Meeting Intensity as an Indicator for Project Pressure

Exploring Meeting Profiles

Olga Liskin, Kurt Schneider, Stephan Kiesling
Software Engineering Group
Leibniz Universität Hannover, Germany
E-Mail: {olga.liskin, kurt.schneider, stephan.kiesling}@inf.uni-hannover.de

Simone Kauffeld
Dept. of Industrial/Organizational and Social Psychology
Technische Universität Braunschweig, Germany
E-Mail: s.kauffeld@tu-bs.de

***Abstract*—Meetings are hot spots of communication and collaboration in software development teams. Both distributed and co-located teams need to meet for coordination, communication, and collaboration. It is difficult to assess the quality of these three crucial aspects, or the social effectiveness and impact of a meeting: Personalities, psychological and professional aspects interact. It is, therefore, challenging to identify emerging communication problems or to improve collaboration by studying a wealth of interrelated details of project meetings.**

However, it is relatively easy to count meetings, and to measure when and how long they took place. This is objective information, does not violate privacy of participants, and the data might even be retrieved from project calendars automatically. In an exploratory study, we observed 14 student teams working on comparable four-month projects. Among many other aspects, we counted and measured meetings. In this contribution, we compare the meeting profiles qualitatively, and derive a number of hypotheses relevant for software projects.

***Index Terms*—team communication; meeting culture; software development**

I. Introduction

Software engineering is a complex task. In most cases, development tasks cannot be fulfilled by an individual person and collaboration is indispensable to success [1]. Insufficient communication, e.g. about requirements, is responsible for a big part of project failures [2].

Meetings play a crucial role for communication. They constitute essential intersections of information, both for co-located as well as distributed teams. Kauffeld has shown that the interactions in team meetings have an influence on the team's and company's overall success [3] [4] [5]. Communication, collaboration, and coordination could be improved, if we knew more about meeting culture.

To gain insights about meeting behavior, it is necessary to observe real projects and understand the teams and the different kinds of meetings [6]. There is a tension between detailed observations and light-weight ones. While detailed observations can provide the best insights, they are difficult to gain and possibly even tedious for the observed team members. Light-weight observations are a lot simpler to conduct but might be too shallow. In this paper, we are interested in light-weight examinations. The time and duration of meetings can be measured easily with practically no burden for the team members. We believe that these measures provide a good indicator for the circumstances in a project, like pressure, and offer quick insights or even warning signs for management. For research, these measures offer a lightweight tool to characterize and categorize communication in projects.

II. Cohort of 14 Comparable Projects

Since 2004, the Software Engineering Group at LUH regularly conducts a software project course for third-year bachelor students. Teams of five students develop a software product for a customer. They need to elicit requirements, design, implement, and test their product. In an acceptance test, the customer evaluates the software product.

In the fall term 2012, 68 students participated in 14 teams (13 teams of 5 students each, one team of 3 students). Some teams had the same task assigned in parallel, but had to work independently of each other.

The teams are directed by a phase model with quality gates. Quality gates are mandatory checks for all teams at the end of a phase. Teams plan, organize, and decide within this coarse grid. Phases are requirements, design/prototyping, and implementation & testing. All students have similar backgrounds; a number of courses on programming and software engineering/software quality are prerequisites.

Some aspects of these software projects are designed to be "close to real projects": Customers have limited time to spend for interviews, validation, etc. Teams have a technical coach who can be asked for help. However, coach and customer are never the same person; coaches know nothing about the project, only about technologies. There is no "advisor", since autonomous and responsible behavior is one of the core learning goals of the course. We have reported on the didactical setup of the software project course before [7]. The notion of "requirements compliance" was introduced as a measure of an important success criterion [8].

III. Study Outline

The explorative study reported here is part of a larger longitudinal study. One of its goals was to test the feasibility of data solicitation and measurement under the given conditions. Besides that, the distribution of meetings was observed as a

Accepted for publication by IEEE. © 2013 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/ republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

potential indicator for communication or collaboration style. In this paper, selected observations are presented and interpreted.

