Computers in Human Behavior 52 (2015) 373–378

Contents lists available at ScienceDirect

Computers in Human Behavior

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

ELSEVIER

Influence of personality types in software tasks choices

Luiz Fernando Capretz [a,*], Daniel Varona [b], Arif Raza [a]

[a] Department of Electrical & Computer Engineering, Western University, London, Ontario N6A5B9, Canada
[b] 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 (L.F. Capretz), dvarona@uci.cu (D. Varona), araza22@uwo.ca (A. Raza).

http://dx.doi.org/10.1016/j.chb.2015.05.050
0747-5632/© 2015 Elsevier Ltd. All rights reserved.

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, Mccaulley, 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 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.

Table 1
MBTI type distribution among software engineers, system analysts and programmers (Schaubhut & Thompson, 2008).

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

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

Download English Version:

https://daneshyari.com/en/article/6838035

Download Persian Version:

https://daneshyari.com/article/6838035

Daneshyari.com