Helius: A Design Thinking Techniques Recommendation System in Software Development

Rafael Parizi
Federal Institute Farroupilha
São Borja, Brazil
rafael.parizi@iffar.edu.br

Lucas Hanauer
Pontifical Catholic University of Rio Grande do Sul
Porto Alegre, Brazil
lucas.hanauer@edu.pucrs.br

Sabrina Marczak
Pontifical Catholic University of Rio Grande do Sul
Porto Alegre, Brazil
sabrina.marczak@pucrs.br

Tayana Conte
Federal University of Amazonas
Manaus, Brazil
tayana@icomp.ufam.edu.br

ABSTRACT

This paper demonstrates Helius, a collaborative Design Thinking (DT) techniques recommendation system. We proposed Helius to support DT practitioners in selecting DT techniques for use in software development, as literature has shown that selecting a set of DT techniques can be a complex decision-making problem. To develop Helius, we followed a Design Science Research framework, starting with a deep understanding of the problem and followed by three iterations of design and validation activities. In total, 40 professionals participated in the validation activities for Helius. The results show that Helius effectively supports the selection of DT techniques and is considered by DT practitioners to be a useful and easy-to-use recommendation system.

Helius Tool's video: https://doi.org/10.5281/zenodo.11398135

KEYWORDS

Software Development, User-centered Design, Design Thinking

1 INTRODUCTION

Software companies have been using Design Thinking as a User-Centered Design approach, bringing the user to the center of the development process. DT aids software teams to collect the user's real needs and to propose viable, feasible and desirable solutions[8].

Literature has shown that given the dynamic nature of DT and the need to efficiently explore the creativity and collaboration of multidisciplinary teams to discover, frame and propose solutions that meet the users' needs, selecting the DT techniques to use becomes a complex decision-making endeavor [13, 14]. In two previous exploratory studies, a Systematic Literature Mapping [12] and a Survey [15], we identified more than 80 DT techniques used in software development. We also figured out that the professionals who use DT have to deal with some challenges when selecting the appropriate DT techniques to use including, for instance, the need of defining contextual factors such as time, the challenge to be solved, etc; time pressure, and lack of participant commitment.

In this context, aiming to support IT professionals to use DT in software development, we designed, implemented and empirically validated Helius. Helius is a collaborative recommendation system that provides 20 recommendation mechanisms classified into personalized and non-personalized recommendations. It uses the ratings provided by users to DT techniques to retro-feed the recommendation algorithms. It also provides access to a community of practices on the experiences of IT professionals on the use of DT in software development.

Thus, this paper demonstrates Helius. It is the result of a long-term research agenda, in which we followed the Design Science Research (DSR) method to identify and define a research problem (the lack of support on the selection of DT techniques) and to iterate in the design and validation approaches to propose a suitable solution to the defined research problem.

This paper is structured as follows: Section 2 details Helius, including its functionality, the collaborative design and validation process with DT practitioners and researchers, its features, and a series of screenshots illustrating its usage. Section 3 describes a validation study conducted for Helius, while Section 4 compares it with similar tools. Finally, Section 5 concludes the paper and outlines future research directions.

2 HELIUS: DT TECHNIQUES RECOMMENDATION SYSTEM

Helius is an outcome of a research agenda following the Design Science Research method. We followed the DSR framework for Software Engineering proposed by Runeson et al. (2020) [16]. The framework suggests starting with a deep understanding of the problem to be solved, followed by iterations of design and validation of the proposed solutions. In addition, the framework also fosters the researchers to constantly evaluate the rigor, relevance and novelty of the artifacts generated.

Figure 1 shows a simplified version of our research design.

2.1 Problem Definition

We began our research by conducting two exploratory studies: a Systematic Mapping Study [12] and an Exploratory Survey [15] on the use of DT in software development. These studies were conducted to identify gaps in both the state of practice and the state of the art regarding DT in software development. Then, aiming to better understand the problem and propose a suitable solution, we conducted a meta-DT session[1]. The meta-DT session followed the double diamond DT model [5]. For the problem understanding activity, we explored the first half of the model (problem space).

[1] We named the DT session a meta-DT session since it focused on finding solutions for a DT problem.

