

***Team Formation and Inclusion in Hackathons:
An empirical analysis and research proposal***

Debra Hevenstone, Oleg Lavrovsky, Nada Endrissat, Oliver Hümbelin

Abstract

Hackathon events focusing on social or public interest depend on participants from diverse and interdisciplinary backgrounds. However, to date, there is no general approach to managing inclusion at hackathons. In this paper we consider team formation's role in supporting inclusion. We describe two case studies with different approaches to team formation: a small event using open assignment and a larger event using an algorithm. Event evaluations suggest that open assignment may have generated teams of sub-optimal size and skill mix, while the algorithm led to dissatisfaction and reassignment, i.e., market "unraveling." Given the lack of documented algorithm use in hackathon team formation and inapplicability of team formation algorithms in other context, we turn to the economics literature on deferred acceptance algorithms proposing an approach to generate suggested matches followed by a second phase of open reassignment, and a data collection strategy to empirically assess whether this method improves inclusion.

1. Introduction

Hackathons are short sprint-like events where programmers and other experts from different professional and organizational backgrounds come together voluntarily to develop software or data prototypes. Over the past decades, they have evolved in different directions, including events where technical experts and the social or public sector work together [4]. While the importance of diversity and inclusion is emphasized in the literature [5, 6], little is known about how to address and successfully manage the inclusion of diverse participants (with respect to any dimension such as race or profession) to ensure the active empowerment and contribution of all participants.

We begin this article by reviewing the literature on inclusion and team formation in hackathons, considering how team formation may improve inclusion. In the second section, we highlight two social sector hackathons, contrasting how team formation strategies impacted inclusion, in particular interdisciplinary cooperation. In the third section we propose a team formation approach and a data collection strategy to assess it. Given the findings, and based on the literature on team assignment algorithms, we propose a new hybrid procedure and a research agenda to rigorously test whether the proposed procedure improves inclusion. Our proposed approach is relevant for a broad range of mid-sized events in which field knowledge and technical skills are combined.

2. Literature Review

2.1. Inclusion

Hackathons are traditionally used within the disciplines of computer science and engineering. However, social sector events depend on a diverse group of participants. A project where the proposed solution ignores the perspectives and needs of user groups or focuses on the technical and ignores field knowledge, is likely to fail to find convincing solutions [16] which is why inclusion and diversity have become a central topic among organizers of hackathons [5,10,12]. Inclusion refers to efforts to increase diversity within organizations or communities, which aims to ensure that all contributions regardless of background or orientation, are valued equally to those of their (technical) peers.

The current literature on hackathon inclusion focuses on women, as they are not only under-represented in computer science, but even more so at hackathons [19]. Research suggests women choose not to participate or feel excluded when participating because of cultural differences, low self-esteem, and male dominance (e.g., “mansplaining” or talking over female peers) [2,10]. Research on transsexual participants has similar findings with additional challenges related to smaller participant pools [12].

The issue of inclusion extends beyond identity to profession or discipline. For instance, programmers may tend to focus more on code than conceptual issues [7] and are used to marathon-style intense work sessions [6]. This can make it difficult to integrate non-technical participants. Inclusion can be about avoiding passive discrimination—making sure one group does not work as usual, failing to tap into the relevant knowledge of minority participants.

One way to address these issues, is to integrate non-technical knowledge by bringing in field experts to support teams [3] or by recruiting and integrating participants with more diverse backgrounds [13]. Alternatively, some events emphasize topical brainstorming over technical goals, which can lead to a mix of projects including both technical and non-technical [17]. Other events put an explicit focus on groups that are under-represented in these spaces like women, offering women-only or women-focused events or addressing topics of interest (e.g., milk pumps). These approaches may lead to successful events, but do not offer a general solution to managing inclusion. One type of general solution is to improve within-team processes, emphasizing, for instance emphasizing brainstorming techniques. However, the ability of organizers to influence these interactions is limited [4]. In the next section we turn to team formation as an opportunity for organizers to actively ensure

diversity and inclusion among participants, a solution that to date has remained largely unexplored among hackathon researchers and organizers.

2.2. Team assignment in Hackathons

Following the crucial recruitment phase, where a diverse participant group is registered, organizers can encourage inclusion by managing team formation. The most recent available reviews of team formation [11] split existing approaches into the two broad fields of “open selection” and “assignment” or “invitation.” Both tend to use specialized tools for hackathons (Dribdat, HackDash, Devpost, and Quill) for registration, recruitment, and project management.

