Design Thinking in Software Requirements: What Techniques to Use? A Proposal for a Recommendation Tool

Rafael Parizi[1], Marina Moreira da Silva[1], Igor Couto[1], Kendra Trindade[1], Matheus Plautz[1], Sabrina Marczak[1], Tayana Conte[2], Heloisa Candello[3]

[1] MunDDoS Research Group – School of Technology
Pontifícia Universidade do Rio Grande do Sul (PUCRS)
Porto Alegre, Brazil
[2] USES Research Group – Institute of Computing (IComp)
Universidade Federal do Amazonas (UFAM)
Manaus, Brazil
[3] Conversation Intelligence group – IBM Research Lab
São Paulo, Brazil
{rafael.parizi,igor.couto}@edu.pucrs.br,
{marina.moreira,matheuz.plautz}@acad.pucrs.br,
kendrapavin@hotmail.com, sabrina.marczak@pucrs.br,
tayana@icomp.ufam.edu.br, heloisacandello@br.ibm.com

Abstract. Design Thinking (DT) has been incorporated into software processes by teams from startups to large companies, used in problem exploration fostering innovative solutions, and integrated into agile methods to satisfy the real stakeholders' needs. DT places the customer needs up-front and helps to empathize with users, examining their behaviors, and producing outcomes focused on users' demand. A set of techniques, like personas and user journey, can be used to support this human-centered approach. Selecting which technique to use might be challenging since factors such as the application scenario, stakeholder engagement level and previous knowledge of the problem-to-be-solved may vary from case to case. Therefore, this paper presents an ongoing research that aims to develop a collaborative tool with the purpose of providing recommendations about potential DT techniques to be used in the support of requirements engineering activities. Our tool proposal was developed from results of a DT session that identified the collaborative tool as a proposal solution, a requirements elicitation activity to define the tool scope, and an interview-based early experimental study with professionals that use DT in industry. Our preliminary results show that our proposal has the potential to contribute to professionals that apply DT in requirements engineering by suggesting the most suitable techniques according to the established context.

Keywords: Design Thinking · Technique · Recommendation System · Collaborative Tool · Experimental Study.

1 Introduction

Design Thinking is a problem-solving approach that brings human-centered design principles at its core [3]. Driven by the search for innovation, it fits the development of new products, services, or processes, from startups to large and complex environments [2], thus being used by software development teams.

As a human-centric approach [14], DT can be used in the early phases of software development process to understand and identify what the customer needs are, providing better support for downstream development activities, mainly those related to identifying a proper solution for the problem-at-hand. The multidisciplinary view offers software teams the voice for dynamic, and driven problem-based collaboration; bringing all involved parties closer together [18].

By bringing the user needs to the center of the discussion, DT also improves team communication and facilitates knowledge domain acquisition, which are well-known issues in software development [11]. Given its interactive and dynamic nature, DT is also considered an easy-in integration with and a way to boost agile development [17]. Despite the use, navigating in this new world might be challenging. Literature offers a plethora of DT tools and methods (or techniques for simplification) that form the toolkit to perform DT activities themselves. Nevertheless, there is a lack of studies mentioning 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.

Therefore, considering the DT leans and how it can support the discovering of user needs and scoping of a solution [5], well-known activities of the Elicitation phase of the Requirements Engineering discipline, our long-term research goal is to support software development professionals' decision in the selection of which DT techniques to use in a certain development scenario. To achieve our goal, we adopted a long-term multi-phase research process, as follows: i) a literature review [16] to inform on the need to provide support on the DT technique selection, ii) a survey to confirm such need from industry professionals[1], and iii) a tool proposal to initially support such decision process and to later establish a community of users that can retro-feed the efficiency of the recommendations. This paper reports on our third phase–the tool definition.

We present a DT session conducted to better understand the need of a recommendation tool, a requirements elicitation activity in which we defined how the tool should work, and an early evaluation of the initial tool low-fidelity prototypes. Results show that the recommendation tool should take into account the context in which the software product will be used and project characteristics (e.g., team expertise), feedback from others using the recommendation, and

[1] We learned in this survey with 158 participants that 1/3 of them had a difficulty of 6 or up points (scale 1 to 10) in selecting a technique. We also learned that 83% of them chose techniques based on the product context, 81% on their previous experience (which includes learning from others), and 67% on the fitness of a certain technique to a certain DT working space phase (e.g, interviews to support discovering). These results support the need for a recommendation tool. Details can be found in [15]

results from the tool usage itself (e.g., the most selected techniques); and that those features would be welcomed by practitioners.

