Hackathons as a Formal Teaching Approach in Information Systems Capstone Courses

Full Paper
SACLA 2019

Walter F. Uys[1,2][0000-0001-7709-9326]

[1] SDL Research Focus Area, North-West University, Mahikeng, NW 2745, SA
[2] CITANDA, University of Cape Town, Cape Town, WP 8000, SA
Walter.Uys@nwu.ac.za

Abstract. Hackathons are hack[ing mar]athons where participants collaboratively and rapidly prototype new applications (apps) over a 24-48 hour period. The potential of hackathons as an informal strategy for stimulating interest in the CS fields is well established. Their application as a formal teaching strategy in the CS/IS curriculum is less prevalent. This paper reports on the introduction of such a closed hackathon in a third year IS capstone course at a South African University. An exploratory case study method was used to evaluate the feedback from the participants and organisers. In the process, the students completed seven novel apps which they had started during the course. They also learned about new technologies and programming interfaces (API's) as well as exhibited growth in personal and inter-personal competencies. Seven fundamental differences between curricular and traditional hackathons are highlighted in this paper. Some suggestions for integrating hackathons in the undergraduate CS/IS capstone course are provided together with possible areas for further research.

Keywords: Capstone Projects, Computer Science, Hackathons, Information Systems, Software Application Development, Teaching Approach, Undergraduates.

1 Introduction

Application development is one of the fastest growing high-paying careers in the United States [1]. Because software development environments are free and easily accessible, there is a perception that anyone can become a developer without any formal education or training [1]. Contrasted against this growth in demand for application developers, is the concerns as to the value of a University degree versus practical experience [2–7]. Thus, there remains concerns as to the workplace readiness of graduates as well as a general lack of soft-skills that are needed in the workplace [8–10].

In HEI's there is a fair understanding of this need, and the capstone course is one of the strategies that are used to fill this gap [6]. Capstone courses provide students with the opportunity to integrate theoretical and practical aspects of the curriculum in such a way as to develop a real-world project that has some benefit to society [11]. There are different models of capstone courses ranging from limited support and no classes (the traditional model) to clearly defined deliverables, extensive tutor/lecturer support and scheduled classes and/or meetings [12]. Most courses find a balance between these two.

Because capstone courses are mainly student-driven through 'learning by doing', the role of the lecturer and lectures are less clear. The role of the lecturer is to transition students from academic/theoretical studies towards real-world professional practice. Some guidance in this process is useful, however indications are that fostering a real-world environment that encourages active learning strategies has greater benefit than the minimally guided approach.

There are many active teaching/learning strategies that can be implemented as an instructional design in CS/IS to narrow the theory/practice gap and develop some of the 'soft-skills' that are so desperately needed by industry. The primary strategy adopted in capstone courses is referred to as project-based learning (ProjBL) [13, 14] which is not to be confused with problem-based learning (PBL) [15]. Some other strategies that can be used are experiential learning [16], work-integrated learning [17], case-based learning [18], game-based learning [19, 20] and virtual learning [21]. Although these strategies share some commonalities, project-based learning emphasises an educational strategy for implementing a reflective practice that is aimed at solving real-world project-based problems [14].

One under-represented approach for implementing ProjBL in CS/IS capstone courses is the hackathon. Whilst colleges and universities are frequently the preferred setting for hackathons [22–24], they are mainly used as an informal approach to expose the youth to CS/IS and leverage their creativity [25, 23, 26]. There remains a paucity of reports of hackathons as an active teaching approach in capstone courses. This means that educators have minimal guidance/tips or techniques for introducing hackathons in the capstone course.

This research questions about how hackathons can be used as a formal educational approach for implementing ProjBL in the CS/IS curriculum. The secondary research questions examine how this approach can be integrated in the normal capstone curriculum as well as exploring the differences between instructional design in a traditional project-based classroom versus integrating a hackathon as a formal approach in the classroom.

This paper reports on the experiences of introducing a curricular hackathon in the third year IS capstone course at a South African Higher Education Institute (HEI). An exploratory case study method was used to examine the empirical context. Supporting material was provided by means of first-person experiences, reports by the organisers, student feedback and student evaluations. These reports were analysed according to the principles of dialectical hermeneutics.

2 Central Concepts

Hackathons originated in 1999 from the voluntary efforts of programmers in order to develop/advance a free/opensource operating system called OpenBSD [27]. Although the concept of a hackathon is in need of a formal definition [28], the following working definition was adopted:

> *"Hackathons are events where computer programmers and others involved in software development, including graphic designers, interface designers and project managers, collaborate intensively on software projects in a short period of time, typically 24-36 hours"* [29].

