ASSESSING PBL WITH SOFTWARE FACTORY AND AGILE PROCESSES: A CASE STUDY TO DEVELOP MOBILE SOFTWARE´S ENGINEERS

Simone C. dos Santos, Andrea Pinto
CIn – Informatics Center, Federal University of Pernambuco
Prof. Luis Freire Avenue, Recife, Brazil
C.E.S.A.R - Recife Center of Advanced Studies and Systems
220, Bione Street, Recife, Brazil
scs@cin.ufpe.br, andrea.pinto@cesar.org.br

ABSTRACT
The increasing and continuous presence of software products and services consumed daily by the society is gaining momentum with the development of the mobile device market. In this scenario, the demand for trained professionals in specific technologies for mobile platforms like Google Android and Samsung BADA, is growing rapidly. The Problem Based Learning method was used to solve practical and real problems inside a Program for teaching Software Engineering techniques to develop Mobile Devices; this program was inspired in the Medical Residency concept, and is called “Software Residency”, in which students learn by doing mobile applications in a real working environment implemented by a software factory. To execute this approach, a teaching and learning method was defined and applied based on authentic environment and authentic assessment, approach that brought relevant results towards the learning effectiveness in this context.

KEY WORDS
Advanced Technology in Education and Training, PBL, Software Factory, Agile Processes.

1. Introduction
The increasing and continuous presence of software products and services consumed daily by the society is gaining momentum with the popularization and development of the market for mobile devices connected in wireless networks. In this scenario, the software industry has been demanding professionals trained in specific technologies for mobile platforms such as Google Android (www.android.com) and Samsung BADA (www.bada.com), and skills such as business vision, management skills, and the interpersonal characteristics essential to the development of solutions that satisfy a consumer market increasingly demanding and agile. To train professionals in this market, it is necessary an effective education model, which allows the development of technical and non-technical skills, as also, the execution of practices focused on real projects with complexities similar to those found in the labor market.

In this context, PBL fits. Frequently defined as a learning instructional method where the learners are immersed in a practical environment, developing relationships between apprentices and improving themselves initiatives [1], PBL represents an excellent alternative to form professionals aligned to the software industry needs, proven by success of several initiatives as described in [2], [3], and [4].

But the implementation of PBL in the software engineering education requires a major care such happens in the medicine field, where this methodology was created. The learning environment needs to reflect the authenticity found on the market, thus ensuring the effectiveness of the method. In practice, a lot of distinctions can be found between the reality of the software engineering industry and the teaching and learning environment based on PBL. The practitioners from the software industry frequently develop their activities in projects executed considering well defined processes, lead by experienced professionals, often supported by technical leaders and a project manager that tracks the client satisfaction criteria as product and process quality, assuring on-time delivery and low costs. These characteristics are quite different of the classroom or the computer lab.

PBL was implemented bringing this reality to its learning environment using the Software Internship concept (“Software Residency”) [5], similar to the medical internship concept, in which the student learns to solve real problems in real situations. According to [2], the authenticity of this knowledge environment could be preserved considering the following elements: a real and practical problem adoption; the definition of the human resources involved with its roles and responsibilities; theoretical content aligned with the problem solving; and adoption of the development and assessment processes used by the industry.

Particularly related to the process perspective, the software industry becomes more demanding of the total project control, thus the project success involves beyond the functional and technical issues, also the management issues like on-time requirements, roles and responsibilities of human capital, good relationship between team members and management of client expectations. Considering also the rapid pace of changes and innovation
in the software engineering area and business environment, the worldwide IT industry has shown a tendency for agile project development, through the adoption of methodologies like Scrum (www.scrum.org) and Kanban (www.channelkanban.com). The results of this adoption reveal that agile methodologies truly can improve the projects results, considering the aspects of quality, client satisfaction, and productivity, without increasing, significantly, the projects costs. So, the use of these methodologies in the learning environment offers the needed authenticity regarding the actual industry environment of mobile software development.

Focusing in the authenticity of PBL methodology and the IT support on the teaching and learning processes, this paper describes an education program to form mobile software engineers. The teaching and learning environment were implemented inside the Software Residency model, controlled by agile development software processes, real applications conception and development; and multidirectional relationship between students and specialized practitioners acting as teachers and tutors.

This education program was executed by the partnership between the C.E.S.A.R Institute (Recife Center of Advanced Studies and Systems) and SIDI (Samsung Institute for Development in Informatics).

