Work and Play in Software Engineering Training: Experiences from the Silicon Mountain

Ngatchu Damen Nyinkeu
School of Information Technology
Catholic University Institute of Buea
Buea, Cameroon

Henry Ngatchu
Department of Computer Science
University of Buea
Buea, Cameroon

***Abstract*—The University City of Buea in Cameroon – Africa has been named the Silicon Mountain of Africa due to the growing number of Software-based, technopreneurship activities within the city. The uniqueness in methodological approach used for educating and training graduates from major universities in this city remains undiagnosed, although its impact is already being felt internationally. This paper reports the experiences from students and alumni, who participated in "Hackathon-like" classroom sessions as well as in a "keep-C-alive" gaming competition, as part of their engineering training. Rich qualitative data was collected from in-depth interviews and observation notes while Yin's approach of embedded analysis was exploited for data analysis. The critical ingredients that build up the students' confidence and competence as well as the influence of the educational setting – created by instructors are expounded upon. Recommendations on how to deal with the challenges of innovative teaching within infrastructural constrains are proffered as an encouragement for colleagues in akin situations.**

Keywords—Silicon Mountain; technopreneurship; Hackathon-like classroom; keep-C-alive

I. INTRODUCTION

In the last five years, the city of Buea – Cameroon has seen the influx of numerous undergraduate students, with very little or no computer background, into computer related disciplines. While Europe and America had over a century to cultivate a culture for the discipline, computer related studies is very much at its introductory and experimentation phase in some parts of Cameroon. The Silicon mountain, as it has been acknowledged by the British Broadcasting Corporation[1] and France24[2], is such an affluent environment for observation and experimentation of software engineering training.

This paper reports the experiences of ex-students who participated in long (24 hours) hackathon-like classroom sessions as well as in "keep-C-alive" gaming tournaments. The instructors' decision to introduce and include these components into the training programs was based on their desire to catalyze the grooming and cultivation of a whole new breed of computer engineers, programmers and system testers. Besides making the tedious training of software engineering a fun activity, hackathon-like classroom sessions and "keep-C-alive" gaming tournaments had lasting impacts on the students and shed more light on engineering training methodologies.

"Keep-C-alive" is a programming tournament that was created to accomplish two main goals in an environment where infrastructural resources are almost non existent and rare premium brain talent was readily available. The tournament, served as a practical structure to train third year software engineering students on project management while providing an interactive team environment for first year students to learn and develop their "C" computer programming language skills. The program ran for 5 weeks, with participants working in teams, to challenge each other in head to head quizzes, scoring points for their teams over the period.

In real life, practical experience of a software engineer working on a project to meet a tight deadline or get to the end of a project, working round the clock is a "no brainer". However, some freshmen can only relate to this as an episode of a sci-fi movie. Hackathon-like classroom sessions – dubbed, "24 hours working Sessions", were established in response to students' hunger for knowledge and enthusiasm to explore new technologies. These sessions created the necessary challenges, the appropriate environment and provided the right tools and equipment for students to be engaged continuously, an opportunity for students to get what conventional lectures could never replicate.

These innovative teaching practices together with many other formal and informal, conventional and non-conventional practices have served a pioneering batch of students who have carved out a very rich culture for software engineers in the tropical rain-forests of Africa. The Silicon Mountain community is gaining prominence in the global village, making invaluable contributions to the software engineering discipline in particular and to the world at large.

II. LITERATURE REVIEW

A. Software Engineering Education

Global trends in information and communication technologies demands a paradigm shift in attitude and competence for both developers and users. In response to this demand, organizations have adopted continuous learning strategies that are self-directed, practical and goal-oriented [23]. These strategies are equally being used by academics with sufficient industry experience, in a bid to better prepare their students. Researchers [9,17] have found that simulating the

1 http://www.bbc.com/news/business-36054263

2 https://afrohustler.com/france24-confirms-silicon-mountain/

2377-570X/17 $31.00 © 2017 IEEE
DOI 10.1109/CSEET.2017.26

work environment in class does not only enhance the technical competence of software engineering students but equally equips them with complementary skills, necessary to handle the challenges of a global work environment.

A multi-verse of teaching approaches are adopted around the world, for both students and professionals. These approaches are geared towards helping students understand software concepts and development processes[18]; identify the relevance of course content and thus develop intrinsic motivation and remain engaged [19]. In more practice-oriented courses, students leverage their potentials and interests and are the architects/owners of their project, gaining fulfillment of solving real problems [16, 21]. Delivering such courses requires skill and dedication from course instructors [20].