There are many different kinds of hackathons each with its own unique approach. Some are referred to as data dives, codefests, code jams, hack-days, sprints, edit-a-thons [30], data-thons [31, 32] and game labs [19]. Over time these events have increasingly become sponsored by corporations such Facebook [33], F-Secure [28] or KPMG and by Governmental agencies such as GovHack [34], CivHack [35] or NDPHack [36]. These sponsorships have transitioned hackathons from their philanthropic roots to become more competitive with teams keeping their innovations closely guarded until they are presented for adjudication [27]. Motivation for participating in hackathons is mainly for financial gain, personal development, having fun or *"the opportunity to meet new people while learning and experimenting with technologies* [28].

Hackathons promote innovation in product and application development, new uses for existing products or apps or new solutions for government, business or education [37–40, 26]. Ideas or innovations may be bottom-up or top down [28] i.e. originating either from the developers or from senior management, thereby fostering an entrepreneurial approach. They typically take place over extended and focussed periods of between 24 and 72 hours in dedicated venues [29]. Participants remain primarily at the venues with limited breakaways for ablutions or eating and optional sleeping during such events [38] although there are reports of virtual participation [22]. Catering such as food, energy drinks and coffee is normally provided [28] and infrastructure such as computers and data projectors may be available, although participants are normally encouraged to provide their own laptops or hardware devices [41].

Hackathons can either be closed or open events. Closed events are internal to organizations [28]. Open events are organised as public or civic events that are open to everybody [35]. Open events are broadly publicised and attract large corporate sponsorship to encourage attendance and participation [28]. Open events mostly focus on a specific topic or theme such as health and fitness [28], healthcare [26, 42], Internet of things (IoT) or wearable devices [41], whereas closed events might be geared around a new product feature or innovations for a specific company such as Facebook [33, 28].

The targeted participants for hackathons are mainly software developers and technical personnel, although teams may be comprised of programmers, analysts, designers, subject-matter experts (SME's), managers and community representatives [38]. Team sizes can vary between three and five people, with anything from five to fifty teams competing at a particular event [24]. Hackathons can also specifically target under-represented groups such as women and historically disadvantaged individuals (HDIs) [43, 28, 44]. There is also general consensus that (good) programmers are a scarce resource at hackathons [27]. Furthermore, it is acknowledged that financial and material support by leadership is important to hosting such events [38].

Hackathons are able to provide "new and exciting opportunities for education and research" and are able to develop project management and communication skills as well as creativity and innovation amongst participants [19]. They are, however, known to restrict sound architectural approach to development and provide for limited testing and implementation opportunities [23]. It is also questionable how much new (programming) skills can be acquired during such a short event, as the perception remains that participants rely on familiar skills and technologies during hackathons [27].

2.1 Related Work

Although there is a lack of published reports of formal hackathons in the undergraduate (UG) curriculum, there are some related studies that can provide insights on how to introduce these in the curriculum.

Rennick et. al [23] leveraged the principles of hackathons to address student learning outcomes in a first-year Engineering program. Even though their intervention was targeted at achieving course-level objectives, the approach that they followed were 'design days' which bear minimal resemblance with informal hackathons. The main outcome was new designs and not software applications. The purpose of the 'design days' was to improve collaboration between students and staff, expose students to engineering design concepts, integrate knowledge from accross the curriculum and stimulate creative thinking. The primary curriculum outcomes of the 'design days' were improved teamwork, better understanding of design, and greater student engagement. Non-curriculum outcomes were a highly creative and fun/engaging event for the students and increased motivation to participate in such events.

A related study which was hosted at a South African University in conjunction with a research institute [22], examined the efficacy of the hackathon approach to stimulating students enrolments and interest in CS. Six events were hosted over three years based on the typical hackathon format. Participation was voluntary and anyone (not only CS students) were invited to participate in the event. The hackathon gave students the opportunity to learn and network with SMEs and to be part of larger project teams that are focussed on rapidly developing socially relevant solutions. The primary outcomes raised students' exposure to mentoring, work-integrated learning, and collaborative learning. Limited integration with the formal coursework was achieved due to the open nature of the hackathon. The curricular benefits were latent, with students reporting an improved social and practical understanding of CS concepts, further developing their interest and passion for participating in the field and of changing their perceptions of CS.

As can be seen from the basic definition of a hackathon, and how they have been used in higher education based on two exemplary case studies, the reader will agree that the question remains of how a hackathon can be integrated in the curriculum. The following section will review the materials and methods for this case study and then proceed with a description of the case study context.

3 Methods and Materials

For this research, an exploratory case study approach was adopted [45]. This approach is particularly suited for novel studies where the experiences of participants and the context of action is important [46]. This case reports on normal classroom activities and student evaluations. In compliance with the ethical procedures of the university, no interviews or surveys were performed. Participants' names and identities have been kept anonymous. The case study was supported by first-person reports and reflections, reports by the hosting organisation, student feedback and evaluation.

The case study evidence was analysed using dialectical hermeneutics [47]. The method is based on topical analysis [48]. It allows for the analysis of key topics or issues in a corpus of text, discourse or particular event. Topical analysis functions by providing a method of comparison for analysing similarities and differences between related topics, definitions, artifacts or concepts [49].

