Vázquez-Ingelmo, A., García-Holgado, A., Garcia-Peñalvo, F. J., Pérez-Sénchez, P., Antúnez-Muiños, P., Sánchez-Puente, A., Vicente-Palacios, V., Dorado-Díaz, P. I., & Sánchez, P. L. (2023). Refactoring User Interfaces Through a Data-Driven Framework: a Case Study in the Health Domain. In 2023 XLIX Latin American Computing Conference (CLEI). Proceedings CLEI 2023. (La Paz, Bolivia, 16-21 October 2023). IEEE. https://doi.org/10.1109/CLEI60451.2023.10346102

Refactoring user interfaces through a data-driven framework: a case study in the health domain

Andrea Vázquez-Ingelmo
GRIAL Research Group
Universidad de Salamanca
(https://ror.org/02f40zc51)
Salamanca, Spain
andreavazquez@usal.es

Alicia García-Holgado
GRIAL Research Group
Universidad de Salamanca
(https://ror.org/02f40zc51)
Salamanca, Spain
aliciagh@usal.es

Francisco José García-Peñalvo
GRIAL Research Group
Universidad de Salamanca
(https://ror.org/02f40zc51)
Salamanca, Spain
fgarcia@usal.es

Pablo Pérez-Sánchez
University Hospital of Salamanca, CIBERCV and Biomedical Research Institute of Salamanca (IBSAL)
Salamanca, Spain
pperezsanc@saludcastillayleon.es

Pablo Antúnez-Muiños
University Hospital of Salamanca, CIBERCV and Biomedical Research Institute of Salamanca (IBSAL)
Salamanca, Spain
pantunezm@saludcastillayleon.es

Antonio Sánchez-Puente
University Hospital of Salamanca, CIBERCV and Biomedical Research Institute of Salamanca (IBSAL)
Salamanca, Spain
asanchezpu@saludcastillayleon.es

Víctor Vicente-Palacios
Philips Clinical Science
Western Europe
victor.vicente.palacios@philips.com

Pedro Ignacio Dorado-Díaz
Biomedical Research Institute of Salamanca (IBSAL) and University of Salamanca, Statistics Department
Salamanca, Spain
acho@usal.es

Pedro Luis Sánchez
University Hospital of Salamanca, CIBERCV, Biomedical Research Institute of Salamanca (IBSAL) and University of Salamanca, Cardiology Department
Salamanca, Spain
plsanchez@saludcastillayleon.es

***Abstract*— User interfaces (UIs) play a crucial role in defining user experiences and influencing the success of software products. While UI design has traditionally been subjective and iterative, data-driven approaches are becoming increasingly popular to ensure that Uis meet user needs and expectations. However, contextual factors such as the application domain can present challenges for designing Uis that are both effective and efficient. This is particularly true in the health domain, where Uis must be adapted to specific tasks and user expertise to maximize the support provided by software systems. Moreover, the urgency of delivering fully functional systems in short periods can relegate UI design to a second plane. This paper presents a framework proposal for refactoring and improving Uis using a data-driven approach, providing an efficient and systematic methodology to address not solved UI issues introduced during previous software development processes. The proposed framework has been successfully applied to two medical platforms, demonstrating the importance of data-driven approaches for UI refactoring in domains with particular necessities.**

Index Terms—user interfaces, human-computer interaction, data-driven, medical platform, health platform

This research was partially funded by the Ministry of Science and Innovation through the AvisSA project grant number (PID2020-118345RB-I00). This work was also supported by national (PI14/00695, PIE14/00066, PI17/00145, DTS19/00098, PI19/00658, PI19/00656 Institute of Health Carlos III, Spanish Ministry of Economy and Competitiveness and co-funded by ERDF/ESF, "Investing in your future") and community (GRS 2033/A/19, GRS 2030/A/19, GRS 2031/A/19, GRS 2032/A/19, SACYL, Junta Castilla y León) competitive grants.

