



## Influence of personality types in software tasks choices



Luiz Fernando Capretz<sup>a,\*</sup>, Daniel Varona<sup>b</sup>, Arif Raza<sup>a</sup>

<sup>a</sup> Department of Electrical & Computer Engineering, Western University, London, Ontario N6A5B9, Canada

<sup>b</sup> Department of Informatics, University of Informatics Science, Havana, Cuba

### ARTICLE INFO

#### Article history:

#### Keywords:

Human factors in software engineering  
Software life cycle  
Human aspects of software development  
Software psychology  
Empirical software engineering

### ABSTRACT

According to psychology, not everybody can excel at all kinds of tasks. Thus, chances of a successful outcome of software development increase if people with particular personality types are assigned to their preferred tasks in the project. Likewise, software development depends significantly on how software practitioners perform their tasks. This empirical study surveys 100 Cuban software developers, who also teach or study at the University of Informatics Sciences in Havana, Cuba. This work aims to find possible patterns that link personality types to role preferences in a software life cycle. Among the various roles, system analyst, software designer, and programmer are found to be the most preferred among the participants. In contrast, software tester and software maintainer happen to be the least popular roles among software engineers.

© 2015 Elsevier Ltd. All rights reserved.

### 1. Introduction and background

Software engineering has been one of the most prominent professions over the last 20 years, and it is projected to evolve even further. Engineering software comprises stages in distinct areas, such as analysis, design, programming, testing, and maintenance. Today, specialties within software engineering are as diverse as in any other profession. Additionally, software engineers need to communicate more effectively with users and team members, thus the people dimension of software engineering is as important as technical expertise.

Software project managers have always faced the problem of assigning tasks to the right people within a team in such a fashion that increases the chances of successful project completion (DeMarco & Lister, 1999). Different ideas have been tried to use diverse ways to maximize performance (Curtis, Hefley, & Miller, 2001) and make choices in the software engineering process (van Solingen, Berghout, Kusters, & Trienekens, 2000). Those ideas involve: motivation (software engineers tend to perform better if they are motivated to do specific tasks), the environment, and personality type, or a combination of these factors. Motivation and the environment are known to influence task performance. Motivation is generally a powerful element in the performance of task goals, especially in the IT field (Gallivan, 2004; Hall, Sharp, Beechman, Badoo, & Robinson, 2008). However, motivation is often insufficient

for influencing task accomplishment on its own. Feldt, Angelis, Torkar, and Samuelsson (2010) state that environmental factors alone cannot improve task performance. Hence, there are multiple factors involved in the performances of software engineers (Moore, 2000). This study specifically investigates the role of individual preferences in software projects, while neglecting the elements of motivation and environment, which have been the focus of most scholarly research on this topic. Thus this work exclusively investigates the role of individual preferences in software projects, focusing explicitly on how personality types affect preferences for specific software roles, not performance in executing them.

Several studies investigate the relationship between software engineer personalities and performance by identifying associations between particular personality types and specific tasks in software development. For example Choi, Deek, and Im (2008) and Da Cunha and Greathead (2007) address specific issues related to programming. There are conflicting evidences that personality alone is not a good predictor of programming performance. Acuna and Juristo (2004) introduce a capability-person relationship model that can be used by software project managers to assign tasks to people based on soft skills. Acuna, Juristo, and Moreno (2006) report that properly assigning people to development roles is crucial for creating productive teams, and their human capacity-based procedure can aid managers at small- to medium-sized software organizations.

Ritcher and Dumke (2015) adapt the Big Five method for software engineering with a Failure Mode and Effect Analysis method that models the human factor as a risk factor in the software engineering process and examines methods to evaluate psychological

\* Corresponding author.

E-mail addresses: [lcapretz@uwo.ca](mailto:lcapretz@uwo.ca) (L.F. Capretz), [dvarona@uci.cu](mailto:dvarona@uci.cu) (D. Varona), [araza22@uwo.ca](mailto:araza22@uwo.ca) (A. Raza).

characteristics to diagnose expected productivity. Capretz and Ahmed (2010) present a better understanding of the general preferences of software engineers in software life cycle phases and map these phases to the Myers-Briggs Type Indicator (MBTI) dimensions taking into consideration desirable soft skills that appear in job ads. As far as these studies are concerned, not only performance and task choices are affected by personality type, but also by other factors, such as motivation and the surrounding environment.

A wide variety of psychological instruments are used for career counseling and behavior prediction. In understanding the influence of personality on software development tasks there exists a wide variety of personality frameworks (e.g., Five-Factor theory, Keirsey Temperament Sorter, etc.). The MBTI (Myers, Mccauley, Quenk, Hammer, & Manual, 1998) is one of the most popular tools used in workplaces to analyze personality types. According to the MBTI, a person is measured across four dimensions by his/her preferences: energizing, attending, deciding, and living.

