How to organize a hackathon – A planning kit

Preprint · August 2020

DOI: 10.48550/arXiv.2008.08025

9 authors, including:

Alexander Nolte
Eindhoven University of Technology
140 PUBLICATIONS **1,347** CITATIONS
SEE PROFILE

Ei Pa Pa Pe Than
Carnegie Mellon University
28 PUBLICATIONS **520** CITATIONS
SEE PROFILE

Abasi-Amefon Obot Affia
University of Tartu
17 PUBLICATIONS **198** CITATIONS
SEE PROFILE

Arun Kalyanasundaram
Carnegie Mellon University
11 PUBLICATIONS **313** CITATIONS
SEE PROFILE

How to organize a hackathon – A planning kit

Alexander Nolte
University of Tartu, Estonia
Carnegie Mellon University, Pittsburgh, PA, USA
alexander.nolte@ut.ee

Ei Pa Pa Pe-Than
Carnegie Mellon University, Pittsburgh, PA, USA
eipa@cmu.edu

Abasi-amefon Obot Affia
University of Tartu, Estonia

Chalalai Chaihirunkarn
Carnegie Mellon University, Pittsburgh, PA, USA

Anna Filippova
GitHub Inc., San Francisco, CA, USA

Arun Kalyanasundaram
Google LLC, Mountain View, CA, USA

Maria Angelica Medina Angarita
University of Tartu, Estonia

Erik Trainer
Fidelity Investments, New York, NY, USA

James D. Herbsleb
Carnegie Mellon University, Pittsburgh, PA, USA
jdh@cs.cmu.edu

ABSTRACT

Hackathons and similar time-bounded events have become a global phenomenon. Their proliferation in various domains and their usefulness for a variety of goals has subsequently led to the emergence of different formats. While there are a multitude of guidelines available on how to prepare and run a hackathon, most of them focus on a particular format that was created for a specific purpose within a domain for a certain type of participants. This makes it difficult in particular for novice organizers to decide how to run an event that fits their needs. To address this gap we developed a planning kit that is organized around **12 key decision** that organizers need to make when preparing and running a hackathon, and the tradeoffs that drive decision-making. The main planning kit is available online while this report is meant as a downloadable and citable resource.

Author Keywords

Hackathon; Planning kit; Organizers; Collaboration; Social computing

INTRODUCTION

Hackathons and similar time-bounded events have become a global phenomenon [63] drawing interest from practitioners and researchers alike [18, 43]. During such events participants typically form teams and engage in intensive collaboration over a short period of time to complete a project that is of interest to them [55]. Due to their versatility as means to develop innovative ideas [7, 9], add features to existing software [81], foster learning [1, 22, 46], tackle civic and environmental issues [84, 5, 6, 58] and build new or expand existing communities [49, 81, 45] they have been adopted by various domains such as entrepreneurship [47, 10], small-medium size enterprises [34], large corporations [50, 31, 59], (higher) education institutions [57, 23, 8], civic engagement groups [28, 36, 29], (online) communities [2, 15] and others. This spread into various domains has subsequently led to a large variety of different hackathon formats making it difficult in particular for inexperienced organizers to decide how to run an event that fits their needs.

While there are a multitude of guidelines available on how to prepare and run a hackathon, most of them focus on a particular format such as collegiate events [41], civic events [62, 12], science hackweeks [32, 3] or others that were created for a specific purpose within a domain for a certain type of participants. Such guidelines are a great resource for organizers who already know what type of hackathon they want to organize. They however often lack information about why certain aspects of a proposed hackathon format were designed in the presented way and how the format can be adapted to fit different needs than the ones the format was designed for. To address this gap, we developed a planning kit that is organized around **12 key decision** that we found to be crucial to consider when organizing a hackathon. The insights presented in the planning kit are based on our work researching, supporting and co-organizing hackathons for the past five years [54, 52, 19]. We intentionally did not go into the details of how to organize an event in general such as how to secure a room, book catering and ensure that participants have WiFi access. Instead we focus on the specifics that organizers need to consider when planning a hackathon that aims to foster a specific goal for a specific audience in a specific domain. The main planning kit is available online[1] while this report is meant as a downloadable and citable reference. We would thus like to encourage the reader to consult and use the online version since we will continue updating it in the future.

The remainder of this report is structured as follows. We will first provide example timelines for two specific types of hackathons before elaborating on the 12 aforementioned

[1] https://hackathon-planning-kit.org/

decision. The example timelines are mainly aimed for first-time organizers. They walk the reader through the different decision points and show examples for how and why they could decide for one or the other option. Afterwards we will present the 12 decisions in detail before providing a short overview of ongoing and future work.

We would like to encourage everyone that uses this planning kit or that has experience organizing, supporting or researching hackathons to provide feedback or suggestions on how to improve the planning kit. Please feel free to contact us[2] or directly propose changes on GITHUB[3].

EXAMPLE TIMELINES

The following example timelines show an idealized procedure for the organization of two common types of hackathons that focus on fostering entrepreneurship and the establishment of a community. These timelines indicate time and outcome of key decisions.

ENTREPRENEURIAL HACKATHON

This timeline shows an example for a medium sized hackathon (between 100 and 150 participants) which aimed to attract entrepreneurs and foster innovative projects that can become successful businesses.

4 months before the hackathon:

- *Goal:* Fostering the regional development of the start-up ecosystem related to the cyber security domain.
- *Theme:* Cyber security
- *Competition / cooperation:* Decision for a competition style event. Teams can win prizes ranging from tech gadgets to start-up coaching and participation in accelerator programs.
- *Duration / breaks:* Discussion about and decision for a tentative date for a 48-hour event starting in the afternoon of the first day and ending in the afternoon of the third day.
- *Agenda:* Discussion about decision for a tentative agenda that includes daily checkpoints, a final pitch presentation and an award ceremony.
- *Participant recruitment:* Creation of an information hub. Contacting local universities, start-up hubs, tech companies, accelerator programs and government agencies to spread news about the event through their networks. Start of the social-media campaign.
- *Stakeholder involvement:* Discussing with representatives of aforementioned groups about their interest in the event and invitation to participate as mentors, give thematic talks and provide sponsorship and prizes.

3 months before the hackathon:

- *Participant recruitment:* Creation of an online form that covers participantsâĂŹ contact details, their current profession and their projected role during the hackathon. Registration requires the payment of a small nominal fee that will be refunded after participation.
- *Ideation:* Decision for a pitch-style ideation process at the hackathon. Participants can indicate if they have a project idea for the hackathon and provide a short description as part of the registration form.
- *Mentoring:* Identification and invitation of a diverse group of individuals who can provide mentorship related to cyber security, various programming languages, design, entrepreneurship, marketing and others. Decision for a combination between mentor teams and individual on-demand support.

1 month before the hackathon:

- *Team formation:* Teams will form around ideas. They cannot have less than 3 members, have to be of similar size and include individuals with diverse expertise and interests including cyber security, programming, design and entrepreneurship.
- *Stakeholder involvement:* Finalization of sponsor agreements including prizes and talks at the hackathon.
- *Participant recruitment:* 1-day competitive ideation events in three cities close to the main hackathon location during which participants can start working on ideas and form teams. Winners receive travel support for the main hackathon.

1 week before the hackathon:

- *Agenda:* Adding final event agenda including thematic talks, trainings and talks by sponsors during each day to the information hub.
- *Ideation:* Adding information about the pitch procedure to the information hub.
- *Mentoring:* Introduction of mentors on the information hub.

Hackathon day 1:

- *Agenda:* Welcoming words by the organizers, presentation of hackathon agenda including idea pitches, mandatory checkpoints for idea proposers, talks and trainings, expected outcome (pitch presentation) and jury. Reiteration of information hub and contact details for organizers and mentors.
- *Stakeholder involvement:* Introduction of sponsors and supporting individuals and institutions.
- *Mentoring:* Introduction of mentors, their area of expertise and their role during the hackathon.
- *Ideation:* Participants pitch ideas in front of organizers, mentors and other participants including information about which expertise they perceive to be required. Everyone can pitch. Not only participants that submitted ideas through the registration form.
- *Team formation:* Ideas are written on large sheets of paper that idea proposers hang on the walls in the foyer of the hackathon venue. Participants that did not pitch ideas go

[2] contact-info@hackathon-planning-kit.org

[3] https://github.com/herbsleb-group/herbsleb-group.github.io

around and talk to idea proposers, discuss their expertise and voice their interest. Idea proposers select suitable team members based on interest and expertise. Ideas that do not gain sufficient interest by other participants are abandoned and the proposers of these ideas join other teams.

- *Agenda:* Idea proposers present their teams. Quick check by the organizers if the teams are of roughly equal size and if all teams have sufficient expertise to start working on their projects. Teams start hacking.
- *Mentoring:* Mentors meet and form teams with diverse expertise. Each mentor team is assigned to a group of hackathon teams that they support during the hackathon. Mentors focus on their teams but also support others if necessary.

Hackathon day 2:

- *Agenda:* Idea proposers present their progress in front of organizers and mentors at the beginning of the day (1st mandatory checkpoint).
- *Mentoring:* Mentors meet, discuss potential difficulties that certain teams face and decide for mentors with related expertise to support them.
- *Stakeholder involvement:* Thematic talk before lunch time.
- *Agenda:* First pitch training for idea proposers shortly before next checkpoint.
- *Agenda:* Idea proposers present their progress in front of organizers and mentors at the end of the day (2nd mandatory checkpoint).
- *Mentoring:* Mentors meet, discuss potential difficulties that certain teams face and decide for mentors with related expertise to support them.

Hackathon day 3:

- *Agenda:* Idea proposers present their progress in front of organizers and mentors at the beginning of the day (3rd mandatory checkpoint).
- *Mentoring:* Mentors meet, discuss potential difficulties that certain teams face and decide for mentors with related expertise to support them.
- *Agenda:* Second pitch training for idea proposers before lunch time.
- *Agenda:* Third and final pitch training for idea proposers a few hours before the final pitches.
- *Agenda:* Final pitches of idea proposers in front of all participatnts, jury, organizers, mentors and online audience (live stream).
- *Competition / cooperation:* Online voting for audience favorite, jury decision and award ceremony.
- *Duration / breaks:* Group pictures, networking, end of the hackathon and departure.

After the hackathon:

- *Continuity planning:* Organizers share summary of the hackathon on information hub, connect interested teams with stakeholders and periodically contact winning teams about their progress.

COMMUNITY HACKATHON

This timeline shows an example for a small-scale hackathon (< 50 participants) which aimed to bring together interested researchers, students and practitioners and form a community around a novel resource.

4 months before the hackathon:

- *Goal:* Formation of a community around a novel data resource which contains a virtually complete collection of all open source projects around the world.
- *Theme:* Development of research ideas and initial prototypes that utilize the resource.
- *Competition / cooperation:* Decision for a cooperation style event that focuses on joint exploration of the resource.
- *Duration / breaks:* Discussion about and decision for a tentative date.
- *Participant recruitment:* Identification of key individuals in industry, universities and scientific communities that could benefit from the resource and that can support the recruitment of individuals that would be interested in and would benefit from using the resource.
- *Stakeholder involvement:* Discussions with these key individuals as well as developers and maintainers of similar resources about their interest in the resource.
- *Ideation:* Decision to ask participants for initial ideas through the registration form and conduct additional ideation during the hackathon.

3 months before the hackathon:

- *Participant recruitment:* Invitation of potential participants through previously identified key individuals. Registration through an online form that covers their contact details, open source handle, preferred programming languages and interests in the resource. Selection of participants based on interests.
- *Ideation:* Ask invitees to propose initial ideas for hackathon projects through the registration form.
- *Continuity planning:* Invitation of selected participants and key individuals to common communication channel. Creation of an information hub to spread information about the hackathon and the resource.
- *Mentoring:* Identification and invitation of individuals who are familiar with the resource and relevant technologies to serve as mentors. Decision for dedicated mentors that are assigned to individual teams.

1 month before the hackathon:

- *Specialized preparation:* Development of documentation for the resource including sample code for selected project ideas that were submitted through the registration form. Sharing of documentation through communication channel.
- *Duration / breaks:* Decision for a 3-day event starting in the afternoon of the first day and ending in the afternoon of the third day with breaks over night.
- *Agenda:* Development of Development of a first complete agenda that focuses on hacking during the day with breaks breaks during each day for socializing and networking.
- *Ideation:* Planning for ideation session at the beginning of the event. Card based brainstorming with questions focusing on the usability and usefulness of the resource.
- *Team formation:* Decision that teams will form around ideas and that members should come from different institutions.

1 week before the hackathon:

- *Specialized preparation:* Pre-hackathon webinar to introduce participants to the capabilities and usage of the resource. Interaction during webinar that allows participants to connect to the resource and run code samples.
- *Mentoring:* Introduction of mentors and their area of expertise at the webinar.

