Hackathons as a Formal Teaching Approach in Information Systems Capstone Courses

Walter F. Uys[1,2](✉)

[1] SDL Research Focus Area, North-West University, Mahikeng, South Africa
walter.uys@nwu.ac.za
[2] CITANDA, University of Cape Town, Cape Town, South Africa

Abstract. Hackathons are 'hacking marathons' in which participants collaboratively and rapidly prototype new applications over a 24–48 h period. The potential of hackathons as a strategy for stimulating interest in the CS fields is well known. Hackathons share many similarities with capstone courses, however their application as a formal teaching approach in the CS/IS curriculum is less prevalent. This paper describes the introduction of a curricular hackathon in a 3rd-year IS capstone course at a South African university. An exploratory case study was conducted to evaluate feedback from the participants and organizers. In the process, the students completed seven new applications which they had conceptualized during the course. They also learned something about new technologies and programming interfaces as well as they exhibited growth in personal and inter-personal competencies. Seven fundamental differences between curricular and traditional hackathons are highlighted. Suggestions for integrating hackathons into undergraduate CS/IS capstone courses are provided together with possible areas for further research.

Keywords: Information systems education · Capstone courses · Software application development · Hackathons · Project-based learning

1 Introduction

Application development is one of the fastest growing high-paying careers in the USA [6]. Because software development environments are free and easily accessible, there is a perception that anyone can become a successful developer without any formal education or training [6]. Contrasted against this growth in demand for application developers is the *concern about the value of a university degree versus practical experience* [12,16,31,48,50].[1] Thus there remain concerns w.r.t. the *workplace-readiness* of graduates as well as a general lack of soft-skills that are needed in the workplace [3,21,40].

[1] See also https://www.diskonto.net/2019/02/18/skills-vs-degrees-debate/.

B. Tait et al. (Eds.): SACLA 2019, CCIS 1136, pp. 79–95, 2020.
https://doi.org/10.1007/978-3-030-35629-3_6

In higher education institutions (HEI) there is a fair understanding of this need, and the *capstone course* is one of the strategies used to fill this gap [50]. 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 [27]. 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 [2]. Most courses find a balance between these two models. 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 globally needed in the software- and IT-industry: see [32] for comparison. The primary strategy adopted in capstone courses is referred to as *project-based learning* (ProjBL) [22,26] which is not to be confused with problem-based learning (PBL) [60]. Other strategies that can be used are experiential learning [14], work-integrated learning [59], case-based learning [56], game-based learning [23,38] as well as virtual learning [36]; for a broader overview see [13]. Though these strategies have some commonalities, project-based learning emphasizes an educational strategy aimed at solving real-world project-based problems [22].

An under-represented approach for implementing ProjBL in CS/IS capstone courses is the hackathon [15]. Whilst colleges and universities are frequently the preferred setting for hackathons [33,44,57], they are mainly used as an informal approach to expose the youth to CS/IS and leverage their creativity [29,44,52]. Hackathons share many key characteristics with capstone courses [41], yet their use in the software engineering (SE) and computer science curriculum is not widely reported [18,41,42]. This means that educators have minimal guidance/tips or techniques for introducing hackathons in their courses.

This paper responds to this desideratum by showing how hackathons can be used as a formal educational approach for implementing ProjBL in the CS/IS curriculum—specifically in our *3rd-year Information Systems Development* course for the *Bachelor of Commerce* (BCom) degree. The secondary research question is about the differences between traditional versus curricular hackathons. We describe our experiences of introducing a curricular hackathon in our course at a South African HEI. We do so by outlining the central concepts of hackathons, recapitulating related work, outlining our research method, and presenting our findings and recommendations for implementing hackathons in the future under-graduate (UG) IS curriculum.

2 Central Concepts and Related Work

Hackathons originated in 1999 from the voluntary efforts of programmers in order to develop/advance a free/open-source operating system called OpenBSD [45]. Although the concept of a hackathon is in need of a more precise definition [24], we adopted the following

Working Definition: ***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 h [34].

There are many different kinds of hackathons—each with a unique approach. Some are referred to as data-dives, code-fests, code-jams, hack-days, sprints, edit-a-thons [35], data-thons [1,4], code-camps [42] or game-labs [23]. Over time these events have increasingly become sponsored by corporations such Facebook [9], F-Secure [24] or KPMG, as well as by governmental agencies such as Gov-Hack [43], CivHack [17] or NDPHack.[2] 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 [45]. 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"* [24].