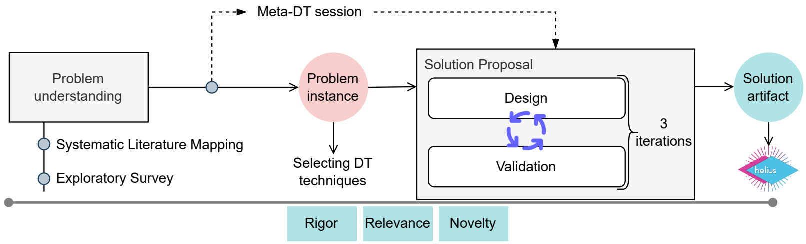


Figure 1: Research Design based on the DSR-framework

Table 1: Iterations and activities

It	Design	Validation	# Participants	Main artifact(s) generated
1	**Requirements elicitation**: Brainwriting, Insight Cards, Affinity Diagrams, Personas, Service Blueprints, User Journey and Low-level fidelity prototypes	Interview-based study; Qualitative data analysis	5 experts in DT from industry	List of requirements; 19 low-level fidelity prototypes
2	**Requirements refining:** transcription of the 19 low-level fidelity prototypes into 62 high-level prototypes and refined the blueprints and user journeys	Perceived Usefulness and Ease of Use through the Technology Acceptance Model (TAM); Feature analysis comparison using DESMET method	7 moderators using DT in the software development	Refined list of requirements; 62 high-level prototypes
3	**Decision-making modeling & Helius implementation:** interview-based study based on Grounded Theory to identify the decision-making elements behind the selection of DT techniques; implementation of 20 recommendation mechanisms of DT techniques in Helius (personalized and non-personalized recommendations)	Empirical validation of Helius with professionals from industry through interviews and using the System Usability Scale method	12 (interview) + 16 (validation) DT moderators in software development	Helius: DT techniques recommendation system

A Requirements Analyst with more than 7 years of experience in DT conducted the meta-DT session. Ten professionals from academia and industry participated in the session. The moderator presented the results of our exploratory studies to the participants. The presentation served as a seed to support the problem identification and definition. After 1 hour of discussing the topic, the participants identified that DT practitioners consider the selection of DT techniques as a decision-making problem and proposed the following problem instance:

> Literature offers a plethora of DT tools and methods (or techniques for simplification) that form the toolkit to perform DT activities. Nevertheless, there is a lack of strategies to support the decision process of which techniques to use and detailing which contextual factors (e.g., previous knowledge about the problem to be solved, customer engagement, etc.) affect such decision.

2.2 Iterative Design and Validation Approach

Once the research problem was defined, we performed 3 iterations in the DSR-based method to design and validate Helius. In each iteration, we designed and validated a set of artifacts. A range of DT practitioners participated in both design and validation activities, providing us with insights and feedback to develop our solution.

Table 1 summarizes each iteration, describing the design and validation activities that we performed, the number of participants involved and the artifacts generated.

2.2.1 Iteration 1 – Solution Proposal and Early Evaluation. We started Iteration 1 during the second half of the Meta-DT session, by exploring the second diamond of the Double Diamond DT model which focuses on a solution proposal to an identified problem [9–11]. The participants proposed a computational resource to support DT practitioners in the selection of DT techniques in software development. The resource should be able to recommend DT techniques based on the experience of the use of DT techniques by a community of DT practitioners, i.e., a collaborative recommendation system. Next, as a design activity, we performed a requirements elicitation process. To validate the requirements, we interviewed experts in DT.

2.2.2 Iteration 2 – Solution Refining and Initial Validation. In Iteration 2, we used the feedback that we collected in Iteration 1 as a starting seed to refine our proposed solution. Thus, in the design activity we transcribed the low-level fidelity prototypes into high-level prototypes based on the user journeys and service blueprints. To validate the refined version of Helius, we first conducted a study using the Technology Acceptance Model (TAM) [6]. We collected the Perceived usefulness and the perceived ease of use from 7 professionals from industry who used Helius (prototyped version). Next, we compared Helius to similar tools using DESMET [7]. DESMET is a feature analysis method that allows the comparison of alternative solutions. We find out that Helius innovates in comparison to other similar tools, as we show in [9].

