Information and Software Technology 64 (2015) 113–131

Contents lists available at ScienceDirect

Information and Software Technology

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

Communication and personality profiles of global software developers

Sherlock A. Licorish, Stephen G. MacDonell *

Department of Information Science, University of Otago, PO Box 56, Dunedin 9054, New Zealand

ARTICLE INFO

Article history:
Received 10 September 2014
Received in revised form 22 January 2015
Accepted 9 February 2015
Available online 24 February 2015

Keywords:
Human factors
Communication
Personality
SNA
Data mining
Psycholinguistics

ABSTRACT

Context: Prior research has established that a small proportion of individuals dominate team communication during global software development. It is not known, however, how these members' contributions affect their teams' knowledge diffusion process, or whether their personality profiles are responsible for their dominant presence.
Objective: We set out to address this gap through the study of repository artifacts.
Method: Artifacts from ten teams were mined from the IBM Rational Jazz repository. We employed social network analysis (SNA) to group practitioners into two clusters, *Top Members* and *Others*, based on the numbers of messages they communicated and their engagement in task changes. SNA metrics (density, in-degree and closeness) were then used to study practitioners' importance in knowledge diffusion. Thereafter, we performed psycholinguistic analysis on practitioners' messages using linguistic dimensions that had been previously correlated with the Big Five personality profiles.
Results: For our sample of 146 practitioners we found that Top Members occupied critical roles in knowledge diffusion, and demonstrated more openness to experience than the Others. Additionally, all personality profiles were represented during teamwork, although openness to experience, agreeableness and extroversion were particularly evident. However, no specific personality predicted members' involvement in knowledge diffusion.
Conclusion: Task assignment that promotes highly connected team communication networks may mitigate tacit knowledge loss in global software teams. Additionally, while members expressing openness to experience are likely to be particularly driven to perform, this is not entirely responsible for a global team's success.

1. Introduction

All but the smallest software development endeavors rely on the productive work of teams, whether co-located or (increasingly commonly) dispersed across locations. Prior research across numerous contexts has established that the intricacies of team dynamics may be revealed by studying members' communication[1] [1,2], and that good communication is essential for building positive interpersonal relations in teams [3]. Among the specific assertions considered previously, research has revealed linkages between informal hierarchical communication structures and team performance for geographically distributed teams [4]. Team communication has also been linked to coordination efficiency [5] and to the quality of resultant software artifacts [6]. Thus, studying the details in and of team communication can provide valuable insights into the human processes involved during software development, including the importance of team members in communication structures, along with the reasons for, and consequences of, communication and coordination actions.

Similarly, aspects of team composition and team members' social and behavioral traits are also said to influence the outcomes of group-based activities. Such issues have been considered from multiple perspectives, including sociology and behavioral psychology relating to social identity [7], social capital [8] and personality psychology [9]. According to contemporary thinking, as well as observed practice in software development, individuals bring unique sets of knowledge[2] and skills to their collaboration during group work. These collective experiences (both prior and in-project), and in particular, those personal qualities that 'connect' during interactions, are influenced by participants' social and behavioral traits. Variations in these traits are said to determine if, and how, team

* Corresponding author.
E-mail addresses: sherlock.licorish@otago.ac.nz (S.A. Licorish), stephen.macdonell@otago.ac.nz (S.G. MacDonell).

[1] The terms "communication" and "interaction" are used interchangeably throughout this the paper to mean the exchange of information.

[2] Information or expertise acquired through previous experience or formal training.

http://dx.doi.org/10.1016/j.infsof.2015.02.004
0950-5849/

members interact and the likelihood of teams being cohesive and productive [10].

Practitioners' communications and personalities and their effect on team members' behaviors and team output have therefore been receiving increasing attention in the software engineering research literature. For instance, Bird et al. [11] studied CVS records and mailing lists and concluded that the more software development an individual does the more coordination and controlling activities they must undertake. One of our own prior studies [12] found that just a few team members dominated project communication, and that these developers were crucial to their teams' organizational, intra-personal and inter-personal processes. Abreu and Premraj [13] observed the Eclipse mailing list and found that increases in communication intensity coincided with higher numbers of bug-introducing changes, and that developers communicated most frequently at release and integration time. From a personality perspective, a study of 47 professional software engineers in ten Swedish software development companies found significant associations between personality factors and software engineers' behaviors [14]. Gorla and Lam's study [15] of the personalities of 92 high-performing IS professionals in Hong Kong uncovered that extroverted programmers outperformed those who were intuitive. Wang [16] also offered support for the linking of personality to team performance when reviewing 116 software project outputs.

