# An examination of personality traits and how they impact on software development teams

Murat Yilmaz[a], Rory V. O'Connor[b,c], Ricardo Colomo-Palacios[d], Paul Clarke[b,c]

*[a]Department of Computer Engineering, Çankaya University, Turkey*
*Phone: +90 312 233 13 58, Fax: +90 312 233 10 26, e-mail: myilmaz@cankaya.edu.tr*
*[b]Lero, the Irish Software Research Center, Ireland*
*[c]School of Computing, Dublin City University, Ireland*
*[d]Østfold University College, Faculty of Computer Sciences, Norway*

## Abstract

**Context:** Research has shown that a significant number of software projects fail due to social issues such as team or personality conflicts. However, only a limited number of empirical studies have been undertaken to understand the impact of individuals' personalities on software team configurations. These studies suffer from an important limitation as they lack a systematic and rigorous method to relate personality traits of software practitioners and software team structures.

**Objective:** Based on an interactive personality profiling approach, the goal of this study is to reveal the personality traits of software practitioners with an aim to explore effective software team structures.

**Method:** To explore the importance of individuals' personalities on software teams, we employed a two-step empirical approach. Firstly, to assess the personality traits of software practitioners, we developed a context-specific survey instrument, which was conducted on 216 participants from a middle-sized software company. Secondly, we propose a novel team personality illustration method to visualize team structures.

**Results:** Study results indicated that effective team structures support teams with higher emotional stability, agreeableness, extroversion, and conscientiousness personality traits.

**Conclusion:** Furthermore, empirical results of the current study show that extroversion trait was more predominant than previously suggested in the literature, which was especially more observable among agile software development teams.

*Keywords:*
Software Team Visualization, Interactive Personality Assessment, Software Developer Personality Traits, Social Aspects of Software Development

## 1. Introduction

People are key for software development processes [1] and personnel factors are one of the elements that can impact the effectiveness and productivity of software teams [2]. These teams rely on the communication skills and abilities of a software development team, and therefore the productivity of a software organization is affected by the levels of social cohesion among its members [3]. In spite of its importance, much of the research and practice has focused mainly on technological or process-related factors instead of organizational, social or psychological factors [4]. However, the literature also reports several studies devoted to investigate the personality types of software practitioners and their connection with project results. A recent paper by Cruz et al. [5] illustrates the relative importance of personality studies for the discipline and the prevalent role of personality characteristics in these studies. MBTI (Myers-Briggs Type Indicator) and FFM (the five-factor model) [6] are the most commonly used approaches for assessing personality profiles particularly in software engineering context, although the second is currently gaining popularity among personality psychologists [5].

Personality traits are considered as stable characteristics of individuals that can be used for selection, measured in percentages, and expect to be normally distributed in a population. On the other hand, personality types are constructs that denote independent groups. They might heavily depend on culture and time, which are frequently illustrated in a binary format and accepted as controversial [7]. To manage higher levels of interaction among software practitioners, it is important to note that individuals may have different personality traits. We believe that awareness of such a difference can greatly improve team motivation. To address the issues mentioned above, we suggest an interactive (game-like) questionnaire in which the personality traits of an individual can be retrieved by using questions based on the situations derived from the software domain instead of using a conventional (i.e. non-interactive and static) questionnaire. The questions are based on a set of situations, which requires a decision, and shall be administered using face-to-face interactions. After asking each question, the responses shall be recorded to investigate the personality traits of the practitioners. Such an innovative approach is crucial for achieving more accurate personality measurements. Similar to a game, our proposition has the potential to keep people individually motivated with the idea of social success [8], and the results have the potential to improve their ability to understand each other's thoughts and feelings [9]. Most importantly, unlike classical approaches, such a situation-based questionnaire approach also removes the concerns related to asking software development personnel a set of private questions.

Frequently used within the domains of psychology and education, an interactive assessment [10] is a type of assessment where the examiner interacts with a subject or a group of individuals in a richer form than simply asking questions or giving paper-based instructions. In addition, the assessor provides the examinees with some substantial *game-like* assistance within the assessment during the onboarding process, and particularly for creating an appropriate scaffolding[1] mechanism [11]. The notion of adopting game related concepts can deliver several benefits. Inherently, it could offer solutions for society based problems using game-like characteristics [8]. Therefore, it is not surprising to discover that interactive approaches have great social and economic benefits [8]. From the outset, such approaches enable us to form collective social organizations, which ultimately produce considerable advantages for operating complex concerns. On the other hand, to constitute better software development teams, we should benefit from personality assessments, which may also particularly be useful for improving team productivity in the software development process.

In addition to the all arguments above, Kaluzniacky's words eloquently support this: "*Eventually, perhaps an actual IT personality diagnostic instrument could be developed containing only questions with an IT work context, but paralleling closely the questions in [a personality assessment] itself.*" [12, pp. 55]. This study aimed to investigate the following hypothesis: *Personality characteristics of individuals in a software development organization can be revealed*

[1] It is a common characteristic of all kind of games, which motivates participant to engage with the activity for a substantial outcome.

*and illustrated by using a context specific interactive assessment technique.*

The remainder of the paper is organized in the following manner. Section two provides an overview of the personality types and traits. It reviews the literature for its application in software engineering research. The third section details the methodological approach we used to conduct the case study, and details the interactive approach. The next section continues with an industrial implementation, which offers illustrations of practitioners' traits and overall team characteristics. Section five provides a discussion of the findings, major conclusions of the study, limitations and suggestions for future work.

## 2. Research Background

A general assumption is that personality characteristics significantly affect the human behavior [13]. Although most people agree that types of behaviors vary in different situations, for most individuals there is an observable pattern of consistency in their behaviors. The classical approach to organizing individuals according to their personality characteristics requires performing a personality questionnaire. Several psychometric tests are used to capture non-context dependent behavior. For example, they are used to assign personnel to the right job, by exploring soft skills. Some of these tests are primary mental ability tests, wonderlic personnel tests and programmer aptitude tests [14]. However, observing individual's verbal decisions in several different context-dependent situations may also reveal the personality characteristics of an individual. Based on Jungs theories [15], MBTI is a psychometric instrument defined by Myers et al. [16] to analyze personality types. The most used test consists of around one hundred forced-choice items regarding four dimensions: Extroversion/Introversion (EI), Sensing/Intuition (SN), Thinking/Feeling (TF) and Judging/Perceiving (JP). Sixteen distinct personality types are defined on the basis of combining these preferences into boxes labelled using four-letter codes.

The Big Five Inventory (BFI) is a framework which considers five factors (i.e. Openness, Conscientiousness, Extroversion, Agreeableness, Neuroticism) that are essential to classifying individual differences in terms of personality characteristics [17]. The model suggests viewing personality with respect to five comprehensive dimensions (i.e. traits), which is one of the widely reorganized models by personality researchers (see Table 1). Five dimensions can be briefly identified as follows: Openness is a personality characteristic which represents individuals who enjoy new experiences and adventures. These people are found to be curious, imaginative, and creative. Conscientiousness individuals are well-organized, purposeful, serious in dealing with problems with a sense of duty. They are achievement oriented. Extroversion is a personality trait that represents individuals' social energy levels who are frequently found to be talkative, active and assertive. Agreeableness is a personality trait that measures an individual's kindness, and they are likely to be gentle, cooperative, helpful and trustful. Individuals with high neuroticism are more concerned about emotional experiences such as sadness, anxiety, depression, and they may be temperamental with self-pitying, and ultimately they may show emotional behaviours [18].

### *2.1. Personality Research in Software Engineering*

A considerable amount of literature has been published on personality research in software engineering [5]. A number of instruments were designed to assess personality characteristics of software practitioners such as BFI [17], MBTI [16], and the Keirsey temperament sorter [19]. MBTI is a tool aimed at understanding personality characteristics while BFI is an instrument for measuring personality traits, which has been found to be beneficial for personnel selection.

| Dimension | High Scorers | Low Scorers |
|---|---|---|
| Openness | Open to adventures, artistic, creative | Practical, traditional, rational |
| Conscientiousness | Well-organized, self directed | Unplanned, without sufficient attention |
| Extroversion | Outgoing, enthusiast, socially active | Enjoy being alone, independent, mostly quite |
| Agreeableness | Gentle, empathetic, helpful | Lack of empathy, complicated |
| Neuroticism | Negative emotional state, sadness, anxiety | Emotionally more stable, risk taker |

**Table 1:** *Potential Outcomes for the Personality Traits of Big Five*

Ultimately, BFI is preferred as a tool to assess five personality traits of individuals with its clear construct structure and reliability [20].

Thus far, a number of studies have attempted to measure the personality characteristics of software practitioners using the BFI framework. By using the International Personality Item Pool (IPIP) [21] 50-item five-factor personality test, Feldt el at. [22] investigated the personality connections between viewpoints and behaviors of 47 software practitioners. They concluded that software engineers are a homogeneous group with respect to its personality structure and suggested that personality views of software practitioners should be used by managers to assess software project failures. To investigate the success in pair programming tasks, Chao and Atli [23] assessed four personality traits in pair programming. However, they failed to find a statistically significant relationship between quality of code and their selection of personality traits to explore compatibility of partners. Hannay et al. [24] examined the personality traits of the five factor model on pair programmers and their efficiency in a study comprising 196 software practitioners. They confirmed that more investigations should be made to determine the relationship between personality and performance related factors for pair programmers. Rodrigues and Rebelo [25] suggested that software practitioners with a proactive personality might be likely to affect individuals' job performance while there was no connection observed between big five personality traits and performance prediction. Based on two big five personality factors, agreeableness and conscientiousness, Acuna et al. [26] demonstrated that there is a significant connection between these personality traits and team job satisfaction. Additionally, they found that autonomy improves their satisfaction levels while a relationship has been identified between extroversion and software product quality. According to Kosti et al. [27], although some research has been carried out on the potential factors (e.g. intelligence and self-compassion) affecting personality characteristics and its impact on software practitioner performance and viewpoints, there have been few empirical investigations and therefore there is little published data on such issues. Their analysis suggested that software practitioners should be considered in two main clusters: intense personalities and non-intense ones. To sum up, several earlier research efforts confirmed that psychometric assessments and related tools are useful to analyze and predict software practitioners' preferences during software development.

Numerous studies have attempted to explain personality characteristics of software practitioners by using MBTI and the Keirsey temperament sorter. Rutherford [28] conducted a study to build project teams for software engineering classes. He claimed that a team with a variety of personalities had more skills in problem solving. In addition, in their ethnographic study, Karn and Cowling [29] conducted a performance analysis of student teams by comparing team effectiveness on a yearly basis. Their findings suggest that building heterogeneous teams in terms of individual personalities brought different ideas to teamwork, which improved the team's productivity. In addition, Karn et al. [30] conducted a qualitative analysis to investigate how the dynamics of software teams can be related to personality research, particularly for XP projects. The results of this analysis indicated that (i) personality type configurations were important for team effectiveness, and (ii) teams with high cohesion were found more competitive.

Capretz [31] surveyed software engineering students by using a personality assessment scale, and concluded that a variety in personality characteristics should form better teams to improve the quality of products. Sach et al. [32] analyzed five different studies conducted to investigate personality preferences in the software engineering domain where they found that these studies supported each other. A detailed examination of the relationship between personality characteristics and variance of performance in code was undertaken by Da Cunha and Greathead [33], who proposed that a productivity difference among the individuals might emerge when teams were organized according to their personality attributes. In a study in 2004, Gorla and Lam [34] reported on the personality types of 92 practitioners that were structured in small teams using Keirsey's temperament sorter. Their goal was to find a possible connection between personality types of teams and their performance. Not surprisingly perhaps, it was found that extroverted practitioners communicate better than an introverted individuals, therefore extroversion was a preferable type of personality particularly for tasks that require more social interactions.