Hackathon day 1:

- *Agenda:* Welcoming words by the organizers, presentation of hackathon agenda and expected final submission (source code and presentation slides) and reiteration of communication channels and information hub.
- *Stakeholder involvement:* Introduction of supporting individuals and institutions.
- *Mentoring:* Introduction of mentors, their area of expertise and their role during the hackathon.
- *Agenda:* Participants introduce themselves to each other.
- *Ideation:* Card based brainstorming. Participants write ideas on cards and share them with the organizers.
- *Agenda:* Break for participants to socialize and network and for organizers to integrate ideas that were submitted through the registration form and to pre-structure brainstorming cards into thematic clusters that can be the basis for hackathon projects.
- *Ideation:* Discussion of pre-structured clusters and adjustment based on participant input.
- *Team formation:* Participants select cluster / project based on their interests while observing the rule that they should be from different institutions. Adjustments to ensure that teams are roughly of equal size.
- *Mentoring:* Mentors join teams and support them to connect to the resource, scope their project and help with technical issues. Mentors focus on their teams but also support others if necessary.
- *Agenda:* At the end of the day teams introduce their members, share their concrete project idea and their plans for the next day (1st checkpoint).
- *Duration / breaks:* Social dinner at the end of the day.

Hackathon day 2:

- *Agenda:* At the beginning of the day the organizers lay out the agenda for the day and reiterate the expected final submission. Teams explain their project ideas and share their plans for the current day (2nd checkpoint).
- *Duration / breaks:* Lunch break.
- *Agenda:* After lunch teams share their progress, problems they ran into and their plans for the rest of the day (3rd checkpoint)]
- *Agenda:* Social game during the afternoon.
- *Agenda:* At the end of the day teams share their progress, problems they ran into and their plans for the final day (4th checkpoint).
- *Duration / breaks:* Social dinner at the end of the day.

Hackathon day 3:

- *Agenda:* At the beginning of the day the organizers lay out the agenda for the day and reiterate the expected final submission. Teams share their progress, problems they ran into and their plans for the remainder of the time (5th checkpoint).
- *Agenda:* Final presentations of teams before lunch. Discussions about the content of the presented projects and problems the teams encountered during the hackathon.
- *Continuity planning:* Teams share presentations and code repositories through communications channel.
- *Duration / breaks:* Lunch break, group pictures, end of the hackathon and departure.

After the hackathon:

- *Continuity planning:* Organizers distribute summary of the event directly **After the hackathon:** and provide regular updates about the resource through communications channel.
- *Stakeholder involvement:* Organizers suggest for stakeholders to share publications and other outcomes they produce using the resource through communications channel.

KEY DECISIONS

In the following we will outline the twelve key decisions that organizers should consider when planning a hackathon. For each decision, we provide information about **when** organizers should consider making it, **who** should be involved in the decision, **how** to make the choice and implement its result, and discuss potential **tradeoffs** among the various options. The order in which the decisions are presented is deliberate but not strict. Moreover, not all decisions are relevant for each hackathon. Prospective organizers should thus perceive

the following list as a suggestion for which decisions can be important but decide for themselves which ones they consider for their specific purpose.

Goal

Goal setting is key for hackathon design. All parties should be clear about their goals going into an event and consider the attainability and clarity of these goals. A failure to consider them could result in disappointment among participants and organizers.

When?

Setting goals for the hackathon should take place *before any planning of the event begins*. A common timeframe is to form goals about 4 months before the start of the event. Before deciding for one or multiple goals the organizers can consult with projected future participants (section 5.5) and potential stakeholders (section 5.4) about the value and feasibility of the projected goals and continue discussions about details after the general direction has been set.

Who?

Organizers, stakeholders, and participants are involved in setting goals for their hackathon. In practice, organizers often set initial goals and modify them in collaboration with stakeholders, which is done in the early phase of hackathon planning. Participants are often also asked to express their goals for the event through e.g. a pre-hackathon survey (section 5.5). It is important to note here that the organizersâĂŹ goals may not always be aligned with those of the participants. For example, organizers might aim to foster entrepreneurship whereas participants simply aim to pursue their interest related to a particular topic or project. It is not necessary that organizers and participants have identical goals, but organizers should be aware of potential goal disparities and take necessary actions to maximize the satisfaction of participants with different goals [42]. The participants, for example, might have a superset of the organizersâĂŹ goals, or the organizersâĂŹ goals may be achievable even if only a subset of participants shares them. The organizers should be aware though if goals are incompatible, (e.g. the organizers want to benefit a charity, while the participants mainly want to start a business), the participants may have unrealistic expectations and leave disappointed. It would be wise for organizers to adapt their recruiting approach (section 5.5) and their messaging to attract attendees with compatible goals.

How?

There are several goals that organizers may aim to achieve when organizing a hackathon. Clarity of goals is strongly related to participant satisfaction and outcome quality [20].

The most common goal around which hackathons are organized is the **production of artifacts or products**. We observe two different strategies that organizers use to achieve this goal. The first strategy involves the recruitment of experts from different fields (section 5.5) and the facilitation of brainstorming (section 5.8) to produce cross-pollinated project ideas. Brainstorming has also been found to be associated with better outcomes for self-identified minorities [20]. The organizers using this strategy should devote most of their effort on recruiting stakeholders (section 5.4) from diverse groups that will benefit from meeting and working with each other. For example, they might include local startup communities, volunteers and activists, non-profits, accelerator programs, investors, technology companies and others. Moreover, organizers should design their events in a way that fosters participants to meet and work with people outside their regular networks and contribute to initiatives outside their usual scope. The sections dedicated to ideation (section 5.8) and duration/breaks (section 5.7) contain helpful ideas to facilitate networking among participants. One common example are regular checkpoints (section 5.10) to ensure that participants communicate with each other about their projects several times during the event. For example, a series of hackathons organized by Garage48 [25] aims to foster the startup ecosystem by connecting entrepreneurs with domain experts, investors, accelerators and others. Similarly, BioInnovation Days [53] gathers together students, researchers, mentors, and startups to develop prototypes in the domain of bio-medicine. The Global hack [70] is a recent example of an online movement where enthusiasts from different countries hosted several hackathons to address the global pandemic.

The second strategy is teaming up experts and novices as mentors and apprentices, which enables the positive side effects of vertical **networking and learning**. Hackathons adopting this strategy should focus their efforts on identifying experts in the field and themes of their events, abilities of participants, modularization of projects to support parallelism, and the use of relevant mentoring strategies [49] (section 5.11). One example of this type of hackathon is Astro Hack Week [65] where a diverse group of participants ranging from novices to experts in the fields of astronomy, physics, statistics, machine learning, and data science gather together to learn skills necessary for the analysis of astronomical data through tutorials and then solve open problems in the astronomical community. Another example is a series of hackathons organized by the team at Space Telescope Science Institute (STScI) managing the science program and operations of the Hubble Space Telescope (HST) [53] where astronomers and software engineers are teamed up to facilitate fruitful cross-domain discussions among them. World of Code [77] is another example for a hackathon where students are teamed up with faculty members to learn how to do quantitative research on the behaviors of open source software (OSS) projects through the worldâĂŹs largest OSS dataset.

Trade-offs

1. **Goal compatibility (Organization goals vs personal goals):** Although the organizational goals may not necessarily be compatible with personal goals of the participants [42], the huge disparity among goals will lead to dissatisfying with experience and artifacts generated. One essential responsibility of organizers is then to identify any potential misalliance among goals and ways to mitigate it. This could be done by clear a presentation of organization goals, selective recruitment of participants (section 5.5), upfront negotiation of goals, and adaptation and customization of the hackathon process and outcome based on goals (section

5.12).

2. **Illustrating a concept vs producing something useful:** Teams can either aim to develop illustrative proofs of concepts, or software that can actually be used to help accomplish a task. In our experience, teams that have not worked together extensively (i.e. flash or ad hoc teams) are likely to spend more time discussing what to build and how to go about it, and how to pitch their projects to potential stakeholders, and they should thus aim to produce **lightly-engineered demos and videos**, with mock-up data, which require little engineering effort. For teams that have worked together extensively, as in a corporate hackathon where intact existing teams participate, it is often desirable for them to aim for a prototype with **sufficient functionality that could be used almost immediately in the real context** [56].

3. **Challenging vs achievable:** Research has shown that groups accomplish more and are more inclined to continue working on a project when they have goals that are challenging [48]. Goals that are too easy fail to motivate them to do their best. On the other hand, goals that are so challenging that they are, or seem to be, impossible to achieve can also be demotivating. Choosing appropriate goals involves an understanding both of the participants capabilities and interests, as well as a realistic view of what can be achieved in a short time frame. One useful approach for managing this tradeoff is to set (or encourage each team to set) both a relatively easy goal, and a more difficult âĂIJstretchâĂİ goal. This gives teams the opportunity to feel that they achieved something meaningful, as well as motivation to go beyond this and attempt something very challenging.

4. **Goal clarity:** Although hackathons should ideally be designed to achieve goals of both organizers and participants, there are circumstances where their goals are not aligned [20, 42]. This occurs, for example, when organizers hold a hackathon to connect professionals working across different fields, but participants wish to develop viable product prototypes. It is important for organizers to recognize such potential goal conflicts and plan accordingly to be reasonably able to achieve their intended goals. One way to address this tradeoff is for organizers to clearly state their goals during recruitment (section 5.5) so that participants know what to expect.

Theme

Most hackathons focus on a specific theme. A theme refers to a specific topic or cause that motivates the organization of a hackathon which in turn sets its boundary. A theme helps organizers to address a particular topical area and devise solutions for problems within this area.

When?

Organizing a hackathon commonly starts with a theme. Themes are generally developed in conjunction with the goals of a hackathon (section 5.1), typically 4 months ahead of the hackathon.

Who?

While the decision for a specific theme resides with the organizers, it is advisable to discuss it with relevant stakeholders such as non-profits, investors, educational institutions or others (section 5.4).

How?

The organizers should decide the initial theme of the event and invite potential stakeholders interested in their proposed theme to discuss its feasibility and develop project ideas related to this theme. Themes can range from general to specific, e.g. fighting a global crisis [70], mining a large-scale open source data source [77], building cybersecurity solutions and teaching cybersecurity skills [25, 1], learning data science through solving astronomical and geo challenges [65, 69], computational biology [51], neuroscience [74], environmental issues [84], and high performance computing [72].

The theme focuses potential project ideas on immediate needs or outstanding challenges in the chosen area. One such need is to **speed up development work**. A series of hackathons by a team of the Space Telescope Science Institute (STScI) that manages science programs and conducts experiments on the Hubble Space Telescope (HST) [53] is one such example. In this case events were organized to speed up the conversion of tools to a language, bringing together individuals who are using such tools for their work. Hackathons organized by Garage48 [25] are other examples of speeding up the development of cybersecurity solutions. Another need is solving outstanding challenges which could take a form of technical debt or scientific advancement. A team participating in a corporate hackathon we studied [44], for example, utilized the time during that hackathon to implement a tool to automate recurring manual work.

Another reason for choosing a specific theme is **hack to train**. Astro hack week [65] and Geo Hack Week [69] are examples of this type where participants are first trained to apply basic analysis and coding skills and gather domain knowledge before moving on to solve challenges in the respective domains. SingaporeâĂŹs BrainHack of the Organization for Human Brain Mapping (OHBM) in 2018 [74] is another example. There the organizers held parallel training and hacking tracks to get participants up to speed with technologies and approaches they might require to work on projects in this domain.

A theme might also be chosen to **build resources for a specific community**. One example for this is the World of Code (WoC) hackathon [77] that aimed to bring interested researchers and together and identify requirements for a large archive of open source data. Another example are events organized by the Open Bioinformatics Foundation (OBF) which aims at improving and extending existing resources and infrastructure and publishing papers about the accomplished work [51].

Trade-offs

1. **Speed vs quality:** Hackathons operated under the themes related to of speeding or accelerating could quickly advance

the development work in a specific domain with proper implementation. Getting things done fast and quick however does not mean that such hackathons produce quality artifacts, i.e. they might be workaround solutions and need code cleanup or dependency reduction, and thus follow-up work is usually required to ensure the quality of the developed artifacts.

2. **More general vs more specific:** Themes can be very specific, e.g., building an app to monitor occupancy levels in a local shelter, or much more general, e.g., help local governments support sustainability. Very specific themes can be effective in helping to ensure the work actually serves a useful purpose but runs the risk of not matching up well with the interests of a large group of potential participants. More general themes allow more latitude for participants to develop projects that align with their personal interests but run a greater risk of not matching up with a real need. To address this tradeoff organizers should make sure they understand the motivations potential participants. To achieve this, they may want to conduct a survey to assess their attitudes toward different theme variations.

Competition / cooperation

One key decision that organizers need to make when designing a hackathon is whether to provide external incentives in the form of prizes, which introduces a strong element of competition into an event.

When?

The decision whether to have a competitive or collaborative hackathon should be made several months before an event. Competitive events, in particular, take considerable time to organize. Organizers must find credible judges, decide on suitable award criteria, determine prizes and prize categories, and ensure that prizes can be handed out at the end of an event. If the prizes have considerable monetary value or will involve a time commitment from experts whose time is very limited (e.g., the opportunity to pitch the teamâĂŹs winning idea to potential investors), there may be considerable work for the organizers to secure sponsorship.

Who?

