TECH START-UP: BRIDGING THE GAP BETWEEN ACADEMIA AND THE MARKET

C. M. F. A. RIBEIRO, S. O. QUEIROZ, M. A. FREIRE e F. A. ALEIXO[*]
Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte
fellipe.aleixo@ifrn.edu.br [*]

Article received in August/2015 and accepted in October/2015

DOI: 10.15628/empiricabr.2015.3334

ABSTRACT

This paper presents an integrated approach for software development process, which aims to help the transition of innovative ideas with market potential, emerged from academic environment. This integrated process was applied in an undergraduate course at IFRN, to guide the generation of a commercial version of LINKEST product and another group of products in their academic version. Specific details about the results are presented and discussed as follows.

KEYWORDS: Entrepreneurship, Academic Process, Start-up.

STARTUP DE TECNOLOGIA: ESTABELECENDO UMA PONTE ENTRE A ACADEMIA E O MERCADO

RESUMO

Este artigo apresenta uma abordagem integrada para processo de desenvolvimento de software, que tem como principal objetivo auxiliar o processo de transição de ideias inovadoras com potencial de mercado, surgidas em ambiente acadêmico. Este processo integrado foi aplicado em curso de graduação no IFRN, para orientar a geração da versão comercial do LINKEST, bem como de um conjunto de outros produtos em suas versões acadêmicas. Detalhes específicos dos resultados são apresentados e discutidos no decorrer do trabalho.

PALAVRAS-CHAVE: Empreendedorismo, Processo acadêmico, Startup.

1 INTRODUCTION

According to Global Entrepreneurship Monitor report [1], Brazil's level of early-stage entrepreneurial activity (TEA - 17.2%) is higher than Germany (5.3%) and the US (13.8%), countries driven by innovation, as well as China (15.5%) and India (6.6%). The same report points out that just over half (55.5%) of Brazilian population perceives good opportunities to start-up in the area where they live. Only the United States (50.9%) and Mexico (48.9%) have comparable rates, placing Brazil among the countries with generally more positive perceptions regarding available opportunities.

Despite those positive figures, World Economic Forum report on entrepreneurial ambition and innovation (GEM, 2015) [2] shows that Brazil businesses are not innovative and do not expect to hire a lot of employees. Experts cited "Education and training" as a constraint (61%), but also as an enabler (23.8%), suggesting that greater investment in education is a requirement in order to Brazil to reach a leading position of entrepreneurship initiatives. Indeed, lack or insufficient training is presumably the cause for many young people give up on their intention to start an enterprise [3].

Start-up companies are usually related to small early-stage businesses. This work focuses on tech start-ups, which are Internet-based companies that are very sensitive to market mood, since they have to deal with high risk and hard competition. Due to the low initial cost, this kind of company is of particular interest for young Brazilian entrepreneurs, fact that motivates this work. More specifically, we are concerned to investigate strategies that can be used as a bridge between academic innovative ideas and the real market.

In this context, software development process can be considered a valuable strategy to be investigated, as it can serve as an efficient mechanism, not only to mitigating risks of the development process itself, but also because of its potential to collaborating for a gradual transition from an academic life cycle to the entrepreneurship.

This work presents an approach for the integration of agile methodologies for [4] and the lean start-up methodology [5]. The proposal was adopted as part of the academic curriculum of an undergraduate course at Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte (IFRN). It has been validated as a tool to help students to transform academic projects into business. Details and results of the integrated process proposed discuss in this paper, which is organized as follows.

Besides this introduction, Section 2 is dedicated to present methodologies that served as a basis for this proposal, namely Agile methodologies and the Lean Start-up, which has been successfully used for many companies around the world. Sections 3 and 4 discuss details of the proposed integrated approach and a validation scenario, respectively. A real case study is subject of Section 5, and finally Section 6 presents some conclusions and future work.

2 METHODOLOGIES FOR TECH START-UPS

2.1 AGILE METHODOLOGIES

In software development projects, a well-established software development process is necessary to guide the timeline of deliverables and to ensure that every member of the development team understands the expectations of the stakeholders. Agile methodologies are a group of software development methods based on iterative development and continuous feedback from all stakeholders throughout the development cycle in accordance with the agile manifesto [6] that articulates the common principles and beliefs underlying these methods.

