MASTER EUROPÉEN ERASMUS MUNDUS EN INGÉNIERIE DES MÉDIAS POUR L'ÉDUCATION (EUROMIME)
MASTER EUROPEO ERASMUS MUNDUS EN INGENIERÍA DE MEDIOS PARA LA EDUCACIÓN (EUROMIME)
MESTRADO EUROPEO ERASMUS MUNDUS EM ENGENHARIA DE MIDIAS APLICADO À EDUCAÇAO (EUROMIME)

Master Ingénierie des Médias pour l'Éducation
Faculté des Lettres et Langues
Université de Poitiers, France

Máster Universitario en Estrategias y Tecnologías para la Función Docente en la Sociedad Multicultural
Facultad de Educación
Universidad Nacional de Educación a Distancia, España

Mestrado Ciências da Educação
Faculdade de Motricidade Humana
Universidade de Lisboa, Portugal

PETESE – Pedagogical Ergonomic Tool of Educational Software Evaluation: An anasynthesis applied to the case study of GGBook

PETESE – Uma ferramenta para validação ergonômica e pedagógica de *softwares* educativos:
a metodologia *anasynthesis* aplicada ao estudo do caso do GGBook
PETESE – Une outil d'évaluation ergonomique et éducatif pour les *softwares* éducatifs:
une méthodologie *anasynthesis* appliquée à l'étude de cas de GGBook
PETESE - Una herramienta para la evaluación ergonómica y educativa para *softwares* educativos:
una metodología *anasynthesis* aplicada al estudio del caso GGBook

Stéphanie Coomans

Prof. Gilberto Lacerda Santos

06.2015

Foreword

This master thesis is the end of my European Master in Media Engineering for Education and closes as well my student academical course. The mix of my bachelor in social communication, my passion of design and the learnings about the educational field, have give birth to this thesis.

Indeed, ergonomics is already present since a long time in my communication and designing processes and this thesis gave me the opportunity to debthen this concept and integrate it in the educational field and its young appearance of educational software.

The elaboration of this thesis was intense and I thank my family and my friends for the transatlantic energy they brought me. I thank in particularly my promoter Prof. Dr. Gilberto Lacerda Santos for its necessary guidance, support and tips as well as its encouragements to discover the academical world through scientific papers and international congresses. I would also like to thank the members of the Abaco Laboratorio of the UnB for the uncountable discussions, interesting exchanges, always welcoming collaboration and their presence in moments of doubts. Finally, I thank the experts and the professors that made this research possible by answering my questions and giving suggestions.

ABSTRACT (EN)

Author: Stéphanie Coomans
Title: PETESE – Pedagogical Ergonomic Tool of Educational Software Evaluation: An anasynthesis applied to the case study of GGBook
University: Euromime - Université de Poitiers, France - Universidade de Lisboa, Portugal - Universidad Nacional de Educación a Distancia, España
Promoter: Prof. Dr. Gilberto Lacerda Santos
Year: 2013 - 2015

In the last few years, education and technology combine and created growing opportunities for teaching and learning, among them educational software. However, some educational software on the market are badly organized and contain ergonomical errors, who interfere in the teaching-learning process. Therefore, pedagogical usability is from huge importance during the development of new educational software, before their launching on the market. In response to the existant rarely complete, badly presented, uneasy to use, abstract and unspecific checklists and guidelines, this study proposes a pedagogical ergonomical tool of educational software evaluation (PETESE). This tool is constructed based on an anasynthesis methodology with literature criteria in the field of ergonomics, education and mathematics. In a second part, the PETESE is applied to the educational software GGBook, a numeric book developed by the Abaco's lab (University of Brasilia) based on the GeoGebra environment and integrating facilities between the graphics and the operations elements. In this case study, the development tool as well as future-user analyse the software based on the PETESE. The results of this research show the importance of a pedagogical ergonomical evaluation that raises important elements forewards and possible improvements before the educational software enters the schools and homes.

Keywords: PETESE, educational software, pedagogical usability, predictive evaluation, GGBook.

ABSTRACT (PT)

Autor: Stéphanie Coomans
Titulo: PETESE – Uma ferramenta para validação ergonômica e pedagógica de *softwares* educativos: a metodologia *anasynthesis* aplicada ao estudo do caso do GGBook
Universidade: Euromime - Université de Poitiers, France - Universidade de Lisboa, Portugal - Universidad Nacional de Educación a Distancia, España
Diretor de pesquisa: Prof. Dr. Gilberto Lacerda Santos
Ano: 2013 - 2015

Nos últimos anos, surgiram através da combinação entre a educação e a tecnologia novas oportunidades de ensino e aprendizagem, no qual podemos destacar os *softwares* educativos. No entanto, alguns *softwares* educacionais disponíveis no mercado são mal organizados e contêm erros ergonômicos que podem prejudicar ou até mesmo inviabilizar um processo de ensino-aprendizagem. Neste sentido, a usabilidade pedagógica é de grande importância durante o processo de desenvolvimento de *softwares* educativos, antes que estes sejam lançados oficialmente no mercado. Em contraponto aos *checklists* existentes, que são raramente completos, mal apresentados, difíceis de usar e pouco objetivos, este estudo propõe uma nova ferramenta de avaliação à ergonomia pedagógica de *softwares* educativos, o PETESE. Essa ferramenta foi idealizada com base na metodologia de *anasynthesis* e concorda com os mais recentes critérios estabelecidos pela literatura nas áreas da ergonomia, educação e matemática. Para fins deste estudo, o PETESE foi aplicado ao software educativo GGBook, que trata-se de um software de apoio ao ensino de Matemática que integra um ambiente de texto com o ambiente gráfico do software mundialmente conhecido GeoGebra, de forma a obter funcionalidades de um livro de matemática digital e dinâmico. Para analisar os resultados, optou-se pela abordagem qualitativa do estudo de caso, no qual o PETESE foi avaliado pela equipe técnica do *software* e também pelos seus futuros usuários. Os resultados obtidos por meio desse estudo, nos confirmaram a grande importância da avaliação ergonômica pedagógica, visto que ela revelou elementos importantes de melhoria para o *software* educativo GGBook, que se executados antes de seu uso em sala de aula, pode aumentar suas chances de sucesso.

Palavras chaves: PETESE, *software* educativo, ergonomia pedagógica, evaluação preditiva, GGBook.

ABSTRACT (FR)

Auteur: Stéphanie Coomans
Titre: PETESE – Une outil d'évaluation ergonomique et éducatif pour les *software* éducatifs: une méthodologie *anasynthesis* appliquée à l'étude de cas de GGBook
Universités: Euromime - Université de Poitiers, France - Universidade de Lisboa, Portugal - Universidad Nacional de Educación a Distancia, España
Promoteur: Prof. Dr. Gilberto Lacerda Santos
Année: 2013 - 2015

Depuis ces dernières années, le domaine de l'éducation intègre les ressources technologiques, ce qui donne lieux à de opportunités grandissantes de nouvelles techniques d'enseignement et d'apprentissages dont les *software* educatifs. Cependant, bon nombre d'entre eux, sur le marché actuel, contiennent des problèmes ergonomiques qui interfèrent dans le processus d'apprentissage. Pour cela, l'ergonomie pédagogique est de grande importance lors du développement de nouveaux *software*. En réponse aux méthodes actuelles d'évaluation ergonomiques peu complètes, difficile d'utilisation et peu concrètes, cette étude propose un nouvel outil d'évaluation d'ergonomie pédagogique de software educatif (PETESE). Cet outil a été construit à partir de la methodologie anasynthèse basée sur des critères de literature du champs de l'éducation, de l'ergonomie et des mathematiques. Ensuite, ce PETESE a été appliqué à un software educatif précis, le GGBook. Ce livre numérique développé par le laboratoire Abaco (université de Brasilia) intégrant l'environnement de GeoGebra en facilitant l'outil texte et les élements d'opérations. A travers cette étude de cas, le PETESE est utilisé tant par les membres de l'équipe de développement que par les futurs utilisateurs. Les résultats de cette recherche confirment l'importance de l'évaluation d'ergonomie pédagogique qui souligne des possibles améliorations ainsi que des élements à résoudre avant le lancement du *software*.

Mots clés: PETESE, *software* educatifs, ergonomie pédagogique, evaluation prédictive, GGBook.

ABSTRACT (ES)

Autor: Stéphanie Coomans
Título: PETESE - Una herramienta para la evaluación ergonómica y educativa para *softwares* educativos: una metodología *anasynthesis* aplicada al estudio del caso GGBook
Universidades: Euromime - Université de Poitiers, France - Universidade de Lisboa, Portugal - Universidad Nacional de Educación a Distancia, España
Promotor: Prof. Dr. Gilberto Lacerda Santos
Año: 2013 - 2015

Desde hace unos años, el área de la educación está integrando, en sus prácticas, diversos recursos tecnológicos. Esto da lugar a oportunidades crecientes de nuevas técnicas para la enseñanza y el aprendizaje, entre las cuales se destacan los *software* educativos. Sin embargo, muchos de estos *software*, en el mercado actual, conllevan problemas ergonómicos que pueden interferir en el proceso de aprendizaje. Por eso, la ergonomía pedagógica es de gran importancia en el desarrollo de nuevos *software*. En respuesta a los métodos actuales de evaluación ergonómica, poco completos, difíciles de utilización y poco concretos, este estudio propone una nueva herramienta de evaluación de ergonomía pedagógica de *software* educativo, el PETESE. Primero, construimos la herramienta, siguiendo la metodología *Anasyntheses*, basada en criterios de la literatura del área de la educación, de la ergonomía y de las matemáticas. Luego, el PETESE fue aplicado a un *software* educativo específico, el GGBOOK. Este libro digital, desarrollado por un equipo del laboratorio Ábaco (Universidad de Brasilia), integra el ambiente del programa GeoGebra, reuniendo la herramienta texto y los elementos de operaciones. En este estudio de caso, el PETESE fue usado tanto por los miembros del equipo de desarrollo, como por los futuros utilizadores. Los resultados de esta investigación confirman la importancia de la evaluación ergonómica pedagógica, la cual subraya posibles mejorías así como problemas a ser resueltos en el GGBook antes de su difusión.

Palabras clave: PETESE, *softwares* educativo, ergonomía educativa, evaluación predictiva, GGBook.

TABLE OF CONTENT

LIST OF FIGURES & TABLES

List of figures

List of tables

1 Presentation Of The Research

Summary

This section presents our research. It starts by a general introduction presenting our theme, our objectives, our methodology as well as the structure of our work and its relevance. In a second time, we present our general theoretical context, and finish on the presentation of our objectives.

General Introduction

Situation of our work

This research is part of the European Master of Media Engineering for Education (EUROMIME). This two year training master takes place in the University of Poitiers (France), the University of Lisboa (Portugal) and the University of Distance Education in Madrid (Spain). This program, part of the Erasmus Mundus, gives on one hand, keys to design, develop and implement new technologies in the field of education, and on the other hand, it trains researchers specializing the study of the use of these technologies in the educational field.

This research work integrates itself in the research dimension of Euromime. Indeed, the aim is to join a support laboratory of the Master in Latin America where the thesis advisor works. For this reason, this thesis work is involved in the on-going research laboratory Abaco of the Faculty of Education of the University of Brasilia (Brazil) and supervised by the Professor Gilberto Lacerda Santos.

The objectives of the research are to develop the research competences and the personal learning of the student. This will be evaluated by three jury members, each one of a different organizational university, based on a checklist of criteria for each part of the thesis. The final note counts for 34 ECTS.

Theme and objective

This work is situated in the actual theme of education and technology, and more specifically in the area of educational software. Our general questioning is the following: “What can be done in educational software to fulfill the educational objectives while preserving the quality patterns of the software?”. It joins thus the area of technology and education, on which we add the area of mathematics, the field of the educational software we integrate our work in, GGBook.

The objective of this research is double. On one hand, the aim is to develop a tool of evaluation of educational software in order to respond to the needs described in the general context above. On the other hand, the objective is to apply this tool in the specific case of the educational software GGBook in order to improve this software.

This research leads thus to two different approaches: 1/ a theoretical approach that leads to the development of an evaluation tool, through the method of anasynthesis; 2/ an empirical research that integrates that tool and experiences it through the case study of a specific educational software, GGBook. This is illustrated in the following figure:

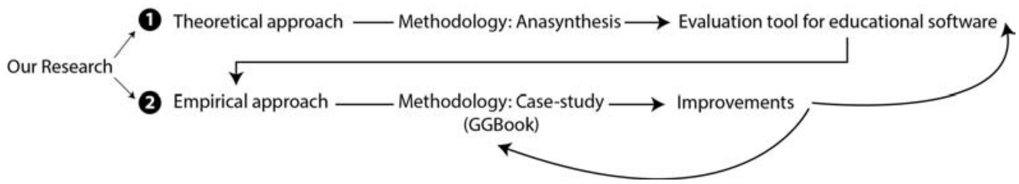


Figure 1 : Our research objectives

Our more specific objectives are detailed on p.23

Methodology

The methodology used in our work is double. The theoretical part is based on the method called anasynthesis that allows establishing a structured and rigorous framework through an analysis of theory regarding the field of education, mathematics and ergonomics. Such a method is constructed in four stages: the identification, the analysis, the syntheses and the validation.

This theoretical analyzing resulting in our referential for pedagogical usability of mathematical software of discovering learning is applied in the second part of our research. Indeed, the empirical part is a case study and the application of our theoretical analyses in practice. It will as well pursue other objectives like thinking critically about the referential and optimize the highlighted problems and propose a concrete application of the referential through the amelioration of the analyzed educational software for mathematics, GGBook. This will be done through an objective description of GGBook's content, an expert's vision through heuristic evaluation, a user's analysis through a questionnaire and an application of the changes.

Our methodology is discussed more in details in our third chapter (p.73).

Structure

The structure of this research will follow the classical order of a thesis. After this introduction part with a contextualization and a clear definition of the objectives, an analysis of state of art allows us to construct the three first stages of the anasynthesis (section 2). It is divided into 4 parts: a part about education that discusses the concept of educational software to discovery learning; a second part about mathematics that defines the important criteria of mathematical software; a third part about engineering and ergonomics developing the quality of software design, the evaluation and the pedagogical usability; finally, part four will present our referential.

This will be followed by the next section that details our methodology (section 3). Based on a anasynthesis and a case study, our methodology will be detailed in this section, through a definition of the methodologies, their process, the data collecting and reporting.

In the fourth part, the research gets completed with an empirical research based on the education software of mathematics, GGBook. The structure is the same as the case study: a description of GGBook, an expert's vision, a user's vision, and an application of the changes.

In conclusion, the fifth section will summarize the major findings of this research by responding to the research questions. Some attention will also be given to the limits of this study and some possible future questions.

Research Context

General context

The third industrial revolution, or technological revolution, impacted the lives and the world population allowing a globalization of information (Rifkin, 2011). This leads to the rise of activities that use technology and opened new horizons in different fields. It is the case of education and technology that combined and created growing opportunities for teaching and learning. This has influenced a new culture of learning with new ways to construct the knowledge.

Nowadays, the availability of the software grows while the cost of the hardware decreases and promotes access. Consequentially, the society raises the necessity of new ways of learning where the TICs provide more creativity and motivation that allow the construction of knowledge and learning in a more significant way. Like Frey and Sutton say, “the multimedia learning allows for a level of interactivity that can enhance both the online text and traditional lecture-based classroom environments, and is growing at all levels of education” (2010, p. 491). It also allows the user to learn at its own rhythm based on its abilities. The getting and giving of information is thus transformed by both the way the media can be integrated into the system as well as the interactive aspects of that system.

The new technologies of information and communication affect directly the teaching methods, practices and resources; and studies have indicated that the work with the new technologies is not as simple as imagined, requiring preparation, basic skills and material available (Libâneo, 2007). This has created big challenges to traditional teaching. While some teachers are reluctant to take the challenge, others make use of computer technology to enhance teaching and learning. The educational software, however, overlap the books and allow the inclusion of videos and animations that satisfy the self-teaching and the curiosity, while the books can be there as support to the learner. It is thus important to join the field of education (content) with the field of technology (means) to fulfill both needs as good as possible.

Among the opportunities of the use of computers in the area of education, the educational softwares have appeared massively on the market. They differ from the other types of software for their clear emphasis on human learning and knowledge acquisition (De Diana & Van Schaik, 1993).

The educational softwares have the opportunity to integrate multimedia and interaction as well as for the students as for the teachers. The educational practices should also emphasize the creation of environments learning, in which students build their knowledge and the teacher start to be the one who teaches and becomes one who guides and encourages the process. In other words, beside the facilitation of students' learning, the educational software also are part of the teachers' work and parents' educational choices. It is thus from high importance to analyze the resources that are brought through these new technologies to capture, treat, organize, systemize, conserve and transmit the information according to the intrinsic pedagogical objectives.

However, "a large part of those educational software have been badly organized and poorly documented" (Garcia, Garcia, & Pacheco, 2009, p. 93). Indeed, developed tools that define and implement educational objectives while preserving technical quality patterns are rare. Silva and Vargas (1999) explain this by the difficulty of the mix of the different areas of ergonomics, TICS, pedagogy and psychology. Yet, a good evaluation of the system is essential to see its quality and its effects before applying it to the classroom or making its entry in families.

From a historical point of view, the need to measure the quality of educational software is issued from two older research themes. From one side the evaluation of the teaching materials like the school textbooks; and from the other side, the evaluation of software and human/machine interfaces (Crozat, Trigano, & Hû, 1999). For both of those areas, formal institutions or even government have defined what is effective and what are the criteria of evaluation. However, what concerns educational software, it is less clear (Fino, 2013). This has caused several authors to think about the necessity to define a methodology for its development and its evaluation, using some tools of software engineering in which they include pedagogical aspects.

Every evaluation has to be constituted by three questions: 'What?', 'Who?' and 'When'? Like Heller mentioned: "The challenge is to decide what to evaluate, who will carry out the process and when has it to take place" (Heller, 1991, p. 285).

Concerning the first question it is important to mention that a lot of evaluations on educational software can be found in the literature in the form of guidelines, criteria or checklists. However, Plaza and al. (2009) conclude, after a study of several evaluations models, that there is a need for unifying criteria and standardization that mainly present the same points of attention.

However, a good evaluation takes into account the cognitive aspects as well as the aspects of usability and allows telling if certain aspects are present or not. The pedagogical aspects include an evaluation of convenience and the feasibility of the software in an educational situation based on the specific situations of the process of learning. The technical aspects are based on an evaluation of the usability, in other words, the quality of the interaction user/computer through the interface of the software looking for an efficacy ad an efficiency of the interaction (Atayde, Teixeira, & Pereira, 2003). However, the existing evaluations do not always include the content coherence and the graphic interface; they are mostly oriented in the field of engineering and ergonomics. The aim of this research is thus to integrate the characteristics of usability with those of learning; where the area of ergonomics focuses on 'learning the system to use it' and learning focuses on 'learning through the system'.

The question of 'what' is evaluated depending also of the reason behind the evaluation. Three main reasons can be highlighted according to Puustinen and al. (2006). First, teachers that want to see which is the best software; secondly, to give feedback on the effectiveness of educational software in the process of creation; or thirdly for the reuse of educational software that was not initially planned for it in the context of a redesign.

In the literature, the 'who' evaluates is usually seen as the teachers. Indeed, the models, checklists and criteria of pedagogical usability frequently speak out to them so they can effectively choose and use educational software considering their possibilities and limitations and see if it reflects their instructional practice. The quality of the softwares however, relies on those who create them. With the market concurrence of nowadays and the enthusiasm of the creation, it is sometimes good to remember the indispensable criteria a software should enhance, as well the technical as the educational criteria, that are not always part of the area of the conceptors.

In this same logic, most of the pedagogical evaluations of educational software happen after the release of the product, before its entering in the classroom. However, software has to be evaluated before purchase at least once during the ongoing process of development. Finally, it can be observed that the majority of the studies do not focus on a specific field. It consists more precisely in general important aspects, which make the application not always appropriated.

In this context, applying appropriated guidelines for building usable software goes through the following steps: 1/ a catalogue of all the existing recommendations 2/ an analysis and comparison of recommendations to get a set of agreed recommendations 3/ a classification of the recommendations. This research focuses exactly on this, building a pedagogical usability referential for educational software.

Context of GGBook, Abaco's project

The mathematical field has seen those last years the creation of lots of software and applications. Among them, GeoGebra, "a free and multi-dynamic mathematics software for all levels of education that joins geometry, algebra, tables, graphing, statistics and calculus in one easy-to-use package" (GeoGebra, 2014).

This dynamic software created in 2001 is intended for teachers and students. Both can use GeoGebra to visualize variables, vectors, points, integrals and functions and work with mathematics in an easy way. According to Hohenwarter, the founder of this application, "it is a nice way to play with mathematics and it turns it into something you can grab" (Hohenwarter, 2010). "The unique feature of GeoGebra is the integration of dynamic geometry software and computer algebra system into a single tool for mathematics education" (Manizade & Lundquist, 2009, p. 1567).

Despite their worldwide community, their continuous development and their awards, researchers have observed problems of users handling with GeoGebra. The major problems are the text tool and some equation's functions, which require a good knowledge of LaTeX as well as an important mathematical background and make the software difficult to use for teachers and students (Claudio, De Lima, Ferreira, Lacerda Santos, & Nobriga, 2012).

In reaction to those obstacles, Abaco's lab[1] supported by the Euromime Consortium, develops the GGBook prototype, a new online interface[2] for the GeoGebra environments that integrates text editors and equations more easily through dynamic narration of mathematics, the aim being to construct digital and dynamic books for the learning of mathematics. This software implements features that would facilitate the observed difficulties by the evolution of the environment of the visual view of the text and the accessibility of the LaTeX language.

---

[1] The Abaco Laboratorium is a group of interdisciplinaries researchers on Informatics and education linked to the post graduated program in education of the Universidade de Brasília (UnB).
[2] Avalable on: http://GGBook.com.br

GGBook is thus innovative on several points. First, it integrates different registers of representations of mathematics on a dynamical way. This means that the users can see the relation that exists between the different representations, for example, if something is modified in the register of algebra, it also changes in the register of geometrics. A second characteristic of the program is that text is also dynamic and integrated in the same logic. Finally, the facilitation of history and revision makes it easy to identify the order or to modify it.

The interface of GGBook integrates the interface of GeoGebra. It is divided into two parts: text and graphics. The main elements of this prototype are the following one's:

- The two parts of the interface are dynamically related one to another.
- The toolbar appears dependently from the part in which the user is. If the user is in the text part, the text tool will appear. If the user is in the graphic part, the graphic toolbar will appear.
- The exercises are like pages of a book that allow going from one page to another.

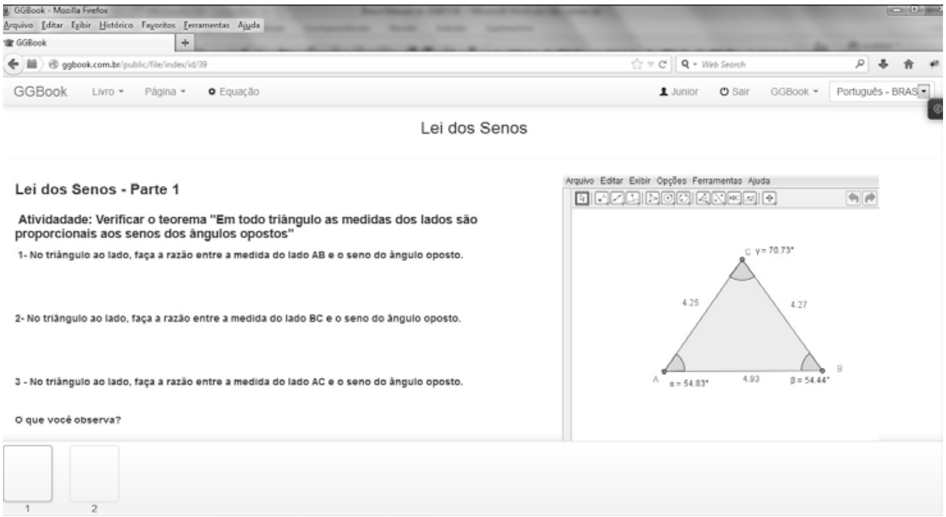


Figure 2 : The interface of GGBook

The targeted public of the software are the professors and the students. Because the others of the research team have worked the part of the professors, it is in that perspective that we follow and also lead our research on the professors. Finally, it is important to mention that this work is a continuity of the PhD of Jorge Nóbriga (2015)[3] as well as the thesis of Bruno Ferreira (2015)[4] and the former Euromime student Valentin Oros[5]. Renan de Lima, actual student of the program will also tackle this software in his future thesis.

[3] Jorge Nóbriga analyzed GGBook as a software that allows the construction of dynamic narrations of mathematics.
[4] Bruno Ferreira integrates a module of gamification into the software GGBook in order to stimulate gamification according to the learning style of the user.
[5] Oros V., Lacerda Santos G. & Nobriga J., GeoGebra in Romanian: the challenges of localising an educational software into a specific socio-cultural context. Geogebra International Journal of Romania, 4(1), 2014.

Relevance of our research

This study contributes primarily to the academic debate by its elaboration of a referential that joins the three fields: education, technology and mathematics. Through the review in the literature of the criteria and characteristics of each field followed by an interaction of it, the referential we propose is quite precisely related to a specific kind of educational software, mainly mathematics of discovery learning. This is rarely the case of the existing one's presenting mostly general guidelines applicable to all areas and all types of learning perspectives.

Moreover, nowadays politically speaking, the question of technology in schools is highly debated. From an economical perspective, the consumerism pushes a high commercialization of materials, whose quality raises questions. Such a referential can help the designers and developers to focus on the important characteristics of an educational software before releasing it on the market.

Indeed, our referential is primary aimed to help the developers and the designers during the creation process of the educational software before its release in the commerce. This is rarely the case in the one's we have founded that focus on the professors, in order to help them in the choice of the election of an educational software for their course.

However, we are conscious of the importance of the professor's role in the educational criteria of the software, as they are part of the target public of most of the educational software (it is the case for GGBook). For this reason, our referential is not only based on the theory, but also tested among professors giving feedback on characteristics of importance for the developers.

Research Questions & Objectives

The main goal of this research is to propose a pedagogical usability referential for mathematical software of discovering learning dedicated to the instructional designers to use during the design process of the software and to apply those criteria to the educational software of mathematics developed by the Abaco team. In this research, we address the following research questions and its related objectives:

RQ1: Which are the elements that have to be taken into account while evaluating ergonomics of mathematical software of discovering learning?

Objectives: → 1. Define important elements for discovering learning
→ 2. Identify characteristics of effective mathematical software
→ 3. State the ergonomics criteria for effective mathematical software

RQ2: What is the contribution in the literature in the field of pedagogical usability evaluation?

Objectives: → 4. Gather the existent pedagogical usability evaluations
→ 5. Retain the pedagogical usability criteria that arise the most

RQ3: How can the above-founded results help the pedagogical usability of the mathematical software, and more specifically the educational software GGBook?

Objectives: → 6. Develop a referential based on the above selected information
→ 7. Application and evaluation of the referential to the practical case of GGBook

The different objectives correspond to the intersections of the three different fields that we are merging, the central common point referring to the development of our referential, like explained in the graphic below.

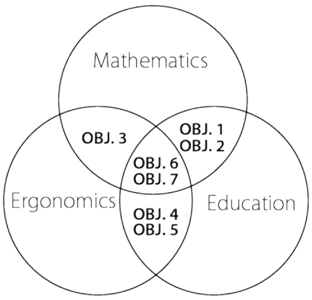


Figure 3 : Intersection of our three research field with our specific objectives

Literature Review 2

Summary

This literature review is divided in four main parts: a part about education that discusses the concept of educational software to discovery learning; a second part about mathematics that defines the important criteria of mathematical software; a third part about engineering and ergonomics developing the quality of software design, the evaluation and the pedagogical usability; finally, part four will present our referential. This corresponds to the three main circles of our Figure 3 : **Intersection of our three research field with our specific objectives**

PART 1: EDUCATION

FROM EDUCATIONAL SOFTWARE TO DISCOVERY LEARNING

Predictive evaluation of effective ***educational software*** *design:*
development of a pedagogical usability referential for mathematical software of ***discovering learning***

Introduction

This part in the area of education will first define our concept of educational software in order to give our definition of educational software as well as its important characteristics. In a second step, we will discuss several typologies of educational software. They give us a better understanding of educational software and allow us to explain our positioning and approach in this complex concept. This will be followed by the learning principles and approaches with a discussion on the behaviorist approach and cognitivist approach. Because our evaluation tool is more specifically aimed to mathematical software of discovering learning, the constructivist approach and the discovery learning will be taken described as well and their criteria highlighted. The cognitive theory of multimedia learning and the cognitive overload will conclude our first part, and bring a support to discovering learning as well as additional criteria and guidelines to take into account. Finally, a conclusion will summarize this first literature part and respond to our first objective: define important elements to discovering learning.

Definition of educational software

Educational software is composed of the following terms: ‘educational’ and ‘software’. If all authors agree on the term ‘software’ as being a component of any computing system that refers to programs that are activated by hardware (Crovello, 1982), the definition of the term ‘educational’ raises debate according to the point of view of the interested person. Ramos (2005) explains that for some authors, educational software are software that are specifically created and destined to be used in educative situations. For other authors, educational software are all software that are used in a context of learning-teaching. Finally, others consider educational software as all software that provide content.

To understand better the differences of view, it is important to go back to the first ideas of utilization of the computers in the area of education that are strongly determined by the previous experiences of teaching through machines with the famous examples of Pressey[6] and Skinner[7]. Those mechanical teaching machines instruct students without a human teacher and provide immediate feedback with a score counter, and sometimes rewards. Later, people like Papert (1993) defended that computers should be used to learn, think and discover instead of teaching by repetition. Both visions highlight that educational software are programs where the objective is to favor the process of learning, in other words, programs that have as aim to lead the students to construct a determinated knowledge related to a didactic content.

In the recent literature, few authors are giving a global definition of what they consider an educational software. It is the case for Fino (2013) that defines the ideal educational software as a software that gives the learners access to a diversified extrapolation and that allows them to control the courses of the events and negotiate the sequences of the operations. A software where the initiative belongs completely to the learner, and where the errors can result in new opportunity of learning. In conclusion, it has to be created in the aim of learning, testing things and gaining competencies.

Puustinen, Bakerz and Lundz (2006) also define the concept of educational software. According to them, this type of software is build on a coherent set of relations between the tools that learners have at their disposal (e.g. text, graph, multiple choices), the actions that can be carried out with the tools (e.g. reading, data entry), the learners' characteristics (e.g. prior knowledge), and the pedagogical goals that can be achieved as a result (e.g. understanding the processes underlying the greenhouse effect, becoming a more responsible citizen), in given situations (e.g. at home, at school, at work).

In this work, we refer to the definition of educational software of Lacerda Santos (2009). According to him, educational software could be described as a space to propose construction of knowledge created with a pedagogical aim and a mission to deepen the cognitive human process. Effective educational software can be differed from a normal software by the following characteristics (Lacerda Santos G. , 2006, p. 22):

- consistent educational objectives that can be applicable to a larger context and fulfills the way of working of the teachers, the mission of a school and the demands of the society.

[6] Sidney Pressey, 1924, inventor of the automatic teacher, the first electronic device used in schools.
[7] Burrhus Frederic Skinner, 1954, inventor of the teaching machine, the first self-management machine teaching machine.

- adequate curriculum that respond to the expectations of the school in terms of construction of the knowledge and also that valorizes the production of knowledge by the students.
- possibility of integration of different languages of communication.
- valorization of the potential of the student and the multiple intelligences of the people.
- integration of the ergonomically principles as well as the pedagogical content and social consistency.
- possibility of doing collaborative works.
- possibility of integration of teachers work without limitations of possibilities.
- degree of complexity that allows the manifestation of the process of learning that vary from one person to another.
- integration of evaluation models with more flexible educational approaches.
- integration of the student in the software to facilitation the learning through reflection, questioning, and all the other elements that take the student away from his passiveness.

We also add a crucial element that Gamboa and al. (2003) highlight in their article 'New Strategies in Educational Software', which is the intrinsic motivation. Indeed, in order to work fine, educational software need also to be attractive and attainable for students.

Our definition of educational software responds thus to the following questions: [What] it refers to a program that includes both content (information) and instructional methods (techniques) to help the learners [How] through media to construct knowledge with a pedagogical aim [Why] in order to deepen their knowledge.

However, in the literature, we encounter other words that include a definition of educational software. It is the case for example of 'learning material', 'e-learning product', 'multimedia material' and 'courseware'. We join the following definitions of those broader terms:

<u>Learning material:</u>

By learning material we concur with the definition of Nokelainen (2006) that defines it as a webpage, application or module in a learning platform that has its own user interface, evaluation and defined learning goal.

<u>E-learning product:</u>

We define e-learning using the definition of Mayer & Clark (2011) seeing it as instructions delivered on a digital device such as a computer or a mobile device that is intended to support learning and has the following characteristics: it is stored on servers or memory; it includes content relevant to the learning objective; it uses media elements to deliver content; it uses instructional methods (examples, feedback,…) to promote learning; it may be instructor led (synchronous) or designed for self placed study (asynchronous); and it helps learners build new knowledge and skills linked to individual learning goals or improvement.

(Interactive) multimedia material:

A multimedia material includes content that is characterized by the integration of multiple media elements (audio, video, graphics, text, animations, etc.), into one synergetic and symbiotic whole that results in more benefits for the user than what one element can provide individually. When the user is controls the 'what', 'when' and 'how' of the elements that are delivered, it becomes an interactive multimedia material (Phillips, 1997).

Courseware:

A courseware is an educational software designed to provide instruction or training more specifically for classroom use (Nokelainen, 2006).

