

# KNOWING SOFTWARE ENGINEER'S PERSONALITY TO IMPROVE SOFTWARE DEVELOPMENT

Luis G. Martínez, Juan R. Castro, Guillermo Licea, Antonio Rodríguez-Díaz and Carlos F. Alvarez  
*University of Baja California, Calzada Tecnológico 14418, Tijuana, 22300, Mexico*

**Keywords:** Software Engineering, Decision Making, Roles, Psychometrics.

**Abstract:** Nowadays organizations work to improve their software development process, with a purpose to reduce costs, improve quality and increase planning reliability. That is why decision making pertaining to role assignment in software engineering developing projects is one of the most important factors that affect the software development process in organizations. We should not only consider individual's abilities and capabilities for better team performance but also consider knowing their personality traits to carry out the most suitable role in an effective working team. Through compilation of studies with RAMSET (Role Assignment Methodology for Software Engineering Teams) methodology some personalities and typologies have been identified to perform certain type of roles, thus helping us build a better, cohesive and less conflictive team. Our methodology based on personality has revealed appropriate and adequate personality patterns for assignment of best advisable performing roles in software development, according not only to capabilities of people and role demands but also taking into consideration personality traits, thus showing that knowing software engineer's personality can improve software development process.

## 1 INTRODUCTION

Psychological assessment instruments have been used for over sixty years and have reached a mature stage for predicting career selecting and behavior. Diverse personality tests like Jung, Myers-Briggs, Keirsey, Big Five, among other projective tests, can be used to know the sociopsychological characteristics and personality of individuals besides abilities for job placement and hiring, especially in assigning individuals to form a working team (Leaetta and Frederick, 2000) (Rodriguez, 2004) (Rothstein and Goffin, 2006).

Effective use of psychometric instruments can add value to an organization. When used in selection and structured interview process, they enable companies to select more accurately those people who will perform best in a role, although the instrument does not predict success in a role, it does, however, identify preferences for that role (Pittenger, 1993).

Decision making pertaining to role assignment in software development process is one of the most important factors an organization has to consider. Use of effective decision making software for everyday planning and task management is on the rise in modern organizations. Web pages and

Decision Groups commonly offer services for decision making and personnel selection with diverse methodologies applying psychometrics.

This paper presents a personality trait assessment derived from implementing different personality tests, although software engineering attracts people of all psychological types certain traits are clearly more represented than others discovering personality patterns in software engineering developers.

The paper is organized as follows: section 2 is a brief background of personality types and traits related with software engineer's role performance; section 3 refers to the methodology and personality tests used in our case studies; section 4 presents data results of these tests obtaining personality patterns for software engineering roles, concluding in section 5 with observations obtained from experience.

## 2 BACKGROUND

Actual studies have been focusing on integration and building of Software Engineering Teams, addressing different factors that influence their performance like team abilities, team administration, efficiency, development methods (Guinan et al., 1998), diversity (Smith et al., 2001), size (Biffel and Halling,

2003), genre (Teague, 2002), personality (Gorla and Lam, 2004) and roles (Dubinsky and Hazzan, 2006).

DeMarco and Lister (1999) make many observations about managing software development implying that the major problems in development are not technical but social. Calitz et al. (1997) outlines some of the increased demands on IT personnel. He identifies a shift from a traditional IT environment (programming from specifications, little end user contact, specified delivery times) to an environment of growing competition, fast-changing technologies, and more sophisticated and demanding end-users. This shift is reflected in the personality traits of successful IT staff that are investigative, realistic, enterprising and, more recently, social. Some of these social skills have been identified by Goldstein (1988), agreeing with supervisors and employees in four factors: communication skill, job attitude, business knowledge, and technical skill.

