UNIVERSIDAD
NACIONAL
DE COLOMBIA
SEDE MEDELLÍN
FACULTAD DE MINAS

DYNA
http://dyna.medellin.unal.edu.co/

Agile human centered methodologies to develop educational software

Carina S. González-González, Pedro Toledo-Delgado & Vanesa Muñoz-Cruz

Escuela de Ingeniería y Tecnología. Universidad de La Laguna, La Laguna, Tenerife, Spain {cjgonza;petode;vmunoz}@ull.edu.es

Received: February 02nd, 2015. Received in revised form: March 24th, 2015. Accepted: September 28th, 2015

Abstract
This paper presents a methodological proposal that is based on powerful current approaches to the software development process: agile methodologies (SCRUM, Lean UX), user-centered design (UCD) and user experience design (UXD). Specifically, we have adapted some of the characteristics of the agile methodologies, which have allowed us to produce useful prototypes that are focused on the users while receiving constant feedback from them. The main goal of the proposed methodology is to identify usability problems and UX factors in the early stages of educational software development. Furthermore, we have also applied and validated the proposal during the framework development of an educational software project for users with special educational needs. A specific case taking into account key usability issues to support students with special education needs (SEN) is presented. In conclusion, we state that the proposed methodology could be a good design process philosophy to follow in the development of software projects in general, and in educational projects in particular.

Keywords: UCD, UXD, SCRUM, Human Factors, Usability, Prototyping.

Metodologías ágiles centradas en personas para desarrollar software educativo

Resumen
Este artículo presenta una propuesta metodológica basada en los enfoques actuales del proceso de desarrollo de software: las metodologías ágiles (SCRUM), centradas en el usuario (UCD) y el diseño de experiencia de usuario (UXDy Lean UX). Así, hemos adaptado algunas de las características de las metodologías ágiles que nos han permitido producir prototipos centrados en los usuarios con retroalimentación constante de ellos. El objetivo principal de esta propuesta es identificar problemas de usabilidad y factores UX en las primeras etapas de desarrollo de software educativo. Además, hemos aplicado y validado esta propuesta en el marco de un proyecto de software educativo para los usuarios con necesidades educativas especiales. La metodología propuesta puede ser una buena filosofía de diseño a seguir en los proyectos de desarrollo de software en general, y en proyectos educativos en particular.

Palabras clave: UCD, UXD, SCRUM, HCD, Agile UX, Lean UX, Factores Humanos, Usabilidad, Prototipado.

1. Introduction

The user-centered design (UCD) philosophy indicates that the software development process must be focused on the users rather than on the developers [1]. Nowadays there seems to be an increase in the concern regarding user satisfaction and in the willingness to provide a software solution that fulfills the needs of the costumer. This seems to be more important than including other functionalities that developers find useful and that generally only contribute to making the application more complex and less usable.

Furthermore, UCD is mostly approached by sticking with the four traditional phases of the software development (analysis of requirements, design, implementation and deployment), which usually means development based on a traditional heavyweight work methodology. The great novelty of the UCD is that it pays special attention to the customers and final users of the product. Hence, prototypes are included as a crucial part of the process, as they are fundamental to understand requirements and to improve

DYNA 82 (193), pp. 187-194. October, 2015 Medellín. ISSN 0012-7353 Printed, ISSN 2346-2183 Online
DOI: http://dx.doi.org/10.15446/dyna.v82n193.53495

usability. These prototypes are continuously validated during meetings, in which this functional software is examined carefully by the experts, customers and final users, usually including a usability test with them [2].

Moreover, rapid or agile development models [3] are being introduced to usability engineering. This approach is based on the mix of agile processes (especially Scrum) with user-centered techniques (e.g. [4]). For example, consider design and user involvement inside sprints [5]. In contrast with the traditional software development model, the agile approach involves the user in the design, development and testing. Additionally, Agile methodologies such as Scrum [6] or XP [7] avoid fixed phases and downplay the role of the analysis and design stages, instead focusing on the implementation by an iterative approach. Therefore, requirements are met through client feedback on the implemented uncompleted software during several meetings. Design and architecture emerge out of refactoring, after coding [8]. Furthermore, agile methodologies, and Scrum particularly, provide us with a fixed duration for each one of the iterations, allowing the team to fix the sprints duration and, therefore, we are given a schedule and framework to develop certain amount of functionalities [9].

Some characteristics of agile methodologies can be very useful in the user-centered design. For example, the principle of producing useful prototypes is in keeping with the aim of user testing that it desired for new functionalities. In addition, one of the benefits of agile methodologies is that they welcome changing requirements, despite how advanced the project is [10]. Hence, lightweight development boosts individual motivation in the development team and avoids the excessive amount of documentation and rigid phases of traditional approaches that in general discourage and demotivate the development team. In terms of UCD, a lightweight methodology cannot be used in its entirety due to its quality of being developer-centric. This is a cause of the disagreement expressed by some authors; [11] nevertheless, there are many ideas that come from agile methodologies that could be found interesting while trying to start a project that has the aim of having an UCD approach and motivating the development team while at the same time speeding up the actual development. In fact, there are some authors [12], as well as companies [13] who have already successfully mixed UCD with an agile development methodology: with a high rate of satisfaction.

