The Life Cycle of a Cutting-edge Technology Course
A Coaching Experience on Android[*]

Maria J. Blesa, Amalia Duch, Joaquim Gabarró and María J. Serna
ALBCOM Research Group
Universitat Politècnica de Catalunya - Barcelona Tech, E-08034 Barcelona, Spain

Keywords: Android, Programming, New Technologies, Google, Smart Phones, AndroidEDU, Web 3.0.

Abstract: What is the role that a university should play in the spreading of cutting-edge technologies? It is argued here that one possibility is to bring focused cutting-edge technology courses in the standard curriculum. It is contended that such courses have shorter life-spans than conventional subjects and, consequently, their implementation needs to be more dynamic. These claims are backed by discussing the *life-cycle* of an Android course running biannually from Spring 2010 to Spring 2013 at Universitat Politécnica de Catalunya. The *rise* phase of this course (which lasted two semesters) was a challenging experience that motivated students and lecturers to play a cooperative and active role in the creation of true working Android applications. The course held *stable* for two semesters while student motivation began to fall as smart phones increasingly became everyday objects. During these two phases the course was offered as *extra curricular* in the undergraduate phase. Two added factors were instrumental in the *decline* (or *fall*) phase: the availability of on-line information and the fact that the course became a requirement of a master's curriculum.

1 INTRODUCTION

Concrete IT technologies shapes our believes and behavior (Norman, 1988; Norman, 1998) and in particular, the media affect the way of teaching (Postman, 1984). From an economic point of view, technological development determines (at least in part) the role of countries in recent history. In particular the rise and fall of technological leadership deserves careful study (Nelson and Wright, 1992). In fact, the degree of technological development of a country seems to be influenced by the way in which technology has been taught (see Section B of (Nelson and Wright, 1992)). So, the importance of the role of cutting-edge technology courses is recognized not only among economists and historians but also among participants in educational meetings (Frank and Waren, 1982):

> *One of the more difficult tasks in this era of adopting curricula, is to keep a program current with technology.*

This paper analyzes the complete life-cycle (rise, stability and decline) of the course: Programming Workshop on Android Applications for Google Phones (PWAAGP from now on). This course was a cutting-edge technology course based on Android and Google phones. It originates from a synergy between lecturers from Universitat Politécnica de Catalunya and Google. Due to the very special characteristics of PWAAGP , the life of this course was short (Spring 2010-Spring 2013). Lecturers experienced such a course as an exciting (sometimes exhausting) experience in a highly volatile environment and they view its evolution as a personal challenge. The purpose of this paper is then to explain why the authors believe that such a course has necessarily a short life-cycle life and that its fall is not a failure of the organizers but a consequence of the intrinsic nature of the course. Moreover, it is claimed that many of the situations encountered in PWAAGP are common to any other IT cutting-edge technology courses.

To the best of our knowledge, there is information about the life-cycle of various teaching approaches using IT technologies as tools (Bryan, 2013). However, information about the natural life-cycle of courses on IT cutting-edge technologies is scarce.

[*]This work was supported by project 2009-SGR1137 of the Generalitat de Catalunya and grants TIN2012-37930 and TIN2007-66523 of the Spanish Government.

J. Blesa M., Duch A., Gabarró J. and J. Serna M..
The Life Cycle of a Cutting-edge Technology Course - A Coaching Experience on Android.
DOI: 10.5220/0004841301590166
In *Proceedings of the 6th International Conference on Computer Supported Education* (CSEDU-2014), pages 159-166
ISBN: 978-989-758-021-5

2 CONTEXT AND GOALS

Nowadays it is well established that universities should be a driving force for innovation. A technical university is a living and multi-headed organization and a fundamental issue to considered is how it deals with cutting-edge technologies and what is the role it should play in their standardization.

