

The Social Developer: The Future of Software Development

Designing Corporate Hackathons With a Purpose

Ei Pa Pa Pe-Than, Institute for Software Research, Carnegie Mellon University

Alexander Nolte, Institute of Computer Science, University of Tartu

Anna Filippova, GitHub Inc.

Christian Bird, Microsoft Research

Steve Scallen, Microsoft Garage

James D. Herbsleb, Institute for Software Research, Carnegie Mellon University

In hackathons, small teams work together over a specified period of time to complete a project of interest. Hackathons have become increasingly popular as a means to surface and prototype innovative and creative ideas for products, but their impact often goes beyond product innovation. Based on our empirical studies of 10 hackathons held by scientific communities, a corporation, and universities as well as the review of published literature, we discuss that hackathons can be organized around goals such as enriching social networks, facilitating collaborative learning, and workforce development. We also discuss design choices that can scaffold the organization of hackathons and their trade-offs. Design choices include identifying a suitable mixture of attendee skills, the selection process for projects and teams, and whether to hold a competitive or collaborative event. Hackathons can achieve multiple goals if designed carefully.

Keywords: Computer-supported collaborative work, Human Factors in Software Design, Collaborative Computing, Software Development, Innovation and Technology.

To remain competitive in the global market, tech companies are required constantly to deliver new products and services that offer value to their customers. Many of these companies have adopted various development strategies not only to shorten the product development cycle but also to optimize the capabilities of developers to create innovative products and features. As a consequence, the way that developers organize their efforts is constantly evolving, in response to new business needs and technical changes. Examples of such changes are global or distributed development, agile teams, and DevOps deployment practices.

Hackathons are a relatively new form of organizing for product innovation that is taking on increased importance, and in fact seems to have become a part of the development work of nearly every major tech company and university computer science environment. Hackathons started as competitive events for young developers in the mid to late 2000s but were quickly adopted in different domains such as education,⁹ civic engagement¹⁰ and

in corporations of all sizes. In general, hackathons are time-bounded events, typically of 2-5 days, during which people gather together and form teams, each of which attempts to complete a project of interest to them.^{4,14} The teams are usually collocated, and often composed of people with diverse backgrounds, experience, and expertise. In a corporate hackathon, employees generally form teams of three to five people and work intensively, primarily to produce working prototypes of ideas that could be integrated into existing products or serve as a basis for new products or services.¹ One such example is Microsoft's annual OneWeek global hackathon. Every year in summer, Microsoft employees are given the opportunity to leave their day-to-day jobs for a week (about 18,000 did so in 2017), team up with others, and hack on a project of the team's choosing. Other tech companies such as Facebook and Google also run similar company-wide hackathons each year as well as multiple smaller internal and external hackathons.¹ Hackathons generally combine several features that foster innovation. For example:

- They often bring together people with diverse expertise and experience.
- The work hours are relatively focused and interruption-free.
- They occur outside the usual constraints of processes, goals, and management.
- They provide the opportunity to run a project, assess its feasibility and uncover potential pitfalls with minimal risk to daily operations.
- Participants work on something they are passionate about.

In addition to hackathons' potential to foster innovation,¹ they may also be used to reduce stovepiping by creating new social connections,⁷ to provide learning opportunities,⁵ and to develop and exercise new technical and leadership skills in a low-risk environment. Hackathons can serve many different goals, and the relative importance of these goals can vary dramatically from one company to the next, and from one hackathon to the next.

Hackathons can also be designed in many different ways. They may, for example, provide different kinds of incentives, and have widely varying processes for selecting projects and teams. For those contemplating hackathons, one important question to ask is: *How can hackathons be designed to achieve specific goals?* Although one can easily find information online about how to organize a hackathon,^{8,12} most of the information is based solely on the organization of one specific style of event and does not consider the continuation of hackathon projects afterwards.² Most importantly, none of the research to date compares hackathons across different design elements to evaluate their effectiveness with respect to the intended goals of the events.

