Human Factors in Software and Systems Engineering, Vol. 94, 2023, 64–74
https://doi.org/10.54941/ahfe1003771

Hackathon-Based Software Development: Lessons Learned From an Internal Corporate Hackathon

Georgios Christou

Core Business Technologies, East Providence, RI 02914, USA

ABSTRACT

This article discusses the qualitative evaluation of the results of a corporate internal hackathon, detailing its design, execution, and results. The article begins by noting the factors motivating the decision to perform an internal hackathon. Then the article describes the way the hackathon was structured to fit the corporate environment, the method followed to attack and solve the problem, as well as the outcome of the project undertaken and the effects on the team that participated in it. The article also examines the reasons behind the team's attendance to the hackathon, and the intangible rewards that the team members reported. The results of the evaluation are that there is value in using the hackathon method for the development of new solutions, as well as for the integration of those solutions into the corporation's existing software offerings. Another result of note is that several intangible rewards were expressed by the software developers. Examples of these intangible benefits included personal growth in the context of software development and strengthening of the team bond, thereby helping the team work more efficiently, with better communication amongst its members. Finally, the article proposes a software design and implementation methodology which suits the development done during a hackathon.

Keywords: Software engineering, Corporate hackathon, Systems modelling, Software development, Methodology, Framework

INTRODUCTION

Software development companies look for ways to create innovative products while keeping productivity levels high among their developers (Blackburn et al., 1996). One of these ways that has been prevalent in the past few years is the Open Innovation Model (Chesbrough, 2003), which suggests that "a company commercializes both its own ideas as well as innovations from other firms and seeks ways to bring its in-house ideas to market by deploying pathways outside its current businesses. Note that the boundary between the company and its surrounding environment is porous, enabling innovations to move more easily between the two." (Chesbrough, 2003, p. 37).

However, sometimes these companies do not need new products, but rather innovative solutions to problems of interoperability (Peter, 1996) across their software offerings. This type of innovation cannot be implemented through the model of Open Innovation (Bigliardi et al., 2021) because of

non-disclosure issues, time-to-market, and other confidentiality problems that have to do with disclosure of such products before their introduction to the market (Trott & Hartmann, 2009).

There is, however, an interesting approach that the Open Innovation Model uses to create working products that at least proves that a concept or idea is implementable: the hackathon. A hackathon is in essence a crowdsourcing event; a competition where the contestants are "small teams that work over a specified period to complete a project of interest" (Pe-Than et al., 2019, 15). There are several types of hackathons with different types of goals for the competing teams. There are hackathons that cater towards students, young professionals, those that provide help in the creation of startups, etc. All of them though, abide by a few general rules: The hackathon is a limited-time event, and the problem, however ill-defined, is usually given to the participants by the hackathon-organizing authority.

There are several advantages to leveraging a hackathon for innovation. The process requires a very rapid development cycle, during which a proof-of-concept of the idea is created. This idea is usually innovative, and that is why hackathons are attractive to companies that can use the Open Innovation method. The very rapid development cycle component of a hackathon, which is perhaps downplayed when discussing innovation, becomes extremely important for companies that need to show that a product can actually be created, showcase its capabilities, and provide tangible proof to the management team that the product can be delivered in the way that it was envisioned and designed.

The question that we ask in this article is whether the hackathon model can be used for actual development inside the corporate environment and, if yes, how this can be applied to result in a usable solution by the corporation. Our answer comes from the qualitative evaluation of a hackathon-based event that was held at Core Business Technologies, a fintech company based in Rhode Island, USA. The participants of the event were a designated product team of five engineers, a software architect, the Senior Vice President (SVP) of Product Management and the SVP of Engineering. It must be kept in mind that both SVPs not only managed the team, but were also considered part of the development team, writing significant parts of the code during the event.

The team was broken down into two sub-teams: a back-end development team that used the Go Programming Language and a Systems team that used the Perl programming language. The back-end team was responsible for creating a REST API that would allow any software developed by the company to connect and send/receive messages for payments through SMS, and the Systems team was responsible for integrating one of the company's softwware assets to the back-end along with supporting screens and reports for the payment through SMS functionality.