Aware of this fundamental need a team of lecturers (belonging to the ALBCOM research group of the Departament de Llenguatges i Sistemes Informàtics of the Universitat Politécnica de Catalunya decided to participate in Google's *AndroidEDU EMEA* program (AndroidEDU EMEA, 2013) in November 2009. *Google* launched a call for project proposals to universities through its *AndroidEDU EMEA* program (AndroidEDU EMEA, 2013), the aim of this program was to promote the development of teaching experiences using the *Android* operating system for mobile phones. To participate in this program, a university site had to submit a proposal for the development of an innovative system in mobile engineering that, requiring a limited number of mobile smart phones, could be addressed to a broad audience. The proposed system should also be scalable, and the proposal should promote the practical teaching of computer science issues for students.

The proposal submitted by the authors of this paper was one of the selected ones, and to facilitate its implementation *Google* donated 20 smart phones to the Departament de Llenguatges i Sistemes Informàtics of the Universitat Politécnica de Catalunya.

This is the origin of the *Programming workshop on Android applications for Google Phones* (PWAAGP). Due to the special typology of both the topic and the students, this course had a life-cycle different from regular technical courses.

This course is used as an example of a focused cutting-edge technology course with the belief that the experience obtained on this course is applicable to other similar cases. It is also claimed here that a main characteristic of this kind of courses is that they have a very short life-cycle that corresponds to the time needed by the technology to spread and become popular.

The life-cycle of this course is characterized into three shorter periods that are called *rise*, *stability* and, *decline* (or *fall*) and that follow the progression of Android from cutting-edge IT technology towards its market take-up.

3 ORIGINAL COURSE

The PWAAGP course started as an *extra curricular* subject that could be taken by students from different majors or specialties. The quota of students in the workshop was restricted by the number of phones we had. PWAAGP was a 4-ECTS[2] very practical course that required from enrolled students to have a good knowledge of Java programming language and a good general knowledge of algorithms and data structures. The course ran for ten weeks, and it was scheduled so that every week a new topic is explained through various examples and putting special emphasis in their implementation. Every week, the complexity of the issues treated was gradually increased.

Having these premises in mind, we fixed the course contents as described in the following subsection. Those contents conformed the main elementary fundamentals on which an application (or App) can be implemented. Although the contents remained almost fixed over time, that was not the case for the teaching strategy and methodology used. Conditioned by how the profile of the students was changing, and by how the technology was getting popular, we had to adapt the way those contents were taught and, consequently, they way the students were evaluated.

3.1 Contents

The first session was dedicated to the installation of the programming environment and all the development tools needed along the course. The applications and the widgets make up the highest layer in the project architecture of the Android system. The second session of the course explored the structure of these applications and presented more advanced widgets (e.g., static and dynamic lists). The topic of the third session was location. It included the embedded use of the GPS, together with the use of some hybrid web applications combining maps with other types of information. Session four introduced the SQLite, the small and powerful SQL database engine included in Android. SQLite is also used in PHP, iPhone, Skype, Mozilla Firefox, etc.).

Android inherits the *threads* directly from the Java programming language. Threads allow the execution of several tasks in parallel. This feature is especially useful when programming mobile phones, where of-

[2]ECTS states for European Credit Transfer System, which is a standard for comparing the study attainment and performance of students of higher education across the European Union and other collaborating European countries. One academic year corresponds to 60 ECTS-credits that are equivalent to $1500-1800$ hours of study.

ten different activities are running at the same time in the device. Threads also help on doing a more efficient and user-friendly management of the applications. Threads were addressed in session five.

The subject of session six is of particular importance in Android: *Intents* and *Remote Procedure Calls*. These effective inter-process communication mechanisms increase the modularity and independence of the processes run by the applications, and facilitates the reuse and replacement of components.

In addition to its basic utility as phoning device, smartphones are also used as mobile devices to access the Internet. The seventh session was dedicated to the study of external communications by means of the WebKit browser, HTTP connections, web services, and Bluetooth. Some fundamental aspects of the graphic libraries in Android were introduced in the eighth session. In session nine, the students learned how to handle phone calls and messages while running other applications, and how to provide automatic responses to SMS. As a natural conclusion to the course, the last session was devoted to how to publish an application on the (formerly called) *Android Market*, renamed nowadays as *Google Play App Store* (Google Play App Store, 2013). Often this session was also used to have an invited talk from someone working in a mobile communications' company. That provided the students with real information, impressions and feedback from the outside world and allowed them to establish connections with the industry.