In relation to user centered design and Agile methodology, a philosophy has been developed recently: Lean UX [5]. This philosophy has three main components: design thinking, Lean production and Agile development. Lean UX considers the software as a hypothesis of what the user might need. So, the user test attempts to confirm if this hypothesis is valid or not. In this sense, the users must be represented in the development process [14,15]. The most common methods to represent users in agile software development are “personas” or “protopersonas”. These methods are actively utilized in user stories. User story is a de facto standard way to describe system functionality in Agile [16]. User stories refer to personas who desire to achieve identified goals through specific actions. As such, our approach uses the “personas” method.

Thus, in the design and development of our educational software projects, we found that it is even more important for designers to have a framework for identifying usability problems in early designs and also in more mature prototypes. So, we decided to adopt some interesting ideas from user-centered design (UCD), user experience design (UXD) and agile methodologies (SCRUM, Lean UX) [17] in order to be able to produce useful prototypes from an early stage [18-20]. Additionally, our proposal also allows us to focus on the software design by making use of the users’ requirements.

In this paper we use our methodological approach to investigate the interaction design and usability of an interactive interface to solve addition and subtraction algorithms tailored for students with learning difficulties: Down syndrome (DS). We focus on finding the human factors in software design relating to the particular characteristics of DS students. The rest of the paper is organized as follows: first, we describe the Agile UXD approach that applies to our project. We then present the educational software design and studies on it (participatory design, persona technique, prototyping testing, usability testing) that have been carried out by experts. Finally, conclusions are presented.

2. Development of an educational software with agile methodologies

The most important roles in the learning process are those of the teachers and students. Thus, we will have to pay attention to two different roles involved in the development of e-learning platforms: the learners and the educational experts. Therefore, to be successful, the development of e-learning tools should follow a user-centered design that is focused on the two roles within the e-learning projects.

From the educational experts we will obtain advice in the first stages of the development as they are familiar with the learning process and they know the best practices to ease the learning process. In the later stages we will take into consideration the opinion of the future users of the tool: the learners. In their case, the main source of feedback will be the usability tests that they will carry out. As in the case of standard UCD, this approach usually involves the use of heavyweight methodologies that not only usually produce fewer amounts of useful prototypes during the development process, but also make the meetings more few and far between, thereby reducing the participation of users and experts during the development stage. Having less useful prototypes and meetings, we cannot carry out enough usability tests to have the views of users and experts reflected in the system. Moreover, the changing requirements are much more difficult to introduced. These factors could lead to dissatisfied end users [20].

The development of e-learning applications through agile methodologies, in which the design is traditionally developer-centered, could result in a lack of usability tests with users and experts, as well as insufficient advice from the educational experts. This is often a very serious shortcoming for the project as frequently, when the experts and learners are set aside throughout the e-learning tools development process, the final product does not meet their needs. This means that it has been developed without their advice and this means dismal failure and a waste of time and money.

However, we strongly believe that better outcomes can be achieved in the development of e-learning systems with a combination of agile development, based on short sprints and the development of iterative incremental prototypes. These should have usability tests performed by learners and educational experts during various review meetings of different sprints. All the feedback provided by them should then be taken into consideration thus preserving a learner-centered design while undertaking an agile development. Hence, our final development methodology, especially after the first weeks of analysis and design, was a variation of SCRUM mixed with UCD. In our methodology, the sprints had a one-week duration and, as a result, always produced a prototype. The product backlog was set at the very first stage of the project and was enriched with new ideas throughout the successive sprints. Moreover, in every sprint planning meeting, a series of tasks from the product backlog were planned for the subsequent sprint, as a sprint backlog. Furthermore, we had sprint review meetings and sprint retrospectives. In the first of these two meetings, we reviewed and discussed the advances during the sprint, and we usually carried out some tests with educational experts and sometimes also with end users (children). In the sprint retrospectives the IT team reflected on the past sprint, considering possible improvements and solutions to the problems encountered. The roles in the meetings were rather similar to those of the original SCRUM, were the educators had the role of Product Owner and one member of the IT team was the Scrum Master.

The approach consists of several stages that are iteratively executed: requirements analysis, user analysis, prototyping and evaluation. UCD is specified in ISO 13407– Human Centered Design Processes for Interactive Systems (ISO 13407:1999) [21]. It is both a broad philosophy and variety of methods, but the important concept is that users are involved in the entire process [22]. In this sense, the roots of User Experience Design (UXD) [23] can be found in the principles of Human Centered Design, which can be summarized as:

- Positioning the user as a central concern in the design process.
- Identifying the aspects of the design that are important to the target user group.
- Developing the design iteratively and inviting users' participation.
- Collecting evidence of user-specific factors to assess a design.