Hackathons promote innovation in product and application development, new uses for existing products or apps or new solutions for government, business or education [25,46,51,52,54]. Ideas or innovations may be bottom-up or top-down [24], i.e.: originating either from the developers or from senior management, thereby fostering an entrepreneurial approach. They typically take place over extended and focused periods of between 24 and 72 h in dedicated venues [34]. Participants remain primarily at the venues with limited breakaways for ablutions or eating and optional sleeping during such events [46] although there are reports of virtual participation [33]. Catering such as food, energy drinks and coffee is normally provided [24] and infrastructure such as computers and data projectors may be available, although participants are normally encouraged to provide their own laptops or hardware devices [10].

The targeted participants for hackathons are mainly software developers and technical personnel, although teams may be comprised of programmers, analysts, designers, subject-matter experts, managers and community representatives [46]. Team sizes can vary between three and five people, with anything from five to fifty teams competing at a particular event [57]. Hackathons can also specifically target under-represented groups such as women and historically disadvantaged individuals (HDIs) [7,24,53]. There is also general consensus that (good) programmers are a scarce resource at hackathons [45]. Furthermore, it is acknowledged that financial and material support by leadership is important to hosting such events [46].

[2] See http://www.ndp2030hackathon.gov.za/ndp-2030.

Hackathons can either be closed or open events. Closed events are internal to organizations [24]. Open events are organized as public or civic events that are open to everybody [17]. Open events are broadly publicized and attract large corporate sponsorship to encourage attendance and participation [24]. Open events mostly focus on a specific topic or theme such as health and fitness [24], healthcare [52,55], Internet of things (IoT) or wearable devices [10], whereas closed events might be geared around a new product feature or innovations for a specific company such as Facebook [9,24].

Hackathons are able to provide *"new and exciting opportunities for education and research"* [23] as well as to develop project management and communication skills in addition to creativity and innovation amongst participants [18]. They are, however, known to restrict sound software-architectural approaches to development and provide only limited testing and quality-assurance opportunities [44]. It is also questionable how much new (programming) skills can be acquired during such short events. The perception remains that participants rely on familiar skills and techniques [45] and have only *"limited time to interact with industry experts or learn valuable hard and soft skills relevant to SE"* [41].

Though there is a shortage of published reports of formal hackathons in the UG CS/IS curriculum [18,41,41], there are some related studies that can provide insights into how to introduce those into the curriculum. For example, [19] describes a method for implementing a Hackathon for UG course projects. That approach followed a software development methodology over a period of 24 h with 22 students divided into seven groups. The purpose of that hackathon was to develop an Internet-of-Things application using *Rasberry Pi* devices. The event was hosted outside the university according to a traditional hackathon format. The event was however bespoke and took students through the entire SDLC and was thus not integrated into a capstone course.

A follow-up paper by the same author [18] describes the experiences of developing student projects in a continuous 10 h 'Hack Day' on a Saturday at the end of the semester. Although a 'Hack Day' does not have the full time commitment of a 24–72 h Hackathon, it appears to offer similar benefits. Students were organized into groups of three. They were co-located to avoid groups working remotely as well as to fostering collaboration amongst students and allowing the instructor to provide support. Feedback was obtained from the 24 students in the course, whereby the results indicated that the students learned more during the experience than with traditional classes, and that the event fostered greater motivation and engagement amongst the students and closer relationships with the instructor. Further feedback indicated a high degree of task focus during the event as well as an increase in time management skills as a result of the time pressure during the event.

The principles of hackathons were also applied to address student learning outcomes in a first-year Engineering curriculum [44]. Even though this intervention was targeted at achieving course-level objectives, the approach followed included 'design days' which bear minimal resemblance with informal hackathons. The main outcome was new designs (not software applications).

The purpose of those ‘design days’ was to improve collaboration between students and staff, to expose students to engineering design concepts, to integrate knowledge from across the curriculum, and to stimulate creative thinking. The primary curriculum outcomes of the ‘design days’ were improved teamwork, better understanding of design, and greater student engagement. Non-curricular outcomes were a highly creative and fun/engaging event for the students as well as increased motivation to participate in such events.