Within each dimension, there are two opposite poles: Extroversion (E) – Introversion (I), Sensing (S) – Intuition (N), Feeling (F) – Thinking (T), and Perceiving (P) – Judging (J). Sixteen distinct personality types are defined on the basis of combining these preferences; each type is denoted by four letters. These distinctions have an influence on career choice because people tend to choose occupations that are related to their personality type.

Within the first dimension, Extroverts get their energy from interactions with people, are outgoing, and prefer to work with other people, whereas Introverts get their energy internally and prefer to work alone. Secondly, the S–N dimension is related to the way in which people acquire information. In particular, sensing people receive information from their five senses and are attuned to the practical, hands-on, common-sense approach to information; intuitive individuals are more focused on complex interactions, theoretical implications, and new possibilities. The third dimension, T–F, is concerned with how people make decisions. Specifically, thinkers prefer to analyze logical/objective data. In contrast, feelers respond to situations depending on their feelings about that situation and often want work that provides services to people. Finally, the Judging type prefers work that has a need for order, whereas the Perceiving type prefers tasks that require adapting to changing situations.

The MBTI has its critics (Petinger, 1993) who point out shortcomings with its statistical structure and other limitations (Boyle, 1995). We should be cautious about its possible misuse in organizational and occupational settings. However, MBTI continues to be the most popular instrument used in profiling the personality types of software engineers (Capretz, 2014).

Myers et al. (1998) assert that an individual's interest in jobs is mainly determined by the S–N and T–F dimensions. These pairs are responsible for the cognitive scales that influence the extent to which people feel attracted to and are satisfied with their career choices: STs prefer activities that require the use of established knowledge and are observant and detail-orientated, they are reluctant to try new innovative solutions; NTs are creative and, consequently, enjoy symbolic abstract relations and seek to find patterns rather than dealing with details. Additionally, they like to create new knowledge rather than applying or improving existing techniques. NTs are more creative than STs because Ns see possibilities beyond the given facts and look for patterns and relationships. Thus, when NTs join both theoretical mindset with their tendency to extrapolate beyond the details, they can identify new principles. The extroversion–introversion and judgment–perception dimensions determine individuals' personal attitudes.

Most studies concerning the MBTI distribution among students and engineering professionals demonstrate that ISTJ, INTP, and ESTJ are over-represented personality types, whereas ENFJ and

**Table 1**

MBTI type distribution among software engineers, system analysts and programmers (Schaubhut & Thompson, 2008).

ISTJ	ISFJ	INFJ	INTJ	E	I
se 17.3%	se 3.6%	se 2.2%	se 9.0%	se 42.8%	se 57.2%
sa 17.7%	sa 4.8%	sa 2.0%	sa 6.7%	sa 48.7%	sa 51.3%
p 19.4%	p 5.0%	p 2.6%	p 7.6%	p 38.5%	p 61.5%
ISTP	ISFP	INFP	INTP	S	N
se 8.1%	se 1.6%	se 3.9%	se 11.5%	se 52.0%	se 48.0%
sa 5.7%	sa 3.0%	sa 4.3%	sa 7.1%	sa 57.9%	sa 42.1%
p 9.1%	p 3.3%	p 5.4%	p 9.1%	p 58.3%	p 41.7%
ESTP	ESFP	ENFP	ENTP	T	F
se 4.7%	se 2.0%	se 3.8%	se 9.7%	se 78.9%	se 21.1%
sa 5.6%	sa 2.3%	sa 4.8%	sa 7.1%	sa 71.9%	sa 28.1%
p 5.0%	p 2.1%	p 4.4%	p 5.4%	p 71.4%	p 28.6%
ESTJ	ESFJ	ENFJ	ENTJ	J	P
se 12.7%	se 2.1%	se 2.0%	se 6.0%	se 54.8%	se 45.2%
sa 14.1%	sa 4.7%	sa 2.2%	sa 7.9%	sa 60.1%	sa 39.9%
p 9.9%	p 4.5%	p 1.3%	p 5.9%	p 56.2%	p 43.8%

Note 1: se means "software engineers," sa means "system analysts," and p mean "programmers".

Note 2: Sample of: 1326 subjects for se, 2493 subjects for sa, 1719 subjects for p.

INFJ types are underrepresented (Capretz, 2003). The personality distribution of application software developers can be seen in Table 1, which presents data taken from the book *MBTI Type Tables for Occupations* (Schaubhut & Thompson, 2008) for software engineers, system analysts, and programmers.