Teams were free to meet at any time and in any location; virtual meetings via skype, chat, or even email were an option. Each team's project leader had to report held meetings, their duration, and goals via an online tool. Teams had to report each team meeting that was (or was supposed to be) attended by all team members. This is a subclass of all meetings.

We decided to restrict our count to that subclass because spontaneous "meetings" of two or three people are difficult to distinguish from informal chat. Reporting them all would also put a heavy burden on teams, and tend to infringe privacy of individuals. Plenary meetings, on the other side, tend to be planned in advance and publicly scheduled in project calendars. In fact, our teams where invited (but not forced) to use a planning calendar integrated with the mandatory online reporting tool. The exploratory study mainly served to prepare and fine-tune the subsequent longitudinal study that we plan to conduct next year.

Data collection used different sources:

- Student teams had to report time and duration of (plenary) meetings via an online tool.
- Customers were asked to rate the success and progress of teams, as well as the conformance of implemented features with their requirements. In particular, they compared final products to their own requirements at the end of the project – which may have evolved from the initial requirements under the impression of discussions or prototypes. This concept was called "requirements compliance" and is described by Schneider et al. [8].
- At the end of each project, a post-mortem session using the LIDs technique [9] was held, giving students the opportunity to reflect on their projects. As a side effect, drastic events and their impact can be recognized and considered in data interpretation.

IV. Comparing Meeting Profiles

We have created profiles showing *meeting intensity* as *hours spent in plenary meetings per day*. These profiles, like Figure 1, allow to characterize a team's meeting behavior.

For each team, all meetings are visualized as lines over a common timeline. Quality gates are depicted as triangles in each of those timelines; they occurred at the same time for all teams. Our experience from eight years of conducting these projects taught us that quality gates cause pressure.

Some team profiles refer to teams who had the same customer and the same project task to solve. Those teams never had a meeting together, neither with nor without their customers. Figure 1 shows the meeting profiles of two teams with the same project task.

Although there are differences, both profiles look similar. Only one meeting (in A) took longer than five hours. There are a rather low number of meetings spread over the entire project. Meetings tend to be shorter and fewer in the last phase (during implementation and test) than before. These teams considered their projects doable and not too difficult. We call this profile "moderate meetings".

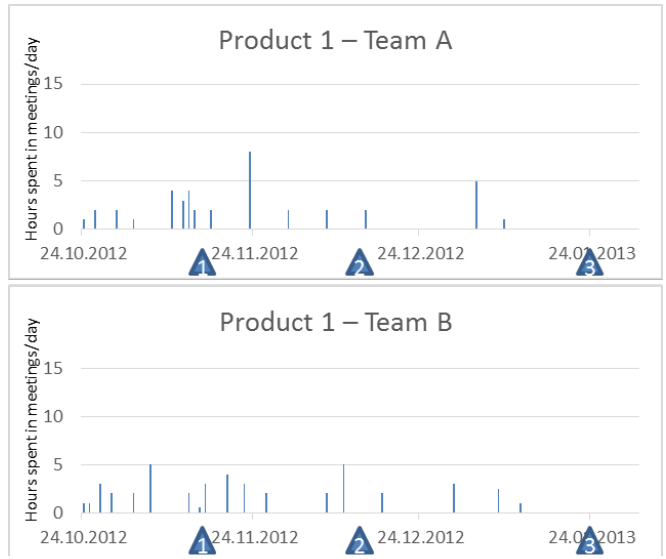

Figure 1: Two teams with same project task (Product 1) The vertical axis shows a line for each meeting, with its duration in hours.

The project in Figure 2 had a different topic, a real external customer, but again the moderate meeting profile. However, there is a slight increase in meetings intensity in the implementation and test phase. Many other projects had a moderate meeting profile.

Comparing Figure 2 (Team A) with Figure 3Figure 3 (Team B), it is interesting to see how different teams dealt with the same topic: While there is a moderate beginning, meeting frequency and durations soar in the final phase. This is surprising, since writing code should not necessarily benefit from meeting or spending time together.

Figure 4 shows the opposite extreme on the spectrum of meeting profiles. Shortly before quality gate 3 (and even after it, before customer acceptance test), there is a surge of meetings. Note the excessive duration of some meetings (over 12 hours).