Five broad domains or dimensions of personality have been scientifically discovered to define human personality, in contemporary psychology they have been called the "Big Five" factors of personality. The initial model was reported by Tupes and Cristal (1992), later Goldberg (1993) extended it to the highest level of organization, and it is known as the "Five Factor Model" or FFM (Costa and McCrae, 1992), is a purely descriptive model of personality. Traits Openness, Conscientiousness, Extraversion, Agreeableness, and Neuroticism (OCEAN), are positively related to teamwork and must be considered to improve team's performance. Bernstein et al. (2008) used the Big Five personality framework to predict the relationship between team member personality and team effectiveness in teamwork and decision making areas.

Neuroticism trait is sometimes referred as Emotional Stability and Openness as Intellect. Openness (O) is a disposition to be imaginative, inventive, curious, unconventional and autonomous; it has an appreciation for art, emotion, adventure and a variety of experience. Conscientiousness (C) comprises of two related facets achievement and dependability, it has a tendency to show self-discipline, to be efficient, organized and aim for achievement, plans rather than behave spontaneously. Extraversion (E) represents a tendency to be sociable, outgoing and assertive, with energy, passion and excitement. Agreeableness (A) is a tendency to be trusting, friendly, compassionate, cooperative, compliant, caring and gentle. Neuroticism (N) represents a tendency to exhibit poor emotional adjustment and experience negative or unpleasant emotions easily, such as anxiety, insecurity, depression and hostility.

Also Jung's different personality dimensions are associated to different career and jobs corresponding to the individual personality type (United States Department of the Interior). Jobs associated with engineering and software are: ISTJ type is related with engineer, programmer and chief information officer; INTJ type includes programmer and informational graphics; INTP type relates with software designer, system analyst, computer programmer, data base manager; ISTP type includes computer repair person, programmer, software developer; ESTP takes in entrepreneur, technical trainer, analyst; ENTJ an administrator and program designer.

Gorla and Lam (2004) made a personality type analysis to describe the most effective personality attributes for software development team roles; she found that a *team leader* with (I) Intuitive characteristics outperformed a (S) Sensing leader as intuitive persons are picture oriented and have an innovative ability to assess alternate solutions. Also a (F) Feeling team leader outperformed a (T) Thinking leader, because a feeling person is people oriented and makes decisions based on how they affect individuals, making him more effective. Capretz (2003) implies that software engineers and psychological types are clearly related, as suggested by this study. More specifically their work suggests that software engineers are most likely to be STs, TJs or NTs.

Furthermore projective tests have also been used in management (Soley and Smith, 2008) to assess achievement motivation and other drives, in sociology to assess the adoption of innovations, and in anthropology to study cultural meaning. The application of responses is different in these disciplines than in psychology, because the responses of multiple respondents are grouped together for analysis by the organization's research commission, rather than interpreting response meanings given by a single subject.

A projective test is a personality test designed to let a person respond to ambiguous stimuli, presumably revealing hidden emotions and internal conflicts. Because the stimulus is ambiguous, the patient must impose his or her own structure. In doing so, thoughts, feelings, and themes, some of which are unconscious, are projected into the material. Projective tests can provide an interesting source of information regarding the person's unique view of the world, and they can be a useful supplement to information obtained with other assessment tools.

The Tree Test is a projective test initiated by Emil Jucker and Koch (1986) contributes with formulations and interpretations related in

graphology as quality strokes, zone and sheet placement. Every graphical product instates the psychic life of an individual; trees are symbolically associated with man's inner person and can describe their growth and emotional stability. The tree's crown represents the subject's fantasies, mental activities, his thoughts, spirituality and reality conception, includes foliage and branches. The trunk is related with affections and relationships. The unconscious world or instincts are symbolized in the root as it relates the material, physical, earthly life, sexuality, and reality criteria (Griffiths 1988). Psychodiagnostics analyzes specific characteristics from the drawing; in Root we can select sketching type and size as it represents the past and reflects a person's dependency. For Trunk we can select form, area, height, sketch intensity and curvature, the trunk depicts the present and reflects person's affectivity. For Foliage we can select form, size and extra features, it symbolizes achievements or goals reached.