I. Introduction

User interfaces (Uis) are crucial means for people to engage with technology. In the current digital age, we continuously interact with technology through different devices and interaction patterns. Uis play a critical role in defining user experiences and influencing a software product's success, from web platforms to mobile applications. As a result, it is vital to ensure that user interfaces are designed to match users' needs and expectations.

Historically, UI design has been a subjective and frequently iterative process, with designers making judgments regarding layout, functionality, and aesthetics based on their intuition and expert knowledge. However, this technique can be time- and resource-consuming, and it does not always ensure the most significant benefits for users.

On the other hand, contextual factors could hamper the design process, such as the application domain. For example, software (and target users) framed within specific and complex domains might ask for particular interaction patterns or UI components that do not necessarily match generic UI guidelines. In these contexts, it is crucial to perform several iterations to capture every domain-specific requirement and validate them through domain experts.

One of these specific fields is the health domain. In this domain, primary users are experts in different medical specialties, such as cardiology, hematology, traumatology, etc. In this sense, Uis should be adapted to the specific tasks and user expertise to get the most out of the support provided by software systems in highly specific domains.

However, in these contexts, the urgency of delivering fully functional systems in short periods can relegate UI design to a second plane. This situation could lead to platforms that meet every functional requirement demanded but with low user experience (UX) and usability, devaluing the final product.

Considering these factors, this work presents a framework proposal for refactoring and improving Uis using a data-driven approach. This framework provides an efficient and systematic methodology to address not solved UI issues introduced during previous software development processes.

The proposed framework has been successfully applied in the health domain with two platforms previously developed and integrated into the Cardiology Department of the University Hospital of Salamanca: CARTIER-IA [1], a platform for managing medical structured and imaging data, and KoopaML [2-4], a platform to support health experts in the generation and application of machine learning (ML) pipelines.

The rest of this paper Is organized as follows. Section II provides an overview of existing UI development frameworks and UI development within the medical domain. Section III details the proposed framework to tackle UI refactoring in highly specific domains. Section IV presents two successful use cases of the proposed framework with two platforms framed in the medical domain. Finally, section V discusses the obtained results from the two use cases, and section VI highlights the main conclusions derived from this work.

II. Related works

The integration of Agile methods and User-Centered Design (UCD) techniques is not something new. According to Hussain et al. [5] "adopting the agile design process centered on the user by their teams resulted in the improvement of quality and usability of the developed product" [6].

Despite there are some clear differences among those methods (i.e., in agile the client is part of the process to provide feedback to the development team, meanwhile in UCD all the process is focused on the user), previous works have analyzed the common trends and recommendations available in the literature to integrate into a single process [5, 7-12]. The review conducted by Silva da Silva et al [8] in 2011 presents a high level framework to integrate processes and artifacts of agile and UCD methods based in the trends found.

Moreover, [13] proposed five principles to integrate Agile development and UCD: separate product discovery and product creation; iterative and incremental design and development, user-centered agile approaches should support software design and development in short iterations and in an incremental manner; design and development should proceed in parallel interwoven tracks; continuous stakeholder involvement; and tangible and up-to-date artifacts should be used to document and communicate product and design concepts.

Jurca et al. [7] identified some UX recommended practices and artifacts among the selected studies (concept maps, cognitive walkthrough variants, workshops, lo-fi prototypes, interviews, scenarios, and meetings with users). On the other hand, it also highlights important problems in Agile-UX methodologies, specifically, the major problem identified was that the UX-designer was over worked and overly distributed among the Agile teams.

The systematic literature review conducted by Caballero et al. [9] identified several studies that report the use of UCD tools in agile software development, such us U-SCRUM which define two product owners to have one only focused on usability [14] or POLVO tool for building low-fi prototypes to improve and increase the agility of interface prototyping in agile developments [15]. The authors identified a lack of interaction design in agile methods. However, they identify "efforts to incorporate the analysis, modeling and integration of user interaction/experiences in the agile design loop".

Salah et al. [11] identified challenges related to Agile User-Centered Design Integration (AUCDI) including lack of time for upfront activities, difficulty of prioritizing UCD activities, difficulty of modularization/chunking, optimizing the work dynamics between developers and UCD practitioners, workload of UCD partitioners and different challenges related to perform usability testing.

