A New Paradigm for Teaching Mobile Application Development

Youry Khmelevsky[*]
Computer Science Department
STH, Okanagan College
1000 KLO Rd., Kelowna, BC V1Y 4X8
ykhmelevsky@okanagan.bc.ca

Volodymyr Voytenko
School of Applied Computing
Sheridan College Institute of Technology
7899 McLaughlin Rd, Brampton, ON L6Y 5H9
volodymyr.voytenko@sheridancollege.ca

ABSTRACT

We present a new paradigm for teaching mobile application development, focusing on software development and software engineering capstone projects with industrial sponsors. We support our new approach by showcasing a number of student projects conducted at three different post secondary institutions, which includes applications development for different mobile platforms.

In summary we discuss how to improve mobile applications development, and summarize new paradigm for mobile application development implementations in undergraduate programs based on our 5-7 years of our teaching experiences with mobile applications development in different courses.

CCS Concepts

•Human-centered computing → Mobile devices; •Social and professional topics → Computer science education; •Software and its engineering → Object oriented development;

Keywords

Algorithms; Design; Performance; Reliability; Experimentation; Security; Human Factors; Standardization; Languages, Verification.

1. INTRODUCTION

"In 2015, global mobile app revenues amounted to 41.1 billion U.S. dollars". The projection for 2020 is over 101 billion U.S. dollars [1].

> "With smartphones and other web-enabled mobile devices growing quickly in popularity, higher education institutions have the opportunity - and challenge - to leverage this young medium to provide dynamic content regarding services, events, and support documentation, among others" [2].

Apps programming for smartphones and mobile devices is still a hot topic nowadays. Colleges and universities are actively looking to implement new related courses:

> "Students are excited about and are asking for courses in it and faculty are scrambling to build those courses. Departments see mobile application development as a way to attract new students to computer science while providing a fertile environment for teaching fundamental computer science topics like memory management, networking, human-computer interaction, etc." [3].

During the last few decades mobile phones, tablets, different gaming consoles and gadgets have become widely adopted. On the other hand, many mobile applications can be used on different platforms, including traditional PCs and even servers [4]. Mobile devices and especially mobile phones provide new opportunities for learning "embedded software concepts, techniques and practices" [5].

Software development and software engineering curriculums have become traditional at computer science departments in the last few decades [6]. Students should learn software and system design for mobile applications, mobile devices architecture, and software development process as early as possible [7], [8].

2. RELATED WORKS

The instruction of mobile application development should include: (a) "Start Early, But It is Never Too Late", (b) "Set Up a Virtual Environment", (c) "Provide Students with Mobile Devices" and (d) "Instructor Resources" [9]. Authors in [10] inform, that mobile computing:

> "has deeply affected the way of teaching computer science. Many mobile devices (e.g., smartphones, tablets, etc.), with the powerful CPUs and the amount of memory they have, cannot be considered limited environments anymore. The main differences compared to traditional environments are: limited battery life, unreliable (and with variable speed) wireless connection, interaction with the physical world (sensors like accelerometer, gyroscope, barometer, etc.). The

[*] Also Affiliated with Computer Science, UBC Okanagan Campus, Kelowna, BC Canada

WCCCE '16, May 06-07, 2016, Kamloops, BC, Canada
© 2016 ACM. ISBN 978-1-4503-4355-8/16/05...$15.00
DOI: http://dx.doi.org/10.1145/2910925.2910937

> evolution of mobile devices is different from the traditional desktop environments as new kinds of devices are appearing in the marketplace that are not general-purpose but their usage is focused (e.g., tracking fitness activities)" [10].

Some challenges of teaching mobile applications are "too complex and too big" [11]. Students like to have more practical sessions with a tutor, just to follow practical examples. The authors reported that time restrictions are very critical and really they could not feasibly implement everything they planned.

Authors in [12] describe "an effective way of introducing iPhone application development to undergraduate students". They found that the students "can quickly grasp the essentials in iPhone application development. These include the unique syntax features in Objective-C and new Apple's programming language Swift, the iPhone SDK and core frameworks, development tools, and design patterns".

