Personality in Software Engineering: preliminary findings from a systematic literature review

Shirley S. J. O. Cruz, Fabio Q. B. da Silva, Cleviton V. F. Monteiro, Ceei ro C. F. SMntos, CsSMvell, MC. M. t os SMntos
Center of Informatics
Federal University of Pernambuco
Recife, Brazil
{ssj,fabio,cvfm,pcfs,irms}@cin.ufpe.br

Abstract — Background – The influence of individual personalities on individual tasks and team work has been a concern in software engineering over the past 50 years. However, how to use personality analysis and what it can offer for the practice of software engineering is still subject to debate among researchers. Aim – The goal of this work is to identify the methods used, topics addressed, personality tests applied, and the main findings produced in the research about personality in software engineering. Method – We performed a systematic literature review of peer reviewed studies published between 1970 and 2010. Results – Data extracted from 42 studies shows that pair programming and team building are the most recurring research topics and MBTI is the most used test. Conclusions – Contradicting evidences were found that may 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 of personality tests in practice, it shows several opportunities for researchers.

Keywords – indivicual personality, personality tests, software engineering, systematic literature review.

I. INTRODUCTION

The influence of individual personalities on individual tasks and team work has been a concern in software engineering from the 1960s (Weinberg 1998; Lee & Shneiderman 1978; Shneiderman 1980). The quest for reliable methods and instruments to predict individual performance on certain tasks (Da Cunha & Greathead 2007), to build effective and motivated teams (Gorla & Lam 2004), or, in general, to find the "best person for the job" (Capretz & Ahmed 2010) are among the central goals of the research about personality in software engineering.

However, how to use personality analysis and what it can offer for the practice of software engineering is still subject of a warm debate among researchers (Hardiman 1997;Kerth, Coplien & Weinberg 1998). This is not a surprise given that personality is one of the most complex constructs in social sciences and personality analysis one of the most difficult tasks in psychology (Costa & McCrae 1992). In fact, McDonald & Edwards (2007) found methodological faults in the use of personality tests in several studies in software engineering that cast doubts about the validity of the results.

Nevertheless, used appropriately, the study of personality conducted using psychometrics and personality tests can assist individuals to their work through a better understanding of preferred styles for individual working, communication, learning, management, being managed, and team-working. Indeed, reliable models for predicting individual performance or building effective teams can be important tools to improve the practice of software engineering.

Despite five decades of research efforts producing diverse and relevant results, there is no comprehensive and systematic review of the studies about personality in software engineering. Given the complexity of this research topic, the diversity of theoretical traditions related to the study of personality, and the variety of personality tests available, a mapping of the published studies is important to summarize and integrate findings, and to identify opportunities for future research.

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, the research method, the type of subjects (students or professionals), and, when applicable, the personality tests used. Besides, we attempted to integrate the results that show effects of personality in individual or team performance, although this integration was not always possible due to the differences among the studies.

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

II. BACKGROUND AND RELATED WORK

There is no consensus in psychology on the definition of personality and a deeper discussion of this subject is out of the scope of this work. Nevertheless, we found important to use a definition to clearly define the focus of this review and to guide the review team during the process of inclusion and exclusion of primary studies. According to Ryckman (2004), personality is "the dynamic and organized set of characteristics possessed by a person that uniquely influences his or her cognitions,

motivations, and behaviors in various situations". This definition is general enough to allow the inclusion of studies covering a wide range of personality theories and research methods. Besides, the definition clearly separate personality from other constructs like cognition, motivation, and behavior, which are not the central interest of this review.

The study of personality has an abundance of theoretical traditions in the field of psychology, including traits, types, behavioral, and psychoanalytic theories. Of those traditions, traits and types theories are among the most used in organizational psychology (Anderson et al. 2002) and in the studies about personality in software engineering. This review focuses on these two traditions.

Traits are defined 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" (American Psychiatric Association 2000). The traits theories assume that traits are stable over time, differ between individuals, and influence behavior. Personality types are related to the classification of people with respect to psychological differences among them. Types distinguish from traits in that the latter comes in different levels or degrees, whereas types are discrete.

Most of the studies about 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 stimulus with the assumption that personality is unconscious and the individual's responses will reveal its inner characteristics. Objective tests measure personality by self-assessment questionnaires with the underlying assumption that personality is primarily conscious and can be directly accessed.

All studies in this review that use personality tests use a form of objective test. This preference may be due at least two reasons. First, objective tests are considered more reliable and valid than projective ones. Second, objective tests are easier to apply, thus giving the (false) impression that it can be administered by researchers without deeper background in psychology and psychometrics. While this is true for the application of the test, McDonald & Edwards (2007) warn that interpretation of the result and its implications for the work practice is not straightforward and requires properly trained professionals.

The only study we found that reviews the literature about personality in software engineering is presented by McDonald & Edwards (2007). This review surveyed published articles in software engineering focusing on the application and interpretation of personality tests. The authors reviewed 40 papers published between 1984 and 2004, from which 13 distinct empirical studies that used personality tests were analyzed in depth. This analysis focused on "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 trying to identify 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 showed several methodological problems with respect to reliability and validity of the test instruments, and 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.

Although this review presents important results for the research in the theme, it has one important limitation. There are no explicitly stated search and inclusion/exclusion processes, therefore it is not possible to evaluate if the sample of 13 articles is not biased. In fact, we do not consider the study a systematic review. Besides, our review has a broader goal of mapping all studies about personality in software engineering, including those that use personality tests.

