Informatics in Education, 2021, Vol. 20, No. 1, 1–17

DOI: 10.15388/infedu.2021.01

Drivers and Indicators of Innovation to Educational Software

Rafael BARBOSA[1], Ricardo SOUZA[1,2]

[1]*Graduate Program of Applied Informatics (PPGIA), Federal Rural University of Pernambuco (UFRPE), Rua Dom Manoel de Medeiros, s/n, Dois Irmãos, CEP 52.171-900 – Recife – Brazil*
[2]*Department of Computer Science (DC), Federal Rural University of Pernambuco (UFRPE) Rua Dom Manoel de Medeiros, s/n, Dois Irmãos, CEP 52.171-900 – Recife – Brazil*
e-mail: s.rafaelbarbosa@gmail.com, ricardo.souza@ufrpe.br

Received: December 2019

Abstract. Software is usually considered enabler for innovation in education. Thus, it is necessary to investigate: What leads to innovation in education and which, therefore, should be considered in the development of educational software? How to support decision-making for choosing software to be used in teaching and learning? How to evaluate software that enables, transforms, or supports innovation in education? In order to answer these questions, a research work was conducted that resulted in an approach called DEISE, which is concerned with drivers that lead to innovation in education and indicators that measure the fostering of software to skills need to students for 21st century learning. Through a web system developed for technological support to the DEISE approach, a set of educational software was evaluated by educators and the results show the index of innovation in education for each evaluated software and a comparison between educational software of similar purpose.

Keywords: innovation in education, 21st century skill, driver of innovation, indicator of innovation, educational software.

1. Introduction

Factors such as globalization, new technologies, migration, international competition, ever-changing markets, and environmental and political challenges are motivating a new educational model that is better suited to the challenges of the 21st century (Luna Scott, 2015a). Among the main problems pointed out by experts is that students are not being equipped with the skills necessary for a satisfactory and productive personal and professional life (Luna Scott, 2015a). For example, learning to learn is a skill that leads to lifelong continuing education.

According to World Economic Forum report (Partovi, 2018), we are facing the challenge of redefining elementary education to keep up with the skills evolution students need to solve, innovate and succeed. Students need to improve skills and learning as a matter of urgency to address persistent global challenges (Saavedra & Opfer, 2012). The introduction of new content in the elementary school curriculum, such as concepts of computer science and computational thinking, aims primarily to stimulate student skills such as creativity, problem solving, ethics and collaboration (Partovi, 2018).

To bring about the changes required in the educational model, we must innovate. Innovation is considered a key element in the knowledge-based economy and fundamental to the continuous improvement of education and the enhancement of learning outcomes, equity, cost efficiency and student satisfaction (Looney, 2009). Innovation in education manifests itself through processes and products that significantly improve teaching and learning as well as facilitate student development. Digital technologies, especially software products, are tools that if properly used can enable, transform, or support innovation in education.

Support by software products to the teaching and learning process is already a reality in many educational institutions, providing a virtual extension of the classroom in which students can learn at their own pace and provides an alignment of education with the new generation of students fluent in technology. However, the offer of educational software is quite extensive and larger than the demand. A question that then arises is how to make the decision to select software to support the teaching and learning?

According to UNESCO report (Luna Scott, 2015b), innovation in education is strongly related to providing students with 21st century skills which consist of a combination of critical thinking, autonomy, creativity, collaboration and communication. However, the ability to measure innovation is essential to an education improvement strategy (OECD, 2014). On the other hand, evaluation is vital to the innovation process (Looney, 2009). Those implementing innovative projects need to evaluate their effectiveness and make the necessary adaptations. Evidence of the impacts of new approaches is also essential for successful dissemination.

Software products, considered as differentials to improve teaching and learning and for student development, need to be evaluated against indicators that measure innovation in education. Indicators can be defined as measures used to describe and analyze actual situation, assess the achievement of objectives and goals and its changes over time, and predict trends.

However, educational software assessment approaches generally address only issues of interest of any kind of software, such as human-computer interaction and usability, as well as other software quality requirements such as performance, portability, and maintainability. In this way, another question arises: How do we measure educational software as enabler of skills needed to the students for 21st century learning and working?

Evaluation by indicators serves to measure the innovation of educational software already developed. However, in the case of developing new educational software, it is also necessary to identify which factors, if satisfied, contribute to achieving innovation in education. Such factors are called innovation drivers. Driver is therefore a factor that "leads to" or "bridges" innovation. Thus, the last question that motivates the development of this work is: What factors should be considered in the educational software development process, which are drivers for innovation in education?

This paper then presents the development of an approach called DEISE (acronym in Portuguese language to ***De****terminantes e* ***I****ndicadores para* ***S****oftwares* ***E****ducacionais*) which specifies: drivers used as guideline to educational software development and indicators used to measure educational software as enabler of skills needed to the students for 21st century learning and working. In addition, a web-based system was developed for technological support to the DEISE approach. Through web system developed a set of educational software was evaluated from specified indicators.

In addition to this introductory section, this paper is structured into the following sections: Theoretical background; Specification of the DEISE approach; Results and Discussion.

2. Theoretical Background

According to OECD (2010), innovation in education involves dynamic change that adds value to the educational process and results in measurable results, whether in terms of stakeholder satisfaction or student performance. For Fullan (2007), educational innovations must contain at least three elements: (1) the possible use of new or revised materials (instructional resources such as curriculum materials or technologies); (2) the possible use of new teaching approaches (i.e. new teaching strategies or activities); and (3) the possible change in beliefs (e.g. pedagogical assumptions and theories underlying new specific policies or programs).

