The role of mental models in FLOSS development work practices[a]

Kevin Crowston[1] and Barbara Scozzi[2]

1 Syracuse University, Syracuse (USA), crowston@syr.edu
2 Politecnique of Bari, Italy, bscozzi@poliba.it

Abstract. Shared understandings are important for software development as they guide to effective individual contributions to, and coordination of, the software development process. In this paper, we present the theoretical background and research design for a proposed study on shared mental models within Free/Libre Open Source Software (FLOSS) development teams. In particular, we plan to perform case studies on several projects and to use cognitive maps analysis to represent and compare the mental models of the involved members so as to gauge the degree of common knowledge and the development of a collective mind as well as to better understand the reasons that underlie team members actions and the way common mental models, if any, arise.

1. Introduction

This paper examines the role of shared mental models in work practices, i.e., the way people coordinate, communicate, learn and make decisions, and the way such models emerge within Free/Libre Open Source Software (FLOSS) development teams. The difficulties of distributed software development are emphasized in the literature on software development and distributed teams ([1]; [2]). The lack of a common organizational setting or functional background can make socialization, communication and coordination processes difficult, so reducing team performance and increasing the need for explicit coordination and learning among members ([3]; [4]). Languages and cultural differences can lead to misunderstanding, reducing the effectiveness of communications ([5]; [6, p.1]). Furthermore, because teams rely on computer-mediated communication, it can be difficult for members to develop the informal relationships and communications necessary to address interpersonal issues [7] . However, the case of FLOSS development presents an intriguing counter-example. Effective FLOSS development teams somehow profit from the advantages and evade the challenges of distributed software development [8].

To understand the origin of work practices, we focus specifically on the role of mental models (e.g., conceptions of the project, other team members, users, competitors or programming standards) that guide team members' behaviours and shape their actions. In this paper, we present the theoretical background and research design for a proposed study on shared mental models. The goals of the study are 1) finding evidence for the existence of shared mental models that shape team work practices and 2) trying to assess how such models arise. The

study is part of a larger research project aimed to identify the dynamics through which self-organizing distributed teams develop and work.

2. Theory: Mental models and software development

Shared mental models, as defined by Cannon-Bowers and Salas [9, p. 228] "are knowledge structures held by members of a team that enable them to form accurate explanations and expectations for the task, and in turn, to coordinate their actions and adapt their behavior to demands of the task and other team members". Research suggests that shared mental models help improve performance in face-to-face [10] and distributed teams [11]. Shared mental models can enable teams to coordinate their activities without the need for explicit communications ([12]; [13]). Without shared mental models, individuals from different teams or backgrounds may interpret tasks differently based on their individual backgrounds, so making collaboration and communication difficult [14]. The tendency for individuals to interpret tasks according to their own perspectives and predefined routines is exacerbated when working in a distributed environment, with its more varied individual settings.

Studies have identified the importance of shared understanding for software development ([15]; [16]). Curtis et al. [7, p.52], note that "a fundamental problem in building large systems is the development of a common understanding of the requirements and design across the project team." They go on to say that, "the transcripts of team meetings reveal the large amounts of time designers spend trying to develop a shared model of the design". The problem of developing shared mental models is likely to particularly affect FLOSS development, since FLOSS team members are distributed, have diverse backgrounds, and join FLOSS teams in different phases of the software development process ([17]; [18]). In short, shared mental models are important as guides to effective individual contributions to, and coordination of, the software development process.

Based on [19], we identify socialization, conversation and recapitulation as the means through which shared mental models are built. First, new members joining a team learn how they fit into the process being performed through socialization, e.g., by following a "joining script" [20]. Members need to be encouraged and educated to interact with one another so as to develop a strong sense of "how we do things around here". Barley and Tolbert [20 p. 100] similarly note that socialization frequently "involves an individual internalizing rules and interpretations of behaviour appropriate for particular settings". Second, conversation is critical in developing shared mental models. It is difficult to build shared mental models if people do not talk to one another and use common language. Meetings, social events, hallway conversations and electronic mail or conferencing are all ways in which team members can get in touch with what others are doing and thinking (interestingly though, many of these modes are not available to FLOSS teams). Finally, [19] stress the importance of recapitulation. To keep shared mental models strong and viable, important events must be "replayed", reanalyzed, and shared with newcomers. The history that defines who we are and how we do things around here must be continually reinforced, reinterpreted, and updated.

