DEVELOPMENT METHODOLOGIES FOR EDUCATIONAL SOFTWARE: THE PRACTICAL CASE OF COURSEWARE SER_E

António Pedro Costa[1], Maria João Loureiro[1], Luís Paulo Reis[2]

[1] CIDTFF/DDTE/UA - Centro de Investigação em Didáctica e Tecnologia na Formação de Formadores, Departamento de Didáctica e Tecnologia Educativa, Universidade de Aveiro.
Aveiro, Portugal
apcosta@ua.pt; mjoao@ua.pt

[2] LIACC/DEI/FEUP – Laboratório de Inteligência Artificial e Ciência de Computadores, Departamento de Engenharia Informática, Faculdade de Engenharia da Universidade do Porto.
Porto, Portugal
lpreis@fe.up.pt

Abstract

Traditional software development methodologies, usually designated as engineering methodologies, are very bureaucratic, in what documentation and rigid control mechanisms is concerned, and not applicable to different software development projects. In the last years, new methodologies called "agile or light" as been purposed. This paper describes Hybrid User-Centered Development Methodology (HUCDM). The main assumptions of the methodology are related with the constitution of a multidisciplinary team, with various skills, as well as the evaluation of the resource, that involved external elements. The methodology, being used in the context of a small software enterprise, is also being evaluated (formative evaluation). That evaluation sleeked to ensure that the enterprise can develop educational resources with recognized quality that is simultaneously viable from the economical point of view. The paper presents a brief description of traditional and current software development methodologies, followed by a brief description of the developed resource – Courseware Ser_e - The Human Being and Natural Resources – and of the adopted development methodology. Some insights of the evaluation being performed are also presented.

Keywords – Educational Courseware, Software Development Methodologies, User Centered Design

1 INTRODUCTION

Software development is a very complex activity. In many cases, software is developed without being properly planned, supported by "short term" decisions [1]. This development approach may work for small applications/projects, but if the system grows, the difficulty to add new functionalities also increases. Moreover, Shneiderman & Plaisant claim that 60% of software development projects failed to achieve theirs goals [2]. The same authors point out that this problem exists because in most projects there is poor communication within the multidisciplinary team or between developers and users. The choice of a software development methodology usually takes into account economics and competitive advantages. If a wrong methodology is selected, the more likely will the project deadline be exceeded, there will be failures and, consequently, business loss. If we take into account the diversity of users, software purposes and constants changes in technology, adopting the same methodology for all the software development projects hardly will be a good choice.

In this contribution a development methodology is proposed that seek to address a specific need of a software development company (Ludomedia): to develop educational resources with recognized quality that are simultaneously viable from the economical point of view. This problem led to establishment of a partnership with University of Aveiro in Portugal.

After this brief introduction, section 2 presents a summary of software development methodologies, from traditional to current ones. In the section 3, we describe an educational resource, with emphasis on its functionalities, as well as the development methodology adopted. The methodology comprises: planning of educational guidelines, storyboard design, implementation, evaluation and operation and maintenance; and adopted some of the UCD principles. Finally section 5 presents the paper conclusions and points out possible future work.

2 METHODOLOGIES FOR SOFTWARE DEVELOPMENT

A software process is a set of activities that leads to the production of a software product. Key development stages, common in the majority of the software development processes are: software specification, software design and implementation and software evolution [3]. Recently, new approaches have emerged, designated “agile or light” processes, are replacing “bureaucratic or heavy” approaches [4]. In the next sessions we describe brief these methodologies.

2.1 Traditional methodologies

The first methodologies derived the most common process of system engineering processes. Figure 1, presents the process designated as waterfall model or software life cycle [3].