3. MOBILE APPS DEVELOPMENT IN OUR COURSES AND PROJECTS

At Okanagan College (OC) we involved industrial sponsors since 2005 in both our programs: Bachelor of Computer Information Systems Degree (BCIS) [13] and Computer Information Systems Diploma (CIS) [14]. Brief information about CIS/BCIS courses can be found online[1].

Currently we involve industrial sponsors starting from the first capstone project course COSC 224 "Projects in Computer Science" (CIS/BCIS programs) in Winter, but students begin project requirements analysis and initial design in the Fall in COSC 236 "Object-Oriented Systems Analysis and Design". In the BCIS program students have COSC 470 "Software Engineering" and COSC 471 "Software Engineering Project" in which they start "design and implementation of large, multi-module-program systems" [15]. In the last decade the Computer Science department was able to offer 2 additional courses every second year for mobile devices software development, COSC 419 "Topics in Computer Science: iOs App Development" for iPhone platforms and COSC 419 "Topics in Computer Science: Android App Development" for Android platforms.

The first capstone projects with mobile devices started in 2009 from a small in-kind grant from Nokia [16]. Students were able to develop a prototype with an industrial client from Europe and were able to present research results at IEEE's professional SPCloud workshop in Europe [16]. In the following years students selected more and more mobile components in the capstone projects. We have tried to combine mobile parts in the COSC 470/471 capstone projects with the small mobile projects in COSC 419, but we found that such a combination doesn't work well. In COSC 419 students should learn how to do programming for mobile devices at first and only then they can start project development for mobile devices. We consider small projects in COSC 419 as a 'Spike' project before the projects in COSC 470/471. In COSC 419 students learn about different mobile platforms' applications development. We found that we have 2 separate group of students, one of the groups likes Android applications development, the other group likes Apple mobile applications development. A few years ago, students in COSC 470/471 insisted to develop a mobile part of the project for both platforms in spite of the industrial sponsor request to support just iPhone/iPad platforms for the mobile client. Additionally during the last few years we included Amazon EC2 servers [17] as a back-end for the student projects [18]. On the other hand, when we have privacy concerns from industrial clients we use our own Hybrid Cloud resources at OC [18], [19].

Starting from 2011 we are offering a list of 30-40 project topics to students from industry and academia, but they choose which project to do. Starting from 2013 we were awarded five grants, including four grants supported by "The Natural Sciences and Engineering Research Council of Canada or NSERC". The last research project has just started on February 1st, 2016, which will last for 3 years, until February 2019. Almost all NSERC grants include mobile device developments, mobile devices testing, or both. In the last couple of years, some students in the COSC 224 2nd year capstone project chose mobile application development even without any previous background in mobile application development. More information about all our software development projects can be found at [20], [21], [22].

Sheridan College (Sheridan) offered mobile application development courses in 2010 within the "Computer Systems Technology - Software Development and Network Engineering" programs [7], which made mobile devices a core topic in CS education at Sheridan. iPhone, Google's Android, and Blackberry's OS platforms are taught at Sheridan, including Java ME, Java SE, and Objective-C programming languages. In the Fall of 2014 Sheridan launched the Mobile Applications Development Degree program.

In the new Sheridan's program, students learn iOS applications development using "Objective-C (PROG 31632), applications with MIDP and CLDC support, including BlackBerry/BB10 platform (PROG38448) and complex Android applications using Java SE and Android SDK (PROG39402)" [7]. Sheridan uses simulated and emulated environments within the mobile development courses, but students practice with real devices - smartphones and tablets. Sheridan hosted several annual Mobile Computing Faculty Summits in 2011 and 2012, supported by RIM, TELUS and Xtreme Labs Inc.

At OC students liked learning a new programming language Swift, from Apple, and they were able to finish the COSC 224 capstone project development with Swift in 2015. We will discuss it in the following section. "It's still a challenge today to make mobile devices and mobile application development a core topic in our Computer Science education curricula" [9]. We should support our students strong desire to learn mobile applications development as early as possible, immediately after the first year's introductory programming and data structure courses.