Also, Argumanis et al. [16] distinguished three significant challenges integrating Scrum and UCD: (1) insufficient importance assigned to usability and user needs; (2) the insufficient time assigned for upfront activities in Scrum; (3) and the communication problems between designers and developers.

The integration of Agile methods and UCD continue appearing in the literature, however, we have not found any work related to the integration of those methods for refactoring already developed systems with a particular focus on Uis.

III. Framework to refactor user interfaces

The framework merges data collection and user testing approaches with the SCRUM methodology [17]. It is important to remark that this framework is set to be applied to already-developed platforms that need to refactor and improve their Uis and UX and do not currently ask for major, non-UI-related functional features.

First, following the SCRUM methodology, an initial backlog of known UI issues is identified through qualitative interviews with the product owners (domain experts). This initial step collects major issues that have been identified in production. Other techniques can be used for this purpose, such as heuristic evaluation with domain experts and UX experts. The identified issues are classified by screen and feature and associated with a priority score.

Then, a set of sprints need to be planned to address the identified issues. In this process, the data collection processes should be defined. Data collection might vary depending on each sprint's tasks, as some features could need different testing approaches based on their particular implications in the system.

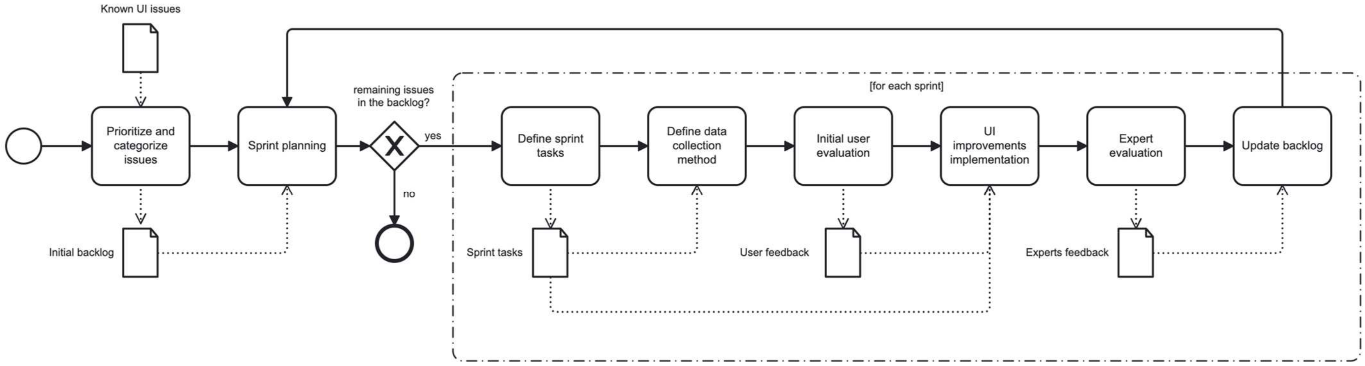

Fig. 1. BPMN diagram of the UI refactoring framework.

Following the selected data collection method, an initial user evaluation is carried out at the beginning of the sprint with a small sample of end users. This initial data collection will provide information to make informed decisions regarding the UI improvements during the sprint. At the end of the sprint, the domain experts perform an additional qualitative evaluation of the introduced improvements.

If the domain experts identify further issues during this step, these will be included in the backlog along with their priority score and classification.

Figure 1 provides an overview of the framework in Business Process Modeling Notation (BPMN).

IV. Case study in the medical domain

This section provides two applications of the proposed framework in the medical domain. The framework has been applied to improve the UIs of two web platforms targeted at the medical audience.

A. *CARTIER-IA*

As introduced, the CARTIER-IA platform [1] is a web application that allows medical institutions to upload DICOM images and associate them with structured data.

The platform's primary goal is to centralize and unify the variety of data sources found in medical institutions to facilitate data analysis. While the platform’s functionality complied with the required features, the UI presented different usability issues.