A personality Tree Test does not define us a total projection or whole image of the personality, but sheds valuable information of the sketcher. Relevance and merit of applying this test in combination with other objective tests, like Jung's and Big Five can reveal appropriate and adequate personality patterns for assignment of best advisable performing roles to members of working teams in software engineering development, according not only to capabilities of the people and demands of the role but also taking into consideration personality traits.

### 3 METHODOLOGY

Teaching of Software Engineering in the Computer Engineering Program at the University of Baja California, Tijuana Mexico is being conducted with development of real software projects applying RAMSET: a Role Assignment Methodology for Software Engineering Teams based on personality; what is unique about this methodology is a combination of Sociometric techniques, Psychometrics and Role Theory consisting of the next steps:

- a) Survey for abilities and skills.
- b) Implementation of Personality Tests.
- c) Carrying out Personal Interviews.
- d) Implementation of Sociometric Technique.
- e) Assignment of Team Roles.
- f) Follow up of Team Role fulfilment.

RAMSET methodology begins with a student's survey enumerating related courses of software engineering he has taken, to know which programming languages and data base managers he is expert in. The next step is a series of personality tests; they could be Jung, MBTI, Big Five, Keirsey and other projective tests.

Subsequently we make an informal interview to know different aspects of his personality, what he likes to do, how he perceives himself after college, how he develops individually in the real world and with others. After that a sociogram technique is applied to identify affinity for integration of teams. Based on test results and interview information a role is recommended to the instructor so individual members of each team develop a specific team role with all its functions.

Adopted team roles were selected from roles defined by Tomayko (1996): architect, responsible for construction, coordination and supervision of the project; analyst, responsible for finding and following up on resources, requirement analysis and specifications; developer, responsible for implementation and design; and programmers, technical specialist in charge of building code. In addition we considered a document specialist in charge of joining documents and skilled in writing, and a presenter and image specialist responsible for sales, distribution and image design.

Our case study has taken into account Jung and Big Five objective tests, where individuals require selecting options from multiple choice questionnaires equating personality to collective degrees of behaviour into Jung Types or Five Factor types. Also a projective test is used where the test taker is asked to draw a Tree, these drawings provide a measure of self-perceptions and attitudes obtaining information concerning an individual's sensitivity, maturity, flexibility, efficiency, degree of integration and interaction with the environment. Although this test throws subjective information based on the point of view and perception of the evaluator.

### 4 RESULTS

In a period of three years from 2007 through 2009, work of our case studies consisted of implementing RAMSET methodology to obtain personality patterns related with software engineering roles performed in team projects. Data accumulated during this period was 88 Jung test results, 80 Big Five test results, 74 drawing trees. With a weighted mean method applied on frequencies of Jung and

Tree data tests, tables 1 and 2 were obtained. Weighted mean is the number obtained by adding the product of  $\alpha_i$  times the  $i$ th number in a set of  $N$  numbers for  $i = 1, 2, \dots, N$ , where  $\alpha_i$  are numbers (weights) such that  $\alpha_1 + \alpha_2 + \dots + \alpha_N = 1$ .

Table 1: Jung Type Software Engineering Role Weights.

	ANA	ARC	DEV	DOC	TST	PRS
J1*	.513	.524	.113	.47	.385	.233
J2	.12	.272	.101	.171	.189	.384
J3	.267	0	.031	.077	.28	.209
J4	.053	.061	.346	.068	0	0
J5	0	0	.189	.051	.056	.093
J6	.047	.054	.025	.06	.049	0
J7	0	.048	0	.043	.042	.081
J8	0	.041	.019	.034	0	0
J9	0	0	.088	.026	0	0
J10	0	0	.075	0	0	0
J11	0	0	.013	0	0	0

\*J1=ESTJ, J2=ENTJ, J3=ESFJ, J4=ISTP, J5=INTJ, J6=ISTJ, J7=ISFJ, J8=ESFP, J9=ISFP, J10=ESTP, J11=ENTP

Table 2: Tree Test Software Engineering Role Weights.