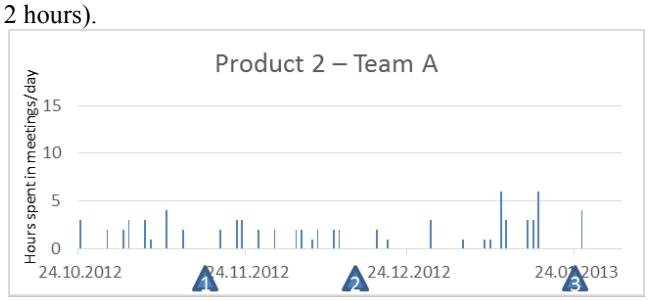

Figure 2: A different task and team with "moderate meeting profile"

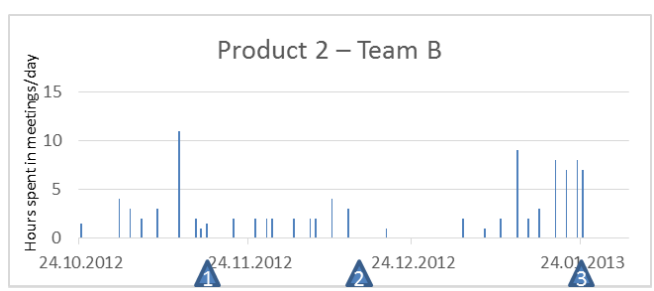

Figure 3: Many long meetings during implementation and testing phase

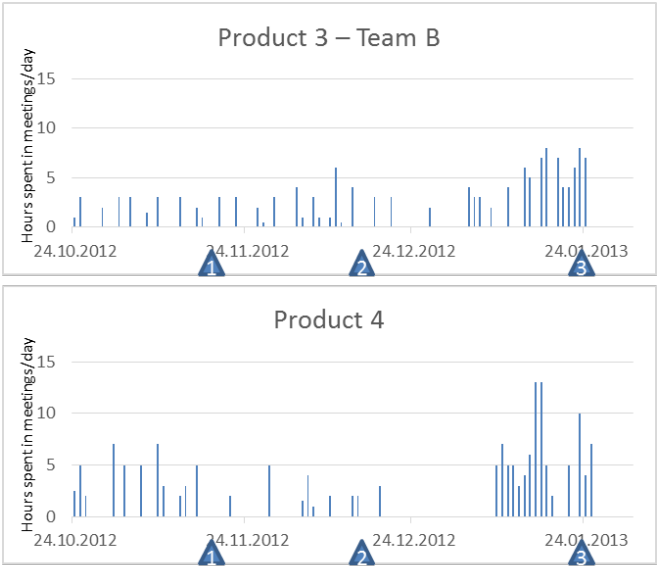

Figure 4: Massive meeting profiles

Although there may be exaggerations in student reporting, they actually spent a lot of time in the public meeting room. Many teams complained about working long hours, but not all of them spent so much time in meetings.

We investigated the communication goals of the *massive meeter* teams. Figure 5 shows the meeting goals for Product 3 – Team B. While in the first two phases (before Quality Gates 1 and 2) the main goals are to plan, exchange information and collaborate on creative tasks, these goals only play a minor role in the last phase. There, the teams mostly meet to sit together while everybody works on her or his own tasks or to solve a problem together.

Surprisingly, all teams had a rather moderate meeting profile during the first two thirds of the project. Only during implementation and test (after the holiday break), some would turn into massive meeters. Interestingly, none of the above-mentioned teams differed drastically in terms of customer evaluations or requirements compliance [8]. The other eight team profiles reside on different locations on the spectrum from moderate to massive meeters.

