Integrating Hackathons with Software Engineering Courses: Best practices and Lessons Learned

Amir Zeid
Department of Computer Science and Information Systems, American University of Kuwait

***Abstract*—**

Hackathons are programming marathons. There are different types of hackathons: hacking for a good cause, hacking for company members to improve team communication among other types.

In this paper, we explore how hackathons can be used to improve the learning experience of software engineering students. We will reflect as well about the importance of multidisciplinary approach in team formation and how it can contribute to students with different backgrounds. We will give real examples from our experience in integrating a hackathon with software engineering course. The hackathon activities help to fulfill the learning outcomes of the course. Success stories and best practices will be discussed.

***Index Terms*—Software Engineering Courses, Integrating contests with Software Engineering.**

I. Introduction

Hackathons (or programming marathons) have been increasingly gaining popularity among technology circles as a way of breeding new ideas, and promoting prototype production in a less risky and a more entertaining setting. Hackathons were originally introduced by companies in the 2000's to encourage innovation and creativity in developing new products. But, shortly afterwards it expanded to include other sectors and public domains, and to involve more communal engagement. In recent years, new branches of hackathons have emerged to particularly address issues of public interest, like addressing governance and public life or supporting a pressing social or environmental cause. These have been commonly referred to as civic hackathons.

Although hackathons are held nowadays in less formal and more entertaining settings, their intensive nature (requiring completion of a project within 1 to 3 days) elicits stressful moments, team conflicts, and sometimes cultural and gender tensions. This has drawn the interest researchers to examine the socio-cultural aspects of hackathons as well as their learning outcomes. The three-day event can offer insightful observations into inter-team interaction patterns that can aid research in the field of collaborative software development.

This research was conducted at American University of Kuwait (AUK). AUK offers software engineering course as a required course for Computer Science, Information Systems and Computer Engineering degrees. The course is offered annually in both semesters (Fall and Spring) for at least 100 students per year.

In this paper, we document our experience with hackathons in improving the learning outcomes of software engineering undergraduate courses. We also discuss the socio-cultural factors that come into play during the process, and how they can be addressed to minimize their negative impact on the learning process. In our discussion, we draw examples from an environmental hackathon that we organized to come up with solutions to pressing marine life issues around the world and within our local community. In light of these examples, we will underpin some challenges and learnt lessons that have been compiled over two years of holding the event.

The remaining of this paper is organized as follows:

- Section 2 includes background about course and environment where the course is offered.
- Section 3 discusses the structure and format of environmental hackathons that have been adopted.
- Section 4 discusses learning outcomes of software engineering courses and how hackathons can be mapped to them
- Section 5 discusses success stories and lessons learnt out of hackathons
- Section 6 concludes the paper, summarizes the contributions and points to future enhancements.

II. Background

A. Software Engineering Course Learning Outcomes

CSIS-330 (Software Engineering) is associated to ABET accredited programs at AUK. Consequently, the main learning outcomes of the course are related to ABET requirements:

1. Describe the main requirements and design for software projects using UML [ABET 2]
2. Make the right decision on the choice of tools to develop a medium sized project [ABET 6]
3. Select the right process model for software projects [ABET 6]
4. Understand the importance of managing requirements in software development. [ABET 2]
5. Understand the importance of documentation and manuals, their structure and production. [ABET 1,6]

978-1-5386-1174-6/18/$31.00 ©2018 IEEE

6. Deploy and effectively use a CASE tool. [ABET 6]
7. Design, Develop and debug a medium sized software project [ABET 2]
8. Function effectively in a software development team [ABET 5]
9. Present technical requirements and design presentations. [ABET 3]
10. Understand the applicability of ACM-IEEE Code of ethics [ABET 4]

B. ABET Requirements

The following are the ABET requirements for Computer Science and Information Systems [1]:

1. Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions.
2. Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline.
3. Communicate effectively in a variety of professional contexts.
4. Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.
5. Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline.
6. Apply computer science theory and software development fundamentals to produce computing-based solutions.

C. Software Engineering Course outline

Table-1 illustrates the course outline of the software engineering course.

TABLE I. COURSE OUTLINE

Week 1-2	Introduction to Software Engineering
Week 3-5	Process Models, Evolution and Validation Activities: Simulation Games to learn different process models
Week 6-7	Requirements Specifications Functional Requirements Non-Functional Requirements
Week-8-9	Requirements Presentations
Week 10-14	OO Design using UML Class Diagrams Sequence Diagrams State Diagrams Activity Diagrams
Week-15-16	Final Presentations and Demos

D. Grade distribution

Typical grade distribution of the course is as follows:

- Group activities 10%
- 2 Exams 40%
- Project (Forming Groups, Proposal and presentation, First Presentation and SRS Document, Design Presentations and Documents, Projects Demos and Final Documents (Last week) 20%
- Final Exam 30%

