Universal Access in the Information Society
https://doi.org/10.1007/s10209-022-00893-9

LONG PAPER

Responsive inclusive design (RiD): a new model for inclusive software development

Carlos Moreno Martínez[1] · Joaquín Recas Piorno[2] · Juan José Escribano Otero[3] · María Guijarro Mata-García[2]

Accepted: 2 June 2022

Abstract
The design and development of technological solutions based on software for all types of people, including people with disabilities, is still a pending issue in most software application development projects today. Situations like the 2020 pandemic drastically reflect how people with disabilities tend to be left outside the application design and construction guidelines. There are multiple initiatives and previous works that advocate user involvement from the beginning of the project; however, in this work, we go a step further by presenting a model for designing and constructing software applications (RiD—Responsive inclusive Design) defined for inclusive software. RiD extends the involvement of the user with disabilities to the entire software life cycle, in different roles, and taking into account the changing nature of the user profile in the evolution of the product. This article also presents the EDICO case study, an accessible and inclusive scientific editor for the Spanish National Organization of the Blind (ONCE), which was successfully implemented applying the RiD principles.

Keywords Inclusive design · Case study · Software engineering · RiD

1 Introduction

The 2020 pandemic and lockdown has revealed difficulties in understanding what we can and cannot do in each phase of the de-escalation. It is difficult to know how to behave in the *"new normal"*, especially for people with disabilities. How will an autistic person, a person with Down syndrome, or a person with cognitive delay with low reading abilities and with concentration problems, know how to behave when something changes again in this new reality? Or a blind person? In Spain, the ONCE Foundation [12] has created a series of videos on its YouTube channel that aim to raise awareness, regarding the serious problem posed by the measures implemented without thinking about the diverse functional diversity of people (example:#BarresimoCOVID). Other organizations have published lists of recommendations to make spaces more accessible and maintain restrictions due to the pandemic [44].

It is necessary to develop technological tools to explain these concepts in an understandable way, adding support elements to meet the new standards. There are many ways to develop these tools, but not all are correct. One can, for example, design and develop a tool for this purpose and then, once completed, apply a new development cycle to make it accessible, for instance, to visually impaired people. This is a wrong way of posing the solution to the problem. By doing so, the adaptation of the final application is affected by previous design decisions and both, the developers and the companies that will finance the project, will be forced to carry out two developments: first they will implement the application following a standard methodology, and then they will adapt it to make it *accessible*. This adaptation implies an increase in development costs, which means, in many cases, that it is not implemented to save costs.

✉ Carlos Moreno Martínez
carlos.moreno@universidadeuropea.es

Joaquín Recas Piorno
recas@ucm.es

Juan José Escribano Otero
juanjose.escribano@ufv.es

María Guijarro Mata-García
mguijarro@ucm.es

1 Science, Computing and Technology Department, Universidad Europea, Madrid, Spain

2 Computer Architecture and Automatic Control Department, Universidad Complutense, Madrid, Spain

3 Higher Polytechnic School, Universidad Francisco de Vitoria, Madrid, Spain

Another way is to develop the application with accessibility as a design principle embedded in the development process. Application design stages with an active participation of people with disabilities, such as blind people, and people who tries to support inclusion in society, is a way to start in the right direction.

The model proposed in this article allows the comprehensive design and subsequent effective use of technology by groups with disabilities. The user experience must necessarily involve the design of specific aspects (inclusive subsystems for hearing and vision disabilities, etc.), but the specific and individual usage patterns of these people and their environment must also be taken into account. Technology must adapt to the intellectual level and functional independence of the individual, and not the other way around [45]. This model is what we call Responsive inclusive Design (RiD), and it consists of principles for the design, development and evolution of inclusive technology, applied to a software development life cycle framework. The model inherits and extends the design philosophy of RD (Responsive Design), transferring the adaptation of the system to the user, the person, and not the device. That is the reason for the R, that the system reacts (adapts) to the user and its functionalities in the same way that an RD application adapts (and reacts) to the graphical functionalities of the device.

This article is structured in the following sections: Sect. 2 lists some of the relevant previous work with user-oriented design and concepts such as usability, as well as a summary of how development models and methods have been designed, and adapted to incorporate accessibility and inclusion as key facets in the development of inclusive systems. Section 3 presents the details of the RiD model. Section 4 describes a case study in which this model has been applied in the implementation of an accessible and inclusive scientific editor, EDICO, in the field of education for visual impaired people. Section 5 provides the most relevant outcomes from the case study. As a final point, conclusions and the future applications of this research are presented in 6.

2 Related work

The need to propose software design and development methods that improve the success in the adoption of systems by the user has been analyzed by various studies. These researches seek insight into the factors that affect the relationship between user and technology.

In [9], an acceptance model is defined in which the approach consists of considering the perspective of the user's beliefs, intentions and attitude as determining factors in the use of information systems. This model has been widely studied and criticized (e.g. in [20, 21]). In further studies, a change in orientation is observed, putting the user at the center of the analysis when assessing the usefulness of the systems, through the concept of usability.

The work of Brooke [4], who proposed the System Usability Scale (SUS) to measure usability, starts from the basis of taking into account the context of the user in the measurement of the system, understanding the context as the physical, organizational and social features of the user to whom the technological solution is addressed. This concept has subsequently been expanded by other works, where we find proposals to evaluate the degree of technological affinity through surveys as a tool to measure the user's level of maturity in the use of the systems [10, 39].