3.1 Mobile Apps Development in Software Engineering Capstone Projects

Besides our mobile courses, we significantly increased usage of mobile application systems and components in our software development, engineering projects and our graduate capstone projects. Those projects promoted student engagement in hands-on research, significantly improving their knowledge and skills, especially those related to team management and industrial clients collaboration.

The first hurdle that both students and faculty face is

[1] http://tinyurl.com/ztcwyan

the topic selection, the topic chosen must relate to mobile device implementation. The key of an application development project is to find a target application that solves real life problems. Because of the nature of the problem, the first group of the projects require to develop only the mobile application version. In this case the desktop application version is not suitable or would not bring any benefits to the client. One of the examples belongs to this category - a wide spectrum of context-aware applications. For this projects, students used different device sensors (motion, environmental, position, etc.) and communication (including users' interaction, identification and personalization) to understand the user's context. These components work together to continuously acquire, model, and notify the user of the mobile application. The recognition of the context and automation of anticipated response by offering contextually relevant information and service is the main challenge of context-aware systems.

Having the mobile context advantage, students face higher level of technological complexity and a number of problems. Many context information sources may be different, unreliable, sometimes unavailable or replaced dynamically. Secondly, one of the most important issues is a task of deriving high-level context information by constructing low-level context information. This can be very complex and sometimes impossible.

In this case, potential errors could occur because of a wrong interpretation of the context. And finally, with the development of context-aware applications, the privacy issue is increasingly capturing our attention. To address those challenges we have to use an encryption mechanism for data and to make sure that the sandboxing mechanism works correctly on a device, in order to guarantee that malware without root privileges cannot access data stored in an applications context.

The second project type - the situation when students require to design and develop mobile and desktop applications at the same time, but for different product functions. For example, the desktop application could be responsible for application administration, server-side database management and support, and user-side mobile application provides a friendly and reliable client interface to make requests to the server.

Finally, the third type of the projects may include a mobile and a desktop standalone version at the same time, and they should have almost the same capabilities. It is very important that the students understand that building a mobile application version is not just a scaling down a PC application version to a different screen size. We have to design and create many appropriate mobile user interfaces by using the same design model(s) and the same access code. Experience has shown that it is near impossible to find an appropriate approach that solves the problem perfectly. As practice showed, none from approaches perfectly fits all solutions in this case. Students work through exercises in product brainstorming, sketching, prototyping, designing, user testing, and system architecting.

4. SELECTED STUDENTS' PROJECTS

During the last 7 years at OC, Sheridan and UBCO we had many small, mid-size and large capstone projects with mobile platform applications development. All projects at OC as well as at UBCO had industrial sponsors (projects, which were supervised by the co-author of this paper) [7, 18, 19, 20, 21, 23]. All industrial sponsors were requested to provide a reference for the project(s) and they were invited for the final project presentations. Unfortunately, not all industrial sponsors were able to be present during the final project presentations. Usually the chair of the department and faculty members evaluated student projects and even in several cases the College administration and public were invited for the project presentations. After the project presentation professors discussed the project results internally at the department. Some of the best projects with mobile applications are discussed below.

4.1 An Astronomical Calendar Apps for iPhone / iPad

An apps-server Astronomical Calendar for iPhone / iPad was developed by the students at UBCO in COSC 319 Software Engineering project course (Winter 2012) [7], but another off-line Astronomical Calendar app was developed by OC students using the new Apple programming language called Swift in COSC 224 (Winter 2015). Both apps have almost the same implemented functions and the same goals, but the first one is using on-line Swiss Ephemeris [24], hosted on a server at OC. The second one uses a local database on iPhone to search astronomical data for the specific date and time. In the first application the computational resources of iPhones in 2012 were not enough for the full implementation on iPhone.

4.1.1 An Apps-Server Calendar

This mobile application uses the same Swiss Ephemeris [24] mathematical library. On the other hand, students were able to port a Java based Astronomical Calendar, what was developed by the COSC 310 students in 2009 at UBC O. The developed Java application was published online at Source Forge. The project development duration was around one and half months. Students used a LAMP stack on the server side for iPhone calculations support.

