Hackathons in computer science education: monitoring and evaluation of programming projects

Zarema S. Seidametova[1], *Zinnur S.* Abduramanov[1] and *Girey S.* Seydametov[1]

[1]*Crimean Engineering and Pedagogical University, 8 Uchebnyi per., Simferopol, 95015, Crimea*

Abstract. There are several ways in making learning activities more engaging and interesting to computer science students. In addition to traditional higher education (curriculum, project-based approaches, lectures, labs), we consider competitive approaches such as hackathons to develop hard and soft skills. Computer science education needs to change requirements for hard and soft skills. In order hard skills university CS education has to capture the changes behind the trends, such as big data, artificial intelligence, cloud computing, etc. Developing soft skills is important teamwork, end-user awareness, collaboration, etc. This study aims to present the various ways to implement hackathons in the context of CS education. We also present a taxonomy of hackathons based on our experiences and observations from 2016 to the present. We aim to share our lessons learned on the following issues: (1) How can hackathons be designed in CS education to teach students necessary skills and competencies; (2) what kinds of programming projects monitoring and evaluation we need during hackathons.

Keywords: computer science education, programming project, soft and hard skills, agile methods, hackathon, collaboration

1. Introduction

Nowadays time-based events, such as hackathons, code camps, or code fests, are used to learn and teach industrial software development using intensive collaboration to complete a project. Like teaching programming and software development courses, time-based events (particularly, hackathons) can be considered as successful learning tools. Hackathons have been adopted not only in computer science, software engineering education [2, 12, 18, 22, 27–29] but in enterprises [9, 30], corporations [23], virtual communities [3, 16, 25] and others. Most hackathons are organized with specific goals to create some innovative technology or product.

Briscoe and Mulligan [4] describes a hackathon as a digital innovation that can be used to teach computer science concepts. Falk, Kannabiran and Hansen [7], Kollwitz and Dinter [17], Medina Angarita and Nolte [19], Nolte et al. [24] discuss the important aspects of hackathon organization, an approach to the business environment involving experts from companies.

The topic of hackathons is reflected in the papers of Falk Olesen and Halskov [8], Frey and

z.seydametova@gmail.com (Z. S. Seidametova); abduramanov.z.s@gmail.com (Z. S. Abduramanov); s.girey.s@gmail.com (G. S. Seydametov)
http://cepulib.ru/index.php/ru/resursy/personalii/47-s-personalii/169-sejdametova-zarema-sejdalievna (Z. S. Seidametova); https://kipu-rc.ru/sotrudniki/103-abduramanov-zinnur-shevketovich.html (Z. S. Abduramanov); https://kipu-rc.ru/fakultet-ekonomiki/kafedra-prikladnoj-informatiki.html?id=107 (G. S. Seydametov)
0000-0001-7643-6386 (Z. S. Seidametova); 0000-0002-2818-4759 (Z. S. Abduramanov); 0000-0002-1004-4141 (G. S. Seydametov)

Luks [10], Gama et al. [11], Medina Angarita and Nolte [20], Pe-Than et al. [26]. Nolte, Chounta and Herbsleb [23] note the importance of hackathons as an educational environment for the formation of students' skills in developing software products as a course project. Cobham et al. [6], Imam and Dey [15] describe a methodology for involving stakeholders (universities, IT companies, students, volunteers, etc.) to participate in the hackathon. Byrne, Sullivan and O'Sullivan [5], Guerrero et al. [13] present possible forms of organizing hackathons, and also describe the advantages of hackathons. Abduramanov, Ibraimov and Seydametov [1]investigated the automation of tools for tracking the results of hackathon participants, as well as checking the work of students.

Mtsweni and Abdullah [21] analyze the factors influencing the motivation and professional interest of computer science students. Horton et al. [14], Uys [32] describe teamwork methods, as well as important aspects of agile application development methodology that need to be taught to students.

Despite the popularity of hackathons, there are no more researches and studies that have the answers to the following questions: what kind of approach to organizing hackathons has a greater pedagogical effect (for example, short-term or long-term event), what stakeholders should be involved, what are the practical differences in organizing hackathons. To study this problem, we use such methods as a literature review, own experience in organizing time-based events, surveys of hackathon participants (students), mentors, and representatives of the IT industry.

2. Objectives and taxonomy of hackathons

The universities' computer science and software engineering departments need to redevelop their teaching approaches because IT industries are rapidly changing and developing the requirements for both hard and soft skills [31, 33]. In terms of skills, curriculum guidelines such as IT professional and educational standards, ACM, and IEEE Curricula (both in Computer Science and Software Engineering) aim to accommodate the increasingly interdisciplinary nature of computer science and software development.

Hard skills are the skills that are easy to observe, measure, and demonstrate – for example, these are programming skills, calculating skills, or swimming skills. Without hard skills, people cannot effectively engage in a certain type of activity. Soft skills are more about communication skills. Soft skills are difficult to measure, but these are the skills that are most effective in demonstrating and applying hard skills. Soft skills are essential in any kind of activity. In terms of soft skills, the emphasis is shifting towards teamwork, end-user awareness, and understanding of the software application development context.