The most widespread approach is “open selection,” meaning that the teams form themselves. This could be a series of pitches at the start of an event, followed by participants moving between pitch facilitators, ultimately choosing a team. Organizers can also intervene by creating language and skill mix suggestions or narrowing the search space by filtering out projects that do not have requirements matching the participants’ skill sets [9].

Alternatively, organizers can directly manage assignment. Organizers can manually assign people based on project and participant characteristics or can use algorithms or statistical approaches. One example is the software Entropy [5] which creates a participant pool from applicants by matching the distribution of applicant characteristics to the distribution of project requirements. Alternatively, at very large online hackathons, as in our second case study, organizers use algorithms to assign participants to specific projects—an approach we could not find in the academic literature on hackathons.

While open selection allows participants to choose projects with certain goals in mind: to learn new skills, advance their careers through networking and portfolio development, or pursue their interests [13], it can lead to too-large or too-small groups or to lopsided mixes of backgrounds [7,9]—problems which assignment can alleviate. On the other hand, team assignment can lead to participant dissatisfaction, team-switching, and drop-out. The challenge is thus how to best match participants in teams. Team assignment algorithms provide a promising avenue to address this challenge.

2.3 Team assignment algorithms

Existing literature discusses algorithms for team assignment in classrooms and in on-line labor markets. Classroom applications focus on diversity or similarity across different dimensions, but teams typically do not have specific characteristics and there are no participant preferences [8]. Some of the algorithms we studied use fitness functions, optimizing for balance, again not considering participant preferences –an approach suitable for the classroom but not for a hackathon [18]. Algorithms used in on-line labor markets do consider skill sets specific to team needs, again incorporating team balance, but minimize total costs with continuous participant bids, again with no participant preferences. Such a strategy could backfire with unraveling in a voluntary competitive situation like a hackathon [1].

Looking towards the economics literature “deferred acceptance algorithms” originally proposed by David Gale and Lloyd Shapley in 1962, have impacted managed markets [14], with variants being adopted to manage diverse matching problems including school choice, organ donor assignment, and medical residency assignments.

Gale and Shapley proposed an algorithm to solve the stable matching or “marriage” problem, i.e., in a two-sided market how to find a set of matches that offer a “stable solution,” (one in which no two parties who are not currently matched would reject their current match to be together). A stable solution avoids “unravelling,” (a rush to match outside of the algorithm), “congestion” (parties seeking to match in shorter and shorter time periods), and “strategic play” (the misrepresentation of preferences). The many-to-one variant (i.e., teams) is called the “college admissions” problem.

In the deferred acceptance algorithm both sides of the market rank their preferences. Then agents on one side make proposals in order of preference. Those who receive more proposals than they can accept, reject their least preferred, but do not immediately accept the others. Rather, they hold onto them, waiting for better offers, as other parties on the same side of the market might subsequently reject individuals who then come to them as a second choice, and are preferred to those who are provisionally accepted. Final acceptance is deferred to the end of the algorithm when no new proposals are made. The algorithm yields a stable solution that is optimal for the offering side of the market. The dominant strategy is the truthful representation of preferences.

In practice, deferred matching algorithms can use a centralized clearing house or a decentralized process, as in the case of clinical psychology, where there is a given time window in which hospitals make offers and residents hold them, immediately informing their prior match when a better offer

comes along. Notably, there can be additional constraints, as when couples wish to be matched with jobs together [14].

To conclude, team assignment algorithms are one avenue to improve inclusion. While much is known about team algorithms in the context of classroom and open markets, its usefulness in the context of hackathons is rarely discussed, providing us with a research opportunity to test its applicability in this specific context. Using two case studies, our research questions is: *What are the benefits and downsides of open versus algorithmic team assignment with respect to inclusion among hackathon participants/teams?*

3. Case Studies of Team Assignment at Inter-disciplinary Hackathons

3.1 Methodology

We selected two cases with similar need for interdisciplinary cooperation but divergent approaches to team assignment. VersusVirus was selected as a rare case of a Hackathon using an algorithm for team assignment. With ten years of data available to us on hosted hackathons, VersusVirus is the only one that used a team assignment algorithm. Looking towards secondary data, we could find no evidence of other Hackathons using algorithms. Hack4SocialGood was selected as a counterpoint, using open assignment in an interdisciplinary context. The two events were similar in being interdisciplinary, on-line, putting collaboration over competition, and having interdisciplinary cooperation as an explicit goal. The events were dissimilar in size, with our proposed procedure applicable to approximately this range in size (50 to 1000 participants).