In the following semester the same team was able to finish a more complicated iPhone and Android gaming platform in COSC 319 [7].

4.1.2 A Swift Based Off-Line Astronomical Calendar

Another team at OC was able to implement and publish on the Apple Store a different implementation of the same application by using the new Apple programming language called Swift in 2015 within the COSC 224 capstone project course. It was a brand new project, including a new design and new implementation. Students used just an idea from previous projects. A team of 4 students were able to import about 400MB of NASA ephemeris into iPhone and developed an API for the astronomical tables. In this project students demonstrated that new mobile applications have enough CPU resources to avoid middle-ware applications for computational tasks. The application calculated sun-rise and sun-set, as well as moon-rise and moon-set for the selected location and printed a table of the calculated results on the iPhone screen. In previous projects, students relied on the server applications, but now everything can be done on an off-line mobile phone.

4.2 A Mobile Personal Financial Platform

A Mobile Personal Financial Platform project was developed in 2012 for both Android and iPhone. Some technical and management problems arose during the project development. The PhoneGap framework was utilized by the students during the project. They finished a prototype for the Android platform. They lacked project management skills and have never before developed any applications for mobile devices in the past. But the project prototype was finished successfully and on time [7].

4.3 Mobile Network Analyser for WTFast

A local industrial client within COSC 470/471 SW Engineering project course and COSC 224 Capstone Project course requested the creation of a mobile Android/ iPhone application for an existing customers' Gamer Private Network (GPN) application in 2014. The primary use of the Mobile Application was to provide a service of quality control for the gamers. A COSC students conducted the initial research in COSC 236 Object Oriented Analysis and Design course in 2014 and then in 2015 a prototype was developed.

4.4 Other related projects

Mobile Inventory Tracking System capstone project for the iPhone/iPad and Android was finished by the COSC 470/471 students in the Winter 2011.

Another project team of 10 students finished in the Winter 2013 (COSC 471) a "New Data Analytical" project for both mobile platforms (iPad/Android) with backend server in the Cloud (Amazon's EC2). The project was started in COSC 470 in the Fall of 2012.

The first capstone project with industrial client WTFast was started in the Fall of 2013.

5. CONCLUSION

As we discussed in [7] and as it was confirmed in [25] and in [9]:

- Students enjoyed capstone projects with mobile devices and industrial clients. We considered it as the project courses improvement.

- We found, that almost every capstone project can include a mobile application development [7]. We were able to achieve this goal in the last several years, starting from 2011. More and more students enjoy combing client-server, n-Tier and app-server architectures within the projects. We are flexible enough to accepts suggestions from the students during the capstone projects and especially during the final project presentations and within anonymous capstone project reviews. Our students suggested:

  - To extend COSC 224 in Winter by starting projects analysis and design in the Fall's COSC 236. It added more workload to the students in COSC 236, but reduced students workload within the following COSC 224. In this new capstone projects approach students have better motivation to learn abstract models, design diagrams and design patterns.

  - In the two first years of CIS/BCIS programs at OC we don't have room for mobile applications development courses. Traditionally such courses should be started from year 3, but our most ambitious and strongest students started mobile applications development learning at the same time when they take COSC 236 (or maybe even earlier) by themselves. They usually have enough time from September to December to learn basic principles and in many cases new programming languages by themselves for their first simple mobile applications development in the following COSC 224 capstone project course. Instructors mentor them and help with the analysis, design, project documentation and testing/evaluation.

  - In our research projects, especially the ones supported by NSERC grants, students start to work with mobile devices even from the 1st year of study (2nd semester). Some students come and ask for a textbook(s) or professional books from the departmental library just to learn different topics ahead, in addition to the taken courses.

The students capstone projects with mobile applications development components may be enhanced by the following:

- We should offer an application development course for mobile devices in the 2nd year [7].

- We should support students by mobile phones and tablets, but not only by software emulators in class [9]. We were able to purchase 24 CodeRebel concurrent licenses for the Apple xServer (Dual Quad Core 2.4 GHz server with 3 TB Raid 5 and 48 GB Ram) and eventually in 2014 we purchased 20 Mac Minis for the apps development. For testing purposes we use iPhone and iPad. Many students already have Android devices.

