Information and Software Technology 57 (2015): 141 – 156

DOI: http://dx.doi.org/10.1016/j.infsof.2014.09.002

Are team personality and climate related to satisfaction and software quality? Aggregating results from a twice replicated experiment

Silvia T. Acuña [a,*], Marta N. Gómez [b], Jo E. Hannay [c], Natalia Juristo [d], Dietmar Pfahl [e]

[a] Escuela Politécnica Superior, Universidad Autónoma de Madrid, Calle Francisco Tomás y Valiente 11, 28049 Madrid, Spain
[b] Escuela Politécnica Superior, Universidad CEU San Pablo, Campus de Monteprincipe, 28668 Boadilla del Monte, Madrid, Spain
[c] Modeling and Simulation Research Program, Norwegian Defence Research Establishment (FFI), P.O. Box 25, NO-2027 Kjeller, Norway [d] Facultad de Informática, Universidad Politécnica de Madrid, Campus de Montegancedo, 28660 Boadilla del Monte, Madrid, Spain
[e] Institute of Computer Science, University of Tartu, J. Liivi 2, Tartu 50409, Estonia

A B S T R A C T

Context: Research into software engineering teams focuses on human and social team factors. Social psy-chology deals with the study of team formation and has found that personality factors and group pro-cesses such as team climate are related to team effectiveness. However, there are only a handful of empirical studies dealing with personality and team climate and their relationship to software develop-ment team effectiveness.
Objective: We present aggregate results of a twice replicated quasi-experiment that evaluates the rela-tionships between personality, team climate, product quality and satisfaction in software development teams.
Method: Our experimental study measures the personalities of team members based on the Big Five per-sonality traits (openness, conscientiousness, extraversion, agreeableness, neuroticism) and team climate factors (participative safety, support for innovation, team vision and task orientation) preferences and perceptions. We aggregate the results of the three studies through a meta-analysis of correlations. The study was conducted with students.
Results: The aggregation of results from the baseline experiment and two replications corroborates the following findings. There is a positive relationship between all four climate factors and satisfaction in software development teams. Teams whose members score highest for the agreeableness personality fac-tor have the highest satisfaction levels. The results unveil a significant positive correlation between the extraversion personality factor and software product quality. High participative safety and task orienta-tion climate perceptions are significantly related to quality.
Conclusions: First, more efficient software development teams can be formed heeding personality factors like agreeableness and extraversion. Second, the team climate generated in software development teams should be monitored for team member satisfaction. Finally, aspects like people feeling safe giving their opinions or encouraging team members to work hard at their job can have an impact on software quality. Software project managers can take advantage of these factors to promote developer satisfaction and improve the resulting product.

1. Introduction

People are a fundamental and critical concern in software development success or failure. Some research has taken this aspect into account and incorporated people into the software process [1,45,54,57,64,69]. These researchers analyze people individually establishing their relationships to the activities performed within the project. Citing DeMarco and Lister [25], "Most software development projects fail because of failures with the team running them". There is recognition that software process productivity and efficiency is critically dependent on human and social factors [15]. Although there is the perception that developers work together to perform interdependent tasks and team interrelationships are complex, there are only a few empirical studies

* Corresponding author. Tel.: +34 91 4972275; fax: +34 91 4972235.
E-mail addresses: silvia.acunna@uam.es (S.T. Acuña), mgomez.eps@ceu.es (M.N. Gómez), Jo.Hannay@ffi.no (J.E. Hannay), natalia@fi.upm.es (N. Juristo), dietmar.pfahl@ut.ee (D. Pfahl).

[2,3,26,35] that characterize and compare software team factors leading to higher effectiveness.

Our research takes the lead from social psychology research, following in the footsteps of Barrick and Mount [8] and Barry and Stewart [10] who related personality to individual and team effectiveness. We empirically analyze relationships during software development between team personality and team climate with software product quality and team satisfaction. We present the results of a quasi-experiment and two replications. We aggregate the results of the three studies through a meta-analysis of correlations. Knowledge like this can lead to guidance for team managers on how to form better teams according on team member personalities and by introducing group dynamics to resolve conflicts or improve team cohesion (team processes).

In team performance literature, the basic team behavior model is based chiefly on McGrath's input-process-output model [51]. This is a simple but very effective approach to groups. This model starts by evaluating the group inputs, like members, their qualities and characteristics, as well as the elements of the group environment. These input factors are combined and interact to form group processes, like cohesion, conflict and team climate, etc. In turn, these group processes have an impact on group effectiveness (output). This model divides the examined variables into three basic components: people, team processes and team effectiveness.

As regards people, we evaluate the personality factors that social psychology research suggests are applicable to team operation and results [53]. Personality factors determine personal preferences, opinions, attitudes, values and characteristics. In other words, everyone has a different personality topology, and this is what differentiates that individual from everybody else. While early studies on the effects of personality in the context of software engineering tended to use the Myers-Briggs Type Indicator (MBTI) [55] for personality measurement [19], personality psychologists have been apt to agree more recently on the advantages of using the Five Factor Model (FFM) or Big Five [22] as a parsimonious and comprehensive framework of personality traits [18].

Therefore, our experimental study measures the personalities of team members based on five personality dimensions:

1. *Neuroticism* is a broad dimension that includes traits like anxiety, moodiness, irritability or frustration.
2. *Extraversion* is a trait of people who take a trusting and enthusiastic view of others, which is associated with being sociable, assertive and talkative.
3. *Agreeableness* is a trait of showing altruistic concern and emotional support towards other people.
4. *Openness to experience* denotes especial inquisitiveness about new ideas, values, feelings and interests.
5. *Conscientiousness* is a trait of perseverant, scrupulous and responsible behavior.

These five personality factors are measured before starting the teamwork using the NEO Five Factor Inventory (FFI) [22]. We measure team personality by averaging the scores of each team member for each individual personality factor. We use the mean as an aggregation measure of the group data following Barry and Stewart [10], Barrick et al. [9], van Vianen and De Dreu [66] and English et al. [28]. More recently, Peslak [56] has also used the mean of the individual scores of each team member to get software developer team constructs.

With respect to team processes, team climate is a topic that has not been researched much in software development. However, social psychology researchers like Burch and Anderson [17] claim that climate influences team development, affecting personal relationships within teams that are vital to the success or failure of teams and the work that they do [24,44,49,70]. We examined four team climate factors [67,68]:

1. *Participative safety*: how much trust participating team members feel there is within the group when explaining their opinions and ideas.
2. *Support for innovation*: support lent by the team to innovative ideas.
3. *Team vision*: how clearly the team defines goals.
4. *Task orientation*: how much effort the team puts into achieving excellence in what it does.

These four factors are measured before the project using the Team Selection Inventory (TSI) [4,17] test to establish subject climate preferences. They are measured after the project using the Team Climate Inventory (TCI) [5–7] test to establish the subjects' perceptions of the climate. We measure both team climate preference and team climate perception by averaging the scores of each team member for each individual team climate preference and perception factor, respectively. We use the arithmetic mean as the group aggregation index to get team preferences and perceptions for each of the four team climate factors.

Team effectiveness refers to the extent to which the team achieves its goals like productivity, delivery time and product or service quality. According to Hackman and Oldham [34] and Gladstein [32], effectiveness also has to do with aspects that are not explicitly part of the team's objectives, such as satisfaction or workability. In research into team effectiveness, the factors usually measured and assessed in the output are satisfaction and performance. Our empirical study examines both these factors:

1. *Satisfaction* indicates the extent to which a team member sees eye to eye or agrees with his or her team mates regarding the work method, generated atmosphere, objective attainment, etc.
2. *Developed software quality* assessed by analyzing the code and documents of the projects developed by the teams and the participation of the team members observed during project development.

All teams are groups of students who should complete a term-length class assignment in the original quasi-experiment and the two replications. The teams participating in the baseline experiment and the first replication develop the same project. This is a moderately complex project where participants design and implement a software system. The teams apply an adaptation of the XP agile methodology [11,13,14]. All team members collaborate on software development. Each team establishes its work plan. Sometimes the whole team works simultaneously on the same design, algorithm, code or test, whereas on other occasions tasks are shared out and developments are checked by other team members before they are built into the system.

The teams in the second replication analyze and design a software system based on a requirements specification for the same project. In this case, the teams apply the Unified Software Development Process [41] to develop the software. Each team establishes its work plan. The iterations are performed over short time periods (weekly), as a result of which errors or development modifications are revealed sooner and are easier for students to correct.

The aim of the research is to take this experimentation forward. Individual studies yield preliminary results. It is good experimental practice in software engineering to replicate at the same site or at other sites to find out whether the results are consistent. We then synthesized the results by means of meta-analysis. Meta-analysis is used in order to combine studies: (a) that do not use exactly the same metrics or (b) where raw data are not available. In our case, the software product quality response variable uses a different metric for each of the two different software product types in the second replication as compared to the baseline quasi-experiment and the first replication. Additionally, there is no way of converting the evaluation conducted in the second replication to the equivalent dimension of the evaluation conducted in the baseline quasi-experiment and its first replication. Under these circumstances, a joint analysis (e.g., a blocked ANOVA) is not feasible. Meta-analysis can overcome these difficulties.

In summary, the idea of replication is to check whether the results hold. If they do, the results are more reliable, even if we do not discover anything new. By means of aggregation we get a larger sample. This increases the statistical power (compared with individual experiments), which then leads to statistically significant results.

This paper is structured as follows. Section 2 describes related work on team building in the software process. Section 3 presents the methodological details of the study and the methods for statistical meta-analysis. Section 4 summarizes the results of the analysis. In Section 5, we discuss implications for theory and practice. Finally, Section 6 states the conclusions of this research.