Computer based-learning / technologies:

Frequently based on constructivist and cognitivist learning theories, these environments focus on teaching both abstract and domain-specific problem solving. It includes microworlds computer environments where learners could explore and build), simulations (computer environments where learner can play with parameters of dynamic systems) and hypertext (Roschelle, Pea, Hoadley, Gordin, & Means, 2000).

Typology of educational software

Typology of educational software guarantees a more comprehensive delimitation of the concept. Therefore, we bring foreword different examples of educational software typology and we will give our position regarding the concept.

The most commonly typology met in the literature is the one that differs the different types of educational software based on the different uses that the software can have in education.

- Tutorials: Software where the information is organized according to a specific pedagogical sequence. They are used to present new information and have the advantage of an immediate feedback and an evaluation of the performance.

- Exercises: Software that uses questions and answers to usually review an already seen content. Those exercises, mostly repetitive, chronometer and in the form of a game, are used for courses that require some memorization and repetition.
- Programations: Software where the student learns to program with the computer, in other words languages of programation.
- Applications: This includes text processors and electronica spreadsheets.
- Simulations: Software that simulates real situations and where the student can test, take decisions, analyze, synthesize and apply its knowledge. This allows an exploration of situations that otherwise would be difficult to realize because of a high level of risks, dangerous manipulations, complexity, price or a long period of time.
- Games: Usually used to entertain, they are used a lot in education nowadays to learn in a different way than through the book.
- Problem solvings: Those present situations that stimulate the student to meet strategies to resolve problems through a series of operations that allow the application of knowledge (Oliveira M. , 1997).
- Databases: Gathering of information that can be reused to other evaluations, analyses, comparisons or syntheses.

Baumgartner and Payr (1998) gather those categories and distinguish three different aspects of software use:

- The software as a subject: The aim is to learn the use of software, programming languages and build applications to have skills and qualifications necessary for the market. This refers to the category ‘programations’ above.
- The software as a tool: this characterizes the use of application software the students use to realize a project or to carry out tasks in various subjects. They are defined as ‘pedagogically neutral’ because their design is open and depends of the user. This includes the categories ‘applications’ and ‘database’.
- The software as a medium: the software itself vehicles simple or complex content. It transports thus, in an implicit or explicit way, theories about the user’s learning process. This is valid for ‘tutorials’, ‘exercises’, ‘simulations’, ‘games’ and ‘problem solving’.

This typology fits us better because it highlights the category of software we are focusing on. Indeed, while speaking about educational software we will refer to the software as a medium, the softwares that vehicle learning content.

Lê and Lê (2007) propose a metaphor approach to differ the different educational software and the way to evaluate them. The categories are the followings :

- Software as a tool: Equally to the authors mentioned above, it refers to software that perform functions such as calculating, drawing, editing, communicating, etc. Those software are examined the best with a checklist that takes into account some of the following aspects: the background of the target learners, the ICT literacy awareness, user-friendly features, help facilities and hardware support.
- Software as an instructor: This metaphor represents the instructionalist point of view where the lessons are arranged from introductory to advanced levels and the Initial-Response-Evaluation model[8] is applied. The evaluation of such a model takes into account questions referring to the instructionalist point of view such as: Does the software have clearly stated learning objectives? Is the software structured in terms of complexity and its developmental stages? Does the software provide items for testing?
- Software as a facilitator of learning: This metaphor represents the constructivism point of view, where the central point is the role of the learner in the learning process. The important underlying questions for such programs are the following: Does the software promote curiosity and inquiry? Does the software give choices for learners to control their own learning? Does the software present tasks for problem solving? Dies the software provide collaborative learning experiences? Does the software provide interaction with others? Does the software provide educational tools and relevant resources for learners and teachers?
- Software as a virtual class: Also often called courseware, those educational software are learning programs designed to teach a specific skill. Flexibility, interactivity, multimedia power, resources, learning experience and learning evaluation are some of the elements that need to be evaluated.

This last classification allows us to be more concrete in our concept of educational software. In this work, we refer thus to educational software as a medium that facilitates the learning through a constructivist approach. It also gives a first idea of the important questions that have to be raised and will be discussed more in detailed in the following chapters.

[8] The Initial-Response-Evaluation model consist in three stages: 1/ an initial stage where the content is introduced with a definition, description and explanation; 2/ a response stage where the learners are expected to gain knowledge and know how to bring it into practice; 3/ an evaluation stage with an evaluation of the learning takes place through responses given by the learners.

In the literature, other typologies can be found based on the targeted public. This is for example the case of Basque and Lundgren-Cayrol (2003) that analyzed 24 different typologies of ICT and propose three categories of educational software: centered on the teacher, centered on the school and centered on the learner. However, we will not make this difference since the educational software we refer to can be part of the three categories.

Learning principles and approaches

The use of educational software is different according to the learning approaches. A behaviorist approach or constructivist approach will not integrate the computer the same way. Niederhauser and Stoddart (2001) examine the relationship between the teachers' instructional perspectives and their use of technology in instruction. The results show that the pedagogical perspective has a powerful influence on how the teachers use technology. However, technology promotes not only the use of constructivist approaching to teaching and learning like we could think. Moreover, one educational software can integrate elements of both approaches.

The behaviorist approach

The behaviorist theories have been the first to influence the use of computers in education; Skinner and Pressey mentioned above being some examples. Papert (1990) sees it as insctructionalist where the computer is based on the action of transmission of information to the learner. Educational technology is than a didactic teaching machine with a linear transmission. It is based on the objectivist view that learning should involve students in mastering and replicating the knowledge and skills transmitted to them in school by the teacher, the authoritative role (Baumgartner & Payr, 1998).

Computer programs based on this paradigm are widely used in traditional classrooms in the form of drill and practice-based integrated learning systems. In this theory, the computer is a tool for hierarchically structuring a sequence of activities as a way to make patters of material and managing the stimulus/response/feedback loop that constitutes the behavioral conditioning process. The learner then receives feedback regarding whether he or she has provided the right answer.

This theory consists thus in educational software that use traditional methods and leads to a standardization of the user behavior based on what the software offers (Lacerda Santos & Lacerda, 2009). However, everyone has its own way of learning and discovering and some learning material need other approaches to reach a good understanding.

The cognitivist approach

In contrast to behaviorism, cognitivist approaches attempt to infer the internal process that take place in learning while looking at the learning process itself rather than just the results. This raises the question of the role of the professors always seen as producer of knowledge. In fact, the teachers can be said to have the role of a tutor.

This student-centered approach can be separated in different subfields. It is the case for example of generative learning that links the new content to the pre-existing knowledge and where teaching therefore becomes a process leading the learner to construct meaning from the actions (Wittrock, 1991). Self-learning that focuses on the learners determination of themselves of the goals and contents, forms and means, as well as the place of their learning (Zimmerman, 1990); and experimental learning that is defined as learning through reflection on doing (Kolb, 1984).

The constructivist approach

Another subfield of cognitivist is constructivism. This paradigm appeared with the development of the human knowledge in psychology and is characterized by individuals developing knowledge and understanding by forming and refining concepts (Piaget, 1952). Learning is thus ongoing reconstruction process where learners should be assisted in some way to construct and refine concepts in personally meaningful way and the teacher helping to create this learning atmosphere that stimulates the student to construct knowledge (Moretto, 2002).

Educational technology can be a tool of thinking and reflection. The computer provides students with the experiences that allow them to discover and reinvent concepts. Students are given access to a variety of open-ended applications that they use to help construct more complex understandings. The learner acts as an active seeker of information who revises and updates his or her knowledge through the process of gathering new information and experience new elements.

Learning is thus seen as a process of perceptions and construction of intern knowledge based on personal interpretation. This theory of learner-centered approach promotes open-ended constructivist software such as interactive and educational games, exploratory software and tool programs. The important use of computers in education facilities the development of micro-worlds used to encourage exploration and discoveries for the construction of knowledge. The learning environments should be provided for the learning to explore the behavior of systems, environments or artifacts.

Based on the constructivist goals of to Vygotsky, several authors like Jonassen (1994), Wilson (1996), Ernest (1995), Holebein (1996), give principles for instructional design that have in mind to integrate this paradigm in their learning environment. This is a recapitulative of the most recurrent principles of those authors:

Summary of principles for constructivist learning and teaching software

1/ Multiple perspectives and representations of concepts and content are presented and ecouraged.
2/ Goals and objectives are derived by the student or in negotiation with the teacher or system.
3/ Teachers serve in the role of guides, monitors, coaches, tutors and facilitators.
4/ Activities, opportunities, tools and environments are provided to encourage metacognition, self-analysis, self-regulation, self-reflection and self-awareness.
5/ The student plays a central role in mediating and controlling learning.
6/ Learning situations, environments, skills, content and tasks are relevant, realistic, authentic and represent the natural complexities of the real world.
7/ Primary sources of data are used in order to ensure authenticity and real-world complexity.
8/ Knowledge construction and not repreduction is emphasized.
9/ This construction takes place in individual contexts and through social negotiation, collaboration and experience.
10/ The learner's previous knowledge constructions, beliefs and attitudes are considred in the knowledge construction process.
11/ Problem-solving, higher-order thinking skills and deep understanding are emphasized.
12/ Errors provide the opportunity for insight into student's previous knowledge constructions.
13/ Exploration is a favoured approach in order to encourage students to seek knowledge independently and to manage the pursuit of their goals.
14/ Learners are provided with the opportunity for apprenticeship learning in which there is an increasing complexity of tasks, skills and knowledge acquisition.
15/ Knowledge complexity is reflected in an emphasis on conceptual interrelatedness and interdisciplinary learning.
16/ Collaborative and cooperative learning are favoured in order to expose the learner to alternative viewpoints.
17/ Schaffolding is facilitated to help stuents perform just beyond the limits of their ability.
18/ Assessment is authentic and interwoven with teaching.

Figure 4 : Principles for constructivist learning and teaching software

Since our research integrates itself in a further expression of constructivist learning, we will complete those criteria with more specific points of the discovery learning perspective.

<u>Explorative learning or discovery learning</u>

According to Bruner (1961) discovery learner happens when learners are stimulated by the learning environment, are actively engaged in solving problems, independently acquire their own experience and carry out experiments. The main aim of education should be to create autonomous learners. This happens through the three modes of representation: enactive representation, iconic representation, symbolic representation. In other words, learning starts with an action based information on which an image will be linked and where the information is integrated in the form of a symbol. This learning progress through different intellectual stages allow the interaction with the world by exploring and manipulating objects and as a result, the learner may be more likely to remember concept and knowledge discovered on their own in contrast to the transmission model.

The method of discovery learning has been recognized as success from several authors (Malone, 1982), (Carroll, 1982), (Scheiderman, 1983) since it allow learners to engage with the learning tasks not only in active ways but also constructively to allow them to go beyond the presented information. It promotes autonomy, responsibility and independence, encourage active engagement and develops the problem solving skills.

However, studies have been conducted to prove that discovery learning has an unfavorable effect specifically with beginning learners. According to Mayer (2003), although constructivist-based approaches might be beneficial, a free exploration in a complex environment can generate heavy working memory because of its lack of structure. Quick guided instructions for the beginners could be a solution (Sweller and al., 1990) as well as feedback (Alfieri and al., 2011).

Based on the previous foundations, the following principles can thus be added for the case of a discovery learning educational program.

Summary of principles for discovery learning and teaching software

1/ Learners are exposed to a minimal guidance (e.g.: manuals, simulations, examples,….).
2/ The target information must be discovered by the learner within the confines of the task and its material.
3/ The exploration of complex phenomena or learning domains that require heavy loads on working memory are avoided.
4/ The instructions should be organized appropriately in order to allow people of all levels to access.
5/ Teaching practices should employ scaffolded tasks that have support in place as learners attempt to reach some objective.

Figure 5 : Principles for discovery learning and teaching software

Cognitive theory of multimedia learning and cognitive overload

As mentioned before, discovery learning is criticized by some authors because of its possibility of cognitive overload. Mayer and Moreno (2003) give some concrete advice to reduce this cognitive overload in a multimedia context.

The theory of multimedia learning is based on the cognitive learning theory that explains how mental processes transform information received by the eyes and ears into knowledge and skills in human memory. It is based on three assumptions about how the mind works in multimedia learning: dual channel, limited capacity and active processing. Mayer and Moreno (2003) define it in the following figure:

Assumption	Definition	Based on
1. Dual channel	Humans possess separate information processing channels for verbal and visual material: - an auditory/verbal channel - a visual/pictorial channel	Paivio's dual-coding theory (1986) Baddeley's theory of working memory (1998)
2. Limited capacity	There is only a limitated amount of processing capacity available in the verbal and visual channels	Chandler and Sweller's cognitive load theory (1999) Baddeley's theory of working memory (1998)
3. Active processing	Learning requires substantial cognitive processing in the verbal and visual channels.	Wittrock's generative-learning theory (1989) Mayer's selecting-organizing-integrating theory of active learning (1999,2002)

Figure 6 : Three assumptions about how the mind works in multimedia learning (Mayer & Moreno, 2003)

Those three assumptions are integrated within the modes of knowledge representations:

- physical representations: e.g. words or pictures that are presented to the learner;
- sensory representations: in the ears or eyes of the learner;
- shallow working memory representations: e.g. sounds or images attended for the learner;
- deep working memory representations: e.g. verbal and pictorial models constructed by the learner
- long-term memory representations: e.g. the learner's relevant prior knowledge.

Those models of knowledge representations follow the three important cognitive processes the learner engages: the selecting process, the organizational process and the integrating process.

This is illustrated in the following figure:

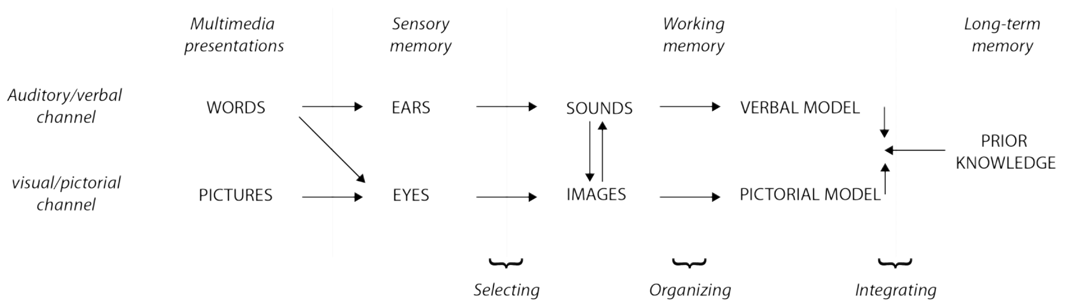


Figure 7 : Cognitive theory of multimedia learning (Mayer & Moreno, 2003)

If the capacities for physically presenting words and pictures is virtually unlimited, such as the storing knowledge in long-term memory, the capacity for mentally holding and manipulating words and images in working memory is however limited. When the processing demands evoked by the task exceeds the processing capacity of the cognitive system, the situation is called cognitive overload.

Three kinds of cognitive demands can be characterized:

- Essential processing: it refers to the cognitive process needed to make sense of the material. It includes the whole process of the cognitive theory of multimedia learning, from the selection to the integration.
- Incidental processing: it refers to the cognitive process that is not required to understand the material but is primed by the design of the learning task.
- Representational holding: it refers to the cognitive process aimed at holding a mental representation in working memory over a period of time.

To reduce the overload of information that the learner can not assimilate because it exceeds its cognitive capacities, a reduction of incidental processing as well representational holding can be involved. Mayer and Clarck (2011) propose several ways to reduce cognitive load in multimedia learning through proven guidelines for consumers and designers of multimedia learning. The guidelines are related to the following principles:

The principles of multimedia design:	*The related guidelines:*
1/ Multimedia representation principle it is better to present an explanation in words and pictures than solely in words	- Use relevant graphics and text to communicate content - Use animations to demonstrate procedures; use a series of stills to illustrate processes - Use explanatiory visuals that show relationships among content topics to build deeper understanding - Use relevant graphics explained by audio narration to communicate content
2/ Contiguity principle when giving a mutlimedia explanation, present corresponding words and pictures contiguously rather than separately	- Integrate text nearby the graphic on the screen - Allow learners to play an animation before or after reviewing a test description - Avoid covering or separating information that must be integrated for learning
3/ Coherence principle when giving a multimedia explanation, use few rather than many extraneous words and pictures	- Use simpler visuals to promote understanding - Avoid irrelevant graphics, stories, and lengthy text - Avoid irrelevant videos, animations, music, stories and lenthy narrations
4/ Personalization principle	- Write in a conversational style using first and second person - Use virtual coaches/agents to deliver instructional content such as examples and hints - Script audio in a conversational style using first and second person - Script virual coaches to present instructional content such as examples and hints via audio
5/ Segmentation principle	- Break content down into small topic chunks that can be accessed at the learner's preferred rate - Break content down into small topic chunck that can be accessed at the learner's preferred rate using a continue or next button - Use a continue and replay button on animations that are segmented into short logical stopping points
6/ Pretraining principle	- Teach important concepts and facts prior to procedures or precesses - When teaching concepts and facts prior to procedures or processes, maintain the context of the procedure or process - Teach important concepts and facts prior to procedures or processes
7/ Temporal contiguity principle	- Do not allow separation of visuals and audio that describes the visual
8/ Redundancy principle	- Do not present words as both onscreen text and narration when there are graphics on the screen

Figure 8 : Principles of multimedia design and their guidelines

The principles demonstrate that it is possible to take a learner-centered approach to instructional technology and are important for the construction of our referential for mathematical software of discovering learning.

Conclusion

In this chapter we have first explained what we consider to be an 'educational software' and define it as a space to propose construction of knowledge created with a pedagogical aim and a mission to deepen the cognitive human process. The characteristics mentioned to differ an educational software from an other software such as the valorization of the potential of the student, the adaptation to each learner complexity and the facilitation of the learning through reflection questionings and other elements that take the student away from his passiveness are also elements mentioned in the constructivist approached developed after.

In a second time, we have answered to the first objective of this research project: **define important elements to discovering learning** through the importance of elements such as encouraging the active engagement; promote autonomy, responsibility and independence; develop problem solving skills, providing feedback.

After the findings of the different elements an educational software should take into account, we synthesizing the different informations of this chapter over constructivism, discovering learning and the cognitive theory of multimedia learning, and could conclude the following interesting criteria to integrate in our referential:

Summary of principles regarding our first part: Education

Multimedia representation:
- It is better to present an explanation in words and pictures than solely words (multimedia representation principle)
- When giving a multimedia explanation, present corresponding elements contiguously rather than separately (contiguity principle)
- When giving using multimedia use few rather than many extraneous words and pictures (coherence principle)
- Adapt the multimedia content to the targeted public (personalization principle)
- Do not present words as both onscreen text and narration when there are graphics on the screen. (redundancy principle)

Educational objectives:
- The educational objectives are consistent and derived by the student. They also fulfill the way of working of the teachers, the mission of the school and the demands of the society.
- Activities, opportunities, tools and environments are provided to encourage metacognition, self-analysis, self-regulation, self-reflection and self-awareness.
- The program promotes knowledge construction and is not based on reproduction. Moreover, exploration is a favored approach in order to encourage students to seek knowledge independently and to manage the pursuit of their goals.
- The construction takes place in individual contexts and through social negotiation, collaboration, cooperation and experience in order to expose the learner to alternative viewpoints.
- Scaffolding is facilitated to help students to perform just beyond the limits of their ability.

Tasks:
- The tasks are relevant, realistic, authentic and represent the natural complexities of the real world.
- The tasks arouse problem-solving, higher-order thinking skills and deep understanding.
- Assessment is authentic and interwoven with teaching.
- The target information must be discovered by the learner within the confines of the task and its material.
- Break content down into small topics that can be accessed easier (segmentation principle).

Role of the teacher:
- Teachers serve in the role of guides, monitors, coaches, tutors and facilitators.

Assessing learning:
- Errors provide the opportunity to insight into student's previous knowledge constructions.
- Learners are exposed to a minimal guidance.
- The instructions should be organized appropriately in order to allow people of all levels to access.

Figure 9 : Summary of principles regarding our first part, education

PART 2: MATHEMATICS

FROM EFFECTIVE TEACHING OF MATHEMATICS TO MATHEMATICAL SOFTWARE

Predictive evaluation of ***effective*** *educational software design:*
development of a pedagogical usability referential for ***mathematical software*** *of discovering learning*

Introduction

We develop this part of the area of mathematics because the aim of our work is to evaluate the educational software of mathematics, GGBook. Therefore, we researched criteria for effective learning and teaching of mathematics, the integration of technology in mathematical education as well as criteria for mathematical educational software design and dynamic mathematical software. This will be concluded by our second and third objective: identifying characteristics of effective mathematical teaching and ergonomics criteria for effective mathematical software.

Effective learning and teaching of mathematics

The International Academy of Education[9] (Anthony & Warlsaw, 2009), gives principles of effective teaching of mathematics. It consists in 10 principles that have to be understood in a larger network that includes the school, home and community. The principles are the following one's:

[9] The International Academy of Education is a not-for-profit association that promotes educational research and its disseminations and implementations.

Summary of principles principles of effective teaching of mathematics

1. Having an ethic care
    a. Caring about the development of students' mathematical proficiency
    b. Caring about the development of students' mathematical identities
2. Providing opportunities to work both independently and collaboratively to make sense of ideas
    a. Independent thinking time
    b. Partners and small groups
    c. Whole class discussions
3. Planning mathematics learning experience that enables students to build on their existing proficiencies, interests and experiences
    a. Connecting learning to what students are thinking
    b. Using students' misconnections and errors as building blocks
    c. Appropriate the challenge
4. Understanding that the tasks and examples they select influences how students come to view, develop, use and make sense of mathematics
    a. Mathematical focus
    b. Problematic task
    c. Practice activity
5. Supporting students in creating connections between different ways of solving problems, between mathematical representations and topics, and between mathematics and everyday experience
    a. Supporting making connections
    b. Multiple solutions and representations
    c. Connecting to everyday life
6. Using a range of assessments practices to make student's thinking visible and to support students' learning
    a. Exploring students reasoning and probing their understanding
    b. Teacher questioning
    c. Feedback
    d. Self and peer assessment
7. Ability to facilitate classroom dialogue that is focused on mathematical argumentation.
    a. Scaffolding attempts at mathematical ways of speaking and thinking
    b. Developing skills of mathematical argumentation
8. Shaping mathematical language by modeling appropriated terms and communicating their meaning in ways that students understands
    a. Explicit language instructions
    b. Multilingual contexts and home language
9. Selecting tools and representation to provide support for students' thinking [tools: symbols, graphs, diagrams, images, textbook,…]
    a. Thinking with tools
    b. Communicating with tools
    c. New technologies (calculators, applications, TBI, mobile device,…)
10. Developing and using sound knowledge as a basis for initiating learning and responding to the mathematical needs of all their students
    a. Teacher content knowledge
    b. Teacher pedagogical content knowledge
    c. Teacher knowledge in action
    d. Enhancing teacher knowledge

Figure 10 : Summary of principles principles of effective teaching of mathematics

Another report of the International Academy of education (Grouws & Cebulla, 2000), explains other principles to improve the student achievement of mathematics. Next to the highlighted principles, we make a possible lecture of it regarding virtual environments.

General: Classroom	*Applied: Educational software*
The opportunity to learn	The educational software is based on a concrete mathematical objective with sub-tasks leading to it. It has also to be adequate to the curriculum of the user.
The focus on meaning	A simple learning context, a clear structure and connections and understandable instructions are present.
Learning new concepts and skills while solving problems	The software has tasks that develop the skills of the user. This can be done based on previous concepts in order to build new knowledge.

Opportunities for practice and invention	The program contains exercices of practice of already seen concepts, but allows as well possibilities of invention to the learner.
Openness to students solution methods and student interaction	The software allows discovery learning and several ways are available to arrive to the solution. It contains as well mechanism of interaction with the user.
Small group learning	The software contains elements of cooperation and collaboration.
Whole class discussion	The software has a section where discussion can take place.
Number sense	There is an intuitive feel for the presented content. The user does not precise an involvement of the teacher.
Concrete materials	The software presents content related to daily situations.
Student's use of calculator	The students need to use mathematical tools in order to achieve the tasks.

Figure 11 : Principles to improve the students' achievement of mathematics

This can be completed by elements that Stronge highlights in his book of Qualities of Effective Teachers (2007, p. 117):

General: Classroom	*Applied: Educational software*
Establish instructional procedures	The educational software has clear instructions of use and gives informations regarding its utilization.
Manage students behavior and activities through clear expectations	The presence of feedback in the software.
Provides easy access to instructional materials	The instructions are clear and easily available.
Manages emergency situations as they occur	Positive and learning management of the mistakes of the users.
Provides positive reinforcement	The software encourages the users in their positive actions.
Exhibits consistency in management style	The software is consistent.
Maintain a personal work space that can be personalized	Personalization is available in the software.

Figure 12 : Principles of effective teaching

An effective learning environment (real or virtual) should thus embed the following important elements for mathematical education (Ministery of Education of Canada, 2003):

- foster positive beliefs and attitudes about mathematics;
- value prior knowledge and make connections between important concepts in mathematics, the learner's world, and other subjects;
- build a community of mathematics learners where mathematics is seen, heard, and felt;
- focus on important mathematical concepts;
- encourage learning through problem solving;
- make strong links with the home and community;
- use resources that aid understanding;
- recognize and support the important role of the teacher;

• be supported by principals, senior administrators, and school boards;

According to the National Council of Teachers of Mathematics (2000), this can be applied through six principles:

Summary of principles of effective learning and teaching of mathematics

- Equity: high expectations and strong support for all students.
- Curriculum: collection of activities that must be coherent, focused on important mathematics, and well articulated across the grades.
- Teaching: understanding of what students know and need to learn and challenging and supporting them to learn it well.
- Learning: Students must learn mathematics with understanding, actively building new knowledge from experience and prior knowledge.
- Assessment: Assessment should support the learning of important mathematics and furnish useful information to both teachers and students.
- Technology: Technology is essential in teaching and learning mathematics; it influences the mathematics that is taught and enhances students' learning.

Figure 13 : Principles of effective learning and teaching of mathematics

An effective teaching framework provides thus a balance of teaching strategies, student groupings, and types of activities.

Integrating technology in mathematics education

The first great technological change in mathematics teaching is the arrival of the pocket calculator in the 1970s. Indeed, the changes in pedagogy where quiet high and allowed to: "increase the importance of experimental and discovery learning, strengthen modeling and mathematical concepts, enhance application tasks, reduce the importance of manual computational skills, increase the importance of algorithms" (Weigand & Weth, 2002, p. 4).

This first introduction of technology in the field of mathematics was perceived as controversial and raised some pedagogical questions. If the impact of this tool affected some mathematical topics, however, it did not really change the objectives, methods and assessments in mathematical education (Fey & Hirsch, 1992).

Some years later, the computer enters the school with as aim to prepare the students for to this new technology. If during the beginning the focus laid on the use of the computer itself, it focuses later more on the content and the first mathematical software where created. They consisted mainly in drill-and-practice programs, computer-assisted instructions and geometry software.

The first dynamic programs that emerged are Sketchpad and Cabri Geometry, that allowed the creation of dynamic figures in which points and segments can be dragged while preserving the properties defining the figure. However, neither of those programs was conceived with clear educational aims in mind (Scher, 2000). The general idea was more to bring an occasion for conjecturing and creativity for students and teachers in the mathematics classrooms.

Only later, specific experimental environments where created with as aim the collaborative learning and the student exploration and encouragement. Dynamic geometry systems are thus seen as ‘‘providing a setting in which students can construct and experiment with geometrical objects and relationships’’ (Hoyes & Noss, 2003, p. 333). As Pereira (2002) explains, there is a mathematical change from a static deductive activity to an exploratory, inductive activity that emphasizes the heuristics involved in discovering results where the role of the mathematic student becomes almost one of the scientist.

In this change, teachers need to accept that learning might take place in the computer-based situation without reference to a paper-and-pencil environment. The NCTM[10] expresses this phenomenon clearly: “ The effective use of technology in the mathematics classroom depends on the teacher. Technology is not a panacea. As with any teaching tool, it can be used well or poorly. Teachers should use technology to enhance their students’ learning opportunities by selecting or creating mathematical tasks that take advantage of what technology can do efficiently and well — graphing, visualizing, and computing”. (NCTM, 2000, p. 25) According to Lawless and Pellegrino (2007) technology can make it quicker and easier to teach the same things in routine ways. Moreover, it allows teachers to adopt new and better approaches to instruction and change the context of learning if the technology is used responsively with the goal of enriching students’ learning of mathematics.

From a study (Lins, 2003) on the use of the dynamic geometrical software in classrooms appears that dynamic geometry gives the student a higher possibility to explore, propose, and try to demonstrate. Other highlights elements are the fact that it makes student work with figures easier, faster and more accurate; it guides students to learn by themselves; and promotes student conviction, understanding and remembering.

[10] NCTM: National Council of Teachers of Mathematics was founded in 1920 and integrates more than 80.000 members throughout the United States and Canada. It is a voice mathematics education, supporting teachers to ensure equitable mathematics learning of the highest quality for all students through vision, leadership, professional development, and research. More information : http://www.nctm.org

Dynamic mathematical education software

Different types of software tools are used for mathematical education. Computer Algebra Systems (CAS), Dynamic Geometry Software (DGS) and Spreadsheets are the current mathematical educational software for teaching and learning mathematics (Drijvers & Trouche, 2007).

They focus on different parts of mathematical topics, however, with the high expansion of software, the limits are blurred and characteristics can be joined. In this context, a new type of educational software is designed with a purpose to join the advantages of the above described software: the dynamic mathematics software. This includes thus a wider range of mathematical contents, grade levels and teaching methods.

The CAS are designed to facilitate the manipulations of mathematical expressions in symbolic form. Generally, they deal with the symbolic and numeric representation of mathematical objects. They allow for manipulating a variety of algebraic expressions and functions, and can deal for example with basic mathematical operations, simplification, factorization, derivatives, integrals, sequences, and matrices. Moreover, they allow plotting graphs of functions and equations. Usually, they operate using keyboard input and enable users to implement their own algorithms using commands and a special syntax (Fuchs & Hohenwarter, 2005).

The DGS are software witch are predominantly used for the construction and analysis of tasks and problems in elementary geometry and provide basic mathematical objects such as points, segments, lines, circles, vectors, and conic sections. They are operated mainly with the mouse by activating different geometric tools and applying them to the drawing pad. They differ from the other software because of their drag mode (relations and dependencies between the objects are maintained while an object is updated dynamically), their customized tools (adapted toolbar with commands) and their traces (allowing users to examine movements and dependencies between the objects) (Strässen, 2002).

The spreadsheets are computer applications that allow the display of text or numeric values in table cells organized in rows and columns. It is usually used for statistical calculations.

Dynamical mathematics software provide the above mentions characteristics into a same package and propose the combination of those features in a higher degree of interaction between them. Our studied case, the educational software of mathematics GGBook, is one good example of dynamical mathematics software.

In their article, Ruuthwen and al. (2008) give recommendations for an effective dynamical mathematical software. They are presented in this figure:

Recommendations for an effective dynamical mathematical software

1/ Prepared teacher use of a dynamic figure to support persuasive mathematical presentation
- Maintaining student attention through dynamic presentation and tactical questioning
- Making properties apprehensible and convincing to students through dragging
- Making it easy for students to identify properties by pre-empting possible confusions
- Avoiding the disadvantages of student software use through teacher presentation

2/ Structured student use of dynamic geometry to support mathematically principled interaction
- Developing student awareness of space and shape through exploring dynamic figures
- Giving students experience of geometrically principled interaction with the software
- Focusing student attention on mathematical essentials through structured software use
- Supporting students in questioning unexpected results and learning from them

3/ Structured student use of a customized dynamic figure to provide a mathematical reference model
- Building student understanding through paired investigation of the dynamic model
- Prompting students to appreciate errors by relating solutions back to the dynamic model
- Making software use manageable for students by preparing figures and measurements
- Bringing out key points through guiding student use of the customized figure

4/ Guided student use of dynamic geometry to support mathematically disciplined expression
- Giving students experience of finding rules and patterns within abstract geometry
- Emphasizing mathematical rules through clarifying student instructions to the computer
- Making mathematical properties noticeable through prompting dragging by students
- Making learning less vague through getting students to write a rule clearly

Figure 14 : Recommendations for an effective dynamical mathematical software

Criteria for mathematical educational software design

According to Hennessy (2000), learning Mathematics through the use of technology simplified learning and increase students understanding and confidence. Teoh and Fong (2005) demonstrate that the teaching and learning using dynamic visualization approach helped students to better understand the concepts taught. Moreover, Marzita and Rohaidah (2004) explain that the use of interactive multimedia is necessary in enticing students attention and increasing students understanding of mathematics. Good mathematical software is thus an important role.

According to Vergnaud (1997) mathematical software enter in a constructivist perspective of resolution of problems and this needs to be in the center of the operationally. Gladcheff and al. (2001) as well explains that a mathematical software should have the following characteristics:

Characteristics for a mathematical software

- situations where the students discover new significations of content
- situations that enlarge their competences and concepts
- a base on cognitive processes
- enabling reasoning
- adopted strategies during the process of resolution
- stages of development relative to the skills involved and characterization of the various problems and their level of complexity
- the mathematical educational objective is clear and the tasks work the necessary aspects to meet the proposed objectives
- the mathematical language is correct and at the level of understanding of the targeted public
- the mathematical software presents a summary of what is working after the end of each session
- the software presents interaction and has features where the learner can explore mathematical knowledge
- the software is smart and does not underestimate the learner
- it presents realistic situations of mathematics
- the feedback is positive even if the user fails
- the feedback allows also the learner to reflect on his errors and try to correct it without intervention of the teacher
- the educational software offers a feedback of the student progress
- the exercices represent the reality of the learner
- it allows multiple paths to the solution

Figure 15 : Characteristics for a mathematical software

According to those authors, an effective mathematical software will be the one that allows interaction with the user about what concerns the concepts and ideas of mathematics, the discoveries, the results, the hypotheses, the tests, etc.