Often, universities and IT industry representatives see the importance of hard and soft skills differently. Learning is becoming more interactive, that is why the importance of collaborative learning and teamwork is increasing. At the same time, hackathon-like activities that teach rapid team-based software development are becoming increasingly popular for teaching computer science and software development. Such approaches can help to teach students to collaborate in teams, to help each other in achieving learning goals. Communication and collaboration have become important factors to consider when teaching students and are essential skills in the 21st century.

Hackathons are project-based and work-integrated events that address real-world problems or challenges posed by IT industry representatives.

Events (hackathons) aimed at learning through interaction in a common project may differ in duration, forms of implementation, and participants coverage.

Table 1 shows the classification (taxonomy) of Hackathons. Hackathons can be different in duration, objectives, goals, stakeholders, and achieved results.

Table 1
Hackathon's classification (taxonomy).

Time-based events	Goals	Stakeholders and beneficiaries	Results
Hackathon Duration: 24 or 48 hours	The rapid introduction of new computer science and software development topics (topics that are important but not presented in the curriculum). It is considered to approach as "learning by doing".	Universities, educators, students, IT industry	Implementation in computer science concepts, skills improvement, new knowledge, rapid prototypes
Hackathon Duration: 1 or 2 weeks	Understanding of professional topics and concepts. It is considered to approach as "learning by doing".	Universities, educators, students, IT industry	Deep knowledge and skills on the hackathon's topic, working prototypes
Team building for projects (freshmen students)	Communication and acquaintance. Learning process motivation. Idea generation. Focus on interaction	Universities, educators, students	Understanding own strengths and weaknesses in programming.
Specialization in IT	Acquaintance with software development technologies.	Universities, educators, students	Knowledge of the technology and the acquisition of relevant skills.
Hackathon as an exam or a capstone project	Testing and evaluating the skills using a real project environment.	Universities, educators, students	Knowledge and skills that every student should have.
Competition	Innovative approaches and solutions to the given problem and the implementation.	Universities, educators, students, mentors	Ability to work with a real problem, innovative opportunities, implementation, participation in competitions.
Industry hackathon	Results presented: product and solutions on a company-defined topic using special tools and methodologies. Idea generation and diffusion	Students, IT industry	Ability to work in real conditions and the company's environment.

3. Hackathon model

One of the hackathon goals is to deepen and expand the professional competencies of computer science or software engineering students. The approach to solving this problem is to apply the existing theoretical knowledge and practical skills of students in real professional activity. This approach can be realized through the flow of students in the framework of the educational process of various types of practices and the involvement of professional software developers to contact students outside of the educational process.

Hackathons were designed as time-based events that help programmers and software developers collaborate on dome software ideas or projects. As we showed in the previous section, hackathons can last from one day up to a month. Mostly the motivation behind hackathons can be connected with educational or social goals.

Hackathon as a time-based and work-integrated event includes multiple desirable features, for example, stakeholders and students can face-to-face communicate and interact, a better understanding of how to work together in different teams.

To gain information that can be useful for the organizing of the hackathons we provided a survey among the individuals (*N*=104) who took part in hackathons before as participants (63,46 %), mentors (17,31 %), organizers (4,81 %), or judges (14,42 %) (figure 1).

We got the following answers on the question about who are the main beneficiaries of hackathons (figure 2): hackathon participants – 43,9 %, organizers – 29,27 %, society – 22,76 %, it depends on hackathon – 3,25 % and nobody – 0,81 %.

We asked during the survey about the advantages and disadvantages of hackathons. The answers you can see in table 2. Table 2 describes some of the advantages and disadvantages that hackathon participants can expect.

Table 2
Some advantages and disadvantages of hackathons

Advantages	Disadvantages
Team engagement, team members get the opportunity to work together on a project, as well as fulfill their roles in the project	Lack of time to work on other projects during the Hackathon (for example, for students, this is completing assignments in academic disciplines)
Development of a common understanding of software development	Lack of individual work due to the need to interact with the rest of the team
The creative process in a real project, coinciding with the acquisition of the new skills	All received artifacts such as codes, diagrams, or data are not always used
The intensity and quick results	Exhaustion
Training and acquisition of new competencies	High intensity of the event and uneven load
Solving non-standard tasks	Restricting the use of tools and technologies
Obtaining insights for the implementation of future projects	Weak motivation with a small number of participants
To be connected with a real problem, to have innovative opportunities, implementation	The unequal level of training of team members
Additional financial (material) income	The possible bad faith of hackathon's organizers
Public speaking skills	