2.2.3 Iteration 3 – Solution Evolution and Validation. In Iteration 3 we aimed to improve Helius by designing the mechanisms responsible for recommending DT techniques based on the experiences of DT practitioners (the collaborative feature of Helius). In the Design activity, we used Grounded Theory as a research method to interview professionals and collected the decision-making behind the selection of DT techniques [17]. Our goal was to understand how professionals decide which techniques to use and implement recommendation mechanisms based on that decisions. To validate the recommendation mechanisms, we conducted an empirical study with 16 professionals from the software industry. We interviewed them and also collected data using the System Usability Scale (SUS) method [3]. The results showed that Helius contributes to the selection of DT techniques by recommending techniques based on the experience of other professionals.

2.3 Helius’ features in a nutshell

As a result of the 3 iterations, we developed Helius as a collaborative DT techniques recommendation system that considers the users’ ratings to DT techniques to retro-feed the recommendation algorithms. Aiming to support both novices and experts on the use of DT techniques in software development, Helius provides the following features:

- *Creation of DT techniques projects, allowing the user to require DT techniques recommendations*: the user may create a DT techniques portfolio (project for simplification) including a title, description, participants, context, domain, date, and the DT techniques that she wants to use. The addition of DT techniques might be done in two different ways: (i) adding DT techniques that the user already knows or has used before or (ii) asking Helius for recommendations of DT techniques;
- *Review of the DT techniques used in software development*: the user can rate the techniques she has used, including a description of the experience of using the technique, the participants of the DT activity when using the technique, other techniques that were used in combination, the cost of using a DT technique and the time spent to use it;
- *Access to a community of practice* on the use of DT techniques: the user may access a space to collect the experiences of other users by using DT techniques;
- *Access to a DT techniques combination graph:* the user may access a graph that draws vertices and edges between techniques, showing how the user has been using the DT techniques in combination. For instance, the graph aims to complete the statement: “Users who are using XyZ techniques are also using ABC techniques”;
- *Access to a catalog of DT techniques:* The user can get detailed information on DT techniques from Helius. The information includes: name, how to use, when to use, materials needed, features and references;
- *Follow other professionals who used DT techniques to discover their experiences with DT techniques:* the user can follow other users and see how they have used DT techniques for software development.

2.4 Helius’ Architecture

Figure 2 shows the architecture of Helius, a multi-platform recommendation system designed to facilitate the selection of DT techniques. The architecture includes the front end, back end, and a recommendation module.

We developed Helius’ front-end using Flutter[2], a versatile programming language that supports deployment on the Web, Mobile devices (including iOS and Android), and Desktops. The front-end interacts with the users, allowing them to request DT technique recommendations and receive responses. It communicates with the back end via HTTP/JSON.

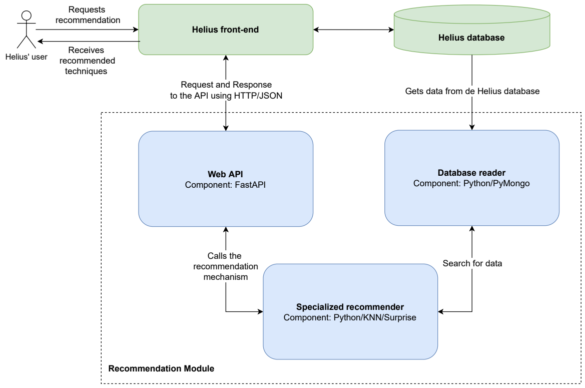

Requests recommendation
Receives recommended techniques
Helius' user
Helius front-end
Helius database
Request and Response to the API using HTTP/JSON
Gets data from de Helius database
Web API
Component: FastAPI
Database reader
Component: Python/PyMongo
Calls the recommendation mechanism
Search for data
Specialized recommender
Component: Python/KNN/Surprise
Recommendation Module

Figure 2: Helius architecture

The Helius database is a MongoDB[3] instance, chosen for its flexibility, scalability, and efficiency. MongoDB’s flexible document schemas allow for easy adaptation to changing requirements, while its powerful querying and analytics capabilities support the complex data operations required by Helius.

The back end comprises multiple components, including the Web API and the Database reader. The Web API, implemented using FastAPI[4], handles incoming requests from the front end, processes them, and interacts with the database and recommendation module. FastAPI was chosen for its high performance and ease of integration with Python-based components.