- We should support instructors with software development resources for the mobile application development courses, including mobile devices and development tools [7].

Additionally, we would like stress on the following new approaches in mobile and capstone project courses:

- "Teaching students how to develop applications for one particular platform in 'a one size fits all' method is no longer a viable solution" [12]. In the last several years students in capstone projects almost always made a decision to develop applications for both platforms, for Android and for iPhone / iPad, even if the client wants to see only a "PC solution" or just 1 mobile platform.

- The native mobile applications have better performance to compare with Java Script or HTML 5 in many cases. It's nice to have one apps for both Android and iPhone, but practically our industrial clients reported very slow performance and bad scalability.

- Using different types of Clouds in all our capstone projects, including Private, Public and Hybrid Clouds improves motivation for the students [18]. EC2 from Amazon is a very good example of the Private Cloud platform for the student projects. Atlassian is another example of the Private Cloud applications, which offers free Cloud resources for version control (BitBucket with SourceTree) and Jira with Confluence for the software engineering project management.

- In mobile applications development, Cloud should be used for permanent storage mostly, including encryption on both client and Cloud sites. All processing should be done mostly on mobile devices. Many mobile devices have enough processing power for many applications, but network interruption or lack of Internet services (or too expensive services) create problems for the networked applications on mobile devices.

- Middleware application servers for mobile devices should be used only when it's really necessary. We had several projects in the past where students developed mobile applications with the main processing part on a middle server. The developed projects were dead in 6 - 12 months, because the middle-ware or application servers require additional support and maintenance. This is a typical problem with the mobile applications development.

- All data processing and computations may be done on mobile devices in many cases, but Cloud may be used for the data exchange and for endusers data storage. In many cases it's better to use native programming languages or code generators like Lua for both Android and for Apple mobile platforms in multi-platform projects.

- All data and traffic must be encrypted on both sides and even during data transmission, but this is another challenge for the students.

- Mobile devices can share processing power, especially by using Peer-to-Peer ("P2P") technology. On the other hand it's better to avoid integration of mobile computational resources via Internet.

- Another emerging trend is coming when mobile devices need only a data plan without a voice plan. This gives a lot of opportunities in developed countries, where data plans for mobile devices are very cheap. Telephone or VOIP communications can be used though the data plan only.

- A typical bottleneck in mobile software development is the middleware and internet now, but computational power is doubled or even quadrupled every year in accordance with Moore's Law.

- Students developed a new version of an astronomical calendar without middleware by using NASA ephemeris tables with the size of about 400MB in 2015. That means we are already ready to deploy large applications with heavy computational requirements just on widely available mobile devices.

- If mobile devices still need additional computational power and additional large storage online a transition from permanent online to periodically synchronized storages could be implemented via WiFi sessions on mobile devices via internet or by P2P technology.

- User experience improvement requires new approaches with mobile applications development in the students capstone projects, including new Mobile-Cloud applications development. "It overcomes obstacles related to the performance in mobile environment, guaranteed for the availability of resources, assures security, provide intelligent services, transforming the internet computing infrastructure, and broad network access to mobile users" [26].

6. ACKNOWLEDGMENTS

Our thanks to all Computer Science students at OC, who participated in the capstone project courses COSC 224 in CIS/BCIS and COSC 470 / COSC 471 in BCIS programs. Many student projects were in support to NSERC CCI ARD and Engage College grants, including "GAUGE: Exact Positioning Systems For Sport and Healthcare Industries" as well as GPN-Perf (2014) and GPN-Perf2 (2016-2019) grants.

Our thanks to "Amazon Web Services, Inc." for the great development, software engineering capstone and research projects support. We are grateful for a research grant from "AWS Grants for Research and Education". We would like to thank Atlassian for supporting us with Jira, Bitbucket, Bamboo and SourceTree, as well as "IBM Education" for the IBM RSA licenses, what we used for the project analysis and project design.