In principle, UXD is no different from UCD [24]. However, UXD adds important dimensions, such as User Experience (UX) factors [25]. While traditional usability factors were largely related to performance and smooth interaction, new UX factors relate to effect, interpretation and meaning. Some UX factors, such as social and aesthetic aspects, are likely to be very different in character from the traditional concerns [26]. This presents UX practitioners with significant challenges in terms of identifying which UX factors they need to consider when embarking on a design project. In any case, it is usual that a design team will only be able to deal with a few critical UX factors that influence the suitability of the design for a typical usage situation. Consequently, a big challenge for design teams is to make sense of the available information during the early phases of the UXD process. In order to capture UX factors during the early development stages, we believe that heuristics can particularly be a fast and cost-efficient but still effective and accurate evaluation method of user experience.

Heuristic evaluation is one of the so-called expert-based usability inspection methods [27]. It is an efficient analytical and low-cost usability method that is to be repeatedly applied during a development process, starting at the very beginning of a project design circle [28]. In a traditional user-interface evaluation, three to five experts (in the field of the application, usability or both) inspect a system according to recognized and established usability principles. Thus, our research group is formed by computer science engineers, educators, graphic designers and psychologists, which allow us to test the tools with the experts. However, we also tested tools with children as final users, who were supervised by the interdisciplinary team to capture UX factor requirements from their usage.

In the next section, we describe the application of the agile human centered methodologies to a particular case: the EMATIC software design and development.

3. Educational software proposal: EMATIC

EMATIC is a multi-device web tool that is oriented towards digital tablets and mobile phones, to teach basic mathematical operations to children with SEN. The software is adapted to mobile devices and aimed at two distinct profiles: adults such as the specialists who need to create quick and easy educational and cognitive rehabilitation activities and SEN children who need friendly and fun software with which to perform the activities recommended by specialists.

Due to the special characteristics of our target population, for the design of our first prototype, we took into account the methodologies that guide universal and inclusive design [29-31]. Moreover, we also considered the seven principles proposed by the Center for Universal Design at North Carolina State University (1997) [32]:

- Equitable use: the design is useful and marketable to people with diverse abilities.
- Flexibility in use: the design accommodates a wide range of individual preferences and abilities.
- Simple and intuitive use: use of the design is easy to understand, regardless of the user's experience, knowledge, language skills, or current concentration level.
- Perceptible information: the design communicates necessary information effectively to the user, regardless of ambient conditions or the user's sensory abilities.
- Tolerance for error: the design minimizes hazards and the adverse consequences of accidental or unintended actions.
- Low physical effort: the design can be used efficiently and comfortably and with a minimum of fatigue.

- Size and space for approach and use: appropriate size and space is provided for approach, reach, manipulation and use regardless of the user's body size, posture or mobility.

Additionally, for the learning design, we consider students' main difficulties in specific learning areas. Errors provide us with information to improve the teaching-learning process in our system. Research with children without disabilities shows a very wide range of errors in basic mathematic operations [33,34].

Furthermore, several requisites were considered when designing the interface of our first prototype in an effort to overcome the problems faced by DS children. These were problems such as language problems, difficulty in the transference and consolidation of learning and fine motor skills problems.

Technologically, the system was developed in Python language and one can chose between MySQL, PostgreSQL and Oracle as a database engine. The interfaces are based on HTML5, CSS3 and JavaScript. This web platform gives access to different users of the application (students, teachers or administrators), and allows each role to access only the granted processes and assigned actions (Fig. 1).

Furthermore, the system has been designed with modular software architecture. The topic that is to be taught is defined in the *domain module*. The individual characteristics of the student and the individual knowledge about the domain are stored in the *student model*.

EMATIC also includes a module for creating educational activities, which allows the educational professionals to design custom activities for a pupil or group of students (*authoring module*). Those designs may be either completely fixed, or allow randomized variations of the activity in an adapted sequence to the student. This module allows pedagogical experts to be able to design and construct activities strategies (*tutor module*). These strategies may include very different behaviors such as: free navigation through activities, fixed sequences, conditional sequences and gamifications. Furthermore, teachers can manage users, groups and activities through the *management module*, and the students can complete their activities using the execution module. Finally, results may be explored using the *visualization module*. Individual record visualizations are especially suited to the use of students, parents or tutors and teachers. The visualization and analysis of aggregated group results may be accessed by teachers and education directors.

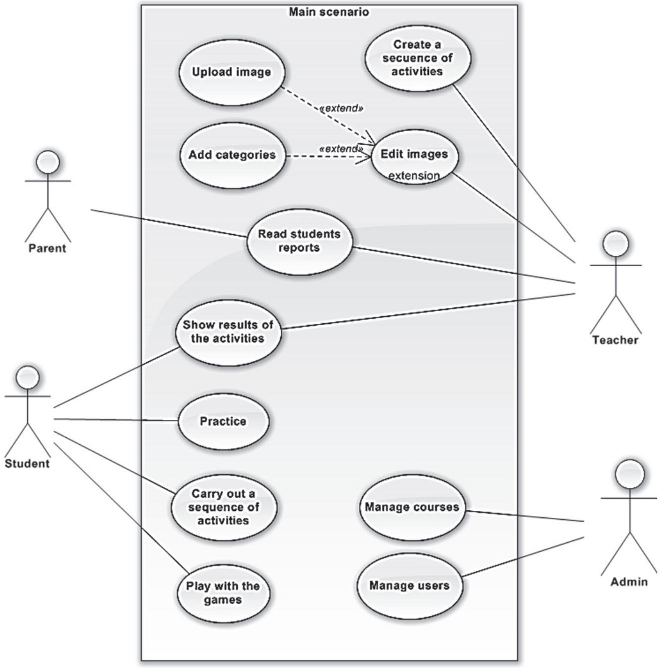

Figure 1. Use case diagram of the EMATIC scenario.
Source: The authors.

