Analysis and Visualization of Multimodal Socio-Technical Information of Free/Libre and Open Source Software (FLOSS) Projects

Oscar Hernán Paruma Pabón

A thesis submitted in

fulfillment of the requirement for the award of the

Degree of Magíster en Ingeniería - Ingeniería de Sistemas y Computación

FACULTAD DE INGENIERÍA

UNIVERSIDAD NACIONAL DE COLOMBIA - SEDE BOGOTÁ

November, 2016

I hereby declare that this thesis entitled "Analysis and Visualization of Multimodal Socio-Technical Information of Free/Libre and Open Source Software (FLOSS) Projects" is the result of my own research except as cited in the references. This thesis has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Student: Oscar Hernán Paruma Pabón

Date: November, 2016

Advisor: Fabio A. González O., Ph.D.

Co-Advisor: Jairo Aponte, Ph.D.

Co-Advisor: Jorge E. Camargo, Ph.D.

Dedication

A quienes siempre me han apoyado cuando emprendo nuevos proyectos. A quienes hacen grandes esfuerzos para que
Mis esperanzas e ilusiones se conviertan en realidad. A quienes esperan pacientemente el día que vuelva a ser suyo el tiempo que me prestaron para
Ir en busca de un sueño. A quienes me impulsan a seguir adelante y no desfallecer ante los obstáculos que se presentan. A quienes dejan ver en su rostro la
Felicidad, la alegría y el orgullo al saber que juntos cumplimos un nuevo objetivo.
A quienes me motivan para ser mejor días tras día. A quienes celebran como propio cada nuevo triunfo haciendo que ese
Momento quede grabado en la memoria y quiera revivirlo una y otra vez. A quienes transforman las partes difíciles, los sinsabores y las desilusiones en momentos
Inolvidables que hacen que todo haya valido la pena. A quienes seguirán ahí de manera incondicional, dando más de
Lo que reciben, sin esperar nada a cambio, ofreciendo una sonrisa, un abrazo y una palabra de aliento en el
Instante adecuado, porque solo eso basta para reconfortar el alma, despejar la mente y sentir que no hay nada más importante y valioso que la familia.
A Lucy, Juan Sebastián, Tomás Felipe, Adriana, Janeth. **A MI FAMILIA**. Gracias. Porque sin ustedes nada sería igual.

Acknowledgments

Al emprender nuevos proyectos contamos con el apoyo de las personas que siempre han estado junto a nosotros sin importar los sacrificios que llegan con cada nueva aventura. A medida que avanzamos en el camino que debemos recorrer para llegar a la meta nos encontramos con personas que, aún sin conocernos, nos impulsan para seguir adelante y no desfallecer ante los innumerables obstáculos que se presentan y que sería muy difícil sortear sin su compañía y orientación. En este momento, al ver que el objetivo se cumplió, agradezco a todos quienes me colaboraron para que hoy pueda decir **Lo Logramos**.

A mi familia, Lucy Pabón, Juan Sebastián Paruma Chaparro, Tomás Felipe Castillo Camelo, Adriana María Castillo Camelo, Lucy Janeth Paruma Pabón, por estar siempre a mi lado y por su apoyo incondicional. A los profesores Fabio González, Jairo Aponte y Jorge Camargo por su dirección y asesoría durante la elaboración de mi tesis. A los profesores Felipe Restrepo y Mario Linares por su revisión rigurosa al documento de tesis y por sus observaciones y sugerencias para lograr una versión mejorada del mismo. A las profesoras y profesores que orientan los cursos ofertados para la Maestría en Ingeniería - Ingeniería de Sistemas y Computación, Profesora Jenny Marcela Sánchez, Profesora Elizabeth León, Profesor Jean Pierre Charalambos, Profesor Eduardo Romero, por transmitir su conocimiento y ser parte del proceso de formación de nuevos investigadores. A las compañeras y compañeros que integran los Grupos MindLab y Software Evolution & Source Code Analysis, por sus valiosos aportes durante las sesiones de seguimiento y por dar a conocer su trabajo y avances en los seminarios.

A Dios y a todas las personas que de una u otra manera contribuyeron para que la ilusión de los primeros días ahora sea una realidad.

Muchas gracias.

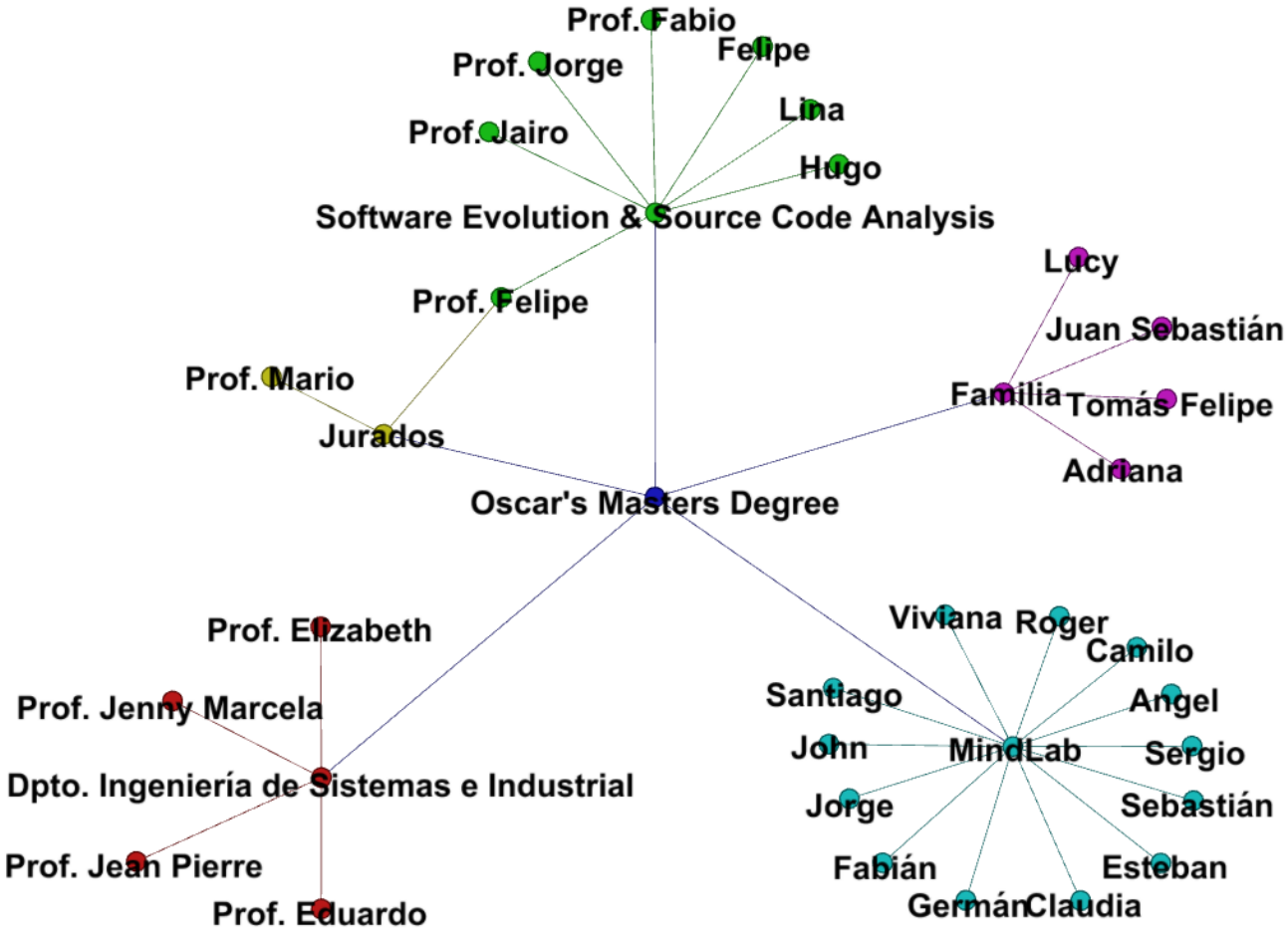


Figure 0.1: Graph of acknowledgments.

Abstract

Personality traits influence most, if not all, of the human activities, from those as natural as the way people walk, talk, dress and write to those most complex as the way they interact with others. Most importantly, personality influences the way people make decisions including, in the case of developers, the criteria they consider when selecting a software project they want to participate. Most of the works that study the influence of social, technical and human factors in software development projects have been focused on the impact of communications in software quality. For instance, on identifying predictors to detect files that may contain bugs before releasing an enhanced version of a software product. Only a few of these works focus on the analysis of personality traits of developers with commit permissions (committers) in Free/Libre and Open-Source Software (FLOSS[1]) projects and their relationship with the software artifacts they interact with. This thesis presents an approach, based on the automatic recognition of personality traits from e-mails sent by committers in FLOSS projects, to uncover relationships between the social and technical aspects that occur during software development processes. Experimental results suggest the existence of some relationships among personality traits projected by the committers through their e-mails and the social (communication) and technical activities they undertake.

[1] `www.gnu.org/philosophy/floss-and-foss.en.html`

Contents

Chapter 1

Introduction

Most of the work about Software Engineering focus on technical aspects, e.g. source code artifacts, and ignore social aspects, that is, the impact that communities of users and developers (and their interaction) have on the evolution of the project. Taking into account the social aspects of software development and software developed, as in one of the branches of Software Engineering (Social Software Engineering - SSE[1]), and if studies take into account elements such as the way developers work, cooperate, communicate, and share information; emerges the hypothesis that social aspects influence the way software projects evolve over time [64].

Software development processes involve different actors (requirements engineers, software engineers, software architects, software developers, test engineers, project managers, coordinators, users), each engaged in certain stages of the software life cycle and with specific knowledge of both the process and the product being developed. A group of such people form a community and continuous interaction among members of the community leads to the formation of a social network that revolves around the software evolution, establishing an interdependent relationship. Since project managers of commercial software carefully design the organizational structure of the project taking into account available resources, activities assignment, geographical aspects; in Open Source projects a pre-designed organizational structure is not observed and any existent structure is dynamic, self-organized, latent, and usually has not been explicitly defined [13].

Communication is a key success factor in any software project, especially in FLOSS projects which, usually, are developed in a geographically distributed manner. When no regular communication is maintained during the project implementation, the software quality is affected and it is reflected in the number of injected bugs [64, 1]. FLOSS projects are no different from the proprietary software development only by the type of licensing, used tools nor motivations, but also in their communication style [26]. As a typical behavioral trait of members of the FLOSS community can be noticed that some people tend to be more active in the mailing list while others are more active in the version repository [77]. Changes in the community (e.g., unexpected departure of a core person or a new community taking the project) or in the software product (e.g., a significant change in the base code) could influence the way in which the project will evolve over time, and a better understanding of the impact this could have would support the emergence of predictive models, guidelines, and best practices that could be applied by communities to optimize existing processes, communicate effectively, and make the software more attractive to their developers and users [64].

Repositories associated with software development projects contain lots of data and information that reflect the evolution of the software through each of the stages of the development process that have been

[1] `en.wikipedia.org/wiki/Social_software_engineering`

addressed. Having access to the history of the software allows us to know this software since its inception, identify the changes it has had over time, identify critical points, strengths and weaknesses, having inputs for decision-making about updates, modifications, adding new features and incorporating new technologies, and may even be useful to prevent or minimize recurring errors. However, since this information is usually available in a variety of formats (i.e., in a structured and unstructured way) it is not easy to interpret, and in most cases it is not easy to take advantage the most of it, or simply it is not used due to the lack of tools that facilitate the analysis, the inference and conclusions drawing [45, 58, 59, 65, 78, 81, 89]. Using Version Control Systems (VCS), Issue Tracking System (ITS), and Mailing Lists is currently a standard practice in software development projects that facilitates effective collaboration among stakeholders. Data stored by these systems, related to the history of changes, have multiple uses in software maintenance and evolution. For instance, a project manager can use information from the history of changes to assign new tasks to the most appropriate developers; test people can identify who is responsible for a bug and when it was injected into the source code; a developer can identify which parts of the system were modified during the implementation of certain features. Software repositories contain the data needed to answer these and other questions that support various maintenance tasks such as impact analysis, design improvements, refactoring, guidance on changes in software, check the integrity of a change, logical coupling detection [90, 87, 10, 69, 84, 88, 4, 25].

It is well known that visualization provides effective ways to break down the complexity of information, and it has shown to be a successful means to study the evolution of software systems [30]. That is why some works have combined the techniques of data mining and visual analysis [84, 31, 33, 30, 36, 39, 9]. As in other problems of analysis of large data sets, visualization is used to show to the users the information obtained through mining software repositories. Mining software repositories is an important activity in software evolution because the extracted data are used to address the changes in the software and to support different software maintenance and evolution tasks. The information mined from software repositories offers an overview of the changes made to the software system. To get a complete picture, the extracted data must be filtered, integrated, and presented in a visual way to the users [84].

Software Engineering and software development processes involve complex social processes in which the kind of communication and cooperative interaction among stakeholders determines to a large extent the quality of a collaboratively developed product [52]. Throughout the history of Software Engineering it has been repeatedly found that the humans involved are a key factor in determining project outcomes and success. However, the amount of research that focuses on human aspects has been limited compared to research with technology or process focus [56]. A growth in Empirical Software Engineering research has coincided with the greater focus on human factors of Software Engineering [52] and the interest in gaining a better understanding of how personality traits influence many aspects of task-related individual behavior has lead to conduct studies using personality models (e.g., MBTI[2], Big Five[3]). The results of such studies have contributed to a growing understanding in how to best associate personnel with the various tasks in a software project [3, 21]; in the identification of correlations between personality types and performance in software evolution and maintenance tasks [29, 51]; in finding relationships between personality composition of teams and the team performance [40, 68]; and in understanding to some extent the relationships between personality, team processes, task characteristics, product quality and satisfaction in software development teams [2]. This thesis fits into the group of works using the Big Five model [16, 66, 2, 86, 37, 63, 51] and the aim is to propose a method to uncover relationships between social and technical aspects from personality traits projected by the committers through the e-mails they send to the mailing lists of the FLOSS projects

[2] en.wikipedia.org/wiki/Myers-Briggs_Type_Indicator
[3] en.wikipedia.org/wiki/Big_Five_personality_traits

to which they contribute.

1.1 Problem identification

Social factors in Software Engineering (Social Software Engineering - SSE) and the relationship of these factors with technical aspects in software development processes have a growing interest in the community studying software evolution [64, 13, 1, 12, 14, 32, 34, 46, 47, 54, 55, 60, 72, 73].

Taking into account the large number of FLOSS projects that exist and the number of developers geographically distributed that contribute to its evolution, it becomes necessary to develop a methodology that allows, in a semi-automatic and unsupervised way, to build datasets from multimodal data[4] , to be scalable when processing small, medium-sized and even big and unlabeled datasets, and to infer personality characteristics from text. Such a methodology will be proposed and validated in this thesis.

This thesis aims to identify relationships between social and technical aspects of software evolution using data analysis techniques that assist in the identification of such relationships from data stored in different information sources associated with Free/Libre and Open-Source Software (FLOSS) projects. Additionally, because of the large amount of information stored in software repositories, characteristics most relevant to the analysis being performed must be identified.

Based on the above, the research problem is framed in the identification of relationships emerging between different aspects related with software evolution (developers, source code, commits) and social activities (communication) of community members through mailing lists.

The main motivation of this thesis is to study the relationships between software artifacts, mainly source code, and developers' personality traits to gain insight into the factors influencing developers to get involved in one or another FLOSS project. It is expected these insights may help to explain the implicit mechanisms that lead to self-forming software teams, and how developers' personality marks are reflected in some technical actions such as the commits they do.

The present thesis seeks to answer the following research questions:

- **RQ1**: What personality traits can be identified through communications among software developers involved in FLOSS projects?,
- **RQ2**: What personality traits stand out according to the projects the software developers are involved in and the technical activities (commits) they carry out in those projects?, and
- **RQ3**: What relationships can be observed between the social activities[5] (communication through the project mailing lists) of the committers and personality traits characterizing the technical groups they belong to?

[4] In the context of this thesis, multimodal data refer to data coming from multiple, different, and independent sources.
[5] For the scope of this thesis, social activities refer to communication among developers through the project mailing lists.

1.2 Objectives

1.2.1 General objective

To propose a methodology to identify relationships between socio-technical aspects from personality traits projected by the committers through e-mails they send to the mailing lists of the Free/Libre and Open Source Software (FLOSS) projects to which they contribute.

1.2.2 Specific objectives

- To build data sets from multiple information sources associated with FLOSS projects to be used in experimentation and evaluation stages.
- To define a feature extraction strategy to be applied to the built data sets.
- To propose a strategy to uncover relationships between socio-technical aspects of the evolution of Free/Libre and Open Source Software (FLOSS) projects.
- To validate the proposed methodology applying it to at least two case studies.

1.3 Results

Experimental results showed that personality traits scoring higher ($\geqslant 80\%$) and with nearly similar values through all technical clusters (e.g. *Intellect*, *Liberalism*, *Imagination*, *Openness, Cautiousness, Adventurousness*, and *Achievement striving*) could be considered as personality factors characterizing the project, i.e. people involved in the project will most likely exhibit high values in these personality traits. On the other hand, personality traits scoring lower ($\leqslant 25\%$) allow to identify relationships with the technical aspects, differentiating personality features among different technical clusters.This statement stems from the fact that technical activities in three of the four case studies showed *Self-expression* (present in Eclipse Platform, OpenStack, and Wikimedia projects), *Stability* (present in Eclipse Platform OpenStack, and Wikimedia projects), *Excitement* (present in Eclipse Platform, Xen, and Wikimedia projects), *Morality* (present in Eclipse Platform, OpenStack, and Xen projects), and *Structure* (present in Eclipse Platform, OpenStack, and Wikimedia projects) as personality traits characterizing technical clusters while social activities in two of the four case studies (Xen and Wikimedia projects) showed that committers having a tendency to actively participate in the mailing lists belong to technical clusters in which *Activity level*[6] is one of the most representative personality trait.

One of the most interesting results, although it was observed in only one (Eclipse Platform project) of the four case studies, was to find that committers grouped in a technical cluster[7] scoring high values in the

[6] In the context of this thesis, activity level was defined as the averaged number of times that project directories have been touched by committers in each technical cluster. Just average values $\geqslant 0.07$ were taken into account. It is computed from the number of times that a project directory has been touched by the committers of each technical cluster divided by the sum of the times that all project directories have been touched by the committers of each technical cluster.

[7] In the context of this thesis, technical cluster refers to one of the k partitions performed by the clustering algorithm on a given dataset (the technical dataset in this case), and formed by data objects (committers in this case) sharing similar

Artistic interests facet respect to other clusters contributes mainly to a project related to graphical elements, i.e., Eclipse Platform UI. Such preferences or behaviors were identified just from messages sent by committers to mailing lists.

The *IBM Watson Personality Insights*[8] service seemed to be a suitable NLP tool to analyze personality traits from text when is impractical to apply personality tests to each participant of a study, however it is necessary to design and carry out experiments to test the validity of the perception of personality traits derived from such service (e.g. using a control group to which can be applied personality tests and contrast these results with those obtained from the *IBM Watson Personality Insights* service) in order to ensure the reliability of the analysis and the conclusions drawn at the end of the study.

1.4 Contributions

Most of the results reported in the literature studying human aspects in Software Engineering, specifically those related with personality traits, are based on personality assessment questionnaires designed by psychologists and applied directly to the software team members. This thesis addresses the problem of uncover relationships among personality traits and the social and technical activities performed by the software team proposing a novel approach which involves the use of several tools (please refer to subsection 3.3Tools) and clustering techniques to extract personality traits just from the text developers write in their e-mails.

To the best of our knowledge, this is one of the first studies analyzing personality traits of software developers and their relationships with social and technical aspects in FLOSS projects.

Preliminary results were reported in a paper entitled "*Finding Relationships between Socio-Technical Aspects and Personality Traits by Mining Developer E-mails*"[67] and it was accepted for the 9th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE 2016)[9].

1.5 Document organization

The rest of this document is structured as follows: Section 2 presents a mapping study about Human Factors in Software Engineering. Section 3 describes the Socio-Technical Analysis Methodology proposed in this thesis. Section 4 groups the experiments conducted and the results obtained for four case studies; Section 5 presents the conclusions. Threats to validity are in Section 6, and lastly, in Section 7 we propose some future work.

characteristics (i.e., committers who touched nearly the same project files/directories).

[8] `www.ibm.com/watson/developercloud/doc/personality-insights/`

[9] `www.chaseresearch.org/workshops/chase2016`

Chapter 2

Human and Social Aspects in Software Engineering - A Mapping Study

Part of the work done in this thesis was focused on gain insights into the most relevant topics about human and social aspects in Software Engineering. Using a top-down approach, i.e., going from the general to the particular, and following the guidelines for mapping studies [17] we searched for published papers that may contain relevant research results related with the goal of this thesis and we selected the most appropriate from these after further examination. From the selected papers (please refer to section 7) we identify the topics that they becomes involved with and as a result we obtain a mapping between topics and documents which is summarized in Table 2.1.

Topic	**Reference**
Social aspects	[75, 50, 79, 80]
Ethnography	[75, 52, 79]
Gender	[79, 80]
Communication	[47, 50, 11, 46, 7, 71, 83, 1, 12, 74, 48, 5, 60, 76, 44, 49]
Technical aspects - Skills	[18, 28, 43, 19, 53, 80, 51]
Human factors	[6, 75, 3, 50, 70]
Personality traits derived from personality tests	[35, 86, 37]
Big Five	[16, 66, 86, 37, 63, 51]
MBTI	[15, 18, 20, 28, 43, 19, 52, 79]
NEO Personality Inventory	[16, 86]
International Personality Inventory Pool (IPIP)	[86, 51]
Eysenck Personality Questionnaire-Revised (EPQ-R) short version	[35]
Personality traits automatically derived from text - Big Five	[61, 62, 8, 23, 24, 41, 22, 57]
Experience	[6, 3]
Human factors - Skills	[19, 53]
Organizational aspects	[82, 75, 66, 3, 52, 63]
Teams	[6, 15, 16, 66, 63, 52, 57]
Client relationships	[52]
Management	[52]
Psychoempirical Software Engineering	[42]

Table 2.1: Mapping between topics and references.

Figure 2.1 summarize and connect topics related with human and social aspects in Software Engineering.

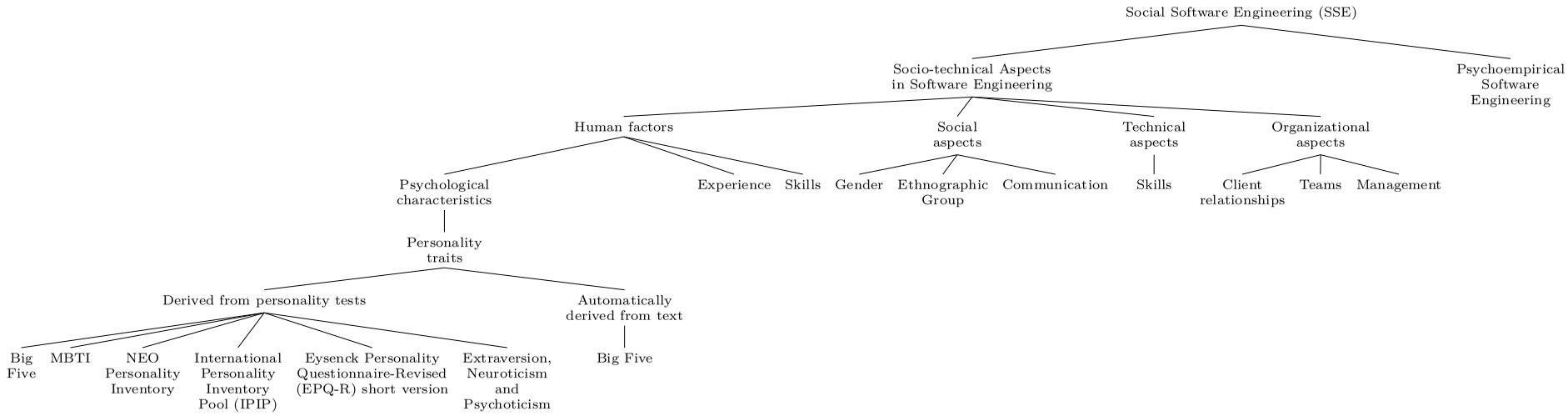


Figure 2.1: Summary of topics about human and social aspects in Software Engineering.

Below is an expanded description of the mapping between topics and papers to which they are related.

2.1 Human factors

Some authors agree that most of the research on the software process has focused on technical aspects [82, 75, 51] due to the misconception that, as stated by Sommerville and Rodden [75] "process could be represented as programs which could be executed by automated or human agents", however, after understanding that processes are carried out by people and not by machines there has been growing interest in research on the influence of human factors in the software building process.

Human factors have been a topic of interest in the community of researchers in Software Engineering and it is reflected in the amount of published works about this topic [6, 75, 16, 66, 18, 20, 3, 28, 50, 43, 19, 70, 51]. The group of human factors includes psychological characteristics, personality traits, experience, and skills and each of these factors has an effect on the software development activities and on the developed product.

John et al. [50] describes a workshop on human and social factors of software engineering in which the importance of these factors on the success of software development activities and the resulting product is highlighted. The workshop focuses on soft aspects in software development such as the human communication and the social environment of software developers, and including and combining approaches of software engineering with social science they try to point out solutions and conditions for human-centered software engineering. The topics discussed in John et al.'s work cover a wide spectrum including social factors in software process improvement and software development, the impact of personality types on software development and behavioral patterns in Software Engineering, empirical and qualitative research approaches for social factors in software development, ending with the review of communication and collaboration aspects of software development. As a result of the workshop discussions, John et al. refer that to get a full understanding of a social system like a software project, qualitative research is needed in order to understand what the important factors are and how and why they may influence. Additionally, John et al. identify the relations among personality, skills, and roles as one of the most important topics of human and social factors research in Software Engineering, the same as the forms of management and levels of freedom for the software engineers in software companies which is confirmed by a phrase mentioned in the workshop *"Happy programmers write better code"*. Finally and seeking to attract more commercial interest in this topic, John et al. argue that getting the human and social aspects right in software development will increase productivity, improve quality and customer and user satisfaction which translates into an increase in the perceived capital, and the knowledge about human and social factors will be highly valued by industry since improved organization of work practices is essential to be competitive.

2.1.1 Personality traits derived from personality tests

Personality traits are defined by the American Psychiatric Association as "enduring patterns of perceiving, relating to, and thinking about the environment and oneself that are exhibited in a wide range of social and personal contexts." [27, 51]. Personality tests are widely used in most studies on personality in Software Engineering mainly to identify personality traits that characterize software developers and to determine how those traits impact and are reflected in software development activities [16, 66, 27, 51].

A work referenced by Cruz et al. [27] in their mapping study mentions that "Myers-Briggs Type Indicator (MBTI) is the most popular approach for assessing personality profiles in Software Engineering. Nevertheless, the Five Factor Model (FFM - also known as the Big Five) of personality is currently gaining popularity among personality psychologists."

Among the most popular personality tests are the Myers-Briggs Type Indicator (MBTI), NEO Personality Inventory (assesses the Big Five personality traits), and International Personality Inventory Pool (IPIP, used in the Big Five personality traits assessment) [51].

Although neither MBTI[1] nor the Big Five[2] are considered by all psychologists to be universally accepted [18], many researchers are employing them for a variety of purposes [27].