Usability inspection methods have been used over the years to validate software design [15], some of these methods proposed by [30] include the Pluralistic walkthroughs as a way of collaborative work carried out by user representatives, domain experts and developers [3].

Functional dependency is defined by [1] as *"Functional independence is the ability to carry out activities of daily living safely and autonomously"*. In [8], aspects such as the individual's demographics, cognitive abilities and previous experience in the use of technology are evaluated in relationship to technology adoption as a factor for functional independence. The findings could be applied in the proper design of training programs for technology users, specially for older people.

In recent years, the concept of usability has been expanded to encompass accessibility in technology when the user has some type of disability (e.g. in [22]). Today, there are standards and a collection of best practices for software development, which recommend user involvement from very early stages of design (e.g. [17]). The research presented in [43] proposes the analysis of the context of use of the application, as well as the implementation of iterative development methods with assessment of user needs through prototypes in order to improve the design and development process. In turn, de facto industry standards for the development of accessible digital content have emerged, [7, 46], as well as accessible design guidelines from widely used software manufacturers (e.g., [25, 37]). For website development, the concept of Responsive Design (RD) [2, 13] focuses on the device and the layout options provided by the technology.

However, there are still great challenges in meeting accessibility goals for all and in the extensive use of accessible technology (e.g., [18, 34]). What are the reasons for this gap? According to the aforementioned study by Raja [34], there are several reasons. The lack of regulatory frameworks and the higher economic cost in developing accessible technology is one of them. The absence of awareness, knowledge and capacity to develop these applications is another highlighted reason. Focusing on the latter, we see that there are reasons originating from the user herself (awareness, knowledge) and from people dedicated to the implementation (development capacity).

One way to deal with this situation is to apply user-centered design from early stages in the development life cycle, which allows creating digital technology for people with a lower degree of maturity in the use of information technology [23]. The work of [19] also highlights the need of a user-centric design strategies for people with impairments and presents a design model that considers user capability and the population profile.

In [41], a successful case study is presented, together with a design methodology for a self-service ticketing system that involves users, in this case elderly people, in early stages of the design process. This project applies a design method that includes a user needs assessment based on the concept of self-efficacy (*"in one's capabilities to organize and execute the courses of action required to produce given attainments"*).

In the specific case of assistive technology design applied to learning applications for students with disabilities, specifically with visual impairment, we found several previous works and experiences that apply the concept of user-centered design. Taylor's study [42] reviews previous works on this subject, as well as various proposals and digital learning tools for this group of users, and proposes to apply the principles of Universal Design in Instructions (UDI), such as the use of information perceptible through accessibility devices and tools (e.g., refreshable braille displays, screen readers, and screen magnifiers). The same authors also propose the equitable use of the content to ensure that access instructions are available in formats that allow an equal or at least similar use for students with different abilities and conditions.

Similarly, [38] proposes a development model for multimedia software design for visually impaired students. It is based on the conversion of 2D and 3D models of the real world to an acoustic representation that the student can perceive. In this case, the design process begins with the identification of the students' cognitive skills, for example, problem solving or drill-and-practice exercises on the computer, such as those recommended in [14]. In addition to a model for the design of the application, the author proposes the use of special editors for students and teachers. The design process includes a usability assessment and evaluation metrics about the impact on cognitive abilities. Among the recommendations proposed in this work is the application of haptic and auditory technology as an alternative to traditional human-machine interfaces.

Another example that highlights the importance of an adapted design is found in [40]. The authors describe the work carried out in a wide-ranging project for the design and development of a prototype of an e-Learning system for mathematics students with visual impairments (ACED—Adaptive Content for Evidence-based Diagnosis). They propose a reference framework for the design of adaptive programs, focused on students with visual disabilities for mathematics subjects, where the key to the design lies in the adaptation of the system to the student's needs. The authors cover aspects of systems design, as well as the design of training content, learning instructions, and teaching methods for the user community. They propose adapting the content to the student's needs through what these authors call micro-adaptation (what content is presented) and macro-adaptation (how it is presented).

Focusing on the design process itself from the point of view of software engineering, many authors argue that software engineering should incorporate the emotion experienced by the user when using the system as a fundamental element of requirement engineering. They propose the use of *"emotional objectives"* as a type of requirements in Software design models, and analyze the impact of insufficient specification of requirements from various sources. It is therefore essential to achieve a design adapted to people with disabilities (visual in the case treated in this article), to allow the generation of these positive emotions and avoid emotions that provoke rejection ([6, 28, 29, 35]).

3 Method

Any system that deals with disabilities must take into account the voice of the experts as well as that of the end users, the beneficiaries of that system. Unfortunately, this is not always the case and very often it is only considered after the development, in the testing phase. This approach (develop first, adapt later) leads the solution into an increased development cost motivated by disability. RiD, and other models based on the concept "inclusive by design", propose to transfer that voice to the early stages of development, when it is simple and cheap to decide solutions that later serve their purpose because they are already adapted. RiD also proposes to include that opinion in all stages of development, from design, to system testing, through the intermediate stages. Care methodologies such as PCP (Person-Centered Planning) [32], which appeared in the 80s in the context of intellectual disability, used by many assistance associations, such as "Plena Inclusión" in Spain [24], show the power of involving people with disabilities in everything that is related to the development of solutions for their integration and normalization in society.