Organizers, sponsors, and other stakeholders (section 5.4) involved in setting goals for a hackathon should consider whether and how to introduce competition to their event. It is also extremely important to consider the goals of participants – picking a structure that does not further their goals will make recruiting (section 5.5) difficult. If participant goals are not clear to the organizers, a brief survey or interviews with a sample of the population of interested participants can be very helpful.

How?

Cooperative events are typically structured around a **common goal** (section 5.1) or **theme** (section 5.2). An example for cooperative events is a series of cooperative hackathons [17] that was organized to accelerate the development of integrated web services in the field of bioinformatics. Teams there worked on projects involving data standardization and interoperability of tools and services among others. Other cooperative hackathons aim to perfect an artifact. For example, the Debian Linux project launched the Debcamp event [67], a hacking session before the Debian Conference, where software developers whose work depends on the Debian Linux software distribution worked together to promote the redistribution, general availability, and mutual compatibility of software. Another example is a corporate hackathon series organized by STScIâĂŹs Hubble Space Telescope team that aimed to accelerate the migration of data analysis tools implemented in an old obsolescing language to a contemporary one [53].

Competitive events focus on teams winning prizes. Winners can be selected by a **jury** or based on a **popular vote**. Some hackathons also award special prizes to projects that meet specific challenges posed by sponsors or other stakeholders. If winners are selected by a jury, it is necessary to invite judges and to specify suitable judging criteria. For judges, the organizers may consider recruiting (domain) experts from a university, community leaders, or representatives from tech companies [37]. Some commonly used judging criteria include appeal to market, creativity, originality, completeness and level of difficulty. For a competition to be perceived as fair, it is important that judging criteria are well known in advance and that organizers make sure that participants do not simply turn up with solutions that have been built prior to the event. There are different approaches to award prizes based on a popular vote. Voting can e.g. be limited to on-site participants only or it can be replaced or augmented by online voting. The latter requires an online voting system as well as final presentation sessions to be live-streamed or presentation materials to be distributed online.

In order to select the winning teams, the organizers should create a dedicated session at the conclusion of a hackathon to allow each participating team to pitch and demonstrate their idea to the entire audience of an event. This session could take the form either of an **on-stage presentation** or a **(science) fair**. In the former approach, each team is typically given a set time to present their prototypes in front of the entire event audience. For this it is important to inform participants in advance how their presentation will be evaluated, the presentation format, and the time limit. The commonly used presentation format starts with a brief introduction of team members and problems that they tried to solve, which is followed by a live demo [64]. It is also possible that teams record a short introductory video and focus on their demo during the presentation. In a (science) fair style presentation session [50], each team is commonly allocated a booth in a large-enough space to set up their presentations and judges and visitors visit each booth to interact with the teams. This approach facilitates a greater face-to-face interaction between participants, judges, and other attendees. At the Microsoft OneWeek Hackathon we observed [50, 44], teams could both attend a science fair and upload a video to an online platform where other participants and observers could vote on their favorite projects. This online voting enables individuals who are not able to attend the fair in person to participate in winner selection.

Prizes can vary greatly, with tech gadgets, cash prizes and

opportunities for continued development of winning ideas probably being the most common. The opportunities for further development can take the form of providing additional resources, computing power, freeing up participantsâĂŹ time to work on their project post-hackathon, or simply the opportunity to pitch their idea to top executives or investors. Awarding cash prizes is not always feasible or recommended approach [38]. Major League Hacking (MLH) for example provides a few non-cash prize ideas [27] such as laptops, headphones, conference tickets, etc. as alternatives. Moreover, hackathon organizers need to decide for **how many prizes** they offer in relation to the number of participating teams. Offering many prizes at a small event might reduce their perceived value which in turn can negatively affect participant motivation [48].

Trade-offs

1. **Competition vs cooperation:** Competition is suitable for hackathons aiming to create innovations because, under competitive pressure, teams are likely to generate unique solutions to differentiate themselves from other competing teams and put more effort in their projects. However, competition discourages communication among teams, and hence is not suitable for hackathons aiming to enrich networking among participants beyond their own teams, or to engage participants in a common goal. On the other hand, cooperation works well whenever it furthers the organizersâĂŹ and participantsâĂŹ goals, such as promoting a civic cause, providing different pieces of an integrated solution, or learning about programming tools or a particular domain. To reduce the severity of this trade off, it might be advisable to de-emphasize the prizes in a competitive event so that participants do not over-emphasize competition, e.g., by having prizes that are largely symbolic rather than great cash value. Heavily emphasizing prizes might put off many potential participants especially if they feel that the odds of winning a significant prize are low, that their idea might not work well with the proposed judging criteria or that judging might not be fair.

2. **Jury vs popular vote:** Experts are more appropriate as judges if the desired criteria are technically complex. They will be much more able to determine if a prototype is actually feasible, for example, and if it actually addresses the problem claimed. On the other hand, if the desired result is something that addresses a widely-experienced need, or to produce something that seems cool or stylish, a popular vote may be more suitable.

3. **High value vs low value prizes:** High value prizes create a more serious atmosphere and level of competitiveness, and organizers will have to be very careful about finding skilled judges and applying the pre-specified criteria in a way that can be perceived as fair by hackathon participants. The rules will have to be very clear, e.g., about whether teams can form and start work prior to the hackathon and whether they can bring code they have written with them to the hackathon. If the highest possible level of professionalism, skill, and innovation is the goal, high value prizes are a good choice. Low value prizes are better when organizers and participants have a variety of goals such as learning, expanding social ties, or attracting new people to a community. As the value of prizes approaches zero (e.g., special hats or shirts), competitive hackathons can look much more like collaborative hackathons. With inherently competitive populations, however, even symbolic prizes can make some teams behave very competitively.

Stakeholder involvement

Hackathons commonly focus on a specific theme (section 5.2) or take place in a specific domain. It thus appears reasonable to include stakeholders related to this theme or domain to participate in the organization, execution and follow-up of a hackathon. They can provide valuable input, help set the stage of an event, make it more engaging and fun for participants and support the sustainability of hackathon outcomes (section 5.12). Deciding about how and when to involve stakeholders in the planning and execution of a hackathon is thus a crucial decision that organizers have to take because it will fundamentally shape the experience of participants during the hackathon.

When?

Organizers should think about which stakeholders to involve early in the planning process because their input might have a considerable impact on the design of the event itself. The way each stakeholder participates in the planning, execution and follow-up of a hackathon is then subject to individual planning and can potentially happen later in the planning process. Stakeholders and their role in relation to the hackathon should however be decided upon and announced prior to the start of the hackathon to be able to include them in information material.

Who?

While conducting a traditional stakeholder analysis [61] might be too time-consuming, organizers certainly should think about who might be interested in or affected by the outcomes of the hackathon they plan to organize. Depending on the theme of the event organizers might want to involve university departments (e.g. for a collegiate event [76]); investors, incubators, and customers (e.g. for an entrepreneurial event [25]); managers and executives (e.g. for a corporate event [50]); volunteers and activists (e.g. for a civic event [66]); scientists and technical experts (e.g. for a scientific event [53]). These are just a few examples for potential stakeholders. The decision for whom to involve and how to involve them ultimately lies in the hands of the organizers.

How?

Much like the decision for which stakeholders to involve, the decision for how to involve them allows many options. In the following we will outline common examples for stakeholder involvement. These should however not be perceived as exhaustive. Organizers should discuss options with stakeholders and decide for a model that fits their particular event.

One common way of involving stakeholders in a hackathon is as **sponsors**. This can include them providing resources in exchange for being mentioned on the hackathon website, in handouts or on posters at the hackathon site or them providing

specialized equipment or sponsoring awards or specific activities during a hackathon. The website of Major League Hacking provides a good overview on how to attract sponsors including different sponsorship options [39]. The advantage of this approach is that the outline and organization of a hackathon remains solely in the hands of the organizers while sponsors provide additional resources for the hackathon to take place.

Another common way of stakeholder involvement is to invite them as **speakers**. Similarly, stakeholders can also hold **training sessions** during an event e.g. related to specific technologies they are familiar with and that participants might use for their projects. Both approaches allow stakeholders to be present during an event and provide useful context and input for participants that they can utilize when planning and working on their projects. Moreover, it leaves the option for participants to decide whether to use the input for their projects or not.

Another common way of involving stakeholders in a hackathon is for them to serve as **mentors** (section 5.11) or **jurors**. Serving as mentors – in comparison to the aforementioned role as speakers or trainers – allows stakeholders to directly work with participants, provide targeted feedback and steer them into a specific direction [49] and foster learning [1]. Utilizing stakeholders as jurors can also be beneficial because they can provide realistic project assessments based on their area of expertise and again provide useful feedback to participants when discussing their verdict.

In addition to serving in the aforementioned roles, stakeholders can also **provide access to additional resources** in the form of datasets, documentation and access to interested parties such as potential future customers or domain experts [13]. This requires them to be accessible during the hackathon but it allows participants to seek input and advice on demand.

For some hackathons it might also be feasible for stakeholders to **propose specific challenges** for the participants to address [49]. We observed this model mainly in scientific hackathons where scientists proposed project areas or challenges that were related to their area of expertise or that would help them in their work. Participants can then choose which challenge or project to address and how to address it.

Finally, stakeholders can of course also attend an event as **participants** or serve as **co-organizers**. This is particularly common for corporate hackathons where organizers and participants often are employees of the same company that organizes the event [44]. Involving stakeholders as participants can however be difficult especially in an open event that welcomes individuals from various domains and backgrounds since stakeholders might be inclined to take over projects and adjust them to fit their ideas or goals.

Trade-offs

1. **Depth of stakeholder involvement:** The main difference between the aforementioned models of stakeholder participation is how much they can influence what happens during a hackathon. In some cases, it might be useful for stakeholders to be deeply embedded e.g. when a hackathon aims to solve specific issues within a certain domain such as the development of software artifacts that fit within an existing ecosystem. This might however limit interest by projected participants thus making it hard to attract and retain participants during a hackathon. Balancing these two sides can be difficult for organizers. One way of addressing this tradeoff is for organizers to allow stakeholders to provide input but limit their interaction with and active participation in participants projects.
2. **Open project selection vs selection among proposed challenges:** Most hackathons allow participants to work on any project they want. This approach can foster creativity and interest because it allows participants to work on any theme they are passionate about. It will however likely also lead to participants working on projects that might or might not be related to the goals (section 5.1) organizers had when organizing their event or projects that might not be useful for the domain the hackathon was organized in. Providing specific challenges ensures that participants work on projects that are relevant to individuals outside the context of the hackathon thus increasing the probability of projects to live on after the hackathon has ended (section 5.12). It does however limit choice for participants and thus might lead to limited interest and frustration. One way to address this tradeoff is for organizers to propose larger topic areas of themes that guide participants to a specific direction but allow for them to develop their own idea related to this direction.

Participant recruitment

Participant recruitment is one of the most crucial elements of hackathon design. After defining goals (section 5.1) and themes (section 5.2) for the hackathon, organizers should ask themselves: Who would be the target audience for an event? When should they start recruitment? How to draw interest and attention to an event? We will provide suggestions for those in the following.

When?

Participants need to be recruited and have to register before the hackathon. The exact time that recruitment needs to occur varies based on the scope of the event, the degree to which the target audience is known, and the amount of planning needed for potential participants to take part in a hackathon (i.e. location and time). As such the organizers should discuss who they would want to participate at the very beginning of the event organization before they develop or deploy any recruitment strategy. Organizers sometimes recruit participants up to a year before an event as e.g. in the case of events for specialized communities such as Astro Hack Week [65] or Geo Hack Week [69]. Other events such as collegiate hackathons sponsored by Major League Hacking typically start recruitment about two months before the actual event [40].

Who?

Taking their event goals (section 5.1) and themes (section 5.2) into consideration, the organizers should identify the characteristics of the target audience they aim to recruit. For some events with broad appeal, *college students who have taken a programming class* may be sufficiently specific. For others,

e.g. *PhD level astronomy students* the recruitment efforts will have to be very targeted and provide compelling motivation for that particular population. Stakeholders (section 5.4) that are connected to the targeted audience can support this recruitment process.

How?

There are two general strategies for participant recruitment: **open and closed**. The organizers should decide which strategy to use based on the goals of their hackathon. Open recruitment targets a wide range of participants with the aim to diversify participation. As such, open recruitment is typically used for hackathons whose main goal is to build a community around the cause or theme. The aforementioned hackathons Astro Hack Week [65] and Geo Hack Week [69] used open recruitment inviting anyone with an interest in astronomy and data science. Alternatively, the organizers could also use closed recruitment thus only inviting specific participants who e.g. are internal to a specific community. Examples for such hackathons are MicrosoftâĂŹs OneWeek Hackathon [44] and the STScI hack days [53] who only recruited among their employees. Many tech companies hold such events to foster innovation, to promote a more open and innovative culture, and to help create richer and farther-reaching social ties [55]. If the event is a community event, organizers need to identify and invite individuals who might be interested in the particular cause or theme that serves as the glue for the community. Hackathons mostly are recurring events that are sometimes held by groups, such as scientific research communities, who have continuous needs to train members, maintain or implement new features, or work on interoperability issues for shared tools. The HackWeek Toolkit provides detailed suggestions for defining a suitable audience and scope an event to the needs of specific communities [3].