It is worth noting that this course was thought as a general post-intermediate course in programming and it was not planned for a wide audience. It is interesting to compare the contents of this course with other proposals. Consider for instance the topics covered in (Muppala, 2011). Both courses cover roughly the same topics, however in (Muppala, 2011) much more emphasis is done in the real-time aspects.

3.2 Material

This course has used a significant amount of educational and support material, including electronic devices, paper books and online material.

Google Phones. The *AndroidEDU Program* from Google awarded our course proposal with 10 phones Android Dev 1, and 10 mobile Phone Nexus One (HTC G1, 2013; Nexus One, 2013) (see Figure 1). The Android Dev Phone 1 was a variant of the HTC T-Mobile G1 designed exclusively as a tool for application developers to Google's Android OS. Both phones run on Android (1.0 Phone in Dev 1 and 2.2

(a) Android Dev Phone 1 (b) Nexus One

Figure 1: Smartphones provided by Google in 2009.

in Nexus One), have Wi-Fi (802.11 b/g), Bluetooth, quadband GSM radio and USB. The storage ability is considerably higher in the Nexus One (512MB RAM, 4 GB microSD card) than in the Dev Phone 1 (192MB RAM, 1 GB microSD card) as well as its processing power (1 GHz CPU vs. 528 MHz CPU).

Although the Android software development kit (SDK) includes an acceptably good emulator, the students found always very attractive and interesting the possibility of having real smartphones in their hands to test their applications. At the first editions of the course, the provided smartphones were of the latest technology (and therefore quite expensive) and they were not yet available to the general public. By that time, not many smartphones were available in the market. That was a major plus for the course, since the students were extremely interested in such a novel hardware and showed themselves eager for knowing how to program it using Android which, was also something to discover by that time. However, as the time passed, smartphones became more powerful and popular. That happened relatively fast. In a short time, namely three to four years, our smartphones became old and the students themselves owned better ones. Also Android established itself as an important and well-known system and everyone had heard about it at the later editions of the course. Thus, both the hardware and the software involved in the course, were gradually losing their attraction to the students.

Bibliography. At the first edition of the workshop, there were very few books on Android. The following ones were used:

- M. Murphy. *Beginning Android*. Apress Ed. 2009
- Z. Mednieks R. Rogers, J. Lombardo and G. Blake Meike. *Android Application Development*. O'Reilly Media Ed. 2009

Both are very practical, provide a good knowledge of the Java programming language, and are very focused on the implementation. They include many discussions about the code of various applications, and show how the efficiency of the programs can be improved by gradually adapting them to consider the specific restrictions imposed by limited memory and battery life in mobile devices. An important feature of these two books is that the publishers offer on the Internet the source code for all programs referred in them, so it is easy to access the code, to test it, modify it and reuse it. In our opinion, they remain still two of the best books to follow for teaching Android. New editions of them exist (Murphy and Allen, 2011; Rogers et al., 2010). Since 2009, there was a burst of edited books and e-learning support material about Android on the Internet (guides, videos, etc..). Among the recently appeared references, it is worth mentioning (Haseman, 2008; Ableson et al., 2009; Mednieks et al., 2012; Burnette, 2009; Ledford, 2009; Miller, 2008) because of the quality of their contents.

Another essential online resource for this course is the Google's *Android Developer's guide* (Android Developers' Guide, 2013). In 2009, this guide was very scarce in contents and very chaotic, which made it often difficult to find specific information. Nowadays, the guide is very complete and describes basically all what can be done with Android in a very well organized way. The examples documented therein are also very appropriate.

Website and Online Forum All the contents and materials of the course, as well as projects from previous editions of the course can be found at `http://albcom.lsi.upc.edu/tpaagp/`. An online forum for students and teachers can also be found.

3.3 Evaluation

The evaluation of the students was divided into two parts: a part of continuous assessment and the evaluation of a final project. Although these two parts were always present, the way they were brought into practice had to be adapted because the profile of the students, the technology and the relation between them both, changed in the meanwhile.