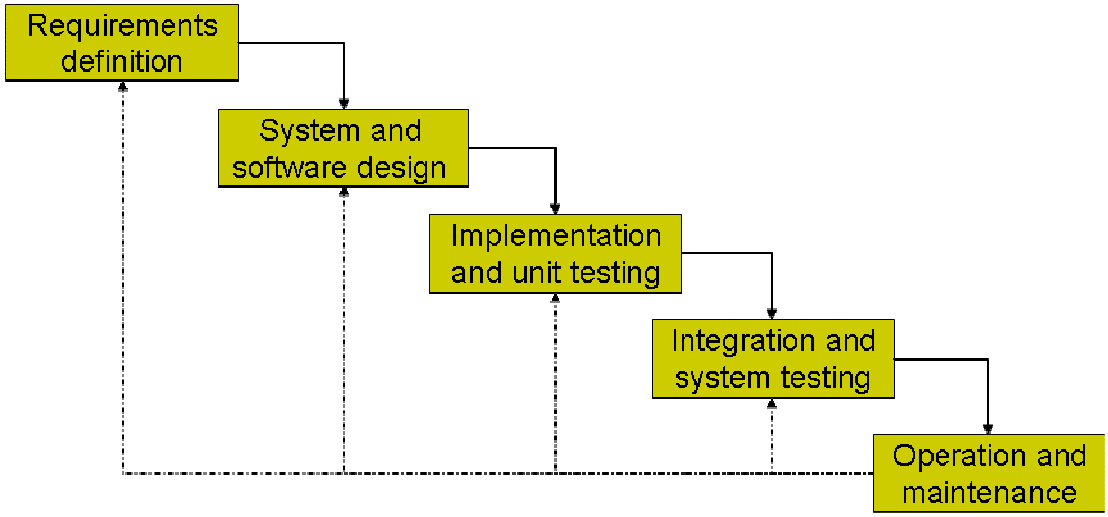

Fig. 1. The software life cycle

Some examples of traditional methodologies are:

- **Waterfall:** this methodology promotes the development of well defined projects and requirements. One of the its characteristics is the clear formalization of goals, documentation and results;
- **Incremental:** which is a result of the integration in the waterfall methodology of critical and larger projects with the need of high stability;
- **Prototyping:** the prototyping methodology is an iterative form of experimentation with the purpose of obtaining information for the development process;
- **Spiral:** is an evolving model, which combines the iterative nature of prototyping with characteristic of the waterfall methodology.

2.2 Current methodologies

The agile methods are adaptive and “embrace” the change. It tried to be a process that adapts and evolve with change, to the point that it have changed itself [1].

The agile methods based on certain principles, such us, customer involvement, incrementally delivery, people not process, embrace change and maintain simplicity [3], will be evidenced in two agile methodologies which are described:

- Extreme Programming (XP): is a software development methodology oriented to small teams, that want developed systems quickly and with constant changes in requirements. These teams work in short iterations, producing so incremental and analyzing requirements as they are described by users. The software developed with this method, should be simple [3, 5, 6, 7]. This methodology is based on four fundamental values: communication, feedback, simplicity, courage [8] and twelve procedures/practices [5].

Most of procedures/practices present on Figure 2 are old, tested and proven, often forgotten, including the most processes. XP is based on the synergy of a whole, where each is a strengthening of other.