The RiD model follows a process approach to software development as proposed by the ISO/IEC/IEEE 12207 standard [16]. Specifically, the processes of categories 6.4 and 7.1 of the aforementioned standard covers the recommended activities from the conception and analysis of the user's need, through the analysis of systems, design, development, validation and testing, to the implementation and post-implementation verification.

RiD is, therefore, conceived as a model that follows the software life cycle approach [33, 36]. RiD particularizes existing models by applying User centric product design, and an iterative process with specific add-ons for disabled users:

(a) the user is an active actor in the design and validation of the system from the beginning of the project;
(b) it is recommended to organize the development project based on iterative models in the design and verification phases, involving users with inclusive needs and/or field experts; and
(c) it promotes the design beyond the product delivery, including into the product design the changing nature of user characteristics over time (e.g. the degree of blindness that evolves over time).

RiD is aimed as a general model, but with specific implementation based on the characteristics of the user groups.

In order to achieve this, RiD defines ten specific principles for the software life cycle and implementation of inclusive software applications. These principles govern the activities of the life cycle, from the conception of the software and the definition of requirements by experts in the field, through iterative design with the user, the joint work between disability experts and experts in the development of inclusive software, up to the controlled and guided deployment by experts and users with disabilities. It also includes a final phase dedicated to continuous improvement that guarantees that the software will remain valid and applicable even when the user suffers a change in the disability degree.

Figure 1 summarizes the fundamental aspects of the RiD model, and its principles are described in the following.

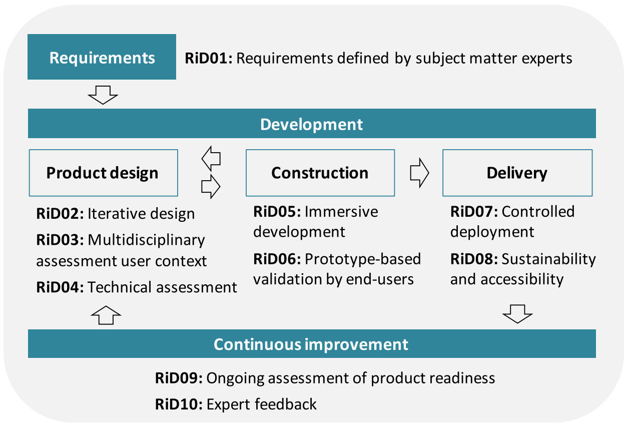

Requirements
RiD01: Requirements defined by subject matter experts
Development
Product design
Construction
Delivery
RiD02: Iterative design
RiD03: Multidisciplinary assessment user context
RiD04: Technical assessment
RiD05: Immersive development
RiD06: Prototype-based validation by end-users
RiD07: Controlled deployment
RiD08: Sustainability and accessibility
Continuous improvement
RiD09: Ongoing assessment of product readiness
RiD10: Expert feedback

Fig. 1 RiD principles in the inclusive software life cycle

RiD01: Requirements defined by business experts, disability experts and inclusive technology experts The design of the software system is based usually on the application domain (e.g., computer solution for teaching and learning mathematics). For RiD applications, it should necessarily be based on specific requirements defined by subject matter experts in user disability conditions, for example, teachers or students with different levels of visual disabilities, and experts in inclusive technologies. This approach allows the alignment of functional and inclusive requirements.

RiD02: Iterative design RiD promotes iterative design in the Software life cycle, with design acceptance criteria conditioned on the active involvement of application domain experts and subject matter experts in disability (and/or user representatives with disabilities). In this sense, any software development with inclusive requirements should consider an iterative approach to refine the end-product until it aligns and meets criteria and requirements from all points of views.

RiD03: Multidisciplinary assessment of end-user circumstances The circumstances of the user must be analyzed from a multidisciplinary perspective using collaborative design and validation methods. It is of utmost importance to fully capture the user circumstances for inclusive software. Specifically in the context of RiD this entails:

1. correctly capturing the current and future situation of disability (e.g., users with visual impairment who suffer a degenerative disease that increases their degree of disability over time);
2. capturing the degree of maturity in the use of technologies by the potential user population; and
3. defining the requirements for the maintenance and inclusive evolution of the product. If this user circumstances analysis approach is not carried out, the developed product runs the risk of not serving its purpose in the future.

RiD04: Technical assessment of inclusive requirements (SW tool perspective) Within the iterative design stage, software engineers specialized in inclusive development are part of the team. Software experts perform a technical feasibility validation of the requirements.

RiD05: Immersive development The list of validated requirements moves to the construction phase. Software Engineers from the multidisciplinary team implement the requirements, in close collaboration with domain experts, applying a work format with direct involvement, even on-site with the expert user.

RiD06: Prototype-based validation by end-users The developments are based on prototypes, which are validated by end users or end user representatives. In case of finding any non-conformity, it goes back to the iterative design validation phase.

RiD07: Controlled deployment The final product delivery is carried out by the client's personnel in controlled deployments, which involve a third validation with the participation of users or representatives of selected users, before the definitive deployment open to the entire user base.