The main issues were related to the displayed information on the application’s screens, as significant amounts of data, in addition to the DICOM images, needed to be displayed (Figure 2). On the other hand, additional issues were found in the navigation flow, error recovery, and the system’s help and documentation. These UI issues were collected through a heuristic evaluation with experts [18], which allowed the prioritization and categorization of each issue to plan the UI improvements.

Following the proposed framework, sprints were organized targeting the identified issues. Micro-user evaluations were conducted with a small set of domain experts testing the screens subject to improvement in the current sprint. These evaluations provided additional insights to tackle the UI improvement in an informed manner.

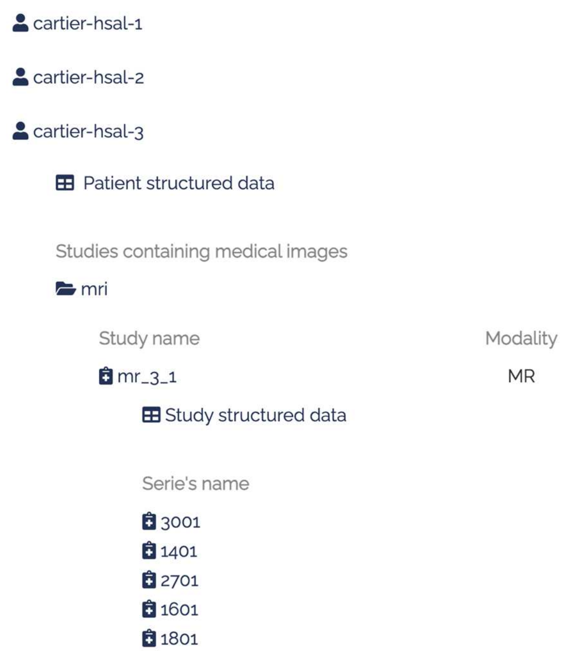

Fig. 2. Information about patients, DICOM studies, and structured data in the CARTIER-IA platform before introducing the improvements.

Figure 3 shows one of the improvements in the UI. In contrast to Figure 2, in which users needed to click through all the tree elements to obtain the target information (which was one of the concerns of the domain experts), the new interface version allowed users to explore the data in a two-dimensional way.

When selecting a patient, their structured data is displayed below, while their available studies are beside them. This approach is repeated for studies and DICOM files. After implementing the improvements, the same domain experts were asked to test and validate the new interface version.

B. *KoopaML*

The KoopaML platform [19] is a system that enables non-expert users to build and execute ML pipelines. KoopaML aims to offer intuitive and educational interfaces to learn how to apply ML models in the medical domain.