The OECD (2017) proposes an innovative learning structure, called "7 + 3", consisting of seven principles for learning environments and three fundamental dimensions for innovation in education. The principles for learning environments are: (1) to recognize students as key participants, encouraging their active engagement and developing their understanding of own activity as learners; (2) be based on the social nature of learning, actively encouraging well-organized cooperative learning; (3) teaching professionals should be highly attuned to student motivations and the fundamental role of emotions in performing tasks; (4) accommodate individual differences between students; (5) demand hard and challenging work from students, but without excessive burden; (6) operate with clear expectations and implement assessment strategies consistent with those expectations, providing formative feedback to support learning; and (7) promote connection with other areas of knowledge and subjects.

The dimensions of innovation in education are: (1) to innovate the pedagogical core of the learning environment, either in the main elements (students, educators, learning content and resources) or in the dynamics that connect them (pedagogy and formative assessment, use of time and organization of educators and apprentices); (2) know the outcome and monitor the level of learning achieved by different teaching strategies and innovations; and (3) work with different media such as families and communities, higher education, cultural institutions, media, business, and especially other schools and learning environments, to directly shape the pedagogical core and learning leadership.

Rethinking pedagogy is a central activity for innovation in education. Pedagogy is the set of teaching and learning practices that shape the interaction between teachers and students. Pedagogy provides teachers with decision-making possibilities on how they teach and may appropriate the use of digital technology as a complement rather than as a substitute for teaching (Peterson *et al.*, 2018).

Among pedagogical approaches that represent alternatives to the traditional teaching and learning are: Game-based learning (Oblinger, 2004) – provides immediate feedback, allowing students to test hypotheses and learn from their actions within the game, and enable self-assessment through scoring mechanisms and range of different difficulty levels; Flipped classroom (Bergmann & Sams, 2016) – instructional content is provided in advance, and during class, students seek to apply theoretical knowledge in interactive group learning and / or problem-solving activities with teacher guidance; and Problem-Based Learning (PBL) (Berbel, 1998; Prince, 2004). – encourages active student participation in the search of knowledge, encourages group work to identify what needs to be learned to solve a particular problem.

Traditional pedagogy emphasizes the memorization or application of simple procedures that do not promote in the students skills such as critical thinking and autonomy (Luna Scott, 2015a). Students need to engage in learning that has genuine relevance to them personally and to the communities in which they are inserted. Real-world experiences coupled with engagement and collaboration provide opportunities for students to build and organize knowledge, engage in research, investigation, writing and analysis, and communicate effectively. The collaborative process makes students consider new uses for knowledge (apud Darling-Hammond *et al.*, 2015).

UNESCO has produced a report (Luna Scott, 2015a) that compiles several studies, such as the P21 framework (P21, 2011) and Delors report (Delors, 1996), which define the so-called skills required for students for 21st century learning. Skill is a developed aptitude or ability to use one's knowledge effectively and readily in execution or performance (Merriam-Webster, 2016). Some these 21st century skills can be fostered through educational software such as:

- **Autonomy** – Student's ability learning how to learn, set his/her learning goals, plan and monitor learning, produce new knowledge without the formal aid of an educator. Educational software can act as a kind of virtual tutor, guiding the student on a learning path.
- **Adaptability** – Students' ability to adapt to different forms of teaching and learning and to apply knowledge in different ways to solve problems. Educational software can support different learning paradigms, such as flipped classroom, PBL, etc.
- **Analytical thinking** – Students' ability to analyze information and use logic to address problems. Educational software can enable different ways or strategies to solve a problem.
- **Collaboration** – Students' ability to interact effectively with other students, participate in a collective process to accomplish a task (teamwork) and share knowledge. Educational software can assist in the development of collaborative activities, such as shared writing of a document online.
- **Communication** – Students' ability to communicate with others or to communicate understanding of knowledge acquired. Educational software can enable syn-

chronous and asynchronous communication between students, as well as provide visual means for demonstrating learned content.
- **Creativity** – Students' ability to question the status quo and articulate different ideas to address a problem and then choose the one that best fits the context. Educational software can assist creative activities, such as storytelling, digital prototyping, visual mapping, etc.
- **Critical thinking** – Students' ability to access, analyze and synthesize information needed to compare evidence, evaluate competing alternatives and make responsible decisions. Educational software can provide feedback about different solution alternatives.
- **Knowledge transformation** – Students' ability to make diverse transformations between tacit and explicit knowledge. Educational software can allow different pieces of knowledge be combined into a new form of knowledge.

3. Specification of the DEISE Approach

The 21st century skills are related to emerging development models (Ananiadou & Claro, 2009). An education system aligned with today's demands must be concerned that students acquire these skills which are considered essential for his/her personal and professional life. The DEISE approach aims to contribute to innovation in education by providing a strategy to drive development and to evaluate educational software from the perspective of fostering the skills needed to the students for 21st century learning.

3.1. *Conceptual Model*

Fig. 1 illustrates the Conceptual Model which provides an overview of the concepts, and their relationships, that guided the development of the DEISE approach. The DEISE approach is based primarily on drivers and indicators of innovation in education. Conceptually, drivers lead to innovation in education, while indicators measure innovation in education. The drivers are technical and pedagogical factors that can be considered during the educational software development process to direct and increase the chance of success in the pursuit of innovation in education. Indicators are factors that assess whether educational software fosters students' 21st century learning skills, which are among the key goals of innovation in education.

The DEISE approach aims to support the educational software development and evaluation. This strategy was implemented through a web-based system. The possible users of the developed web-based system are: educational software industry; educational institutions; educators; and education experts.