2.1.1.1 Big Five

The five-factor model (or the Big Five) provide a taxonomy by which personality can be consistently defined and measured [16]. The traits of the Big Five are labeled and defined as: *Extraversion*, which is associated with being talkative, active, outgoing, sociable, confident, and enthusiastic; *Agreeableness*, which is associated with being good-natured, gentle, cooperative, forgiving, generous, and hopeful; *Conscientiousness*, which is associated with being self-disciplined, responsible, industrious, thorough, organized, and scrupulous; *Neuroticism* (or *Emotional Stability*), which is associated with being moody, worring, insecure, inhibited, emotional, and depressed; and *Openness to Experience*, which is associated with being imaginative, broad-minded, sensitive, intellectual, curious, and original [16, 66].

Relying on one of the most popular models in contemporary personality psychology research [51], Buchanan [16] explores the impact of Big Five personality patterns[3] on group cohesiveness and group performance on creative tasks, and as a result of his study establishes patterns of three Big Five traits (moderate levels of Extraversion, high levels of Openness to Experience and high levels of Conscientiousness) as potential predictors of group performance on creative tasks. Neuman et al. [66], based on the purpose of investigating the effectiveness of using personality to staff work teams, studied the relationship between team personality composition and work-team performance assessing individuals participating in the research on a broad range of Big Five personality traits. As Neuman et al. observed, for each specific trait of the Big Five, either TPE (team personality elevation, i.e. the average level of a given trait within a team) or TPD (team personality diversity, i.e. the differences in personality traits found within a team) predicted team performance and for the traits of *Conscientiousness*, *Agreeableness*, and *Openness to Experience*, TPE was positively related to team performance while TPD of *Extraversion* and *Emotional Stability* was positively related to team performance. Finally, Neuman et al.'s study suggests that, besides considers the magnitude of individual differences among candidates, similarity of individual trait differences should be considered when making team selection decisions.

Kanij et al. [51] based their study on Software Testers arguing the task set, mindset, and work approach of the testing profession is different to those of other software development practitioners, raising the question of whether the personality of software testers may be different to other people involved in software development.

[1] `en.wikipedia.org/wiki/Myers-Briggs_Type_Indicator#Criticism`
[2] `en.wikipedia.org/wiki/Big_Five_personality_traits`
[3] A Big Five personality pattern is a combination of individual Big Five traits, e.g., moderate levels of Extraversion, high levels of Openness to Experience and high levels of Conscientiousness [16].

The data they worked with corresponds to personality profiles defined by the Big Five Factor model of a large group of Software Testers and a large group of people involved in other roles of software development in industry. Profiles were compared to determine if there are significant commonalities or differences and the results of data analysis indicate that Software Testers are significantly higher on the *Conscientiousness* factor than other software development practitioners, while there is no significant difference in mean scores on the other factors.

2.1.1.2 MBTI

The Myers-Briggs Type Indicator (MBTI[4]) is a well-known instrument for measuring and understanding individual personality types [19], and it has been used for more than three decades to determine personality types [18]. MBTI describes 16 psychological types which result from the dynamic interplay of four pairs of preferences or dichotomies: extroversion (E) and introversion (I), sensing (S) and intuition (N), thinking (T) and feeling (F), and judging (J) and perceiving (P). The 16 types are typically referred to by an abbreviation of four letters; the letters of each of the four type preferences (e.g., ISTJ, ENFP, INTP, ...). All preferences are equally important. No preference is superior over any other preference, and no type is superior over any other type. [20].

Capretz [20] provides a personality profile of software engineers according to the Myers–Briggs Type Indicator and the results of his study suggest that software engineers are most likely to be ST (Sensing and Thinking), or TJ (Thinking and Judgment), or NT (Intuition and Thinking). Capretz concluded that "although software engineering attracts people of all psychological types, certain traits are clearly more represented than others in this field and, as a matter of fact, the software field is dominated by introverts, who typically have difficulty in communicating with the user." Most recently, Capretz and Ahmed [19] used the Myers-Briggs Type Indicator to mapping job and skills requirements to personality types for each of the activities involved in software engineering processes such as system analysis, software design, programming, testing, and maintenance. Capretz and Ahmed claim that "assigning people with personality types best suited to a particular stage increase the chances of the project's successful outcome."

Another study by Capretz [18], based on the fact that individuals with similar interests tend to go toward certain professions, aims to compare the personality profile of a group of software engineering students to engineering students in general at the University of Western Ontario, London, Canada in order to determine the personality differences among software engineers and all other engineers, according to the Myers-Briggs Type Indicator. Results shows that introverts are more common among software engineering students and, in general, engineers lean towards thinking types, as opposed to feeling and engineering students have more judging types (goal-oriented and value systems and order) than perceptive types (value a more adaptive or spontaneous approach). Minor differences were noted between the profile of all-engineering students and the software engineering students. The differentiating factor can be noticed in the distribution of ESTPs (Extraversion-Sensing-Thinking-Perception) and INFJs (Introversion-Intuition-Feeling-Judgement), wherein the software engineering sample contains more than double the number of ESTPs, but less than half of INFJs, compared to the all-engineering sample. Capretz's study confirms that occupations should attract particular personality types, and similar occupations should have similar personality type distributions.

A list of characteristics accepted as part of the profile of software professionals include [18]: low need for

[4] `en.wikipedia.org/wiki/Myers-Briggs_Type_Indicator`

social interaction, high need for challenge and achievement, low motivation towards management responsibilities, low identification with authority, low tolerance for interpersonal conflicts, loyalty to the profession rather than the employer, optimism regarding time estimates, systematic-methodical approach to problem-solving, and interest in stable and secure work.

MBTI has also been part of the research done by Da Cunha and Greathead [28], Greathead [43], Karn and Cowling [52], and Varona et al. [79].

2.1.2 Personality traits automatically derived from text - Big Five

Studies such as those conducted by Yarkoni [86], Golbeck et al. [37] and Gill [35] have sought to identify personality traits from text (blogs, twitter, email). As referred by Gill, *"Personality is projected linguistically"* and *"Personality can be perceived through language"*. The way the people write and speak and the words they use relates to their personality traits, so one can say there is a strong relationship between personality and the use of language, especially when people write or talk about topics of their choice[86].

While in some studies it would be possible to acquire personality information by asking the user or author directly [62], in most cases, as in the present thesis, it is not possible to assess the personality of people that are part of the analysis (unseen subjects or unseen individuals) through personality tests (or personality description questionnaires). Researchers have made efforts for automatic recognition of personality through language in conversation and text as in the case of the work done by Mairesse et al. [62] and Mairesse and Walker [61], where models for personality recognition are automatically learned from different corpora and sources of personality evaluation. Mairesse et al. and Mairesse and Walker reports experimental results for recognition of all Big Five personality traits in conversation and text. From the data sources they used in their experiments, they extracted a set of linguistic features as frequency counts of 88 word categories from the Linguistic Inquiry and Word Count (LIWC)[5] dictionary and 14 additional features from the MRC Psycholinguistic Database[6]. Specifically, their models contain features characterizing many aspects of language production: speech acts, content and syntax (LIWC), psycholinguistic statistics (MRC), and prosody. Their results confirms previous findings linking language and personality and further show that for some traits, any type of statistical model (for classification: C4.5 decision tree (the most easy to understand), Nearest Neighbour, Naive Bayes (performing the best), Ripper, Adaboost and Support Vector Machines with linear kernel and for regression: M5 regression tree, REPTree decision tree and SMOreg) performs significantly better than the baseline (for classification, the base line was a model returning the majority class, and for regression the baseline was a model returning the mean personality score), but ranking models (RankBoost) perform best overall. It is important to note that feature selection is an essential aspect to consider, as some of the best models only contain a small subset of all available features [62]. In addition, they reported that *Extraversion*, *Emotional Stability* and *Conscientiousness* are easier to model, and recognition models based on observed personality perform significantly better than a baseline returning the average personality score, as well as better than models using self-reports. Finally, they conclude that personality can be recognized by computers through language cues.

A couple of works based on the premise that some of the LIWC measures are correlated with the Big Five personality traits are those done by Bazelli et al. [8], and Licorish and MacDonell [57]. Bazelli et al.

[5] `liwc.wpengine.com`
[6] `www.psych.rl.ac.uk`

use LIWC to analyze questions and answers posted on StackOverflow and to define the personality traits of the posts' authors. They found out that the top, medium, and low reputed authors differ in Neuroticism, Extraversion, Openness, Agreeableness, and Conscientiousness. According to Bazelli et al.'s results, top reputed authors are less neurotic, more extroverted and open compared to medium and low reputed users, and authors of up voted posts express significantly less negative emotions than authors of down voted posts. Licorish and MacDonell examined personality through language use and provided insights into personality variations among members of distributed and global software development (GSD) teams. They mined the IBM Rational Jazz repository and used social network analysis (SNA) to cluster team members working across a set of teams into two groups (Top Members and Others). Then, Licorish and MacDonell performed linguistic analysis to explore personality reflected in developers' messages, and related this evidence to records of activity in project history logs. Their results show that the Top Members demonstrated more openness to experience than the Other practitioners. Additionally, practitioners involved in usability-related tasks were found to be highly extroverted, and coders were most neurotic and conscientious.

Celli et al. [23, 24, 22], in an attempt to solve the main problems with the supervised approaches to personality recognition from text, such as limitations in data annotation and language dependency besides the language dependency of the resources (LIWC and MRC) and considering that when training models on a specific domain or language are not very effective if used on different domains, they present *PR2* (or PeaR - adaptive Personality Recognition system in the online version[7]), a natural language processing (NLP) tool for personality recognition that performs unsupervised classification of Big Five personality types from unlabeled text using language-independent features. Tests conducted in English and Italian reached acceptable performance values ranging from 62% to 73% of correct predictions, depending on the size and quality of the data.

Within the group of works presenting visualization tools to show graphically personality traits derived from text are *IBM Watson Personality Insights*[8] and *PersonalityViz* [41].

Gou et al. [41] present *PersonalityViz* as an interactive visualization tool to help people explore and understand their personality traits (and those of others) and their changes over time derived from social media using the Linguistic Inquiry and Word Count (LIWC) text analysis tool and LIWC/Big Five personality correlations to compute personality scores of a person from their tweets. As a complementary feature, *PersonalityViz* build a temporal model of personality with a series of personality scores based on sets of tweets from different time frames enabling users to explore temporal trends of their personality. They used Sunburst visualization metaphor to show the Big Five personality traits and facets scores and a timeline view of personality features to show the changes of both Big Five traits and their facets over time.

2.1.3 Human factors - Experience

Experience appears in some studies [6, 3] as a factor not explicitly controlled nor observed, but it is mentioned in the description of the people involved in the experiments conducted by the authors. For instance, participants in the Basili et al.'s study [6] were "advanced undergraduate and graduate students in the Department of Computer Science. None were novice programmers, all had completed at least four semesters of programming course work, several were about to graduate and take programming jobs in government or

[7] `personality.altervista.org/pear.php`
[8] `watson-pi-demo.mybluemix.net`

industry, and a few had as much as three years of professional programming experience.", and in the experimentation phase described by Acuña and Juristo [3] they mentioned that "in four of the projects selected at random according to statistical principles, the people were assigned to roles according to the team manager's preferences, that is, by experience, which is how it is usually done in software development projects."

2.1.4 Human factors - Skills (Soft Skills)

Skills are the central topic in the work done by Capretz and Ahmed [19]. After analyzing job descriptions for software engineers, they identified that job advertisements generally divide software engineering skill requirements into two categories: hard and soft skills. Soft skills include personality traits, social interaction abilities, communication, and personal habits. Hard skills are described later in section 2.3 Technical aspects - Skills (Hard Skills). Capretz and Ahmed argue that assigning people with personality types best suited to a particular stage increases the chances of the project's successful outcome and it could be achieved by mapping soft skills and psychological traits to the main stages of the software life cycle.

Kimmelmann [53] identified a lack of knowledge to understand which competencies are necessary to reach specific status in Open Source (OS) projects and she focused her work in the identification of competencies that are relevant in order to work as an OS developer in comparison to a developer job in the proprietary software sector. Kimmelmann summarizes the results saying that technical competencies will be a stable aspect of a successful developer and social competencies are connected to a growing importance for future success in Open Source projects as users and developers are becoming more diverse in their cultural/linguistic background or personal expectations onto Open Source software. Kimmelmann remarks that Open Source developers are not part of an inner circle of equals but members of a global community which includes users without technical experience and this makes English language skills, empathy, communication skills and the ability to provide constructive feedback necessary.

Kimmelmann defined social competencies as those related with interpersonal skills required to support the software development and distribution process or the person's own career in an explicit way, and personal competencies as those comprising attitudes, values and motivation of the software developer.

Some aspects of the results presented by Kimmelmann that are worth noting for their relevance to this thesis are listed below:

> " ...
>
> *Being a successful member of an OS project is not limited to technical competencies. ... Indeed it is the social competency field that is crucial across all levels of career. The high importance of these competences rejects the prejudice of the social incompetent "nerd" in a very comprehensive way.*
>
> *Communication skills and ability for team work are the most important competencies at all as developing OS software is mainly organized by email-communication and through global virtual teams.*
>
> *The rights of committers are connected with trust from the community and its clients. ... The status can be approved by the public examination of the applicant's work: "code talks". That does not only mean to be competent and to do the right thing but to live the philosophy of social give*

and take as well as to show a social competent behaviour to other members.

. . . As work itself is accessible to everyone in the community, the contributions of everyone are also achieved for open review by its members. "The net never forgets". This public profile is not limited to technical competencies. Social competencies can be observed by following the GitHub-Repositories and mailing-lists as well.

. . . "

2.2 Social aspects

Software Engineering is a collaborative, knowledge-intensive, human-centric activity. Architects, designers, developers, users, testers, and managers communicate, share knowledge, exchange artifacts, and coordinate their efforts in order to create and maintain large and complex software systems. Understanding the social aspects of Software Engineering is, therefore, crucial to understanding how to build software effectively, improving the creation and maintenance of software systems as well as the management of software projects [80].

As stated by John et al. [50] and [80], "Software is developed for people and by people", so human and social factors have a strong impact on the success of software development activities and the resulting product.

Sommerville and Rodden [75] realized that the existing approaches to software process modelling are too mechanistic for describing processes which are dominated by human activities, therefore they turned to social sciences and adopted methods of process analysis which might be applicable to study software processes.

Social aspects in software development and software processes includes communication, knowledge sharing, motivation, cooperativeness, and collaboration [50].

2.2.1 Gender

Gender appears in Varona et al.'s study [79] as an aspect not explicitly controlled but observed in the results. It is mentioned in the description of the people involved in the experiments conducted by Varona et al., where the ratio between genders was approximately even with 48% males to 52% females in the sample. Considering participants' gender, results of Varona et al.'s study shows no significant differences among MBTI personality types extraversion (E), judgment (J) and perception (P) which have the same distribution. However, intuition (N) and sensing (S) show the opposite behaviour (60% males, 40 % females in N; 42% males, 58% females in S), and also shows a relatively higher percentage of males in thinking (T) (55 % males, 45 % females) and a higher percentage of females in feeling (F) (31 % males, 69 % females).

Vasilescu [80] provides a quantitative study in order to assess the representation and social impact of gender in online communities. Vasilescu's study was an attempt to quantitatively evaluate the presence of women in specific software-development-related online communities, and to compare their levels and duration of engagement with respect to the male counterparts.

2.2.2 Ethnography

Ethnographic research methods were originally founded by social anthropologists to aid them in their understanding of different cultures and environments. Those methods are now used in many disciplines (e.g. Empirical Software Engineering - ESE) in which research involving humans is important [52].

Ethnography involves an observer spending an extended period of time living in a society or working environment and making detailed observations of its practices. Subsequent analysis of these observations reveals information about the structure, organization and practices which take place in that environment. Ethnography is useful because is concerned with what actually happens rather than some notional definition of what should happen. [75].

For the studies of software processes, Sommerville and Rodde [75] adopted an approach based on an ethnographic investigation and they discovered that there were three areas where ethnography was valuable in understanding software processes: explicit identification of ad hoc cooperation, identification of individual process interpretations, and identification of organizational influences on software processes. Sommerville and Rodde also found that ethnography was a very effective method of discovering process rationale, however, ethnography was not an efficient way of discovering the overall process structure. Ethnographic analysis was very useful for understanding human interactions in the process and for discovering process subtleties which would not normally be represented in process models[75].

In a work that focuses on identifying personality types in a specific ethnographic group, Varona et al. [79] surveyed a group of Cuban software engineers and, using the MBTI instrument (Form M, spanish version), they identify their personality types. As result, the most prominent personality type was a combination of Extraversion, Sensing, Thinking and Judging (ESTJ) with a representation of 26% among the surveyed Cuban software engineers, followed by ESTP (Extraversion, Sensing, Thinking and Perceiving) with 13% and ISTJ (Introversion, Sensing Thinking and Judging) with 10%.

2.2.3 Communication

Large-scale software development projects require a lot of communication and coordination amongst the project workers. Communication and coordination activities influence (and are influenced by) the design, structure and evolution of software systems. Open Source software projects conduct all their activities in public, and every Open Source project includes one or more public mailing lists wherein project stakeholders can communicate and coordinate their activities. The entire trace of these mailing lists are archived and available for study. These archives, along with the source code repositories constitute a unique and valuable resource for the study of communication and coordination activities in software projects [11]. Effective communication among members of a software development team is considered to be a critical factor in the success of software projects [60], so it could be the social aspect most studied in Open Source projects [47, 50, 11, 46, 7, 71, 83, 1, 12, 74, 48, 5, 60, 76, 44, 49].

Communication between developers plays a very central role in team-based software development for a variety of tasks such as coordinating development and maintenance activities, discussing requirements for better comprehension, and assessing alternative solutions to complex problems. However, the frequency of communication varies from time to time. Sometimes developers exchange more messages with each other

than at other times [1]. Abreu and Premraj [1] investigated whether developer communication has any bearing with software quality by examining the relationship between communication frequency and number of bugs injected into the software. The data used for their study were drawn from the bug database, version archive, and mailing lists of the JDT subproject in Eclipse. Results showed a statistically significant positive correlation between communication frequency and number of injected bugs in the software, and it was noticed that communication levels of key developers in the project do no correlate with the number of injected bugs.

Emails related to the development of a software system contain information about design choices and issues encountered during the development process. Exploiting the knowledge embedded in emails with automatic tools is challenging, due to the unstructured, noisy and mixed language nature of this communication medium. Natural language text is often not well-formed and is interleaved with languages with other syntaxes, such as code or stack traces [5]. Bacchelli et al.[5] proposed an approach, based on a combination of parsing techniques and machine learning methods, to classify the contents of development emails in five predefined categories.

MacKellar [60] presents a case study of group communication patterns in a software engineering course. Using data collected from communication diaries kept by the students, communication among students was analyzed in terms of modality, success and purpose of each communication event, and some basic measures from social network analysis such as indegree and outdegree were computed. Differences among the groups on these measures were compared with respect to the success or failure of each group. Findings revealed that unsuccessful groups relied on less effective communication modalities, had more failed communications, and did not interact with the successful groups. In particular, no member of unsuccessful teams emerged as an information broker (individuals with a high ranking on the *wsi*[9] measure), in contrast to what happened in successful groups.

The growing interest in the usage of online social media channels (e.g., Facebook, Twitter, LinkedIn) has attracted the open source software community. Open source projects are often found to adopt an identity on these social media channels (e.g., Apache Solr/Lucene2 on Twitter, MySQL3 on Facebook) in order to disseminate project-related information (release announcements, major bug fixes) or gather feedback/questions posted by the users. Software developers contributing to open source projects also exists on social media channels. Quite often, they discuss, debate, or share experiences with others relevant to a software project using hashtags (e.g., #apache, #maven, #hadoop). Hence, the discussions covering open source projects are not limited to dedicated forums or mailing lists, there also exists huge amount of information on the social media channels. However, on the social media channels, less technical details relevant to the project's architecture, code or bugs are discussed. Much of the information available is regarding the experiences or announcements particular to a software project, and such valuable information should not be ignored.[49].

Iqbal's work [49] focuses on the comparison and analysis of the social behavior of software developers in different communication channels. Iqbal's work was motivated by the lack of research analyzing the behavior of software developers communication with each other on the mailing list/issue tracker and their communication on social media channels (e.g., Twitter). Results reported by Iqbal showed a very low correlation between developers communication on Twitter and mailing lists. Further, the social communication between software developers on Twitter is comparatively lower than the communication through traditional communication channels (i.e., mailing lists, bug tracking systems).

[9]The weighted successful indegree *wsi* was computed for each node P as $wsi = n - f$ where n is the number of communication events directed to P, and f is the number of failed communication events directed to P.

2.3 Technical aspects - Skills (Hard Skills)

Hard skills are the technical requirements and knowledge a person should possess to perform a task, including the theoretical foundations and practical experience a person should have to comfortably execute the planned task [19].

Reaffirming the importance of studying socio-technical aspects and human factors in the software development process and because of the lack of knowledge to understand which competencies are necessary to reach specific status in Open Source projects, Kimmelmann [53] present her work about relevant competencies for successful Open Source developers. The results are based on the analysis of interviews with Open Source software developers, their project managers and human resource managers in Open Source software companies. Participants were asked to describe relevant competencies of Open Source software developers depending on their position in Open Source projects. The profile of successful developers on their way to the committer status distinguishes among technical (T), social (S) and personal (P) competencies. Kimmelmann defined technical competencies as those corresponding to relevant technical knowledge, documented technical experience, and attitudes that are relevant for the successful implementation of Open Source software.

2.4 Organizational aspects

In a review about productivity factors in software development, Wagner and Ruhe [82] give a special consideration of human factors in software engineering which, as they explain, are often not analyzed with equal detail as more technical factors further that more than a third of the time a software developer is concerned with other kind of work, not just technical work. They point out one study in the 80's decade as "the first and most comprehensive work on the soft factors (all non-technical factors) influencing productivity in software development" and they highlight this as the boost of a stronger interest in soft factors in the 90's decade, resulting on studies focused on characteristics of groups and their influence on productivity and showing the positive correlation between the intensity of internal communication and the project success. One of the main contributions of Wagner and Ruhe's work is a list of soft and technical factors influencing productivity in software development.

Sommerville and Rodden's[75] work discusses human, social and organizational factors affecting software processes and, to remark, they discuss how to analyze software processes as human rather than technical processes.

With regard to software process enhancements, Acuña and Juristo [3] proposed a capabilities-oriented software process model to assigning people to roles according to their capabilities and the capabilities demanded by the role, and they validated empirically that this approach improves software development and influences the effectiveness and efficiency of software development. Acuña and Juristo affirm that "Our proposal is one of the very few approaches aiming to connect labour psychology and software production."

2.4.1 Management and Client Relationships

In an initial study of the effect of personality on group cohesion in software engineering projects, Karn and Cowling [52] selected a group of teams working on real industrial projects and those teams were observed in team, client, and manager meetings. The main aspects that were recorded during these meetings included the effect of personality type on behaviour towards teammates, clients, and managers; and how these aspects related to the amount of disruption, positive ideas, and equipoise brought forth from each team member. The objectives in Karn and Cowling's work were to identify combinations of personality types that will result in a smooth software engineering team, to show pairs and even triads of individuals who clash on a regular basis and have a tendency to disrupt the workings of the team, and to give deep qualitative descriptions of which personality types will work well and the area of the project they will excel in. Karn and Cowling's findings indicated that personality can seriously affect the cohesion of teams in meetings and certain personality types do have a positive, negative or a combination of both effects on the well being of a software engineering team.

2.4.2 Teams

Basili and Reiter [6] remark that factors directly related to the psychological nature of human beings play a major role in software development. In their study they focused on the effects of two human factors: the size of the programming team and the methodological discipline employed. Basili and Reiter concluded that research into the effects of human factors on software is dependent on suitable measurement of several non-functional software features such as reliability, maintainability, modifiability, cost-effectiveness, complexity, comprehensibility, and readability, but because of the difficulty to characterize and quantify these non-functional features it is necessary to use some metrics directly related with programming which are so well-defined and can be quantified. Basili and Reiter reported findings about the effects of team programming and methodological discipline upon non-functional software features demonstrating that the larger programming team size and the use of a disciplined methodology had beneficial effects on the development process and the developed product. Moreover, the disciplined methodology increased software reliability beyond that achieved by either individual programmers or programming teams using an ad hoc approach.

In an attempt to identify those factors that could help organizations to improve their software development process, Martínez et al. [63] proposed 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. This is achieved by applying RAMSET, a Role Assignment Methodology for Software Engineering Teams based on personality to obtain personality patterns related with software engineering roles performed in team projects. Martínez et al. concluded that knowing software engineer's personality can help to build a better, more cohesive and less conflictive team. Additionally, combining some personality tests gives more valuable information for decision making as it could help to predict situations inside the working team.

Along the same line of work, seeking to associate personality with the software process, Bradley and Hebert [15] proposed a model that can be used to analyze the personality type composition of an information system development team, and highlight the impact of personality type on team productivity.

2.5 Psychoempirical Software Engineering

Psychoempirical Software Engineering refers to the research in Software Engineering with psychology theory and measurement. It is emerging as a novel proposal in the field of SSE.

In their recent work, Graziotin et al. [42] described the challenge to conduct proper affect-related studies with psychology, provided a comprehensive literature review in affect theory, and proposed guidelines for conducting psychoempirical software engineering.

Chapter 3

Socio-Technical Analysis Methodology

In this chapter, the proposed methodology in this thesis is presented. The steps to be followed for studing socio-technical relationships in FLOSS projects are described. The datasets and tools used to validate the proposed methodology are also presented in this chapter.

3.1 Methodology

Because of the specificity of the study conducted in this thesis, it was necessary to define a methodology to study socio-technical relationships[1] in FLOSS projects. The main goal of the methodology is to automate as much as possible each of the steps that must be performed previous to obtain the data that will be analyzed to identify relationships between socio-technical aspects from personality traits projected by the committers through emails they send to the mailing lists of the FLOSS projects to which they contribute.

The proposed methodology starts by defining the best representation of the data describing the social and technical aspects of the developers in the software development process to, thereafter, build the datasets to be used in the experimental stage. The representation used for technical data[2] was binary vectors. Each vector represents whether a committer touched or not each file of the project. For personality data, the representation were the personality traits characterizing software developers, which they project through their emails.

An exploratory analysis was performed to become familiar with the data and to identify potential inconsistencies that should be corrected.

For each research question to be answered, a specific experiment was configured and carried out. The first experiment was intended to answer **RQ1**: What personality traits can be identified through communications among software developers involved in FLOSS projects?, so that, at this stage, the *IBM Watson Personality Insights* service[3] becomes more prominent. The dataset consisted of emails sent by committers to the mailing lists of the project and their subprojects.

[1] In the context of this thesis, socio-technical relationships refer to connections between technical activities (e.g., commits) or technical aspects (e.g., source code artifacts), and social activities (e.g., communications among software developers through mailing lists).

[2] In the context of this thesis, technical data refers to data obtained from the source code repository, and represent whether a committer touched or not each file of the project.

[3] `www.ibm.com/watson/developercloud/doc/personality-insights/`

The goal of the second experiment was to answer **RQ2**: What personality traits stand out according to the projects the software developers are involved in and the technical activities (commits) they carry out in those projects? and, at this stage, the personality traits identified in the previous stage were used, and using clustering techniques (k-means and spectral clustering), the personality traits characterizing each of the resulting clusters were identified.