After the target audience is identified, organizers should set up a **website which should contain basic information about the event** such as dates and venue, contact information to enable interested participants to communicate with the organizers, and a registration form. The process of setting up and publishing a website can be streamlined by e.g. creating a repository on GitHub as in the case of the World of Code (WoC) hackathon [78]. It is helpful for the organizers to decide on a particular hosting platform, e.g., GitHub, before the event. The organizers should then publicize the website to potential interested participants. When using a GitHub project URL organizers can also encourage interested participants to communicate with them via GitHub issues. In addition to providing information the organizers should also provide contact information and be accessible for potential participants via email and Slack.

The organizers should also create a **pre-event registration** form using e.g. Google Forms which should be accessible from the website. Through this registration form organizers can also collect additional information about participantsâĂŹ skills and background, preference of projects and areas, and goals and expectations if they decide to do so. Forms can also include open text boxes to allow participants to propose project ideas (section 5.8) or indicate if they are planning to come as a team (section 5.9).

Promoting and advertising an event is typically done by disseminating the previously discussed website to potentially interested individuals through various channels such as processional networks, mailing lists, student groups, university departments, personal networks, and social media such as twitter and Facebook groups depending on the target audience [40]. The organizers could also send direct email invitations to individuals who might be interested in taking part in the hackathon as participants or who might promote the event to potentially interested individuals.

While **open selection** of participants is often preferred because that it allows every registrant to take part in the event, some hackathons organizers decide to **carefully select participants** [4]. Reasons for selection might be constraints such as the maximum capacity of a venue, funding, etc. as well as to broaden participation from various communities. As described earlier, the choice of the strategy is very much dependent on the event. For example, if the goal of the event is to broaden participation in a specific software development community and most registrants are predominantly from a single institution or background, the organizers might want to consider deploying other strategies to broaden their reach. For hackathons with a learning goal, the organizers should a mix of more and less experienced participants. Likewise, hackathons that aim to foster entrepreneurship or the development of sustainable artifacts might want to attract participants from diverse backgrounds and expertise [48]. One approach that seems promising is organizing a mini-hackathon with participants from the community that they wish to attract prior to the main event. This approach has been successfully deployed by the She Innovates [75] hackathon. This is an all-women hackathon which aims to get women participants familiarized with the hacking process before they move on to events with more diverse participants.

Moreover, **participant recruitment and selection should start as early as possible**, at best right after the event theme and goals are formulated. This gives organizers more time to adjust their recruitment strategy when needed, increasing the chance to attract a sufficient number of participants that fit their desired profile. For large events, using an online tool can be helpful for the selection process. Entrofy [68] is an example for such a tool that allows organizers to extract a subset of registered participants based on certain attributes e.g. gender, career stage, etc., and given value (i.e. the percentage of distribution for each attribute).

Trade-offs

1. **Open vs selective recruitment:** An open recruitment strategy is advisable for hackathons that aim to facilitate networking among participants or hope that they would find new collaborators for future work. However, if the goal of the hackathon is to have a more concrete outcome e.g. creating a working prototype, or learning a specific tool, selective recruitment may help bring in participants most able to contribute or most able to benefit. Selective recruitment helps organizers to diversify participation or select a desired mix of skills and abilities that best serve the pur-

pose of the hackathon. This in turn helps create teams that have the skills and expertise to achieve their project goals which can then foster long-term project continuation [48]. For hackathons with selective recruitment, it is important to make the selection process as transparent as possible e.g. by letting potential participants know how the selection process works, how participants are selected if there are more eligible participants for each category, or which attributes are given greater weight than others, etc. This helps ensure that even if they are not selected, participants might feel the fairness of the selection process and might not feel discouraged to participate in similar events in the future.

2. **Open vs closed hackathon:** In a closed hackathon, only members of a particular organization are eligible to participate. This allows the organizers and participants to freely discuss and work on non-public topics such as new products, competitive strategy, and proprietary technologies. It also makes it easier for teams to coordinate their work, since members of the same organization generally share a culture, technical language, and role expectations. On the other hand, open participation allows a much broader mixture of people from different backgrounds, domains, and areas of expertise, which facilitates innovation, learning and community building.

Specialized preparation

Organizers might want to run a hackathon related to a specific theme (section 5.2), in a specific domain or utilize specific software and hardware during their event that are not commonly available to participants. Such events thus potentially require the organizers to provide trainings, access to licenses or hardware for participants to be able to work on projects during this hackathon.

When?

Preparation activities can be done remotely, onsite, or both. If the theme (section 5.2) and goals (section 5.1) will likely require specialized technical knowledge (e.g. particular tools, languages, or frameworks) or domain knowledge (e.g. community needs, or a scientific field) it is important to develop ways to bring participants up to speed before (usually 1 to 3 weeks) or very early during the event. The organizers may also want to facilitate team meetings if teams are formed in advance (section 5.9) so that they could discuss project scope and plan, assign tasks, and experiment with technologies to be used during the hackathon. Assuming participants have free time and sufficient motivation, this can help the work move along more quickly when the hackathon begins.

Who?

The organizers should work with mentors (sections 5.9 and 5.11, [49]) to coordinate training programs. Mentors will in fact often be running those programs, since they are typically chosen based on their expertise. For hackathons when team formation (section 5.9) occurs in advance, it is advisable that the team leaders are chosen, so they can organize team meetings. Organizers can also encourage projected participants to prepare for the hackathon by e.g. setting up a code base in advance [48] or study technologies that they might want to use for their project.

How?

The organizers first need to identify **what technologies and topics are necessary** for people to participate in their event and to what extent participants should know about them before coming to the event. One common way to help achieve this is for organizers to arrange training programs in which participants are taught specific technologies or domain knowledge that they would need to use at the event. These programs can consist of webinars developed by the organizers, or pointers to existing resources that are available, e.g. on Youtube, or the Coding Academy website [14] and that projected participants can use to prepare themselves [48].

The organizers should ensure that the tutorial materials are **accessible** to all participants. This typically includes posting them on the hackathon website, the collaboration platform through which the hackathon is organized, e.g. GitHub, or other document sharing tools, e.g. GoogleDrive.

Tutorials are often delivered as pre-recorded videos with interactive Q&A at scheduled times. If the training, for example, is related to the configuration of the development environment, participants can watch a pre-recorded video, replicate the steps shown in the video, and communicate with trainers during the Q&A session and/or via emails. Alternatively, tutorials can be delivered live by a mentor (section 5.11, [49]) which guides a group of participants through activities interactively. For groups of larger size, it might be advisable to form smaller subgroups of perhaps 5 to 10 participants, each guided by one mentor. In practice, live tutorials can present a scheduling challenge, as it might not always be possible to find a common time for all participants particularly when they are geographically distributed. Moreover, tutorials may also need to be customized based on the participantsâĂŹ skill levels, e.g. novices need foundational knowledge first before learning advanced skills while experienced participants might want to skip such basics (section 5.5). In such situations, it is advisable that organizers cluster participants into groups of similar skills and provide appropriate materials for each group. The World of Code (WoC) hackathon [77] is an example where mentors trained participants in small group tutorials in real-time a few days before the event via Skype.

For hackathons that wish to train participants onsite, we have observed two approaches that can be effective. The first approach is **train to hack** as done by Astro Hack Week [65] and Geo Hack Week [69]. In this approach, participants spend most of their time hacking, while also spending a considerable amount of time (e.g. 25-50%) on training of particular skills and domain knowledge required to conduct research work afterwards. The second approach is participants **alternating between hacking and training**. An example of the second approach is high performance computing (HPC) hackathons which run in parallel with the super computing conference [72] where participants distribute their time between hacking and attending conference sessions. BrainHack 2018 of the Organization for Human Brain Mapping (OHBM) in Singapore [74] is another example that allows participants to swap between

hacking and attending training sessions during a concurrent track.

In case the hackathon involves **specialized hardware**, the organizers might want to ensure that it arrives at the hackathon site early so that it can be set up before the participants arrived. Moreover, organizers might want to ensure that a specialist is on-site during the entire event that can help with technical issues.

Trade-offs

1. **Pre-recorded videos vs real-time training:** Pre-recorded training videos resemble a traditional mode of instruction that offers limited interaction between the participants and trainers. While questions can be addressed in a live session after the training, spontaneous adaptation of the training to attain better learning outcomes is not easily achievable. Adaptation and customization are possible in real-time interactive training, as human trainers are aware of the difficulties that participants are experiencing and can quickly act to mitigate such difficulties. The latter however is not always feasible for larger groups.

2. **Training before vs training at the hackathon:** Pre-event training permits more hack time as opposed to onsite training which requires participants to split their time between hacking and training. Pre-event training, however, demands participantsâĂŹ willingness to spend some time for training before the event, and that delivered in real-time settings demands both trainers and participants are concurrently presented which can present scheduling problems.

Duration / breaks

When organizing a hackathon, organizers have to decide when to start, when to end and when to take breaks in between. These decisions are crucial because they can influence who would be motivated to come, whether attendees can maintain a high level of motivation, how the event will be perceived and how participants engage with each other beyond working on their projects.

When?

The overall timeline of a hackathon needs to be decided on and announced early during the planning process since it serves as a basis for recruitment material (section 5.5) and for peoplesâĂŹ decision to attend the event. The overall timeline should include dates and times (including start, end and potential overnight breaks) for each day. Other breaks during the hackathon can potentially be decided on and announced later.

Who?

The decision for when a hackathon will take place, how long and it should be and how many breaks it will have is commonly taken by the organizers. For this decision they can consult projected participants (section 5.5), mentors (section 5.11, [49]) and other stakeholders (section 5.4). Including external stakeholders is especially advisable when the hackathon focuses on a specific theme (section 5.2), takes place in a specific domain or aims to attract participants (section 5.5) from backgrounds that the organizers are not particularly familiar with.

How?

When thinking about a hackathon most people will probably think about an event that starts on a Friday afternoon, ends on a Sunday, runs overnight and has little to no breaks in between with teams just tirelessly hacking away on their project [26]. While this is a common hackathon format it certainly is not the only one. Organizers can decide for their event to take place at any point during the week and have breaks overnight as well as during the day. When deciding about the timing of their particular event, organizers should take the following aspects into account.

They should consider the background of their **projected participants** (section 5.5). While it might be ok for students to participate in an event during the week and stay up overnight, this might not be possible for people that have fixed working times or busy family lives. Corporate events we studied often took place during regular working hours and participants could choose to go home for the night or stay and continue working [44], while civic events often take place in the evening and can be spread out over multiple weeks with breaks during hacking times to allow for participants to network [66].

Another aspect to consider when deciding for the duration of an event is the **context or domain** (section 5.2) an event takes place in. In a corporate setting it might be feasible to focus on regular working hours because relevant stakeholders that can e.g. serve as mentors or provide thematic input might only be available during certain times. These times can however be considerably different e.g. in a civic context where stakeholders may be more likely to be available after regular working hours.

It is also important for organizers to consider their **goals** (section 5.1) for organizing a hackathon when deciding about when to start, when to end and when to take breaks in between. If their goal is for teams to develop polished prototypes, they might want teams to focus on their project and thus not take too many breaks to not affect their productivity and rhythm. If the organizersâĂŹ goals should however be for participants to network, they might want to consider regular breaks during which participants can socialize.

Breaks can also serve as opportunities for organizers to convene and discuss with mentors (section 5.11, [49]) and stakeholders (section 5.4) and potentially alter the course of an event. For example, during a community hackathon we studied, the organizers took time during a break to sort ideas proposed by participants to structure the following team formation process (section 5.9, [78]).

Trade-offs

1. **Overnight vs breaks during the night:** The main advantage of organizing a hackathon that takes place overnight is that participants have more time to work on their projects. Working overnight can however take a toll in that productivity can be expected to drop during the night and the following morning. Breaks during the night limit the available working time but allow for participants to get some rest and engage with activities beyond the hackathon. To address this tradeoff organizers might consider providing the option

to work during the night by e.g. keeping the venue open but leave it to the participants whether they would like to take a break. To avoid participants feeling social pressure to work during the night, organizers could also emphasize that it might be helpful to take a break or organize an activity that could reasonably mark the end of the hacking day such as a dinner.

2. **Weekend vs during the week:** Organizing a hackathon during a weekend might make it more likely for people to participate since many projected participants can be expected to be busy during the week. It might not be advisable though to organize a hackathon for corporate employees during a weekend. Organizing a hackathon during the week might also provide access to stakeholders that will not be available during the weekend. To address this tradeoff, organizers could utilize a mixed format where the start of the hackathon is during the week and it ends during the weekend. This would allow participants to access stakeholders during the first crucial phases of a hackathon when ideas are formed.

3. **Short vs long:** Deciding on the overall duration of a hackathon can be difficult. An event needs to be long enough for participants to be able to make progress on their projects, but it should not drag on endlessly because participants might lose interest. To refocus interest organizers could ask stakeholders to give talks, provide examples or otherwise engage participants with the theme of the event. Generally, it is not advisable though to drag an event on for too long. One of the characteristics of a hackathon after all is that it takes place over a limited time span. An overall hacking time of about 48 hours divided over multiple days has proved to be a good rule of thumb.