4 Case Study

There are sufficient commonalities between the traditional hackathon and the curriculum hackathon such as we implemented here for us to refer to it as a hackathon.

4.1 The Aims and Objectives

Some of the main similarities between the two are that the event was a focussed event that occurred at a particular venue (a computer lab on campus), over a fixed period of 24 hours. Catering, coffee and drinks as well as PCs were provided and attendees had access to a kitchen and ablution facilities. Students and lecturers were encouraged to stay awake and present at the venue for the entire period although there were some exceptions.

The event was scheduled over a Friday/Saturday and timed closer to the end of the second semester (19-20 October) so that there were no conflicts with other courses, assignments, tests or exams. After the hackathon, the students would then have 2-3 weeks to complete the documentation/reporting for examination purposes. Students were divided into seven teams of three students each. There were different roles in each team such as project manager, analyst, developer.

The emphasis at the hackathon was the delivery of a working system and the documentation for the system was drafted afterwards for assessment purposes.

Although there were no incentives offered at the hackathon, students were informed that the top three teams would be selected to participate at the national SITA NDP2030 hackathon [36] which itself had a prize of R100K ($6000) for the winning team.

4.2 The Hackathon Phases

The typical hackathon can be represented by means of the classical IS Input, Process and Output model [28]. The input phase is the pre-hackathon phase where ideation and team building take place. The process phase is the actual hackathon where intense hacking occurs and results are demonstrated. The post-hackathon phase is where teams decide to continue with the idea, form new teams or grow the teams and adopt new technologies and develop plans for funding.

In our case, the pre-hackathon phase was approximately 12 weeks. During this time, the students formed their teams, conceptualised their ideas, developed a business case, designed their apps, started building them and elicited requirements from other stakeholders. One of the teams was also responsible for planning the event and had to facilitate the event t-shirts, catering and permission for the event. The week before the event, the preparations began in earnest and all the students were involved in final preparations for the event. On the final days before the event, the coffee/tea/cups and barbeque were purchased and the catering orders confirmed.

The hackathon event started at 18h00 (with a planned start of 17h00) on the Friday, and after initial presentations and motivation by the organising team, the students had dinner. After dinner the students hacked for 4 hours and presented their progress to the entire forum at 24H00. After the presentations the students had snacks and then continued to hack till 5AM where they presented again to the forum.

Departing from the traditional hackathon approach we had a morning exercise routine on the second day where some physical games were played by the students in order to energise them after a long night of intense coding. Some of the lecturers and facilitators attended a Colour Run that happened to be on during this time. After exercise and ablutions, the students started hacking again from 8H00 till 12H00. At 12H00 the students had lunch and hacked again from 13H00-17H00 at which time they submitted their final projects.

> *"the results that we reached at the hackathon were amazing we managed to get most features of the application working during those 24 hours."* (TLM)

At the conclusion of the formal hackathon activities, students, lecturers and facilitators were treated to a typical South African barbeque (Sish Nyama) at the Soccer Institute. Security requirements were that no outside students were allowed at the event, no loud music, no alcohol and that activities needed to end at 21H00. At any rate, the students were so exhausted by that time that they were glad to pack up and go to their respective residences by then.

During the post-hackathon phase, the students made the final changes to their apps in order to capture screenshots for their project report and also to prepare for the final assessment presentations/demonstrations. They were also tasked to develop an individual report outlining their contributions. For this report they were advised at the beginning of the semester to keep a record of their tasks and activities. The final demonstrations and presentations were held four weeks later, and it gave the students the opportunity to present their solutions to invited representatives from government, industry and the University.

4.3 Projects

During the first semester, students were tasked to develop a project plan that included the business case, user requirements, project scope and costs as well as high-level designs and GANTT charts. In the second semester the students started implementing these system development projects. These ideas were initially conceptualised by the students and implemented through various iterations and interactions with the lecturers and stakeholders.

Table 1. App Development Systems/Teams.

Team/App	Description
Residence Control System	Caters for monitoring and managing visitors access to student residences in order to avoid ‘illegal’ squatting.
eLicence App	A mobile app that registers and represent a virtual drivers licence. It also allows security officers to validate physical and virtual licences as well as check for outstanding fines/expired licences. It has a front-end and back-end app.
Soapy Shine Car Wash	A carwash ‘loyalty‘ app that allows members to check availability/queues, book and be notified at any of the participating carwashes.
Billboards Innovation	An app that facilitates the remote management and hosting of adverts on electronic billboards.
Stay Residence Booking System	An ‘Airbnb’ for student accommodation. Intended mainly for finding and booking of ‘approved’ university ‘digs’ or accommodation.
Virtual Housing Project	A VR system that converts 2D plans to 3D virtual walkthroughs in order to visualise the property/architectural design.
Clinic Appointment App	A queuing system for government hospitals and clinics such as those found in banks/mobile service providers. Allows for segregation of emergency/clinic/hospital and dispensary patients.