Finally, the third experiment was intended to answer **RQ3**: What relationships can be observed between the social activities (communication through the project mailing lists) of the committers and personality traits characterizing the technical groups they belong to? For this purpose a graph (a social network) representing e-mail communication from committers to mailing lists was created. Using the results obtained in the above stages, the more distinctive personality traits of the nodes (committers) connected to the hubs (mailing lists) in the graph were identified.

This methodology was validated by applying it to four case studies. The methodology proposed is depicted in Figure 3.1.

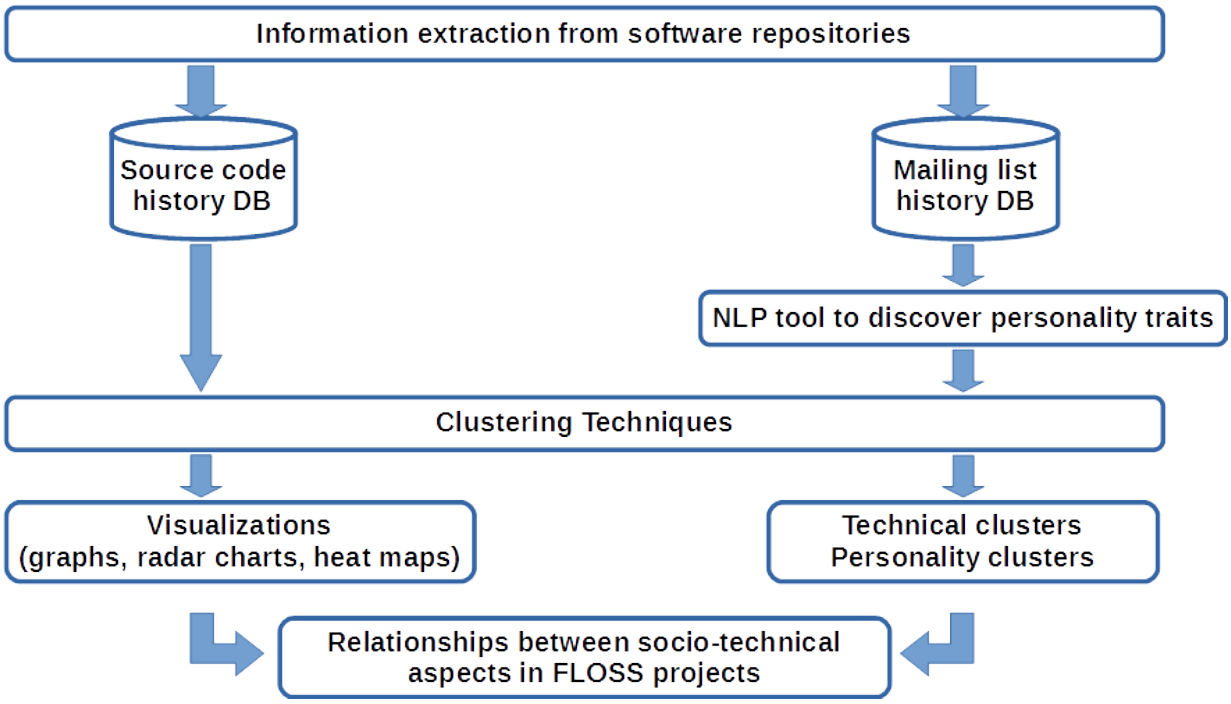


Figure 3.1: Socio-Technical analysis methodology diagram.

The details of each stage are described below:

3.1.1 Restoring databases from the dumps (source code repository and mailing lists)

In order to concentrate the efforts in later stages, no information retrieval was performed to obtain data from source code repositories and mailing lists since, despite being possible using the Metrics Grimoire set of tools, it requires considerable investment of time. Therefore, we used publicly available and previously created database dumps[4], [5] of each of the FLOSS projects which will be analyzed.

[4] gsyc.es/~jgb/repro/2015-msr-grimoire-data/
[5] bitergia.com/projects.html

Those dumps were restored in their corresponding database. Each database complies with the data structure defined by the tools CVSAnalY[6] and Mailing List Stats[7]. CVSAnalY extracts information out of source code repository logs and stores it into a database. Mailing List Stats stores into a database all the information contained in an e-mail (mbox files). CVSAnalY and Mailing List Stats are part of the MetricsGrimoire toolset.

From here, unstructured data available at source code repositories and e-mail archives (mbox files) are now in a structured way easily accessible through SQL standard queries.

3.1.2 Preliminary data exploration

To learn about the data of each project to be used in the experiments, a preliminary exploration was conducted. Information about the number of committers contributing to the project, the number of commits sent to the source code repository, the number of people who wrote to the mailing lists of the project, and the number of messages sent to these mailing lists gives an idea of the dimension of the data and its relevance to be used in the study.

At this stage are evidenced some correlations such as those between the number of committers and the amount of commits sent to the source code repository of a project. Also, the information about the number of messages sent to the mailing lists could be an indicator of the amount of data (mainly the amount of text) available to conduct the experiments at the stage in which personality traits are derived from text written by people.

At this stage, based on the results of the preliminary exploration, we determine whether or not to use the data of a project.

3.1.3 Datasets construction (social, technical and personality data)

Having the data in a structured way is less time consuming when creating the datasets that will be used later, but it is mandatory to know the structure established by the container in which the data are housed, a database in this case. Fortunately, the database schema of each of the Metrics Grimoire tools used in this work, CVSAnalY and Mailing List Stats, is available[8], [9].

At this point, the task corresponds to write the set of SQL statements to be executed on the database to retrieve the needed data. The main constructed datasets correspond to CSV files, one of them containing the information of files touched by a committer, represented as binary vectors where '1' means that the committer touched the file. Rows corresponds to committers and columns corresponds to files. To help in the analysis process, the level of granularity up from files to directories. Thus, the resulting dataset contains information in a binary format about directories touched by a committer, where '1' means that the committer touched the directory. Complementary information that could be extracted from this dataset is the activity level of the committers respect to the directories they touched, i.e. how much the committers tend to touch a file

[6] metricsgrimoire.github.io/CVSAnalY/
[7] metricsgrimoire.github.io/MailingListStats/
[8] github.com/MetricsGrimoire/CVSAnalY/wiki/Database-Schema
[9] github.com/MetricsGrimoire/MailingListStats/wiki/Database-Schema

or directory in a greater or lesser proportion respect to other files or directories they touches. As can be determined the project to which a directory belongs, also can be determined the committers' activity level for each project to which they contribute. Figure 3.2 shows an example of how the technical dataset looks.

Directory Committer	Directory_1	Directory_2	Directory_3	Directory_4	Directory_5	Directory_6	Directory_7	Directory_8	...
Committer_1	1	1	1	0	0	0	0	0	...
Committer_2	0	0	0	0	0	0	0	0	...
Committer_3	0	0	0	0	0	0	0	0	...
Committer_4	1	1	1	1	1	1	1	1	...
Committer_5	0	0	0	0	0	0	0	0	...
Committer_6	0	0	0	0	0	0	0	0	...
Committer_7	1	0	0	0	0	0	0	0	...
Committer_8	0	0	0	0	0	0	0	0	...
Committer_9	1	1	1	1	1	1	1	1	...
Committer_10	1	1	1	1	0	0	0	0	...
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.

Figure 3.2: Example of how the technical dataset looks.

The other dataset contains the scores of personality traits for each committer. At first, messages sent for committers to the mailing lists of the project they contributes to were retrieved from the database and then concatenated to fulfill the recommendation of the *IBM Watson Personality Insights* service, which says "*For statistically significant results, you need at least 3500 words and ideally 6000. You can still play with the demo if you have at least 100 words, but you should take those results with a grain of salt.*"[10], therefore, committers for those who could get a number of words equal to or greater than 3500 were those that were included in this dataset. The resulting corpus of text for each committer was analyzed with the *IBM Watson Personality Insights* service which returns scores for personality characteristics related to the personality models Big Five, Needs[11] , and Values[12] . The scores returned as a percentage "*indicates the extent to which the individual's writing discloses the associated characteristic when compared with a sample population*" [13]. Thus, the information in this dataset is represented with decimal values varying in the range between 0 and 100. Figure 3.3 shows an example of how the personality dataset looks. These couple of datasets are the input for the later stages.

3.1.4 Identifying technical and personality groups by applying clustering techniques

To find technical and personality groups of data objects that share similar characteristics, cluster analysis by using spectral clustering (to find technical clusters) and k-means clustering (to find personality clusters) was performed. These algorithms receive as a parameter the number of clusters (k) in order to partition a dataset. This parameter was obtained through the elbow method by plotting the result of the within-cluster sum of squared errors (SSE) for different values of k. We used three distance metrics (Euclidean, Jaccard,

[10] `http://personality-insights-livedemo.mybluemix.net/`. Currently, this statement was replaced with: "You can play with the demo with as little as 100 words, but for a more accurate analysis, you need more words."
[11] `http://www.ibm.com/watson/developercloud/doc/personality-insights/models.shtml#outputNeeds`
[12] `http://www.ibm.com/watson/developercloud/doc/personality-insights/models.shtml#outputValues`
[13] `https://www.ibm.com/watson/developercloud/doc/personality-insights/inout.shtml#outputJSON`

Committer	Agreeableness	Altruism	Cooperation	Modesty	Morality	Sympathy	Trust	Conscientiousness	...
Committer_1	2,07%	2,39%	51,65%	4,84%	4,13%	96,06%	10,92%	58,33%	...
Committer_2	7,96%	17,33%	85,27%	13,58%	15,71%	96,53%	28,24%	88,72%	...
Committer_3	4,29%	3,64%	81,83%	16,70%	11,38%	95,35%	40,89%	98,14%	...
Committer_4	6,49%	19,31%	82,44%	5,43%	21,41%	98,18%	36,67%	94,43%	...
Committer_5	4,64%	3,81%	45,42%	3,11%	6,30%	65,23%	14,93%	70,23%	...
Committer_6	3,31%	1,37%	11,72%	6,91%	3,19%	19,19%	6,46%	22,32%	...
Committer_7	2,37%	5,52%	67,49%	6,96%	5,49%	98,13%	13,40%	82,51%	...
Committer_8	24,88%	47,13%	88,61%	12,34%	38,72%	97,11%	70,24%	66,80%	...
Committer_9	3,27%	1,81%	71,83%	1,41%	5,89%	93,33%	13,20%	49,78%	...
Committer_10	1,65%	0,97%	36,94%	1,41%	1,66%	76,19%	1,89%	28,66%	...
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.

Figure 3.3: Example of how the personality dataset looks.

and Hamming for technical clustering; and Euclidean, City block, and Cosine for personality clustering) in plotting the Elbow curve trying to minimize the subjectivity associated to this method when choosing the appropriate value for k.

Looking at the point at which the SSE value changes significantly, was selected $k_t = 5$ for technical clustering and $k_p = 3$ for personality clustering.

Additionally, spectral clustering can perform clustering for an affinity matrix which, in the case of this thesis, corresponds to a matrix based on the Jaccard similarity coefficient computed between the vectors representing technical data. Data representation (binary vectors representing if a committer touched or not a file of a project) suggests that is more convenient to use a similarity metric like Jaccard similarity coefficient (used in this thesis) than Euclidean distance.

Just to clarify, a technical clustering corresponds to the result of applying the clustering algorithm (spectral clustering algorithm in this thesis) to the data representing the directories a committer has touched (i.e. a directory containing a file modified by a committer and sent by him to the repository).

On the other hand, personality clustering refers to the result of applying the clustering algorithm (k-means clustering algorithm in this thesis) to the data representing personality traits inferred from committers' texts (emails sent by committers to mailing lists).

3.1.5 Identifying personality traits that characterize each technical group

Each personality cluster has associated some personality traits that characterize and distinguish its members. From the results of the *IBM Watson Personality Insights* service it is possible to identify which personality traits are dominant in each cluster, becoming differentiating features, and what personality traits have similar values across all groups. In addition, it is known which technical group is associated to each committer.

By computing the entropy for each of the *Big Five* dimensions and facets, *Needs* and *Values*, it is possible to determine which of these attributes provide more information or become a differentiating factor when analyzing the technical groups, depending on the personality traits of the committers who are part of them. The lower the entropy, the greater the variation of the values of the corresponding attribute for the technical clusters, i.e., the attribute turns out more informative. This allow to characterize the group or groups in

which this feature is present and in which the focus should be on when making an analysis of each cluster.

Since it is known the technical cluster where each committer belongs to and the personality traits of committers belonging to each technical cluster, the centroids of personality traits for each technical cluster can be computed by averaging the values of the personality traits of committers in each technical cluster.

As a visual aid, radar charts were generated by plotting the 10 lowest values of entropy for the Big Five dimensions and facets, Needs, and Values of personality centroids for each technical cluster.

3.1.6 Visualization of social (communication) network

Using the information obtained from the e-mails sent by committers to the project mailing lists it is possible to build a weighted adjacency matrix which can be visualized as a graph representing e-mail communications. The graph shows committers and mailing lists as nodes and an edge is drawn between nodes (committers and mailing lists) when a committer sent a message to a mailing list. Mailing lists perform as hubs in the graph as this nodes has a high number of connections because messages sent by the committers are received there. The thickness of the edge between a committer and a mailing list represents the amount of emails sent by the committer to the list. Additionally, the color of the nodes representing committers corresponds to the technical cluster to which the committer belongs to.

For convenience and to expedite the analysis of the results obtained, only committers that have sent more than 10 e-mails to any of the lists were taken into account.

3.1.7 Identification of social and technical relationships

Finally, from the result obtained in the previous steps it is possible to identify relationships between socio-technical aspects in the FLOSS projects under study, such as personality traits that stand out according to the projects the software developers are involved in, and relationships observed between the social activities (communication through the project mailing lists) of the committers and the personality traits characterizing the groups they belongs to.

3.2 Datasets

Building on the work done by Gonzalez-Barahona et al. [38] was used the data of the Eclipse project[14] and OpenStack project[15] available at [16], with information from the following repositories: source code management (git for Eclipse and OpenStack), issue tracking (Bugzilla for Eclipse and Launchpad for OpenStack), mailing lists (archived in mbox format for Eclipse and OpenStack), and code review (Gerrit for Eclipse and OpenStack). Data of the Xen and Wikimedia projects were obtained from the Projects section at the Bitergia website[17]. From the dumps that are provided by Metrics Grimoire[18] and Bitergia, the databases were

[14] www.eclipse.org/eclipse
[15] www.openstack.org
[16] gsyc.es/~jgb/repro/2015-msr-grimoire-data
[17] bitergia.com/projects.html
[18] metricsgrimoire.github.io/

restored and the datasets used in the experimental stage were built.

Since the goal is to identify relationships between social and technical aspects in the evolution of FLOSS projects, the source code repository and the mailing lists are the most relevant data for the purpose of this thesis. Specifically, the data of the Eclipse Platform subproject was used, which in turn is divided into the following components [**?**]: Ant - Eclipse/Ant integration, Workspace (Team, CVS, Compare, Resources) - Platform resource management, Debug - Generic execution debug framework, Releng - Release Engineering, Search - Integrated search facility, SWT - Standard Widget Toolkit, Text - Text editor framework and UI - Platform user interface, runtime and help components. The OpenStack project data used in the experiments were those related to the Core Services: Cinder, Glance, Keystone, Neutron, Nova, and Swift; and those related to the Optional Services: Ceilometer, Heat, and Horizon. The Xen project data used in the experiments were those related to Xapi, Xenopsd, Xcp-rrdd, Xcp-networkd, Squeezed, SM, MirageOS, and Linux Kernel 2.6. Finally, the Wikimedia project data used in the experiments were those related to apps-firefox-wikipedia, apps-ios-wikipedia, apps-android-wikipedia, wikimania-scholarships, analytics-wikistats, labs-toollabs, apps-android-commons, apps-ios-commons, mediawiki-core, mediawiki-extensions-AccessControl, mediawiki-extensions-AccountInfo, mediawiki-extensions-ActivityMonitor, mediawiki-extensions-CSS, mediawiki-extensions-VisualEditor, mediawiki-extensions-Graph, mediawiki-extensions-Wikidata, mediawiki-extensions-WikiLove, and labs-maps.

3.3 Tools

Most of the tools used in this thesis are Python packages used in a broad range of scientific and academic research fields. Scikit-learn[19] was used for clustering, matplotlib[20] was used for plotting, NumPy[21] and SciPy[22] were used for scientific computing, pandas[23] was used for data manipulation, and NetworkX[24] was used for network visualization.

On the other hand, with regard to the study of social aspects identified from communications among software developers, the tool used was *IBM Watson Personality Insights*[25]. The *IBM Watson Personality Insights* service [86, 85] [26] can detect personality traits reflected in text written by a subject. This was particularly useful for this work since it was unfeasible to apply a personality test to each of the committers who contribute to the FLOSS projects under study.

[19] scikit-learn.org
[20] matplotlib.org
[21] www.numpy.org
[22] www.scipy.org
[23] pandas.pydata.org
[24] networkx.github.io
[25] www.ibm.com/smarterplanet/us/en/ibmwatson/developercloud/personality-insights.html
[26] www.ibm.com/smarterplanet/us/en/ibmwatson/developercloud/doc/personality-insights/science.shtml

Chapter 4

Case Studies and Results

To validate the proposed methodology for Socio-Technical Analysis, data from four FLOSS projects (Eclipse Project[1], OpenStack Project[2], Xen Project[3] and Wikimedia Project[4]) were used to run experiments and to analyze the results.

4.1 Case Study - Eclipse Project

4.1.1 Preliminary data exploration

To learn about the Eclipse Platform project data to be used in the experiments, a preliminary exploration was conducted. The results are summarized in Table 4.1. The date range for which data were obtained is between January 01st, 2003 and January 01st, 2015.

[1] www.eclipse.org/eclipse
[2] www.openstack.org
[3] www.xenproject.org
[4] wikimediafoundation.org/wiki/Our_projects

No	Project / Subproject	Description	Committers	Commits	Mailing List Senders	Mailing List Messages
1	Eclipse Platform	The Platform defines the set of frameworks and common services that collectively make up infrastructure required to support the use of Eclipse as a component model, as a Rich Client Platform (RCP) and as a comprehensive tool integration platform.	46	6829	405	939
2	Platform Text	Platform Text is part of the Platform UI project and provides the basic building blocks for text and text editors within Eclipse and contributes the Eclipse default text editor.	33	5911	71	454
3	Platform UI	Platform UI consists of several components, which provide the basic building blocks for user interfaces built with Eclipse. Some of these can be reused in arbitrary applications, while others are specific to the Eclipse IDE.	112	25110	375	5069
4	RelEng	Platform RelEng provides release engineering services for the Eclipse Project team, maintaining the build scripts that are used to massage the source from the developer to a download at eclipse.org.	4	205	232	22716
5	Resources	The resources component provides the fundamental model underlying the IDE portion of the Eclipse Platform. This includes the central concepts of resources (projects, folders, and files), builders, natures, resource change listeners, etc. The resources component contains no GUI, and can be run in a completely headless Eclipse application.	28	3077	180	1561
6	SWT	SWT, the Standard Widget Toolkit, is an open source widget toolkit for Java designed to provide efficient, portable access to the user-interface facilities of the operating systems on which it is implemented. SWT can be used independently of the rest of the Eclipse Platform.	46	21984	1125	5967

Table 4.1: Eclipse Platform project - Number of registers.

4.1.2 Technical and personality groups

To find technical and personality groups of data objects that share similar characteristics, cluster analysis through spectral clustering was performed. The algorithm receives as a parameter the number of clusters (k) in order to partition a dataset. This parameter was identified through the elbow curve by plotting the result of the within-cluster sum of squared errors (SSE) for different values of k.

Looking at the point at which the SSE value changes significantly, was selected $k_t = 5$ for technical clustering and $k_p = 3$ for personality clustering.

Figures 4.1 and 4.2 shows elbow curves used to select the k value for technical and personality clustering.

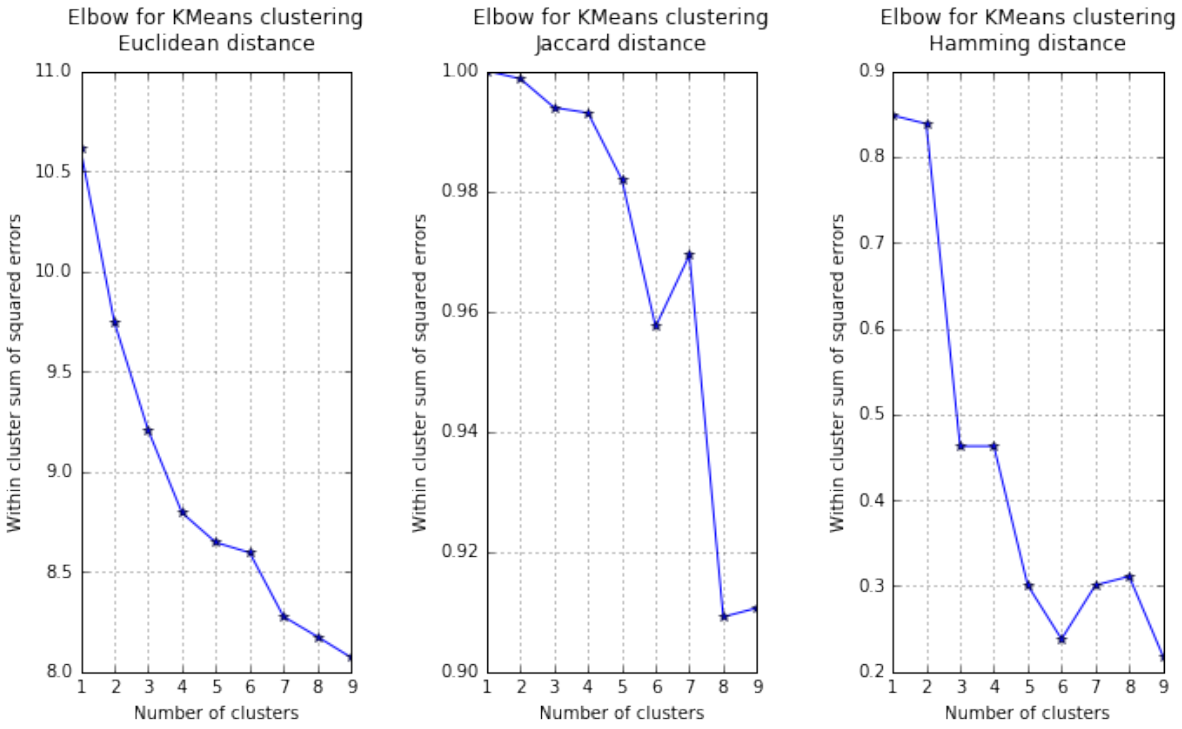


Figure 4.1: Eclipse Platform project - Elbow curves for tecnical clustering

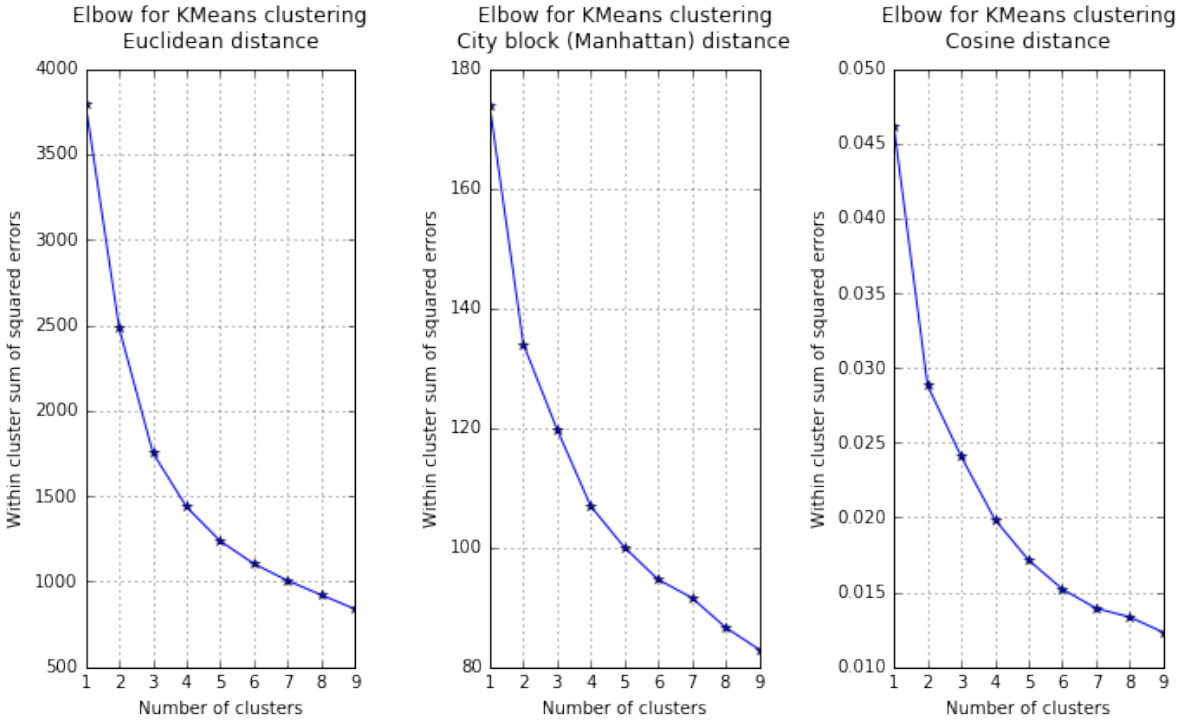


Figure 4.2: Eclipse Platform project - Elbow curves for personality clustering

Tables 4.2 and 4.3 show the results for technical clustering. The projects touched[5] by committers in

[5]In the context of this thesis, "to touch a file" is the action performed by a developer which have an effect in the source code

technical clusters were obtained averaging the number of times that project directories have been touched by the committers in each cluster. Only values that represent a participation or contribution of the committers to the project greater than or equal to 7% are taken into account.

Technical cluster	**Number of committers**
0	107
1	11
2	12
3	24
4	13
Total	168

Table 4.2: Eclipse Platform project - Results for technical clustering.

Technical cluster	**Eclipse Platform**	**Eclipse Platform Runtime**	**Eclipse Platform SWT**	**Eclipse Platform Team**	**Eclipse Platform Text**	**Eclipse Platform UI**
0	*	*	0.13	*	0.07	**0.67**
1	**0.24**	0.09	*	*	*	**0.56**
2	*	*	**0.73**	*	*	0.19
3	0.07	0.07	0.08	**0.27**	0.12	**0.39**
4	*	*	*	*	*	**0.84**

* *Values < 0.07*

Table 4.3: Eclipse Platform project - Results for technical clustering. Averaged number of times that project directories have been touched by committers.

In addition, for each technical cluster the number of committers who touched each project was computed, as shown in Table 4.4. Hence, to understand the meaning of technical groups the results presented in Tables 4.3 and 4.4 were taken into account. It was noticed that most committers from all technical clusters, except for technical cluster 2, contribute to Eclipse Platform UI project. In fact, there is great participation of technical cluster 0 (83 committers) and a high activity of the technical cluster 4 (0.84) with reference to this project. Analyzing participation in other projects, it was observed that committers from cluster 0 tend to be more present in the Eclipse Platform SWT project (35) just as committers from cluster 2 (12), while committers from cluster 1 lean toward Eclipse Platform Runtime (10), and committers from cluster 3 tend to work in Eclipse Platform Team project (24). Furthermore, it was noticed uniformity in cluster 4 as all the committers belonging to this group (13) contribute to Eclipse Platform SWT, Eclipse Platform Team, Eclipse Platform Text and Eclipse Platform UI projects, with more activity in the latter project (0.84).

