

The Benefits and Limitations of Knowledge Management in Global Software Development

Torgeir Dingsøy¹, Knut-Helge Rolland², M. Letizia Jaccheri²,
¹SINTEF ICT, ²Dept. of Computer and Information Science NTNU,

torgeir.dingsoyr@sintef.no, [knutrr|letizia}@idi.ntnu.no](mailto:{knutrr|letizia}@idi.ntnu.no)

Abstract

The role of knowledge management practices and tools in global software development will be explored by empirical investigations. These investigations will look at global software development processes by taking into special account multicultural factors and will rely on both quantitative methods for project selection and qualitative methods for in depth study of the single project contexts.

1. Introduction

Software development work is becoming global in the sense that development work is increasingly carried out in teams that are geographically separated across national boundaries and cultures. This trend is captured in commercial software production where parts of the development activities often are outsourced to low-cost countries, as well as in ‘open source software’ projects where development sometimes is global in scope.

The problem we want to discuss in this position paper are the fact that while software development becomes increasingly distributed and global in nature, much of the techniques and tools for improvement still assume that individuals are co-located. For example, the principle of pair-programming in the XP approach was established to improve learning and knowledge-exchange among programmers [2] and traditional code inspection methods often assume face-to-face interaction in terms of more or less formal meetings.

Software development conducted in a distributed fashion is often referred to as ‘Global Software Work’ (GSW). More accurately, Sahay defines GSW as “software work undertaken at geographically separated locations across national boundaries in a coordinated fashion involving real time and asynchronous interaction” [9]. Given these characteristics, it becomes clear that GSW involves different kinds of complexities compared to traditional software development where members of the project are more or less co-located and are therefore able to share their experiences through face-to-face communication.

Drawing from past experience reported in the literature on global software development, we can

single out some factors that are of profound relevance. Previous studies have shown how cultural issues are directly related to and influence how software is developed and managed. It has been shown that teams from different cultures tend to prefer dissimilar approaches to architectural design. For example, there are differences in how abstractions are chosen and what architectural patterns are used [3]. Clearly, an understanding of these issues would be of profound importance for managing global software projects successfully. Moreover, recent research has underscored the challenge of adopting common and standardized practices and tools in global software work and the need for developing a specific competence to do global product development [8].

However, while the above literature points at some practices and competencies for successfully conducting global software development, little is said about how such competencies can be learned and adopted in other organizations and domains. Considering the limited capability of learning in many software development projects and organizations [6], it is then important to investigate how organizations and individuals can facilitate learning both within specific global software development initiatives and between different projects.

In this position paper we will outline a research design for studying the limits and benefits of Knowledge Management (KM) practices and tools for improving organizations’ capabilities of continuous learning in global software development projects.

2. Arenas to learn from

Global software development has existed in some forms since the early eighties in both volunteer contexts and commercial ones, such as banking applications. One example of an environment that does global software development is the Open Source community [5]. The Open Source community consists of a myriad of projects, which vary in number of developers and their organization roles, kind and size of developed software, degree of involvement of commercial actors, popularity, vitality, degree of success, and duration of the project. The slogan “the success of open sources projects” seems to stem from the big success of projects such as Linux and Apache [7].

The software research community has devoted a lot of attention to the open source world, as can be deduced for example from the series of open source workshops in the ICSE context. Many are the questions which are of interest of the software community when looking at the open source world. Which are the successful projects and how do we define this notion of success in a not profit world? Is success a function of project vitality? Is success a function of the impact the open source software has on the commercial market, such as for example the Linux operating system?

Given that successful open source project are the focus of our interest, how and what do we want to learn from them and how do we want to transfer the learned knowledge into commercial contexts or even into educational ones? One possibility is that of making hypotheses about the reasons or causes for success of open source projects, and then trying to describe these reasons in order to disseminate them. How does Open source project implement knowledge management? Is it the software process model of open source project which is the cause of success? If we regard a process model as descriptions of tasks, practices, responsibilities, tools, and document types, which of these elements is of most importance for project success?

3. Knowledge Management: Benefits and limitations

Learning in the context of software development has often been limited to different kinds of information technology support for learning and knowledge-exchange [4]. For example, there has been much focus on reusing life-cycle experience, processes, and products for software development in terms of having an ‘Experience Factory’ [1]. Likewise, the information systems literature has emphasized introduction of ‘Knowledge Management Systems’ in order to support organisation-wide knowledge-exchange and learning. Arguably, these technologies and knowledge-sharing practices can play important roles in facilitating learning in software organizations. However, as there exists a wide range of different KM practices and tools, there is thus a growing need for investigating empirically what kind of KM that is relevant for global software development. A salient point related to global software development is also that systematic practices and tools for KM are perhaps even more relevant for software development in (globally) distributed settings there lack of more informal face-to-face interaction must be substituted with other ways of coordinating work and ways for facilitating mutual learning between distributed development teams. On the other hand, establishing KM practices and tools across different cultural settings might also be especially challenging due to cultural differences in how knowledge is formed, structured and utilized in different countries [10].

