

Tool Support for Globally Distributed Scrum

Morten Esbensen
It University of Copenhagen
Rued Langgaards Vej 7
2300 Copenhagen S, Denmark
mortenq@itu.dk

Jakob E. Bardram
It University of Copenhagen
Rued Langgaards Vej 7
2300 Copenhagen S, Denmark
bardram@itu.dk

ABSTRACT

Global software development promises advantages compared to collocated software development however, it also brings challenges associated with the introduced distances. To harvest the advantages and handle the challenges, appropriate processes and tools need to be employed. Through the use of agile development methodologies such as Scrum, global software development has successfully implemented such processes. In this position paper we argue, that to fully capitalize on the advantages of Scrum in global software development, appropriate tools needs to be developed. We propose that the use of activity-based tools can help leverage the advantages of Scrum and overcome the challenges of distributed collaboration and illustrate this proposal with examples from an observational study.

Author Keywords

Global software development; Scrum; activity-based computing; tool support.

ACM Classification Keywords

H.5.3. Group and Organization Interfaces: Collaborative computing, Computer-supported cooperative work

INTRODUCTION

Global software development (GSD) promises advantages such as lower development costs, wider pool of skilled labour and around the clock development [7]. The introduction of distance however also poses several challenges and successfully harvesting the advantages requires companies to implement strategies and make use of appropriate tools.

Agile practices have at the same time become a favorite method for many companies. Promoting collaboration and communication over documentation and negotiations, these practices seek to improve efficiency in delivering working software on time that meets the expectations of clients. One such agile practice is Scrum. Adhering to the agile manifesto of collaboration and adaptation to change [4], Scrum proposes a software process driven by close collaboration between client and vendor and in between the development

team. Furthermore, Scrum proposes a number of meetings being carried out throughout the process and a process for managing and implementing requirements in an iterative manner.

One particular interest to researchers have been the use of agile methods such as Scrum in GSD (cf. [9, 11]). Despite the seemingly conflict – Scrum advocates collocation while GSD introduces distances – global Scrum has proven successful in many cases. Modifying Scrum methods to fit with the distances of time and space that GSD introduces, companies have successfully harvested some of the advantages of Scrum in a distributed environment.

GSD AND SCRUM

While there seems to be problems related to the introduction of Scrum in a global context, several studies point out that adapting Scrum processes to a distributed environment can improve GSD [5], and it has even been described as essential to overcome communication and collaboration problems [13].

Adapting the traditional Scrum processes and practices have been observed in several instances. With distribution over several time-zones the daily Scrum meeting has been changed by moving it from the start of the day to a suitable time for a distributed Scrum team or meetings have been held at the different locations and afterwards synchronized [8].

In general, there has been a focus on studying global Scrum and the processes but less on the tools. This is despite the fact that tools have been identified as a key research area within agile GSD [6, 8].

TOOLS FOR DISTRIBUTED SCRUM

We argue that there is a need for research in to how tools can support global Scrum such that the tools and processes support each other, and that activity-based computing supports the existing practices of Scrum while leveraging some challenges of GSD.

Activity-based computing (ABC) is a computational paradigm that seeks to move the basic computational unit from files and applications to *activities* [2]. Activities are reflections of the actual task of a user and this paradigm thus seeks to support users in their actual intentions of work rather than just provide support for the tools they use. The introduction of activity-based tools to knowledge work can help reduce the complexities of working with fragmentation – the fact that knowledge workers have to deal with and switch between many different artifacts and activities [1].

ABC has been introduced to GSD before [3, 14], however we argue that ABC carries specific advantages to distributed *agile* software development. In particular, we argue for three advantages of using ABC to support agile GSD:

1. Software development, global or local, is a highly complex task involving numerous artifacts, tools and services. This fact requires actors to engage in a large amount of configuration work – the work involved in locating, opening and arranging such items [10]. One of the advantages of activity-based applications is their ability to support this form of configuration, allowing the user to focus on the activity at hand rather than the configuration of the needed tools.
2. ABC seeks to move the basic computational structure from the files and application to a unified structure – the activity. This principle supports the idea of distributed collaboration very well. Files and applications as we know them are highly linked to the personal (local) computer. Activities on the other hand are intrinsic collaborative and consider not only the artifacts and tools involved, but also the community around them.
3. Activity-based applications adhere to the same principles that underlie the processes of Scrum. ABC’s holistic representation of the actual intention of work over applications or files, fits very well with the agile manifestos emphasis on interactions and collaboration over processes and documentation.

In summary, we propose that using an activity-based approach to build tools for global Scrum thus helps shape these tools to support global Scrum.

ILLUSTRATION

During a two week study of a distributed Scrum team working from Denmark and India, we were able to observe how a distributed Scrum collaboration unfolded. Indeed, with central Scrum processes such as daily meetings, retrospective meetings and estimation meetings used, we observed a sprint where tasks were implemented on time and collaboration happened with close coordination and communication between different sites. Still however, we observed instances where the tool support seemed inadequate forcing overhead coordination and communication.