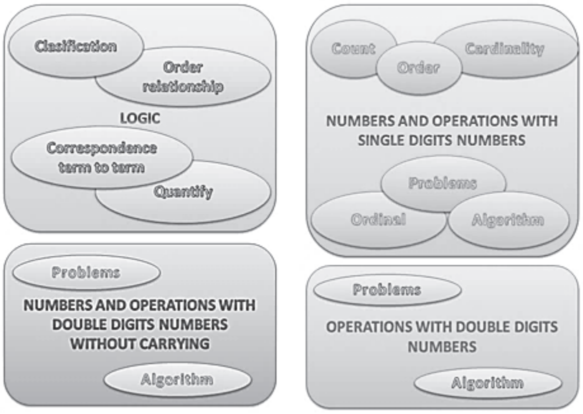
Clasification
Order relationship
LOGIC
Correspondence term to term
Quantify
Count
Order
Cardinality
NUMBERS AND OPERATIONS WITH SINGLE DIGITS NUMBERS
Problems
Ordinal
Algorithm
Problems
NUMBERS AND OPERATIONS WITH DOUBLE DIGITS NUMBERS WITHOUT CARRYING
Algorithm
Problems
OPERATIONS WITH DOUBLE DIGITS NUMBERS
Algorithm

Figure 2. Activity categories in EMATIC.
Source: The authors.

3.1. Domain Model's Module

This domain model's module contains subject knowledge and the rules and relations among its concepts. The tutor module can obtain the knowledge to be imparted from this module.

The EMATIC domain is made up from the concepts of logic, numbers and operations with single and double digit numbers, problems and addition and subtraction algorithms. Relating to these concepts, there are 10 different types of activities categories (classification, relationships, mapping, quantifiers, counting, recognition, cardinality, ordering, ordinality and problems) [35] and more than 150 task templates (Fig. 2).

3.2. Student Model's Module

The student model stores all the information on the pupil (knowledge, preferences, learning styles, etc.) It is composed by the student profile (persistent information such as cognitive age or disability) and the student's records (collected data through the interaction with the system). EMATIC software facilitates the processing of interaction data for cognitive reasoning tracing of the child during the undertaking of a task or set of tasks. So, based on the domain model, the student model is responsible for the generation of the student knowledge state.

3.3. Authoring Module

A setting up of the activities is possible in order to limit randomness when creating personalized exercises and group exercises for the students. The setting up of an activity depends on the particular definition of a type of activity in the system (classification, ordering, relations, etc.). Teachers can record a custom audio for the instruction in order to replace the "Text To Speech Agent" (TTS). They can also set up the maximum time for testing and the maximum number of attempts to pass the activity. Moreover, teachers can choose and replace the picture's collections to be used in the activity, selecting the most suitable for each type of activity.

3.4. Tutor Module

The tutor module contains knowledge about teaching strategies in order to adapt the strategy, taking into account the student module information (knowledge state and profile). Thus, the student model delivers the style which best suits the user, then the tutor module selects the most suitable pedagogical style according to the particular characteristics of each student. Moreover, this module has the gaming mechanics of the systems.

3.5. Management Module

In the management module, teachers can create and manage their students and their groups by assigning particular characteristics to students. Teachers may be in charge of several groups, for example, they can have different students in diverse institutions. Besides, teachers can assign the student's group with a new activity setting.

3.6. Execution Module

Typical systems only record the final results of the activities and save them. After, they analyze the interactions made during the resolution of a given task. EMATIC stores all data on user interaction, which is recorded ¨step-by-step¨ by the system in order to determine and infer the child's reasoning during the task execution.

Each exercise evaluates some cognitive ability. These exercises are randomly generated based on the collections and images loaded into the database. When a student selects an exercise, it is generated and then shown to the student. When the student responses, it is stored in the database with specific data (images that show collections relate, etc.) This defines the features that are shown to the student.

The results of the session are compiled and interpreted by the tutor module. The result obtained is the knowledge (or not) of the subjects taught. This also updates the goals that need to be fulfilled in the session.

3.7. Visualization Module

EMATIC includes a data visualization module, which provides a specialist tool to discover patterns in certain learning difficulties as a result of data mining. Moreover, teachers can analyze different groups of students with the same problem (for example, Asperger) (Fig. 3). Students can also see their results through the visualization module, but with a specific adaptation to their profiles.