It was only during the hackathon that the students completed and presented their final systems as outlined in Table 1.

4.4 Outcomes

During the hackathon, students learned how to work with new technologies, tools and software development platforms e.g. they learned how to develop in Java on

Android Studio, database development and using technologies such as XAMMP, PHP, umajin, cross-platform development architectures and the use of APIs such as Googles authentication features and maps in Android Studio as well as interfacing with bulk SMS providers that were not part of their under-graduate coursework.

> *"I got to understand more about cross-platform developmentI did more research on it* ***during the hackathon*** *as we were busy with the coding and development. I also learned more about APIs (Application Programming Interfaces) as we had to do research on how were going to integrate them into our application to work better with other existing applications (i.e. Google Maps)"* (KJT).

Students provided some insightful comments as to the efficacy of the approach through entries in the University administered anonymous Student Experience Survey that was completed online at the end of the semester.

Table 2. Learning experiences and suggested improvements.

What did you like about the teaching and learning experiences of this module?
To communicate and be able to work in a group, to participate in class. This class prepares us for the life outside university. (ST1)
The experiences were great, it taught us about our pre-professional lives. (ST2)
It also taught me to grow as an adult and be more responsible. (ST3)
The learning styles used. (ST4)
Creativity (ST5)
It is like doing real events that affects my life in a very tangible way. (ST6)
What suggestions do you have to improve the teaching and learning experiences of this module?
This module needs time, to be able to complete the project, as there is so much to do. (ST7)
Introduce the work integrated learning strategy in other modules.(ST8)
Every group must be assigned a supervisor. (ST9)
The use of study guide. (ST10)
Nothing. (ST11)
More practicals, field trips and recruits of sponsors and for practicals like Microsoft, hackathon and other IT related companies. (ST12)
The class experience was very new, I don't think there is much more that I can add. (ST13)
Its perfect! (ST14)

What was surprising were the unintended soft-skills that the students developed during the process [10]. They learned a lot about teamwork, project management skills (the first semester complement of this module), time management under pressure, punctuality, responsibility, creativity, bug-fixing and presentation skills.

> *"The project on its own requires that each individual has to play a role in making progress. There is no time for dependency. Teamwork pays off, but most importantly, the ability to communicate with other people is very essential when you are working on something so volatile."* (ZAN)

Although these skills are not considered to be the main outcomes of an IS course [50], they are recognised as critical for the successfull integration of graduates in the workplace [51, 7]. It is thus important to note that the hackathon can be regarded as a suitable active learning approach for developing these soft-skills, which in practice are hard to teach.

As lecturers we found that the event provided students the opportunity to focus purely on the completion of their projects without other distraction as well as provide personal challenges to stay awake for the duration and to work under pressure.

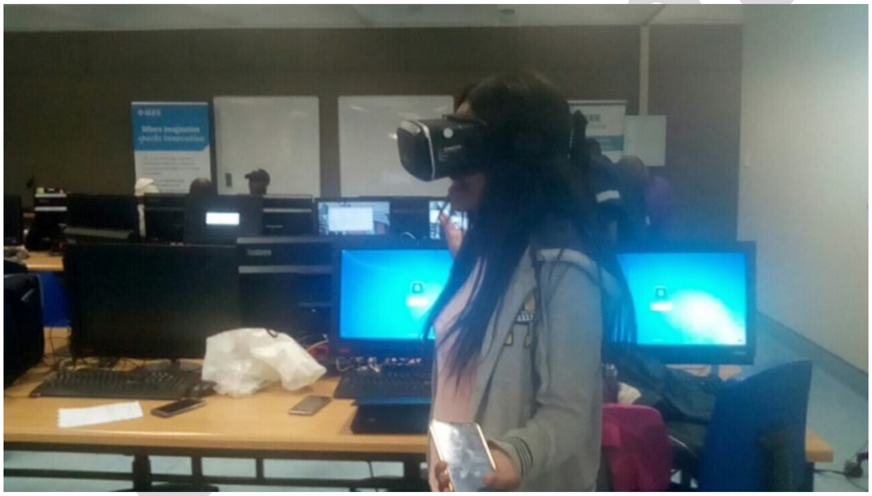

Fig. 1. Demonstrating the VR Housing project during the hackathon.

Without a suitable control group, it is difficult to assess the effect or outcomes that the hackathon had on the quality of the students' solutions, and what they learned during the process as compared to the traditional approach. Even the students' marks would not provide a fair representation of learning as the students in other years and the projects would be different. Assessing software development progress in industry is an ongoing challenge [52] and related metrics such as lines of code, function points or completed features could be used to assess the value that the hackathon had on the students' projects progress. Ultimately, the measure of the success of such an approach would be to track these students to industry and see how they adapt in practice.

5 Discussion

Some similarities/differences between the curricular and the traditional hackathon are explored next.

