Collab2US: a Tool for Collecting User Stories from Collaborative Boards

Pedro Lucas Dornelles and Rafael Parizi

Lardev Research Group – Federal Institute Farroupilha – São Borja/RS – Brazil
pedro.2023002752@aluno.iffar.edu.br, rafael.parizi@iffar.edu.br

Abstract. Software teams are increasingly adopting User-Centered Design (UCD) approaches to navigate complex problems, focusing on empathy, creativity, and iterative learning for a deep understanding of users' needs. Despite UCD's advantages, transitioning from conceptual prototypes to concrete requirements, such as detailed user stories, remains challenging. This paper introduces Collab2US, a tool designed to automate the conversion of UCD outcomes into structured user stories. Collab2US tackles the challenge of converting collaborative ideation outcomes into a structured product backlog of software requirements. It supports IT professionals' workflow by integrating Miro for ideation and Trello for backlog organization, streamlining the process of turning collaborative ideas into user stories. An initial evaluation of Collab2US has shown it can aid software teams in turning team insights into user stories for product backlogs.

Keywords: Software Development · Collaborative Work · User Stories

1 Introduction

Software teams have long embraced User-Centered Design (UCD) approaches to handle complex problems by incorporating empathy, creativity, and iterative learning [1]. This design philosophy motivates teams to deeply understand users' needs, redefine problems, and propose innovative solutions through rapid prototyping and validation [12].

Despite the benefits of using UCD approaches, transitioning from broad, conceptual outcomes as solution prototypes to concrete requirements, such as detailed user stories for product backlogs, still represents an open gap [14]. In addition, the shift to remote work has further highlighted the importance of collaborative tools in this context [10], emphasizing the need for methodologies that support seamless ideation and prototyping across distributed teams.

In this scenario, our research focuses on supporting IT professionals in effectively translating user-focused insights generated from UCD processes into requirements. To do this, we developed Collab2US, a tool that automates the conversion of UCD outcomes into structured user stories.

To enhance the transition from brainstorming to execution, Collab2US leverages tools such as Miro to enable teams to visually organize their ideas and insights. This approach facilitates a more seamless move from ideation to practical

Proceedings of the 27th Workshop on Requirements Engineering (WER24), August 07-09, 2024, Buenos Aires, Argentina.
DOI: 10.29327/1407529.27-22

application. Early outcomes suggest that Collab2US aids in incorporating UCD insights into the requirements engineering phase. This process ensures that software development is more directly aligned with the requirements and preferences of users, leading to more user-friendly and effective software solutions.

The remainder of this paper is organized as follows: Section 2 introduces the Collab2US tool, adhering to the Design Science Research (DSR) methodology, which encompasses Problem Understanding, Solution Design, and Early Validation. Section 3 presents the preliminary results from using the tool, and Section 4 concludes with final considerations and outlines future work.

2 Collab2US tool

This section details the Collab2US tool. Section 2.1 outlines our approach to problem understanding, while Section 2.2 sheds light on the solution design. Section 2.3 offers a brief initial validation.

Collab2US aims to bridge the gap between the outcomes of collaborative ideation and prototyping sessions inspired by User-Centered Design methods and the creation of a backlog containing a list of software requirements. The aim is to contribute to addressing the research question: **How can software teams effectively translate UCD-based artifacts into backlog items, ensuring that user-centric insights are seamlessly integrated into the Requirements Engineering process?**

Inspired by the DSR methodology for the conception and design of Collab2US, we incorporated insights from Runeson et al. (2020) [13], targeting practical problems through the design and analysis of artifacts within their context. DSR is focused on enhancing problem spaces by generating tailored, human-designed solutions. DSR is characterized by its emphasis on the novelty of the artifact and its theoretical and practical contributions.

Figure 1 details our DSR-based research methodology. We implemented the Design Science Research framework iteratively, beginning with the Problem Understanding activity, followed by Solution Design and Validation activities.