Apart from introducing our students to Android, this course was aimed for enhancing them to other important abilities. For example, to work in groups, to develop a big project in a continuous and progressive way, to discuss troubles and their solutions, to brainstorm, to defend their work in English to a wider audience, to promote their work in social media platforms, etc. For working up these goals into practice, students were always asked to work in 2-3 people teams.

Continuous Assessment. At the end of each session, a programming exercise was left as homework to be solved for the next session. In order to provide the students with a general overview of the different parts conforming a non-trivial Android application, the exercise left as homework one week was, as far as possible, an extension of the one from the previous week. At every session, one team presented how did they solved the homework, and their solution was discussed among all the other students and teachers.

For evaluating this part, it was weekly taken into account how the homework was done, how did the students publicaly present their solutions to their colleagues, and also their ability to discuss and defend their solutions to the problems found.

Final Projects. The students were also asked to develop a final project. The final projects were of greater difficulty than the exercises performed along the course, and each team is assigned with a different one. Usually the final projects were proposed by the lecturers, although we exceptionally agreed sometimes on projects proposed by students. The teams had approximately two months to work on them. During that time no more pressential sessions of the workshop were scheduled.

For the evaluation of the final project, the students had to provide the programming code of the developed application, and a technical report about it. They also had to make a public oral presentation of their project in which they explained to their colleagues how they designed and programmed their application, which problems they found, how they solved them, etc. In that presentation, they also had to show that their application was properly installed and successfully running in their phones. In the first editions of the course, the students were also asked to film a video in which they promoted their application to a broader audience. Those videos are available in an dedicated channel on YouTube. All the material to be evaluated had to be in English.

We briefly explain below some of the projects undertaken to date. More information about them and all the other projects can be found at the website of the course.

Our city, Barcelona, is a metropolis with more than 1.5 million inhabitants, within a urban area of 4.5 million people. Moreover, it gets around 7 million tourists every year. That was always an inspiring scenario for us when thinking about projects to propose to the students. The proposed projects were aimed to help, by means of the mobile technology, manage some important city services as, for example, the bike rental service, car parking, car sharing, taxi finder, etc., and supporting and promoting the tourism. It is worth to highlight that most of the Apps resulting from these projects were a big novelty by the time they were developed, e.g.:

- *gBicing.* The *Bicing* system for bike rentals has been a quite successful idea that has considerably increased the mobility within Barcelona in an ecological and sustainable way. This application allows to manage in real time the location and occupancy of the Bicing's bike stations.
- *Easy Parking.* This application helps the user to find it by means of the GPS coordinates of the car location and the use of Google Maps. When sharing the car with friends, the application gives the possibility to determine whether the car is available and an estimation of the time to retrieve it. Furthermore, this application allows to reserve the car for a period of time.
- *Guide and vote.* This application guides the user along his visit to a site and the several points-of-interest in it. Each point of interest is ranked according to the interest of its previous visitors. Once visited, the user can also vote himself.

Another type of projects were those implementing interactive games. Projects implementing some games were always welcomed by the students. They were always aimed to be played by a group of people and in a distributed way, which is quite of a novelty. E.g.:

- *Distributed Tetris.* Tetris is probably the most popular puzzle video game ever. This application allows to play Tetris in a distributed way by using radio-based communication. Not only one, but several players play the same game against the machine, and so they share the same board and have full information about the game.

4 LIFE-CYCLE

In this section, the three phases detected in the life-cycle of the PWAAGP course are described. They define the natural evolution of the course along six semesters, with a total amount of hundred student inscribed in it.

4.1 Phase 1: Rise

Covers the period from Spring 2010 to Fall 2010. Starts with the participation in the AndroidEDU EMEA program. It is offered as an *elective* subject opened to all undergraduate students. This period has its own sociological points.

- Lecturers, most of them teaching programming courses, are far from to be familiar with Android. They aim to socialize Android (open code) as an alternative to the Apple-iPhone perspective. There are very few books and the web pages are incomplete and change quickly. As information is difficult to obtain, lecturers introduce the material in formal lectures. The amount of work required to prepare a lecture is significant.
- Students don't have smart phones. They appreciate very much the opportunity to work with a real mobile phone (provided by the university). The class-group is quite heterogeneous containing computer science and telecommunications engineers. Students are highly motivated and become good friends. The relationship between computer scientists and telecommunications engineers is excellent. Some students believe that good skills in Android can help them find interesting jobs.
- Relationship between students and lecturers is very good. Students accept quite well the limited Android capabilities of the lecturers to teach some topics or to solve some specific problem. They are very collaborative and help to solve problems. The chat of the course is a fundamental communication channel.
- The evaluation is project oriented. The project is proposed at the end of the training process. As the project is ”something new” it involves an important amount of extra work.