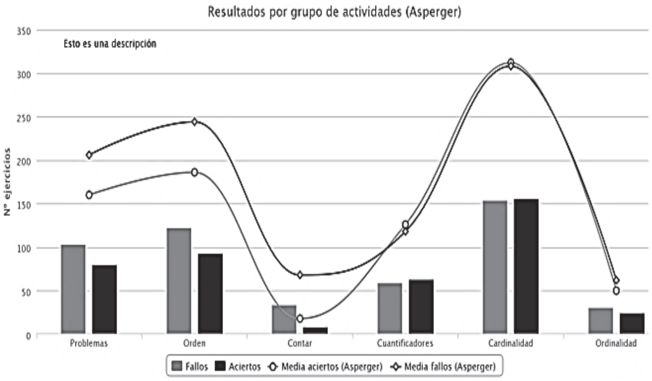

Figure 3. Results by activity group (features).
Source: The authors.

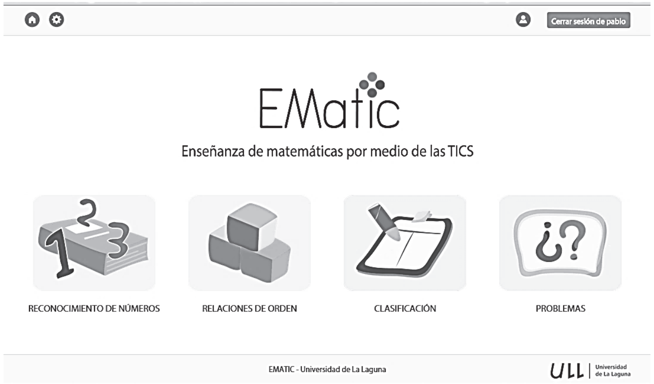
Cerrar sesión de pablo
EMatic
Enseñanza de matemáticas por medio de las TICS
RECONOCIMIENTO DE NÚMEROS
RELACIONES DE ORDEN
CLASIFICACIÓN
PROBLEMAS
EMATIC - Universidad de La Laguna
ULL Universidad de La Laguna

Figure 4. EMATIC welcome interface.
Source: The authors.

4. Evaluation

The EMATIC interfaces have been designed based on usability standards for both a mobile device (such as a tablet) and a desktop computer (Fig. 4). This allows for interaction with visual elements using mouse and touch on the tablet. However, we consider that is necessary to endeavor to overcome the problems faced by SEN children (such as language problems, difficulty in with transference and consolidation of learning and final motor skills problems).

As has been stated above, the seven principles proposed by the Centre for Universal Design at North Carolina State University (1997) have also been considered [31].

Researchers and designers' experiences are very different to those of SEN users, which make it difficult to identify with or relate to these users. Involving people with impairments in the design process is very challenging, especially when these impairments affect cognitive functions or communication. In order to achieve the set research goals, the participatory design (PD) approach [36] was used. The study involved a group of experts in mathematics, psychologists and teachers from University of La Laguna. Different kinds of techniques

Table 1.
Example of the "persona" template used in the validation.

Character		
Name	Luciana	
Age	16	
Gender	Female	
Education Level	Childhood Education (1st level)	
Context		
When?	She uses the computer from Monday to Friday, (30 minutes per session)	
Where?	1.1 Tenerife Down Association	
With who?	She uses the computer as a support her classes in the company of her teacher and classmates.	
Mission		
Goals	*General goal:* To develop her reasoning and to use this to solve problems in her daily life. *Specific goals*: a) To develop her levels of perception. b) To develop concepts of quantity. c) To develop logical thinking. d) To know the basic vocabulary of mathematical expressions. e) To develop her knowledge of and use of calculation (addition and subtraction) in simple operations.	
Description		
Luciana is 16 and has Down syndrome (moderate), she has been integrated into school and she is currently in an adult classroom. Quantitative data that can help determine Luciana's development: -Mental Age: approximately, 6 years -Verbal-Score: 37 -Manipulative-Score: 43 -IQ-55 (according to her chronological age)		

Source: The authors.

Table 2.
Heuristics evaluated

Heuristic / Severity Ranking	**1 %**	**2 %**	**3 %**	**4 %**	**5 %**	**6 %**	**7 %**	**8 %**	**9 %**	**10 %**	**11 %**
1	61,4	70,83	47,92	66,67	59,26	62,50	69,70	50	72,22	54,17	43,02
2	0	8,33	6,25	14,29	3,70	4,17	10,61	8,33	0	12,50	29,01
3	7,89	8,33	6,25	2,38	3,70	20,83	10,61	0	5,56	4,17	11,02
4	9,65	6,25	20,83	4,76	14,81	4,17	3,03	25,00	16,67	12,50	8,05
5	6,14	6,25	18,75	11,90	18,52	8,33	6,06	16,67	5,56	16,67	8,9

Source: Adapted from Nielsen and Pierotti [39].

were conducted with experts: persona [14,15, 37] (Table 1), prototype testing and interactive redesign and usability evaluation [39].