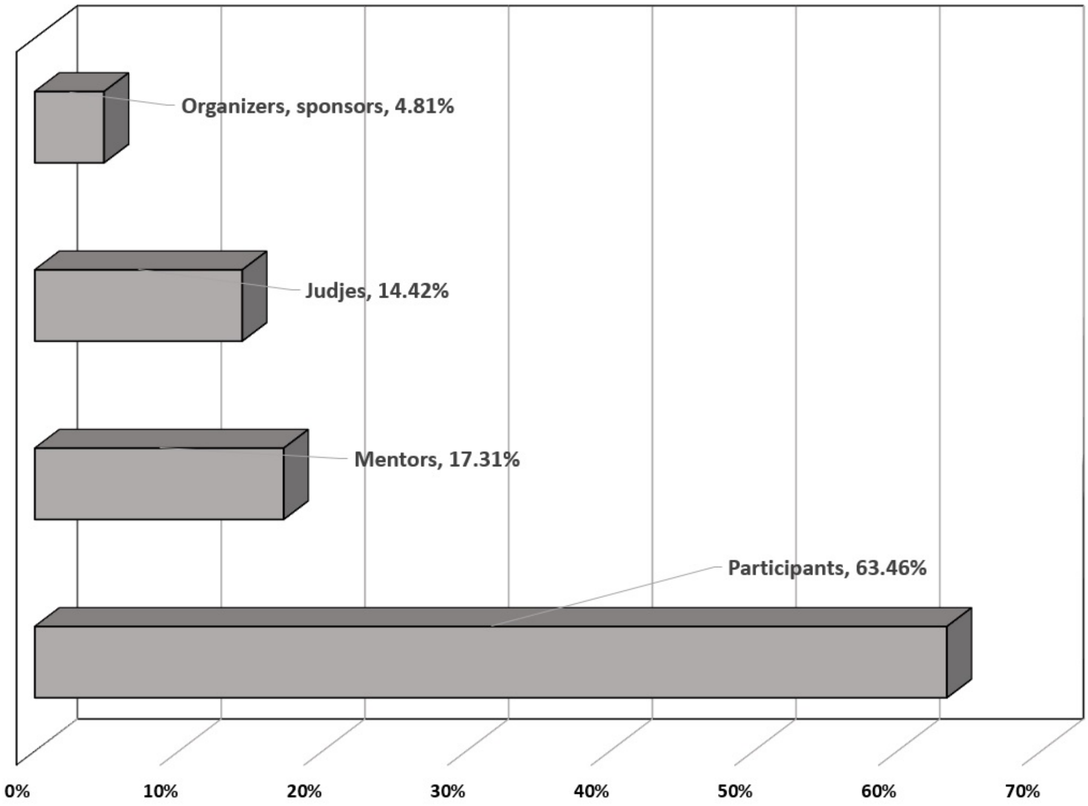

Figure 1: Participants of the survey.

We got the following answers to the question "An important quality of a Hackathon participant?": knowledge and skills in software development – 61,9 % of participants, knowledge of domain area – 28,6 %, teamwork skills – 81 %, ability to work in stressful conditions – 71,4 %, extraordinary thinking – 50 %, other (flexibility, testing skills, etc.) – 2,4 %. Results on answering the question "Do you plan to further participate in Hackathons (in different roles) or assist in hackathons organization?" were "yes, of course" – 69,1 %, "maybe" – 23,8 %, "no" – 7,1 %.

From the results of the survey described below, the hackathon model is presented in figure 3. The purpose of the model is to capture key elements that are essential toward stimulating and maintaining students' interest in computer science and software engineering hackathons.

4. Guidelines for organizing of the hackathon

After questionary and based on the experience of hackathons organizers it is important to consider the following guidelines for organizing the hackathons:

1. *Venue and location.* Hackathons should be inspiring, so the venue and the location of the hackathon (audiences, online space, etc.) are really important. It is important to

ht]