RiD08: Sustainability Like *"manufacturing"* processes, product evolution processes are created by design to include people with disabilities in the role of support personnel. All deployment is accompanied by adapted and inclusive documentation, both for end users and for maintenance and product evolution personnel.

RiD09: Ongoing assessment of product readiness for each individual user In parallel to the deployment, the continuous improvement phase begins. RiD establishes guidelines for subsequent maintenance, based on the continuous assessment of the product regarding its suitability for each individual user, through the capture of incidents reported by the end user.

RiD10: Expert feedback on product readiness Part of the continuous evolution phase is the capture of feedback from experts, and relevant users, regarding new functionality and specific functional improvements, and the analysis of user behavior, to provide insight into new opportunities for application adaptation.

4 Results: the EDICO case study

4.1 About EDICO

EDICO arises from the need of the Spanish National Organization of the Blind (ONCE) for the creation of an accessible and inclusive scientific editor. ONCE is a non-profit organization of general interest oriented to the integration into society of blind people, or people with other disabilities, with the aim of improving their personal autonomy and quality of life.

EDICO is an editor designed by ONCE, and developed by the Complutense University of Madrid (UCM), so that students from scientific areas and with visual impairment can work both inside and outside the classroom [5]. Conceived to be used from primary school to university courses, EDICO should allow both blind or visually impaired and sighted users to manipulate mathematical texts, as well as to represent chemical compounds, in the same way as with a notebook. EDICO should also allow the teacher to supervise the students' work and prepare tests for them to take in a controlled and inclusive environment (Fig. 2).

4.2 EDICO project development under the RiD model

Once the general idea of EDICO was specified, the design and implementation of EDICO began with the RiD model in mind from its earliest phase until the deployment of the final prototype.

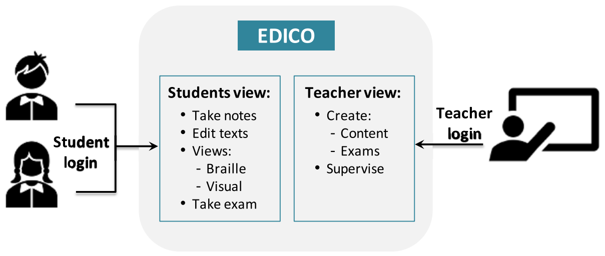


Fig. 2 General EDICO functionality

The first step was to form the three project teams.

1. *ONCE Education Team* is composed of two experts in the teaching and learning processes of visually impaired people. One of the experts was a sighted teacher of mathematics, physics and chemistry at the ONCE Educational Resource Center for blind people, and the other a blind mathematical professor.
2. *UCM Technological Development Team* is composed of four computer engineers, two junior computer engineers in charge of software development and two senior engineers as supervisors.
3. *ONCE's Tiflotechnological Testing and Advisory Team* is team formed by computer engineers and users affiliated with ONCE. They established the programming language, the environment, as well as the accessibility features. Formed by a senior developer, specialized in accessible technologies, and a testing team, specialized in systematic testing of software accessibility for blind people.

These work teams followed the work process shown in Fig. 3. This figure shows the 5 phases required to complete the project and the relation to RiD principles, which will be detailed in the following.

4.2.1 Definition of basic requirement by experts stage (RiD01)

In this first phase, the ONCE Education group generates a general specification document that broadly includes the minimum functionality desired for EDICO, as described below:

1. The first stable version should be available two years after the start of the project;
2. The program must allow the inclusion of blind people in the classroom for the following science subjects:

   (a) Mathematics

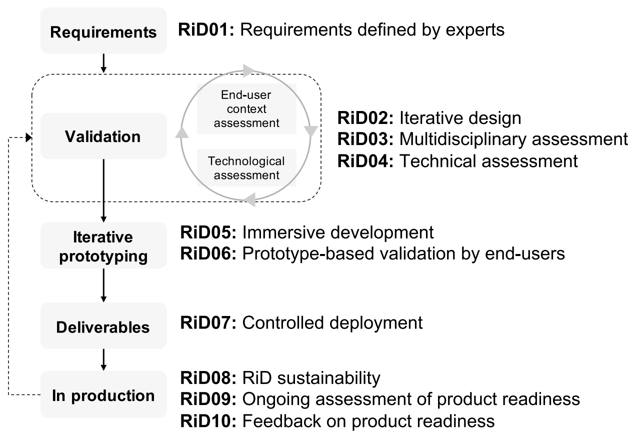

Requirements
RiD01: Requirements defined by experts
Validation
End-user context assessment
Technological assessment
RiD02: Iterative design
RiD03: Multidisciplinary assessment
RiD04: Technical assessment
Iterative prototyping
RiD05: Immersive development
RiD06: Prototype-based validation by end-users
Deliverables
RiD07: Controlled deployment
In production
RiD08: RiD sustainability
RiD09: Ongoing assessment of product readiness
RiD10: Feedback on product readiness

Fig. 3 RiD applied to EDICO

(b) Physics
(c) Chemistry;

3. It must be designed to be used by blind, or visually impaired users, as well as by sighted people;
4. It should simplify the generation, completion and correction of exams for blind people by sighted teachers;
5. It must be able to adapt the interface to users with different degrees of visual impairment and knowledge (students of different educational levels and teachers);
6. It must support screen reading software such as JAWS [11] and NVDA [31];
7. It must support the use of a braille display;
8. It should be available for Windows, Mac and Linux operating systems.