III. Review Method

Kitchenham (2004) adapted guidelines for performing systematic literature reviews (SLRs) in medicine for SLRs in software engineering. Later, using concepts from social science (Petticrew & Roberts 2006), Kitchenham & Charters (2007) updated the guidelines. The literature differentiates two broad types of systematic reviews (Petticrew & Roberts 2006), including:

- Conventional SLRs (Petticrew & Roberts 2006), which aggregate results about effectiveness of a treatment, intervention, or technology, and are related to specific research questions like: *Is intervention I on population P more effective in obtaining outcome O in context C than comparison treatment C?* (resulting in the PICOC structure (Petticrew & Roberts 2006))
- Mapping (or scoping) Studies (MS) (Arksey & O'Malley 2005) aim to identify all research related to a specific topic, i.e. to answer broader questions related to trends in research. Typical questions are exploratory: *What do we know about topic T?*

In this article, a mapping study of the research about personality in software engineering was performed. This work is classified as a secondary study since a review of primary studies was performed. The guidelines of Kitchenham & Charters (2007) were followed to plan and execute the review, and the structure used by Dybå et al. (2008)was followed to organize this article. Our goal is to collect evidence that can be use to guide research and practice, therefore we consider this review to be part of the evidence based software engineering effort (Kitchenham, Dybå & Jørgensen 2004).

A. Research Questions

The construction of the review protocol starts with the careful choice of the research questions that will guide de review process. In this review, our goal is to produce a mapping of the research about personality in software engineering and seek to answer the following specific questions:

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

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

RQ3: *What are the personality tests applied in the studies and the type of participants (professionals or students)?*

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

B. Inclusion and Exclusion Criteria

Studies were eligible for inclusion in the review if they presented research on personality in software engineering. Given the abundance of theoretical traditions related to the study of personality, we decided to include only those that addressed traits or type theories. Studies addressing other individual characteristics (e.g., behavior, cognition, competence, abilities, roles, etc.) were excluded.

No restriction was imposed on the type of the studies, thus articles reporting empirical studies (based on direct observation or experiments), theoretical studies (based on an understanding of the theme from experience or reference to other works), industrial experience reports, and literature reviews were eligible. Empirical studies that used either students or professional software developers as subjects were included. Only studies written in English were included.

Studies showing opinion pieces, viewpoints, or purely anecdotal evidence, and those presenting in progress research or incomplete results were also excluded.

C. Data Sources and Search Strategy

A broad search process was performed looking for primary studies published between January 1970 and December 2010, combining automatic and manual search. Manual Search was performed on relevant journals and conference proceedings (TABLE I). The researchers looked for title of all published articles in each source used in the manual search.

TABLE I. MANUAL SEARCH SOURCES

ACM Computer Surveys
ACM Transactions on Software Engineering Methodologies
Communications of the ACM
Empirical Software Engineering Journal
Evaluation and Assessment of Software Engineering
IEE Proceedings Software (now IET Software)
IEEE Software
IEEE Transactions on Software Engineering
Software Practice and Experience
Information and Software Technology
Int. Conference on Software Engineering
Int. Symposium on Empirical Software Engineering and Measurement
Journal of Systems and Software

The automatic search was performed in five search engines and indexing systems: ACM Digital Library; Elsevier ScienceDirect; El Compendex; IEEEXplore Digital Library; Scopus. All automatic searches were performed on the entire paper. To achieve high coverage, the search string used in the automatic search (Figure 1) was constructed based on two keywords: "personality" and "software engineering". Synonyms for software engineering used in other SLRs (e.g., Beecham et al. (2008)) were used. In consultancy with specialists in personality psychology, no synonym for personality was found to be necessary.

("software engineering" OR "software development" OR "agile development" OR "systems engineering" OR "systems development" OR "software project") AND ("personality")

Figure 1. Search String

D. Study Selection

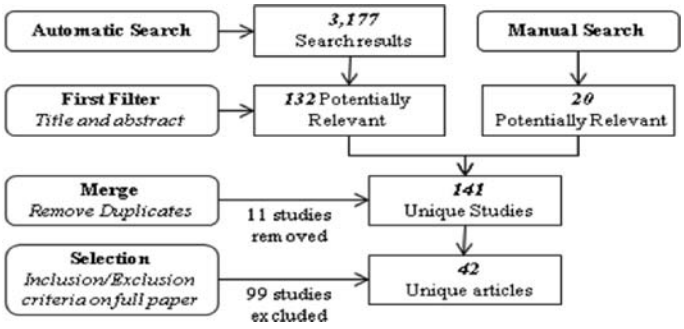

Figure 2. Stages of the Study Selection Process

The study selection process is illustrated in Figure 2. Results from the automatic search (n=3,177) were evaluated by four researchers looking at the title and excluding the studies that were clearly not relevant. The resulting articles (n=132) were merged with 20 potentially relevant studies found in the manual search, and 11 duplicates were removed. Finally, the four researchers applied the inclusion and exclusion criteria on the set of potentially relevant studies (n=141), resulting in 42 articles selected for the review (Appendix A).

E. Data Extraction

Data extraction was carried out guided by an extraction form implemented in MS Excel™. Each article was assigned a unique identifier (S1 – S42). The following information was extracted for each article: the year of publication, authors, research topic addressed, type of study (empirical, theoretical, literature review, and industry experience report), and the research question. For the empirical studies, the research method and the subject of investigation (professional or student) were also extracted. The methods presented by Easterbrook et al. (2007) were used to classify the research methods: experiments or quasi-experiments, case-studies, surveys, ethnography, and action research. The types of research question presented by Easterbrook et al. (2007) were used to classify the research questions and extract their variables.