Cruz, Silva, and Capretz (2015) present a comprehensive systematic literature review of personality in software engineering. Other researchers have studied characteristics and traits in personality types for certain roles in software engineering (Clark, Walz, & Wynekoop, 2003; Evans & Simkin, 1989; Teague, 1998), and (Varona, Capretz, Pinero, & Raza, 2012). System analysts and programmers are among the most explored roles in these studies. Nevertheless, we do not find a straight relationship among their preferences and personality types. Since there is a logical relationship between the task preferences and proportions of personality types in software engineering, our study seeks to provide evidence of relationships between personality types, task preferences, and roles in order to obtain conclusive results for systems analysts, designers, programmers, testers, and maintainers. In particular, an empirically validated study of actual software developers is used to investigate these relationships.

## 2. Research motivation

As a discipline, software engineering consists of many roles and responsibilities from the perspective of a project team.

The definition of roles significantly depends on the project characteristics and the development process. While there are a wide range of roles in software development, this investigation focuses on some well-known defined roles: analyst, designer, programmer, tester, and maintainer (Capretz & Capretz, 1996).

The MBTI type distributions of some software professionals are presented in Table 1 (Schaubhut & Thompson, 2008).

Software engineers review, design, create, and test software for basic computer applications, including operating systems, compilers, and computer networks. They expand existing or launch new general software applications and may also examine or design databases. They establish operational specifications and study requirements using computer science, engineering, or mathematics. Software engineering can be seen as an umbrella career for these specialized tasks (Capretz, 2002).

System analysts must be able to understand system essentials, and to create an abstract model of the application in which user

needs are met. They also require a significant amount of interaction with users/clients to understand the scope and constraints of the software solution, and to propose procedures to improve software capabilities and workflow.

Software designers are responsible for modeling the proposed software system as well as evaluating and validating the designed model to ensure its quality before generating the software code. According to [Shatnawi and Alzu'bi \(2011\)](#), unlike other engineering products software systems are not tangible. Due to their complexity, software quality assurance is essential to achieve software of the highest quality. Nevertheless, designers identify appropriate components by experimenting with a variety of schemes in order to discover the optimal way of refining the application for the client. Ideally, software designers should have sufficient skills for producing a model that supports the programmer activities and provides an accurate translation of the customer requirements into a finished software product that is stable and operates on real computers.

Programmers transfer project specifications into detailed algorithms for coding into a computer language. They create and write computer programs to perform calculations and store, locate, or retrieve data or other information. The process of programming requires the identification of control structures, relevant variables, and data structures as well as a detailed understanding of the syntax and specifics of a programming language. In particular, programmers need to attend to details and exhibit an open, logical, analytical style. MBTI type distribution among computer programmers is also shown in [Table 1 \(Schaubhut & Thompson, 2008\)](#).

The task of testing is focused on identifying faults; there are many ways to make testing efforts more efficient and effective. Testing strategies are neither random nor haphazard; rather, they should be approached in a methodical and systematic manner. The process of debugging errors can be a frustrating and emotionally challenging activity that can lead software engineers to restructure their thinking and decisions. [Takamatsu, Sato, Oyama, and Kurihara \(2014\)](#) realize that software testing is costly. They recommend automated testing and consider it to be an effective way to reduce the burden of testing. However, testing requires persistence, especially because of the need to choose from an enormous range of possibilities and to keep a high level of attention to detail.

Software maintenance involves keeping applications operational, reacting quickly to problems in order to restore service, meet or exceed the agreed level of service, and transmit confidence to the user community. Specifically, maintenance personnel should ensure that users believe in their support team as dedicated and competent individuals. Unfortunately, we could not find data providing MBTI type distribution among designers, testers, or maintainers to compare with results observed in this study.

### 3. Research methodology

This paper aims to identify software engineers' preferred roles on the basis of their distinct personality types. We have used an empirical method to relate personality types and preferences with software engineering roles. This study constitutes an important link between personality types and tasks to which engineers are inclined to. In order to understand whether individual personality affects software engineers' choices, specific evidence from the field are required. Consequently, 100 software developers from the University of Informatics Sciences (UCI) in Havana, Cuba, were surveyed and interviewed. For the sake of clarification, UCI is an atypical university, where both students and professors are directly

engaged in software projects. Their average experience as software developers is five years. So it is correct to consider UCI students/professors as software practitioners, as they develop software products that are exported to several countries in Latin America and Europe.