Most of the existing studies on shared mental models remain conceptual, though a few empirical studies in this area have investigated the relationship be-

tween team or organizational factors and the presence of shared mental models. However, while increasing attention has been lately devoted to the topics of knowledge creation, knowledge sharing and learning within the FLOSS development teams, (e.g. [22], [13]; [23]) to our knowledge no other studies have yet looked in detail at shared mental models for FLOSS development. For example, [23] focus on how knowledge is created and shared based on a case study, the KDE project. However, the study does not specifically examine which process aspects/practices are/are not shared and how extensive the sharing process is. [13] try to assess the importance of shared mental models for project coordination, but do not directly investigate the presence of shared mental models. Our project will therefore address this gap in the literature.

3. Research methodology

In this section, we describe the research methodology we will be adopting for the study. To achieve our goal, we plan to perform case studies on several FLOSS projects. In order to ensure that we are studying team large enough to have interesting work dynamics, we have selected projects with more than seven core developers. Different FLOSS projects are being examined and the attendant team members contacted. All the team members of the projects willing to take part to the study will be interviewed.

Interviews will be based on a semi-structured protocol designed to identify how team members interpret their role and the other members' roles, how they act and the reasons for their behaviours, eventual tacit norms and practices and the way such practices have arisen. To address the first set of concerns, the interview protocol will be organized in the following sections.

- *Developer demographics.* Descriptive data about developers, such as areas of expertise, formal role, years with the project, other projects in which they participate as well as perception of their role and other members' role in the project.
- *Project rules and norms.* Any explicitly stated norms or rule as perceived by developers.
- *Project environment and constraints.* The environment in which the team operates, constraints that they have to deal with, customers and competitors.
- *Development strategy.* The overall approach to project development.
- *Development process.* Process by which the software is developed (activities, dependencies, coordination mechanisms), tools and technology used for software development, as well as to submit and handle bugs, patches and feature requests, decision-making processes.
- *Team organization.* Team structure and specific team roles.
- *Socialization conversation and recapitulation.* Actions related to socialization, conversation and recapitulation as perceived by developers.

As to the latter aspects, in the interviews we will identify specific actions that can help building shared mental models. Therefore, the interview protocol will assess how and if socialization, conversation and recapitulation occur within the teams.

4. Analysis: Cognitive mapping techniques

Interview transcriptions will be analyzed using cognitive mapping techniques [24]. Cognitive maps are graphic tools used to represent concepts and ideas a person associates to a given issue (i.e., the topic of the map). Cognitive maps can be used with an explicative, a predictive, and/or a reflective purpose [25]. In this project, cognitive maps will be adopted for an explicative purpose, i.e., finding evidence of the existence of shared mental models, the way models shape team work practices and arise within FLOSS development teams.

Different methodologies have been proposed in the literature to develop cognitive maps. For data collection, the main approach consists of the administration of semi-structured interviews ([26], [27]). Some scholars have also developed more structured schemes [28] or models to make people self-interview, e.g. the self-Q technique by Bougon [29]. To develop maps, documents can also be used rather than interviews.

Based on the interview text, maps will be created by using a technique called Documentary Coding Method [30], which involves identifying the main concepts cited by the respondents and the relationships among them. A cognitive map is characterized by two ontologies, namely concepts and causal links among them [24]. Concepts represent ideas, opinions and key issues associated to the topic of the map. Concepts are linked by causal relationships, which can be mainly distinguished in cause/effect (which do not imply intentionality) or means/end relationships. Concepts are graphically represented by nodes and relationships by arrows. Concepts that represent the cause or the means to achieve a given goal are situated at the arrow's tail, concepts that represent the effect or the end at the arrow's head.

Different methodologies to analyze and compare maps also exist. In most studies quali-quantitative metrics, e.g. number of heads, tails, domain and centrality, are used [31]. Ad hoc metrics have also been defined to compare maps. The most well-known have been developed by [32]. In our study, maps will be analyzed by measuring/examining at least the following quali-quantitative metrics:

- *Map complexity.* It is given by the number of concepts on the map and the link/concept ratio.
- *Heads and Tails* Map heads are concepts represented by nodes that only have arrows going inside. They represent developers' final end/goal and/or the effects of their perception. Tails are concepts represented by nodes that only have arrows going outside. They explain/describe the causes of some perceptions and/or means to be adopted to achieve goals.
- *Domain and Centrality.* Domain and centrality provide information about the importance of concepts. In particular, a concept domain is given by the number of direct links. On the contrary, by the centrality analysis both direct or indirect links are used to assess the importance of concepts, so providing information on those concepts that are often unconsciously considered as the most relevant.
- *Sets.* Sets are groups of concepts that deal with a specific issue or topic. By counting the number of concepts mentioned in the maps for each set it is possible to assess the importance/complexity associated to the object of the set. We also will investigate the characteristics of concepts within sets (i.e. the number of heads, goals, and domain and centrality).