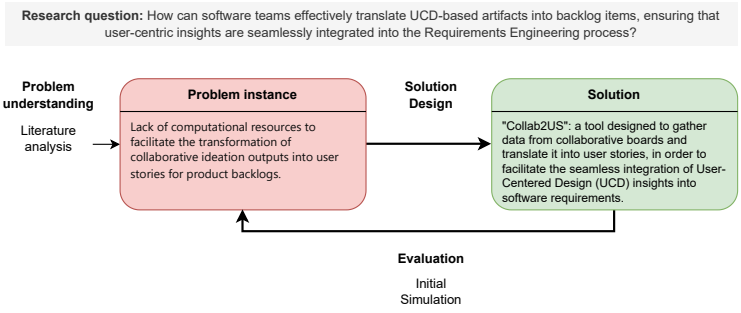

Fig. 1. DSR-based Solution Design

2.1 Problem Understanding

To understand the problem, we analyzed studies from literature, including (i) Parizi *et al.*(2022) [7], which emphasizes the growing adoption of UCD techniques in software development, highlights challenges and the lack of tools to support the use of Design Thinking (DT) in software development, including the proposal for a DT techniques recommendation system [6,8,9]; (ii) Alhazmi *et. al* (2020)[2], which presents an integrated framework of DT and Scrum specifically for requirements engineering management, underscoring the need to overcome certain endeavor, such as achieving a comprehensive understanding of customer needs and addressing the implications of challenges like frequent requirements changes, and; (iii) Correa *et. al* (2021)[11] who, despite recognizing the benefits of applying UCD techniques in software development, underlined the challenges of their integration with agile methods. So, based on the studies' results, we defined the following problem instance (Figure 1): Lack of computational tools to facilitate the transformation of collaborative ideation outputs into user stories for product backlogs.

2.2 Solution Design

Inspired by the problem instance, we proposed Collab2US to support software teams by automating the gathering of insights from collaborative boards and converting these insights into user stories. In addition, our proposal is to create and add the stories into a Kanban-style project board that serves as the Product Backlog. Figure 2 illustrates the application scenario for Collab2US.

Collab2US integrates 2 tools (i) Miro[1], which provides a synchronous and remote collaboration environment open to ideation activities, and (ii) Trello[2], which offers a space for activity management and project administration. The tool is available at `https://collab2usvfinal.vercel.app/`. In the first version of Collab2US, we focus on the use of Miro and Trello once they are two tools widely known by software development professionals.

To use the tool, users must provide an API (Application Programming Interface) key for Miro and both an API key and a connection Token for Trello. These details can be easily obtained on their respective platforms. Once connected to Collab2US, users will gain access to their Miro boards and have the capability to generate user stories from the information gathered by Collab2US. Furthermore, they can create a Trello board to organize the product backlog.

The initial version of Collab2US requires as input Miro boards structured into frames according to a template [3] (Figure 3): ≺Personas≻, ≺Features≻, and ≺Intents≻, defined as follows:

≺Personas≻ represent the target users and their respective contexts, profiles, and needs within the project (e.g. Keynote speaker);

[1] `https://miro.com`
[2] `https://trello.com`
[3] `https://miro.com/app/board/uXjVKd2aXrk=/?share_link_id=915205396090`