In this study, 100 Cuban software developers were surveyed, including students in senior Informatics Sciences Engineering courses and professors at UCI. There were seven senior students and 93 professors among the subjects; both students and professors were directly involved in software development. They were invited to take part in this study voluntarily. The sample contained 47% males and 53% females. It is the country's policy that all Cuban universities have a balanced ratio between males and females to avoid gender bias. The student age range was between 22 and 23 years of age, while professors ranged from 23 to 27 years of age. The MBTI instrument, Form M, Spanish language version, was used to identify their personality types. Although this instrument is self-assessed, an MBTI certified practitioner was present and processed the data.

As far as the individual roles were concerned, prior to collecting this information about role preferences, the subjects were given the role definitions. The seven students in the sample were working as software engineers in international projects at the moment of the study. All students had at least two years of experience as software practitioners. Professors in the sample had five to eight years of experience in software development. Among the subjects 30% were analysts, 11% designers, 28% programmers, 12% testers, 12% maintainers, and 7% had been project leaders.

After the MBTI was applied, participants were queried about their preferences for the various roles: analyst, designer, programmer, tester, and maintainer. Specifically, participants were asked to state if they preferred, did not prefer, or were neutral about these roles. In particular, participants were instructed to focus only on their overall preferences and to disregard any particular software development tasks that they might have been performing at the moment the research was conducted.

### 4. Results and discussions

In the MBTI distribution of the 100 software developers ([Table 2](#)), specific poles dominate within each dimension. Specifically, the number of extroverts (63%) is almost double that of introverts (37%). Similarly, sensing individuals (72%) predominate over intuitive people (28%), thinking types (75%) are three times as common as feeling types (25%), and judging types (61%) outnumber perceiving individuals (39%).

With reference to the individual poles, it is immediately clear that 'Ts' and 'Ss' (75% and 72% respectively) are over-represented in the sample, whereas 'Fs' and 'Ns' (25% and 28% each) are under-represented.

**Table 2**  
Cuban software engineers representation among the 16 MBTI types ( $n = 100$ ).

ISTJ	ISFJ	INFJ	INTJ	E	I
10%	7%	1%	6%	63%	37%
ISTP	ISFP	INFP	INTP	S	N
5%	2%	1%	5%	72%	28%
ESTP	ESFP	ENFP	ENTP	T	F
15%	6%	3%	2%	75%	25%
ESTJ	ESFJ	ENFJ	ENTJ	J	P
25%	2%	3%	7%	61%	39%

In terms of personality types, ESTJ is the most popular type (25%), while ESTP (15%) and ISTJ (10%) are also relatively common. In combination, these three personality types represent half of the sample. In contrast, the least represented types are INFJ and INFP, each indicating 1% of the subjects, while ISFP, ENTP, and ESFJ only accounted for 2% of the total sample, as shown in Table 2.

For statistical analysis, a Chi-Square (non-parametric) test was applied. The distribution of the observed results (Table 2) was not found to be highly significant ( $\alpha < 0.001$ ), as compared to the expected distribution data in (Table 1). The data for role preference appear in Table 3. Each row depicts the personality type, the number of individuals comprising each category, and the number of individuals preferring each role. Among the roles: analysts, designers, and programmers are the most popular, with analyst being the most preferred of all. In contrast, testers and maintainers were the least popular. The definitions of each role were explained to the subjects before they were asked about their preferences.

#### 4.1. Analyst

In order to check the statistical difference between the surveyed sample and the MBTI type distribution the number of system analysts in Table 1 was taken as the expected distribution of personality types for analyst. A Statistical Binomial Test was performed for each 16 personality type presented in Table 3, specifically those denoting preference for the role "Analyst." From these tests it can be inferred that: 21 (84%) of the individuals were from the group of 25 ESTJ subjects, 14 (93%) individuals from 15 ESTP subjects, eight (80%) out of 10 ISTJ subjects, six (86%) out of seven ISFJ subjects, four (67%) out of six INTJs, six (100%) out of six ESFPs, and four (80%) out of five INTPs preferred the role "Analyst."

As expected, values of neutral and non-preference are not available, therefore generalized inferences cannot be made. Although descriptive analysis framed on the sample shows: 43% ENTJs, 16% ESTJs, 10% ISTJs, and 7% ESTPs do not prefer the role "Analyst." In contrast, 14% ENTJs, 14% ISFJs, and 10% ISTJs express themselves as neutral about this role.

#### 4.2. Designer

As there were no previous studies exposing what can be taken as expected values for this role, we just discuss the preferences shown in Table 3. Several personality types within the surveyed sample prefer designing, including 100% INTPs, 83% INTJs, 80%

ISTJs and ISTPs, 72% ESTJs, 71% ENTJs, 60% ESTPs, and 50% ESFPs. On the other hand, 33% ESFPs, 27% ESTPs, 29% ISFJs, 24% ESTJs, 20% ISTJs, 17% INTJs, and 14% ENTJs dislike designing. Lastly, 43% ISFJs, 17% ESFPs, 14% ENTJs, 10% ISTJs, 13% ESTPs, and 4% ESTJs state their neutrality about the role "Designer."