	<i>Hack4SocialGood</i>	<i>VersusVirus</i>
organizer	university	consortium
challenges	social organizations	organizer brainstorm
N participants	40	800
N evaluation	24	580
% technical participants	43	93
% women	37	41
incentives	cash prizes	coaching & financing
team assignment	open	algorithm

Both events were evaluated with participant surveys. The Hack4SocialGood evaluation focused on inclusion, while the VersusVirus evaluation was more general. The Hack4SocialGood evaluation instrument collected background information on gender, age, educational attainment, field of employment, and prior hackathon participation as well as extensive questions on inclusion and teamwork (see the online Appendix). VersusVirus had two sources of information (that could not

be linked): registration data and post-event evaluation. From registration data we know the distribution of individuals' self-assessed "hacker skills" (e.g., programming) and hacker type (e.g., nerd and language; see Appendix for full list). From the evaluation we know likely gender (from handle names), prior hackathon experience, hours work on projects as well as general qualitative questions about the participant experience (see online Appendix).

We use only qualitative comments from the evaluations to assess team assignment and inclusion at these two events. While the Hack4SocialGood evaluation asked the relevant questions, only 14 respondents were active event participants. However, the open fields following each question means we have several in-depth comments on inclusion and team formation. In contrast, the VersusVirus event was large, but the evaluation used exclusively open-ended questions and did not ask about inclusion or team experience. However, many VersusVirus respondents chose to comment on team formation and inclusion because of the strong positive/negative experiences detailed in the next section.

3.2 Open team assignment at Hack4SocialGood

The Hack4SocialGood (<https://hack4socialgood.ch>) focused on helping social service organizations. Organizations hosted specific projects including data and visualization tasks, such as an assessment of social assistance expenditures, as well as software development, such as an online platform for a social service provider to interact with clients and caregivers. Organizers helped social service organizations prepare their challenges and the event was run on an open-source hackathon platform (Dribdat).

Inclusion is particularly relevant when there are diverse participants. Organizers diversified recruitment by partnering with various local social and technical organizations, yielding 40 registered participants: 8 researchers, 15 from the social sector, and 17 with a technical background with 37% female. Unlike most other hackathons, technical participants were a minority. Most participants commented that they were motivated by fun or learning (e.g., "I wanted to gain an impression of digital solutions") while just two participants reported wanting to use their technical skills for social good (e.g., "to apply my knowledge to a good cause.")

Individuals were sorted into teams using "open assignment." Each challenge had a designated chat room with only one team per challenge. In each chat room, a designated host from the organization explained and supported the challenge. In some rooms there were too few participants, so people self-reassigned. The largest team was five, while the smallest two teams had two people each.

Given that the event was dominated by field, not technical, knowledge, the question of inclusion focused on minority technical, not field, experts. With 43% technical participants it would have been possible to integrate technical participants to solve the challenges, but these individuals were not evenly distributed. In response to the question, “which skills were missing from your team?” no one reported a lack of field knowledge, instead mentioning technical skills, web design, graphic design, web development and IT. From the participant perspective technical skills were thus scarce and many respondents suggested increasing participation. Alternatively, participants suggested re-orienting the event away from the technical.

Hack4SocialGood Comments

Active participants were at the lowest limit.

Too few participants.

Relatively few “technies” at this hackathon.

I highly recommend doing a joint event.

It would be interesting to open the Hackathon to an interdisciplinary public.

Challenges could also be used for social problems where technology is not in the foreground (hacking in a broader sense). This would open up the field of participants.

Overall, the Hack4SocialGood event attracted a substantial minority of technical participants who were not equally distributed across teams. Some participants felt that technical skills were lacking, requesting better recruiting, and re-orienting the format away from technical goals. Alternatively, the scarce technical resources could have been better distributed through organizer facilitation.

3.3 Algorithm-led team assignment at VersusVirus

[VersusVirus](https://www.versusvirus.ch/) (<https://www.versusvirus.ch/>) was an interdisciplinary hackathon focused on the COVID-19 pandemic. More than 5,000 people signed up, with 800 participants present at the start of the event. The organizers broadly defined 14 overarching topics, ranging from the protection of risk groups to alternative education forms with specific project ideas (250) within each topic. Participants could also propose projects. The resulting solutions ranged from online apps to check the crowd density of public places in real time, to online tools for education and socializing.