While it has been observed that just a few members tend to dominate team communication during software development [17], little is known about these members' role in project knowledge diffusion. Knowledge diffusion is *the spread or transfer of knowledge from one part of a network (or individual) to another* [18]. Perhaps these highly active members communicate densely on specific tasks, and so they may be no more important in their team's overall knowledge diffusion process than those who communicate with fewer messages on comparatively higher numbers of software tasks. To this end, inferring practitioners' importance in knowledge diffusion based solely on the number of messages they contribute may be biased, resulting in unfulfilled expectations regarding practitioner performance and negative project consequences.

Similarly, there has been limited research focused on studying the potential influence of personality on members' involvement in knowledge diffusion during distributed and global software developments (GSDs). This is despite the belief that such work should help us to understand the potentially complex team dynamics in these environments [19]. Such explorations would seem to be particularly necessary given that these very teams are often challenged by reduced levels of awareness, group identification and shared understandings, due to team members' separation [20]. Thus, studying personality and behavioral issues in these settings, and understanding the potential impact of these variables on the performance of practitioners, should lead to recommendations that are likely to influence positive project outcomes [14].

In addressing these gaps we have led multiple explorations in an effort to contribute understandings around global team dynamics. For instance, we previously examined the role of core developers in global software teams to provide initial insights into the reasons for their exaggerated presence [12]. Additionally, we examined the potential influence of personality in global teams, providing insights into the profiles of Top Members [21]. In the current study we provide an extension to these works [12,21], and bring together two distinct but related threads introduced above, communication and personality. We investigate the importance of active communicators in knowledge diffusion and the distribution of these practitioners' personalities as evident in language use, in order to provide insights into communication and personality variations among members in such a global setting. We mined the IBM Rational Jazz repository and used social network analysis (SNA) to cluster practitioners working across a set of teams into two groups (Top Members and Others, defined below). A number of SNA metrics (density, in-degree and closeness – refer to Section 3.2 for details) were then used to study practitioners' importance in knowledge diffusion, and these were triangulated with the exploration of particular linguistic usage. Finally, we performed further linguistic analysis to explore personality reflected in developers' messages, and related this evidence to records of activity in project history logs. We then relate the personality profiles of practitioners' to their involvement in knowledge diffusion. The findings from these activities are reported here.

This work makes multiple contributions. Firstly, we demonstrate that various analysis techniques may be systematically employed to deliver reliable and internally consistent results when studying human-related issues in empirical software engineering. Secondly, we extend previous work studying the communication and personality profiles of global developers. Finally, we provide recommendations for those tasked with leading globally distributed software development projects.

In the next section (Section 2) we present related work, and outline our specific research direction. We then describe our research setting in Section 3, introducing our procedures for data collection and measurement. In Section 4 we present our results, and discuss our findings. Section 5 then outlines implications of our results, and we identify potential threats to validity in Section 6. Finally, in Section 7 we draw conclusions.

2. Related work

We review related work in this section. In Section 2.1 we introduce research that has utilized repository data, and textual communications in particular, to explore human-centric aspects of software development processes. We then examine personality theories and models and how these have been applied to the study of software development practitioners' behaviors, in Section 2.2. Section 2.3 then addresses how personality may be studied from textual communications, with particular emphasis on the benefits of such analyses for global software development. Finally, we present our research questions in Section 2.4.

2.1. Analyses of textual communication

Software repositories and software history data have emerged as valuable sources of evidence of practitioners' interactions and communications [17,22] (questions over data quality notwithstanding [12]). Accordingly, researchers have exploited process artefacts such as electronic messages, change request histories, bug logs and blogs to provide unique perspectives on the activities occurring during the software development process [1,2]. In particular, previous work has focused heavily on studying communication patterns of software teams to explore and explain the knowledge diffusion process.

For instance, the Debian mailing list was used by Sowe et al. [23] to observe knowledge sharing among developers, with the authors finding that no specific individual dominated knowledge sharing activities in the Debian project. Crowston et al. [24] examined the work of the developers of five small open source software (OSS) projects using multiple explanatory approaches, including the principle of Bradford's law, and found that core groups of developers comprised only a small number of those contributing to the projects. Crowston and Howison's related study [25] found some OSS projects to be highly centralized (with just a few members communicating), and this pattern was especially pronounced for smaller projects. Additionally, it was revealed that most OSS projects had a hierarchical social structure, although there was