Table 4.5 shows the results for personality clustering and the heat map in Figure 4.3 depicts the results of each Big Five dimension and facet, each Need, and each Value (rows) by each personality cluster (columns). To facilitate and improve interpretability just the top-10 (the lowest) entropy values for Big Five dimensions and facets, Needs, and Values are shown. As recommended by the *IBM Watson Personality Insights* service and for statistically significant results, at least 3,500 words written by each committer were analyzed. To get enough text for each committer, his/her e-mails sent to the project mailing lists were concatenated.

repository such as file creation, modification or deletion. Something similar apply for directories and projects.

Technical cluster	Eclipse Platform	Eclipse Platform Runtime	Eclipse Platform SWT	Eclipse Platform Team	Eclipse Platform Text	Eclipse Platform UI
0	11	17	**35**	22	19	**83**
1	7	**10**	6	5	7	**11**
2	2	0	**12**	7	11	**12**
3	13	12	13	**24**	14	**24**
4	9	6	**13**	**13**	**13**	**13**

Table 4.4: Eclipse Platform project - Number of committers touching projects in technical clusters.

The way in which the entropy calculation was performed is based on what Neuman et al. [66] named team personality elevation (TPE), "a team's mean level on a particular personality trait or set of personality traits, i.e. characterizing a team as high in TPE on extraversion would mean that for the team as a unit, members would be sociable, talkative, and assertive. This does not imply that all team members score high on this trait, just that there are at least some members whose scores **elevate the average for the team**." Thus, at first the distribution of the Big Five dimensions and facets, Needs, and Values was calculated for committers belonging to the same technical cluster, whether or not in the same personality cluster, and then the entropy for each Big Five dimensions and facets, Needs and Values was computed.

Personality cluster	Number of committers
0	42
1	24
2	2
Total	68

Table 4.5: Eclipse Platform project - Results for personality clustering.

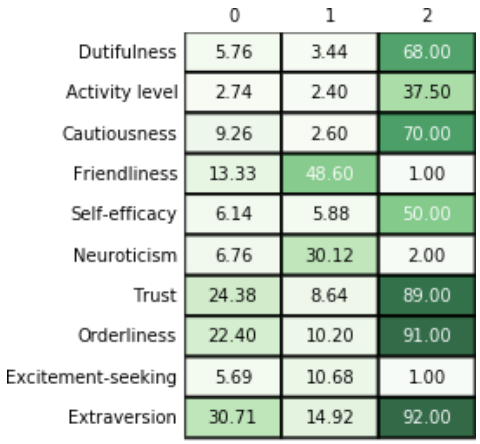


Figure 4.3: Eclipse Platform project - Heat map of the most discriminative factors for the personality clustering.

Then, in answering the **RQ1**, the personality traits can be identified through communications (e-mails sent by committers to mailing lists) between software developers involved in FLOSS projects, which are those corresponding to the Big Five dimensions and facets, Needs, and Values[6]. As highlighted in the heat map,

[6] www.ibm.com/smarterplanet/us/en/ibmwatson/developercloud/doc/personality-insights/models.shtml

personality cluster 2 groups the committers with the highest scores in personality traits such as *Extraversion* (92%), *Orderliness* (91%), *Trust* (89%), *Cautiousness* (70%), and *Dutifulness* (68%); and the lowest values in *Excitement-seeking* (1%), *Friendliness* (1%), and *Neuroticism* (2%). Personality traits characterizing cluster 1 by its moderately high values are *Friendliness* (48.6%), and *Neuroticism* (30.12%), opposed to cluster 2 which has very low values in those personality dimensions. Finally, personality traits standing out in cluster 0 are *Trust* (24,38%), *Orderliness* (22.40%), and *Extraversion* (30.71%), which are lower than those of cluster 2, but higher than those of cluster 1.

4.1.3 Personality traits characterizing technical groups

Each personality cluster has associated personality traits that characterize and distinguish its members. From the results of the *IBM Watson Personality Insights* service it is possible to identify which personality traits are dominant in each cluster, becoming differentiating features, and what personality traits have similar values across all groups. In addition, it is known which technical group is associated to each committer.

By computing the entropy for each of the Big Five dimensions and facets, Needs and Values, it is possible to determine which of these attributes provide more information or become a differentiating factor when analyzing the technical groups, depending on the personality traits of the committers who are part of them. The lower the entropy, the greater the variation of the values of the corresponding attribute for the technical clusters, i.e., the attribute turns out more informative. This allow to characterize the group or groups in which it is presented, and in which it is necessary to focus on when making an analysis of each cluster.

Because the technical cluster to which each committer belongs is known as well as the personality traits of committers belonging to each technical cluster, the centroids of personality traits for each technical cluster can be computed. Figure 4.4 show just the 10 lowest values and the 10 highest values of entropy for the Big Five dimensions and facets, Needs, and Values of personality centroids computed by averaging the values of the personality traits of committers in each technical cluster.

From the results reported in Figure 4.4 and Table 4.3, **RQ2** can be answered. Personality traits scoring higher ($\geqslant 80\%$) and with nearly similar values through all technical clusters (e.g. *Liberalism*, *Imagination*, *Openness*, *Intellect*, *Cautiousness*, *Adventurousness*, *Self-enhancement, and Achievement striving*) could be considered as personality factors characterizing the project, i.e. people involved in the project will most likely exhibit high values in these personality traits. On the other hand, personality traits scoring lower ($\leqslant 25\%$) allow to identify relationships with the technical aspects, differentiating personality features among the different technical clusters.

Figure 4.5 summarizes personality traits by technical cluster allowing to visualize which personality traits are more representative in each technical cluster. From this representation can be noticed the dominant facets for the different technical clusters. For instance, committers grouped in the technical cluster 4 scored high values in the *Artistic interests* facet in comparison with other clusters, and they contributes mainly to a project related to graphical elements, i.e., Eclipse Platform UI. Furthermore, a high value in *Structure* Need[7] (25.88%), and a low value in *Self-transcendence* Value[8] (8.5%) regarding the other clusters could explain why the committers of the technical cluster 4 contribute to only one project.

[7] `www.ibm.com/smarterplanet/us/en/ibmwatson/developercloud/doc/personality-insights/models.shtml#outputNeeds`
[8] `www.ibm.com/smarterplanet/us/en/ibmwatson/developercloud/doc/personality-insights/models.shtml#outputValues`

	0	1	2	3	4
Artistic interests	6.23	4.11	3.25	4.08	10.63
Activity level	19.50	7.89	5.75	14.17	8.88
Self-expression	10.40	6.89	3.25	9.08	9.88
Stability	8.67	4.89	3.75	4.75	8.13
Excitement	4.80	2.56	2.00	2.58	4.00
Orderliness	9.67	9.78	8.00	5.50	5.13
Morality	6.87	6.89	3.75	4.08	5.13
Emotionality	3.40	3.22	2.50	1.83	2.00
Structure	22.73	16.00	13.00	18.17	25.88
Self-transcendence	15.37	16.67	17.50	15.50	8.50

(a) The 10 lowest values of entropy (most discriminative personality factors for technical clusters) for the Big Five dimensions and facets, Needs, and Values of personality centroids.

	0	1	2	3	4
Liberalism	98.23	97.78	98.00	98.25	98.75
Imagination	97.87	97.67	98.25	98.58	98.63
Openness	97.23	96.67	97.50	97.67	98.13
Intellect	98.30	97.89	98.75	99.25	99.25
Cautiousness	91.67	93.33	91.25	92.00	93.75
Adventurousness	91.17	90.67	92.25	93.92	94.38
Self-enhancement	84.90	82.33	85.75	88.17	85.63
Achievement striving	82.70	79.11	80.00	85.08	83.63
Conscientiousness	68.90	69.56	70.50	67.00	73.00
Self-consciousness	51.23	45.22	44.25	50.00	46.75

(b) Top 10 highest values of entropy (personality factors characterizing the project) for the Big Five dimensions and facets, Needs, and Values of personality centroids.

Figure 4.4: Eclipse Platform project - Heat map of personality centroids for each technical cluster.

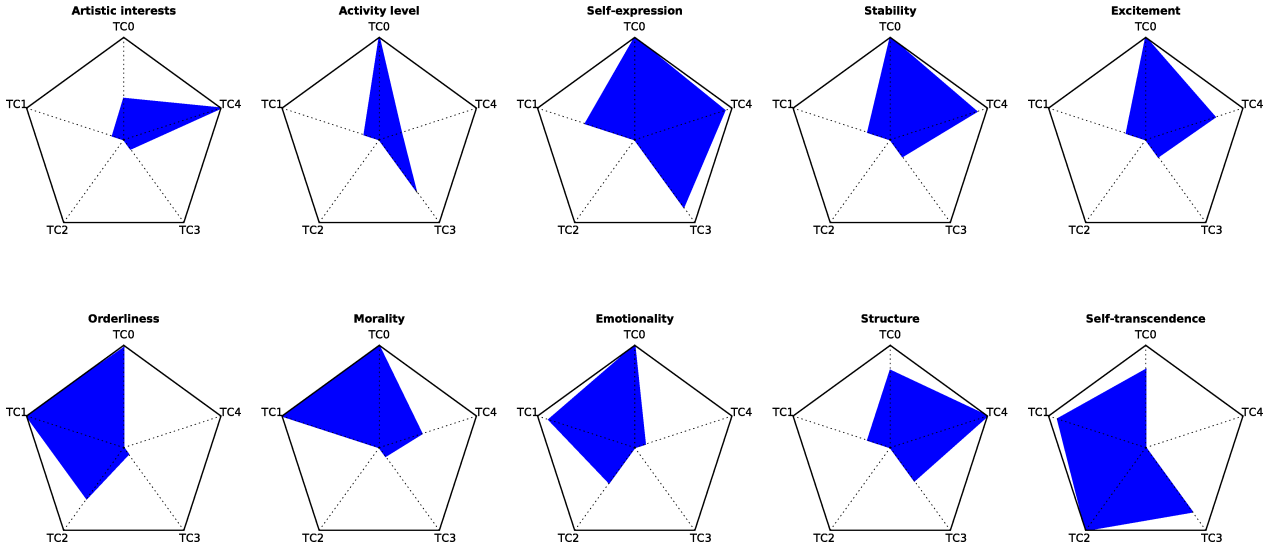


Figure 4.5: Eclipse Platform project - Radar charts of personality traits by technical cluster (TC).

From Figure 4.6 and based on values in Figure 4.4a it can be said that the distribution of personality traits is something similar for the couples TC0-TC3 and TC1-TC4 regarding the feature *Activity level* (*Extraversion* facet) alike for the couples TC0-TC3 and TC1-TC2 regarding the feature *Self-transcendence* (Human Value) and for the couples TC0-TC4 and TC1-TC3 regarding the features *Structure* (Need) and Stability (Need). The couples TC0-TC1 and TC2-TC3 have a similar behavior regarding the feature *Morality* (*Agreeableness* facet) as the couples TC0-TC1 and TC3-TC4 regarding the feature *Orderliness* (*Conscientiousness* facet). Observing the full set of technical clusters is conspicuously the lack of *Excitement* (Need) and *Emotionality* (*Openness* facet).

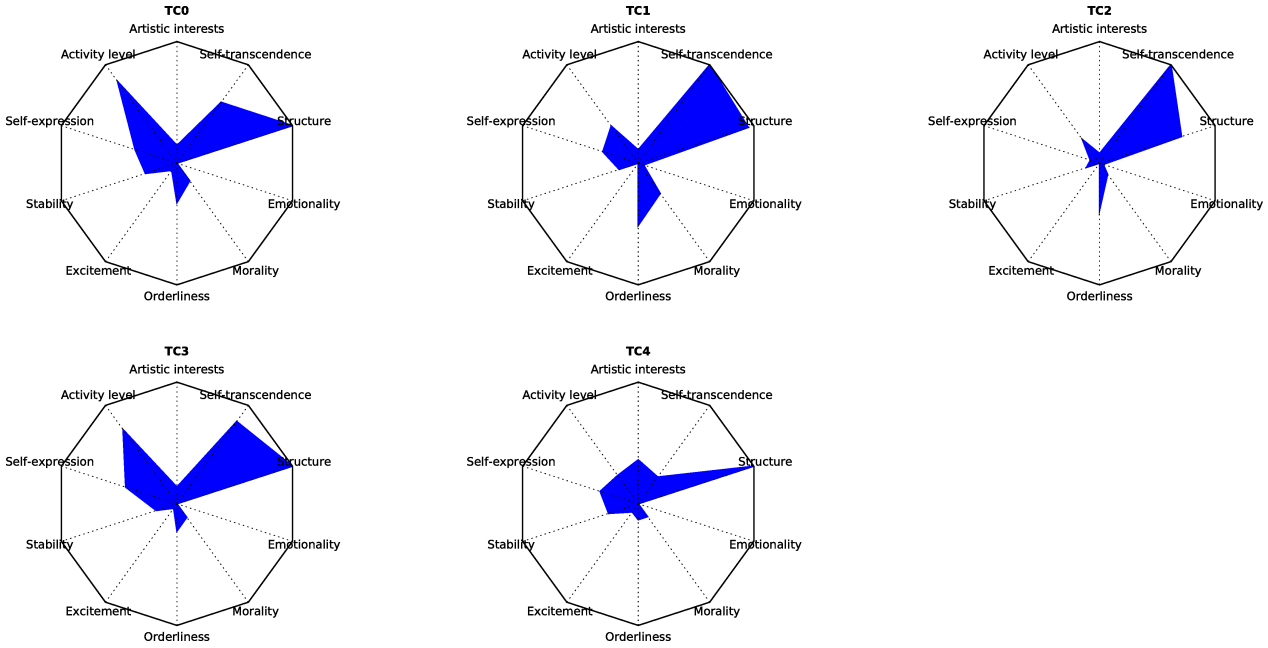


Figure 4.6: Eclipse Platform project - Radar chart of technical clusters characterized by personality traits.

4.1.4 Visualizing the social network - from committers to mailing lists

Using the e-mails sent by committers to the Eclipse Platform project mailing lists, a graph representing e-mail communications was built. The graph in Figure 4.7 shows committers and mailing lists (PlatformDev, Search, Text, Core, RelEng, UI, SWT, Team, i.e., red circles) as nodes. The thickness of the edge between a committer and a mailing list represents the amount of emails sent by the committer to the list. Additionally, the color of the nodes representing committers corresponds to the technical cluster the committer belongs to (cluster 0: blue, cluster 1: yellow, cluster 2: orange, cluster 3: purple, cluster 4: green). Only committers that have sent more than 10 e-mails to any of the lists were taken into account.

Figure 4.7 help to answer **RQ3**. Committers belonging to technical group 3 (purple circles) are those distributed through all mailing lists (red circles), with a participation, in each mailing list, of a different number of committers belonging to this cluster. This could be because of the ranking they have in the combination of personality traits such as *Self-transcendence* (15.50%),*Structure* (18.17%), and *Activity level* (14.17%).

Figure 4.7 also show that some representatives (2 out of 7) of technical cluster 1 (yellow circles) have

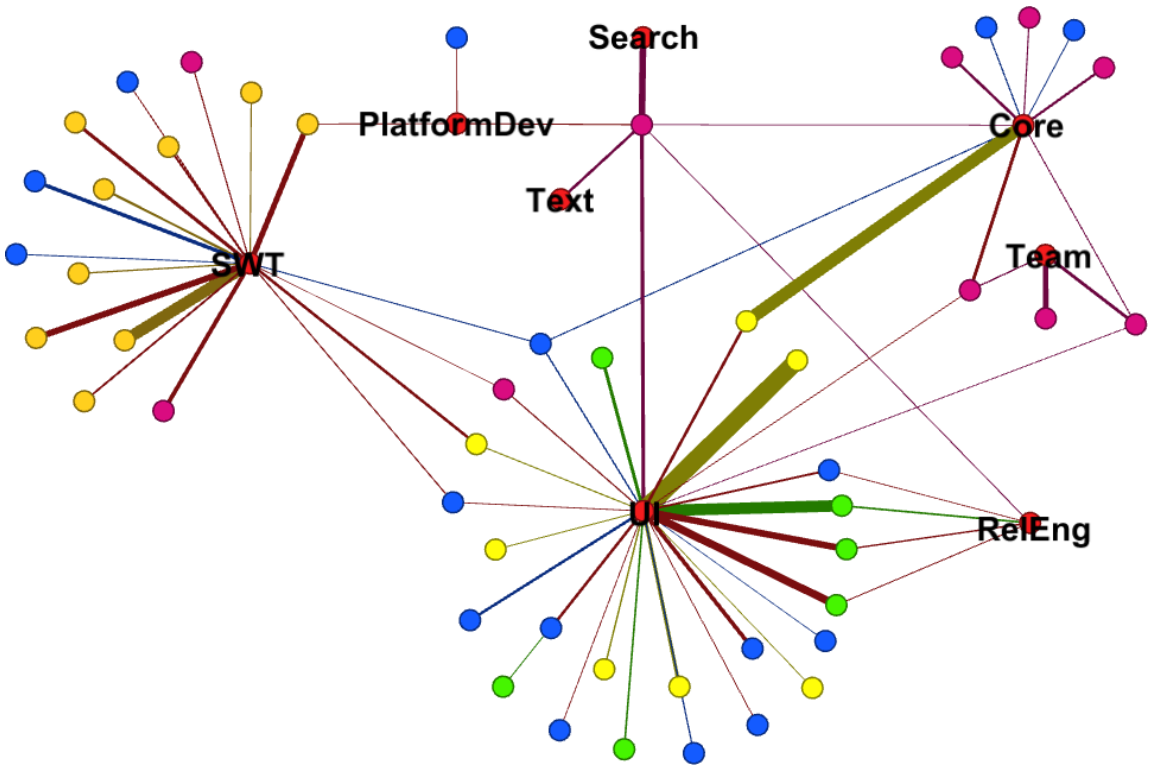


Figure 4.7: Eclipse Platform project - Social (email communication) network. From committers to mailing lists.

a tendency to actively participate in the lists to a greater extent that representatives of other technical clusters, which could be related to personality traits *Self-transcendence* (16.67%), *Structure* (16.00%), and *Conscientiousness* (69.56%).

4.2 Case Study - OpenStack Project

4.2.1 Preliminary data exploration

Results of preliminary exploration conducted over OpenStack project data are summarized in Table 4.6. The date range for which data were obtained is between May 27th, 2010 and February 05th, 2015.

No	Type of Service	Project / Subproject	Description	Committers	Commits	Mailing List Senders	Mailing List Messages
1	Core	Swift	The OpenStack Object Store project, known as Swift, offers cloud storage software so that lots of data can be stored and retrieved with a simple API.	197	4576	3367	72446
2		Keystone	Keystone is the identity service used by OpenStack for authentication (authN) and high-level authorization (authZ).	297	6974		
3		Nova	Nova is an OpenStack project designed to provide power massively scalable, on demand, self service access to compute resources.	827	34742		
4		Neutron	Neutron is an OpenStack project to provide "networking as a service" between interface devices (e.g., vNICs) managed by other Openstack services (e.g., nova).	447	9547		
5		Cinder	Cinder is a Block Storage service for OpenStack. It is designed to present storage resources to end users that can be consumed by the OpenStack Compute Project (Nova).	405	5535		
6		Glance	The Glance project provides a service where users can upload and discover data assets that are meant to be used with other services.	294	4458		
7	Optional	Horizon	Horizon is the canonical implementation of Openstack's Dashboard, which provides a web based user interface to OpenStack services including Nova, Swift, Keystone, etc.	412	6783		
8		Ceilometer	The Ceilometer project is a data collection service that provides the ability to normalise and transform data across all current OpenStack core components with work underway to support future OpenStack components.	208	3642		
9		Heat	Heat is the main project in the OpenStack Orchestration program. It implements an orchestration engine to launch multiple composite cloud applications based on templates in the form of text files that can be treated like code.	220	7470		

Table 4.6: OpenStack project - Number of registers.

4.2.2 Technical and personality groups

Again, looking at the point at which the SSE value changes significantly in the elbow curve for technical and personality clustering, the best candidates for technical clustering were $k_t = 5$, $k_t = 6$, and $k_t = 7$ and for personality clustering were again $k_p = 3$, $k_p = 4$, and $k_p = 5$. For convenience and to expedite the comparison and analysis with results obtained in the other case studies $k_t = 5$ was selected for technical clustering and $k_p = 3$ for personality clustering. At this point it should be clarified that these values of k were selected considering that the quality of the results are not going to be affected and seeking that the analysis can be done in an objective and reliable manner.

Figures 4.8 and 4.9 shows elbow curves used to select the k value for technical and personality clustering.

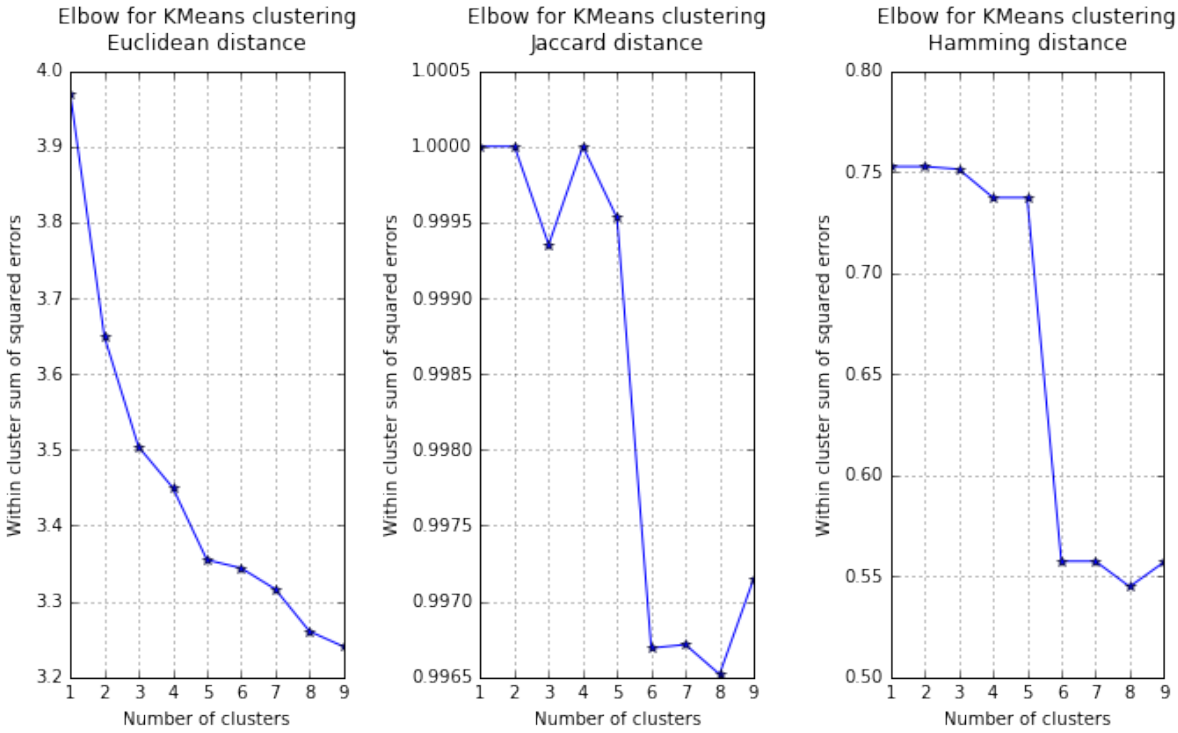


Figure 4.8: OpenStack project - Elbow curves for tecnical clustering

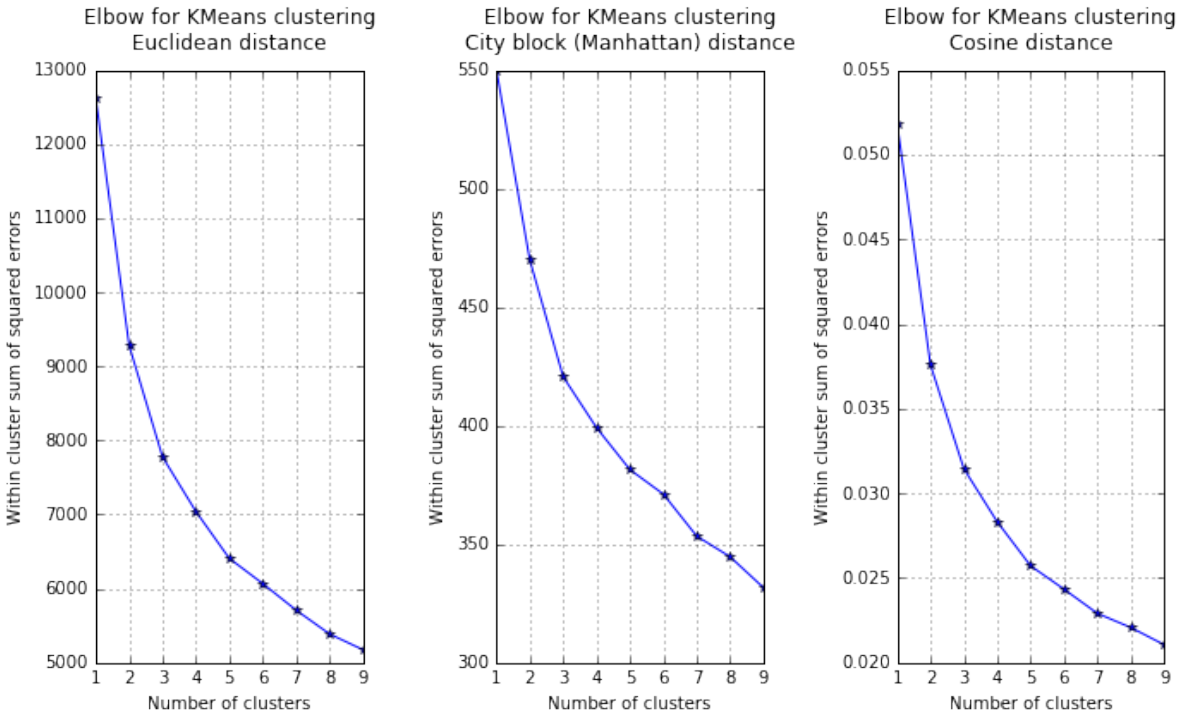


Figure 4.9: OpenStack project - Elbow curves for personality clustering

Tables 4.7 and 4.8 show the results for technical clustering. Same considerations as in the former case study were taken into account.

Technical cluster	**Number of committers**
0	112
1	1101
2	105
3	425
4	118
Total	1861

Table 4.7: OpenStack project - Results for technical clustering.

Table 4.9 shows, for each technical cluster, the number of committers who touched each project. As in the former case study, to understand the meaning of technical groups the results presented in Tables 4.8 and 4.9 were taken into account. In this case, committers from technical clusters 1, 3 and 4 contribute the most to Nova project, which is one of the oldest components of OpenStack, along with Swift project. In fact, there is great participation of technical cluster 1 (474 committers) and a high activity of the technical cluster 3 (0.63) with reference to this project. Committers from technical cluster 0 prefer to contribute to Neutron project, evidencing a share of 111 committers with 0.47 of activity, while cluster 2 is most active (0.51) in Horizon project with a share of 105 committers. Analyzing participation in other projects, it is observed that committers from cluster 0 tend to be more present in the Nova project (103), while committers from cluster 1 to 3 lean toward Swift project (432, 98, and 316 respectively). Furthermore, it is noticed some uniformity in cluster 4 as all the committers belonging to this group (118) contribute to Swift and Nova projects, with more activity in the latter project (0.34).