We thank our reviewers for the comments and the helpful advise on how to improve our paper.

7. REFERENCES

[1] Statista, "Worldwide mobile app revenues in 2015, 2015 and 2020 (in billion u.s. dollars). http://www.statista.com/statistics/269025/worldwide-mobile-app-revenue-forecast/."

[2] B. Siegfried, "Enhanced student technology support with cross-platform mobile apps," in *Proceedings of the 39th Annual ACM SIGUCCS Conference on User Services*, ser. SIGUCCS '11. New York, NY, USA: ACM, 2011, pp. 31–34. [Online]. Available: http://doi.acm.org/10.1145/2070364.2070373

[3] P. E. Dickson, "Cabana: A cross-platform mobile development system," in *Proceedings of the 43rd ACM Technical Symposium on Computer Science Education*, ser. SIGCSE '12. New York, NY, USA: ACM, 2012, pp. 529–534. [Online]. Available: http://doi.acm.org/10.1145/2157136.2157290

[4] J. Bishop, "Devices and services meet mobile development," in *Proceedings of the 2013 ACM Workshop on Mobile Development Lifecycle*, ser. MobileDeLi '13. New York, NY, USA: ACM, 2013, pp. 35–36. [Online]. Available: http://doi.acm.org/10.1145/2542128.2542139

[5] J. K. Muppala, "Teaching embedded software concepts using android," in *Proceedings of the 6th Workshop on Embedded Systems Education*, ser. WESE '11. New York, NY, USA: ACM, 2011, pp. 32–37. [Online]. Available: http://doi.acm.org/10.1145/2077370.2077375

[6] D. Petkovic, G. Thompson, and R. Todtenhoefer, "Teaching practical software engineering and global software engineering: Evaluation and comparison," in *Proceedings of the 11th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education*, ser. ITICSE '06. New York, NY, USA: ACM, 2006, pp. 294–298. [Online]. Available: http://doi.acm.org/10.1145/1140124.1140202

[7] Y. Khmelevsky and V. Voytenko, “Strategies for teaching mobile application development,” 18th Western Canadian Conference on Computing Education, Ed. North Vancouver, BC: Capilano University, May 3rd 2013. [Online]. Available: http://wccce2013.ca/wp-content/uploads/2013/05/wccce2013_submission_8.pdf

[8] H. Mitsui, H. Kambe, and H. Koizumi, “Use of student experiments for teaching embedded software development including hw/sw co-design,” *Education, IEEE Transactions on*, vol. 52, no. 3, pp. 436–443, Aug 2009.

[9] Q. H. Mahmoud, “Best practices in teaching mobile application development,” in *Proceedings of the 16th Annual Joint Conference on Innovation and Technology in Computer Science Education*, ser. ITiCSE ’11. New York, NY, USA: ACM, 2011, pp. 333–333. [Online]. Available: http://doi.acm.org/10.1145/1999747.1999844

[10] J. Bishop, M. Hauswirth, A. Sillitti, and S. Stokes, “Mobile computing and education: (panel),” in *Proceedings of the 2nd Workshop on Programming for Mobile & Touch*, ser. PROMOTO ’14. New York, NY, USA: ACM, 2014, pp. 33–34. [Online]. Available: http://doi.acm.org/10.1145/2688471.2688483

[11] P. Alston, “Teaching mobile web application development: Challenges faced and lessons learned,” in *Proceedings of the 13th Annual Conference on Information Technology Education*, ser. SIGITE ’12. New York, NY, USA: ACM, 2012, pp. 239–244. [Online]. Available: http://doi.acm.org/10.1145/2380552.2380620

[12] B. Yan, D. Becker, and C. Hecker, “An effective way of introducing iphone application development to undergraduate students,” *Journal of Computing Sciences in Colleges*, vol. 26, no. 5, pp. 166–173, 2011.