Highsmith [7] describes agility as the ability to sense and response to business prospects in order to stay inventive and aggressive in an unstable and rapidly shifting business environment. In this context, agile methodologies are playing an important role in software development companies, their emphasis on frequent communication, transparency of progress and short iterations have been encouraging the use of them in industry, mainly in start-ups companies. According to the "Agile Development Survey [8], the reasons for companies adopt agile practices are accelerate product delivery (59%) or enhance their ability to manage changing priorities (56%). Otherwise, it pointed out as main benefits of agile development the ability to manage changing priorities (87%) and the team productivity (84%). Anyway, these are important indicators for tech start-up companies.

The key benefits of agile methodology [9] are: (i) handling change of requirements due the direct customer involvement; (ii) fault detection due continuous testing feedback; (iii) increased performance due the daily standup meetings that allows to exchange valuable information and to fine tune improvements continuously; (iv) iterative and incremental delivery due the small increments that enable the early feedback from customers and end users; (iv) flexibility of design because it has to be made flexible in order to handle changes easily; and (v) improvement in quality due refactoring, testing, improved communication, and faster turnaround time for blocking bugs.

Scrum is a general-purpose project management framework most commonly used in software development projects [10] [8]. The project progress is managed in a series of iterations called sprints, whose typical duration is 2-4 weeks. The scrum roles are: (i) the development team; (ii) the product owner and (iii) the scrum master. The teams are comprised of multi-functional members and each team member has the autonomy to determine how and when to complete its work. The product owner leads the development effort outlining work in a centred requirements board called product backlog, and prioritizing it based on business value. The product owner acts to facilitate the scrum process, in order to sure that the team is as productive as possible.

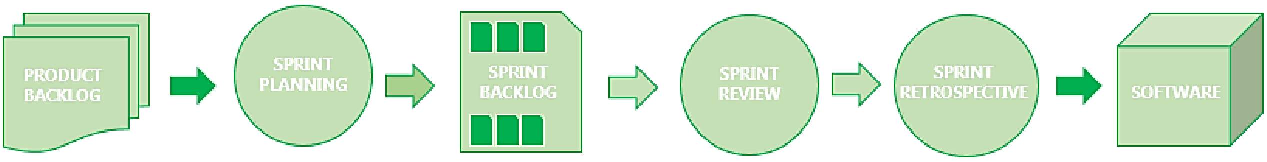


Figure 1. Basic Scrum components
(source: elaborated by the authors based in Scrum process definition)

Figure 1 shows the basic Scrum components. For each sprint, a set of user requirements are selected do be developed during the sprint planning meeting, in the end of the sprint, during the sprint review, team shows what they accomplished during the sprint. After that, lessons learned and improvement ideas are discussed during the sprint retrospective.

Other reference in Agile methodologies is OpenUP [11], whose components are designed to enhance collaboration among team members and to amplify benefits to stakeholders avoiding unnecessary formality and deliverables. OpenUp defines four phases related to the software development life cycle: Inception, Elaboration, Construction and Transition. Besides these phases, OpenUP divides the project development into interactions, which keep a close relationship with Scrum sprints. After each interaction, a new version or build of software is delivered.

The main goal of Scrum, Open UP and Agile methodologies in general is to mitigate the risks of software development, by reducing life cycle duration and by establishing many "build-measure-learn" steps, which will lead to a product that can better attend users' expectations. This is precisely the same purpose of Lean Startup methodology, subject of next subsection.

2.2 LEAN START-UP

The management of a start-up brings a lot of challenge. Therefore, new management ideas have emerged in order to support this kind of companies. Lean startup is a method for developing businesses and products proposed by Eric Ries [5]. It provides a scientific approach to creating and managing startups and minimize the products time to market getting customer feedback as soon as possible. The idea is focused on create a minimum viable product to get early feedback from customer.