ROOT'S WEIGHTS						
	ANA*	ARC	DEV	DOC	TST	PRS
R1	0.103	0.182	0.441	0.030	0.067	0.050
R2	0.276	0.636	0.441	0.727	0.333	0.200
R3	0.621	0.182	0.118	0.242	0.600	0.750
TRUNK'S WEIGHTS						
	ANA	ARC	DEV	DOC	TST	PRS
T1	0.174	0.174	0.25	0.091	0.097	0.316
T2	0.652	0.174	0.656	0.455	0.452	0.632
T3	0.174	0.652	0.094	0.455	0.452	0.053
FOLIAGE'S WEIGHTS						
	ANA	ARC	DEV	DOC	TST	PRS
F1	0.225	0.153	0.340	0.243	0.243	0.130
F2	0.6	0.307	0.545	0.540	0.162	0.695
F3	0.15	0.512	0.090	0.162	0.540	0.087
F4	0.025	0.025	0.022	0.054	0.054	0.087

\*ANA=Analyst, ARC=Architect, DEV=Developer-programmer, DOC=Documenter, TST=Tester, PRS=Image and Presenter  
R1=null, R2=none, R3=with, T1= straight, T2=wave, T3=trapeze, F1=circular, F2=cloud, F3=fruit, F4=null

If we only consider Jung's Test we observe that ESTJ type is a highly qualified individual to perform different roles, data suggests Analyst or Architect as best roles; they are highly (E) extroverted and have good (J) judgment, qualities favourable to relate with other people and take important decisions. An

ISTP type is a Developer or Programmer, commonly (I) introverted and (T) thoughtful to his work with high logic for problem solving. An ESTJ is a Tester although others can perform this role and for Presenter we recommend an ENTJ, ESTJ or ESFJ. Data also indicates that types ESFP, ISFP, ESTP or ENTP should not be assigned as analysts, architects or testers, as they lack traits to better perform these roles.

From Table 2 we can analyze each attribute with its highest value, for example, when Root (R) is null (R1) the most probable role is Developer-Programmer (Q3). Without visible root (R2) we can assign Architect (Q2) or Documenter (Q4). Any sketch of root (R3) we are talking about an Analyst (Q1), Tester (Q5) or Presenter (Q6). Individuals performing this later role have been related with their own personal image, and a high percentage draw roots (R3) even highlighting thick roots, meaning that this individual likes to draw attention to him; he wants to be noticed and depends of what other people say.

There are fewer differences on the Trunk (T), but we can distinguish an Architect (Q2) from others because he draws the trunk in a trapeze shape (T3). The Foliage (F) distinguishes an Architect (Q2) and a Tester (Q5) from other roles as they draw trees with Fruits (F3), others draw cloudy (F2) type most of the times. We can clearly distinguish two roles from others, they are Architect (Q2) as he has the only combination of without root (R2), trapeze (T3) and fruits (F3); and the Tester (Q5) is the only one with root (R3), trapeze or wavy (T3 or T2) and fruits (F3). Drawing fruits means he has a clear view of what he wants to do, he has achieved personal goals in life, giving him the serenity to take charge of any project obtaining the final product, qualities of a leader and an architect.

Table 3: Big Five Software Engineering Role Means.

	O	C	E	A	N
ANA	50.61	63.14	51.57	45.84	64.28
ARC	54.15	67.23	57.30	52.15	64.15
DEV	51.66	52.77	44.11	54.47	59.22
DOC	55.00	61.28	51.92	52.00	58.85
TST	51.71	66.13	56.00	47.85	63.33
PRS	52.22	60.88	46.88	55.77	65.77

Big Five personality test results are presented in Table 3 showing means of each trait for every role. Each row is a personality vector unique for every Role, no two rows have exactly the same values for

every one of their attributes, and this gives us a significant difference between each Role to recommend based on the Big Five Personality Test.