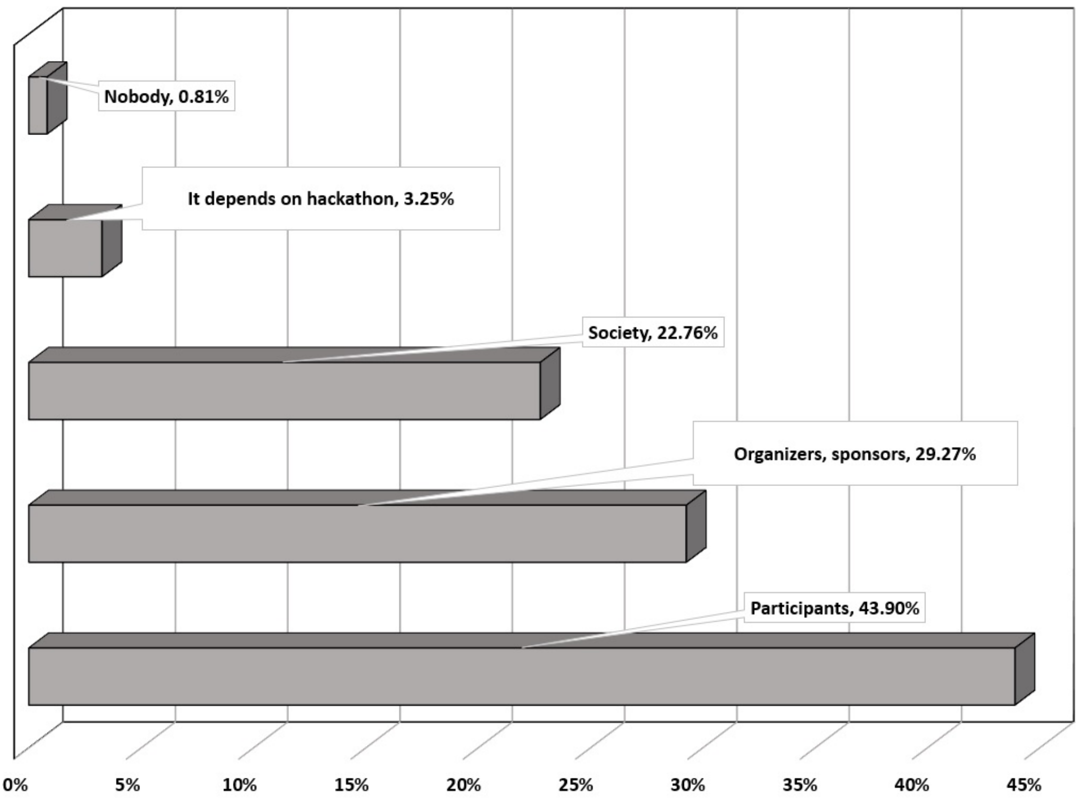

Figure 2: Answers to the question "Who are the main beneficiaries of hackathons".

allocate a specific location for the event and allow participants to fully focus on their tasks during the hackathon with minimal distraction due to external factors. Moreover, areas of relaxation and stress relief are desirable, this is useful for maintaining a creative atmosphere. It is also advisable to ensure that all key players: participants, mentors, organizers, experts, judges are available to check consistency throughout the process. Physical space and resources are essential to a successful collaboration.

2. *Timely information and support of the participants.* Hackathons are usually very tedious, so organizers should take steps to support the participants so that they remain focused, active, and motivated. Discussions and presentations should be limited in time. It is useful to have a well-planned and uncomplicated schedule so that the work on the project develops gradually and progressively. A presentation on the project at the end of each stage and a final presentation at the end of the event will provide an opportunity for a wider group of people (stakeholders, experts, participants) to see the results of the work.

3. *Time.* The level and composition of team members are important. Therefore, it is necessary to choose a time for the hackathon when the participants are not participating in other

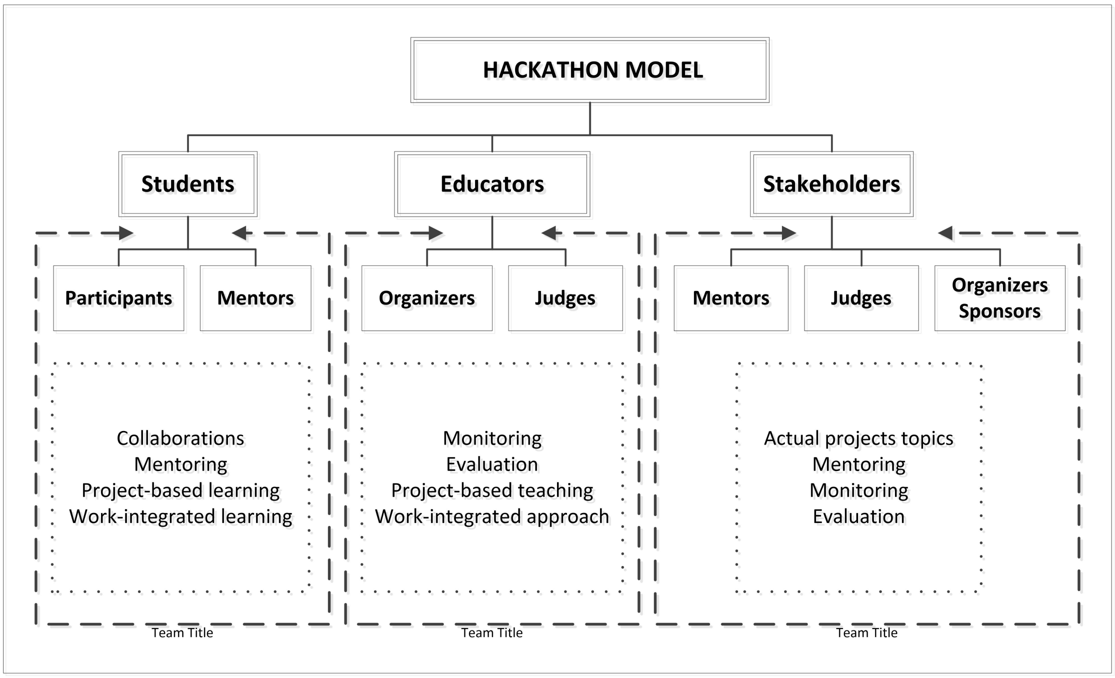

Figure 3: Hackathon model.

events. This creates difficulties in organizing the hackathon but allows you to extract real value from the event.

4. *Teamwork.* Meeting and working closely with other team members involved in the project is beneficial for all involved and is a very productive way to achieve results. At the end of the event, all participants have a better understanding of the problems and tasks they were trying to solve. During the hackathon, there is also a significant exchange of knowledge and technical information between participants, team members. In addition, the hackathon provides an opportunity to work through complex technical problems with mentors, experts, software development industry with diverse backgrounds and technical skills.