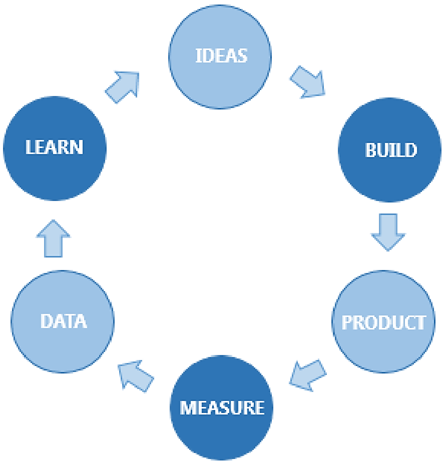


Figure 2. Lean Startup abstract process
(source: Eric Ries [5] **)**

The lean startup abstract process (Figure 2) defines a continuous cycle of learning for innovation. It is defined as a build-measure-learn cycle [5]:

- Build Step: the hypothesis test is used to build the idea. A minimum viable product (MVP) is built;
- Measure Step: appropriate metrics are collected for testing the hypothesis. Metrics that can demonstrate cause and effect question;

- Learn Step: the experiment results are analyzed then learn whether to pivot or persevere.

This fast cycle of innovation helps to eliminate uncertainty inherent in the business environment. It accelerates the feedback loop.

In the scenario of tech startup, agile methodologies and lean startup principles can be naturally combined, in order to produce innovative software products. As they commonly operate in an environment of uncertainty, the fast cycle proposed by the use of these methodologies can potentially mitigate risks and reduce costs of software developments projects, by contributing for the company success as a whole.

3 AN INTEGRATED APPROACH FOR SOFTWARE DEVELOPMENT IN TECH STARTUPS

3.1 Conceptual Foundations

The basic concepts adopted by this work can be grouped into two complementary categories. The first one is related to the academy concerns, and the second one about market issues. Indeed, as an integrated approach, the starting point of this work is to point out similarities and differences between those environments, in order to harnessing the best from both.

Educational institutions have their own set of values that are essential to reinforcing academic integrity. They provide a foundation for responsible conduct in students´ lives after graduation. Besides this ethical background, the academic environment should also provide technical abilities achieved not only from theoretical content, but also from practical experiences. In this regard, the proposal is based on the specific curriculum components of TADS – Systems Analysis and Development Technologies – undergraduate course at IFRN. This course is under execution and has prepared professionals for more than 10 years, which are recognized for their expertise in software development.

The market mood is dictated by aspects related to demand, trends, and the continuous movement of competitors towards improvement of their market share. This produces profound changes in the way the explicit knowledge that is learnt at the academy is applied in real scenarios. Actually, a given methodology, which has proved its value in challenging situations at big companies, has not its applicability in small companies automatically granted. In the case of tech start-ups, which are precisely the focus of this proposal, it is critical to review methodologies, methods, and techniques in order to assess their feasibility in case of high risk and very limited resources.

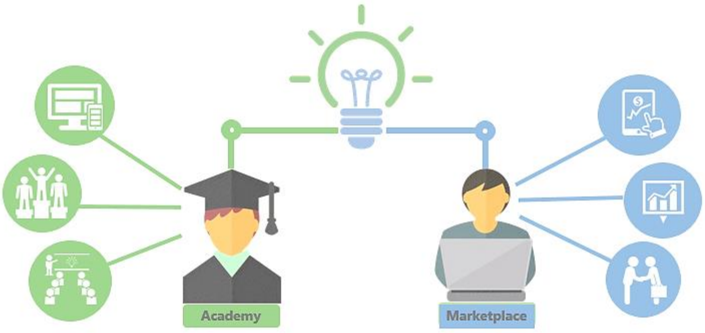


Figure 3. The motivated scenario (source: elaborated by the authors)

Despite the differences between academy and market, some innovative ideas have succeeded in surpassing the limits of academia towards the market, as depicted by Figure 3. This is precisely the kind of situation that motivates this proposal, whose core components are subject of next subsection.

3.2 Core Elements

The integrated approach is made up by two kinds of methodologies. Agile methodologies give the product development guidelines, as detailed at section 2.1, and the Lean Startup [5] gives the necessary market perspective (section 2.2). Figure 4 summarizes the integration of these two perspectives.