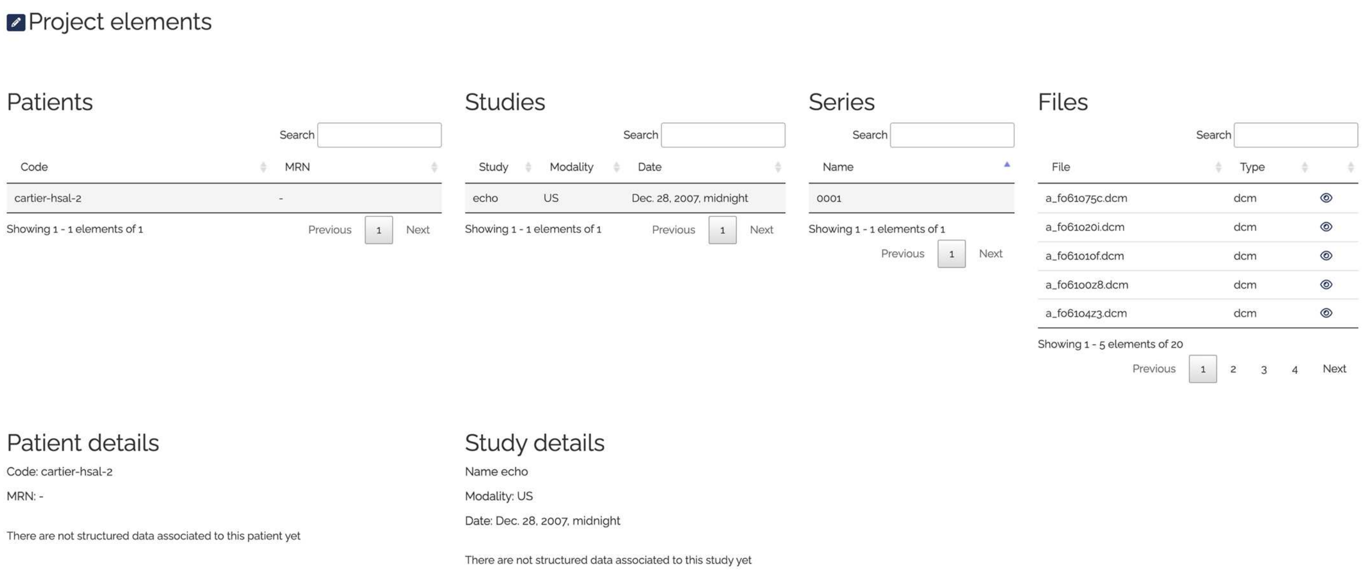

Fig. 3. Information about patients, DICOM studies, and structured data in the CARTIER-IA platform after introducing the improvements.

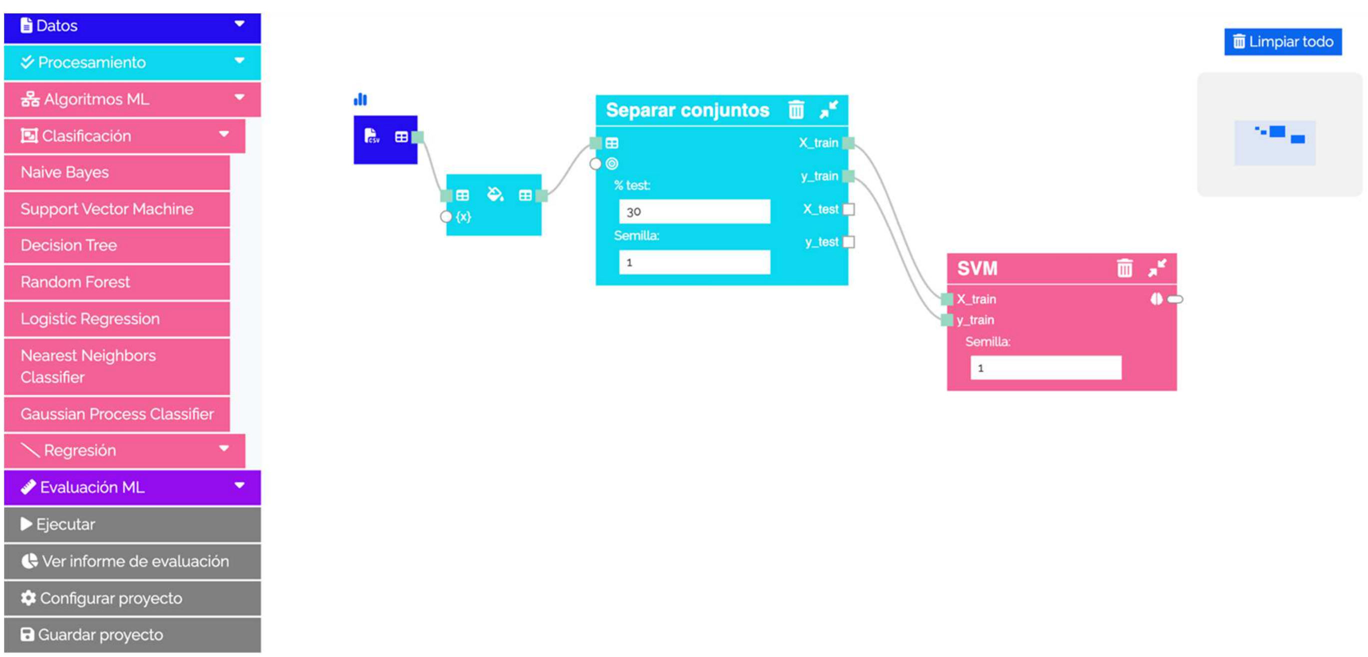

Fig. 4. Workspace of KoopaML before the UI improvement process (contents in Spanish).