4.2.2 Requirements validation and refinement stage (RiD02, RiD03, RiD04)

Starting from the basic requirements document generated in the previous phase, an iterative requirements validation stage is conducted. During this stage, two weekly meetings were planned, in which the different work teams studied in detail the initial requirements from a technical point of view:

1. Attending the first of the two weekly meetings was attended by the ONCE education team, the UCM supervisors, and typhlotechnology expert from ONCE. The objective of this meeting was to perform a first analysis of the feasibility, from a technological point of view, of every requirement specified by the ONCE education team.
2. In the second meeting, the entire technical team of the UCM, together with the typhlotechnology expert, evaluated in detail the feasibility of the implementation of the requirements and designed the architecture to be developed, both at a technological and functional level, taking into account accessibility restrictions. On occasions, during these meetings, possible modifications of the requirements were also raised in order to create a functional tool in a reasonable time.

It is relevant to note that, during this iterative stage, some aspects considered desirable by Education had to be modified, such as the support for the use of the NVDA tool, due to the integration problems of the revision system. Development also had to be limited to a single platform, Windows, using the Visual Studio IDE [27] and the C# programming language, since, unlike other development environments, offers specific functionality that greatly facilitates inclusive software development due to its integration with JAWS [26].

In the initial phase of EDICO, user profiles were specified in which the requirements of each user were established. This does not generate different versions of the program, but adapts the program to the user who is going to use it. These profiles will adapt the contents to the need, for example, if we import a file in latex and we have set that we have the need for contrast, it will show it and work with the file with the corresponding adaptation. If the user profile is defined for a blind person, all the content of the program will be adapted with text labels that can work with the screen reader, as NVDA or JAWs, and will be displayed in Braille through the Braille Line.

4.2.3 Iterative prototyping stage (RiD05, RiD06)

Once the final specifications had been defined, bearing in mind the feasibility of its implementation within the established time limits, the EDICO editor was implemented in an iterative way.

For each module of the specifications that was being implemented, tests were carried out by two test teams:

1. The sighted teachers of the ONCE Educational Resource Center (CRE) and affiliates belonging to the Braille Mathematics Commission, made up of blind experts in the area of mathematics. These tests were not only limited to verifying that the application met the specifications, but the team also suggested changes and extensions in the functionality, that were added after being evaluated by the education and development teams;
2. Once the various EDICO modules met the specifications, a team of blind testers from ONCE, who were not experts in education, but were experts in the management of accessible technologies, checked their correct operation using a standard test form.

4.2.4 Deliverables (RiD07, RiD08, RiD09, RiD10)

The first functional prototype, corresponding to version v1.1.0.0, was released in September 2018. This deliverable had two favorable reports: one by the education team and another carried out by the ONCE accessibility testing team. Subsequently, this initial version was made available to all ONCE affiliates, users with visual disabilities, who had the need to study subjects in the field of science and regardless of whether or not they were knowledgeable about technologies or the use of typhlotechnological tools. This initial product was released under continuous product evolution and version control.

EDICO also has a specific report platform, where bugs found by users are received and cataloged with the date of the incident and a description by the affected user. Using this information, the development team can make a report where the following fields are assigned to the incident: unique identifier, priority and a summary where the affected versions and classes are specified, along with the proposed solution, the status of the incident and finally, the version in which the bug is corrected.

The next deliverable, v2.0.0.0 was released two years later. This new version was tested by end users, the visually impaired, as well as by sighted people who need these support tools for their teaching.