In this paper, we first discuss a number of goals around which a hackathon can be organized, and then some of the design choices that can foster achieving such goals and describe some key design trade-offs. As with all events where people gather, there are the usual needs for space, food, facilities, promotional material, etc. Since these are not specific to hackathons – except perhaps for greater demand for electricity and bandwidth – we will focus only on hackathon-specific choices. Our discussion is based on our empirical studies of 10 hackathons, including hackathons by scientific communities, a very large scale internal corporate hackathon, and university hackathons. As part of our research, we have attended and ethnographically observed hackathons, completed 103 interviews with organizers and participants, and administered 4 post-hackathon surveys. We have also reviewed the published literature directed to both researchers and practitioners and have integrated this information into our results. Although we draw on experience and literature representing a variety of hackathon types, here we narrow our focus to corporate hackathons since they are the most expensive, hold the greatest promise for commercial advantage, and present unique problems given their embeddedness in a corporate context.

Goals: Organizational and Personal

Designing an effective hackathon involves a careful consideration of goals set out by both organizers and participants. The organizers need to be aware that their goals for hackathons may, and often will, be different from those of participants. Failure to consider a possible divergence in goals may result in not being able to recruit or to leverage the fullest potential of participants and may detract from participant satisfaction and outcome quality.

Some of the common goals for corporate hackathon *organizers* include:

- ***Enrich intra-company networks and reduce stovepiping:*** Motivate and provide an opportunity for people from different parts of the company and on different levels of seniority, who are unlikely to have opportunities to communicate and work together, to form teams and collaborate.
- ***Change the culture within the company:*** Encourage people to contribute to initiatives that are outside of the scope of their regular work and encourage creativity and outside-of-the-box thinking.
- ***Workforce development:*** Encourage participants to explore new roles like product or project managers, and to expand their technical skill set by facilitating a self-driven and collaborative learning environment.
- ***External image:*** Show potential future employees the company is innovative and open to change.

In comparison to organizers' goals, participants might have similar as well as different goals in mind such as:

- ***Having fun:*** Escape the constraints of company product plans and preset development processes and allow participants to work at their own pace on things they care about.
- ***Learning:*** Learn new technologies and tools, and more about their current projects, and other skills such as collaboration, leadership and project management.
- ***Win prizes:*** Achieve monetary or other prizes such as recognition by leadership and/or their peers.
- ***Expand personal networks:*** Grow individual professional networks within the company beyond the boundaries of their everyday work.
- ***Foster career:*** Impress current managers through taking on new roles during the hackathon or draw other departments' attention to a participant's skills.
- ***Get the needed work done:*** Take advantage of this opportunity to pursue a project that is a high priority to an individual or team, but low priority to managers allocating resources.

Design Choices

In this section, we elaborate on a set of core design choices that can be used to shape the design of hackathons for particular purposes and describe key design trade-offs. Table 1 summarizes the design choices and supportive goals in the context of corporate hackathon.

Table 1. A Summary of Hackathon Design Choices and Related Goals

<i>Design Choices</i>	<i>Strategies</i>	<i>Organizational Goals</i>	<i>Personal Goals</i>
Collaboration vs Competition	Collaboration <ul style="list-style-type: none"> Facilitate shared or interdependent goals Integrate social elements and unconference sessions Enable team switching at predefined intervals 	<ul style="list-style-type: none"> Enrich intra-company networks and reduce stovepiping Collaborative learning Change the culture within the company Improve external image 	<ul style="list-style-type: none"> Expand one's personal network Learning Foster career Having fun
	Competition <ul style="list-style-type: none"> Compete for prizes Prizes range from cash to opportunities for continued development of winning ideas Subject matter experts are usually invited as judges Awards range from winners of challenges to popularity awards 	<ul style="list-style-type: none"> Product innovation Change the culture within the company Improve external image 	<ul style="list-style-type: none"> Win prizes Get the needed work done Foster career Having fun
Attracting attendees with relevant skillsets	<ul style="list-style-type: none"> Identify individuals who are enthusiastic about hackathons Distribute promotional materials timely and effectively Use various invitation approaches: incentives, targeted invitation, and participant selection using software tools 	<ul style="list-style-type: none"> Product innovation Collaborative learning Workforce development Advance technical work 	<ul style="list-style-type: none"> Expand one's personal network Learning Having fun Get the needed work done
	<ul style="list-style-type: none"> Mentoring, tutorial sessions, and brainstorming A manageable ratio of (at least 2:1) novices to advanced team members 	<ul style="list-style-type: none"> Encourage diversity and inclusion Lower the barrier to participation of novices Collaborative and spontaneous learning Workforce development 	<ul style="list-style-type: none"> Learning Foster career
Selection of projects	<ul style="list-style-type: none"> Participants propose own project ideas 	<ul style="list-style-type: none"> Product innovation Encourage a sense of autonomy 	<ul style="list-style-type: none"> Creativity and innovation Get the needed work done Learning Foster career Having fun
	<ul style="list-style-type: none"> Participants pick organizers' proposed project ideas 	<ul style="list-style-type: none"> Collaborative learning Workforce development Progress in targeted areas 	<ul style="list-style-type: none"> Foster career Learning
Team formation	<ul style="list-style-type: none"> Self-organization of teams by recruiting members with required skills Tools such as Hackbox are 	<ul style="list-style-type: none"> Enrich intra-company networks and reduce stovepiping 	<ul style="list-style-type: none"> Expand one's personal network Foster career Having fun