Education experts are professionals who have experience using software products to assist the teaching process. They can use the web-based system to evaluate such educational software by their level of compliance with specified indicators. The edu-

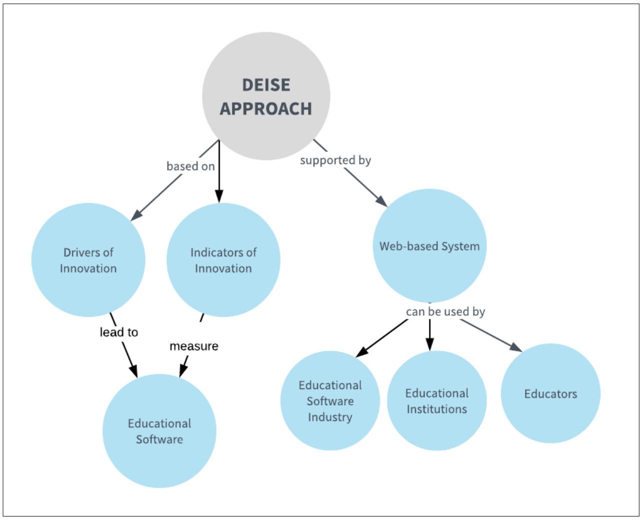

Fig. 1. DEISE Approach Conceptual Model.

cation software industry comprises individual developers and development companies who can use the web-based system to: consult the set of drivers as a checklist to guide the software development process; and consult the level of innovation of the developed educational software, calculated from the assessments made by the education experts. Educational institutions and educators can use the web-based system to assist with decision-making about selecting educational software to provide students with specific learning skills based on innovation indicators according to education expert assessments.

3.2. *Drivers of Innovation*

The drives of innovation serve as guideline to educational software development and are grouped into two categories: technical, related to computational processing and other technological issues; and pedagogical, related to teaching and learning practices.

In the DEISE approach, drivers of innovation were identified from a systematic literature review mapping. (Barbosa & Souza, 2018) and from analysis of technical reference documents in education produced by the Organization for Economic Cooperation and Development – OECD (OECD, 2014; Peña-López, 2017). Table 1 shows the identification and description of the drivers of innovation, as well as their source and category.

The OECD is an international organization aimed at fostering discussion and sharing of experiences between governments on the economy and social welfare, as well as seeking solutions to common problems, including those in the area of education. (`http://www.oecd.org/education`).

Table 1

Drivers of Innovation

Driver:	Computational processing capacity.
Description:	The educational software works properly on machines (e.g. microcomputers, mobile devices, etc.) with basic computational processing configuration.
Source:	(Valiente, 2010)
Category:	Technical
Driver:	Active teacher participation in the educational software development process.
Description:	Teachers participate from the conception (idea and requirements) to the functional tests (evaluation and verification) of the educational software.
Source:	(Valiente, 2010)
Category:	Technical
Driver:	Educational software maintainability.
Description:	Easy, accurate and economical educational software maintenance.
Source:	(OECD, 2017)
Category:	Technical
Driver:	Teacher proficiency in using the educational software.
Description:	Teacher training level to use the educational software in teaching activities.
Source:	(Avvisati, Hennessy, Kozma, & Vincent-Lancrin, 2013; OECD, 2017; Valiente, 2010)
Category:	Technical
Driver:	Educational software availability.
Description:	Educational software availability for use (anywhere and anytime) by students and teachers.
Source:	(Istance & Kools, 2013)
Category:	Technical
Driver:	Trial and evaluated educational software before distribution.
Description:	Students, teachers, and education experts experiment and evaluate a beta version of the educational software to provide improvement suggestions for the final version.
Source:	(Valiente, 2010)
Category:	Technical
Driver:	Feedback on student performance.
Description:	Feedback on student performance in learning activities conducted through the educational software.
Source:	(OECD, 2017; Valiente, 2010; Veraszto, do Amaral, & Barreto, 2013)
Category:	Pedagogical
Driver:	Challenging environment and student reward.
Description:	Educational software provides different levels of difficulty to perform tasks and provides rewards according to student performance and involvement.
Source:	(Booyens, Molotja, & Phiri, 2013; Hosie, Schibeci, & Backhaus, 2005; Istance & Kools, 2013)
Category:	Pedagogical
Driver:	Parent participation in the student learning process.
Description:	Educational software enables parents to actively participate in the student learning process (e.g., requesting specific assignments, providing rewards, tracking performance, etc.).
Source:	(OECD, 2017)
Category:	Pedagogical
Driver:	Integration of educational software into teaching planning.
Description:	Level of integration of the educational software with the academic activities provided for in the teaching planning (e.g., compulsory use of the educational software to perform academic tasks).
Source:	(Hosie *et al.*, 2005; Istance & Kools, 2013; Valiente, 2010)
Category:	Pedagogical
Driver:	Educational software provides teacher with formative student assessment.
Description:	Educational software allows assessment and monitoring of individualized student performance.
Source:	(Istance & Kools, 2013)
Category:	Pedagogical

3.3. *Indicators of Innovation*

Indicator of innovation is used to measure and monitor innovative performance, the capacity to innovate and setting guidelines. In the DEISE approach, indicators of innovation are aligned with one of the key-goal education which consists in to enable students with skills needed for 21st century learning and working, as reported by OECD (Ananiadou & Claro, 2009) and UNESCO (Luna Scott, 2015a). Table 2 presents the identification, description and skill-related of the indicators of innovation specified well as example of educational software that can be used to foster this skill.