2. Related work

More and more software engineering research focuses on studying team composition and, therefore, the personality of team members. However, they do not all analyze its effect on the team task performance and team member satisfaction. Feldt et al. [30,31] conducted a comprehensive survey with 47 software engineers at ten Swedish software organizations in order to investigate the relations between personality traits and software engineering related views, attitudes and work preferences. Some of their results indicate that personality dimension conscientiousness correlates with certain attitudes towards work style, openness to change and task preference, and higher levels of openness to experience is linked to preferring to take responsibility for a whole project and not individual parts. Research conducted by Peslak [56] presents a descriptive and correlational experimental study to evaluate the impact of personality with respect to team processes, project success and, finally, team personality diversity. The results suggest that personality is not related to team processes but does have to do with project success, whereas team personality diversity is unconnected with project success. Sfetsos et al. [61] ran a controlled experiment to research pair programming from the viewpoint of developer personality composition and how this affects the effectiveness of the pair. The results show that communication, effectiveness and workability is better in pairs with heterogeneous personalities. In their empirical study, Walle and Hannay [65] researched the nature of collaboration in pair programming and the effects of personality on collaboration in pair programming. They concluded that personality affects collaboration and personality diversity increases communicativeness and collaborativeness.

Some correlational studies have been conducted in academia [58–60] to analyze the relationship of personality factors like conscientiousness, openness to experience, or neuroticism, respectively, with the performance of software developers that practice pair programming. The results of the first study suggest that conscientiousness does not have a significant effect on performance, although this might be due to the fact that the tasks performed throughout the experiment were short. However, this study also examined the other personality factors, and the results revealed that openness to experience had a direct positive correlation with pair performance. The second experiment did not find any relationship between neuroticism and performance.

Cruz et al. [23] presented a systematic literature review on personality in software engineering covering some of the above research. The results suggest that most of the analyzed studies report empirical research on the influence of personality on both pair programming and team effectiveness. They also signify that empirical studies have not been widely replicated and that have not tested models proposed in theoretical studies. They conclude hence that empirical studies require replication to consolidate knowledge that is potentially useful for new research and is likely to influence software engineering practice. The research reported here presents a meta-analysis of replications that we have conducted of the baseline experiment [2,3].

Besides, there is very little literature on studies including team climate in the field of software engineering. The study conducted by Thamhain and Wilemon [63] examines team effectiveness and team member satisfaction to determine which factors have a positive (interest and motivation, technical management and leadership, professional development, etc.) and negative (unclear team goals, low safety at work, power struggles, bad management, etc.) influence. Climate factors include participative safety. The concept of participative safety appears to be related to the concept of trust discussed in virtual teams [42]. They found that the perception of high team trust is related to performance.

Seger et al. [62] researched the idea of relationships between specific indices of organizational climate (team/management climate), level of individual self-efficacy as a personality attribute, and software practitioners' readiness for agile software development. The research results can help organizations predict the readiness of employees to implement agile methods and/or to work effectively in an agile environment.

Goparaju et al. [33] classified the factors affecting software development team performance and stressed the soft (non-technical) factors affecting the performance of software development teams. They found that soft factors, such as team climate, team diversity, team innovation, team member competencies and characteristics, top management support and team leader behavior, have an effect on software development team performance. Mutual trust and communication effectiveness are found to be the prioritized factors affecting software development team performance.

Some researchers have examined factors like developer team cohesion, satisfaction, team productivity, etc. to compare agile methods and traditional methods [27,48]. But none of the reviewed papers examines the relationships between the Big Five personality factors, team climate and software product quality or team satisfaction, and they therefore have a different goal to our research. In our research, we focus on these relationships during the development of small-sized software.

Table 1 summarizes the studies analyzed individually above and considered relevant to this research. Table 1 lists the personality traits and climate aspects considered by each study that warrant inclusion in this research with respect to performance and/or team satisfaction.

3. Method

The baseline quasi-experiment and its two replications are designed as correlational research [21]. The baseline quasi-experiment has two parts. Each part was published separately. The first part analyses the relationships of development team personality factors with respect to developed software product quality and team member satisfaction. This research is described in Acuña et al. [3]. The second part focuses on the study of team climate [5–7] with respect to product quality and software development team member satisfaction in the quasi-experimental study. This

Table 1
Aspects analyzed in the related work.

	Personality					Climate		Performance	Satisfaction
	Team composition	Team diversity	Conscientiousness	Openness to Experience	Neuroticism	Team Climate	Participative Safety		
Peslak [56]	X	X						X	
Sfetsos et al. [61]	X	X						X	X
Walle and Hannay [65]	X	X							
Feldt et al. [30,31]			X	X	X				
Salleh et al. [58]			X					X	
Salleh et al. [60]				X				X	
Salleh et al. [59]					X			X	
Thamhain and Wilemon [63]						X		X	X
Jarvenpaa and Leidner [42]							X	X	X
Seger et al. [62]						X		X	X
Goparaju et al. [33]	X	X				X		X	
Dybå and Dingsøyr [27]								X	X
Macias et al. [48]								X	X

Table 2
Independent and dependent variables of the empirical study.

Independent variables			Dependent variables
General	Specific		
Personality	– Neuroticism – Extraversion – Openness to experience – Agreeableness – Conscientiousness		Satisfaction Software quality
Team Climate	– Preferences – Perceptions	– Participative safety – Support for innovation – Team vision – Task orientation	

research is reported in detail in Acuña et al. [2]. In the following sections we describe the goals, hypotheses, variables, participants, subjects, measurement instruments, and quasi-experimental procedure of the three studies. This report conforms to the guidelines for reporting empirical research in SE [43].

3.1. Goals, hypotheses and variables

The goal of this research is to analyze the relationships between two independent variables and two dependent variables. The two independent variables are:

- Big Five personality factors (neuroticism, extraversion, openness to experience, agreeableness and conscientiousness) at team level.
- Climate preferences and perception factors within a team (participative safety, support for innovation, team vision and task orientation).

The two dependent variables are:

- Quality of the developed software.
- Mean satisfaction of team members.

Input and process measures (personality, team climate preference, team climate perception and satisfaction) were collected at the individual level and aggregated to team level. In order to build this team construct, the intra-class correlation (ICC) index can be used to decide whether data can be aggregated. The ICC(1) index compares the inter-group with the intra-group variance [20,28,46]. The higher the ICC(1) index is the greater the variance at the individual level attributable to the relevant team will be. Normally, an ICC(1) of over 0.20 is considered to justify aggregation [29]. In all three studies, all the aggregate variables are significantly higher than this threshold. For example, this applies in the baseline quasi-experiment for: (a) the neuroticism (0.55), extraversion (0.53), openness to experience (0.54), agreeableness (0.52), and conscientiousness (0.52) personality factors; (b) the participative safety preferences (0.55), support for innovation preferences (0.53), team vision preferences (0.54) and task orientation preferences (0.52), participative safety perceptions (0.60), support for innovation perceptions (0.58), team vision perceptions (0.59) and task orientation perceptions (0.55) team climate factors; and (c) the team member satisfaction (0.56) response variable. Therefore, we can generate team-level variables by aggregating the scores of the members of each team. Particularly, we used the arithmetic mean as a measure for aggregating the data for each variable examined in this study at group level in order to represent the average behavior, team climate preferences and perceptions and satisfaction, except for the software quality response variable which was measured directly for the team.

The team's instructor provided the deliverable ratings, and then we assessed the quality of the software product developed by each team.

Table 2 summarizes the independent and dependent variables analyzed in all three quasi-experiments.

Table 3 lists the alternative hypotheses derived from the objectives defined for the empirical study and the factors that they involve.

In the studies non-professional subjects (undergraduate students) develop a (toy) software project. The baseline quasi-experiment uses an adaptation of the agile XP method within a laboratory environment (on-line). However, the second replication uses an incremental iterative process to develop a software product.

We have identified three variables that we do not intend to study but might affect the results of the study (Table 4).

- Previous software development knowledge and experience of software development team members. This variable is likely to influence the quality of the resulting software (dependent variable). It is composed of two issues: software design experience and procedural programming experience. The study participants are students with similar levels of knowledge and ability. Specifically, the students taking the course are normally distributed. This variable is blocked using a random method of assigning participants to teams, which are the subjects of the quasi-experimental study. Randomization assures that two or more teams are likely to be equivalent. This control technique prevents undesired variables systematically affecting the study results.
- Software development project (functionalities that the subjects are to implement during the experiment). The system under development could have an effect on the quality of the resulting software. In this case, the proposed project is equivalent for all teams. If the projects were of differing complexity or chosen by each team, this would be a factor that would affect the resulting software. If the teams work on problems of differing complexity, the results of the teams may not be comparable. This would invalidate the findings. This variable is blocked by setting the same software development project for all teams or a project of equivalent complexity.

Knowledge of the proposed software development method (XP for two of the quasi-experiments and incremental iterative development for the other quasi-experiment). Team productivity may vary depending on team member knowledge of the proposed development method. In order to control this variable and try to establish a similar level of knowledge, all study participants receive training sessions on the XP and incremental iterative methods generally and especially on the XP practices applied in the project.

3.2. Subjects

The baseline quasi-experiment was conducted with 2nd-year computing students at the Universidad Autónoma de Madrid's School of Computing during the 2004/05 academic year. The initial study was first replicated during the 2005/06 academic year again with 2nd-year students from the same university. The second replication was performed with senior undergraduate computing students at the Universidad Politécnica de Madrid's School of Computing during the 2005/2006 academic year. Table 5 shows the characterization of the quasi-experiment and the replications conducted.

A total of 105 students participated in the initial empirical study, of which 83 were male (79%) and 22 were female (21%). A total of 66 students participated in the first replication, of which 50 were male (75.8%) and 16 were female (24.2%). A total of 136 students participated in the second replication, of which 98 were male (72.1%) and 38 were female (27.9%).

The teams were formed at random and their members were blind to the quasi-experimental conditions and hypotheses. Teams used a common agile methodology (an XP [13] adaptation to the educational environment) and C or C++ programming languages in the first two quasi-experiments, whereas an incremental iterative process was used for software development in the second replication. In all three studies development took place throughout the whole semester, and instructors played the role of users.

3.3. Data collection procedure

The design of the quasi-experimental study is divided into two phases: before and after. *Before* refers to when the project kicks off, the time during which the teams are being set up, but no team work has yet been done. This is the period when the NEO FFI personality test and TSI questionnaire were handed out and collected back in. *After* is, according to estimated project time, when the project is 95% complete (week 23) just before project development comes to an end. This period coincides with the end of the course and is when the developed software is delivered. The questionnaires handed out at this point measure the TCI climate perception and the satisfaction of the teams after their team work. Table 6 shows when the questionnaires are administered across the different quasi-experiment phases.