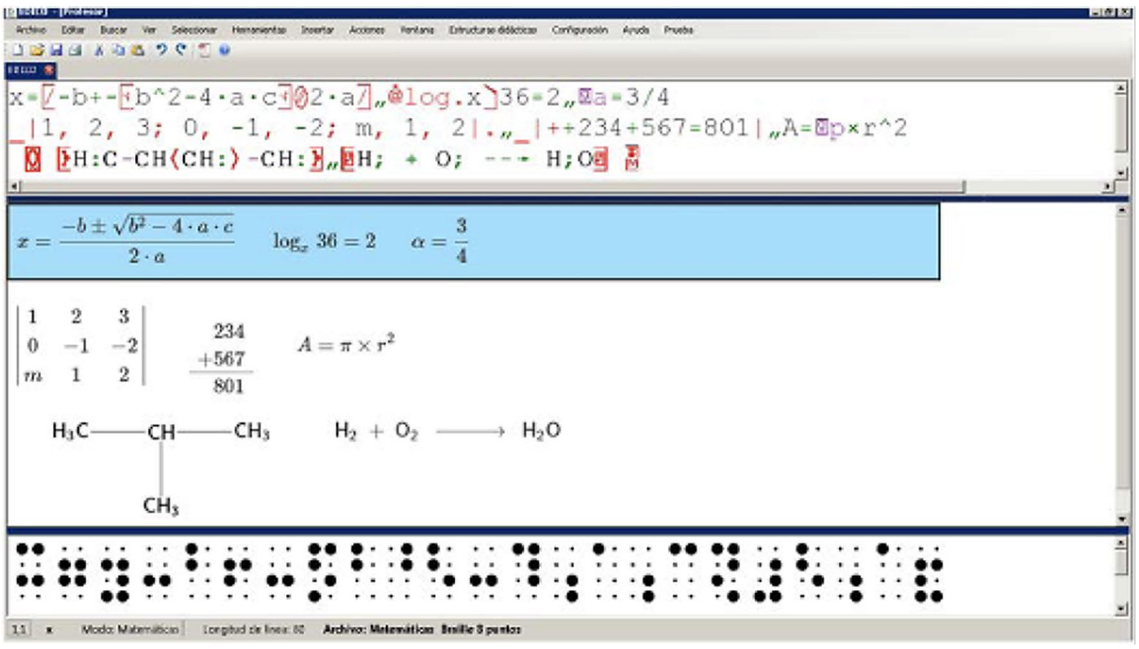
4.2.5 Final result: EDICO program

In its current production version, EDICO is a program that works in Windows environments that integrates with the Jaws screen reader and with the braille display [11], whose main objective is to normalize the blind or visually impaired student inside the classroom. The editor displays the scientific texts on screen using three panels, as can be seen in Fig. 4:

1. Linear editor window (Fig. 4 top frame): displays the linearized code for visual expressions. The screen reviewer verbalizes the content using the corresponding mathematical expressions (integral, division, single bond, ion, etc.).
2. Graphic window (Fig. 4 central frame): expressions in visual format, as they would be written in a physical notebook.
3. Braille window (Fig. 4 bottom frame): Braille display of the line where the linear editor window focus is.

EDICO allows people with visual disabilities to interact with the rest of the students and teachers, since the views offered allow them to share and work in real time on the same document:

1. The blind user can write using both the qwerty keyboard and the braille display, receiving the information contained in an auditory way, through the screen reviewer, and by touch, through the braille line;



```
x=/-b+-b^2-4·a·c 2·a/,,log.x 36=2,,a=3/4
_|1, 2, 3; 0, -1, -2; m, 1, 2|.,,_|++234+567=801|,,A=p×r^2
 H:C-CH(CH:)-CH:,,H; + O; ---> H;O
```

$$x=\frac{-b\pm\sqrt{b^2-4\cdot a\cdot c}}{2\cdot a}\qquad \log_x 36=2\qquad \alpha=\frac{3}{4}$$

$$\begin{vmatrix}1&2&3\\0&-1&-2\\m&1&2\end{vmatrix}\qquad \begin{array}{r}234\\+567\\\hline 801\end{array}\qquad A=\pi\times r^2$$

$H_3C-CH(CH_3)-CH_3$ $\quad H_2+O_2\longrightarrow H_2O$

Fig. 4 Scientific editor EDICO

2. The sighted user will use the qwerty keyboard and will see the result on the screen through the graphic window.

5 Discussion and limitations

The RiD model for inclusive software design and development defines principles for the software life cycle that puts the user, their disability profile and their future evolution in the use of technology, at the center of the design, and advocates for the active involvement of user, and user representatives, throughout the development process.

As a result of the application of the RiD model, the EDICO application has been designed and developed through a collaboration between the ONCE (National Organization of the Spanish Blind) and the UCM (Complutense University of Madrid). This new application allows the inclusion of people with visual disabilities into classrooms of middle and higher education. For this, it has adaptations based on the use of refreshable braille display or braille terminal and Speech synthesizers. This application is currently being used in classrooms of different educational levels, as well as in the Spanish-speaking countries of Central and South America, with more than 7000 downloads. In turn, this application has been installed, as a proof of concept, in computer workstations of the 26 faculties that make up the Complutense University of Madrid, helping to provide an accessible workstation in all areas of knowledge and within the reach of all students. EDICO also takes into consideration other roles of visual impaired people, such as maintenance and test users during the evolution of the software, following RiD principles.

RiD is a general model, and visual impairment has been taken into account in the application in EDICO. RiD has not yet been applied to a project where there are multiple user profiles (multiple disabilities), for example, deaf-blind people, or blind people with a motor disability.

The experience in the implementation in EDICO tells us that a limitation related to the possible imports of documentation to the developed software must be addressed. Any document that a user wants to import from the tool must be accessible, RiD does not impose limitations on imports to accessible software.

Through the development of EDICO it has been contemplated that the particularities of each accessible module must be taken into account in the RiD software model itself, such as in the case of visual impairment and its adaptation, the different languages with which to communicate in Braille. EDICO is designed to work exclusively with Braille in Spanish, therefore it is not compatible with other languages.

6 Conclusion and future work

RiD is a suitable model to develop inclusive software, for users with a specific disability, based on the integral involvement of users and experts in the field throughout the entire life cycle of the product.

The described case study shows the successful application of the RiD model to an inclusive scientific editor software, applied to the field of education of people with blindness. Currently, ONCE is asking to extend the functionality to incorporate other locations. Likewise, the client organization itself is involved in the active maintenance of the platform based on direct user feedback. Moreover, entities from other countries have shown interest in using this system (Italy, South America).