The tests were used to complement each other. The topics for testing were adapted from the 11 usability heuristics, with there being more than one hundred items to evaluate the user interface design (Nielsen [27] and Pierotti [39]).

We also conducted a focus group with experts, combining this with the persona technique [14, 15] in order to analyze the weaknesses and straightness of the interface, taking into consideration a user profile.

5. Results

The heuristics evaluated were the following:

1. The visibility of system status
2. Matches between systems and the real world
3. User control and freedom
4. Consistency and standards
5. Error prevention
6. Recognition rather than recall
7. Flexibility and efficiency of use
8. Aesthetic and minimalist design
9.Helping users recognize, diagnose and recover from errors
10. Help and documentation
11. Skills

Each heuristic was measured according the following severity, numbers becoming increasingly severe (possible answers):

1. No usability problem: "I don't agree that this is a usability problem"
2 - Cosmetic problem: "I think it's an esthetical usability problem"
3 - Minor usability problem: "I think it's a low usability problem"
4 - Major usability problem; important to fix: "I think it's a high usability problem"
5 - Usability catastrophe; imperative to fix: "I think it's a catastrophically usability problem"

The results of heuristic evaluation are shown in Table 2.

The system should inform users of system status, giving appropriate feedback within a reasonable timeframe. More than half of responders (61%) think that there is no usability problem, but other people think that there is an esthetical problem (15%). A catastrophic usability problem is lower than the other options (6%).

In relation to language, the system must use the language that users are familiar with, words or phrases that are known, rather than the terms that are used in the system, so that the user does not find it difficult to use the system. Most responders (71%) think that there is no problem with this usability issue.

When users choose an option from the system by mistake, the system should have the undo and redo options to provide users with an easy way out without them having to use extended dialogue. This is an important aspect to improve in EMATIC, which has a 19% critical usability problem and 21% of high usability problem.

Consistency is a fundamental design principle for usable user interfaces. It dictates that users should follow general rules and conventions in the system. Regarding this issue, however, the results show a low usability problem (67%). However, a catastrophically usability problem is moderately significant with 12%.

Error messages should be expressed in plain language, precisely indicate the problem and be constructive. On this issue, while the 59% of experts think that this is not a usability problem, 33% of respondents were found to have answered to have found it to be between a catastrophically and high usability problem.

Even better than good error messages is a careful design that prevents a problem from occurring in the first place. In this usability heuristic we found that 63% do not think that it is a usability problem; however, 12% think that it is high or extreme usability problem. So, a careful design is needed t prevents problems before they occur.

The system should minimize the amount of information that the user must remember. Instructions to use the system should be visible or accessible to the user when required. Ematic has made significant effort in this area, as the results show (70% of experts do not think that it is a usability problem).

Regarding the user efficiency and flexibility heuristic, it is important that the system allows itself to be customized. The results show us much more needs to be done on this issue (42% claim a high and extreme usability problem).

Design is much more than making something look pretty. Studies have shown that designs that look good have a higher probability of being used, and users also perceive that the system is easier to use [40]. The interface should not contain information that is not relevant or is rarely used, because each additional unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility. For this reason, we believe that the results obtained are positive (72% do not think that it is a usability problem).

The help and documentation of the system has to be easy to find, and focused on user tasks. 29% of experts think that this heuristic is a high or extreme usability problem. Therefore it is an important issue to improve in the next version of EMATIC.

To conclude, we analyzed the skills necessary to be able to fully use system. It is important to remember that the following data must be contextualized in a Down syndrome environment. Within this condition we have found the highest usability problems (57%). At the same time, this issue is a major challenge for the EMATIC system: to create a usable, accessible and adaptive system for the Down syndrome profile, taking into account their limitations and skills.

Moreover, as result of the focus group, the weaknesses of the interface that were detected by experts are the following:

- The interface is not available for people with hearing or visual disabilities, and this is a high usability problem.
- The randomness in the creation of activities causes several problems, for example: drawings of activities that cannot be understood or distinguished from the background, and exercises that have no logical pictures or semantic sense. Sometimes the interface is highly charged with objects, and on other occasions there are only a few objects.
- There is little accuracy when objects are being dragged and dropped in an area.
- It is necessary to give aural feedback along with the results of exercises.
- A more expressive virtual agent is needed.
- Sometimes the computer energy saver interrupts the activity.
- The tablet keyboard is complex to enter text answers (in our case, it is needed only for login).
- Regarding the skills needed to work with the tablet, the experts think that it could be difficult for people with SEN to work autonomously.
- It is easy to exit the program unintentionally.
- If the user makes a mistake it is not corrected.

Experts commented on the following issues being strengths:

- It is excellent to have specific support for tablets.
- It could be a highly motivating tool if applied properly.
- The advantage of the touch device is that it is a more intuitive interface for children or people with disabilities.
- The customization of the user interface and its adaptation to different levels and abilities are good features of EMATIC.

Another important issue that experts suggested was about the menus. They should be shown through pictures, not text-based and the access to the system should be through avatars related to each user.

6. Conclusions

In this paper we presented a software development approach based on agile human centered methodologies (SCRUM, UCD, UXD and Lean UX). Also, we focused on the usability analysis and human factors that influence the interface design.