All the personality factors, as well as the team climate preferences and perceptions and satisfaction are evaluated on Likert scales in the questionnaires listed in Table 6. Subjects were told that there are no right or wrong answers and that their responses should be accurate and truthful, taking into account the scoring system (1 = totally disagree, 5 = totally agree).

The questionnaires were administered to all individuals on each of the courses. The result is a separate score for each individual and each variable. As already mentioned, because this research was performed at team level and the questionnaires were administered to individuals, the data had to be aggregated in order to build the

Table 3
Hypotheses associated with the empirical study.

Factors		Response variables	
		Quality	Satisfaction
Personality	Neuroticism Extraversion Openness to experience Agreeableness Conscientiousness	H_1. – There is a relationship between all team member *personality* factors and developed software *quality*	H_2. – There is a relationship between all team member *personality* factors and software development team *satisfaction*
Climate preference	Participative safety Support for innovation Team vision Task orientation	H_3. – The team climate factor *preference* is related to a better quality software product	H_5. – The team climate factor *preference* is related to a better software development team *satisfaction*
Climate perception	Participative safety Support for innovation Team vision Task orientation	H_4. – The team climate factor *perception* is related to a better quality software product	H_6. – The team climate factor *perception* is related to a better software development team *satisfaction*

Table 4
Controlled variables in the empirical study.

Controlled variable	Control mode
Software development knowledge and experience	**Randomization** used to form teams by a purely mechanical method (flipping a coin)
Software development project	Assignment of all teams to **the same or an equally complex project**
Knowledge of the proposed software development method	**Homogenization** of knowledge through training

Table 5
Characterization of quasi-experiments.

	Quasi-experiment UAM 0405	Replication no. 1 UAM 0506	Replication no. 2 UPM 0506
Year/total years	2nd/4	2nd/4	5th/5
Subject	Data Structures and Algorithms	Data Structures and Algorithms	Software Engineering
Duration	4 months	4 months	9 months
Participants	105	66	136
Teams	35	22	34
Participants/team	3	3	4
Task	Design and Implementation	Design and Implementation	Analysis and Design
Development process type	Adapted XP Process	Adapted XP Process	Unified Process

Table 6
Phases and measurement instruments in the empirical study.

Measured variables	Before	After	Code and project documents
Personality	Spanish version of the NEO FFI test [22]		
Team climate preferences	TSI [4]		
Team climate perceptions		TCI [5,7]	
Satisfaction		Gladstein's questionnaire [32]	
Software quality			Developed software product rating

team construct. Therefore, we can generate team-level variables by aggregating the scores of the members of each team. We used the arithmetic mean as a measure for aggregating the data for each variable examined in this study at group level, except for the software quality response variable that was measured directly for the team.

All these Spanish version questionnaires were anonymized. The teams were identified by a team number and the participants by a team member number (1, 2, 3 or 4). This subject and participant identification was the same for all the questionnaires at each measurement time. Information was gathered by means of measurements taken before and after project development. The questionnaires were handed out and collected back in on the dates established previously for each phase of each study. When the students handed in the questionnaires, we looked through them to check that they had answered all the questions.

The final quality of the software products output by the teams was evaluated when all the projects had been completed, although each of the deliverables submitted by the teams throughout the course was evaluated separately. To measure the quality of the software product in the baseline experiment and the first replication, we analyzed the code and product documents. The criteria and metrics used to evaluate the quality of the design and the code generated by the team were taken from SWEBOK 2004 [40]: modularization (number of modules and coupling), testability (number of defects detected by the automatically executed test case set), functionality (number of satisfied requirements), reuse (number of reused modules within the project), programming style (guidelines defined at the course website). These criteria were evaluated on a scale from 1 to 4 points. A score of 1 (poor) means that the software product satisfied up to 25% of the aspects that the ideal solution would for each metric. A score of 2 (acceptable) means that software product satisfied from 26% to 50% of the aspects that the ideal solution would. A score of 3 (good) means that the software product satisfied from 51% to 75% of the aspects that the ideal solution would. A score of 4 (excellent) means that the software product satisfied from 76% to 100% of the aspects that the ideal solution would.

The progress made in the exercises was also recorded to quantitatively document each team members' participation. Of the total software product quality grade, 20% assessed participation. The remaining 80% assessed the other criteria. The grade is weighted on a scale from 0 to 10 points. This is how each team's development project was graded using the weighted grading formula (1).

$$\begin{aligned}\text{Grade} = {} & (((\text{Modularization} * 2 + \text{Testability} * 2 \\ & + \text{Functionality} * 2 + \text{Reuse} * 2 + \text{Style} * 2)/4) * 0.8) \\ & + ((\text{Participation} * 10/4) * 0.2). \end{aligned} \quad (1)$$

The above formula was applied to each scheduled deliverable, except for the first deliverable on which reusability was not evaluated. Formula (2) for calculating the final project grade from the partial deliverables was established as:

$$\begin{aligned}\text{Final grade} = {} & \text{Deliverable 1 grade} * 0.20 \\ & + \text{Deliverable 2 grade} * 0.20 \\ & + \text{Deliverable 3 grade} * 0.30 \\ & + \text{Deliverable 4 grade} * 0.30. \end{aligned} \quad (2)$$

Software quality was rated in the second replication according to the grading of the projects developed by the teams through expert assessment. Grades are the result of analyzing the delivered documentation and each team member's participation. The subject instructors provided the grades after correcting and rating the generated products. The team project quality was rated on a scale from 0 to 3, where 0 is the lowest grade and 3 is the top grade. The rubric was based on the correctness of the following software products: use case diagram, extended use cases, system sequence diagram, operation contracts, domain model, design sequence diagrams and class diagram. Special attention was paid to product consistency (e.g., the system sequence diagram and operation contracts match). All projects that correctly specify the system are graded equally, that is, there is no one correct solution, and there are many error types or omissions since there are a lot of artefacts to be checked. For example, special attention was paid to the correctness of extends and includes relationships in use case diagrams. Extended use cases were checked to assure that the sequence of both principal and alternative events was correct, the messages contained the correct I/O data, etc. Errors or omissions were not penalized individually but weighted within the context of the respective artefact. If the use case or contract was understandable and reasonable despite the errors, the penalization was low. If not, the penalization was higher. The penalizations ranged from 0.01 to 0.2 depending on the gravity of the error or omission.

The distribution of the response variables for both the baseline quasi-experiment and the two replications is normal according the Shapiro–Wilk test. The assumption of normality could not be rejected for the software quality variable with p-values of 0.215, 0.185, and 0.605 for each study covered by the meta-analysis, respectively. Neither could the assumption of normality be rejected for the satisfaction variable with p-values of 0.263, 0.362, and 0.104 for each study covered by the meta-analysis, respectively.

3.4. Internal and external validity

Quasi-experiments, like experiments, have to consider both internal and external validity. Before running the quasi-experiment, we identified six major internal validity threats that were dealt with as follows:

1. *Team member motivation and competencies.* Not all the team members have the same competencies, like intrapersonal and interpersonal skills, or motivation. This threat was mitigated by forming teams at random.
2. *Knowledge of XP and Unified Process and development experience.* Not all teams have the same knowledge of XP or the Unified Process and the same experience in the task to be developed. This threat was mitigated by forming teams at random and organizing a training session on the XP methodology and Unified Process applied to project development.
3. *Participant attendance.* In order to avoid (or at least to account for) different student training, we made it compulsory for students to attend classes and included attendance in the grading of the developed software product. The measure was successful since attendance was high.
4. *Process conformance.* It is hard to guarantee that the students do all the specified tasks in the required order or adequately schedule their work. This threat was mitigated by allocating two instructors to every 60 students. These instructors were responsible for monitoring project development. Specifically, the instructors made a note of the tasks performed and their order in an observations column on the team lists for each session. If team members were not doing the specified tasks in the right order, instructors discussed this question with the teams and went through the process guide supplied to participants. Process conformance was generally satisfactory.
5. *Potential impact of XP on climate.* Agile methods are known to facilitate and motivate team building. So, the usage of the agile XP process might influence the climate and the software built and leads to less generalizable study results. In our empirical study, however, the XP practices were confined to programmer practices related to technical design, construction and testing activities. Coaching practices that could potentially influence climate and therefore the results, such as participative sessions aimed at achieving accepted responsibility and cover for the team or at conveying a faithful reflection of the state of affairs [12], were not applied.
6. *Software project to be developed.* The results of the teams working on different problems or problems of different complexity might not be comparable. To prevent this threat, all the teams in each study developed the same or an equally complex project according to a structured or object-oriented approach.

With respect to external validity, our results are generalizable to academic environments, as the study participants were students. For them to be generalized to an industrial setting, special-purpose quasi-experimental studies at software developer organizations would have to be planned and designed. However, if we do detect any positive relationships, the resulting recommendations, reinforced by the meta-analysis, might be applicable in the software industry, as has proved to be the case with some other aspects of software development [38]. Additionally, such relationships could be used to formulate the research hypotheses to be tested in software developer organizations.

4. Statistical analysis

We conducted separate meta-analyses for each of the independent variables on the two dependent variables software quality and satisfaction. We used the meta-analysis technique of correlations based on Z transformation. We used Comprehensive Meta-Analysis[1] v2 to calculate the meta-analytic estimates (Comprehensive Meta-Analysis is a trademark of Biostat Inc.).

We estimated the meta-analytic effect size both under the assumption of the random-effects model and under the fixed-effects model. These models lead to different significance tests and confidence intervals for meta-analytic results [39].

The fixed-effects model assumes an unknown and fixed population effect size that is estimated by the studies in the meta-analysis. All the studies in the meta-analysis are seen as drawn from the same population, and variances in effect sizes between individual studies are viewed as due to subject variability [36].

The random-effects model, on the other hand, assumes an unknown and stochastic population effect-size distribution. That is, the true effect of, say, extraversion varies around a mean μ. This accounts for the possibility of the effects of personality or team climate depending on situational variables and other factors (both known and unknown) that are not taken into consideration in the analysis. Variances in effect sizes are then seen as being due to subject variability and also to inter-study variability, since each study is seen as approximating a different part of the true effect size distribution [16,36].

Both models relate to an unknown population parameter. The approaches are hence referred to as *unconditional* [47]. The choice of which model to use is based on theory, past empirical findings, and on insights as to what the included studies describe. This choice is made prior to the analysis.

