Workshop
Agile Knowledge Sharing for Distributed Software Teams

Björn Decker[1], Hans-Jörg Happel[2], Walid Maalej[3]

[1] empolis GmbH
Europaallee 10. D-67657 Kaiserslautern
Bjoern.Decker@empolis.com

[2] FZI Forschungszentrum Informatik
Haid-und Neu-Str. 10-14, D-76131 Karlsruhe
happel@fzi.de

[3] Technische Universität München
Boltzmannstr. 3, D-85748 Garching
maalejw@in.tum.de

1 Motivation

The nature of software engineering projects is rather complex. They involve people with different backgrounds, working in different phases and activities. Software engineering is a field where constant technology change takes place, rendering the work of involved people highly dynamic, in the sense that discovering solutions for new problems belongs to the daily tasks. Knowledge in software engineering is diverse and its proportions are immense and continuously expanding. Important needs that drive the use of knowledge management are capturing and sharing process and product knowledge, acquiring knowledge about the application domain and new technologies, as well as knowing who knows what.

In order to address these issues, software engineering practitioners have adopted various forms of organizational practices, collaboration technologies and knowledge management systems such as lessons learned databases. However, it turns out that those often centrally organized and heavyweight solutions are difficult to apply and maintain in distributed development settings with frequently changing collaborators from different organizations [RM06]. Accordingly, such dynamic distributed development settings – as also found in the Open Source communities – raise new challenges for collaboration and knowledge sharing for software teams:

- Distributed development hinders knowledge sharing because of reduced communication bandwidth. Despite progress in communication technology, distributed teams have less context information and encounter more awareness difficulties. Research has shown that distributed teams are less efficient than collocated ones [HM03, SB98].

- Agile methodologies embrace tight collaboration and informal communication, manifesting in practices such as frequent “Scrum” meetings or pair programming. These practices dismiss “unnecessary” formal overhead like extensive documentation.

Traditional centralized knowledge management solutions fail to address these challenges. They require large upfront configuration and investment as well as stable, long-term environments. However, in dynamic software projects, developers avoid spending extra effort for following formal and extensive knowledge management policies. Knowledge Management methods and tools have to adapt themselves with the current development practices.

2 Topics

Despite of the practical relevance, current research around these topics is fragmented. Various disciplines and communities, such as the Artificial Intelligence, Knowledge Management and Software Engineering ones, have overlapping interests. However proposed approaches are mostly undertaken from only one perspective. Moreover, different research methods such as empirical, process-oriented or tool-oriented ones, lead to unrelated, partial and sometimes even contradictory results.

The workshop “Agile Knowledge Sharing for Distributed Software Teams” [AK07] explores enablers of new, agile approaches for coordination and knowledge sharing in distributed teams. Major topics addressed at the workshop are:

- Lightweight and unobtrusive tools for knowledge capture and sharing in distributed teams.

- Methods for supporting knowledge sharing in agile and/or distributed development teams.

- Developer observation tools and frameworks for context awareness.

- Tools for intelligent developer assistance, such as search and recommendation tools.

- Empirical studies of developers’ collaboration and information behaviour.

- Scientific analysis of the relation between agile processes and knowledge management (e.g. witch activities in agile processes enable knowledge capture and sharing).

In the workshop, we aim at bringing together practitioners and researchers from various disciplines and backgrounds. Thereby, one major goal is to establish a community around the crosscutting topic of Agile Knowledge Sharing in Distributed Software Teams.

References

[AK07] Aknowledge Plattform http://www.aknowledge.org

[HM03] Herbsleb, James D.; Mockus, Audris: Formulation and preliminary test of an empirical theory of coordination in software engineering. In: Proceedings of the 9th European software engineering conference held jointly with 11th ACM SIGSOFT international symposium on Foundations of software engineering, ACM Press, 2003, pp. 138-137.

[SB98] Seaman, Carolyn B.; Basili, Victor R.: Communication and Organi-zation: an Empirical Study of Discussion in Inspection Meetings. In: IEEE Transactions on Software Engineering 24 (1998), No. 7, pp. 559-572.

[RM06] Ramesh, B., Cao, L., Mohan, K., and Xu, P. 2006. Can distributed software development be agile?. Commun. ACM 49, 10 (Oct. 2006), 41-46.