Today's software engineering education should emphasize requirements management; distributed system architectures, and component-connector architectures as well as project management education, that covers business-related concerns such as cost/benefit analysis, cultural and linguistic issues and market analysis [22].

B. The hackathon experience

Hackathons are events "in which computer programmers and others, involved in software development, collaborate intensively over a short period of time on software projects."[3] They could either be tech-centric or focus-centric, yet organized to be inclusive [12, 4] and in an inspiring environment, with self-organizing groups and non-mandated leaders.

Tech-centric hackathons usually gravitate around a specific technology or group of technologies, with the aim of improving or enhancing the technology and or related products. Such hackathons facilitate the learning of new technologies and gives participants a hands-on experience with these technologies. An excellent case of such tech-centric hackathon is reported in [12] and the findings revealed that background knowledge or interest in the relevant technology stimulates participants' motivation for joining the hackathon. Participants also reported that the hackathon enabled them to recognize a space for collaboration within competition and helped them diversify their perception. Tech-centric hackathons generally have very specific and specialized set of participants.

On the other hand, focus-centric hackathons use software development to contribute to a social issue or address a business objective. Case studies [5, 8] of focus-centric hackathons have been published and their community value are as intriguing as their academic worth. In [5], for instance, a hackathon was used to solve problems faced by homeless people by creating applications for information sharing. The project brought together a wide variety of participants and reported the value of building strong relationships with academic institutions and non-governmental organizations as a major lesson learned. In [8], the benefits of hosting a hackathon are presented. The paper narrates the experiences of Edmonton Public Library – Canada, in hosting their first hackathon, to commemorate the International Open Data Day. The findings revealed that the event helped the library to gain both information and relationships necessary to release meaningful datasets and put itself in an excellent position to understand and respond to the interests and needs of the open data community.

Research [11] has shown that hackathons can be used to de-construct the conception that computer science is a complex and asocial discipline and could stimulate and maintain students' interest in computer related disciplines. The friendly and informal nature of the work environments mandated by the hackathon spirit reduces the work pressure and the like-minded nature of the team eases the work. It is no wonder that the second and third reason why people participate in hackathons are networking and social change; after their desire to learn.

C. Programming contests

Researchers have established the worth of programming contests as an enjoyable blend of theoretical and practical aspects of computer science[14]. Technologically advanced societies have galvanized on the benefits of such contests at national levels [10] enhancing the organization, structure and management of academic tournaments [7] even for children [13]. The pedagogic benefits of participating in programing contests are tremendous and begin as participants prepare for the contest. They will normally work through and discuss sample problems as well as review the official solutions [1]. As they continue to invest themselves and their time into the preparation, participants develop new insights into the problem domain and its related disciplines. Programming contest participation corroborates highly with student enthusiasm for computer science studies and better career prospects after graduation[12]

Garcia-Mateos and Fernandez-Aleman [6] propose a change in the delivery of a programing course such that (i) the final exam is replaced with a series of activities in a continuous evaluation context; and (ii) making those activities more appealing to the students, in the form of a contest. Besides simplifying course management – especially assessment of assignments for populated courses, their approach improves students' self-assessment skills and encourages them to work independently. They also found reduced dropout rate by 27% and double the pass rate for the course. Such achievements are desirable for most courses in higher education.

In general, contests in computer studies stimulates better motivation; increase active learning and encourages autonomous work in participants. In addition, it promotes a kind of reflective feedback from the learning process, which can be harnessed from a research perspective. For instance, a contest on secure coding [2] did not only strengthen participants' appreciation of secure software development but equally tested the level of security of different software. An analysis of data collected during the contest also helped researchers to characterize certain practices relating to secure coding—shorter programs correlated with better security scores. In addition, the contest data revealed that teams with knowledge of different programming-languages wrote more secure codes and teams that were successful in building secure codes were also successful at finding security bugs.

III. METHODS

This paper reports the experiences of ex-students who participated in long (24 hours) hackathon-like class room sessions as well as in "keep-C-alive" gaming tournaments. These approaches to software engineering education are non-conventional at the universities where these students studied. The research participants, procedure and data collection process are described in the following subsections.

A. The participants

Fifteen (15) ex-students (two female and thirteen male) from two universities were purposefully sampled based on their experiences in the phenomena under investigation and their willingness to share these experiences. Thirteen of the participants graduated with a Bachelor of Science (B.Sc.) in Software Engineering, one with a B.Sc. in Telecommunication Systems and another with a B.Sc. in Computer Networks. Both participants who are non-Software Engineering majors participated in the "keep-C-alive" gaming tournaments and hackathon-like class room sessions. The participants have been coded (R1, R2, … R15) on the basis of when their responses to participate in the research were received.