Taken together, an implication of previous research on personality demonstrates that it is important to reveal the personality characteristics in a software team [35]. Such a team with a variety of different skilled people with distinctive perspectives should be beneficial when coping with the dynamic tasks of a software development process [36]. For instance, some of the personality characteristics such as introversion vs. extroversion may have more impact on one development phase, and can be used for selecting individuals for the different stages of a software development process with respect to their so-called *soft skills* [37]. As a consequence of the growing complexity of software development activities, there is no single personality characteristic found that fulfills all roles in software development. Moreover, an ethnographic study targeting the actualization of MBTI measures on software teams by Karn and Cowling [38] reached a conclusion that specific personality types are more gravitated towards some specific type of development activities, as well as some specific roles. A recent empirical study suggests that there is a pattern that connects personality characteristics to role choices in software development activities where assigning tasks to individuals regarding personality types improves the chance of success [39].

There are two main reasons why a novel approach is essential. Firstly, from an industrial perspective, personality assessment is considered as a tool in the service of exploring the social characteristics of software practitioners during the activities of software development [12]; all the studies reviewed so far, however, suffer from the fact that there is no systematic and rigorous method to relate the personality characteristics of software practitioners to the formation of a software team structure. Kaluzniacky [12] agrees with this, pointing out that a tool for assessing the personality characteristics of IT practitioners should be constructed. Secondly, most of the research conducted in literature is qualitative, which means that the research to date has tended to focus on individuals, for example, who can either be extroverted or introverted without any quantification such as percentage of extroversion-introversion level. Most importantly, evidence suggests that it is hard to assign an individual a personality type especially those found to be near to the cut-off values, and further there is no revised or updated version of such personality assessments based on these issues above [40]. Nevertheless, "*[personality assessments] are administered over two million times each year and is used by eighty-nine companies in the Fortune 100 and by the U.S. military. Thousands of research studies, journal articles, and dissertations have focused on it. It is used in couples counseling, family therapy, team building, and career counseling*" [41, pp. 37].

In light of these remarks, this study aims to measure the personality traits of software practitioners based on the Big Five model on a quantitative scale, which can be useful, particularly for illustrating the personality characteristics on a software team level.

### *2.2. An interactive Approach for Personality Profiling*

Psychological testing is considered a research field that observes individuals' behavior to investigate hypothesized constructs such as intelligence, personal attitudes or personality traits. Consequently, it can be based on testing a relationship between an individual and an object (e.g. cards, images, etc.). A famous technique for psychological testing is the inkblot (Rorschach) test in which subjects are required to offer a response to a set of inkblots. In addition, different empirical tests are performed to investigate the social and emotional characteristics of individuals. These tests, for example, may analyze whether the answers of a participant are defensive or argumentative.

Depending upon the complexity of tasks and human interactions, in software engineering settings, a precise computational model for profiling software practitioners is challenging to construct. In light of this remark, we used an interactive method for personality profiling akin to psychiatric tests. Similar to the psychiatric assessments (i.e. card-based tests) used in the field of psychology, in our approach participants answer a set of questions. We have termed this novel approach *Qualitative Simulation, a scenario based information gathering, analyzing and evaluation method which relies on a set of interactive game elements (e.g. a deck of cards, turn-based play, feedback) for better results.*

This approach was partly based on concrete ideas from the relational frame theory [42], which is considered a rigorous approach to understanding the empirical and conceptual connections between verbal knowledge and behavioral process. Therefore, questions on the cards utilized triggering keywords and an attached situation. Ultimately, the goal was to utilize the connections between human language and derived stimulus relations. Based on real life and context dependent situations, our approach was likely to use operationalized scenarios and events that can happen during the software development life cycle.

In addition, the approach uses *context cards* that are equipped with personality trait questions, which were derived from several situations captured from the events observed in the software industry [43]. Grounded on software development concepts, these context cards are used to operationalize a situational approach where the goal is to reveal the personality traits of software practitioners using real-life situations (e.g. see [44]). Here we suggest that such a systematic approach not only reveals the personality profile of an individual but also understands the personality profiles of software teams as a whole.

### *2.3. An Iterative Qualitative Approach for Card Creation Process*

To bridge the gap between obtained empirical data and a theory at a conceptual level, grounded theory was introduced by Glaser and Strauss [45]. It is a systematic approach for collecting and analyzing qualitative data consistent with the theory of *symbolic interactionism*, i.e. an individual's definition of a situation that causes an action [46]. It also promotes the idea of continuous engagement with the empirical findings while segmenting the key thoughts among the conceptual elements by possible operationalization of coding, comparing and memoing of the raw data. In fact, grounded theory has been used successfully in software engineering both for theory development [see e.g., 47, 48], and in a reduced form for the purpose of robust concept identification [49].

Partially based on the grounded theory methodology, we built the card creation process as an iterative qualitative approach for a number of reasons: (i) to our knowledge, no study exists in the literature that offers an inductive approach and therefore allows for the emergence of a psychometric structure based on the experiences (or situations) captured from the individuals in a software development organization, (ii) the grounded theory analysis is a well-established research methodology which has sophisticated guidelines for conducting effective empirical research in fields such as software engineering, partially based on the exchange of tacit knowledge which requires social interaction, (iii) based on the industrial experience of the researchers, grounded theory promotes the notion of *theoretical sensitivity*, which relates to the ability of researchers to understand the important elements of the data and its empirical contributions to the card creation process.

## 3. Methodology

### *3.1. Case Study Research*

A case study is a multi-dimensional activity that is frequently used for seeking answers to scientific inquiries. It is a classical approach usually when one or more instances of a phenomenon is investigated in its natural environment at a bounded time. While the quality of the results and the combination of its components vary, case study found to be an appropriate methodology for conducting empirical research in software engineering landscapes [50]. Yin [51] suggests that a case study should be *an empirical inquiry*, which examines a state-of-the-art phenomenon in the real-life situations based on multiple information resources with unclear boundaries in the given context.

### *3.2. Research Process*

To evaluate the initial prototype of context cards that was proposed in a pilot study [43], we conducted a series of sequential activities explained in the preliminary findings [52]. The prototype evaluation was started with a set of interviews. Secondly, we consulted with three experts: an English-language expert who checked the questions for language usage and understandability, a software development manager who examined the questions for their relevance with software development activities, an organizational psychologist who criticized the tone of the questions. We also asked their general opinion regarding the content of the questions in terms of clarity, readability and relevance. Based on this initial feedback, 12 questions were revised [52]. Next, 15 participants were tested with the Big Five Inventory [53] and results were compared using correlation analysis [52].

In this section, we detail the research process we conducted in the case study. The details of the research process (see Figure 1) comprises the following steps:

1. First, we form our hypothesis: the personality characteristics of individuals in software development teams can be revealed and illustrated by using a context specific interactive assessment technique.

2. Secondly, we review the literature for the use of personality assessments in the software engineering domain.

3. Thirdly, we construct a survey instrument; (i) we identify the themes, which outlines the main body of the questionnaire, (ii) we create a deck of 60 cards, (iii) we discuss the results with a group of experts to validate the content of the questions.

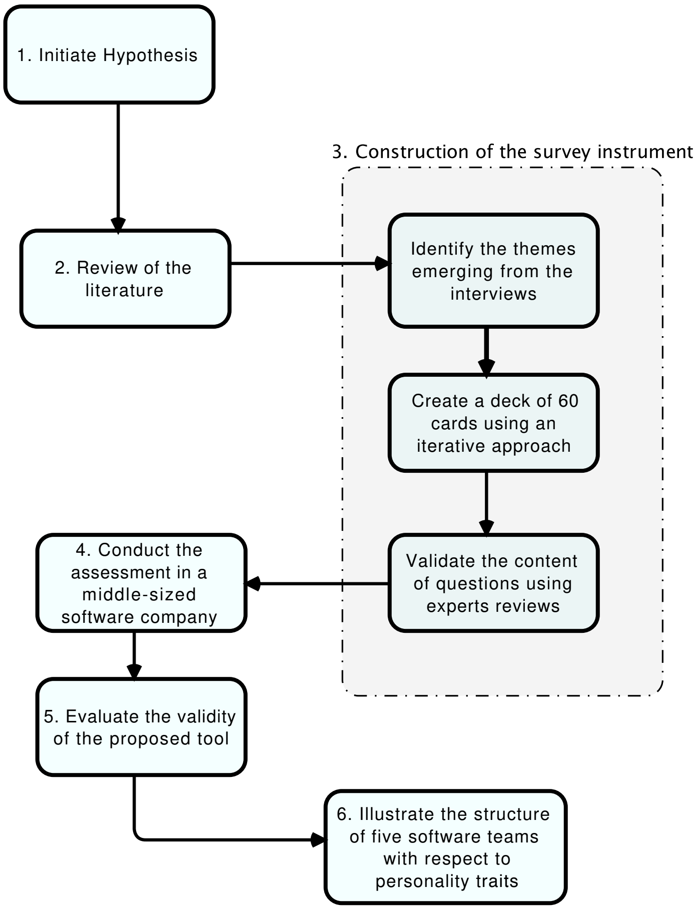

1. Initiate Hypothesis
2. Review of the literature
3. Construction of the survey instrument
Identify the themes emerging from the interviews
Create a deck of 60 cards using an iterative approach
Validate the content of questions using experts reviews
4. Conduct the assessment in a middle-sized software company
5. Evaluate the validity of the proposed tool
6. Illustrate the structure of five software teams with respect to personality traits

**Figure 1:** *The Steps Involved in Our Systematic Research Process*

4. Fourth, using these cards, we conduct the personality assessment with 216 software practitioners.

5. Fifth, we evaluate the validity of the proposed tool.

6. Finally, in the sixth step, based on our findings, we illustrate five software teams with respect to the personality traits of its members, and investigate how they are structured using the Big Five personality traits.

### *3.2.1. Assessment Procedure*

This subsection describes the necessary requirements that are proposed, and procedural steps, which were taken, to conduct the interactive assessment. The defined procedure inherits a set of game design patterns. It was conducted in a sequential order where the responses were recorded on a sheet of paper.

- To conduct the assessment, we decided that a small meeting room isolated from the work environment would be suitable.
- To asses team based characteristics properly, practitioners' were invited as a team to promote social courage for more reliable responses.
- A form to record the answers to questions was supplied.
- The assessment started with an introductory talk, and later a question from a deck of cards was read to each team by a moderator.
- The order of cards remains preserved during the assessment where the questions in these cards correspond to one of the Big Five personality dimensions.
- Participants were encouraged to interact and discuss the details of any questions.
- After a question was asked, participants were required to select from two alternative answers and saved their responses according to the form supplied.
- The moderator waited for each team member to finish answering the question.
- To acquire scaffolding for the interactive assessment, each question was built on a theme and a picture related to that theme was shown on each round.
- There was a fixed sequence for all respondents to go through all the cards. The process continues until 60 cards were discussed.
- Lastly, social discussions with teammates were encouraged between questions regarding the situations introduced.

To sum up, unlike traditional type assessments, we constructed an interactive team-driven approach which was based on game design concepts as well as face to face interactions. During these interactions, participants' responses were captured. These recordings were analyzed and results interpreted. At this point, situational context cards were employed to observe the participants' verbal behaviors so as to discover not only the personality characteristics of individuals but also the personality structure of the software teams as a whole.

## 4. Industrial Case Study

In order to evaluate our personality-profiling framework, an industrial case study was carried out within a middle-sized software development organization. Founded over 15 years ago, the selected firm is a well-established company in delivering software products. This company was primarily chosen due to its enthusiasm to participate in this study. To improve their organizational productivity, the management team was interested in receiving more information about software team structures. Secondly, although the sample size of a case study is likely to depend on available time and resources, a middle-sized company with more than two-hundred employees is adequate enough to collect data for the research [54]. The company has a number of team-based development activities, and use both agile and traditional type of methods where the team size varies from four and forty. From an industrial point of view, similar to an interactive game-like approach, an interactive BFI-44 [20] based personality questionnaire was created, conducted, and evaluated on-site with 216 software practitioners. As the arrangements were kindly requested by the management, all individuals participated.