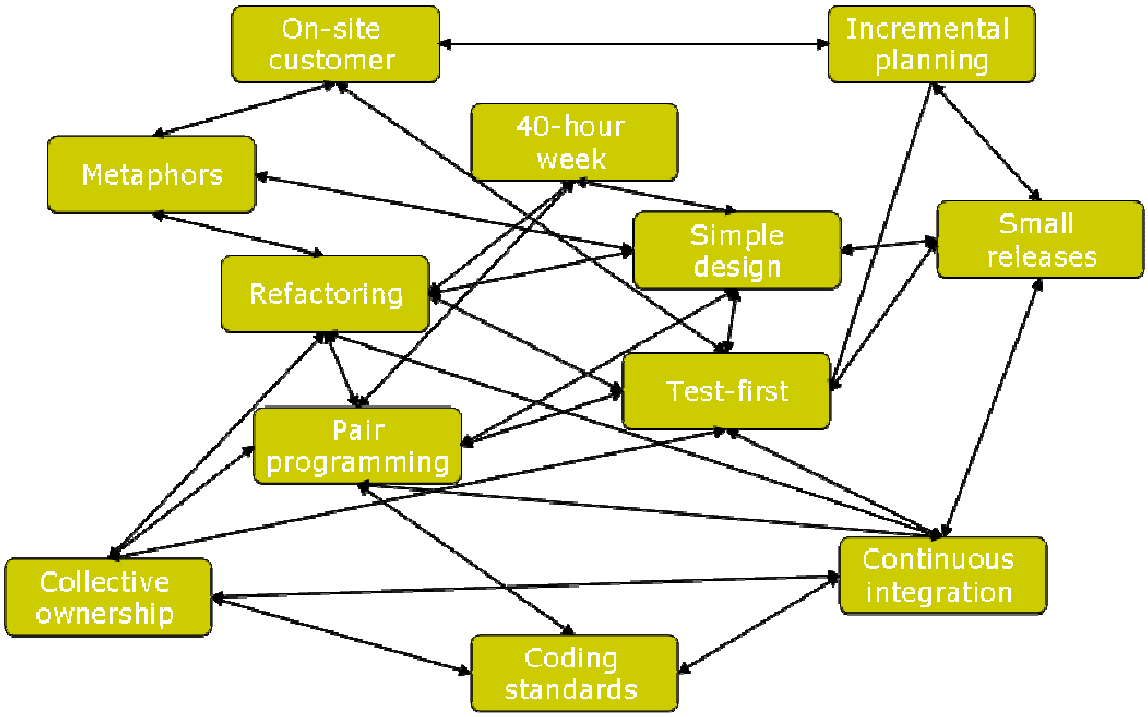

Fig. 2. Extreme Programming (XP) procedures/practices

- Scrum: The Scrum methodology focuses on the fact that defined and repeatable processes only work for problems defined and repeatable, with defined and repeatable people in defined and repeatable environments. The Scrum divides a project into 30 days iterations (called sprints). Before commencing a sprint, the functionality required for the sprint must be defined and then left to the team to develop. The objective is to stabilize the requirements during the sprint. Meanwhile, management is continuous during the same. Every day is held a small meeting, designated as scrum, in which the team defines what will be made on the next day. They inform the management of the impediments that must be resolved and what was done so that management knows in which stage is the project [9].

Probably the best agile method is XP [5]. However, there are other agile approaches, such as, Crystal, Adaptive Software Development, DSDM and Feature Driven Development.

2.3 User Centered Design methodologies

The term UCD, serves to describe processes of a project where end-users have influence on how it is conducted. Some types of UCD seek advice from users about their needs they have in particular education area, involve them at specific shares during development process. Furthermore, there are methods which users have a larger presence, doing part of team, this is, are involved with members throughout all process [10]. Authors like Facer & Williamson, among others, reinforce that UCD is a methodology that combine, among other things, with a user participation and formative prototypes evaluation [11]. According to the standard ISO/DIS 13407 (Human Centered Design Process for Interactive Systems), the UCD projects are governed by four design principles: multidisciplinary team organization, participatory design, interaction between user and system and interactive design.

Abras [10] on Table 1, identifies the advantages and disadvantages of User Centered Design. It is possible to conclude that principal disadvantages are cost and time. Moreover, Shneiderman & Plaisant claim that “UCD methodologies leads to systems that generate fewer problems during development and have lower maintenance costs over their lifetime”[2, p.118].

Table 1. Advantages and Disadvantages of User Centered Design

Advantages	Disadvantages
Products are more efficient, effective and safe.	It is more costly.
Assists in managing users' expectations and levels of satisfaction with the resource.	It takes more time.
Users develop a sense of ownership for the product.	May require the involvement of additional design team members (i. e. ethnographers, usability experts) and wide range of stakeholders.
Products require less redesign and integrate into the environment more quickly.	May be difficult to translate some types of data into design.
The collaborative process generated more creative design solutions to problems.	The product may be too specific for more general use, thus not readily transferable to other clients, thus more costly.

Currently, there are already enterprises that integrated UCD assumptions into their practices. The case of IBM, and its project called "Ease of Use". They may also offer management strategies to keep projects on track and to facilitate effective collaboration among teams that include both business and technical participants [3].