Another study examined the efficacy of the hackathon approach to stimulate students’ enrollments and interest in CS [33]. Six hackathons were hosted at a university over a period of three years. Participation was voluntary after the students had been invited to participate in the event. The hackathon gave students an opportunity to learn and network with subject matter experts and to be part of larger project teams that were focused 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, as well as of changing their perceptions of CS by-and-large.

As it can be seen from the basic definition of hackathons and from how they have been used in higher education in the case studies recapitulated above, there remains the *question of how a hackathon can be better integrated into the curriculum.* This question is addressed in the subsequent sections.

3 Methods and Materials

For this paper an *exploratory case study* [61] was carried out. This method is suited for novel studies where the experiences of participants and the context of action is important [5]—in our case: normal classroom activities and student evaluations. In compliance with our university’s regulations, participants’ names and identities have been kept anonymous, and statements were attributed with three-letter acronyms (TLA) representing the corresponding persons. The participants completed an informed-consent form which outlined the purpose of our study, the use of the students’ answers, and the confidentiality of their information. The students were also told that their voluntary participation would have no bearing on their final marks (course results). Our case study was supported by first-person reports and reflections, additional reports by the hosting organization, as well as anonymous student feedback and evaluation. No personal interviews were conducted. The answers obtained were analyzed by means of *topical analysis* [47] which can reveal key topics or issues in a corpus of text, a discourse or some particular event. Topical analysis provides a method for comparing and analyzing similarities and differences between related topics, definitions, artifacts or concepts [20].

4 Case Study

There are sufficient commonalities between a traditional hackathon and our curriculum hackathon for us to refer to our object of study as 'hackathon'. Some of the main similarities between the two are that our event was a focused event that occurred at a particular venue (a computer lab on campus), over a fixed period of 24 h. 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 of the South African academic year (19–20 October) so that there were no conflicts with other courses, assignments, tests or exams. After the hackathon, our students had 2–3 weeks to complete their documentations 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.

4.1 Aims and Objectives

The emphasis at our hackathon was the *delivery of a working system*, whereby 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 abovementioned SITA NDP2030 hackathon which itself carried a prize of 100'000 South African Rand (≈US$6'000) for the winning team.

4.2 Phases

The typical hackathon can be represented by means of the classical IS Input-Process-Output model [24]. 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 took approximately 12 weeks which overlapped with the traditional capstone curriculum. During this time, the students formed their teams, conceptualized 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 students were involved in final preparations for the event. On the final days before the event, drinks and meals were purchased and the catering orders confirmed.

The hackathon event itself (process phase) started at 18h00 (planned was 17h00) on the Friday, and, after initial presentations and motivation by the organizing team, the students had dinner. After dinner the students ‘hacked’ for 4 h and presented their progress to the entire forum at midnight. After the presentations the students had snacks and then continued to hack till 05h00 when they presented their results again to the forum.

Departing from the traditional hackathon approach we had some physical exercise in the morning of the second day whereby some sports games were played by the students to get energetic again after a long night of intense coding. Thereafter, the students started hacking again from 08h00 till 12h00. At noon the students had lunch and resumed hacked again 13h00–17h00; then they submitted their final projects. According to one participant (TLM),

> “*the results that we reached at the hackathon were amazing: we managed to get most features of the application working during those* 24 h”.

At the conclusion of the formal hackathon activities, students, lecturers and facilitators were treated to a barbecue. Thereafter, our students were so exhausted that they were glad to ‘pack up’ and go home.

During the post-hackathon phase (output phase), the students made the final changes to their apps in order to capture screenshots for their project report and also to prepare for their final assessment presentations and demonstrations. They were also required to write individual reports to explain their individual contributions to the their groups. For this report they had been advised already at the beginning of the semester to keep a record of their personal tasks and activities. The final demonstrations and presentations were held four weeks later. This gave the students the opportunity to explain their solutions to invited representatives from government, industry and the university.

4.3 Projects

During the first semester (February to June) of our academic year, our students were required 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 conceptualized by the students and implemented through various iterations and interactions with the lecturers and stakeholders. Only during the hackathon the students completed and presented their final systems; (see Table 1).