MOTIVATION TOWARDS USING THE HACKATHON PROCESS

Hackathon events use the same formula for their operation. The event will draw in participants that are interested in the hackathon's specific topic, usually from the world of computer technology. The participants form teams, either pre-determined or self-formed during the event's beginning. These

teams will then work atypically long hours to produce a digital product that provides a solution to either a specific problem defined by the hackathon's organizers, or to a problem that each team chooses to solve, backed by a reason of significance. At the end of the hackathon, each team presents their solution to a panel of judges, either to win a prize, or with the hope that a corporate sponsor may pick up their solution (Zukin & Papadantonakis, 2017). As mentioned earlier, there are several problems with this approach when applied to a corporate setting, particularly when working with technologies and solutions that have been developed inside the corporation, which cannot be shared to the public (Trott & Hartmann, 2009). Despite the drawbacks of an open-to-the-community hackathon, there are several good reasons that closed hackathons can be leveraged for innovation at a corporation. For example, we posit that the immersion of the hackathon participants into a social structure where everyone is working toward solving a common problem or by using similar technologies toward the solution of different problems creates an environment conducive to collaborative and innovative thinking (Chai & Freeman, 2019). Also, the hackathon environment ensures that the participants are not distracted by external intrusions, such as having to share their time between several tasks which is something inherent in a contemporary work environment. As such, a hackathon environment lends itself towards rapid project development, particularly when the required goal is sufficiently delineated to the participants. In other words, when a project has a deadline looming, insulating the team that is working towards the resolution of the project in a hackathon environment is perhaps more beneficial than allowing the team to proceed with business-as-usual in the corporate environment.

With the aforementioned project in the corporation, it was decided that the hackathon approach would be tried for two reasons. First, to test whether it was a viable approach to the project's development and resolution. Second, to examine the implications of such an intense development effort on the participating team and evaluate the intangible effects (both good and bad) on the team.

The hackathon solution was proposed by the SVP of Engineering to the team, with the caveat that only if the whole team agreed to this approach the hackathon would be performed. The response of the team was an overwhelming yes. In fact, the suggestion created interest and excitement to the participants.

It was decided that the hackathon would take place in four days, from Monday to Thursday, during which days the team would all be collocated at the corporation's offices, in seclusion from any other department. It was also decided that a strategy session would take place the Sunday before, so that the whole team would know what the problem would be, decide on the solution methodology and technologies, and create a preliminary architecture for the solution.

INTERVIEW RESULTS AND DISCUSSION

Here we discuss the process that was taken to host the hackathon from its inception to the resolution of the project. The results discussed were

Table 1. The interview questions.

No	Interview Questions
1	What were the reasons for you wanting to take part in the hackathon?
2	Why did you feel that the hackathon was a good practice to resolve the project?
3	What were the challenges that you faced during the hackathon?
4	What, if any, rewards did you take from your participation in the hackathon?
5	Do you believe that the project was concluded successfully at the end of the hackathon?
6	Can you describe the method followed for the initialization, development, and resolution of the project during the hackathon, in your own words?
7	Would you participate in another hackathon? If yes or no, why? Does your answer come with any concerns?
8	What is the greatest lesson that you have taken away from this experience?

taken from direct observations during the hackathon, as well as from semi-structured interviews with the participants. The initial interview questions are shown in Table 1.

Interview Results

The interviews were performed through video conference software. There were seven (7) interviewees in total. The interviewees were all the members of the team that took part in the hackathon except the researcher. Interviewees were all informed that everything they said during the interview would be confidential. They were also informed that their interview answers would not be used verbatim, but rather the summation of the interview answers for each question would be used in any resultant research produced.

Question 1: What were the reasons for you wanting to take part in the hackathon?

The team members mentioned that one of the main drivers was the realization that management vis a vis the sales team was impatient to have a solution delivered to satisfy a market need. They also agreed that this process would provide a better environment where each team member would get to know the others better, especially in the context of a distributed work force. Members hoped to develop better communication between them and with their managers, and they were also hoping to build closer relationships between themselves to augment virtual standups and other Scrum ceremonies.

Question 2: Why did you feel that the hackathon was a good practice to resolve the project?

Before the hackathon, several of the team members admitted that they were not entirely convinced that the hackathon process would be ideal to resolve the project. However, after the hackathon, their answers reached a consensus that the hackathon process had indeed worked. The team members reported

that they felt that they were more focused on the one project, because the hackathon environment ensured that they would not be distracted by any other task or requirement from anyone else in the company. However, they also conceded that they felt more pressure in that environment, because they felt the weight of the expectation that the project should be finished by the end of the hackathon. This contrasted with the understanding that during a regular sprint the team is not expected to complete an entire project in such a small amount of time.