- The Logical User-Centred Interactive Design methodology (LUCID), consists of six steps [12], which are described:

1. Envision – align the agendas of all stakeholders, support users' need for highly usable product;
2. Discovery – study users to determinate high-level requirements;
3. Design Foundation – develop a conceptual design and create a key screen prototype to convey the visual style. Usability test the design, revise, and repeat;
4. Design Detail – flesh out high-level design into complete specifications;
5. Build – support the production process through review and late-stage change management;
6. Release – conduct usability and measure user satisfaction.

The first step, envision, unlike many software-development efforts that are launched without a clear concept of the resource or without agreement among stakeholders, LUCID creates a "high concept" for the resource – that is, a brief statement that defines goals, functionality and benefits. Like most UCD methodologies, LUCID employs rapid prototyping and iterative usability testing.

3 COURSEWARE SERE

3.1 Presentation

Courseware Ser_e includes several types of software (simulations, experimental work, search,…) with educational activities specified in the exploration guidelines. The activities are targeted at both teachers and students (software users). The software objectives are: to promote the understanding of the impact that the human activity has in the natural resources; to understand that the future of mankind will necessary imply a more responsible attitude towards the currently used energy sources, mainly oil and forest resources. The courseware aims to approach the relationship between human activity and exploitation of natural resources and the environmental, social and economic consequences, of this exploitation.

The Courseware was designed to be used in the classroom, with students of 1^{st} and 2^{nd} basic education cycles (preferably from 8 years), particularly the 3º to 6º schooling years, with the guidance of their teachers, although its exploration can be adapted to other levels of schooling, as well other contexts.

As an introduction to the didactic exploration of Courseware Ser_e, it is proposed to view an animation (top left screen of Figure 3) in situations that appear problematic for the Human Being related to the depletion of natural resources (centered on forest biomass and oil). The animation serves as a starting point to a phase of problematization guiding of work research in regard to, for example, the use of

natural energy resources or the exploration of simulations on the impact that the increase in population and the levels and patterns of "consumption" of oil may have access to natural resources.

The resources of Courseware Ser_e, include: an educational software (version on CD-ROM and online, see at: http://sere.ludomedia.pt), the guidelines for the Didactic Exploration – Teacher and the guidelines of Records Sheet - Student/User and User Manual. The software online version allows accessing other resources, such as a library. In the User Manual information related to screens navigation and icons used in the software can be found.

The educational software is divided into two non sequential main phases: Phase 1 - Oil and Phase 2 – Forests. The teacher/student can choose by which of the phases and activities he wants to start the exploration.

Fig. 3. Courseware Sere example screenshots

Regarding activities and for example, in some screens the user is led to reflect on where and how they used natural resources (oil and forest), throughout a research process and making records in tables or graphs. The screen of Figure 3 containing a world map is an example of how are recorded the oil production or levels of consumption that exist in various parts of the world.

The guidelines (Figure 4) were developed to support the operation of the software. In the guidelines for the Didactic Exploration – Teacher different activities are proposed, structured as follows: 1) Activity Purpose; 2) Exploration Context; 3) Exploration Methodology. The guidelines for students are composed mainly of record sheets.

Fig. 4. Guidelines for didactic exploration

Like previously described, Courseware Ser_e is available in two different format: CD-ROM and online. In the online version, the teacher and students may have access to many functionalities that support different activities exploration, such as (Figure 5):

- Information and resources access that related with Courseware subject, available in it's library;
- Socialization between different users of didactic resources, through Web 2.0 tools (forums, chats, wikis, glossary,…), as well a collaborative/cooperative work around activities, logbooks creation, among others.

It also aims to promote skills development of students/users, not only at a thematic level, likewise, ICT use and, fight users isolation through sharing of ideas, stories, problems, experiences, among others mutual interest aspects, and joint construction of solutions.