4.2 Phase 2: Stability

Covers the period from Spring 2011 to Fall 2011. PWAAGP is also offered as an elective subject.

- As web information and books about Android become more popular, regular lectures become less and less important. Students continue to be highly motivated. Slowly they become much more independent of the formal lectures and autonomous learning becomes more and more important. Formal lectures begin to disappear because students find them redundant. The role of the lecturers shift to the organizational aspects of the course.

4.3 Phase 3: Decline

Corresponds to the period from Spring 2012 to Spring 2013. Many factors contribute to the decline of this course. First of all the smart phones become a commodity. Getting information about Android is much more easy. For instance, several student associations offer Android courses. On the other hand, as it was a “successful” course, the lecturers were asked to adapt it to the standard continuous evaluation of optional courses so as to integrate it in the near future in a master curricula. The course was lately offered as a elective course in a computer science master degree.

- Formal lectures disappear because most students find them redundant. The role of lecturers become mostly supervising. The course is reorganized as lab sessions. The session assignments are highly guided. Less free-room for creativity.
- Evaluation continues to be project oriented. The project is cut in small parts and the different parts have to be completed along the course. At the end of the course only an integration remains to be done. The number of good projects fall rather radically.

5 RESULTS

The grades of the students in each term have an important correlation with the three stages of the life-cycle of this course, as seen in Table 1.

Rise. One can see in the first two rows of Table 1 that most of the enrolled students have completed the final project with excellent results, very few of them did not work as expected and for unknown reasons some of them abandoned the course (the lecturers suspect that this was due to the significant workload required). Given these results and the feedback received from the students of these two editions of the course it can be said that they had a positive overall assessment of the course and in particular they:

- found the course contents new and interesting,
- liked to have teachers that guide the search for knowledge more than experts in the field,
- appreciated the research required by the subject,
- found the support material good and appropriate,
- appreciated the existence of weekly exercises,
- agreed with the method of assessment,
- enjoyed the final projects and found them interesting and of appropriate difficulty and,
- were convinced that the course provided useful knowledge for their future careers.

In general, the results indicate, in our opinion, that a high interest in the subject of study compensated non-experienced lecturers and/or non-sophisticated assessment mechanisms. For lecturers, the results observed exceeded by far their expectations. First, students were much more proactive and motivated than they usually are in other more conventional courses, which resulted in student projects of high quality, both in their contents and in their presentation. Second, there was an interesting change of role for the lecturers, who were acting as a learning *coaches* rather than as conventional lecturers. Third, the use of new technologies in this workshop brought new insights to classical computer science contents such as algorithmic schemes and data structures. In view of the results obtained, new technologies have shown to be a very useful motivating element for teaching that, in our opinion, is still unexploited. This experience forced us to take new positions and educational challenges. From the viewpoint of the university, this course give us the opportunity to develop university-industry relations.

Stability. The third and fourth rows of Table 1 (in sepia) correspond to the results of what was called the stability phase. In contrast with the results of the rise phase, almost all the students have excellent final projects–specially in the edition of Fall 2011, so that, it is impossible to grade any of them with honors (MH) since it was impossible to distinguish a best project among all of them. For the whole phase, only one student failed and six of them dropped-out. However, even if in Fall 2011 there is a peak in students performance, they were not as motivated as before.

For lecturers these symptoms suggested that the course workload was heavy and did not correspond to the number of credits students were granted. At this point, lecturers decided to change the evaluation method and the course entered in its decline phase.