The external ring of Figure 4 represents the innovation management processes. Modern approaches state that innovation must be developed in a continuous manner [12]. The first process of innovation management is related to **prospection** (1.1) of market. Analysis of data about similar products is one of the activities of this process, which looks for establishing a new value to the clients. The second process namely **ideation** (2.1) manages with creativity. Indeed, there is no innovation without good ideas. However, prospection and ideation are not enough. It is necessary to establish a strategy to select the best form to implementing such an ideas. This process is called **strategy development** (2.2).

Resource mobilization (3.1) is a requirement for innovation **implementation** (3.2). In the context of startups, this process is extremely critical, since startups team is commonly small and multidisciplinary. It means that it is not possible to dedicate resources for innovation management activities exclusively. The evaluation process (4.1) is essential to understand the effectiveness of the strategy adopted. According to the integrated approach, the innovation management processes serve as an interface between the market and the startup.

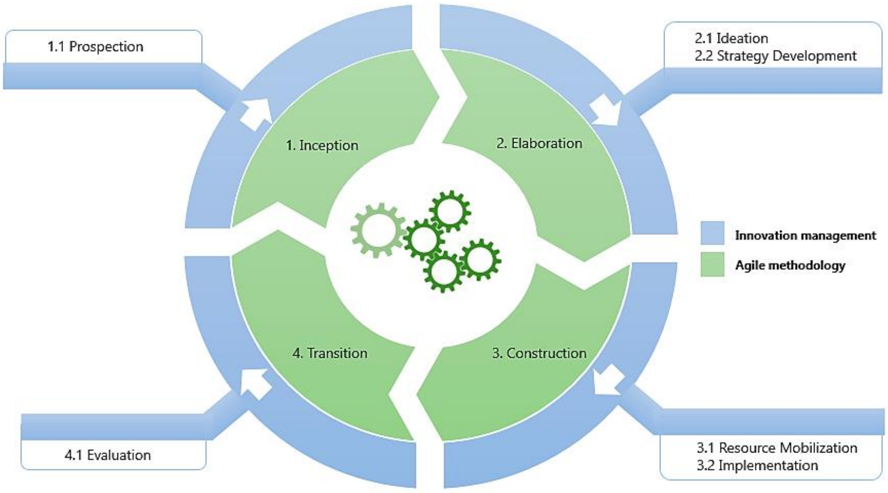


Figure 4. The Integrated Approach Overview
(source: elaborated by the authors)

The internal ring of Figure 4 represents software development process based on agile methodologies, as discussed in Section 2. At this abstraction level, the four phases defined by OpenUP are illustrated, namely Inception (1), Elaboration (2), Construction (3), and Transition (4).

It is worth noting that inside of the iterations of these phases work the Scrum dynamic, as illustrated by Figure 1.

During the Inception phase, the main concern is to guarantee that the scope and requirements are clearly understood by stakeholders. Indeed, statistics show that the agreement between stakeholders on what exactly the software will provide is one of the most important source of conflict, as demonstrated by the well-known Pareto principle (80-20) [13].

The Elaboration phase is dedicated to describe in a more technical way the functionalities that will be provided. Details about software architecture, strategy for risk mitigation, among others, are also defined. The Construction phase deals with programming and other technical tasks, which makes it possible to present an executable version of software. The Transition phase is an opportunity to get some insights from lessons learned, as well as to validate the user satisfaction level.

3.3 Integration of core elements

Having detailed the core components individually, it is essential explain how such components can cooperate to help tech startups in their efforts to produce innovative products. In this context of integration, explicitly represented by Figure 4, it is important to notice the continuous movement represented by arrows at the external ring, the innovation management perspective. In fact, continuous improvement is a basic tool to survive in very competitive scenarios that tech startups have to face.

Another key feature of the proposed integration approach is the external position of the innovation management process and the internal position of the agile methodology. This is due the necessity of every startup to stay opened to the market, in order to understand its movement of change, avoiding investments in wrong direction and harnessing good opportunities. This openness characteristic is critical to synchronize internal efforts with external demands.