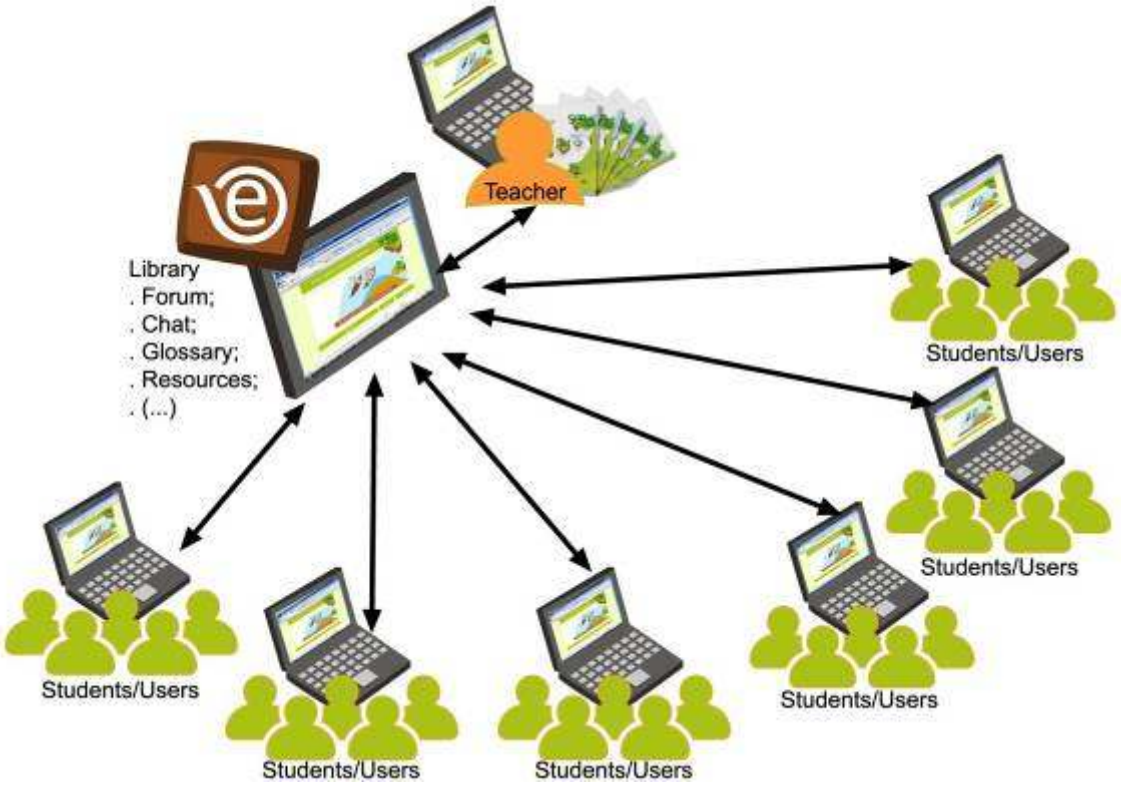

Fig. 5. Courseware Ser_e online exploration

3.2 Development methodology used

The so called HUCDM, used to develop the above described resource, is based both in principles of the UCD and agile methodologies. For instance, in what UCD is concerned, the end users and external experts where involved in different development stages, to evaluate the prototypes, and not only in the end. On the other hand, since the each educational activity is related with different screens, small releases could be tested and evaluated while others were being programmed. As presented in the next paragraphs, the planification was done in an incremental way.

A first stage was the constitution of a multidisciplinary team with elements with various skills such as Didactic of Science (DS), Educational Technology (ET), Project Management, Design, Programming and Usability, for the Courseware Sere -The Human Being and the Natural Resources development. The used courseware development methodology sought to answer research questions related with the implementation of user centered software development methodologies.

To reduce the development time and cost, two of the main UCD disadvantages [10], the team chose to involve the end users (teachers and students) only in the resource evaluation task. The resource (particular the storyboard) was also subjected to evaluation by external experts [6] which is considered essential, regardless of the adopted methodology. Figure 6 synthesizes the development process of the Courseware Sere.