4.4 Results

The course’s lecturers found that the event gave the students the opportunity to focus solely on the completion of their projects without other distractions. The event also imposed personal challenges to the students, such as to stay awake for its entire duration and to work under pressure. During the hackathon, students learned how to work with new technologies, tools and software development platforms. For example, they learned how to develop in Java on Android Studio;

Table 1. App development: teams and systems

Team/App	Specification
Residence Control System	Monitoring and managing visitors access to student residences to avoid illegal squatting
e-License App	A mobile app to register and represent 'virtual' driver licenses. It must allow road officers to validate physical and virtual licenses as well as to check for outstanding fines or license expiry. It has a front-end sub-app and back-end sub-app
Soapy Shine Car Wash	A car wash loyalty app that allows members to check a shop's availability (or current queue length), to book and be notified at any of the participating shops
Billboards Innovation	Facilitates the remote management and hosting of advertisements on electronic billboards
Stay Residence Booking System	A variation of 'AirBnB' for student accommodation: intended mainly for finding and booking of 'approved' university 'digs'
Virtual Housing Project	A VR system to convert 2D plans into 3D virtual walkthroughs for visualizing the architectural features of real estate properties
Clinic Appointments	A queuing system for public hospitals and clinics, similar to those found in banks or other mobile service providers. It must be able to distinguish emergency cases from dispensary patients

they learned database development as well as the use of technologies such as XAMMP, PHP, or UMAJIN. They were exposed to cross-platform development architectures and the use of APIs (such as Google's authentication features and maps in Android Studio) as well as to interfacing with bulk SMS providers which had not been part of their under-graduate coursework so far. Consequently,

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

Surprising were the unintended 'soft skills' the students developed during the project; see [40] for comparison. The students learned much about teamwork, project management (the first semester complement of this module), time management under pressure, punctuality, responsibility, creativity, bug-fixing as well as presentations an audience. Although these skills are not considered to be the main outcomes of an IS course [28], they are recognized as critical for the successful integration of graduates into the workplace [11,48]. Thus the hackathon can be regarded as a suitable 'active learning' method for developing these 'soft skills' which are hard to teach in the lecture hall.

The students also 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; (see Table 2). In particular:

> "*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).

Table 2. Learning experiences and suggested improvements

What did you like about the teaching and learning experiences of this module?	Stud.
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 affect 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 [*more*] *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, [*too*]	(ST8)
Every group must be assigned a supervisor	(ST9)
The use of a study guide	(ST10)
More practicals, field trips and recruiting of sponsors for practicals (like Microsoft, hackathon- and other IT-related companies)	(ST11)
The class experience was very new; I don't think there is much more that I can add	(ST12)
It's perfect!	(ST13)

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, i.e.: what they learned during the process as compared to a traditional approach. Even the students' final marks would not provide a fair representation of learning, as students and project topics in other years would differ individually. Assessing software development progress in industry is an ongoing challenge [39], though the usual metrics (such as lines of code, function points or completed features) could also be used to assess the value that the hackathon had on the students' projects progress. Ultimately, the measure of success for such an approach would be to 'track' these students into the industry and see how they are faring there in practice.

5 Discussion

Some similarities and differences between the curricular and the traditional hackathon are explored next, as per our secondary research question.

5.1 Curricular Hackathons

Scope and Purpose. The scope and purpose of the curricular hackathon is much 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 focused 24 h 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 problems that students were experiencing in completing their projects during the semester due to competing demands from other courses.

Conceptualization of Projects. Unlike in traditional hackathons, the ideas for the projects originated mainly from the students themselves. This was due to limited participation from outside stakeholders. In [33], by contrast, project ideas originated from schools, non-profit organizations, expert and, in some instances, from computer students. After coming up with their ideas, however, our students were encouraged to engage with other lecturers and industry stakeholders in the field for which the solutions were intended. This meant that the projects or ideas were not necessarily aligned with 'national priorities'. It is suggested that in curricular hackathons, ideas for projects such as those from the list of national priorities are given to students to choose from.

Time Frame. As the hackathon was held at the end of the capstone course, students had the entire year to conceptualize and plan their projects. The students were required at the beginning of the year to produce ideas for innovative apps that address some organizational, societal or academic needs. In the first semester the students developed the business case, user requirements, system requirements, prototypes and project plans. In the second semester they did systems analysis and design, system architecture, use-cases, and user interfaces; then they and started building the system.