2. Planning the Software Residency

The “Software Residency” program [5], based in the Medical Internship model, identifies two core elements: (1) the formal education in relevant knowledge through an education entity and; (2) the deepening in practices inside a specific knowledge area, acquired in real software development environment. This definition is clearly aligned to the PBL education method, highlighting the need of a formal education, practical environment and actual solving problems inside this environment. By means of simplification, the term “BADA’s Residency” will be used to make reference to this program throughout this paper.

The educational objective of the BADA’s Residency was to instruct software engineers in the development of mobile applications using the Samsung BADA’s platform.

The Software Factory model was used as a practical knowledge environment to run the BADA’s Residency, involving the SIDI Institute as the software factory client, and the C.E.S.A.R Institute as the education institution. It is important to emphasize that the Software Factory concept is related with software development based on well structured and integrated units, with clear roles and responsibilities, supported by tools and mature processes, as described in [6].

The BADA’s Residency was run in a 10 months period, starting in August, 2010 and finishing on May, 2011: the first two months were dedicated to program planning, including recruiting and preparation of the support team; while the Software Factory was run in eight months. This Residency was formed by 26 students, grouped in three teams of 8 to 9 members, each one. The students had a technical profile and were undergraduate students in information technology area with 50% of graduation concluded, at minimum, but without any experience in the Samsung BADA platform.

Based on the PBL methodology implementation in the Software Engineering education described in [2], four core program elements were mapped: Actual Problem, Human Capital Roles, Content of Disciplines and Assessment Process.

2.1 The Actual Problem

The program goal was to develop two mobile applications per student, one low-level complexity application and the other one of high-level complexity. Applications with low level of complexity lasted approximately 2 to 3 weeks of development, considering that the graphical interface and required texts were ready, while the high level application lasted about 4 to 6 weeks.

The ideas for the applications were taken from a repository with preliminary ones identified by specialists, designers, the client, tutors and the students themselves. The students identified ideas during the practical classes related to conception and identification of products. The ideas were firstly selected by the team members and then sent to the client for validation.

All developed applications were made available worldwide through Samsung’s online store, in three idioms, Portuguese, Spanish and English.

2.2 Human Capital

The Human Capital involved in the BADA’s Residency was organized in four groups: (1) Management Team, composed by the academic coordinator, project manager and a client representative; (2) Educational Team, formed by teachers, technical tutors and a PBL tutor; (3) Technical Team, formed by the internship students; (4) Support Team, with practitioners from user experience area, designers, test engineers, content writers and translators. Figure 1 shows Human Capital organization.

![Figure 1: Human Capital in the BADA’s Residency.](image-url)
the support team formed by one User Experience (UX), one graphical interface Designer and one Test Engineer. The translation was done by a team of three writers that supported all interns.

A group formed by teachers was responsible to teach the disciplines specifically planned to this program (more details in the section “Content Disciplines”). Both technical leader and PBL tutor had the goal of disseminate the learning among the teams and preserve the PBL methodology.

Finally, a group formed by an academic coordinator (supporting the educational program and assessment processes), a project manager (responsible by the project management) and a client representative (responsible to conception and validation of applications), completed the team.

2.3 Content Disciplines

The content of the BADA’s Residency was created considering the students profiles and the results expected by the client.

The recruiting of the students was done with the aid of an announcement, describing the program goals, the basic requirements that would be used in recruitment and the student role description.

The selection process consisted of technical knowledge tests, curriculum analysis and personal interviews. Therefore, early in the process it was already possible to identify the profile of the students. Based on that, it was identified the need to instruct them in the C++ programming language, which is fundamental to understand the BADA platform, assuring that all the students had the same level of understanding regarding to this language.

Based on these information, the disciplines were divided in three technical modules: Conception of BADA Applications, which covered techniques and concepts used for product conception by the design perspective; BADA Fundamentals, including logic of algorithms and C++ programming language for BADA; and API (Application Programming Interface) BADA, covering the basic and advanced components of the BADA platform.

Additionally, it was included two more modules; a PBL training in the beginning of the program, assuring that all program members knew the PBL principles; and a discipline related to agile software development processes, focusing in Kanban and Scrum methods.

2.4 Learning and Assessment Processes

An effective PBL methodology is strongly process-oriented, since the approach needs to be planned to ensure that theory and practice will walk together and aligned. Moreover, learning needs to be accompanied by instruments that can assess its effectiveness [7].

As detailed in [8], the BADA’s Residency adopted an authentic assessment process [9] based on three perspectives: (1) Content, with the conceptual, practical and contextual assessments done inside the program disciplines; (2) Process, according to reviews made during project meetings, as required by Scrum and Kanban and other project status meetings; (3) Output, based on the analysis of the BADA applications delivered by the students.