5.1 Curricular Hackathons

Scope and purpose: The scope and purpose of the curricular hackathon is a lot narrower than the traditional hackathon, which typically takes an idea from concept to prototype stage during the event. The objective of the curricular hackathon is to provide the students with a focussed 24 hour period in which to complete the projects that they had been working on over the course of the semester. The purpose for introducing the hackathon was in response to the challenges that they were experiencing in completing their projects during the semester due to competing demands from other courses.
Conceptualisation of projects: Unlike traditional hackathons, the ideas for the projects originated mainly from the students. This was due to limited participation from outside stakeholders. In Mtsweni and Abdullah [22], project ideas originated from "community schools, non-profit organisations, expert and in some instances, from computer students". After coming up with their ideas, however, students were encouraged to engage with other lecturers and industry stakeholders in the field in which the solutions operated and then. This meant that the projects or ideas were not necessarily aligned with the national initiatives. It is suggested that in curricular hackathons, ideas for projects such as those from the National imperatives are offered to students to choose from.
Timeframe: Unlike a traditional hackathon, students had the entire year to conceptualise and plan their projects. The students were tasked at the beginning of the year to come up with an idea for an innovative apps that solves a specific organisational, social or academic need. In the first semester they developed the business case, user requirements, system requirements, prototyping and project plans. In the second semester they did the systems analysis and design, system architecture, use-cases, user interfaces and started building the system.
Closed event: Another basic difference between the two is that curricular hackathons are closed events as opposed to open events for traditional hackathons. Our event was restricted to third year IS students who were enrolled in the course. The reason for excluding other students were that this group of students had been working on their projects from the beginning of the semester. Inviting other participants at such a late stage would detract from the focus that students in the course had in completing their projects, as well as disadvantage those who had not been working on an idea/solution over the course of the year.
Incentives: Also, unlike the traditional hackathon, students earned marks for completing particular aspects of their solution throughout the year. There were key points where students needed to interact with stakeholders from industry in order to develop their ideas/designs as well as present them to lecturers in the faculty. Marks were allocated for documentation, apps and presentations during the semester in terms of their formative assessments. The final assessment was

a presentation of their workin gprojects to industry stakeholders, lecturers from the department as well as the external examiners. Students were also evaluated on the project documentation as well as the software code that was submitted at the end of the hackathon. All the material that was developed by the students during the course was uploaded to the institutional E-Learning system.

Compulsory: In contrast to the free/open culture of traditional hackathons [53], curricular hackathons restrict the voluntary nature of traditional hackathons, yet still allow for the philanthropic [27] and socially relevant [22] ideals. Firstly, participation in the hackathon was compulsory. Secondly, students were tasked to develop or come up with a socially relevant solutions, although this might not always be the case, especially if organisational pressures or corporate funding prevails. Thirdly, it is suggested that students were not necessarily motivated by the social cause of their solution, but by obtaining marks, and thus may have complied with the design brief merely to pass the course. They still did however have a large degree of freedom in choosing which topic they wish to focus on and the technologies or design they wished to use.

Intellectual property: Questions were also raised by the students as to the ownership of the intellectual property that was developed during the hackathon. It is suggested that the same regulations that pertains to academic research be applied to hackathons. In the end however, it appears as if few students or groups intend to incubate or continue with the projects which they developed during the hackathon. The sustainaility of such projects should be designed in the activities in order to move away from disposable assignments [54] to renewable projects.

5.2 Teaching Approach

One of the clear advantages of integrating the hackathon in the traditional capstone course [55] is that it places students in a high-pressure, team-based learning environment where they need to perform, much like in industry. In contrast, the traditional environment places no such demands, resulting in students remaining undecided on particular system or technology decisions that need to be made and rushing their projects at the end of the semester. Limitations of this approach are that it requires additional management and teaching skills, time and resources from the lecturer and department that are not necessarily catered for by traditional curriculum teaching activities. The process can be facilitated by external parties that ease the transition of hosting a hackathon; however this does incur additional costs.

Facilitation: In our case, we made use of the services of a professional organisation (PRO) to run the event and the residence catering services (RCS) to provide the catering. Funding for this was provided by senior management in the Faculty. PRO was responsible for advertising the event, managing the schedule as well as providing transport and accommodation for the mentors that were brought in from other companies and regions.

Mentoring: In addition, the curricular hackathon emphasises an apprenticeship model where students are guided by experienced mentors from industry, lecturers and senior students. The mentors were responsible for motivating the students at the start of the event, and for providing feedback and advice during presentations throughout the night as well as technical advice. Because the event was to be held on Campus after academic hours, we needed to get permission from the campus security services, the director of student life, the SRC president as well as the manager of the soccer institute where the end function was to be held. This was all arranged by one of the groups of students with the guidance of the lecturer.

In summary, curricular hackathons are closed events that are directed at accelerating students' capstone projects in a focussed 24 hour session that is hosted on campus by experienced facilitators. Students are dividided into teams of betweeen 3 and five students at the start of the semester. The projects are conceptualised and developed during the semester and completed at the hackathon. Participation is compulsory and projects/presentations are assessed for marks during the hackathon and at their final project presentations. These events require different expertise and resources to facilitate than traditional teaching and/or capstone projects demand. In addition, integrating academic/assessment requirements in a traditional hackathon also presents its own unique challenges for facilitators.