However, the analysis techniques associated with the two models can also be used merely to characterize the studies with respect to each other without any reference to a population effect size. Heterogeneity measures of the observed data are used to determine which model to use. The heterogeneity measures are

[1] Standard open-source statistical packages such as R could have been used to conduct all statistical calculations in our study.

calculated under the assumption of a fixed-effects model. If heterogeneity is non-significant, a fixed-effects model is an appropriate characterization of data. Otherwise, a random-effects model best characterizes the data. The results of such *conditional* approaches should, however, not be confounded with statements regarding population parameters.

We conducted our analysis from both an unconditional and a conditional perspective. For the unconditional perspective, we focus on the fixed-effects model, because the studies are all close replications of each other, and we assume that situational variables and possible moderating factors are similar in all three.

In the conditional approach, we tested whether there were genuine differences underlying the results of the studies (heterogeneity), or whether the variation in the findings was compatible with mere chance.

In the following, we give an overview of the technical details of the meta-analysis. For further elaborations, see e.g. [16,36,50]. In simple terms, the meta-analytic effect size is estimated by a weighted average of the effect sizes of the individual studies, where the weights are computed on the basis of each study's variance (with the inter-study variance added to each study variance when using the random-effects model).

In our case, effects are correlations. Although not as common referred to in terms of "effect sizes" as, e.g., differences between middle values, correlations are, nonetheless, treated in the meta-analysis literature as expressing effects. However, when faced with correlational data, an additional step is required prior to the meta-analysis proper. The standard error, and hence the variance of a correlation is a function not only of sample size but also of the magnitude of the correlation, i.e. the actual effect size: a larger positive or negative correlation has a smaller standard error. Thus, the weight given to a particular study in the meta-analysis would depend on the study's effect size, which is clearly not intended. We avoid this problem by first converting all correlations to Fisher's Z metric (which should not be confused with the Z statistic), whose standard error is not confounded with effect size. All computations for the meta-analysis are then performed using Fisher's Z, and then the results are converted back to the original correlational measure. Let ρ be a correlation. The transformation from the correlation to Fisher's Z is given by

$$Z = \frac{1}{2}\ln\frac{1+\rho}{1-\rho} = \tanh^{-1}(\rho), \quad (3)$$

and the standard error of Z is

$$\sigma_Z = \frac{1}{\sqrt{N-3}}. \quad (4)$$

Let k be the number of studies in the meta-analysis. Let T_i be the standardized effect-size estimate of study i (in our case, T_i is Fisher's Z transformation of the correlation estimate r_i of study i). In the fixed-effects model, the estimate $\overline{T_\bullet}$ of the assumed fixed population effect size, and the estimate's variance $v_\bullet$, are

$$\overline{T_\bullet} = \frac{\sum_{i=1}^{k} w_i T_i}{\sum_{i=1}^{k} w_i} \qquad v_\bullet = \frac{1}{\sum_{i=1}^{k} w_i}, \quad (5)$$

where $w_i = 1/v_i$ is the weight assigned to study i, i.e. the reciprocal of the variance v_i for study i. Thus, $\overline{T_\bullet}$ is a weighted mean over the effect sizes of the individual studies, where studies with less variance are given greater weight. In the random-effects model, the weights are based on between-study variance in addition to within-study variance v_i. Specifically, the estimate $\overline{T_\bullet^\#}$ of the mean μ of the assumed population effect-size distribution, and the estimate's variance $v_\bullet^\#$, are

$$\overline{T_\bullet^\#} = \frac{\sum_{i=1}^{k} w_i^\# T_i}{\sum_{i=1}^{k} w_i^\#} \qquad v_\bullet^\# = \frac{1}{\sum_{i=1}^{k} w_\bullet^\#}, \quad (6)$$

where $w_i^\# = 1/v_i^\#$, for $v_i^\# = v_i + \tau^2$. Here, τ^2 is the additional between-study variance

$$\tau^2 = \begin{cases} \frac{Q-df}{C} & \text{if } Q > df \\ 0 & \text{if } Q \leqslant df \end{cases}, \quad (7)$$

where the degrees of freedom $df = k - 1$, and Q represents the total variance:

$$Q = \sum_{i=1}^{k} w_i (T_i - \overline{T_\bullet})^2. \quad (8)$$

In Eq. (7), C is simply a scaling factor that ensures that τ^2 has the same denomination as within-study variance, i.e. $C = \sum w_i - \sum w_i^2 / \sum w_i$.

In fact, Q is a statistic that indicates heterogeneity, and the one that we used for this purpose. A significant Q rejects the null hypothesis of homogeneity and indicates that the variability among the effect sizes is greater than what is likely to have resulted from subject level variability alone [36]. We also calculated the I^2 statistic, which indicates heterogeneity in percentages:

$$I^2 = \begin{cases} 100\%(Q - df)/Q & \text{if } Q > df \\ 0\% & \text{if } Q \leqslant df \end{cases}. \quad (9)$$

A value of 0 indicates no observed heterogeneity, whereas 25% is equivalent to low, 50% to moderate, and 75% to high heterogeneity [37].

5. Results

We present the meta-analysis results in terms of personality factor effects on software quality and satisfaction and team climate factor effects on software quality and satisfaction.

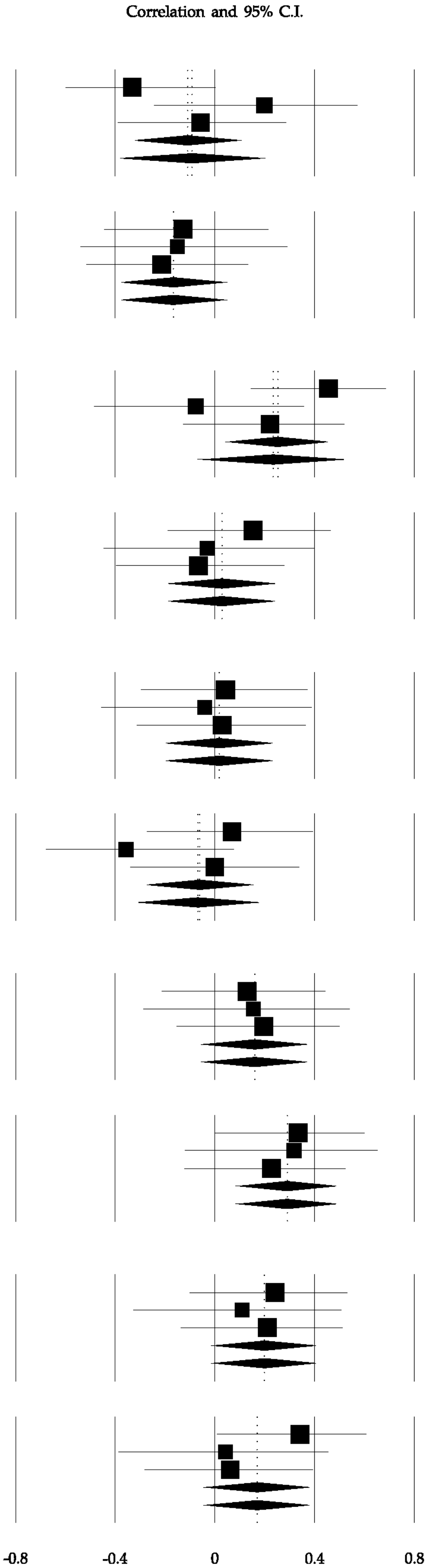
5.1. Effects of team personality on software quality and on satisfaction

Fig. 1 summarizes the meta-analyses for each of the five personality factors on software quality and satisfaction.

All estimates are given in the original metric (i.e. not in terms of Fisher's Z, which is only used in the calculations). For each meta-analysis, the *Overall fixed model* row reports the meta-analytic estimate tanh $(\overline{T_\bullet})$ under the fixed-effects model, its 95% confidence interval, along with the probability p of observing the calculated estimate under the hypothesis that the true population correlation is zero. Similarly, the *Overall random model* row gives the meta-analytic estimate tanh $(\overline{T_\bullet^\#})$ under the random-effects model. The heterogeneity measures are given on the far right of the *Overall fixed model* row (Fig. 1). Heterogeneity is in the small-to-medium region for *Neuroticism/Software Quality, Extraversion/Software Quality* and *Openness/Satisfaction*, although heterogeneity is non-significant in all three cases. In all three cases, the effects in study UAM 0506 are opposite to the other two studies.

Table 7 shows the correlations for the meta-analyses of the personality factors that have statistically significant relationships with respect to software quality and satisfaction. Additionally, qualitative information is added for each quasi-experiment. Thus, the direction of the correlation (positive or negative) is symbolized by a '+' or '–' sign, and its significance level by the darkness of the shading (a darker shade is more significant). It also specifies the heterogeneity indices of the quasi-experiments in the meta-analysis.

Unlike the other two studies (Fig. 1), the value for the extraversion personality factor with respect to software quality is negative