At the proposed integrated approach, the two rings overlay is not made by chance. Actually, all phases and activities, either those related to innovation management or software development, must be synchronized. Otherwise, the new product could reach the market later or with features not aligned with users expectations.

In regard the specific phases, there are four integration points. The first one establishes synchronization between prospection phase (1.1), at the innovation management perspective, and the inception phase at Agile methodology perspective. Indeed, as the main goal of these phases is to propose new values to the clients, avoiding waste of resources by scope creep situation, for example, it is reasonable to suppose that the integration can foster the achievement of such benefits.

At the second integration point, new ideas that emerged from ideation phase (2.1) and the strategy to deliver them will certainly cause some impact on respective software functionalities. Therefore, descriptions of these functionalities must also be reflected on guidelines and orientations to the construction phase. It necessarily implies into analysis and selection tasks, in a process of identifying the choices with better chances to be accepted by the market.

The third integration point correlates the construction phase (3), at the software development perspective, and two innovation management processes, namely resource mobilization (3.1) and strategy implementation (3.2). A special attention must be paid at this integration point, as resources availability in big companies is very different from startups, where a small multidisciplinary team has to deal with all demand. This apparent constraint is actually an opportunity to create an environment where innovation becomes a natural part of each task.

The fourth point of integration relates the transition phase (4) of Agile methodology, and the evaluation phase (4.1) of innovation management process. This integrated phase occurs after the conclusion of a minimum viable product (MVP), stated by Lean startup methodology. In tech startups, this is a special moment to collect data and to reason about the innovative aspects of the product. Putting in other words, it is represented by the measure and learning steps of build-measure-learn Lean startup principles.

As mentioned at section 3.1, this integrated approach targets academic environment, where new ideas with market potential can appear. The main goal is to provide a means to make a smooth transition from academy to the market, more precisely to startups companies. Despite the fact that the validation of this proposal is an ongoing process, the premises adopted are based on a solid ten-year experience at TADS. To fully understand the premises and the specific context that motivated this work, the next section presents the main aspects of theoretical-practical approach adopted as a curricular component of TADS.

4 USAGE SCENARIO

This section presents a ten-year experience of TADS course at IFRN. More specifically, the theoretical-practical approach used as a curriculum component. Two different, but interrelated aspects are discussed. The first one is the strategy for problem definition, which motivates software development. The second one is related to the process of giving gradual autonomy for students, when it comes to make decisions about the project. These key principles that support the proposed integration approach are subject of next subsections.

4.1 Towards new solutions for real problems

The TADS curse of the IFRN has as one of its intents to provide to its students the experience of working in real problems. It is actually organized in six semesters. At the third to fifth semesters the students are organized in teams that are conducted to experience the development of software systems. This organization was made to offer three stages in the experience in software development process. Groups of five students have to deal with client demand and other software development activities, including analysis, programming, test and documentation. By the end of the term, students have to present an executable version of software, which conforms to client requirements and the technical characteristics asked as academic requirement.

At the first stage, the teacher guides the students through a software development process. In this stage, the teacher acts as the client of the proposed software. At the second stage, the students receive more responsibilities in the conduction of the software development process. At the third and final stage, the students have the full responsibility about the software development process. In this final stage, one of the objectives was to afford the experience of working in real problems. At this final stage, three different strategies for problem definition were adopted in the related software development projects. Each one presents advantages and disadvantages, later discussed.

The first strategy was to use real clients; where the problem definition was derived from the corporative demand. This strategy allowed to getting closer to market partners, which is very important to establishing collaborative projects. Having a corporative client, students had the possibility to practice very consolidated technics for requirements elicitation. It was also important to give students a real vision of marketplace. This experience was rich, but became clear that the different rhythms of the market and the academy were more harmful then beneficial. The strong differences between timetables prevented the effectiveness of software development and implementation by using this strategy. Meetings were also a problem, because it usually respected availability of stakeholders, not academic demands. The potential disruption of group arrangement made it almost impossible to resume the same project in the next term, reducing the quality and coverage of functionalities developed. Finally, the management of the client expectations just increased the pressure for results. This scenario was responsible for some looses in the teaching and learning process.