2 Background

DT is a structured approach to problem-solving by exploring the needs of users and transforming an idea into an acceptable solution via a prototype. Hiremath and Sathiyam [8] argue that DT is increasingly used in software development companies as a tool for innovation. It offers iterative learning from the beginning of the development cycle, including continuous improvement [6].

Requirements Engineering (RE) literature has also increased its interest on the topic. For instance, Hehn and colleagues' work on approaches to tailoring and integrating DT and RE [5], and Hehn, Uebernickel, and Fernandez's [7] integration of DT with software engineering processes aiming to improve the quality of requirements are examples of how RE can benefit from DT.

The approach is inherently multidisciplinary, involving professionals and participants from different domains and knowledge areas. It also uses empathy as the underlying concept to seek for what is technologically appropriate and strategically feasible to be built when proposing a solution [2].

Moreover, Brenner and Uebernickel [1] consider that DT can be conceptualized in the light of three perspectives: as a process, as a mindset, and as a toolbox. The first perspective defends that DT is a set of distinct and iterative steps (e.g., empathize, define, ideate, prototype, test) that abstract working spaces to explore the problem, propose a solution and validate it, which can be adapted and executed non-sequentially according to the necessity. Literature (e.g., [4,12,13,21]) present different process models to abstract these working spaces. The second perspective argues that DT as a mindset inspires a human-centered view, placing the user as the center of attention, seeking to understand what the problem is, among others. Finally, the third perspective supports DT as a 'toolbox', i.e., a set of techniques (e.g., persona, user journey) that support the achievement of the planned process steps. It acts as an 'operational layer' that makes it possible to obtain a solution to a problem. Our research assumes this third view as the focus of interest.

Despite the large number of studies reporting on the use of DT in software engineering [19], we still have no consolidated knowledge on how developers choose DT techniques, what criteria they consider for supporting such selection, and what sources they consult for their decisions. Initiatives such as the Luma Institute's Innovating for People Toolkit [13]–which explains the meaning of each technique and suggests related techniques for a certain purpose–, or the IDEO Toolkit [9]–which recommends techniques based on a predefined set of questions that a professional might ask about how to use DT–, are examples that demonstrate that choosing a technique is not trivial. Although highly used by practitioners, both cases fail to consider the context in which DT is being used. For instance, if the team has no previous information about the stakeholders, selecting interviews might be of more interest to learn about their profile rather than a focus group session in which people might not get along and waste effort. DTA4RE - Design Thinking Assistant for Requirements Elicitation [20] is a first step towards aiming to tailor the recommendation process. However, DTA4RE is still limited in the number of contextual factors it considers to recommend a technique and on not taking into account any kind of feedback from those who use the recommendations. Our research contributes towards filling in this gap.

3 Research Methodology

Inspired on the results of our preliminary literature review and survey, as cited in Section 1, we posed the following research question: *How can software development professionals select the most suitable Design Thinking (set of) technique to support requirements engineering*?

Figure 1 illustrates our 3-steps research methodology to answer the posed research question. First, to understand and define the problem of how to select the suitable DT techniques for supporting requirement engineering, we performed a "meta-DT" session (Step 1). We named it "meta-DT" since we conducted a 3 hours-long Design Thinking session to better understand the need to propose a recommendation system to support the selection process of DT techniques and to define how the tool should work. A total of 10 people participated in the activity–5 of them from industry and with an average of 3 years of experience working with DT, and 5 graduate students with DT as a research topic and 2 of them also with industry experience.

The session was conducted by a requirements analyst with 2.5 years of experience in DT and organized into the following activities: i) problem understanding based on the presentation of the working question; ii) problem definition which aimed to further discuss the need for a tool; iii) ideation using the Brainwriting DT technique to generate ideas; iv) convergence activity using the Affinity Diagram DT technique to find similar features; iv) prototyping the solutions (paper-based prototypes defined by two mixed groups of 5 members); and v) choice of solutions and presentation by each one of the groups through a voting activity. As a result of this session, the participants concluded that the solution would be the development of a recommendation tool associated with a community building environment through feedback, and in the future, rely on gamification mechanisms to engage tool users and promote rich feedback to maintain the community.