5. *Flexibility.* When organizing the hackathon and during the hackathon, it is important to find a balance between setting rigid goals and methods and fairly open and poorly defined objectives. This balance is difficult to achieve during a hackathon as it requires active collaboration. However, flexible formulation of recommendations and methodologies allows for an appropriate degree of flexibility. This approach allows you to combine goal achievement, allows to emerge new ideas, and also develops innovative thinking.

6. *Size of a team.* It has been found that the optimal size of the team is between 5 and 6 team members. If the team is too large, then subgroups naturally from within the team, and there is a danger that the subgroups will start to work separately.

7. *Experts, judges.* At the end of each stage of the hackathon, a meeting is held in which non-team members participate (remotely or in-person) in a session to review and discuss project progress and possible next steps. Experts and the judges are allowed to ask questions or ask about specific points of interest in the projects.

8. *Duration.* It was found that the duration should be close to the maximum limit of effective intensive work. Long hackathons allow for more ambitious tasks than those considered during short-time hackathons (for example, a 24-hour hackathon or weekend). It is difficult to schedule and motivate a large group of participants for a long-time hackathon when they have other education activities or work responsibilities.

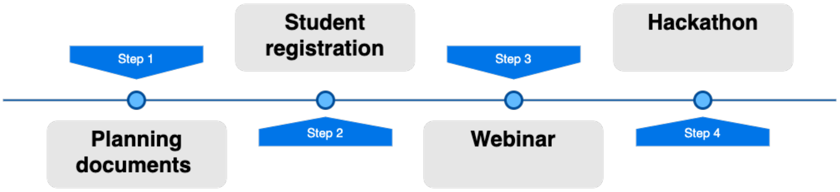

Figure 4: Preparation activities before the hackathon.

Preparation for the hackathon should start 6 months before the event is scheduled. Figure 4 provides an overview of the timeline:

Step 1 – planning documents, mentors' selection, software engineers working in the context of the event.

Step 2 – registration of the students, formation of the teams.

Step 3 – introduction to the event context, event's domain areas.

Step 4 – kick off the hackathon.

Hackathon activities should include the following stages of software development:

1. Preparation
   - The study of the task and the subject area. Definition of the concept of the developed software product. Finding the answer to the question: "What should the product do?"
   - Market research and search for similar products. Highlight the killer feature of the product.
   - Interviewing potential users to identify competitive advantages. Creating a project description.
2. Planning
   - Forming and adding tasks to Project Management services.

   - Study of the technology stack for the project and the selection of optimal solutions.
   - Score tasks by the hour. Highlighting tasks for the MVP version.
   - Forming tasks for increments and sprints using the Scrum methodology. Distribution of tasks between team members.

3. Programming and testing
   - Initializing the project and creating a Git repository.
   - Sprint 1.
   - Sprint 2.
   - Sprint 3.
   - Sprint 4.
4. User Acceptance Testing
   - Verification of the developed product by end-users for efficiency and convenience.
   - Defining the list of tasks that require execution in the next versions of the product.
5. Presentation
   - Presentation creation: project description (in one sentence); product functionality and killer features; problems solved by the product; potential users of the product; the stack of used technologies; plans for further product development.
   - Preliminary (closed) presentation for mentors and jury members. Live demonstration of a workable project.

Completed projects (software products) of Hackathon are assessed by the jury according to the following criteria:

- The quality of the development process is controlled during the project development process;
- Design – estimated the beauty and style of the project;
- Functionality – a set of the declared functions of the software product is checked, satisfying the requirements of the customer's technical specification and the needs of potential users;
- Reliability – estimated completeness of the project, its fault tolerance, and recoverability.
- Security – properties ensuring data confidentiality, their integrity, and availability are checked;
- Convenience – features of the software product are taken into account, which allows the user to easily work with data;

- Efficiency – the ratio of the level of services provided by the software product to the user under specified conditions, to the volume of resources used is assessed;
- Accompanying – estimated ease of analysis of the software, the ability to make changes to the software in connection with changing user needs;
- Portability – adaptability to different hardware environments, installation flexibility is checked;
- Q-factor – the value of the software product, reasonableness, and rationality of the used solutions and technologies are taken into account;
- Presentation – the quality of the performance, demonstrating the developed software product, is evaluated.

5. Technical features of the hackathon and evaluation criteria

To monitor and evaluate the development of the Hackathon's assignments we developed a web-based accounting system. The architecture of this system is presented on the figure 5. A web-based accounting system provides access (limited by category) in desktop and mobile versions to Hackathon's participants, organizers, judges, and mentors.

The system provides the following capabilities:

1) registration and authorization for participants, mentors, judges, etc.;
2) choice of roles in the system: participant, team mentor, administrator, etc.;
3) a description of the terms of reference for the participating team;
4) control and assessment of the passing of the stages of product development;
5) iterative the team and mentor cooperation (receiving and making edits at each stage of development);
6) tracking the status of teams (passed stages, the number of points received).