TABLE II. TYPES OF RESEARCH QUESTIONS

RQ Category	Sub-category	RQ Code
Exploratory	Existence	E
	Description and Classification	DCL
	Descriptive-Comparative	DCO
Base-rate	Frequency Distribution	FD
	Descriptive-Process	RP
Relationship	Relationship	R
Causality	Causality	C
	Causality-Comparative	CC
	Causality-Comparative Interaction	CCI
Design	Design	D

TABLE III. EXAMPLE OF EXTRACTION OF RESEARCH QUESTION COMPONENTS

ID	Research Question (RQ)	RQ Code	RQ Structure	RQ Components	Topic
S15	(H01): Heterogeneous developer personalities and temperaments do not affect pair performance.	C	What effect does X have on Y?	X = heterogeneous developer personalities and temperaments Y = pair performance	Pair Programming

F. Synthesis of Findings

The data extracted from each study were integrated in categories representing the research topic addressed, research method used, the personality tests applied, and the type of participants in the research (subjects of investigation). Frequencies of each component in the categories were presented using column charts. Bubble charts were built to relate two or more categories, thus providing several combinations of the data.

The categories of research topics were built using a form of thematic analysis (Dixon-Woods et al. 2005), in the following steps. First, the research question was extracted from the primary study and classified according to the definition of Easterbrook et al. (2007) (TABLE II). The categories are defined as follows:

- Exploratory - questions that attempt to understand the phenomena and identify useful distinctions to clarify its understanding;
- Base-rate – once the phenomena is understood, base-rate questions aim to identify the normal patterns of the phenomena occurrence;
- Relationship – questions that aim understand the relationship between two phenomena occurrence, specifically whether the occurrence of one is related to the occurrence of the other;
- Causal – established the relationship, causal questions tries to understand why the relationship occurs by attempting to identify cause and effect;
- Design – these type of question focuses in designing better ways to do software engineering.

Second, the variables of the research question were extracted using templates for each type of question. Third, similar variables were grouped in themes. Finally, after all variables have been grouped, the groups were given names that better represented the research topic. TABLE III shows an example of data extraction of a research question of S15 in which the dependent variable Pair Performance was grouped with others that refer to pair programming, and at the end this group was categorized under the research topic Pair Programming. Using this procedure, the categories of all research topics were constructed.

IV. RESULTS

A. Overview of the Studies

The study selection process (Figure 2) resulted in 42 studies selected for data extraction and analysis (TABLE VII). Figure 3 depicts the temporal distribution of primary studies. Observe that 83% (35/42) of primary studies have been published after 2004. This indicates that, although the human factor in software engineering has been acknowledged and researched since the 1970s, research focusing in personality is much more recent, with the vast majority of the studies developed in the last decade.

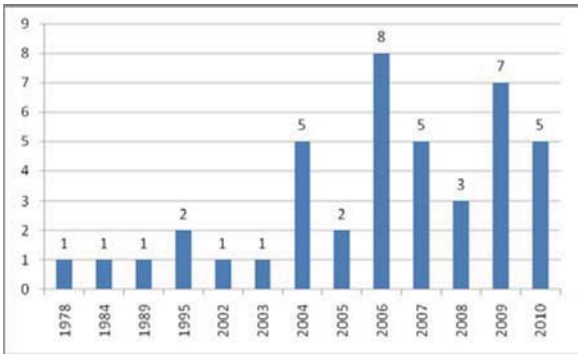

Figure 3. Temporal Distribution of Primary Studies

TABLE IV. STUDIES TYPES

Study Type	Number	Percentage
Empirical	36	86%
Theoretical	6	14%
Total	42	100%

TABLE V. RESEARCH METHODS

Research Method	Number	Percentage
Experiment	12	33%
Quasi-Experiment	2	6%
Case Study	11	30%
Survey	10	28%
Ethnography	1	3%
Total	36	100%

TABLE VI. SUBJECTS OF INVESTIGATION

Subject of Investigation	Number	Percentage
Students	20	56%
Professionals	15	41%
Both	1	3%
Total	36	100,00%

TABLE VII. SUMMARY OF THE STUDY SELECTION PROCESS

Search Strategy	Source	Search Results	Potentially Relevant	No Access	Not Relevant	Repeated	Incomplete	Relevant Studies	Search Efficacy	First Filter Efficacy
		(a)	(b)					(c)	(c/a)	(c/b)
Automatic	IEEEXplore	229	15	0	5	0	4	6	2,62%	40,00%
	ACM	604	46	0	21	19	0	6	0,99%	13,04%
	ScienceDirect	919	18	0	9	0	0	9	0,98%	50,00%
	EI Compendex	125	29	1	2	14	0	12	9,60%	41,38%
	Scopus	1300	24	1	8	9	0	6	0,46%	25,00%
Manual	Proceedings and Journals	-	20	0	6	11	0	3		15,00%
	Total	3177	152	110				42		

Eight six percent (36/42) of the primary studies report empirical research (TABLE IV), which increases the strength of evidence found in this review (Atkins 2004). The empirical studies used a variety of research methods (TABLE V) and nearly 40% (14/36) performed controlled experiments or quasi-experiments which also provides strong evidence. Finally, the studies used professionals from industry and students as subjects in the investigations (TABLE VI).

B. Answers to the Research Questions

This section presents the synthesis of the findings of this review organized according to the four research questions. We present the synthesis of empirical and theoretical studies separated to allow a clear separation of empirically based evidence from non-tested propositions and models.

1) Synthesis of the Empirical Studies

RQ1: *What are the research topics investigate in the research about personality in software engineering?*

The research topics were categorized using the procedure explained in Section III.F. Since a given study can contain more than one research question, it can be related to more than one research topic. Therefore, the sum of the percentages in the chart of Figure 4 is greater than 100%.

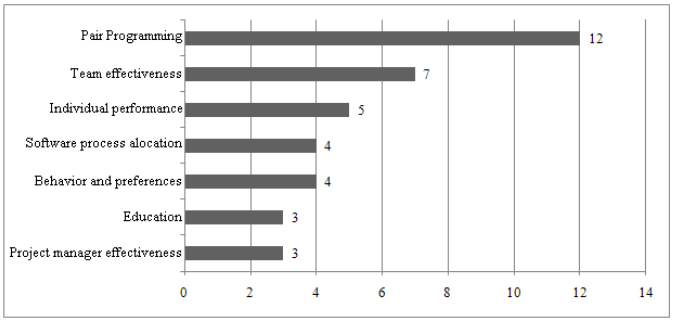

Figure 4. Research Topics

We will briefly describe each research topic from the point of view of how they are influenced by personality as investigated in the primary studies.