The Recommendation module is the core of Helius, consisting of specialized recommenders implemented in Python. It uses the K-Nearest Neighbors (KNN) algorithm and the Surprise library to generate recommendations. This module includes 20 different recommendation mechanisms, classified into personalized and non-personalized. Personalized recommendations take into account the user’s profile, leveraging the Pearson Correlation Coefficient to identify similar users and suggest techniques based on their preferences. Non-personalized recommendations provide the Top-N DT techniques, such as the most used or best-rated techniques, applicable to all users.

2.5 Helius’ Screenshots & Usage

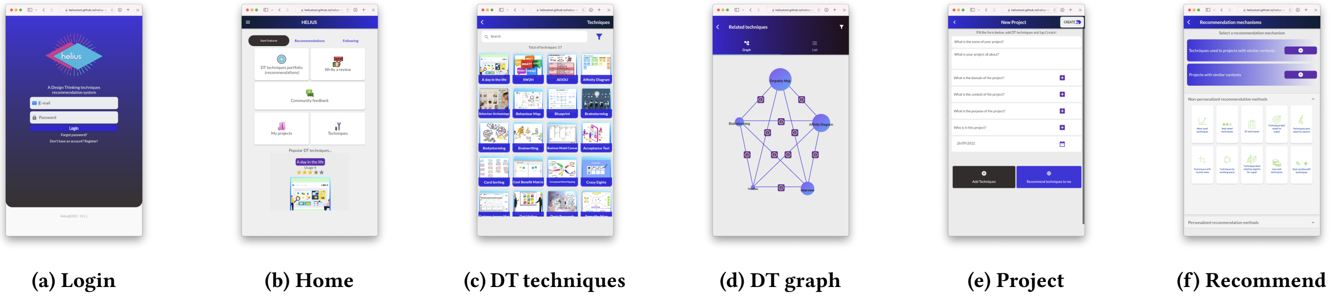
Figure 3 shows a subset of screenshots of Helius. To use Helius, the user might execute the following tasks:

1. **Access Helius** by clicking on: https://heliustool.github.io/helius-web/.
2. **Create an user account**. After, **log in** in Helius by using your user account data (Figure 3a);
3. Once logged in, Helius shows the Home screen (Figure 3b). Then, the user might **access a list of DT techniques** by clicking on the Techniques button. Helius will open a list of DT techniques (Figure 3c);

[2] https://flutter.dev/

[3] https://www.mongodb.com/

[4] https://fastapi.tiangolo.com/



(a) Login (b) Home (c) DT techniques (d) DT graph (e) Project (f) Recommend

Figure 3: A sample of Helius screenshots

(4) Through the Home Screen (Figure 3b) the user can access a graph of DT techniques (combinations of techniques) by clicking on a drawer menu (top left menu). Then, Helius draws the graph of DT techniques (Figure 3d);
(5) The Home Screen also provides access to the creation of a DT techniques project, a **portfolio of DT techniques** (Figure 3e) where the user might require for recommendations (Figure 3f). Helius provides 20 recommendation mechanisms of DT techniques, classified into personalized and non-personalized mechanisms;

3 VALIDATION STUDY

This section describes an empirical study designed to validate Helius, aiming to assess its effectiveness in aiding DT practitioners in selecting suitable DT techniques for software projects. We followed to empirical research guidelines proposed by Wohlin *et al.* (2012) [19] and Travassos (2002) [18]. Our study covered phases such as scoping and planning, execution, data analysis and results presentation, as illustrated in Figure 4.

3.1 Scoping and Planning

To define the scope of our study, we employed the Goal-Question-Measure (GQM) template to outline the context, goal, research questions, and measures [2]. Specifically, the study targeted practitioners involved in DT sessions or projects within the software development industry (Table 2). The central research question posed was: "How did DT practitioners perceive Helius's recommendations?" We focused on practitioners' perceptions of Helius's recommendations and their influence on decision-making processes.

Next, we planned our study by addressing ethical considerations, including preparing a participant consent form[5]. Our study focused on professionals directly involved in conducting DT sessions within software development settings.