In the Content perspective, the assessment was done by the teacher, based on the discipline and practical activities, the interactions with the students and; tracking the applications development.

In the Process perspective, the assessment was done by both the PBL tutor and project manager along the applications development. This tracking was done during the project status assessment meetings (Status Report), as also through meetings recommended by Kanban and Scrum methodologies, frequently made daily and weekly, using as reference the software process development and the criteria defined by the PBL tutor. As criteria, it’s important stand out: general view of activities and project phases, schedule tracking, on-time delivering goals, risk management, and the identification of the process weakness and the improvement opportunities.

Lastly, in the Output perspective, the assessment happened by analyzing the source code of the BADA applications developed by the students, considering its integration with the components and services generated by the support team, like graphical interface, and test execution artifacts as well. These analyses were made throughout the applications’ development considering the following criteria: suitability of the BADA architecture pattern; structure of the source code; source code clarity; interface and system integration, and; validation and verification of the applications. The assessment process was conducted by both the lead tutor and the technical tutor of each team.

3. Monitoring and Executing the BADA’s Residency

The BADA’s Residency was implemented in three phases: the first one in 3 months, with focus on high complexity applications; the second one in 2 months, with focus on medium complexity applications; the last one in three months, with focus on high complexity applications. In the first and second phases, the applications were developed by teams, whereas in the third one, each application was developed by a student individually.

During these phases, the continuous assessment process was planned for the Residence BADA, allowing improvements to the program during its execution.

3.1 Content Aspect

The vast majority of content planned for the training program was introduced in the first phase: C++ programming language, Conception of BADA Applications, BADA Fundamentals, BADA API, BADA
basic components. Bada advanced was introduced in the second program phase. Concepts and fundamentals about Processes and Methods (Scrum, Kanban and PBL) were also introduced in the first phase.

The classes were conducted part-time, while their practices were carried out within the context of applications in the respective development phase. The alignment between the disciplines content and the applications context brought motivation to the students to know and to investigate the BADA platform, enhancing content in the subjects difficult to understand and reducing the content of the simplest subjects. A practice adopted by the students on their own initiative was to read the technical documents of BADA platform even before the discipline taking to the classroom the questions and difficulties found to apply the concepts. Good results were confirmed by the students’ performance during the classes, these results were raised during the assessment process in the Content aspect.

Figure 2 shows the performance of the students in the disciplines per development team.

![Figure 2: Assessment considering the content aspect.](image)

Considering the scale from 0 to 10, with satisfactory performance above the average 7.0 (seven), the graph of Figure 2 shows that the performance of all groups was satisfactory, with above average grades and an evolution almost always continuous. Points of difficulty can be observed in the C++ and Media disciplines, due to the approach of teaching and the complexity of content, respectively. Comparing the teams’ performance, we can also observe various points of convergence between the teams, highlighting the team A with the best general performance.

### 3.2 Process Aspect

In the first phase of the program, an empirical process to develop BADA application was used. Based on this process, and using the PDCA (P – Plan, D - Do, C – Check and A – Act) cycle as a reference model, four phases were mapped, as illustrated in Figure 3.

In the planning phase (PLAN), the UX team drew a navigational model for the applications that would be developed, discussing them with the software engineers (the students), the tutors and current teachers. In general, a big effort was spent in this phase, since it was required a deep understanding of all application elements, including, in several times, the elaboration of specific content that were validated by the client. One example of an application in this context is the “Quick Cooking”, that shows some food recipe to the application users, for which was needed an authorization by the recipe’s authors so the application could be published referencing the source of the receipts.

In the planning phase (PLAN), the UX team drew a navigational model for the applications that would be developed, discussing them with the software engineers (the students), the tutors and current teachers. In general, a big effort was spent in this phase, since it was required a deep understanding of all application elements, including, in several times, the elaboration of specific content that were validated by the client. One example of an application in this context is the “Quick Cooking”, that shows some food recipe to the application users, for which was needed an authorization by the recipe’s authors so the application could be published referencing the source of the receipts.

In the planning phase (PLAN), the UX team drew a navigational model for the applications that would be developed, discussing them with the software engineers (the students), the tutors and current teachers. In general, a big effort was spent in this phase, since it was required a deep understanding of all application elements, including, in several times, the elaboration of specific content that were validated by the client. One example of an application in this context is the “Quick Cooking”, that shows some food recipe to the application users, for which was needed an authorization by the recipe’s authors so the application could be published referencing the source of the receipts.