Pair Programming is a practice mainly used associated agile methodologies in which two programmers work collaboratively on the same code and sharing the same computer. The potential benefits for software development are, among others, reducing defects, improving quality and communication. Therefore, the work associated with this research topic investigating the influence of personality in implementing this practice, in particular in which case pairs formed by individuals with different personality perform better then uniform pairs.

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

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

Software Process Allocation is studied considering that the work in software development is diversified and multidisciplinary, involving tasks such as analysis, design, coding, testing, among others. Therefore, the allocation of individual to roles in a software team is seen as a critical factor for project success because certain personality traits may be better suited to perform certain tasks.

Behavior and Preferences is investigate given that certain attitudes and preferences of the software engineers are influenced by their personality traits or types, understand how this relationship is defined is important to provide a general understanding of how these attitudes and preferences explain the work styles, habits and preferences for tools and processes.

The influence of personality in the *Education* of students of software engineering work is seen by some researchers as a key factor for successful learning. Therefore, these researchers are seeking to understand of how the style and teaching practices can be tailored to the specific personality to improve the efficacy of the learning process.

Project Manager Effectiveness is a research topic explored from the perspective of how personality traits or types affect their leadership behavior and how this behavior impacts individual and team's satisfaction, and, consequently, the success of the project.

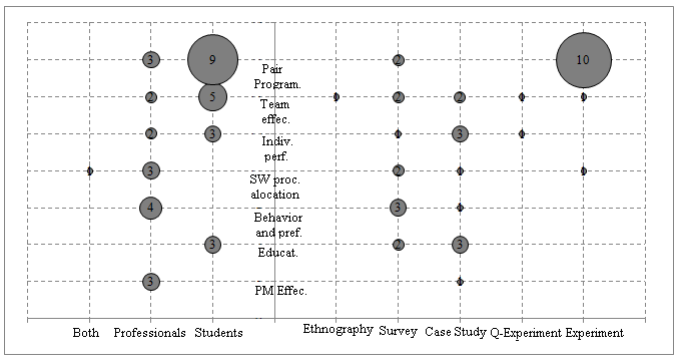

Figure 5. Mapping of Research Topics, Methods, and Subjects

RQ2: *What are the research methods used in the studies and in which 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 regarding the type of participants. In Figure 5 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 and the number of studies is indicated in the center of the circle.

The distribution between types of participants in the studies is relatively well balanced: 56% (20/36) use students and 42% (15/36) use professionals. Nevertheless, studies using professionals are better distributed among the research topics, unlike the research with students who are focused mainly on pair programming. The case study and survey studies are well distributed among the research topics, but experiment is very focused on pair programming, which may indicate the difficulty of using experiments on other topics, and also employing professionals.

RQ3: *What are the personality tests applied in the studies and the type of participants (professionals or students)?*

Personality tests used in the primary studies are shown in Figure 6.

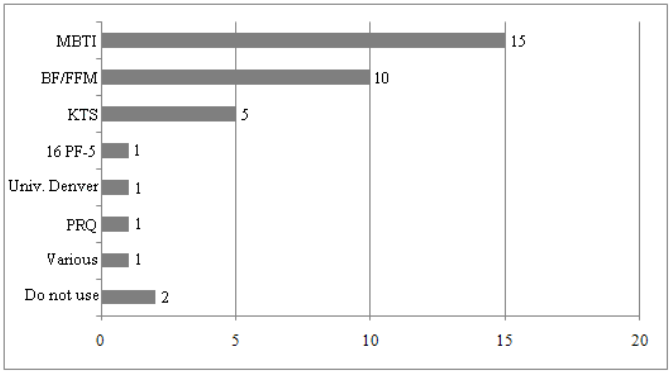

Figure 6. Personality Tests

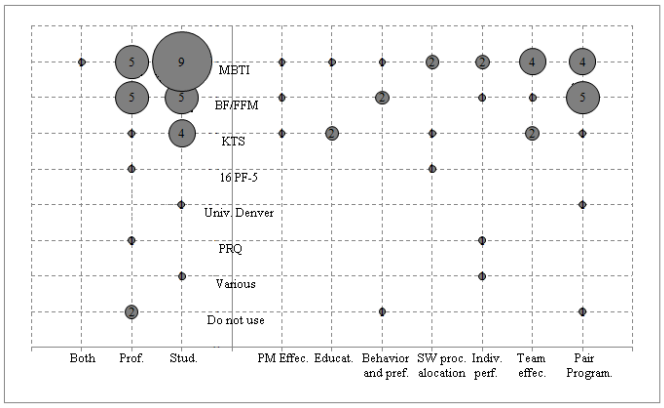

Figure 7. Mapping of Research Topics, Personaliy Tests, and Subjects

The Myers-Brigg Type Indicator (MBTI) is used in 42% (15/36) of the studies. Combined with the 14% (5/36) that used the Kersey Temperament Sorter (KTS), 56% (20/36) of studies use tests based on Jung's Personality Types Theory. Tests based on the Five Factor Model (FFM) (Costa & McCrae 1992), in particular the NEO-PI test, have been used in 28% (10/36) of the studies. The remaining studies used a variety of other specific tests.