#### 4.3. Programmer

The benchmark distribution of personality types for programmers has been taken from Table 1. A Statistical Binomial Test was carried out for each of the 16 personality types depicted in Table 3, specifically those denoting preference for the role "Programmer". From these tests we inferred, with high significance, that: 19 (76%) individuals out of 25 ESTJ subjects, 10 (67%) out of 15 ESTP subjects, five (71%) out of seven ISFJ or ENTJ subjects, five (83%) out of six ISFJ subjects, four (67%) out of six ESFPs subjects, and five out of five ISTPs preferred programming.

Expected values of neutral and non-preference regarding programmers are not available; thus general inferences cannot be made in this case. Descriptive analysis framed on the sample indicates that: 60% ISTJs, 33% both ESTPs and ESFPs, 16% INTJs, 16% ESTJs, and 14% both ISFJs and ENTJs do not like programming. In contrast, 20% INTPs, 14% ISFJs, and ENTJs, 10% ISTJs, and 8% ESTJs declare their neutrality about programming.

#### 4.4. Tester

Regarding tests activities the results are as follows: 30% ISTJs preferred to do testing. Conversely, 83% INTJs, 80% ISTPs, 71% ENTJs, 67% ESFPs and ESTPs, and 57% ISFJs do not like testing. Personality types that reported neutral feelings for this role included 30% ISTJs, 29% ENTJs, 20% ISTPs and INTPs, 17% ESFPs, 14% ISFJs, 13% ESTPs, and 4% ESTJs.

#### 4.5. Maintainer

Finally, the following personality types preferred maintenance: 40% ISTPs, 29% ISFJs and ENTJs, 28% ESTJs, 20% ISTJs, 17% ESFPs, and 13% ESTPs. On the other hand, 83% ESFPs, 67% INTJs, 57% ISFJs, 53% ESTPs, 48% ESTJs, 43% ENTJs, 30% ISTJs, and 20% INTPs and ISTJs did not like maintenance. Lastly, 60% INTPs, 50% ISTJs, 40% ISTPs, 33% ESTPs, 33% INTJs, 29% ENTJs, 24% ESTJs, and 14% ISFJs expressed their neutrality about this role.

**Table 3**  
Summary of MBTI role preferences.

Types	Portion	% of total	Analyst			Designer			Programmer			Tester			Maintainer		
			1	0	-1	1	0	-1	1	0	-1	1	0	-1	1	0	-1
ISTJ	10	10	8	1	1	8	1	1	3	1	6	3	3	4	2	5	3
ISFJ	7	7	6	1	0	2	3	2	5	1	1	2	1	4	2	1	4
INFJ	1	1	1	0	0	1	0	0	1	0	0	0	0	1	0	0	1
INTJ	6	6	4	0	2	5	0	1	5	0	1	1	0	5	0	2	4
ISTP	5	5	2	1	2	4	0	1	5	0	0	0	1	4	2	2	1
ISFP	2	2	2	0	0	0	0	2	2	0	0	0	1	1	1	1	0
INFP	1	1	0	0	1	1	0	0	1	0	0	0	1	0	0	1	0
INTP	5	5	4	0	1	5	0	0	2	1	2	1	1	3	1	3	1
ESTP	15	15	14	0	1	9	2	4	10	0	5	3	2	10	2	5	8
ESFP	6	6	6	0	0	3	1	2	4	0	2	1	1	4	1	0	5
ENFP	3	3	2	0	1	1	1	1	3	0	0	0	0	3	0	0	3
ENTP	2	2	2	0	0	2	0	0	0	0	2	2	0	0	0	0	2
ESTJ	25	25	21	0	4	18	1	6	19	2	4	5	1	19	7	6	12
ESFJ	2	2	2	0	0	1	0	1	1	0	1	1	0	1	0	0	2
ENFJ	3	3	3	0	0	2	1	0	0	1	2	1	0	2	1	0	2
ENTJ	7	7	3	1	3	5	1	1	5	1	1	0	2	5	2	2	3

Note: 1 means "prefer," 0 means "neutral," and -1 means "do not prefer."

Once preferences have been related to each personality type, the following conclusions can be made about individual indicators, as displayed in Table 4.

High preference percentages of extroverts and introverts related to the analyst role can be noticed from Fig. 1. 84% out of 63 extroverts prefer to be analysts, 70% of introverts want to be designers, whereas 65% of introverts prefer doing programming. This is well in line with the suggestions by Capretz and Ahmed (2010) and Ahmed, Capretz, and Campbell (2012), who recommend that extroverts are more suitable to be appointed as system analysts. Their recommendations of introverts for the jobs of software designers and/or programmers are also supported by the outcome of this study.