Thus, the literature seems to suggest that there are both potential benefits and limits for improving learning through KM in global software development.

Research should be conducted in order to illuminate these benefits and limits in more detail, and for increasing the understanding of global software development in general.

4. Research design

In order to study the new landscape of software development, it is relevant to draw from different research disciplines and perspectives in a cross disciplinary and multi-perspective approach. On the research method side, we will be open to both the empirical software engineering community and the community of research that focuses on explaining software development in relation to a broader social, organisational, and economical context.

Our investigations will look at global software development processes by taking into special account multicultural factors. On the one hand, there is a need to use descriptive statistics to get an orienteering map in the complex world of global software development. If we look at open source projects for example, it is of great help to classify them by evaluation parameters, like number of active users, lines of code, age, vitality, number of represented countries which help us to compare them and to choose those which we want to study in depth.

In this way, we will select a couple of case studies to capture and describe learning practises and networks in real-world settings, how tools, techniques, or concepts are employed. One case will involve Open Source development, and one will involve global software development in a commercial setting. We will select cases involving software development environments in at least three countries, and projects that run over a period of at least two years.

Data from the case studies will be collected according to the following two principles:

- Multiple sources of evidence. We will collect data from several sources, such as documentation, archival records, interviews, direct observations, participant-observation, and physical artefacts. Analysis will be assisted by using the qualitative data analysis tool Nvivo.
- Case study database. We will document the data collected in the case studies in terms of notes, documents, tabular materials, narratives, photographs, and video, and organise it in a case study database. For this purpose we will use the facilities offered by eRoom, which provides a shared, secure workplace on the Web for the project team.

5. Conclusion

In this position paper, we have argued that software development is increasingly global, which makes the complexity of software development larger due to changing technologies, methods, geographical location and multicultural arrangements. Knowledge

management tools have the objective to reduce the problems of complexity in global software development. Through investigations of practices in organizational learning, we seek to reach a better understanding of how knowledge management tools and learning issues in general function in global software work. Such an understanding can give the software engineering community better abilities to see what kind of knowledge management practices and tools that are suitable for global development, as well as new insights on key practices from the open source community.

8. References

- [1] V. R. Basili, Caldiera, G and Rombach, H.D., "The Experience Factory," in *Encyclopedia of Software Engineering*, vol. 1, J. J. Marciniak, Ed.: John Wiley, 1994, pp. 469-476.
- [2] K. Beck, *Extreme Programming Explained*: Addison-Wesley, 190 pp., 2000.
- [3] G. Borchers, "The Software Engineering Impacts of Cultural Factors on Multi-cultural Software Development Teams," presented at 25th International Conference on Software Engineering (ICSE'03), 2003.
- [4] T. Dingsøy and R. Conradi, "A Survey of Case Studies of the Use of Knowledge Management in Software Engineering," *International Journal of Software Engineering and Knowledge Engineering*, vol. 12, num. 4, pp. 391 – 414, 2002.
- [5] J. Feller and B. Fitzgerald, *Understanding Open Source Software Development*: Addison Wesley, 211 pp., 2002.
- [6] K. Lyytinen and D. Robey, "Learning Failure in information systems development," *Information Systems Journal*, vol. 9, pp. 85-101, 1999.
- [7] A. Mockus, R. T. Fielding, and J. D. Herbsleb, "Two case studies of open source software development: Apache and Mozilla," *ACM Transactions on Software Engineering and Methodology (TOSEM)*, vol. 11, num. 2, pp. 309-346, 2002.
- [8] W. J. Orlikowski, "Knowing in Practice: Enacting a Collective Capability in Distributed Organizing," *Organization Science*, vol. 13, num. 3, pp. 249-273, 2002.
- [9] S. Sahay, "Global software alliances: the challenge of 'standardization'," *Scandinavian Journal of Information Systems*, vol. 15., pp. 3-21, 2003.
- [10] G. Walsham, "Knowledge Management: The Benefits and Limitations of Computer Systems," *European Management Journal*, vol. 19, num. 6, pp. 599-608, 2001.