In the planning phase (PLAN), the UX team drew a navigational model for the applications that would be developed, discussing them with the software engineers (the students), the tutors and current teachers. In general, a big effort was spent in this phase, since it was required a deep understanding of all application elements, including, in several times, the elaboration of specific content that were validated by the client. One example of an application in this context is the “Quick Cooking”, that shows some food recipe to the application users, for which was needed an authorization by the recipe’s authors so the application could be published referencing the source of the receipts.
frequently related to the understanding and accuracy of the application requirements.

Second, as the applications of the first phase were simpler, with short term, it became necessary that the project manager could track the development of several applications in parallel, solving many conflicts and impediments. This issue highlighted the need of finding a process that could offer great visibility and transparency to the activities.

Third, although the initial applications were selected considering their low complexity (since the intern software engineers were learning to develop in the BADA platform), there were activities involving mainly the graphical interface design, the content creation, translation and test execution, all of them done by the support teams and demanded by almost all applications at the same time. Additionally, the continuous validations of the client along the development process, required changes in graphical or content components, overloading these support teams. In this context, the management of the resources sharing became very complex.

Faced by these initial problems, the program adopted the Kanban approach, a Japanese technique that follows the Lean principles, emerged from TPS (Toyota Product System).

### A. Adopting Kanban Method

Kanban is a project management methodology that introduces improvements in the software development cycle. It is a precursor approach in the use of a visual controlling mechanism where we can track the activities throughout the phases of the value chain. Kanban is based in a simple idea; the number of work in progress (WIP) should be limited. New work should only be initiated when another one is completed or when an automatic function do the pull action [10].

A visual board with post-its is typically used with the purpose of visually control the activities. In the BADA’s Residency, the board was divided in three swim lanes, each one representing a team. Big post-its were used to represent the tasks, and small post-its in the color of blue, green and yellow were used to identify team members’ profile: usability engineers and designers represented by the pink ones, software engineers represented by the yellow and test engineers represented by the blue ones. The Figure 4 shows the Kanban’s visual board of the BADA’s Residency.

The columns of the Kanban’s visual board were defined in the following way: (1) ideas approved by the client, representing the development backlog; (2) benchmarking, corresponding to the search of similar applications on competing websites; (3) prototype and navigational model, when the initial application graphical interface are generated; (4) development, phase in which the applications are developed; (5) applications submitted to Samsung’s online store, the final phase of the process.

The WIP was defined and limited for all workflow phases, and the lead time (time between the moment that one idea was selected to enter in the workflow, and the moment when it was submitted to the online store) was measured. In this way, the process allowed better requirements validation (represented by columns 1, 2 and 3) and an improved visibility of the major activities flow.

![Figure 4: Kanban’s visual board in BADA’s Residency.](image)

With the use of limited WIP in the Kanban system, any activity blocked for any reason stopped the entire system. This major characteristic of Kanban caused cooperation improvements inside the team, and also a better use of the human resources of the support team and, hence, an improvement in the process productivity.

The analysis of the Kanban adoption in the first phase allowed quick changes to reduce the recurrence of problems in the second phase. However, a new challenge was identified. In the process flow mapped using Kanban there was a small visibility related to the development phase of the applications in the Development column. Considering that in the third phase, each student individually had to develop an application, it was necessary to use an approach with more refined control. Therefore, it was chosen to use the Scrum methodology, a more prescriptive (“with more rules to follow”) methodology, in addition to Kanban.

### B. Using Scrum in the last phase

Scrum is an empirical approach focused in persons, developed for environments where requirements arise and change quickly. The implementation of Scrum is based on task definition and prioritizations (Product Backlog), made with the client support, and grouped in development units with short duration (maximum of 4 weeks), called Sprints. The process is governed by an actor called Scrum Master and aggregates monitoring and feedback activities. These activities are conducted generally in daily meetings of short duration, involving the entire team to pinpoint and correct any shortcomings and/or impediments on the development process [11].

The Scrum method also uses a visual board, usually divided in three columns: Backlog, Doing and Done. Each column represents the actual status of the activity. The tasks execution beginning is marked by its removal from the column “Backlog” to the “Doing”, and the conclusion is evidenced by the move to the “Done” column.

One Scrum’s visual board was created for every application of this phase (Figure 5). Big post-its
represented the different tasks and the small ones highlighted the human resources’ profile, just like in the Kanban’s visual board: usability/designers/content engineers (blue), software engineers (yellow) and test engineers (green). Pink post-its were used for impediments.