In contrast to intuitive respondents, sensing individuals report higher preference for the design and/or maintenance activities respectively (63% over 39% regarding design and 24% over 14% regarding maintenance) as presented in Fig. 2.

While mapping skills to MBTI personality types, Capretz and Ahmed (2010) found intuitive individuals more likely to thrive in design and programming, which is also supported by this study's results. However, they believe sensing people are more suitable for testing and maintenance jobs. In contrast, only 21% of sensing individuals showed their preference for testing and 24% of them wanted to do software maintenance. Furthermore, a high percentage of both sensing and intuitive individuals opted to work as analysts. This came as a surprise, if we compare it with the observation of other researchers (Capretz & Ahmed, 2010; Evans & Simkin, 1989).

Capretz and Ahmed (2010) assert that Fs are a preferable choice for systems analyst's job; whereas Ts are more likely to prefer programming. In this study, however, there is also a remarkable preference for designing (75%) among thinking individuals as compared to only 44% among feeling individuals. It is notable that a substantial number of feeling individuals expressed preferences for the "Analyst" role. The interests in "Tester" and "Maintainer" roles remain almost equal for both dimensions, as shown in Fig. 3.

In the judging and perceiving dimensions, there were not much difference with regards to role preference percentages. However, the high preference for the role of analysts by both judging (79%) and perceiving (82%) is remarkable. This is followed by their preference for the roles of designers and programmers, as presented in Fig. 4. This is surprisingly in conflict with the theoretical view, according to which sensing and judging individuals are more suitable for software testing as observed by Kanij, Merkel, and Grundy (2013).

Yet another way of combining the preferences is by the temperaments: SJ, SP, NF, and NT. The temperaments lens are very useful for career counselling.

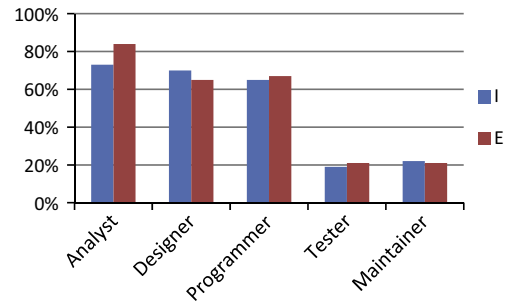


Fig. 1. E-I preference indicator.

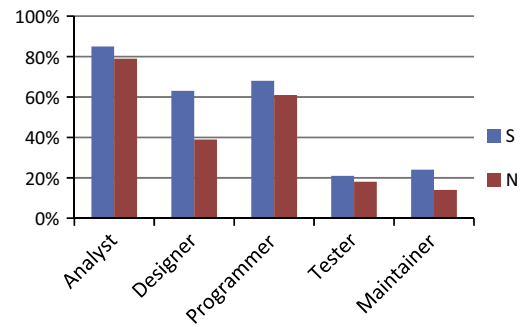


Fig. 2. S-N preference indicator.

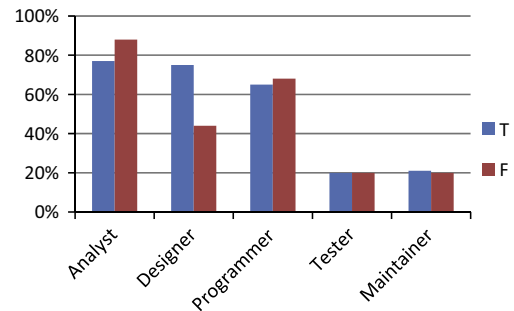


Fig. 3. T-F preference indicator.

The predominant temperament, as can be seen in Table 4, within the sample is SJ, with 45% of individuals. In contrast, the least popular temperament is NF, with only 8% of people having these traits. The temperament results for SJ and NF are similar to those in previous studies (Miller & Zhichao, 2004; Varona et al., 2012).

Table 4  
Individual indicator and temperament distribution.