Table 2

Indicators of Innovation

Indicator:	I01. Educational software as enabler to self-managed learning.
Description:	Enable the student self-learning, set his/her own learning goals, plan and monitor his/her own learning, evaluate his/her own progress.
Skill:	Autonomy.
Example:	MinecraftEdu (education.minecraft.net) provides instructions and means for the student to chart and develop his/her own learning strategy.
Indicator:	I02. Educational software as enabler to different learning styles and paradigms.
Description:	Enable the student self to adapt to different forms of learning, such as active learning, gamification, etc.
Skill:	Adaptability.
Example:	Khan Academy (khanacademy.org) can be used as an extension of the traditional classroom for e-learning and flipped classroom.
Indicator:	I03. Educational software as enabler to personalized learning.
Description:	Enable the student learning at his/her own pace and proficiency level.
Skill:	Autonomy and adaptability.
Example:	Symbaloo (symbaloo.com) allows the availability of different learning paths for the student to choose from which is most appropriate for his/her pace and level of knowledge.
Indicator:	I04. Educational software as enabler to teamwork.
Description:	Enable individual contribution of the student in a collective effort to accomplish a task.
Skill:	Collaboration.
Example:	Google Docs (google.com/docs) enables collaborative making of online documents, spreadsheets, and presentations where a student can contribute, and changes made can be viewed by the group in real time.
Indicator:	I05. Educational software as enabler to peer learning.
Description:	Enable the student to share knowledge and learn by teaching.
Skill:	Communication and collaboration.
Example:	Stack Overflow (stackoverflow.com) is an online community of practice in which software development learners share knowledge and teach how to solve problems of collective interest.
Indicator:	I06. Educational software as enabler to communication between individuals.
Description:	Enable the student to interact and exchange information synchronously (e.g. chat, video conference, etc.) and asynchronous (e.g. discussion forum, e-mail, etc.).
Skill:	Communication.
Example:	Google Classroom (classroom.google.com) provides an environment where students can communicate with each other or with the mediator in different ways, such as group conversations, direct messages, discussion boards, etc.

Continued on next page

Table 2 – continued from previous page

Indicator:	I07. Educational software as enabler to creative demonstration of learning.
Description:	Enable the student to demonstrate acquired and internalized knowledge from creative way (e.g. telling a story).
Skill:	Creativity and communication.
Example:	ClassDojo (www.classdojo.com) allows students to share academic activities outcomes of different ways, such as drawings, photos, videos and notes.
Indicator:	I08. Educational software as enabler to problem solving.
Description:	Enable the student to use divergent thinking to meet different alternatives/ideas for solving a problem and then use convergent thinking to make the best choice from alternatives/ideas met.
Skill:	Collaboration, creativity, analytical thinking and critical thinking.
Example:	Lightbot (lightbot.com) is a game that allows the students to develop different programming logic to reach the goal and then choose the most efficient one.
Indicator:	I09. Educational software as enabler to knowledge transformation.
Description:	Enable the student to externalize (make explicit) knowledge acquired or recombine (synthesize) different pieces of knowledge in a new form.
Skill:	Communication, knowledge transformation and analytical thinking.
Example:	MindMeister (mindmeister.com) is a graphical tool that allows students to synthesize the knowledge acquired at different levels of abstraction through a Mind Map.

3.4. *Development of the Web-Based System*

The web-based system was developed to support DEISE approach. The Laravel framework (`www.laravel.com`), based on programming language PHP, was used to web-based system development. The Laravel framework provides dependencies management, integration with diverse databases, as well as utilities and components to allow modular development of a web-based system.

The design pattern MVC (Model-View-Controller) (Verma, 2014) extended with layer Route (Coleman, 2016) was used to drive logical architecture of the web-based system developed. The design pattern MVC aims to separate data presentation from manipulation of this data by system business rules (Sommerville, 2011). The layer Route is responsible for managing the various predefined Uniform Resource Locator (URL) that can be requested by the user through a browser. From the user's request, a specific route is triggered to control calls to the elements of the other layers so that the appropriate data be presented to users through a web page.

Fig. 2 shows a UML Sequence Diagram for educational software evaluation scenario through interaction between logical view classes from user request to gauge chart returned which presents innovation index of the educational software evaluated.

The web-based system to support DEISE approach can be accessed through URL www.abordagemdeise.com/en. Fig. 3 presents user interface (UI) of the web-based system which show indicators of innovation. Fig. 4 presents UI to evaluate educational software from indicators of innovation. First, a user selects educational software (for example, MOODLE) and then, for each indicator, he/she assigns one of the following values: 1 (no meet); 2 (don't know); 3 (partially meet); 4 (fully meet).

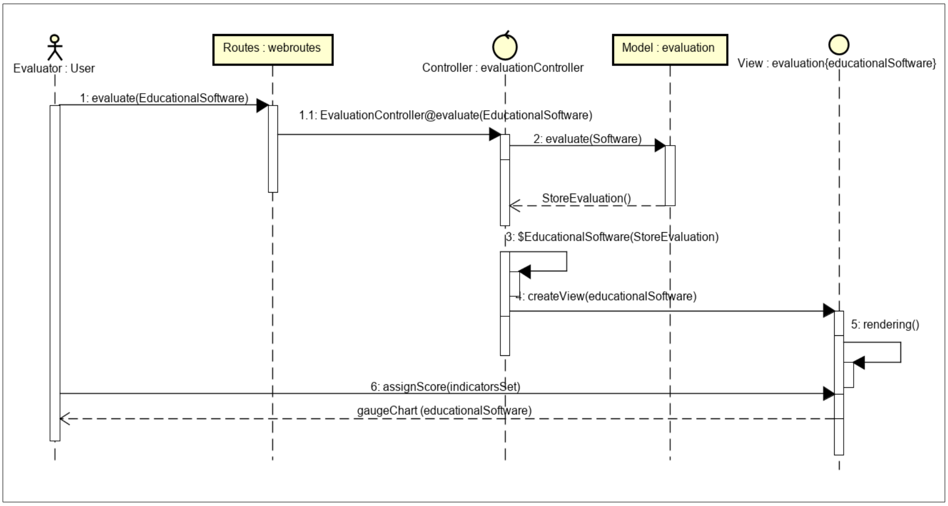

Fig. 2. Dynamic scenario for educational software evaluation.

