Team Familiarity, Goal Setting, and Process: A Case Study of a Corporate Hackathon

Ei Pa Pa Pe Than
Carnegie Mellon University
Pittsburgh, PA 15213, USA
eipapapt@cs.cmu.edu

Alexander Nolte
James D. Herbsleb
Carnegie Mellon University
Pittsburgh, PA 15213, USA
{aun, jdh}@cs.cmu.edu

Anna Filippova
GitHub Inc.
San Francisco, CA 94107, USA
annafil@gmail.com

Christian Bird
Microsoft Research
Redmond, WA 98052, USA
cbird@microsoft.com

Steve Scallen
Microsoft Garage
Redmond, WA 98052, USA
sscallen@microsoft.com

The 2nd Workshop on Hacking and Making at Time-bounded Events: Current Trends and Next Steps in Research and Event Design, CHI 2018, April 22, 2018, Montréal, Canada.

Abstract

Time-bounded events such as hackathons are increasingly popular, becoming a common feature of many large software companies including Google, Facebook, and Microsoft. With a widespread adoption of hackathons and a wide range of decisions about how to form teams and manage events, it is important to understand how characteristics of teams can impact desired outcomes. In particular, hackathons often include teams who are strangers before the event as well as teams who have regularly worked together. It is not clear how these different levels of familiarity impact choice of projects, coordination, and team dynamics. We collected interview data from members of five teams who participated at the 2017 Microsoft OneWeek Hackathon. We found that "pre-existing teams" (higher familiarity) used the hackathon space to get needed but non-routine work done, and chose projects that were riskier and long-term and set higher expectations on outcomes. In contrast, newly formed "flash teams" (lower familiarity) aligned their goals with official hackathon outcomes of lightly-engineered demos and videos, and had a substantial focus on personal goals. Flash teams experienced more conflict and misaligned expectations, yet were largely satisfied with the experience and intend to participate in the future.

Author Keywords

Hackathons; time-bounded events; collaboration; familiarity; goal setting; team process; team dynamics; mixed methods study.

Introduction

Time-bounded intensive events have become increasingly popular in recent years, variously called hackathons, data dives, codefests, hack days, sprints, edit-a-thons, map-a-thons, and so on. Their popularity is attested to by the fact that collegiate hackathons alone attracted more than 65,000 students from 16 different countries in 2017 [5]. Further, the scope of hackathons has broadened from the tech industry to other sectors and disciplines such as astronomy, arts and humanities, biology, social goods, and many more, taking on many different forms such as collaborative or competitive, and focused on innovation, community building, or learning [2, 7].

Regardless of design variations, all hackathons share a set of common features. People divide into small groups to innovate, improve, learn, and network within a specified timeframe, typically 2-5 days. These groups consist of people of often diverse backgrounds, experience, and expertise, and gather in one location. Due to their potential to leverage collective intelligence and foster innovation outside the usual constraints and processes of the workplace, hackathons have become a common feature of large software companies including Facebook, Google, and Microsoft. Yet little is known the ways different compositions of team harness, or fail to harness, the members’ creative capabilities.

Prior research on team familiarity (e.g., [1, 3]) suggests that teams with higher familiarity of members have higher performance. In particular, prior studies have found that when members of a team work together over time, they have increased familiarity with the task domain and with each other, clearer expectations and communication, a common knowledge base, and better coordination. Another factor important for team coordination process is goal setting (e.g., [4]). As goals or conscious ideas regulate people actions, the choice of goals may determine team coordination process.

Hackathons where teams vary widely in familiarity provide an interesting context in which to study familiarity and the strategies that non-familiar teams develop to accomplish work in extremely compressed time scales where members may be required to deliver a substantial result 2-5 days after they first meet. In addition, hackathon teams are generally free to work on anything they want with very few constraints other than time and potentially a desire to appeal to a judge. Goal setting becomes a critical process for enabling the team to work together effectively. It is therefore interesting to examine how both familiar and unfamiliar teams set goals and approach their attainment. Thus, we aim to address the following research questions to advance our understanding of team process in time-bounded settings:

RQ: In time-bounded settings, how do teams with higher familiarity differ from those with lower familiarity in 1) setting goals, and 2) coordinating their work?

Methods

We chose the 2017 Microsoft OneWeek Hackathon to study. This is a Microsoft’s annual global event held with more than 16,000 participating employees in

2016. The data was collected using a mixed methods approach which included interviews, observation, and survey. In particular, our research group interviewed event organizers and each team leader prior to the event, one member of the research group shadowed one team during hackathon days, interviewed members of that team within a week after the hackathon and again three months after the hackathon. We administered a survey at the science fair which was held on the last day of the event where participants showcased or demoed their projects. Of these five teams we shadowed, two were pre-existing teams with members who regularly worked together before (P1-7), and three were newly-formed "flash" teams whose members had (mostly) not worked together before (F1-16).

All interviews were transcribed and analyzed following empirical grounded theory procedures describe by Strauss and Corbin (1998) [6], using Deedose, a web-based qualitative data analysis software. First, three authors conducted open coding on the interview data, in which familiarity, goal setting, and coordination were used as sensitizing concepts. In the second phase, we shared and wrote descriptive memos. We then discussed in a highly collaborative manner and combined codes that had similar meanings to yield second-ordered codes or themes. The resulting coding scheme was used for the remaining text.