Discussing the quantitative and qualitative results, it should be noted that the final rating of evaluation by the jury members of the developed software products of the Hackathon was the sum of two components:

1. Rating evaluation of the development process of a software product by a team, taking into account in the web-based accounting system (the highest possible rating is 3000 points) (table 3).
2. Rating evaluation of the developed software jury (the maximum possible rating is 3000 points) (table 4).

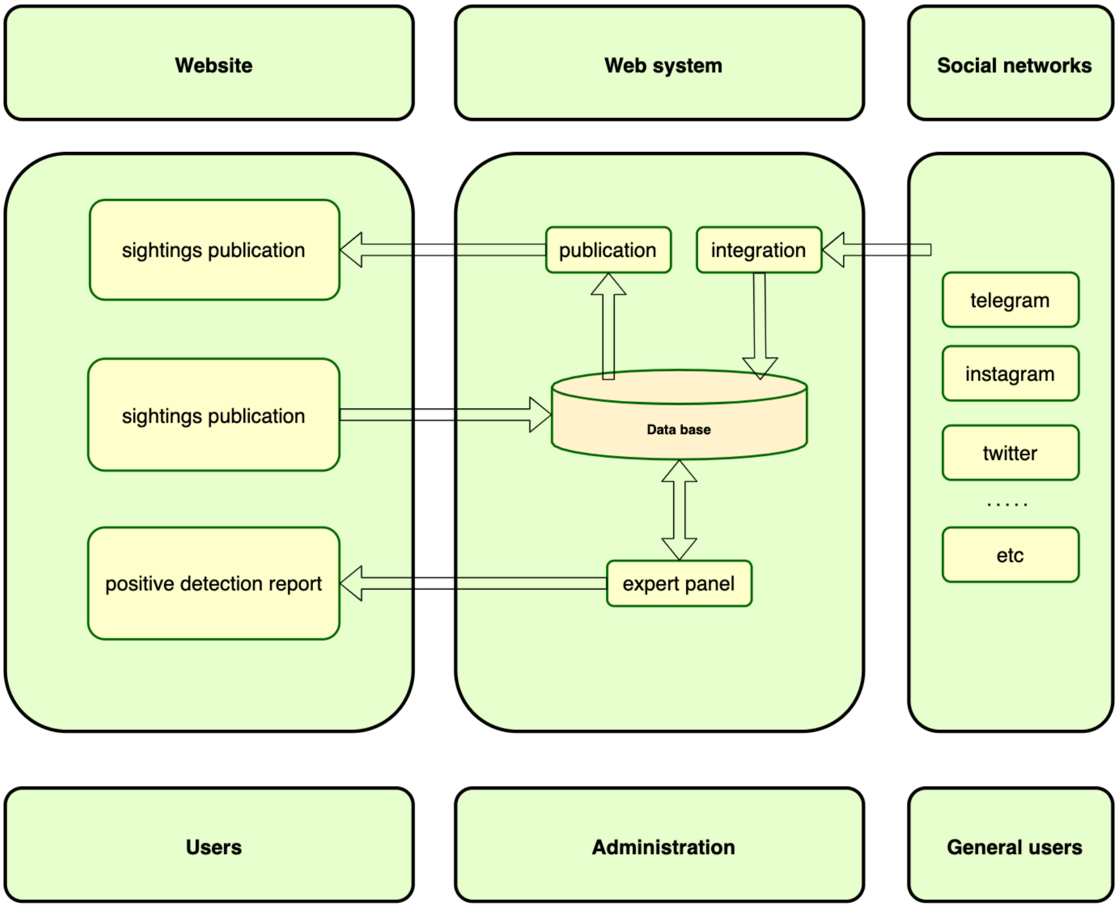

Figure 5: System architecture.

6. Conclusions

Hackathons are growing in popularity in computer science education and encourage the practical ingenuity of putting together some inherent elements that are universally required, often via programming. The value of hackathons is in providing an opportunity for students to meet and create new links in the IT-industry medium. Hackathon is not only a prototyping exercise technically speaking, but it is similarly a prototyping exercise of new working and personal collaborations for the participants.

There are a lot of approaches to the implementation of hackathons in the context of computer science education. We presented the taxonomy of hackathons, the advantages and possible disadvantages of hackathons. Hackathons as pedagogical technology can be used to teach students the necessary skills and competencies for developing programming projects and software applications.

Table 3
Criteria for assessing the quality of the development process (accounting system).

N	Criteria for assessing the quality of the development process	Maximum score
1	Domain area	100
2	Similar products	100
3	Project Description	200
4	Task formation	100
5	Technology stack	100
6	Task highlighting for MVP version	100
7	Forming tasks for increments sprints	100
8	Creating a Git repository	100
9	Sprint 1	400
10	Sprint 2	400
11	Sprint 3	400
12	Sprint 4	400
13	User Testing	100
14	Tasks for future periods	100
15	Presentation development	200
16	Demonstration of the project	100

Table 4
Criteria for assessing the quality of the development process (jury).