4. **Time for work vs time for breaks:** Depending on the goals of the organizers it might be advisable to organize multiple breaks during each day for participants to be able to get away from hacking and socialize. Having many breaks will however cut into the time participants will have for their projects and might leave participants frustrated because they did not make sufficient progress. To address this tradeoff organizers could use breaks such as breakfast, lunch or dinner for participants to socialize.

Ideation

One of the main motivations for individuals to attend a hackathon is the prospect to work on an exciting project. It is thus crucial for organizers to think about how to support participants to come up with interesting and attainable project ideas they can work on during a hackathon. There is also evidence that some ideation approaches, such as traditional brainstorming, can help self-identified minorities feel more welcome and their ideas more accepted during the event [20].

When?

Ideation typically takes place before the hackathon, but organizers can also plan for a dedicated ideation session at the beginning of the event itself.

Who?

Participants typically propose their own ideas. Especially for ideation during the event, trained facilitators can help the participants generate ideas efficiently and harmoniously. It is also possible to guide ideation towards a certain direction that appears feasible and useful to organizers and / or connected stakeholders (section 5.4).

How?

The most common ideation approach is for participants to develop ideas that are related to the **theme(s) of a hackathon** (section 5.2). Hackathon themes are often intentionally broad covering areas such as civic technologies [11], environmental sustainability [71], entrepreneurship [47] and others to allow for a large variety of ideas to fit under their banner. Organizers and stakeholders can also decide to narrow the scope of potential ideas by proposing specific problems (areas) that participants should address. This approach is suitable for targeted events (section 5.1) that e.g. aim to develop technologies for a specific community [53]. It is important to leave space for participants to develop ideas that are of interest to them, to ensure their motivation to participate.

Ideation can take place **before or during a hackathon**. For larger audiences it might be advisable to collect ideas prior to an event using technologies such as Google Docs or GitHub issues. It is important to use technologies that projected participants are familiar with. Collecting ideas through such technologies not only allows participants to describe their ideas but also enables others to comment, provide feedback and express interest. They also allow organizers and stakeholders to pre-screen ideas, adjust their ideation approach and capture ideas for future use beyond the context of a particular hackathon (section 5.12).

Conducting a separate ideation session **at the beginning of a hackathon** using common approaches [16] such as brainstorming [30] might lead to more interaction between participants, organizers and mentors (section 5.11) and foster ideation. It also allows organizers to guide ideation by asking targeted questions [82] and clustering ideas e.g. based on participant interest. Ideation during a hackathon does takes away time for hacking though especially for larger audiences.

Collecting a **sufficient number of interesting ideas** is crucial for a successful hackathon because ideas usually are the basis for team formation (section 5.9). Collecting many interesting ideas is, however, not the only aspect to consider during ideation. While being challenging enough to be interesting for participants to attempt and potentially continue after an event [48], ideas should also be attainable. This means that they need to be doable during the short duration of a hackathon, that the team that attempts them has or can quickly attain the skills required to complete a project based on that idea, and that there are sufficient resources available at the hackathon, including, for example, specialized hardware, licenses, cloud resources, or others (section 5.6). Organizers, stakeholders and mentors can support teams to select suitable ideas and help them scope their project during the hackathon.

Finally, it is important to consider that **some ideas might be**

extremely popular while others do not draw much attention. If some ideas prove extremely popular, the idea can sometimes be split into parts, or several teams can be formed to pursue the same idea. If ideas are not popular at all it should be clear to all participants that they will not be attempted during the hackathon. Participants proposing ideas thus have to be prepared to let go of their idea and potentially join a different team and work on something else.

Trade-offs

1. **Priming vs open ideation:** Leaving ideation completely open and in the hands of the participants can lead to them to coming up with ideas that are not, or only marginally, related to the goal of the hackathon, or with ideas that are not doable due to other constraints imposed by the setup of the event (time, specialized resources, available skills, etc.). If the primary goal is just to have fun or some basic exposure to coding and its possibilities, this may be fine. On the other hand, imposing strict limitations on the ideation process by e.g. limiting participants to address specific challenges proposed by organizers or stakeholders can in turn negatively affect the motivation of participants to attend an event and take on the proposed challenges. To address this tradeoff, it is thus advisable to always leave room for participants to develop their own ideas even when proposing challenges.

2. **Individual ideation vs group ideation:** This tradeoff is common for most creativity techniques. Asking participants to develop ideas individually and share them after ideation has ended typically leads to more diverse ideas since people tend to follow the direction of ideas that have already been proposed. Some participants might however also benefit from others sharing their ideas because it can foster their imagination. One way of dealing with this tradeoff is to take a two-step approach by asking participants to submit individual ideas prior to the hackathon and then sharing them at the beginning of the event allowing other ideas to be added. Moreover, posing multiple (potentially contradicting) ideation themes might also help participants to come up with diverse ideas.

3. **Time for ideation vs time to hack:** Conducting the ideation at the beginning of or during a hackathon provides organizers with an opportunity to steer its direction, emphasize ideas that they perceive to be best related to their goals and allow participants to develop additional related ideas. It does however also cut into the time that remains for hacking. This tradeoff becomes more problematic for larger hackathons because each participant should have the chance to propose ideas to keep the morale up which might not be possible at larger events. For larger events it is advisable to move ideation online or to ask participants to send ideas before the event. Ideation prior to an event can also allow participants to familiarize themselves with the idea, create common ground and start learning about potentially required technologies [1, 50].

4. **Too large vs too small:** Ideas should be interesting and challenging but at the same time doable during the short duration of a hackathon. One approach to deal with this trade-off would be to let participants propose wild ideas first that can then be scaled down to doable projects through mentoring. In order to avoid mismatched expectations, it is important for participants to be gently encouraged to be realistic in what they can hope to accomplish during an event. Prototypes where only a few example features are implemented simply, and difficult technical challenges such as analyzing substantial data sets or developing APIs are simply faked. Such compromises are common and often necessary.

Team formation

Another important decision for organizers is selecting an appropriate strategy for selecting projects and forming teams. Teams are typically formed from the recruited participant pool (section 5.5), around projects of interest to them.

When?

Participant recruitment (section 5.5) and ideation (section 5.8) are prerequisites for the team formation process. Team formation and project selection can take place either **before the event** or **at the beginning of the event**. Each has its advantages and disadvantages, as we will discuss in the trade-offs below. Even if the intent is to choose teams and projects at the beginning of the event, organizers should expect that some participants may join the hackathon as a team, with firm ideas about what they want to work on, and with whom.

Who?

There are three roles involved in the team formation and project selection process. These roles are project **proposers**, **moderators**, and **joiners**. The proposer refers to someone who pitches a project idea at the event. This role can be taken by participants, organizers or stakeholders (section 5.4). The joiners are participants who selected the project they are interested in, sometimes also selecting a role that they would like to play at the event. For example, during MicrosoftâĂŹs OneWeek Hackathon [56], project proposers specified roles required for their proposed projects and other participants joined the project teams by taking one of these roles (cf. Microsoft HackBox [21]). The organizer or a dedicated person takes the role of moderator who facilitates the team formation process in order to configure project teams with skills, expertise, background, and reasonable size required to complete the projects they aim to work on. For hackathons at scale, it is important to assist the moderator with a tool that facilitates the matching of participants and projects.

How?

In order to successfully form teams with skills required to complete the projects proposed at a hackathon, it is often helpful to have a diverse participant pool (section 5.5). Organizers try to attract suitable participants as part of participant selection and recruitment process which has to be finished before team formation.

Teams can be formed either by **open selection**, **assignment**, or a **hybrid** strategy. In open selection, participants select projects and roles that they want to play based on their interest from the list of all available projects and roles. In the assignment strategy, a mediator assigns projects and roles to

participants. The hybrid strategy narrows down the participantâĂŹs search space by filtering out projects and roles that seem to be of lesser interest to participants or that they might be less qualified for. For assignment and hybrid strategy, it is important to gather participantsâĂŹ needs and expectations beforehand to optimize team formation. Information like that is typically collected through the registration process as part of participant recruitment.

Forming teams **before a hackathon** requires suitable online tools such as Google Docs, Google Forms or GitHub issues. Suitable tools need to support project listing and sign up. For example, in the STScI hack days [53] we observed that Google Forms were used to collect project preferences and skills. Based on this information, the organizers configured teams of 3 to 6 participants with a good mix of skills. Some hackathons, e.g. Steelhacks [76], suggest participants to form teams of 5. These tools work well for smaller events (say, 50 or fewer participants) but a more sophisticated tool would be required for larger scale such as MicrosoftâĂŹs OneWeek Hackathon. They deployed the online tool HackBox [21] that allowed participants to create projects proposal, sign up for projects and search for additional members with specific skills or interests.

It is common for teams to form **at the beginning of a hackathon** (section 5.10). This process needs to be fairly efficient so that teams will have sufficient time to actually work on their project. One common approach is to allow participants that have a project idea to pitch it in front of the other participants and write each idea down on a flip chart or whiteboard (section 5.8). The remaining participants are then given some time to walk around and chat with the project proposers and select a project they would like to work on. It is common to aim for teams of similar size between 3 and 6 members. It is particularly important for competitive hackathons (section 5.3) to have teams of similar size since a large difference between team sizes could lead to an unfair disadvantage for some teams. Moreover, large teams should be avoided because they typically require additional coordination effort which can limit the time a team has to actually work on their project. They thus need to potentially be split up, and ideas that do not draw much interest are generally abandoned. In another small-scale hackathon we observed, organizers asked the participants to rank proposed projects in order of their preference in Google Docs and participants were assigned to the project on a first-come-first-serve basis. This process might not be feasible for events at scale and using a tool like HackBox would be necessary even if team formation occurred at the hackathon.

For hackathons where teams are formed on site, it is sometimes desirable for organizers to propose projects, and either post descriptions in advance, develop brief pitches and make them available to participants as videos, or describe them at the beginning of an event. This approach is particularly useful when the goal (section 5.1) of a hackathon is, for example, to introduce newcomers to a particular domain, tool set, or scientific community. In these cases, it is very difficult for the projected participants to develop with feasible and appropriate project ideas themselves.

Trade-offs

In the following, we describe a number of trade-offs between various strategies used to configure teams in hackathons. It is important to note here that these trade-offs are not independent, and organizers should consider balancing them when making decisions about team formation and project selection.

1. **Forming teams before vs at a hackathon:** Pitching projects and even forming teams before an event can help to get the project work under way more quickly at the hackathon itself. However, there are some costs to this approach. Projects proposed before an event, even if there is an opportunity to pitch at the event itself, are likely to be chosen since participants have become familiar with them. They will not have the benefit of being discussed face to face with the potential for cross-fertilization and innovation this can provide. Moreover, if teams are chosen before an event, there is generally very little interaction, during the hackathon, among participants on different teams. If growing a community and forming broader social networks are important goals, it is generally advisable to pitch and discuss ideas at the hackathon itself.

2. **Proposing projects by participants vs by organizers:** Most hackathons allow participants to define their own project ideas, and this is often a primary motivation for participants, to do something fun and acquire skills they want. For some hackathons, however, participants are simply not in a good position to formulate projects that are feasible and appropriate for the theme of an event. They may lack technical skills, domain knowledge, or both. In these cases, it is desirable for organizers to define projects that will help the participants. This approach combines well with pre-event tutorials and dedicated mentors (section 5.11, [49]) because teams will likely need a lot of help to make progress. Since they are not pursuing their own passion, motivation for participation needs to be carefully considered though which may consist of things like valuable contacts for their future profession, potential job offers, or developing skills that are in demand. Recruiting materials should lay these benefits out convincingly.

3. **Open selection vs assignment:** Open selection of teams is common for hackathons organized around themes (section 5.2) (e.g. particular civic issues, making use of specific data sets, etc.) and for hackathons designed just for fun or, for exposure to programming or prototype development or entrepreneurship. Open selection is beneficial in the sense that it allows participants to choose what they want to work on in contrast to the strict assignment approach that does not consider participants' motivations, needs and expectations. However, there are some costs associated with open selection. For example, teams may not have members with skills required to complete a desired project. This can not only lead to frustrations during the hackathon but might also negatively affect the probability of project continuation after an event has ended [48]. A lack of diversity may also inhibit a teamâĂŹs ability to create innovative ideas and solutions and again affect project continuation after an event [48]. For this type of hackathons, it is often helpful to have

webinars and pointers to resources prior to the actual event. This put participants in a much better position to formulate realistic and on-target project ideas quickly. Another way to alleviate this trade-off is having a balanced - hybrid - approach which could provide participants with choices which are closely related to their goals and expectations.

4. **Large vs small teams:** Organizers should expect that even if they try to have teams with reasonable size, teams may be larger than the desirable size of 3 to 6 participants. Larger teams are more likely to encounter coordination problems compared to smaller teams. This problem might even be more significant in teams consisting of members who have not collaborated before because they do not have a common knowledge about each otherâĂŹs skills and working practices [56], which could lead to them not being able to generate the outcomes they want. Hackathons with open selection are more likely to suffer this problem as no moderation is applied to the team formation. To minimize this issue, the organizers should try to ensure teams have reasonable size regardless of the team formation strategy they use.

