Combining Challenge-Based Learning and Design Thinking to Teach Mobile App Development

Kiev Gama, Fernando Castor, Pedro Alessio, André Neves, Cristiano Araújo,
Rafael Formiga, Francisco Soares-Neto, and Higor Oliveira
Federal University of Pernambuco (UFPE)
Recife, Brazil

***Abstract*—This paper describes a set of practices, techniques and conceptual tools that were tailored by combining ideas from Challenge-based Learning, Design Thinking and Lean Startup. While the former focuses on intrinsically motivating the students to engage in the learning process by tackling real-world problems aligned with their sense of purpose, the latter emphasizes aspects of the system being built, such as feasibility, viability, and desirability. The framework conception is as an innovative approach for mobile application development in long-term courses (1- or 2-year) involving programming, design, and business. We applied the methodology in the context of a regular undergraduate course including students with a diverse computing and non-computing backgrounds and collected their feedback during the process.**

***Index Terms*—Challenge-based Learning; Design Thinking; Innovation and Creativity**

I. Introduction

Recently, many new educational approaches [1]–[5] have been proposed as means to better engage students in learning activities, improving their motivation to learn as well as their ability to build new knowledge. What all these new approaches share is an interest in catering for a new generation of students with a vast wealth of information available at the tips of their fingers. These students have a tendency to be independent learners, as long as they are learning something that interests them. These approaches emphasize learning by doing, instead of simply knowing [6].

Traditional higher education specialized on Science, Technology, Engineering and Mathematics (STEM) is focused on building a solid foundation that will result in specialists those domains, without much attention to creative, cross-disciplinary problem-solving skills [7]. As an attempt to fill that gap of college graduates lacking in creativity and innovation there is a trend to push Arts+Design into what is being called STEAM, offering students more than high-tech skills [8].

Going in that direction of bringing to students more innovative ways of thinking on problem solutions, by focusing on creative thinking and problem-solving skills when teaching technology, we propose a framework on top of Challenge-Based Learning (CBL) [9] to teach mobile app development in an interdisciplinary environment. The framework that we called CBL+DT consists of a set of practices, techniques and conceptual tools for students in mobile application development. We combine CBL with design thinking [11] and lean startup [23] principles. While CBL focuses on intrinsically motivating the students in the learning process, the other two approaches focus on aspects of the system being built, such as feasibility and desirability.

The framework was originally conceived as an innovative approach for intensive training (20h/week) on mobile application development in long-term interdisciplinary courses (1- or 2-year) involving programming, design, and business. Here we describe the CBL+DT framework and the student feedback we had after instantiating it in the context of a regular undergraduate semester course for mobile application development, which was a different and much more constraining setting than the original scenario where the framework was conceived.

The paper is structured as follows. Section II describes related work and background on the fundamental concepts used in this paper. Section III presents the CBL+DT framework. Section IV details the undergraduate course in which we applied CBL+DT and presents results about the students feedback. Finally, section V concludes the paper.

II. Background

A. Design Thinking

Design is an activity that is part of the conception and configuration of artifacts, using a human-centered approach. It can be divided into three spaces: (1) inspiration, the "problem or opportunity that motivates the search for a solution", (2) ideation, the "process of generating, developing, and testing ideas", and (3) implementation, "the path that leads from the project stage into peoples lives" [11]. Three important aspects of an insight or idea may lead to innovation: an idea should be **desirable** by the user, **viable** as a product in the market, and **feasible** in terms of technology and its possibilities.

One of the challenges that can be daunting to novices in design thinking is how to manage the vast number of ideas and research material produced by an interdisciplinary team and the ways those ideas and data could converge to a concrete solution. Ideation techniques such as brainstorms or SCAMPER [12], among many others, can be applied to help in that context. The generated ideas can be presented, filtered and chosen by the three aspects described previously (desirable, viable and feasible) and refined to the phase when they will become a final prototype and finished product.