Decline. Instead of grading a big final project at the end of the term, lecturers decided to grade, the continuous development of a single project (the same for all students) that was appropriately divided in modules to be delivered weekly. The notes of the students when this evaluation method was just established corresponds to Row 5 in Table 1 (the first in brown). There it can be seen clearly that the notes are divided into two groups: the students that performed the minimum effort just to pass the course and the ones that were really motivated and wanted to obtain the maximum grade. And this marked the end of the course since in Fall 2012 it was not offered and in Spring 2013 it became an optional master course, only 5 students enrolled and their interest was very limited, as shown in the last row of Table 1.

6 LESSONS LEARNED

What has been discussed so far is a six semester experience with a course on Android as an example of

Table 1: Statistics of the PWAAGP course. In each column it is reported the number of students that obtained the corresponding grade (going from sufficient (S) to honors (MH).

		Enrolled	NP	S	A	N	E	MH
Spring 09/10	2009-2	14	1	0	2		8	3
Fall 10/11	2010-1	20	6	0	1	1	12	0
Spring 10/11	2010-2	21	3	0	0	7	8	3
Fall 11/12	2011-1	21	3	1	0	0	17	0
Spring 11/12	2011-2	21	1	0	8	0	8	4
Fall 12/13	2012-1	0						
Spring 12/13	2012-2	5	0	3	0	2	0	0

teaching cutting-edge technologies. Although this experience was influenced by particular circumstances depending on the prevailing university context, timings, students, and work team, one can extract the following lessons:

Lesson 1. Technical Universities Should Play a Role in the Dissemination of Cutting-edge Technologies. Nowadays it is well-acknowledged that universities are expected to collaborate with leading firms in technology-transfer projects and public funds are usually available to this end. Cutting-edge technologies courses are a great opportunity to achieve this kind of collaboration and to promote the dissemination of new technologies. This collaboration is advantageous for universities because they can get free access to equipment, motivation for its students and lecturers, social visibility, and so on. Leading companies also benefit from cutting edge technology courses, since they get, among other things, feedback on products from researchers and good students, new and fresh ideas and proposals (from course prototypes), adept future engineers and public visibility.

Lesson 2. Universities Should Provide Flexible Structures to Include Short Life-cycle Courses. Our experience shows that the benefits of early adoption of technology are substantial but returns diminish rather quickly. Many universities (e.g., ours) have convoluted procedures for curriculum change—elaborate proposals, fixed and infrequent submission deadlines, several committees. It will be difficult to bring cutting-edge technology in the classroom unless some sort of ad-hoc technology-adoption policy that makes short life-cycle subjects easier to approve, implement and update is put in place.

Lesson 3. Lecturers Should Watch for Challenging Teaching Opportunities. From Lessons 1 and 2, it is a natural consequence that the right way to introduce cutting-edge technologies courses maintaining their dynamism is bottom-up. Therefore it corresponds to lecturers and leading technological firms to look for collaboration opportunities with the goal of disseminating promising technologies.

Lesson 4. Cutting-edge Technology Courses Are Extremely Dynamic. In Spring 2010, Android was emerging as a new operating system and at that time it could be considered as a cutting-edge technology but after three short years it was no longer the case. Now, Android is, a standard operating system (in fact, the most used) for smartphones, tablets, smart TVs . This standardization was accompanied by several factors: (a) The phones received quickly became obsolete and, soon, students began to own their own phones; (b) courses on Android mushroomed in internet, companies and, academies; (c) Abundant information became available in several formats (books, web-pages, web-sites, etc.); (d) Several companies developing mobile applications emerged; and (e), Being proficient in Android passed from being a plus to being a requirement.

Lesson 5. A Life-cycle Appears to Exist. As shown in this paper, in only six editions the course completed its life-cycle. One can see that this life-cycle matched the evolution of Android from cutting-edge to a market standard. In fact, this case suggests that the life-cycle of college courses that introduce a cutting-edge technology is linked to the progression of the technology towards its market take-up. The claim is that cutting-edge technology courses should form part of college curricula, but always being aware that their life-cycle is short and dictated by the standardization of the technology.