In order to avoid the mentioned situation, the real corporate client strategy was replaced by a fictitious one represented by the professor. At this second strategy, the demand was induced and the professor had to play a double role, as a mentor and as a client. The main advantage of this approach is to permit the accomplishment of the academic requirements in time by following a very disciplined routine. During the application of this strategy, a side effect could be perceived: the professor becomes overwhelmed and the flip side is just not establishing an appropriate environment for professional practices, which of course include not attend users expectations. This strategy could create some confusion in the students, receiving conflicting instructions by the client and instructor roles, from the same person. This strategy was adopted during a while, but it was abandoned just because this reason. These experiences brought a question: what would happen if the team itself defines the system to be developed? At this point, was decided that the students would be stimulated to propose ideas to solve real and community problems.

The strategy used nowadays is to give students freedom to choose a real problem they believe there is a potential market opportunity to explore it. Despite its recently adoption, the exercise of observing real demands have waken up entrepreneurship characteristics in many students. This is very interesting in the actual economic scenario, not only in local basis, but also all over the world. The scope of the proposed projects was delimitated to the main theme of: smart cities. Each team could choose the areas of: (i) Government and Agency Administration, (ii) City Planning and Operations, (iii) Buildings, (iv) Energy, (iv) Water, (v) Transportation, (vi) Education, (vii) Smarter Care, (viii) Social Programs and (ix) Public Safety [14]. The knowledge involved in the application of this strategy extrapolates the conventional software development process theory. This experience adds important knowledge to the students that soon will leave the academy, and will face the real market challenges. In this context, the integrated approach is particularly feasible to support innovative ideas transition, from academy to the market.

This strategy still not perfect, and some answers are still persecuted. For example, as the team acts as product owner, the requirements remain unclear until system versions with reasonable number of functionalities are produced. Often, when the system reaches this maturity level, the project is close to its ending. Even with this limitation, the experience provided by this strategy is quite satisfactory.

4.2 Improving management maturity level for software development

The second key aspect of integrated approach that deserves a special attention concerns gradual change in maturity level of students. This is also an assisted process during software development experience. This aspect has not suffered hard changes during ten-year covered by this work. This is due to the fact that it does not depend on external factors, since three curricular components provide opportunity to realize such a strategy. Despite highly recommended students could change the project they had worked during the previous term. To keep the same project during three terms has proved to be an important way to improve quality level of software developed.

The main aim to use the three stages mentioned in the previous section was to slowly upgrade the maturity level in the management discipline of the software development process. In the first stage occurs the first project development module, when the professor performs the project manager role. The professor defines the phases, activities and tasks to be performed by the team. The professor defines and distributed activities and to make all technical decisions, giving students guidelines that they can follow in real projects. This first stage is also the first contact of the students with a complete software development process. The professor uses this context to enhance student leaning.

Based in the lessons of the first stage, at the second stage students have the chance to play different roles, such as project manager, developer, software architecture, etc. To share the experience of performing the project manager role with the whole team, a student is chosen to perform the project manager role at the beginning of iteration. The teacher acts as a mentor, guiding the project manager to conduct the activities during the iteration. At the end of each iteration is realized a review meeting, to discuss the problems and the success of the iteration.

Finally, during the third and last project development experience, students have achieved the maturity level necessary to make their own decisions, as reported in the next section. Each team can choose a different software development process as base. The team also chooses the roles performed by each member, and agrees with the responsibilities of each role. At this stage, the maturity acquired at the previous stages is put at proof. The maturity at this stage allows the team to face some responsibilities beyond the software development process. The teams can be charged to analyze and develop: (i) opinion polls, (ii) market research, (iii) business plan [15], (iv) feasibility analysis, etc., exercising the entrepreneurship of the involved students [5].

5 A REAL CASE STUDY

This section is dedicated to present a real case study of transition between academy and the market, involving a group of students of TADS, who developed a software named LINKEST (www.linkest.net). Lessons learned from this rich experience are presented as a testimonial according to the point of view of a participant and coauthor of this paper. The experience of producing LINKEST was divided into two moments, which can be better explained by using software academic and commercial versions, described as follows.