B. Challenge-based Learning

The Challenge-based Learning (CBL) framework originally emerged in a high-school context as an approach to foster

978-1-5386-1174-6/18/$31.00 ©2018 IEEE

learning while solving real-world problems [9]. It later evolved to another version [5] targeting also higher education. CBL is based on experiential learning, which considers learning as a continuous process grounded in experience, taking into account the environment in which the learner is inserted [2]. The CBL approach has similarities to Problem-based Learning (PBL) [10]. While in PBL there is a predetermined problem within an action domain, in CBL students follow their passions to find and choose a challenge to be solved. An idea of global importance (e.g., sustainability, health) is linked to the problem in question, but with local applicability [1].

CBL divides the activity of learning into 3 phases: Engagement, Investigation and Action. **Engagement** is the first phase, when students start defining and exploring a *Big Idea* (e.g. health) which is a broad concept they want to explore. It is followed by an *Essential Question* (e.g. what do I need to do to be healthy?) that helps contextualize and personalize the Big Idea. The last step in this phase is a *Challenge* (e.g. engage sedentary people to exercise) that determines a call to action. **Investigate** is the phase where students and teachers take part in a research journey on topics related to the challenge. It starts with *Guiding Questions* (e.g. what are the typical obstacles for exercising?) to give a direction to the students' research path toward the knowledge needed to build a solution for their challenge. The *Guiding Activities* (e.g. Web search, field research, surveys, interviews) and *Resources* (e.g. online databases and journals, public libraries, experts in the field) help answering the Guiding Questions. The *Analysis* of the findings resulting from the Guiding Activities can be reports and presentations where the learners (students) present what they acquired in that process. **Act** is the phase where the solutions are actually implemented targeting a real audience. In the *Solution*, students conceive through prototypes and tests. The *Implementation* puts the solution into practice. The *Evaluation* gathers and evaluates data (e.g., interviews, questionnaires), allowing to reflect if expectations were fulfilled.

C. Related Work

The intersection of distinct domains enhances student engagement and learning, helping to unlock creative thinking and innovation [13]. Efforts to stimulate cross-curricular collaboration between STEM and Arts+Design – what is being called STEAM – are a way to foster the curiosity and self-motivation of college students [8]. In the case of software development, there are very recent attempts to bring design and creativity techniques to classes, such as Chaordic Learning [3], which employs a creative problem solving approach where students take charge of their learning with support from teachers, and École 42 [14], focused on peer-to-peer learning, where students just follow the projects proposed by the pedagogic team [15]. Both curricula address creativity and design but do not explicitly mention design thinking. We found approaches to teach mobile application development [4], [16] infusing design thinking methods (e.g., personas, storyboards, user journey maps, prototyping) into the project life-cycle, but it increased the already heavy load of material and assignments.

These previous examples show two scenarios for project-based courses: (1) the intention to use creativity techniques without design thinking being explicitly addressed and (2) too much detail about design thinking techniques in a short timespan, bringing a heavy load. We aim at an intermediary point where students can be guided to apply design techniques in class projects having little effort but enough information.

III. Proposed Framework

The initial setting in which the CBL+DT framework was conceived consisted of an interdisciplinary 2-year training program for mobile applications development targeting the iOS platform, for students from interdisciplinary backgrounds. In that program, we had the initial purpose of using CBL with undergraduate students to teach the construction of apps targeting real users. Technical content involved design, mobile application development and entrepreneurship.

We introduced adaptations to CBL, suggesting appropriate methods and techniques to enrich the motivational process based on classic ideation, design thinking and lean startup. Our proposed framework embeds techniques from those domains aiming to generate creative and useful ideas. In contrast to the original CBL approach, where the Investigate phase comes after the Engage phase, in our the adapted CBL+DT framework the former does not exist by itself. It is rather orthogonal to the Engage and Act phases. In our approach (Fig. 1), the CBL's investigate phase (guiding questions/activities/resources, and analysis) permeates the whole process, instead of being a preliminary phase as in the original CBL framework.