With respect to occupational and identity diversity, the most common reported skill was programming with about 5% reported having exclusively nontechnical skills (e.g., social/education). Although a strong majority spoke English, 12% of registrants reported speaking only French or German. Of those participants whose genders we could infer from their handle, 59% were men.

With thousands of people signing up for the event and just a two-week lead-time, the organizers decided to use an algorithm to assign teams, using the aforementioned registration data including self-assessed skill and type as well as preferred topics. In addition, they could sign up in groups as a team.

The organizers defined each topic as needing certain skills and all teams were required to have at least one of each. Ten people were assigned per team, anticipating a 20% no-show rate. After assignment, at the event, teams could jointly select a specific project.

The algorithm went as follows:

- For those who signed up as a team, but had less than 10 people
 - Select a random individual
 - Check if the team needs their skill or type
 - If there is a match, add the person to the team, if not, continue
 - When the team has all skills and types, assign random individuals until 10 spots are filled

- For all others
 - Sort into 4 language pools: English speaking, only German, only French, only Italian.
 - Within the only-French and only-Italian speakers, create random 10-person teams.
 - Within the English and only-German groups do the following for each topic:
 - Create an empty team for each topic. Within team:
 - Randomly select an unassigned person. Test if they have a necessary type or skill.
 - If they have a necessary type or skill, assign them, if not pick another random person and repeat
 - Once teams have all necessary types and skills, assign random people until 10-person teams are achieved.

- If there are still unassigned individuals, take first one, select their topic of interest, and create an empty team with this topic. Fill with unassigned individuals as before.

The algorithm ran into several obstacles. First, with an 80% no-show rate rather than the anticipated 20%, many teams were empty or had just 1 or 2 members at the beginning of the event. Second, the structure of 250 projects (invisible at registration) subsumed into 14 broader topics was confusing and many found there were interesting projects under unchosen topics. Third, some teams who signed up together did not want to admit additional members. Fourth, there was an interpersonal element that motivated individuals to switch teams. Finally, there were some communication gaps with participants having insufficient proficiency in the languages where they reported proficiency or having not closely read topic definitions. These issues led to a manual re-assignment process that endured for almost half of the event, with around a third of all participants switching.

The evaluation highlights participant frustration with the team formation process. Although only 10% judged the overall negatively, 9.5% offered *unsolicited* feedback about team formation. Common themes were confusion around the subgrouping of specific projects under broader themes and chaos during re-assignment, critiqued as “chaotic” and “Kafkaesque.” Despite the logistical challenges, the algorithm was reasonably successful with respect to inclusion. Twenty-two participants lauded the geographic and professional diversity they encountered in their teams.

VersusVirus Comments

Very good team spirit and openness to non-tech solutions

Possibility to meet people, not just IT

Great experience in interdisciplinary work

Energy of the participants, diversified background of team members

I met many dedicated, smart, helpful, funny and curious experts of all kinds of specialties

I almost feel like a techy, and I am no techy whatsoever!!

No Roesti Graben. Global swiss emulsion.

I find the idea incredible, I like the "One Switzerland," meaning solidarity within the whole country

I met foreign, but clever and motivated people

There were three critiques related to inclusion on dimensions not considered by the organizers: 1. engaging broader disciplines (humanities and skilled trades) and 2. including older participants or those with family obligations.

To summarize, the two events were similar in that both needed to integrate those with field and technical expertise. They were different in that the first was very small—the organizers did not consider using an algorithm for team formation, while in the second, the use of an algorithm was almost a forced practical measure. The team formation approach impacted team size, mix, and the match of the team’s strengths to the projects they worked on. For both events, evaluations suggest inclusion mattered most with respect to skill, not identity, as is often the focus of academic literature (see section 2.1).

Based on our findings, we suggest that open assignment can lead to teams with insufficient skill mixes, while algorithmic assignment can lead to dissatisfaction and disorganized manual re-assignment (called “unravelling” in the economics literature on deferred acceptance algorithms). In the next section we propose a hybrid team assignment approach using a deferred acceptance algorithm for “suggested teams” along with a flexible yet organized process for reassignment, as well as a data collection strategy to evaluate the approach.

4. A Proposal for Team Assignment

Team assignment at hackathon events overlaps with markets in which deferred acceptance matching algorithms are already used, barring some key differences. Similarly, there is a two-sided many-to-one market, with the problem of couples in medical matches akin to pre-formed teams registering. There are differences in that hackathon team sizes are not strictly fixed, and that teams have skills they need, but not strict ranked preferences. As in medical matches there is a human element, but in this case, team assignment itself impacts participant preferences (feedback) whereas in medical matches, the team character is already defined by existing members who are not part of the matching process.