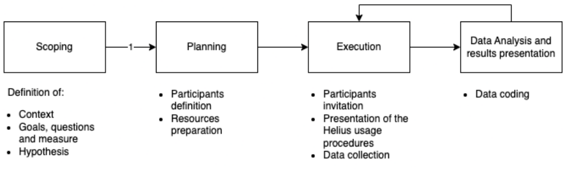

Scoping
1
Planning
Execution
Data Analysis and results presentation
Definition of:
• Context
• Goals, questions and measure
• Hypothesis
• Participants definition
• Resources preparation
• Participants invitation
• Presentation of the Helius usage procedures
• Data collection
• Data coding

Figure 4: Activities of the empirical evaluation of Helius

Table 2: GQM template for evaluating Helius

Item	Description
To analyze	the use of Helius, the DT techniques recommendation system
with the purpose of	supporting the decision-making for selecting DT techniques,
with respect to	select the DT techniques through the collaboration of DT practitioners
under the perspective of	DT practitioners using DT in software development
in the context of	software development

3.2 Execution

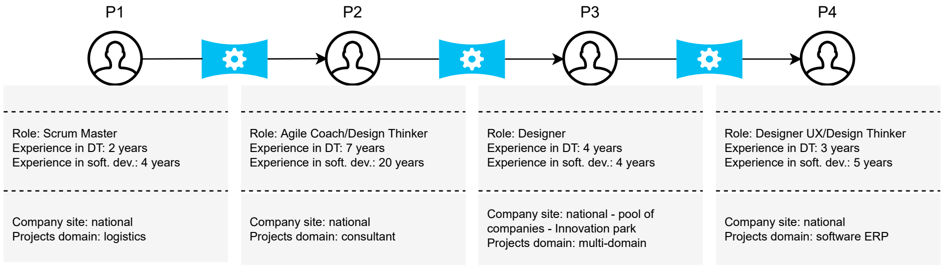
The execution phase of our empirical study involved inviting and selecting participants, introducing Helius, and detailing data collection and analysis procedures. We carried out the study in two main steps: (1) an interview study and (2) a questionnaire-based feedback collection. This approach supported us to guarantee the reliability of the data and the validity of our findings.

3.2.1 Step 1 – Interview-based Study. Step 1 of our empirical study introduced DT practitioners to Helius, encouraging its use in their professional activities. We invited 8 practitioners working in software companies; 4 agreed to participate - P1-P4 (Figure 5). We conducted an initial presentation about Helius on the Zoom platform[6], where participants familiarized themselves with Helius' features. After using Helius for approximately 15 days (see Table 3), we collected feedback in follow-up meetings.

3.2.2 Step 2 – Questionnaire-based Study. In Step 2 of our validation study, we launched the refined Helius tool based on the feedback from Step 1. Using LinkedIn[7], we invited DT practitioners to join our study; 12 professionals actively using DT in software development responded (see Table 4). We sent details of Helius' features and the study's objectives to the participants. They then used Helius to create projects, add and rate DT techniques, and interact with the community feature. We gathered data through an online questionnaire. To assess Helius' usability, we utilized the System Usability Scale (SUS) [4], which includes 10 statements rated on a 5-point Likert scale to measure user satisfaction.

[5] This project was submitted to and approved by the Ethics Committee, ensuring compliance with ethical research standards.

[6] https://zoom.us
[7] https://www.linkedin.com



Role: Scrum Master
Experience in DT: 2 years
Experience in soft. dev.: 4 years

Company site: national
Projects domain: logistics

Role: Agile Coach/Design Thinker
Experience in DT: 7 years
Experience in soft. dev.: 20 years

Company site: national
Projects domain: consultant

Role: Designer
Experience in DT: 4 years
Experience in soft. dev.: 4 years

Company site: national - pool of companies - Innovation park
Projects domain: multi-domain

Role: Designer UX/Design Thinker
Experience in DT: 3 years
Experience in soft. dev.: 5 years

Company site: national
Projects domain: software ERP

Figure 5: Empirical study – Participants of the Step 1

Table 3: Helius usage by the study's participants

	Features			
P#	**Project created**	**Project's domain**	**Techniques' rating**	**Users' experiences (Community of practice)**
P1	1	Hotel management system	9 (Affinity diagram, Brainstorming, Card Sorting, Empathy map, Feedback Grid, Insight Cards, Interview, Personas, Paper prototyping)	yes
P2	1	ERP to control devices in hospitals and schools	3 (A day in the life, Ideas Menu, Insight Cards)	yes
P3	1	Mobile Application for testing Helius	5 (Interview, Affinity Diagram, Try it yourself, Brainstorming, Empathy map)	yes
P4	2	Simulating system Testing project	8 (Observation, Stakeholder Map, 5w2H, Exploratory research, Card Sorting, Empathy map, A day in the life, I like I wish) 5 (Personas, Interview, Empathy Map, Insight Cards, Feedback Grid)	yes