Lesson 6. Motivation Changes. During the *rise* phase of this course, Google was keen on the dissemination of Android among college students, the students were highly motivated and well-disposed to learn and committed substantial amount of time to this purpose. Lecturers were motivated also by, both, the challenges of learning and teaching Android and the commitment of students. This situation was kept for a couple of semesters until students ceased to see this knowledge as an advantage (for instance for getting a job). and became less inclined to follow the guide of teachers who couldn't keep up with the pace of that technology. From the lecturers point of view, updating course contents from one semester to the next, involved a substantial amount of work that had a modest lasting value, minor recognition from the university and, at the end, no recognition from students.

7 CONCLUSIONS AND OPEN QUESTIONS

An experience teaching a course on *Android applications for Google Phone*s is presented. The course originated from the *AndroidEDU* program at 2009 providing an interesting interplay between the university and private IT firms as Google.

This course exhibited three stages: rise, stability and fall. The experience spanned six semesters, coinciding with the introduction, expansion and popularization of Android as a cutting-edge technology. The lessons learned along this period are summarized in a previous section.

The last stage of this course (accompanied with a loss of students' interest) has been attributed to a life-cycle that coincides with the standardization of cutting edge technologies, in this case of the Android system. Although this is a possible conclusion supported by the development of the course, other alternatives, such as lecturers' choices and actions taken to change and adapt the course to the new environment, may also explain the phenomenon. In such a case several open questions arise: how should lecturers and course designers adapt to the transformation of a cutting edge technology into a mainstream one? How should universities accompany this process? What role should IT firms play?

Although not conclusive, student assessments support the main features of the life-cycle. In the future, more accurate student assessments should be designed and applied for the duration of the course in order to be conclusive.

It is also claimed, that the life-cycle described in this work is common to every cutting edge technology course, but, a comparison with other cutting edge technologies courses is still to be done. This effort may be supported by knowledge on diffusion of innovations (Rogers, 2003).

One of the assets of the experience here described is the integrative nature of the contents of this course (also noted by (Muppala, 2011)) it remains to be explored how similar integrative benefits can be brought into main stream courses and with types of content.

REFERENCES

Ableson, F., et al. (2009). *Unlocking Android: A Developer's Guide*. Manning Publications.

Android Developers' Guide (2013). developer.android.com/guide.

AndroidEDU EMEA (2013). sites.google.com/site/androideduemea.

Bryan (2013). The rise and fall and rise of MOOCs. blogs.cofc.edu/tlt/2013/01/28/the-rise-and-fall-and-rise-of-moocs.

Burnette, E. (2009). *Hello, Android. Introducing Google's Mobile Development Platform*. The Pragmatic Bookshelf.

Frank, R. and Waren, A. (1982). Teaching new technologies. In *13th SIGCSE symposium on Computer science education*, page 268.

Google Play Store (2013). play.google.com/store.

Haseman, C. (2008). *Android Essentials*. Apress Ed.

HTC G1 (2013). www.htc.com/www/product/g1.

Ledford, J. (2009). *Web Geek's Guide to the Android-Enabled Phone*. QUE Ed.

Mednieks, Z., et al. (2012). *Programming Android: Java Programming for the New Generation of Mobile Devices*. O'Reilly Media.

Miller, M. (2008). *Google-pedia: The ultimate Google resource*. QUE Ed.

Muppala, J. K. (2011). Teaching embedded software concepts using Android. In *Proceedings of the 6th Workshop on Embedded Systems Education*, pages 32-37.

Murphy, M. and Allen, G. (2011). *Beginning Android 4*. Apress Ed.

Nelson, R. R. and Wright, G. (1992). The rise and fall of american technological leadership: The postwar era in historical perspective. *Journal of Economic Literature*, 30(4):1931–1964.

Nexus One (2013). www.google.com/phone/detail/nexus-one.

Norman, D. A. (1988). *The Psychology of Everyday Things*. Basic Books.

Norman, D. A. (1998). *The Invisible Computer: Why Good Products Can Fail, the Personal Computer Is So Complex, and Information Appliances Are the Solution*. MIT Press.

Postman, N. (1984). *Amusing Ourselves to Death. Public Discourse in the Age of Show Busines*. Penguin.

Rogers, E. M. (2003). *Diffusion of Innovations, 5th Edition*. Free Press.

Rogers, R., et al. (2010). *Android Application Development*. O'Reilly Media.