We propose evaluating a two-phase assignment procedure. In the first phase, a deferred acceptance algorithm generates a “suggested” team, followed by a second phase where dissatisfied individuals can switch teams (i.e., semi-open assignment). In the next two subsections we detail our proposal and a data collection strategy to evaluate it.

4.1 Phase I: Algorithmic pre-assignment

We define the two sides of the market as “groups” and “participants.” Participants are those registered for the event. Groups are empty sets based on the number of total topics and the number of registered participants. Topics can be designed by the event hosts or additionally submitted by participants ahead of time. The organizers choose S , the optimal group size and A , the anticipated no-show rate. Groups are assigned S/A empty slots and with P participants. Groups are then equally divided over topics ($G = (PA)/(S)$) with the remainder being randomly assigned. (See the Appendix for pseudo-code.) This prioritizes *de facto* organizer preferences for all topics to be addressed over participants’ preferences, in the case of overwhelming preference for one or a few topics.

We describe a basic scenario with only two features: topics and skills, which increases the chance of finding a stable match. Initialized groups have S/A empty slots, a topic, and a list of necessary skills. For each group, the required skills are distributed equally across slots. The remainder of slots are randomly assigned across the relevant skills—assuring redundancy in skills within the group even after attrition as compared to the algorithm used in the case study. Participants provide two pieces of information: 1. Skill(s) and 2. Topic(s): a ranking of their preferences for which topic they would like to work on. The algorithm begins with a list of empty groups, with each slot having a specified skill. Assignment proceeds as follows:

- Loop through the list of participants
 - Participants loop through groups starting with their first-choice topic, offering to join.

- Groups provisionally accept participants as follows:
 - If there are unfilled slots: preferably in a slot with their skill, if no matching skill, in a random slot.
 - If there is a filled slot with (one of) their skill(s), but the current occupant does not have the skill. The group releases the prior occupant & triggers a flag indicating there was a release this round
- At the end of the round, those without a group repeat the offering-loop in order of declining topic preferences. If rejected by all topics on their list, participants apply randomly.
- The loop ends when all participants are assigned, and none is released.

This core algorithm can have variations such as: 1. assigning a specific ratio of slots by skill within group or 2. participants ranking skills or 3. allowing teams to pre-register or not and considering pre-registered groups as exclusive or not.

4.2 Phase II: Open re-assignment

Algorithmic assignment is advantageous in that it will increase diversity and inclusion by evenly distributing skills across groups and maximizing participant preferences in topics while creating stable matches. Further, shy or minority members might feel more justified in their contribution when confronted by majority group team members. Nevertheless, participants can become excited about a topic or working with a specific person during the event, might dislike the personalities in their assigned group, or attrition might exceed expectations, leaving some groups too small.

Organizers should set aside a period in which all groups are publicly accessible and dissatisfied individuals can provisionally sit in on another group, choosing their definitive group at the end of the period (as is currently the case at most hackathons). During the reassignment period there are no constraints on groups size or skill mix, with the expectation that groups can determine whether the group size or mix will work for them. The better the algorithm, the less self-reassignment should occur.

Using a deferred acceptance algorithm matching followed by open reassignment may manage inclusion in a way that offers participants their best possible experience and generates inclusive productive groups. It is important to objectively assess whether this is the case. In the following section, we set out a data collection agenda.

4.3 Data Collection Agenda

To evaluate our proposed procedure, we suggest data collection at three levels: event, group, and participant. Event data is understood to be meta data, collected by organizers and submitted to the platform organizer. Group- and participant-level data is collected via online platforms through registration before the event, during the event as projects are developed, and following the event via feedback. Information should be linkable through both project and individual identifiers used consistently throughout registration, participation, and feedback.

Before the event, organizers must note all the variables related to the team assignment approach (whether the algorithm was used, number of skills listed, etc.). In addition, diversity or inclusion indicators should be noted (e.g., the percent of speakers by gender and specialty).

Prior to the hackathon, information from the algorithm run should be recorded: individual-group assignment, the percentage of participants assigned to groups not requiring their skills, and the percent of participants assigned to their first (or other order) preference, and the percent of groups with skills not covered by participants. Additionally, the individual level registration data should be saved including participants' skills, topic rankings, gender, and age.