Technical cluster	Swift	Keystone	Nova	Neutron	Cinder	Glance	Horizon	Ceilometer	Heat
0	0.12	*	0.28	**0.47**	*	*	*	*	*
1	0.23	0.08	**0.28**	0.09	*	*	0.12	*	0.08
2	0.12	*	0.19	*	*	*	**0.51**	*	*
3	0.14	*	**0.63**	*	0.09	*	*	*	*
4	0.11	0.07	**0.34**	0.10	0.07	0.07	*	*	0.13

* *Values < 0.07*

Table 4.8: OpenStack project - Results for technical clustering. Averaged number of times that project directories have been touched by committers.

Technical cluster	Swift	Keystone	Nova	Neutron	Cinder	Glance	Horizon	Ceilometer	Heat
0	93	17	**103**	**111**	11	11	22	23	31
1	**432**	134	**474**	180	84	52	208	72	105
2	**98**	42	93	48	27	13	**105**	9	29
3	**316**	36	**424**	80	126	48	50	35	56
4	**118**	112	**118**	105	87	83	95	59	114

Table 4.9: OpenStack project - Number of committers touching projects in technical clusters.

Table 4.10 shows the results for personality clustering, and the heat map in Figure 4.10 depicts just the top-10 (the lowest) entropy values for the Big Five dimensions and facets, Needs, and Values (rows) by each personality cluster (columns).

Personality cluster	Number of committers
0	39
1	60
2	31
Total	130

Table 4.10: OpenStack project - Results for personality clustering.

As highlighted in the heat map (Figure 4.10), personality cluster 2 groups the committers with the highest scores in *Conservation* (80%), and the lowest values in *Assertiveness* (1.94%), *Gregariousness* (1.88%), *Friendliness* (1.87%), *Emotionality* (1.84%), and *Extraversion* (1.56%). The personality trait that characterizes cluster 1 by its moderately high value is *Self-transcendence* (57.7%), opposed to cluster 2 which has a relatively low value (7.09%) in this dimension of Human Values. Finally, personality traits standing out in cluster 0 are *Self-transcendence* (45,23%), *Altruism* (30.82%), *Assertiveness* (24.72%), and *Conservation* (20.41%), which are lower than those of cluster 2 but a little bit higher than those of cluster 1, as in the case of *Conservation* (20.41% for cluster 0 vs. 11.88% for cluster 1) and *Assertiveness* (24.72% for cluster 0 vs. 18.98 for cluster 1).

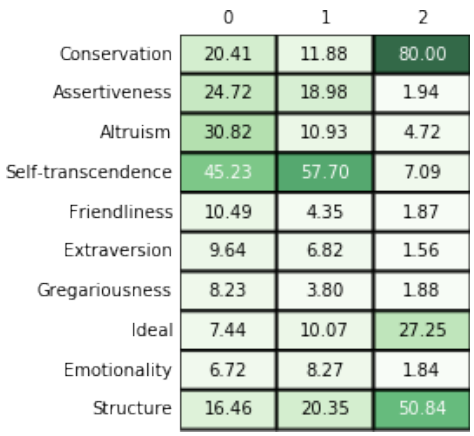


Figure 4.10: OpenStack project - Heat map of the most discriminative factors for the personality clustering.

4.2.3 Personality traits characterizing technical groups

Remaining the premise that each personality cluster has associated personality traits that characterize and distinguish its members, by computing the entropy for each of the Big Five dimensions and facets, Needs and Values, it is possible to determine which of these attributes provide more information or become a differentiating factor when analyzing the technical groups, depending on the personality traits of the committers who are part of them, allowing to characterize the group or groups in which it is presented, and in which it is necessary to focus on when making an analysis of each cluster.

Knowing the technical cluster where each committer belongs and the personality traits for committers belonging to each technical cluster, it is possible to compute the centroids of personality traits for each technical cluster. Again, to facilitate and improve interpretability, Figure 4.11 shows just the 10 lowest values and the 10 highest values of entropy for the Big Five dimensions and facets, Needs, and Values of personality centroids computed by averaging the values of the personality traits of committers in each technical cluster.

From the results reported in Figure 4.11 and Table 4.8, **RQ2** can be answered. Personality traits scoring higher ($\geqslant$ 80%) and with nearly similar values through all technical clusters (e.g. *Intellect*, *Openness*, *Liberalism*, *Imagination*, *Adventurousness*, *Achievement striving*, *and Cautiousness*) could be considered as personality factors characterizing the project, i.e. people involved in the project will most likely exhibit high values in these personality traits. On the other hand, personality traits scoring lower ($\leqslant$ 25%) allow to identify relationships with the technical aspects, differentiating personality features among the different technical clusters.

Figure 4.12 summarizes personality traits by technical cluster allowing to visualize which personality traits are more representative in each technical cluster. From this representation can be noticed the dominant facets for the different technical clusters. Looking for relationships between personality traits characterizing technical clusters and the involvement of the committers in the OpenStack projects, can be noticed that committers belonging to technical cluster 2, who exhibit the lowest values of personality traits shown in Figure 4.12, contribute actively to the Horizon project which is an Optional (Non-Core) Service. In contrast, committers belonging to technical cluster 0, who exhibit the higher values in most personality traits (excluding *Conservation*, *Morality*, *Gregariousness*, and *Self-expression*), are more involved in the Neutron project which

	0	1	2	3	4
Conservation	34.43	37.23	12.44	36.38	21.57
Morality	25.29	20.39	10.22	28.14	18.46
Structure	30.07	29.45	12.00	29.29	20.36
Gregariousness	6.14	4.52	2.56	4.48	6.32
Friendliness	8.79	5.29	4.11	4.81	7.07
Self-expression	7.00	8.58	3.56	6.24	6.04
Trust	53.43	49.39	23.22	48.10	40.75
Stability	10.07	10.06	4.89	7.10	7.57
Activity level	29.64	22.35	13.33	26.05	25.21
Altruism	24.79	13.55	14.89	13.33	17.93

(a) The 10 lowest values of entropy (most discriminative personality factors for technical clusters) for the Big Five dimensions and facets, Needs, and Values of personality centroids.

	0	1	2	3	4
Intellect	96.64	97.45	97.44	97.00	96.39
Openness	94.93	95.84	96.22	94.90	94.86
Liberalism	95.86	95.61	96.11	96.29	94.57
Imagination	94.21	95.84	96.33	95.43	94.11
Adventurousness	89.79	90.55	85.33	88.14	87.25
Achievement striving	79.64	81.06	77.44	76.57	77.29
Cautiousness	87.57	93.13	88.56	88.76	88.43
Conscientiousness	64.36	71.84	64.78	64.86	65.96
Self-enhancement	83.50	73.65	80.89	76.71	73.04
Self-efficacy	65.00	64.48	64.67	57.57	68.04

(b) Top 10 highest values of entropy (personality factors characterizing the project) for the Big Five dimensions and facets, Needs, and Values of personality centroids.

Figure 4.11: OpenStack project - Heat map of personality centroids for each technical cluster.

is a Core Service. It should be clarified that Swift and Nova projects were not taken into account for the analysis as these are practically common to all technical clusters.

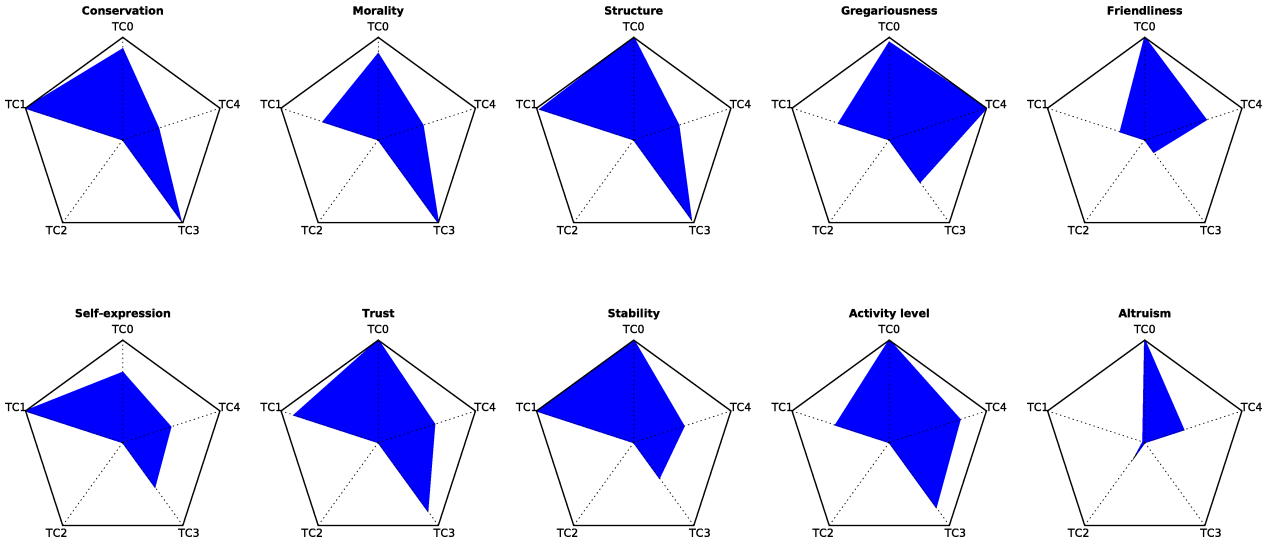


Figure 4.12: OpenStack project - Radar charts of personality traits by technical cluster (TC).

Observing the distribution of personality traits through each of the technical clusters, some personality traits exhibit a behavioral pattern as can be seen in Figure 4.13. Clearly it shows that *Trust* facet, belonging to *Agreeableness* dimension, is a trait that stands out in all technical clusters while *Gregariousness* (*Extraversion* facet), *Friendliness* (*Extraversion* facet), and *Self-expression* (Need) are features practically absent from the personality traits of the OpenStack project committers. On the other hand, *Activity level* (*Extraversion* facet) is a trait that is present in a proportion nearly uniform in all the technical clusters. It does not stands out nor is entirely absent.

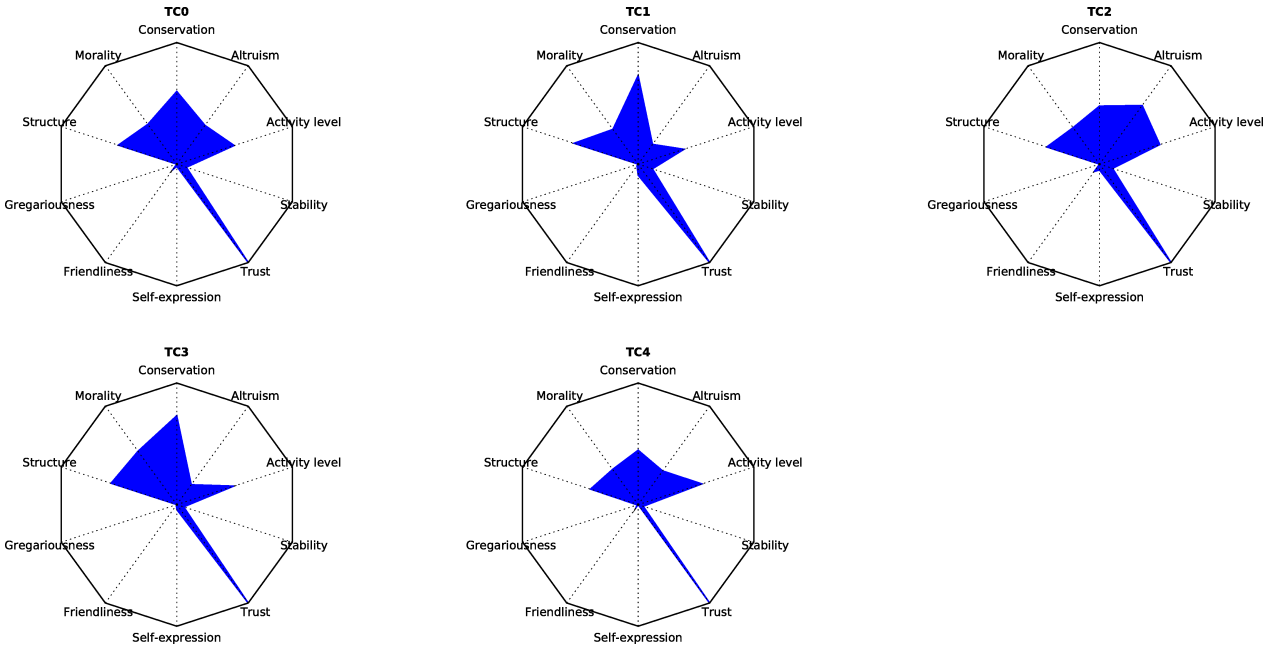


Figure 4.13: OpenStack project - Radar chart of technical clusters characterized by personality traits.

4.2.4 Visualizing the social network - from committers to mailing lists

Because of OpenStack project does not have a mailing list for each service (Swift, Keystone, Nova, Neutron, Cinder, Glance, Horizon, Ceilometer, Heat) but concentrates e-mail communication in mailing lists that deal with specific topics (e.g. Openstack List - The OpenStack General mailing list; Community List - The OpenStack Community team is the main contact point for anybody running a local OpenStack Group; Foundation List - General discussion list for activities of the OpenStack Foundation; OpenStack-dev List - OpenStack Development Mailing List; OpenStack-docs List - OpenStack Documentation Mailing List; among others[9]), in this case study the visualization of the social network show just nodes representing committers connected to nodes representing mailing lists without a clearly recognizable behavioral pattern, so it does not provide additional nor relevant information that might be useful for analysis.

The graph in Figure 4.14 shows committers and mailing lists (OpenStack, OS_stablemaint, OS_announce, OS_infra, OS_operators, OS_tc, OS_hpc, OS_docs, OS_qa, i.e., red circles) as nodes. The color of the nodes representing committers corresponds to the technical cluster to which the committer belongs to (cluster 0: blue, cluster 1: yellow, cluster 2: orange, cluster 3: purple, cluster 4: green).

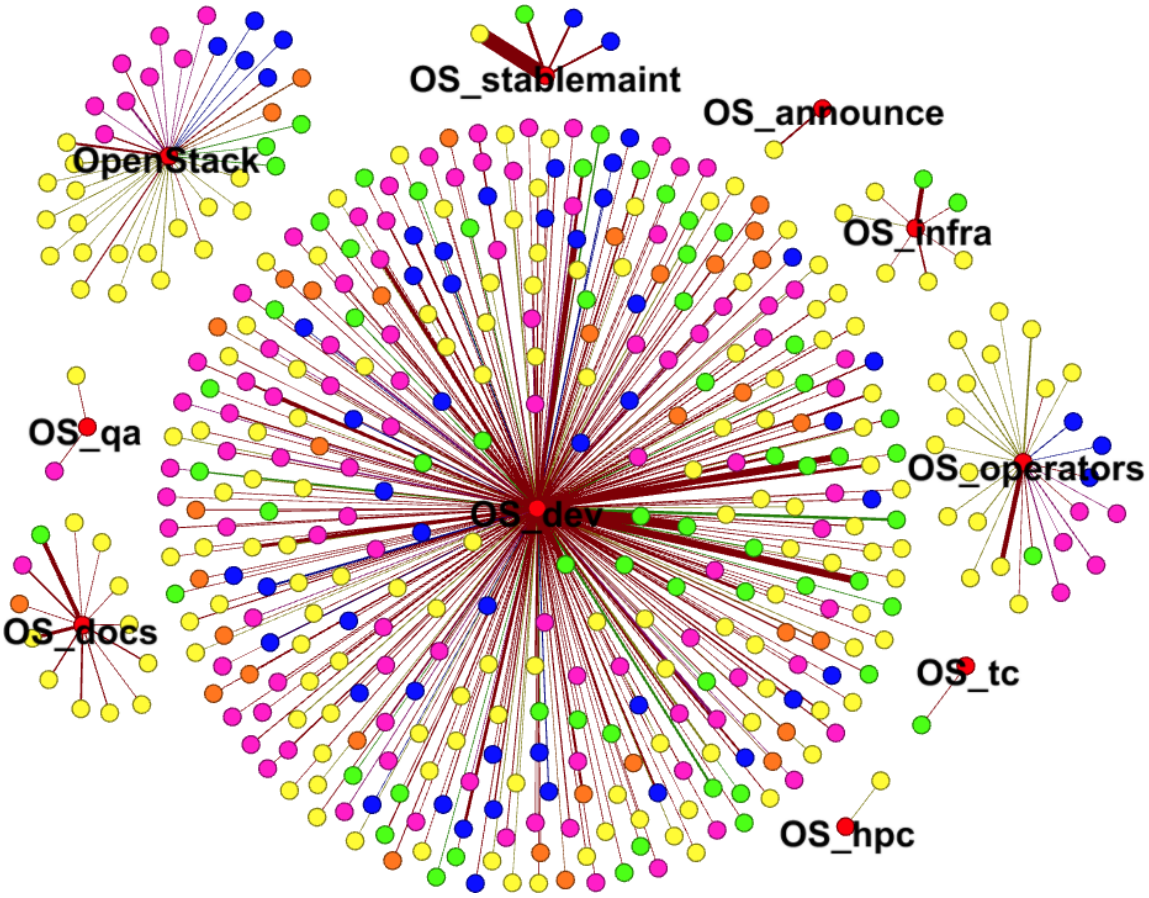


Figure 4.14: OpenStack project - Social (email communication) network. From committers to mailing lists.

[9] lists.openstack.org/cgi-bin/mailman/listinfo

4.3 Case Study - Xen Project

4.3.1 Preliminary data exploration

Results of preliminary exploration conducted over Xen project data are summarized in Table 4.11. The date range for which data were obtained is between July 18th, 2007 and March 04th, 2016.

No	Project / Subproject	Description	Committers	Commits	Mailing List Senders	Mailing List Messages
1	Xen	Xen Project is a hypervisor using a microkernel design, providing services that allow multiple computer operating systems to execute on the same computer hardware concurrently.	169	17640	3762	238800
2	Xapi	Manages a cluster of Xen hosts, co-ordinating access to network and storage.	98	8723	894	7447
3	Xenopsd	A low-level "domain manager" which takes care of creating, suspending, resuming, migrating, rebooting domains by interacting with Xen via libxc and libxl.	25	1209		
4	Xcp-rrdd	A performance counter monitoring daemon which aggregates "datasources" defined via a plugin API and records history for each.	68	5729		
5	Xcp-networkd	A host network manager which takes care of configuring interfaces, bridges and OpenVSwitch instances.	69	5714		
6	Squeezed	It is a single host ballooning daemon which "balances" memory between running VMs.	68	5542		
7	SM	Storage Manager plugins which connect Xapi's internal storage interfaces to the control APIs of external storage systems.	10	489		
8	MirageOS	Mirage is an exokernel (also called a Cloud Operating System) for constructing secure, high-performance network applications across a variety of cloud computing, embedded and mobile platforms.	21	988	223	5222
9	Linux-2.6	The Linux kernel is a Unix-like computer operating system kernel. Original release date: 17 December 2003 Maintainer: Linus Torvalds	54	3255	905	23030

Table 4.11: Xen project - Number of registers.

4.3.2 Technical and personality groups

As in the above case studies, looking at the point at which the SSE value changes significantly in the elbow curve the best candidates were again for technical clustering $k_t = 5$, $k_t = 6$, and $k_t = 7$ and for personality clustering $k_p = 3$, $k_p = 4$, and $k_p = 5$. For convenience and to expedite the comparison and analysis with results obtained in the other case studies $k_t = 5$ was selected for technical clustering and $k_p = 3$ for personality clustering. It bears repeating that these values of k were selected considering that the quality of

the results are not going to be affected and seeking that the analysis can be done in an objective and reliable manner.

Figures 4.15 and 4.16 shows elbow curves used to select the k value for technical and personality clustering.

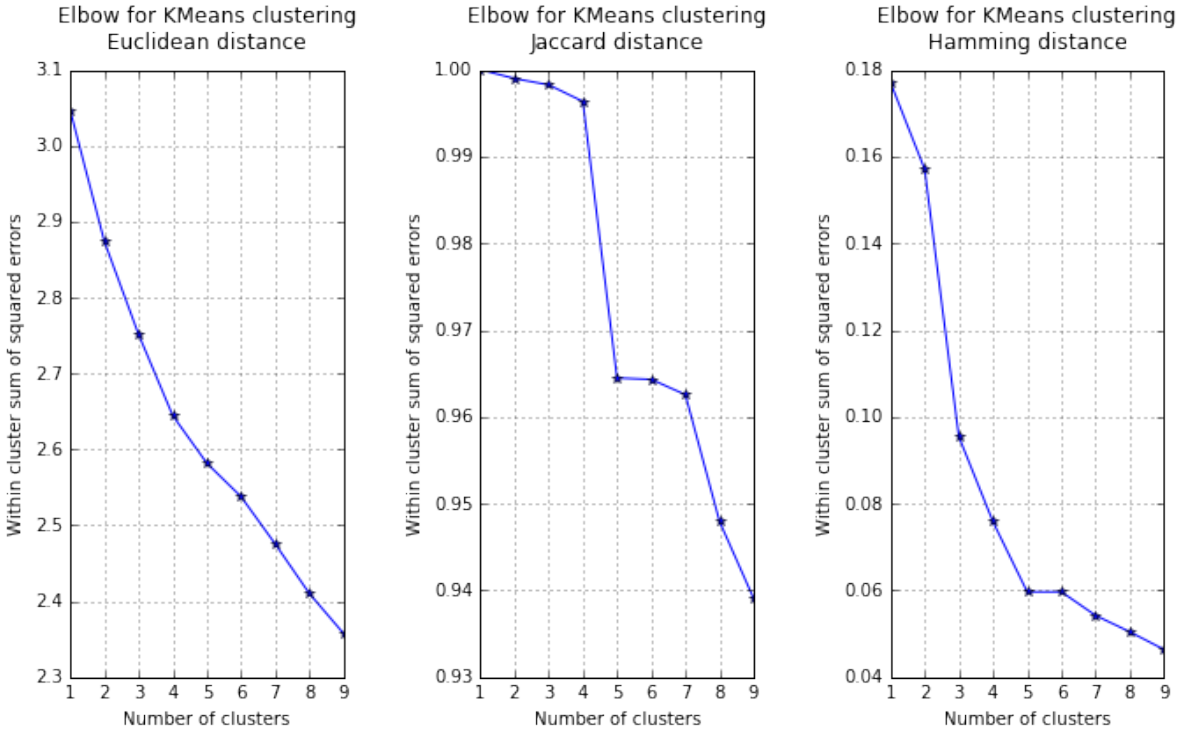


Figure 4.15: Xen project - Elbow curves for tecnical clustering

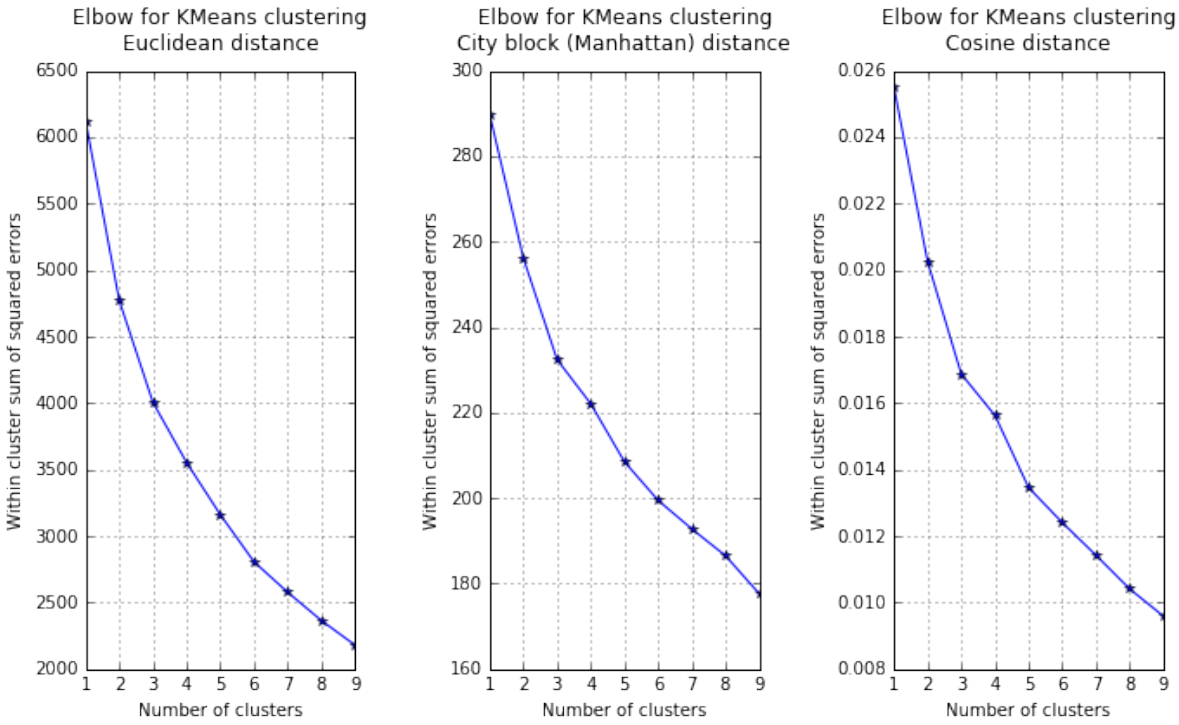


Figure 4.16: Xen project - Elbow curves for personality clustering

Tables 4.12 and 4.13 show the results for technical clustering. Same considerations as in the former case studies were taken into account.

Technical cluster	**Number of committers**
0	194
1	93
2	18
3	14
4	10
Total	329

Table 4.12: Xen project - Results for technical clustering.

Table 4.14 show, for each technical cluster, the number of committers who touched each project. As in the former case studies, to understand the meaning of technical groups the results presented in Tables 4.13 and 4.14 were taken into account. In this case, committers from technical clusters 1, 2 and 4 contribute the most to Xen and Linux-2.6 projects, which can be considered as the central projects of this case study. Actually, there is great participation of technical cluster 1 (93 and 33 committers contributing to Xen and Linux-2.6 projects respectively); a high activity of the technical clusters 1 and 4 (0.78 and 0.64 respectively) with reference to the Xen project and a considerable activity of the technical cluster 2 (0.51) with reference to the Linux-2.6 project. Committers from technical clusters 0 and 3 prefer to contribute to Xen-Api project, evidencing a share of 69 (technical cluster 0) and 14 (technical cluster 3) committers with 0.33 (technical cluster 0) and 0.49 (technical cluster 3) of activity. Analyzing participation in other projects, it is observed that committers from cluster 0 tend to be more present in the Xen project (59), while committers from cluster

Technical cluster	Xen	Xen-Api	Xcp-Networkd	Xenopsd	Xcp-rrdd	Squeezed	SM	Mirage	Mirage-www	Linux-2.6
0	**0.25**	**0.33**	*	0.10	*	*	*	*	0.11	0.09
1	**0.78**	*	*	*	*	*	*	*	*	**0.20**
2	**0.41**	*	*	*	*	*	*	*	*	**0.51**
3	0.09	**0.49**	*	**0.12**	*	*	*	*	*	*
4	**0.64**	*	*	*	*	*	*	*	*	**0.32**

* *Values < 0.07*

Table 4.13: Xen project - Results for technical clustering. Averaged number of times that project directories have been touched by committers.