In Figure 7, we present the combination of the research topics from RQ1 and the types of participants from RQ2, with the personality test used in the study. Among the 20 studies that applied MBTI and KTS, only 11 provided the results for the sample of participants (S10, S12, S14, S15, S16, S20, S22, S29, S30, S33, S39). Regarding the application of tests based on FFM, 5 presented the results for the sample of participants (S2, S6, S7, S8, S41). Therefore, one could think of performing meta-analysis to integrate these results. However, considering the poor description of and great differences in the research contexts, lack of details of the statistical treatment of the data, and several methodological problems related to test application and interpretation (as those discussed by McDonald & Edwards (2007)), meta-analysis would not provide reliable results.

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 evidences about the influence of the personality on the tasks and processes in software engineering, organized according to the research topics (Figure 4). Regarding this synthesis, the reader must be aware that we did not investigate the consistency among the operational definitions of the constructs used as outcomes in the studies. In fact, most studies did not provide characterization of these definitions that would allow such investigation. Furthermore, we also did not check the consistencies of the population among the studies, again for lack of information in most of the studies. Therefore, the discrepancies among the results that are presented in the synthesis may be the result of different operationalization of the evaluated outcomes, to differences in the contexts or population, or differences in the version of the tests applied and the way results of the application were analyzed. To facilitate the understanding of the synthesis, we put the personality test together with the reference of the study,

for instance, S2-FFM means the study S2 that uses a test based on FFM.

Pair Programming: most of the studies in this topic tested whether the composition of the pair with respect to differences or similarities in personality, influenced the performance of the pair. Seven studies presented conflicting evidences: three found that pairs with distinct personality types (heterogeneous) perform better than homogeneous pairs (S10-MBTI, S15-KTS, S25-MBTI), three found no relationship between heterogeneous and homogeneous pairs (S3-FFM, S41-FFM, S42-FFM), and one study showed that personality clash is one of the most serious problems in pair programming (S18-None), but did not report which personality types cause more clashes. One study (S5-Denver) found evidence that pairs formed by explicitly combining certain personality traits (Open-mindedness and Responsibility) result in code with better quality. On the other hand, one study argues that personality has low predictive value for performance of pairs when compared with expertise, task complexity, and country (S2-FFM) and another study shows that personality has no influence on communication, satisfaction, trust, and compatibility of the pair (S9-MBTI). With respect to collaboration between members of the pair, one studied found that variability in personality improves collaboration (S4-FFM).

Team Effectiveness: the studies showed evidence that personality relates to project success (S33-MBTI), and also to code quality, individual satisfaction, and team cohesion (S8-FFM). Other studies show that personality diversity in teams is not directly related to team effectiveness (S26-KTS, S30-MBTI, S33-MBTI), while one study 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 (S23-KTS). Another study showed that teams with predominance of Introversion experience lower effectiveness due to communication problems (S20-MBTI). Finally in this topic, one study lends evidence that heterogeneous teams are "optimum" when solving unstructured tasks while homogeneous teams are "optimum" when solving structured tasks (S39-MBTI).

Individual Performance: articles in this topic investigated the best fit between certain software engineering tasks and personality. One of the studies identified that a particular personality type is positively related to performance of the task of code review (S27-MBTI). Using a combination of the results of several specific tests, one study found that personality exhibits a predictive relationship with object oriented programming (S40-Various). On the other hand, one study showed no significant relationship between personality and programming performance (S6-FFM) and a second found no significant difference in personality between exceptional (high performers) and non-exceptional programmers (S14-MBTI). Finally, it was found that personality affects individual job satisfaction (S35-PRQ) in two aspects: challenges at work and feeling of respect and appreciation.

Software Process Allocation: the four empirical studies classified in this topic investigated the relationship between personality and performance, with respect to technical roles in the software process. From the results of surveys of personality types of software engineers, two studies propose how to map personality types to technical roles (S12-MBTI, S26-KTS). On the other hand, one study failed to demonstrate the influence of personality and did not consider this factor on the proposed competency model (S16-MBTI).

Behavior and Preferences: The four empirical studies classified in this topic provide evidence that: personality influences attitudes toward judgment and decision making of the software engineers (S7-FFM); personality factors affect how individuals react to or prefer techniques, methods and processes (S32-FFM, S38-None); systems analysts tend to be technically oriented (S37-MBTI).

Education: the three empirical studies classified in this topic provide evidence that: a form of education with specific activities based on the variety of personality of students is more effective (S22-MBTI, S29-KTS) and that the assembly of groups based on the personality of each member maximizes the strengths and minimizes weaknesses, promoting a better way of learning for students (S24-KTS).

Project Manager Effectiveness: in this topic the studies show that: there is no pattern linking the leadership behavior of the project manager with personality traits (S11-MBTI); personality of the project manager is related to project success (S17-FFM); heterogeneity between the project manager and the team members regarding to some aspects of personality are related to team performance (S26-KTS).

TABLE VIII. SUMMARY OF THEORETICAL STUDIES

ID	Research Topic	Test	Outcomes
S01	Software process allocation	MBTI	The study proposes a mapping between the job requirements of each role in software development, interpersonal skills (soft skills), and the corresponding personality types most suitable.
S13	Software process allocation	MBTI	The study proposes a model showing the influence of personality variables in four stages of the software development process.
S19	Personality Test Application	MBTI	This study, described in Section II, presents a critical appraisal of 13 empirical studies that used personality tests in software engineering. The study analysis the reliability of the test instruments and the evidences of adequate application and interpretation of results. Recommendations to potential participants, practitioners, and researchers are provided
S28	Job Retention	MBTI	The study suggests that certain personality types influence the decision paths in relation to changes in employment in software engineering.
S34	Project manager effectiveness	MBTI	The study suggests a personality type that best fits the role of a project manager.
S36	Education	MBTI	The study suggests that a form of education with specific activities based on the variety of personality of students is more effective.

2) Synthesis of Theoretical Studies

The six theoretical studies are summarized in TABLE VIII. All theoretical studies based their model construction or propositions on Jung's Personality Type Theory and, in particular, on the MBTI test, although they did not actually use the test. Job Retention and Personality Test Application are the two topics that are only addressed by theoretical studies.