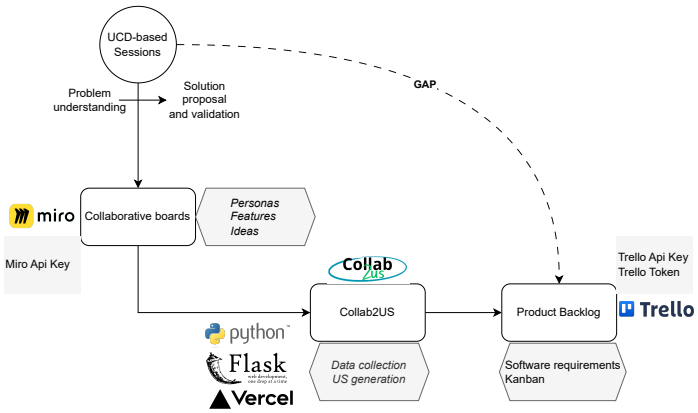

Fig. 2. Collab2US application scenario

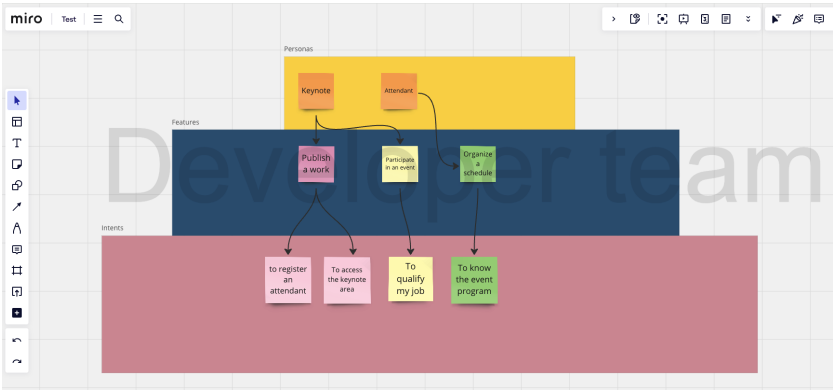

Fig. 3. Miro Template for Collab2US

≺Features≻ denote the functionalities or specific capabilities that the product must provide to meet the needs of the personas (e.g. To publish a work); ≺Intents≻ encapsulate the intentions or goals users have when interacting with the product or service, reflecting the actions they wish to perform (e.g. To access the keynote area).

Once the user stories for the selected board have been generated, the user can proceed to create a Trello board through Collab2US to serve as the product backlog. Collab2US uploads the user stories to Trello, organizes the kanban with spaces for To Do, Doing, and Done, and places the user stories, organized by persona and feature, in the To Do space. Finally, the user is notified by the tool whether the backlog creation process has been successfully completed.

Table 1 displays the set of features of Collab2US. For the implementation of Collab2US, we utilized the Python language. We connected to the APIs of Miro, to collect data from collaborative boards, and of Trello, to generate boards

Table 1. Collab2US' features

Features	
Connect to Miro to collect insights in collaborative boards	Generate user stories based on the relationships of the collected elements
Collect the Miro boards of the connected user	Present the generated user stories to the user
Analyze if the Miro board is in accordance with the required template	Connect to Trello to organize the product backlog
Collect the personas defined in the selected Miro board	Generate a Trello board referring to the selected Miro Board
Collect the features defined in the selected Miro board	Generate lists of cards in Trello following the Kanban model To Do, Doing, and Done
Collect the intents defined in the selected Miro board	Create a card for each generated user story in the To Do list
Capture the connections made between the elements Personas, Features, and Intents	Create Tag labels for the cards using the Personas and Features
Notify the user that the cards have been created in Trello	Provide help on how to use the tool

with requirements in the form of user stories. We employed the Python Flask framework[4] to implement a web version of the tool, which is hosted on a web server provided by the Vercel platform[5].

2.3 Tool's Early Evaluation

To conduct an initial evaluation of the tool, we collaboratively created two boards on Miro, following the model suggested in Section 2.2. During this preliminary phase, our goal was to deeply understand the capabilities of the tool and its alignment with our expectations based on predefined requirements. The collaborative creation of the boards allowed us to explore the tool's functionality in real-time, facilitating a practical assessment of its features and usability. This approach provided valuable insights into how the tool could support our workflow and enhance efficiency in translating collaborative ideas into user stories.

3 Preliminary Results

To initially evaluate Collab2US, we utilized two test boards created in Miro. To this end, the authors of this paper simulated a DT session to assist in proposing a system for aiding the organization of an academic event.

The process began with an ideation activity in Miro (Figure 3). Next, the ideas were organized according to the template provided by the Collab2US tool (Personas, Features, and Intents). With the Miro board prepared, we utilized the Collab2US tool (Figure 4). In Figure 4–A, we connected to the tool by entering the Miro API Key (to connect to Miro and gather the board data) and the Trello API Key and Token (to generate the board, cards, and user stories in Trello).

Subsequently, Collab2US displayed the available boards in Trello for the connected account (Figure 4–B). Then, through the "Generate" button, Collab2US

[4] `https://flask.palletsprojects.com/en/3.0.x/`
[5] `https://vercel.com/`

generated the user stories from the Miro board. Collab2US then presented the Personas, Features, and Intents, as shown in Figure 4–C. Finally, via the "Export to Trello" button, the tool created a Trello board containing the system features proposed by the participants (Figure 4–D). Figure 5 displays the Trello Board generated in Trello by the Collab2US tool, with the system requirements.

As a result, the initial version of the Collab2US tool shows potential to assist development teams in translating insights developed in activities based on UCD methods, using Miro, into user stories in Kanban format, contributing to the construction of the product Backlog with the system requirements.