6 Conclusion

Hackathons provide a unique blend of active learning approaches [56, 29, 23, 57, 41, 14, 37, 19] in a focussed 24-72 hour event. Although they are widely hosted at Colleges and Universities [22–24], their role as a formal teaching approach in the curriculum is not well understood.

This paper reports on the three phases [28] of a curricular hackathon that was introduced in a capstone IS course, namely the pre, hackathon and post-hackathon phase. During the 12 week pre-hackathon phase teams were formed, ideas conceptualised, projects planned and development commenced, much like a traditional capstone course. During the 24 hour hackathon event the students completed their projects through a process of mentorship and regular feedback. Functional support for the students was provided by means of catering and other facilities. In the post-hackathon phase of 4 weeks, the students finalised their projects, completed their documentation, and presented their apps to industry and academic stakeholders.

The students found that the hackathon was an extremely valuable activity that prepares them for the demands of the working environment. They learned more about working with new technologies and tools, doing cross-platform development and using Google APIs and public SMS services. More importantly they developed a number of soft-skills during the process [10]. They learned a lot about teamwork, project management skills, time management under pressure, punctuality, responsibility, creativity, bug-fixing and presentation skills.

We found it valuable to use the services of a professional company to facilitate the hackathon, as academics are not prepared for the kinds of activities that are required. Furthermore, we found that it required senior management support and additional financial resources that are not necessarily catered for in traditional teaching activities. Additionally, it was effective in accelerating the completion of student projects, especially at the end of the semester when they were pressured by other courses to prepare for exams and final reports. Finally, we learned that a curricular hackathon can be a fun event that stimulates students interests in the discipline and creates envy amongst non-participants.

Our suggestions are to host two hackathon events during the year . The first event at the start of the year should be open to all IS/CS students in order to expose all UG to the concept and allow the third years to conceptualise their projects. The second, closed event should be held towards the end of the semester and restricted to the final year students for them to complete their projects. Furthermore, the use of source-code repositories should be encouraged in order to manage and monitor the software development progress before and after the hackathon.

This research indicates that a curriculur hackathon is a viable approach to facilitating workplace skills in the CS/IS curriculum and provides some guidelines on how this was done as well as highlighting some of the primary differences and similarities when compared to the traditional hackathon. Improved means of assessment is called for in evaluating hackathons in the curriculum and further research may need to examine what it means to ‘hack and what students ‘hack during hackathons as well as assessing ‘how and ‘what they learn from these activities.

7 Acknowledgements

We would like to thank the senior staff in the faculty who supported this project both financially and administratively; the directors of student life and campus security; the facilitation company who assisted us in planning and hosting the event; the management team, from the student group, who did additional work in order to ensure the events success; as well as the other lecturers and students who participated in the event. A special word of thanks to the external examiner for suggesting such an intervention and for attending the final presentations and providing guidance during the process. No other funding or support were provided for this project.

References

1. Bhardwaj, P.: This Is the Fastest-Growing Six-Figure Job in America and it Doesn’t Require a Degree (feb 2019), http://money.com/money/5635712/this-is-the-fastest-growing-six-figure-job-in-america-and-it-doesnt-require-a-degree/.
2. Skills vs. Degrees -The Debate, https://www.diskonto.net/2019/02/18/skills-vs-degrees-debate/, last accessed 2019-06-23.
3. Gallet, F.: The education vs experience debate. Bona Magazine (oct 2015), https://showme.co.za/lifestyle/the-education-vs-experience-debate/
4. Carr, W.: Theories of theory and practice. Journal of Philosophy of Education **20**(2), 177–186 (1986)