V. DISCUSSION

A. Implications for Research and Practice

As far as we are aware of, this is the first attempt to review the literature about personality in software engineering in a systematic way. Our findings show a great concentration of empirical research on the study of the effect of personality in pair programming, followed by the studies on the influence of personality in team effectiveness. The other research topics received much less attention. Furthermore, no external replication of empirical studies was found and only one paper reported an internal replication. The models proposed in the theoretical studies were not tested in empirical studies. One of the implications of these findings for the research community is that this area of research has open many opportunities to be explored. In particular, 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.

For the practitioners, the direct application of the results presented in the studies must be conducted with care. The conflicting evidences suggest 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 & Edwards (2007) in that the interpretation of the results of personality tests and its implications for the work practice is not straightforward and requires properly trained professionals. We also believe that a careful analysis of the context in which the research was conduct is necessary to assess the possibility of generalizing the results to other settings.

B. Limitations of this Review

The most common limitations in a systematic review are the possible biases introduced in the selection process and inaccuracies of the data extraction. These are also the main possible limitations of this review. The research protocol developed based on well established guidelines is the measure taken to prevent selection bias. The combination of automatic search in several engines and manual search on relevant publications 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 a third party or in consensus meetings.

Moreover some software engineering papers that did not use any of the search terms defined in the protocol may not have been found, e.g. studies that used any personality test but did not use the word personality explicitly.

VI. CONCLUSION

This review analyzed 3,177 studies from automatic and 20 from manual search processes, of which 42 provided answers to four research questions. These studies investigated 9 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 showed that 6 different personality tests have been most commonly used. While MBTI largely dominates the studies, tests based on FFM, in particular NEO-PI, are becoming more popular.

Considering the importance of the effect of human factors in many aspects of software engineering, the amount of research about the effects or influences of personality in the field is very small. Besides, the evidences are weak and in many cases inconclusive. More research effort is required if we want research results that can influence the practice of software development.

This research area is multidisciplinary in nature, since it necessarily combines knowledge from software engineering and humanities, in particular psychology. However, most of the studies have been conducted by researchers from the software engineering area without collaboration with psychologists or researchers from related areas (at least this collaboration is not explicitly mentioned in the papers). We argue that collaborative work is imperative on this research theme.

Finally this research should be extended to incorporate deeper analysis and comparison between studies, particularly to understand the effect size of the outcomes encountered.

ACKNOWLEDGMENT

Professor Fabio Q. B. da Silva holds a research grant from the Brazilian National Research Council (CNPq), process #314523/2009-0. Cleviton V. F. Monteiro currently receives scholarships from Capes.

REFERENCES

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

Arksey, H.; O´Malley, L. Scoping studies: towards a methodological framework, International Journal of Social Research Methodology, 8(1), 19-32, 2005.

Atkins, D. et al. Grading quality of evidence and strength of recommendations, BMJ 328(1490), 2004.

Beecham, S. et al. Motivation in Software Engineering: A systematic literature review. IST, 50(9-10), 860-878, 2008.

Capretz, L., & Ahmed, F. Making Sense of Software Development and Personality Types. IT Professional, (February), 6–13. IEEE Computer Society, 2010.

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

Da Cunha, A. D., & Greathead, D. Does personality matter? an analysis of code-review ability. *Communications of the ACM*, *50*(5), 109-112, 2007.

Diagnostic and Statistical Manual of Mental Disorders (DSM-IV-TR). Fourth Edition, Text Revision, American Psychiatric Association, 2000.

Dixon-Woods, M., et al. A. (2005). Synthesising qualitative and quantitative evidence: a review of possible methods. *Journal of health services research & policy*, *10*(1), 45-53.

Dybå, T.et al. Empirical studies of agile software development: A systematic review. Information and Software Technology, 50(9-10), 833-859, 2008.

Easterbrook, S. M.; et al. Selecting Empirical Methods for Software Engineering Research. In F. Shull, J. Singer and D. Sjøberg(eds) Guide to Advanced Empirical Software Engineering, Springer, 2007.

Gorla, N., & Lam, Y. W. Who should work with whom?. *Communications of the ACM*, *47*(6), 79-82. doi: 10.1145/990680.990684, 2004.

Hardiman, L.T. Personality types and software engineers. IEEE Computer 30, 10 (Oct. 1997), 10.

Kerth, N.L, Coplien, J., and Weinberg, J. Call for the rational use of personality indicators. IEEE Computer 31, 1 (Jan. 1998), 146–147.

Kitchenham, B.; Dybå, T.; Jørgensen, M. Evidence-based Software Engineering. Proceedings of the 26th International Conference on Software Engineering, (ICSE '04), IEEE, Washington DC, USA, pp 273 – 281, 2004.

Kitchenham, B.A. Procedures for Undertaking Systematic Reviews. Joint Technical Report, Computer Science Department, 2004, Keele University (TR/SE-0401) and National ICT Australia Ltd (0400011T.1), 2004.

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

Lee, J., & Shneiderman, B. Personality and programming: Time-sharing vs. batch preference. In Proceedings of the ACM '78 Annual Conference - Volume 2, pp. 561–569, 1978. doi: 10.1145/800178.810092.

Mcdonald, B. S., & Edwards, H. M. (n.d.). Who Should Test whom?. *Communications of the ACM*, 50(1), 67-71, 2007

Petticrew, M.; Roberts, H. Systematic Reviews in the Social Sciences. Blackwell Publishing, 2006.

Ryckman, R. *Theories of Personality.* Belmont, California: Thomson/Wadsworth, 2004.

Shneiderman, B. Software Psychology: Human Factors in Computer and Information Systems, Winthrop Publishers, 1980.