	essential to have a better mix of skills		
	<ul style="list-style-type: none"> • Team assignment by organizers 	<ul style="list-style-type: none"> • Advance technical work • Collaborative learning 	<ul style="list-style-type: none"> • Foster career • Learning
Pre-work before or at the event	<ul style="list-style-type: none"> • Pre-meetings, where teams can divide the work into manageable tasks, assign roles, and pretest technologies 	<ul style="list-style-type: none"> • Improve team efficiency by alleviating slow-start problem • 	<ul style="list-style-type: none"> • Win prizes • Expand one's personal network • Foster career
	<ul style="list-style-type: none"> • No pre-meetings, facilitate ideation and brainstorming in teams at the event 	<ul style="list-style-type: none"> • Encourage diversity and inclusion • Foster the integration of diverse ideas and perspectives into final product 	<ul style="list-style-type: none"> • Expand one's personal network • Learning
Post-work	<ul style="list-style-type: none"> • Pitching output to seek home for project to continue 	<ul style="list-style-type: none"> • Increase the visibility of projects to a larger audience • Increase the chance of project continuation 	<ul style="list-style-type: none"> • Foster career • Gain recognition • Seek an opportunity for project continuation

Fostering Competition

One key choice has to do with incentives structured either to favor competing or cooperating. People generally take part voluntarily in hackathons, but various design features and incentives can shape their participation in either a competitive or cooperative direction. Collaborative hackathons are typically designed to enhance interaction among participants, thereby establishing or deepening connections that can foster longer-term collaboration post-hackathon.⁷ This can be achieved by facilitating inter-team interactions with shared or interdependent goals and/or injecting social elements into the hackathon agenda.³ One common approach used in collaborative-style hackathons is having “unconference” sessions¹³ during which participants give short technical briefings or pitch project ideas. Afterwards participants can be encouraged to wander around the room, discuss with the owner of an idea that they are interested in and offer suggestions. This situation increases the chance of participants to meet new people and generate cross-pollination among ideas. Another approach is team fluidity where participants switch between teams at specified intervals which allows members to meet others and exchange information about their projects.¹⁰ However, our interview data suggests that this approach might lead to frustration among participants and reduced commitment if they feel they are forced to switch before their work is completed. Hence, this must be done carefully, with attention to participant goals, but can be effective if participants also want to focus on building their personal networks. If they are more focused on exploring their own ideas and polishing a prototype, efforts to impose fluidity among projects may prove quite unproductive. A collaborative-style hackathon would be suitable to reduce stovepiping, facilitate collaborative learning, enhance personal networks, and advance a cause shared among participants.