3 lean toward Xenopsd project (14). Furthermore, it is noticed some uniformity in technical clusters 2 and 4 as all the committers belonging to these groups (18 and 10 respectively) contribute to Xen and Linux-2.6 projects, with more activity in the Xen project (0.64) for cluster 4 and more activity in the Linux-2.6 project (0.51) for cluster 2, while all the committers belonging to technical cluster 3 (14) contribute to Xen-Api and Xenopsd projects with more activity in the former project (0.49).

Technical cluster	Xen	Xen-Api	Xcp-Networkd	Xenopsd	Xcp-rrdd	Squeezed	SM	Mirage	Mirage-www	Linux-2.6
0	**59**	**69**	9	35	4	4	7	10	26	22
1	**93**	4	0	3	0	0	0	1	0	**33**
2	**18**	2	0	2	0	0	0	2	2	**18**
3	7	**14**	11	**14**	7	11	2	4	6	3
4	**10**	5	1	2	0	0	1	0	0	**10**

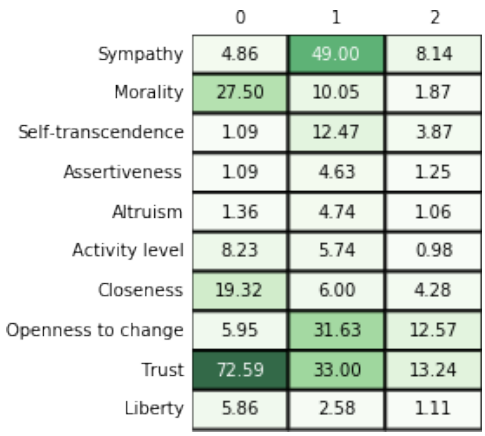
Table 4.14: Xen project - Number of committers touching projects in technical clusters.

Table 4.15 shows the results for personality clustering, and the heat map in Figure 4.17 depicts just the top-10 (the lowest) entropy values for the Big Five dimensions and facets, Needs, and Values (rows) by each personality cluster (columns).

Personality cluster	Number of committers
0	22
1	19
2	108
Total	149

Table 4.15: Xen project - Results for personality clustering.

As highlighted in the heat map (Figure 4.17), personality cluster 0 groups the committers with the highest scores in *Trust* (72.59%), and the lowest values in *Openness to change (5.95%), Sympathy* (4.86%), *Self-transcendence* (1.09%), *and Assertiveness* (1.09%). Personality traits characterizing cluster 1 by its moderately high values are *Sympathy* (49%), *Trust* (33%), and *Openness to change* (31.63%). With reference



	0	1	2
Sympathy	4.86	49.00	8.14
Morality	27.50	10.05	1.87
Self-transcendence	1.09	12.47	3.87
Assertiveness	1.09	4.63	1.25
Altruism	1.36	4.74	1.06
Activity level	8.23	5.74	0.98
Closeness	19.32	6.00	4.28
Openness to change	5.95	31.63	12.57
Trust	72.59	33.00	13.24
Liberty	5.86	2.58	1.11

Figure 4.17: Xen project - Heat map of the most discriminative factors for the personality clustering.

to personality clusters 0 and 1, cluster 2 characterizes by lower values in most of their personality traits showed in Figure 4.17, not exceeding 13% as in the case of *Trust* (13.24%), *Openness to change* (12.57%), and *Sympathy* (8.14%). Furthermore, the lower value among all the personality traits in Figure 4.17 is for *Activity level* (0.98%) (*Extraversion* facet) in cluster 2.

4.3.3 Personality traits characterizing technical groups

Applying the next step of the proposed methodology as was done in the previous case studies, the entropy for each of the Big Five dimensions and facets, Needs and Values was computed to determine which of these attributes provide more information or become a differentiating factor when analyzing the technical groups. Centroids of personality traits for each technical cluster were obtained using the information about the technical cluster where each committer belongs and the personality traits for committers belonging to each technical cluster. Figure 4.18 shows just the 10 lowest values and the 10 highest values of entropy for the Big Five dimensions and facets, Needs, and Values of personality centroids computed by averaging the values of the personality traits of committers in each technical cluster.

Based on the results reported in Figure 4.18 and Table 4.13 to answer **RQ2** for this case study, personality traits scoring higher ($\geqslant$ 80%) and with nearly similar values through all technical clusters (e.g. *Intellect*, *Liberalism*, *Imagination*, *Openness, Cautiousness, Adventurousness*, *and Achievement striving*) could be considered as personality factors characterizing the project, i.e. people involved in the project will most likely exhibit high values in these personality traits. By observing the personality traits scoring higher ($\geqslant$ 80%) in all case studies presented in this work, can be said that those are practically the same, differing mainly in the percentage in which these are present in each cluster. This may correspond to a recurring pattern in FLOSS projects. On the other hand, personality traits scoring lower ($\leqslant$ 25%) allow to identify relationships with the technical aspects, differentiating personality features among the different technical clusters.

Figure 4.19 summarizes personality traits by technical cluster allowing to visualize which personality traits are more representative in each technical cluster. From this representation can be noticed the dominant facets for the different technical clusters. Looking for relationships between personality traits characterizing technical clusters and the involvement of the committers in the Xen projects, can be noticed that committers belonging to technical cluster 3, who exhibit the highest values in most of the personality traits shown

	0	1	2	3	4
Excitement	1.92	1.41	1.60	1.60	10.30
Liberty	2.02	1.16	1.40	1.60	8.10
Activity level	3.63	0.91	1.73	3.40	9.20
Morality	12.31	2.54	2.60	5.40	2.00
Self-transcendence	4.38	4.09	5.27	16.20	3.00
Altruism	2.02	1.09	1.47	3.80	1.10
Assertiveness	2.02	1.38	1.40	4.00	1.10
Orderliness	3.67	1.80	2.53	5.60	1.60
Sympathy	15.17	10.13	14.20	26.40	6.10
Closeness	8.04	4.96	4.33	4.40	13.30

(a) The 10 lowest values of entropy (most discriminative personality factors for technical clusters) for the Big Five dimensions and facets, Needs, and Values of personality centroids.

	0	1	2	3	4
Intellect	99.17	99.88	99.67	99.00	99.70
Liberalism	99.62	99.96	99.87	98.80	99.90
Imagination	99.50	99.96	99.87	98.80	100.00
Openness	98.38	99.77	99.67	98.40	99.70
Cautiousness	93.44	95.45	94.33	92.40	95.10
Adventurousness	96.17	98.45	98.00	95.40	98.10
Achievement striving	83.50	86.48	86.20	83.60	85.40
Conscientiousness	73.44	78.34	80.40	74.00	73.50
Self-discipline	52.79	56.64	59.87	49.80	54.70
Self-enhancement	78.10	73.41	68.20	62.60	70.60

(b) Top 10 highest values of entropy (personality factors characterizing the project) for the Big Five dimensions and facets, Needs, and Values of personality centroids.

Figure 4.18: Xen project - Heat map of personality centroids for each technical cluster.

in Figure 4.19, such as *Self-transcendence*, *Altruism*, *Assertiveness*, *Orderliness* and *Sympathy*, contribute actively to the Xen-Api and Xenopsd projects, unlike committers belonging to the other technical clusters (0, 1, 2, and 4) who contributes mainly to Xen project and secondly to Linux-2.6 project. Committers belonging to technical clusters 1 and 2 exhibit similar lower values of personality traits, mostly in *Excitement*, *Liberty*, *Morality*, *Altruism*, *Assertiveness*, and *Closeness* and contributes to the same projects (Xen and Linux-2.6 projects), but despite of the marked differences observed among personality traits characterizing technical cluster 4 with respect to those characterizing technical clusters 1 and 2, mainly in *Excitement*, *Liberty*, *Activity level*, and *Closeness*, seems that committers belonging to one of these technical clusters (1, 2, and 4) have managed to engage and collaborate together as they work and contribute to the same projects (Xen and Linux-2.6 projects), although in varying proportions.

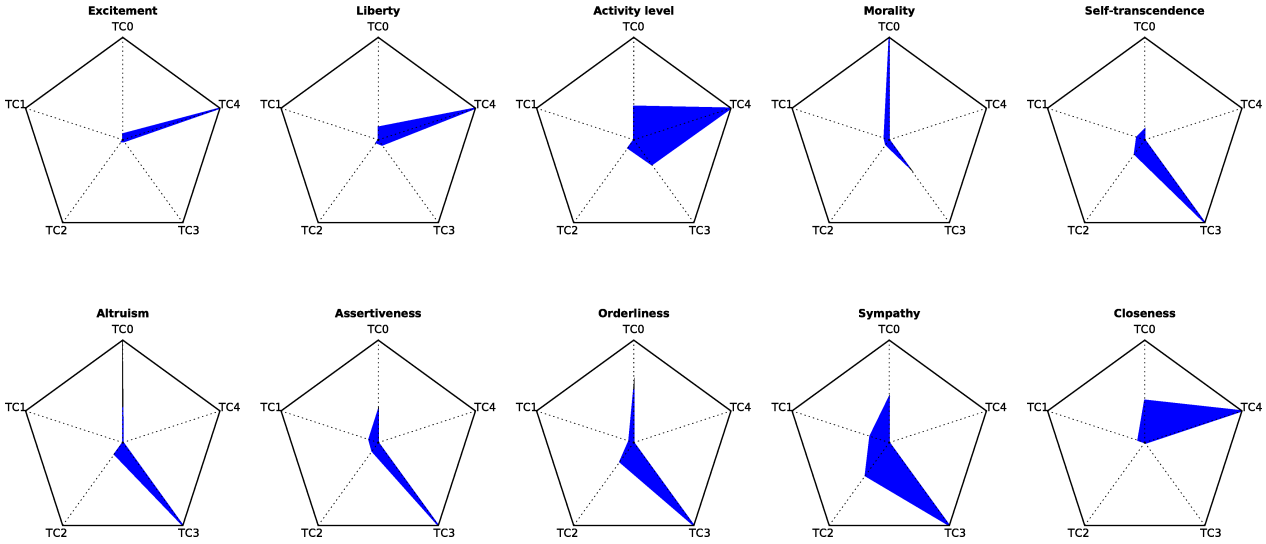


Figure 4.19: Xen project - Radar charts of personality traits by technical cluster (TC).

Observing the distribution of personality traits through each of the technical clusters, some personality traits exhibit a behavioral pattern as can be seen in Figure 4.20. Clearly it shows that *Sympathy* facet, belonging to *Agreeableness* dimension, is a trait that stands out in all technical clusters, except for cluster 4 in which the most protruding personality trait is Closeness (Need) while *Excitement* (Need), *Altruism* (*Agreeableness* facet), and *Liberty* (Need) are features practically absent from the personality traits in most of the Xen project committers, particularly those belonging to technical clusters 0 to 3. On the other hand, *Morality* (*Agreeableness* facet) and *Self-transcendence* (Value) are traits that not stand out nor are entirely absent.

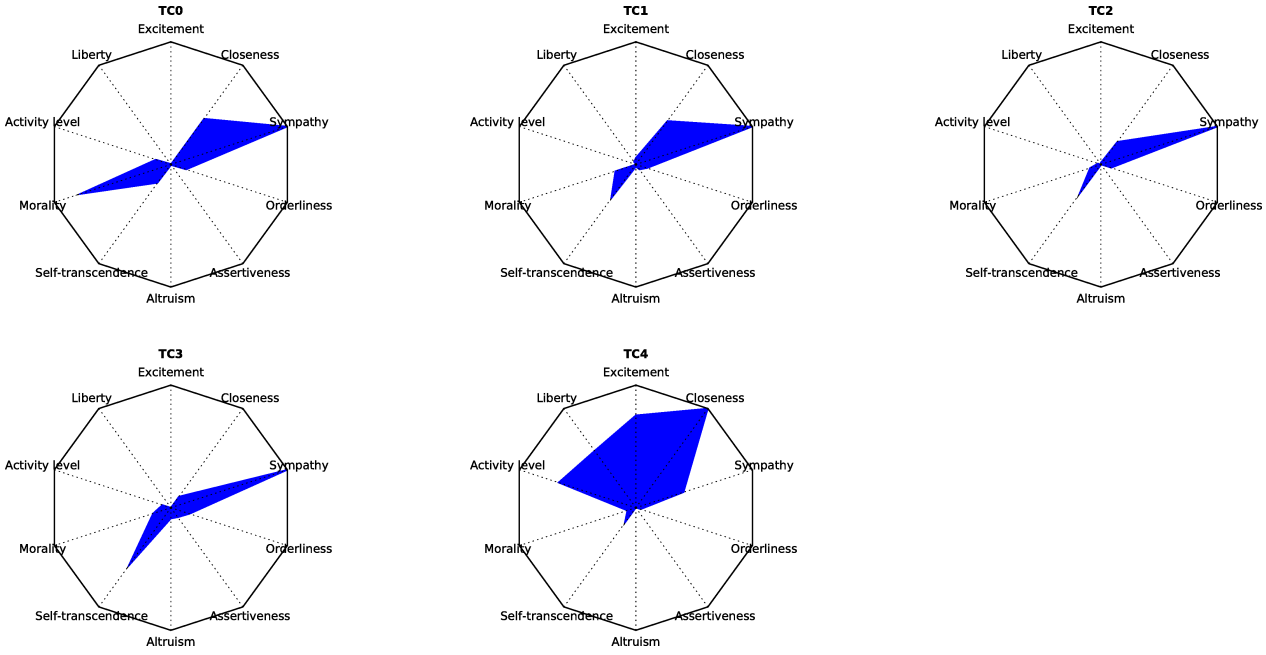


Figure 4.20: Xen project - Radar chart of technical clusters characterized by personality traits.

4.3.4 Visualizing the social network - from committers to mailing lists

As in the first case study, using the e-mails sent by committers to the Xen project mailing lists, a graph representing e-mail communications was built. The graph in Figure 4.21 shows committers and mailing lists (XenDevel, XenUsers, XenApi, MirageOSDevel, LinuxKernel, i.e., red circles) as nodes. The thickness of the edge between a committer and a mailing list represents the amount of emails sent by the committer to the list. Additionally, the color of the nodes representing committers corresponds to the technical cluster to which the committer belongs to (cluster 0: blue, cluster 1: yellow, cluster 2: orange, cluster 3: purple, cluster 4: green). Only committers that have sent more than 10 e-mails to any of the lists were taken into account. In the graph is evident the majority of representatives belongs to cluster 0 (blue circles).

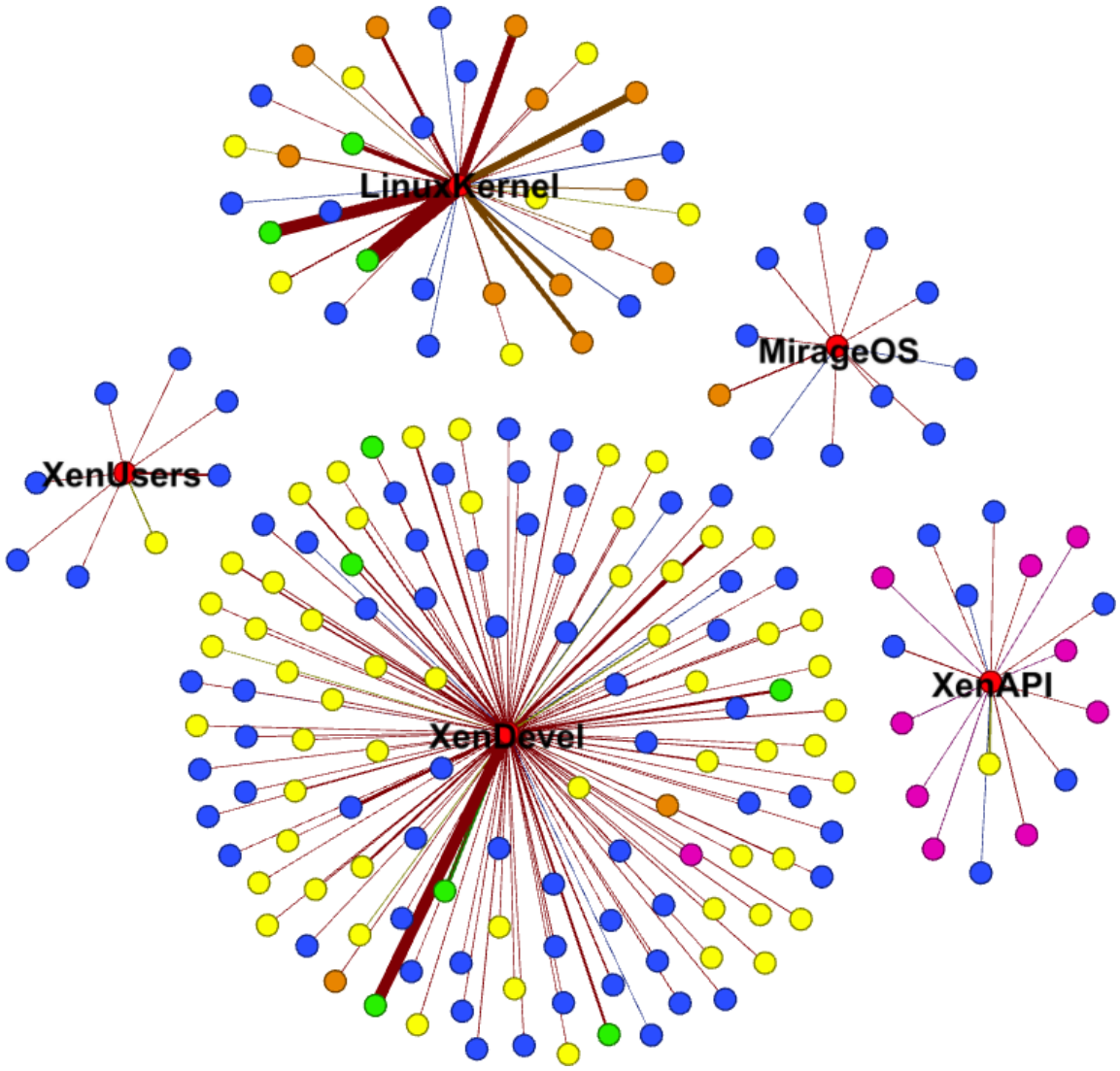


Figure 4.21: Xen project - Social (email communication) network. From committers to mailing lists.

Figure 4.21 reveal there is no direct communication among committers and there are no emails linking mailing lists of each project. Apparently, communication only occurs through mailing lists, which suggests that members of the Xen project community meet the Open Source axiom stated in their mailing list informative page[10], which says:

[10] www.xenproject.org/help/mailing-list.html

"If it hasn't happened in an email list, it hasn't happened" – Open Source axiom

Figure 4.21 help to answer **RQ3** for this case study. Committers belonging to technical groups 0 (blue circles) and 1 (yellow circles) are those distributed through all mailing lists, except MirageOS in the case of group 1. This could be because of the ranking they have in the combination of personality traits such as *Sympathy* (15.17% for cluster 0, and 10.13% for cluster 1), and *Self-enhancement* (78.10% for cluster 0, and 73.41% for cluster 1).

Figure 4.21 also shows that some representatives (3 out of 9) of technical cluster 4 (green circles) have a tendency to actively participate in the lists to a greater extent that representatives of other technical clusters, which could be related to personality trait *Activity level* (9.20%).

4.4 Case Study - Wikimedia Project

4.4.1 Preliminary data exploration

Results of preliminary exploration conducted over Wikimedia project data are summarized in Table 4.16. The date range for which data were obtained is between April 14th, 2003 and December 14th, 2015.

No	Project / Subproject	Descripton	Committers	Commits	Mailing List Senders	Mailing List Messages
1	apps-firefox-wikipedia	Wikipedia Mobile for Firefox OS	15	134	3584	110587
2	apps-ios-wikipedia	Wikipedia IOS app	16	1171		
3	apps-android-wikipedia	Wikipedia app for Android	27	2673		
4	wikimania-scholarships	Wikimania Scholarships Application. Collect information from scholarship applicants and provide a review management workflow for processing the applications.	23	518		
5	analytics-wikistats	Wikimedia Statistics Source code for stats.wikimedia.org	14	786		
6	labs-toollabs	Tool Labs tools and configuration	16	123		
7	apps-android-commons	The Wikimedia Commons Android App	15	587	479	7466
8	apps-ios-commons	Wikimedia Commons uploader app for iOS	15	806		
9	mediawiki-core	MediaWiki core	447	79623	4822	49135
10	mediawiki-extensions-AccessControl	MediaWiki extension AccessControl	11	47		
11	mediawiki-extensions-AccountInfo	MediaWiki extension AccountInfo	12	160		
12	mediawiki-extensions-ActivityMonitor	MediaWiki extension ActivityMonitor	8	69		
13	mediawiki-extensions-CSS	MediaWiki extension CSS	18	139		
14	mediawiki-extensions-VisualEditor	Visual editor for Wikitext documents.	86	11468		
15	mediawiki-extensions-Graph	Creates data-driven dynamic graphs embedded in wiki pages	22	313		
16	mediawiki-extensions-Wikidata	MediaWiki extension Wikidata	15	1259		
17	mediawiki-extensions-WikiLove	MediaWiki extension WikiLove	46	1027		
18	labs-maps	Maps and OpenStreetMap tools	2	11	120	1392

Table 4.16: Wikimedia project - Number of registers.

4.4.2 Technical and personality groups

As in the above case studies, looking at the point at which the SSE value changes significantly in the elbow curve for technical and personality clustering, the best candidates for technical clustering were $k_t = 4$, $k_t = 5$, and $k_t = 6$ and for personality clustering were again $k_p = 3$, $k_p = 4$, and $k_p = 5$. For convenience and to expedite the comparison and analysis with results obtained in the other case studies$k_t = 5$ was

selected for technical clustering and $k_p = 3$ for personality clustering. It is recalled that values of k were selected considering that the quality of the results are not going to be affected and seeking that the analysis can be done in an objective and reliable manner.

Figures 4.22 and 4.23 shows elbow curves used to select the k value for technical and personality clustering.

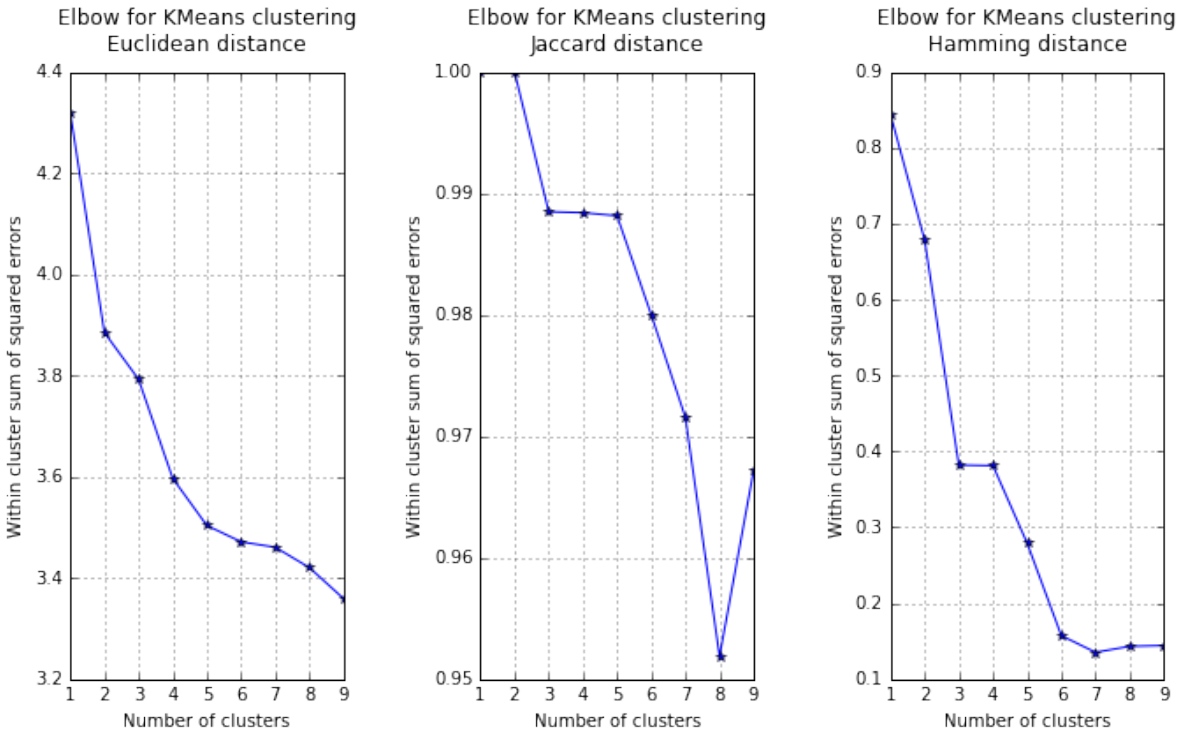


Figure 4.22: Wikimedia project - Elbow curves for tecnical clustering

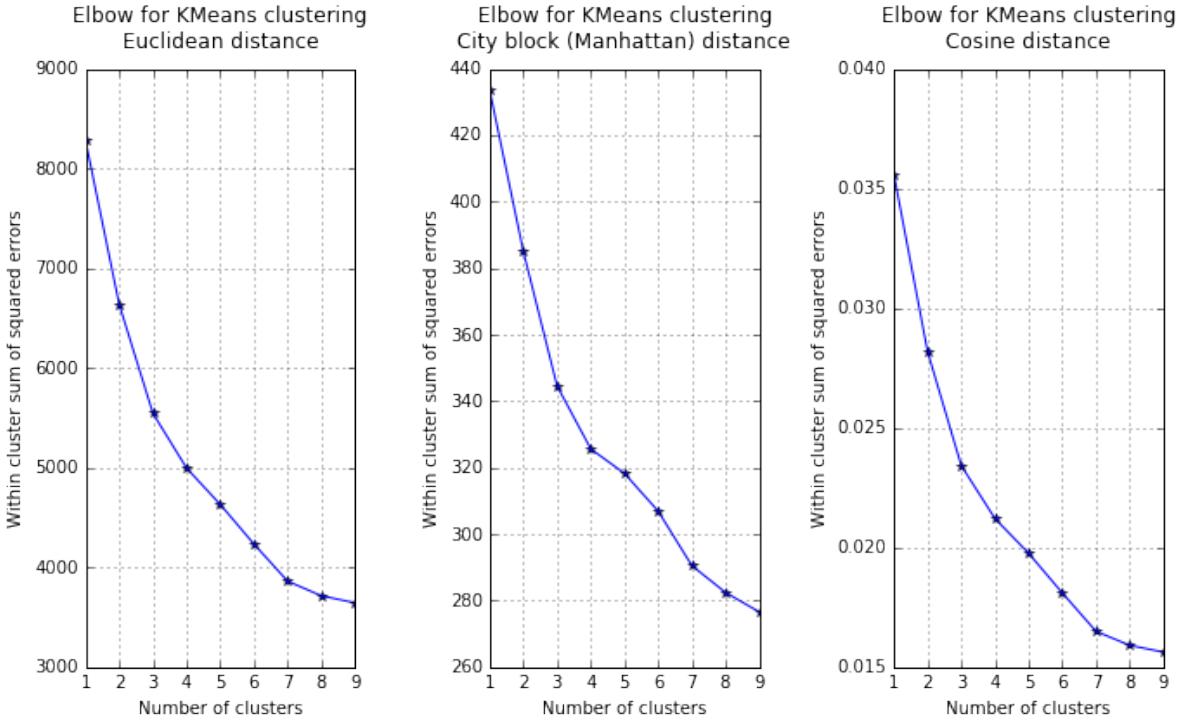


Figure 4.23: Wikimedia project - Elbow curves for personality clustering

Tables 4.17 and 4.18 show the results for technical clustering. Same considerations as in the former case studies were taken into account.