The basic idea in developing the new instrument is to measure the personality traits more precisely using a percentage scale. In particular, the instrument guides us to highlight the difference between the same personality characteristics with different ratios. For example, it takes the variation into account that helps us to explore team-based personality traits as a whole. In addition, a card-based game-like approach is used to improve the quality of the assessment.

### *4.1. Crafting the Instrument and Protocols*

This part of the study is comprised of sequential steps, which requires a significant amount of time on the part of the participants.

Furthermore, we intentionally initiated the card creation process by investigating Big Five Factors (dimensions) of personality for two reasons. First, it is a version of the BFI-44 test that is freely available[2]. We thought using a common template would be helpful for evaluators to work on our cards. Secondly, we preferred to be compatible with the Big Five inventory for improving the outputs of this study.

Our approach comprises three main steps: (i) initiation phase, (ii) card creation phase, (iii) comparison phase (see Figure 2).

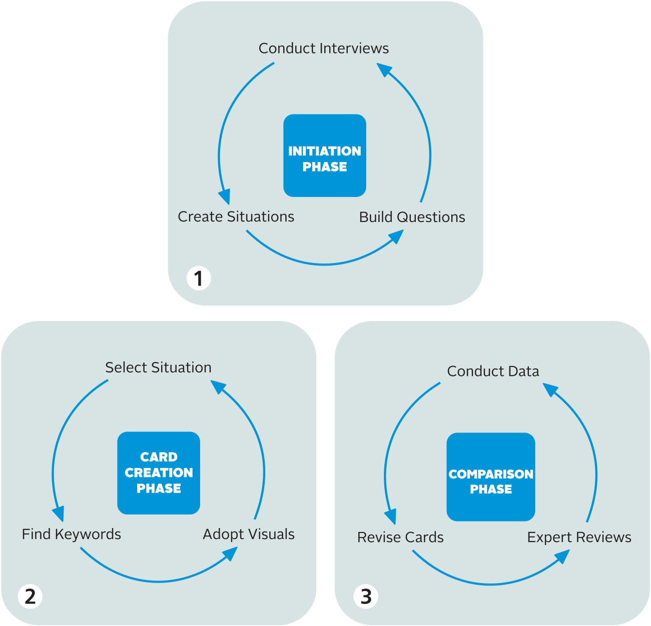


**Figure 2:** *The Systematic Process for Creating Context Cards*

[2] Please see https://www.ocf.berkeley.edu/ johnlab/bfi.php

#### *4.1.1. Initiation Phase*

The initiation phase started with searching for a set of context dependent situations from the industrial settings that were transformable into hypothetical situations, which were later used in the card design process. To improve the findings through peer confirmation, we reviewed the questions of BFI-44 (i.e. a 44-item personality assessment inventory for measuring the big five dimensions of personality [55]) for potential themes, which our context dependent questions should be based upon. To store the codings transcribed from the interviews, a codebook was also created.

By following Creswell's [56] advice, we selected 20 participants - who were highly experienced in software development - for this part of the study (10 were selected for semi-structured interviews, 10 were selected for the expert reviews). From among the initial cohort of 60 software practitioners who were willing to participate in the study, we selected 10 individuals for the interviews based on their availability, work experience and age. All participants were above 30 years of age and they had at least five years experience in the software development domain. The interviewees were informed about personality research in software engineering and requested to submit a BFI-44 test by email. Next, they received a consent letter and a set of interview questions such as "*What do you think is missing in a personality test like this for identifying software engineers personality characteristics?*", "*Can you think of any domain specific situation that can replace a question from the questionnaire?*", etc. Later, follow-up discussions about the personality characteristics were recorded for transcription. During the process, the transcriptions were segmented based on several coded parameters such as similarities and contradictions in speech, and then they were processed before starting the analysis.

Based on the preliminary outcomes, 5 of 10 interviewees were selected for half an hour one-to-one interviews. For the next iteration, we selected 3 participants who were from the research and development department of the company for a one hour extra discussion. Finally, we discussed the findings of the previous iterations with a usability expert, experienced software practitioner and an expert in organizational psychology.

Furthermore, additional data concerning a number of business situations were collected by one of the researchers who participated in a number of meetings.

| Participant ID | Title | Age | Years of Experience | Education |
|---|---|---|---|---|
| P12 | IT Specialist | 33 | 6 | MSc. |
| P36 | Product Owner | 47 | 7 | PhD. |
| P44 | Software Architect | 37 | 12 | BSc. |
| P57 | Software Developer | 31 | 6 | BSc. |
| P99 | Software Developer | 33 | 7 | BSc. |
| P106 | R&D Team Lead | 39 | 14 | PhD.. |
| P112 | Software Tester | 32 | 4 | MA. |
| P73 | System Analysis | 34 | 9 | BA. |
| P51 | R&D Team Member | 32 | 7 | MSc. |
| P97 | R&D Team Member | 31 | 5 | MSc. |

**Table 2:** *Participants' Information*

Table 2 outlines the profile of the 10 participants including their roles (titles), age, years of experience and level of education. The first iteration was conducted on ten people. The interviews were analyzed from audio recordings by using a tool for scientific transcription called *f5*. Additionally, we used the TAMS analyzer as a Computer Assisted Qualitative Data AnalysiS

(CAQDAS) tool while processing our data.

### *4.1.2. Validation of the Codebook*

To aid in coding the segments of information, we created a codebook. The goal was to create relationship between themes and potential keywords, which might be beneficial for the card creation process. The data segment, which had a potential keyword, was given a new code and added to this codebook. When the transcription was completed, the researcher reviewed each independent information segment for the themes of interests where these segments were checked for potentially new concepts. Whenever researchers found a potential item stored in a transcribed segment, they compared it with the initially coded themes and personality constructs.

To validate our codebook, the coding scheme was discussed with the research and development team of the middle-sized software company. Initial results allowed us to derive the initial personality type keywords by open coding, and further we enhanced our results by using several documents and reports where the themes were for Extroversion; (i) talkative, (ii) assertive, (iii) energetic, (iv) active, (v) approachable, (vi) outgoing. For Openness to experience; (i) curiosity, (ii) innovation, (iii) flexibility, (iv) immersion, (v) open-minded (vi) expressiveness. For Agreeableness; (i) agreeable, (ii) directness, (iii) altruistic, (iv) optimistic, (v) tolerant (vi) idealistic. For Neuroticism; (i) tense, (ii) haste, (iii) uncertainty, (iv) tension, (v) temperamental, (vi) insecure. For Conscientiousness; (i) unselfishness, (ii) compassion, (iii) obedient, (iv) persistent, (v) indulgence, (vi) Industrious. In addition, the TAMS analyzer allowed us to tag a set of potentially suitable keywords for the next phase. To select the keywords for our deck of 60 cards, we analyzed the frequency of similar keywords with respect to different interviewees. These keywords were discussed with several rounds of feedback and revisited by a user experience designer who also guided the picture selection process.

The mapping between the identified categories versus constructed question cards are presented in Table 3.

In an effort to evaluate the keywords extracted from preliminary resources, *a concept map* [57] was created particularly based on the themes and the keywords. After conducting a theme analysis of the transcripts from the previous step, the codebook was validated using the information stored, and the frequency of each keyword in the transcript was categorized for further analysis.

The examination and comparison of the data was a continuous process where findings were taken back to selected participants from the study until an agreement has been either confirmed or verified. Whenever a situation was identified it was compared with several other previous situations for understanding their similarities as well as the differences. After finishing these tasks, we used axial coding to organize and combine the keywords with the selected categories from the transcribed documents and memos.

### *4.2. Card Creation Phase*

To prepare the proposed instrument, we initiated the card creation phase as follows. After having conducted a focus group with the management team of the company, we prepared an initial version of the cards by analyzing company-wide the business situations to find a match that could highlight a sub-personality characteristic extracted from BFI-44 items. With the help of the keywords, a constant comparative process was utilized where the collected situations from company documents were compared with the basic patterns of the Big Five questionnaire. Similar to any other personality assessment tests, participants were given two different selections to reflect their personality alignments. For each of the 60 cards, we selected a situation and a keyword from the

| Identified Themes | Derived Questions on Cards | |
|---|---|---|
| Talkative | 1, 8 | Extroversion |
| Assertive | 15, 22 | |
| Energetic | 29, 36 | |
| Active | 50, 57 | |
| Approachable | 41, 47 | |
| Outgoing | 43, 60 | |
| Curiosity | 2, 17 | Openness |
| Innovation | 23, 24 | |
| Flexibility | 30, 44 | |
| Immersion | 9, 10, | |
| Open-minded | 16, 37 | |
| Expressiveness | 45, 51 | |
| Agreeable | 3, 31, | Agreeableness |
| Directness | 38, 52, | |
| Altruistic | 42, 58 | |
| Optimistic | 53, 59 | |
| Tolerant | 4, 5 | |
| Idealistic | 19, 25 | |
| Tense | 32, 46 | Neuroticism |
| Haste | 18, 26 | |
| Uncertainty | 34, 56 | |
| Tension | 11, 12 | |
| Temperamental | 33, 39 | |
| Insecure | 49, 55 | |
| Unselfishness | 6,7 | Conscientiousness |
| Compassion | 13,14 | |
| Obedient | 20,21 | |
| Persistent | 27,28 | |
| Indulgence | 35,48 | |
| Industrious | 40, 54 | |

**Table 3:** *The identified 30 themes and trait related keywords with respect to the derived questions on cards*

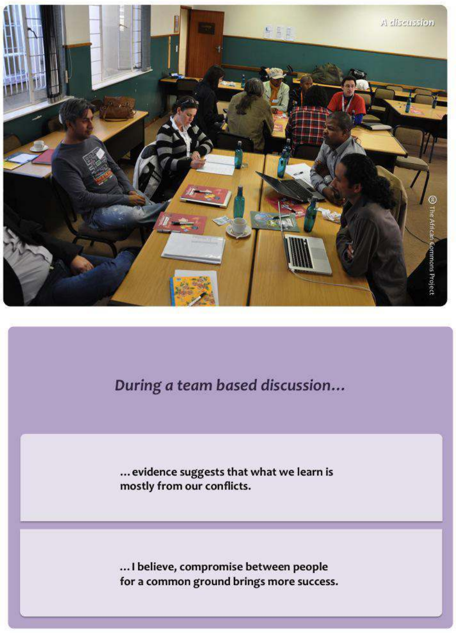
codebook. Next, we chose a picture with creative commons license, which reflected the keyword and the situation that was previously formed (see Figure 3 for a sample card).

As a result, we designed 60 cards (see Figure 3 for a sample). All cards had two faces. The front had a picture and a keyword that defined the theme of a picture. The goal was to visually prepare the participants for the hypothetical situation that was written on the other side of cards. Each situation had two different answers, which indicated the participants' inclination on a personality trait, e.g. being introverted or extroverted.

To sum up, created (context) cards were designed to highlight situations specific to software development domains where participants had always two answer choices embedded in each card. A number of questions defined a situation (e.g. cause and effect relationship) and sought for a single answer or an action.

### *4.3. Comparison Phase*

Finally, in the third phase, the cards were revised based on both the data collected from the next wave of interviews and the opinions of software practitioners from the industry. To substantiate the reliability of this part of the study, we consulted 10 experts (see Table 4) from the software industry and academia to discuss our findings. All cards were designed by using the



*During a team based discussion...*

...evidence suggests that what we learn is mostly from our conflicts.

...I believe, compromise between people for a common ground brings more success.