Weinberg, G.M. The Psychology of Computer Programming, silver anniversary ed. Dorset House Publishing, 1998. (First Edition, 1971).

APPENDIX A – SELECTED PRIMARY STUDIES

S1 Capretz, L. F.; Ahmed, Faheem. Making Sense of Software Development and Personality Types. IT Professional 12, no. 1 (January 2010): 6-13. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5403172.

S2 Hannay, J. E.; Arisholm, E.; Engvik, Harald; Sjoberg, Dag I.K. Effects of Personality on Pair Programming. IEEE Transactions on Software Engineering 36, no. 1 (January 2010): 61-80. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5089333.

S3 Salleh, Norsaremah; Mendes, Emilia; Grundy, John; Burch, Giles St. J. An empirical study of the effects of personality in pair programming using the five-factor model. 2009 3rd International Symposium on Empirical Software Engineering and Measurement. IEEE, 2009. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5315997.

S4 Walle, Thorbjorn; Hannay, J. E. Personality and the nature of collaboration in pair programming. 2009 3rd International Symposium on Empirical Software Engineering and Measurement. IEEE, 2009. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5315996.

S5 Chao, J.; Atli, G. Critical Personality Traits in Successful Pair Programming. AGILE 2006 (AGILE'06). IEEE, 2006. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1667566.

S6 Darcy, D.P. Exploring Individual Characteristics and Programming Performance: Implications for Programmer Selection. Proceedings of the 38th Annual Hawaii International Conference on System Sciences. Vol. 15. IEEE, 2005. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1385889.

S7 Feldt, Robert; Angelis, Lefteris; Torkar, Richard; Samuelsson, Maria. Links between the personalities, views and attitudes of software engineers. Information and Software Technology 52, no. 6 (June 2010): 611-624. http://linkinghub.elsevier.com/retrieve/pii/S0950584910000029.

S8 Acuña, Silvia T.; Gómez, Marta; Juristo, Natalia. How do personality, team processes and task characteristics relate to job satisfaction and software quality?. Information and Software Technology 51, no. 3 (March 2009): 627-639. http://linkinghub.elsevier.com/retrieve/pii/S0950584908001080.

S9 Choi, K. S.; Deek, Fadi P.; Im, Il. Pair dynamics in team collaboration. Computers in Human Behavior 25, no. 4 (July 2009): 844-852. http://linkinghub.elsevier.com/retrieve/pii/S0747563208001817.

S10 Choi, K. S.; Deek, Fadi P.; Im, Il. Exploring the underlying aspects of pair programming: The impact of personality. Information and Software Technology 50, no. 11 (October 2008): 1114-1126. http://linkinghub.elsevier.com/retrieve/pii/S0950584907001292.

S11 Strang, Kenneth David. Examining effective technology project leadership traits and behaviors. Computers in Human Behavior 23, no. 1 (January 2007): 424-462. http://linkinghub.elsevier.com/retrieve/pii/S0747563204001773.

S12 Capretz, L. F. Personality types in software engineering. International Journal of Human-Computer Studies 58, no. 2 (February 2003): 207-214. http://linkinghub.elsevier.com/retrieve/pii/S1071581902001374.

S13 Bishop-Clark, C. Cognitive style, personality, and computer programming. Computers in Human Behavior 11, no. 2 (January 1995): 241-260. http://linkinghub.elsevier.com/retrieve/pii/074756329400034F.

S14 Turley, Richard T.; Bieman, James M. Competencies of exceptional and nonexceptional software engineers. Journal of Systems and Software 28, no. 1 (January 1995): 19-38. http://linkinghub.elsevier.com/retrieve/pii/0164121294000782.

S15 Sfetsos, Panagiotis; Stamelos, Ioannis; Angelis, Lefteris; Deligiannis, Ignatios. An experimental investigation of personality types impact on pair effectiveness in pair programming. Empirical Software Engineering 14, no. 2 (November 2008): 187-226. http://www.springerlink.com/index/10.1007/s10664-008-9093-5.

S16 Derro, Mary Ellen; Williams, Christine R. Behavioral competencies of highly regarded systems engineers at NASA. 2009 IEEE Aerospace conference. IEEE, 2009. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4839712.

S17 Wang, Yi; Li, Fan. How Does Project Managers' Personality Matter? Building the Linkage between Project Managers' Personality and the Success of Software Development Projects. Proceeding of the 24th ACM SIGPLAN conference companion on Object oriented programming systems languages and applications - OOPSLA '09. New York, New York, USA: ACM Press, 2009. http://portal.acm.org/citation.cfm?doid=1639950.1640049.

S18 Begel, Andrew; Nagappan, Nachiappan. Pair Programming: What's in it for Me?. Proceedings of the Second ACM-IEEE international symposium on Empirical software engineering and measurement - ESEM '08. New York, New York, USA: ACM Press, 2008. http://portal.acm.org/citation.cfm?doid=1414004.1414026.

S19 McDonald, Sharon; Edwards, Helen M. Who should test whom?. Communications of the ACM 50, no. 1 (January 2007): 66-71. http://portal.acm.org/citation.cfm?doid=1188913.1188919.

S20 Karn, John; Cowling, Tony. A follow up study of the effect of personality on the performance of software engineering teams. Proceedings of the 2006 ACM/IEEE international symposium on International symposium on empirical software engineering - ISESE '06.

New York, New York, USA: ACM Press, 2006. http://portal.acm.org/citation.cfm?doid=1159733.1159769.

S21 Layman, L. Changing Students' Perceptions: An Analysis of the Supplementary Benefits of Collaborative Software Development. 19th Conference on Software Engineering Education & Training (CSEET'06). IEEE, 2006. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1617342.

S22 Layman, Lucas; Cornwell, Travis; Williams, Laurie. Personality types, learning styles, and an agile approach to software engineering education. ACM SIGCSE Bulletin 38, no. 1 (March 2006): 428. http://portal.acm.org/citation.cfm?doid=1124706.1121474.