Closed Event. Another difference between the two is that curricular hackathons are closed events as opposed to open events for traditional hackathons. Our event was restricted to 3rd-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 year. Inviting other participants at such a late stage would have detracted from the educational focus described above, and would also have disadvantaged those participants who had not already been working on a project during the course of the year.

Incentives. Also, unlike in a 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 and designs as well as to present them to lecturers in the faculty. Marks were allocated for documentation, apps and presentations during the semester according to formative assessments. The final assessment was a presentation of students' working projects to industry stakeholders, lecturers from the department, and 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,[3] curricular hackathons restrict the voluntary nature of traditional hackathons, yet still allow for the philanthropic [45] and socially relevant [33] ideals. Firstly, participation in the hackathon was compulsory. Secondly, students were required to develop socially relevant solutions, although this might not always be feasible, especially if organizational pressures or corporate funding prevails. Thirdly, it is conjectured 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 had, however, a large degree of freedom in choosing which topic they wanted to focus on as well as the technologies or design scheme they wished to apply.

Intellectual Property. Questions were also raised by the students as to the ownership of the intellectual property that was developed during the hackathon. It was suggested that the same regulations that pertain to academic research be applied to hackathons. In the end, however, it appears as if few students or groups intended to incubate or continue with the projects which they developed during the hackathon. The sustainability of such projects should be designed into the activities in order to move away from disposable assignments to renewable projects [58].

5.2 Teaching Approach

One of the clear advantages of integrating the hackathon into the traditional capstone course [49] is that it puts students into a high-pressure, team-based learning environment where they need to perform much like in industry. By contrast, the traditional capstone course imposes no such demands, resulting in students remaining undecided on particular system or technology decisions that

[3] For comparison see http://www.gnu.org/philosophy/free-sw.html.

need to be made, and rushing their projects at the end of the semester. Some other challenges and opportunities with this approach are discussed below.

All-nighter. Some concerns have been raised as to the effects of sleep deprivation amongst the students. Research shows that students are accustomed to 'all-nighters' due to academic and social pressures and that acute sleep deprivation can have a physical but not necessarily a cognitive impact on healthy university students [37]. In order to ameliorate this effect (based on the organizers past experience of 72 h hackathons) an exercise session was held in the morning to boost the participants vitality. These concerns can also be addressed by changing the format to a 'code camp' [42] which is a coding event that is hosted over the period of a week during normal class times. The test week would be an ideal time for such a 'code camp' and would provide greater opportunities for the students to interact with industry experts and to develop the soft skills that are expected of software engineers.

Facilitation. Another limitation of the hackathon approach is that it requires additional management, teaching facilitation, 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 will incur additional costs. In our case, we used the services of a professional organization (PRO) to run the event as well as the residence catering services to provide the food. PRO was responsible for advertising the event, managing its 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 emphasizes 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 held on campus after academic hours, we needed to obtain permission from the campus security services, the director of student life, the students' representative council, and the manager of the soccer institute where the end function was held. This was all arranged by one of the groups of students with the guidance of the lecturer.

Accelerating Projects. Additionally, we found that the hackathon 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 as well as 'awe' amongst non-participants.

Closed Event. One concern raised by students from other years and other departments was why they could not participate in the curricular hackathon. This was explained to them due to the closed nature as part of the IS capstone course. Our suggestions are now to host two hackathon events during the year:

The first event at the start of the year should be open to all students in order to expose them to the concept and to allow the third-year students to conceptualize their projects. A second, closed event should be held towards the end of the semester and be restricted to the final-year IS students for them to complete their projects for marks.

Asessing Progress. It is difficult to assess the degree of software development progress that groups can achieve during hackathons. We suggest that source-code repositories such as *GitHub* be used to manage and monitor the software development progress throughout the semester and during the hackathon.

In summary, curricular hackathons are closed events that are directed at accelerating students' capstone projects in a focused 24 h session that is hosted on campus by experienced facilitators. Students are divided into teams of between 3 and 5 students at the start of the semester. The projects are conceptualized and developed during the semester and completed at the hackathon. Participation is compulsory. Projects (presentations) are assessed for marks during the hackathon and at the final project-day. These events require other expertise and resources to facilitate than traditional teaching and/or capstone projects. Last but not least the integration of academic (assessment) requirements into a traditional hackathon creates a number of additional challenges for the facilitators.