Another interesting study is the one of the Curatelli and Martinengo (2012) that define some criteria to overcome mathematics learning difficulties in educational tools. Among them:

Criteria to overcome mathematics learning difficulties in educational tools

- an adaptation of the mathematical educational tool to the target users
- specifications on the learning strategy to be followed
- customization of the exercices and of their parameters
- immersion in real fields of experience with situations as close as possible to real life
- overall similar layout

Figure 16 : Criteria to overcome mathematics learning difficulties in educational tools

Conclusion

In this second part of the literature review we have concentrated ourselves on the field of mathematics through an analysis of effective learning and teaching of mathematics and a history of the integration of technology in mathematics. This is followed by the definition of dynamic mathematical software and finalized with the criteria for mathematical educational software design.

Two objectives have also been reached in this part. First, **identify characteristics of effective mathematical teaching.** Indeed, we have found various principles among them opportunities of collaborative work, support of problem solving, the focus on the meaning, opportunities for practice and invention,… Those characteristics can be guided by other transitionals principles of equity, curriculum, teaching, learning, assessment and technology. Secondly, **state the ergonomics criteria for effective mathematical software.** The characteristics of a mathematical software are described in this section and include namely: enabling reasoning, clear mathematical objectives, interaction, feedback, etc.

While bringing together the different information and needs of this part of the mathematical field, we could conclude the following criteria to integrate to our referential:

Summary of principles regarding our second part: Mathematics

- the mathematical language is correct and at the level of understanding of the targeted public
- the activities are coherent, focused on important mathematics, and well articulated across the grades.
- customization of the exercises and of their parameters
- immersion in real fields of experience with situations as close as possible to real life
- coherent layout
- the mathematical educational objective is clear and the tasks work the necessary aspects to meet the proposed objectives
- the mathematical software presents a summary of what is working after the end of each session
- the software presents interaction and has features where the learner can explore mathematical knowledge
- the software is smart and does not underestimate the learner
- situations that enlarge their competences and concepts
- the feedback is positive even if the user fails
- the feedback allows also the learner to reflect on his errors and try to correct it without intervention of the teacher
- the educational software offers a feedback of the student progress
- it allows multiple paths to the solution situations where the students discover new significations of content
- encourage learning through problem solving;
- use resources that aid understanding;
- recognize and support the important role of the teacher;
- the educational software has clear instructions of use and gives informations regarding its utilization.
- the instructions are clear and easily available
- the software encourages the users in their positive actions
- the software contains elements of cooperation and collaboration.
- the software has a section where discussion can take place.

Figure 17 : Summary of principles regarding our second part, mathematics

***Predictive evaluation** of **effective** educational **software design**:*

*development of a **pedagogical usability** referential for mathematical software of discovering learning*

Introduction

To understand the recent explosion of pedagogical usability checklists and guidelines for educational material, it is important to highlight where they are coming from, the field of ergonomics for general software. For this reason, we have decided to primarily discuss what is effectiveness and quality of a software design, followed by the contextualization of predictive evaluation and the concept of usability. Only after this general perspective, we deal more specifically with the evaluation of educational software through standards, guidelines and models. The conclusion will resume the chapter and resolve three more of our objectives.

Definition of effectiveness and quality of software design

Software quality is also a term that arouses debate. Traditionally, software quality has been classified either as functional quality, based on the requirements and specifications of the software, and structural quality, referring to the robustness and maintainability of the software (Coté, Suryn, & Georgiadou, 2006). Later on, as the industry focus has shifted from functionality to improving quality new ways of thinking have appeared and a distinction is made between the capability of the software to conform to the requirements, and the customer value, or the "what we think, feel or sense as a result of the objective reality", like Shewhart (1931) defines it.

Kitchenham and Pfleeger (1996) differ clearly the different definitions of quality. According to them, there are five perspectives:

- <u>The transcendental perspective</u>: that deals with the metaphysical aspects and views quality as the way to what the software ought to be.

- The user perspective: which looks at the product in a given context of use; in other words, if the characteristics of the software meet the user's needs.
- The manufacturing perspective: representing quality as conformance to requirements.
- The product perspective: that sees the quality of a software through its inherent characteristics of the product.
- The value-based quality perspective: that define quality according to the value of the various stakeholders.

Those five perspectives that take technical ergonomical criteria as well as the user experience into consideration, is a prerequisite in the development of a quality model, together with establishment of the users' and developers' requirement early in the system's life cycle.

One quality model that respects those criteria is the ISO/IEC 9126 related to the software product evaluation – quality characteristics and guidelines for their use. Indeed, this norm integrates the different perspectives of quality in three categories:

- the quality in use: the user's view of the quality of the software product when it is used in a specific environment and a specific context of use. It focuses on the extent to which users can achieve their goals, rather than measuring the properties of the software itself.
- the external quality: the set of characteristics of the software fro an external view.
- the internal quality: the totality of characteristics of the software product from an internal view.

The relationship between them, is the following: the internal quality verified by internal quality requirements indicates the external quality validated by external quality requirements and indicates at its turn the quality in use defined by the users' uses, needs and feedbacks.

The internal and external quality model is composed of quality characteristics and sub characteristics presented as followed:

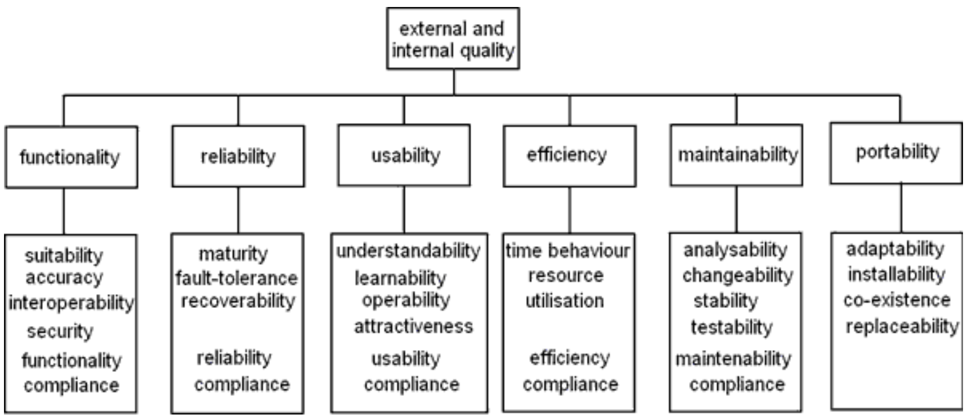


Figure 18 : Principles for internal and external quality from ISO/IEC, 2001

The quality in use model represents the following components:

Figure 19 : Principles of quality in use model from ISO/IEC, 2001

With both models, an overall quality can be predicted based on the five perspectives developed above, with the exception of the transcendental perspective that refers to quality as something is recognized and not defined. For that reason, it cannot be explicitly implemented in a software product.

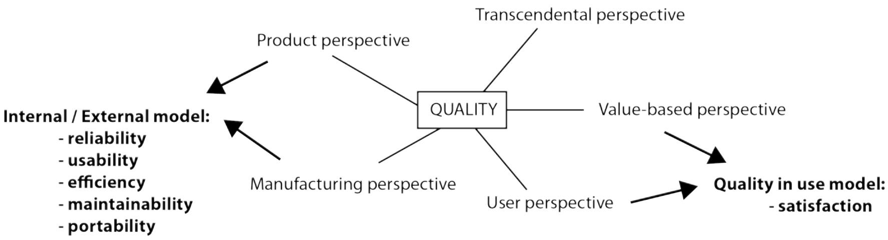


Figure 20 : The relationship between the ISO norm 9126 and the perspectives of quality

If the quality mentioned so far concerns software in general terms, very little is mentioned only about the quality and effectiveness of educational software, referring all mainly to the evaluation of effective educational software design.

However, according to Maguire (2001) the quality of an educational software depends of its intrinsic characteristics as well as its power to reach the objectives and necessities of the users. Moreover, McDougall and Squires (1994) affirm that the effectiveness of an educational software lies in the curriculum embedded in the design and the aspects related to the learning of the student. Finally, Mayer and Clark (2011) highlight the importance of the environment in which the training is developed, in other words, the aspects of the context of use.

It can thus been observed that those aspects related to the educational software take over exactly the same perspectives of quality mentioned for the normal software, by keeping always in mind the criteria from Lacerda Santos (2009) mentioned in the first section.

Software processing and predictive evaluation

There are many definitions of the term evaluation. The common agreement is that evaluation is about assessment of quality of a product, task, program or activity; the aim being to compare what something is, to what it ought to be, in order to facilitate judgment about the value of that thing. “What the software is” refers to the rational construction of pedagogical and ergonomic foundations, mostly the product of the designer’s original intention. “What the software ought to be” refers to the ideal of en effective educational software as described above and that takes into consideration the clarification after the evaluation work (Puustinen, Bakerz, & Lundz, 2006).

Three main reasons can be highlighted for the evaluation of educational software. First, an evaluation that the teachers do to see what is the best software to use in their teaching according to their educational principles; Secondly, an evaluation by developers or designers to give a feedback on the effectiveness of an educational software in the process of creation; and finally, in the case of a re-designing of a software that was not initially produced for educational purposes. In other words, one evaluation is done after the release of the educational software, the two others, before. Evaluating software before it is used with learners like an expert review or evaluation organizations' checklist approach may be classified as predictive evaluation (Squires & Preece, 1996). Moreover, Sanchez and al. (2004) mention that the strategy of evaluation needs to take three important elements into consideration during its process of creation: an auto-evaluation from the producers, an evaluation from experts and an evaluation for and from the users.

Different theoretical models exist on the life cycle of software that take into consideration all the aspects of production of software from the initial stage until its maintenance. It is the case for example of the Waterfall model of Royce (1970), the Spiral model of Boehm (1986) or the Linear model of Pressman (2006). However, in order to highlight the best the evaluation phases in the process, we refer to the evolutionary process flow. This model executes the different activities from the communication until the deployment in a circular way, like illustrated in this figure:

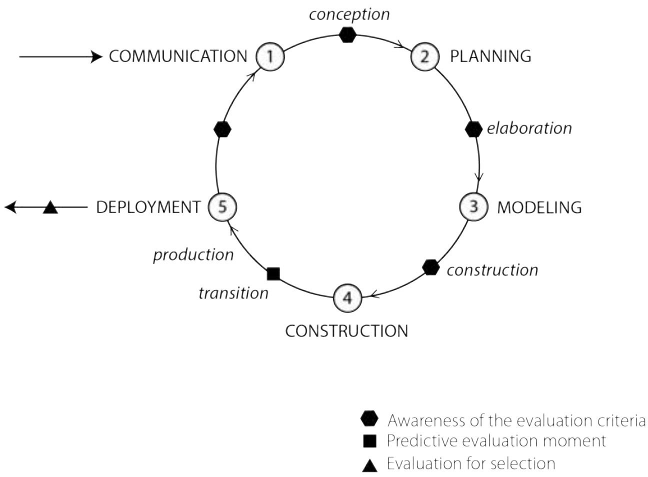


Figure 21 : Evolutionary process flow of Pressman (2006) integrating the evaluation moments

The evolutionary process flow consists in five activities that can be gathered in five phases.

- The conception phase: The conception phase consist in the communication of the customer on the planning of the activities and the generic process of the customer. During this phase, the requirements are identified, a rough architecture of the system is proposed and a plan is created for the iterative development.

- The elaboration phase: The elaboration phase includes both the planning and the modeling activities of the process. A refine and expands of the preliminaries use is executed and the architecture of the model includes other views like the use, the design, and the analysis. The baseline of the system is executed and viable.

- The construction phase: The construction phase develops the software components that make all the operations. Analyses and design choices are completed to reflect the final version of the software.

- The transition phase: The transition phase is the last part of the construction activity and the first part of the deployment activity of the generic process. During this phase, the software is given to end users for beta testing and user feedback reports for the defects and other changes. It is also in this phase that the team creates support documentation, because at the end of this phase, the software becomes usable software.

- The production phase: This phase monitors the software on its on-going use. The support of the operating software is provided and the defect reports and requests for changes are submitted and evaluated.

In this process of creation, the evaluation moments are also highlighted. Indeed, it is recommended to evaluate the software in the transition phase, where a primary version of the software can be tested. However, the evaluation criteria can be in mind of the designers and developers since the inception phase and being referred to during the elaboration and construction phase.

During this evaluation in the creation process, some important characteristics are analyzed such as the efficiency, the facilitation of maintenance, the facility of use and the level of confidence. This can be related to the definition mentioned above on the term 'quality' as well as the usability criteria that follow.

Usability

Usability comes from the area of Human-computer-interaction and focuses on the ease of use and learnability of a human-made object. The International standard ISO 9241-11 defines usability as "the extent to which a product can be used by specified users to achieve a specified goal with effectiveness, efficiency and satisfaction in a specified context of use" (ISO 9241-11, 1998).

Shackel (1991) argues the importance of the usability design as part of the system design process and defines the usability goals are the following one's:

- Effectiveness: effective to use (e.g.: the general goals,…)
- Efficiency: efficient to use (e.g.: number of steps, one-click option,…)
- Memorability: easy to remember how to use (e.g.: meaningful icons, command names,…)
- Learnability: easy to learn (e.g.: tutorials, complexity of the tasks,…)
- Utility: have good utility (e.g.: powerful tools,…)
- Safety: safe to use (e.g.: preventing users from serious errors, recovery functions,…)

There are many recommendations on how to reach those goals and to improve software system usability, even though the techniques often differ. The following table integrates the important authors of this domain with their contribution to usability:

Author (date)	Description	Type
Smith and Mosier (1986)	Give guidelines for design of user interface software in sex functional areas: data entry, data display, sequence control, user guidance, data transmission and data protection.	Usability guidelines
Mayhew (1992)	User interaction design guidelines for interfaces (menu, fill-in forms,…) and user interaction (graphics, color, windows,…).	Usability guidelines
Bastien & Scapin (1997)	Ergonomical principles used to guide the conceptors to ergonomic considerations. It focuses on the guidance, workload, explicit control, adaptability, error management, consistency, significance of codes and compatibility.	Ergonomics principles
Hix & Hartson (1993)	Design of guidelines including support for the user planning (focus on the user, memory limitations,…), the user translation (effective affordances, errors prevention, user control,…), the user physical actions (location, physical affordance, size,…), the user assessment (feedback, information, outcome,…) and issues independent of user actions (user perception, displays layout, modalities,…).	Usability guidelines
Nielsen (1993)	Usability principles including he consistency, feedback, graphic design and color, system failures, response time, prevent errors, help and documentation,…	Design heuristics
Rubin (1994)	Usability guidelines that take into consideration the usefulness, effectiveness, learnability and likeability.	Usability guidelines
Preece and al. (1994)	The authors give advice regarding the factors of usability like the facilitation of learning, facilitation of use, satisfaction of the user, flexibility,…	Usability principles
Marcus (1995)	Graphical user interface guidelines with dimensions of the interface object, colors to use and advice regarding alignment and labeling techniques.	Usability guidelines
Schneiderman (1998)	Rules of interface design about consistency, shortcuts, feedback, design, error handling, easy actions, control and memory load.	Usability rules
Constantine & Lockwood (1999)	Rules for good user-interface design gathered in principles: the structure principle, the simplicity principle, the visibility principle, the feedback principle, the tolerance principle and the reuse principle.	Usability principles
Welie (2003)	Focus on usability patterns: visibility, affordance, natural mapping, constraints, conceptual models, feedback, safety and flexibility.	Usability patterns
Tidwell (2005)	Patters designers should take into account while in the development of their product: safe exploration, instant gratification, satisficing, changes in midstram, deferred choices, incremental construction, habituation, spatial memory, prospective memory streamlined repetition, keyboard only and other people's advice.	Usability patterns

Table 1 : The important authors concerning usability

Beside the usability, Preece and al. (2011), highlight the importance of the user experience and comes up with a model joining the usability measurements with the user experience. This is represented visually as followed:

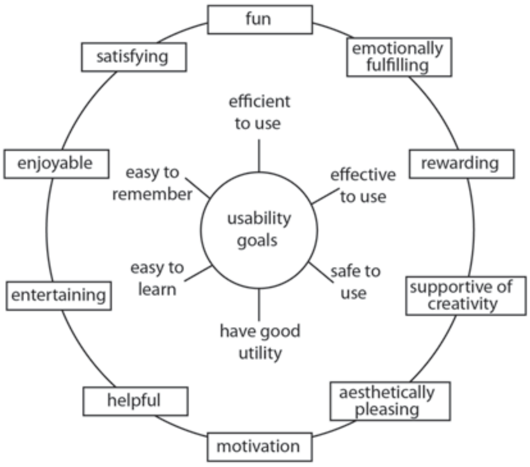


Figure 22 : Interaction design model of Preece and al. (2011)

The usability goals are at the center of interaction design, and user-experience goals are on the outer ring of the diagram, in other words, they are secondary goals. This takes over the idea developed in the previous section about quality with the internal/external model and the quality in use model that integrate objective criteria with more subjective one's.

From this analyses and relationships between those different contributions in the field of software ergonomics, ERGOLIST[11] groups the best all the above-mentioned criteria and brings it concretely through a checklist (SoftPólis, 2011):

[11] ERGOLIST consists in 3 modules : 1/ a checklist that helps to evaluate the ergonomical quality of a software ; 2/ a module of questions that helps to understand in an informal way what consists the different criteria of the checklist ; 3/ a module of recommendations that can help in the evaluation. It has been realized by professionals in usability and the collaboration between SoftPolis, Nucleo Softex-2000 de Florianopolis, and LabIUtil. It can be accessed online: http://www.labiutil.inf.ufsc.br/ergolist/check.htm

Ergonomical criteria

1. Promptness: Verify that the system informs and guides the user during the interaction.
2. Grouping by location: Make sure that the spatial distribution of items translates the relationships between the information.
3. Grouping by format: Check the shapes of the items as a means to convey associations and differences.
4. Feedback: Rate the quality of the immediate feedback to user actions.
5. Readability: Check the legibility of the information in the system screens.
6. Brevity: Check the size of codes and terms presented and introduced to the system.
7. Minimum Actions: Check the length of the dialogues established to achieve the user goals.
8. Density of Information: Evaluate the informational density of screens displayed by the system.
9. Explicit Actions: Verify that the user who is explicitly commands the system actions.
10. User Control: Evaluate the user possibilities to control the chain and carrying out the actions.
11. Flexibility: Make sure the system allows you to customize presentations and dialogues.
12. User Experience: Evaluate whether users with different levels of experience have equal opportunities to succeed in their goals.
13. Protection of errors: Verify that the system provides opportunities for the user to prevent errors.
14. Error Messages: Rate the quality of error messages sent to users in trouble.
15. Correction of errors: Check the facilities provided so that the user can correct their mistakes.
16. Consistency: Evaluate whether is maintained consistency in the code design, screens and dialogs with the user.
17. Meanings: Evaluate whether the codes and descriptions are clear and meaningful to users of the system.
18. Compatibility: Check the system's compatibility with the expectations and needs of the user in his task.

Figure 23 : Ergonomical principles

The integration of usability in the sphere of pedagogy is very controversial. According to most of the authors, usability is important and should be taken into consideration to develop educational software that is efficient, effective and gives satisfaction to the user (Carvalho, 2001). Nokelainen points out that in addition to the dialogue between a user and a system, “the pedagogical usability of a system or a learning material is also dependent on the goals set for a learning situation by the student and the teacher” (Nokelainen, 2006, p. 180).

However, usability is seen most of the time as in terms of operating systems and does not consider the achievement of educational goals. This can lead to usability features that do not realize educational purposes. “Just because an interface is easy to use, does not mean that it is designed appropriately from an educational perspective. There is an essential relationship between the two which must be addressed to ensure good educational software design” (Squires & Preece, 1996, p. 15).

In the literature, lots of authors have developed evaluations and recommendations about pedagogical usability. Those will be developed and analyses now in the following section.

Evaluation of educational software

In the evaluation process, there are essentially two types of review: formal and informal. (Duchastel, 1987) The informal review refers to a simple personal review of the developer/designer or the teacher. It is the simplest and quickest approach to software evaluation. The evaluator bases generally his appreciation on a personal try-out of the product and its experience with other similar software. This is quite subjective but can bring important information foreword for a timeliness review. The formal review considers more criteria developed by educational organizations or authors. It is important to mention that 'informal' does not refer to a lack of details from the evaluations, just as 'formal reviews' is not a synonym of perfect evaluation. The validity lies on the experience and the expertise of the evaluator.

Lansdale and Ormerod (1995) define three categories regarding the evaluations: 1/ standards, 2/ guidelines and checklists and 3/ models. Each of them are taken separately and developed for the case of educational software.

1/ Standards

Standards are guidelines that are discussed by specialized institutions and that turn out into norms. The most famous one's are the ISO norms developed by the International Organization of Standardization[12]. In our context of ergonomic evaluation of effective educational software design, we stay with that complementarity of two main areas: the computing field and the educational field. Since there are no standards nor specific ISO norm for this, we have selected the following ISO norms that are related to our research field and classified them according to four categories, corresponding to the composants of our research: software interface and usability, evaluation and quality of software, software development and learning quality management (see figure below).

---

[12] The International Organization of Standardization is an international non governmental association whose aim is to promote an international standardization with the intention to facilitate the changes of products and services between countries. The standards are defined by ISO technical specifications, which establish rules and criteria and therefore define characteristics to ensure that products, services or processes are suitable for use.

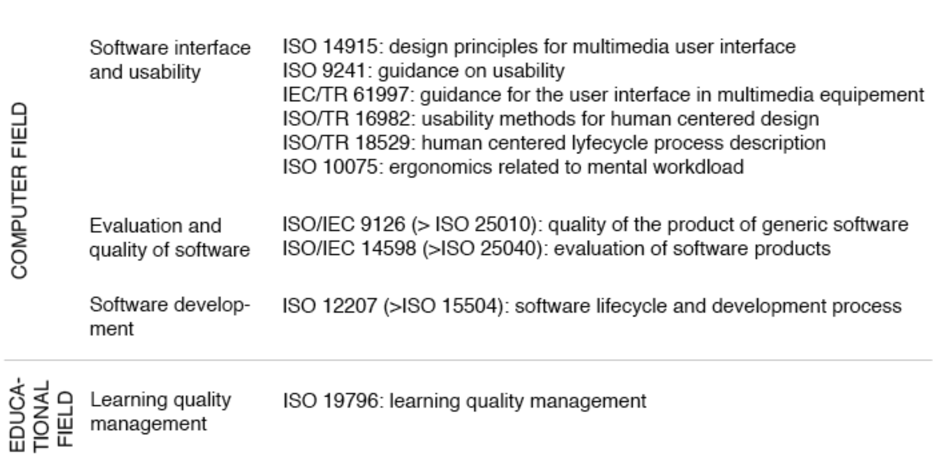


Figure 24 : ISO norms related to our research field

The following table resumes each ISO norms:

ISO norm	Description
The ISO 14915 (2012)	The ISO 14915 establishes design principles for multimedia user interfaces and provides a framework for handling the different considerations involved in their design. This norm gives requirements and recommendations for the ergonomic design of multimedia applications mainly intended for professional and vocational activities such as work or learning. It consists of the following parts: 1/ a general introduction to the standard; 2/ recommendations for navigation structures and aids, media controls, basic controls, media control guidelines for dynamic media; 3/ general guidelines for media selection and combination, 4/ computer based training, computer supported cooperative work, kiosk systems, on-line help and testing and evaluation.
The ISO 9241 (1998 & 2006)	The ISO 9241-11 has as aim to design and evaluate computers seeking usability and providing users to reach their goals and needs. This norm clarifies the benefits in terms of performance and user satisfaction. The norm is composed out of 17 parts. However the parts that are interesting for our anasynthesis are the following one's: part 10 on the dialogue principles, part 11 about the usability and part 12 on the presentation of the information.
The IEC/TR 61997 (2001)	The purpose of these guidelines is to take note of those inconveniences in the operation of multimedia equipment observed today, and to specify check-points that should be given primary consideration in the development of good multimedia products and systems that the general, non-professional user can use with confidence. This technical report gives general principles and detailed design guidance for media selection, and for mechanical, graphical and auditory user interfaces.
The ISO 16982 (2002)	This ISO norm provides information on human-centered usability methods, which can be used for design and evaluation. According to this norm, there are different methods: observations, measurements of the performances, critical incidents, questionnaires, interviews, loud thinking techniques, conception and collaborative evaluation, methods of creativity, methods based on the documents, approaches based on models, evaluations by experts and automatic evaluations. As well as factors that influence the choice of the method such as: process in the lifecycle, characteristics of the users, characteristics of the task, the product itself, constraints of the project, degree of expertise in ergonomics. It details the advantages, disadvantages and other factors relevant to using each usability method.
The ISO TR 18529 (2000)	This norm is intended to assist those involved in the design, use and assessment of lifecycle processes for systems, hardware and software. It presents a definition of the processes, which comprise a human-centered approach, and lists their components, outcomes and the information used and produced. The intention is to inform the users

	of process models who want to take account of human-centered processes in system, hardware and software lifecycles. It consists in 7 steps: 1/ ensure content in system strategy, 2/ plan and manage the process, 3/ specify the stakeholders and organizational requirements, 4/ understand and specify the context of use, 5/ produce design solutions, 5/ evaluate design against requirements and 7/ introduce and operate the system.
The ISO 10075 (2001)	This norm describes design principles necessary to avoid impairing effects and to improve working conditions. The standard is aimed at designers and manufacturers to make provisions for effective and efficient work and focus on tasks, equipment, environment, and organization, with a view to optimizing, rather than minimizing, mental workload, that is, avoiding the extremes of a workload that is either too heavy or too light. The charges are divided into informative charges (senses and perception), cognitive charges (think, learn, concentrate, memorize), and emotional charges (sensation, initiatives, feeling).
The ISO/IEC 9126 (1991 & 2001) and ISO/IEC 25010 (2011)	ISO/IEC 9126, replaced by the ISO/IEC 25010, is an international standard that refers to the software engineering and more specifically to the product quality. In other words the standards provide a framework for organizations to define a quality model for a software product. However, it leaves up to each organization the task of specifying precisely its own model.
The ISO/IEC 14598 (1999) and ISO/IEC 25040 (2011)	The ISO/IEC 14598, replaced by the ISO/IEC 25040, defines the requirements and recommendations for the practice implication of evaluation of software. Like a guide, it defines metrics of quality of software that can be used to evaluate products that are already produced or products that are in development.
The ISO 19796 (2005)	This norm consists in the harmonization of multiple approaches of quality used in the field of education and training. The objective is to help the responsible, representatives of quality, the developers of the system and the users, to develop their own system of quality. This norm is thus more a tool that gives a common language of quality, a model to help with the development and the amelioration of quality. This norm includes a model that takes different stages into consideration: 1/ the needs analysis, 2/ the framework analyses, 3/ the conception design, 4/ the development production, 5/ the implementation and 6/ the learning process realization.

Table 2 : The ISO norms and their description

After an in depth analyses of the different norms, we retain the important elements that contribute to the construction of our theoretical referential. From this analysis, we can retain the following important criteria concerning the software interface and usability:

Ergonomical criteria

Adaptation to the task
- understanding of the tasks that needs to be done
- the capacities, necessities and objectives are connected to the users
- there is a variability in the tasks

To be clear: the users need to understand the system easily
- not additional information
- help system available
- metaphors that facilitate the user
- adequate design of displays, signals and controls

To be controllable: the users can control the system
- the rapidity of the system is compatible with the capacities of the users
- the speed can be controlled by the users
- the system needs to give information on the users about the possible navigations

Respect of the expectations of the users
- give information on the type of task
- know the time that stays for a task

Tolerance of the mistakes
- request the confirmation of the user
- transparency in the information to the user

Ability to individualization
- individualization with the culture, experience,…

Self learning
- have mechanism of help
- promote actions of "learning by doing"

Presentation of the information
- clarity
- conciseness
- consistency
- readability
- comprehensibility

Figure 25 : Criteria concerning the software interface and usability

Moreover for the usability methods, it indicates the type of evaluation regarding the software process. For the acquisition and aprovisionnement, the best methods are the observations of the users and the methods based on the documents. For the development and analyses of the requirements: questionnaires / interviews and observations of the users. For the development and architectural conception: measurements related to the performance and methods of creativity. For the development and qualification test: measurements related to the performance and questionnaires / interviews, and finally, for the maintenance and operation: observation of the users and evaluations by experts. Most of those techniques are integrated in our methodology and developed in the third chapter of our work.

Concerning the workload, the related ISO's norms confirm the cognitive overload developed in the first part of the theory by paying attention to the ambiguity of the tasks, the requirements, the adequacy of the information, the ambiguity of the information, the long-term memory load, the probability of the errors, etc. Those are developed by Mayer and Clark (2011) that adapt those criteria's with concrete guidelines for multimedia products (see part 1 of our literature).

The development of the process of software is also explicated in the ISO norms. According to the norms, the lifecycle of a software includes 5 steps: 1/ the acquisition phase with the activities involved in initiating a process (initiation, request for proposal preparation, contract, negotiation of the changes, update of the contract, acceptance and completion); 2/ the supply phase that develops the project management plan with information about the project such as different milestones that need to be reached; 3/ the development that integrates the design, the creation and the testing of the product; 4/ the operation, witch includes the activities like assisting users in working with the created software product; and finally 5/ the maintenance-tasks to keep the product up and running. It can be observed that those steps are similar to the evolutionary process flow mentioned above.

Regarding the 'quality' aspect of software, the ISO norms describe the following characteristics: functionality, reliability, usability, efficiency, maintainability and portability. Those have already been developed in the part of the literature regarding the term 'quality'.

Finally, the last information that comes out the analyses of the different ISO's is the requirements and recommendations for the evaluation of software products. The model contains 5 steps: establish the evaluation requirements, specify the evaluation, design the evaluation, execute the evaluation and conclude the evaluation. The parallel can be made with the anasynthesis methodology we are using in the creation of our evaluation tool.

2/ Guidelines and checklists of educational software's evaluation

Guidelines and checklists are another way to evaluation educational software. Guidelines are recommendations to orient the design of the evaluation of certain aspects of the interface. These guidelines are advisory and offer information for improve the software, that the evaluator can choose to adopt it or not. They do not constitute a complete analysis, but rely on guiding informations.

Checklists are based on guidelines and standards and consist in a list of questions grouped into categories, whose answers give an indication of the quality of the software. Those checklists mostly allow to question and identify the quality, problems and limitations of the software in question. It is a type of analyses of the different aspects that needs to be considered in the evaluation of the quality of the product. The components of a checklist differ from one author to another. This technique has the advantage to be of lower cost and a quiet rapid method. Very popular in the early use of educational software, they are now arousing polemics. Indeed, authors like Squires and Preece (1999) identify problems in the use of checklist, illustrating that "the failure of the checklist approach is to address the integration of usability and education issues" (Squires & Preece, 1999, p. 471).

Lots of checklists and guidelines can be founded in the literature proposing an evaluation system for educational software. In this work, we will refer to 15 of them, the most recurrent in the literature. They are the following:

Name (date)	Description
MicroSIFT (1982)	The MicroSIFT checklist integrates both content of instructional quality concerned with educational issues, as well as a technical quality section with usability concerns. The MicroSIFT checklist is composed out of two different sections: 1/ In the first section contains information such as: - objective information: the title, the version, the producer and the required hardware and software. - subjective information: the instructional purpose, instructional techniques, estimated time of the interaction of the students,... 2/ the second section contains a list of 21 items about content, instructional quality and technical quality. The evaluations are asked to score each item on a scale of strongly agree, agree, disagree, strongly disagree or not applicable.
Reeves & Harmon (1993)	The authors propose criteria for educational software that take into consideration as well pedagogical criteria as interface criteria. They propose two types of criteria: pedagogical criteria (14) and interface criteria (10).
Niquini (1996)	Niquini proposes guidelines for educational software and focuses primary on the pedagogical criteria. It is categorized based on the objectives, the content, the didactic, the capacity of interaction, the presentation of the content.
Coburn (1998)	Coburn realizes a list of questions in order to evaluate an educational software during its production process. The questions are related to the following categories: Objectives, content, presentation of that content, didactic, the capacity of interaction and process.