Preliminary Findings

In this paper, we report some of our preliminarily results based on the analysis of interview data collected within a week after the hackathon.

Goal setting: We found that pre-existing teams utilized the hackathon as a dedicated time and space to get the needed non-routine work done (P4, P5). Their chosen projects seemed to be riskier and have long-term potential compared to those of flash teams. In contrast, flash followed the goals that the hackathon set out for them, and aimed at producing lightly-engineered demos of their solutions (P4, P5, P6). In addition, majority of flash team members used the hackathon space to explore new skills and roles unrelated to their regular work, and network with and learn from people who are outside of their regular workgroup (F2, F4). In contrast, pre-existing team members' participation was closely related to their regular job (P4, P5). In that regard, flash teams were more inclined toward innovation, suggesting that there is a trade-off between familiarity and innovation in terms of goal setting.

Coordination: Pre-existing teams' members uniformly picked out tasks that they were familiar with or they could leverage their existing skills or knowledge (P1, E1, E3). In contrast, flash teams' members adopted divisions of labor loosely based on standard team roles at Microsoft (e.g., developer, marketing, UX designer, program manager) and performed activities expected to be performed by these roles (F5, F6, F16). When organizing their processes, pre-existing teams were found to fall back on their regular work practices whereas flash teams' members tended to coordinate based on their taken roles (P4, F16). Here, it is important for flash teams' members to modify their taken roles considering the constraints of hackathon.

Expectation mismatch: We found that flash teams were prone to a problem of expectation mismatch, especially when some members had no prior hackathon experience. These newcomers seemed to have set high or unrealistic expectations on outcomes of the hackathon, either by putting their personal goals first or holding to their professional engineering norms (F6, F15) rather than focusing just on a demo. Such expectation mismatches led to confusion or mild conflicts in flash teams.

Conflict avoidance: Both types of team seemed largely to avoid open conflict during hackathon. This may be due to the short duration of the hackathon which did not really allow time for conflict resolution, and that flash team members were unconcerned about allowing unspoken conflict to continue, since they were unlikely to continue to work as a team after the hackathon (P6, F11).

Conclusion

Our results suggest that goals and expectations set out by hackathon teams are contingent on how familiar their members are with each other. Depending on the types of goal they pursued, teams adopt different mechanisms to organize themselves in such a way that would maximize the goal attainment. Having prior hackathon experience enabled them to realize the differences between hackathon and regular work and modify their hackathon roles accordingly.

Author Background

Ei Pa Pa Pe Than is a postdoctoral researcher in the Carnegie Mellon University's Institute for Software Research. Her research focuses on understanding how technologies can be leveraged to create new forms of collaboration that improve engagement, productivity, and outcome quality.

Alexander Nolte is a postdoctoral researcher at the School of Information Sciences at the University of Pittsburgh and a research associate at the Institute for Software Research at Carnegie Mellon University.

James Herbsleb is a Professor of Computer Science at Carnegie Mellon University, where he serves as Director of the PhD program in Societal Computing. His research interests focus on global software development, open source, and more generally on collaboration and coordination in technical domains.

Anna Filippova is a data scientist at GitHub Inc.

Christian Bird is a researcher at Microsoft Research working on Empirical Software Engineering.

Steve Scallen is a principal design researcher at Microsoft Garage.

Acknowledgements

We wish to thank Microsoft Garage team and Microsoft Research for collaborating with us on this study. We also thank members' of hackahton teams we shadowed for sharng their expreinces.

References

1. J Alberto Espinosa, Sandra A. Slaughter, Robert E Kraut, and James D. Herbsleb. 2007. Familiarity, complexity, and team performance in geographically distributed software development. *Organization science, 18*(4), 613-630.
2. Anna Filippova, Erik Trainer, and James D. Herbsleb. 2017. From diversity by numbers to diversity as process: supporting inclusiveness in software development teams with brainstorming. In *Proceedings of the 39th International*

Conference on Software Engineering (ICSE '17), 152-163. DOI: https://doi.org/10.1109/ICSE.2017.22

3. Pamela J. Hinds and Catherine Durnell Cramton. (2013). Situated coworker familiarity: How site visits transform relationships among distributed workers. *Organization Science, 25*(3), 794-814.
4. Edwin A. Locke. 1968. Toward a theory of task motivation and incentives. *Organizational Behavior and Human Performance, 3*, 157-189.
5. Major League Hacking (MLH). Retrieved January 31, 2018 from https://mlh.io
6. Anselm Strauss, and Juliet Corbin. 1994. Grounded theory methodology. *Handbook of qualitative research, 17*, 273-285.
7. Erik H. Trainer, Arun Kalyanasundaram, Chalalai Chaihirunkarn, and James D. Herbsleb. 2016. How to Hackathon: Socio-technical Tradeoffs in Brief, Intensive Collocation. In *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing (CSCW '16)*, 1118-1130. DOI: https://doi.org/10.1145/2818048.2819946