Technical cluster	**Number of committers**
0	334
1	5
2	33
3	20
4	119
Total	511

Table 4.17: Wikimedia project - Results for technical clustering.

Table 4.19 show, for each technical cluster, the number of committers who touched each project. As in the former case studies, to understand the meaning of technical groups the results presented in Tables 4.18 and 4.19 were taken into account. In this case, committers from technical clusters 0, 2, 3 and 4 contribute the most to Mediawiki-Core and Wikimedia-Wikimania-Scholarships projects, which can be considered as the central projects of this case study. Actually, there is great participation of technical clusters 0 and 4 (230 committers belonging to cluster 0 and 119 committers belonging to cluster 4 contributing to Mediawiki-Core project, and 89 committers belonging to cluster 0 and 118 committers belonging to cluster 4 contributing to Wikimedia-Wikimania-Scholarships project); a high activity of the technical clusters 3 and 4 (0.55 and 0.57 respectively) with reference to the Mediawiki-Core project and a considerable activity of the technical cluster 2 (0.48) with reference to the same project. Committers from technical cluster 1 prefer to contribute to the Mediawiki-Extensions-Wikidata project, evidencing a share of 5 committers with 0.84 of activity.

Technical cluster	Apps-Android-Wikipedia	Mediawiki-Core	Mediawiki-Extensions-VisualEditor	Mediawiki-Extensions-Wikidata	Wikimedia-Wikimania-Scholarships	Wikipedia-ios
0	0.10	**0.40**	*	*	**0.17**	0.08
1	*	**0.08**	*	**0.84**	*	*
2	*	**0.48**	0.13	0.11	**0.14**	*
3	*	**0.55**	*	0.14	**0.17**	*
4	*	**0.57**	*	0.10	**0.28**	*

* *Values < 0.07*

Table 4.18: Wikimedia project - Results for technical clustering. Averaged number of times that project directories have been touched by committers.

Furthermore, it is noticed some uniformity in technical clusters 1 and 3 as all the committers belonging to these groups (5 and 20 respectively) contributes to the Mediawiki-Core, Mediawiki-Extensions-Wikidata, and Wikimedia-Wikimania-Scholarships projects, with more activity in the Mediawiki-Extensions-Wikidata project (0.84) for cluster 1 and more activity in the Mediawiki-Core project (0.55) for cluster 3.

Technical cluster	Apps-Android-Wikipedia	Mediawiki-Core	Mediawiki-Extensions-VisualEditor	Mediawiki-Extensions-Wikidata	Wikimedia-Wikimania-Scholarships	Wikipedia-ios
0	11	**230**	21	35	**89**	37
1	0	**5**	0	**5**	**5**	**5**
2	8	**33**	24	31	**32**	23
3	13	**20**	12	**20**	**20**	11
4	3	**119**	9	72	**118**	7

Table 4.19: Wikimedia project - Number of committers touching projects in technical clusters.

Table 4.20 shows the results for personality clustering, and the heat map in Figure 4.24 depicts just the top-10 (the lowest) entropy values for the Big Five dimensions and facets, Needs, and Values (rows) by each personality cluster (columns).

Personality cluster	Number of committers
0	32
1	27
2	19
Total	78

Table 4.20: Wikimedia project - Results for personality clustering.

As highlighted in the heat map (Figure 4.24), personality cluster 1 groups the committers with the highest score in *Self-transcendence* (66.74%). Personality traits characterizing cluster 2 by its moderately high values are *Conservation* (46.30%), *Curiosity* (32.05%), and *Structure* (34.90%). With reference to personality clusters 1 and 2, the cluster 0 characterizes by moderately low values in most of their personality traits showed in Figure 4.24, ranging from 10% to 22% as in the case of *Curiosity* (10.31%), *Structure*

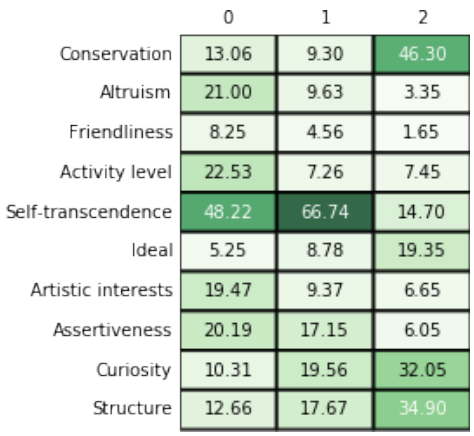


Figure 4.24: Wikimedia project - Heat map of the most discriminative factors for the personality clustering.

(12.66%), *Conservation* (13.06%), *Artistic interests* (19.47%), *Assertiveness* (20.19%), *Altruism* (21%), and *Activity level* (22.53%). Furthermore, the lowest value among all the personality traits in Figure 4.24 is for *Friendliness* (1.65%) in cluster 2.

4.4.3 Personality traits characterizing technical groups

Applying the next step of the proposed methodology as was done in the previous case studies, the entropy for each of the Big Five dimensions and facets, Needs and Values was computed to determine which of these attributes provide more information or become a differentiating factor when analyzing the technical groups. Centroids of personality traits for each technical cluster were obtained using the information about the technical cluster where each committer belongs and the personality traits for committers belonging to each technical cluster. Figure 4.25 shows just the 10 lowest values and the 10 highest values of entropy for the Big Five dimensions and facets, Needs, and Values of personality centroids computed by averaging the values of the personality traits of committers in each technical cluster.

Based on the results reported in Figure 4.25 and Table 4.18 to answer **RQ2** for this case study, personality traits scoring higher ($\geqslant$ 80%) and with nearly similar values through all technical clusters (e.g. *Intellect*, *Openness, Liberalism*, *Imagination*, *Adventurousness*, *Cautiousness*, and *Sympathy*) could be considered as personality factors characterizing the project, i.e. people involved in the project will most likely exhibit high values in these personality traits. By observing the personality traits scoring higher ($\geqslant$ 80%) in all case studies presented in this work, can be said that those are practically the same, differing mainly in the percentage in which these are present in each cluster. This may correspond to a recurring pattern in FLOSS projects. On the other hand, personality traits scoring lower ($\leqslant$ 25%) allow to identify relationships with the technical aspects, differentiating personality features among the different technical clusters.

Figure 4.26 summarizes personality traits by technical cluster allowing to visualize which personality traits are more representative in each technical cluster. From this representation can be noticed the dominant facets for the different technical clusters. Looking for relationships between personality traits characterizing technical clusters and the involvement of the committers in the Wikimedia projects, can be noticed that committers belonging to technical cluster 1, who exhibit the highest values in most of the personality traits shown in Figure 4.26, such as *Stability*, *Structure*, *Excitement*, *Closeness* and *Curiosity*, contribute actively

	0	1	2	3	4
Activity level	21.78	2.00	4.25	20.17	5.09
Stability	5.56	30.00	4.00	11.83	8.78
Structure	16.00	67.00	9.25	26.17	23.22
Friendliness	9.33	2.00	3.50	3.00	2.87
Excitement	3.33	14.00	3.50	5.83	5.13
Altruism	19.67	3.00	10.75	8.67	6.39
Assertiveness	18.11	3.00	19.50	8.67	12.91
Closeness	8.89	28.00	7.75	11.67	12.74
Curiosity	16.17	50.00	14.25	26.83	24.13
Self-expression	4.17	7.00	2.50	9.67	4.09

(a) The 10 lowest values of entropy (most discriminative personality factors for technical clusters) for the Big Five dimensions and facets, Needs, and Values of personality centroids.

	0	1	2	3	4
Intellect	97.39	99.00	98.25	97.83	97.30
Openness	96.00	98.00	97.50	96.83	96.39
Liberalism	95.67	99.00	97.00	97.50	97.04
Imagination	95.39	99.00	96.75	96.83	96.74
Adventurousness	89.72	92.00	89.75	90.50	88.00
Cautiousness	89.44	92.00	85.00	91.33	86.30
Sympathy	84.94	91.00	94.75	81.83	81.17
Achievement striving	74.22	64.00	66.25	71.33	63.52
Cooperation	77.44	78.00	68.25	72.67	63.39
Conscientiousness	67.28	55.00	62.50	63.00	55.26

(b) Top 10 highest values of entropy (personality factors characterizing the project) for the Big Five dimensions and facets, Needs, and Values of personality centroids.

Figure 4.25: Wikimedia project - Heat map of personality centroids for each technical cluster.

to the Mediawiki-Extensions-Wikidata[11] project, unlike committers belonging to the other technical clusters (0, 2, 3, and 4) who contributes mainly to Mediawiki-Core project and secondly to Wikimedia-Wikimania-Scholarships project. Committers belonging to technical clusters 3 and 4 exhibit similar values of personality traits, mostly in *Structure, Friendliness*, *Excitement*, *Closeness*, and *Curiosity* and contributes to the same projects (Mediawiki-Core and Wikimedia-Wikimania-Scholarships projects), but despite of the marked differences observed among personality traits characterizing technical clusters 0 and 2 with respect to those characterizing technical clusters 3 and 4, mainly in *Structure*, *Excitement*, *Altruism*, *Closeness*, and *Curiosity*, it seems that committers belonging to one of these technical clusters (0, 2, 3, and 4) have managed to engage and collaborate together as they work and contribute to the same projects (Mediawiki-Core and Wikimedia-Wikimania-Scholarships projects), although in varying proportions.

[11] `en.wikipedia.org/wiki/Wikidata`

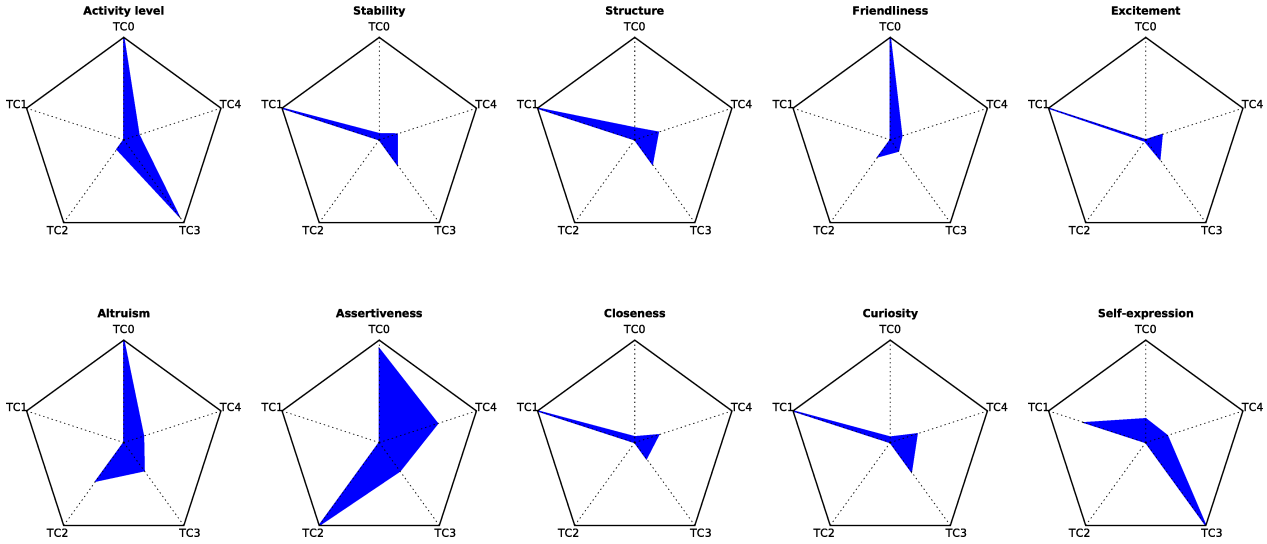


Figure 4.26: Wikimedia project - Radar charts of personality traits by technical cluster (TC).

Observing the distribution of personality traits through each of the technical clusters, some personality traits exhibit a behavioral pattern as can be seen in Figure 4.27. Clearly it shows that *Curiosity* and *Structure* categories, belonging to the Needs model, are traits that stand out in all technical clusters, while *Friendliness* (*Extraversion* facet), and *Self-expression* (Need) are features practically absent from the personality traits in most of the Wikimedia project committers, particularly those belonging to technical cluster 2. On the other hand, *Closeness* (Need), and *Assertiveness* (*Extraversion* facet) are traits that are present in a proportion nearly uniform in all the technical clusters. These does not stand out nor are entirely absent.

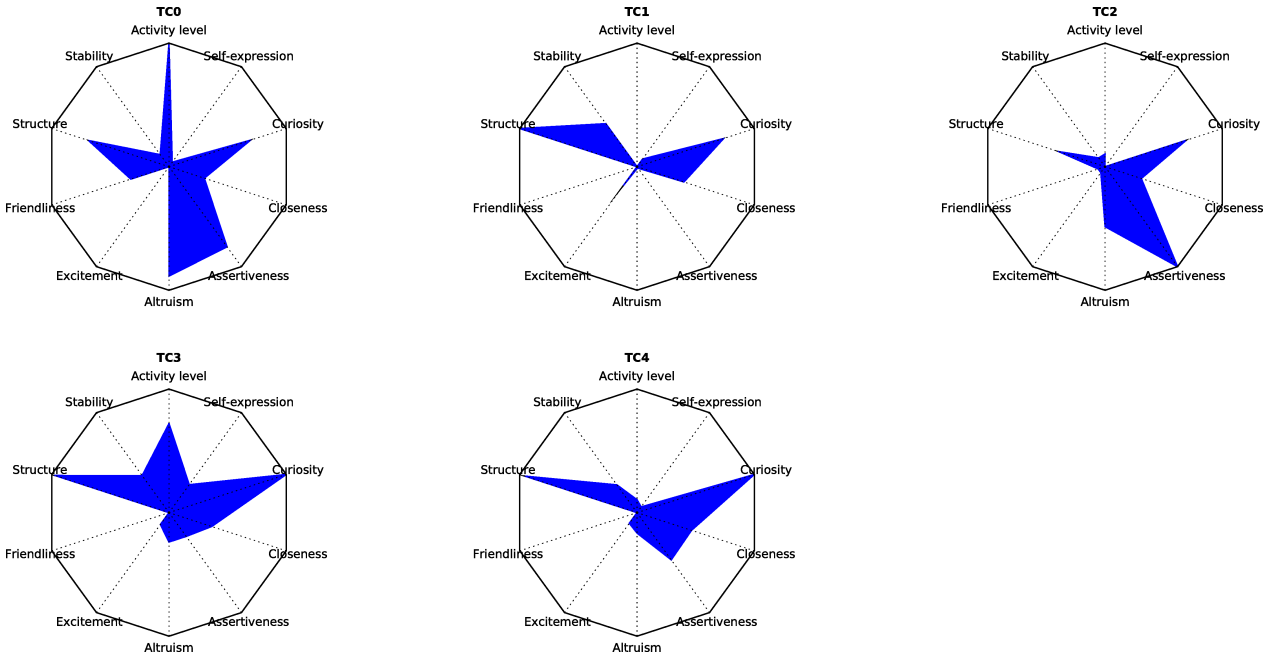


Figure 4.27: Wikimedia project - Radar chart of technical clusters characterized by personality traits.

4.4.4 Visualizing the social network - from committers to mailing lists

As in the first and third case studies, using the e-mails sent by committers to the Wikimedia project mailing lists, a graph representing e-mail communications was built. The graph in Figure 4.28 shows committers and mailing lists (MediaWiki-Enterprise, Wikitech-Ambassadors, MediaWiki-API, MediaWiki, Wikitech, XML-Data-Dumps, Pywikipedia, Maps, Translators, MediaWiki-i18n, WikiText, Commons, i.e., red circles) as nodes. The thickness of the edge between a committer and a mailing list represents the amount of emails sent by the committer to the list. Additionally, the color of the nodes representing committers corresponds to the technical cluster to which the committer belongs to (cluster 0: blue, cluster 1: yellow, cluster 2: orange, cluster 3: purple, cluster 4: green). Only committers that have sent more than 10 e-mails to any of the lists were taken into account. In the graph is evident the majority of representatives belongs to cluster 0 (blue circles). Cluster 1 (yellow circles) is the one with the fewest representatives. Only two (2) committers belonging to that cluster are part of the graph.

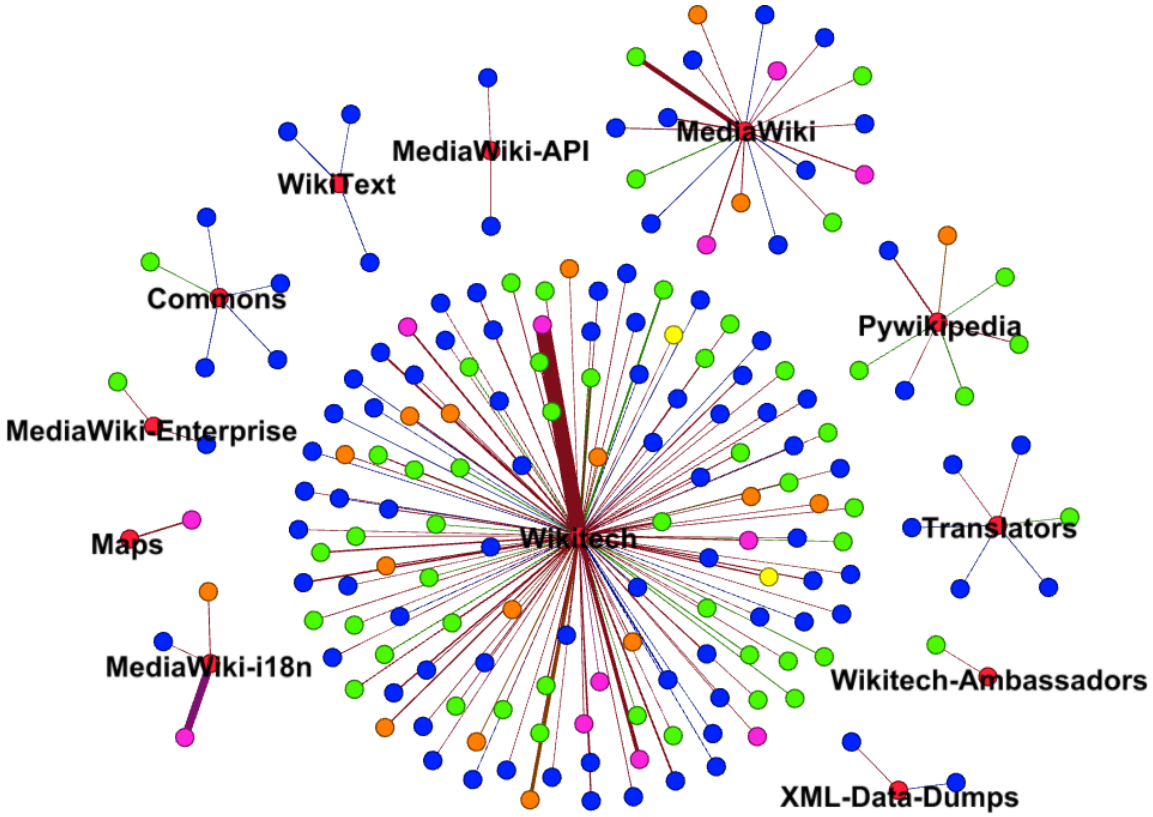


Figure 4.28: Wikimedia project - Social (email communication) network. From committers to mailing lists.

Figure 4.28 reveal there is no direct communication among committers and there are no emails linking mailing lists of each project. Apparently, communication only occurs through mailing lists.

This figure help to answer **RQ3** for this case study. Committers belonging to technical group 0 (blue circles) are those distributed through all mailing lists, except Maps and Wikitech-Ambassadors. This could be due to the ranking they have in the combination of personality traits such as *Activity level* (21.78%), *Altruism* (19.67%), and *Assertiveness* (18.11%). The graph also show that representatives of technical cluster 3 (purple circles) have a tendency to actively participate in the lists in greater extent than representatives of other

technical clusters, which could be related to personality traits *Curiosity* (26.83%), *Structure* (26.17%), and *Activity level* (20.17%).

4.5 Discussion

The large number of existing FLOSS projects may suggest that four case studies are not sufficient to draw conclusions that can be generalized, however the results obtained in this work show some common patterns between the analyzed projects while allow to validate the proposed methodology.

The answers to research questions (**RQ1** to **RQ3**) were derived from the results obtained during the experimental phase carried out for each of the case studies. Overall, it was observed that personality traits of software developers allow to identify relationships between their social and technical activities. As mentioned above, to determine whether the findings of this work may or may not be generalized is necessary to conduct experiments on a larger number of FLOSS projects. What can be said with a little less uncertainty is that each project has characteristic traits both technical and social and people who contribute to its development print out on it a bit of their personality traits.

Empirical evidence shows that personality traits scoring higher ($\geqslant 80\%$) and with nearly similar values through all technical clusters (e.g. *Intellect*, *Liberalism*, *Imagination*, *Openness, Cautiousness, Adventurousness*, and *Achievement striving*) could be considered as personality factors characterizing the project, i.e., people involved in the project will most likely exhibit high values in these personality traits. By observing personality traits scoring higher ($\geqslant 80\%$) in all case studies presented in this work, can be said that those are practically the same, differing mainly in the percentage in which are present in each cluster. This may correspond to a recurring pattern in FLOSS projects. On the other hand, personality traits scoring lower ($\leqslant 25\%$) allow to identify relationships with the technical aspects, differentiating personality features among different technical clusters. This statement stems from the fact that technical activities in three of the four case studies shows *Self-expression* (present in Eclipse Platform, OpenStack, and Wikimedia projects), *Stability* (present in Eclipse Platform OpenStack, and Wikimedia projects), *Excitement* (present in Eclipse Platform, Xen, and Wikimedia projects), *Morality* (present in Eclipse Platform, OpenStack, and Xen projects), and *Structure* (present in Eclipse Platform, OpenStack, and Wikimedia projects) as personality traits characterizing technical clusters while social activities in two of the four case studies (Xen and Wikimedia projects) show that committers having a tendency to actively participate in the mailing lists belong to technical clusters in which *Activity level* is one of the most representative personality trait.

Although it was observed in only one (Eclipse Platform project) of the four case studies, it was found that committers grouped in a technical cluster scoring high values in the *Artistic interests* facet contributes mainly to a project related to graphical elements, i.e., Eclipse Platform UI. Such preferences or behaviors were identified just from messages sent by committers to mailing lists.

IBM Watson Personality Insights service seems to be a suitable NLP tool to analyze personality traits from text when is impractical to apply personality tests to each participant of a study, however it is necessary to design and carry out experiments to test the validity of the perception of personality traits derived from such service (e.g. using a control group to which can be applied personality tests and contrast these results with those obtained from the *IBM Watson Personality Insights* service) in order to ensure the reliability of the analysis and the conclusions drawn at the end of the study.

In order to check whether the identified patterns are recurrent in FLOSS projects it is necessary to extend the study to a larger number of projects and it is desirable to involve professionals from areas such as sociology and psychology to analyze the results from perspectives complementing the point of view of engineering to

draw conclusions supported in interdisciplinary knowledge.

This work is a preliminary study aimed at supporting the setting up of efficient work teams in software development projects based on an appropriate mix of stakeholders taking into account their personality traits.

Chapter 5

Conclusions

FLOSS projects are characterized by a high component of social interaction where a large number of people with great technical skills contribute from different parts of the world, in most cases without knowing each other. Within this context, a preliminary study aimed at uncovering relationships between the social and technical aspects that occur during software development processes was conducted. Experimental results suggest the existence of relationships among personality traits projected by the committers through their e-mails and the social (communication) and technical activities they undertake.

Personality traits influence most, if not all, of the human activities, from those as natural as the way people walk, talk, dress and write to those most complex as the way they interact with others. For this reason, it appears to be a good choice to assist in meeting the goals proposed in this thesis, and it is considered the centerpiece of the work done. In this regard, services such as *IBM Watson Personality Insights* are crucial to analyze personality traits from text, when is impractical to apply personality tests to each participant of a study. By having the personality characteristics (a total of 52) inferred by the service, it was possible to identify relationships established, either solely from personality traits, or those established from both personality traits and social activities of the committers related to communication through the project mailing lists.

It was observed that personality traits of software developers allow to identify relationships between their social and technical activities, however, to determine whether the findings of this work may or may not be generalized is necessary to conduct experiments on a larger number of FLOSS projects. What is apparent is that a software project is characterized not only by technical aspects, it also is marked by personality traits that developers print out on it.

As evidenced by analyzing the graph representing social activities, is not enough to focus on just one personality trait to identify patterns due to the complexity of the personality and its constitutive factors. Thus, is necessary to take a comprehensive and detailed look at each one of the dimensions, facets and categories of the three personality models (Big Five, Needs and Values) to be able to draw conclusions most closely related to the behavior the data try to show us.

By observing personality traits scoring higher ($\geqslant 80\%$) in all case studies presented in this work, can be said that those are practically the same (i.e. *Intellect*, *Liberalism*, *Imagination*, *Openness, Cautiousness, Adventurousness*, and *Achievement striving*), differing mainly in the percentage in which these are present in each cluster. This could correspond to a recurring pattern in FLOSS projects.

Personality traits scoring lower ($\leqslant 25\%$) allowed to identify relationships with the technical aspects, differentiating personality features among different technical clusters.

Decision making at the time of forming software development teams should not be based solely on technical skills; it should be taken into account social skills that could be obtained from personality traits. This could create a better working environment and get as result a better software product.

Human behavior is so complex. It involves several aspects and interrelated variables that should be considered and analyzed together, and being aware that the analysis of personality traits demand specific knowledge in psychology it is necessary to engage in this kind of studies professionals in areas such as social and human sciences, e.g. psychologists and sociologists. It is not enough to rely only on the point of view of the Engineering as could be omitted certain aspects that should be considered.

Chapter 6

Threats to validity

What should have in mind is that the main aim of this work is to explore whether it is possible to extract personality traits from developer e-mails, and try to uncover relationships among those traits and the social and technical activities performed by the software team. As a feasible way to achieve this goal, was proposed a novel approach to collect, process, and analyze the relevant data, which involves the use of several tools and clustering techniques.

Preliminary results presented in this work may be affected by several validity threats inherent in the proposed approach. To mention just the most important ones, those results depend on an automatic analysis of developer e-mails performed by the *IBM Watson Personality Insights* service, instead of personality assessment questionnaires designed by psychologists and applied directly to the software team members. Moreover, the experiments are limited to the mailing lists and code base of four systems only. Thus, variables such as the project domain, the system size, the team size, and the quality and availability of the text could influence the effectiveness of the proposed approach.

Validity threats identified in each of the categories are described below:

- Internal validity

Because of the recommendation by the *IBM Watson Personality Insights* service about the input text used to infer individuals' intrinsic personality characteristics, it was necessary to select the group of committers (i.e., the text sent by each committer to the mailing lists) that would be part of the study, and to exclude those not meeting the recommendation. This restriction has the effect that no all committers that contribute to projects in each case study are taken Into account and their personality traits are not computed nor considered into the cluster analysis and, therefore, they (the committers) are not included in any cluster.