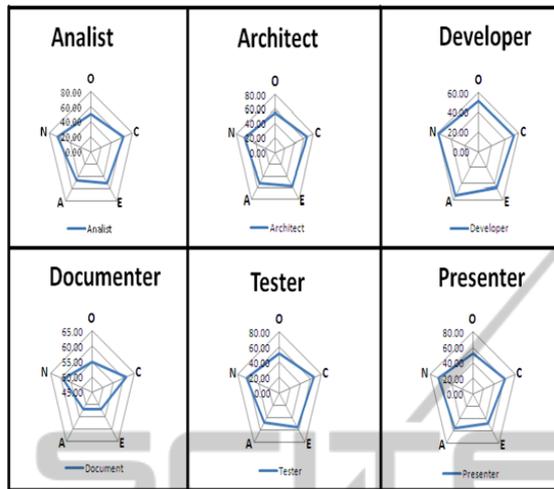


Figure 1: Big Five Personality Patterns (B5P).

We displayed them with a radar chart type in figure 1, obtaining unique Big Five Patterns (B5P) for Software Engineering Roles, showing significant differences between each role. Trait (E) is high for architect and tester, low for a developer. Trait (C) is high for most roles except for developer. Trait (A) is high for a presenter and developer, but low for an analyst. This can give us a glimpse of specific traits for a particular role. Thus one trait does not define the personality of a role, but a personality vector with all traits involved can give us differences between each role.

Assessment of these results has ascertain that a low degree of (E) is definitively recommended to place this person as a Developer-Programmer, this indicates a person highly Introverted, for him is difficult to relate with others, although his high degree in trait (A) is an asset as he is very cooperative, trusting and compliant, attributes for a good programmer, also the B5P figure of developer has a high degree of (O) indicating to be imaginative and creative, qualities for code design. Trait (N) is a most significant trait as envelopes a wide range of roles, those with low degree of (N) or better said with a high degree of (ES) Emotional Stability is a quality of a leader presenting security, reassurance and selfconfidence. For a high degree of (E) as noted in B5P figures, an architect, analyst and tester present this quality, indicating that these roles are best suited for outgoing people, they can relate easily with others, with passion and excitement reaching goals and objectives.

Qualitative data shows that teams have been fulfilling their expectations and fulfilling their roles. The majority of opinions on integrating a work team for a software development project have been positive. The teams formed have been successful and cooperative; have finished their jobs with great accomplishment. Particular opinions acknowledge that working as a team the job gets done better.

Knowing their strengths and weaknesses gave them more confidence and a better attitude to approach assigned role responsibilities. Understanding their role and knowing what qualities they can contribute, gives them a sense of belonging and a feeling of contribution to the team, a place and a reason for that role in team performance dynamics.

Quantitatively we can make reference to project time delivery, before implementing RAMSET methodology we worked with 10 teams, where 8 of them were late on delivery, with RAMSET we worked with 18 teams where 5 of them were late on delivery, a great improvement after organizing and integrating teams based on personality as shown in figure 2.



Figure 2: On time delivery percentage.

## 5 CONCLUSIONS

We know that personality is an important factor for team performance, thus is latent the difficulty to assign adequate roles to each member for success in team performance. Some personalities and typologies have been identified to perform certain type of roles, concluding that knowing software engineer's personality can help us build a better, more cohesive and less conflictive team.

Combination of personality tests gives more valuable information for decision making as they can help predict situations inside the working team.

If we know our software engineer's personality, we can assign a secure and self confident individual

leadership of the team as an architect, if he is cooperative, trusting and compliant he can very much perform as a developer-programmer, if he is self-disciplined, efficient, organized, but insecure he will be most adequate as a tester, if he is imaginative, inventive, curious, outgoing and autonomous role analyst is the perfect choice.

Future work consists on creating a software supporting tool developed under a fuzzy approach, integrating each personality test to build a decision making model support tool. This software will facilitate process of role assignment, which results in a choice of role selection for individuals in working team projects. Building a decision making model as future work will facilitate role assignment process for human resource personnel at selecting candidates for software engineering jobs.

Our methodology based on personality has revealed appropriate and adequate personality patterns for assignment of best advisable performing roles to members of working teams in software engineering development, according not only to capabilities of the people and demands of the role but also taking into consideration personality traits, thus showing that knowing software engineer's personality can improve software development process.