Study	Correlation	95% C.I.	p	Relative weight fixed	Relative weight random	Heterogeneity Q	Heterogeneity p	Heterogeneity I^2
Neuroticism								
Software Quality								
UAM 0405	-.33	-.60 .00	.05	39.02	36.63			
UAM 0506	.20	-.24 .57	.38	23.17	27.33			
UPM 0506	-.06	-.39 .29	.75	37.80	36.05			
Overall fixed model	-.11	-.32 .11	.32			3.69	0.16	45.73
Overall random model	-.09	-.37 .20	.54					
Job Satisfaction								
UAM 0405	-.13	-.44 .21	.46	39.02	39.02			
UAM 0506	-.15	-.54 .29	.51	23.17	23.17			
UPM 0506	-.22	-.52 .13	.22	37.80	37.80			
Overall fixed model	-.17	-.37 .05	.13			0.13	0.94	0.00
Overall random model	-.17	-.37 .05	.13					
Extraversion								
Software Quality								
UAM 0405	.46	.14 .68	.01	39.02	36.44			
UAM 0506	-.08	-.48 .36	.73	23.17	27.65			
UPM 0506	.22	-.13 .52	.21	37.80	35.91			
Overall fixed model	.25	.04 .44	.02			3.92	0.14	0.00
Overall random model	.23	-.07 .50	.13					
Job Satisfaction								
UAM 0405	.15	-.19 .46	.38	39.02	39.02			
UAM 0506	-.03	-.45 .40	.89	23.17	23.17			
UPM 0506	-.07	-.40 .28	.71	37.80	37.80			
Overall fixed model	.03	-.19 .24	.80			0.86	0.65	0.00
Overall random model	.03	-.19 .24	.80					
Openness								
Software Quality								
UAM 0405	.04	-.30 .37	.81	39.02	39.02			
UAM 0506	-.04	-.46 .39	.85	23.17	23.17			
UPM 0506	.03	-.31 .36	.87	37.80	37.80			
Overall fixed model	.02	-.20 .23	.87			0.09	0.96	0.00
Overall random model	.02	-.20 .23	.87					
Job Satisfaction								
UAM 0405	.07	-.27 .39	.70	39.02	37.99			
UAM 0506	-.36	-.68 .07	.10	23.17	24.94			
UPM 0506	.00	-.34 .34	1.00	37.80	37.07			
Overall fixed model	-.06	-.27 .15	.58			2.51	0.28	0.00
Overall random model	-.07	-.30 .17	.59					
Agreeableness								
Software Quality								
UAM 0405	.13	-.21 .44	.46	39.02	39.02			
UAM 0506	.15	-.29 .54	.50	23.17	23.17			
UPM 0506	.20	-.15 .50	.27	37.80	37.80			
Overall fixed model	.16	-.06 .36	.14			0.07	0.96	0.00
Overall random model	.16	-.06 .36	.14					
Job Satisfaction								
UAM 0405	.33	.00 .60	.05	39.02	39.02			
UAM 0506	.32	-.12 .65	.15	23.17	23.17			
UPM 0506	.23	-.12 .52	.20	37.80	37.80			
Overall fixed model	.29	.08 .47	.01			0.24	0.89	0.00
Overall random model	.29	.08 .47	.01					
Conscientiousness								
Software Quality								
UAM 0405	.24	-.10 .53	.17	39.02	39.02			
UAM 0506	.11	-.33 .51	.63	23.17	23.17			
UPM 0506	.21	-.14 .51	.24	37.80	37.80			
Overall fixed model	.20	-.01 .40	.07			0.23	0.89	0.00
Overall random model	.20	-.01 .40	.07					
Job Satisfaction								
UAM 0405	.34	.01 .61	.04	39.02	39.02			
UAM 0506	.04	-.39 .45	.86	23.17	23.17			
UPM 0506	.06	-.28 .39	.73	37.80	37.80			
Overall fixed model	.17	-.05 .37	.12			1.78	0.41	0.00
Overall random model	.17	-.05 .37	.12					

Fig. 1. Meta-analyses of correlations on software quality and job satisfaction of Five Factor Model traits.

Table 7
Significant correlations and heterogeneity of the personality traits with software quality and satisfaction.

RESPONSE VARIABLES	FACTORS	UAM 0405	UAM 0506	UPM 0506	META-ANALYSIS		
					CORRELATION	HETEROGENEITY INDICES	
						Q	P
Software Quality	**Extraversion**	+		+	**0.25***	3.92	0.14
Satisfaction	**Agreeableness**	+			**0.29***	0.24	0.89

*α<0.05

for the UAM 0506 replication. However, Table 7 shows that the result of the meta-analysis does not suggest that they are heterogeneous. The setup of the first two studies, UAM 0405 and UAM 0506, is apparently identical (see Table 5): site, instructor, year, participants/team, subject, duration, task, development process type, etc. There are, however, some differences, such as sample size and instructor experience, which might be causing the results to vary and acting as moderator variables in the experiment. The results are rendered unpredictable by the influence of these unstudied variables on quasi-experiments, which falsifies the apparent homogeneity of the studies. As we mentioned earlier, the setup of the first two studies, UAM 0405 and UAM 0506, and the third study, UPM0 506, is different (see Table 5). The meta-analysis suggests that there is a significant correlation between the factor extraversion factor and software quality. However, the correlation level is very low. This would limit the usefulness of controlling for it in real-life software development projects. Additionally, the quasi-experiments, UAM 0405 and UPM 0506, already returned significant correlations (with different confidence levels) for this relationship. Until we identify the moderator variables, our findings are limited to this provisional result of the meta-analysis. Thus we can claim that, irrespective of team size, software development type (XP Process or Unified Process) and development task type (design and implementation or analysis and design):

- There is a relationship between the extraversion factor and software quality.
- There is no relationship between neuroticism, openness experience, agreeableness and conscientiousness factors and software quality.

Table 7 shows that the correlation of the agreeableness factor with respect to satisfaction is significant in the meta-analysis, but the correlation level is low. This relationship was significant at a 95% confidence level in the UAM 0405 study, but not in the other two replications, UAM 0506 and UPM 0506. The meta-analysis suggests that all the quasi-experiments are homogeneous (Fig. 1 and Table 7). A visual examination of the forest plot, where the confidence intervals and the result of the meta-analysis overlap, also points to homogeneity. As already mentioned, the setup of the first two studies, UAM 0405 and UAM 0506, is different to the third, UPM 0506 (see Table 5), whereas the setup of the first two studies, UAM 0405 and UAM 0506, is identical (see Table 5). We have confirmed by means of the meta-analysis the processes that common sense dictates take place during teamwork, that is, team members with an agreeable personality are more friendly with and attentive to others. This raises satisfaction with team membership and makes them want to continue working together. Team size may account for this *agreeableness/satisfaction* relationship. Relationships are easier in small teams. Satisfaction with team membership comes quicker in smaller than in larger teams. This supports the results suggesting that, irrespective of software development type (XP, Unified Process) and development task type (design and implementation or analysis and design):

- The relationship of the agreeableness personality factor with respect to satisfaction is significant.
- There is no relationship between neuroticism, extraversion, openness experience and conscientiousness factors and satisfaction.

5.2. Effects of team climate on software quality and satisfaction

As already mentioned, we analyzed each of the team climate factors for climate preferences at the start of development (Fig. 2) and climate perceptions after software system development (Fig. 3). The results of the meta-analysis for each climate factor are discussed below.

Most team climate preference factors: participative safety, support for innovation and task orientation factors did not have a significant correlation with respect to the **software quality** in any of the studies. Only the team vision factor had a confidence level of 90% for the UAM 0405 study. However, the correlation coefficient output by the meta-analysis for these four factors is not significant either. The results for these factors suggest that all the quasi-experiments are homogeneous (Fig. 2). Therefore, we have empirically confirmed the processes that common sense dictates take place during teamwork, that is, team climate preferences do not have an impact on the quality of the software produced by the team.

A similar thing applies to all four team climate preferences with respect to satisfaction. Most of the team climate preference factors: participative safety, team vision and task orientation factors did not have a significant correlation with respect to the **satisfaction** in any of the studies. The support for innovation factor had a confidence level of 90% level in the UPM 0506 study only. But the correlation coefficients output by the meta-analysis for these four factors are not significant either. The results for these factors suggest that all the quasi-experiments are homogeneous (Fig. 2). We have empirically confirmed the processes that common sense dictates take place during teamwork, that is, team climate preferences do not have an impact on satisfaction among team members.

In contrast to the questionnaire administered prior to teamwork, the inventory taken toward the end of the teamwork shows more substantial effects. Fig. 3 shows small-to-large overall effects of team climate perceptions. Only *support for innovation/software quality* and *team vision/software quality* are non-significant. Heterogeneity is in the medium region (non-significant) for *support for*

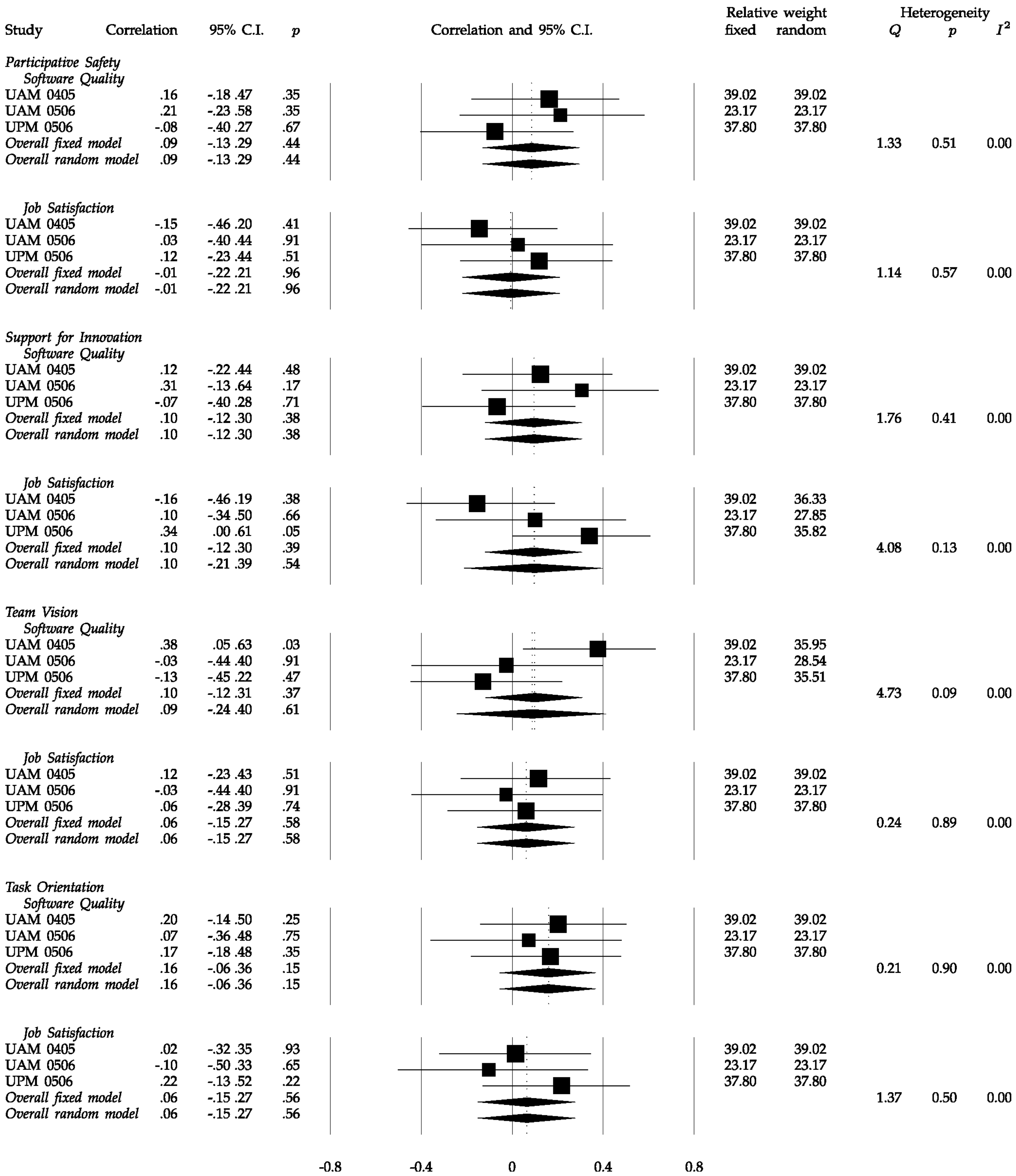


Fig. 2. Meta-analyses of correlations on software quality and job satisfaction of team climate.

innovation/software quality, *support for innovation/satisfaction*, and *team vision/software quality*.