Cluster analysis conducted in this thesis requires to estimate the number of clusters (k) in which the datasets will be partitioned. In this thesis, the elbow method (or elbow criterion) was used to obtain the value of this parameter. Since the elbow criterion is a visual method, it has implicit the subjectivity of the observer to select the point at which the "elbow" is found, so, it may happen that results may vary for different values of k.

Entropy was the measure used to determine which of the personality traits provide more information or become a differentiating factor when analyzing the technical groups. Similar results were obtained by computing standard deviation (σ) for each of the *Big Five* dimensions and facets, *Needs* and *Values*. Small changes in the results could be expected by using information gain as attribute selection measure.

- Construct validity

The NLP tool used to discover individuals' personality traits from text (in this thesis, the *IBM Watson Personality Insights* service) is a critical component in the proposed methodology as final results depend on its precision and reliability to infer personality characteristics, and how close the tool's results are to the scores the author of the text would receive by taking an actual personality test. Since most of the time a third-party NLP tool will be used, there will not be any control over it, and the best thing to do is to select the better available tool.

For a full set of individuals' contributions to a FLOSS project it is necessary to integrate information from multiple data sources corresponding to different kinds of repositories (e.g., mail archives and version control systems). However, developers may use different aliases in different software repositories, and even different aliases in the same software repository. To integrate information about individual contributions, it is needed a unique identity representing the same contributor across different repositories and different projects. To this end, it is necessary to use an identity merging algorithm. Identity merging remains challenging in mining software repositories, and despite of the work done to improve the existing approaches, all of them produce false positives and false negatives [80]. Identity information, which is part of the datasets used in this thesis, has gone through a process of identity merging carried out using the tool Sorting Hat[1], a tool to manage identities.

Representation of technical data used in this thesis (i.e., binary vectors representing whether a committer touched or not a file of the project) may have omitted some relevant information available in mail archives and version control systems such as commit messages, patches attached to e-mails, source code metrics, developers' coding style, and emotions, attitudes, or sentiments developers tried to express in messages sent to the mailing lists. Including this kind of information in the representation of technical data could lead to other conclusions.

- External validity

Projects that are part of the case studies are not a representative sample of a particular software category or domain, nor a specific programming language, or end user license, or focused on meeting a specific requirement. Selection criteria were those related with number of contributors, time under development, and mainly the public access to mail archives and source code repositories. Applying these criteria allowed to collect enough data to validate the proposed methodology, however same criteria do not allow to ensure that results are generalizable to other projects, or development teams, or FLOSS communities.

[1] `github.com/MetricsGrimoire/sortinghat`

Chapter 7

Future work

In the short term and in order to validate the *IBM Watson Personality Insights* service outputs, it is proposed to apply personality tests based on the Big Five, Needs and Values personality models to a group of developers that are part of software teams in software factories. This will determine if the service is reliable in terms of the attributes of personality inferred from digital communications such as e-mails, text messages, tweets, and forum posts.

To get the most out of the data available in repositories (source code, mailing lists, issue trackers, IRC channels), it is suggested to include the commit messages in the text used to identify developer' personality traits; and it is recommended to take into account the messages sent to the issue tracker and communications that take place at IRC channels.

By knowing the relationships between the socio-technical aspects that occurs in the software development process and the personality traits of the stakeholders involved in this process, it is possible to extend this study to the setting up of efficient work teams based not only on their technical skills but also on their personality traits.

It could be interesting to analyze a greater number of FLOSS projects in order to identify personality traits present in developers communities and, based on this information, to recommend projects, modules or activities (development, design, test, bugs fixes, documentation, coordination) they may be interested in contributing.

Complementing the identification of relationships among personality traits and the social and technical activities performed by the software team, this work can be extended by performing sentiment analysis and mining emotions from text written by developers, and look for correlations and common aspects among personality traits and sentiments or emotions embodied in the analyzed text.

Under the assumption that personality changes over time, it can be performed an analysis of the evolution of developers' personality traits as software evolves and identify how it relates to the quality of the software product and how it looks reflected in the evolution of developers community.

Because personality traits define the preferences and human behavior, another aspect that can be explored is the identification of roles in software teams from developers' personality traits. This could be helpful when suggesting candidates for certain positions or activities within the developers community.

Glossary

Activity level / Energetic:	Lead fast-paced and busy lives. They do things and move about quickly, energetically, and vigorously, and they are involved in many activities. It is an Extraversion facet.
Activity level:	In the context of this thesis, activity level was defined as the averaged number of times that project directories have been touched by committers. Just average values $\geqslant 0.07$ were taken into account. It is computed from the number of times that a project directory has been touched by the committers of each technical cluster divided by the sum of the times that all project directories have been touched by the committers of each technical cluster.
Altruism / Altruistic:	Find that helping others is genuinely rewarding, that doing things for others is a form of self-fulfillment rather than self-sacrifice. It is an Agreeableness facet.
Artistic interests:	Love beauty, both in art and in nature. They become easily involved and absorbed in artistic and natural events. With intellect, this facet is one of the two most important, central aspects of this characteristic. It is an Openness facet.
Assertiveness / Assertive:	Like to take charge and direct the activities of others. They tend to be leaders in groups. It is an Extraversion facet.
Conscientiousness:	Is a person's tendency to act in an organized or thoughtful way. It is one of the dimensions of the Big Five model.
Conservation / Tradition:	Emphasize self-restriction, order, and resistance to change. It is one of the dimensions of the Values model.
Curiosity:	Have a desire to discover, find out, and grow. It is one of the categories of the Needs model.
Elbow method:	One method to validate the number of clusters is the elbow method. The idea of the elbow method is to run k-means clustering on the dataset for a range of values of k (say, k from 1 to 10), and for each value of k calculate the sum of squared errors (SSE). The SSE is defined as the sum of the squared distance between each member of the cluster and its centroid. Plotting k against the SSE, you will see that the error decreases as

k gets larger. This is because when the number of clusters increases, they should be smaller, so distortion is also smaller. The idea of the elbow method is to choose the k at which the SSE decreases abruptly. This produces an "elbow effect" in the graph.
bl.ocks.org/rpgove/0060ff3b656618e9136b
www.quora.com/How-can-we-choose-a-good-K-for-K-means-clustering

Friendliness / Outgoing / Warmth: Genuinely like other people and openly demonstrate positive feelings toward others. It is an Extraversion facet.

IBM Watson Personality Insights: The IBM Watson™ Personality Insights service provides an Application Programming Interface (API) that enables applications to derive insights from social media, enterprise data, or other digital communications. The service uses linguistic analytics to infer individuals' intrinsic personality characteristics, including Big Five, Needs, and Values, from digital communications such as email, text messages, tweets, and forum posts. The service can automatically infer, from potentially noisy social media, portraits of individuals that reflect their personality characteristics.
www.ibm.com/watson/developercloud/doc/personality-insights

Multimodal data: In the context of this thesis, multimodal data refer to data coming from multiple, different, and independent sources.

Needs: This personality model describes at a high level which aspects of a product are likely to resonate with the author of the text. The model includes twelve categories of needs based on Kotler's and Ford's work in marketing.
The IBM Watson Personality Insights service evaluates twelve categories of needs: Excitement, Harmony, Curiosity, Ideal, Closeness, Self-expression, Liberty, Love, Practicality, Stability, Challenge, and Structure.
http://www.ibm.com/watson/developercloud/doc/personality-insights/models.shtml#outputNeeds

Self-enhancement / Achieving success: Seek personal success for themselves. It is one of the dimensions of the Values model.

Self-transcendence / Helping others: Show concern for the welfare and interests of others. It is one of the dimensions of the Values model.

Socio-technical relationships: In the context of this thesis, socio-technical relationships refer to connections between technical activities (e.g., commits) or technical aspects (e.g., source code artifacts), and social activities (e.g., communications among software developers through mailing lists).

Structure: Exhibit groundedness and a desire to hold things together. They need things to be well organized and under control. It is one of the categories of the Needs model.

Sympathy / Empathetic:	Are tender-hearted and compassionate. It is an Agreeableness facet.
Technical cluster:	In the context of this thesis, technical cluster refers to one of the k partitions performed by the clustering algorithm on a given dataset (the technical dataset in this case), and formed by data objects (committers in this case) sharing similar characteristics (i.e., committers who touched nearly the same project files/directories).
Technical data:	In the context of this thesis, technical data refers to data obtained from the source code repository, and represent whether a committer touched or not each file of the project.
Values:	This personality model describes motivating factors that influence the author's decision-making. The model includes five dimensions of human values based on Schwartz's work in psychology. The IBM Watson Personality Insights service infers five values: Self-transcendence / Helping others, Conservation / Tradition, Hedonism / Taking pleasure in life, Self-enhancement / Achieving success, and Open to change / Excitement. `http://www.ibm.com/watson/developercloud/doc/personality-insights/models.shtml#outputValues`

References

[1] Roberto Abreu and Rahul Premraj. How developer communication frequency relates to bug introducing changes. In *Proceedings of the joint international and annual ERCIM workshops on Principles of software evolution (IWPSE) and software evolution (Evol) workshops - IWPSE-Evol '09*, pages 153 – 158, New York, New York, USA, 2009. ACM Press.

[2] Silvia T. Acuña, Marta Gómez, and Natalia Juristo. How do personality, team processes and task characteristics relate to job satisfaction and software quality? *Information and Software Technology*, 51(3):627–639, mar 2009.

[3] Silvia T. Acuña and Natalia Juristo. Assigning people to roles in software projects. *Software: Practice and Experience*, 34(7):675–696, jun 2004.

[4] SN Ahsan, MT Afzal, and S Zaman. Mining effort data from the oss repository of developer's bug fix activity. *Journal of IT in ...*, page 14, 2010.

[5] Alberto Bacchelli, Tommaso Dal Sasso, Marco D'Ambros, and Michele Lanza. Content classification of development emails. In *2012 34th International Conference on Software Engineering (ICSE)*, pages 375–385. IEEE, jun 2012.

[6] Victor R. Basili and Robert W. Reiter. An Investigation of Human Factors in Software Development. *Computer*, 12(12):21–38, dec 1979.

[7] Olga Baysal and Andrew J. Malton. Correlating Social Interactions to Release History during Software Evolution. In *Fourth International Workshop on Mining Software Repositories (MSR'07:ICSE Workshops 2007)*, pages 1–8. IEEE, may 2007.

[8] Blerina Bazelli, Abram Hindle, and Eleni Stroulia. On the personality traits of StackOverflow users. In *IEEE International Conference on Software Maintenance, ICSM*, pages 460–463, 2013.

[9] Xu Ben, Shen Beijun, and Yang Weicheng. Mining Developer Contribution in Open Source Software Using Visualization Techniques. In *2013 Third International Conference on Intelligent System Design and Engineering Applications*, pages 934–937. IEEE, jan 2013.

[10] D Beyer and A Noack. Mining co-change clusters from version repositories. page 18, 2005.

[11] Christian Bird, Alex Gourley, Prem Devanbu, Michael Gertz, and Anand Swaminathan. Mining email social networks. In *Proceedings of the 2006 international workshop on Mining software repositories - MSR '06*, pages 137–143, New York, New York, USA, 2006. ACM Press.

[12] Christian Bird, Nachiappan Nagappan, Harald Gall, Brendan Murphy, and Premkumar Devanbu. Putting It All Together: Using Socio-technical Networks to Predict Failures. In *2009 20th International Symposium on Software Reliability Engineering*, pages 109–119. IEEE, nov 2009.

[13] Christian Bird, David Pattison, Raissa D'Souza, Vladimir Filkov, and Premkumar Devanbu. Latent social structure in open source projects. In *Proceedings of the 16th ACM SIGSOFT International Symposium on Foundations of software engineering - SIGSOFT '08/FSE-16*, page 24, New York, New York, USA, 2008. ACM Press.

[14] F Bolici, J Howison, and K Crowston. Coordination without discussion? Socio-technical congruence and Stigmergy in Free and Open Source Software projects. *Socio-Technical Congruence ...*, pages 1 – 9, 2009.

[15] John H. Bradley and Frederic J. Hebert. The effect of personality type on team performance. *Journal of Management Development*, 16(5):337–353, jul 1997.

[16] LB Buchanan. *The impact of big five personality characteristics on group cohesion and creative task performance*. PhD thesis, 1998.

[17] David Budgen, Mark Turner, Pearl Brereton, and Barbara Kitchenham. Using Mapping Studies in Software Engineering. In *Proceedings of PPIG*, volume 2, pages 195–204, 2008.

[18] LF Capretz. Are software engineers really engineers? *World Transactions on Engineering and Technology ...*, page 4, 2002.

[19] L.F. Capretz and F. Ahmed. Making Sense of Software Development and Personality Types. *IT Professional*, 12(1):6–13, jan 2010.

[20] Luiz Fernando Capretz. Personality types in software engineering. *International Journal of Human-Computer Studies*, 58(2):207–214, feb 2003.

[21] Luiz Fernando Capretz and Faheem Ahmed. Why do we need personality diversity in software engineering? *ACM SIGSOFT Software Engineering Notes*, 35(2):1–11, mar 2010.

[22] F Celli and M Poesio. Pr2: A language independent unsupervised tool for personality recognition from text. *arXiv preprint arXiv:1402.2796*, page 4, 2014.

[23] Fabio Celli. *Adaptive Personality Recognition from Text.* PhD thesis, 2013.

[24] Fabio Celli, Fabio Pianesi, David Stillwell, and Michal Kosinski. Workshop on Computational Personality Recognition : Shared Task. *Proceedings of the Workshop on Personality Recognition*, 2006:1 – 5, 2013.

[25] K.K. Chaturvedi, V.B. Sing, and Prashast Singh. Tools in Mining Software Repositories. In *2013 13th International Conference on Computational Science and Its Applications*, pages 89–98. IEEE, jun 2013.

[26] Kevin Crowston and James Howison. The social structure of free and open source software development. *First Monday*, 10(2):1 – 27, feb 2005.

[27] Shirley Cruz, Fabio Q.B. da Silva, and Luiz Fernando Capretz. Forty years of research on personality in software engineering: A mapping study. *Computers in Human Behavior*, 46:94–113, may 2015.

[28] AD Da Cunha and D Greathead. Code review and personality: is performance linked to MBTI type? *TECHNICAL REPORT SERIES- ...*, page 18, 2004.

[29] Alessandra Devito Da Cunha and David Greathead. Does personality matter?: an analysis of code-review ability. *Communications of the ACM*, 50(5):109–112, may 2007.

[30] M. D'Ambros, M. Lanza, and M. Lungu. Visualizing Co-Change Information with the Evolution Radar. *IEEE Transactions on Software Engineering*, 35(5):720–735, sep 2009.

[31] Marco D'Ambros, Michele Lanza, and Martin Pinzger. "A Bug's Life" Visualizing a Bug Database. In *2007 4th IEEE International Workshop on Visualizing Software for Understanding and Analysis*, pages 113–120. IEEE, jun 2007.

[32] Cleidson de Souza, Jon Froehlich, and Paul Dourish. Seeking the source: software source code as a social and technical artifact. In *Proceedings of the 2005 international ACM SIGGROUP conference on Supporting group work - GROUP '05*, pages 197 – 206, New York, New York, USA, 2005. ACM Press.

[33] Stephan Diehl. *Software Visualization Visualizing the Structure, Behaviour, and Evolution of Software*. Springer, 2007.

[34] Kate Ehrlich, Giuseppe Valetto, and Mary Helander. Seeing inside: Using social network analysis to understand patterns of collaboration and coordination in global software teams. In *International Conference on Global Software Engineering (ICGSE 2007)*, pages 297–298. IEEE, aug 2007.

[35] Alastair James Gill. *Personality and Language: The projection and perception of personality in computer-mediated communication*. PhD thesis, University of Edinburgh, 2003.

[36] Mathieu Goeminne and Tom Mens. A framework for analysing and visualising open source software ecosystems. In *Proceedings of the Joint ERCIM Workshop on Software Evolution (EVOL) and International Workshop on Principles of Software Evolution (IWPSE) on - IWPSE-EVOL '10*, pages 1 – 6, New York, New York, USA, 2010. ACM Press.

[37] Jennifer Golbeck, Cristina Robles, Michon Edmondson, and Karen Turner. Predicting Personality from Twitter. In *2011 IEEE Third Int'l Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third Int'l Conference on Social Computing*, pages 149–156. IEEE, oct 2011.

[38] Jesus M. Gonzalez-Barahona, Gregorio Robles, and Daniel Izquierdo-Cortazar. The MetricsGrimoire Database Collection. In *2015 IEEE/ACM 12th Working Conference on Mining Software Repositories*, pages 478–481. IEEE, may 2015.

[39] Antonio Gonzalez-Torres, Roberto Theron, Francisco J. Garcia-Penalvo, Michel Wermelinger, and Yijun Yu. Maleku: An evolutionary visual software analysis tool for providing insights into software evolution. In *2011 27th IEEE International Conference on Software Maintenance (ICSM)*, pages 594–597. IEEE, sep 2011.

[40] Narasimhaiah Gorla and Yan Wah Lam. Who should work with whom?: Building Effective Software Project Teams. *Communications of the ACM*, 47(6):79–82, jun 2004.

[41] Liang Gou, Jalal Mahmud, Eben Haber, and Michelle Zhou. PersonalityViz: a visualization tool to analyze people's personality with social media. In *Proceedings of the companion publication of the 2013 international conference on Intelligent user interfaces companion - IUI '13 Companion*, pages 45–46, New York, New York, USA, 2013. ACM Press.

[42] Daniel Graziotin, Xiaofeng Wang, and Pekka Abrahamsson. Understanding the affect of developers: theoretical background and guidelines for psychoempirical software engineering. In *Proceedings of the 7th International Workshop on Social Software Engineering - SSE 2015*, pages 25–32, New York, New York, USA, 2015. ACM Press.

[43] D Greathead. MBTI personality type and student code comprehension skill. *Proceedings of the 20th Workshop on the Psychology ...*, page 13, 2008.

[44] A Guzzi, A Bacchelli, and M Lanza. Communication in open source software development mailing lists. *Proceedings of the 10th ...*, pages 277–286, 2013.

[45] Avdo Hanjalic. ClonEvol: Visualizing software evolution with code clones. In *2013 First IEEE Working Conference on Software Visualization (VISSOFT)*, pages 1–4. IEEE, sep 2013.

[46] J Howison, K Inoue, and K Crowston. Social dynamics of free and open source team communications. *Open Source Systems*, pages 319 – 330, 2006.

[47] J. Huffman Hayes. Do you like Pina Coladas? How improved communication can improve software quality. *IEEE Software*, 20(1):90–92, jan 2003.

[48] Walid M Ibrahim, Nicolas Bettenburg, Emad Shihab, Bram Adams, and Ahmed E Hassan. Should I contribute to this discussion? In *2010 7th IEEE Working Conference on Mining Software Repositories (MSR 2010)*, pages 181–190. IEEE, may 2010.

[49] A Iqbal. Analyzing Social Behavior of Software Developers Across Different Communication Channels. *International Conference on Software ...*, pages 1 – 6, 2013.

[50] Michael John, Frank Maurer, and Bjørnar Tessem. Human and social factors of software engineering. *ACM SIGSOFT Software Engineering Notes*, 30(4):6, jul 2005.

[51] Tanjila Kanij, Robert Merkel, and John Grundy. An Empirical Investigation of Personality Traits of Software Testers. In *2015 IEEE/ACM 8th International Workshop on Cooperative and Human Aspects of Software Engineering*, pages 1–7. IEEE, may 2015.

[52] JS Karn and AJ Cowling. An Initial Study of the effect of personality on group cohesion in software engineering projects. page 49, 2011.

[53] Nicole Kimmelmann. Career in Open Source? Relevant Competencies for Successful Open Source Developers. *it - Information Technology*, 55(5):9, jan 2013.

[54] I Kwan, A Schroter, and D Damian. Does Socio-Technical Congruence Have an Effect on Software Build Success? A Study of Coordination in a Software Project. *IEEE Transactions on Software Engineering*, 37(3):307–324, may 2011.

[55] Irwin Kwan and Daniela Damian. Extending socio-technical congruence with awareness relationships. In *Proceedings of the 4th international workshop on Social software engineering - SSE '11*, pages 1 – 8, New York, New York, USA, 2011. ACM Press.

[56] Per Lenberg, Robert Feldt, and Lars-Göran Wallgren. Towards a behavioral software engineering. In *Proceedings of the 7th International Workshop on Cooperative and Human Aspects of Software Engineering - CHASE 2014*, pages 48–55, New York, New York, USA, 2014. ACM Press.

[57] Sherlock A. Licorish and Stephen G. MacDonell. Personality profiles of global software developers. In *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering - EASE '14*, pages 1–10, New York, New York, USA, 2014. ACM Press.

[58] L López-Fernández. Applying social network analysis techniques to community-driven libre software projects. *International Journal of . . .*, pages 27 – 48, 2006.

[59] L Lopez-Fernandez, G Robles, and JM Gonzalez-Barahona. Applying social network analysis to the information in CVS repositories. pages 1 – 5, 2004.

[60] Bonnie MacKellar. A Case Study of Group Communication Patterns in a Large Project Software Engineering Course. In *2012 IEEE 25th Conference on Software Engineering Education and Training*, pages 134–138. IEEE, apr 2012.

[61] Francois Mairesse and Marilyn Walker. Words Mark the Nerds: Computational Models of Personality Recognition through Language. *28th Annual Conference of the Cognitive Science Society*, pages 543–548, 2000.

[62] François Mairesse, Marilyn A. Walker, Matthias R. Mehl, and Roger K. Moore. Using Linguistic Cues for the Automatic Recognition of Personality in Conversation and Text. *Journal of Artificial Intelligence Research*, 30(1):457 – 500, 2007.

[63] Luis G. Martínez, Juan R. Castro, Guillermo Licea, Antonio Rodríguez-Díaz, and Carlos F. Alvarez. KNOWING SOFTWARE ENGINEER'S PERSONALITY TO IMPROVE SOFTWARE DEVELOPMENT. In *Proceedings of the 6th International Conference on Software and Database Technologies*, pages 99–104. SciTePress - Science and and Technology Publications, 2011.

[64] T Mens and M Goeminne. Analysing the evolution of social aspects of open source software ecosystems. *. . . Workshop on Software Ecosystems (. . .*, pages 1 – 14, 2011.

[65] E Moritz and M Linares-Vásquez. ExPort: Detecting and Visualizing API Usages in Large Source Code Repositories. *cs.wm.edu*, page 6, 2013.

[66] G. A. Neuman, S. H. Wagner, and N. D. Christiansen. The Relationship between Work-Team Personality Composition and the Job Performance of Teams. *Group & Organization Management*, 24(1):28–45, mar 1999.

[67] Oscar Hernán Paruma-Pabón, Fabio A. González, Jairo Aponte, Jorge E. Camargo, and Felipe Restrepo-Calle. Finding relationships between socio-technical aspects and personality traits by mining developer e-mails. In *Proceedings of the 9th International Workshop on Cooperative and Human Aspects of Software Engineering - CHASE '16*, pages 8–14, New York, New York, USA, 2016. ACM Press.

[68] Vreda Pieterse, Derrick G. Kourie, and Inge P. Sonnekus. Software engineering team diversity and performance. In *Proceedings of the 2006 annual research conference of the South African institute of computer scientists and information technologists on IT research in developing couuntries - SAICSIT '06*, pages 180–186, New York, New York, USA, 2006. ACM Press.

[69] Martin Pinzger, Harald Gall, Michael Fischer, and Michele Lanza. Visualizing multiple evolution metrics. In *Proceedings of the 2005 ACM symposium on Software visualization - SoftVis '05*, page 67, New York, New York, USA, 2005. ACM Press.

[70] L Pirzadeh. *Human Factors in Software Development: A Systematic Literature Review.* PhD thesis, 2010.

[71] Peter C. Rigby and Ahmed E. Hassan. What Can OSS Mailing Lists Tell Us? A Preliminary Psychometric Text Analysis of the Apache Developer Mailing List. In *Fourth International Workshop on Mining Software Repositories (MSR'07:ICSE Workshops 2007)*, pages 23–23. IEEE, may 2007.

[72] A Sarma. Challenges in measuring, understanding, and achieving social-technical congruence. *... - Technical Congruence ...*, pages 1 – 12, 2008.

[73] Walt Scacchi. Managing Software Engineering Projects: A Social Analysis. *IEEE Transactions on Software Engineering*, SE-10(1):49–59, jan 1984.

[74] Emad Shihab and Ahmed E. Hassan. On the use of Internet Relay Chat (IRC) meetings by developers of the GNOME GTK+ project. In *2009 6th IEEE International Working Conference on Mining Software Repositories*, pages 107–110. IEEE, may 2009.

[75] I Sommerville and T Rodden. Human, social and organisational influences on the software process. *Software Process*, page 21, 1996.

[76] Megan Squire. How the FLOSS Research Community Uses Email Archives. *International Journal of Open Source Software and Processes*, 4(1):37–59, jan 2012.

[77] François Stephany, Tom Mens, and Tudor Gîrba. Maispion: A Tool for Analysing and Visualising Open Source Software Developer Communities. In *Proceedings of the International Workshop on Smalltalk Technologies - IWST '09*, pages 1–9, New York, New York, USA, 2009. ACM Press.

[78] A. Telea and L. Voinea. Interactive Visual Mechanisms for Exploring Source Code Evolution. In *3rd IEEE International Workshop on Visualizing Software for Understanding and Analysis*, pages 1–6. IEEE, 2005.

[79] D Varona, LF Capretz, and Y Piñero. Personality types of cuban software developers. *Global Journal of Engineering ...*, page 5, 2011.

[80] BN Vasilescu. *Social aspects of collaboration in online software communities.* PhD thesis, 2014.

[81] Lucian Voinea and Alexandru Telea. Multiscale and multivariate visualizations of software evolution. In *Proceedings of the 2006 ACM symposium on Software visualization - SoftVis '06*, page 115, New York, New York, USA, 2006. ACM Press.

[82] S Wagner and M Ruhe. A systematic review of productivity factors in software development. *language*, page 6, 1980.

[83] PF Xiang, ATT Ying, and P Cheng. Ensemble: a recommendation tool for promoting communication in software teams. *Proceedings of the ...*, page 2, 2008.

[84] Xinrong Xie, Denys Poshyvanyk, and Andrian Marcus. Visualization of CVS Repository Information. In *2006 13th Working Conference on Reverse Engineering*, pages 231–242. IEEE, 2006.

[85] H Yang and Y Li. Identifying user needs from social media. *IBM Research Division, San Jose*, page 11, 2013.

[86] Tal Yarkoni. Personality in 100,000 Words: A large-scale analysis of personality and word use among bloggers. *Journal of Research in Personality*, 44(3):363–373, jun 2010.

[87] A.T.T. Ying, G.C. Murphy, R. Ng, and M.C. Chu-Carroll. Predicting source code changes by mining change history. *IEEE Transactions on Software Engineering*, 30(9):574–586, sep 2004.

[88] L Yu. Mining change logs and release notes to understand software maintenance and evolution. *CLEI Electron Journal*, page 10, 2009.

[89] Junji Zhi and Gunther Ruhe. DEVis: A tool for visualizing software document evolution. In *2013 First IEEE Working Conference on Software Visualization (VISSOFT)*, pages 1–4. IEEE, sep 2013.

[90] T. Zimmermann, S. Diehl, and A. Zeller. How history justifies system architecture (or not). In *Sixth International Workshop on Principles of Software Evolution, 2003. Proceedings.*, pages 73–83. IEEE, 2003.