However, due to the complexity of the platform's content (terminology, flows, etc.), users could not completely exploit its powerful features.

In this sense, the platform was evaluated by medical staff with the think-aloud protocol [20] to obtain the initial backlog of improvements to be implemented.

Most usability issues were related to the workspace in which the pipelines are designed (Figure 4). Users found several difficulties both in terms of terminology and interaction patterns.

For each sprint, the selected data collection method was a digital data analysis tool that captures the interaction of users within websites (Hotjar.js, https://www.hotjar.com/). Along with the qualitative feedback obtained in the initial evaluation, users' interaction data provided handy insights to address the identified UI issues.

At the time of writing this manuscript, the UI improvements proposed for KoopaML are a work in progress. However, we have already validated the prototype of the new UI with the domain experts, obtaining favorable results. For example, based on the difficulties found, we plan to restructure the whole workspace to restrict the number of options and guide the user throughout the pipeline design process with hints and shortcuts (Figure 5). Also, instead of configuring each node in place, as shown in Figure 4, we plan to reserve a workspace section to configure each node on demand (left-bottom section of Figure 5), leaving more space for the pipeline structure and making the configuration process more coherent.

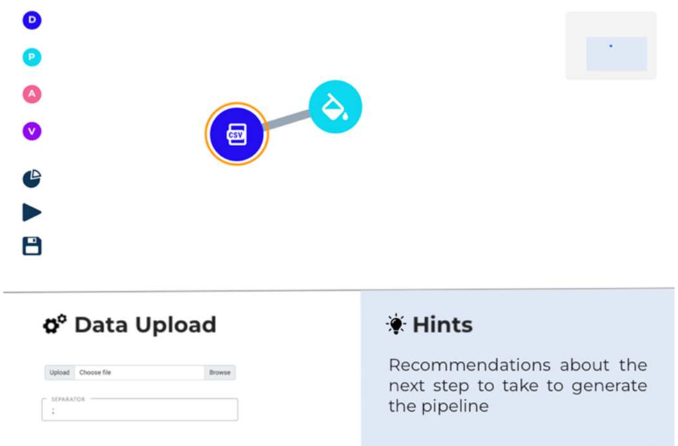

Fig. 5. Prototype of KoopaML's workspace after the UI improvement process.

V. Discussion

The proposed framework has been designed to provide an agile methodology to tackle UI issues in already-developed systems. It is important to remark that this framework is not designed to address the development of new systems, as for this matter, there are already several validated methodologies and frameworks [21-24].

This framework was born from the need to refactor the interfaces of some systems that we previously developed in the medical context. Since in this domain the functionalities were prioritized before the interface design (because of the urgency of integrating the tools into the department's technological ecosystem), the interface's problems were identified when the systems were already in production.

Given the fact that the main functional features of the systems were successfully validated, we needed to focus only on tackling UI issues, in need of a new framework to streamline this process.

The framework takes advantage of the SCRUM methodology by dividing the issues into different sprints, which enables a better prioritization of the development of the improvements. On the other hand, it integrates "micro-evaluations" at the beginning of each sprint, providing insightful information to identify the best UI design to achieve better usability levels. These evaluations are not fixed, so different UI testing techniques can be selected depending on each specific sprint issue.

We pilot this framework with two platforms integrated into the Cardiology Department of the University Hospital of Salamanca. The first platform, CARTIER-IA, had several UI problems, mostly related to the amount of information displayed on the screen. We conducted a set of evaluations to identify the specific sections of the screen that increased the information saturation, ending up with a cleaner result (Figure 3), which the domain experts successfully validated.

On the other hand, we are currently employing this framework to update the workspace of KoopaML, a platform to generate ML pipelines visually. Given that this platform is focused on easing the development of ML flows for non-expert users, providing a usable and intuitive interface is crucial. Several problems were found in the production version of the system related to the complexity of the interaction patterns and terminology. We conducted think-aloud evaluations and captured the interactions of users to identify the elements that hampered the design process of the ML pipelines. We will implement the improvements already validated with the domain experts in subsequent sprints.