5. Merriam, S.: Some thoughts on the relationship between theory and practice. In: Merriam, S.B. (ed.) New Directions for Continuing Education, pp. 87–91. No. 15, Jossey-Bass, San Francisco, CA (1982)
6. Scott, E., Alger, R., Pequeno, S., Sessions, N.: The Skills Gap Observed Between IS graduates and the Systems Development Industry: A South African Experience. In: The Informing Science & IT Education Conference. Cork, Ireland (2002)
7. Scholtz, B., Cilliers, C., Calitz, A.: Bridging the Enterprise Systems (ES) Skills Gap : The South African Challenge. SACLA (10), 3–5 (2010)
8. Andrews, J., Higson, H.: Graduate employability, 'soft skills' versus 'hard' business knowledge: A european study. Higher Education in Europe **33**(4), 411–422 (2008)
9. Fajar, A., Hidajat, J.: Relationship among Soft Skills , Hard Skills , and Innovativeness of Knowledge Workers in the Knowledge. Prodecida Social and Behavioral Sciences **52**, 35–44 (2012)
10. Pieterse, V., van Eekelen, M.: Which Are Harder? Soft Skills or Hard Skills? **642**, 160–167 (2016). https://doi.org/10.1007/978-3-319-47680-3
11. Leidig, P.M., Ferguson, R., Leidig, J.: The use of community-based non-profit organizations in information systems capstone projects. ACM SIGCSE Bulletin **38**(3), 148 (jun 2006)
12. Adams, L., Daniels, M., Goold, A., Hazzan, O., Lynch, K., Newman, I.: Challenges in teaching capstone courses. ACM SIGCSE Bulletin **35**(3), 219 (2004)
13. Laware, G.W., Walters, A.J.: Real world problems bringing life to course content. Proceedings of the 5th conference on Information technology education - CITC5 '04 pp. 6–12 (2004)
14. Janse van Rensburg, J.T., Goede, R.: A Reflective Practice Approach for Supporting IT Skills Required by Industry Through Project-Based Learning. CCIS **963**, 253–266 (2019)
15. Yaqinuddin, A.: Problem-Based Learning as an Instructional Method. Journal of the College of Physicians and Surgeons Pakistan **23**(1), 83–85 (2013)
16. Clark, R.W., Threeton, M.D., Ewing, J.C.: The Potential of Experiential Learning Models and Practices in Career and Technical Education & Career and Technical Teacher Education. Journal of Career and Technical Education **25**(2), 46–62 (2010)
17. Winberg, C., Garraway, J., Engel-Hills, P., Jacobs, C.: Work-Integrated Learning: Good Practice Guide. No. 12 (2011), http://www.ru.ac.za/media/rhodesuniversity/content/communityengagement/documents/Higher_Education_Monitor_12.pdf
18. Ward, R.: Active, collaborative and case-based learning with computer-based case scenarios. Computers and Education **30**(1-2), 103–110 (1998)
19. Jennett, C., Papadopoulou, S., Himmelstein, J., Vaugoux, A., Roger, V., Cox, A.L.: Case Study 3: Students' Experiences of Interdisciplinary Learning While Building Scientific Video Games. International Journal of Game-Based Learning **7**(3), 93–97 (2017)
20. Perrotta, C., Featherstone, G., Aston, H., Houghton, E.: Game-based learning: latest evidence and future directions. NFER Research Programme: Innovation in Education, Slough, UK (2013)
21. Padayachee, I., van der Merwe, A., Kotzé, P.: Virtual learning system usage in Higher Education A study at two South African institutions. South African Computer Journal **57**(December), 32–57 (2015)
22. Mtsweni Jabu, Abdullah Hanifa: Stimulating and maintaining students' interest in Computer Science using the hackathon model. The Independent Journal of Teaching and Learning **10**(1), 85 – 97 (2015)

23. Rennick, C., Hulls, C., Wright, D., Milne, A.J., Li, E., Bedi, S.: Engineering design days: Engaging students with authentic problem-solving in an academic hackathon. In: ASEE Annual Conference and Exposition. vol. June. Salt Lake City, UT (2018)
24. Warner, J., Guo, P.J.: Hack.edu: Examining How College Hackathons Are Perceived By Student Attendees and Non-Attendees. ICER '17 Proceedings of the 2017 ACM Conference on International Computing Education Research pp. 254–262 (2017)
25. Lyndon, M.P., Cassidy, M.P., Celi, L.A., Hendrik, L., Kim, Y.J., Gomez, N., Baum, N., Bulgarelli, L., Paik, K.E., Dagan, A.: Hacking Hackathons: Preparing the next generation for the multidisciplinary world of healthcare technology. International Journal of Medical Informatics **112**(October 2017), 1–5 (2018)
26. Silver, J.K., Binder, D.S., Zubcevik, N., Zafonte, R.D.: Healthcare Hackathons Provide Educational and Innovation Opportunities: A Case Study and Best Practice Recommendations. Journal of Medical Systems **40**(7) (2016)
27. Richterich, A.: Hacking events: Project development practices and technology use at hackathons. Convergence: The International Journal of Research into New Media Technologies pp. 1–27 (2017)
28. Komssi, M., Pichlis, D., Raatikainen, M., Kindström, K., Järvinen, J.: What are Hackathons for? IEEE Software **32**(5), 60–67 (2015)
29. Nandi, A., Mandernach, M.: Hackathons as an Informal Learning Platform. Proceedings of the 47th ACM Technical Symposium on Computing Science Education - SIGCSE '16 pp. 346–351 (2016)
30. Nolte, A., Pe-than, E.P.P., Herbsleb, J., Filippova, A., Bird, C., Scallen, S.: You Hacked and Now What? Exploring Outcomes of a Corporate Hackathon **2**(November) (2018)
31. Aboab, J., Celi, L.A., Charlton, P., Feng, M., Ghassemi, M., Marshall, D.C., Mayaud, L., Naumann, T., Mccague, N., Paik, K.E., Pollard, T.J., Resche-rigon, M., Salciccioli, J.D., Stone, D.J.: A 'datathon' model to support cross-disciplinary collaboration. Science Translational Medicine **8**(333), 1–5 (2016)
32. Anslow, C., Brosz, J., Maurer, F., Boyes, M.: Datathons: An Experience Report of Data Hackathons for Data Science Education Craig. In: SIGCSE '16. pp. 615–620 (2016)
33. Burnham, K.: Inside Facebook's Hackathons: 5 tips for hosting your own (aug 2012), http://www.pcadvisor.co.uk/news/internet/3377232/inside-facebooks-hackathons-5-tips-for-hosting-your-own/.
34. Powell, D.: Want to run your first internal hackathon? Here are some tips from KPMG (2017), https://www.smartcompany.com.au/startupsmart/advice/want-run-first-internal-hackathon-tips-kpmg/, last accessed 2018-08-12.
35. Gama, K.: Crowdsourced Software Development in Civic Apps - Motivations of Civic Hackathons Participants **2**(Iceis), 550–555 (2017)
36. SITA NDP 2030 Mega Hackathon, http://www.ndp2030hackathon.gov.za/ndp-2030, last accessed 2019-02-20.
37. Lara, M., Lockwood, K.: Hackathons as Community-Based Learning: a Case Study. TechTrends **60**(5), 486–495 (2016)
38. Rosell, B., Kumar, S., Shepherd, J.: Unleashing innovation through internal hackathons. In: IEEE Innovations in Technology Conference. Warwick, RI (2014)
39. Senghore, F., Campos-Nanez, E., Fomin, P., Wasek, J.S.: Applying social network analysis to validate mass collaboration innovation drivers: An empirical study of NASA's International Space Apps Challenge. Journal of Engineering and Technology Management - JET-M **37**, 21–31 (2015)