**Figure 3:** *A Two-faced Situational Context Card Example*

insights from software practitioners, which were evaluated by using an organizational psychologist and a graphic designer. In particular, a graphic designer assessed the graphical aspects of the instrument; its implications for the participants were discussed with an organizational psychologist. To achieve more accurate results, all reviewers (see Table 4) were carefully selected with the help of the management team of the company.

| Expert ID | Title | Age | Years of Experience | Expertise |
|---|---|---|---|---|
| E1 | Software Manager | 46 | 20 | Software Management |
| E2 | UX Designer | 36 | 7 | Usability |
| E3 | Graphical Designer | 30 | 4 | Graphic Design |
| E4 | Software Practitioner | 31 | 6 | Software Development |
| E5 | Clinical Psychologist | 43 | 16 | Psychology |
| E6 | Organizational Psychologist | 39 | 11 | Psychometrics |
| E7 | Instructional Designer | 38 | 9 | Software Engineering |
| E8 | Academic | 40 | 14 | Empirical Studies |
| E9 | Academic | 45 | 17 | Psychology |
| E10 | Academic | 58 | 25 | Research Management |

**Table 4:** *Expert Reviewers' Information*

The assessment model is partially based on *game playing*, which is a powerful technique that helps individuals to make meaningful decisions among available set of actions [58]. Consequently, evidence suggests that it is more motivating to allow participant to respond more interactively rather than answering a series of static questions [59]. Ultimately, our goal was to introduce these cards to an individual or a group of participants in a game form to facilitate the identification process of their personality traits.

### *4.4. Evaluation of the Instrument*

Table 5 shows the number of software practitioners with the Big Five personality traits and their percentage values in our sample population. Participants completed the BFI-44 by responding to each item in the inventory on a 5-point scale ranging from 1 (Strongly disagree) through 2 (Disagree), 3 (Neutral), 4 (Agree) to 5 (Strongly agree). Based on the raw score means and standard deviations, we developed local norms that we used to interpret personality traits percentiles relative to the sample population. Percentile equivalents for BFI-44 scores showed that participants were found high in extroversion, average in three traits namely openness, agreeableness, and conscientiousness. Lastly, participants found low in neuroticism.

| Personality Trait | Raw Score Mean±Standard Deviation | Percentile | Interpretation |
|---|---|---|---|
| Extroversion | 33.3±5.3 | 85 | Relatively high |
| Openness | 21.2±4.1 | 57 | About average |
| Agreeableness | 30.7±3.2 | 66 | About average |
| Conscientiousness | 21.2±2.4 | 62 | About average |
| Neuroticism | 11.2±3.6 | 29 | Relatively low |

**Table 5:** *The BFI-44 Inventory Scores Expressed as Mean Raw Scores, Standard Deviations, and Percentile values for each personality trait including all participants (n=216).*

In a classical viewpoint, Myers and McCaulley [16] suggest that in personality assessments, questions are quite general. Unlike a typical personality test, we argue that the content of the questions should be based on situations in terms of their impact on the personality traits, and therefore we used a group of expert reviewers to assess the quality and the importance of the content, i.e. the questions represents the problems happening in a software management context.

Furthermore, a survey instrument sets the stage for understanding the personality content with respect to several situations, which may eventually be evaluated to form a better instrument. In particular, an instrument should have the ability to represent the best possible cases that should be embedded in the survey questions. This was in accordance with our previously conducted interviews, which revealed the fact that situations could be interpreted within a level of importance; therefore, they should not be considered equally important.

The expert reviews and panel discussion were conducted at the company's headquarters with these individuals where all items were examined independently. Table 4 outlines the profile of the 10 participants including their roles (titles), age, years of experience and expertise. The experts were first asked to respond to a questionnaire developed for the assessment of the survey instrument where they rated content clarity, concreteness of its details, and importance of each question using a 4-point Likert scale (1=not clear, 4=very clear).

Concurrently, participants were also asked to assess the relevance of the the keywords and the situations that were extracted, which can be seen in Table 3. Our next goal was to quantify the extent of agreement among the individuals.

Based on the participants' ratings, the content validity index (CVI) was calculated as follows:

$$CVI = \frac{Number\ of\ raters\ giving\ a\ rating\ of\ 3\ or\ 4}{Total\ number\ of\ raters} \quad (1)$$

where a coefficient computed as 0 indicates a total lack of agreement whereas 1 indicates a total agreement [60]. Experts[3] suggested that items should be considered agreed if CVI was > 80%, where a value between 70 and 79 percentage agreement was considered questionable, and below that point was agreed upon as an unacceptable level of content validity of that item.

Among the participants, there was a 98% agreement on the relevance for the extracted keywords, only one was re-worded. Of the instrument's 60 items, the expert panel rated 52 as agreed and 8 as questionable where 6 items were rated as unacceptable for their clarity. All unacceptable items were eliminated, and new variants were introduced, and further argued with the panel. In addition, all questionable items were discussed with the participants to make a series of enhancements. Lastly, the final form of questionnaire was approved by the management team of the software development organization.

### *4.5. Software Team Visualization*

The goal of this part of the study is the visualization of software teams based on their personality traits. Consequently, our empirical findings were based on the percentage of traits, which allowed us to plot personality traits for team-based evaluations.

There are 12 questions for each dimension of personality trait. In order to calculate the percentage of the personality traits of a person, the number of questions that person selected as first choice (choice among two that shows the categorized personality traits) were calculated and then turned into a percentage value based on the fact that if all questions for that personality trait were selected, the percentage value would be accepted as 100%. For example, if 3 questions of 12 were selected as option one, which designates extroversion trait, the extroversion percentage would be 25%.

Based on the proposed scheme, the total percentages of the identified personality traits with respect to job titles of the sample population are shown in Table 6.

| **Title** | **Quantity** | **Extrov. %** | **Openness %** | **Agree. %** | **Cons. %** | **Neuro. %** |
|---|---|---|---|---|---|---|
| Product Owner | 7 | 67 | 54 | 47 | 67 | 21 |
| Software Developer | 24 | 41 | 68 | 25 | 35 | 23 |
| Software Testers | 8 | 79 | 35 | 42 | 37 | 28 |
| Software Specialist | 20 | 61 | 42 | 48 | 41 | 39 |
| System Analyst | 4 | 58 | 51 | 62 | 43 | 26 |

**Table 6:** *Overall Average Percentage of the participants with roles versus their personality traits. Extrov., Extroversion; Agree., Agreeableness; Cons., Conscientiousness; Neuro., Neuroticism; Openness, Openness to Experience.*

To increase the reliability of measures, we explore the personality traits of the same set of software practitioners using five factor model [61]. The results, as shown in Table 6, indicate that software testers were more extroverted than the other roles in the sample population followed by system analysts, product owners, and software specialists. Of the initial cohort, software developers were highly openness to experience. Relatively, a high level of agreeableness was observed in system analysts' personalities. Product owners scored high on the trait of conscientiousness. The majority of software specialist who responded to the questionnaire felt that they were more inclined to be high the personality trait of neuroticism.

[3] E5, E6, E7 (See Table 4)

We conducted the Big Five Inventory assessment (BFI-44 item scale) [20] and compared the results using correlation analysis. Table 7 illustrates the results of the instrument and five factor model. It can be inferred from the table that Extroversion dimension was highly correlated with the extroversion where openness measured with our tool was correlated with BFI-44 measured value for openness, Agreeableness was associated with agreeableness (BFI-44), and conscientiousness (BFI-44) was correlated with proposed conscientiousness measure. There was a strong connection between neuroticism (BFI-44) and neuroticism characteristic from proposed personality assessment tool ($N = 216$, $p < 0.01$).

| | Extrov. | Openness | Agree. | Cons. | Neuro. |
|---|---|---|---|---|---|
| Extroversion (BFI-44) | **0.93** | 0.12 | 0.19 | 0.32 | 0.28 |
| Openness (BFI-44) | 0.39 | **0.68** | 0.22 | 0.14 | 0.19 |
| Agreeableness (BFI-44) | 0.09 | 0.25 | **0.74** | 0.15 | 0.22 |
| Conscientiousness (BFI-44) | 0.23 | 0.18 | 0.17 | **0.79** | 0.12 |
| Neuroticism (BFI-44) | 0.22 | 0.17 | 0.24 | 0.04 | **0.81** |

**Table 7:** *The correlations between proposed instrument and BFI-44. Extrov., Extroversion; Agree., Agreeableness; Cons., Conscientiousness; Neuro., Neuroticism; Openness, Openness to Experience. The correlations are shown in bold typeface on the diagonal ($p < 0.01$).*

### *4.6. Personality-Team Radar*

A radar chart (graph) is a visual method of illustrating multivariate data in a two dimensional polar chart for the analysis of multiple variables where a set of variables represented together. This is suitable as a tool for comparison among a set of items [62]. Although a variant of team radar graph has previously been used in the software process improvement domain as " (the agile team radar" [see e.g., 63, 64]), to the best of our knowledge there is no prior study that combines personality traits and team radar concepts.

This research proposes a novel use of Personality-Team radar (or a personality radar chart) with coordinates of five dimensions of personality traits guide researchers to explore a team to visualize which traits are dominating the group characteristics. To illustrate team-based traits, our radar form includes all five personality traits (i.e. Openness, Conscientiousness, Extroversion, Agreeableness, Neuroticism), each shown in an axis. We used a percentage scale (ranging between 0% and 100%) inside a team radar, which guided us to represent an individual on the Big Five dimensions.

Figure 4 illustrates the conceptual figure for an empty Personality-Team radar graph.

### *4.7. Personality Measurements in Effective Software Teams*

To investigate the relationship between personality and team characteristics of software development teams, we interviewed a number of software team leaders to single out the available teams for this part of the study. We asked a series of questions to team leaders such as (1) *What kind of agile development methodology you are using?* (2) *How many individuals work in your software team and what are their titles (i.e. roles?)* (3) *What is your iteration length?* (4) *What are the best practices you are frequently using in your software development efforts?*

Apart from individual qualifications and their work experience as a team, a major selection criterion is the team's availability to work with us, due to the time limited issues associated with a commercial software team. After careful consideration based on their team attributes, a number

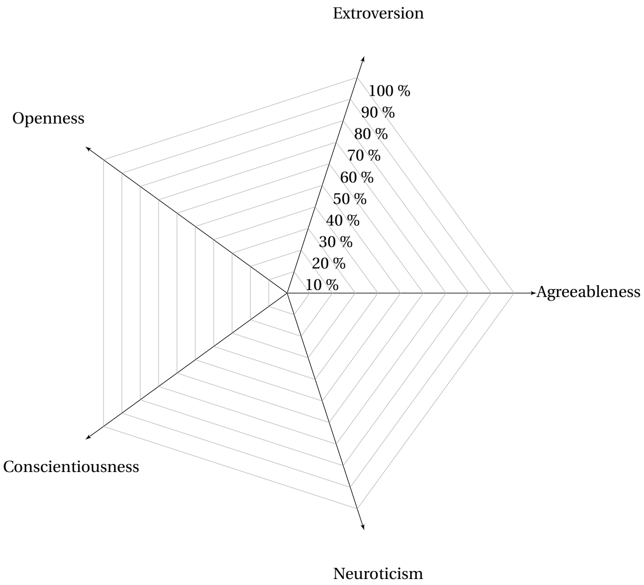


**Figure 4:** *A Personality-Team Radar Template*

of key performance indicators were selected to assist the productivity evaluation. For example, the on-time delivery ratio (ODR), which compares the number of user stories completed with the number of committed user stories, was employed. Team velocity (TV) was also examined (in terms of story points as an average in 5 sprints), as was the team's defect removal efficiency (TDRE) (as a percentage) and team availability (five teams were selected for this part of the study). Specifically, we enrolled 63 software practitioners: 7 Project managers, 24 Software developers, 8 Software Testers, 20 Software Specialist, and 4 System Analysts. In addition, all members of the five teams had at least one year working experience, which consist of individuals who were working on homogeneous tasks for the same project for about six months period.

Table 8 shows the values that were obtained from selected teams especially for measuring their potential progress or success with respect to the proposed key performance indicators. In light of this information, we selected and conducted the proposed approach on five software teams, and illustrated their personality characteristics by using a Personality-Team radar as follows:

| Team Name | ODR | TV | TDRE |
|---|---|---|---|
| Camelot | 0.78 | 120 pts. | 72% |
| Hector | 0.63 | 92 pts. | 84% |
| Finn | 0.82 | 125 pts. | 66% |
| Laran | 0.72 | 118 pts | 68% |
| Triskele | 0.40 | 30 pts. | 44% |

**Table 8:** *Key performance indicators for the selected software teams*

*Team Camelot*

*Camelot* was a software development team of twelve people. This team was a partner engagement team, which worked directly with customers on-site during several implementation processes. This team followed a hybrid agile approach to develop software with continuous deployment (i.e. release automatic software builds) to remove obstacles early. Their iteration length was 30 days. They performed daily stand-up meetings, pair programming when there was a challenging task, and preferred planning poker for estimating and planing. A highly experienced software developer from the team acted as the team lead as well as the scrum master for the whole process.