Finally, it is necessary to point out that the framework's performance depends on the existing architecture of the system to improve. If the functional features are highly coupled with the UI components, then changes to the backend would be required, slowing down the process.

VI. Conclusions

This work proposes a framework for refactoring UIs following an agile methodology and data-driven decision-making processes. This framework aims to provide a set of systematic steps to improve UIs from already-developed systems.

The framework has been successfully integrated with two systems framed in the medical domain. Using this approach, we introduced new UI designs based on previously identified issues and issues found during the initial evaluations in each sprint.

Following this framework has reported different benefits, including a quick way to address problems detected in production and identifying problems that would otherwise have been discovered later after introducing the improvements, requiring additional time to fix them.

Future research lines are focused on continuing to validate this framework by integrating it into the improvement process of other systems' UIs.

References

[1] F. J. García-Peñalvo *et al.*, "Application of Artificial Intelligence Algorithms Within the Medical Context for Non-Specialized Users: the CARTIER-IA Platform," *International Journal of Interactive Multimedia and Artificial Intelligence,* vol. 6, no. 6, pp. 46-53, 2021, doi: 10.9781/ijimai.2021.05.005.

[2] A. Vázquez-Ingelmo *et al.*, "Bringing Machine Learning Closer to Non-Experts: Proposal of a User-Friendly Machine Learning Tool in the Healthcare Domain," in *Proceedings of the Ninth International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM'21) (Barcelona, Spain, October 26-29, 2021)*, M. Alier and D. Fonseca Eds., (ICPS: ACM International Conference Proceedings Series. New York, NY, USA: ACM, 2021, pp. 324-329.

[3] F. J. García-Peñalvo *et al.*, "KoopaML: A Graphical Platform for Building Machine Learning Pipelines Adapted to Health Professionals," *International Journal of Interactive Multimedia and Artificial Intelligence,* 2023, doi: 10.9781/ijimai.2023.01.006.

[4] A. García-Holgado *et al.*, "User-centered design approach for a machine learning platform for medical purpose," in *7th Iberoamerican Workshop, HCI-COLLAB 2021, Sao Paulo, Brazil (September 8–10, 2021)*, vol. 1478, P. Ruiz, V. Agredo Delgado, and A. Kawamoto Eds.: Springer, 2021, pp. 237-249.

[5] Z. Hussain, W. Slany, and A. Holzinger, "Current State of Agile User-Centered Design: A Survey," Berlin, Heidelberg, 2009: Springer Berlin Heidelberg, in HCI and Usability for e-Inclusion, pp. 416-427.

[6] J. Beux, E. Bellei, L. Brock, A. C. Bertoletti, and C. Hölbig, "Agile Design Process with User-Centered Design and User Experience in Web Interfaces: A Systematic Literature Review," *LAJC,* vol. 5, no. 2, pp. 53-60, 2018, doi: 10.5281/zenodo.5708753.

[7] G. Jurca, T. D. Hellmann, and F. Maurer, "Integrating Agile and User-Centered Design: A Systematic Mapping and Review of Evaluation and Validation Studies of Agile-UX," in *2014 Agile Conference*, 28 July-1 Aug. 2014 2014, pp. 24-32, doi: 10.1109/AGILE.2014.17.

[8] T. Silva da Silva, A. Martin, F. Maurer, and M. Silveira, "User-Centered Design and Agile Methods: A Systematic Review," in *2011 Agile Conference*, 7-13 Aug. 2011 2011, pp. 77-86, doi: 10.1109/AGILE.2011.24.

[9] L. Caballero, A. M. Moreno, and A. Seffah, "How Agile Developers Integrate User-Centered Design Into Their Processes: A Literature Review," *International Journal of Software Engineering and Knowledge Engineering,* vol. 26, no. 08, pp. 1175-1201, 2016, doi: 10.1142/s0218194016500418.