Step 2 focused on eliciting requirements for the elected recommendation tool solution. Carried out by the 5 graduate students, the elicitation activity step was also inspired on the use of DT techniques. Upon a 1.5 hour-long session in which each of the students suggested the use of a set of DT techniques, by unanimous decision we selected i) User Journeys[2] to represent the needed steps to a user to achieve their (business) goals, ii) Service Blueprints[3] to describe how the 'service' (tool features) may be offered to the user (touch-points), covering the

[2] User Journeys: http://bit.do/DTA4RE-user-journeys
[3] Service Blueprint: http://bit.do/DTA4RE-serviceblueprints

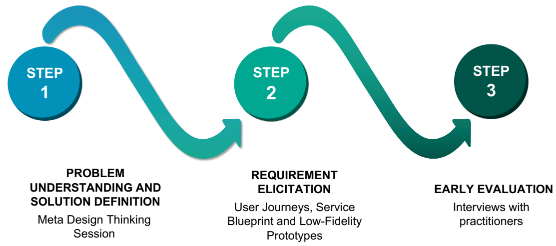


Fig. 1. Our step-wise research methodology

entire journey, identifying points for improvement and business opportunities, and iii) Low-Fidelity Prototypes[4] to define how users will interact with the tool. A sample of those are presented in Section 4.2[5]. Specifications were defined by 3 of the students and reviewed by the 2 others until a consensus was reached.

Finally, Step 3 aimed for an early evaluation of our tool specification with industry practitioners as a means to identify whether we were missing any relevant feature. We interviewed 5 practitioners (a product designer, a service designer, a product owner, a business analyst, and an IT manager) of 2 multinational IT companies, a large Brazilian TV broadcasting company, a cooperative bank and an IT provider with an average of 3.5 years of experience with DT in software development. We first openly asked them how they select the techniques they use to later present them with our recommendation tool idea and a sample of printouts of low-fidelity prototypes to discuss in details. Four of the interviews took place during coffee breaks of the DT track of an industry-based developers' conference and another one over the phone. They lasted an average of 20 min.

4 DT Techniques Recommendation Tool

This section details the activities performed to define and specify the DT techniques recommendation tool. It also presents the resulting specification, i.e., the tool proposal, and its early evaluation.

4.1 Step 1. Meta-Design Thinking Session

Our meta-DT session was guided by the the following question: *In what ways would we as Requirements Analysts/Software Engineers/Designers be able to choose the techniques to succeed in using DT to support software requirements?*

In a two-minutes time slot and using post-its, each participant described what she thought was important to provide relevant information to interested parties about the established problem. This ideation activity resulted in 24 proposed insight cards to the established problem. Then, to gain a deeper understanding

[4] Prototypes: http://bit.do/DTA4RE-prototypes

[5] The full report can be found at http://bit.do/CIBSE2020DesignThinking

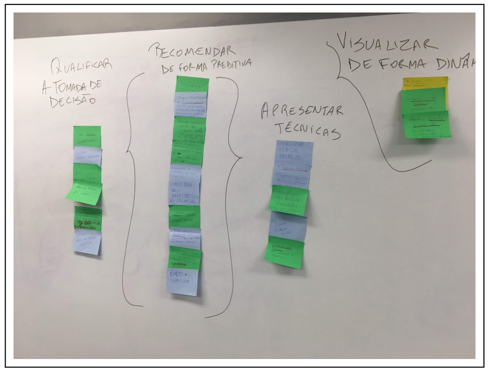


Fig. 2. Affinity Diagram identifying the DT tool macro features

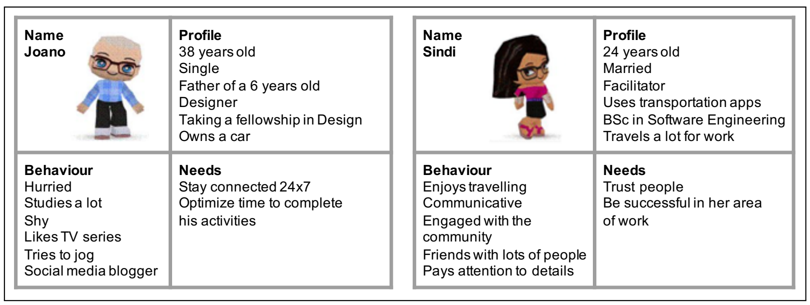


Fig. 3. Personas Sindi and Joano

of this scenario, an Affinity Diagram was organized (see Figure 2). This diagram allows the organization and grouping of results (insight cards) according to their similarities, dependencies, or affinities, generating a diagram that results in macro areas or features that delimit the addressed theme [22].