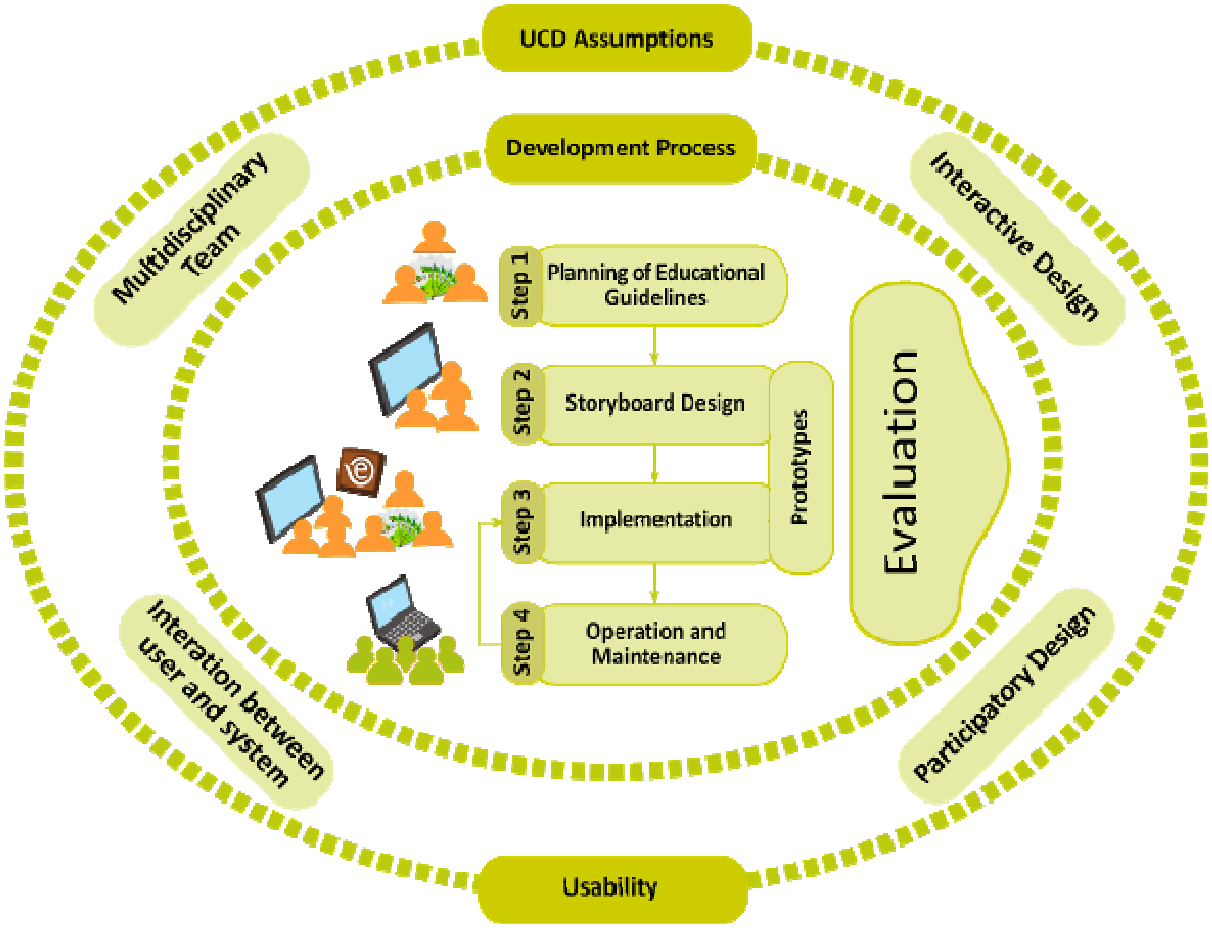

Fig. 6. Development process of Courseware

The development process of the Courseware Sere includes four main steps and two transversal stages:

- Step 1, Planning of Educational Guidelines: definition of the aims, thematic and target public by researchers in DC and TE. Aspects related with the architecture, navigation and the screen's design of the resource, were also discussed at this step, as well as the trademark and patent registration, and agreements associated on copyright, since two institutions are involved in the process (University of Aveiro and Ludomedia).