[10] O. Sohaib and K. Khan, "Integrating usability engineering and agile software development: A literature review," in *2010 International Conference On Computer Design and Applications*, 25-27 June 2010 2010, vol. 2, pp. V2-32-V2-38, doi: 10.1109/ICCDA.2010.5540916.

[11] D. Salah, R. F. Paige, and P. Cairns, "A systematic literature review for agile development processes and user centred design integration," presented at the Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering, London, England, United Kingdom, 2014. [Online]. Available: https://doi.org/10.1145/2601248.2601276.

[12] C. Salvador, A. Nakasone, and J. A. Pow-Sang, "A systematic review of usability techniques in agile methodologies," presented at the Proceedings of the 7th Euro American Conference on Telematics and Information Systems, Valparaiso, Chile, 2014. [Online]. Available: https://doi.org/10.1145/2590651.2590668.

[13] M. Brhel, H. Meth, A. Maedche, and K. Werder, "Exploring principles of user-centered agile software development: A literature review," *Information and Software Technology,* vol. 61, pp. 163-181, 2015/05/01/ 2015, doi: https://doi.org/10.1016/j.infsof.2015.01.004.

[14] M. Singh, "U-SCRUM: An Agile Methodology for Promoting Usability," in *Agile 2008 Conference*, 4-8 Aug. 2008 2008, pp. 555-560, doi: 10.1109/Agile.2008.33.

[15] J. Gonçalves and C. Santos, "POLVO - Software for Prototyping of Low-Fidelity Interfaces in Agile Development," Berlin, Heidelberg, 2011: Springer Berlin Heidelberg, in Human-Computer Interaction. Design and Development Approaches, pp. 63-71.

[16] D. Argumanis, A. Moquillaza, and F. Paz, "Challenges in Integrating SCRUM and the User-Centered Design Framework: A Systematic Review," Cham, 2020: Springer International Publishing, in Human-Computer Interaction, pp. 52-62.

[17] K. Schwaber, "Scrum development process," in *Business Object Design and Implementation: OOPSLA'95 Workshop Proceedings 16 October 1995, Austin, Texas*, 1997: Springer, pp. 117-134.

[18] A. Vázquez-Ingelmo *et al.*, "Usability Study of CARTIER-IA: A Platform for Medical Data and Imaging Management," in *HCII 2021: Learning and Collaboration Technologies: New Challenges and Learning Experiences*, Virtual event, 2021: Springer International Publishing, in Learning and Collaboration Technologies: New Challenges and Learning Experiences, pp. 374-384.

[19] F. J. García-Peñalvo *et al.*, "KoopaML: A graphical platform for building machine learning pipelines adapted to health professionals," *International Journal of Interactive Multimedia and Artificial Intelligence,* In Press.

[20] R. Jääskeläinen, "Think-aloud protocol," *Handbook of translation studies,* vol. 1, pp. 371-374, 2010.

[21] M. Selway, M. Stumptner, W. Mayer, A. Jordan, G. Grossmann, and M. Schrefl, "A Conceptual Framework for Large-scale Ecosystem Interoperability," in *Conceptual Modeling. Lecture Notes in Computer Science*, vol. 9381, P. Johannesson, M. L. Lee, S. W. Liddle, A. L. Opdahl, and Ó. Pastor López Eds. Cham: Springer, 2015, pp. 287-301.

[22] H. Edison, X. Wang, and K. Conboy, "Comparing Methods for Large-Scale Agile Software Development: A Systematic Literature Review," *IEEE Transactions on Software Engineering,* vol. 48, no. 8, pp. 2709-2731, 2022, doi: 10.1109/TSE.2021.3069039.

[23] J. Hunt, "Feature-Driven Development," in *Agile Software Construction*. London: Springer London, 2006, pp. 161-182.

[24] J. M. Rivero, J. Grigera, G. Rossi, E. Robles Luna, F. Montero, and M. Gaedke, "Mockup-Driven Development: Providing agile support for Model-Driven Web Engineering," *Information and Software Technology,* vol. 56, no. 6, pp. 670-687, 2014/06/01/ 2014, doi: https://doi.org/10.1016/j.infsof.2014.01.011.