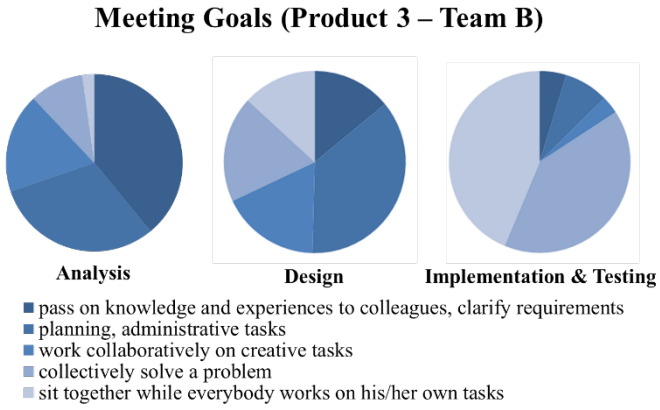

Figure 5: Distribution of Meeting Goals (Product 3 – Team B)

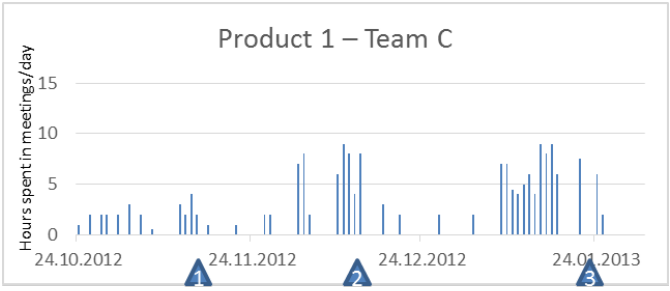

Figure 6: Many more meetings than in Figure 4 above

From the data presented above, more difficult projects could lead to the massive meeting profile, while easier projects favor the moderate meeting profile. Looking at Figure 6, this interpretation may be differentiated.

This shows a third team working on Product 1. However, there is a much higher intensity than with teams A and B of meetings in the final phase, more like in the massive meeting profile. Unlike that, however, this is the only project in which we observed a relatively high meeting density *during the earlier phases*. The unique aspect of this project was its staffing: since there were 68 students all together, one team had only three members: Product 1 – Team C. Although the same customer knew there was a smaller team, and asked fewer features, this team of three students obviously felt a higher pressure throughout the project. Again, their paradoxical reaction was to spend more of their limited time together, in meetings.

V. Interpretation and Discussion

A. Threats to Validity

The exploratory study should not be over-generalized. Obviously, there are several threats to validity. Internal validity may have suffered from self-reporting. However, we tried to minimize the motivation to cheat: There was no reward for high or low meeting intensity. Excessively long meetings may not be possible in many companies. The findings may not be generalized to highly different projects or environments without further evaluations. However, we are confident to use them on other cohorts of that same type of student projects (external validity). We defined the term meeting intensity to depend on frequency and duration of meetings only. This definition does not refer to the cognitive intensity of discussions (construct validity).

B. Theses for Discussion

Measuring meetings is easy and objective, and the information can be visualized easily. A meeting can be reported easily by listing it in a calendar. Many teams already have a shared calendar. Further, measuring meetings does not invade privacy, because this information is public within a company. From this information, it is easy to create a bar chart like presented in the figures above. The charts give a quick overview, allow to qualitatively assess the situation, recognize some simple patterns (like the massive meeting pattern) and compare different projects.

Teams seem to meet more under pressure. One would expect that under time pressure, a team will try to do as many tasks as possible in parallel, which means that everybody would work on their own. What we saw, however, was that teams started meeting a lot more often when they sensed more pressure. Looking into meeting goals (Figure 5), we found that the teams in the last phase meet to be able to coordinate quickly and help themselves more quickly when problems arise.

This observation is especially interesting, when thinking about distributed teams. They often need even more time for communication than co-located teams. At the same time, they cannot increase their meeting times as easily. They might be dependent on resources like video conference rooms, as well as have general limitations like different time zones.

The meeting rate in later phases cannot be predicted from early meetings. In the first two thirds of the project, most teams show the same meeting behavior. Hence, it is not possible to predict from these two phases if a team will get into the massive meeting mode at a later point in the project. What we saw with the team that consisted of three members, however, was that if the pressure was already high in the earlier phases, the meeting rate also was high in these earlier phases.

C. Benefits of Monitoring Meeting Intensity

Excessive stress situations in software projects may be overlooked by management – or only recognized when it is too late to mitigate the situation. Meeting intensity can provide an indicator to recognize stressful situations within a project. High or massive meeting intensity can be a sign for high pressure. Given this simple indicator, management has a chance to help – a matter of risk management.