In competitive-style hackathons, teams generally compete for prizes.^{6,14} Prizes can vary greatly, with cash prizes and opportunities for continued development of winning ideas as perhaps the most common. The opportunities for further development can take the form of providing additional resources, freeing up participants time to work on the project post-hackathon, or simply the opportunity to pitch the idea to a top executive.⁶ Experts are invited as judges, and winners are typically decided based on predefined criteria such as

appeal to market, creativity, originality, and completeness. Some hackathons also award projects that receive the highest number of votes from attendees, or that meet specific challenges posed by executives. The competitive pressure is likely to incentivize teams to put more effort in their projects, with an aim to generate more unique solutions and differentiate themselves from other competing teams. Hence, competitive elements could be used to facilitate product innovation. However, as competition tends to hinder communication between teams, competitive hackathons might not be appropriate to enrich networking among participants beyond participating teams. In large hackathons, with relatively few teams able to win prizes, many or most teams may consider themselves unlikely to win and may either feel demotivated or participate for other reasons and essentially ignore the competition, which was reflected in our interviews with many participants of competitive hackathons. In order to help the teams less interested in and motivated by competition to participate and benefit in other ways from the hackathon, it may be best to de-emphasize the prizes, and not focus on them as the sole or even primary reason to participate.

Attracting a mixture of attendee skills

It is crucial to garner interest by potential participants in order for the hackathon to be successful. This requires promotional material and the identification of individuals within the company who are enthusiastic about participating in and are willing to spread the news about the hackathon.¹³ The promotional material should be distributed through suitable channels depending on the company culture. Example channels include poster, email, and enterprise social networks or Slack. This material should not only make clear that the hackathon has management support but also encourage potential participants to create ideas, form teams and prepare individually or as a group prior to the event.

To attract attendees with relevant skill sets, hackathons employ various approaches including participation incentives, targeted invitation, and actual participant selection by organizers. Some hackathons with targeted invitation recruit attendees from distinct communities and invite individuals that they want to be in the hackathon personally while others encourage newcomers and minorities like women software developers with offers of additional training.¹¹ We have observed hackathons where organizers select participants using software tools such as Entrofy (<https://github.com/dhuppenkothen/entrofy>) to diversify participants over a range of criteria.

A hackathon consisting of attendees with diverse skills can facilitate innovation and learning due to attendees being able to generate and assess ideas from various perspectives. During our observations of multiple hackathons, we found that participants frequently got involved into conversations happened among other team members. These situations led to providing useful suggestions which were based on participants' expertise and experience. In this regard, having diverse participants can facilitate spontaneous learning and creativity among participants. Skill diversity, however, can reduce productivity and technical output since it may take more time for attendees to be on the same page during the discussion and execution of a hackathon project as evident in our observation of Microsoft's OneWeek Hackathon. Skill diversity presents a potential trade-off between the generation of innovative ideas (high diversity) and technical progress (low diversity).

Selection of projects and team formation

Hackathons can allow participants to (1) come up with their own ideas or (2) pick from a set of ideas provided by the organizers. In the first approach, participants propose project ideas at the beginning of or prior to the event,⁶ and recruit team members. This can be supported through a web-based platform where people post project ideas and advertise roles required for their projects. This first approach is likely to encourage new and innovative products since the participants are free to inject, discuss, and combine their own ideas. Encouraging participants to propose ideas before the event has the advantage

that teams can be more prepared, hence more efficient at the hackathon. In fact, in the competitive events we have observed, the winning teams always have fairly extensive preparation. However, organizers always have to anticipate that some participants will come to the hackathon without a team and will need to pitch ideas and form teams at the event. Discouraging preparation, on the other hand, provides more opportunity to discuss and refine ideas with a larger set of people,⁴ and it encourages participants to meet people who they might not have a chance to meet otherwise, and hence enrich one's personal network. The second approach can be effective when a primary goal is learning,⁵ and the organizers can devise projects that address specific learning goals.⁹ It can also be very useful if the organizers' primary goal is to accomplish specific, high priority work. A key trade-off here is the creativity and fun that developing and working on one's own idea provides versus the difficulty that novices, in particular, may have in formulating a feasible project, and the possibility that high priority work will be neglected.

Another important aspect is how to organize teams to have a desired mix of skills for each project. One possibility is to have a moderator assign participants to teams once their skills have been identified. Another, and more common, approach, is to allow teams to self-organize, running the risk that teams may end up with members with very similar backgrounds. In fact, homophily –the well-established tendency of people with similar traits to hang out together – tends to happen when there is no other basis for organizing teams, as we have observed in hackathons attended by distinct professional groups. Software tools that allow participants to pitch project ideas and a mix of skills required for the proposed project can foster skill diversity in self-organized teams. This can also allow someone to identify opportunities to sign on for roles in teams that will allow them to develop new skills.