There are related tools for generating user stories, each addressing different aspects of software development. For instance, AQUSA [4] focuses on evaluating and improving the quality of user stories using Natural Language Processing techniques. The Semi-Automatic User Stories Generation tool [5] measures user experience by creating user stories through intelligent agents that simulate character interactions. In contrast, Collab2US facilitates the initial conversion of collaborative design ideas into structured user stories by integrating Miro and Trello, streamlining the ideation-to-backlog process.

4 Final Considerations and Future Work

In this work, we presented Collab2US tool to bridge the gap between collaborative ideation and prototyping sessions inspired by UCD methods and the creation of a software requirements backlog. Drawing from the DSR methodol-

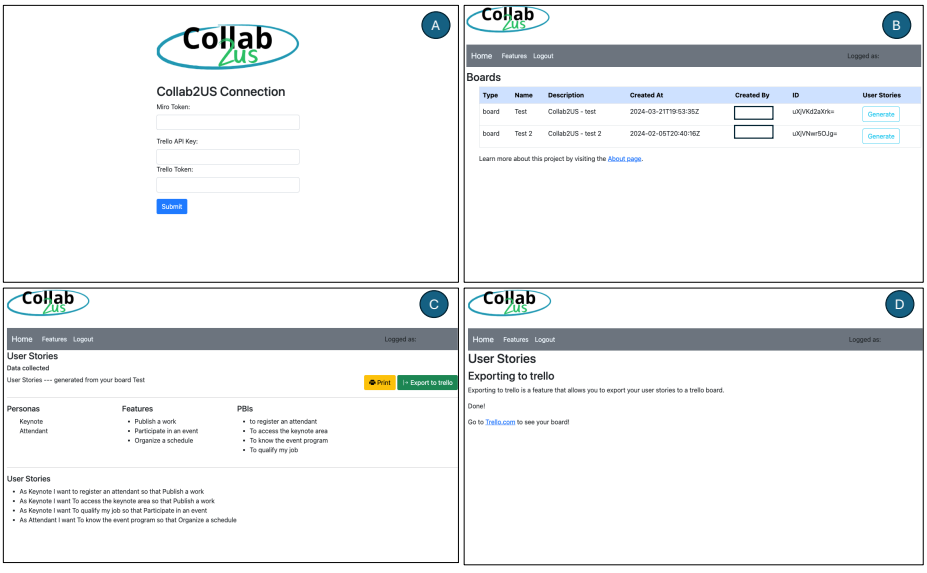

Fig. 4. Collab2US Screens

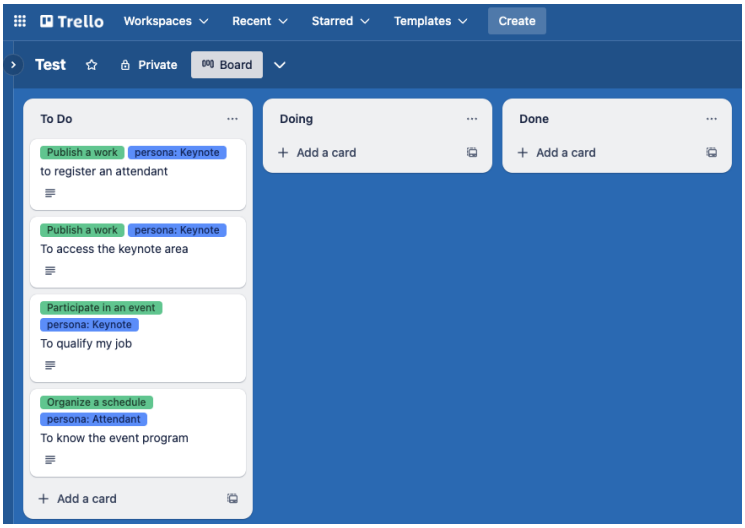

Fig. 5. Trello board generated by Collab2US

ogy, we addressed the challenge of translating collaborative outputs into user stories for product backlogs.

Through an iterative process involving problem understanding, solution design, and early validation, Collab2US has shown promise in facilitating a seamless transition from ideation to implementation. Our initial evaluation, leveraging the Miro tool for collaborative board creation and Trello for backlog organization, indicates that Collab2US effectively captures user needs and supports agile project management practices. This work lays the foundation for future enhancements and wider adoption, aiming to improve software development workflows by integrating UCD insights into agile methodologies.