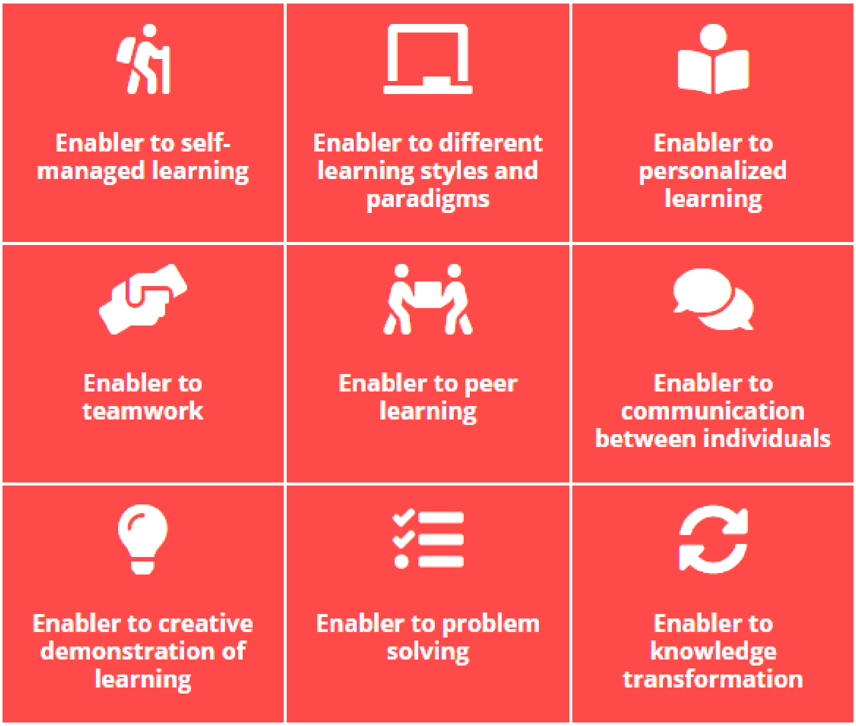

Fig. 3. User interface of the indicators of innovation.

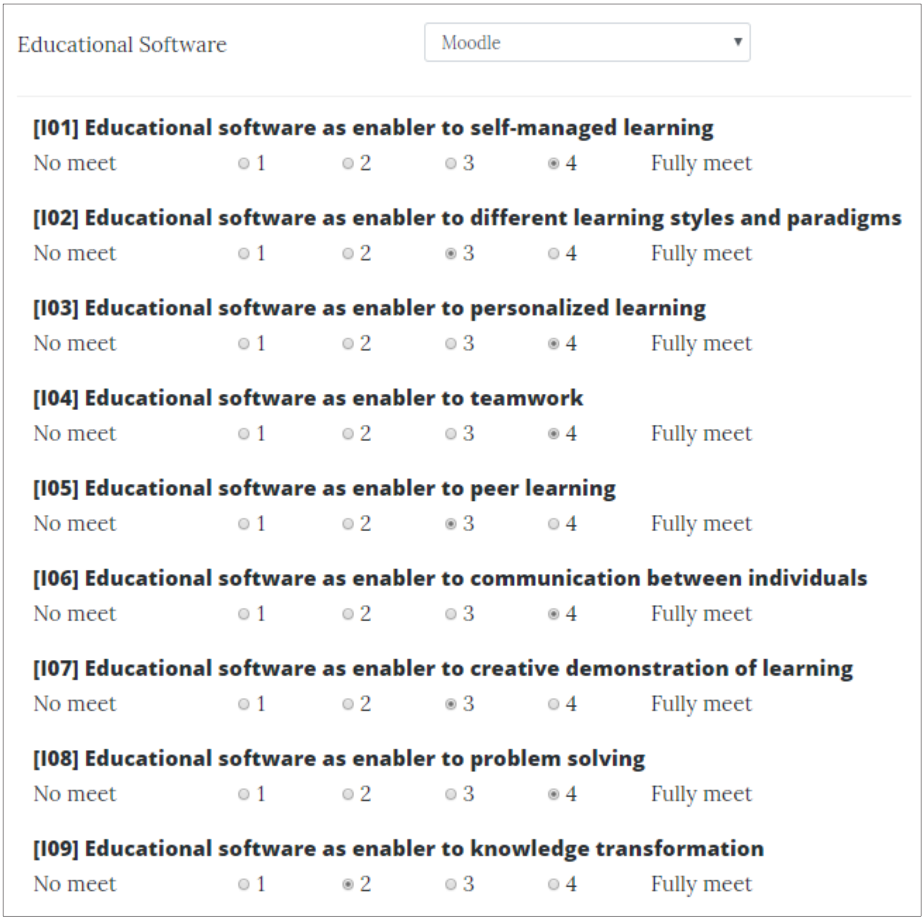

Fig. 4. User interface of the educational software evaluation.

3.5. *Evaluation of Educational Software*

The evaluation of educational software was a stage in the DEISE approach development process. Fig. 5 shows the workflow, using the BPMN notation (OMG & Notation, 2008), carried out to evaluate educational software. A workflow represents the sequence of activities (steps) performed over a period to achieve a goal or work.

There are two roles responsible for carrying out the activities, grouped by different swimlanes: (1) Author (of the work), responsible for the planning and preparation of the environment for the educational software evaluation, as well as for analysing the results of the evaluations and adjusting the web-based system developed from evaluators' feedback; and (2) Evaluator, responsible for evaluation from one or more educational software through web-based system, as well as providing (optionally) feedback on the evaluation made.