The four resulting identified macro features (in Portuguese, from left to right) are: i) qualified decision making; ii) prediction-based recommendation; iii) techniques presentation, and; iv) dynamic visualization of recommendations.

Next, the participants were separated into two mixed groups composed each of practitioners and graduate students to carry out a prototyping activity. These groups are named Group 1 and Group 2 from now on. The moderator introduced two personas, Joano and Sindi, presented in Figure 3 and previously created for the session. Each persona was assigned to each of the groups to work on. Group 1 was assigned to the Joano persona and Group 2 to Sindi.

Through prototyping, for about 30 minutes, each participant individually designed a first round of paper-based low-fidelity prototypes to address the 4 identified macro features. Figure 4 shows the prototypes related to the macro feature iii) present the techniques. Subsequently, each group was instructed to review the designed prototypes, group them by features and vote by posting

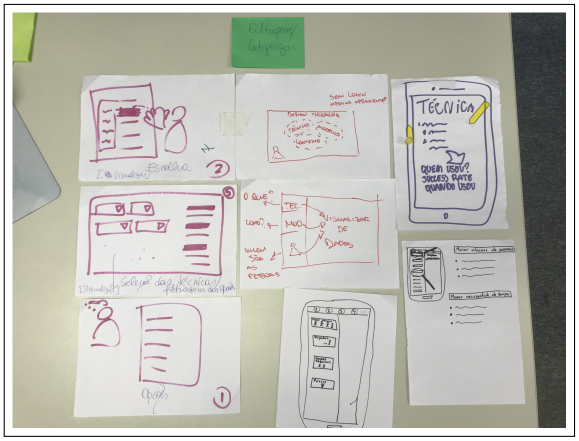


Fig. 4. First round low-fidelity prototypes for the macro feature iii)

Table 1. Identified features per persona grouped by macro feature

Macro Feature	**Joano Persona (Group 1)**	V_1	**Sindi Persona (Group 2)**	V_2
i) Qualified decision making	- Organize Techniques by Category (7a) - Qualify Decision Making (7a, 7b, 7d)	0	- Suggest Complementary Technique (7a) - Organize Techniques by Category (7a) - Qualify Decision Making (7a, 7b, 7d)	4
ii) Prediction-based recommendation	- Get to know similar cases (7b, 7c, 7d) - Consider the context of use of DT (7a, 7c) - Provide feedback (7d)	1	- Suggest tools predictively (7a) - Receive examples of the application of the technique (7b, 7c, 7d) - Consider the context of using DT (7a, 7c) - Provide feedback (7d)	2
iii) Techniques presentation	-Visualize techniques and models (7a, 7b, 7c) - Search for technique recommendations in a systematic way (7a, 7c)	3	- Visualize the techniques and models (7a, 7b, 7c)	0
iv) Dynamic visualization	- Simulate the use of combined techniques for a particular purpose (7c)	6	- Dynamic and interactive visualization (7a, 7b)	2

a yellow post-it on the grouped set of prototypes by macro feature that most represented what the tool should offer.

Table 1 summarizes the identified features per persona grouped by macro feature and the respective amount of votes attributed to them (columns (V_1) and (V_2)). The voting aimed to highlight whether certain macro features stood out and deserved priority attention or a deeper discussion. Each participant could choose between one to three features. Features with zero voting indicate that none of the participants considered it a priority. The groups differ on what they consider to be most relevant. While Group 1 considered that the visualization of techniques was the most relevant macro feature (6 out of 10 votes), Group 2 was torn among 3 of the macro features, with no predominant decision. Given the distinct persona behaviors and needs this divergent scenario was not considered an issue. The table also maps the prototyped features in round 2 as described in Section 4.2 (e.g., (7a)) to the discussed features in the meta-DT session (prototype round 1) as indicated in the referred table (e.g., Feature i)).

