (2011). Personality in software engineering: Preliminary findings from a systematic literature review. In *EASE'11: 15th annual conference on evaluation & assessment in software engineering* (pp. 1–10). IEEE.

da Silva, F. Q. B., Cruz, S. S. J. O., Gouveia, T. B., & Capretz, L. F. (2013). Using meta-ethnography to synthesize research: A worked example of the relations between personality and software team processes. In *ACM/IEEE international symposium on empirical software engineering and measurement* (pp. 153–162). http://dx.doi.org/10.1109/ESEM.2013.11.

da Silva, F. Q. B., Suassuna, M., França, A. C. C., Grubb, A. M., Gouveia, T. B., Monteiro, C. V. F., et al. (2012). Replication of empirical studies in software engineering research: A systematic mapping study. *Empirical Software Engineering, 19*(3), 501–557. http://dx.doi.org/10.1007/s10664-012-9227-7.

Dickersin, K., Scherer, R., & Lefebvre, C. (1994). Systematic reviews: Identifying relevant studies for systematic reviews. *British Medical Journal, 309*(6964), 1286–1291.

Dixon-Woods, M., Agarwal, S., Jones, D., Young, B., & Sutton, A. (2005). Synthesising qualitative and quantitative evidence: A review of possible methods. *Journal of Health Services Research and Policy, 10*, 45–53.

Dyba, T., & Dingsøyr, T. (2008). Empirical studies of agile software development: A systematic review. *Information and Software Technology, 50*, 833–859.

Easterbrook, S., Singer, J., Storey, M-A., & Damian, D. (2007). Selecting empirical methods for software engineering research. In F. Shull, J. Singer, & D. Sjøberg (Eds.), *Guide to advanced empirical software engineering*. Springer.

Hannay, J. E., Arisholm, E., Engvik, H., & Sjoberg, D. I. K. (2010). Effects of personality on pair programming. *IEEE Transactions on Software Engineering, 36*(1), 61–80.

Kitchenham, B. & Charters, S. (2007). *Guidelines for performing systematic literature reviews in software engineering*. Technical report EBSE-2007-01. School of Computer Science and Mathematics, Keele University.

Kitchenham, B., Dybå, T., & Jørgensen, M. (2004). Evidence-based software engineering. In *26th International conference on software engineering* (pp. 273–281). IEEE.

Mcdonald, B. S., & Edwards, H. M. (2007). Who should test whom? *Communications of the ACM, 50*(1), 66–71.

Miller, J., & Yin, Z. (2004). A cognitive-based mechanism for constructing software inspection teams. *IEEE Transactions on Software Engineering, 30*(11), 811–825.

Petticrew, M., & Roberts, H. (2006). *Systematic reviews in the social sciences*. Blackwell Publishing.

Pocius, K. E. (1991). Personality factors in human–computer interaction: A review of the literature. *Computers in Human Behavior, 7*(3), 103–135. http://dx.doi.org/10.1016/0747-5632(91)90002-I.

Ryckman, R. (2004). *Theories of personality*. Belmont, California: Thomson/Wadsworth.

Salleh, N., Mendes, E., Grundy, J., & Burch, G. S. J. (2009). An empirical study of the effects of personality in pair programming using the five-factor model. In *3rd International symposium on empirical software engineering and measurement* (pp. 214–225). IEEE.

Salleh, N., Mendes, E., & Grundy, J. C. (2014). Investigating the effects of personality traits on pair programming in a higher education setting through a family of experiments. *Empirical Software Engineering, 19*(3), 714–752.

Sjøberg, D. I. K., Hannay, J. E., Hansen, O., Kampenes, V. B., Liborg, N.-K., & Rekdal, A. C. (2005). A survey of controlled experiments in software engineering. *IEEE Transactions on Software Engineering, 31*(9), 733–753.

Varona, D., Capretz, L. F., Piñero, Y., & Raza, A. (2012). Evolution of software engineers' personality profile. *ACM SIGSOFT Software Engineering Notes, 37*, 1–5.

Zhang, H., Babar, M. A., Xu, B., Li, J., & Huang, L. (2011). An empirical assessment of a systematic search process for systematic reviews. In *15th Annual conference on evaluation & assessment in software engineering* (pp. 56–65). http://dx.doi.org/10.1049/ic.2011.0007.