When teams are organized to have a mix of skills, they will most likely consist of both novices and experts. Here, mentoring and tutorial sessions will be helpful for novices. We found brainstorming to be a useful technique that allows everyone to feel their ideas are heard and seems particularly effective in helping those who identify as minorities to feel satisfied with the process.^{4,11} A hackathon we have studied holds a separate event only with minorities before they take part in a larger event with more diverse participants. We have also observed hackathons where novices are encouraged to spread themselves out among teams with more experienced members, who are encouraged to help bring the novices up to speed. Keeping a manageable ratio of (at least 3:1) novices to experienced team members facilitates learning without too great a sacrifice of technical progress.

Pre-work and Post-work

It is advisable for teams that aim to develop a fully functioning prototype during the hackathon to meet before the event and divide their project into manageable work packages, assign responsibilities and pretest technologies that they are going to use. This allows them to be as efficient as possible during the hackathon. For teams that are not prepared to start with idea at the event, it is worthwhile to consider the best ways to facilitate ideation and brainstorming in teams.⁴

In order for hackathon prototypes to have impact, follow-up activities have to be prepared by both organizers and participants. Organizers should provide opportunities for teams to promote their prototypes to a larger audience.⁶ At Microsoft's OneWeek Hackathon, the organizers provided support for creating a video demonstrating each project and allowed participants to publicize their project and video through a web-based platform and a "science fair" at the end of the hackathon.

We found that the continuation of a project beyond the hackathon mainly depends on a market need and a project's fit to the existing products. Finding a suitable home where a prototype can mature can be difficult, and generally requires individual networking and determination on the part of the participants. If a hackathon is aimed to become a recurring event, organizers might also want to evaluate them to improve future hackathons. This could be done by post-event surveys, ethnographic observations, and interviews.

Conclusion

Hackathons are successfully used as a new form of organizing product innovation in response to new business needs and technical changes due to its ability to create prototypes and assess their feasibility within a relatively short period of time. However, designing a hackathon involves a careful upfront planning and consideration of goals that both organizers and participants have set for the event and for themselves. Based on our studies of hackathons held by different communities as well as our review of extant literature, we have identified various goals that organizers and participants may aim to achieve from hackathons, showed how such events can be designed to achieve specific goals, and identified potential design trade-offs. Our results suggest that in addition to product innovation, hackathons can be used with great success as a tool for achieving a variety of goals such as enriched intra-company network and preparing employees for future changes and positions.

References

1. G. Briscoe, "Digital innovation: The hackathon phenomenon," Working papers of the Sustainable Society Network, 2014.
2. A. Carruthers, "Open Data Day Hackathon 2014 at Edmonton Public Library," Partnership: The Canadian Journal of Library and Information Practice and Research, vol. 9, no. 2, 2014, p.1.
3. A. Decker, K. Eiselt, and K. Voll, "Understanding and improving the culture of hackathons: Think global hack local," Proc. Frontiers in Education Conference (FIE), 2015, pp. 1-8.
4. A. Filippova, E. Trainer, and J. D. Herbsleb, "From diversity by numbers to diversity as process: supporting inclusiveness in software development teams with brainstorming," Proc. 39th International Conference on Software Engineering, 2017, pp. 152–163.
5. A. Fowler, "Informal stem learning in game jams, hackathons and game creation events," Proc. Int'l Conf. on Game Jams, Hackathons, and Game Creation Events, 2016, pp. 38-41.
6. M. Komssi, D. Pichlis, M. Raatikainen, K. Kindström, and J. Järvinen, "What are hackathons for?," IEEE Software, vol. 32, no. 5, 2015, pp. 60–67.
7. S. Möller, E. Afgan, M. Banck, R. J. Bonnal, T. Booth, J. Chilton, ... and M. Kalaš, "Community-driven development for computational biology at Sprints, Hackathons and Codefests," BMC bioinformatics, vol. 15, no. 14, p. S7, 2014.
8. Major League Hacking (MLH), "The MLH Hackathon Organizer Guide,"; <https://guide.mlh.io>.
9. A. Nandi, and M. Mandernach, "Hackathons as an informal learning platform," Proc. 47th ACM Technical Symposium on Computing Science Education, 2016, pp. 346-351.
10. E. Porter, C. Bopp, E. Gerber, and A. Volda, "Reappropriating Hackathons: The Production Work of the CHI4Good Day of Service," Proc. 2017 CHI Conference on Human Factors in Computing Systems, 2017, pp. 810-814.
11. G. T. Richard, Y. B. Kafai, B. Adleberg, and O. Telhan, "StitchFest: Diversifying a College Hackathon to broaden participation and perceptions in computing," Proc. 46th ACM Technical Symposium on Computer Science Education, 2015, pp. 114-119.
12. Smart Chicago, "Civic Innovation Toolkit: How to run a civic hackathon,"; <http://www.smartchicagocollaborative.org/civic-innovation-toolkit-how-to-run-a-civic-hackathon/>
13. A. Stoltzfus et al., "Community and Code: Nine Lessons from Nine NESCent Hackathons," F1000Research, vol. 6, 2017.
14. E. H. Trainer, A. Kalyanasundaram, C. Chaihirunkarn, and J. D. Herbsleb, "How to