Table 8 shows that there was a significant correlation of the participative safety factor again with respect to software quality in all studies. The task orientation factor perception in replication UPM 0506 also had a significant correlation. Apart from these values, Table 8 also shows that correlation coefficients output by the meta-analysis are significant for the perception of both factors. The meta-analysis suggests that the studies of the participative safety and task orientation team climate factor perceptions are homogeneous. A visual examination of the forest plot, where the confidence intervals and the result of the meta-analysis overlap, also points to homogeneity. Remember that the setup of the first to studies, UAM 0405 and UAM0506, and the third, UPM 0506, is different (see Table 5). So, we have confirmed by meta-analysis that participative safety and task orientation team climate factor

Study	Correlation	95% C.I.	p	Relative weight fixed	Relative weight random	Heterogeneity Q	Heterogeneity p	Heterogeneity I^2
Participative Safety								
Software Quality								
UAM 0405	.34	.01 .61	.04	39.02	39.02			
UAM 0506	.49	.08 .75	.02	23.17	23.17			
UPM 0506	.43	.11 .67	.01	37.80	37.80			
Overall fixed model	.41	.22 .57	.00			0.38	0.82	0.00
Overall random model	.41	.22 .57	.00					
Job Satisfaction								
UAM 0405	.63	.38 .80	.00	39.02	39.02			
UAM 0506	.66	.34 .85	.00	23.17	23.17			
UPM 0506	.76	.56 .87	.00	37.80	37.80			
Overall fixed model	.69	.56 .79	.00			1.02	0.60	0.00
Overall random model	.69	.56 .79	.00					
Support for Innovation								
Software Quality								
UAM 0405	-.05	-.37 .29	.79	39.02	36.46			
UAM 0506	.47	.06 .75	.03	23.17	27.63			
UPM 0506	.07	-.28 .40	.70	37.80	35.92			
Overall fixed model	.13	-.09 .33	.25			3.90	0.14	0.00
Overall random model	.15	-.16 .43	.34					
Job Satisfaction								
UAM 0405	.70	.48 .84	.00	39.02	36.01			
UAM 0506	.80	.57 .91	.00	23.17	28.42			
UPM 0506	.46	.14 .69	.01	37.80	35.57			
Overall fixed model	.65	.51 .76	.00			4.61	0.10	0.00
Overall random model	.66	.44 .81	.00					
Team Vision								
Software Quality								
UAM 0405	.30	-.04 .57	.08	39.02	37.05			
UAM 0506	.40	-.03 .70	.07	23.17	26.58			
UPM 0506	-.05	-.38 .29	.77	37.80	36.37			
Overall fixed model	.19	-.02 .39	.08			3.23	0.20	0.00
Overall random model	.20	-.07 .45	.15					
Job Satisfaction								
UAM 0405	.50	.19 .71	.00	39.02	39.02			
UAM 0506	.60	.24 .82	.00	23.17	23.17			
UPM 0506	.36	.03 .62	.04	37.80	37.80			
Overall fixed model	.47	.29 .62	.00			1.21	0.55	0.00
Overall random model	.47	.29 .62	.00					
Task Orientation								
Software Quality								
UAM 0405	.22	-.12 .52	.21	39.02	39.02			
UAM 0506	.27	-.17 .62	.22	23.17	23.17			
UPM 0506	.34	.00 .61	.05	37.80	37.80			
Overall fixed model	.28	.07 .46	.01			0.28	0.87	0.00
Overall random model	.28	.07 .46	.01					
Job Satisfaction								
UAM 0405	.71	.49 .84	.00	39.02	39.02			
UAM 0506	.80	.57 .91	.00	23.17	23.17			
UPM 0506	.68	.44 .83	.00	37.80	37.80			
Overall fixed model	.72	.60 .81	.00			0.87	0.65	0.00
Overall random model	.72	.60 .81	.00					

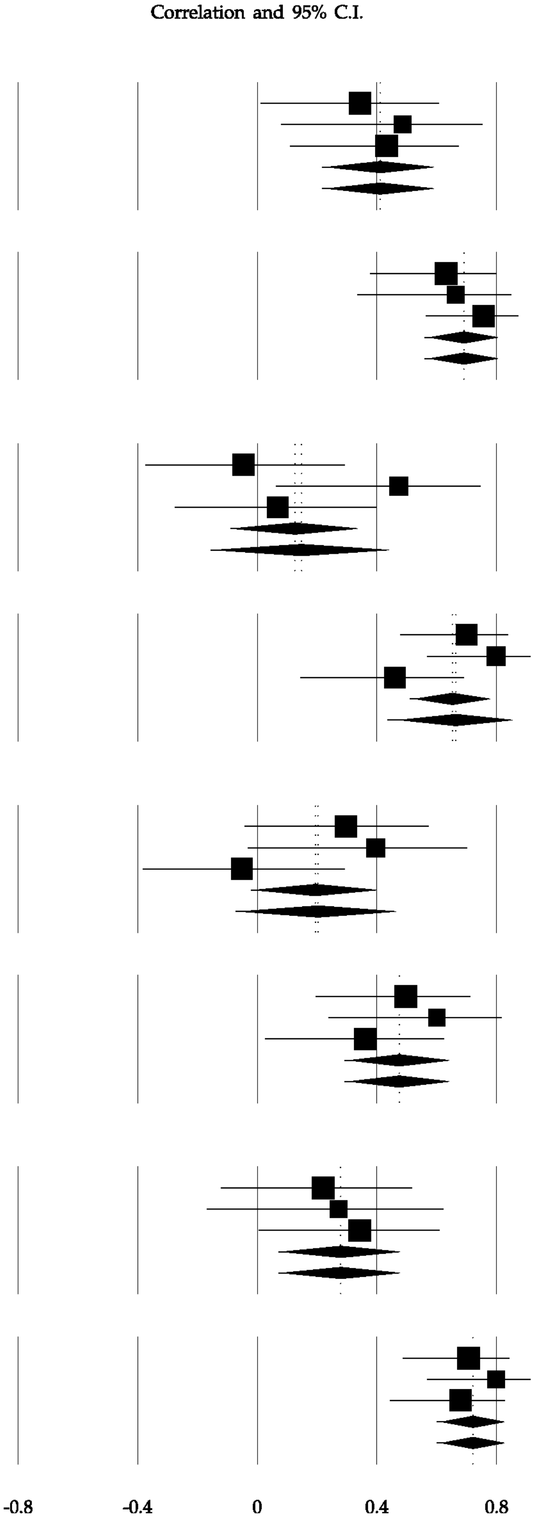


Fig. 3. Meta-analyses of correlations on software quality and job satisfaction of team climate post-test.

perceptions do affect the quality of the developed software product. This means that a work environment where all team members feel that it is safe to make suggestions and show an interest in achieving excellence at work has a positive impact on developed software quality. Therefore, all this supports the earlier results, that is, that irrespective of the team size, development process type (XP Process or Unified Process) and development task type (design and implementation or analysis and design):

- The participative safety and task orientation team climate perceptions are related to software quality.
- The support for innovation and team vision are not related to software quality.

Finally, the meta-analysis indicates that all the studies of team climate factor perceptions are homogeneous. The results are shown in Fig. 3 and Table 9, and are clear from a visual examination of the

Table 8
Significant correlations and heterogeneity relationship of team climate perceptions with software quality.

FACTORS	UAM 0405	UAM 0506	UPM 0506	META-ANALYSIS		
				CORRELATION	HETEROGENEITY INDICES	
					Q	P
Participative safety	+	+	+	**0.41***	0.36	0.82
Task orientation			+	**0.28***	0.28	0.87

*α<0.05

Table 9
Significant correlations and heterogeneity relationship of team climate perceptions with satisfaction.

FACTORS	UAM 0405	UAM 0506	UPM 0506	META-ANALYSIS		
				CORRELATION	HETEROGENEITY INDICES	
					Q	P
Participative safety	+	+	+	**0.69***	1.02	0.60
Support for innovation	+	+	+	**0.65***	4.61	0.10
Team vision	+	+	+	**0.47***	1.21	0.55
Task orientation	+	+	+	**0.72***	0.87	0.65

*α<0.05

forest plot, where the confidence intervals and result of the meta-analysis overlap. Table 9 shows that the correlations are significant (99% and 95%) for all four climate factor perceptions. Apart from these qualitative values, it also includes the significant correlation coefficients in the meta-analysis for the four climate factor perceptions. The setup of the first two studies is identical, whereas the third setup is different, as already mentioned and shown in Table 5. Consequently, all the team climate factor perceptions can be said to be related to satisfaction. We do get some near-heterogeneous (non-significant) values for the support for innovation perception factor, but the meta-analysis suggests homogeneity. Therefore, all this corroborates the finding that team climate factor perceptions are related to satisfaction, irrespective of the team size, development process type (XP Process or Unified Process) and development task type (design and implementation and analysis and design).