- Step 2, Storyboard Design: development of the preliminary ideas concerning the educational activities and the content, defined in the previous step, as well as related with the interaction, particularly the navigation and interface. This step involved researchers in DS and ET in collaboration with a designer and a programmer. As Bassani, Passerino, Pasqualotti and Ritze [13] or Carvalho [14] refer, it is considered that the design of scenarios resulting from this step are essential to understand the context of use of the resource and to represent some of the interactive situations to be implemented in the software.

- Step 3, Implementation: this step was divided into two sub-steps implemented simultaneously. The technical part was composed by the software design and programming and the user guide development. The educational part requested the specification in detail of aspects beyond those already specified in the storyboard, as the initial animation and the guidelines for the teachers and the students. During this task, the multidisciplinary team tested and adjusted the content of the guidelines to the desired exploration of the software. This step involved the permanent collaboration of all team members.

- Prototypes: The prototypes include early paper prototypes, key screens and running prototypes and were developed in close collaboration between all the members of the team. Among others, the team identified interface issues that have implications for the technical architecture, which, in some cases, led to changes of the educational guidelines of the resource. Software prototyping was also used, in

the development process, in order to explore particular software solutions and to support user interface design.

- Evaluation: aiming to assess both the resource and its development process. This phase crosses all the above mentioned steps. At the end of step 2, the storyboard evaluation was performed by external elements to the multidisciplinary team, namely: end users, teachers of the 1st and 2nd basic education cycles, and researchers at ET and DS. In addition to numerous internal evaluation and testing sessions of the resource, running prototypes (the first version) were evaluated: 1) by teachers and experts in software engineering, during V Iberian Seminar - I Ibero-American Seminar, Science - Technology - Science Education in Society; 2) by future teachers (3rd year of the 1st cycle of a Higher Education degree in primary teachers training). 3) by teachers of the 1st and 2nd basic education cycles [15].

The evaluation focuses on technical and didactic aspects of the first version of the courseware and is being organised in several workshops. The workshops are practical sessions, with a maximum duration of 120 minutes, in which the evaluators, in small groups, of two to three elements, explore one of the courseware phases and respond to a questionnaire. The first part of the evaluation questionnaire is divided in two groups with closed questions about the educational potential of Courseware Sere: (a) a list of issues related to user interaction with the software; (b) pedagogical and content aspects associated with the designed activities. In the second part, open questions, ask for a synthesis on the relevance and potential of educational resource, the Courseware Sere. Finally, the third part, seeks comments about the working session and the evaluation tool. Moreover, the comments made by the groups, during the exploration of the courseware, can be registered in online forums.

Currently, the team is performing a new evaluation step, involving mainly children of the 1st and 2nd basic education cycles. This evaluation will take place in the classroom context and a new questionnaire will be used to, provide a clear and objective measure of the user's view about the suitability of the courseware [16]. The next evaluation step concerns the development process.

- Step 4, Operation and Maintenance: This step involves correcting errors, technical and educational, which were not discovered in earlier stages of the life cycle of the courseware. It permit the improvement of the system units and enhance the system's services as new requirements are discovered [3, 17]. There are three types of maintenance:

- Corrective – concerns problems such as:
    - graphical elements which were not very perceptive;
    - interactivity of some screens should be increased;
    - contents presented may be too complex for students of the first cycle;
    - vocabulary used may be difficult to understand for some students.
- Perfective - for example propositions of additional functions, like:
    - to give students/users the possibility to fill the activities worksheets online (online version of the student guide);
    - record students/users log. That function will permit to trace the students/users progression.
- Preventive - for example:
    - technology changes used to facilitate the implementation of new functionalities, as well as to improve the system adaptability.

4 CONCLUSION AND FUTURE WORK

Traditional software development methodologies work for non-interactive or software packages with scarce interaction. However, it become insufficient when the implementation involves graphical interfaces, with multiples possibilities of interaction, since it is difficult, or impossible, to anticipate a precise definition of all the requirements and the specifications of this type of resources [18].

Software development is a learning iterative process, dripped and organized as the process is conducted [17]. Since software is produced for a variety of users and given the importance of the

contexts, new development methodologies are required. Although, the selection of the methodology depends on the environment where the project is inserted and on a set of variables that, sometimes, are not defined in advance, the methodologies assist in the software development, minimize uncertainty and allow the achievement of the expected final result, as efficiently as possible. Software development processes may be improved in many enterprises [3], but the successful developers also know that careful attention to UCD issues, at the early steps of software development, dramatically reduces both development time and cost [2].