Types	Quantity	Analyst	Designer	Programmer	Tester	Maintainer					
I	37	73%	27	70%	26	65%	24	19%	7	22%	8
E	63	84%	53	65%	41	67%	42	21%	13	21%	13
S	72	85%	61	63%	45	68%	49	21%	15	24%	17
N	28	79%	22	39%	11	61%	17	18%	5	14%	4
T	75	77%	58	75%	56	65%	49	20%	15	21%	16
F	25	88%	22	44%	11	68%	17	20%	5	20%	5
J	61	79%	48	69%	42	64%	39	21%	13	23%	14
P	39	82%	32	64%	25	69%	27	18%	7	18%	7
<i>Temperament</i>											
SP	27	85%	23	59%	16	74%	20	11%	3	22%	6
SJ	45	84%	37	66%	29	64%	28	25%	11	25%	11
NT	20	65%	13	85%	17	60%	12	20%	4	15%	3
NF	8	75%	6	63%	5	63%	5	13%	1	13%	1

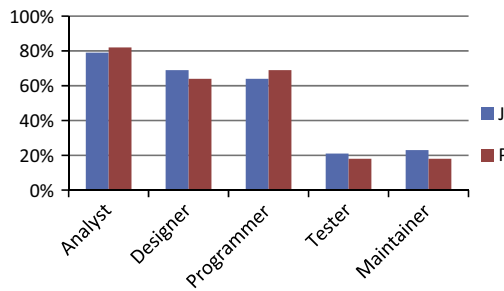


Fig. 4. J-P preference indicator.

However, SP was the second most represented temperament type, rather than NT, which was the opposite case in those two studies.

The temperaments that are more assertive about their preferences are NF and SP. Whereas, SJ and NT are certain about the role they want to assume during software development.

The most highly-valued preference for the SP temperament was the “Analyst” role, which was followed by the “Programmer” role. Furthermore, the NTs have a clear preference for the “Designer” role, while the SJ and NF temperaments preferred the role “Analyst.” Lastly, the NFs preferred the roles “Designer” and “Programmer.”

## 5. Conclusion

An empirical study was conducted to map some opposing psychological traits, such as extroversion-introversion, sensing-intuition, thinking-feeling, and judging-perceiving to the main tasks of a software life cycle. This work addresses the fundamental issue of the impact of human factors in software development. This research provides evidence of relationships between software engineers’ MBTI types and role preferences. Although the MBTI does not predict success in a particular career, it identifies individual preferences for specific occupations. We are not trying to predict performance based on personality types, we are attempting to study types preference for certain software roles. It is safe to assume that individuals are more motivated to finish tasks that they prefer or enjoy doing.

The results indicated by the personality types INFJ, ISFP, INFP, ESFP, ENFP, ENTJ, ESFJ, and ENFJ are not relevant because of the sample size. In each of these cases, our data values are in the range of one and three individuals, thus it lacks statistical significance.

However, distinctive patterns are evident in the relationship between personality types and preferences for software engineering roles. This study revealed particular role preferences for some personality types. It has been inferred, with high significance, that 84% ESTJs, 93% ESTPs, 80% ISTJs, 86% ISFJs, 67% INTJs, 100% ESFPs, and 80% INTPs indicate analysis in their preferences. While 76% ESTJs, 67% ESTPs, 71% ISFJs and ENTJs, 83% ISFJs, 67% ESFPs, and 100% ISTPs prefer programming. It can be concluded that assigning a person with specific psychological characteristics to tasks in the software life cycle best suited for their traits increases the chances of a successful outcome for the project.

Overall, the roles “Analyst”, “Designer”, and “Programmer” were the most popular ones among all personality types, whereas “Tester” and “Maintainer” roles were identified as the least sought after ones. The margins of neutral positions can be decreased through further inquiries about software engineer preferences.

Finally, it is important to point out that the study of the relationships between personality types and software roles sheds more light as to how software development is affected by individual type

indicators and how important it is to match the right people to roles in software engineering.