6. Discussion

The meta-analysis produced three main results:

- There is a rather weak relationship between *personality* and *quality* or *satisfaction*. Software quality is related to the extraversion factor but not to the neuroticism, openness to experience, agreeableness and conscientiousness factors. Satisfaction is related to the agreeableness factor but not to the neuroticism, extraversion, openness to experience and conscientiousness factors.
- No relationship was observed between the *climate preferences* and *quality* or *satisfaction*, that is, none of the team climate factor (participative safety, support for innovation, team vision and task orientation) preferences are related to either software quality or to satisfaction.
- There is a clear relationship between team *climate perceptions* and *quality* or *satisfaction*, that is, the perception of participative safety and task orientation within team climate are related to software quality, whereas the perceptions of all the team climate factors (participative safety, support for innovation, team vision and task orientation) are related to satisfaction.

We have found that all these results are independent of the applied development process type (XP, Unified Process). Regarding the connections between **personality** and **quality** or **satisfaction**, one of the primary quasi-experiments (specifically, UAM 0405) identified a statistically significant correlation between *extraversion* and *quality* on one side ($r = 0.46$, $p = 0.01$), and *agreeableness* and *satisfaction* on the other ($r = 0.33$, $p = 0.05$). This effect was not replicated in the subsequent experiments, but it is confirmed by the meta-analysis ($r = 0.25$, $p = 0.02$ – using the fixed-effects model – and $r = 0.29$, $p = 0.01$, respectively). No other effect was observed in either of the primary studies or the meta-analysis.

The observed effects make sense [8,10]. In a software development environment, extraversion is a necessary characteristic for making innovative proposals, criticizing existing designs and improving the quality of the final product irrespective of the applied development process type (XP or Unified Process). Agreeableness is also an important characteristic for team cohesion for developments using both XP and the Unified Process. It is remarkable, however, that the related correlations, albeit significant, are quite low for both development process types ($r < 0.3$ in both cases). It is also surprising that other factors, particularly *conscientiousness* (as hard-working people are expected to generate better outcomes), are not related to quality or satisfaction irrespective of the development type used by teams. Our results show that only extroversion and agreeableness are positively related to software quality and team member satisfaction, respectively.

With regard to **climate preference**, the meta-analysis is crystal clear: none of the correlations were significant. This is not surprising, as, with only one exception with respect to quality (the relationship between *team vision* and *quality* in quasi-experiment UAM 0405) and another exception with respect to satisfaction (the relationship between *support for innovation* and *satisfaction* in quasi-experiment UPM 0506), no effects at all were observed in the primary studies. This suggest that team climate preferences are not related to either software quality or satisfaction, irrespective to team size, development process type (XP Process or Unified Process) and development task type (design and implementation and analysis and design).

Finally, regarding **climate perceptions**, we found that both high team *participative safety* and *task orientation* perceptions were significantly related to better software irrespective of team size, development process type (XP Process or Unified Process) and development task type (design and implementation or analysis and design). Besides all four team climate factors perceptions were significantly related to the highest satisfaction. These results are easy to interpret: individual perceptions positively or negatively affect team outcomes, but are not affected by the type of development process used, XP or Unified Process, the size of the teams or the development task type (design and implementation or analysis and design).

This set of findings is strongly supported by information gathered from the team debriefing questionnaires in each quasi-experiment. Respondents consistently stated that a high interaction among team members is essential for developing software following both an agile methodology and an incremental iterative development process. All participants are equally engaged in the development of their project and are responsible for the success or failure of the resulting product. The project manager role is not defined, and all team members are equally committed to the project. Hence, traits like sociability, talkativeness, communicativeness, affability and responsible behavior appear to be conducive to the development of high quality software, as well as to the satisfaction of the development team members. It is important to track team climate in a development project since it seems to be one (of many) indicators of the quality of the software to be delivered. Nevertheless, even if team managers promote team climate factors (participative safety and task orientation), there is no guarantee of the team producing high quality software as we are studying correlations not cause-effect relationships. In a team where there is participative safety members feel comfortable about exchanging questions, opinions, ideas, etc., whereas a team that has task orientation is committed to achieving its objectives, excellence, and the highest possible standards and it stresses appraisals, inspections and evaluations of how work methods as a means of achieving the goals.

The team exerts a strong influence on individuals. This influence tends to cancel out team members' individual characteristics which have less of a bearing on team outcomes. Team climate is clearly related to the satisfaction level, but quality is harder to improve. The factor that was observed to be most related to quality was participative safety.

7. Conclusions

The research problem addressed in this paper is to analyze some personality and team climate factors, which social psychologists have found to influence team effectiveness, the quality of the product output by the development process and team satisfaction. Looking to extend these results, we analyze the relationships between these factors and their impact on teams to improve team formation.

On the one hand, this paper presents a correlational meta-analysis that includes the results of a correlational study of a quasi-experimental [2,3] and two replications. All these studies have two clearly separate parts. The first part analyses the relationships of development team personality factors with respect to the developed software product quality and the level of team member satisfaction. The second part focuses on the study of team climate regarding product quality and software development team members. Therefore, the goals of this research were to apply a data aggregation method to check which team member personality factor and team climate relationships were reinforced with respect to development team effectiveness. Specifically, team effectiveness is measured by the quality of the software product developed by the team, as well as team member satisfaction. On the other hand, the replications used different types of development process, XP and Unified Process, and therefore it was also possible to draw conclusions from the results of the meta-analysis about the relationship between teams and the task.

Some social psychology research point to three of the Big Five primary personality traits – neuroticism, openness to experience and conscientiousness – as being relevant for effective team operation and results [52], whereas other report two of the Big Five personality traits – extraversion and conscientiousness – as having a positive influence on team performance [8–10]. Nonetheless, in this paper, we consider all five personality factors in order to determine which factors are related to software product quality and software development team member satisfaction.

In this research we again look at the team climate factors that West and Anderson [68] considered important for team effectiveness in order to determine which factors are related to the quality of the software product and software development team member satisfaction. Specifically, the climate factors are participative safety, support for innovation, team vision and task orientation. By measuring these four factors before the project we can establish climate preferences, and by measuring them after the project, climate perceptions.

The meta-analysis aggregates the results of the quasi-experiment and its two replications in order to corroborate the following conclusions. First, we confirm the relationship between the extraversion personality factor and the quality of the resulting software product. Extraversion is beneficial for communication among team members and eases their software development activities. Additionally, we can say that the more satisfied teams have the most affable members, which improves team cohesion. Second, the study of team climate confirms that team climate preferences are not related to team effectiveness. However, the same does not apply to team climate perceptions, that is, what work environment develops in the team and its impact on team effectiveness. According to these results, the perceived team climate for all the factors can be said to determine team member satisfaction with software development irrespective of the type of development process enacted. Additionally, when team members perceive or experience a safe climate in which to put forward ideas, the quality of the software product is higher, irrespective of the development process set up to produce the final product. The same thing applies when there is a climate encouraging team members to work towards building the best product (task orientation), because they then manage to build an even better product.

The findings of this paper target any software development process, both agile and non-agile (Unified Process) methods. Taking into account some personality factors, like agreeableness or extraversion, we can improve the formation of more efficient software development teams. Likewise, controlling the team climate or environment that develops within software development teams can be an important factor for team member satisfaction. But, additionally, some aspects, such as people feeling safe about putting forward their opinions or ideas or encouraging all team members to strive to do a good job, will result in quality software.

All this knowledge can be used to define guidelines for team managers on how to form efficient teams according to team member personalities or by introducing group dynamics in order to resolve conflicts or improve group cohesion in order to build a better team climate.

Acknowledgments

This research has been funded by the following projects: Experiment Replication and Synthesis Technologies in SE (MICINN TIN2011-23216) and Go Lite (MICINN TIN2011-24139).