N	Criteria for assessing the quality of the development process	Maximum score
1	Design	350
2	Functionality	450
3	Reliability	450
4	Security	300
5	Usability	250
6	Efficiency	150
7	Maintainability	150
8	Portability	300
9	Presentation	300

References

[1] Abduramanov, Z.S., Ibraimov, R.I. and Seydametov, G.S., 2019. Automation of monitoring and verification tools for students on hackathon. *Information and computer technologies in economics, education and social sphere*. vol. 4 (26), pp.79–86.

[2] Abduramanov, Z.S., Seydametov, G.S. and Ametov, O.M., 2017. Project IT-Academy CSE4S: experience of cooperation of the Applied Informatics Department with the professional IT-community of Crimea. *Information and computer technologies in economics, education and social sphere*. vol. 1 (15), pp.144–152.

[3] Bolton, W.S., Ng, S., Lam, A., Kinch, J., Parchment, V., Foster, W.P., Zimmermann, M.R., Teh, J.Q., Simpson, A., Sharma, K., Kerstein, R., Burke, J., Chapman, S.J., Culmer, P.R. and Jayne, D.G., 2021. Virtual hackathon to tackle COVID-19 unmet needs. *BMJ Innovations*, 7(2). Available from: https://doi.org/10.1136/bmjinnov-2020-000456.

[4] Briscoe, G. and Mulligan, C., 2014. Digital innovation: The hackathon phenomenon. Available from: https://bit.ly/3eNcdpo.

[5] Byrne, J.R., Sullivan, K. and O'Sullivan, K., 2018. Active learning of computer science using a hackathon-like pedagogical model. *2018 Constructionism; Vilnius, Lithuania; Research Council of Lithuania*. pp.138–150.

[6] Cobham, D., Gowen, C., Hargrave, B., Jacques, K., Laurel, J. and Ringham, S., 2017. From hackathon to student enterprise: an evaluation of creating successful and sustainable student entrepreneurial activity initiated by a university hackathon. *EDULEARN17 Proceedings*. IATED, 9th International Conference on Education and New Learning Technologies, pp.789–796. Available from: https://doi.org/10.21125/edulearn.2017.1172.

[7] Falk, J., Kannabiran, G. and Hansen, N.B., 2021. What do hackathons do? Understanding participation in hackathons through program theory analysis. *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: Association for Computing Machinery.

[8] Falk Olesen, J. and Halskov, K., 2020. 10 years of research with and on hackathons. *Proceedings of the 2020 ACM Designing Interactive Systems Conference*. New York, NY, USA: Association for Computing Machinery, DIS '20, p.1073–1088. Available from: https://doi.org/10.1145/3357236.3395543.

[9] Filippova, A., Trainer, E. and Herbsleb, J.D., 2017. From diversity by numbers to diversity as process: Supporting inclusiveness in software development teams with brainstorming. *Proceedings of the 39th International Conference on Software Engineering*. IEEE Press, ICSE '17, p.152–163. Available from: https://doi.org/10.1109/ICSE.2017.22.

[10] Frey, F.J. and Luks, M., 2016. The innovation-driven hackathon: One means for accelerating innovation. *Proceedings of the 21st European Conference on Pattern Languages of Programs*. New York, NY, USA: Association for Computing Machinery, EuroPlop '16. Available from: https://doi.org/10.1145/3011784.3011794.

[11] Gama, K., Alencar, B., Calegario, F., Neves, A. and Alessio, P., 2018. A hackathon methodology for undergraduate course projects. *2018 IEEE Frontiers in Education Conference (FIE)*. pp.1–9. Available from: https://doi.org/10.1109/FIE.2018.8659264.

[12] Gama, K., Alencar Gonçalves, B. and Alessio, P., 2018. Hackathons in the formal learning process. *Proceedings of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education*. New York, NY, USA: Association for Computing Machinery, ITiCSE 2018, p.248–253. Available from: https://doi.org/10.1145/3197091.3197138.

[13] Guerrero, C., Mar Leza, M. del, González, Y. and Capó, A. Jaume-i, 2016. Analysis of the results of a hackathon in the context of service-learning involving students and professionals. *2016 International Symposium on Computers in Education (SIIE)*. pp.1–6. Available from: https://doi.org/10.1109/SIIE.2016.7751857.

[14] Horton, P.A., Jordan, S.S., Weiner, S. and Lande, M., 2018. Project-based learning among engineering students during short-form hackathon events. *2018 ASEE Annual Conference & Exposition*. Available from: https://doi.org/10.18260/1-2--30901.

[15] Imam, A. and Dey, T., 2021. Tracking hackathon code creation and reuse. *2021 2021 IEEE/ACM 18th International Conference on Mining Software Repositories (MSR).* Los Alamitos, CA, USA: IEEE Computer Society, pp.615–617. Available from: https://doi.org/10.1109/MSR52588.2021.00085.

[16] Imam, A., Dey, T., Nolte, A., Mockus, A. and Herbsleb, J.D., 2021. The secret life of hackathon code. 2103.01145.

[17] Kollwitz, C. and Dinter, B., 2019. What the hack? – towards a taxonomy of hackathons. In: T. Hildebrandt, B.F. van Dongen, M. Röglinger and J. Mendling, eds. *Business Process Management*. Cham: Springer International Publishing, pp.354–369.