## References

- Acuna, S. T., & Juristo, N. (2004). Assigning people to roles in software projects. *Software: Practice and Experience*, 34, 675–696.
- Acuna, S. T., Juristo, N., & Moreno, A. M. (2006). Emphasizing human capabilities in software development. *IEEE Software*, 23(2), 94–101.
- Ahmed, F., Capretz, L. F., & Campbell, P. (2012). Evaluating the demand for soft skills in software development. *IEEE IT Professional*, 14(1), 44–49. <http://dx.doi.org/10.1109/MITP.2012.7>.
- Boyle, G. J. (1995). Myers-Briggs Type Indicator (MBTI): some psychometrics limitations. *Australian Psychologist*, 30(1), 71–74.
- Capretz, L. F. (2002). Are software engineers really engineers? *World Transactions on Engineering and Technology Education*, 1(2), 233–235.
- Capretz, L. F. (2003). Personality types in software engineering. *International Journal of Human Computer Studies*, 58(2), 207–214. [http://dx.doi.org/10.1016/s1071-5819\(02\)00137-4](http://dx.doi.org/10.1016/s1071-5819(02)00137-4).
- Capretz, L. F. (2014). Bringing the human factor to software engineering. *IEEE Software*, 31(2), 102–104. <http://dx.doi.org/10.1109/MS.2014.30>, March–April.
- Capretz, L. F., & Ahmed, F. (2010). Making sense of software development and personality types. *IEEE IT Professional*, 12(1), 6–13. <http://dx.doi.org/10.1109/MITP.2010.33>. IEEE Press.
- Capretz, L. F., & Capretz, M. A. M. (1996). *Object-oriented software: design and maintenance*. 981-02-2731-0. Singapore: World Scientific.
- Choi, K. S., Deek, F. P., & Im, I. (2008). Exploring the underlying aspects of pair programming: The impact of personality. *Information and Software Technology*, 50(11), 1114–1126.
- Clark, J. G., Walz, D. B., & Wynkoop, J. L. (2003). Identifying exceptional application software developers: A comparison of students and professionals. *Communications of the Association for Information Systems*, 11(8), 137–154.
- Cruz, S., Silva, F. Q. B., & Capretz, L. F. (2015). Forty years of research on personality in software engineering: A mapping study. *Computers in Human Behavior*, 46(1), 94–113. <http://dx.doi.org/10.1016/j.chb.2014.12.008>. Elsevier.
- Curtis, B., Hefley, W. E., & Miller, S. A. (2001). *People Capability Model (P-CMM)*. Technical Report no. CMU/SEI-2001-MM-001, Software Engineering Institute, Carnegie Mellon University.
- Da Cunha, A. D., & Greathead, D. (2007). Does personality matter?: An analysis of code-review ability. *Communications of ACM*, 50(5), 109–112.
- DeMarco, T., & Lister, T. (1999). *Peopleware: Productive projects and teams* (2nd ed.). Dorset House.
- Evans, G. E., & Simkin, M. G. (1989). What best predicts computer proficiency? *Communications of ACM*, 32(11), 1322–1327.
- Feldt, R., Angelis, L., Torkar, R., & Samuelsson, M. (2010). Links between the personalities, views and attitudes of software engineers. *Information and Software Technology*, 52(6), 611–624.
- Gallivan, M. J. (2004). Examining IT professionals’ adaptation to technological change: The influence of gender and personal attributes. *SIGMIS Database*, 35(3), 28–49.
- Hall, T., Sharp, H., Beechman, S., Badoo, N., & Robinson, H. (2008). What do we know about motivation? *IEEE Software*, 25(4), 92–94.
- Kanij, T., Merkel, T., & Grundy, J. (2013). An empirical study of the effects of personality on software testing. In *26th IEEE international conference on software engineering education and training (CSE&T)*, May, San Francisco (pp. 239–248).
- Miller, J., & Zhichao, Y. (2004). A cognitive-based mechanism for constructing software inspection teams. *IEEE Transactions on Software Engineering*, 30(11), 811–825.
- Moore, J. E. (2000). One road to turnover: An examination of work exhaustion in technology professionals. *MIS Quarterly*, 24(1), 141–168.
- Myers, I. B., Mccauley, M. H., Quenk, N. L., Hammer, A. L., & Manual, M. B. T. I. (1998). *A guide to the development and use of the Myers-Briggs type indicator*. Palo Alto, CA: Consulting Psychologists Press.
- Petinger, D. (1993). Measuring the MBTI... and coming up short. *Journal of Career Planning and Employment*, 54(1), 48–52.
- Ritcher, K., & Dumke, R. (2015). *Modeling, evaluating, and predicting IT human resources performance*. CRC Press.
- Schaubhut, N. A., & Thompson, R. C. (2008). *MBTI type tables for occupations*. Mountain View, CA: CPP.
- Shatnawi, R., & Alzu’bi, A. (2011). A verification of the correspondence between design and implementation quality attributes using a hierarchical quality model. *IAENG International Journal of Computer Science*, 38(3), 225–233.
- Takamatsu, H., Sato, H., Oyama, S., & Kurihara, M. (2014). Automated test generation for object-oriented programs with multiple targets. *IAENG International Journal of Computer Science*, 41(3), 198–203.
- Teague, J. (1998). Personality type, career preference and implications for computer science recruitment and teaching. In *3rd Australasian conference on computer science education* (pp. 155–163). New York, NY: ACM.
- van Solingen, R., Berghout, E., Kusters, R., & Trienekens, J. (2000). From process improvement to people improvement: Enabling learning in software development. *Information and Software Technology*, 42, 965–971.
- Varona, D., Capretz, L. F., Pinero, Y., & Raza, A. (2012). Evolution of software engineers’ personality profile. *ACM SIGSOFT Software Engineering Notes*, 37(1), 1–5. <http://dx.doi.org/10.1145/2088883.2088901>.