## REFERENCES

- Bernstein, A., Radosevich, D. J., Clesca, C. F., Lynn, J., (2008) Personality predictors of teamwork behaviors and decision-making style. *Review of Business Research*.
- Biffi, S., Halling, M. (2003) Investigating the Defect Detection Effectiveness and Cost Benefit of Nominal Inspection, In *IEEE Transactions on SE* (Vol. 29 no. 5, pp. 385-397).
- Calitz, A. P., Watson, M. B., Gideon, D. (1997) Identification and Selection of Successful Future IT Personnel in a Changing Technological and Business Environment. In *ACM SIGCPR 97*, San Francisco.
- Capretz, L. F. (2003) Personality types in software engineering *Int. J. Human-Computer Studies*, 58, 207-214, Elsevier Science Ltd.
- Costa Jr, P. R., McCrae, R. R. (1992) *Revised NEO Personality Inventory (NEO-PI-R) and NEO Five-Factor Inventory (NEO-FFI) manual*. Odessa, FL. Psychological Assessment Resources.
- DeMarco, T., Lister, T. (1999) *Peopleware: Productive Projects and Teams* 2nd edition. New York: Dorset House Publishing Co.
- Dubinsky, Y., Hazzan, O. (2006) Using a Roles Scheme to Derive Software Project Metrics, *Journal of Systems Architecture* 52(11), 693—699.
- Goldberg, L. R. (1993) The structure of phenotypic personality traits. *American Psychologist*, 48, 26-34.
- Goldstein, David K. (1988) An Updated Measure of Supervisor-rated Job Performance for Programmer/Analysts. In *SIGCPR'88*, ACM Digital Library.
- Gorla, N., Lam, Y. W., (2004) Who Should Work With Whom? Building Effective Software Project Teams, *Communications of the ACM*, (Vol. 47 No. 6, pp. 79-82).
- Griffiths, R. (1988) *Graphology: early recollections and the tree test*. The Author West Vancouver British Columbia.
- Guinan, P. J., Coopridge, J. G., Faraj, S., (1998) Enabling Software Development Team Performance During Requirements Definition: A Behavioral Versus Technical Approach, *Information Systems Research*, 9(2), 101-125.
- Koch, K., (1986) *El test del árbol: el dibujo del árbol como medio psicodiagnóstico auxiliar*. Kapelusz, Buenos Aires
- Leaetta, M. H., Frederick, L. O. (2000). Personnel Selection: looking toward the future-remembering the past. *Annual Reviews Psychology*, 51, 631-664.
- Pittenger, D. J., (1993) The Utility of the Myers-Briggs Type Indicator. *Review of Educational Research* 63(4), 467-488.
- Rodríguez, J., (2004). *Formación de grupos de desarrollo de software*. Ediciones Yoltéotl.
- Rothstein, M., Goffin, G. R. D. (2006). The use of personality measures in personnel selection: What does current research support? *Human Resource Management Review*, 16(2), 155-180.
- Smith, D. C., Bwecher, M., Burns-Howell, J., Kyriakides, J., (2001) Creating High performance IS Teams, In *SAICSIT*, Pretoria South Africa.
- Soley, L., Smith, A. L. (2008). *Projective Techniques for Social Science and Business Research*. Milwaukee: The Southshore Press.
- Teague, J., (2002) Women in Computing: What brings them to it, what keeps them in it? in *SIGCSE* (Vol. 34 no. 2, pp. 147-158).
- Tomayko, J. E. (1996) *Teaching a Project-Intensive Introduction to Software Engineering*, SEI Carnegie Mellon University Tech. Rep., Pittsburgh Pennsylvania.
- Tupes, E. C., Christal, R. E., (1992) Recurrent Personality Factors Based on Trait Ratings. *Journal of Personality*, 60, 225-251.
- U. S. Department of the Interior. Retrieved from: <http://permanent.access.gpo.gov/websites/doi.gov/www.doi.gov/octc/typescar.html>