![Figure 5. Scrum board in BADA’s Residency.](image)

The use of Scrum at the BADA’s Residency provided some important contributions. The presence of a product owner, responsible for the application conception and validation, allowed clearer applications' requirements. Other positive point was the use of agile requirements approach, based on “user’s stories”, easing the communication between de owner of the product and the technical team. The Scrum Master role was shared between the technical tutors and the project manager, which conducted the prioritization of activities, the daily meetings to follow the applications development and the assessment of impediments. The Scrum’s visual board, composed with the “Backlog”, “Doing” and “Done” tasks, provided greater visibility and transparency to the tasks flow in each application, and the identifications of general issues shared by many of them. Thus, using both methodologies, Kanban and Scrum, the project reached the necessary agility and the requirements control.

C. The team performance

The assessment process in the Process perspective was performed along the applications development, about both processes, Kanban and Scrum. As an instrument of evaluation were used forms with questions about the objective and subjective understanding of the two methodologies. The ability to follow procedures was assessed during the meetings scheduled in the methodologies, such as the Scrum meeting and status report presentation, exploring the keypoints presented at section 2.4. Figure 6 shows the performance graph in Process perspective for the three teams, related to the 14 applications in the second and third phase.

![Figure 6. The process performance per team.](image)

Looking at Figure 6, it is observed that the performance related with this criteria was maintained with average grade of more than 8.25. This high performance is explained by the intensive use of agile development methodologies, focusing on the high relationship between those involved, and monitoring of tutors dedicated to ensuring the adoption and compliance of these processes.

3.3 Output Aspect

The results of BADA’s Residency in the Outputs perspective were very positive. Figure 7 shows some applications’ screens.

![Figure 7. Bada Applications in Samsung App Store.](image)

During its execution, 55 applications for mobile devices were generated in the three phases: three teams developed 16 in the first one; the same groups developed 9 applications, more complex than the previous ones, in the second phase, and 30 applications were developed in the third phase, exceeding the initial planning.

All of applications were internationalized and translated to three different languages: English, Portuguese and Spanish. Most of them received positive reviews by the users of the Samsung Apps Store, averaging 3.5 stars out of 5. Particularly, the application called “Children’s Timetable” remained in the website’s “Top List” (with 5 stars) during Mother’s Day celebrations, praised by reviews and high number of downloads (Figure 8).
The performance of students in the Output perspective throughout the Residence BADA was superior to 9.0 out of 10, demonstrating the high quality of the applications code. Figure 9 shows this performance of the same group of applications showed in the Figure 6.

4. Conclusion

The PBL implementation as an education methodology in the Software Engineering field has gained good results because of its practical and collaborative context. The use of an authentic environment of software development to teach and learn, with all the complexity and diversity attached to it, make it possible the use of PBL in this area reaching the same level of reality of the hospitals – which has been used by PBL applied to the medical education for many decades with a great success.

With this motivation, the case study described in this paper presented an effective PBL approach, created in a practical knowledge environment based in Software Factory and agile methodologies, inside a real context with real problems.

The efficiency of this approach was proven by the amazing results. Among these results we can highlight the performance of students above the industry media (in general, 7.0), and the development of 55 applications with quality standards approved by the client and the mobile devices consumers (the end users), from the analysis of the applications’ download. We can highlight also as important result, the hiring of 40% of the students by the involved institutes.

Finally, the motivation of students along to the program, observed from their performance and commitment was investigated at the end of the program, through satisfaction evaluation questionnaire composed by items about program quality, education methodology, team interrelationship and teachers’ quality. The results showed a level of satisfaction with 100% above the level of very good (4 out of 5).

As future works, we highlight the analysis of the results with the use of this same approach in two more Software’s Residency ongoing, in the Telecom industry. In these works, others market instruments have been used to assess the students under the process perspective, like the Nokia Test, a questionnaire commonly used to evaluate the Scrum adoption in software development teams. Additionally, other kinds of teaching and learning methods should be investigated, considering the better approach to fit the educational objectives of each residency and client’s purpose. These results will be part of one future analysis as well.

Acknowledgements

The results presented here were developed as part of joint project between CESAR and SIDI (Samsung Development Institute for Information Technology), funded by Samsung Electronics Amazonia Ltda., using resources of the "Law of Informatics" on Brazil’s electronics industry, No 8.248/91. Additionally, this program would not have obtained the results presented without the involvement and commitment of students and the whole technical and academic teams.

References