Usability is an issue that is becoming increasingly important in software development. In the current environment, in which software systems are aimed at a special audience (SEN children) and increasingly at experts in a management system, usability is a fundamental attribute that dictates the success of a software application.

For this reason, it is important to invest in usability to build the system. It is very usual to find serious usability errors when the project is about to end, and then time may have to be extended in order to correct such problems. Once it is known how the user interacted with the system, and thus why the problems occurred, we can endeavor to improve the phase.

The most important usability problems found were about user control and freedom, user support and error recovery and skills needed to use the application. Moreover, our best findings were related with and esthetical, minimalist design and user language.

Currently, EMATIC is correcting the different bugs reported that affect flexibility and efficiency, design, support, languages and sounds, consistence, error prevention, etc. in order to meet special users and professional's expectations who interact with it.

Future work will look at the validation of the system in different areas of SEN children, having already improved the software.

Acknowledgments

We would like to thank the experts in the fields of Didactics of Mathematics and Psychology who have participated in this study. We are especially very grateful to the Tenerife Downs Association for their support and collaboration with our research.

References

[1] Cecil, R., Clash of the titans: Agile and UCD. UXmatters [Online] 2006. Available at: http://www.uxmatters.com/mt/archives/2006/12/clash-of-the-titans-agile-and-ucd.php.

[2] Colfelt, A., Bringing user centered design to the agile environment boxes and arrows. [Online] 2010. Available at:http://www.boxesand arrows.com/view/bringing-user.

[3] Martin, R.C., Agile software development: Principles, patterns and practices. Pearson Education. 2002. ISBN 0-13-597444-5.

[4] Memmel, T., Gundelsweiler, F. and Reiterer, H., Agile human-centered software engineering. Proceedings of the 21st British HCI Group Annual Conference on People and Computers, 1. pp. 167-175, 2007.

[5] Gothelf, J. and Seiden, J., Lean UX: Applying lean principles to improve user experience. O'Reilly Media, Inc., Sebastopol, CA, 2013.

[6] Takeuchi, H. and Nonaka, I., The new product development game. Harvard Business Review, 1986.

[7] Beck, K., Extreme programming explained: Embrace change. Addison-Wesley Professional, 1999.

[8] Neil, C.J. and Laplante, P.A., Paying down design debt with strategic refactoring. Computer, 39 (12), pp. 131-134, 2006. DOI: 10.1109/MC.2006.435

[9] Smite, D., Moe, N.B. and Gerfalk, P.R.J., Agility across time and space: Implementing agile methods in global software projects. Springer Publishing Company, 2010. DOI: 10.1007/978-3-642-12442-6

[10] Detweiler, M., Managing UCD within agile projects. ACM Interactions, 14 (3), pp. 40-42, 2007. DOI: 10.1145/1242421.1242447

[11] Armitage, J., Are agile methods good for design?. ACM Interactions, 11 (1), pp. 14-23, 2004. DOI: 10.1145/962342.962352

[12] Mcinerney, P., UCD in agile projects: Dream team or odd couple?. ACM Interactions, 12 (6), pp. 19-23, 2005. DOI: 10.1145/1096554.1096556

[13] Sy, D., Adapting usability investigations for agile user-centered design. Journal of Usability Studies, 2 (3), pp 112-132, 2007.

[14] Grudin, J. and Pruitt, J., Personas, participatory design and product development: An infrastructure for engagement. Proceedings of PDC, pp. 144-161, 2002.

[15] Pruitt, J. and Grudin, J., Personas: practice and theory. Proceedings of Conference on Designing for User Experiences (DUX2003), ACM Press, pp 1-15, 2003. DOI: 10.1145/997078.997089

[16] Liikkanen, L., Kilpiö, H., Svan, L. and Hiltunen, M., Lean UX - The next generation of user-centered agile?, Proceedings of the 8th Nordic Conference on Human-Computer Interaction (NordiCHI '14). ACM, pp. 1095-1100, 2014.

[17] Schwaber, K., Agile project management with SCRUM. Microsoft Press, 2009.

[18] Nielsen, J., Agile development projects and usability, Nielsen Norman Group, [Online] 2008. [date of reference February 25th of 2014] Available at: http://www.useit.com/alertbox/agile-methods.html.

[19] Nielsen, J., Agile user experience projects. Nielsen Norman Group. [Online] 2009. [date of reference February 25th of 2014] Available at: http://www.useit.com/alertbox/agile-user-experience.html.

[20] Nielsen-Norman Group. Report agile usability: Best practices for user experience on agile development projects, Nielsen Norman Group 2nd edition. [Online] 2008. [date of reference Feb 25th of 2014] Available at: http://www.nngroup.com/reports/agile/.

[21] Garreta, M. and Santanach, F., Open to the users' needs: Combining user-centered design, standards and open source software. Proceedings of open ED Conference, [Online] 2010. [date of reference Feb 25th of 2014] Available at: http://openaccess.uoc.edu /webapps/o2/handle/10609/4865.