B. Procedure and Data collection

An email was drafted, vetted and sent to a selected number of ex-students to solicit their willingness to participate in the research by sharing their experiences from "keep-C-alive" gaming tournaments and hackathon-like class room sessions. The participants were further engaged in synchronous online chat to enrich the researchers' exploration of their experience.

Embedded analysis (Yin, 2003) of the qualitative data was carried out using Strauss and Corbin (1998) line-by-line approach. The findings are discussed in the proceeding section, grouped under the respective research questions.

IV. FINDINGS AND DISCUSSIONS

The initial research questions were geared towards finding out the memorable aspects of the "Keep-C-alive" tournament, for the ex-students as well as the remarkable points from their participation in the 24 hours sessions. The exploration of the participants' experience highlighted the value of the Silicon Mountain context and infrastructural challenges. The findings from the various questions are synthesized and discussed in the subsections below.

A. How has the Silicon Mountain tech environment shaped you for today's tech world?

The "*Silicon mountain environment has been an eye opener to endless tech opportunities*" (R10), especially "*… to meet other tech oriented minds … [in] this conducive environment …*"(R7). It has "*Created an environment where I can exchange ideas with other … enthusiast … Extended my network …helped me appreciate the strengths of others especially in areas where I am lacking. Transformed my mindset by instilling in me the notion of thinking big and starting small with faith.*"(R3) "*… I also got to know some credible people who were doing great things in industry with their skills and passion. These served as an inspiration for me.*"(R4) More importantly, It gives me "*… a sense of belonging, a sense of purpose, a sense of achievement, … as a lamp light through the many exposures … a motivation towards my tech aspirations.*"(R1).

The "*activities (seminars, conferences and programming meetups) organized by facets of silicon mountain have brought both developing and existing technologies to mind, creating awareness …"*(R12). Students "*… learned the existence of various tools and new technologies [even] as they came up real time…*"(R13) and "*… with many start-ups, [they] could test the knowledge acquired in the classroom in areas such as web development, android & standalone systems.*"

Almost all the participants pointed out the inspiring nature of the silicon mountain environment. Three remarkable quotations from the sixth, eleventh and fifteenth respondents are stated below.

> "*… What I get are motivation and inspiration. You become who you surround yourself with. Being surrounded by people who are making things happen and creating opportunities for themselves is a huge motivation for me …*"(R6)

> "*… I enjoyed their passion in improving technology in Cameroon and their determination to work in teams no matter the numerous obstacles that tech faces in Cameroon*"(R11)

> "*… The competitiveness and oneness of the community has been a great source of motivation.*"(R15)

The major theme from the data is the integration of networking and learning opportunities in an inspiring tech-community. This echoes a synergy between academia and industry and as this synergy is channeled towards the society, it creates a jungle of entrepreneurial opportunities.

B. What were the memorable aspects of the Keep-C-alive tournament for you?

The tournament was structured to give senior students the opportunity to manage a team of junior students. The data reveals three levels of reflection namely: Manager reflecting on management strategy; Manager reflecting on the impact of the tournament on participants and Participants reflecting on the tournament. With respect to strategy, R10 writes "*Tedious and assiduity are key things learnt about tech tasks/jobs in the "Keep-C-alive" project*" and R4 says "*Three things come to my mind ... Collaboration, algorithms and differences.*" and goes further to describe how collaborating with another class mate helped them to jointly manage their teams. "*Ideally each person was supposed to manage the team assigned to them. However, [my classmate] and I joined our teams together and we both managed them as a single team. I was in charge of logistics and planning issues and communication, while he took charge of technical issues for the team like solving bugs and environment issues.*"

It was interesting to see team managers, though students, reflecting on the impact of the tournament on participants. R5 was excited when his team won the tournament "*When my Team won the challenge. it exposes me to introducing C-programming to Young Engineering students Yes my team won…* " and it changes his self-esteem. R11 says "*The most memorable moments during the tournament were: the difficulty in teaching beginners programming and the anxiety and collaboration among the team members to win the tournament*

at all costs". This insinuates that the student had developed some understanding, and hopefully appreciation, of the pedagogic process and could easily develop into a facilitator in industry. R1 expresses similar views *"... in the area of the challenge problems which enable those participating in the competition to explore their thinking limits. In solving these problems using algorithms, it helps develop the logical reasoning of the participants ... the event itself which involves the coming together of those interested in this area ... There is always something to learn from someone during this event. Lots of sharing of ideas which bring about new research interest and extends to generate new teams for further work."*