6 Conclusion

Hackathons provide a unique blend of active learning approaches [8,10,22,23,25,30,34,44] in focused 24–72 h events. Although they are widely hosted at colleges and universities [33,44,57], their role as a formal teaching approach in the CS/IS curriculum was hitherto not well understood.

This paper describes the three phases [24] of a curricular hackathon conducted in a *3rd-year undergraduate capstone course* for *information systems design and development*. During a 12 week pre-hackathon phase the teams were formed, ideas conceptualized, projects planned and development commenced (much like a traditional capstone course). During the 24 h 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 a post-hackathon phase of 4 weeks, the students finalized their projects, completed their documentation, and presented their software apps to industrial and academic stakeholders.

Our students found that the hackathon was a valuable activity that prepared them for the demands of the industrial work environment. They learned more about working with new technologies and tools, doing cross-platform development, using Google APIs and public SMS services. More importantly they developed a number of important 'soft skills' [40] during the process. They learned much about teamwork, project management, time management under pressure, punctuality, responsibility, creativity, bug-fixing and presentation of achievements.

Thus a curricular hackathon is a viable approach to facilitating workplace skills in the CS/IS curriculum that complements traditional capstone courses. This paper provides some guidelines on how curricular hackathons can be implemented and highlights some of the most important differences and similarities in comparison to traditional hackathons. Some challenges and limitations of this approach are also outlined.

Future work should evaluate the efficacy of 'code camps' as opposed to curricular hackathons. Improved means of assessment are also called for in evaluating curricular hackathons. Further research may also look at what it means to 'hack', and how these approaches foster workplace skills amongst new generations of ICT students.

Acknowledgments. The research described in this paper was done as part of the author's participation in our regional *Institutional Teaching Excellence Awards* (ITEA). Thanks to the senior staff of our 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 event's success; as well as the other lecturers and students who participated in the event. Thanks to the external examiner for suggesting such an educational intervention as well as for attending the final presentations and providing guidance during the process. Thanks for the financial contribution made by the schools' directors towards hosting the hackathon. Last but not least thanks the anonymous reviewers for their critical feedback as well to the SACLA'2019 conference's audience for some insightful remarks following the presentation of this work.

References

1. Aboab, J., et al.: A 'Datathon' model to support cross-disciplinary collaboration. Sci. Transl. Med. **8**(333), 1–5 (2016)
2. Adams, L., Daniels, M., Goold, A., Hazzan, O., Lynch, K., Newman, I.: Challenges in teaching capstone courses. ACM SIGCSE Bull. **35**(3), 219 (2004)
3. Andrews, J., Higson, H.: Graduate employability, 'Soft Skills' versus 'Hard' business knowledge: a European study. High. Educ. Eur. **33**(4), 411–422 (2008)
4. Anslow, C., Brosz, J., Maurer, F., Boyes, M.: Datathons: an experience report of data hackathons for data science education. In: Proceedings of SIGCSE 2016, pp. 615–620 (2016)
5. Benbasat, I., Goldstein, D.K., Mead, M.: The case research strategy in studies of information systems. MIS Q. **11**(3), 369–386 (1987)
6. Bhardwaj, P.: This is the Fastest-Growing Six-Figure Job in America and it Doesn't Require a Degree (2019). http://money.com/money/5635712/this-is-the-fastest-growing-six-figure-job-in-america-and-it-doesnt-require-a-degree/
7. Blanco, L.: BeSmart hackathon: HBCU students hacked into their futures in silicon valley. Black Enterp. **49**(2), 54–57 (2018)
8. Bowen, L.M.: The limits of hacking composition pedagogy. Comput. Compos. **43**, 1–14 (2017)
9. Burnham, K.: Inside Facebook's Hackathons: 5 Tips for Hosting Your Own (2012). http://www.pcadvisor.co.uk/news/internet/3377232/inside-facebooks-hackathons-5-tips-for-hosting-your-own/