References

[1] S.T. Acuña, N. Juristo, Assigning people to roles in software projects, Softw.: Pract. Exp. 34 (2004) 675–696.
[2] S.T. Acuña, M. Gómez, N. Juristo, Towards understanding the relationship between team climate and software quality – a quasi-experimental study, Empirical Softw. Eng. 13 (4) (2008) 401–434.
[3] S.T. Acuña, M. Gómez, N. Juristo, How do personality, team processes and task characteristics relate to job satisfaction and software quality?, Inf Softw. Technol. 51 (2009) 627–639.
[4] N. Anderson, G.J. Burch, The Team Selection Inventory, ASE, NFER-Nelson, Slough, 2003.
[5] N. Anderson, M. West, The Team Climate Inventory, ASE, NFER-Nelson, Windsor, 1994.
[6] N. Anderson, M. West, Measuring climate for work group innovation: development and validation of the team climate inventory, J. Org. Behav. 19 (1998) 235–258.
[7] N. Anderson, M. West, The Team Climate Inventory: User's Guide, second ed., ASE, NFER-Nelson, Windsor, 1999.
[8] M.R. Barrick, M.K. Mount, The Big Five personality dimensions and job performance: a meta-analysis, Pers. Psychol. 44 (1991) 1–26.
[9] M.R. Barrick, G.L. Stewart, M.J. Neubert, M.K. Mount, Relating member ability and personality to work-team processes and team effectiveness, J. Appl. Psychol. 83 (1998) 377–391.
[10] B. Barry, G.L. Stewart, Composition, process and performance in self-managed groups: the role of personality, J. Appl. Psychol. 82 (1997) 62–78.
[11] K. Beck, Extreme Programming Explained: Embrace Change, Addison-Wesley, Reading, MA, 1999.
[12] K. Beck, W. Cunningham, A laboratory for teaching object-oriented thinking, in: Proceedings of Object-Oriented Programming Systems, Languages and Applications (OOPSLA '89), SIGPLAN Notices, vol. 24, no. 10, 1989, pp. 1–6.
[13] K. Beck, M. Beedle, A. Cockburn, W. Cunnimgham, M. Fowler, et al., Agile Manifesto, 2001. <http://agilemanifesto.org/>.
[14] K. Beck, M. Fowler, Planning Extreme Programming, Addison-Wesley, Boston, MA, 2001.
[15] B.W. Boehm, C. Abts, W.A. Brown, S. Chulani, B.K. Clark, E. Horowitz, R. Madachy, D.J. Reifer, B. Steece, Software Cost Estimation with COCOMO II, Prentice Hall, Upper Saddle River, NJ, 2000.
[16] M. Borenstein, L.V. Hedges, H. Rothstein, Meta-analysis-fixed Effects Versus Random Effects, Tech. Rep., Biostat Inc., 2007. <meta-analysis.com/downloads/Meta-analysis_fixed_effect_vs_random_effects.pdf>.
[17] G.J. Burch, N. Anderson, Measuring person-team fit: development and validation of the team selection inventory, J. Manage. Psychol. 19 (4) (2004) 406–426.
[18] G. Burch, N. Anderson, Personality as a predictor of work-related behavior and performance: recent advances and directions for future research, in: G.P. Hodgkinson, J.K. Ford (Eds.), Int'l Review of Industrial and Organizational Psychology, John Wiley & Sons, 2008, pp. 261–305.
[19] Luiz Fernando Capretz, Personality types in software engineering, Int. J. Hum Comput Stud. 58 (2) (2003) 207–214.
[20] D. Chan, Functional relations among constructs in the same content domain at different levels of analysis: a typology of composition models, J. Appl. Psychol. 83 (2) (1998) 234–246.
[21] T.D. Cook, D.T. Campbell, Quasi-Experimentation Design and Analysis Issues for the Field Settings, Houghton Mifflin, Boston, MA, 1979.
[22] P.T. Costa Jr., R.R. McCrae, NEO Personality Inventory. Revised, Psychological Assessment Resources, Odessa, Florida, 1992 (Spanish Version. 2002. TEA Ediciones, Madrid).
[23] S.S.J.O. Cruz, F.Q.B. da Silva, C.V.F. Monteiro, P.C.F. Santos, I.R.M. Dos Santos, Personality in software engineering: preliminary findings from a systematic literature review, in: Proceedings of the 15th Annual Conference on Evaluation & Assessment in Software Engineering (EASE 2011), 2011, pp. 1–10.
[24] L.A. Curral, R.H. Forrester, J.F. Dawson, M.A. West, It's what you do and the way that you do it: team task, team size, and innovation-related group, Eur. J. Work Org. Psychol. 10 (2) (2001) 187–204.
[25] T. DeMarco, T. Lister, Peopleware: Productive Projects and Teams, second ed., Dorset House, New York, 1999.
[26] T. Dybå, E. Arisholm, D.I.K. Sjøberg, J.E. Hannay, F. Shull, Are two heads better than one? On the effectiveness of pair programming, IEEE Softw. 24 (6) (2007) 10–13.
[27] T. Dybå, T. Dingsøyr, Empirical studies of agile software development: a systematic review, Inf. Softw. Technol. 50 (2008) 833–859.
[28] A. English, R.L. Griffith, L.A. Steelman, Team performance: the effect of team conscientiousness and task type, Small Group Res. 35 (6) (2004) 643–665.
[29] S. Faraj, L. Sproull, Coordinating expertise in software development teams, Manage. Sci. 46 (12) (2000) 1554–1568.
[30] R. Feldt, R. Torkar, L. Angelis, M. Samuelsson, Towards individualized software engineering: empirical studies should collect psychometrics, in: Proceedings of the International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE '08), ACM, New York, 2008, pp. 49–52.
[31] R. Feldt et al., Links between the personalities, views and attitudes of software engineers, Inf. Softw. Technol. 52 (6) (2010) 611–624.
[32] D.L. Gladstein, Groups in context: a model of task group effectiveness, Adm. Sci. Q. 29 (1984) 499–517.
[33] P.S. Goparaju, F. Ayesha, P. Sanghamitra, Soft factors affecting the performance of software development teams, Team Perform. Manage. 17 (3–4) (2011) 187–205.
[34] J.R. Hackman, G.R. Oldham, Work Redesign, Addison-Wesley, Reading, MA, 1987.
[35] J.E. Hannay, T. Dybå, E. Arisholm, D.I.K. Sjøberg, The effectiveness of pair programming: a meta-analysis, Inf. Softw. Technol. 51 (2009) 1110–1122.
[36] L.V. Hedges, J.L. Vevea, Fixed- and random-effects models in meta-analysis, Psychol. Methods 3 (4) (1998) 486–504.
[37] J.P. Higgins, S.G. Thompson, J.J. Deeks, D.G. Altman, Measuring inconsistency in meta-analyses, Br. Med. J. 327 (4) (2003) 557–560.
[38] M. Höst, B. Regnell, C. Wohlin, Using students as subjects – a comparative study of students and professionals in lead-time impact assessment, Empirical Softw. Eng. 5 (3) (2000) 201–214.
[39] J.E. Hunter, F.L. Schmidt, Fixed effects vs. random effects meta-analysis models: implications for cumulative research knowledge, Int. J. Sel. Assess. 327 (8) (2000) 272–292.
[40] IEEE, Guide to the Software Engineering Body of Knowledge-SWEBOK Version 2004, IEEE Computer Society, California, 2004.
[41] I. Jacobson, G. Booch, J. Rumbaugh, The Unified Software Process Development, Addison Wesley, Reading, MA, 1999.
[42] S.L. Jarvenpaa, D.E. Leidner, Communication and trust in global virtual teams, Organ. Sci. 10 (6) (1999) 791–815.
[43] A. Jedlitschka, M. Ciolkowski, D. Pfahl, Reporting experiments in software engineering, in: F. Shull, J. Singer, D.I.K. Sjøberg (Eds.), Guide to Advanced Empirical Software Engineering, Springer, London, 2008, pp. 201–228.
[44] K.A. Jehn, A qualitative analysis of conflict types and dimensions in organizational groups, Adm. Sci. Q. 42 (1997) 530–557.
[45] M.I. Kellner, R.J. Madachy, D.M. Raffo, Software process simulation modelling: Why? What? How?, J Syst. Softw. 46 (1999) 91–105.
[46] D.A. Kenny, L. La Voie, Separating individual and group effects, J. Pers. Soc. Psychol. 48 (2) (1985) 339–348.
[47] M.W. Lipsey, D.B. Wilson, Practical Meta-Analysis, Sage, Thousand Oaks, CA, 2001.
[48] F. Macias, M. Holcombe, M. Gheorghe, A formal experiment comparing extreme programming with traditional software construction, in: Proceedings of the Fourth Mexican International Conference on Computer Science (ENC 2003), 2003, pp. 73–80.
[49] M.A. Mañas, V. González-Romá, J.M. Peiro, Work-Group Climate. Determinants and Consequences, Universidad de Almería, Instituto de Estudios Almerienses, Almeria, Spain, 1999.
[50] J. Miller, Applying meta-analytical procedures to software engineering experiments, J. Syst. Softw. 54 (1–2) (2000) 29–39.
[51] J.E. McGrath, Groups: Interaction and Performance, Prentice-Hall, Englewood Cliffs, New Jersey, 1984.
[52] E. Molleman, A. Nauta, K.E. Jehn, Matching tasks and workers in teams: a multi-level approach, in: Proceedings of the 6th International Workshop on Teamworking, School of Technology and Society. Malmö University, Malmö, 2002, pp. 288–310.
[53] E. Molleman, A. Nauta, K.A. Jehn, Person-job fit applied to teamwork: a multilevel approach, Small Group Res. 35 (2004) 515–539.
[54] E. Moore, Personality characteristics of information systems professionals, in: Proceedings of the Conference on SIGCPR, 1991, pp. 140–155.
[55] I.B. Myers, H.M. McCaulley, N.L. Quenk, A.L. Hammer, MBTI Manual (A Guide to the Development and Use of the Myers Briggs Type Indicator), third ed., Consulting Psychologists Press, 1998.
[56] A. Peslak, The impact of personality on information technology team projects, in: Proceedings of the SIGMIS-CPR'06, Claremont, California, USA, 2006, pp. 273–279.
[57] D. Pfahl, O. Laitenberger, G. Ruhe, J. Dorsch, T. Krivobokova, Evaluating the learning effectiveness of using simulations in software project management education: results from a twice replicated experiment, Inf. Softw. Technol. 46 (2004) 127–147.
[58] N. Salleh, E. Mendes, J.C. Grundy, G.St.J. Burch, An empirical study of the effects of conscientiousness in pair programming using the five-factor personality

model, in: Proceedings of the ACM/IEEE International Conference on Software Engineering, vol. 1, 2010, pp. 577–586.
[59] N. Salleh, E. Mendes, J.C. Grundy, G.St.J. Burch, The effects of neuroticism on pair programming: an empirical study in the higher education context, in: Proceedings of the ACM-IEEE International Symposium on Empirical Software Engineering and Measurement, 2010.
[60] N. Salleh, E. Mendes, J. Grundy, The effects of openness to experience on pair programming in a higher education context, in: Proceedings of the 24th International Conference on Software Engineering Education and Training (CSEE&T 2011), IEEE Computer Society, 2011, pp. 149–158.
[61] P. Sfetsos, I. Stamelos, L. Angelis, I. Deligiannis, An experimental investigation of personality types impact on pair effectiveness in pair programming, Empirical Softw. Eng. 14 (2009) 187–226.
[62] T. Seger, O. Hazzan, R. Bar-Nahor, How does Readiness for Agile Development Relate to Team Climate and Individual Personality Attributes?, Springer-Verlag, Berlin, Heidelberg, 2007.
[63] H.J. Thamhain, D.L. Wilemon, Building high performing engineering project teams, IEEE Trans. Eng. Manage. 34 (3) (1987) 130–137.
[64] R. Turley, J. Bieman, Competencies of exceptional and nonexceptional software engineers, J. Syst. Softw. 28 (1) (1995) 19–38.
[65] T. Walle, J.E. Hannay, Personality and the nature of collaboration in pair programming, in: Proceedings of the 3rd. International Symposium on Empirical Software Engineering and Measurement (ESEM 2013), IEEE Computer Society, 2009, pp. 203–213.
[66] A.E.M. van Vianen, C.K.W. De Dreu, Personality in team: its relationship to social cohesion, task cohesion, and team performance, Eur. J. Work Org. Psychol. 10 (2) (2001) 97–120.
[67] M.A. West, The social psychology of innovation in groups, in: M.A. West, J.L. Farr (Eds.), Innov. Innovation and Creativity at Work: Psychological and Organizational Strategies, Wiley, Chichester, UK, 1990.
[68] M.A. West, N. Anderson, Innovation in top management teams, J. Appl. Psychol. 81 (1996) 680–693.
[69] J. Wynekoop, D. Walz, Investigating traits of top performing software developers, Inf. Technol. People 13 (3) (2000) 186–195.
[70] A. Zander, Making Groups Effective, second ed., Jossey-Bass, San Francisco, CA, 1994.