Table 4: Participant's demographic data (Questionnaire)

P#	Background	Current role	Experience in DT (Yrs)	Experience in Software Development (Yrs)	Company domain	Company site
P5	Computer Science	Lead Design Thinking / Cloud Support Specialist	4 - 5	4 - 5	Software house	Multinational
P6	Design	UX Designer/Design Thinker	2 - 3	4 - 5	Personal loan solution	National
P7	Computer Science	Developer/Lecturer	2 - 3	4 - 5	Software for education	Multinational
P8	Design	UX Designer	0 - 1	0 - 1	Software house	National
P9	Management	Product Designer	2 - 3	2 - 3	Software house	National
P10	Management	Agile Manager	4 - 5	+7	Software house	Multinational
P11	Design	Head of Design	+7	4 - 5	Software house	National
P12	Computer Science	Software Engineer	2 - 3	4 - 5	Health systems	National
P13	Computer Science	Product Manager	+7	+7	Tech & innovation	Multinational
P14	Computer Science	Data scientist/UX analist	4 - 5	+7	Data Dashboards	National
P15	Design	UX designer	2 - 3	2 - 3	B2B solutions	National
P16	Design	Team Leader	2 - 3	4 - 5	Bank systems	Multinational

3.3 Data Analysis and Results Presentation

We began data analysis immediately after each interview and questionnaire response, enabling ongoing refinement of Helius. The insights gathered from both steps were synthesized to evaluate Helius' effectiveness and utility in supporting DT practitioners in selecting appropriate DT techniques. This comprehensive study not only informed further development of Helius but also validated its application in real-world DT contexts.

The feedback for RQ used a 5-item Likert scale [1] (Table 5 for interviews and Table 6 for questionnaires) and revealed varied perceptions of the relevance of recommendations. Some practitioners appreciated being introduced to new techniques, while others preferred recommendations that aligned more closely with their previous experiences. This feedback prompted enhancements in Helius, specifically allowing users to access comprehensive information about techniques as they are recommended, thus improving usability and aiding better decision-making in selecting appropriate DT techniques.

Table 5: Participant's perceptions of the Helius' recommendations - Interview Study

Participants	Likert-item	Recommendations' importance
P1	① ② ③ ❹ ⑤	Somewhat important
P2	① ② ③ ④ ❺	Important
P3	① ② ③ ④ ❺	Important
P4	① ② ③ ❹ ⑤	Somewhat important

❶ ② ③ ④ ⑤ - Not at all important
① ❷ ③ ④ ⑤ - Somewhat unimportant
① ② ③ ④ ⑤ - Neutral
① ② ③ ❹ ⑤ - Somewhat important
① ② ③ ④ ❺ - Important

In assessing the usability of Helius, we employed the System Usability Scale, a reliable tool for measuring user satisfaction, which consists of 10 statements rated on a 5-point Likert scale. The overall SUS score for Helius was 71.75, categorizing it as "Good" according to the scale proposed by Brooke (1996) [4].

Table 6: Participant's perceptions of the Helius' recommendations - Questionnaire Study

Participants	Likert-item	Recommendations' importance
P5	①②③❹⑤	Somewhat important
P6	①②③❹⑤	Somewhat important
P7	①②③❹⑤	Somewhat important
P8	①②③④❺	Important
P9	①②③④❺	Important
P10	①②③④❺	Important
P11	①②❸④⑤	Neutral
P12	①②③❹⑤	Somewhat important
P13	①❷③④⑤	Somewhat unimportant
P14	①②③❹⑤	Somewhat important
P15	①②③❹⑤	Somewhat important
P16	①②③❹⑤	Somewhat important

❶②③④⑤ - Not at all important
①❷③④⑤ - Somewhat unimportant
①②❸④⑤ - Neutral
①②③❹⑤ - Somewhat important
①②③④❺ - Important

Figure 6 illustrates these results and confirms the study's conclusiveness with a significant participant number. Despite the positive usability feedback, participants suggested several enhancements to improve Helius's interface and functionality, such as better interface flows, availability of downloadable content, and tailored user levels to cater to varying expertise levels in DT.