In the (personality) team analysis phase, we revealed the personality traits of a team with 12 individuals. Table 9 shows the job titles versus some of the personality characteristics of the team Camelot. It is apparent from this table that individuals such as product owners, system analysts who may be socially active for their positions are more extroverted. Not surprisingly perhaps, three of the team members were found highly extroverted. As team Camelot was considered as one of the most productive teams of the development organization (see Table 8), the observed increase in extroversion could be attributed to the new tasks and activities of software development that require more socially interactive teams. The results showed significantly high values of personality traits; emotional stability, agreeableness, extroversion, and conscientiousness.

| Participant ID | Job Title | E% | O% | A% | C% | N% |
|---|---|---|---|---|---|---|
| P57 | Product Owner | 92 | 67 | 58 | 67 | 17 |
| P38 | Software Developer | 33 | 67 | 50 | 67 | 42 |
| P40 | Software Developer | 25 | 42 | 75 | 75 | 8 |
| P34 | Software Developer | 83 | 42 | 50 | 42 | 42 |
| P47 | Software Developer | 42 | 67 | 75 | 67 | 25 |
| P13 | Software Developer | 42 | 67 | 58 | 33 | 8 |
| P45 | Software Developer | 58 | 42 | 42 | 42 | 8 |
| P37 | Software Tester | 83 | 67 | 58 | 58 | 25 |
| P44 | Software Tester | 83 | 42 | 67 | 67 | 17 |
| P23 | Software Tester | 83 | 33 | 50 | 67 | 17 |
| P24 | Software Tester | 58 | 67 | 75 | 33 | 42 |
| P29 | System Analyst | 75 | 42 | 33 | 50 | 25 |

**Table 9:** *Team Camelot with roles versus members' traits. E, Extroversion; O, Openness to Experience. A, Agreeableness; C, Conscientiousness; N, Neuroticism.*

Figure 5 presents Table 9 data on a radar graph. From the graph, we can see that the emotional stability reported significantly more than neuroticism. For the other traits, most of the individuals are in the range between 33% and 75%, which indicates that the team has high values for agreeableness, extroversion, and conscientiousness personality characteristics while neuroticism was below 43%.

To illustrate personality characteristics on a personality radar, we identified the team members and calculated their percentage for each trait. For another example, consider participant 38 from Table 9. On our scaling system this individual would be marked 33% on extroverted, 67% on openness, 50% on agreeableness, 67% on conscientiousness, and finally 42% on agreeableness scale. This approach allows us to see what percentage the five traits level of individuals are, on a team trait structure. To the best of our knowledge, such a percentage-based assessment result on a team level has not previously been published.

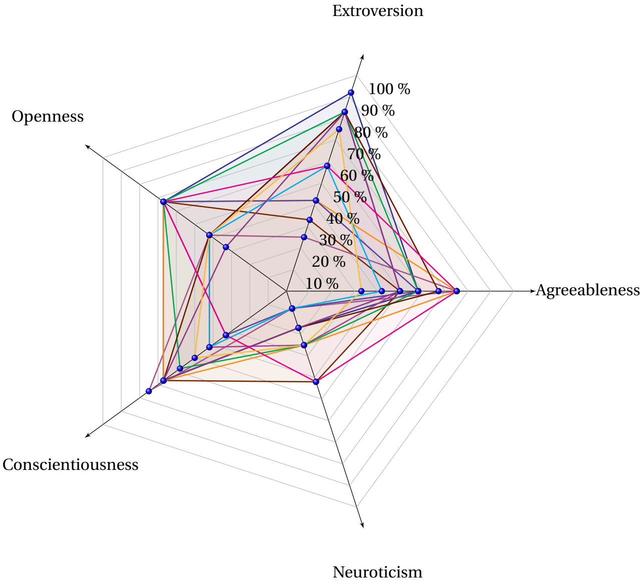


**Figure 5:** *Team Radar for Team Camelot*

*Team Hector*

*Hector* was a software development team of eight people, which was responsible for on-site customer training in product configuration, management, and deployment. The team customized their software process for handling difficult cases and finding and fixing a variety of technical problems. They used a tailored version of scrum to design and develop software in such a way that makes it easy to deploy, configure and manage. Their iteration length was 45 days. To track the progress of each team member, the team lead was also responsible for collecting timesheets for measuring the actual deliveries and also for exploring the time that team members spent and their task completion dates.

Team Hector consisted of eight software practitioners; four developers, two specialists, an analyst, and a product owner. Table 10 shows participants' id, job titles, and the percentage of personality traits found for the individuals from team Hector. Similar to team Camelot, team Hector was also populated with individuals with higher scores in agreeableness, extroversion, and conscientiousness and team had an inclination to be balanced in positive personality traits.

Figure 6 presents Table 10 data on a Personality-Team radar graph.

*Team Finn*

*Finn* was a software development team of ten people. Using extreme programming to deal with volatile requirements, this team performed solution design, technical installation and setups where it worked directly with customer support team and companies sales department commonly. They prefered to focus on technical practices, i.e. refactoring and regression testing to improve the quality of their software products. Their iteration duration was 15 days. They frequently used planning poker for estimation and pair programming to improve the quality of their work. Their motto was to reduce the cost of change by producing software artifacts that are readily

| Participant ID | Job Title | E% | O% | A% | C% | N% |
|---|---|---|---|---|---|---|
| P2 | Software Developer | 75 | 58 | 67 | 92 | 25 |
| P3 | Software Specialist | 67 | 58 | 42 | 67 | 17 |
| P8 | Software Specialist | 58 | 50 | 42 | 50 | 8 |
| P9 | Software Developer | 67 | 42 | 83 | 58 | 33 |
| P10 | Software Developer | 83 | 33 | 42 | 42 | 8 |
| P12 | System Analyst | 67 | 58 | 83 | 58 | 17 |
| P25 | Software Developer | 100 | 58 | 83 | 50 | 25 |
| P62 | Product Owner | 92 | 42 | 58 | 83 | 33 |

**Table 10:** *Team Hector with roles versus personality traits. E, Extroversion; O, Openness to Experience. A, Agreeableness; C, Conscientiousness; N, Neuroticism.*

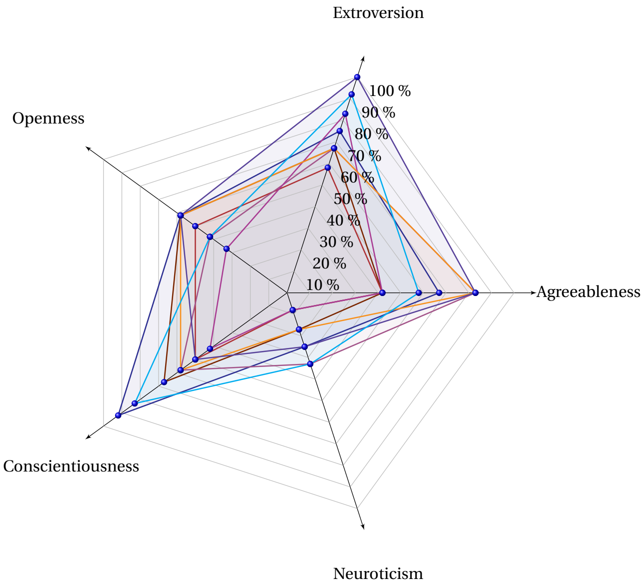


**Figure 6:** *Team Radar for Team Hector*

maintainable by proposing a simple but an elegant design.

Table 11 provides the personality characteristics data for team Finn. From this data, we can see that there were ten software practitioners with roles such as software specialist, software developer, software tester, system analyst, and a product owner in the team. Similar to Hector and Camelot, results indicate that team Finn also showed relatively high agreeableness and extroversion traits.

Figure 7 shows the data in Table 11 on a Personality-Team radar graph.

### *Team Laran*

*Laran* was a software development team of sixteen people who work closely with domain experts to implement software products. They collaborate with other teams (i.e. dealing with a variety of personalities) in the organization where some of their tasks require interpersonal skills including poise, tact and diplomacy. Their iteration duration was 30 days. Laran used a tailored version of Scrum as their software development methodology and they frequently used burndown

| Participant ID | Job Title | E% | O% | A% | C% | N% |
|---|---|---|---|---|---|---|
| P5 | Software Specialist | 75 | 67 | 75 | 75 | 25 |
| P6 | Software Tester | 75 | 33 | 58 | 67 | 17 |
| P7 | Software Specialist | 92 | 42 | 50 | 42 | 8 |
| P11 | System Analyst | 58 | 58 | 75 | 33 | 25 |
| P15 | Product Manager | 67 | 25 | 75 | 75 | 25 |
| P16 | Software Developer | 75 | 42 | 58 | 67 | 33 |
| P17 | Software Specialist | 75 | 67 | 58 | 58 | 8 |
| P18 | Software Developer | 92 | 57 | 92 | 33 | 17 |
| P59 | Software Developer | 67 | 42 | 75 | 67 | 42 |
| P61 | Software Developer | 42 | 33 | 58 | 42 | 33 |

**Table 11:** *Team Finn with roles versus personality traits. E, Extroversion; O, Openness to Experience. A, Agreeableness; C, Conscientiousness; N, Neuroticism.*

charts to monitor their sprint status, with planning poker employed to determine end durations of tasks. ROI (return on investment) values for item-based prioritization were evaluated and ultimately, they regularly performed 15 minute daily stand-up meetings.

As can be seen from Table 12 that Laran has the roles of software developer, software specialist, system analyst, scrum master, software testers, and a product owner, and it is also highly populated with extroverted individuals.

| Participant ID | Job Title | E% | O% | A% | C% | N% |
|---|---|---|---|---|---|---|
| P1 | Software Developer | 75 | 67 | 58 | 67 | 33 |
| P28 | Software Specialist | 75 | 58 | 67 | 50 | 8 |
| P30 | System Analyst | 58 | 42 | 67 | 67 | 17 |
| P31 | Software Specialist | 75 | 83 | 75 | 67 | 8 |
| P32 | Software Specialist | 75 | 42 | 50 | 75 | 25 |
| P33 | Scum Master | 67 | 67 | 83 | 67 | 8 |
| P35 | Software Tester | 67 | 58 | 42 | 58 | 17 |
| P36 | Software Specialist | 83 | 58 | 58 | 75 | 8 |
| P39 | Software Tester | 67 | 75 | 58 | 75 | 25 |
| P42 | Software Developer | 58 | 58 | 83 | 58 | 8 |
| P43 | Software Specialist | 67 | 50 | 83 | 67 | 17 |
| P51 | Software Tester | 33 | 50 | 42 | 83 | 8 |
| P52 | Software Specialist | 75 | 67 | 67 | 58 | 25 |
| P53 | Software Developer | 92 | 58 | 75 | 67 | 8 |
| P54 | Software Specialist | 67 | 75 | 67 | 83 | 8 |
| P55 | Product Owner | 92 | 83 | 58 | 83 | 17 |

**Table 12:** *Team Laran with roles versus personality traits. E, Extroversion; O, Openness to Experience. A, Agreeableness; C, Conscientiousness; N, Neuroticism.*

The single most striking observation to emerge from the data in comparison with other teams was that this team has the strongest values for agreeableness, extroversion, openness, and conscientiousness. Figure 8 shows the data in Table 12 on a personality radar graph.