10. Byrne, J.R., O'Sullivan, K., Sullivan, K.: An IoT and wearable technology hackathon for promoting careers in computer science. IEEE Trans. Educ. **60**(1), 50–58 (2017)
11. Calitz, A.P., Greyling, J.H., Cullen, M.D.M.: South African industry ICT graduate skills requirements. In: Proceedings of SACLA 2014 Annual Conference of the Southern African Computer Lecturers' Association, pp. 135–145 (2014)
12. Carr, W.: Theories of theory and practice. J. Philos. Educ. **20**(2), 177–186 (1986)
13. Carroll, J.M. (ed.): Innovative Practices in Teaching Information Sciences and Technology — Experience Reports and Reflections. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-03656-4
14. 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. J. Career Tech. Educ. **25**(2), 46–62 (2010)
15. Duhring, J.: Project-based learning kickstart tips: hackathon pedagogies as educational technology. In: Proceedings of NCIIA, pp. 1–8 (2014)
16. Gallet, F.: The Education vs. Experience Debate. Bona Magazine (2015). https://showme.co.za/lifestyle/the-education-vs-experience-debate/
17. Gama, K.: Crowdsourced software development in civic apps – motivations of civic hackathons participants. In: ICEIS, vol. 2, pp. 550–555 (2017)
18. Gama, K.: Developing course projects in a hack day: an experience report. In: Proceedings of ITiCSE 2019, Aberdeen, pp. 388–394 (2019)
19. Gama, K., Alencar, B., Calegario, F., Neves, A., Alessio, P.: A hackathon methodology for undergraduate course projects. In: Proceedings of FiE 2018 Conference (2019)
20. deGaray, J.: Aristotelism of difference. Found. Sci. **13**(3/4), 229–237 (2008)
21. Hendarman, A.F., Tjakraatmadja, J.H.: Relationship among soft skills, hard skills, and innovativeness of knowledge workers in the knowledge economy era. Procedia Soc. Behav. Sci. **52**, 35–44 (2012)
22. Janse van Rensburg, J.T., Goede, R.: A reflective practice approach for supporting IT skills required by industry through project-based learning. In: Kabanda, S., Suleman, H., Gruner, S. (eds.) SACLA 2018. CCIS, vol. 963, pp. 253–266. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-05813-5_17
23. 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. Int. J. Game-Based Learn. **7**(3), 93–97 (2017)
24. Komssi, M., Pichlis, D., Raatikainen, M., Kindström, K., Järvinen, J.: What are Hackathons for? IEEE Softw. **32**(5), 60–67 (2015)
25. Lara, M., Lockwood, K.: Hackathons as community-based learning: a case study. TechTrends **60**(5), 486–495 (2016)
26. Laware, G.W., Walters, A.J.: Real-world problems bringing life to course content. In: Proceedings of 5th Conference on Information Technology in Education, pp. 6–12 (2004)
27. Leidig, P.M., Ferguson, R., Leidig, J.: The use of community-based non-profit organizations in information systems capstone projects. ACM SIGCSE Bull. **38**(3), 148 (2006)
28. Loveland, T.R.: Teaching personal skills in technology and engineering education: is it our job? Tech. Eng. Teach. **1**, 15–20 (2017)
29. Lyndon, M.P., et al.: Hacking Hackathons: preparing the next generation for the multidisciplinary world of healthcare technology. Int. J. Med. Inform. **112**, 1–5 (2018)