[13] Okanagan College. (2016, April) Bachelor of Computer Information Systems Degree program description. [Online]. Available: http://www.okanagan.bc.ca/Programs/Areas_of_Study/science/computer-science/Programs/BCIS_degree.html

[14] ——. (2016, April) Computer Information Systems Diploma program description. [Online]. Available: http://www.okanagan.bc.ca/Programs/Areas_of_Study/science/computer-science/Programs/CIS_Diploma.html

[15] ——. (2016, April) BCIS and CIS course descriptions. [Online]. Available: http://webapps-5.okanagan.bc.ca/ok/Calendar/CourseDescriptions?COSC

[16] R. Grmek, Y. Khmelevsky, and D. Syrotovsky, “Automated inventory tracking system prototype in cloud,” in *High Performance Computing and Simulation (HPCS), 2011 International Conference on High Performance Computing & Simulation.* Istanbul, Turkey: In Cooperation with the ACM, IEEE, IFIP, Co-Sponsored by IEEE Turkey, ASIM, EUROSIM, CASS, JSST, LSS, PTSK, TSS, Bahcesehir University, July 4-8 2011, pp. 435–441.

[17] Amazon Web Services, Inc. or its affiliates. (2016, April) Amazon EC2 - Virtual Server Hosting.

[18] Y. Khmelevsky and V. Voytenko, “Hybrid cloud computing infrastructure in academia.” in *WCCCE 2015 - the 20th Western Canadian Conference on Computing Education, At May 8-9, 2015.* Vancouver Island University (VIU), Nanaimo, British Columbia, Canada., 2015.

[19] Y. Khmelevsky, “SW Development Projects in Academia,” in *Proceedings of the 14th Western Canadian Conference on Computing Education*, ser. WCCCE ’09. New York, NY, USA: ACM, 2009, pp. 60–64. [Online]. Available: http://doi.acm.org/10.1145/1536274.1536292

[20] ——, “Research and teaching strategies integration at post-secondary programs,” in *Proceedings of the 16th Western Canadian Conference on Computing Education*, ser. WCCCE ’11. New York, NY, USA: ACM, 2011, pp. 57–60. [Online]. Available: http://doi.acm.org/10.1145/1989622.1989638

[21] Y. Khmelevsky, V. Ustimenko, G. Hains, C. Kluka, E. Ozan, and D. Syrotovsky, “International Collaboration in SW Engineering Research Projects,” in *Proceedings of the 16th Western Canadian Conference on Computing Education*, ser. WCCCE ’11. New York, NY, USA: ACM, 2011, pp. 52–56. [Online]. Available: http://doi.acm.org/10.1145/1989622.1989637

[22] Y. Khmelevsky, G. Hains, and C. Li, “Automatic code generation within student’s software engineering projects,” in *Proceedings of the Seventeenth Western Canadian Conference on Computing Education*, ser. WCCCE ’12. New York, NY, USA: ACM, 2012, pp. 29–33. [Online]. Available: http://doi.acm.org/10.1145/2247569.2247578

[23] Y. Khmelevsky, L. Burge, M. Govorov, and G. Hains, “Distance learning components in cs and gis courses,” in *Proceedings of the 16th Western Canadian Conference on Computing Education*, ser. WCCCE ’11. New York, NY, USA: ACM, 2011, pp. 17–21. [Online]. Available: http://doi.acm.org/10.1145/1989622.1989627

[24] M. Arges Nadin, “Rendezvous with the stars and the universe, too,” *Journal of the Royal Astronomical Society of Canada*, vol. 107, p. 202, 2013.

[25] J. Barbosa, R. Hahn, D. N. F. Barbosa, and C. F. R. Geyer, “Mobile and ubiquitous computing in an innovative undergraduate course,” in *Proceedings of the 38th SIGCSE Technical Symposium on Computer Science Education*, ser. SIGCSE ’07. New York, NY, USA: ACM, 2007, pp. 379–383. [Online]. Available: http://doi.acm.org/10.1145/1227310.1227444

[26] K. Mohiuddin, A. Islam, A. Alam, and A. Ali, “24x7x365: Mobile cloud access,” in *Proceedings of the CUBE International Information Technology Conference*, ser. CUBE ’12. New York, NY, USA: ACM, 2012, pp. 544–551. [Online]. Available: http://doi.acm.org/10.1145/2381716.2381820