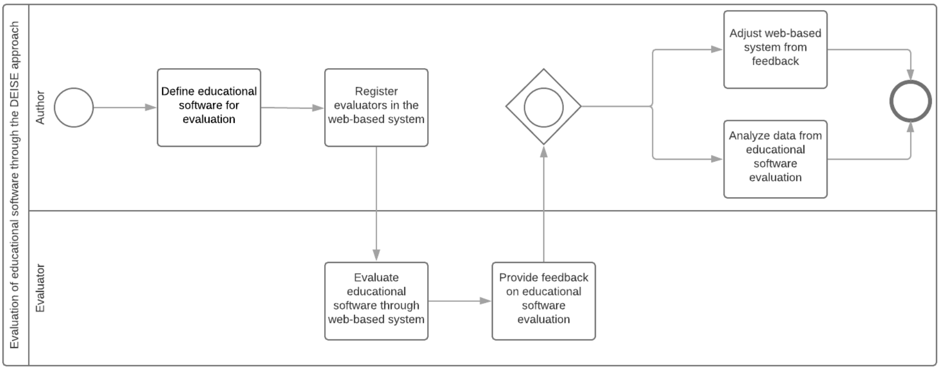

Fig. 5. Workflow for evaluation of educational software.

The definition of the educational software to be evaluated was based on a survey whose participants were educators with experience in using educational software in the teaching process. The educational software most cited for they were: Scratch (Resnick *et al.*, 2009); App Inventor (Wolber, Abelson, Spertus, & Looney, 2011); MOODLE (Moodle, 2018); Google Classroom (Iftakhar, 2016); Google Suite (Kakoulli-Constantinou, 2018); and Lightbot (Kazandzhy, 2017).

The evaluators are teachers of different grade level with classroom experience in using at least one of the defined educational software for evaluation. The evaluators (29 in total) were registered in the web-based system. The author sent for evaluators an e-mail contained respective access credentials and general instructions for educational software evaluation. Only 19 evaluators performed at least one educational software evaluation through web-based system of the DEISE approach. The amount of evaluation per educational software was: Scratch – 16; MOODLE – 07; Google Suite – 07; App Inventor – 05; Google Classroom – 03; and Lightbot – 02.

Some evaluators provided feedback on educational software evaluation through web-based system of the DEISE approach. From evaluators' feedback, the author has made the following adjustments to the web-based system usability: changed the colours of the gauge chart showing the innovation index to education for evaluated educational software; on the educational software evaluation user interface, added a hyperlink to web page which presents the set of indicators of innovation; and on the drivers of innovation web page, inserted caption to differentiate technical and pedagogical drivers.

The last activity performed consisted of analyze data from educational software evaluation. The results will be presented in next section.

4. Results

As previously described (Section 3.4), an educational software evaluation through web-based system of the DEISE approach (Fig. 3) consists of to assign a score from 1 to 4 for each indicator of innovation (Table 2). The assigned score (AS) can be: 1 – educational software doesn't meet the indicator; 2 – evaluator doesn't know whether educational software meets the indicator; 3 – educational software partially meets the indicator; or 4 – educational software fully meets the indicator.

For assigned score (AS) is associated an assigned value (AV) as following: If AS = 1 or 2 then AV = 0; If AS = 3 then AV = 0.5; If AS = 4 then AV = 1. The Innovation Index to Education (I2E) is average of the assigned values (AV). For example, educational software MOODLE was evaluated by seven evaluators. Table 3 presents the assigned score (AS) for each indicator from I01 to I09, informed by each evaluator (EV) from EV1 to EV7, and respective assigned value (AV). The Innovation Index to Education (I2E) is presented by indicator and for the educational software in the last two rows of the Table 3, respectively. I2E per indicator is average of the AV. I2E for educational software is average of the I2E per indicator. The educational software MOODLE obtained the following I2E per indicator on a scale from 0 to 1:

- I01. Educational software as enabler to self-managed learning: 0.86 (86%)
- I02. Educational software as enabler to different learning styles and paradigms: 0.71 (71%);
- I03. Educational software as enabler to personalized learning: 0.71 (71%);
- I04. Educational software as enabler to teamwork: 0.79 (79%);
- I05. Educational software as enabler to peer learning: 0.64 (64%);
- I06. Educational software as enabler to communication between individuals: 0.86 (86%);
- I07. Educational software as enabler to creative demonstration of learning: 0.57 (57%);
- I08. Educational software as enabler to problem solving: 0.64 (64%);
- I09. Educational software as enabler to knowledge transformation: 0.71 (71%).

The educational software MOODLE was better evaluated in the indicators I01 and I06 and worst evaluated in the indicator I07. In general, educational software MOODLE

Table 3

MOODLE Software Evaluation Data

Ind. / EV	[I01]		[I02]		[I03]		[I04]		[I05]		[I06]		[I07]		[I08]		[I09]	
	AV	AS	AV	AS	AV	AS	AV	AS	AV	AS	AV	AS	AV	AS	AV	AS	AV	AS
EV1	4	1	3	0.5	2	0	2	0	3	0.5	3	0.5	3	0.5	3	0.5	3	0.5
EV2	4	1	3	0.5	4	1	4	1	4	1	4	1	4	1	4	1	3	0.5
EV3	1	0	3	0.5	3	0.5	3	0.5	3	0.5	3	0.5	1	0	1	0	3	0.5
EV4	4	1	4	1	4	1	4	1	4	1	4	1	4	1	4	1	4	1
EV5	4	1	4	1	4	1	4	1	3	0.5	4	1	2	0	3	0.5	4	1
EV6	4	1	4	1	4	1	4	1	4	1	4	1	4	1	4	1	4	1
EV7	4	1	3	0.5	3	0.5	4	1	2	0	4	1	3	0.5	3	0.5	3	0.5
I2E by indicator		**0.86 86%**		**0.71 71%**		**0.71 71%**		**0.79 79%**		**0.64 64%**		**0.86 86%**		**0.57 57%**		**0.64 64%**		**0.71 71%**
I2E for Educational Software	**0.72 (72%)**																	