5.1 LINKEST academic version

According to LINKEST team, many aspects of software development process are source of concerns for students. To define a real problem that deserves a computational solution and make an effective plan to implement it, by following good practices, is definitely not a trivial task. Solid knowledge acquisition at academy is pointed out as a basis upon which practical experience can succeed.

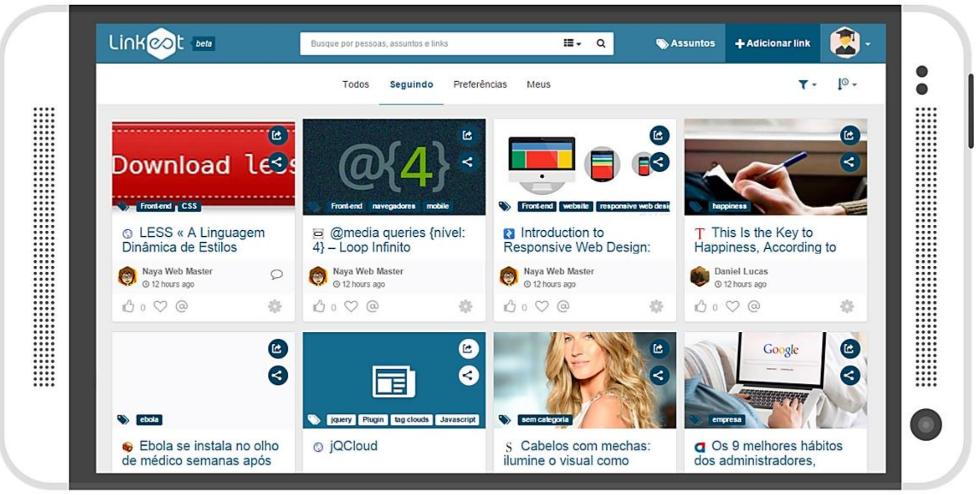
The gradual increasing of autonomy is also mentioned as a key point of the methodology adopted at TADS. This strategy allowed the team to exercise some roles, e.g. project management, as the maturity level increases.

Deep understanding of the users' need is also mentioned as a challenge, but also as good opportunity to transform LINKEST from an academic idea into business. Early versions, students and professors played the role of customers of the LIKEST project. During two academic terms, LINKEST was gaining professional features. According to the team, agile methodology was critical to reach the expected results. The fast and short cycle of Scrum sprints made it possible to keep the scope and attributes agreed, as well as to establish synchronization among team members.

At the end of academic process, according to LINKEST team, the bunch of functionalities developed were very well assessed by professors. However, by asking real users, it became clear that any new development effort should be preceded by a deep investigation of users needs. Such perception came only during a pitch elevator session, to get an opportunity to participate in an incubator call. The team concluded that there is a clear gap originated from the disconnection between academic process and market expectations.

5.2 LINKEST commercial version

The transition to market, mediated by IFRN incubator, inaugurated a different moment to LINKEST team. Technical issues were replaced by business ones, according to participants. It was clear, at that moment, that among risks elicited since then, competitors and their similar software must be considered a priority. Academic premises, such as zero error versions, were also replaced by a necessity to deliver as soon as possible, a minimum viable product (MVP version) to collect data from user experience.



LESS « A Linguagem Dinâmica de Estilos
Naya Web Master
12 hours ago
0

@media queries {nível: 4} – Loop Infinito
Naya Web Master
12 hours ago
0

Introduction to Responsive Web Design:
Naya Web Master
12 hours ago
0

This Is the Key to Happiness, According to
Daniel Lucas
12 hours ago
0

Ebola se instala no olho de médico semanas após

jQCloud

Cabelos com mechas: ilumine o visual como

Os 9 melhores hábitos dos administradores,

Figure 5. LINKEST beta version

Knowledge on business models and business strategies became as fundamental, as cloud computing service models to guarantee scalability and performance. An innovation strategy has to be built from scratch, by using competitors marketing plan as a basis. However, such a strategy is not enough to get market share. The Lean startup methodology was then adopted as a means to get fast background to help running startup business, minimizing the cost of building unnecessary functionalities.