40. Tsjardiwal, L.: Hackathons: An Effective Communication Tool for Innovation within an Organization? Thesis, Erasmus University Rotterdam (2016), https://pdfs.semanticscholar.org/360b/1ebef9c6c75aafa20b6b7831a3aeb82a55b3.pdf
41. Byrne, J.R., O'Sullivan, K., Sullivan, K.: An IoT and Wearable Technology Hackathon for Promoting Careers in Computer Science. IEEE Transactions on Education **60**(1), 50–58 (2017)
42. Wang, J.K., Pamnani, R.D., Capasso, R., Chang, R.T.: An Extended Hackathon Model for Collaborative Education in Medical Innovation. Journal of Medical Systems **42**(12) (2018). https://doi.org/10.1007/s10916-018-1098-z
43. Blanco, L.: BeSmart Hackathon: HBCU Students Hacked into their futures in Silicon Valley. Black Enterprise **49**(2), 54–57 (2018)
44. Tobor, N.: The Largest All-Female Hackathon In South Africa (aug 2017), https://www.iafrikan.com/2017/08/12/girlcode-the-largest-all-female-hackathon-in-south-africa/.
45. Yin, R.K.: Case Study Research: Design and Methods, vol. 5. SAGE Publications Limited, London, U.K., 2nd edn. (1994)
46. Benbasat, I., Goldstein, D.K., Mead, M.: The Case Research Strategy in Studies of Information Systems. MIS Quarterly **11**(3), 369–386 (1987)
47. Myers, M.D.: Dialectical hermeneutics: a theoretical framework for the implementation of information systems. Information Systems Journal **5**(1), 51–70 (1995)
48. Aristotle: Topics. In: Ross, W.D. (ed.) The Works of Aristotle, vol. I, chap. 100-164. Clarendon Press, Oxford, 1928 edn.
49. de Garay, J.: Aristotelism of difference. Foundations of Science **13**(3-4), 229–237 (2008)
50. Loveland, T.R.: Teaching personal skills in technology and engineering education: Is it our job? technology and engineering teacher **1**(April), 15–20 (2017)
51. Calitz, A.P., Greyling, J.H., Cullen, M.D.M.: South African Industry ICT Graduate Skills Requirements. In: South African Computer Lecturer Association. pp. 135–145. Port Elizabeth (2014)
52. Pfleeger, S.L.: Software metrics: Progress after 25 years? IEEE Software **25**(6), 32–34 (2008)
53. What is free software? (1996), http://www.gnu.org/philosophy/free-sw.html, last accessed 2019-02-21.
54. Wiley, D.: What is Open Pedagogy? (2013), http://opencontent.org/blog/archives/2975, last accessed 2019-02-12.
55. Scott, E.: From Requirements to Code: Issues and Learning in IS Students Systems Development Projects. Journal of Information Education Technology (JITE) **7**, 1–13 (2008)
56. Bowen, L.M.: The Limits of Hacking Composition Pedagogy. Computers and Composition **43**, 1–14 (2017)
57. Maaravi, Y.: Running a research marathon. Innovations in Education and Teaching International **55**(2), 212–218 (2018)