[18] Kopeć, W., Balcerzak, B., Nielek, R., Kowalik, G., Wierzbicki, A. and Casati, F., 2018. Older adults and hackathons: a qualitative study. *Empirical software engineering*, 23(4), pp.1895–1930. Available from: https://doi.org/10.1007/s10664-017-9565-6.

[19] Medina Angarita, M.A. and Nolte, A., 2019. Does it matter why we hack? – Exploring the impact of goal alignment in hackathons. *Proceedings of 17th European Conference on Computer-Supported Cooperative Work*. European Society for Socially Embedded Technologies (EUSSET), pp.1–15. Available from: https://doi.org/10.18420/ecscw2019_ep01.

[20] Medina Angarita, M.A. and Nolte, A., 2020. What do we know about hackathon outcomes and how to support them? – a systematic literature review. In: A. Nolte, C. Alvarez, R. Hishiyama, I.A. Chounta, M.J. Rodríguez-Triana and T. Inoue, eds. *Collaboration Technologies and Social Computing*. Cham: Springer International Publishing, pp.50–64.

[21] Mtsweni, J. and Abdullah, H., 2015. Stimulating and maintaining students' interest in computer science using the hackathon model. *The independent journal of teaching and learning*, 10(1), pp.85–97. Available from: https://doi.org/10.10520/EJC179017.

[22] Nandi, A. and Mandernach, M., 2016. Hackathons as an informal learning platform. *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*. New York, NY, USA: Association for Computing Machinery, SIGCSE '16, p.346–351. Available from: https://doi.org/10.1145/2839509.2844590.

[23] Nolte, A., Chounta, I.A. and Herbsleb, J.D., 2020. What happens to all these hackathon projects? Identifying factors to promote hackathon project continuation. *Proc. ACM Hum.-Comput. Interact.*, 4(CSCW2). Available from: https://doi.org/10.1145/3415216.

[24] Nolte, A., Pe-Than, E.P.P., Filippova, A., Bird, C., Scallen, S. and Herbsleb, J.D., 2018. You hacked and now what? - Exploring outcomes of a corporate hackathon. *Proc. ACM Hum.-Comput. Interact.*, 2(CSCW). Available from: https://doi.org/10.1145/3274398.

[25] Nolte, A., Pe-Than, E.P.P., Obot Affia, A. amefon, Chaihirunkarn, C., Filippova, A., Kalyanasundaram, A., Angarita, M.A.M., Trainer, E. and Herbsleb, J.D., 2020. How to organize a hackathon – a planning kit. Available from: https://arxiv.org/abs/2008.08025.

[26] Pe-Than, E.P.P., Nolte, A., Filippova, A., Bird, C., Scallen, S. and Herbsleb, J.D., 2019. Designing corporate hackathons with a purpose: The future of software development. *IEEE Software*, 36(1), pp.15–22. Available from: https://doi.org/10.1109/MS.2018.290110547.

[27] Pirker, J., Kultima, A. and Gütl, C., 2016. The value of game prototyping projects for students and industry. *Proceedings of the International Conference on Game Jams, Hackathons, and Game Creation Events*. New York, NY, USA: Association for Computing Machinery, GJH&GC '16, p.54–57. Available from: https://doi.org/10.1145/2897167.2897180.

[28] Porras, J., Khakurel, J., Ikonen, J., Happonen, A., Knutas, A., Herala, A. and Drögehorn, O.,

2018. Hackathons in software engineering education: Lessons learned from a decade of events. *Proceedings of the 2nd International Workshop on Software Engineering Education for Millennials*. New York, NY, USA: Association for Computing Machinery, SEEM '18, p.40–47. Available from: https://doi.org/10.1145/3194779.3194783.

[29] Rainville, B., 2015. *Creating value for teachers: Product development at the intersection of education and technology*. Doctoral dissertation. Harvard Graduate School of Education. Available from: https://dash.harvard.edu/handle/1/16645022.

[30] Saravi, S., Joannou, D., Kalawsky, R.S., King, M.R.N., Marr, I., Hall, M., Wright, P.C.J., Ravindranath, R. and Hill, A., 2018. A systems engineering hackathon – a methodology involving multiple stakeholders to progress conceptual design of a complex engineered product. *IEEE Access*, 6, pp.38399–38410. Available from: https://doi.org/10.1109/ACCESS.2018.2851384.

[31] Striuk, A.M., 2018. Software engineering: First 50 years of formation and development. *CEUR Workshop Proceedings*, 2292, pp.11–36.

[32] Uys, W.F., 2020. Hackathons as a formal teaching approach in information systems capstone courses. In: B. Tait, J. Kroeze and S. Gruner, eds. *ICT Education*. Cham: Springer International Publishing, pp.79–95.

[33] Varava, I.P., Bohinska, A.P., Vakaliuk, T.A. and Mintii, I.S., 2021. Soft skills in software engineering technicians education. *Journal of physics: Conference series*, 1946(1), p.012012. Available from: https://doi.org/10.1088/1742-6596/1946/1/012012.