To minimize cognitive overload, we provide a small set of recommendations and templates with basic guidance. Students can follow the techniques autonomously in each CBL phase:

1) Investigate: This phase is pervasive throughout the entire process. It will be present in the transitioning of each activity or step (Big Idea, Essential Question, Challenge, Solution, Implementation, and Evaluation). The techniques (focus group, interviews, questionnaires, etc) used in the Investigate phase are supporting methods for requirements engineering [17], being useful to validate hypotheses about users and scenarios.

2) Engage: In this phase, the learners will get familiar with the domains in which they choose to explore. In those phases, they will define and investigate the domain, setting new guiding questions and answering them. The steps are Big Idea, Essential Question, and Challenge. In each step, we propose a set of techniques that help learners to choose and refine ideas.

Big Idea: One of the design techniques proposed for this step is Brainwriting [18], when learners can freely discuss their concerns, individual preferences and beliefs. A Big Idea can also be picked from a set of big ideas suggested by teachers. Scenarios, urban tribes, or public spaces can be identified as places to make observations.

Essential Question: Here the learners can perform a detailed review of the literature [19], to learn about the chosen domain. Students should summarize all the research about a large field of the Big Idea and propose a question that is only answered by more detailed research on the user and her

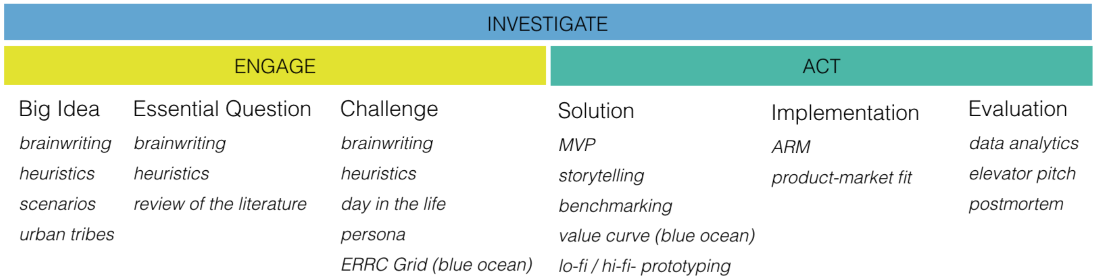

Fig. 1. In our proposed CBL+DT framework we embed different Design Thinking and Lean Startup techniques (in italics) throughout the process.

context. If the students already have information about the user, we propose to compile it all to build a Persona [20].

Challenge: The proposed design techniques for this step are Brainwriting, Day in the Life [21], the Eliminate-Reduce-Raise-Create (ERRC) Grid from the Blue Ocean Strategy [22], and Personas (if not yet provided by the students). The Grid is abstracted and broken down into four questions: *what can we [eliminate—reduce—raise—create] [in—from] people's live?*.

3) Act: The Act phase takes all the research and objectives to start constructing the mobile app that will help to solve the problem identified and described previously.

Solution: In this step, students agree on a Minimal Viable Product (MVP) [23], with a minimal set of features to address the core of their challenge. Next, they build a low-fidelity paper prototype [24] to rapidly present their ideas to the group and the teachers. After getting feedback on that, they start to build a high-fidelity prototype [25] and present it again to new users to validate details (e.g., navigation) and to have more feedback. The hi-fi prototype also helps the team identifying new tasks and planning the activities of the future phases.

Implementation: The implementation step is where the learners go for the coding steps and the conceiving steps of the product. The ARM (acquisition, retention, monetization) strategy [26] will help them find a revenue model if they want their app to become a product. They may iterate back to previous phases and "pivot" (i.e., modifying) their idea after prototype feedback. Validation of a product-market fit [23] helps to identify if a potential customer would pay for the app and if it could become a viable business.

Evaluation: In this step, data retrieved through direct feedback from users or through the mobile app store can help the analysis and evaluation of the app solution. The analysis may eventually lead to improvements in the app. Finally, the students are invited to present the finalized product as a pitch [27] and demonstrate it to an open audience. For evaluation and feedback, we also promote a postmortem meeting, taken from agile methods [28], where students can share with their team and teachers what went wrong and what went right.