S23 Pieterse, Vreda; Kourie, Derrick G; Sonnekus, Inge P. Software engineering team diversity and performance. Proceedings of the 2006 annual research conference of the South African institute of computer scientists and information technologists on IT research in developing couuntries - SAICSIT '06. New York, New York, USA: ACM Press, 2006. http://portal.acm.org/citation.cfm?doid=1216262.1216282.

S24 Rutherfoord, Rebecca H. Using personality inventories to form teams for class projects. Proceedings of the 7th conference on Information technology education - SIGITE '06. New York, New York, USA: ACM Press, 2006. http://portal.acm.org/citation.cfm?doid=1168812.1168817.

S25 Katira, Neha; Williams, Laurie; Wiebe, Eric; Miller, Carol; Balik, Suzanne; Gehringer, Ed. On understanding compatibility of student pair programmers. Proceedings of the 35th SIGCSE technical symposium on Computer science education - SIGCSE '04. New York, New York, USA: ACM Press, 2004. http://portal.acm.org/citation.cfm?doid=971300.971307.

S26 Gorla, Narasimhaiah; Lam, Yan Wah. Who should work with whom?. Communications of the ACM 47, no. 6 (June 2004): 79-82. http://portal.acm.org/citation.cfm?doid=990680.990684.

S27 Da Cunha, Alessandra Devito; Greathead, David. Does personality matter? An Analysis of Code-Review Ability. Communications of the ACM 50, no. 5 (May 2007): 109-112. http://portal.acm.org/citation.cfm?doid=1230819.1241672.

S28 Mourmant, Gaëtan; Gallivan, Mike. How personality type influences decision paths in the unfolding model of voluntary job turnover. Proceedings of the 2007 ACM SIGMIS CPR conference on 2007 computer personnel doctoral consortium and research conference The global information technology workforce - SIGMIS-CPR '07. New York, New York, USA: ACM Press, 2007. http://portal.acm.org/citation.cfm?doid=1235000.1235032.

S29 Galpin, Vashti C.; Sanders, Ian D.; Chen, Pei-yu. Learning styles and personality types of computer science students at a South African university. Proceedings of the 12th annual SIGCSE conference on Innovation and technology in computer science education - ITiCSE '07. New York, New York, USA: ACM Press, 2007. http://portal.acm.org/citation.cfm?doid=1268784.1268844.

S30 Miller, J. A cognitive-based mechanism for constructing software inspection teams. IEEE Transactions on Software Engineering 30, no. 11 (November 2004): 811-825. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1359772.

S31 Acuña, Silvia T.; Juristo, Natalia. Assigning people to roles in software projects. Software: Practice and Experience 34, no. 7 (June 2004): 675-696. http://doi.wiley.com/10.1002/spe.586.

S32 Feldt, Robert; Torkar, Richard; Angelis, Lefteris; Samuelsson, Maria. Towards individualized Software Engineering: Empirical Studies Should Collect Psychometrics. Proceedings of the 2008 international workshop on Cooperative and human aspects of software engineering - CHASE '08. New York, New York, USA: ACM Press, 2008. http://portal.acm.org/citation.cfm?doid=1370114.1370127.

S33 Peslak, Alan R. The Impact of Personality on Information Technology Team Projects. Human Factors (2006): 273-279.

S34 Brewer, Jeffrey L. Project Managers, Can We Make Them or Just Make Them Better?. Proceedings of the 6th conference on Information technology education - SIGITE '05. New York, New York, USA: ACM Press, 2005. http://portal.acm.org/citation.cfm?doid=1095714.1095754.

S35 Gallivan, Michael J. Examining IT Professionals' Adaptation to Technological Change: The Influence of Gender and Personal Attributes. ACM SIGMIS Database 35, no. 3 (August 2004): 28-49. http://portal.acm.org/citation.cfm?doid=1017114.1017119.

S36 Capretz, L. F. Implications of MBTI in software engineering education. ACM SIGCSE Bulletin 34, no. 4 (December 2002): 134. http://portal.acm.org/citation.cfm?doid=820127.820185.

S37 Smith, D C. The personality of the systems analyst: An Investigation. ACM SIGCPR Computer Personnel 12, no. 2 (December 1989): 12-14. http://portal.acm.org/citation.cfm?doid=1036387.1036389.

S38 Lee, Jeanne M; Shneiderman, Ben. Personality and programming: Time-Sharing vs. Batch Preference. Proceedings of the 1978 annual conference on - ACM '78. New York, New York, USA: ACM Press, 1978. http://portal.acm.org/citation.cfm?doid=800178.810092.

S39 White, K. A preliminary investigation of information systems team structures. Information & Management 7, no. 6 (December 1984): 331-335. http://linkinghub.elsevier.com/retrieve/pii/0378720684900284.

S40 Cegielski, Casey G.; Hall, Dianne J. What makes a good programmer?. Communications of the ACM 49, no. 10 (October 2006): 73-75. http://portal.acm.org/citation.cfm?doid=1164394.1164397.

S41 Salleh, Norsaremah; Mendes, Emilia; Grundy, John; Burch, Giles St J. The Effects of Neuroticism on Pair Programming: An Empirical Study in the Higher Education Context. Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement - ESEM '10. New York, New York, USA: ACM Press, 2010. http://portal.acm.org/citation.cfm?doid=1852786.1852816.

S42 Salleh, Norsaremah; Mendes, Emilia; Grundy, John; Burch, Giles St. J. An Empirical Study of the Effects of Conscientiousness in Pair Programming using the Five-Factor Personality Model. Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering - ICSE '10. New York, New York, USA: ACM Press, 2010. http://portal.acm.org/citation.cfm?doid=1806799.1806883.