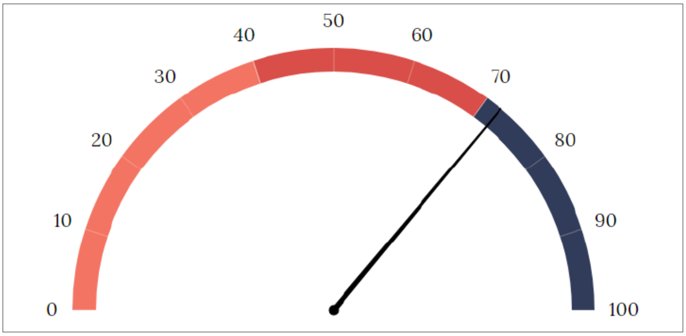

Fig. 6. Gauge chart for I2E of the educational software MOODLE.

obtained I2E 0.72 (72%). Fig. 6 shows I2E to educational software MOODLE in gauge chart format, as presented in the web-based system of the DEISE approach.

Educational software of similar purpose can be compared from evaluation carried out by educators (evaluators) through web-based system of the DEISE approach. For example, Scratch and App Inventor are educational software whose purpose is the programming learning through blocks connected to each other like a puzzle, rather than command lines, which makes it more intuitive for the student to think about the algorithm and minimizes the syntactic complexity of programming language.

Fig. 7 presents in radar chart format the Innovation Index to Education (I2E) per indicator (from I01 to I09) of the educational software Scratch and App Inventor. Looking

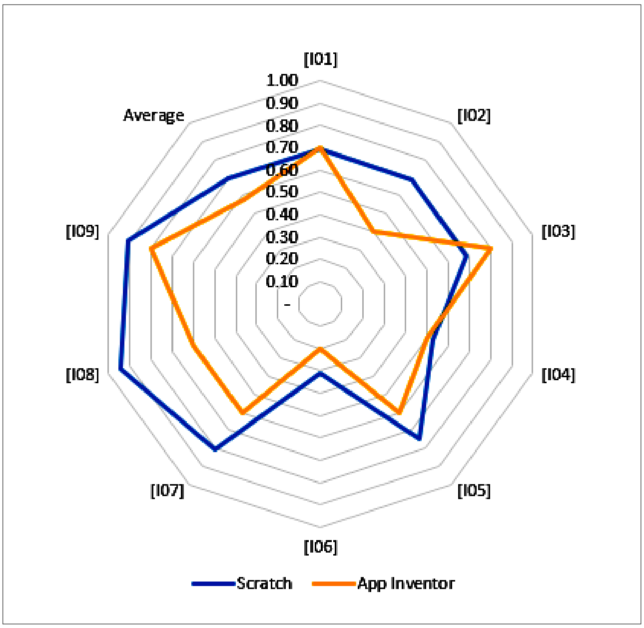

Fig. 7. Radar chart comparative between Scratch and App Inventor.

at Fig. 7, it's possible to see that both value curves has similar format, but Scratch value curve is better than App Inventor value curve. Value curve is drawing by link between the points on the radar chart that represent the I2E value per indicator (from I01 to I09). Value curve paints a picture of the level how educational software enables students with some skills needed for 21st century learning and working.

6. Discussion

From academic point of view, the DEISE approach contributes to the area of innovation in education by presenting a new and differentiated means of evaluating educational software based on enabling students with skills needed for 21st century learning rather than focusing on technical aspects. From technical point of view, the DEISE approach can be used: by educators to evaluate educational software so that the repository of evaluated educational software be evolved and expanded; by teachers and educational institutions to assist in making a decision on which educational software to adopt; and by individual developers or companies as guideline to the educational software development process through the specified drivers and indicators, as well as to obtain innovation index of education of the educational software developed.

The main research outcomes are:

- Indicators of innovation to measure educational software as enabler to: self-managed learning, different learning styles and paradigms, personalized learning, teamwork, peer learning, communication between individuals, creative demonstration of learning, problem solving, and knowledge transformation. These indicators are associated with the following 21st century learning skills: autonomy, adaptability, analytical thinking, collaboration, communication, creativity, and critical thinking.
- Drivers of innovation, grouped in the technical and pedagogical categories, met from a literature systematic mapping. These drivers can be used as guideline to educational software development process.
- Web-based system used to evaluate educational software from indicators specified. From evaluation carried out is calculated the innovation index to education (I2E) per indicator and for educational software.
- Repository of evaluated educational software used to: compare educational software of similar purpose; and support decision-making about educational software choice.

The main limitations of this research work are: drivers of innovation has not been used or evaluated in the context of a real educational software development project; few educational software evaluations carried out by few educators (19) and for only 06 educational software; the DEISE approach has not been used by teachers, educational institutions and educational software industry aiming value delivery planned. In addition to meeting these work limitations, future work opportunities include: designing and deploying a business model based on the DEISE approach; and apply statistical treatment of educational software evaluation data to disregard possible outliers.

References

Ananiadou, K., & Claro, M. (2009). *21st century skills and competences for new millennium learners in OECD countries*.

Avvisati, F., Hennessy, S., Kozma, R. B., & Vincent-Lancrin, S. (2013). *Review of the Italian Strategy for Digital Schools*.

Barbosa, R., & Souza, R. (2018). Um Levantamento dos Determinantes de Inovação em Softwares Educacionais. *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática Na Educação-SBIE)*, 29(1), 437.