During the event, information can be collected via the platform: individual-group assignment following the open assignment phase, the submission stage the group has reached (e.g., prototype, working solution, presentation uploaded), and whether the group received an award: both being measures of productivity. At the individual level, the platform can measure individual engagement, considering the amount of time spent on the platform and the number of uploads and downloads. Post-event researchers can construct measurements of inclusion and diversity such as segregation indices for technical vs. field knowledge within teams.

Information collected during the event might be reliable, but inexact, and as such should be complemented by a post-event survey. Diversity can be measured by asking participants about, for example, prior experience with hackathons and technology as well as professional background. Inclusion can be measured using questions related to belonging and authenticity, self-segregation, communication, stereotype engagement, and feelings of belonging or of one's contribution being valued [12]. Platform developers should offer organizers a standard menu of questions so that data can be aggregated across hackathons.

With this information it would be possible to study how team formation approach impacts group diversity, participant satisfaction, productivity, and feelings of inclusion.

5. Conclusion

Inclusion is an important problem at interdisciplinary hackathon events where engaging all participants leads to better results and better experiences. Most hackathon events we have looked at use open assignment processes. However, there is evidence from Computer Science that team assignment algorithms could achieve better balance and Economics research suggests that deferred acceptance algorithms could improve satisfaction and avoid unravelling. We have found no prior studies of algorithm use for team assignments in hackathons or discussion of appropriate algorithm development or testing in the hackathon context.

We analyzed two events taking opposing approaches to team assignment. Our findings suggest that open selection generated inconsistent team sizes and gaps in necessary competencies while the tested algorithm suffered from participant dissatisfaction and reassignment (unravelling). We propose a deferred acceptance algorithm that should support a better balance than the open assignment approach, and potentially lead to greater satisfaction than the tested algorithm. Recognizing the importance of flexible and dynamic interactions as well as feedback between team composition and preferences, we propose following the algorithm with a second phase of open reassignment. Finally, we outline an approach to data collection that would allow researchers to systematically study team assignment. Our study has several limitations. First, the suggested approach might apply to a subset of all hackathons ranging in size from perhaps 50 to 1000 participants and a focus on interdisciplinary collaboration. Second, the existing empirical evidence on algorithm design in team assignment at hackathons is small. With more empirical evidence, the potential for algorithm improvements will be clearer.

Future research should empirically test algorithmic approaches to team assignment and look at their impact on not only inclusion, but also on productivity and the range of applicability.

6. Literature

- [1] G. Barnabò, A. Fazzino, S. Leonardi, and Chris Schwegelshohn.. “Algorithms for Fair Team Formation in Online Labour Marketplaces“. *Companion Proceedings of The 2019 World Wide Web Conference*, pp. 484–90, 2019.
- [2] J. C. Carver and A. Serebrenik, “Gender in Software Engineering,” *IEEE Software*, vol. 36, no. 6, pp. 76–78, 2019.
- [3] F. Doshi-Velez and Y. E. Marshall, “Hack Ebola with Data: On the Hackathon Format for Timely Data Analysis,” *Procedia Engineering*, vol. 107, pp. 377–386, 2015.