Gamez (1998)	This Técnica de Inspeção Ergonômica de Software Educacional (TICESE) focuses mostly on the quality of the interface and proposes an orientation that helps the responsible to integrate the materials in the educative programs. It is based on criteria of cognitive aspects, ergonomics, psychology and pedagogy. The method consists in a checklist to guide the evaluator. For the ergonomical part, it is inspired on the criteria of Scapin and Bastien, 1993.
Silva (1999)	Silva proposes a checklist that integrates the teaching and ergonomics in the same tool. It is design so that the evaluator has an overview of the elements that should be observe, facilitating his research on these points and is applicable.
Squires & Preece (1999)	The authors propose guidelines for the teachers in order that they evaluate educational software before their use in the classroom. It is a predictive evaluation based on a socio-constructivist learning approach. Those guidelines arise from the usability heuristics (Nielsen, 1994) and the learning evaluations heuristics based on the socio-constructivist learning approach (Atkins, 1993 / Soloway et al., 1996).
Crozat, Trigano & Hû (1999)	The authors propose the EMPI Guidelines: Evaluation des logiciels Multimedia Pédagogiques Interactifs. This method is aimed to evaluate the educational software so the professors can choose the most adapted to their pedagogical objectives. It is based on the authors such as Scapin 86, Senach 90, Bastien & Scapin 94, Bastien & Scapin 97.
Gladcheff (2001)	This author proposes on evaluation system in form of questions that is aimed to check how value can be added in the learning environment in the area of mathematics. It consist in technical aspects as well as aspects of the area of education.
Campos & Campos (2001)	Campos and Campos define that an educational software needs to take into consideration elements of pedagogy as well as elements of the field of ergonomics. For this reason, they define the following categories of evaluation that an educational software should have: pedagogical, facility of use, adaptability, documentation, portability, ROI.
Oliveira (2001)	The author proposes an evaluation of educational software, in form of lists of evaluations that allows the theoretical-methodological orientation in the evaluation and use of these software. The categories are: the content of the software, portability, maintainability, usability, presentation and efficiency.
Gomez, and al. (2002)	The authors propose a checklist for evaluating software, which privileges some teaching and learning aspects to allows teachers to choose and use the software programs considering their possibilities and limitations.
Plaza, and al. (2009)	The authors propose a model of evaluation of educational software based on the characteristics and sub characteristics of the ISO/IEC 9126 family and extends them with the following dimensions: technical aspects, organizational aspects, content aspects and educational aspects.
Da Silva, and al. (2011)	The authors discusses the development of criteria for evaluating educational software that considers technical and specific teaching of reading and comprehension for the learning of the lecture and textual comprehension. It is based on Oliveira and al (2001) and Atayde (2003) for the interaction of student-software-professor, pedagogical fundamentals, content and programing. The other criteria are based on the literature of the use of software in education: Dall'asta (2004), Silva (2009) e Webber (2009).
Cenci & Bonelli (2012)	Criteria's for professors of mathematics to choose a software well while taking into consideration technical and pedagogical aspects. The authors create their checklist based on the diverse criteria of Gomes (2002), Wolff (2008), Vieira (2011), Campos (2001), Silva (2011), for the pedagogical as well as the technical aspects.

Table 3 : Our 15 selected evaluation systems

In this analysis, we have taken the checklists as well as the guidelines. Indeed, they do not differ a lot regarding the content, as one will present the criteria in the form of questions, others the same criteria in a descriptive way. Moreover, it is important to mention that we also took into consideration the checklists and guidelines intended for the teachers. Even if our referential will focus on a predictive evaluation, the selection's criteria of the targeted group is for us an important source of information. The teachers / parents are the one that will select the software one's it is commercialized, so considering their criteria into the previous step of the software process before the release is precious.

The analysis of the 15 checklists and guidelines, gives the following results:

Criteria of the different checklists/guidelines	MicroSIFT (1982)	Reeves & Harmon (1993)	Niquini (1996)	Coburn (1998)	Gamez (1998)	Silva (1999)	Squires & Preece (1999)	Crozat, Trigano & Hû (1999)	Gladcheff (2001)	Campos & Campos (2001)	Oliveira (2001)	Gomez et al. (2002)	Plaza et al. (2009)	Da Silva et al. (2011)	Cenci & Bonelli (2012)
GENERAL IMPRESSIONS															
Reassuring	—	—	—	—	—	—	—	yes	—	—	—	—	—	—	—
innovating	—	—	—	yes	—	—	—	yes	—	—	—	—	—	—	—
easy of use	—	yes	—	yes	—	—	—	yes	yes	yes	—	—	—	—	yes
affordable price	—	—	—	—	—	—	—	—	—	yes	—	—	—	—	—
TECHNICAL CRITERIA															
compatibility of the software	—	yes	—	—	yes	yes	—	yes	—	yes	yes	—	yes	yes	yes
easy installation of the software	—	—	—	—	—	—	—	yes	yes	—	yes	—	—	yes	—
errors recognition, diagnosis and recovery cycle	—	—	—	yes	yes	yes	yes	yes	—	yes	yes	—	—	—	yes
presence of the modalities of utilization	—	—	yes	—	yes	yes	—	yes	—	—	—	—	—	—	—
easy and well explained connection	—	—	—	—	—	—	—	yes	—	—	—	—	—	—	—
possibility to manipulate a lot of data	—	—	yes	—	—	—	—	—	—	—	—	—	—	—	—
maintainability of the product	—	—	—	—	—	—	—	—	yes	—	—	—	—	—	—
presence of an analysis of the results	—	—	—	yes	—	—	—	—	—	—	—	—	—	—	—
USABILITY CRITERIA															
orientation of the user	—	—	—	—	yes	—	—	yes	—	—	—	—	yes	—	—
manageable cognitive load	—	yes	—	—	yes	yes	—	yes	—	—	—	—	—	—	—
minimal actions	—	—	—	—	—	—	—	yes	—	—	—	—	—	—	—
unrestricted learner control	yes	yes	—	—	yes	yes	yes	yes	—	yes	yes	—	yes	yes	—
homogeneity of the presentation of the content and the functions	—	—	yes	—	yes	—	—	yes	yes	—	—	—	yes	—	—
adaptability with the user	—	—	—	—	yes	yes	—	yes	—	—	—	—	yes	—	—
pleasing aesthetics	—	yes	yes	—	—	—	—	yes	—	—	—	—	—	—	—
follows principles of screen design	—	yes	—	—	—	—	—	—	—	—	—	—	—	—	—
capacity of interaction	—	—	yes	yes	—	—	—	—	—	—	yes	—	yes	yes	—
feedback is effectively employed	yes	—	—	yes	—	—	—	—	—	—	yes	—	—	yes	yes
the software can be abandoned at any time without loss of data	—	—	—	—	—	—	—	—	—	—	—	—	—	yes	yes
MULTIMEDIA CRITERIA															
the edition tools are instinctive	—	—	—	—	—	—	—	—	—	yes	—	—	—	—	—
content and language adapted to the public	yes	—	—	—	—	—	—	yes	yes	—	yes	yes	—	—	yes
clear typography	—	—	—	—	—	—	—	yes	—	—	—	—	—	—	—
clean and coherent layout and design	—	—	—	yes	—	—	—	yes	—	—	yes	—	—	—	yes
images, photos, drawings, videos are legal and appropriated	yes	—	—	yes	—	—	—	yes	yes	—	yes	yes	—	yes	yes
the sounds are audible	—	—	—	—	—	—	—	yes	—	—	yes	—	—	—	—
avoiding the redundancy	—	—	—	—	—	—	—	yes	yes	—	yes	—	yes	—	—
coordinated media integration	—	yes	—	—	—	—	—	—	—	—	—	—	—	—	yes
SCENARIZATION CRITERIA															

clear structure	—	—	—	—	—	—	—	yes	—	—	—	—	—	—	—
navigation fidelity	—	yes	—	—	—	—	yes	yes	—	yes	—	—	yes	—	—
elements of narration in the software	—	—	—	—	—	—	—	yes	—	—	—	—	—	—	—
presence of links that open to other information	—	—	—	—	—	—	—	—	—	—	yes	—	—	—	—
DIDACTIC CRITERIA															
approaches to learning /learnability	—	—	—	—	—	—	yes	—	—	—	—	—	yes	yes	yes
match with the curriculum	—	—	yes	—	—	yes	yes	—	—	—	yes	—	—	yes	yes
clear defined and communicated objectives	yes	—	yes	yes	—	—	—	yes	yes	yes	yes	yes	—	yes	yes
presence of an evaluation (type, criteria, positioning,..)	—	—	—	—	yes	—	—	yes	—	—	yes	—	—	—	yes
facilitative teachers' role	—	yes	—	—	—	—	—	yes	—	—	—	—	—	—	—
learning from the errors	—	yes	—	—	—	—	—	—	—	—	yes	yes	—	—	—
intrinsic motivation	—	yes	—	—	—	—	—	yes	—	—	—	—	—	yes	—
integral cooperative learning	—	yes	—	—	—	—	—	—	—	—	yes	—	—	yes	—
validity of the content (clear and complete information)	—	yes	yes	—	yes	—	—	yes	—	—	—	—	—	—	—
comprehension of the content	yes	—	—	—	yes	—	—	—	yes	—	—	yes	—	—	—
personalization (information, settings, adaptation,…)	—	—	yes	yes	—	—	—	yes	—	—	—	—	—	—	—
resolution of the activity	—	—	—	—	—	—	—	yes	—	—	—	—	—	—	—
creation of activities	—	—	—	—	—	—	—	yes	—	—	—	—	—	—	—
resources for help	—	—	—	—	—	—	—	yes	yes	—	—	—	—	—	yes
level of difficulty appropriated to the target audience	yes	—	—	—	—	—	—	—	—	—	—	—	—	—	—
comprehensible without the intervention of an instructor	—	—	—	yes	—	—	—	—	yes	—	—	yes	—	—	—
interdisciplinary perspective	—	—	—	—	—	—	—	—	—	—	yes	—	—	yes	—
different levels of complexity	—	—	—	yes	—	—	—	—	—	—	yes	—	—	yes	—
promotion of creativity	—	—	—	—	—	—	—	—	—	—	—	—	—	—	yes

Table 4 : Analysis of the criteria according to the 15 selected evaluation models

From those results, we can observe that in all 15 evaluations, there are criteria concerning the technique and usability, as well as elements of pedagogy.

This classification of the criteria allows us also to highlight the most frequent and important criteria. The most recurrent criteria are: unrestricted learning control (10/15), clear defined and communicated objectives (10/15), compatibility of the software (9/15), errors recognition, diagnosis and recovery cycle (8/15), appropriation of the multimedia elements (8/15), match with the curriculum (6/15), ease of use (6/15) language and content adapted to the public (6/15) as well as the homogeneity, interaction, feedback and navigation facility with respectively 5 evaluations on 15. However, when we look at the recurrence of the criteria, only two are part of the pedagogical criteria.

Moreover, it is a way to compare the different evaluations systems and bring out their different perspectives. Galvis-Panqueva (1997) points out that an evaluation can highlight more one aspect of the software than another, depending on the type of evaluation and the nature of the software. In other words, different evaluations tests can have different perspectives according to the dimensions of the system. It can be more centered on the artifact and the design, or centered on the learning process and the tasks, or finally centered on the social context and the interaction between the user and the interface.

On the other hand, it makes us found out the difficulty of the selection of the criteria. Those results will be taken into consideration in the next phase of our anasynthesis.

3/ Models of educational software evaluation

Rare are the models of educational software's evaluation in the literature. Only few authors propose methods for the evaluation of educational software. It is the case of Puustinen and al. (2006) that integrate the evaluation criteria into the development of the educational software. Their framework is based on the idea that any educational software should be designed on the basis of a coherent set of relations between the tools that learners have at their disposal, the actions that can be carried out with the tools, the learner's characteristics and the pedagogical goals that can be achieved as a result in a given situation. This GESTALT model is then interrelated to the elements of evaluation of coherence, compatibility, completeness and relevance related to their viewpoint. It shows the important of the criteria of evaluation within the model.

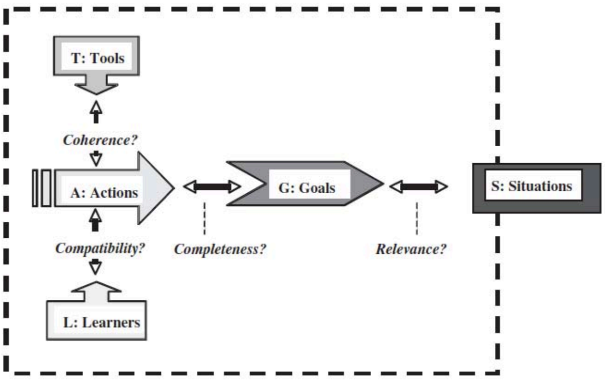


Figure 26 : GESTALT model according to Puustinen and al. (2006)

Another model is model of McDougal and Squires (1994) that is aimed to facilitate the problems of checklists. It takes into account the three important actors: students, teachers and designers. Based on their interactions, they propose specific points of interest for the evaluation of the educational software. The teacher and students perspectives interaction focuses on the integration of the previous student's experiences, the interaction and the responsibility of learning through collaboration, responsibility and autonomy. The designer and students perspective interaction includes whether it is possible to identify the underlying theory of learning, the appropriation of it, the integration of the design, the extent of the learner control and the complexity of the material presented. Finally, the designer and teacher perspectives interaction verifies the relationship of the software to the curriculum.

Atayde and al. (2003) developed the MAQSEI (method of evaluation of the quality of infantile educational software). This very concrete model is constructed in 4 phases: 1/ the phase of recognition and proposition of evaluation, 2/ the phase of planning of the tests, 3/ realization of the tests, 4/ phase of analysis of the data and production of the outcomes. It is this model that is applied in our research.

However, whatever the evaluation model, the objective of it is to guarantee the reproductivity, impartiality, objectiveness and the repetitiveness like emphasizes Rouller (cited in De Alencar, 2007).

After a formal or informal evaluation, a more detailed review can be carried out, applying the educational software on the field to the members of the intended audience. The favorable methods are observations, interviews/questionnaires and satisfaction evaluation (Preece, 1993). Those are described more in detail in our methodology, section 3 of our research.

Conclusion

In this third part, we have focused on the area of engineering and ergonomics through the discussion of concepts like effectiveness and quality of software design, the software processing, the usability and the evaluation of educational software through standards, checklists and models as well as models. Indeed, cognitive ergonomics properly applied optimizes the performance and effectiveness of the system, while concerned with “mental processes such as perception, memory, reasoning and motor response” (IEA, 2014). Applying those principles to the educational aspects can influence on several aspects with the user such as interaction and another construction of knowledge (Alessi & Trollip, 2001).

Through this part, we can affirm our accomplishment of the new objectives. **Gather the existent pedagogical usability evaluations** (objective 4) has been realized through a selection of 15 educational software checklists or guidelines. Given the large number of pedagogical usability evaluations, a selection was needed and has been done on their frequency of appearance in the literature. A resume of the evolutions can be found in Table 1. Another objective that we realize in this part is the fifth one: **Retain the pedagogical usability criteria that arise the most.** Indeed, based on the selection of the 15 evaluations, we have compared their different evaluation criteria and compared the perspectives between them. This can be seen in Table 3.

While making a synthesis of the different informations of this chapter, we could include the following criteria in our referential :

Summary of principles regarding our third part: Ergonomics

Technical criteria :
- the software needs to be compatible with the hardware of the users
- the software needs to be easy to install
- the software recognizes errors, diagnosis and recovers the process as well as prevents the user to make errors
- the error messages are clear
- the errors can be corrected easily
- the modalities of use are explained
- the software is able to manipulate the data that it requires, even in great quantity
- the software will be maintained after its release
- the data generated through the software can be analyzed
- there is a transparency of the information to the user

Usability criteria :
- the cognitive load is manageable
- the software is taught in terms of minimal actions
- the learner has a control on the system and the actions (ex : speed)
- the software adapts itself to the user
- there is a capacity of interaction with the system
- feedback is effectively employed
- the software saves the data
- there are help resources
- the software is innovating and has an added value
- the software is easy to use
- the system guides and encourages the user
- there is the politic of minimal action to achieve the user's goal

Scenarization criteria :
- the structure of the software is clear and quickly understandable
- the navigation is coherent and contents fidelity
- the software has links to external additional information

Didactical criteria:
- there is a match with the curriculum of the targeted public (comprehensible)
- there is a learning approach
- there are clear defined and communicated objectives
- there is a presence of an evaluation
- the user learns from its errors
- the software contents an intrinsic motivation
- it integrates a cooperative learning
- there is a possible personalization (information, settings, adaptation)
- the results of the activities are available
- activities can be created easily
- the level of difficulty is appropriated to the target audience
- the software can be used without the intervention of an instructor
- the software has an interdisciplinary perspective
- it contains different levels of complexity
- it promotes creativity
- users with different levels of experience have equal opportunities to succeed in their goals
- there is a variability in the tasks
- promotes actions of "learning by doing"

Presentation of the information:
- the edition tools are instinctive
- the language is adapted to the public
- the content is clear, consistent concise and comprehensible
- the images, photos, drawings, videos, are appropriated and their use adds a value
- the multimedia elements used are legal
- the sounds are audible
- the multimedia content avoids redundancy with other content
- the integration of the multimedia content is coordinated

Layout :
- the typography is clear
- there is a pleasing design and follows the principles of screen design
- the design is coherent on the different levels of architecture of the software
- there is a homogeneity of the presentation of the content and the functions
- the actions are explicit
- the signals and controls are adequate to the design

Figure 27 : Summary of principles regarding our third part, ergonomics

PART 4: PETESE, OUR PROPOSITION

We enter in the fourth and last part of our literature review, which also corresponds to the synthesis part of our methodology. Indeed, the objective is to gather the conclusions of the three previous parts and to accomplish our next objective: **developing an evaluation referential based on the above selected informations.**

We have decided to call our tool as PETESE: Pedagogical Ergonomical Tool for Educational Software Evaluation. This referential of criteria is thus an answer to the need of the literature that does not propose any predictive tool of evaluation of educational software in the field of mathematics, that integrates the requirements of the field of education, mathematics and ergonomics in order to help the instructional designers before the launching of their educational software on the market.

The different criteria in our referential have been taken from the criteria mentioned in the previous parts of the literature. This can be seen in the following image:

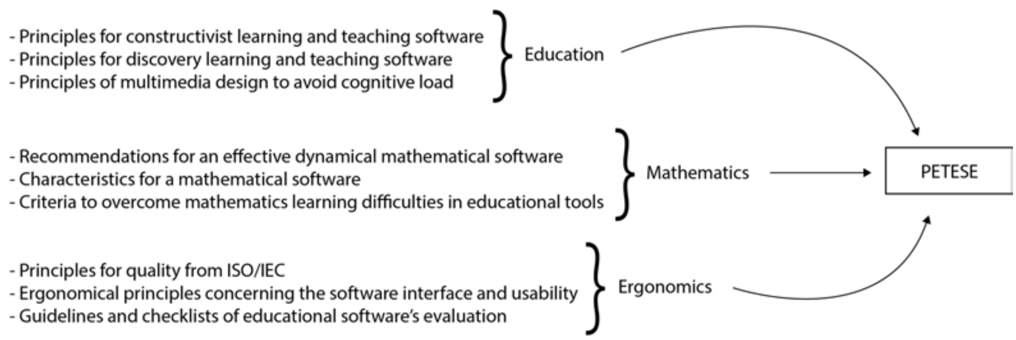


Figure 28 : Composition of PETESE, our pedagogical ergonomical tool of educational software evaluation

To classify the different criteria, we have decided to use the classification system of Gamez (1998), where the author orders the criteria according the following aspects: criteria's related to the technical aspects of the construction of the system, to a pedagogical character, to the interface of the product, to the content of the software and to usability aspects.

PETESE: A REFERENCIAL FOR A PREDICTIVE EVALUATION OF EFFECTIVE DISCOVERY LEARNING MATHEMATICAL SOFTWARE
1. GENERAL ASPECTS
- The software is innovating and has an added value
- The software is easy to use
- The software has an interdisciplinary perspective
- The software contains different levels of complexity
- There is a learning approach that matches with the curriculum of the targeted public
2. PEDAGOGICAL ASPECTS (constructivism/mathematics)
2.1 Educational objectives
- There are clear defined and communicated educational objectives
- The construction takes place in an individual context but integrates elements of cooperation and collaboration
- It promotes creativity
- Exploration is the favored approach and encourages students to seek knowledge independently
- The software encourages self-analyses, self-regulation, self-reflection and self-awareness
2.2 Activities & Tasks
- The activities work the necessary aspects to meet the proposed mathematical objectives
- Tasks are relevant, relisting, authentic and represent the natural complexities of the real world, so close as possible of the targeted public
- The level of difficulty is appropriated to the public and scaffolding facilitates to perform just beyond the limits of its ability
- There is a variability in the tasks
- The results of the activities are available
- Activities can be created easily and customized
- The tasks are problem-solving and develop higher order-thinking skills through situations that enlarge the competences and the concepts
2.3 Role of the teacher
- The software can be used without the intervention of an instructor
- Teachers serve in the role of guides, monitors, coaches, tutors or facilitators
2.4 Evaluation
- The software recognizes and supports the important role of the teacher
- There is a presence of evaluation
- The evaluation is authentic and interwoven with teaching
3. USABILITY ASPECTS
3.1 Guidance & Instructions
- The system guides and encourages the user
- There are help resources
- Instructions are clear and easily available
3.2 Feedback & Motivation
- Feedback is effectively employed
- The software contents an intrinsic motivation
- The software encourages the users in their positive actions
- The feedback is positive even if the user fails
- The feedback allows the learner to reflect on his errors
- The software offers a feedback of the student progress
3.3 Interaction (system & users)
- There is a capacity of interaction with the system
- The software has a section where interaction like discussion and collaboration can take place
3.4 Error prevention
- The software recognizes errors and recovers the process
- It prevents the user to make errors
- The error messages are clear
- The user learns from its errors
3.5 Control & Personalization

- The learner has a control on the system and the actions
- There is a possible personalization of the settings or informations
4. TECHNICAL ASPECTS
4.1 Compatibility
- The software needs to be compatible with the hardware of the users
- The conditions of compatibility are clearly explained
4.2 Installation
- The software needs to be easy to install
- The modality of use are explained
4.3 Data management
- The software is able to manipulate the date that it requires even in great quantity
- The data generated through the software can be analyzed
- There is a transparency of the information to the user
- The software saves the data of the user
4.4 Maintenance
- The software will be maintained after its release
- Updates will take place to keep the software up to date
5. CONTENT
5.1 Language
- The language is adapted to the public
- The mathematical terms are correct and also adapted to the public
- The content is clear, consistent, concise and comprehensible
5.2 Multimedia elements (images/photos/drawings/videos/sound/…)
- Those elements are appropriated, add a value and avoid redundancy with the other contents
- Multimedia elements are used to facilitate the comprehension
- If they come from another source, they are used legally
- They are from good quality (sound audible, images visible,…)
- The multimedia explanations present the elements contiguously rather than presented separately
5.3 Organization
- The content is organized to avoid cognitive load
- The content is reachable in terms of minimal actions
- The content is broken down in small topics that can be accessed easier
- The content can be reached via multiple paths
- The software has links to external additional information
6. INTERFACE
6.1 Navigation
- The structure of the software is clear and quickly understandable
- The navigation is coherent
6.2 Layout
- The typography is clear and adapted to its support
- There is a pleasing design that is coherent on the different levels of architecture of the software
- The actions are explicit
- Icons are representative
- The colors are meaningful and help in the comprehension of the software

Table 5 : Our referential for a predictive evaluation of effective discovery learning mathematical software

3 Methodology And Research Material

Summary

The methodology used to develop a pedagogical usability referential for mathematical software of discovering learning is based on the anasynthesis. However, to test this referential we use the methodology of the case study, and apply it to GGBook. In this chapter, both methodologies will be explained as well as the reasons for our choices. The type of case and the way the data is collected and reported will also be discussed.

Research Techniques:
Anasynthesis & Case Study

Anasynthesis

The anasynthesis, like its name indicates, is formed by the words 'analyze' and 'synthesis' that designate the general process of elaboration of a model, referential or system (Sauvé, 1992). Used in various areas, this method is inspired by the works of Silvern (1980). This method constitutes the theoretical part of our research.

This method has been criticized by some authors with the argument of bringing a diversity of models based on subjectivity (Landry & Auger, 2003). However, the aim of our research is just to offer guidelines through a referential to the instructional designers and developers of educational software of mathematics based on discovery learning. Moreover, D'Amboise and Audet (1996) specify that the notion of objectivity is considered as a myth, the neutrality of the researchers and the processus being relative.

Legendre (2005) defines the anasynthesis according to four steps:

1. The identification: This corresponds to the situation of departure of the research with the construction of the problematic of the research and the formulation of the problem. What concerns the evaluation of educational software we note that there is an abundance of different evaluations materials like checklists, guidelines and models for more general educational software. However, there is an absence of specific evaluation material for mathematical software of discovering learning helping the developers before the release of the software. For this reason, we identify the elements in the theory that help us constructing this referential in the field of education, mathematics and ergonomics.

2. The analyses: This phase corresponds to the analyses of the theoretical data in order to build the referential. This is done through a critical analyze of the content gathered in phase 1. The important criteria of each field are highlighted and gathered in the conclusion part of each of the three theoretical parts.

3. The syntheses: The third phase makes the syntheses of the information of the previous step. The aim is to identify the relevant and coherent elements as well as gather the different elements together in order to produce a first theoretical proposition of referential. It corresponds to the fourth part in the literature part of our work.

4. The validation: The validation is the testing part of the theoretical work produced in phase 3 through a concrete application in a specific case. It is important to mention that the validation does not include a definitive end of the referential's prototype. Indeed, the process of retroaction is present in each stage of the anasynthesis in a perspective of amelioration of the referential.

Visually, applied to our research, the anasynthesis corresponds to the following figure:

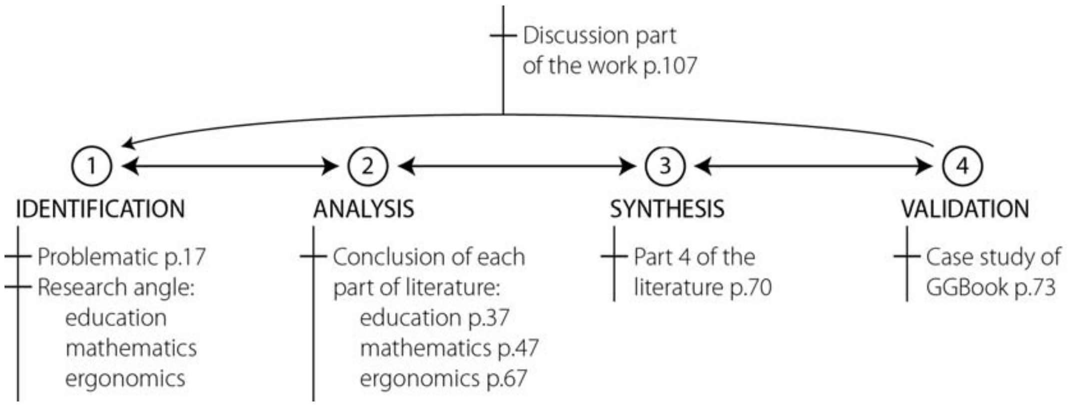


Figure 29 : The anasynthesis applied to our research

This research method, mainly based on theoretical research, presents thus a dominant inductive demarche that manifests through the progressive and iterative construction of the unities that help building the final referential.

The results of this anasynthesis have been published in our article (Coomans & Lacerda, 2015).

Case study

For the fourth step of the anasynthesis, or the evaluation of the built referential, we have chosen to base our approach on the case study method. Indeed, the empirical part of our work is aimed to validate the significance of our theoretical built referential, but also to evaluate precise educational software GGBook.

Thomas (2011) gives the following definition of case study: "Case studies are analyses of persons, events, decisions, periods, projects, policies, institutions, or other systems that are studied holistically by one or more methods. The case that is the subject of the inquiry will be an instance of a class of phenomena that provides an analytical frame within which the study is conducted and which the case illuminates and explicates". In fact, this type of research is thus mostly to answer the questions ‘how’ and ‘why’ on the basis of a concrete example (Yin, 2009).

Focusing on precise cases allows a lot more details to be collected that are richer and of greater depth than what can be obtained by other research methods. The vision of the question is then much more complete and can be seen in a whole and dependent on its context. Indeed, the case is studied in its dimension of time and space. Yin (2009) uses the word “real-time context” to define the investigation and analyze of the case study. It is also a research method that allows flexibility and adaptation in actual time as well as a high crossing checking of the data due to the different sources of information. Researchers can present data from multiple methods, which is also a form of triangulation (Eurome Aid Evaluation Methodology, 2013).

The main critique done to a case study is its non-character of generalization. Based on an example, the conclusions cannot be expended to the general population. To this argument, authors like Flyvbjerg (2001) defend that context dependent knowledge is sometimes more valuable than the search for predictive theories and universals. However, the fact that case study research is not as systematic in its data collection, many perceive it as unscientific. This perceived lack of rigor makes the case study unbiased in its findings and conclusions.

Between the pros and cons, the importance stays the signification put behind. A good example is given by the researcher Hans Eysenck (1976), “who originally did not regard the case study as anything other than a method of producing anecdotes, later realized that sometimes we simply have to keep our eyes open and loop carefully at individual cases, not in the hope of proving anything, but rather in the hope of learning something”.

According to the definition of the case study above, it is the most appropriated methodology to study the application of the pedagogical usability referential. With the question of "how" to evaluate an educational software, the aim is not superficially to a large sample of participants to generalize certain aspects, but rather to take precise cases and to have a deep and detailed knowledge of the application on a particular software. Working in depth on one case, here the educational software GGBook, allows understanding the phenomenon as a whole. More than just criteria's, this methodology enables a complete vision from different angles.

From all the different types of case studies, this research can be defined as being descriptive. The subject is observed and the information gathered is then compared to the pre-existing theory. This descriptive case is based on a single case design, more specifically based on the rational of the single case as typical representative case. As Yin defines: "the objective is to capture the circumstances and conditions of an everyday or commonplace situation" (Yin, 2009, p. 41).

Yin (2009) defines six sources of evidence in collecting the data: documentation, archival records, interviews, direct observations, participant-observation, and physical artifacts. For this research, most of those sources are used. In a chronological order, those are the following methods that are used in our research.

1. Description of GGBook through content analyses:

The documentary information is generally present in each case study. For the educational software GGBook, it consists in the analysis of the interface and mechanisms of the software. Through this research, an explanation and description of the software has been done by observation of the software and the given information.

2. The expert's vision through heuristic evaluation:

An expert evaluation, also known as heuristic evaluation or usability review, is an evaluation of a user interface regarding the accepted pedagogical usability best practices. Our elaborated referential in the theoretical part is applied here. In other words, this evaluation corresponds to the validation step of the anasynthesis. Having experts' points of views is also from high contribution to the analyses of GGBook and will help to obtain a clear feedback with corrective measures. Even thought this evaluation may highlight more minor issues and fewer major issues, it will be completed with the other mentioned tests.

For our research, six experts have been chosen to aggregate the results and obtain an effective expert evaluation. It consists in the members of the development team of GGBook, that to say, the coordinator, a professor of mathematics, a usability expert, a technical expert, a content expert and a designer.

The expert received an Excel sheet with 3 steps. First, they were asked to indicate if the criteria mentioned of the PETESE was present or absent regarding to them in the prototype of the educational software GGBook. Three options of answers were possible: present, absent of I don't know / not applied. In the second step, they were invited to make remarks, give recommendations, suggestions, improvements and comments on their choices. It was asked to minimum write something when they mentioned "absent" in the previous step.

REFERENTIAL FOR A PREDICTIVE EVALUATION OF EFFECTIVE DISCOVERY LEARNING MATHEMATICAL SOFTWARE	APPLICATION OF THE REFERENTIAL TO GGBOOK			REMARKS, RECOMMENDATIONS, SUGGESTIONS, AMELIORATIONS, COMMENTS,... *
	Present	I don't know	Absent	* Respond the minimum when the answer is 'absent' (This can be in portuguese as well)
1. GENERAL ASPECTS	1. GENERAL ASPECTS			1. GENERAL ASPECTS
- The software is innovating and has an added value				
- The software is easy to use				
- The software has an interdisciplinary perspective				
- The software contains different levels of complexity				
- There is a learning approach that matches with the curriculum of the targeted public				
2. PEDAGOGICAL ASPECTS (constructivism/mathematics)	2. PEDAGOGICAL ASPECTS			2. PEDAGOGICAL ASPECTS
2.1 Educational objectives	2.1 Educational objectives			2.1 Educational objectives
- There are clear defined and communicated educational objectives				
- The construction takes place in an individual context but integrates elements of cooperation and collaboration				
- It promotes creativity				

Figure 30 : Extract of the Excel sheet for the experts evaluation (step 1 and 2)

Finally, the experts received questions on the PETESE itself, the objective being that every expert focuses on the questions of its area. The questions were the following:

1. *According to your expertee, do the criteria from the literature represent well the different aspects to take into consideration during the development of a mathematical software?*
2. *Are there important aspects that are not present in the criteria and should be added because of their importance during the development?*
3. *Are there criteria presented in the referential that, according to your expertee, are not important to be mentioned in the referential for a predictive evaluation?*

The results are showed and discussed in the following sections of our work.

3. User's vision through a questionnaire:

The aim of this part is to collect the user's vision on the educational software GGBook. In order to do this, we have built a questionnaire based on the criteria of the PETESE, the same as used in the expert's part. However, to make it more practice and attractive to use as the Excel sheet, we have developed an online questionnaire on "Typeform"[13]. It can be founded online at the following address: https://stefany4.typeform.com/to/HPQtUt The design of our questionnaire is as shown in Figure 31.

[13] Typeform is a free survey builder that makes asking questions easy and human on any device.