While the results are preliminary and controlled, we acknowledge the need for external evaluations in production environments to mitigate potential biases from the developers. For future work, we plan to implement features like Trello and Miro account logins to enhance integration and user experience. Additionally, we will use the Technology Acceptance Model (TAM) [3] to assess professionals' willingness to use Collab2US and conduct real-world experiments to evaluate its effectiveness, efficiency, and user satisfaction. This combined approach will provide a comprehensive assessment of Collab2US's impact on software development by generating user stories from UCD-based boards.

Tool's Video: `https://doi.org/10.5281/zenodo.11398419`

Acknowledgments

The authors gratefully acknowledge the support provided by the Federal Institute of Education, Science and Technology Farroupilha (IFFar).

References

1. Adikari, S., Keighran, H., Sarbazhosseini, H.: Embed Design Thinking in Co-Design for Rapid Innovation of Design Solutions. In: Proc. of the Int'l Conference of Design, User Experience, and Usability. pp. 3–14. Springer, Toronto, Canada (2016)
2. Alhazmi, A., Huang, S.: Integrating Design Thinking into Scrum Framework in the Context of Requirements Engineering Management. In: Proceedings of the International Conference on Computer Science and Software Engineering. pp. 33–45. ACM, Beijing, China (2020)
3. Davis, F.D.: Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology. MIS Quarterly **13**(3), 319–340 (Sep 1989)
4. Lucassen, G., Dalpiaz, F., van der Werf, J.M.E., Brinkkemper, S.: Improving agile requirements: the quality user story framework and tool. Requirements engineering **21**, 383–403 (2016)
5. Olaverri-Monreal, C., Hasan, A.E., Bengler, K.: Semi-automatic user stories generation: To measure user experience. In: 2013 8th Iberian Conference on Information Systems and Technologies (CISTI). pp. 1–6 (2013)
6. Parizi, R., Couto, I., Hanauer, L., Marczak, S., Conte, T.: Helius: On a Recommendation System of Design Thinking Techniques for Software Development based on Professionals' Collaboration. In: Proceedings of the Workshop on Requirements Engineering. vol. 45, pp. 80–93. PUC-RIO, Virtual Conference (Dec 2020)
7. Parizi, R., Prestes, M., Marczak, S., Conte, T.: How has Design Thinking being Used and Integrated into Software Development Activities? A Systematic Mapping. Journal of Systems and Software **187**, 1–27 (5 2022)
8. Parizi, R., Moreira da Silva, M., de Souza Couto, I., Pavin Trindade, K., Plautz, M., Marczak, S., Conte, T., Candello, H.: Design Thinking in Software Requirements: What Techniques to Use? A Proposal for a Recommendation Tool. In: Proceedings of the Ibero-American Conference-American on Software-American Engineering. p. 14. Curran Associates, Curitiba, Brazil (2020)
9. Parizi, R., da Silva, M.M., Couto, I., dos Santos Marczak, S., Conte, T.: A tool proposal for recommending design thinking techniques in software development. Journal of Software Engineering Research and Development (2021)
10. Park, J., Mostafa, N.A., Han, H.J.: StoryWeb: A Storytelling-based Knowledge-sharing Application Among Multiple Stakeholders. Creativity and Innovation Management **29**(2), 224–236 (Apr 2020)
11. Pereira, L., Parizi, R., Prestes, M., Marczak, S., Conte, T.: Towards an Understanding of Benefits and Challenges in the Use of Design Thinking in Requirements Engineering. In: Proceedings of the Annual ACM Symposium on Applied Computing. p. 1338–1345. ACM, Virtual Event, Republic of Korea (2021)
12. Prasad, W.R., Perera, G., Padmini, K.J., Bandara, H.D.: Adopting Design Thinking Practices to Satisfy Customer Expectations in Agile Practices: A Case from Sri Lankan Software Development Industry. In: Proceedings of the Moratuwa Engineering Research Conference. pp. 471–476. IEEE, Moratuwa, Sri Lanka (2018)
13. Runeson, P., Engström, E., Storey, M.A.: The Design Science Paradigm as a Frame for Empirical Software Engineering, vol. 1, chap. 5, pp. 127–147. Springer, Heidelberg, Germany, 1 edn. (2020)
14. Shania, M., Raharjo, T., Fitriani, A.N.: Implementation user-centered design in agile software development: Systematic literature review. Indonesian Journal of Multidisciplinary Science **2**(7), 2812–2831 (2023)