Agenda

Like any other event hackathons need an agenda that outlines which activities will take place at which point in time. The timing and outline of activities can profoundly affect the experience of participants. Organizers thus have to carefully plan which activities they want to conduct during a hackathon for it to be satisfying and engaging.

When?

The agenda should be available at least a few days prior to a hackathon to allow participants and other stakeholders (section 5.4) to familiarize themselves with it and make plans accordingly. Certain activities in the agenda such as organizing speakers and awards might require longer preparation periods and should thus be started earlier during the preparation.

Who?

Organizers typically consult with mentors (section 5.4, [49]) and other stakeholders (section 5.4) such as sponsors and domain experts to decide about which activities will take place during a hackathon. Their timing then is commonly decided by the organizers to create an organic flow during the event itself.

How?

Depending on the domain the hackathon takes place in (section 5.2), the goals (section 5.1) that organizers aim to reach or the type of participants they aim to attract (section 5.5), organizers might want to consider a variety of different activities during an event. In the following we will outline common examples for activities that organizers might want to consider. This list is however by no means complete. Organizers can and should be creative in developing specific activities that fit their particular event.

Hackathons typically **start with a brief welcoming address**. During this address the organizers welcome participants and lay out the organizational details of an event. These should include means of reaching organizers, mentors and other participants during a hackathon such as shared **communication channels, email lists or common document folders** as well as **links to useful resources** related to e.g. the theme of the event or to technologies that participants might use [79]. Organizers should also **introduce mentors** (section 5.4, [49]), **jury and judging criteria** (in the case of a competitive event (section 5.3)), explain **checkpoints** and discuss which **outcome** is expected from each team at the end of the hackathon. Such outcomes can include but are not limited to source code or other technical artifacts and presentations including videos and / or slides. The welcoming address can also include a **thematic keynote** e.g. by a sponsor that provides additional background for the hackathon and sets the tone for the remainder of the event.

Afterwards participants commonly pitch ideas (section 5.8) and form teams (section 5.9) before starting to work on their projects.

During the hackathon organizers commonly also schedule a series of **checkpoints** during which teams report their progress, discuss problems they are facing and outline their plans for the time ahead [73, 25]. These checkpoints should be evenly distributed along the timeline of a hackathon. It is e.g. typical to have checkpoints at the beginning and the end of each day. They provide a great opportunity for organizers and mentors to get an overview of each teamâĂŹs progress and decide which team might need additional support. Checkpoints can be organized in different ways. Some organizers may only ask team leaders to present to organizers and mentors to not break the teamsâĂŹ rhythm. Others prefer all participants to be present during each checkpoint so that teams can share experiences and learn from each other.

Organizers can also schedule additional **talks or training sessions** during an event [25, 1]. These can be related to using common or specialized technologies (section 5.6) that participants might use, provide additional domain background, or teach participants specific skills such as how to successfully pitch their project at the end of the hackathon. Such talks or training sessions can take place once or multiple times as part of the main event. Some organizers even run them as parallel tracks over the entire duration of the hackathon [74]. Such talks should be closely related to the projects that teams are working on during an event to have the desired effect [1].

Depending on the goal of an event, organizers might also organize **social activities** that require participants to interact with each other beyond the teams they work in. These can include short games where participants have to form teams that are different from those they work with during the hackathon and compete for small prizes [35]. Such games can also ease the tension of a hackathon and force participants to move around which can help reduce stress and emphasize the fun aspect of a hackathon. They are particularly useful for hackathons that emphasize networking as a goal. They can however also be frustrating because they can break the rhythm of participants and distract them from their projects [83].

At the end of a hackathon it is common for teams to **present**

their project to the other teams, organizers, mentors and jury (in the case of a competitive event). These presentations can take different forms depending on the outcome outlined at the beginning of the hackathon. They can be organized in the form of pitches (as common in entrepreneurial events), demos (as common in collegiate events) or project presentations (as common in civic and corporate events). In a competitive event these presentations are then followed by a deliberation of the jury and the award ceremony.

It is generally **not advisable to plan too many activities** during a hackathon because all of them will reduce the time teams have to work on their projects (section 5.7) which after all will be one of the main reasons for people to attend a hackathon. It is advisable though to conduct a thorough opening address as outlined before, schedule at least one checkpoint per day and hold final presentations so that all teams can show what they had been working on. The other outlined activities are optional, and organizers need to decide which ones they consider useful for their specific event.

Trade-offs

1. **Input and trainings vs social activities:** Organizers might be inclined to provide as much input to participants as possible especially during a hackathon that is attended by participants who are not necessarily very familiar with the theme of the event. Providing too much input during a hackathon can however confuse and frustrate participants because it breaks their rhythm, and they might feel inclined to change their project idea repeatedly based on the input they received. Moreover, some hackathon organizers might organize social activities for participants to network rather than work on their projects all the time. Striking a suitable balance here is crucial for a successful event. One possible way to mitigate this trade-off could be to have a thematic keynote at the beginning of an event, provide additional resources for participants to refer to during an event and stagger social activities around common breaks such as breakfast, lunch and dinner.

2. **Repeat activities vs single activities:** It can be advisable to run the same talks or training sessions multiple times during an event to allow participants to attend them at the point in time that fits them best. This can however be difficult to organize since it requires presenters and trainers to be available during the entire duration of a hackathon. It might also be advisable to focus input at the beginning of an event so that participants can take maximum advantage of it. In addition, organizers can provide access to useful resources (e.g., instructional videos) that participants can access when needed.

3. **Voluntary vs mandatory checkpoints:** Checkpoints are a great way for organizers and mentors to assess the progress for each team and provide targeted support if necessary. Attending checkpoints can however be tedious and time-consuming for teams and affect their productivity, so they might not be particularly inclined to attend. One way to deal with this tradeoff is to assign mentors to one team or a group of teams (section 5.11, [49]) and ask them to engage with their teams on a regular basis. This allows them to detect issues, inform the organizers, discuss strategies and provide targeted support. This does, however, lose the advantage of familiarizing the team members with the projects and people on other teams.

Mentoring

Mentors are the first substantial point of contact for participating teams. They provide feedback, help them when they have problems and guide them through the hackathon process. Deciding on who to recruit as a mentor and developing a suitable mentoring strategy are thus crucial decisions for every hackathon organizer.

When?

It is important to develop a mentoring strategy and recruit suitable mentors prior to a hackathon. Since they are likely to be busy people and mentoring generally takes a substantial chunk of their time, recruiting weeks or months in advance is desirable. Mentoring itself typically takes place either over the entire duration of a hackathon or at specific points during the event. It can also continue after a hackathon has ended (section 5.12).

Who?

Mentoring requires the collaboration of organizers, mentors and participants. Organizers create a mentoring strategy, recruit mentors and support them to execute the developed strategy during and after a hackathon. Mentors support participating teams based on this strategy. The time commitment asked of the mentors should be made very clear. For example, are they expected to help participants before and/or after the event itself? Are they expected to stay for the entire event, work in shifts, or just be available at checkpoints (section 5.10)?

How?

Prior to a hackathon, organizers have to develop a mentoring strategy and recruit suitable individuals as mentors.

Mentoring strategy: The most common strategy is for mentors to provide individual **on-demand support** during a hackathon based on the mentorâĂŹs expertise. This is appropriate when the participants are generating their own projects (section 5.8), and have the basic skills required to complete them. On-demand mentors typically circulate among teams and/or staff a help desk location where participants can receive assistance when needed. In addition, organizers often set specific checkpoints during which mentors engage with teams, ask for their current progress and provide targeted feedback. Alternatively, the event may have **dedicated mentors** that are assigned to an individual team [49]. This is useful when the participants have significant skill deficiencies, or donâĂŹt have sufficient domain knowledge (section 5.6) to define projects that fit within the hackathon theme (e.g., scientific hackathons aimed at bringing neophytes into a field). For either strategy, it is crucial that mentors are accessible to teams when they need them.

If possible, **in-person mentoring** is highly desirable, although it is possible to mentor, through technical means like Slack, Zoom, Google Meet or other conferencing and messaging

platforms. **Online mentoring** can be more difficult than anticipated because of the limitations of tools, hardware, and the difficulty of mentors being fully aware of the team context. If using a platform like Slack, for example, it can be difficult, time-consuming, and frustrating to describe problems and solutions in text. For conferencing tools like Zoom, it may be difficult for all team members to see a mentor or share the mentorâĂŹs screen on the small screen of a laptop. Moreover, a remote mentor will not have the ability to easily see the confusion, skill deficiencies, or frustration that teams may be experiencing. We thus strongly encourage in-person mentoring whenever possible.

In addition, organizers might want to create mandatory **checkpoints** (section 5.10) during which teams present their progress to mentors and to the other participating teams. Such checkpoints allow for mentors and teams to detect deficiencies that might have remained unnoticed and provide an opportunity for broad feedback to all teams at once.

Another aspect to consider is whether to have **individual mentors** supporting participants or to form **mentor teams** with diverse backgrounds and expertise. Individual mentors allow for a more flexible deployment while mentor teams can provide holistic feedback to participating teams on a broad range of issues. Mentor teams also provide opportunities for less experienced mentors to learn from their more experienced peers. In the case that organizers decided for mentoring teams, it is important to define them prior to or at the beginning of the event.

Organizers also have to decide **how many mentors** to recruit for their hackathon, and how many of them to deploy at specific points during the hackathon. This decision depends on the number of participating teams, the availability of mentors and other recruitment related aspects we will discuss in the following. It also depends on the time of a hackathon since mentor support is mostly needed during the early and late phases of an event. As a rule of thumb using a minimal mentor to participant ratio of 1 to 10 is feasible since teams commonly have fewer than 10 participants.

Recruitment: Depending on the strategy, the organizers have to decide how to recruit suitable mentors. Common aspects for recruitment are the expertise of individuals related to the theme or domain of the event, their technical proficiency related to the technologies that participants might use during a hackathon, their prior hackathon (mentoring) experience and their ability to guide teams and support them to perform to the best of their abilities. A good mentor [33, 60] thus possesses a combination of domain, technical, project management and social skills. To recruit suitable individuals, organizers also have to think about potential benefits for mentors since they will invest a lot of time and effort into mentoring. For example, mentors can sometimes be drawn from tech companies who are hoping to find new recruits among the participants, or from faculty or postdocs looking for talented students.

After recruiting suitable individuals, organizers need to provide them with suggestions on how to mentor [24] teams based on the previously decided mentoring strategy. This includes potential activities before a hackathon such as training webinars as well as their availability during the event either on site or online. It is crucial that mentors are available to teams when they need them because the tight time constraints of a hackathon do not allow teams to get stuck for long. Mentors will be particularly busy during the early and late phases of a hackathon. During the early phases, teams commonly need help scoping their project and technical support to get started. During the late phases, everyone is scrambling to fix last minute issues which can also lead to increased mentor demand.

Mentors need to be introduced to the participants either before or at the beginning of a hackathon. This introduction should include how and when participants can engage with mentors and which mentor can help them with specific topics or issues. It is important to remind mentors that a hackathon is not for them to push their own ideas. It is about helping teams to run their project their way.

In some cases, mentoring can also continue after a hackathon has ended (section 5.12) to e.g. facilitate the continuity of learning or complete the development and integration of a technical artifact. This depends however on the mutual interest of participants, mentors and organizers and should be discussed at best before the end of a hackathon.

Trade-offs

1. **Dedicated mentors vs on-demand mentors:** On-demand mentors - if sufficient in number and covering all the necessary skills - can quickly address the needs of any team while dedicated mentors can build a relationship with a team and be more effective and efficient supporting them and helping them define a project and acquire skills they need. Supplying a larger hackathon with individual team mentors can however prove to be challenging. Moreover, dedicated mentors might be inclined to take over certain aspects of a project which might negatively affect a teamsâĂŹ motivation and give them an unfair advantage especially during a competitive hackathon (section 5.3).

2. **Mentor teams vs individual mentors:** One benefit of mentoring teams is that participants can get comprehensive support related to multiple aspects of their project (domain, technical advice, scoping, etc.) [1]. They are thus particularly useful when using checkpoints because these allow for mentors to take time to address multiple potential issues the team is facing at the same time. Mentor teams also allow less experienced mentors to gain more experience while working with more experienced peers. These benefits however only materialize when the mentors in a team have different expertise and experiences and forming teams requires additional coordination effort by the organizers. Individual mentors on the other hand can flexibly offer targeted advice for teams in need. This does however require teams to know who they should address for specific topics. Individual mentoring also makes better use of each individual mentorâĂŹs time but limit support to the expertise of one mentor at a time only.

3. **Mentor background:** Each mentor should at best be an experienced project manager and domain expert with years of technical experience in the field, lots of hackathons under her/his belt and the ability to solve any problem a team might have. Since this is not always possible, it is important that organizers carefully select mentors with complementing backgrounds and skills. Moreover, participants and mentors need to be aware of the skills of other mentors, to be able to refer participants to suitable mentors if needed. This can be achieved by individually introducing mentors at the beginning of a hackathon, creating short online profile or by e.g. creating colored badges that indicate which kind of support each individual mentor can provide. Moreover, mentoring teams can mitigate this issue if they are formed as discussed before.