Potential risks to validity include our interpretation of the feedback collected in the validation activities (Iterations 1 to 3). To mitigate bias, we held discussions among authors and collected data from diverse professionals in each iteration.

4 SIMILAR TOOLS

This section presents a brief comparision of Helius to other related tools for supporting the selection of DT techniques in software development. We collected the similar tools through a systematic mapping study in Literature [12] (see Table 7).

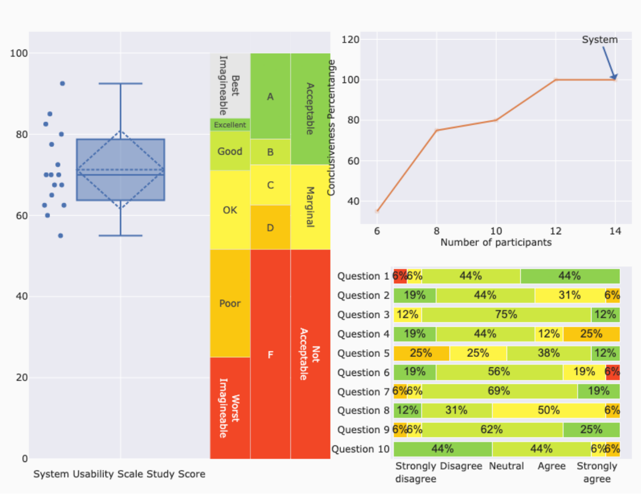


Figure 6: SUS score of Helius calculated using the tool [3]

Table 7: Tools comparison

Feature	DTA4R	IDEO DT	DT@IT	Helius
Creation of DT techniques projects, allowing the user to require DT techniques recommendations	◗	◗	⊘	●
Review of the DT techniques used in software development	⊘	⊘	⊘	●
Access to a community of practice	⊘	⊘	⊘	●
Access to a DT techniques combination graph	⊘	⊘	⊘	●
Access to a catalog of DT techniques	●	●	●	●
Follow other professionals who used DT techniques to discover their experiences with DT techniques	⊘	⊘	⊘	●

● – Feature implemented
◗ – Feature partially implemented
⊘ – Feature not implemented

DTA4RE[8] (*Design Thinking Assistant for Requirements Engineering*) is a DT technique selection assistant that provides a questionnaire related to project context and technique application needs, listing appropriate DT techniques based on user responses. IDEO DT[9] is a web catalog of DT techniques where users can access various techniques, learn about their details such as required time, necessary materials, and participants, and filter them by workspace usage. DT@IT[10] offers a catalog with 12 DT techniques, providing templates for their application.

Unlike these tools, Helius differentiates itself by leveraging collaborative user experiences as a feedback mechanism for DT technique recommendations. This means that recommendations, being collaborative and personalized, can vary for each user or change based on the evaluations of the experiences by other users.

5 FINAL REMARKS

This paper presented Helius, a recommendation system for DT techniques in software development. Our objective is to assist DT practitioners in selecting DT techniques by leveraging the experiences of other professionals, addressing the complex decision-making challenge involved in choosing appropriate DT techniques. In future research, we plan to conduct further evaluations of Helius to assess its effectiveness in enhancing the use of DT as a user-centered design approach within software development.

ARTIFACTS AVAILABILITY

Helius Tool source code: https://doi.org/10.5281/zenodo.12727205

ACKNOWLEDGMENTS

We thank PUCRS BPA 2019, 2020 and 2021 projects (Programa de Bolsas Pesquisa Alunos da PUCRS/Chamada Geral 1/2019, 1/2020 and 1/2021) and CNPq PIBIC 2019/2020 project (Programa Inst. de Bolsas de IC do CNPq) for the undergraduate research assistant scholarships. Tayana Conte thanks CNPq (Grant 314797/2023-8) and Sabrina Marczak thanks CNPq (Grant 313181/2021-7). Rafael Parizi thanks CNPq (Grant 200372/2022-0) and CAPES for the financial support (Code 001).