Bearing in mind that the RiD model presented in this paper is in its first release, EDICO being the first practical example of the model's application, it is our initial goal to make developers and users aware of the RiD model for designing and constructing inclusive software. Possible next steps of the model should include dynamic adaptability through artificial intelligence. The implementation based on the current version of RiD handles adaptation by predefined user profiles, as a form of automatic adaptation based on user input (profile configuration). Inclusive software may benefit significantly from artificial intelligence for the analysis of user behavior and usage patterns, so that it would be able, without the need for explicit user feedback, to provide autonomous adaptation to respond to changes in user conditions.

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature. No funding was received to assist with the preparation of this manuscript.

Availability of data and materials Not applicable.

Code availability Not applicable.

Declarations

Conflict of interest Joaquín Recas Piorno and María Guijarro Mata-García have received research support from Company ONCE. Carlos Moreno Martínez and Juan José Escribano Otero declare they have no financial interests.. The authors have no conflicts of interest to declare that are relevant to the content of this article.

References

1. Australian Institute of Health and Welfare: Person-level of functional independence, Functional Independence Measure score code N. https://meteor.aihw.gov.au/content/index.phtml/itemId/449150 (2021)
2. Bader, W.I., Hammouri, A.I.: Responsive web design techniques. Int. J. Comput. Appl. **150**(2), 18–27 (2016). https://doi.org/10.5120/ijca2016911463
3. Bias, R.G.: The Pluralistic Usability Walkthrough: Coordinated Empathies, pp. 63–76. Wiley, New York (1994)
4. Brooke, J.: SUS: a 'quick and dirty' usability scale. In: Jordan, P.W., Thomas, B., McClelland, I.L., Weerdmeester, B. (eds.) Usability Evaluation In Industry, vol. 189, pp. 207–212. CRC Press, London (1995). https://doi.org/10.1201/9781498710411-35
5. Carenas, J.M., Cabra, A.B., Mata, M.G., Gea, P.C., Hernández, D.H.: Prácticas Edico Qué es Edico? Integración Revista Digital Sobre Discapacidad Visual **72**, 100–108 (2018)
6. Colomo-Palacios, R., Hernández-López, A., García-Crespo, Á., Soto-Acosta, P.: A study of emotions in requirements engineering. Commun. Compute. Inf. Sci. **112**(2), 1–7 (2010). https://doi.org/10.1007/978-3-642-16324-1_1
7. Cueva-Lovelle, J.M., García-Fernández, G., Aguilar, L.J., Núñez-Valdez, E., Sanjuan-Martinez, O.: Security guidelines for the development of accessible web applications through the implementation of intelligent systems. Int. J. Interact. Multimed. Artif. Intell. **1**(2), 79–86 (2009)
8. Czaja, S.J., Charness, N., Fisk, A.D., Hertzog, C., Nair, S.N., Rogers, W.A., Sharit, J.: Factors predicting the use of technology: findings from the Center for Research and Education on Aging and Technology Enhancement (CREATE). Psychol. Aging **21**(2), 333–352 (2006). https://doi.org/10.1037/0882-7974.21.2.333
9. Davis, F.D., Bagozzi, R.P., Warshaw, P.R.: User acceptance of computer technology: a comparison of two theoretical models. Manag. Sci. **35**(8), 982–1003 (1989). https://doi.org/10.1287/mnsc.35.8.982
10. Franke, T., Attig, C., Wessel, D.: Assessing affinity for technology interaction—the affinity for technology assessing affinity for technology interaction (ATI) (2017). https://doi.org/10.13140/RG.2.2.28679.50081
11. Freedom Scientific: Blindness solutions. https://www.freedomscientific.com/products/blindness/ (2021)
12. Fundación ONCE: Nuestros ON Fologüers sufren el #Barresimo-COVID - YouTube. https://www.youtube.com/watch?v=V9Hs0DYm1ig (2020)
13. Gardner, B.S.: Responsive Web Design: Enriching the User Experience. Noblis, Falls Church (2011)
14. Hasselbring, T.S., Goin, L.I., Bransford, J.D.: Developing math automaticity in learning handicapped children—the role of computerized drill and practice. Focus Except. Child. **20**(6), 1–8 (1988)
15. Hollingsed, T., Novick, D.G.: Usability inspection methods after 15 years of research and practice. In: Proceedings of the 25th Annual ACM International Conference on Design of Communication, SIGDOC '07, pp. 249-255. Association for Computing Machinery, New York, NY, USA (2007). https://doi.org/10.1145/1297144.1297200
16. ISO: ISO/IEC/IEEE 12207:2017—Systems and software engineering—software life cycle processes. Technical report, IEEE (2017)
17. ISO: ISO - ISO 9241-210:2019—Ergonomics of human-system interaction—part 210: human-centred design for interactive systems. https://www.iso.org/standard/77520.html (2019)
18. Jaeger, P.T.: Disability and the Internet Confronting a Digital Divide. Lynne Rienner Publishers, Boulder (2011)
19. Keates, S., Clarkson, P.J., Harrison, L.A., Robinson, P.: Towards a practical inclusive design approach. In: Proceedings of the Conference on Universal Usability, pp. 45–52. Association for Computing Machinery (ACM) (2000). https://doi.org/10.1145/355460.355471
20. King, W.R., He, J.: A meta-analysis of the technology acceptance model. Inf. Manag. **43**(6), 740–755 (2006). https://doi.org/10.1016/j.im.2006.05.003
21. Legris, P., Ingham, J., Collerette, P.: Why do people use information technology? A critical review of the technology acceptance model. Inf. Manag. **40**(3), 191–204 (2003). https://doi.org/10.1016/S0378-7206(01)00143-4
22. López Ibáñez, M., Romero-Hernández, A., Manero, B., Guijarro, M.: Computer entertainment technologies for the visually impaired: an overview. Int. J. Interact. Multimed. Artif. Intell. (2021). https://doi.org/10.9781/ijimai.2021.04.008. (**in press**)
23. Lucke, U., Castro, T.: The process of inclusive design. In: Proceedings—IEEE 16th International Conference on Advanced Learning Technologies, ICALT 2016, pp. 446–447. Institute of Electrical and Electronics Engineers Inc. (2016). https://doi.org/10.1109/ICALT.2016.132
24. Madrid, P.I.: Plena Inclusión Madrid | Federación Org. Discapacidad Intelectual. https://plenainclusionmadrid.org/ (2021)
25. Microsoft: Inclusive Design. https://www.microsoft.com/design/inclusive/ (2021)
26. Microsoft: Products and services for everyone. https://www.microsoft.com/en-us/accessibility/ (2021)
27. Microsoft: Visual studio. https://visualstudio.microsoft.com/ (2021)
28. Miller, H.G.: The spark of innovation begins with collaboration. Inside Digit. Ecosyst. **11**(1), 13–19 (2011)
29. Miller, T., Pedell, S., Lopez-Lorca, A.A., Mendoza, A., Sterling, L., Keirnan, A.: Emotion-led modelling for people-oriented requirements engineering: the case study of emergency systems. J. Syst. Softw. **105**, 54–71 (2015). https://doi.org/10.1016/j.jss.2015.03.044
30. Nielsen, J.: Usability inspection methods. In: Conference Companion on Human Factors in Computing Systems, CHI '94, pp. 413–414. Association for Computing Machinery, New York, NY, USA (1994). https://doi.org/10.1145/259963.260531
31. NV Access: NVDA: nonvisual desktop access. https://www.nvaccess.org/ (2021)
32. O'brien, C., Lyle; O'brien, J.: The origins of person-centered planning: a community of. For full text: http://www.soeweb.syr.edu/thechp/wnew (2000)
33. Pressman, R., Maxim, B.: Software Engineering: A Practitioner's Approach, 9th edn. McGraw-Hill Education, New York (2020)
34. Raja, D.S.: Bridging the disability divide through digital technologies. World Development Report pp. 1–37 (2016)
35. Ramos, I., Berry, D.M.: Is emotion relevant to requirements engineering? Requir. Eng. **10**(3), 238–242 (2005). https://doi.org/10.1007/s00766-005-0014-5
36. Ruparelia, N.B.: Software development lifecycle models. ACM SIGSOFT Softw. Eng. Notes **35**(3), 8–13 (2010). https://doi.org/10.1145/1764810.1764814