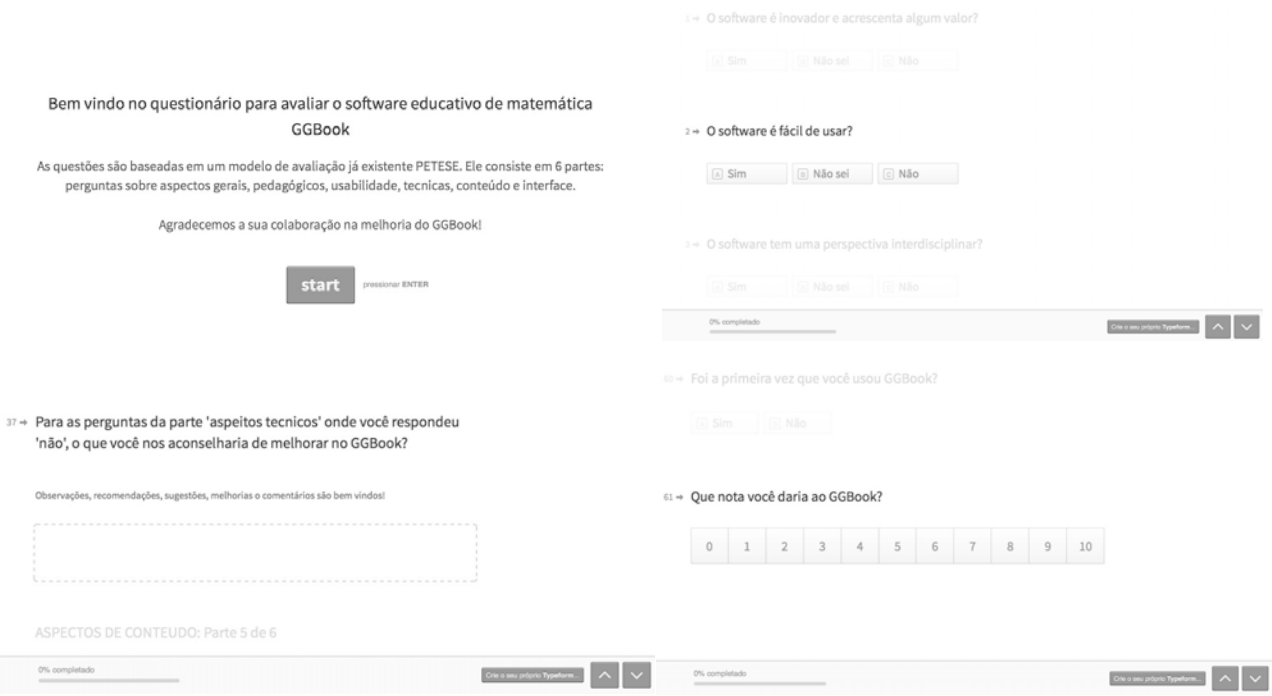


Figure 31 : Extract of the questionnaire for the user's evaluation (step 3)

As the targeted users were professors of the basic education of the public system of the federal district of Brazil, the questions have been asked in Portuguese. Those professors had already experienced the software in a previous experience. It was thus a way to have their satisfaction about the prototype.

After an introduction word, 61 questions were asked. There were three kinds of questions: questions of answers with 'yes / I don't know / no' (principally for the criteria), open questions for long text answers (space for recommendations) as well as a rating question (see Figure 31).

<u>Conclusion:</u>

In conclusion, through those different steps, two types of evaluators can be defined: the experts and the users. If the expert evaluation involves a review of an educational software according to accepted usability principles, the users will bring other elements forewards such as their satisfaction with the system. Those different points of view contribute to the in depth analyses of our case-study.

Moreover, timeliness and costs are practical factors, which play an important role in the choice of methodologies and evaluation structures. Nevertheless, the method and techniques that are used depending on the nature of the content, the focus, the costs and the equipment, the most important is that it will improve the product as long as its results are fed back to the development group and re-integraged in the system.

The following graphic summarizes visually our methodology used in our case study of GGBook:

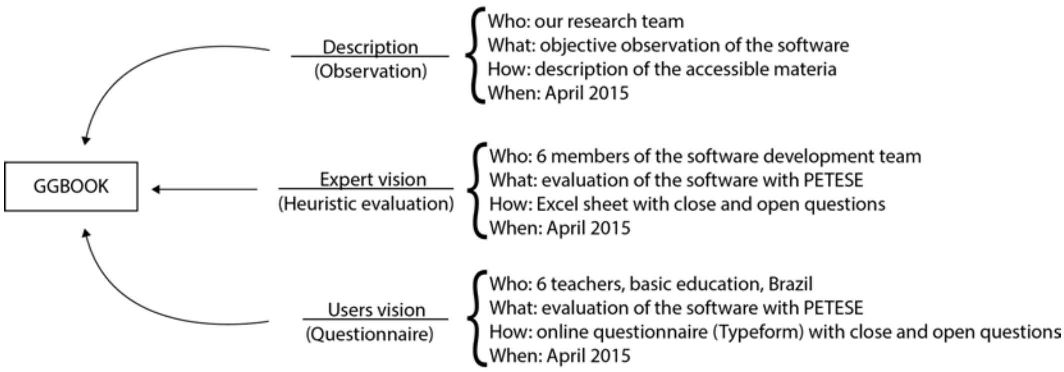


Figure 32 : Research design for the case study of the educational mathematics software GGBook

4 Empirical Research

Summary

In this section, the results of our three different methods are detailed. This mains: 1/ an objective description of the educational website GGBook; 2/ the expert vision on the software and 3/ the vision of the users on the software. For the two last parts, the structure will follow the structure of the PETESE, tool used to evaluate the software.

DESCRIPTION OF GGBOOK

Login of GGBook

To access to the online platform, the user needs to enter in the navigator through the following address: http://www.GGBook.com.br. The following screens then appear, depending if the user already has an account or not:

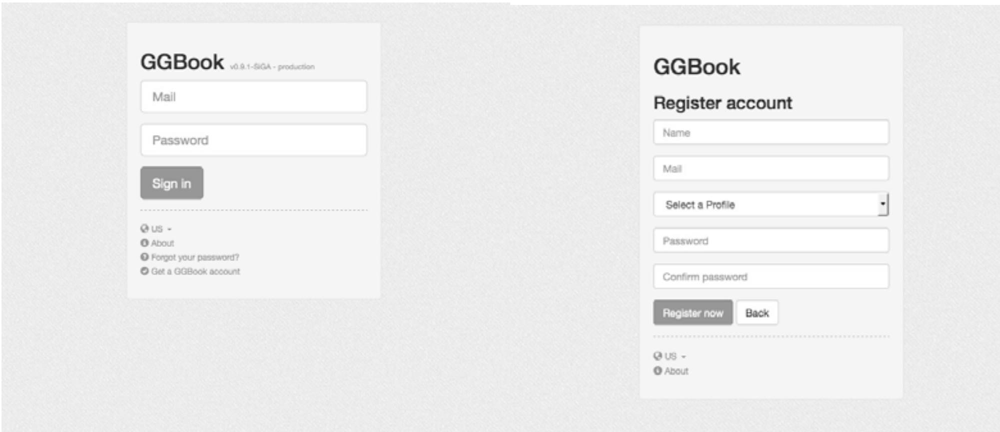


Figure 33 : Login and sign up pages of GGBook

When it is the first time for the user to access the program, the user needs to create an adequate profile (admin, teacher, student) and enter an e-mail address as well as its preference language (French, Portuguese, English, Spanish).

Once the program is entered, the program shows the page of the books; empty when it is the first use of the software; with books, when some have already been created:

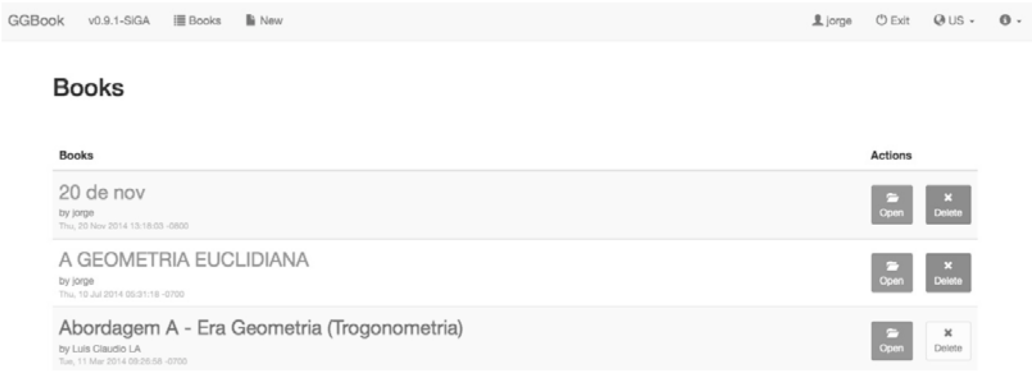


Figure 34 : Welcoming page of GGBook

GGBook's User Interface

Since GGBook joins dynamical geometry with computer algebra, its user interface contains additional components that cannot be found in simple geometry software. The software is divided into four different zones like shown in the following figure: 1/ the menu bar; 2/ the algebraic part; 3/ the graphical representations; and 4/ the book navigation bar.

Figure 35 : The four different zones of GGBooks' interface

1/ The menu bar

The menu bar contains the following elements:

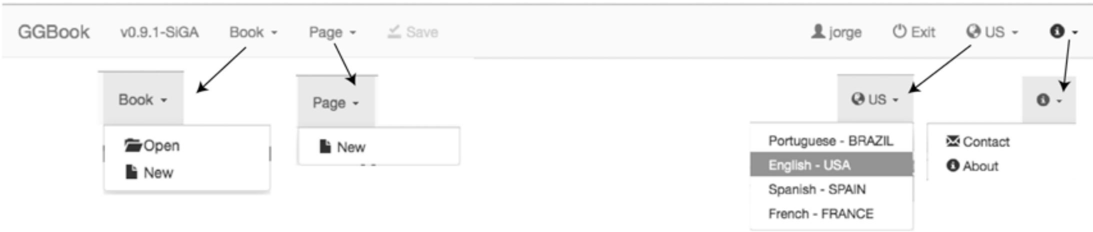


Figure 36 : The menu and the sub-menus de GGBook

- 'GGBook': this link leads to the welcoming page (see figure 30), which is the book page.
- 'Books': this leads to the book page, the same as the page of 'GGBook'. This contains two sub-menus:
  - 'Open': this leads to a pop-up page containing the list of the already created books.
  - 'New': this leads to a pop-up page to create a new book.
- 'Pages': this leads to a creation of a new page, witch adds a new page in the book navigation bar.
- 'Utilisador': this page leads to a page with personal information where the name, e-mail address and passwords can be changed.
- 'Exit': When the user exits the software, it goes directly to the login page, without any message.
- 'Languages': Four languages are available and change immediately the software when they are changed.
- 'Information': The information contains two sub-parts:
  - 'Contact': This opens your mailbox to write an e-mail.
  - 'About': This brings a pop-up page the information and photos of the creators of the software.

2/ The algebraic part

The algebra window is placed on the left hand side of the GeoGebra window. It contains the numeric and algebraic representations of the objects as well as a unique toolbar that appears when the user activates that zone by a mouse-click. Algebraic expressions can be changed directly in the algebra window, using the text tools or the formula tools that are the last line of the toolbar without any coding.

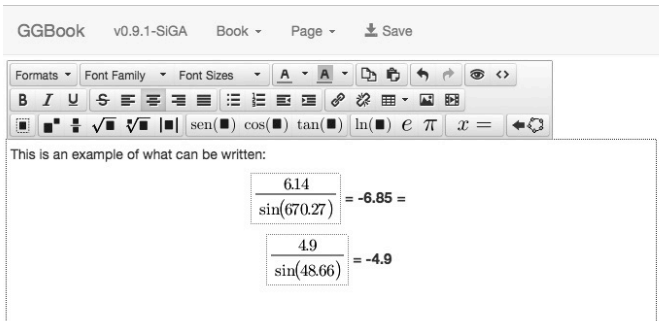


Figure 37 : Algebraic part of GGBook

3/ The graphical representation part

The graphics window is placed on the right hand side of the GGBook window. It contains a drawing pad on which the geometric representations of objects are displayed. The objects can be modified directly by dragging them with the mouse, while new objects can be created using the dynamic geometry tools provided in the toolbar.

This toolbar consists of a set of toolboxes in which the tools are organized. Both the name of the activated tool as well as the toolbar help, which is placed right next to the toolbar, give useful information on how to operate the corresponding tool and, therefore, how to create new objects. In the right corner of the toolbar the Undo and Redo buttons can be found, which enable the user to undo mistakes step-by-step.

The menu bar of GeoGebra is placed above the toolbar. It provides a wide range of menu items allowing the user to save, print, and export constructions, as well as to change default settings of the program, create custom tools, and customize the toolbar.

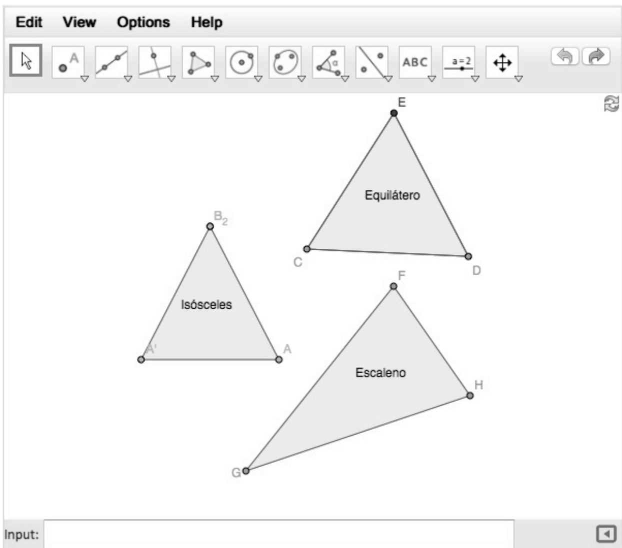


Figure 38: Graphical representation part of GGBook

4/ The book navigation bar

In this last bar, the user can creates different pages of the book and switch easily form one to another. The professor can also change the pages easily to reorganize them.

Creating material with GGBook

GGBook is initially developed with the goal of helping teachers making mathematical exercises or other instructional materials easily without coding. It turned out to also be a very useful tool for letting the students explore and discover mathematical concepts on their own and on an interactive way. For this purpose, the software offers different possibilities to allow a wide range of teachers to realize their own visions of successful instructional materials through a file handling, picture handling and text processing handling.

Teaching mathematics with GGBook

The program linking the construction of geometry with the functionalities of an algebra system opens up a range of application possibilities for teaching mathematics. Teachers can use the software in all grade levels from secondary school up to college and university for a wide range of different mathematical topics.

GGBook can thus be used as:

- a presentation tool
- an analyses of practice daily situations
- discovering learning and lesson enrichment
- student centered teaching
- creating of a dynamic book for the teachers

Added value of GGBook over GeoGebra seen by its conceptors

The developers of the software characterize GGBook as follow (Nobriga, 2015):

- The dynamical aspect allows to see directly all the modifications in the text as well as the graphical part, which is not possible in a statically software.
- The layout of the pages in GGBook allows a rapid navigation from one page to another, which is not the case in most of the software where the user needs to minimalize the windows.
- The text editor and the equations of GGBook allow the user to express himself mathematically in an intuitive way, without any coding knowledge. This is also the case for the graphical part where the user does not particularly need manual abilities or knowledge in a particular software.
- Finally, the program allows to save the instructions done, which is not the case in GeoGebra.

The development team of GGBook

The aim of this expert's vision analysis is to apply our referential to the development team of GGBook. This allowed us to highlight two essential objectives: on one hand, the positioning of each person of the team regarding to the software and; on the other hand, the vision of each expert on the nature of the criteria's themselves.

Before entering in the details of our results, here is the organization of the development team of GGBook:

- Coordinator
  - Content designer
  - Professor of Mathematics
  - Graphic designer
  - Developper specialized in usability
  - Developper specialized in technical aspects

Figure 39 : Organization chart of the GGBook team

It is observable that the functions of the development team are quiet close to the division of our referential. Therefore, each of those experts will have some questions that are unique on their area of expertise.

Part 1: General Aspects

The results of the first part of the questionnaire regarding the general aspects can be founded in Table 6 : **Results of expert analyses > Part 1 : general aspects**

⊕ present ⊘ don't know / not applicable ⊖ absent

1. GENERAL ASPECTS	pedagogy	usability	technique	content	interface	coordinator
- The software is innovating and has an added value	⊕	⊕	⊕	⊕	⊕	⊕
- The software is easy to use	⊕	⊖	⊖	⊕	⊖	⊘
- The software has an interdisciplinary perspective	⊕	⊕	⊕	⊕	⊕	⊕
- The software contains different levels of complexity	⊖	⊖	⊕	⊖	⊕	⊕
- There is a learning approach that matches with the curriculum of the targeted public	⊕	⊕	⊕	⊕	⊕	⊕

Table 6 : Results of expert analyses > Part 1 : general aspects

In this table, it can be observed that for 3 criteria all people of the team agree. In other words, the software is innovating and has an added value, has an interdisciplinary perspective and there is a learning approach that matches with the curriculum of the targeted public. For the other criteria, the members of the team disagree. Indeed, from the point of view of the technical members (developers and interface), the software is not easy to use. For the teacher and the content designer, however, it is. Another interesting point is the level of the complexity of GGBook. For the teacher and content designer, the software just has one level, which for the technical members is the opposite. This could be explained by a misunderstanding of the criteria or another vision of it ; a technical vision being different from a pedagogical point of view.

Regarding the open question on the possible ameliorations of the software, three suggestions are done, like observable in Table 7 : **Results of expert analyses > Part 1 : comments on the general aspects**

	FIELD OF THE EXPERT:	COMMENTS:
1. GENERAL ASPECTS	- Pedagogy:	There is only one level of difficulty in GGBook. It could be interesting to discuss as a team if it is interesting or not to develop levels of complexity.
	- Usability:	From a point of view of the software itself, GGBook does not organize its tools in order of complexity of ability. It could be interesting to think about a way of doing so.
	- Technique:	/
	- Content:	/
	- Interface:	The software is intuitive but sometimes, it is not, like the part of the formules as well as the difficulty of the interface when entering the software for the first time.
	- Coordinator:	/

Table 7 : Results of expert analyses > Part 1 : comments on the general aspects

Part 2: Pedagogical Aspects

The results of the second part of the questionnaire regarding the general aspects can be founded in Table 8.

⊕ present ⊘ don't know / not applicable ⊖ absent

2. PEDAGOGICAL ASPECTS

	pedagogy	usability	technique	content	interface	coordinator
2.1 Educational objectives						
- There are clear defined and communicated educational objectives	⊕	⊘	⊘	⊖	⊖	⊕
- The construction takes place in an individual context but integrates elements of cooperation and collaboration	⊕	⊕	⊕	⊘	⊕	⊕
- It promots creativity	⊕	⊕	⊕	⊕	⊕	⊕
- Exploration is the favoured approach and encourages students to seek knowledge independently	⊕	⊕	⊕	⊕	⊕	⊕
- The software encourages self-analyses, self-regulation, self-reflection and self-awareness	⊕	⊕	⊘	⊕	⊕	⊖
2.2 Activities & Tasks						
- The activities work the necessary aspects to meet the proposed mathematical objectives	⊕	⊕	⊕	⊕	⊕	⊕
- Tasks are relevant, realisting, authentic and represent the natural complexities of the real world, so close as possible of the targeted public	⊕	⊕	⊕	⊕	⊕	⊕
- The level of difficulty is appropriated to the public and schaffolding facilitates to perform just beyond the limits of its ability	⊖	⊘	⊘	⊕	⊕	⊕
- There is a variability in the tasks	⊕	⊕	⊕	⊕	⊕	⊕
- The results of the activities are available	⊕	⊕	⊘	⊕	⊖	⊕
- Activities can be created easily and customized	⊕	⊕	⊕	⊕	⊕	⊕
- The tasks are problem-solving and develop higher order-thinking skills through situations that enlarge the competences and the concepts	⊕	⊕	⊘	⊕	⊕	⊕
2.3 Role of the teacher						
- The software can be used without the intervention of an instructor	⊖	⊖	⊕	⊕	⊕	⊖
- Teachers serve in the role of guides, monitors, coaches, tutors or facilitators	⊕	⊖	⊕	⊕	⊖	⊕
2.4 Evaluation						
- The software recognizes and supports the important role of the teacher	⊕	⊕	⊕	⊕	⊕	⊖
- There is a presence of evaluation	⊕	⊕	⊖	⊖	⊖	⊖
- The evaluation is authentic and interwoven with teaching	⊕	⊕	⊖	⊖	⊖	⊕

Table 8 : Results of expert analyses > Part 2 : pedagogical aspects

In this table, four aspects of pedagogy where analyzed. The educational objectives, regarding the expert of pedagogy are all present in GGBook. However, for the majority of the others, the software does not clearly explicit and communicates the educational objectives. Indeed, some experts have indicated that they didn't know the answer because the educational objective can be defined by the professors and does not necessarily has to be present in the software itself. If the software encourages self-analyses, self-regulation, self-reflection and self-awareness is also not clear for everyone. This can maybe be explained by the difficulty of the criteria that takes four concepts into consideration.

Regarding the aspects of the activities and the tasks, all members of the team agree that the software meets the proposed mathematical objectives, that the tasks are relevant and represent the natural complexities of the real world. The software also contains a huge variability in the tasks and allows an easy creation of the tasks with customization. According to the pedagogical specialist, GGBook has an absence of scaffolding in the tasks. This is also confirmed by the usability and technique point of view. For the interface designer, another problem is present: the results of the activities are not available. However, this can be explained by the fact that the professors correct the exercises and not the software. Here again, it is a result of differences in interpretation of the criteria.

For the role of the professors, the opinions differ. For some, the software can be used without the intervention of an instructor, for others not. This same difference exists for the criteria of the role of guidance and tutor of the teacher. The content specialists gave a possible explanation to this disagreement: the teacher is the one that creates the exercises, so its intervention is essential during the creation, however, the resolution of the students does not need any intervention and the teacher can thus act as a facilitator. This latter depends of the choice of the teacher aside the program, which can explain the different points of views of the answers.

Finally, concerning the evaluation, the pedagogical and usability experts agree on all criteria, while for minimum one of the team members the criteria are absent in GGBook. Once, more, this can be explained by a misunderstanding of the point of view of the criteria itself; an evaluation in the system (GGBook) or an evaluation by the user (teacher)?

What concerns the open question on the possible improvements of the software, three experts make suggestions, like observable in Table 9. The usability expert proposes that the educational objectives are integrated in the software. The content expert proposes an integration of more interaction in the software as well as an evaluation system. Finally, the interface expert supports all the above mentions aspects by giving other ideas of integration.

	FIELD OF THE EXPERT:	COMMENTS:
2. PEDAGOGICAL ASPECTS	- Pedagogy:	/
	- Usability:	The educational objectives are not explicit. This could be done however through the professors or a mecanic of the software could offer it as well.
	- Technique:	/
	- Content:	The interaction is more between the teachers and the students than between the students. However, this could be taken into consideration in a further amelioration of the program. Concerning the evaluation of the software, this could be imagined in a more 'automatic' way through GGBook for instance.
	- Interface:	There is not a clear communicated objective of the software. It could be interesting to mention the aim of the software on the homepage for example with arguments of why it is an interesting program to use. The results of the activities are not given. Why not inventing a way of communication between the professor and the answer of the student. This would also reinforce the role of guide of the professors that could be more active of what is present. Concerning the evaluation part, it is completely absent in the program. However, the presence of it could be a great idea.
	- Coordinator:	/

Table 9 : Results of expert analyses > Part 2 : comments on the pedagogical aspects

Part 3: Usability Aspects

The results of the third part of the questionnaire regarding the usability aspects can be founded in Table 10.

⊕ present ⊘ don't know / not applicable ⊖ absent

3. USABILITY ASPECTS	pedagogy	usability	technique	content	interface	coordinator
3.1 Guidance & Instructions						
- The system guides and encourages the user	⊖	⊖	⊘	⊕	⊖	⊖
- There are help ressources	⊖	⊖	⊘	⊖	⊖	⊖
- Instructions are clear and easily available	⊖	⊖	⊘	⊖	⊖	⊖
3.2 Feedback & Motivation						
- Feedback is effectively employed	⊖	⊖	⊘	⊕	⊖	⊖
- The software contents an intrinsic motivation	⊖	⊕	⊕	⊕	⊖	⊖
- The software encourages the users in their positive actions	⊖	⊖	⊘	⊕	⊖	⊖
- The feedback is positive even if the user fails	⊖	⊖	⊖	⊖	⊖	⊖
- The feedback allows the learner to reflect on his errors	⊖	⊖	⊕	⊕	⊖	⊖
- The software offers a feedback of the student progress	⊖	⊖	⊕	⊖	⊖	⊖
3.3 Interaction (system & users)						
- There is a capacity of interaction with the system	⊕	⊕	⊕	⊕	⊕	⊕
- The software has a section where interaction like discussion and collaboration can take place	⊖	⊕	⊕	⊖	⊖	⊕
3.4 Error prevention						
- The software recognizes errors and recovers the process	⊘	⊖	⊖	⊖	⊘	⊕
- It prevents the user to make errors	⊘	⊖	⊖	⊖	⊘	⊘
- The error messages are clear	⊘	⊖	⊘	⊕	⊖	⊘
- The user learns from its errors	⊘	⊕	⊕	⊕	⊖	⊕
3.5 Control & Personalization						
- The learner has a control on the system and the actions	⊘	⊕	⊕	⊕	⊕	⊕
- There is a possible personalization of the settings or informations	⊘	⊖	⊘	⊕	⊖	⊘

Table 10 : Results of expert analyses > Part 3 : usability aspects

The first impression of the table is an impressive amount of 'absent' and 'I don't know' answers. Indeed, only for one criteria, all experts agree: there is a capacity of interaction with the system, and one criteria everyone almost agree (only one "I don't know"): the learner has a control on the system and the actions. In opposite, there is one criteria where everyone finds that GGBook does not integrate that criteria: the feedback is positive even if the user fails. For several other criteria however, some people didn't know and the all the rest finds it not present in GGBook. It is the case of the prevention of errors, the help resources and the clear instructions.

What concerns the other criteria, we can observe that the usability experts give a good mean of the different experts. Guidance and instructions are thus not very present in GGBook. Feedback and motivation are mainly absent as well, however, the majority says that the software contains an intrinsic motivation. Interaction is well present in the GGBook. Error prevention is mainly not present for the experts with the exception that the learner learns from its errors. This can be explained by the discovery learning perspective integrated in GGBook that allows learning from the mistakes of the users. Finally, we can say that the user in GGBook has control on the system and the actions, however, there is a lack of personalization.

What concerns the possible improvements of the software concerning the usability aspects, four interesting perspectives are given (Table 11). Indeed, the pedagogical expert recognizes the problems and their importance of amelioration. The technical expert reinforces the criteria of intrinsic motivation of GGBook and proposes to lay the accent more on it. The content expert lays the importance on the error messages and the instructions that could be more present and clear. Finally, the interface expert makes as well propositions of better interaction.

3. USABILITY ASPECTS FIELD OF THE EXPERT:	COMMENTS:
- Pedagogy:	The software does not encourage the users, does not give feedback nor has help functionalities. However those elements are from great importance. It is the same regarding the interaction of the software.
- Usability:	/
- Technique:	The intrinsic motivation is high because it is a new way of working with mathematics. This could be put more in advance.
- Content:	GGBook needs help functions, mostly concerning the insertion of data in the Geogebra part and for the edition of equations tool. Technical instructions could also be good. The prevention of the user to make errors could also be developped better.
- Interface:	The motivation part could be rethought. The help functions are important and should also be present. Concerning the feedback, it could be good to imagine a kind of positive feedback. Why not as well a chat or a conversation part to allow interaction. Error messages could be used for instance when the user does something wrong in an exercice.
- Coordinator:	/

Table 11 : Results of expert analyses > Part 3 : comments on the usability aspects

Part 4: Technical Aspects

The results of the fourth part of the questionnaire regarding the technical aspects can be founded in Table 12.

⊕ present ⊘ don't know / not applicable ⊖ absent

4. TECHNICAL ASPECTS	pedagogy	usability	technique	content	interface	coordinator
4.1 Compatibility						
- The software needs to be compatible with the hardware of the users	⊖	⊖	⊖	⊕	⊘	⊕
- The conditions of compatibility are clearly explained	⊕	⊕	⊕	⊖	⊖	⊖
4.2 Installation						
- The software needs to be easy to install	⊖	⊕	⊕	⊕	⊕	⊖
- The modality of use are explained	⊕	⊕	⊘	⊘	⊖	⊖
4.3 Data management						
- The software is able to manipulate the date that it requires even in great quantity	⊘	⊕	⊕	⊕	⊘	⊕
- The data generated through the software can be analyzed	⊘	⊕	⊕	⊕	⊘	⊕
- There is a transparency of the information to the user	⊘	⊕	⊕	⊕	⊕	⊕
- The software saves the data of the user	⊘	⊕	⊕	⊕	⊕	⊕
4.4 Maintenance						
- The software will be maintained after its release	⊘	⊕	⊘	⊕	⊘	⊘
- Updates will take place to keep the software up to date	⊘	⊕	⊘	⊕	⊘	⊘

Table 12 : Results of expert analyses > Part 4 : technical aspects

According the technical expert, the only criteria that GGBook does not complete is the compatibility of the software with the hardware of the user. This is for the simple reason that the software is only accessible online and thus, this criteria is not applicable for GGBook. The installation criteria, are thus as well not applicable for GGBook. However, the modality of use could be added to GGBook even if it is only a web-based software. What concerns the data management and the maintenance, everyone responded that it was completed by the software of that they didn't know.

This can also be noticed by the low content quantity of comments regarding the technical aspects (Table 13). Indeed the experts only highlight the information that could be given to the users about the modality of use.

4. TECHNICAL ASPECTS

FIELD OF THE EXPERT:	COMMENTS:
- Pedagogy:	/
- Usability:	/
- Technique:	The only thing that could cause problems is the low RAM of the user. Maybe this could be adviced at the start.
- Content:	The first time on the program, it will take a little bit more time. This could maybe be informed to the user.
- Interface:	The modality of use could be informed to the user.
- Coordinator:	/

Table 13 : Results of expert analyses > Part 4 : comments on the technical aspects

Part 5: Content's Criteria

The results of the fifth part of the questionnaire regarding the content criteria can be founded in Table 14.

5. CONTENT

⊕ present ⊘ don't know / not applicable ⊖ absent

	pedagogy	usability	technique	content	interface	coordinator
5.1 Language						
- The language is adapted to the public	⊕	⊕	⊕	⊕	⊕	⊕
- The mathematical terms are correct and also adapted to the public	⊕	⊕	⊕	⊕	⊕	⊕
- The content is clear, consistent, concise and comprehensible	⊕	⊕	⊕	⊕	⊕	⊕
5.2 Multimedia elements (images/photos/drawings/videos/sound/...)						
- Those elements are appropriated, add a value and evoid redundancy with the other contents	⊕	⊕	⊕	⊕	⊕	⊖
- Multimedia elements are used to facilitate the comprehension	⊕	⊕	⊕	⊕	⊕	⊖
- If they come from another source, they are used legally	⊘	⊕	⊕	⊘	⊕	⊖
- They are from good quality (sound audible, images visible,...)	⊕	⊕	⊕	⊘	⊕	⊖
- The multimedia explanations present the elements contiguously rather than presented separately	⊕	⊕	⊕	⊘	⊕	⊖
5.3 Organization						
- The content is organized to avoid cognitive load	⊘	⊖	⊕	⊕	⊘	⊕
- The content is reachable in terms of minimal actions	⊕	⊕	⊕	⊕	⊕	⊖
- The content is broken down in small topics that can be accessed easier	⊕	⊕	⊕	⊕	⊕	⊖
- The content can be reached via multiple paths	⊕	⊕	⊘	⊕	⊕	⊖
- The software has links to external additional information	⊘	⊕	⊘	⊕	⊖	⊖

Table 14 : Results of expert analyses > Part 5 : content aspects

From the results appears clearly that the content aspects in GGBook are quiet good. This is confirmed by the content expert that only responds positively, with some ‘I don’t know’s’. This is the case for the language aspects as well as the multimedia elements. However, the coordinator does not always agree what concerns the multimedia elements and the organization of GGBook. This can be explained by a miss understanding of the word multimedia, since GGBook integrates another software (GeoGebra).

Concerning the organization, the usability experts finds that the content can generate a cognitive overload, and the interface experts highlights the weak additional information in GGBook. This latter is also mentioned in the comments of Table 15. The content expert also mentioned the future integration of videos.

5. CONTENT

FIELD OF THE EXPERT:	COMMENTS:
- Pedagogy:	/
- Usability:	/
- Technique:	/
- Content:	It is in the plans that GGBook integrates videos. It is actually not the case, but we believe that it would facilitate the comprehension a lot.
- Interface:	It could be very interesting to have links to external information about the same topic for instance.
- Coordinator:	/

Table 15 : Results of expert analyses > Part 5 : comments on the content aspects

Part 6: Interface Aspects

The results of the sixth, and last part of the questionnaire regarding the interface criteria can be founded in the following table (Table 16).

6. INTERFACE

⊕ present ⊘ don’t know / not applicable ⊖ absent

	pedagogy	usability	technique	content	interface	coordinator
6.1 Navigation						
- The structure of the software is clear and quickly understandable	+	+	+	+	+	+
- The navigation is coherent	+	+	+	+	+	+
6.2 Layout						
- The typography is clear and adapted to its support	+	+	+	/	+	/
- There is a pleasing design that is coherent on the different levels of architecture of the software	+	+	+	/	+	/
- The actions are explicit	/	+	+	+	-	+
- Icons are representative	+	+	+	-	+	/
- The colors are meaningful and help in the comprehension of the software	+	+	/	/	-	-

Table 16 : Results of expert analyses > Part 6 : interface aspects

The results show that the navigation is optimal in the software of GGBook. For the layout criteria, the opinions are divergent. For the interface experts, the actions are not always explicit, and the colors do not always help in the comprehension of the software. This last argument is approved by the coordinator of the software. For the content expert, the icons are not always representative. For all the others, the answers are positive or without opinion.

	FIELD OF THE EXPERT:	COMMENTS:
6. INTERFACE	- Pedagogy:	/
	- Usability:	/
	- Technique:	/
	- Content:	The icons of the editor of the equations needs to be improved.
	- Interface:	The design could be a little more colorfull. Why not a kind of personalization of colors.
	- Coordinator:	/

Table 17 : Results of expert analyses > Part 6 : comments on the interface aspects

The comments explicit more the visible answers of Table 16. Indeed, the content expert suggests improving the equation icons, and the designer suggests a personalization of the colors.