Participants reflected on the tournament and expressed its educational, transformational and entertaining effects. "*The Keep-C-alive tournament introduced me to computer programming. and i remember though challenging i learned a lot of programming skills given that we had managers from a higher class who helped us. One Motivational aspect of that tournament was the price tag for the winners ...*"(R7). In support of the educational value of the tournament, R8 writes "*Reviewing my C programming concepts in a competitive/sports programming environment was really great. [it] tests your instinctive programming abilities in relation to Algorithm development, programming Language syntax and semantics and I think it's a great way for progressive evaluation of programming experience.*" The "*... programming exercises designed for the teams were challenging, frustrating ... but enriching ...*"(R12) and "*Thanks to "Keep-C-alive" I acquired that keen eye necessary when debugging. I got to contribute to a team and my team won a trophy. It felt like having fun & getting credit for it.*"(R9) In support transformational value of the mentorship implemented in the tournament, R2 writes "*... to be able to work with mentors of various types. That is a memorable aspect I will never forget. Collaborating with new people and improving on my social skills is a great tool ...* ".

The comments from the thirteenth and fifteenth respondents captured the entertaining side of the tournament.

"*The memorable aspects of the Keep-C-alive Tournament i have is the manner in which in made me learn (Basic aspects) the language faster. Knowing that we have to compete with a team the next day could only increase the amount of effort i had to put in to ensure I won and had a good mark too. In addition it was fun and a good way to learn while having fun.*" (R13)

"*Well, it was fun and challenging at same time. The fact that we could learn from seniors was good and also compete amongst ourselves. It was fun and engaging. Though little stressful too. It taught many other things like team management, getting things done, and working with people with different characters*"(R15)

C. What are the remarkable points from your participation in the 24hrs sessions?

For some of the participants (R4, R9, R10), the ability to sacrifice sleep for the sake of set goal and objectives was the main lesson of the 24 hour working sessions. R6 concurs with this view and writes "*... The program also taught me about making sacrifices to accomplish a goal. Sometimes to get ahead in life, sacrifices has to be made which include giving up sleep. ...*". Others (R11) feel that "*… The body needs rest to function effectively; programmers work day and night but a good night sleep will make them more efficient.*"

These long working sessions helped students to become more self-disciplined and reduce procrastination in order to get things done fast enough to meet deadlines. They learned to maintain high focus, eliminate distractions and manage their time well, so as to improve their productivity. Students learned to be flexible, to adapted to different work environments and styles. "*... One remarkable point was the food. I loved it it was always available at any point in time and so it was also a moment of communion*"(R13).

"*The session ... permitted me to kick off my third year final project, I was stocked on how to start, and with the guide lines you gave me and me seating for almost 24hrs working on it helped me greatly to progress very fast, and when I was fully into my topic, I could not stop. I kept on progressing quit rapidly later on as well. The final result is I had an A grade for my project.*"(R14)

D. What infrastructural challenges do technology students in the Molyko area face?

Participants were quick to identify challenges and difficulties they encounter and in follow-up conversations, it was clear that these challenges could be transformed into technopreneurship opportunities. The table below gives a list of challenges highlighted by participants.

Table I. Infrastructural challenges of students

SN	Challenge
1	The absence of reliable and affordable high-speed Internet access
2	Constant electric power outages
3	Insufficient exposure to most recent technologies
4	Lack of public laboratories for R&D in technology
5	A platform to expose skills to target audience
6	Lack of well equipped tech hubs and coding spaces with appropriate ergonomics

V. RECOMMENDATIONS AND CONCLUSION

Running the keep-C-alive tournament and organizing hackathon-like classroom sessions has provided a transforming experience for the entire student population. By the end of the last tournament, the buzz for computer programming among the students and in the community as a whole was hyped. The level of collaborative work within the students superseded anything that had been experienced in any other department. The students went on to participate in other international programming tournaments including ACM ICPC and Google Summer of Codes.

Some of the 24 hours sessions comprised of a trip to an appropriate location or destinations and students gained travel experience. However, in order to balance students' benefits with the social impact [5] of these sessions, The tools,

equipment and infrastructure required; The accomplish-able tasks and modalities for accomplishing them together with the evaluation rubric were carefully documented. As a benefit for hosting/sponsoring the keep-C-alive tournament and hackathon-like sessions, students gained internship placements in some of the sponsoring organizations, which benefited from the quality work of the students.

ACKNOWLEDGMENT

We thank the Silicon Mountain tech-community for their openness and willingness to share their experiences.

REFERENCES

[1] A. Bloomfield and B. Sotomayor, "A Programming Contest Strategy Guide," 2016, pp. 609–614.