hackathon: Socio-technical tradeoffs in brief, intensive collocation,” Proc. 19th ACM conference on computer-supported cooperative work & social computing (CSCW 2016), 2016, pp. 1118–1130.

Ei Pa Pa Pe Than is a Postdoctoral Researcher in the Institute for Software Research in the School of Computer Science at Carnegie Mellon University. Her research interest involves understanding how technology can shape new forms of collaboration that foster engagement, creativity and innovation, productivity, and outcome quality, through empirical research methods. She has a masters degree in Information Systems and a PhD in Information Science from the Nanyang Technological University in Singapore. Contact her at eipapapt@cs.cmu.edu.

Alexander Nolte is a Lecturer at the Institute of Computer Science at the University of Tartu and an Adjunct Assistant Professor at the Institute for Software Research at Carnegie Mellon University. His main research interest is on understanding how individuals collaborate and support them by designing and evaluating socio-technical approaches that sparkle creativity, foster innovation and improve collaboration. Alexander has a PhD in Information Systems from the University of Duisburg-Essen in Germany. Contact him at alexander.nolte@ut.ee.

Anna Filippova is a Data scientist at GitHub, Inc. Her work spans across fields of Communications, Organizational Behavior, Social Psychology and Information Systems, using both qualitative and quantitative methods. She is interested in distributed development and open source and free software, and generally in helping people connect and communicate. She holds a PhD in Communication and New Media from the National University of Singapore. Contact her at annafil@gmail.com.

Christian Bird is a Researcher at Microsoft Research working on Empirical Software Engineering, focusing on ways to use data to guide decisions of stakeholders in large software projects. Chris received his bachelor's degree in computer science from Brigham Young University and his Ph.D. in Computer Science from University of California Davis. Contact him at Christian.Bird@microsoft.com.

Steve Scallen is a Principal Design Researcher at Microsoft Garage. He leads development of tools and resources for Hackathons, Hackers, Hack Teams, and Hack Projects. He holds a PhD from the University of Minnesota. Contact him at sscallen@microsoft.com.

James D. Herbsleb is Professor in the Institute for Software Research in the School of Computer Science at Carnegie Mellon University. His interests include collaboration and coordination in software development, taking a socio-technical approach to novel forms of organizing such as globally-distributed development, open source communities and ecosystems, and hackathons. Before joining the faculty at CMU, he led the Bell Labs Collaboratory Project at Bell Laboratories, Lucent Technologies. He has a masters degree in computer science from the University of Michigan, a PhD in psychology and a J.D. in law from the University of Nebraska. Contact him at jim.herbsleb@gmail.com.

Authors' Photos



Ei Pa Pa Pe Than



Alexander Nolte



Anna Filippova



Christian Bird



Steve Scallen



James D. Herbsleb

Tweets

- Hackathons can be used as a means to surface and prototype innovation and creative ideas for products
- If designed with a purpose, hackathons can also enrich inner-company network, reduce stovepiping, and even change company culture
- One key design choice is collaborative vs. competitive, each of which serves different goals and outcomes
- Other important design choices include a suitable mix of attendee skills, and project and team selection process