III. HACKATHON FOR ENVIRONMENT

The hackathon mainly takes place over two three days (weekend). The following specialized hackathon was chosen to experiment with. The Fishackathon (Fish Hackathon) is an international initiative by the U.S. Department of State that aims to raise awareness about using technology to solve marine related problems [2]. The event is organized simultaneously in different cities across the globe. Department of State and its supporting organizers submit the problem set to participating sites. Each participating site has the right to include local marine related problems to the problem set. Teams have the right to pick one or more of the existing problems and try to solve it using technology (mobile phones, drones, or any other relevant technology). We successfully organized the Fishackathon twice (2016, 2018). The hackathon includes the following activities:

A. Call for Participation

General call for participation using social media and university visits to promote the event. The following categories of participants are encouraged to apply:

- IT Gurus: participants with at least two years of professional experience in programming. Their main role is to lead the teams technically.
- Junior Programmers: typically a second or third year student majoring in computer science or engineering.
- Graphics Designers: participants with artistic background to come up with logos and theme designs for the projects.
- Marine biologists: participants with background in environmental issues (mostly marine biology students).
- Business analysts: participants who can develop business cases and feasibility studies.

B. Idea explanation

Participants gather in the first night (typically a Friday night). After formal introductions, the proposed problem set is explained by domain experts. In this particular instance, marine biologists and experts explain the problems. Each problem is simplified and related to the marine environment of the country.

C. Team Leaders pick ideas

IT Gurus (team leaders) are identified. Each team leader picks an idea to work with. He/she then takes the stage to

introduce himself, his background and why he chose this idea to work on. It is the responsibility of the team leader to recruit his team members. However, organizers may interfere to make sure the formed teams are homogeneous to guarantee a fair experience for all participants.

D. Forming a team

Junior participants and other members join team leaders according to their interest. The team leader has the right to accept team members based on their background and the nature of the project. Each team will have at least:

- 1 experienced programmer
- 2 junior programmers
- 1 graphic designer
- 1 marine biologist
- 1 business analyst

The idea is to have a diverse group of participants in each team to create balance between the teams. This will also provide resources to cover the project from different aspects. The diverse nature of team members goes along with supporting liberal arts education and multidisciplinary research. Team formation supports student learning outcome 8.

E. Developing a prototype

The teams work together to develop a prototype to solve one of the identified problems. The teams work from 8 am to 10 pm on the second day. On the third day, teams submit their work and presentations around 5 pm. Coders can develop the prototype using any existing open source technology (no restrictions to using certain tools like some commercial hackathons) It is recommended (but not restricted) that a submitted application run on one of the following platforms: Android devices, iOS device or windows device.

F. Presentations

Presentations are given by each team at the end of the hackathon. Each team gets 5 minutes to present the solution and approach. Presentations are conducted in front of a specialize panel of judges that includes programmers, scientists, business owners and marine researchers. Questions from judges follow the presentations. Presenters are free to use any tool to present their solution as long as they submit a pdf file for the presentation. The presentation follows certain template. The teams receive the template and rubrics for the evaluating the product one day before their presentation. The template includes: problem statement, used resources, possible ways for implementation and different scenarios to use the developed solution. Teams are encouraged to design their own artwork. Some teams opt to produce short videos about their solution. Teams are also encouraged to provide business models to make their solution sustainable.

G. Deliberations and announcing winners

Following the presentations, judges deliberate to decide on the winning three teams. The criteria [2] has 0-3 scale for each category. Participants are evaluated based on the following issues: Quality of the idea/innovation, Impact, User experience of the developed solution and the creativity of the presentation.

IV. Activities of the Course

Course activities include using simulation games to introduce the concepts of difference process models like waterfall, spiral, incremental and rapid prototyping [3]. During the course of 16 weeks, students have to develop a medium sized project. Hackathons are used to introduce the concept of extreme programming in a practical way. The advantages and disadvantages of extreme programming are discussed in classes. Participating students are encouraged to share their experiences with class mates after the event. The course is aligned to other contests of different nature to address the learning outcomes [4].

A. Aligning the Hackathon with Course Activities

1) Developing a prototype

- Teams have freedom to choose whatever development tools and systems analysis tools to use in the contest (outcome 2, 6)
- Teams have freedom to choose the relevant process model (outcome 3)
- Developing the prototype (outcome 7)

2) Final presentation

- Each team has 5 minutes to present their idea in public and in front of a diverse groups of judges (outcome 9)

V. Challenges and Solutions

Most of the challenges we faced were due to the fact that the concept is new to Kuwait. The following are the challenges and solutions.

A. Promoting the event

We mainly used social media outlets to invite participants. We also had campus tour to encourage students from different universities and schools to participate.

B. Funding

Funding a first of its kind event is challenging. We approached private sector companies, government organizations and universities. Organizing the first event, we only had two sponsors (private sector). As for organizing the second event (after having a success story), we had around 5 sponsors.