IV. Instantiation of the Framework

As an attempt to experiment with the CBL+DT framework outside of its original environment, we instantiated it and adjusted it (i.e., bypassed certain activities) to be used in a regular undergraduate semester course. Classes mixed undergraduate students from two groups: (i) *computing students* (Computer Science, Computer Engineering, and Information Systems); and (ii) *computing novices* (Design, Architecture, and Graphical Expression). Students in the latter group have no prior experience with programming.

We present a preliminary analysis of the quantitative and qualitative data about the students' perceptions of the usage of CBL+DT. We collected data through a questionnaire sent by email after their final pitch, with 36 responses out of 39 students from the two classes (92% response rate).

A. Quantitative Data

The quantitative data from the questionnaire concerned three questions using a Likert-scale response format, ranging from Strongly Disagree to Strongly Agree. We analyzed the answers of the computing students and computing novices as two distinct groups, to compare their perspectives. The three questions respectively represent variables concerning the student's perceived confidence in developing iOS apps, the perceived motivation to learn given by the methodology and perceived self-motivation to learn, as detailed in Figures 2 and 3. We performed the non-parametric Mann-Whitney U test to check if the samples of the two groups have statistically significant differences. It allowed retaining the null hypothesis of the variables from the two samples having the same distribution, thus confirming the similarities. This similarity can be also visualized under the perspective shown in the graphs in Fig. 2 and Fig. 3. The major difference between the two groups concerns the confidence in developing an iOS application, where the group of computing novices is slightly less confident (only 9% strongly agree with that) than the computing students (36% strongly agree). In regards to the CBL+DT framework helping students to learn new things, results were very similar for both groups: the answers agreeing with the affirmation represent 76% and 82% of responses from the computing students and computing novices, respectively. Concerning the course motivating them to learn new things, the computing students seemed to be slightly less (76%) motivated than the computing novice (91% in agreement).

B. Qualitative Data

The qualitative data came from open-ended questions: how did the CBL+DT framework contribute to the project concep-

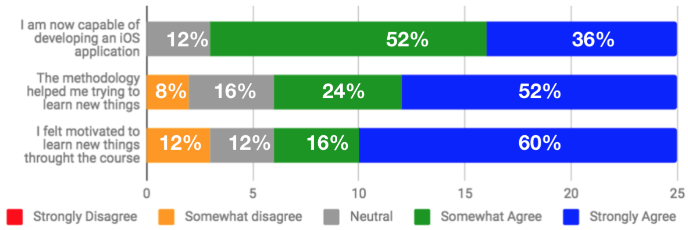

Fig. 2. Responses from computing students (*N=25*).

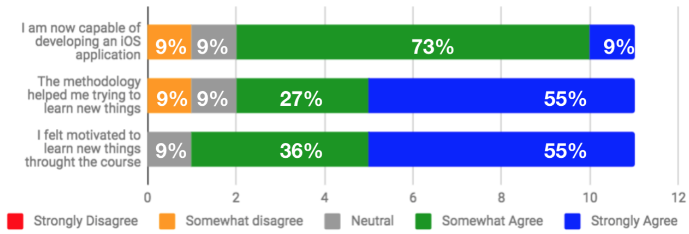

Fig. 3. Responses from computing novices (*N=11*).

tion; what positive aspects did they identify in the course; and what were the negative aspects they identified in the course.

Opinions about how the framework contributed to conceiving the project were mostly related to the idea of guiding or framing the activities toward the project idea. Students mentioned the framework as a "funnel" to a solution. A student that mentioned difficulties using the voting cards said it *"helped guide the steps of the group from the choice of the theme to the challenge"*. Students considered the ability to focus on a real problem and user needs as important:

"The process helped us to deepen our understanding of a problem and to discover problems that could be solved through some application, with a potential impact on people's lives, starting from a very comprehensive and difficult idea to attack." (Computing student)

We also asked students about the positive and negative points they observed in the course. Both computing and novice students were excited about the technology and the laboratory environment. The availability and support of the teaching team in-class and out-of-class hours were also considered positive. The CBL+DT framework was mentioned as a positive aspect by both groups of students as well. Students also highlighted the interdisciplinarity of the teaching team and also the exchange between students with different backgrounds:

"The flexibility of teachers and instructors always willing to make us understand, encouraging and motivating us to [autonomously] search for learning. [...] the methodology was very interesting because it aims at a multidisciplinary professional training that can know concepts from the area of design to programming and even entrepreneurship to understand the whole flow of an app." (Computing novice)

The majority of respondents mentioned the short timespan of the course (20 classes of 4h, 3x a week) as a negative point:

"Time seemed too short sometimes, many topics could get more classes, and learning object-oriented programming requires practice..." (Computing novice)

C. Discussion and Limitations

The fact of the computing novices being less confident about developing iOS apps is understandable since it is their first experience with programming. However, there is a consistency between novices and computing students concerning the utility of the methodology and the students' motivation throughout the course. The students' comments confirm that many of the proposed design thinking and Lean Startup techniques used in the CBL+DT framework could be used seamlessly. For instance, the combination of brainwriting with heuristics led to opinions quickly and easily converging. The opportunity to work with an interdisciplinary approach where in the same course they could use ethnography for user research, design techniques, business and innovation, and programming was appreciated by most computing novices and by some of the computing students. Many of the techniques related to design and innovation could be used after brief instructions and in certain cases with the help of templates that simplified them.

This research presents some threats to validity, such as the limited sample of students following the framework. The power and authority dynamic between instructors and students could have influenced their responses to be biased and affirmative to the instructor's approach. However, the anonymous forms and the negative opinions show that this dynamic may not have influenced the responses.

V. Conclusions

In this paper, we described the CBL+DT framework, which combines challenge-based learning, design thinking and Lean Startup techniques to motivate students to learn how to creatively think and build solutions targeting real user needs by engaging them with actual problems. We combine these techniques in a framework comprising instructions and a small set of templates based on a group of design techniques, trying to avoid the cognitive overload of learning such techniques in detail while also learning mobile application development.

We instantiated this framework in an undergraduate semester course of mobile programming, targeting computing students and computing novices (e.g., Design, Architecture). The method also proposes an interdisciplinary approach in the realm of technological innovation, where design not only focuses on the shape and aesthetics of interfaces but is in the essence of research for opportunities and how to solve problems in a creative way. Many design techniques could be seamlessly used throughout the process. Students felt motivated to learn new subjects by the process prescribed by our methodology and were able to produce functioning apps targeting real-world problems quickly. The framework helped in the convergence of group opinions, and fostered more research around the domain and potential stakeholders, giving more attention to user needs. Approaches going in this direction can materialize STEAM cross-curricular collaborations, resulting in curiosity and self-motivation of students being nurtured.

REFERENCES

[1] Laurence F Johnson, Rachel S Smith, J Troy Smythe, and Rachel K Varon. 2009. Challenge-Based Learning: An Approach for Our Time. New Media Consortium (2009).

[2] David A Kolb. 2014. Experiential learning: Experience as the source of learning and development. FT press.

[3] Stephan Krusche, Bernd Bruegge, Irina Camilleri, Kirill Krinkin, Andreas Seitz, and Cecil Wöbker. 2017. Chaordic learning: a case study. In Proceedings of the 39th International Conference on Software Engineering: Software Engineering and Education Track. IEEE Press, 8796.

[4] Natasha M Costa Valentim, Williamson Silva, and Tayana Conte. 2017. The students perspectives on applying design thinking for the design of mobile applications. In Proceedings of the 39th International Conference on Software Engineering: Software Engineering and Education Track. IEEE Press, 7786.

[5] Mark Nichols, Karen Cator, and Marco Torres. 2016. Challenge Based Learner User Guide. Redwood City, CA: Digital Promise. link-http://cbl. digitalpromise. org/2016/08/30/challenge-based-learning-guide (2016).

[6] Gerald C. Gannod, Janet E. Burge, and Michael T. Helmick. 2008. Using the inverted classroom to teach software engineering. In 30th International Conference on Software Engineering (ICSE 2008). Leipzig, Germany, 777786.