User Story Creation

One of the areas of interest of our observations was the lifecycle of a user story. The Scrum team we observed was fully distributed (e.g. consisting of people in both Denmark and India) with the Scrum master located in Denmark. Creating new user stories was the responsibility of the Scrum master. The process of creating a user story involved several non-integrated tools; the story was written in Microsoft Word, analysis of cost-benefit was noted in Microsoft Excel and the final user story was then transferred to Application Lifecycle Manager – a task-tracking application. Writing the user story also required the Scrum master to talk to a user of the system and consult the lead developer on technical details.

The creation of a user story also involved a discussion of the related UI implementation. However, as all UI developers were located in India, the Scrum master decided not to do so despite the fact that the a similar UI solution had been implemented by the Indian UI developers before.

We propose that the use of ABC in this scenario could help integrate all the required tools and documents – from document editing to communication tools – decreasing the amount of configuration work and leaving more time and focus on the actual writing of the user story.

Scrum Board

The Scrum board is a tool used in many Scrum processes to track the progress of work. It serves as an important awareness tool in the process and it has also been classified as an important boundary object and a help in the articulation work process [12]. The Scrum board however is a highly local tool. We observed this as well in the collaboration. No Scrum board was used and when asked why no board was used, the simple answer was because of the distance.

The need for awareness of team progression was still there though, and to account for the lack of a scum board, the lead developer located in Denmark coordinated over instant messaging the the team in India before the end of each day.

In this example we propose, that the use of an activity-aware distributed Scrum board that integrates with the existing software tools and infrastructures, could provide awareness to the Scrum team thus decreasing the need for coordination.

In summary, these examples illustrate that even when Scrum processes are in place the lack of appropriate tools create situations where overhead coordination and communication is needed to drive the collaboration forward.

CONCLUSION

Global software development can benefit largely from adapting the Scrum to fit with the introduced distances. Still, not much research have been conducted in to adequate tool support. In this position paper, we argue for the need of dedicated distributed Scrum tools to support this form of collaboration. We propose that activity-based tools can help leverage the advantages of Scrum and overcome the challenges of distributed collaboration.

ACKNOWLEDGEMENTS

This research has been funded by the Danish Agency for Science, Technology and Innovation under the project Next Generation Technology for Global Software Development, #10-092313.

REFERENCES

1. Balakrishnan, A. D., Matthews, T., and Moran, T. P. Fitting an activity-centric system into an ecology of workplace tools. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '10*, ACM (New York, NY, USA, 2010), 787–790.
2. Bardram, J., Bunde-Pedersen, J., and Soegaard, M. Support for activity-based computing in a personal

- computing operating system. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '06, ACM (New York, NY, USA, 2006), 211–220.
3. Bardram, J. E. Activity-based computing support for agile and global software development. In *CSCW 2008 Workshop on Supporting Distributed Team Work (VRC2008)* (San Diego, CA, USA, 2008).
 4. Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R. C., Mellor, S., Schwaber, K., Sutherland, J., and Thomas, D. Manifesto for agile software development. <http://agilemanifesto.org>, 2001. [Online - accessed 01.12.2013].
 5. Bose, I. Lessons learned from distributed agile software projects: A case-based analysis. *Communications of the Association for Information Systems* 23, 34 (2008).
 6. da Silva, F. Q. B., Prikladnicki, R., Frana, A. C. C., Monteiro, C. V. F., Costa, C., and Rocha, R. An evidence-based model of distributed software development project management: results from a systematic mapping study. *Journal of Software: Evolution and Process* 24, 6 (2012), 625–642.
 7. Herbsleb, J., and Moitra, D. Global software development. *Software, IEEE* 18, 2 (mar/apr 2001), 16–20.
 8. Hossain, E., Babar, M., and young Paik, H. Using scrum in global software development: A systematic literature review. In *Global Software Engineering, 2009. ICGSE 2009. Fourth IEEE International Conference on* (july 2009), 175–184.
 9. Hossain, E., Bannerman, P. L., and Jeffery, R. Towards an understanding of tailoring scrum in global software development: a multi-case study. In *Proceedings of the 2011 International Conference on Software and Systems Process, ICSSP '11*, ACM (New York, NY, USA, 2011), 110–119.
 10. Houben, S., Esbensen, M., and Bardram, J. A situated model and architecture for distributed activity-based computing. In *In Proceedings of the 2nd International Workshop on Model-based Interactive Ubiquitous Systems*, MODIQUITOUS '12 (2012), 913.
 11. Paasivaara, M., Durasiewicz, S., and Lassenius, C. Distributed agile development: Using scrum in a large project. In *Global Software Engineering, 2008. ICGSE 2008. IEEE International Conference on* (aug. 2008), 87–95.
 12. Pries-Heje, L., and Pries-Heje, J. Why scrum works: A case study from an agile distributed project in denmark and india. In *Agile Conference (AGILE), 2011* (aug. 2011), 20–28.
 13. Ramesh, B., Cao, L., Mohan, K., and Xu, P. Can distributed software development be agile? *Commun. ACM* 49, 10 (oct 2006), 41–46.
 14. Tell, P., and Ali Babar, M. Supporting activity based computing paradigm in global software development. In *Automated Software Engineering (ASE), 2011 26th IEEE/ACM International Conference on*, IEEE (2011), 508–511.