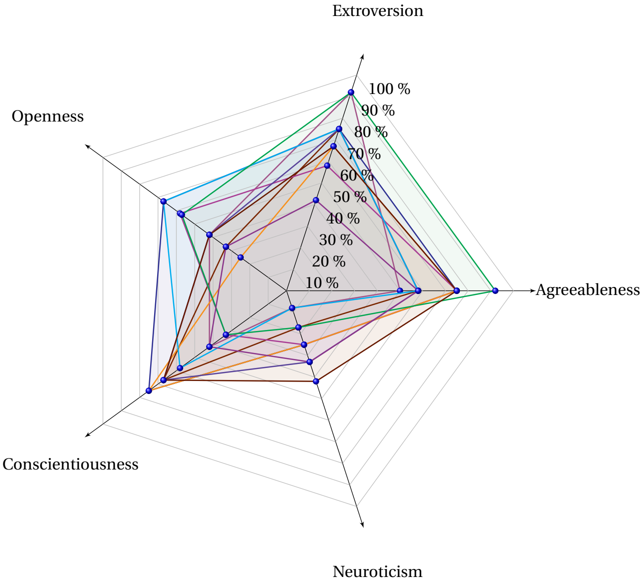


**Figure 7:** *Team Radar for Team Finn*

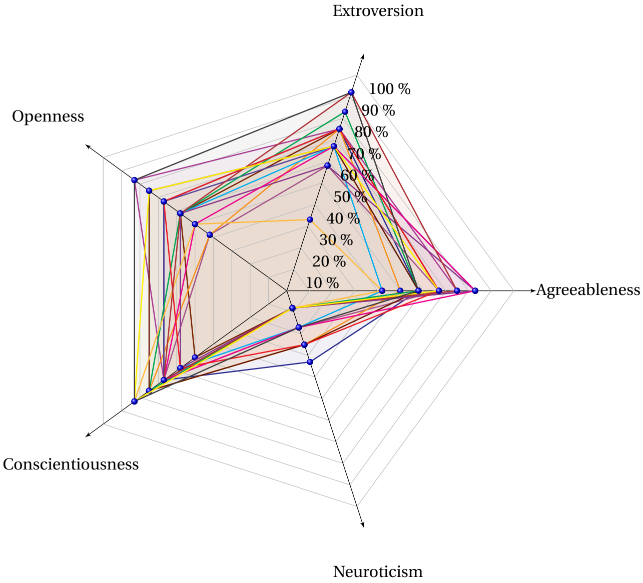


**Figure 8:** *Team Radar for Team Laran*

*Team Triskele*

*Triskele* was a research and development team of five people. All members have either a Masters degree or a PhD, and they continually deal with research oriented tasks using a time-boxed, iterative and relatively traditional methodology. The team adopted a strategy based on an advice taken from the Peopleware [65], with the goal of continuously grouping the members of their team to reduce the occurrence of unnecessary disturbances, which should naturally promote uninterrupted work. Triskele's iteration duration was 30 days. The team utilized pair programming, with code reviews bringing the benefit of experience exchange across the team. In addition, the team was accustomed to developing prototypes for pilot projects with volatile requirements.

Table 13 illustrates the personality characteristics of the team Triskele.

| Participant ID | Job Title | E% | O% | A% | C% | N% |
|---|---|---|---|---|---|---|
| P41 | Product Owner | 17 | 42 | 25 | 50 | 42 |
| P46 | Software Developer | 75 | 42 | 33 | 33 | 33 |
| P48 | Software Specialist | 17 | 58 | 42 | 33 | 8 |
| P49 | Software Specialist | 67 | 25 | 33 | 58 | 17 |
| P50 | Software Developer | 58 | 33 | 50 | 33 | 33 |

**Table 13:** *Team Triskele with roles versus members' traits. E, Extroversion; O, Openness to Experience. A, Agreeableness; C, Conscientiousness; N, Neuroticism.*

It is apparent from Table 13 that relatively low values can be observed for agreeableness, extroversion, openness, and conscientiousness in personality trait percentages for this team. Although the percentage of emotional stability trait was high which seems consistent with previous teams, the most striking result to emerge from the data was that a lack of direction was visualized. During the progress of this study, several issues and strong disagreement among the team members were reported.

Figure 9 below shows the team personality traits of team *Triskele*, scaled on a Personality-Team radar format.

### *4.7.1. A Brief Discussion about Findings*

For team Triskele, the results acquired from the data collected from the research and development team show that the team is relatively disconnected from social gatherings (e.g. they are not directly working with the customers), and there are no testers or a system analyst in their unit. The team did not seem to be strong in conscientiousness, openness, and agreeableness traits as well. It includes only a group of researchers who did not work with customers directly. Therefore, individuals who were selected for such teams were highly introvert. This result was compatible with the early days of software engineering landscapes when teams were working in isolated environments [65], and practitioners were found to be more introverted [5]. According to our results, this type of introversion was only observed in the *research and development* team, as we found all other four *cross-functional* software development teams were higher in extroversion, agreeableness, conscientiousness and openness traits.

In contrast to team Triskele, the other four teams, namely Camelot, Hector, Finn, and Laran, were found to be higher in agreeableness, conscientiousness and openness scales, which were considered to be working in socially interactive settings with a high tendency to be cooperative. In particular, they were the teams which had members with strong focus on stakeholder engagement and continuous delivery. In addition, the testers, system analysts and managers of the other four teams were found to be mostly higher in agreeableness and extroversion. The highly social

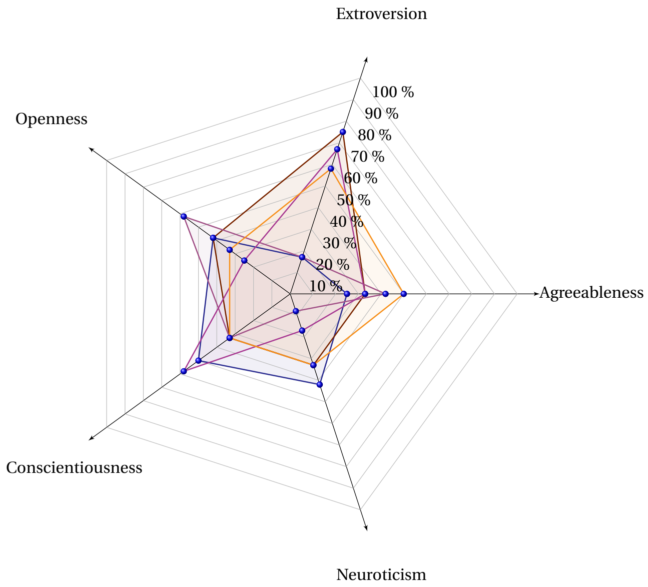


**Figure 9:** *Team Radar for Team Triskele*

nature of agile development tasks is compatible with openness to experience trait, which is defined with good communication and innovation skills, and a detail oriented mindset [66]. Lastly, for the teams working in agile environments, individuals with conscientiousness trait were observed. To be productive in such teams, organizational skills and open and adaptable to change mindset is crucial.

### *4.8. Validation Interviews*

To investigate the possible use of overall findings, one of the authors conducted semi-structured validation interviews [67] on all sixty three software practitioners who were working in the five software teams. The interviews were performed under the consent of an individual from the managerial team of the company as the company policy. The goal of this individual was to check the information collected and assess its potential consequences, and he was told to document the progress of the research.

To understand how software practitioners perceived their team and individual personality traits and their effects on team productivity, a series of personal interviews were conducted. The iterative interviewing technique was used where most of the data analysis happened during the interviews. Initially, interviewees were asked to detail their personality characteristics in different situations during the software development process. A goal was to explore the relations between productivity indicators of software development (e.g. project tasks, team velocity, software methodologies, team structures) and software practitioners' personality traits. Next, the researcher asked the interviewees a set of questions regarding the benefits of the software team visualization.

The researcher asked questions such as "*What do you think about complex or long term tasks, team velocity and its relation with personality traits?*", "*Do you think there is a relationship* *between continuously changing requirements, repetition of tasks and your teammates personality characteristics? If so, which ones?*", "*Does your team benefit from this new visualization perspective?*", "*Do you think personality traits may help you to select a software development process or improve your team productivity to some extent?*"

The interpretation of data collected from the interview transcripts reveals that the extrovert practitioners were described as relatively impatient on complex software development tasks and expressed themselves more intolerant to slow project velocities. In contrast, introvert practitioners were more motivated with long term and complex tasks. People who were found higher on the conscientiousness trait stated that they did not like continuously changing requirements. In addition such individuals claimed that they prefer to follow standard procedures to address encountered problems. Software practitioners who were high in the openness trait presented as largely uninterested in repetition in their tasks. The introverted individuals with higher scores in agreeableness trait preferred not to be in a team leader position while extroverted individuals with higher scores in emotional stability have a keen interest to improve their value by contributing to the team's social structure. Practitioners in conscientiousness trait were found to be more result-oriented, mostly they were not interested in the selection or type of the software development process. Individuals who scored higher in emotional stability trait were relatively good at coping with pressure and stress, and ultimately were less affected by increasing project velocity. To sum up, the preliminary findings of this study suggested that agile teams possess more extroversion personality characteristics. It has been noted that conventional teams and projects may tend to be more introverted, demonstrating a preference for private judgment over explicit extroverted interaction [31, 5].

In addition, majority of the participants reported that software team visualization could be a beneficial artifact for revealing the personality structure of a software team. These views surfaced mainly in relation to software team configuration, e.g. building a dynamic, measurable, and sustainable software team structure. Some interviewees suggested that such a visualization tool would be effective for identifying a social conflict, while others suggest that the approach can be considered as a novel contribution for creating a new personnel selection process.

## 5. Discussions and Conclusions

The primary intent of this study is to utilize an interactive approach to reveal the personality traits of industrial software practitioners who were working in a number of different teams with a variety of different roles in a medium-sized software company. We developed a context dependent questionnaire based on real situations captured from industrial settings. The secondary goal here is to build a team based personality measurement for inferring the personality characteristics of software teams. The results of this study indicate that understanding the differences in Personality-Team Radars can give us insights into how socially compatible teams could be formed. A Personality-Team Radar helps us to explore personality combinations that may work better when building effective team configurations.

Although one sample cannot represent the entire software practitioner community, we made some preliminary observations. We, firstly, experienced that not all traits were equally present in the company. Therefore, there should not be some dominant types for software professionals. However, this would be a fruitful area for further work. Secondly, individuals with traditional tasks matched with the traditional characteristics. Not surprisingly, perhaps, agile teams were found to be more extroverted. The results suggest the possibility of the instrument to contribute a value to software team configuration efforts. In addition, we have observed in Personality-Team

Radars that selected teams are quite balanced (i.e. overall personality characteristics of all teams are in the inner webs of the agreeableness, extroversion, openness, and conscientiousness traits). The teams were selected by the software company as these teams were expected to have effective team structures.

### *5.1. Threats to Validity*

Here, we discuss several potential threats to validity that were addressed. To cope with problems of construct validity, first we assessed the content validity of the questionnaire by using expert judgment (refined by using a number of interviews) in order to compare the conceptual and operational form of each question. Consequently, a group of experts evaluated the content of each question (compare their understandably).

To deal with internal validity problems such as *history effect*, we have not changed our measuring device (i.e. interactive survey instrument) during the course of the study. Furthermore, we have not observed any change of participants' circumstances, which may cause a change in their preferences during the analysis. In the industrial case study, our approach was conducted with each team once; therefore there was no report of experimental deterioration. To avoid the participant effect, the assessment was constructed in a highly interactive form to keep the focus of the individuals on the context rather than a single question.

To cope with the external validity problems, we conducted a pilot study (see [43]), and its results were discussed with several experts from academia. Still, the generalizability of these results is subject to certain limitations for instance the relatively small sample size could limit external validity. Therefore, this study should be replicated with a larger sample size in order to confirm the stability of the initial results and consequently address external validity issues. In fact our systematic approach is well documented, which is easily applicable to an alternate setting. Lastly, we used industrial practitioners instead of college sophomores, which we believe should also increase the precision of the results.