Question 3: What were the challenges that you faced during the hackathon?
The team members expressed two main challenges. The first challenge was to remove themselves from their households' daily routine. Several team members had to make various types of concessions to be present at the hackathon site. The second challenge was to deal with the increased pressure that the team felt, because the team believed it was expected that the project should have been completed and a successful demo be presented to the rest of the organization by the end of the hackathon.

Question 4: What, if any, rewards did you take from your participation in the hackathon?
The team members unanimously answered that the hackathon affirmed, or re-affirmed, their ability to produce good quality code, and that they came out of the hackathon with feelings of pride and a sense of accomplishment. Several team members also suggested that they felt they grew in their respective positions as both developers and managers. Additionally, team members reported that they developed or enhanced their levels of communication and camaraderie with the rest of the team members. The participants reported that the hackathon furthered their individual abilities and, in turn, those of the team to produce good, reliable software products. It was also mentioned that if it had not been for the hackathon, this camaraderie would not have been developed, even though the team members had (and continue to have) daily interactions virtually or in person during their work hours. As research shows (Sias & Cahill, 1998, Choi & Ko, 2020), going through the difficult event together, regardless of virtual or co-located existence, allowed for a transition of work friendships to close friendships between the team members.

Question 5: Do you believe that the project was concluded successfully at the end of the hackathon?
All of the team members answered that while they felt that the project may not have concluded to a sellable product the next day, but the project's software requirements were all fulfilled from the engineering requirements. They also felt that it was a product that could be given to the company's other departments for further processing and packaging into a complete solution.

As such, from a developer's standpoint, all team members affirmed that they believed that the project was completed.

Question 6: Can you describe the method followed for the initialization, development, and resolution of the project during the hackathon, in your own words?
The discussion of the answers for this question is included in the presentation of the software development model discussed in the next section.

Question 7: Would you participate in another hackathon? If yes or no, why? Does your answer come with any concerns?
Every team member answered positively to this question. However, there were three caveats that were stipulated. The first was that there should be additional notice provided before the hackathon for each member to take care of personal life issues ahead of the event. The second was that the team would like to have a longer planning period pre-hackathon, so that each member would feel more prepared and have a better grasp of the problem diving into the hackathon. The third was that any hackathon should take place at a location where team members would be able to have additional break-out rooms, whiteboards, etc. where discussions could be held without disrupting other team members. The location should also have adequate hardware, similar to the setup that each team member has at their own location, so that their development environment does not have significant changes. For example, team members mentioned that while they were used to working on two monitors, the location only allowed them to have one during the hackathon.

Question 8: What is the greatest lesson that you have taken away from this experience?
The team members expressed that their greatest take away was a newfound camaraderie that was built between the team members. The trust between them became stronger, and their communication became much better. The team members also mentioned that they re-affirmed their trust in their skill set, and some felt they grew through the experience and learned more than they felt possible during a regular sprint session. In all, they felt that this was a positive experience despite the aforementioned challenges because they overcame the challenges together.

THE SUGGESTED SOFTWARE DEVELOPMENT PROCESS

Based on the team's answers in question six (6) and our observations during the hackathon, we present a software development methodology that can be replicated either during a corporate hackathon or used for the rapid completion of MVPs and their development into completed software products. The process is shown below in Figure 1.

The proposed software development process begins with certain tasks that must be performed prior to the beginning of the hackathon. These are (1)

Setting the goal of the project, (2) Choosing the basic technologies to be used during the hackathon and (3) setting constraints on the process. The team then is ready to take a deep dive into software development as soon as