Like human faces, the interface is probably the most visible aspect of software. By looking at Figure 5, it is possible to recognize many of decisions made to attend academic requirements and show technical abilities, for instance, by replacing simplicity by big set of functionalities. Six months of market experience and hundred of users later, it becomes clear that it is critical to make effort to change this direction.

6 CONCLUSION AND FUTURE WORK

This paper presented an integrated approach designed to lead innovative ideas that emerged from academic environment towards the market. The motivation for this work is a real situation, based on a decade of teaching experience and preparation of software developers. Having in mind that the training that has been offered has privileged mostly the preparation of employees rather than to encourage entrepreneurship, some changes were made in curricular components. These changes were fully described and discussed in this paper, as well as the usage scenario and a real case study.

As other educational investigations, this is a continuous ongoing work. Therefore, many activities are planned for future development, including the definition of tasks that can be followed at more operational levels. For example, the project manager at startups usually has to add new attributions related to innovation management. Guidelines like those established by PMBOK are extremely valuable as a means to keep the agenda, relating concerns from different knowledge areas, such as risk and communications management. Besides that, it is worth noting that each new academic term, new groups come, new computational solutions have to be developed, what necessarily put some new challenges that must be evaluated and solved.

7 REFERENCES

1. GLOBAL ENTREPRENEURSHIP MONITOR. GEM 2014 Global Report. **Global Report**, 2014. Disponivel em: <http://www.gemconsortium.org/report/information/49079>. Acesso em: 06 jun. 2015.
2. GLOBAL ENTREPRENEURSHIP MONITOR. World Economic Forum Report on Entrepreneurial Ambition Features Insights from Endeavor Entrepreneurs. **World Economic Forum**, 27 Jan 2015. Disponivel em: <http://www.endeavor.org/research/wef-report-on-entrepreneurial-ambition-features-insights-from-endeavor-entrepreneurs/>. Acesso em: 14 Ago 2015.
3. GELDHOF, G.J. ET AL. Fostering Youth Entrepreneurship: Preliminary Findings From the Young Entrepreneurs Study'. **Journal of Research on Adolescence**, v. 24, n. 3, p. 431–446, 2014.
4. PRESSMAN, R. **Software engineering:** a practitioner's approach. New York: Palgrave Macmillan, v. Sixth, 2005. 880 p.

5. RIES, E. **The Lean Startup**. New York: Random House LLC, 2011.
6. AGILE ALLIANCE. Agile manifesto. **Manifesto for Agile Software Development**, 2001. Disponivel em: <http://www.agilemanifesto.org>. Acesso em: 15 ago. 2015.
7. HIGHSMITH, J. A. **Agile software development ecosystems**. Salt Lake: Addison-Wesley Professional, 2002.
8. VERSION ONE. The 9th Annual State of Agile Survey. **Annual State of Agile Survey**, 2015. Disponivel em: <http://www.versionone.com/pdf/state-of-agile-development-survey-ninth.pdf>. Acesso em: 10 ago. 2015.
9. KUMAR, ; BHATIA, P.. Impact of Agile Methodology on Software Development Process. **International Journal of Computer Technology and Electronics Engineering**, v. 1, n. 4, p. 46-50, 2012.
10. AHMAD, S. et al. **Agile Software Development:** Impact on Productivity and Quality. IEEE International Conference on Management of Innovation and Technology. Singapore: IEEE. 2010. p. 287 - 291.
11. ECLIPSE FOUNDATION. OpenUP Introduction. **OpenUP**, 01 jun. 2012. Disponivel em: <http://epf.eclipse.org/wikis/openup/>. Acesso em: 30 jul. 2015.
12. STEFANOVITZI, ; NAGA, M. S. Gestão da inovação de produto: proposição de um modelo integrado. **Associação Brasileira de Engenharia de Produção**, São Paulo, v. 24, n. 2, 2014.
13. SANDERS, R. The Pareto principle: its use and abuse. **Journal of Services Marketing** , v. 1, n. 2, p. 37-40, 1987.