Evaluation of the PETESE

To our three questions regarding the PETESE tool, the different experts answered the following things:

1. According to your expertee, do the criteria from the literature represent well the different aspects to take into consideration during the developpement of a mathematical software?

- Pedagogy:	/
- Usability:	During the development of GGBOOK, the most important criteria were the consistency between mathematical concepts and produced technological artifacts; communication between the student and the teacher; a construction and mathematical communication; and storage activities produced by both the student and the teacher. Even though we used templates and market frameworks for the development of GGBOOK, I think the usability criteria were not considered as they should. Therefore, I see those criteria as very important for any software.
- Technique:	Yes, it makes a highlight of what is prioritized in the use of current tools and techniques. It is also well referenced and understandable in the programming world.
- Content:	In short words, the development of GGBOOK was based on the theory of semiotic representations of Duval. This theory says that for mathematics comprehension is necessary to include the representations of mathematical object records.
- Interface:	Yes, it is good that it takes the differents aspects of the system.
- Coordinator:	Yes. Ergonomic criteria, for exemple, are rarely explained and developers, generally, have no theory support to develop softwares.

Table 18 : Answers of the experts on the first PETESE question

According to the answers, the fact to take the answers from the literature brings a huge credibility to the criteria, even more because the educational software are quiet new and they need a support of other more antigue areas, like the area of ergonomics. Another mentioned point is the comprehension of the criteria that seem quiet understandable for the majority of the people.

2. Are there important aspects that are not present in the criteria and should be added because of their importance during the development?

- Pedagogy:	I think all aspects marked as absent so far could be added to improve the software experience of using both the teacher user perspective and student.
- Usability:	/
- Technique:	I believe that adding more moments of tests could increase the focus on bug fixes rather than developing new features. I believe it is important to add new features, but the stability of which is already available should not be ruled out, and user testing (teachers / students) could change the focus accordingly.
- Content:	It should be good to think on it a bit more, but first time the answer is no.
- Interface:	The aspects that are mentionned in the design part cover well the important points to take into consideration.
- Coordinator:	/

Table 19 : Answers of the experts on the second PETESE question

In Table 19, the main message is that no other criteria needs to be added. The accent of the answers highlight principally the important of the tests moments as well as the possibility it brings to think about improvements.

3. Are there criteria presented in the referential that, according to your expertee, are not important to be mentionned in the referential for a predictive evaluation?

- Pedagogy:	No. I think the instrument brings a set of fairly complete questions. However, I must point out that sometimes I was not sure about which type of user should consider: the teacher who writes the activity or the student who uses the produced activities. For example: "The level of difficulty is appropriated to the public and schaffolding Facilitates to perform just beyond the limits of its ability" We are talking about who the public? The teacher writing the activity or the student who uses the activity produced by the teacher?
- Usability:	No. I believe that the instrument is very complete.
- Technique:	No, although I believe there are most important criteria than others. I think the least important are the ways in which users have no contact directly. The information on which users (teachers / learners) relate are the most important, they are: educational and interface.
- Content:	No. I think it's very complete. I would suggest a field "maybe".
- Interface:	No. I think it is quiete complete.
- Coordinator:	Educational software engeneering is a new field and those kind of software is, generally, developed without the consideration of scientific criteria. This tool is thus important.

Table 20 : Answers of the experts on the third PETESE question

Table 20 shows that all criteria are important and no one would delete one. However, the pedagogical experts highlights the difficulty of some criteria, where it is not always clear to who the criteria is referred to; and the technical software suggests to prioritize the different criteria, because according to him, some are more important than others.

Introduction

In order to see the efficiency of our PETESE analysis tool as well as to have user's vision on GGBook, we have decided, like explained in the methodological part, to apply our same referential of criteria to professors of mathematics that already used GGBook before.

Like it can be observed in the following results, some criteria of the PETESE have not been asked to the users. Indeed, we have decided for some more technical and internal questions to delete them from the tool estimating that the user would not be able to answer them and that the length of unknown questions could have an impact on the quality of the other answers.

Part 1: General Aspects

Concerning the general aspects of GGBook, the teachers all agree on the fact that the software is innovating and has an added value ; the software is easy to use and that there is a learning approach that matches with the curriculum of the targeted public. Only one teacher didn't know if the software has an interdisciplinary perspective, while the others all approve. For the levels of complexity of the system, however, it is not that clear : 2 professors agree, 2 don't and 1 doesn't know.

⊕ present ⊘ don't know / not applicable ⊖ absent

1. GENERAL ASPECTS	User 1	User 2	User 3	User 4	User 5
- The software is innovating and has an added value	⊕	⊕	⊕	⊕	⊕
- The software is easy to use	⊕	⊕	⊕	⊕	⊕
- The software has an interdisciplinary perspective	⊕	⊕	⊘	⊕	⊕
- The software contains different levels of complexity	⊕	⊕	⊖	⊖	⊘
- There is a learning approach that matches with the curriculum of the targeted public	⊕	⊕	⊕	⊕	⊕

Table 21 : Results of the teachers analyses > Part 1 : general aspects

What concerns the possible remarks of the professors : one insists on a general improvement of the software ; the other on the levels of complexity, where the student can also develop and not only build.

	USER:	COMMENTS:
1. GENERAL ASPECTS	- User 1:	/
	- User 2:	I advise to improve the performance of the software because the last time I used it, there were lots of problems.
	- User 3:	/
	- User 4:	It could be good to have more levels of complexity, where the student can not only build but also develop.
	- User 5:	/

Table 22 : Results of the teachers analyses > Part 1 : comments on the general aspects

Part 2: Pedagogical Aspects

What concerns the educational objectives of the software, the answers differ well. Indeed, all agree that there are clear defined and communicated educational objectives as well as the promotion of creativity, however, only three agree that the construction takes place in an individual context, but contains elements of cooperation and collaboration. The exploration to encourage students to seek knowledge independently as well as the self-analyses, self-regulation, self-reflection and self-awareness is seen as well as present as absent.

Concerning the activities and the tasks of, the teachers evaluated that they are easy to create, personalizable, problem-solving and meet the proposed mathematical objectives. 4/5 agree that the level of difficulty is appropriated to the public, the tasks are relevant and realistic and highlight also the variability in the tasks. Where the teachers do not agree, is the availability of the the results of the activities.

It is also clear that GGBook helps the teachers to play the role of guide and coach, however, for 2 out of the 5 users, GGBook the software can not be used without the intervention of an instructor.

The evaluation of the software is not clear, because in all answers related to that topic, at least one person didn’t know what to answer.

⊕ present ⊘ don't know / not applicable ⊖ absent

2. PEDAGOGICAL ASPECTS	User 1	User 2	User 3	User 4	User 5
2.1 Educational objectives					
- There are clear defined and communicated educational objectives	⊕	⊕	⊕	⊕	⊕
- The construction takes place in an individual context but integrates elements of cooperation and collaboration	⊘	⊕	⊘	⊕	⊕
- It promotes creativity	⊕	⊕	⊖	⊕	⊕
- Exploration is the favoured approach and encourages students to seek knowledge independently	⊘	⊕	⊘	⊕	⊖
- The software encourages self-analyses, self-regulation, self-reflection and self-awareness	⊘	⊕	⊘	⊖	⊕
2.2 Activities & Tasks					
- The activities work the necessary aspects to meet the proposed mathematical objectives	⊕	⊕	⊕	⊕	⊕
- Tasks are relevant, realisting, authentic and represent the natural complexities of the real world, so close as possible of the targeted public	⊕	⊕	⊕	⊕	⊘
- The level of difficulty is appropriated to the public and schaffolding facilitates to perform just beyond the limits of its ability	⊕	⊕	⊖	⊕	⊕
- There is a variability in the tasks	⊕	⊕	⊘	⊕	⊕
- The results of the activities are available	⊕	⊕	⊖	⊖	⊘
- Activities can be created easily and customized	⊕	⊕	⊕	⊕	⊕
- The tasks are problem-solving and develop higher order-thinking skills through situations that enlarge the competences and the concepts	⊕	⊕	⊕	⊕	⊕
2.3 Role of the teacher					
- The software can be used without the intervention of an instructor	⊖	⊕	⊕	⊖	⊕
- Teachers serve in the role of guides, monitors, coaches, tutors or facilitators	⊕	⊕	⊕	⊕	⊕
2.4 Evaluation					
- The software recognizes and supports the important role of the teacher	⊕	⊕	⊘	⊕	⊕
- There is a presence of evaluation	⊕	⊕	⊘	⊕	⊘
- The evaluation is authentic and interwoven with teaching	⊕	⊕	⊕	⊕	⊘

Table 23 : Results of the teachers analyses > Part 2 : pedagogical aspects

What concerns the comments of the users (Table 24), user 1 explains that the software needs the teacher to explain in the beginning how it works because of its difficulty in the beginning. User 4 highlights the point of the results of the non presence of the results of the activities that could demotivate the students. Finally, user 5 agrees with user 1 and stresses once more the necessity of the explanations in the beginning.

2. PEDAGOGICAL ASPECTS — USER:	COMMENTS:
- User 1:	In my case I found that all tasks must be accompanied by the teacher, because the software is difficult to use in the beginning.
- User 2:	/
- User 3:	/
- User 4:	The results of the activities are not always available and this somehow may discourage the student in case it is not fixed.
- User 5:	The GGBook is quite interesting and can be used with or without the aid of a teacher. But it is necessary that you make use of the program has well defined goals for learning, for he alone can discover and build concepts or be guided by a teacher in the construction of concepts.

Table 24 : Results of the teachers analyses > Part 2 : comments on the pedagogical aspects

Part 3: Usability Aspects

In the usability aspects, we have decided to leave some of the criteria of the PETESE with the argument that the users, in our case the teachers, could not respond due to a lack of accessibility of the information of a lack of knowledge.

Regarding the guidance and instructions, all teachers agree that there are help resources. And if it is also positive for the majority of the teachers that the system guides the users and gives clear and easily available instructions, user 2 disagrees.

The feedback and motivation is only analyzed by two criteria and gives controversial answers. With a high percentage of “I don’t know’s”, for both criteria, there are as many positive answers are negative answers.

What concerns the interaction, the teachers affirm that there is a capacity of interaction with the system and 3 users also confirm that the software has a section where interaction like discussion and collaboration can take place.

Finally, for the users, the control and personalization are present, with the exception of user 3 that doesn’t know.

⊕ present ⊘ don't know / not applicable ⊖ absent

3. USABILITY ASPECTS	User 1	User 2	User 3	User 4	User 5
3.1 Guidance & Instructions					
- The system guides and encourages the user	⊕	⊖	⊕	⊘	⊕
- There are help ressources	⊕	⊕	⊕	⊕	⊕
- Instructions are clear and easily available	⊕	⊖	⊕	⊕	⊕
3.2 Feedback & Motivation					
- Feedback is effectively employed	⊕	⊖	⊖	⊕	⊘
- The software contents an intrinsic motivation	○	○	○	○	○
- The software encourages the users in their positive actions	⊘	⊖	⊘	⊕	⊘
- The feedback is positive even if the user fails	○	○	○	○	○
- The feedback allows the learner to reflect on his errors	○	○	○	○	○
- The software offers a feedback of the student progress	○	○	○	○	○
3.3 Interaction (system & users)					
- There is a capacity of interaction with the system	⊕	⊕	⊕	⊕	⊕
- The software has a section where interaction like discussion and collaboration can take place	⊘	⊕	⊕	⊕	⊘
3.4 Error prevention					
- The software recognizes errors and recovers the process	○	○	○	○	○
- It prevents the user to make errors	○	○	○	○	○
- The error messages are clear	○	○	○	○	○
- The user learns from its errors	○	○	○	○	○
3.5 Control & Personalization					
- The learner has a control on the system and the actions	⊕	⊕	⊘	⊕	⊕
- There is a possible personalization of the settings or informations	⊕	⊕	⊘	⊕	⊕

Table 25 : Results of the teachers analyses > Part 3 : usability aspects

Conform to Table 26, the questions of the feedbacks as well as the encouraging of the students according to user 2 depends not only of the system itself, but also from the teacher. Indeed, it is important to mention here that the content of GGBook is principally created by the professors itself and not the GGBook. This is why this teacher (user 2) reacts that way.

	USER:	COMMENTS:
3. USABILITY ASPECTS	- User 1:	/
	- User 2:	I think this question of encouraging students largely depends on the teacher and the feedbaks too. I also think that students need to be more familiar with the tool and understand a little of the logical operation.
	- User 3:	/
	- User 4:	/
	- User 5:	/

Table 26 : Results of the teachers analyses > Part 3 : comments on the usability aspects

Part 4: Technical Aspects

For the technical aspects, we have only asked two criteria. For the first one – is there a transparency of the information to the user – four users have responded ‘yes’, and 1 ‘I don’t know’. For the second question – does the software save the data of the user – all answers have been positive.

⊕ present ⊘ don't know / not applicable ⊖ absent

4. TECHNICAL ASPECTS	User 1	User 2	User 3	User 4	User 5
4.1 Compatibility					
- The software needs to be compatible with the hardware of the users	○	○	○	○	○
- The conditions of compatibility are clearly explained	○	○	○	○	○
4.2 Installation					
- The software needs to be easy to install	○	○	○	○	○
- The modality of use are explained	○	○	○	○	○
4.3 Data management					
- The software is able to manipulate the date that it requires even in great quantity	○	○	○	○	○
- The data generated through the software can be analyzed	○	○	○	○	○
- There is a transparency of the information to the user	⊕	⊕	⊘	⊕	⊕
- The software saves the data of the user	⊕	⊕	⊕	⊕	⊕
4.4 Maintenance					
- The software will be maintained after its release	○	○	○	○	○
- Updates will take place to keep the software up to date	○	○	○	○	○

Table 27 : Results of the teachers analyses > Part 4 : technical aspects

For this part of the questionnaire, no comments and improvements have been given like illustrated in Table 29. This can be explained by the few quantity of criteria that we have asked concerning the technical part and also because non of the answers of the users where negative.

4. TECHNICAL ASPECTS – USER:	COMMENTS:
- User 1:	/
- User 2:	/
- User 3:	/
- User 4:	/
- User 5:	/

Table 28 : Results of the teachers analyses > Part 4 : comments on the technical aspects

Part 5: Content's Criteria

The content's criteria were almost all seen as positive. Only user 3 noted a negative point : the software hos no links to external additional information. For the rest, there are only agreements or 'I don't know's', with a strong unanimity on the language criteria compared to the multimedia elements and the organization.

⊕ present ⊘ don't know / not applicable ⊖ absent

5. CONTENT	User 1	User 2	User 3	User 4	User 5
5.1 Language					
- The language is adapted to the public	⊕	⊕	⊕	⊕	⊕
- The mathematical terms are correct and also adapted to the public	⊕	⊕	⊕	⊕	⊕
- The content is clear, consistent, concise and comprehensible	⊕	⊕	⊕	⊕	⊕
5.2 Multimedia elements (images/photos/drawings/videos/sound/...)					
- Those elements are appropriated, add a value and evoid redundancy with the other contents	⊕	⊕	⊘	⊕	⊕
- Multimedia elements are used to facilitate the comprehension	⊕	⊕	⊕	⊕	⊘
- If they come from another source, they are used legally	○	○	○	○	○
- They are from good quality (sound audible, images visible,...)	⊕	⊕	⊘	⊕	⊕
- The multimedia explanations present the elements contiguously rather than presented separately	⊘	⊕	⊘	⊕	⊘
5.3 Organization					
- The content is organized to avoid cognitive load	⊕	⊕	⊘	⊕	⊘
- The content is reachable in terms of minimal actions	⊕	⊕	⊘	⊕	⊘
- The content is broken down in small topics that can be accessed easier	⊕	⊕	⊕	⊕	⊘
- The content can be reached via multiple paths	⊕	⊕	⊘	⊕	⊕
- The software has links to external additional information	⊘	⊕	⊖	⊕	⊘

Table 29 : Results of the teachers analyses > Part 5 : content aspects

Once more, none of the users have make comments. This can maybe be explained by the high quantity of positive points that the user have indicated and where they estimate that no improvements are needed.

5. CONTENT — USER:	COMMENTS:
- User 1:	/
- User 2:	/
- User 3:	/
- User 4:	/
- User 5:	/

Table 30 : Results of the teachers analyses > Part 5 : comments on the content aspects

Part 6: Interface Aspects

For the interface elements, the navigation is seen as positive by all evaluators. The layout is also very positive. However, concerning the icons, the explicitly of the actions and the colors, there is always one user that answered ‘I don’t know’. For the typography and the coherence, this amount goes to two users.

⊕ present ⊘ don't know / not applicable ⊙ absent

6. INTERFACE	User 1	User 2	User 3	User 4	User 5
6.1 Navigation					
- The structure of the software is clear and quickly understandable	⊕	⊕	⊕	⊕	⊕
- The navigation is coherent	⊕	⊕	⊕	⊕	⊕
6.2 Layout					
- The typography is clear and adapted to its support	⊘	⊕	⊘	⊕	⊕
- There is a pleasing design that is coherent on the different levels of architecture of the software	⊕	⊘	⊘	⊕	⊕
- The actions are explicit	⊕	⊕	⊘	⊕	⊕
- Icons are representative	⊘	⊕	⊕	⊕	⊕
- The colors are meaningful and help in the comprehension of the software	⊕	⊕	⊘	⊕	⊕

Table 31 : Results of the teachers analyses > Part 6 : interface aspects

Like observable in Table 32, no users have make comments or improvements for the interface criteria.

6. INTERFACE — USER:	COMMENTS:
- User 1:	/
- User 2:	/
- User 3:	/
- User 4:	/
- User 5:	/

Table 32 : Results of the teachers analyses > Part 5 : comments on the content aspects

Evaluation of the PETESE

For the part of the evaluation of the use of the PETESE from the users, we have asked one direct question that can be visible in Table 33. The answers are quiet controversial such as user 4 that explains that all questions are clear and understandable while user 1 finds a lack of clarity in some questions. The appropriation of some questions is highlighted by user 2 and user 4 that describes some questions as “unnecessary”. Finally, user 5 explains its “I don’t know” by its lack of knowledge about the program.

Concerning the questions, were they comprehensible? Are they efficient to evaluate an educational software of mathematics ? Are there any questions you would have added/removed?

- User 1:	There is a lack of clarity in some questions, but I believe that they help to improve the software development.
- User 2:	I think some questions are not always appropriate, but other are indispensable.
- User 3:	Some questions I think unnecessary, better to be direct than ask the same question ...
- User 4:	All questions are clear and understandable.
- User 5:	I had to mark a lot of time "do not know", because of the lack of knowledge about the program.

Table 33 : Answers of the users on the PETESE question

Conclusion

In this empirical part of our research that corresponded to the application of our PETESE to the educational software of mathematics, GGBook with a case study methodology, we have accomplished our seventh and last objective, namely **application and evaluation of the referential to the practical case of GGBook.** The results will be discussed in the next part of our work.

5 Discussion And Conclusion

Summary

This section includes the general discussions of our research as well for the part of our anasynthesis and our PETESE as for the case study on GGBook. Moreover, the research questions will be answered and followed by a conclusion as well as further information such the limitations of the study and some future investigations.

Discussion

Introduction

Like explained in the methodological part, our chosen method, anasynthesis, is divided in 4 main stages. After an identification, analysis and synthesis that we have build through our literature process, and a validation step through our empirical part (case study of GGBook), it is now time to iterate the process.

This is why, in this discussion part, we will develop a general discussion as well as proposing improvements for both the PETESE and the GGBook. This is nothing more than a logical continuation of our work, like it can be seen in the following table:

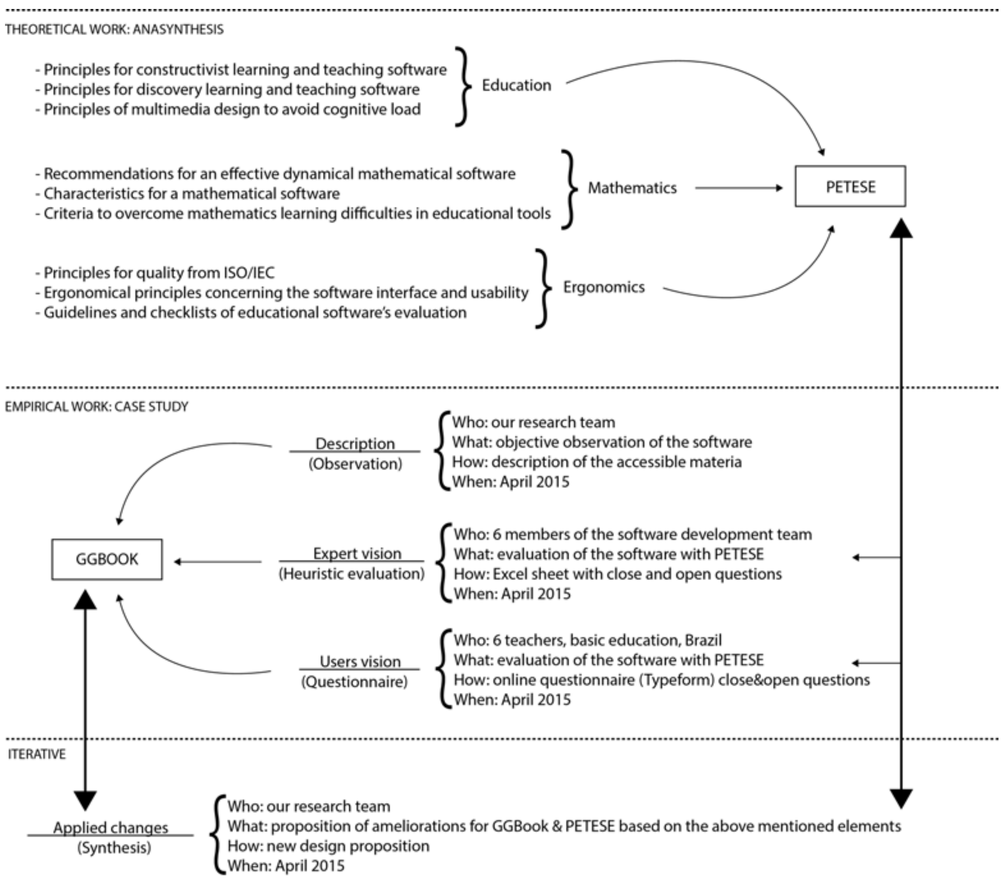


Figure 40 : Iterative process in our research

General discussion and improvements of the PETESE

The application of our theoretical developed tool to a real case brought several elements forwards about our PETESE. The questions of the experts, the once's of the users and our observations of analysis will compose our general discussion about our evaluation tool.

Both, for the six experts and the five users, the PETESE is something interesting. Indeed, they founded the referential much more complete than expected. It covers well the different aspects of a system from an educational point of view until the usability criteria. According to them, the fact that those criteria are based on scientific literature gathered on a systematic method, is also appreciated and from high interest in this new field of educational software engineering.

Moreover, instructional designers and developers are aware of the crucial role of tests in the creation process. However, the way of doing it as well as the aspects to take into consideration, are not always clear. In that context, this tool highlights well the different elements a mathematical software needs to integrate and allows the development team to think about each aspect. "If that aspect is present in the software, it is a way to see if it can be improved, if it is not present, it allows us to discuss and reflect on it, by seeing if it needs to be added or not, which results than more in reflected choices than in omissions", wrote one of our cases.

Nevertheless, all criteria are not always applicable to each educational software, which makes the analysis often confusing. According to others, however, even if one criteria is not applicable for the software, it is always interesting to have it mentioned. This can always bring up new reflections and lines of thoughts.

What the users mainly communicate is the problems of the questions. Lack of clarity and repetitiveness are two arguments that the teachers highlight. However, no more details are given, which leads us to questioning the questions of our questionnaire and our methodology. The experts of the developing team of GGBook concord as well that the questions are not always clearly explicit when referring to the software or the user.

Based on those feedbacks, we want to go back to our original PETESE and bring some modification to it as well as redesigning it for more generalization.

Our new proposition of the PETESE is modular. Indeed, different categories of modules can be put together and generate a specific evaluation. Five main categories exists like it can be seen in Figure 41, corresponding to the five colored tabs. The aim is to build a personalized evaluation according to the educational software that needs to be evaluated. This can be done by a simple drag and drop of the modules (left part) and a hang-up of them on the PETESE (right part).

The first tab (red) contains the type of evaluation and the type of evaluator the PETESE has to adopt. We have imagined different types of evaluations such as a questionnaire, a scale rating, a checklist or a group evaluation. Depending on this choice, the PETESE will adopt the chosen evaluation style. In this tab, it is also important to select the type of evaluator. Until now, we have imagined internal evaluators (people of the development team) and external evaluators (future users for example). This choice will also adapt the following modules because the nature of the criteria are not the same. This differentiation is a main result of what has been observed in our case study with GGBook where the users are not able to respond to more technical criteria for example.

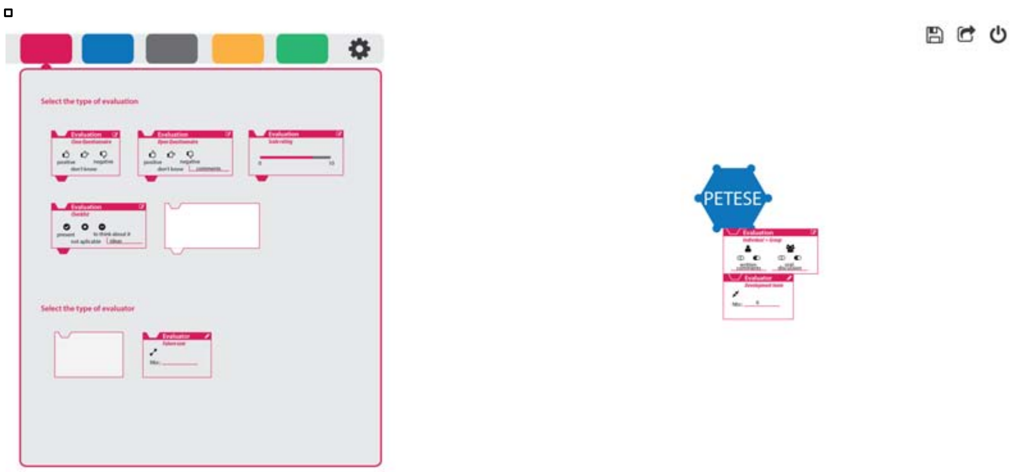


Figure 41 : Reviewed PETESE > Evaluation and evaluator tab

The second tab contains the six modules that according to us compose the PETESE : general aspects, pedagogical aspects, usability aspects, technical aspects, content aspects and interface aspects. Those bleu modules gather the basic criteria for any educational software evaluation. They are, based on the literature, the most important elements for the pedagogical usability. The person who builds up the PETESE, however, can choose the orden of its appearance.

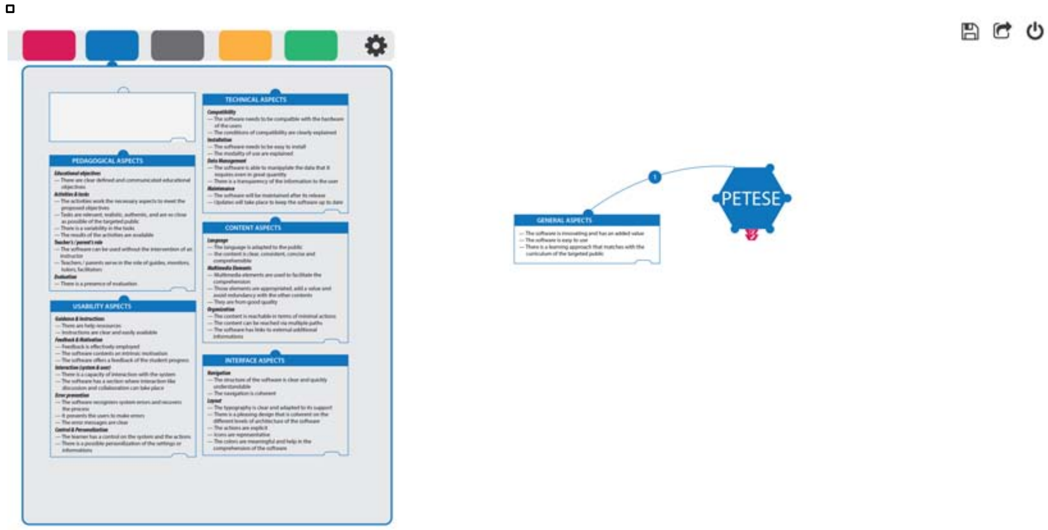


Figure 42 : Reviewed PETESE > The six basic modules

The grey tab contains all modules with more in depth criteria. They can be added to the bleu modules with the basic criteria. Indeed, this distinction has been made after the recommendations in our case study to make levels and graduations among the criteria. This way of separating the essential criteria from the more in depth once's is according to us, a way of prioritizing.

The yellow tab refers to the learning perspective of the software. Indeed, a software based on a behaviorist perspective does not involve the same pedagogical ergonomical criteria than if it was based on a constructionist approach. There, the perspective can be selected, and a selection of adapted criteria will appear. In our case, we have criteria from the discovering learning.

Finally, the green tab. This refers to the educational discipline. In fact, an educational software of mathematics does not have the same requirements as a software for the learning of languages. Like illustrated in Figure 43, the first step is to select the area of the educational software that is going to be analyzed and according to the choice, modules with particular criteria will appear.

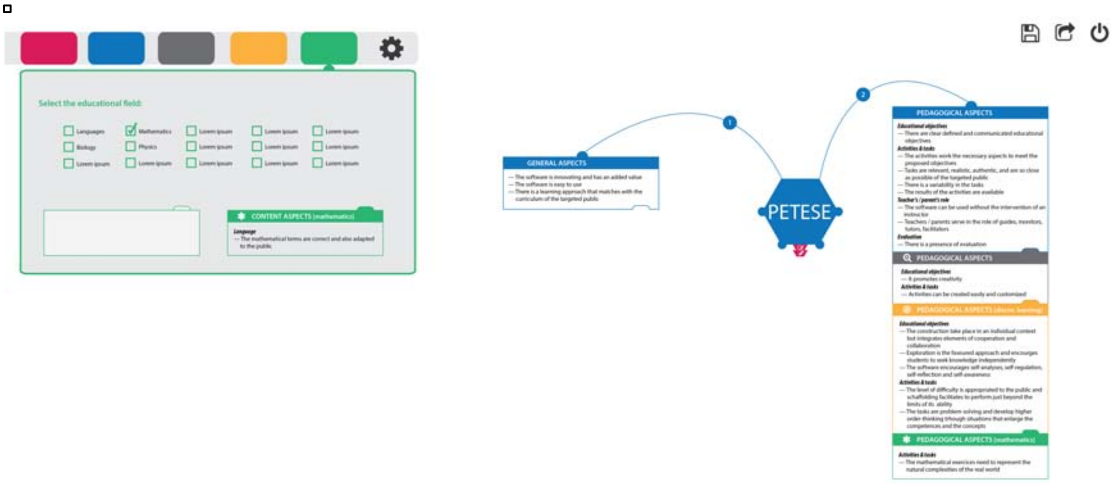


Figure 43 : Reviewed PETESE > The educational field specificity

After a personalized building of the PETESE according to the educational software to evaluate, the PETESE can look like the image below. In this case, we have taken our example for the mathematical educational software GGBook, based on a discovery learning approach. In that analyses we have selected a checklist model of the PETESE for the internal team (6 members). We have also chosen to take all basic models (bleu) with its in depth criteria (grey) on which we also joined the modules of discovery learning (yellow) and mathematics (green) like observable in Figure 44.