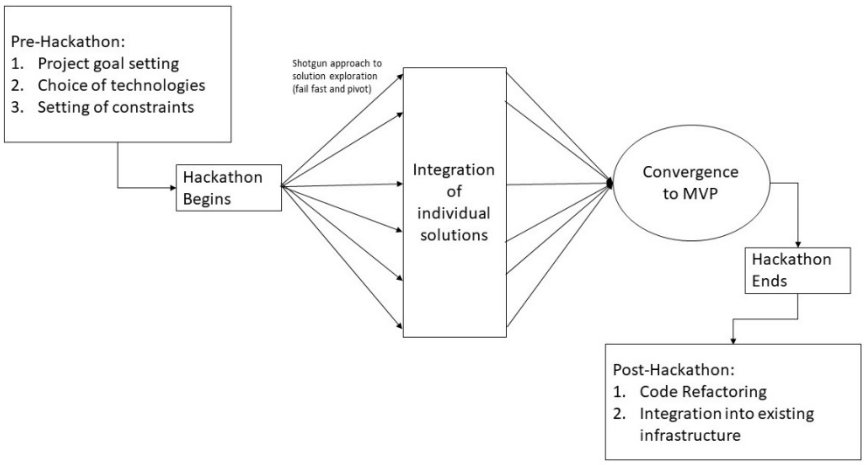

Figure 1: Software development process

the hackathon session begins. During the hackathon the team independently broke into several two-person sub-teams. Each sub-team picked which feature of the project they would tackle and used several elements from extreme programming practices (Beck & Fowler, 2001), such as real-time frequent communication with other sub-teams, constant review and change of requirements as each sub-team understood the problems and features required better, and the management during the hackathon was flat. There was one manager responsible for specifying requirements (SVP Engineering) and each sub-team conformed to those requirements. However, each sub-team was allowed to interpret and code for those requirements as they saw fit, if they adhered to the interface that was decided for the front-end / back-end communication.

The development phase followed a Just-In-Time methodology similar to processes proposed by Adaptive Software Development (Highsmith, 2000). The requirements of the software features were expected to change as the sub-teams' increased understanding of what needed to be done as time progressed. The sub-teams followed a "shotgun" approach to explore viable solutions to the various project requirements. This meant that several approaches would be evaluated fast, perhaps even used, with the expectation that if the evaluated solution did not offer a fast resolution of the requirement in question, then that approach would be dropped in favor of a different one. As such, it was expected from each sub-team to also adopt the "Fail Fast Concept" (Fail Fast | Agile Dictionary) proposed by Agile methods. In fact, at the end of the hackathon it was found that the team had scrapped at least half of the written code because of this approach. However, it was also because of this approach that the sub-teams were able to find solutions to problems fast, without being constrained by one specific approach or solution to any problem they encountered.

Development proceeded with constant communication between the two major sub-teams, the one developing the front-end and the one developing

the back-end. Any changes in requirements were communicated directly from one sub-team to another, without waiting for team meetings. Specifically, any added requirements by either of the two sub-teams were communicated directly and consideration was given to maintain consistency between the front- and back-end integration points. This was perhaps the most important factor that allowed both sub-teams to work independently and know that no matter how the teams solved the problems in the front- or the back-end, the resultant application would be integrated easily.

At the conclusion of the hackathon, a team meeting took place to evaluate the state of the developed product. It was found that the product did work as expected and was a match to the high-level requirements defined at the initial meeting. However, because of the nature of development, the team felt that the product did not go through enough testing, nor was it "polished" enough to market immediately. Therefore, after the hackathon, a regular sprint was devoted to testing and polishing the developed product. This sprint was not to change the functionality of the product, but to bring it up to marketable levels, and provide peace-of-mind to the team that the product had been adequately tested.

DISCUSSION

There are two major lessons that were learned through the process of the corporate hackathon that will be discussed in this section. The first prevalent characteristic during the development phase was that the Scrum process (Sutherland, 2014) used by the team in typical Sprints was not appropriate during the hackathon. On the other hand, the Kanban (Nihon Noritsu Kyokai, 1989) approach felt more natural in its application. Scrum aims to provide a structured approach to the team and the schedule of development whereas Kanban aims to provide the required work items, and then provides a visual way of looking at how many of those items have been completed (Alqudah & Razali, 2017). The reason for Kanban application was that it allowed for the Just-in-Time development that was needed during the project and also allowed for the development of features rather than the development of a working prototype, which was the ultimate goal of the hackathon. Whereas a project of this type can be broken down by features and each feature assigned to a team of one or more developers, it cannot be broken down into working prototypes of smaller sub-projects that can do something functional without having the whole integration. As such, the Kanban approach lends itself towards hackathon-based development.

A second lesson from the experience was that part of the success of the project was because of the team members being allowed to work without having to attend meetings outside of the project's scope, nor being distracted by other requirements from other projects that occurred at the company at the time of the hackathon. The team was focused entirely on completing the project at hand and, thus, the team's momentum was not broken. Rather, it was allowed to build towards their optimum capability. The hackathon allowed the team members to enter the flow state (Csikszentmihalyi, 2008). As Csikszentmihalyi (2008) states "During flow, people typically experience

deep enjoyment, creativity, and a total involvement" – a state that is experienced only through intense focus on a task. The team members entered this state as a team and stayed in this state throughout the hackathon allowing them to perform in a more optimal manner, as some stated during their interviews. We posit that because the team members entered this state as a team, their resultant camaraderie was heightened and their sense of trust of each other was strengthened. However, this last statement is just a supposition without hard data to support it. It does remain one of the open questions of this article.