[7] Margaret E. Maddena , Marsha Baxtera , Heather Beauchampa , Kimberley Boucharda , Derek Habermasa , Mark Huffa , Brian Ladda , Jill Pearona , Gordon Plague (2013). Rethinking STEM education: An interdisciplinary STEAM curriculum. Procedia Computer Science, 20, 541-546.

[8] Michelle H. Land. "Full STEAM ahead: The benefits of integrating the arts into STEM." Procedia Computer Science 20 (2013): 547-552.

[9] Mark Nichols and Karen Cator. 2008. Challenge Based Learning, White Paper, Cupertino, CA: Apple. (2008).

[10] Diana F Wood. 2008. Problem based learning. BMJ: British Medical Journal 336, 7651 (2008), 971.

[11] Tim Brown and Jocelyn Wyatt. 2010. Design thinking for social innovation IDEO. Development Outreach 12, 1 (2010), 2931.

[12] Shanna R Daly, Seda Yilmaz, James L Christian, Colleen M Seifert, and Richard Gonzalez. 2012. Design heuristics in engineering concept generation. Journal of Engineering Education 101, 4 (2012), 601629.

[13] Erik W. Robelen. "STEAM: Experts make case for adding arts to STEM." Education week 31.13 (2011): 8.

[14] 42 Silicon Valley. https://www.42.us.org/

[15] Marta Peris-Ortiz, Juan José Alonso Llera, and Carlos Rueda-Armengot. "Entrepreneurship and Innovation in a Revolutionary Educational Model: cole, 42." Social Entrepreneurship in Non-Profit and Profit Sectors. Springer, Cham, 2017. 85-97.

[16] Maria Palacin-Silva, Jayden Khakurel, Ari Happonen, Timo Hynninen and Jari Porras. "Infusing Design Thinking Into a Software Engineering Capstone Course." Software Engineering Education and Training (CSEE&T), 2017 IEEE 30th Conference on. IEEE, 2017.

[17] Martin Maguire and Nigel Bevan. 2002. User requirements analysis. In Usability. Springer, 133148.

[18] Bernd Rohrbach. 1969. Kreativ nach Regen-Methode 635, eine neue Technik zum Losung von Problemen. Absatz-wirtschaft 12 (1969), 7375.

[19] John W Creswell. 2013. Research design: Qualitative, quantitative, and mixed methods approaches. Sage publications.

[20] Alan Cooper et al. 2004. The inmates are running the asylum:[Why high-tech products drive us crazy and how to restore the sanity]. Sams Indianapolis, IN, USA:.

[21] Matthew Riddle and Catherine Howell. 2008. You are here: Students map their own ICT landscapes. Hello! Where are you in the landscape of educational technology? Proceedings ASCILITE Melbourne 2008 (2008).

[22] W Chan Kim and Rene Mauborgne. 2005. Blue ocean strategy. Harvard Business School Press, Boston (2005), 240.

[23] Roland M Müller and Katja Thoring. 2012. Design thinking vs. lean startup: A comparison of two user-driven innovation strategies. Leading Through Design 151 (2012).

[24] Carolyn Snyder. 2003. Paper prototyping: The fast and easy way to design and refine user interfaces. Morgan Kaufmann.

[25] Kara Pernice. 2016. UX Prototypes: Low Fidelity vs. High Fidelity. (December 2016). https://www.nngroup.com/articles/ux-prototype-hi-lo-fidelity/

[26] Khaled Mohammad Alomari, Tariq Rahim Soomro, and Khaled Shaalan. "Mobile gaming trends and revenue models." International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems. Springer, Cham, 2016.

[27] Aileen Pincus. 2007. The perfect (elevator) pitch. Bloomberg Business Week 18 (2007).

[28] Torgeir Dingsøyr and Geir Kjetil Hanssen. 2002. Extending agile methods: postmortem reviews as extended feedback. In International Workshop on Learning Software Organizations. Springer, 412.