[2] A. Ruef, M. Hicks, J. Parker, D. Levin, M. L. Mazurek, and P. Mardziel, "Build It, Break It, Fix It: Contesting Secure Development," 2016, pp. 690–703.

[3] G. Briscoe, "Digital innovation: The hackathon phenomenon," 2014.

[4] A. Cox, M. Fisher, and others, "Gender and programming contests: mitigating exclusionary practices," *Informatics in Education-An International Journal*, no. Vol 5_1, pp. 47–62, 2006.

[5] N. Linnell, S. Figueira, N. Chintala, L. Falzarano, and V. Ciancio, "Hack for the homeless: A humanitarian technology hackathon," in *Global Humanitarian Technology Conference (GHTC), 2014 IEEE*, 2014, pp. 577–584.

[6] G. Garcia-Mateos and J. L. Fernandez-Aleman, "Make learning fun with programming contests," in *Transactions on edutainment II*, Springer, 2009, pp. 246–257.

[7] J. P. Leal and F. Silva, "Mooshak: A Web-based multi-site programming contest system," *Software: Practice and Experience*, vol. 33, no. 6, pp. 567–581, 2003.

[8] A. Carruthers, "Open Data Day Hackathon 2014 at Edmonton Public Library," *Partnership: The Canadian Journal of Library and Information Practice and Research*, vol. 9, no. 2, p. 1, 2014.

[9] S. Beecham, T. Clear, J. Barr, M. Daniels, M. Oudshoorn, and J. Noll, "Preparing tomorrow's software engineers for work in a global environment," *IEEE Software*, vol. 34, no. 1, pp. 9–12, 2017.

[10] K. Manev, E. Kelevedjiev, and S. Kapralov, "Programming contests for school students in Bulgaria," *Olympiads in Informatics*, vol. 1, pp. 112–123, 2007.

[11] J. Mtsweni and H. Abdullah, "Stimulating and maintaining students' interest in Computer Science using the hackathon model," *The Independent Journal of Teaching and Learning*, vol. 10, no. 1, pp. 85–97, 2015.

[12] G. T. Richard, Y. B. Kafai, B. Adleberg, and O. Telhan, "StitchFest: Diversifying a College Hackathon to Broaden Participation and Perceptions in Computing," 2015, pp. 114–119.

[13] A. Idlbi, "Taking kids into programming (contests) with Scratch," *Olympiads in Informatics*, vol. 3, pp. 17–25, 2009.

[14] V. Khera, O. Astrachan, and D. Kotz, "The internet programming contest: a report and philosophy," 1993, pp. 48–52.

[15] R. K. Yin, "Case study research: Design and methods" 2003, Thousand Oaks, CA: Sage (3rd ed.).

[16] N. D. Nyinkeu, C. T. Katiba and N Henry. "Project-based teaching for information technology students in Africa: A case report" *International Journal of Technology in Teaching and Learning*, vol. 11, no. 2, pp130-139. 2015.

[17] M. B. Blake, "A student-enacted simulation approach to software engineering education," *IEEE Transactions on Education*, vol. 46, no. 1, pp. 124–132, 2003.

[18] E. Ye, C. Liu, and J. A. Polack-Wahl, "Enhancing software engineering education using teaching aids in 3-D online virtual worlds," in *Frontiers in education conference-global engineering: knowledge without borders, opportunities without passports, 2007. FIE'07. 37th annual*, 2007, p. T1E–8.

[19] M. L. Junhua, "Innovations in Software Engineering Education: An Experimental Study of Integrating Active Learning and Design-based Learning," *age*, vol. 23, p. 1.

[20] R. Chatley and T. Field, "Lean learning: applying lean techniques to improve software engineering education," in *Proceedings of the 39th International Conference on Software Engineering: Software Engineering and Education Track*, 2017, pp. 117–126.

[21] S. Huang and D. Distante, "On Practice-Oriented Software Engineering Education," in *Software Engineering Education and Training Workshops, 2006. CSEETW'06. 19th Conference on*, 2006, pp. 15–15.

[22] M. J. Hawthorne and D. E. Perry, "Software engineering education in the era of outsourcing, distributed development, and open source software: challenges and opportunities," in *Proceedings of the 27th international conference on Software engineering*, 2005, pp. 643–644.

[23] H. J. Ellis and G. W. Hislop, "Techniques for providing software engineering education to working professionals," in *Frontiers in Education, 2004. FIE 2004. 34th Annual*, 2004, p. F1C–19.

[24] A. Strauss, and J. Corbin, "Basics of qualitative research: Techniques and procedures for developing grounded theory" (2nd ed.). Thousand Oaks, CA: Sage. 1998.