From a company perspective, this instrument can help in avoiding unnecessary pressure by adjusting phase durations (quality gate dates). In extreme programming, the "sustainable pace" practice reminds developers to avoid excessive overwork [10]. Kanban emphasizes the importance of pull instead of push, which also helps to identify potentials for improvement at bottlenecks [11]. In our observation, there was a co-occurrence of high meeting intensity and project pressure. Meeting intensity is easy to measure. In our concrete case, a profile like in Figures 4 and 6 would indicate to shorten the early phases and to extend the last phase.

D. More Open Questions

The results so far are explorative. They open more questions, like: Why do teams meet more when they are under pressure? How do these meetings look like? Is it justified to call them meetings? What could distributed teams do? What would be the right communication infrastructure to accommodate a late need for such meetings? Would it make sense to intentionally arrange more meetings upfront in order to decrease pressure early on?

VI. Conclusions

Meetings are an important intersection of information for project members. It is necessary to understand meeting culture in order to provide ways to improve communication and collaboration within project teams.

We have shown that by simply documenting meetings and their durations, interesting profiles of a project team's meeting behavior can be depicted. We identified different patterns for these meetings. Especially the massive meeting pattern is particularly interesting, since most teams with this pattern reported a stressful situation and very high pressure afterwards.

We provide an interpretation of the observations and suggest interesting open questions. Measuring meetings could accordingly serve as an instrument to monitor pressure within projects and therefore could enhance risk management.

Meetings are central for communication, collaboration, and coordination. There are many more detailed aspects to research, which we also want to do in a longitudinal study in the future. Here, we have presented an interesting intermediate result – hoping to trigger discussions about our theses and open questions.

References

[1] M. Hoegl and P. Parboteeah, "Autonomy and teamwork in innovative projects," *Human Resource Management*, vol. 45, no. 1, pp. 67–79, 2006.

[2] C. Ebert, *Systematisches Requirements-Engineering und Management: Anforderungen ermitteln, spezifizieren, analysieren und verwalten*. dpunkt-Verlag, 2008.

[3] S. Kauffeld and N. Lehmann-Willenbrock, "Meetings Matter: Effects of Team Meetings on Team and Organizational Success," *Small Group Research*, vol. 43, no. 2, pp. 130–158, 2012.

[4] S. Kauffeld, *Kompetenzen messen, bewerten, entwickeln: Ein prozessanalytischer Ansatz für Gruppen*. Stuttgart: Schäffer-Poeschel, 2006.

[5] S. Kauffeld, S. Grote, and E. Frieling, *Handbuch Kompetenzentwicklung*. Stuttgart: Schäffer-Poeschel, 2009.

[6] P. N. Robillard and M. P. Robillard, "Types of collaborative work in software engineering," *Journal of Systems and Software*, vol. 53, no. 3, pp. 219–224, 2000.

[7] D. Lübke, T. Flohr, and K. Schneider, "Serious Insights through Fun Software-Projects," in *EuroSPI 2004: European Software Process Improvement Conference*, ser. Lecture Notes in Computer Science 3281, T. Dingsor, Ed. Trondheim, Norway: Springer, 2004.

[8] K. Schneider, O. Liskin, H. Paulsen, and S. Kauffeld, "Requirements Compliance as a Measure of Project Success," in *Proceedings of the 4th IEEE Global Engineering Education Conference (EDUCON 2013), in press*, 2013.

[9] K. Schneider, "LIDs: A Light-Weight Approach to Experience Elicitation and Reuse," in *Product Focused Software Process Improvement*, ser. Lecture Notes in Computer Science, F. Bomarius and M. Oivo, Eds., vol. 1840/2000. Springer Berlin / Heidelberg, 2000, pp. 407–424.

[10] K. Beck, *Extreme Programming Explained*. Addison-Wesley, 2000.

[11] D. J. Anderson, *Kanban - Successful Evolutionary Change for your Technology Business*. Blue Hole Press, 2010.