37. Samsung UK: What are the accessibility settings and how do I use them? https://www.samsung.com/uk/support/mobile-devices/what-are-the-accessibility-settings-and-how-do-i-use-them/ (2021)
38. Sánchez, J.: A model to design multimedia software for learners with visual disabilities. In: Kumar, D., Turner, J. (eds.) Education for the 21st Century—Impact of ICT and Digital Resources, vol. 210, pp. 195–204. Springer, Boston (2006). https://doi.org/10.1007/978-0-387-34731-8_21
39. Sauro, J., Lewis, J.R.: Did we meet or exceed our goal? In: Jeff Sauro, J.R.L. (ed.) Quantifying the User Experience, pp. 41–62. Elsevier, Amsterdam (2012). https://doi.org/10.1016/b978-0-12-384968-7.00004-7
40. Shute, V.J., Graf, E.A., Hansen, E.G.: Designing Adaptive, Diagnostic Math Assessments for Sighted and Visually Disabled Students. Technology-Based Education: Bringing Researchers and Practitioners Together, pp. 169–202 (2006)
41. Siebenhandl, K., Schreder, G., Smuc, M., Mayr, E., Nagl, M.: A user-centered design approach to self-service ticket vending machines. IEEE Trans. Prof. Commun. **56**(2), 138–159 (2013). https://doi.org/10.1109/TPC.2013.2257213
42. Taylor, M.A.: Improving accessibility for students with visual disabilities in the technology-rich classroom. PS Political Sci. Politics **49**(01), 122–127 (2016). https://doi.org/10.1017/S1049096515001134
43. Thomas, C., Bevan, N.: Usability context analysis: a practical guide. https://hdl.handle.net/2134/2652 (1996)
44. Todo Disca: Guía de la “Nueva Normalidad” accesible. https://www.tododisca.com/guia-accesible-de-la-nueva-normalidad/ (2020)
45. Vicente, K.J.: The Human Factor: Revolutionizing the Way People Live with Technology, 1st edn. Taylor & Francis Group, Routledge (2004). https://doi.org/10.4324/9780203944479
46. Web Accessibility Initiative: Web Content Accessibility Guidelines. https://www.w3.org/TR/WCAG21/ (2012)