Through cognitive maps analysis we will be able to represent and compare the mental models of the developers about the project and project team so as to gauge the degree of common knowledge and the development of a collective mind as well as to better understand the reasons that underlie team members actions and the dynamics based on which common mental models, if any, arise ([24]; [33]; [34]). We can also examine the distribution of these models, e.g., which parts of the model are shared by most team members and which are common only among the core developers.

The main benefit that derives from the adoption of the maps is the ease of the analysis of different perspectives. The graphical representation facilitates identification of the key issues and the differences among different positions. Moreover, the adopted metrics facilitate the understanding of concepts or relationships not perfectly clear or conscious to individuals. These relationships can be more easily stressed than is the case when other qualitative tools (such as case studies or simple interviews) are used.

Of course, cognitive maps also present some drawbacks. In particular, the stage of the knowledge elicitation (interviews and codification of collected data) is the most critical. This observation is based on the difficulties we encountered in other projects during map development [e.g. 35]. Such consideration is also broadly discussed in the literature. As most of the qualitative research methodologies, the knowledge schemes of the interviewer (i.e., the researcher) can strongly influence the findings. By knowledge scheme we mean the culture, interests and experiences of the interviewer. The researcher's knowledge scheme can influence the way questions are asked (so influencing the answers) and, above all, the way data are analyzed. As already mentioned, there exist some techniques that try to reduce the subjectivity, but they introduce other sources of error [32]. For example, by providing an ex-ante defined list of possible constructs and concepts (though in some cases they can be extended by respondents) the answer possibility of the respondents is limited and can be biased. Based on our previous experience, we have decided to adopt semi-structured interviews so trying to minimize the effects of biases. Despite the drawbacks, we argue that cognitive maps can be effectively used to identify the mental models of the FLOSS team members and to assess if they are shared and how they affect work practices.

5. Expected results

The proposed study will have conceptual, methodological as well as practical contributions. The study fills a gap in the literature with an in-depth investigation of the mental models of FLOSS teams. Furthermore, we will use cognitive maps, which have never been used to investigate mental models within FLOSS development teams. The project will advance knowledge and understanding of FLOSS development and distributed work more generally by understanding the role and the extent of shared mental models within the teams. Understanding the dynamics of action in the teams is important to improve the effectiveness of FLOSS teams, software development teams, and distributed teams in general. As distributed teams are increasingly adopted by firms for a wide range of knowledge work, the study results can indeed be useful for managers willing to adopt distributed teams in their own organization.

References

[1] Bélanger, F. (1998). Telecommuters and Work Groups: A Communication Network Analysis. In Proceedings of the International Conference on Information Systems (ICIS) (pp. 365–369). Helsinki, Finland.

[2] Carmel, E., & Agarwal, R. (2001). Tactical approaches for alleviating distance in global software development. IEEE Software(March/April), 22–29.

[3] Finholt, T., Sproull, L., & Kiesler, S. (1990). Communication and Performance in Ad Hoc Task Groups. In J. Galegher, R. F. Kraut & C. Egido (Eds.), Intellectual Teamwork. Hillsdale, NJ: Lawrence Erlbaum and Associates.

[4] Robey, D., Khoo, H. M., & Powers, C. (2000). Situated-learning in cross-functional virtual teams. IEEE Transactions on Professional Communication(Feb/Mar), 51–66.

[5] . D. Herbsleb, A. Mockus, T. A. Finholt, and R. E. Grinter, "An Empirical Study of Global Software Development: Distance and Speed," in Proceedings of the International Conference on Software Engineering (ICSE 2001). Toronto, Canada, 2001, pp. 81–90.

[6] Massey, A. P., Hung, Y.-T. C., Montoya-Weiss, M., & Ramesh, V. (2001). When Culture and Style Aren't About Clothes: Perceptions of Task-Technology "Fit" in Global Virtual Teams. In Proceedings of GROUP '01. Boulder, CO, USA.

[7] Curtis, B., Walz, D., & Elam, J. J. (1990). Studying the process of software design teams. In Proceedings of the 5th International Software Process Workshop On Experience With Software Process Models (pp. 52–53). Kennebunkport, Maine, United States.

[8] Alho, K., & Sulonen, R. (1998). Supporting virtual software projects on the Web. 7th International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises (Wetice 98).

[9] Cannon-Bowers, J. A., & Salas, E. (1993). Shared mental models in expert decision making. In N. J. Castellan (Ed.), Individual and Group Decision Making (pp. 221-246). Hillsdale, NJ: Lawrence Erlbaum Associates.

[10] Rentsch, J. R., & Klimonski, R. J. (2001). Why do 'great minds' think alike?: Antecedents of team member schema agreement. Journal of Organizational Behavior, 22(2), 107–120.