- [4] A. Filippova, E. Trainer, and J. D. Herbsleb. "From Diversity by Numbers to Diversity as Process: Supporting Inclusiveness in Software Development Teams with Brainstorming," *2017 IEEE/ACM 39th International Conference on Software Engineering*. IEEE, 2017.
- [5] D. Huppenkothen, B. McFee, and L. Norén, "Entropy your cohort: A transparent method for diverse cohort selection," *PLoS ONE*. vol. 15, no. 7, 2020. [6] L. Irani, "Hackathons and the Making of Entrepreneurial Citizenship," *Science, Technology, & Human Values*, vol. 40, no. 5, pp. 799–824, 2015.
- [7] M. Komssi, D. Pichlis, M. Raatikainen, K. Kindström, "What are Hackathons Good For?" *IEEE Software*, vol. 32 no. 5, pp. 60-67, 2015.
- [8] R.A. Layton, M.L. Loughry, M.W. Ohland, and G.D. Ricco. "Design and Validation of a Web-Based System for Assigning Members to Teams Using Instructor-Specified Criteria," *Advances in Engineering Education* vo. 2, no. 1, 2010.
- [9] A. Nolte, E. P. P. Pe-Than, A. O. Affia, C. Chaihirunkarn, A. Filippova, A. Kalyanasundaram, A. Medina, E. Trainer, and J.D. Herbsleb, "How to organize a hackathon-A planning kit." *arXiv preprint arXiv:2008.08025*, 2020.
- [10] L. Paganini and K. Gama, "Engaging Women's Participation in Hackathons: A Qualitative Study with Participants of a Female-focused Hackathon." *International Conference on Game Jams, Hackathons and Game Creation Events 2020*. Association for Computing Machinery, New York, NY, USA, 8–15, 2020.
- [11] E. P. P. Pe-Than, A. Nolte, A., Filippova, C. Bird, S. Scallen, S., and J.D. Herbsleb, "Designing corporate hackathons with a purpose: the future of software development," *IEEE Software*, vol. 36, no. 1, pp. 15-22, 2018.
- [12] R. Prado, W. Mendes, K. S. Gama, G. Pinto, "How Trans-inclusive are hackathons?" *IEEE Software*, vol. 38, no. 2, pp. 26-31, 2020.
- [13] B. Rosell, S. Kumar, and J. Shepherd, "Unleashing innovation through internal hackathons," *IEEE Innovations in Technology Conference*, pp. 1–8, 2014.
- [14] A. E. Roth, "Deferred acceptance algorithms: History, theory, practice, and open questions," *International Journal of Game Theory*, vol. 36, no. 3, pp. 537-569, 2008.
- [15] L. M. Shore, A. E. Randel, and B. Chung, "Inclusion and Diversity in Work Groups: A Review and Model for Future Research," *Journal of Management*, vol. 37, pp. 1262–89, 2011.
- [16] P. Soltani, K. Pessi, K. Ahlin, and I. Wernerer, "Hackathon – a method for Digital Innovative Success: A Comparative Descriptive Study," *Proceedings of the 8th European Conference on Information Management and Evaluation*, 2014.
- [17] N. Taylor and L. Clarke, "Everybody's hacking: participation and the mainstreaming of hackathons." *CHI*, Association for Computing Machinery, pp. 1-2, 2018.
- [18] V. Yannibelli and A. Amandi. "A Hybrid Evolutionary Algorithm Based on Adaptive Mutation and Crossover for Collaborative Learning Team Formation in Higher Education," *International Conferences on Intelligent Data Engineering and Automated Learning*, Springer, Cham, 2017.[19] S. Zweben and B. Bizot, "2014 Taulbee Survey," *Computing Research News*, vol. 28, no. 5, pp. 2-60, 2016.

7. Appendix

Pseudo-code for matching algorithm

```
GN = (PA)/(S) #number of groups
N = int(GN/TN) #number of groups per topic
R = modulo(GN/TN) #remainder groups get random topic
```

```
#create groups to fill with people; teams have topics and slots; topics have a skill list; slots have skills and assigned participants
Left out steps like creation of lists, get set etc... just rough.
for i 1:GN(
    for j 1:N(
```

```

aTeam = Team();
aTeam.Topic =Topic[i];
for k 1:S/A( #for every slot in team
  aTeam.aSlot();
  if k<(length(aTeam.aTopic.skillIist[]))
    (aTeam.aSlot.skill= aTeam.aTopic.skillIist[k],
     else(aTeam.aSlot.skill = aTeam.aTopic.skillIist[randUniform(0:
       length(aTeam.aTopic.skillIist[]))])
    )
  aTeam.aSlot.occupant=NULL;
)))
for i 1:R(
  aTeam = Team();
  aTeam.Topic =Topic[randUniform(1,length(topicList))];
  for k 1:S/A( #for every slot in team
    aTeam.aSlot();
    if k<(length(aTeam.aTopic.skillIist[]))
      (aTeam.aSlot.skill= aTeam.aTopic.skillIist[k],
       else(aTeam.aSlot.skill = aTeam.aTopic.skillIist[randUniform(0:
         length(aTeam.aTopic.skillIist[]))])
      )
    aTeam.aSlot.occupant=NULL;
  )
#Loop through participants finding them a team with a matching topic. Participants have a topic
preference list, a team (initialized at null) and a skill.
N=length(participantList)
Flag = 1;
While Flag==1(
  Flag=0
  for i 1:N(
    aPerson = personList[i]
    while aPerson.team=null(
      for j 1:length(aPerson.topicList)(
        aTeam = teamList[randUniform(1,length(teamList))]
        while(aTeam.topic != aPerson.topic)(aTeam =
          teamList[randUniform(1,length(teamList))])
        offerTeam(aPerson) #refers to function for team object
      )
    )
  )
)
#function for team to reject or accept and assign only those with matching skills until no assignments
aTeam.offerTeam(aPerson){
  for i 1:length(slotList)(
    if(aTeam.slot[i]=null, accept(aPerson);
    if(aTeam.slot[i]!=null & aTeam.Slot.occupant.skill != aTeam.slot[i].skill()) &
    aPerson.skill=aTeam.slot[i].skill(), accept(aPerson));
  )
}
#there is an accept() function for teams that triggers the global flag=1, indicating an assignment was
made, sets the person's team variable, adds the person to their team list and if there is already a
person in the slot that they reject, sets that person's slot to null