Bearing in mind that there is no "ideal" software development methodology. The development methodology of the Courseware Ser_e, HUCDM, incorporate principles of the UCD as well as principles aligned with agile methodologies. We believe that the evaluation of the methodology, in progress, will help to define a software development methodology suitable to be adopted by a small software enterprise and thus to find solutions to the problem identified in the introduction of that contribution.

Planned future work comprises:

- the definition in which steps, of the development process, the end user must intervene (incorporating it or not into the multidisciplinary team);
- the balance of the costs and the benefits of the HUCDM used.

5 REFERENCES

[1] Fowler, M. (2005). The New Methodology. Retrieved 20 April 2009 from http://www.martinfowler.com/articles/newMethodology.html.

[2] Shneiderman, B., & Plaisant, C. (2005). Designing the User Interface - Strategies for Effective Human-Computer Interaction (Fourth Edition ed.): Addison Wesley.

[3] Sommerville. (2007). Software Engineering (Eighth Edition ed.): Addison Wesley.

[4] Bona, C. (2002). Avaliação de Processos de Software: um estudo de caso em XP e ICONIX.

[5] Beck, K. Extreme Programming Explained: Embrace change. Reading, Massachusetts: Ed. Addison-Wesley, 2000.

[6] Bergin, J., Caristi, J., Dubinsky, Y., Hazzan, O., & Williams, L. (2004). Teaching Software Development Methods:The Case of Extreme Programming. Paper presented at the SIGCSE '04.

[7] Vasconcelos, C. R., Garcia, F. P., & Turnell, M. d. F. Q. V. (s.d.). Integrando Usabilidade e a Engenharia de Software: um modelo para o desenvolvimento de sistemas centrados no usuário.

[8] Keith, E. R. (2002). Agile Software Development Processes - A Different Approach to Software Design.

[9] Paelke, V., & Nebe, K. (2008). Integrating agile methods for mixed reality design space exploration. Paper presented at the Proceedings of the 7th ACM conference on Designing interactive systems.

[10] Abras, C., Maloney-Krichmar D., & Preece, J. (2004) User-Centered Design. In S. Publications (Ed.), Encyclopedia of Human-Computer Interaction. Thousand Oaks: Sage Publications.

[11] Facer, K., & Williamson, B., (2004). Designing educational technologies with users. From http://www.futurelab.org.uk/download/pdfs/research/handbooks/handbook_02.pdf

[12] Kreitzberg, C. (1996). Managing for usability, in Aber, Antone F. (Editor), Multimedia: A Management Perspective, Wadsworth, Belmont, CA, 65-88.

[13] Bassani, P. S., Passerino, L. M., Pasqualotti, P. R., & Ritzel, M. I. (2006) Em busca de uma proposta metodológica para o desenvolvimento de software educativo colaborativo. Novas Tecnologias na Educação, 4(1), 1-10.

[14] Carvalho, C. (2003) Conceitos básicos para o desenvolvimento de cursos multimédia - Manual do Formador (1ª ed.) Sociedade Portuguesa de Inovação, Porto.

[15] Costa, A. P., Loureiro, M. J., Reis, L. P., Sá, P., Guerra C., & Vieira, R.M. (2009). *Courseware Sere: Technical and Didactic Evaluation*. V International Conference on Multimedia and ICT in Education (m-ICTE2009), Lisbon 22 to 24 April.

[16] Van Veenendaal, P. W. M (1998). Erik: Questionnaire based usability testing, in Proceedings of European Software Quality Week, Brussels, November.

[17] Miguel, A. (2003). *Gestão de Projectos de Software*: FCA - Editora de Informática.

[18] Gomes, M. C. A. (2000). Avaliação e Ciclo de Vidas das Aplicações Educativas: uma proposta com base na análise no desempenho do aluno. Universidade de Coimbra, Coimbra.