CONCLUSION

In this article we have presented the results of a closed hackathon that took part at Core Business Technologies. We have discussed the intangible rewards for the team that took part in the hackathon, which were distilled through the analysis of interviews with each participant. We have also presented a methodology which was derived from the analysis of the process followed by the team to reach product viability during the hackathon. Finally, we discussed two major take-aways that were taken from the experience, that Kanban is a better choice for hackathon-based projects, and that teams do perform better if they are allowed to only work with focus on one project at a time. We posit that the software development process described here is replicable not only in the confines of a hackathon, but also in the context of larger software engineering projects that take place in software development organizations. We believe that this methodology can produce software solutions faster, given that the two lessons that were discussed are followed. Finally, it needs to be said that having employees work under hackathon conditions is not a viable everyday practice as was evident by the increased pressure the team members felt. This pressure though, did not overbear the team members, who at the end saw that they received several intangible benefits from the process, and felt that the process was a positive experience, despite the additional pressure.

ACKNOWLEDGMENT

The author would like to acknowledge all the team members that participated in the hackathon and the company management for its support of the authoring of this article.

REFERENCES

Alqudah, M., & Razali, R. (n.d.). A comparison of scrum and Kanban for identifying their selection factors.

Anderson, D. J. (2010). Kanban: Successful Evolutionary Change for Your Technology Business. Blue Hole Press.

Beck, K., & Fowler, M. (2001). Planning extreme programming. Addison-Wesley.

Bigliardi, B., Ferraro, G., Filippelli, S., & Galati, F. (2021). The past, present and future of open innovation. European Journal of Innovation Management, 24(4), 1130–1161. 10.1108/EJIM-10-2019-0296

Blackburn, J. D., Scudder, G. D., & Wassenhove, N. V. (1996). Improving speed and productivity in software development: a global survey of software developers. IEEE Transactions on, 22(12), 875–885. 10.1109/32.553636.

Chai, S., & Freeman, R. B. (2019). Temporary colocation and collaborative discovery: Who confers at conferences. Strategic Management Journal, 40(13), 2138–2164. 10.1002/smj.3062

Chesbrough, H. W. (2003). The era of open innovation. MIT Sloan Management Review, 44(3), 35–41.

Choi, Y. & Ko, S. (2020). Roses with or without thorns? A theoretical model of workplace friendship, Cogent Psychology, 7(1). 10.1080/23311908.2020.1761041.

Fail Fast | Agile Dictionary. (n.d.). Agile Academy. Retrieved January 3, 2023, from https://www.agile-academy.com/en/agile-dictionary/fail-fast/

Flores, M., Golob, M., Maklin, D., Herrera, M., Tucci, C., Al-Ashaab, A., Williams, L., Encinas, A., Martinez, V., Zaki, M., Sosa, L., & Pineda, K. F. (2018). How can hackathons accelerate corporate innovation? IFIP International Conference on Advances in Production Management Systems (APMS), 165–175. 10.1007/978-3-319-99704-9_21.

Highsmith, J. A. (2000). Adaptive Software Development: A Collaborative Approach to Managing Complex Systems. Dorset House Pub.

Nihon Nōritsu Kyōkai. (1989). Kanban Just-in Time at Toyota: Management Begins at the Workplace (D. J. Lu, Trans.). Taylor & Francis.

Peter, W. (1996). Interoperability. ACM Computing Surveys, 28(1), 285–287.

Pe-Than, E. P. P., Nolte, A., Filippova, A., Bird, C., Scallen, S., & Herbsleb, J. D. (2019). Designing corporate hackathons with a purpose: The future of software development. IEEE Software, 36(1), 15–22. 10.1109/MS.2018.2901105470.

Sias, P. M. & Cahill, D. J. (1998). From coworkers to friends: The development of peer friendships in the workplace. Western Journal of Communication, 62(3), 273–299. 10.1080/105703198093746110.

Sutherland, J. (2014). Scrum: The Art of Doing Twice the Work in Half the Time. Crown.

Trott, P., & Hartmann, D. (2009). Why 'open innovation' is old wine in new bottles. International Journal of Innovation Management, 13, 715–736.

Zukin, S., & Papadantonakis, M. (2017). Hackathons as Co-optation Ritual: Socializing Workers and Institutionalizing Innovation in the "New" Economy. In S. P. Vallas & A. L. Kalleberg (Eds.), Precarious Work (pp. 157–181). Emerald Publishing Limited. 10.1108/S0277-283320170000031005.