[22] ISO 13407:1999. Human-centred design processes for interactive systems. International Organization for Standardization, Geneva, Switzerland.1999.

[23] Abras, C., Maloney-Krichmar, D. and Preece, J., User-centered design. Berkshire encyclopedia of human-computer interaction. Thousand Oaks: Sage Publications, pp. 763-767, 2004.

[24] Roto, V., Law, E., Vermeeren, A. and Hoonhout, J., User experience white paper. Bringing clarity to the concept of user experience. [Online] 2011. [date of reference February 12th of 2014] Available at:http://www.allaboutux.org/files/UX-WhitePaper.pdf.

[25] Larsen, J.M., Evaluating user experience – how game reviewers do it. Workshop on evaluating user experiences in games, at the 2008 Conference on Human Factors in Computing Systems (CHI '08). 2008.

[26] Constantine, L. and Lockwood, L.A.D., Usage-centered engineering for web applications. IEEE Software, 19 (2), pp. 42-50, 2002. DOI: 10.1109/52.991331

[27] Nielsen, J. and Mack, R.L. (Eds.), Usability inspection methods. John Wiley & Sons, New York, NY, 1994. DOI: 10.1145/259963.260531

[28] Stephanidis C. (Ed.) User interfaces for all – concepts, methods and tools. Lawrence Erlbaum Associates, Mahwah, NJ. 2001.

[29] Nielsen, J., Alternative interfaces for accessibility. [Online] 2003. [date of reference February 25th of 2014] Available at: http://www.useit.com/alertbox/20030407.html.

[30] Lazar, J. (Ed.), Universal usability: Designing computer interfaces for diverse user populations. John Wiley & Sons, 2007.

[31] Center for universal design. Guidelines for use of the principles of universal design. [Online] 2002. [date of reference February 25th of 2014]. Available at: http://www.ncsu.edu/www/ncsu/design/sod5/ cud/about_ud/udprinciples.htm.

[32] Roberts, G.H., The failure strategies of third grade arithmetic's pupils. The Arithmetic Teacher 15 (5), pp. 442-446, 1968.

[33] Cox, L.S., Systematic errors in the four algorithms in normal and handicapped populations. Journal for Research in Mathematics Education, 6 (4), pp. 202-220, 1975. DOI: 10.2307/748696

[34] Muñoz, V., Diseño e implementación de planificadores instruccionales en sistemas tutoriales inteligentes mediante el uso combinado de metodologías borrosa y multiagente. PhD. Thesis dissertation, Department of Computer Science, University of La Laguna, Spain, 2007.

[35] González-González C., Noda, A., Bruno, A., Moreno, L., Muñoz, V., Learning subtraction and addition through digital boards: A down syndrome case. International Journal Universal Access in the Information Society - UAIS, 29 pp. 29-44, 2013. DOI 10.1007/s10209-013-0330-3

[36] Muller, M.J., Participatory design: The third space in HCI. In Jacko, J. and Sears, A. (eds.), Handbook of HCI 2nd Edition. Mahway, NJ, USA: Erlbaum, 2007, pp. 1051-1068. DOI: 10.1201/9781410615862.ch54

[37] Nielsen, L., Engaging personas and narrative scenarios: A study on how user-centered approach influenced the perception of the design process in the e-business group at Astra Zeneca, København: Samfundslitteratur, 2004.

[38] Dow, S., Fortuna, J., Schwartz, D., Altringer, B., Schwartz, B. and Klemmer, S., Prototyping dynamics: Sharing multiple designs improves exploration, group rapport and results. ACM Conference on Human Factors in Computing Systems (CHI'11), pp. 2807-2816. 2011. DOI: 10.1145/1978942.1979359

[39] Pierotti, D., Heuristic evaluation, a system checklist, Xerox Corporation. [Online] 1995. [date of reference March 18th of 2014]. Available at: http://www.stcsig.org/usability/topics/articles/he-checklist.html

[40] Lidwell, W., Butler, J. and Holden, K., Universal principles of design. Beverly, MA: Rockport Publishers. 2003.

C.S. González-González, received her PhD. in Computer Science in 2001 from the University of La Laguna, Spain. She is an associated professor in the Computer Engineering Department at the University of La Laguna, Spain. Her main research focus areas are related to human computer interaction (HCI), natural and adaptive interfaces, serious games and gamification in Education and gender in HCI. Also, she has a great experience in e-learning best practices and LMS systems.

P. Toledo-Delgado, received his MSc. in Computer Science Engineering from the University of La Laguna, Spain in 2004. He has been a PhD. student at the same university since 2004, in the Computer Engineering Department. His current research interests include Reinforcement Learning, Graph Mining, and Workflow Induction.

V. Muñoz-Cruz, received her MSc. in Computer Science in 2001 and her PhD. in Computer Science in 2007 from the University of La Laguna, Spain. She is a professor in the Computer Engineering Department of the University of La Laguna, Canary Islands, Spain. Her current research interests are decision-making, discrete event simulation, intelligent agents, intelligent tutorial systems and computer education.