```

Evaluation Instruments

Hack4SocialGood Evaluation Questions:

- To what extent do the following statements apply (1= not at all, 5=absolutely)
 - Through teamwork I learned about new perspectives
 - Working in the team I came into contact with people in occupations with whom I would not normally have contact.
 - Through teamwork I learned about new approaches to problems in the social sector.

- I was able to contribute my ideas and competencies to the teamwork.
- Our team had the appropriate mix of knowledge and competencies to address the challenges.
- Teamwork added value.
- The hackathon enabled a new kind of collaboration.
- Through the hackathon, I encountered people and perspectives that are relevant to my work.
- The hackathon brought new ideas and innovation to the social sector
- The ideas have the potential to be applied in practice
- The hackathon has encouraged the exploration of innovative approaches in the social sector
- The hackathon opened up new perspectives on the topics of
 - topics of digitization and data use.
 - Through the hackathon I improved my digital knowledge and abilities.
 - Through the hackathon, I have learned new insights and practices that I would like to implement in my day-to-day work.
 - Would you suggest additional hackathons in the social sector?
- Open-ended questions
 - Which competencies were lacking in your team?
 - Further comments on teamwork?
 - For what reason should the Hack4SocialGood (not) be repeated in the future?
- My attitude towards the advancing digitalization in the social sector is: critical/somewhat critical/half-half/somewhat optimistic/optimistic.
- My digital abilities and competency is: beginner/basic/competent/expert

Versus Virus registration data

Skills

allrounder,Allrounder
 businessandsales,Business & Sales
 craftsmen,Craftsmen
 creativeandarts,Creative & Arts
 datawrangling,Data wrangling
 designandmultimedia,Design & Multimedia
 environmentalandsustainability,Environmental & Sustainability
 marketingandpr,Marketing & PR
 medicalandsport,Medical & Sport
 philanthropicandhumanitarian,Philanthropic & Humanitarian
 presentingandpitching,Presenting & Pitching
 programming,Programming
 projectmanagement,Project Management
 serviceandgastronomy,Service & Gastronomy
 socialandeducational,Social & Educational
 strategyandbusinessdevelopment,Strategy & Business
 Development
 sportsmovement,Sports & Movement

Types

conceptualist,Conceptualist, "You support creative minds by breaking down their many ideas and translating them into a concept. You create a tangible, realistic concept from a vague idea. "

creativehead,Creative Head, "You get new creative ideas in 1-minute intervals. Challenges are the playground for new ideas."

doer,Doer, "You are a website developer, designer or a business strategist and maker in general and thanks to you ideas become reality. You translate what is thought into something tangible."

flexible,Flexible,"You jump in where it's needed, because you can use your broad experience to advance the common idea in many ways."

challenger,Challenger, "You are critical and point out to the team when they move into utopian areas and bring the thinking of others back on track."

nerd,Nerd, "You inspire and support the team with your valuable expertise."

fieldresearcher,Field researcher, "You prefer to be in direct exchange with the target group in order to make the result as user- and needs-oriented as possible."

Versus Virus Evaluation Questions (all open-ended except last):

- How did you hear about us?
- How did you experience #VersusVirus?
- How did you experience the online events and moderation?
- How did you experience the #VersusVirus mentoring process?
- How did you experience the "How to" sessions?
- How did you experience the Versus Virus Slack workspace and the communication with the team?
- How did you experience the #VersusVirus overall support, including mail and contact form?
- How did you experience the #VersusVirus website and application?
- How did you experience the #VersusVirus project submission, including deliverables and judging criteria?
- How did you experience the #funversusvirus?
- I was excited to be part of the #VersusVirus Hackathon because:
- During the #versusvirus hackathon, I learned:
- Fun fact about you and your team?
- What can we do better?
- Do you want to tell us something else?
- Would you recommend others to participate in future additions of Versus Virus?
(yes/no)