[8] https://sites.google.com/site/dta4re/
[9] https://www.designkit.org/
[10] https://espspinix.github.io/dt-at-it-toolbox/

REFERENCES

[1] Gerald Albaum. 1997. The Likert Scale Revisited. *Market Research Society Journal* 39, 2 (Feb 1997), 1–21.

[2] Victor R Basili. 1994. Goal Question Metric Paradigm. *Encyclopedia of Software Engineering* 1 (Jan 1994), 528–532.

[3] Jonas Blattgerste, Jan Behrends, and Thies Pfeiffer. 2022. A Web-Based Analysis Toolkit for the System Usability Scale. https://github.com/jblattgerste/sus-analysis-toolkit

[4] John Brooke. 1996. SUS-A Quick and Dirty Usability Scale. *Usability Evaluation in Industry* 189, 194 (Jan 1996), 4–7.

[5] Design Council. 1944. The Design Process: What is the Double Diamond? designcouncil.org.uk/

[6] Fred D. Davis. 1989. Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology. *MIS Quarterly* 13, 3 (Sep 1989), 319–340.

[7] Barbara Ann Kitchenham. 1996. Evaluating Software Engineering Methods and Tool part 1. *ACM SIGSOFT Software Engineering Notes* 21, 1 (jan 1996), 11–14.

[8] Hugo Ferreira Martins, Antônio Carvalho de Oliveira Junior, Edna Dias Canedo, Ricardo Dias Ajax Kosloski, Roberto A. Paldês, and Edgard Costa Oliveira. 2019. Design Thinking: Challenges for Software Requirements Elicitation. *Information* 10, 12 (Nov 2019), 27.

[9] Rafael Parizi, Igor Couto, Sabrina dos Santos Marczak, Lucas Hanauer, and Tayana Conte. 2021. On a Recommendation System of Design Thinking Techniques for Software Development based on Professionals? Collaboration. In *Proceedings of the Workshop on Requirements Engineering*. PUC-RIO, Virtual, 10.

[10] Rafael Parizi, Marina Moreira, Igor Couto, Sabrina Marczak, and Tayana Conte. 2020. A Design Thinking Techniques Recommendation Tool: An Initial and On-going Proposal. In *Proceedings of the Brazilian Symposium of Software Quality* (São Luiz do Maranhão). SBC, 355–360.

[11] Rafael Parizi, Marina Moreira da Silva, Igor de Souza Couto, Kendra Pavin Trindade, Matheus Plautz, Sabrina Marczak, Tayana Conte, and Heloisa Candello. 2020. 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*. Curran Associates, Curitiba, Brazil, 14.

[12] Rafael Parizi, Matheus Prestes, Sabrina Marczak, and Tayana Conte. 2022. How has Design Thinking being Used and Integrated into Software Development Activities? A Systematic Mapping. *Journal of Systems and Software* 187 (5 2022), 1–27.

[13] J. Pereira and R.F.S.M. Russo. 2018. Design Thinking Integrated in Agile Software Development: A Systematic Literature Review. *Computer Science* 138, 1 (Oct 2018), 775–782.

[14] Lauriane Pereira, Rafael Parizi, Matheus Prestes, Sabrina Marczak, and Tayana Conte. 2021. 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*. ACM, Virtual Event, Republic of Korea, 1338–1345.

[15] Matheus Prestes, Rafael Parizi, Sabrina Marczak, and Tayana Conte. 2020. On the Use of Design Thinking: A Survey of the Brazilian Agile Software Development Community. In *Proceedings of the International Conference on Agile Software Development*. Springer, Copenhagen, Denmark, 73–86.

[16] Per Runeson, Emelie Engström, and Margaret-Anne Storey. 2020. *The Design Science Paradigm as a Frame for Empirical Software Engineering* (1 ed.). Vol. 1. Springer, Chapter 5, 127–147.

[17] Anselm Strauss and Juliet Corbin. 1967. *Discovery of Grounded Theory*. Vol. 1. Aldine Transaction.

[18] Guilherme Horta Travassos, Dmytro Gurov, and EAGG Amaral. 2002. *Introdução à Engenharia de Software Experimental* (1 ed.). Vol. 9. UFRJ.

[19] Claes Wohlin, Per Runeson, Martin Höst, Magnus C Ohlsson, Björn Regnell, and Anders Wesslén. 2012. *Experimentation in Software Engineering* (1 ed.). 1, Vol. 1. Springer.