Berbel, N. A. N. (1998). A problematização e a aprendizagem baseada em problemas: diferentes termos ou diferentes caminhos? *Interface-Comunicação, Saúde, Educação*, 2, 139–154.

Bergmann, J., Sams, A. (2016). Sala de aula invertida: uma metodologia ativa de aprendizagem. *Rio de Janeiro: LTC*.

Booyens, I., Molotja, N., Phiri, M.Z. (2013). Innovation in high-technology SMMEs: the case of the new media sector in Cape Town. *Urban Forum*, 24(2), 289–306.

Coleman, A. (2016). Playing with legos: the inner-working of a model-view-controller (MVC) web application. Retrieved February 2, 2019, from Self-Taught Coders website: `https://selftaughtcoders.com/model-view-controller-mvc-web-application`

Darling-Hammond, L., Barron, B., Pearson, P.D., Schoenfeld, A.H., Stage, E.K., Zimmerman, T.D., … Tilson, J.L. (2015). *Powerful Learning: What we Know about Teaching for Understanding*. John Wiley & Sons.

Delors, J. (1996). Education: the necessary utopia. *Learning: The Treasure Within*, 13–35.

Development, O. for E. C. and. (2010). *Inspired by Technology, Driven by Pedagogy: A Systemic Approach to Technology-Based School Innovations*. OECD Publishing.

Fullan, M. (2007). *Leading in a Culture of Change*. John Wiley & Sons.

Hosie, P., Schibeci, R., & Backhaus, A. (2005). A framework and checklists for evaluating online learning in higher education. *Assessment & Evaluation in Higher Education*, 30(5), 539–553.

Iftakhar, S. (2016). Google classroom: what works and how? *Journal of Education and Social Sciences*, 3(1), 12–18.

Istance, D., & Kools, M. (2013). OECD work on technology and education: Innovative learning environments as an integrating framework. *European Journal of Education*, 48(1), 43–57.

Kakoulli-Constantinou, E. (2018). Teaching in Clouds: Using the G Suite for Education for the Delivery of Two EAP Courses. *Journal of Teaching English for Specific and Academic Purposes*.

Kazandzhy, L. (2017). Lightbot: Code Hour. *Children's Book and Media Review*, 38(1), 22.

Looney, J. W. (2009). *Assessment and Innovation in Education*.

Luna Scott, C. (2015a). *The Futures of Learning 2: What Kind of Learning for the 21st Century?*

Luna Scott, C. (2015b). *The Futures of Learning 3: what kind of pedagogies for the 21st century?*

Merriam-Webster. (2016). *Merriam-Webster Dictionary*. Retrieved February 12, 2016, from `https://www.merriamwebster.com/dictionary/process`

Moodle, H. Q. (2018). Moodle. *West Perth, Australia: Moodle HQ*.

Oblinger, D. (2004). The next generation of educational engagement. *Journal of Interactive Media in Education*, 2004(1).

OECD. (2014). *Measuring Innovation in Education: A New Perspective, Educational Research and Innovation*. `https://doi.org/10.1787/9789264215696-en`

OECD. (2017). *Future of Work and Skills*. 24. Retrieved from `https://www.oecd.org/els/emp/wcms_556984.pdf`

OMG, S., & Notation, O. (2008). Software & systems process engineering meta-model specification. *OMG Std., Rev*, 2, 18–71.

P21, P. for 21st C. S. (2011). *Framework for 21st Century Learning*.

Partovi, H. (2018). Why schools should teach the curriculum of the future, not the past. Retrieved May 10, 2019, from World Economic Forum website: `https://www.weforum.org/agenda/2018/09/why-schools-should-teach-the-curriculum-of-the-future-not-the-past`

Peña-López, I. (2017). *The OECD Handbook for Innovative Learning Environments*.

Prince, M. (2004). Does active learning work? A review of the research. *Journal of Engineering Education*, 93(3), 223–231.

Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., … Silverman, B. (2009). Scratch: Programming for all. *Commun. Acm*, 52(11), 60–67.

Saavedra, A. R., & Opfer, V. D. (2012). Teaching and learning 21st century skills: Lessons from the learning sciences. *Retrieved Agust*, 1, 2016.

Sommerville, I. (2011). Software engineering 9th Edition.

Valiente, O. (2010). 1-1 in education: Current practice. *International Comparative Research Evidence And*.

Veraszto, E. V., do Amaral, S. F., & Barreto, G. (2013). Ambientes Tecnológicos Interativos e Educação: Uma Proposta para Inovação Tecnológica. *Revista InovaEduc*, *1*, 1.

Verma, A. (2014). MVC Architecture: A Comparative Study Between Ruby on Rails and Laravel. *Indian Journal of Computer Science and Engineering (IJCSE)*, 5(5), 196–198.

Wolber, D., Abelson, H., Spertus, E., & Looney, L. (2011). *App Inventor*. " O'Reilly Media, Inc."

R. Barbosa is the COO at Prozital, professor at the ELO University. He received the MS from the Federal Rural University of Pernambuco (UFRPE), Graduate Program of Applied Informatics (PPGIA) in Brazil. His research areas of interest are Innovation in Education, Educational Games, Agile and Project Management. In his free time, he likes to play DotA 2.

R. Souza is an associate professor at the Department of Computer Science (DC) and permanent member at the Graduate Program of Applied Informatics (PPGIA) at Federal Rural University of Pernambuco (UFRPE) in Brazil. He received his MS and PhD from the Federal University of Pernambuco (UFPE), Informatics Center (CIN) in Brazil. His research areas of interest are Innovation in Education, Digital Social Innovation, Digital Transformation, Digital Governance, Knowledge Management and Software Product Management.