### *5.2. Limitations*

While this is the first exploratory study to report personality traits of software practitioners using a novel visualization method, a number of potential limitations need to be considered. First, all kinds of personality tests build upon self-report (i.e. subjective evidence), which inherently involves the possibility that participants report false choices. Although a game-like approach was likely to improve participants' motivation to reveal true preferences, to deal with this issue, we informed the participants that the results would be kept confidential. Second, we conducted a cross-sectional study where assessment was conducted at a single point in time. Even though this type of research is relatively cheap and easy to perform, it could only be considered as a snapshot of the current characteristics of the population.

In the current study we used an assessment tool to reveal the personality traits of software practitioners. Individual traits were evaluated using 12 questions, giving a total of 60 questions for the 5 dimensions of interest. Therefore, the results of this study are limited by these questions. However, based on real situations and events, the questionnaire was constructed by a systematic qualitative assessment, and its results were compared with BFI-44 results. Consequently, these findings have shown that the proposed instrument has an ability to assess personality traits.

This study was limited by the presence of an individual from the management team who partially participated some parts of the research. During the validation interviews, however, all participants' private information (e.g. names) was modified through all transcripts to protect participant confidentiality. Participants were assigned to special IDs (e.g. E1, P1, etc.) as can be seen in Tables that shows participants' details. We conducted group sessions with the software teams while collecting data related to personality characteristics. Since it can be difficult to avoid groupthink in specialized task groups such as software teams, we asked every participant to fill out their responses individually without sharing their choices with team members. During the personality assessment, discussions among the group members were encouraged to explore situations however there were no collaborative decision making reported.

Lastly, the findings should be cautiously interpreted because the current investigation was limited to a single company. The results therefore might not be transferable to other software companies directly. Nevertheless, the authors believe that this research provides interesting insights, and it should be replicated in different settings with more software teams to compare findings. Apart from the limitations above, this research demonstrated the successful use of an interactive personality reporting method, and ultimately contributed to the understanding of personality characteristics of software practitioners and teams.

### *5.3. Conclusions*

Software development is a complex activity [68] and part of that complexity can be tackled by harnessing the power of human team dynamics. Furthermore, human characteristics, such as culture and disharmony, are noted to influence software development processes [49]. It is clear from recently published research that a focus on individual and group behavior in software development is both of interest and importance to our field [69]. We therefore suggest that efforts such as those outlined in this paper can serve to improve our understanding of human dynamics in software development with positive benefits for software projects and our discipline as a whole.

In this industrial case study, we have demonstrated that an interactive approach is able to reveal the personality traits of software practitioners particularly for the selected company. Returning to the hypothesis posed at the beginning of this study, it is now possible to state that the personality characteristics of the software practitioners working for that organization were revealed and illustrated by our approach. Our approach contributes to the software development process by illustrating a team's personality structure on the team radar. The results of this study indicate that effective teams that participated in this study consisted of teams of individuals whose personality traits; emotional stability, agreeableness, extroversion, and conscientiousness, were found to be high in their team structure, and were plotted on a team-radar. Such an approach should further improve our ability to build or configure more effective software teams or perhaps improve the process of integrating a new member into a software team structure.

In practice, it will be useful not only for building software development teams but also having a favorable selection from a set of individuals with the same skill set with different personalities. For example, the approach might be useful for exploring the most suitable person from a group of candidates by using an observable variability in aspects of their behavior. In sum, we believe that our approach has the potential to supersede the paper based personality tests, particularly for software development organizations. However, our findings may not be extrapolated to claim such results yet. Further studies, which take these outputs into account will need to be undertaken.

Regardless of being agile or traditional, a software team is formed to respond to key challenges such as increased diversity in activities and the required interactions in a software development process. During the follow-up interviews, we have indicated the agile proponents are more open to new experiences. Nearly all individuals we interviewed have a significant inclination to be agile, and are found to be extroverted. Consequently, our interactive approach shall provide a mechanism for balancing agility and discipline from a team configuration perspective to some extent. Overall, this information has practical benefits in conflict-handling and exploring personality aspects of software teams.

### *5.4. Future Work*

To improve the performance of our interactive approach, more empirical case studies should be performed for revealing personality traits of individuals in different software organizations. For example, new cards may be designed and the study should be balanced to incorporate experience gained through future team-based sessions. Considerably more work will need to be done to explore socially compatible individuals. Based on the collected information, an automated team builder will be developed to suggest possible combinations of practitioners to software management. More investigation on a personality-based team builder would help us to establish a greater degree of accuracy on the subject matter.

To assess the personality traits attracted to the software engineering profession, multi-national research is required. It would be interesting to compare individuals from different countries and analyze its implications on a global extent. This information can also be used to develop a list of compatible personalities or perhaps a lookup table for conflicting personality characteristics both of which may be extremely useful for the team building process in software projects. However, the experience from the field suggests that empirical analyses alone may not be enough to convince management towards adoption.

Equally important, several visualization methods have emerged naturally as a way of presenting our findings during this study. The authors believe that the results of this research should be properly illustrated; therefore, further research should be done to investigate more representation techniques for the gathered data and initial findings. Future studies on the visualization techniques to improve the presentation forms are therefore highly recommended.

## Acknowledgments

This work is supported, in part, by Science Foundation Ireland grant number 13/RC/2094 to Lero, the Irish Software Research Centre (`www.lero.ie`).

## References

[1] R. Colomo-Palacios, C. Casado-Lumbreras, P. Soto-Acosta, F. J. Garca-Pealvo, E. Tovar-Caro, Competence gaps in software personnel: A multi-organizational study, Computers in Human Behavior 29 (2013) 456–461.

[2] R. Colomo-Palacios, C. Casado-Lumbreras, P. Soto-Acosta, S. Misra, F. Garca-Pealvo, Analyzing human resource management practices within the GSD context, Journal of Global Information Technology Management 15 (2012) 30–54.

[3] S. Ryan, R. V. O'Connor, Development of a team measure for tacit knowledge in software development teams, Journal of Systems and Software 82 (2009) 229–240.

[4] P. Lenberg, R. Feldt, L. G. Wallgren, Behavioral software engineering: A definition and systematic literature review, Journal of Systems and Software 107 (2015) 15–37.

[5] S. Cruz, F. Q. B. da Silva, L. F. Capretz, Forty years of research on personality in software engineering: A mapping study, Computers in Human Behavior 46 (2015) 94–113.

[6] R. R. McCrae, O. P. John, An introduction to the five-factor model and its applications, Personality: critical concepts in psychology 60 (1998) 295.

[7] W. T. Singleton, Social skills, Springer, 1983.

[8] J. McGonigal, Reality is broken: Why games make us better and how they can change the world, Penguin Pr, 2011.

[9] P. Yoong, Managing IT Professionals in the Internet Age, IGI Global, 2006.

[10] H. C. Haywood, D. Tzuriel, Interactive assessment, Springer, 1992.

[11] K. Werbach, D. Hunter, For the Win: How Game Thinking Can Revolutionize Your Business, Wharton Digital Press, 2012.

[12] E. Kaluzniacky, Managing psychological factors in information systems work: An orientation to emotional intelligence, Information Science Publishing, 2004.

[13] G. Matthews, I. J. Deary, M. C. Whiteman, Personality Traits, Cambridge University Press, 3 edition, 2009.

[14] D. Mayer, A. Stalnaker, Selection and evaluation of computer personnel-the research history of SIG/CPR, in: Proceedings of the 1968 23rd ACM national conference, ACM, pp. 657–670.

[15] C. Jung, H. Baynes, R. Hull, Psychological types, Routledge, 1991.

[16] I. Myers, M. McCaulley, N. Quenk, A. Hammer, MBTI manual, Consulting Psychologists Press, 1999.

[17] R. R. McCrae, O. P. John, An introduction to the five-factor model and its applications, Journal of personality 60 (1992) 175–215.

[18] P. Kline, The handbook of psychological testing, Psychology Press, 2000.

[19] D. Keirsey, M. Bates, Please understand me: Character & temperament types, Prometheus Nemesis Michigan, 1984.

[20] O. P. John, E. M. Donahue, R. L. Kentle, The big five inventoryversions 4a and 54, Berkeley: University of California, Berkeley, Institute of Personality and Social Research (1991).

[21] L. R. Goldberg, J. A. Johnson, H. W. Eber, R. Hogan, M. C. Ashton, C. R. Cloninger, H. G. Gough, The international personality item pool and the future of public-domain personality measures, Journal of Research in personality 40 (2006) 84–96.

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

[23] J. Chao, G. Atli, Critical personality traits in successful pair programming, in: Proceedings of the conference on AGILE 2006, AGILE '06, IEEE Computer Society, 2006, pp. 89–93.

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

[25] N. Rodrigues, T. Rebelo, Incremental validity of proactive personality over the big five for predicting job performance of software engineers in an innovative context, Journal of Work and Organizational Psychology 29 (2013) 21–27.

[26] S. T. Acuna, M. Gomez, N. Juristo, How do personality, team processes and task characteristics relate to job satisfaction and software quality?, Information and Software Technology 51 (2009) 627 – 639.

[27] M. V. Kosti, R. Feldt, L. Angelis, Personality, emotional intelligence and work preferences in software engineering: An empirical study, Information and Software Technology 56 (2014) 973–990.

[28] R. Rutherfoord, Using personality inventories to help form teams for software engineering class projects, ACM SIGCSE Bulletin 33 (2001) 73–76.

[29] J. Karn, T. Cowling, A follow up study of the effect of personality on the performance of software engineering teams, in: Proceedings of the 2006 ACM/IEEE international symposium on Empirical software engineering, ACM, pp. 232–241.

[30] J. Karn, S. Syed-Abdullah, A. Cowling, M. Holcombe, A study into the effects of personality type and methodology on cohesion in software engineering teams, Behaviour & Information Technology 26 (2007) 99–111.

[31] L. Capretz, Personality types in software engineering, International Journal of Human-Computer Studies 58 (2003) 207–214.

[32] R. Sach, M. Petre, H. Sharp, The use of mbti in software engineering, in: 22nd Annual Psychology of Programming Interest Group 2010.

[33] A. Da Cunha, D. Greathead, Does personality matter?: an analysis of code-review ability, Communications of the ACM 50 (2007) 109–112.

[34] N. Gorla, Y. Lam, Who should work with whom?: building effective software project teams, Communications of the ACM 47 (2004) 79–82.

[35] L. Capretz, F. Ahmed, Making sense of software development and personality types, IT Professional 12 (2010) 6–13.

[36] L. Capretz, F. Ahmed, Why do we need personality diversity in software engineering?, ACM SIGSOFT Software Engineering Notes 35 (2010) 1–11.

[37] F. Ahmed, L. Capretz, P. Campbell, Evaluating the demand for soft skills in software development, IT Professional 14 (2012) 44–49.

[38] J. Karn, A. Cowling, Using ethnographic methods to carry out human factors research in software engineering, Behavior research methods 38 (2006) 495–503.

[39] L. F. Capretz, D. Varona, A. Raza, Influence of personality types in software tasks choices, Computers in Human Behavior 52 (2015) 373–378.

[40] C. Tavris, S. O. Lilienfeld, S. J. Lynn, J. M. Lohr, Science and pseudoscience in clinical psychology, The Guilford Press, 2004.

[41] M. L. Moffitt, R. C. Bordone, The handbook of dispute resolution, Wiley. com, 2012.

[42] S. Hayes, D. Barnes-Holmes, B. Roche, Relational Frame Theory: A Post-Skinnerian Account of Human Language and Cognition, Kluwer Academic Publishers, 2002.

[43] M. Yilmaz, R. O'Connor, Towards the understanding and classification of the personality traits of software development practitioners: Situational context cards approach, in: Software Engineering and Advanced Applications (SEAA), 2012 38th EUROMICRO Conference on, IEEE, pp. 400–405.

[44] M. Yilmaz, R. V. O'Connor, A software process engineering approach to improving software team productivity using socioeconomic mechanism design, ACM SIGSOFT Software Engineering Notes 36 (2011) 1–5.

[45] B. Glaser, A. Strauss, The discovery of grounded theory: Strategies for qualitative research, Chicago:Aldine, 1967.