4. **Participant to mentor ratio:** At first glance it appears that more mentors are always better since more mentors can support more teams. This is however not true in all circumstances since e.g. having many mentors that can help with domain related questions and none that can help with technical problems is not desirable. Moreover, more mentors create more organizational overhead for organizers and might result in conflicting messages to participants since different mentors might provide different advice to teams based on their personal experience and background. Starting with a mentor to participant ratio of 1 to 10 can serve as a rule of thumb. It is however important for organizers still ask themselves is which expertise might be required by participating teams at different points during a hackathon. At the beginning of an event, participants will mostly require help related to scoping their project while later the required support will shift stronger towards domain and technology related questions. Inexperienced teams may also need help setting up their technical environment and tools. When deciding for mentors it is thus important to consider which expertise needs to be available to participants at which point during an event. Moreover, mentoring can also be streamlined by using the previously discussed checkpoints.

5. **Strict guidance vs mentors decide how to engage:** While there are general mentoring [33] and hackathon mentoring [60, 24] guidelines available online, it is important to note that each hackathon is slightly different with different goals, organizers, participants and mentoring requirements. Given this highly context-dependent nature of hackathons, it might be helpful under certain circumstances to advise mentors about when and how to engage with their teams. Under other circumstances organizers might also just let mentors engage with teams at any point in any way they want. Both extremes are not feasible. Providing strict guidance would limit the ability âĂŞ in particular of experienced mentors âĂŞ to provide useful support. Providing no guidance might affect the overall structure of a hackathon because mentors might e.g. be inclined to contact their teams all the time thus affecting their rhythm [1]. A hackathon is an intensive event during which a lot of things happen over a short period of time and mentors are crucial for an event to be successful. Spreading information about who is responsible for and knowledgeable about which topic and setting fixed checkpoints to reel everyone back in can thus help to mitigate this tradeoff. In general, however, the more competitive the event the more it is necessary, in the name of fairness, to provide relatively strict guidance and limits on what sort of assistance mentors should provide.

Continuity planning

Organizers might want to run a hackathon just for everyone to have a good time. They might however also want to organize one for a purpose that extends beyond the conclusion of the hackathon, for example to kickstart a community, teach participants about new technologies or create innovative products and services that will actually be brought to market (section 5.1). Continuity does not come for free though. It needs to be an integral part of the hackathon planning process.

When?

Continuity planning needs to start prior to the hackathon and continues past the end of the event itself. In order to support the continuity of hackathon outcomes, it is important that the event is embedded into a larger strategy.

Who?

Organizers are responsible for developing and deploying a suitable strategy which includes communicating it to the participants of a hackathon. The planning and execution of that strategy needs support by stakeholders (section 5.4) and hackathon participants. Organizers should also be aware that only a few or even none of the participants might share their continuity vision. Participants might just come for the fun, or they might have continuity plans of their own.

How?

Before planning for continuity, organizers and potential stakeholders need to think about **what the outcome of a hackathon should be** and how the continuation of this outcome can be supported. When thinking about hackathon outcomes most people will think of hackathon projects that get turned into startups [80]. The transformation of projects into startups is however not the only potential outcome worthy of being continued. Participants and/or organizers might also aim to extend existing products or services, foster community growth or spread knowledge about certain domains and technologies. Each of these outcomes might require a different continuation strategy.

Based on the decision of which outcome should be continued, hackathon organizers can then focus on **involving potential stakeholders**. Such stakeholders can e.g. be companies if the continuation goal is to develop a product or service, or communities with related or complementary interests if the goal is to start or grow a community. Stakeholders are vital for continuity planning since they can provide background and domain knowledge for hackathon projects, support participants to scope their projects, connect participants to key players after a hackathon has ended, and provide access to learning materials. It is also important for organizers to set suitable expectations for stakeholders and participants since there is only so much that can be achieved during the short timeframe of a hackathon. This makes it all the more important to plan

for what can be done before, during (section 5.10) and after an event to support continuation if that is the organizersâĂŹ goal. Examples for how organizers can support continuation is to suggest for participants to scope their project (section 5.8) before and start learning about technologies they might want to use (section 5.6, [56, 50]).

During a hackathon organizers and stakeholders should **provide an environment for participants that fosters the desired outcome**. If an important goal is for participants to have the opportunity to establish lasting social bonds, the organizers should put an emphasis on activities that allow for them to not only work on their projects in their teams but also to get in contact with other participants. This could happen in the form of games or other social activities. For participants to develop a project that can be continued afterwards, they should encourage participants to form a diverse team that has the skills required to complete that project [49], work on a strategy on how to spread the word about their project after the hackathon [48] or to ensure that what they develop can be easily integrated into an existing code base [50]. Offering prizes at an event (section 5.3) can also be an incentive for teams to continue their projects but they only have a short-term effect [48]. If the goal is to sustain participantsâĂŹ learning, e.g. about technologies they used during the hackathon, it might be useful to propose follow-up projects or connect them to other individuals that aim to learn about the same technology.

Participants and organizers might have different continuation goals after an event [42]. Most participants might not even be interested to continue working on their project after a hackathon or their interest might fade quickly [48]. Since continuation requires extra work from participants, it is important to identify and provide support to those who are interested in continuation. One approach to achieve this is simply to ask participants if they are interested in continuation and based on their response provide support and guidance. This support can âĂŞ depending on the planned outcome âĂŞ come in the form of startup funding, connecting participants to relevant parties that can support them, help them find resources, or simply contacting them from time to time after the hackathon to see what happened and provide targeted support if needed.

Trade-offs

In the following we will discuss strategies to support the continuity of different hackathon outcomes. From the descriptions it should be clear that some outcomes might require approaches and strategies that can negatively affect other outcomes. For example, organizing social gatherings during a hackathon might foster participant networking thus potentially contributing to connection continuity. Such gatherings however eat into the participantsâĂŹ time to work which can negatively affect their project outcome thus potentially jeopardizing its continuation after the hackathon.

1. **Project continuity:** The development of useful artifacts that continue to be developed or that get used after a hackathon is among the most common continuity goals. This requires âĂŞ as discussed before âĂŞ a strong focus on the project as such. It is advisable for teams to meet (section 5.9) and refine their project idea (section 5.6) before the event, validate it with potential stakeholders and then use the hackathon to develop it to a stage that it can be shown to stakeholders [50]. This means that the hackathon essentially serves the purpose of the team focusing on the development of a presentable prototype for their project. It also means that participants should be encouraged to find team members with diverse skill sets that fit the requirements of the project [48] and that team members choose which aspect of a project they work on predominantly based on their current skills and not on what they want to learn about.
2. **Connection continuity:** To support connection continuity it is important to ensure that participants can stay in touch after a hackathon. Technical means such as a shared Slack channel, a shared Google Drive folder or similar tools can be already used leading up to and during a hackathon. These tools enable participants to quickly get on board and start communicating and sharing information about themselves and their projects while the event is still going on. Organizers should also consider creating opportunities for participants to engage with other like-minded participants outside their project teams during and after the hackathon to foster continuous engagement after the hackathon has ended (section 5.10). None of these will work, however, unless the participants are substantially motivated to stay in touch, perhaps to act as sounding board or social support, to network opportunities such as jobs, or to provide each other with needed expertise in various domains. It might also be worthwhile to introduce teams to each other that worked on similar projects which serve as a proxy for the connection continuity.
3. **Learning continuity:** To foster learning continuity it is not only important to provide learning material before, during and after an event (section 5.6) as well as targeted talks and mentoring during the event itself [1]. It is also important to create continuing interests, e.g. through challenges during and after a hackathon. Moreover, it is important to ensure that participants think about their projects early [1] and choose them based on what they want to learn rather than what they already know.

FUTURE WORK

The version of the planning kit in this report represents our current state of knowledge, presented in a usable form. We will continue improving it based on new insights from our ongoing research work as well as experience and research reported in literature. We are also currently experiencing a surge of hackathon events organized online due to the world-wide pandemic. While still partly applicable, the suggestions provided in this planning kit website are based on our experiences related to collocated events. Organizing online hackathons will certainly require additional considerations that are not reflected here yet. We are currently studying these new and upcoming online events and we will update the hackathon planning kit based on our findings as soon as possible.

ACKNOWLEDGMENTS

The authors gratefully acknowledge support by the Alfred P. Sloan foundation, Microsoft Research, Microsoft Garage, the

Science Gateways Community Institute, the Space Telescope Science Institute, Omnibond and Garage48. We would also like to thank Christian Bird, Amy Cannon, Kevin Ellet, Linda Bailey Hayden, Timothy Holston, Sudhakar Pamidighantam, Rajesh Kalyanam, Audris Mockus, Je'aime Powell, Steve Scallen, Kathryn Traxler, Nancy Wilkins-Diehr, Boyd Wilson, Mona Wong and the organizers, mentors and participants of the various hackathons we studied, co-organized and supported.

REFERENCES

[1] Abasi-Amefon Obot Affia, Alexander Nolte, and Raimundas Matulevičius. 2020. Developing and Evaluating a Hackathon Approach to Foster Security Learning. In *Collaboration Technologies and Social Computing*. Springer.

[2] Pantelis Angelidis, Leslie Berman, Maria de la Luz Casas-Perez, Leo Anthony Celi, George E Dafoulas, Alon Dagan, Braiam Escobar, Diego M Lopez, Julieta Noguez, Juan Sebastian Osorio-Valencia, and others. 2016. The hackathon model to spur innovation around global mHealth. *Journal of medical engineering & technology* 40, 7-8 (2016), 392–399.

[3] Arendt, Anthony and Huppenkothen, Daniela. 2020a. HackWeek Toolkit. `https://uwescience.github.io/HackWeek-Toolkit/` [Online; accessed 12-August-2020].

[4] Arendt, Anthony and Huppenkothen, Daniela. 2020b. HackWeek Toolkit - Target Audience and Scoping to Specific Communities. `https://uwescience.github.io/HackWeek-Toolkit/#Objectives/Objectives-and-Goals/#target-audience-and-scoping-to-specific-communities` [Online; accessed 12-August-2020].

[5] Bastiaan Baccarne, Peter Mechant, Dimitri Schuurma, Lieven De Marez, and Pieter Colpaert. 2014. Urban socio-technical innovations with and by citizens. *Interdisciplinary Studies Journal* 3, 4 (2014), 143.

[6] Eric Berger. 2017. Karachi Hackathon Takes on Emergency Medicine Challenges: Solutions Pitched for Resource-Poor Environments. *Annals of Emergency Medicine* 69, 3 (2017), A17–A20.

[7] Gerard Briscoe. 2014. Digital innovation: The hackathon phenomenon. *Creativeworks London* 6 (2014), 1–13.

[8] Irene-Angelica Chounta, Sven Manske, and Ulrich Hoppe. 2017. "From Making to Learning": Introducing Dev Camps as an Educational Paradigm for Re-inventing Project-based Learning. *International Journal of Educational Technology in Higher Education* 14 (2017), 18.

[9] David Cobham, Bruce Hargrave, Kevin Jacques, Carl Gowan, Jack Laurel, Scott Ringham, and others. 2017a. From hackathon to student enterprise: an evaluation of creating successful and sustainable student entrepreneurial activity initiated by a university hackathon. In *9th annual International Conference on Education and New Learning Technologies*. EDULEARN.

[10] David Cobham, Kevin Jacques, Carl Gowan, Jack Laurel, Scott Ringham, and others. 2017b. From appfest to entrepreneurs: using a hackathon event to seed a university student-led enterprise. In *11th annual International Technology, Education and Development Conference*.

[11] Code for America. 2019. National Day of Civic Hacking. `https://www.codeforamerica.org/events/national-day-of-civic-hacking-2019` [Online; accessed 12-August-2020].

[12] Code for America. 2020. OrganizerâĂŹs Playbook. `https://brigade.codeforamerica.org/resources` [Online; accessed 12-August-2020].

[13] Code for Pittsburgh. 2020. Code for Pittsburgh Meetups. `https://www.meetup.com/codeforpgh/events/268968230/` [Online; accessed 12-August-2020].

[14] Codecademy. 2020. Learn Python 3. `https://www.codecademy.com/learn/learn-python-3` [Online; accessed 12-August-2020].

[15] R Cameron Craddock, Daniel S Margulies, Pierre Bellec, B Nolan Nichols, Sarael Alcauter, Fernando A Barrios, Yves Burnod, Christopher J Cannistraci, Julien Cohen-Adad, Benjamin De Leener, and others. 2016. Brainhack: a collaborative workshop for the open neuroscience community. *GigaScience* 5, 1 (2016), 16.

[16] Mihaly Csikszentmihalyi. 1996. *Flow and the psychology of discovery and invention*. Vol. 56. New York: Harper Collins.

[17] Database Center for Life Science. 2008. BioHackathon 2008. `http://hackathon.dbcls.jp/` [Online; accessed 12-August-2020].

[18] Jeanette Falk Olesen and Kim Halskov. 2020. 10 Years of Research With and On Hackathons. In *Proceedings of the 2020 ACM on Designing Interactive Systems Conference*. 1073–1088.