[11] Sutanto, J., Kankanhalli, A., & Tan, B. C. Y. (2004). Task coordination in global virtual teams. Paper presented at the Twenty-Fifth International Conference on Information Systems, Washington, DC.

[12] Crowston, K., & Kammerer, E. (1998). Coordination and collective mind in software requirements development. IBM Systems Journal, 37(2), 227–245.

[13] Espinosa, J. A., Kraut, R. E., Lerch, J. F., Slaughter, S. A., Herbsleb, J. D., & Mockus, A. (2001). Shared mental models and coordination in large-scale, distributed software development. Paper presented at the Twenty-Second International Conference on Information Systems, New Orleans, LA.

[14] Dougherty, D. (1992). Interpretive barriers to successful product innovation in large firms. Organization Science, 3(2), 179–202.

[15] Levesque, L. L., Wilson, J. M., & Wholey, D. R. (2001). Cognitive divergence and shared mental models in software development project teams. Journal of Organization Behavior, 22, 135–144.

[16] Sagers, G. W., Wasko, M. M., & Dickey, M. H. (2004). Coordinating Efforts in Virtual Communities: Examining Network Governance in Open Source. Paper presented at the Tenth Americas Conference on Information Systems, New York, NY.

[17] Edwards, K. (2001, June 11–12). Epistemic communities, situated learning and Open Source Software development. Paper presented at the Epistemic Cultures and the Practice of Interdisciplinarity Workshop, NTNU, Trondheim.

[18] Gasser, L., & Ripoche, G. (2003). Distributed Collective Practices and F/OSS Problem Management: Perspective and Methods. Conference on Cooperation, Innovation & Technologie (CITE2003), University de Technologie de Troyes, France.

[19] Brown, J. S., & Duguid, P. (1991). Organizational learning and communities-of-practice: Toward a unified view of working, learning, and innovation. Organization Science, 2(1), 40–57.

[20] von Krogh, G., Spaeth, S., & Lakhani, K. R. (2003). Community, Joining, and Specialization in Open Source Software Innovation: A Case Study. Research Policy, 32(7), 1217–1241.

[21] Barley, S. R., & Tolbert, P. S. (1997). Institutionalization and structuration: Studying the links between action and institution. Organization Studies, 18(1), 93–117.

[22] Lanzara, G. F., & Morner, M. (2003). The Knowledge Ecology of Open-Source Software Projects. Paper presented at the 19th EGOS Colloquium, Copenhagen.

[23] Hemetsberger, A., & Reinhardt, C. (2004). Sharing and Creating Knowledge in Open-Source Communities: The case of KDE. Paper presented at the The Fifth European Conference on Organizational Knowledge, Learning, and Capabilities, Innsbruck, AU.

[24] Pidd, M. (1996). Tools for thinking modeling management science. Chichester: John Wiley and Sons.

[25] Codara, L. (1998). Le mappe cognitive, uno strumento di analisi per la ricerca sociale e per l'intervento organizzativo. Rome: Carocci Editore.

[26] Eden, C. (1988). Cognitive mapping. European Journal of Operational Research, 36, 1–13.

[27] Laukkanen, M. (1998). Conducting causal mapping research: Opportunities and challenges. In C. Eden & J.-C. Spender (Eds.), Managerial and organizational cognition: Theory, methods and research (pp. 168–191). Thousand Oaks, CA: Sage.

[28] Cossette, P. (1994). Cartes cognitives et organizations. Paris: Eska.

[29] Bougon, M. G. (1983). Uncovering cognitive maps: the Self-Q technique. In G. Morgan (Ed.), Beyond method: Strategies for social research. Beverly Hills, CA: Sage.

[30] Wrightson M.T., (1976). The documentary coding method, in Axelrod R. (ed.), Structure of Decision. The cognitive maps of political elites, Princeton (NJ), Princeton University Press.

[31] Cossette P., Audet M., 1992, Mapping of an idiosyncratic schema, Journal of Management Studies, vol.29, n.3, pp. 325-347.

[32] Markoczy L., Goldberg J., 1995, A method of eliciting and comparing causal maps, Journal of management, vol.21 n.2, pp. 305-333.

[33] Carley, K. M. (1997). Extracting team mental models through textual analysis. Journal of Organizational Behaviour, 18, 533–558.

[34] Carley, K. M., & Palmquist, M. (1992). Extracting, representing and analyzing mental models. Social Forces, 70(3), 601–636.

[35] Albino V., Kuhtz S., Scozzi B. (2004). Cognitive maps on sustainable development in industrial districts: a pilot study, in Johansson B., Karlsson C., Stough, R.R. (eds.), Industrial Clusters and Inter-Firm Networks, Elgar Publ., 149-170, ISBN: 1-84542-010-1.