[46] C. Goulding, Grounded theory: A practical guide for management, business and market researchers, SAGE Publications, 2002.

[47] G. Coleman, R. O'Connor, Using grounded theory to understand software process improvement: A study of irish software product companies, Information and Software Technology 49 (2007) 654–667.

[48] G. Coleman, R. O'Connor, Investigating software process in practice: A grounded theory perspective, Journal of Systems and Software 81 (2008) 772–784.

[49] P. Clarke, R. V. O'Connor, The situational factors that affect the software development process: Towards a comprehensive reference framework, Information and Software Technology 54 (2012) 433–447.

[50] P. Runeson, M. Höst, Guidelines for conducting and reporting case study research in software engineering, Empirical Software Engineering 14 (2009) 131–164.

[51] R. Yin, Case study research: Design and methods, volume 5, Sage Publications, 2008.

[52] M. Yilmaz, R. V. OConnor, P. Clarke, An exploration of individual personality types in software development, in: Proceedings of the 21st European Systems and Software Process Improvement and Innovation (EuroSPI 2014), Springer, 2014, pp. 111–122.

[53] R. R. McCrae, P. T. Costa, Reinterpreting the myers-briggs type indicator from the perspective of the five-factor model of personality, Journal of personality 57 (1989) 17–40.

[54] A. J. Mills, G. Durepos, E. Wiebe, Encyclopedia of case study research, Sage Publications, 2009.

[55] L. R. Goldberg, The structure of phenotypic personality traits., American psychologist 48 (1993) 26.

[56] J. Creswell, Qualitative inquiry and research design: Choosing among five approaches, Sage Publications, 2012.

[57] J. Maxwell, Qualitative research design: An interactive approach, Sage Publications, 2004.

[58] K. Salen, E. Zimmerman, Rules of play: Game design fundamentals, MIT press, 2003.

[59] R. E. Ferdig, Interdisciplinary Advancements in Gaming, Simulations and Virtual Environments: Emerging Trends: Emerging Trends, IGI Global, 2012.

[60] E. R. Lenz, Measurement in nursing and health research, Springer Publishing Company, 2010.

[61] J. Wiggins, The five-factor model of personality: Theoretical perspectives, The Guilford Press, 1996.

[62] R. Basu, Implementing quality: a practical guide to tools and techniques: enabling the power of operational excellence, Cengage Learning Business Press, 2004.

[63] N. Moe, T. Dingsøyr, E. Røyrvik, Putting agile teamwork to the test–an preliminary instrument for empirically assessing and improving agile software development, Agile Processes in Software Engineering and Extreme Programming (2009) 114–123.

[64] M. Ringstad, T. Dingsøyr, N. Brede Moe, Agile process improvement: Diagnosis and planning to improve teamwork, Systems, Software and Service Process Improvement 172 (2011) 167–178.

[65] T. DeMarco, T. Lister, Peopleware: productive projects and teams, Dorset House Publishing Company, 1999.

[66] A. Casado-Rivas, M. M. Archidona, The influence of personality traits on software engineering and its applications, Agile Estimation Techniques and Innovative Approaches to Software Process Improvement (2014) 83.

[67] A. Bryant, K. Charmaz, The Sage handbook of grounded theory, Sage Publications, 2010.

[68] P. Clarke, R. V. O'Connor, B. Leavy, A complexity theory viewpoint on the software development process and situational context, in: Proceedings of the International Workshop on Software and Systems Process, ACM, pp. 86–90.

[69] H. Sharp, Y. Dittrich, C. de Souza, The role of ethnographic studies in empirical software engineering, IEEE Transactions on Software Engineering 42 (2016) 786–804.

## APPENDIX: Situational Context Cards

*When a customer calls with an unexpected problem and requests a team member…*

…I might be the right person to talk them through the issue.

…I refer someone else to handle the call.

*When you are interacting with your teammates…*

…I find myself participating easily.

…I prefer to listen and think.

*Imagine yourself in a face to face conversation, you will usually prefer…*

…to express your ideas.

…to listen to others.

*While two of your teammates are arguing…*

…you participate and also say what is on your mind.

…you follow the conversation.

*All members of your software development team should…*

"…just, focus on their individual tasks."

"…be prepare to be involved in the tasks of other members."

*Consider a meeting with lots of other people from the company, you are waiting for CEO who is late…*

…you try to interact with other employees in the room.

…you prefer to keep your own counsel.

*As a teammate do you think yourself …*

…approachable by others.

…slightly reserved.

*During a work day, are you…*

…prepared to be interrupted by your colleagues on any matters of concern.

…preferred to be left alone with your own tasks.

*You are at a client/customer party, do you…*

…enjoy meeting with new people.

…prefer to spend your time with the people you know.

*Your boss asks you go on a night out with some new customers ...*

... no worries, I would love to, meeting with new people interests me.

... I would prefer that my boss to asks someone else.

*During a daily standup meeting, do you...*

... feel that your motivation is increasing.

... feel that these meetings are probably necessary.

*Considering yourself in a software team, do you prefer...*

... to interact with your colleagues as a group.

... to interact one colleague at a time.

*In your work setting, are you more interested to learn ...*

... from different sources (e.g. colleagues, web).

... from a single and concrete source (e.g. a book).

*When you are starting a new project, do you prefer to work on...*

... the difficult or new parts first.

... the known solutions first.

*Using innovation in software projects...*

... improves the quality.

... potentially jeopardizes the quality.

*In a team based decision-making process, you find yourself arguing against others. Do you argue based on...*

... some of your experiences.

... stick to the facts.

*When your manager says we need to improve productivity of our team by 1.2, are you inclined to interpret it ...*

... in the strict sense of the number.

... as a general objective.

*In a work day, I...*

... do not mind dealing with unexpected challenges.

... would like things to work according to my plan.

*I like...*

... to focus on the details of a situation.

... to have a 'big picture' view.

*When applying for a new job, which skill do you prefer to highlight...*

... how practical you are.

... your creativity and imagination.

*When something goes wrong during the project, do you first...*

... try to stick with the original plan.

... explore new options.

*In a team project do you...*

... address only the specifications in the plan.

... try to add additional features in the time allowed.

*A self-professed visionary with a novel idea...*

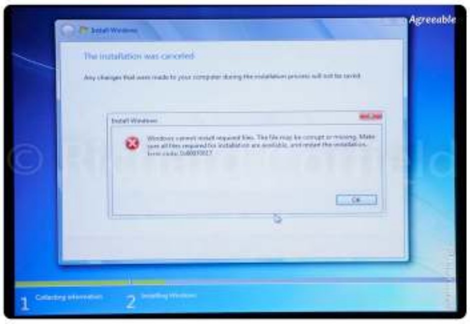
... is somewhat of a distraction.

... may have something of interest.

*In a team, an individual should be able to...*

... express himself/herself directly to his/her teammates even it causes personal conflicts.

... find a different way to explain the situation to avoid conflicts.



*There is an important defect in a module, your team has worked hard but cannot find a resolution...*

... we should discuss the issue with customer to come to a solution.

... we should try to find any workaround to cover it up temporarily.

*When performing a peer review do you prefer to...*

... discuss all problems in detail.

... propose an improvement without mentioning the causes.

*What would you say about messy working environments?*

... I should inform management to take care of the mess.

... I really don't care unless it affects my work.

*A team based agreement...*

... try to build consensus within the team.

... may require a fast execute decision.

*When there is a conflict,...*

... identify the person(s) at the source of the problem.

... accept the group responsibility.

*If the project deadline is overdue, do you prefer to...*

... require all people to stay at work until it is finished.

... volunteer to discuss the issue with the customer.

*A teammate is the one, who...*

... is not afraid to point out my faults.

... supports me at all costs.

*If an important performance specification is not being met...*

... we should report this immediately.

... we should report this at the end of iteration.

*You are eligible for two different positions in a company, do you...*

... accept a job that is most beneficial to the company.

... accept a job for more financial benefits.

*Your company needs someone on site for a month with certain financial benefits, do you...*

... put yourself forward as it is important for the company.

... accept if no one else is suitable.

*A software practitioner should contribute more to...*

... team goals.

... his/her own self-development.

*We have projected that we are not going to be able to achieve the expected quality with the requested module...*

... .we ask for an extension to achieve the intended quality.

... we do not emphasize the issue unless the client brings up the issue as a concern.

*When considering someone who does not fit your team, which sentences would you say...*

"... the individual does not have any idea what he/she is talking about, and wasting our time."

"... I think this individual has skills better suited with another team."

*A team member should...*

"... not easily get excited and show her/his feelings in a conflict."

"... share her/his feelings with her/his teammates and react to their problems."

*Whenever a new task arrives for our team work...*

... I am eager to get on with it otherwise I am concerned.

... I will be inclined to examine the new task in detail before committing.

*In a team based setting,...*

... I decide quickly, act fast, and revisit the situation if necessary.

... I ponder to avoid a bad decision and not to have any doubts about things.

*During the end phase of a finished project, are you more likely to...*

... be focusing on final user acceptance tests.

... be focusing on the next project.

*I would be more relaxed...*

... once all the software decisions have been made.

... to defer the decisions to just the right time.

*We need to buy a new hardware but management would not like to allocate sufficient resources for a good device...*

... I get concerned about such things.

... I do not care much about it.

*During a team based discussion...*

... evidence suggests that we learn from conflicts.

... compromise between people for a common ground brings more success.

*Your team failed to deliver a product on schedule...*

... I will forgive myself and my team, it is quite common in the software business.

... I will blame myself, and I think my team did not deserve any salary bonus for a month.

*You will be moved to a new position, would you prefer to…*

… be a member of one of the best team which has a lot of conflicts with other teams.

… be a member of one of the moderately productive teams of the company.

*While your supervisor is assessing your work unfairly…*

… you make a formal complaint to the upper management.

… you try to talk things through with your supervisor.

*You have to write a short sentence to thank all the people who contributed to the success of the project. You start…*

"The team owes the success of this project to certain individuals… "

"The team shares equally the success of this project… ."

*Your teammate wants you to attend a meeting with him/her, however your participation may not be necessary…*

… I will refuse to go because I am already committed.

… I will accompany him/her because she asked for help.

*Rewards are…*

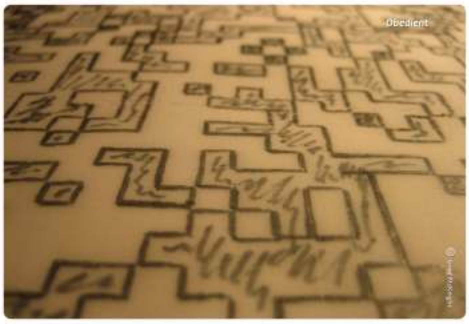
… based on an established performance criteria.

… at the discretion of the manager.

*In a team based discussion, you prefer to be…*

… firm and clear in expressing your thoughts.

… open minded and inspiring in expressing your thoughts.



*Consider the situation that one of your team members will be fired, you propose…*

… the one who is less productive without any further consideration.

… to consider number of factors such as their ability to find another job.

*Consider a difficult task that was assigned to your supervisor, which is overdue …*

… I would throw it to his attention but not be accounted for the problem.

… I would be willing to help him out with the responsibilities.

*A team member should…*

… note individuals' performance.

… be tolerant to the members of the team.

*When you are applying for a role in a project, you prefer...*

...a project with clearly defined aims.

...a project that has some exploratory stages.

*When you are choosing the people you work with...*

...you value their consistency and stability in their behavior.

...you value their new ideas and friendliness.

*A team mate is suffering a personal problem...*

... I don't interfere because it is not related to work.

... I feel for him/her, I can carry out some of his/her tasks.

*If you need to disappoint a teammate?*

...I will do it directly, and frankly.

... I will do it over a period of time.

*Do you prefer to work on a project ...*

...with a lot of targets and technical challenges that can improve your reputation over time.

...with quick prototypes.

*While evaluating someone's compatibility within your team. You would say...*

"...the most important thing is self-discipline, s/he may have to change to fit to our team."

"...we should be able to change the team culture to accommodate his/her characteristics. "