[19] Anna Filippova and Erik Trainer. 2017. Technical Report for the 1st Workshop on Hacking and Making at Time-Bounded Events. *Technical Report CMU-ISR, CMU-ISR-17-104* (2017).

[20] 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 *2017 IEEE/ACM 39th International Conference on Software Engineering (ICSE)*. IEEE, 152–163.

[21] Formidable Labs, Inc. 2020. Hackbox. `https://formidable.com/work/hackbox/` [Online; accessed 12-August-2020].

[22] Allan Fowler. 2016. Informal stem learning in game jams, hackathons and game creation events. In *Proceedings of the International Conference on Game Jams, Hackathons, and Game Creation Events*. ACM, 38–41.

[23] Kiev Gama, Breno Alencar, Filipe Calegario, André Neves, and Pedro Alessio. 2018. A Hackathon Methodology for Undergraduate Course Projects. In *2018 IEEE Frontiers in Education Conference (FIE)*. IEEE, 1–9.

[24] Garage48. 2017. How to mentor teams at a hackathon. https://garage48.org/blog/how-to-mentor-teams-at-a-hackathon [Online; accessed 12-August-2020].

[25] Garage48. 2019. Garage48 Cyber Security 2019. http://garage48.org/events/garage48-cyber-security-2019 [Online; accessed 12-August-2020].

[26] Garage48. 2020. How does it work? http://garage48.org/how-it-works [Online; accessed 12-August-2020].

[27] Gottfried, Jon. 2014. Are Hackathon Prizes the Worst Thing Since Moldy Sliced Bread? https://guide.mlh.io/digital-hackathons/judging-and-submissions [Online; accessed 12-August-2020].

[28] Sarah Hartmann, Agnes Mainka, and Wolfgang G Stock. 2018. Innovation Contests: How to Engage Citizens in Solving Urban Problems? In *Enhancing Knowledge Discovery and Innovation in the Digital Era*. IGI Global, 254–273.

[29] Scott Henderson. 2015. Getting the most out of hackathons for social good. *Volunteer Engagement 2.0: Ideas and insights changing the world* (2015), 182–194.

[30] Heston, Klare. 2020. How to Brainstorm. https://www.wikihow.com/Brainstorm [Online; accessed 12-August-2020].

[31] Chinh Hoang, John Liu, Zubaid Bokhari, and Allen Chan. 2016. IBM 2016 community hackathon. In *Proceedings of the 26th Annual International Conference on Computer Science and Software Engineering*. IBM Corp., 331–332.

[32] Daniela Huppenkothen, Anthony Arendt, David W Hogg, Karthik Ram, Jacob T VanderPlas, and Ariel Rokem. 2018. Hack weeks as a model for data science education and collaboration. *Proceedings of the National Academy of Sciences* 115, 36 (2018), 8872–8877.

[33] Kerpen, Carrie. 2018. If You Want To Be A Great Mentor Do These 5 Things. https://www.forbes.com/sites/carriekerpen/2018/06/18/5-things-great-mentors-do/ [Online; accessed 12-August-2020].

[34] Marko Komssi, Danielle Pichlis, Mikko Raatikainen, Klas Kindström, and Janne Järvinen. 2015. What are Hackathons for? *IEEE Software* 32, 5 (2015), 60–67.

[35] Li, Lori. 2019. 11 Icebreaker Games for Work That Your Team Will Love. https://www.tinypulse.com/blog/sk-work-icebreaker-games [Online; accessed 12-August-2020].

[36] Thomas James Lodato and Carl DiSalvo. 2016. Issue-oriented hackathons as material participation. *New Media & Society* 18, 4 (2016), 539–557.

[37] Major League Hacking. 2019a. Judging & Submissions. https://guide.mlh.io/digital-hackathons/judging-and-submissions [Online; accessed 12-August-2020].

[38] Major League Hacking. 2019b. Swags & Prizes. https://guide.mlh.io/digital-hackathons/event-logistics/ordering-swags-and-prizes [Online; accessed 12-August-2020].

[39] Major League Hacking. 2020a. Getting Sponsorship. https://guide.mlh.io/digital-hackathons/getting-sponsorship [Online; accessed 12-August-2020].

[40] Major League Hacking. 2020b. How to Promote Your Event. https://guide.mlh.io/digital-hackathons/marketing-your-event/how-to-promote-your-event [Online; accessed 12-August-2020].

[41] Major League Hacking. 2020c. The MLH Hackathon Organizer Guide. https://guide.mlh.io/ [Online; accessed 12-August-2020].

[42] Maria Angelica Medina Angarita and Alexander Nolte. 2019. Does it matter why we hack? - Exploring the impact of goal alignment in hackathons. In *Proceedings of 17th European Conference on Computer-Supported Cooperative Work*. European Society for Socially Embedded Technologies (EUSSET).

[43] Maria Angelica Medina Angarita and Alexander Nolte. 2020. What do we know about hackathon outcomes and how to support them? - A systematic literature review. In *Collaboration Technologies and Social Computing*. Springer.

[44] Microsoft Garage. 2017. oneweek hackathon 2017. https://news.microsoft.com/life/one-week-microsoftlife/ [Online; accessed 12-August-2020].

[45] Steffen Möller, Enis Afgan, Michael Banck, Raoul JP Bonnal, Timothy Booth, John Chilton, Peter JA Cock, Markus Gumbel, Nomi Harris, Richard Holland, and others. 2014. Community-driven development for computational biology at Sprints, Hackathons and Codefests. *BMC bioinformatics* 15, 14 (2014), S7.

[46] Arnab Nandi and Meris Mandernach. 2016. Hackathons as an informal learning platform. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*. ACM, 346–351.

[47] Alexander Nolte. 2019. Touched by the Hackathon: a study on the connection between Hackathon participants and start-up founders. In *Proceedings of the 2nd ACM SIGSOFT International Workshop on Software-Intensive Business: Start-ups, Platforms, and Ecosystems*. 31–36.

[48] Alexander Nolte, Irene-Angelica Chounta, and James D Herbsleb. 2020a. What Happens to All These Hackathon Projects? - Identifying Factors to Promote Hackathon Project Continuation. *Proceedings of the ACM on Human-Computer Interaction* 4, CSCW2 (2020), 1–26.

[49] Alexander Nolte, Linda Bailey Hayden, and James D Herbsleb. 2020b. How to Support Newcomers in Scientific Hackathons - An Action Research Study on Expert Mentoring. *Proceedings of the ACM on Human-Computer Interaction* 4, CSCW1 (2020), 1–23.

[50] Alexander Nolte, Ei Pa Pa Pe-Than, Anna Filippova, Christian Bird, Steve Scallen, and James D Herbsleb. 2018. You Hacked and Now What? -Exploring Outcomes of a Corporate Hackathon. *Proceedings of the ACM on Human-Computer Interaction* 2, CSCW (2018), 1–23.

[51] Open Bioinformatics foundation. 2018. OBF hackathons. `https://www.open-bio.org/` [Online; accessed 12-August-2020].

[52] Ei Pa Pa Pe Than, James Herbsleb, Alexander Nolte, Elizabeth Gerber, Brittany Fiore-Gartland, Brad Chapman, Aurelia Moser, and Nancy Wilkins-Diehr. 2018. The 2nd workshop on hacking and making at time-bounded events: Current trends and next steps in research and event design. In *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–8.

[53] Ei Pa Pa Pe-Than and James D Herbsleb. 2019. Understanding Hackathons for Science: Collaboration, Affordances, and Outcomes. In *International Conference on Information*. Springer, 27–37.

[54] Ei Pa Pa Pe-Than and Alexander Nolte. 2019. The 2nd Workshop on Hacking and Making at Time-Bounded Events. *arXiv preprint arXiv:1901.02710* (2019).

[55] Ei Pa Pa Pe-Than, Alexander Nolte, Anna Filippova, Christian Bird, Steve Scallen, and James D Herbsleb. 2019. Designing Corporate Hackathons With a Purpose: The Future of Software Development. *IEEE Software* 36, 1 (2019), 15–22.

[56] Ei Pa Pa Pe-Than, Alexander Nolte, Anna Filippova, Chris Bird, Steve Scallen, and James D. Herbsleb. 2020. Corporate Hackathons, How and Why? A Multiple Case Study of Motivation, Projects Proposal and Selection, Goal Setting, Coordination, and Outcomes. *Human-Computer Interaction* (2020).

[57] Jari Porras, Antti Knutas, Jouni Ikonen, Ari Happonen, Jayden Khakurel, and Antti Herala. 2019. Code camps and hackathons in education-literature review and lessons learned. In *Proceedings of the 52nd Hawaii International Conference on System Sciences*.

[58] Emily Porter, Chris Bopp, Elizabeth Gerber, and Amy Voida. 2017. Reappropriating Hackathons: The Production Work of the CHI4Good Day of Service. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM, 810–814.

[59] Bard Rosell, Shiven Kumar, and John Shepherd. 2014. Unleashing innovation through internal hackathons. In *Innovations in Technology Conference (InnoTek), 2014 IEEE*. IEEE, 1–8.

[60] Savisaari, Olli. 2017. How to be the perfect mentor at a hackathon? `https://www.perfektio.fi/en/blog/how-to-be-a-mentor` [Online; accessed 12-August-2020].

[61] Schuurman, Robbin. 2019. Creating a Stakeholder Analysis: How do you do that? `https://bit.ly/33xZguf` [Online; accessed 12-August-2020].

[62] Tauberer, Joshua. 2017. How to run a successful Hackathon. `https://hackathon.guide/` [Online; accessed 12-August-2020].

[63] Nick Taylor and Loraine Clarke. 2018. Everybody's Hacking: Participation and the Mainstreaming of Hackathons. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 172.

[64] Thackery, L. 2017. Hackathon playbook. `https://hackathon-planning-kit.org/files/hackathon-playbook-external.pdf` [Online; accessed 12-August-2020].

[65] The AstroHackWeek organizers. 2020. Astro hack week 2020. `http://astrohackweek.org/2020/` [Online; accessed 12-August-2020].

[66] The Chi Hack Night organizers. 2020. Chi Hack Night. `https://chihacknight.org/projects.html` [Online; accessed 12-August-2020].

[67] The DebCamp organizers. 2019. DebCamp. `https://wiki.debian.org/DebCamp` [Online; accessed 12-August-2020].

[68] The entrofy organizers. 2017. entrofy. `https://github.com/dhuppenkothen/entrofy` [Online; accessed 12-August-2020].

[69] The Geohackweek organizers. 2019. Geohackweek 2019. `https://geohackweek.github.io/` [Online; accessed 12-August-2020].

[70] The global hack organizers. 2020. The global hack. `https://theglobalhack.com/` [Online; accessed 12-August-2020].

[71] The Green Hackathon organizers. 2019. Green Hackathon. `http://www.greenhackathon.com/` [Online; accessed 12-August-2020].

[72] The HackHPC organizers. 2020. HackHPC. `http://hackhpc.org/` [Online; accessed 12-August-2020].

[73] The Hack@PEARC organizers. 2019. SGCI Hack@PEARC19. https://sciencegateways.org/web/wd/hackathon-2019 [Online; accessed 12-August-2020].

[74] The OHBM Brainhack organizers. 2018. OHBM Brainhack 2018. https://ohbm.github.io/hackathon2018/ [Online; accessed 12-August-2020].

[75] The SheInnovates organizers. 2020. SheInnovates. http://sheinnovates.us/ [Online; accessed 12-August-2020].

[76] The SteelHacks organizers. 2020. SteelHacks. http://steelhacks.com/ [Online; accessed 12-August-2020].

[77] The world of code hackathon organizers. 2019a. World of code hackathon. https://github.com/woc-hack [Online; accessed 12-August-2020].

[78] The world of code hackathon organizers. 2019b. World of code hackathon - schedule. https://github.com/woc-hack/schedule [Online; accessed 12-August-2020].

[79] The world of code hackathon organizers. 2019c. World of code hackathon - tutorial. https://github.com/woc-hack/tutorial [Online; accessed 12-August-2020].

[80] Thorsen, Anna. 2020. 50 Startups That Came From Hackathons. https://www.valuer.ai/blog/50-startups-that-came-from-hackathons [Online; accessed 12-August-2020].

[81] Erik H Trainer, Chalalai Chaihirunkarn, Arun Kalyanasundaram, and James D Herbsleb. 2014. Community code engagements: summer of code & hackathons for community building in scientific software. In *Proceedings of the 18th International Conference on Supporting Group Work*. ACM, 111–121.

[82] Visser, Friso. 2018. 7 tricks to build the perfect brainstorming question. https://bit.ly/30zytw1 [Online; accessed 12-August-2020].

[83] vj, satish. 2016. Please donâĂŹt organize âĂŸfunâĂŹ activities at hackathons. https://bit.ly/2EOdPj4 [Online; accessed 12-August-2020].

[84] Jorge Luis Zapico, Daniel Pargman, Hannes Ebner, and Elina Eriksson. 2013. Hacking sustainability: Broadening participation through green hackathons. In *Fourth International Symposium on End-User Development. June 10-13, 2013, IT University of Copenhagen, Denmark*.