C. Mentors from different domains

Finding qualified mentors who are willing to volunteer their time to help the teams was a real challenge. We had to secure at least 1 mentor in each domain (computing, graphic design, business and marine biology). We depended mainly on volunteers from participating universities and organizations.

D. Turning projects to products

Some of the produced projects were ready to turn to products if proper funding is provided. Unfortunately, up to this point, we did not manage to establish a concrete link to turn the ideas to concrete businesses. Efforts with local organizations are taking place to turn the ideas to reality.

E. Cultural barriers

Kuwait is a conservative country that observes gender segregation in many activities. Percentage of females studying computing in Kuwait is considerably high compared to North America and parts of Europe [5] . Due to this fact, forming teams during the hackathon is challenging. To optimize team communication this issue has to be taken into consideration [6]. Some females prefer to work in all female teams due to culture and tradition.

VI. Success stories

This section includes success stories from two instances of organizing the hackathon in 2016, 2018.

A. First Instance (2016)

- Number of participants: around 40
- Number of projects 6
- Winning teams ideas included global ghost gear initiatives and using image processing to recognize fish samples.

B. Second Instance (2018)

- Number of participants: around 70
- Number of projects 10
- Winning teams ideas included using drones and image processing to identify environmental hazards and monitoring illegal employment in fishing boats.

C. Performance of students who participated in the event

Students who participated showed consistent improvement in course exams and class project. Topics covered in Exam-1 are: process models (including extreme programming), basic software engineering concepts. Exam-2 covers requirements specifications management, team management issues. Topics in both exams are directly related to Hackathon activities. The following figure depicts comparison between Hackathon's participants and the rest of the class.

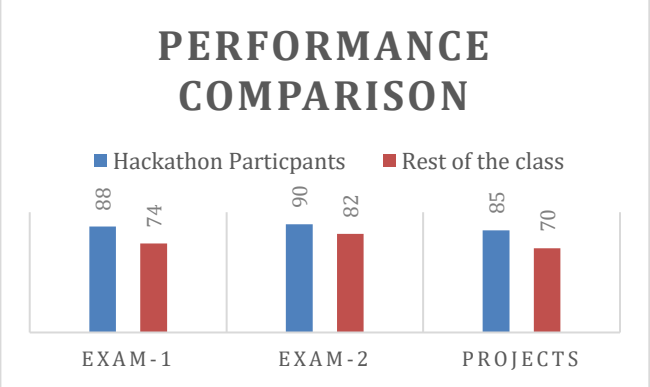

Fig. 1. Performance comparison

D. Achievement of Learning outcomes

Students have to fill two surveys at the end of the semester: evaluation of teaching effectiveness and learning outcomes surveys. They also have the right to comment about the course and its activities in a free form. It has been noticed that the achievement of learning outcomes responses improved after the participation of students in the hackathon. In addition, many positive comments were received from participants. The numerical value for student evaluation of teaching effectiveness for instructors improved in the semesters where the event is organized.

VII. Conclusion

In this paper, we presented a model to integrate international hackathons with software engineering course. The learning experience the students took back was much appreciated. Students who participated in the hackathons frequently got better scores in their course project and related exams. Many of them proceeded to start their own business after succeeding in meeting investors in such events. Participating in the hackathons fulfilled most of the learning outcome of the course in a fun and creative way. Integrating hackathons to software engineering courses proved to be a valid teaching tool to enhance the delivery of most typical learning outcomes of senior courses in computer science programs. Participating in such events contribute to the improvement of students' performance in Capstone projects as well [7].

VIII. BIBLIOGRAPHY

[1] M. Oudshoorn, S. Thomas, R. Raj and P. Allen, "Understanding the new ABET Computer Science Criteria," in *49th ACM techincal symposium on computer science education - SIGCSE 2018.*, 2018.

[2] "Fishackathon 2018," Hackernest, 1 1 2018. [Online]. Available: http://fishackathon.hackernest.com/. [Accessed 1 5 2018].

[3] A. Zeid, "Using simulation games to teach global software engineering courses," in *Frontiers in Education (FIE)*, 2015.

[4] A. Zeid, "A framework to evaluate software engineering student contests: Evaluation and integration with academic programs," in *ICSE.2013*, San Francisco, 2013.

[5] A. Zeid and R. El-Bahey, "Women in computing A case study about Kuwait," in *IEEE Frontiers in Education Conference (FIE)*, Oklahoma City, 2013.

[6] A. Zeid and R. Elbahey, "The impact of introducing single-gender classrooms in higher education on student acheivement levels: A case study in software engineering courses in the GCC region," in *Frontiers in Education (FIE)*, 2011.

[7] A. Zeid, "Integrating International students' contests with computer science capstone: Lessons learned and best practices," in *Frontiers in Education (FIE)*, 2012.