30. Maaravi, Y.: Running a research marathon. Innov. Educ. Teach. Int. **55**(2), 212–218 (2018)
31. Merriam, S.: Some thoughts on the relationship between theory and practice. In: Merriam, S.B. (ed.) New Directions for Continuing Education, pp. 87–91. Jossey-Bass, San Francisco (1982)
32. Motta, G., Wu, B. (eds.): Software Engineering Education for a Global E-Service Economy – State of the Art, Trends and Developments. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-04217-6
33. Mtsweni, J., Hanifa, A.: Stimulating and maintaining students' interest in computer science using the hackathon model. Indep. J. Teach. Learn **10**(1), 85–97 (2015)
34. Nandi, A., Mandernach, M.: Hackathons as an informal learning platform. In: Proceedings of 47th ACMSIGCSE, pp. 346–351 (2016)
35. 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. In: Proceedings of ACM Conference on Human-Computer Interaction, p. 129 (2018)
36. Padayachee, I., van der Merwe, A., Kotzé, P.: Virtual learning system usage in higher education - a study at two South African institutions. South Afr. Comput. J. **57**, 32–57 (2015)
37. Patrick, Y., et al.: Effects of sleep deprivation on cognitive and physical performance in university students. Sleep Biol. Rhythm. **15**(3), 217–225 (2017)
38. Perrotta, C., Featherstone, G., Aston, H., Houghton, E.: Game-Based Learning: Latest Evidence and Future Directions. NFER Research Programme: Innovation in Education, Slough (2013)
39. Pfleeger, S.L.: Software metrics: progress after 25 years? IEEE Softw. **25**(6), 32–34 (2008)
40. Pieterse, V., van Eekelen, M.: Which are harder? Soft skills or hard skills? In: Gruner, S. (ed.) SACLA 2016. CCIS, vol. 642, pp. 160–167. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-47680-3_15
41. Porras, J., et al.: Hackathons in software engineering education – lessons learned from a decade of events. In: Proceedings of ICSE, pp. 40–47 (2018)
42. Porras, J., Knutas, A., Ikonen, J., Happonen, A., Khakurel, J., Herala, A.: Code camps and hackathons in education – literature review and lessons learned. In: Proceedings of 52nd Hawaii International Conference on Systems Science, pp. 7750–7759 (2019)
43. 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/
44. 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: Proceedings ASEE Annual Conference, Salt Lake City (2018)
45. Richterich, A.: Hacking events: project development practices and technology use at hackathons. Converg. Int. J. Res. New Media Tech. **25**, 1000–1026 (2017)
46. Rosell, B., Kumar, S., Shepherd, J.: Unleashing innovation through internal hackathons. In: Proceedings of IEEE Innovation in Technnical Conference, Warwick (2014)
47. Ross, W.D. (ed.): The Works of Aristotle, vol. I. Clarendon Press, Oxford (1928)
48. Scholtz, B., Cilliers, C., Calitz, A.: Bridging the Enterprise Systems (ES) skills gap: the South African challenge. In: Proceedings of SACLA 2014 Annual Conference of the Southern African Computer Lecturers' Association, pp. 3–5 (2010)

49. Scott, E.: From requirements to code: issues and learning in IS students systems development projects. J. Inf. Tech. Educ. **7**, 1–13 (2008)
50. 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: Proceedings of Information Science and IT Education Conference, Cork (2002)
51. 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. J. Eng. Tech. Manag. **37**, 21–31 (2015)
52. 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. J. Med. Syst. **40**(7), 177 (2016)
53. Tobor, N.: The Largest All-Female Hackathon in South Africa (2017). https://www.iafrikan.com/2017/08/12/girlcode-the-largest-all-female-hackathon-in-south-africa/
54. Tsjardiwal, L.: Hackathons: an effective communication tool for innovation within an organization? Thesis, Erasmus Universiteit Rotterdam (2016). https://pdfs.semanticscholar.org/360b/1ebef9c6c75aafa20b6b7831a3aeb82a55b3.pdf
55. Wang, J.K., Pamnani, R.D., Capasso, R., Chang, R.T.: An extended hackathon model for collaborative education in medical innovation. J. Med. Syst. **42**(12), 239 (2018)
56. Ward, R.: Active, collaborative and case-based learning with computer-based case scenarios. Comput. Educ. **30**(1/2), 103–110 (1998)
57. Warner, J., Guo, P.J.: Hack.edu: examining how college hackathons are perceived by student attendees and non-attendees. In: Proceedings of ICER 2017 ACM Conference on International Computing Education Research, pp. 254–262 (2017)
58. Wiley, D.: What is Open Pedagogy? (2013). http://opencontent.org/blog/archives/2975/
59. Winberg, C., Garraway, J., Engel-Hills, P., Jacobs, C.: Work-Integrated Learning: Good Practice Guide #12 (2011). http://www.ru.ac.za/media/rhodesuniversity/content/communityengagement/documents/Higher_Education_Monitor_12.pdf
60. Yaqinuddin, A.: Problem-based learning as an instructional method. J. Coll. Phys. Surg. Pak. **23**(1), 83–85 (2013)
61. Yin, R.K.: Case Study Research: Design and Methods, vol. 5, 2nd edn. SAGE Publ., Thousand Oaks (1994)