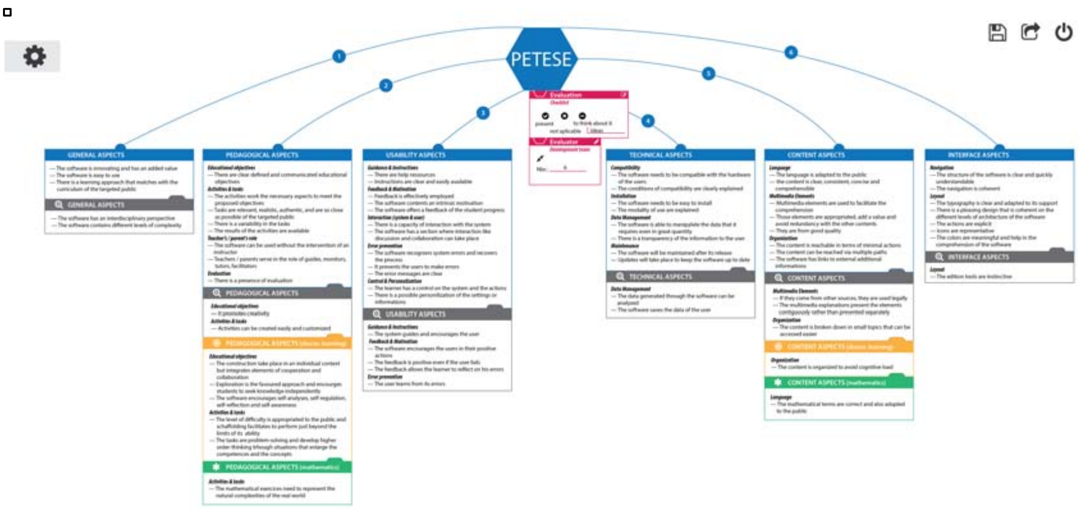


Figure 44 : Reviewed PETESE > Overview

Once the PETESE appears completed, it can be exported and all the modules will make up the personalized evaluation. In our example, it could look like this :

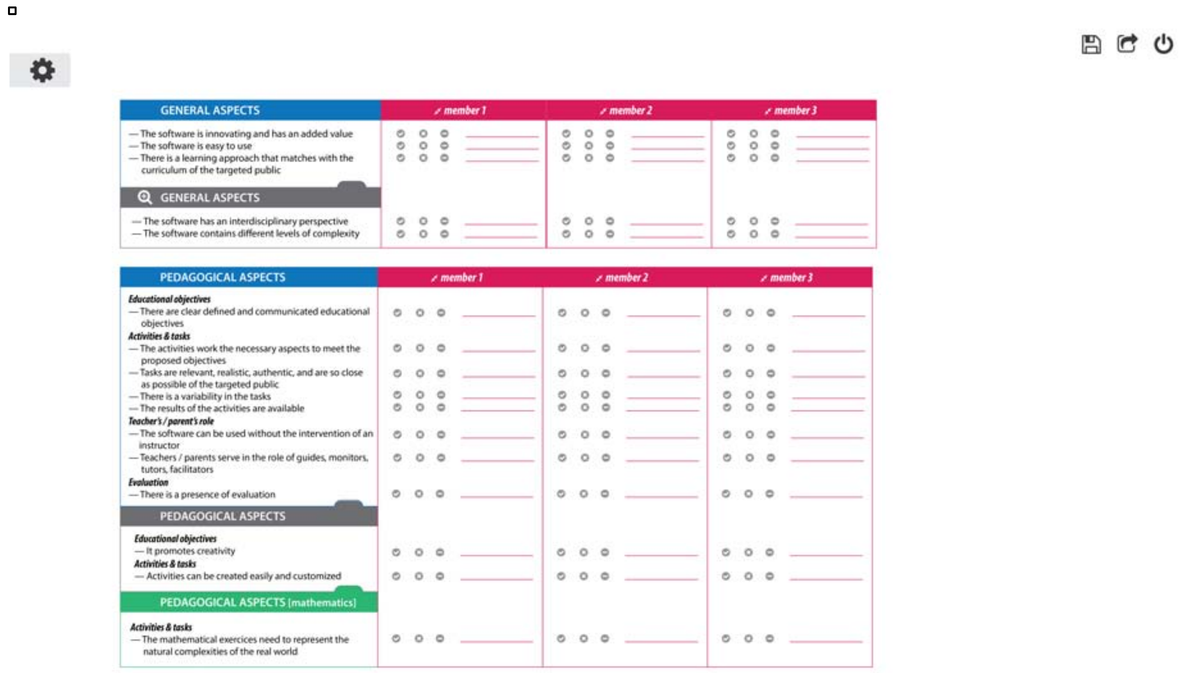


Figure 45 : Reviewed PETESE > Example of personalized built evaluation with PETESE

With this new proposition of PETESE, we not only wanted to respond to the highlighted problems observed in our empirical study with the software GGBook, but also in a larger vision, give a possible universal tool that could give one possible solution to the observed need in the literature of a specific evaluation with pedagogical usability criteria.

General discussion and improvements of GGBook

From the different answers of the case study, in other words, the objective observation, the experts points of view as well as the users points of view (answers of the PETESE as well as the open questions), we can say that the educational software of mathematics GGBook has its strengths and its weaknesses.

Among its strengths, the questioned people highlighted its added value with its interaction between the algebraic part and its graphic part. This free online software is also very open and can be adapted by the teacher to a variety of mathematical topics according to the level of its students. Those tasks are easy to create and customize with daily life explanations. GGBook also promotes creativity and encourages the students in resolving exercises independently through self-analysis and self-reflection. The clear and easily accessible content as well as the simple structure and layout are also seen a positive point.

The main weaknesses of GGBook lay in the facility to use the software since the beginning as well as the lack of guidance and instructions. The point of the evaluation also came a lot forewards because it is actually absent. Other mentioned points are the feedback and the motivation of the students when interacting with the program. Finally, some ask for more personalization.

In response to this, we have taken the mentioned suggestions and propose the following possible improvements to the software GGBook :

A clearer welcoming page with explicit objectives

Like mentioned in our empirical part of our study, GGBook does not have a real welcoming page. Indeed, the first contact with the online software is the login page. However, it could be interesting to have a presentation page that answers the following questions : – What is GGBook ? – For whom is it conceived ? – Why should those people use it ? – How does it work ? That same presentation page of course would keep a possibility of login.

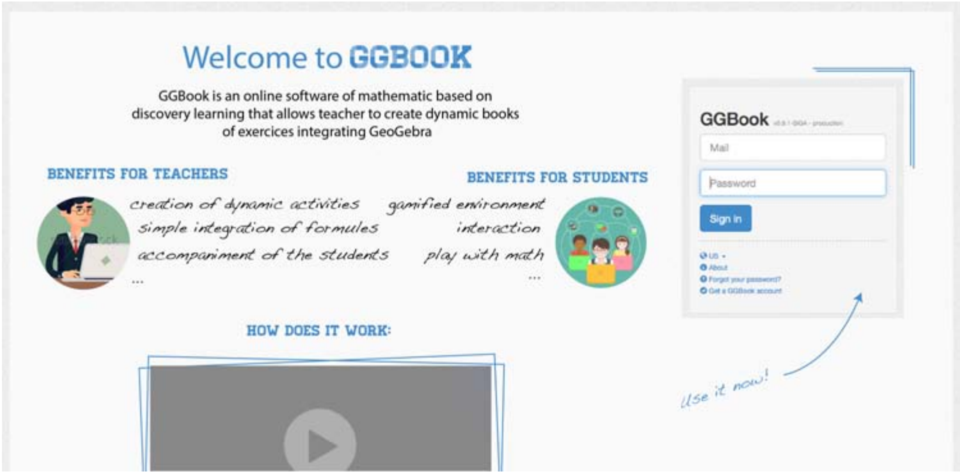


Figure 46 : Proposition of website of GGBook

A dashboard page in the program

When the user enters GGBook for the first time, he arrives on the pages of its books, in other words a white page because the new user never created anything. Like mentioned by the questioned people, this is source of loss for the user. The software has actually never a real homepage. Therefore, we suggest to create a kind of dashboard page that takes the major functionalities of the software as for example : the created books, the response of the students, the shared books, some tutorials,… This page would be what is seen after the login page and play the role of homepage across the software.

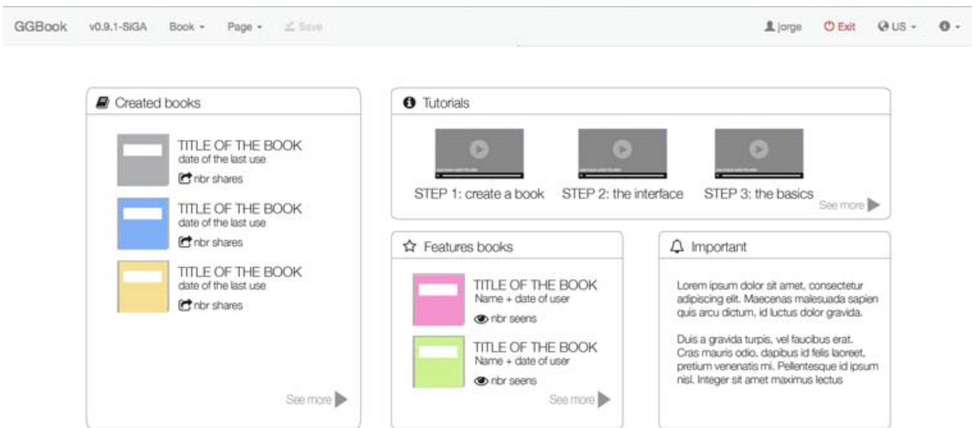


Figure 47 : Proposition of a dashboard in GGBook

The proposition of a guided tour of the software for the first login

After the creation of a first book, we suggest that the user has to possibility to follow a guided tour of the software. This happens through pop-ups that explain the different functionalities. It can for example explain the main tools of the algebraic part, the graphic representation part, as well as the main menu bar.

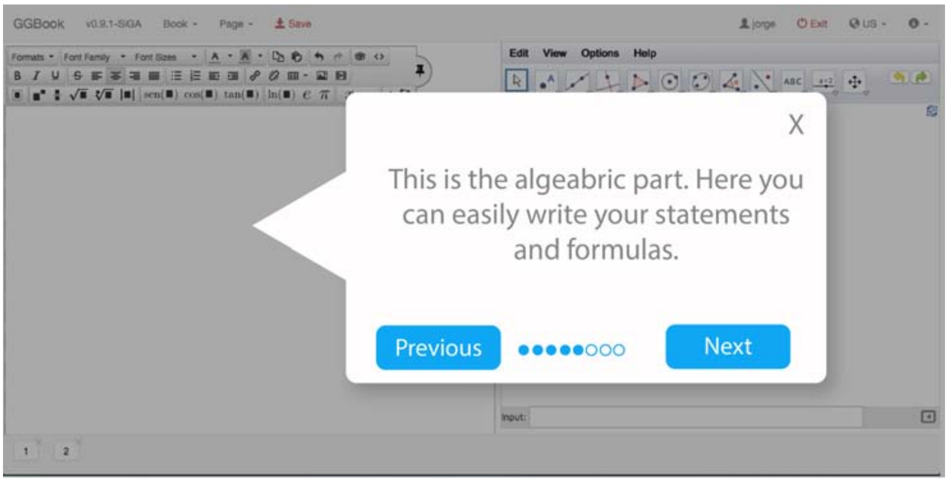


Figure 48 : Proposition of a step of the guided tour in GGBook

A higher personalization of the software

We suggest that the user could personalize the software more like he wants. This could be possible through the choice of a different background color for example or through the choice of the different modules presents on its dashboard as well as ordering it as the user wants.

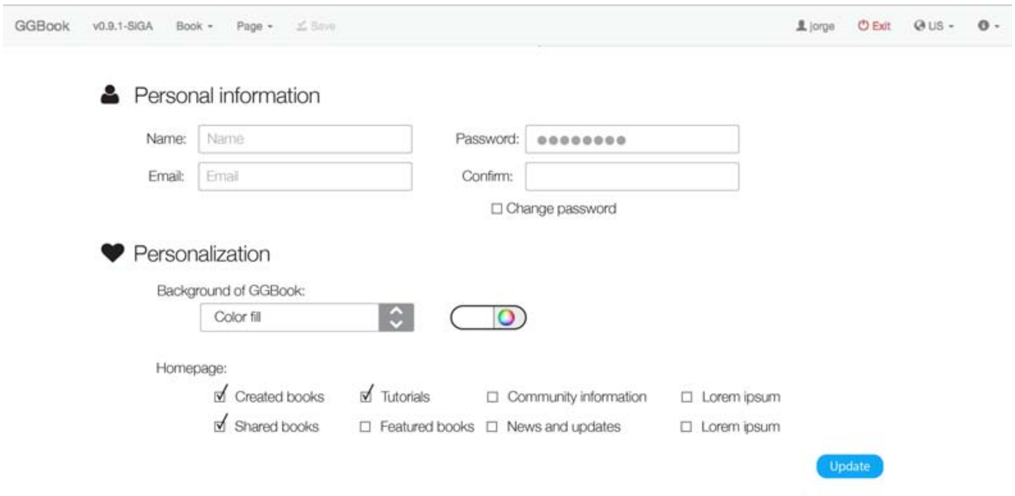


Figure 49 : Possibility of personalization's in GGBook

A safer exit mode

When the user exists GGBook, the software closes immediately. A message like "do you want to save your book/exercise/…" could be interesting and les stressful in case of a wrong manipulation of the system.

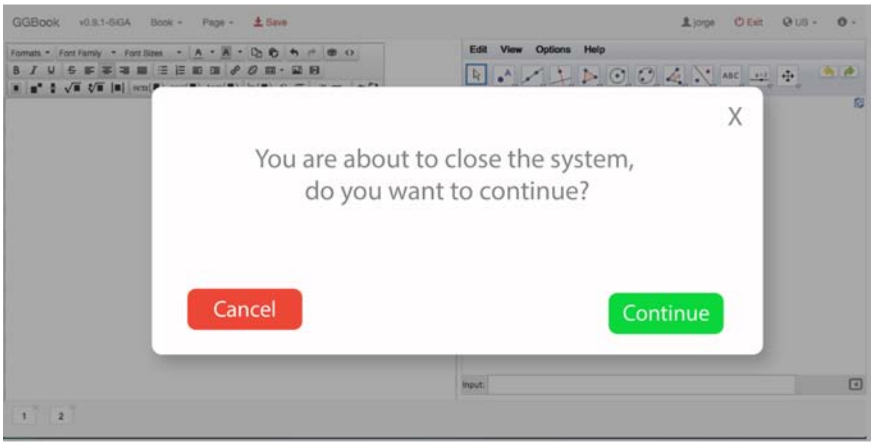
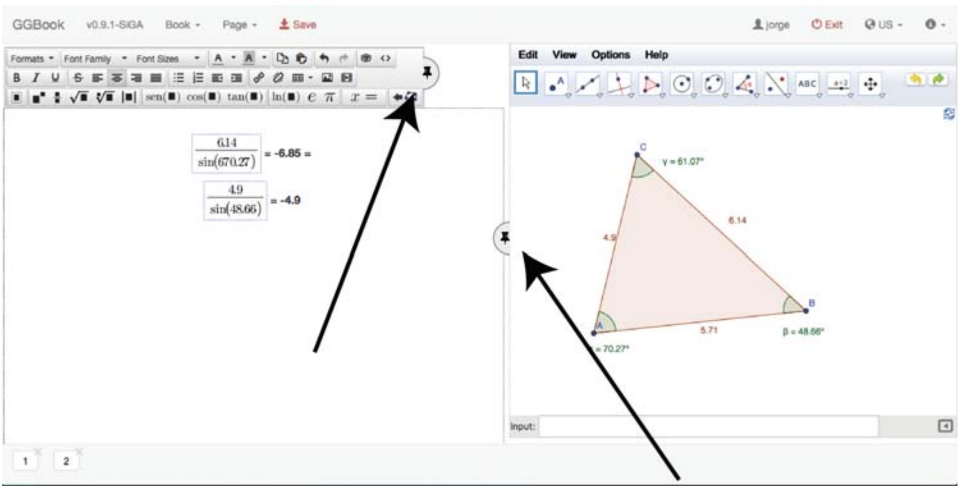


Figure 50 : Proposition of exit message

Controllable areas

The toolbar in the algebraic part is activated only when the user clicks in the algebraic field, and disappears when the user is in the graphical part. However, we think it could be much more interesting if the user itself could chose to maintain the toolbar visible or not. Moreover, when it is not visible, there is a button that would allow it to appear. This way of control on the areas could also be applied to the graphic part or the text part to favorite the working area and hide the area that is not used.



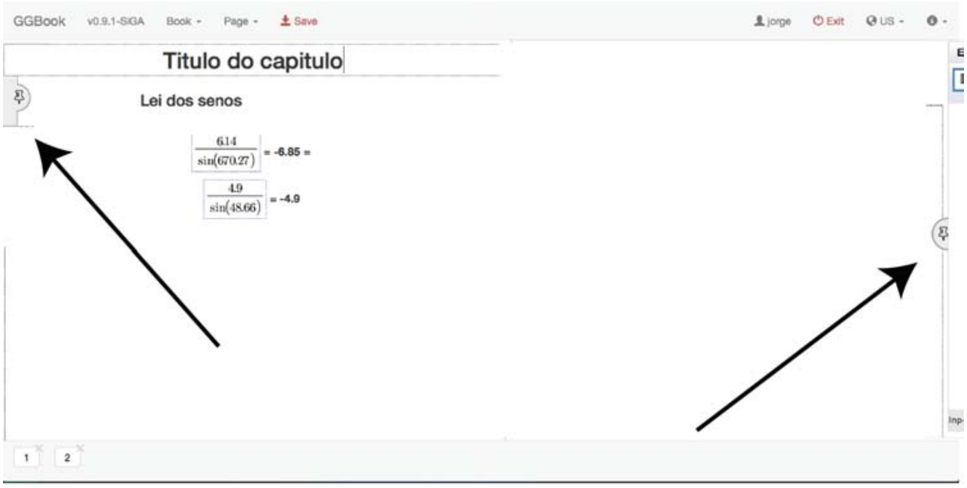


Figure 51 : Proposition of controllable area (open and closed)

Better keyboard manipulation for formulas

It is quiet difficult to understand in the beginning how the operations tools work in the algebraic part. Indeed, the intuitive way would be to do the maximum of the operations with the symbols available on the keyboard. However, this is not the logic of GGBook witch work by clicking on the icons of the toolbar. We suggest thus a better prediction of the system on what the user is writing. In the illustrator example, the user pressed the division sign on its keyboard and in reaction, the system suggests possible divisions.

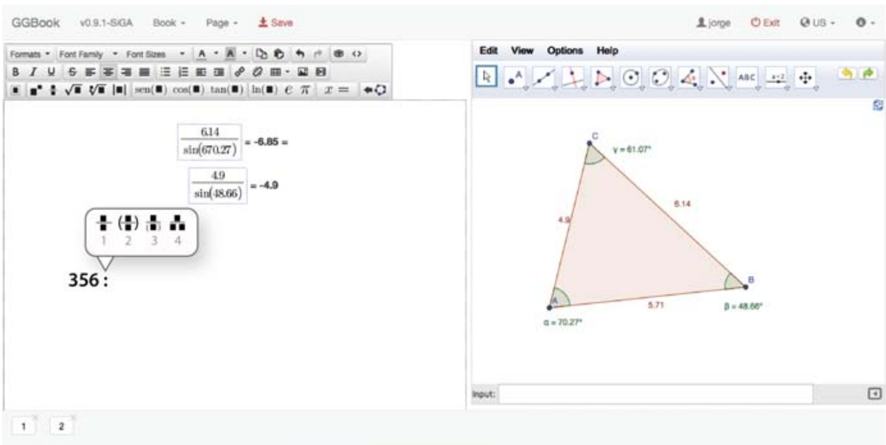


Figure 52 : Proposition of a better keyboard integration in GGBook

Closer integration of GeoGebra

Because GGBook is a text facilitator tool of GeoGebra, we think that a better integration to this latter could be done. Even though the connections between both development teams seem difficult, GGBook could integrate other elements of the GeoGebra community. This could be for example an access for the user to its account in GeoGebra, where this user can import its creations or other elements of the GeoGebra website and community.

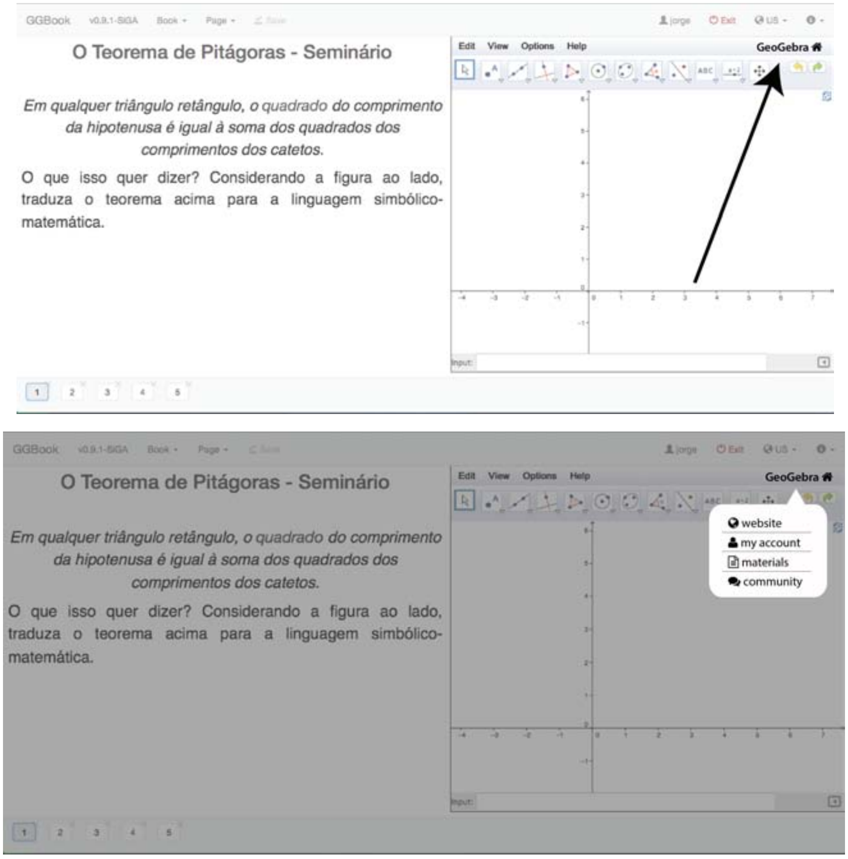
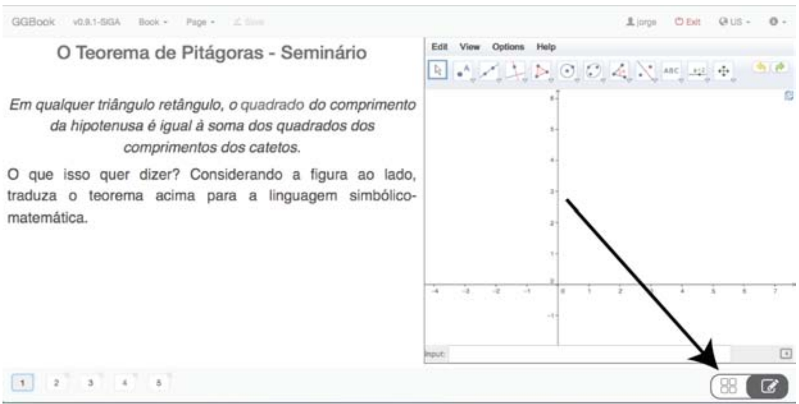


Figure 53 : Proposition of easy access to GeoGebra in GGBook

Overview view of the mathematical book

GGBook, like its name indicates, is a book that integrates pages of exercises. In the actual version of GGBook it is only possible to see the different pages one after the other, however, a global view does not exist. This is what we propose :



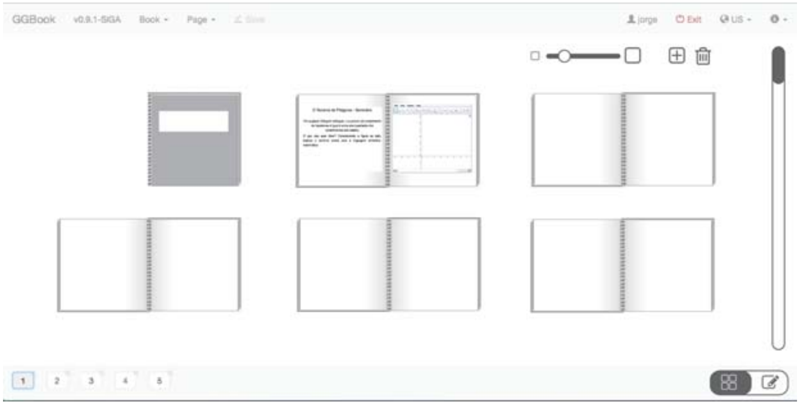


Figure 54 : Overview proposition for GGBook

Access to internet

One comment that has also been make to the software GGBook is its lack of opening to internet and other links to external sources of information. Therefore, we suggest a research engine that looks as well in the software as in GeoGebra, as on internet or other websites of mathematics. The illustrator bellow shows one way of doing this :

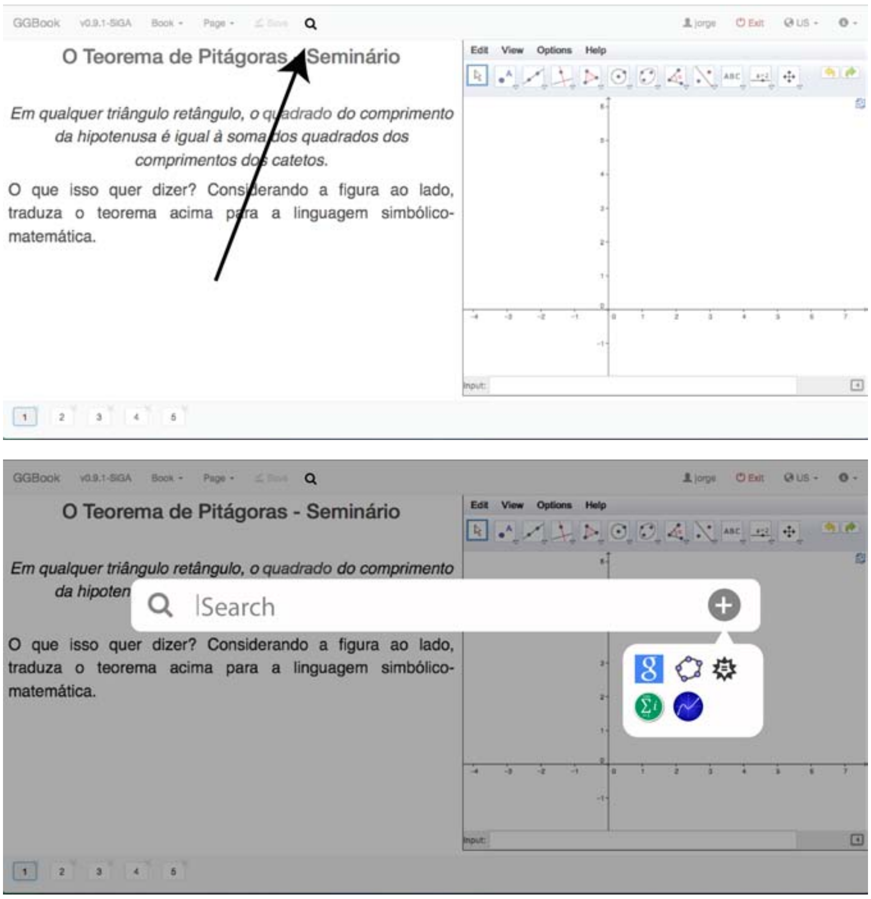


Figure 55 : Search option proposition integrated into GGBook

More motivation, guidance, feedbacks and interaction

To increase those mentioned elements, Bruno Ferreira (2015) in his master thesis on the use of gamification as a didactic strategy for the use of complex educational software proposes a plug-in called SiGA. This system of gamification for education leads the user to its objectives. Messages of guidance and feedback are given as well as recompenses through points and badges. According to the learning style of the user, SiGA will also offer options of collaboration or competition, which stimulate the users to accomplish their learning. We think that the application of SiGA to GGBook, will resolve the lack of motivation, guidance, feedback and interaction.

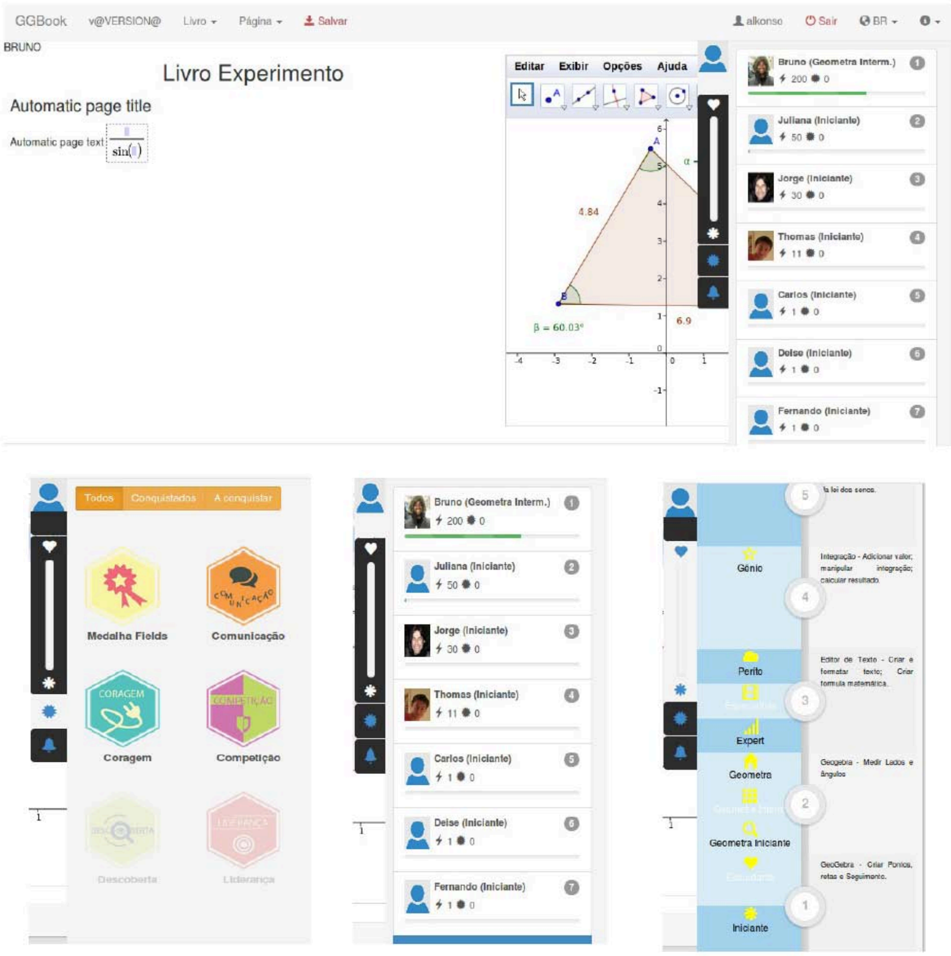


Figure 56 : The proposition of the plugin SiGA integrated into GGBook

Answering the research questions

With all the above mentioned elements, we can respond to the three research questions that have brought us to the development of an pedagogical ergonomical tool for educational software evaluation, the PETESE and its application to the educational software of mathematical learning GGBook.

RQ1 : Which are the elements that have to be taken into account while evaluating ergonomics of mathematical software of discovering learning ?

While evaluating the ergonomics of mathematical software of discovering learning or a general educational software, the most important element to take into account is the balance between ergonomical and pedagogical criteria. Indeed, not only the container (software) needs to be ergonomic, but the content (learning of mathematics) as well. This is why we have looked at ergonomical criteria from the three areas : ergonomy, pedagogy and mathematics.

Concretely, three categories of evaluation processes exists : standards, guidelines and checklists, and finally models. Standards regarding educational software evaluation do not exist and due to the complexity of fields that it requires. Therefore, in order to build our referential, we have looked at the existing standards that are related to our topic ; among them, design principles for multimedia user interface, ergonomics related to mental workload, guidance on usability and learning quality management.

Guidelines and checklists are very present in the literature regarding educational software. However, it has been observed that this overload of evaluation systems are quiet repetitive in terms of criteria. Moreover, most of them focus mostly on the ergonomical part or on a specific field of education. For this reason, we have analyzed and compared 15 checklists of educational software evaluation[14] and observed the most overcoming criteria. Those have been integrated in our evaluation tool.

Models of educational software evaluation are rare. Only few authors propose models and all have their limits : they are mostly very broad nay abstract or only take theoretical mechanisms without proposing any concrete applications. Therefore, we have wanted to propose an evaluation tool that

[14] MicroSIFT, Reeves and Harmon, Niquini, Coburn, Gamez, Silva, Squires and Preece, Crozat and al., Gladcheff and al., Campos and Campos, Oliveira, Gomes and al., Plaza and al., Da Silva and al., Cenci and Bonelli.

can be considered as a model with concrete application and adaptation according the educational software.

Finally, when evaluating ergonomically an educational software, we believe that this has to be done before its launch on the market, in other words, anteriorly and not posteriorly. Predictive evaluation is from huge importance and allows the developing team to focus on all elements of the software as well as already have the first feedbacks of the targeted-public.

According to us, all those above mentioned elements have to be taken into account for a quality evaluation of educational software.

RQ2 : What is the contribution in the literature in the field of pedagogical usability evaluation ?

The literature about pedagogical usability is growing those last years and quiet important in the Portuguese research area. It is however mostly related to the broader and most present area of ergonomics. To understand the developing of pedagogical usability, we have primarily analyzed the area of usability with its famous authors[15]. From there, like explained in the previous research question, we have analyzed the most appeared checklists and recommendations regarding pedagogical usability.

Lots of analyzing criteria founded in those evaluation tools are related to ergonomics. This is the case of criteria like : easy of use, compatibility, errors recognition, adaptability with the user, capacity of interaction, feedback, unrestricted learner control,… However, multimedia, didactic and scenarization criteria have also been highlighted. In tableTable 4, it can be observed that some pedagogical ergonomical evaluation tools take also into account the match with the curriculum of the targeted user, the presence of an evaluation, intrinsic motivation, learning from errors, cooperative learning, the interdisciplinary perspective, etc.

However, those checklists and guidelines are rarely complete, well presented, easy to use, direct applicable and specific for a specific educational field. Therefore, we have founded the necessity to create a new evaluation tool and we have come up with the PETESE, that is in a first place

[15] Smith and Mosier, Mayhew, Bastien and Scapin, Hix and Hartson, Nielsen, Rubin, Preece and al., Marcus, Scheiderman, Constantine and Lockwood, Welie, Tidwell.

specifically for the mathematics softwares based on a discovery learning perspective. In a second step, we have opened our tool and proposed an broader adaptation to all educational fields. We hope that this more concrete model will contribute to this field of pedagogical usability evaluation.

RQ3 : How can the above-founded results help the pedagogical usability of the mathematical software GGBook ?

The above founded results have allowed the construction of our evaluation tool, the PETESE. Its application to the educational software of mathematical learning, GGBook, has allowed us to highlight two important elements : an evaluation of the GGBook software that highlights its weaknesses and strengths as well as some improvements ; and in a second time, through this evaluation, we have been able to make an evaluation of our PETESE with its limits.

To respond the question regarding GGBook, the PETESE has highlighted the following elements :

GGBook is a software that is seen as innovating with an added value by its developing tool as well as its future-users. The major positive points are its promotion of creativity, encouragement in independently learning and its clear ad easily accessible content. Concerning the ergonomical criteria, the future-users find the interface, content and technical quiet good. The main negative points being the usability aspects of guidance and motivation, as well as the pedagogical aspects of the teachers role and evaluation. The experts evaluation gives more detailed results, and with the expertee of each specialized team member, other points are brought foreword, principally regarding the technical, usability and interface parts.

Next to the fact that the PETESE highlights the strengths and weaknesses of the software, it also brought important comments and suggestions of improvements foreword. It is the case of : better guidance with videos, a higher level of complexity, an access to the results of the activities, a way of more feedback, more comprehensive icons, external links with other mathematical content, integration of a chat, etc.

Those informations gathered by the PETESE about GGBook has been transformed in propositions of improvements of the software that are developed in the previous part.

CONCLUSION

General conclusion

The today's technological informational globalized world leads to new fields and ways of acting. Among them, the field of education that adapts to technology, which creates new opportunities of teaching and learning. From that junction appear the educational softwares that bring new elements foreword that a textbook by including multimedia elements, animation and interaction allowing namely self-teaching and a raise of curiosity. However, some educational software on the market are badly organized and contain ergonomical errors, who interfere in the learning process. Therefore, pedagogical usability is from huge importance during the creation process of new educational software.

The main goal of this research is to propose a pedagogical usability referential for mathematical software based on discovering learning in order to respond to the lack of concrete and complete evaluation tools of the literature, as well as applying it to the educational software of mathematics learning, GGBook. This evaluation tool has to particularity to be used during the designing process of the software, in other words before its release in the market.

To reach our objective, we have followed the process of anasynthesis, a method used to develop theoretical referential. The construction take place in four steps. After an identification of the problematic, an analysis of the existent criteria in the literature has been effected. This includes criteria of three different fields. Part one based on the field of education brought us principles for constructivist learning and teaching software, principles for discovery learning and teaching and principles of multimedia design to avoid cognitive load. The second part is the part of mathematics with recommendations for an effective dynamical mathematical software, characteristics for a mathematical software and criteria to overcome mathematics learning difficulties in educational tools . Finally, the third part of our literature review, based on ergonomics, highlighted principles for quality from ISO/IEC, ergonomical principles concerning the software interface and usability and finally, guidelines and checklists of educational software's evaluation.

The synthesis of the above mentioned references, has allowed us to construct our PETESE, pedagogical ergonomical tool of educational software evaluation (part 4 of our literature). In a second time, the tool has helped us in the ergonomical evaluation of the educational software of mathematics GGBook, a text editor and visual representation of the mathematical dynamical software GeoGebra.

The evaluation methodology of the software is the case study. This has been done through three processes. First, we have made an objective description of the software based on direct observations of the software as well as the related documents. Secondly, an expert vision has been applied. The six members of the development team of GGBook have been asked to evaluate the software with the PETESE through an excel sheet of open and close questions. Finally, five future users have answered an online questionnaire taking over the PETESE evaluation criteria. This empirical work allowed thus us on one hand to make an evaluation of GGBook, and on the other hand an evaluation of the PETESE itself.

Through this analyses the strengths and weaknesses of GGBook have been highlighted. Among its strengths, the questioned people highlighted its added value with its interaction between the algebraic part and its graphic part. This free online software is also very open and can be adapted by the teacher to a variety of mathematical topics according to the level of its students. Those tasks are easy to create and customize with daily life explanations. GGBook also promotes creativity and encourages the students in resolving exercises independently through self-analysis and self-reflection. The clear and easily accessible content as well as the simple structure and layout are also seen a positive point. The main weaknesses of GGBook lay in the facility to use the software since the beginning as well as the lack of guidance and instructions. The point of the evaluation also came a lot forewards because it is actually absent. Other mentioned points are the feedback and the motivation of the students when interacting with the program. Finally, some ask for more personalization. The experts and the future users also gave some comments and suggestions of improvements forewards such as: better guidance with videos, a higher level of complexity, an access to the results of the activities, a way of more feedback, more comprehensive icons, external links with other mathematical content, integration of a chat, etc. Those have been taken into consideration in our discussion and advice for improvements of the software.

The application of our theoretical developed tool to a real case brought several elements forwards about our evaluation tool itself. Reactions were positive regarding the concept of the evaluation tool and its complexity. The fact that those criteria are based on scientific literature gathered on a systematic method, is also appreciated and from high interest in this new field of educational software engineering. Moreover, instructional designers and developers are aware of the crucial role of tests in the creation process. However, the way of doing it as well as the aspects to take into consideration, are not always clear. The future-users also mention that all criteria are not always applicable to each educational software, which makes the analysis often confusing. Based on those comments, we have adapted our original PETESE and brought some modification to it.

We can conclude that the PETESE is from great help for the instructional designers and developers in the process of mathematical software design, because it takes all aspects into consideration before launching the software on the market. Moreover, its relevance can be showed by the acceptance of the presentation of this work at the International Event of Ergonomics in Las Vegas in July 2015. This study, however, has been limited to one practice evaluation. In that perspective we have proposed as well a more global version of the PETESE that could be applicable to other educational softwares of other fields. It should be relevant to test this to see the limits of the PETESE and improve it once more.

Limitations of our work and further research

Like all research study, our work has its limits. Firstly, this study has the usual limitations associated with any systematic literature review. In particular, the search process for the construction of the PETESE may have missed some relevant papers. With respect to the search process, we have limited ourselves to the most present articles that came out with the search terms related to the terms 'pedagogical usability evaluation'. This strategy covers thus weakly papers in many national journals and conferences as well as the papers that use unusual terminology. Overall, we do not expect to have missed the most important studies in our selection of the 10 keys authors of ergonomics and 15 most important of pedagogical usability, however, a more systematic or even algorithmically procedure could be made according the available articles in the literature on the above mention terms.

Secondly, our referential has its limits as well. Indeed, our tool has its limits of bringing the vast reality into criteria. In this process, a loose of information can arrive. Even more since the tool has to be conceived and quickly understandable. Other interpretations of that reality could maybe have been done.

Our third limit concerns the methodology. Indeed, in a usability evaluation it is from huge importance to have the future-users vision. For practical reasons, we have decided to do this through a questionnaire. However, we have observed some short answers that often frustrated us because we wanted to understand more in depth the argument of the person. Moreover, in the responses, we often have two completely opposite answers. To understand them better, we think another technique than the questionnaire should have been used. Therefore, we would be curious to see the results of the PETESE if it was orally using the focus group technique where the participants would debate about it. We think that it would bring huge contributions to the PETESE by looking at the miscomprehensions of the participants about the criteria for example, as well as for the analyzed software where we think that the participants would defend there position better.

Another observed limit in our methodology is the fact that some users (teachers) did not all remember very well of GGBook. Indeed, some did not use the software since a long time ago and have not looked at it before answering the questions. To improve this, we think that we should have done maybe a little activity that allows the teacher to remember and reuse quickly the program again before responding the questions. We think that this would certainly influence the responses. Moreover, according to Preece (1993), the data obtained by a software can explain on one hand how the users tackled the given task, where the major difficulties lie and what can be done; on the other hand, this method can highlight some performance measures like frequency of correct task completion, task timing, frequency of errors, etc. For this reason, we also believe that our data could maybe be completed by an indirect observation of that given activity on GGBook to the teachers.

Finally, we hope that further research will take our PETESE and apply it to other mathematical educational software or even other educational software in order to improve it continuously. In our new conception of the tool, further researches need to be done to complete modules, namely in the pedagogical approaches and the other educational fields.

Bibliography

Primary Sources

Reports and documents

GGBook online software, http://ggbook.com.br/.

Coomans, S., & Lacerda, G. (2015). PETESE, a pedagogical ergonomical tool for educational software evaluation. *6th International Conference on Applied Human Factors and Ergonomics (AHFE 2015) and the Affiliated Conferences, AHFE 2015.* Elsevier: Science Direct.

Questionnaires

Content Expert GGBook, Excel Sheet Pedagogical Usability of GGBook, Brasilia, 02/04/2015.

GGBook's Coordinator, Excel Sheet Pedagogical Usability of GGBook, Brasilia, 01/04/2015.

Interface Expert GGBook, Excel Sheet Pedagogical Usability of GGBook, Brasilia, 01/04/2015.

Pedagogical Expert GGBook, Excel Sheet Pedagogical Usability of GGBook, Brasilia, 05/04/2015.

Technical Expert GGBook, Excel Sheet Pedagogical Usability of GGBook, Brasilia, 04/04/2015.

Usability Expert GGBook, Excel Sheet Pedagogical Usability of GGBook, Brasilia, 02/04/2015.

User 1, Typeform Questionnaire Pedagogical Usability of GGBook, Brasilia, 07/04/2015.

User 2, Typeform Questionnaire Pedagogical Usability of GGBook, Brasilia, 09/04/2015.

User 3, Typeform Questionnaire Pedagogical Usability of GGBook, Brasilia, 09/04/2015.

User 4, Typeform Questionnaire Pedagogical Usability of GGBook, Brasilia, 13/04/2015.

User 5, Typeform Questionnaire Pedagogical Usability of GGBook, Brasilia, 19/04/2015.

Secondary Sources

Webography

GeoGebra. (2014). *What is Geogebra*. Consulted on 29/12/2014, www.geogebra.org/cms/en/info.

IEA. (2014). *Definition and Domains of Ergonomics*. (I. E. Association, Producteur) consulted on 23/01/2015, on www.iea.cc/whats/index.html

International Organization for Standardization. (2005). *SO/IEC 19796-1:2005: Information technology - Learning, education and training - Quality management, assurance and metrics* . Consulted on 18/02/2015 , on http://www.iso.org/iso/catalogue_detail?csnumber=33934

International Organization for Standardization. (1998 & 2006). *Ergonomic requirements for office work with visual display terminals (VDTs) - Part 11: Guidance on usability*. Consulted on 18/02/2015, on http://www.iso.org/iso/catalogue_detail.htm?csnumber=16883

International Organization for Standardization. (1999). *ISO/IEC 14598:1999: Information technology - Software product evaluation - Part 1: General overview*. Consulted on 18/02/2015, on http://www.iso.org/iso/catalogue_detail.htm?csnumber=24902

International Organization for Standardization. (2011). *ISO/IEC 25010:2011: Systems and software engineering- Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models*. Consulted on 18/02/2015, on http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=35733

International Organization for Standardization. (2011). *ISO/IEC 25040:2011: Systems and software engineering- Systems and software Quality Requirements and Evaluation (SQuaRE) - Evaluation process*. Consulted on 18/02/2015, on http://www.iso.org/iso/catalogue_detail.htm?csnumber=35765

International Organization for Standardization. (2011). *ISO/IEC 25040:2011: Systems and software engineering- Systems and software Quality Requirements and Evaluation (SQuaRE) - Evaluation process*. Consulted on 18/02/2015, on http://www.iso.org/iso/catalogue_detail.htm?csnumber=35765

International Organization for Standardization. (2002). *ISO/IEC 9126-1:2001 Ergonomics of human-system interaction - Usability methods supporting human-centred design*. Consulted on 18/02/2015, on http://www.iso.org/iso/catalogue_detail?csnumber=31176

International Organization for Standardization. (1991 & 2001). *ISO/IEC 9126-1:2001 Software engineering — Product quality — Part 1: Quality model*. Consulted on 18/02/2015, on http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=22749

International Organization for Standardization. (2001). *ISO/TR 10075:2014, Ergonomic principles related to mental work-load*. Consulted on 18/02/2015, sur http://www.iso.org/iso/catalogue_detail.htm?csnumber=66900

International Organization for Standardization. (2000). *ISO/TR 18529:2000, Ergonomics -- Ergonomics of human-system interaction -- Human-centred lifecycle process descriptions*. Consulted on 18/02/2015, on http://www.iso.org/iso/catalogue_detail?csnumber=31176

International Organization for standardisation. (2001). *IEC/TR 61997*. Consulted on 18/02/2015, on http://www.usabilitynet.org/tools/r_international.htm

International Organization for Standardization. (2012). *Software ergonomics for multimedia user interfaces - Design principles and framework*. Consulted on 18/02/2015, on http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=25578

Hohenwarter, M. (2010, 01 03). *GeoGebra et Tech Awards*. Consulted on 29/12/2013, on www.youtube.com/watch?v=rZnKMwicW_M

SoftPólis, n. S.-2. (2011, 01 15). *Ergolist* . Consulted on 05/03/2015, sur http://www.labiutil.inf.ufsc.br/ergolist/check.htm

Bibliography

Alessi, S., & Trollip, S. (2001). Multimedia for learning: Methods and development.

Alfieri, L., Brooks, P., Aldrich, N., & Tenenbaum, H. (2011). Does discovery-based instruction enhance learning? *Journal of Educational Psychology , 103* (1), 1-18.

Anthony, G., & Warlsaw, M. (2009). *Effective pedagogy in mathematics.* International Academy of Education. Brussels: Educational Practice series: 19.

Atayde, A., Teixeira, A., & Pereira, C. (2003). Uma Metodologia de Avaliação de Qualidade de Softare educacional Infantil. *XIV simpósio Brasileiro de Informática na Educação NCE* , 356-365.

Baumgartner, P., & Payr, S. (1998). Learning with the Internet: A typology of applications. *Proceedings of ED-MEDIA - World Conference on Educational Multimedia and Hypermedia* (pp. 124-129). Charlottesville: AACE.

Basque, J., & Lundgren-Cayrol, K. (2003). Une typologie des typologie des usages des TIC en éducation. *Télé-université* , 2-35.

Bastien, J., & Scapin, D. (1997). Ergonomic citeria for evaluating the ergonomic quality of interactive systems. *Behaviour and Information Technology , 16* (4), 220-231.

Boehm, B. (1986). A Spiral Model of Software Development and Enhancement. *ACM SIGSOFT Software Engineering Notes , 11* (4), 14-24.

Bruner, J. (1961). The act of discovery. *Harvard Educational Review , 31*, 21-32.

Curatelli, F., & Martinengo, C. (2012). Design criteria for educational tools to overcome mathematics learning difficulties. *Procedia Computer Science* , 92-102.

Campos, F., & Campos, G. (2001). Qualidade de Software Educacional. Dans A. Rocha, J. Maldonado, & K. Weber, *Qualidade de software: teoria e pratica* (pp. 124-130). Sao Paulo: Prentice Hall.

Carvalho, A. (2001). Usability testing of educational software: methods, techniques and evaluators. *The Turkish Online Journal of Educational Technology* .

Carroll, J. (1982). The adventure of getting to know a computer. *IEEE Computer, 15* (11), 49-58.

Cenci, D., & Bonelli, S. (2012). Criterios para avaliacoa de software educacionais. *IX Anped Sul, seminario de pesquisa em educacao da regiao sul.*

Claudio, L., De Lima, R., Ferreira, B., Lacerda Santos, G., & Nobriga, J. (2012). GGBOOK: uma interface que integrará os ambientes de texto e gráfico no GeoGebra. *1a Conferencia Latino Americana de GeoGebra*, 03-12.

Coburn, P. (1998). *Informatica na educacoa.* Sao paulo: Livros tecnocos e cientificos.

Constantine, L., & Lockwood, L. (1999). *Software for Use: A Practical Guide to the Models and Methods of Usage-Centered Design.* Boston: Addison-Wesley.

Coté, M., Suryn, W., & Georgiadou, E. (2006). *Software quality model requirements for software quality engineering.*

Crozat, S., Trigano, P., & Hû, O. (1999). EMPI: Une méthode informatisée pour l'évaluation des didacticiels multimédias. *Revue d'Interaction Homme-Machine , 1* (2), 61-87.

Crovello, T. (1982). Computer Software Evaluation: Who, When, Where, Why? *The American Biology Teacher , 44* (7), 429-446.

Eysenck, H. (1976). Introduction. Dans H. Eysenck, *Case stydies in Behaviour Therapy.* London: Routledge and Kegan Paul.

English Computorials. (1982). MicroSIFT courseware evaluation. *Northwest Regional Educational Lab.* , 1-87.

Ernest, P. (1995). The one and the many. Dans L. Steffe, & J. Gale, *Constructivism in Education* (pp. 459-486). New Jersey: Lawrence Erlbaum Assoc.

Duchastel, P. (1987). Structures and methodologies for the evaluation of educational software. *Studies in Educational Evaluation , 13*, 111-117.

Da Silva, A., Da Silva, W., & De França, R. (2011). Uma proposicao de criterios para avaliacao de softwares educativos de lingua portuguesa. *annals of XXII SBIE-XVII WIE* , (pp. 1104-1107). Aracaju.

D'Amboise, G., & Audet, J. (1996). Le projet de recherche en administration: Un guide général à sa préparation. Dans *Chapitre 4: L'approche holistico-inductive.*

De Alencar, R. (2007). *Proposta de um modelo para analise de requisitos de software educativo.* Brasilia: Di ssertação de mestrado apresentada à Faculdade de Educação da Universidade de Brasília como requisito parcial para a obtenção do grau de Mestre em Educação.

De Diana, I., & Van Schaik, P. (1993). Courseware Engineering Outlined: An Overview of Some Research Segues. *ETTI , 30* (3), 191-211.

De Souza, M., Filho, J., & Pequeno, M. (2004). *Um abordagem semi-automatica para a avaliacao comparativa de softare educational de matematica.* UFAM: Simposo Brasileiro de Informatica na Educaçao XV.

Drijvers, P., & Trouche, L. (2007). From artifacts to instruments: a theoretical framework behind the orchestra metaphor. Dans G. Blume, & M. Heid, *Research on Technology in the Learning and Teaching of mathematics: Syntheses and Perspectives.* Dordrecht: Kluwer Academic Publisher.

Fuchs, K., & Hohenwarter, M. (2005). Combination of dynamic geometry, algebra and calculus in the software system GeoGebra. 128-133.

Fey, J., & Hirsch, C. (1992). *NCTM Yearbook:Calculators in mathematics education.* Reston: NCTM.

Ferreira, B. (2015). *O uso da gamificacao como estratégia didatica na formaçao de professores para o uso softwares educativos complexos.* Brasilia: Universidade de Brasilia.

Fino, C. (2013). Avaliar software "educativo". *Actas da III Conferencia International de Tecnologias de Informaçao e Comunicacao na Educacao*, (pp. 689-694). Braga: Universidade do Minho.

Flyvbjerg , B. (2001). *Making Social Science Matter: Why Social Inquiry Fails and How it can Succeed again.* UK: Cambridfe University Press.

Frey, B., & Sutton, J. (2010). A Model for Developing Multimedia Learning Projects. *Merlot Journal of Online Learning and Teaching , 6* (2), 491-507.

Galvis-Panqueva, A. (1997). Software educativo multimidia aspectos criticos no seu ciclo de vida. *Revista Brasileira de Informatica na Educacao, 1*, 1-14.

Gamboa, F., Pérez, L., Galvez, D., Lara, F., Miranda, A., Cabiedes, F., et al. (2003). New Strategies in Educational Software. *Center of Applied Sciences and Technological Development, UNAM*, 26-31.

Gamez, L. (1998). *Ticese: tecnica de inspeçao de conformidade ergonomica de software educacional.* Portugal: dissertacao de mestrado submetida a universidade do Minho.

Garcia, I., Garcia, W., & Pacheco, C. (2009). A Cooperative Application to Improve the Educational Software Design using Re-usable Processes. *Springer-Verslag Berlin Heidelberg , LNCS 5738*, 93-100.

Gladcheff, A., Zuffi, E., & Silva, M. (2001). Um Instrumento para Avaliaçao da qualidade de softwares Educacionais de MAtematicas para o Ensino Fundamental. *Anais do XXI congresso da Sociedade Brasileira de Computacao.*

Gomes, A., Castro, F., Gitirana, V., Alves, M., Melo, M., & Ximenes, J. (2002). Avaliaçao de software educativo para o ensino de matematica. Florianopolis: WIE.

Grouws, D., & Cebulla, K. (2000). *Improving student achievement in mathematics.* International Academy of Education . Brussels : Educational practicies series: 4.

Heller, R. (1991). Evaluating software: a review of the options. *Computers and Education , 17* (4), 285-291.

Hennessy, S. (2000). Graphing investigations using portable technology. *Journal of Computer Assosted Learning , 16*, 243-258.

Hix, D., & Hartson, H. (1993). *Developing User Interfaces: Ensuring usability trhough Product and Process.* New York: John Wiley and Sons.

Hinostroza, J., & Mellar, H. (2001). Pedagogy embedded in educational software design: report of a case study. *Computers and Education* .

Hoyes, C., & Noss, R. (2003). What can digital technologies take from and bring to research in mathematics education? Dans A. Bishop, M. Clements, C. Keitel, J. Kilpatrick, & F. Leung, *Second International Handbook of Mathematics Education.* Dordrecht: Kluwer.

Honebein, P. (1996). Seven goals for the design of constructivist learning environments. Dans B. Wilson, *Constructivist Learning Environements.* New York: Educational Technology Publications.

Juristo, N., Moreno, A., & Sanchez-Segura, M. (2007). Analysing the impact of usability on software design. *The Journal of Systems and Software , 80*, 1506-1516.

Jonassen, D. (1994). Thinking technology: toward a constructivist design model. *Educational Technology , 34* (3), 34-37.

Kitchenham, B., & Pfleeger, S. (1996). Software quality: the elusive target. *IEEE , 13* (1), 12-21.

Kolb, D. (1984). *Experiential Learning: Experience as the source of learning and development .* New Jersey: Printice-Hall.

Komoski, P. (1987). Educational microcomputer software evaluation. Dans J. Moonen, & T. Plomp, *Eurit86: Developments in Educanional Software and Courseware* (pp. 399-404). Oxford: Pergamon Press.

Lacerda Santos, G. (2006). Alguns principios para situaçoes de engenharia de softwares educativos. *Inter-Açao: Rev. Fac. Educ. UFG , 34* (1), 17-36.

Lacerda Santos, G., & Lacerda, R. (2009). Proposta de um modelo analise de requisitos suscetivel de subsidiar o processo de desenvolvimento de softwares educativos. *Nuevas Ideas en Informatica Educativas*, 201-210.

Lawless, K., & Pellegrino, J. (2007). Professional development in integrating technology into teaching and learning: knowns, unknowns and ways to pursue better questions and anwers. *77* (4), 575-614.

Landry, N., & Auger, R. (2003). *Quelques stratégies de clarification conceptuelle et d'appui technique en vue d'une recension d'écrits dans un cadre de recherche en éducation.* Montréal: Rapport de recherche.

Lansdale, M., & Ormerod, T. (1995). *Understanding Interfaces: a Handbook of Human Computer Dialogue.* London: Academic Press.

Lê, Q., & Lê, T. (2007). Evaluation of educational software: Theory into practice. Dans J. Sigafoos, & V. Green, *Technology and Teaching* (Vol. Chap. 11, pp. 1-10). Nova Science Publisher.

Legendre, R. (2005). *Dictionnaire actuel de l'éducation.* Montréal: Editions Guérin Eska.

Libâneo, J. (2007). As tecnologias de comunicaçao e informacao e a formaçao de professores. Dans C. Silva, & M. Rosa, *Didatica e interfaces.* Editora Descubra.

Lindstrom, R. (1995). *Multimedia presentations.* NY: McGrawHill.

Lins, B. (2003). *Actual meanings, possible uses: Secondary mathematics teachers ad dynamic geometrics.* Paper presented at CERME-3.

National Centre for Technology in Education. (2008). *Advice Sheet 21 - Educational Software.*

National Council of Teachers of Mathematics. (2000). *Principles and Standards for School mathematics.* NCTM. Reston: Association Drive.

Niederhauser, D., & Stoddart, T. (2001). Teachers' instructional perspectives and use of educational software. *Teaching and Teacher Education , 17*, 15-31.

Nielsen, J. (1993). *Usability engineering.* Boston: AP Professional.

Niquini, D. (1996). *Informatica na Educaçao: implicacoes didactico-pedagogicas e construçao do conhecimento. .* Brasilia: Universidade catolica de Brasilia UCB.

Nobriga, J. (2015). *GGBook: uma plataforma que integra o software de geometria dinamica GeoGebra com editor de texto e equacoes a fim de permitir a construçao de narrativas matematicas dinamicas.* Brasili: Universidade de Brasilia.

Nokelainen, P. (2006). An empirical assessment of pedagogical usability criteria for digital learning material with elementary school students. *Educational Technology & Society , 9* (2), 178-197.

Mayer, R. (2003). *Learning and instructions .* Upper Saddle River, New Jersey: Prentice Hall.

Mayer, R., & Clark, R. (2011). *E-learning and the Science of Instruction: Proven Guidelines for Consumers and Designers of Multimedia Learning.* San Francisco: Pfeiffer.

Mayer, R., & Moreno, R. (2003). Nine ways to reduce cognitive load in multimedia learning. *Education psychologist , 38* (1), 43-52.

Mayhew, D. (1992). *Principles and Guidelines in Software User Interface Design.* New Jersey: Printice-Hall.

Maguire, M. (2001). Methods to support Human-centered Design. *International Journal of Human-Computer Studies , 55*, 587-634.

Malone, T. (1982). Heuristics for designing enjoyable user interfaces: Lessons from computer games. *Proceedings of the Conference on Human Factors in Computing Systems* (pp. 63-68). New York: ACM.

Manizade, A., & Lundquist, B. (2009). Learning about Proof by Building Conjecture. *The International Group for the Psychology of Mathematics Education , 5*, 1566-1571.

Marzita, P., & Rohaidah, M. (2004). Application of geometer sketchpad in learning mathematics. *Proceeding 2003 Regional Conference on Intergrating Technology in the Mathematical Sciences*, (pp. 193-203).

Marcus, A. (1995). Principles of effective visual communication for graphical user-interface design. Dans Baecker, Grundin, Buxton, & Greenberg, *Readings in Human-Computer Interaction* (pp. 425-441). Palo Alto: Morgan Kaufman.

Ministery of Education of Canada. (2003). *Early Math Strategy: the report of the expert panel.* Ontario.

Moretto, V. (2002). *Constructivismo: A producao do conhecimento em aula.* Rio de Janeiro: DP&A Editora.
Office of Technology Assessment. (1988). *Power on! New Tools for Teaching and Learning .* Washington DC: Us Government Printing Office.

Oliveira, N. (2001). Uma proposta para avaliacao de software educacional. *105f PPGEP/UFSC, Santa Catarina.* Florianopolis.

Oliveira, M. (1997). Vigotsky: aprendizado e desenvolvimento: um processo socio-historico. Sao Paulo.

Puustinen, M., Bakerz, M., & Lundz, K. (2006). GESTALT: a Framework for Redesign of Educational Software. *Journal of Computer Assisted Learning , 22*, 34-46.

Papert, S. (1990). Introduction. Dans I. Harel, *Constructionist Learning.* Cambridge: MIT Media Laboratory.

Papert, S. (1993). *The Children's Machine: Rethingking School in the Age of the Computer.* New York: Basic Books Inc.

Pereira, P. (2002). *Dynamic geometry: Dilemmas of teaching.* Paper presented at Technology and Education Conferences .

Piaget, J. (1952). *The origins of Intelligence in Children.* New York: International Universities Press.

Phillips, R. (1997). *The Developers handbook to Interactive Multimedia: A Practical Guide for Educational Applications.* London: Kogan Page.

Plaza, I., Marcuello, J., Igual, R., & Arcega, F. (2009). *Proposal of a quality model for educational software.* IEEE.

Preece, J. (1993). *A guide to Usability: Human factors in Computing.* Addisson Wesley, The Open University.

Preece, J., Rogers, Y., & Sharp, H. (2011). *Interaction Design: Beyond Human Computer Interaction.* John Wiley & sons .

Preece, J., Rogers, Y., Sharp, H., Benyon, D., Holland, S., & Carey, T. (1994). *Human-computer interaction.* Boston: Addison Wesley.

Pressman, R. (2006). *Engenharia de software.* New York: McGraw-Hill.

Sweller, J., Chandler, P., Tierney, P., & Cooper, M. (1990). Cognitive load as a factor in the structuring of teachnical material. *Journal of Experimental Psychology: General , 119*, 176-192.

Sauvé, L. (1992). *Eléments d'une théorie du design pédagogique en éducatin relative à l'environnement.* Thèse de doctorat, Université du Québec à Montréal.

Sanchez, F., Espinosa, P., Gonzalez, B., Poveda, L., Cifuentes, T., & Fernandez, I. (2004). *Herramienta de evaluacion de multimedia didactico.*

Scher, D. (2000). Lifting the curtain: The evolution of the geometer's Sketchpad. *The Mathematics Educator , 10* (1), 42-48.

Schneiderman, B. (1983). Direct manipulation: A step beyond programming languages. *IEEE computers , 16* (8), 57-69.

Schneiderman, B. (1998). *Designing the User Interface: Strategies for Effective Human-Computer Interaction.* Menlo Park: Addison Wesley.

Silva, C. (1998). *Bases pedagogicas e ergonômicas para a concepçao e avaliaçao de produtos educacionais informatizados.* Florianopolis: Dissertaçao para o Mestrado em Engenharia de Produçao, PPGEP/UFSC.

Silva, C., & Vargas, C. (1999). Avaliaçao da qualidade de software educacional. 1-15.

Silvern, L. (1980). Anasynthesis revisited: systems engineering in a time of education lethargy. *Educational Technology* , 40-46.

Shackel, B. (1991). Usability- context framework design and evaluation. Dans B. Shackel, & S. Richardson, *Human factors for informatics usability* (pp. 21-38). Cambridge: Cambreidge University Press.

Shewhart, W. (1931). *Economic control of quality of manufacture product.* Van Nostrand.

Smith, S., & Mosier, J. (1986). *Guidelines for designing user interface software.* Massachusetts: Mitre corporation.

Soloway, E., Jackson, S., Klein, J., Quintana, C., Reed, J., Spitulnik, J., et al. (1996). Learning theory in practice: case studies in learner-centred design. *CHI'96 Proceedings*, (pp. 189-196). ACM.

Squires, D., & McDougall, A. (1994). *Choosing and Using Educational Software: A teachers' Guide.* London: Falmer Press.

Squires, D., & Preece, J. (1996). Usability and learning: Evaluating the potential of educational software. *Coputers Education , 27*, 15-22.

Squires, D., & Preece, J. (1999). Predicting quality in educational software: Evaluating for learning, usability and the synergy between them. *Interacting with Computers , 11*, 467-483.

Strässen, R. (2002). Research on dynamic geometry software: introduction. *The International Journal on Mathematics Education , 34* (3), 65.

Stronge, J. (2007). *Qualities of Effective Teachers.* Alexandria, Virginia: ASCD.

Rubin, J. (1994). *Handbook of Usability Testing: How to plan, design, and conduct effective tests.* New York: John Wiley & Sons.

Ruthven, K., Hennessy, S., & Deaney, R. (2008). Constructions of dynamic geometry: A study of the interpretative flexibility of educational software in classroom practice. *Computers and Education , 51*, 297-317.

Ramos, J. (1998). *A criacao e utilizacao de micromundos de aprendizagem como estrategia de integracao do computador no curriculo do ensino secundario.* Evora: Dissertaçao de doutoramento.

Ramos, J. (2005). *Sistema de avaliacao, certificacao e apoio a utilizacao de software para a educacao e formacao .* Lisbon: Cadernos SACAUSEF.

Reeves, T., & Harmon, S. (1993). *User interface rating tool for interactive multimedia.*

Rifkin, J. (2011). *The Third Industrial Revolution.* Basingstoke: MacMillan.

Royce, W. (1970). Managing the development of large softwar systems. *IEEE Wescon* (pp. 1-9). The Institute of Electrical and Electronics Engineers.

Roschelle, J., Pea, R., Hoadley, D., Gordin, D., & Means, B. (2000). Chaning how and what children learn in school with compyter-based technologies. *Children and Computer Technology , 10* (2), 76-101.

Teoh, B., & Fong, S. (2005). The effects of geometer's sketschpad and graphic calculator in the Malaysian Mathematics classroom. *Malaysian Online Journal of Instructional Technology , 2* (2), 82-96.

Tidwell, J. (2005). *Designing Interfaces.* O'Reilly Media.

Thomas, G. (2011). A typlogy for the case study in social science following a review of definition, discourse and structure. *Qualitative Inquiry , 17* (6), 511-521.

Vergnaud, G. (1997). The nature of mathematical concepts. Dans T. Nunes, & P. Bryant, *Learning and teaching mathematics: An international Perspective* (pp. 5-28). Hove: Psychology Press.

Weigand, H.-G., & Weth, T. (2002). *Computer in Mathematikunterricht: Neue Wege zu alten Zielen.* Berlin: Spektrum Akademischer Verslag, Heidelberg.

Welie, M. (2003). *Asterdam Collection of Pattenrs in User Interface Design.*

Wilson, B. (1996). *Constructuvist Learning Environments: Case Studies in Intructional design.* Englewood Cliffs NJ: Educational technology Publications.

Winship, J. (1988). Software Review or Evaluation: Are they Both Roses or is one a Lemon? *Paper presented at the Proceedings of the Australian Computer Education Conference.* Perth.

Wittrock, M. (1991). Testing and recent research in cognition. Dans M. Wittrock, & E. Baker, *Testing and cognition.* Prentice Hall: Englewood Cliffs.

Yin, R. (2009). *Case Study Research: Design and Methods. 4th ed.* America: Sage Publications.

Zimmerman, B. (1990). Self-regulated learning and academic achievement: An overview. *Educational Psychologoist , 25* (1), 3-17.

Appendix

Content

For practical reasons, all documents are put on the CD available at the end of this thesis. The following content can be found on:

1/ The digital version of this research

2/ The primary sources

- Report
- Questionnaires
  - Development team
  - Future users