To conclude, each group presented their results to one another and explained their voting and motivations behind it. This discussion promoted common ground among the meta-DT session participants and made possible to the group to conclude that they have reached a tool vision.

4.2 Step 2. Requirements Elicitation

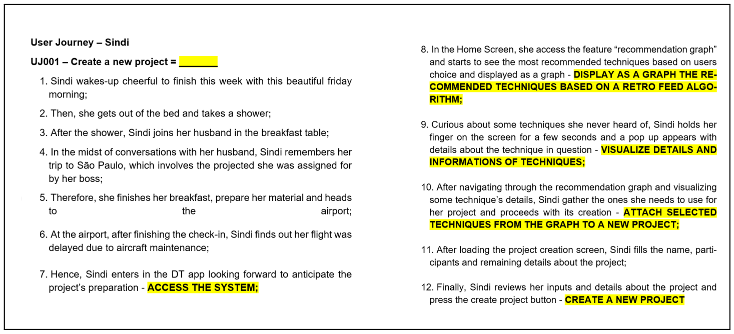
The second step includes User Journeys, Service Blueprints and Prototyping of the screen's flow. This paper presents a sample of the identified features.

User Journeys: Proceeding with the requirements gathering exercise for the tool proposed in this paper, we started with the definition of 5 User Journeys. Figure 5 shows a User Journey developed to Persona Sindi, highlighting her main actions when looking for a technique and aiming to use it in her product development. The identified functional requirements derived from this user journey are highlighted in Figure 5, described next:

- Access the system: user access the tool;
- Recommend techniques based on a self-updated visualization graph: user can receive technique recommendations and self-update the tool, collaborating with other users;
- View techniques details: user can access detailed information about a certain technique, like when, how, and why to use it;
- Attach/Add techniques from the visualization graph to a new project: user have the opportunity to choose a technique and associate it to a new project.
- Create a new project: user can create a project and manage the used techniques and previous experiences, updating the visualization graph.

Service Blueprints: As mentioned, Service Blueprints (SB's) are used to visually present the detailed specification of aspects of a service (business feature), from the user's perspective, and other relevant parts that may be involved.

Figure 6 presents a SB for the persona Sindi, related to the User Journey showed by Figure 5, aiming to get new insights and a deeper understanding of the problem. In this service blueprint, the user initially accesses system (A) and selects from the suggestion graph some technique for his DT project (B). The user can also see information about the selected technique (C) as well as linking to the running project (D). This figure also presents other expected functionalities through the relationships shown in each of the horizontal lanes physical evidence (i), customer actions (ii), contact with backstage actions (iii), and support processes (iv).


User Journey – Sindi

UJ001 – Create a new project = ________

1. Sindi wakes-up cheerful to finish this week with this beautiful friday morning;
2. Then, she gets out of the bed and takes a shower;
3. After the shower, Sindi joins her husband in the breakfast table;
4. In the midst of conversations with her husband, Sindi remembers her trip to São Paulo, which involves the projected she was assigned for by her boss;
5. Therefore, she finishes her breakfast, prepare her material and heads to the airport;
6. At the airport, after finishing the check-in, Sindi finds out her flight was delayed due to aircraft maintenance;
7. Hence, Sindi enters in the DT app looking forward to anticipate the project's preparation - **ACCESS THE SYSTEM;**
8. In the Home Screen, she access the feature "recommendation graph" and starts to see the most recommended techniques based on users choice and displayed as a graph - **DISPLAY AS A GRAPH THE RECOMMENDED TECHNIQUES BASED ON A RETRO FEED ALGORITHM;**
9. Curious about some techniques she never heard of, Sindi holds her finger on the screen for a few seconds and a pop up appears with details about the technique in question - **VISUALIZE DETAILS AND INFORMATIONS OF TECHNIQUES;**
10. After navigating through the recommendation graph and visualizing some technique's details, Sindi gather the ones she needs to use for her project and proceeds with its creation - **ATTACH SELECTED TECHNIQUES FROM THE GRAPH TO A NEW PROJECT;**
11. After loading the project creation screen, Sindi fills the name, participants and remaining details about the project;
12. Finally, Sindi reviews her inputs and details about the project and press the create project button - **CREATE A NEW PROJECT**

Fig. 5. User Journey – View techniques in detail – Persona Sindi

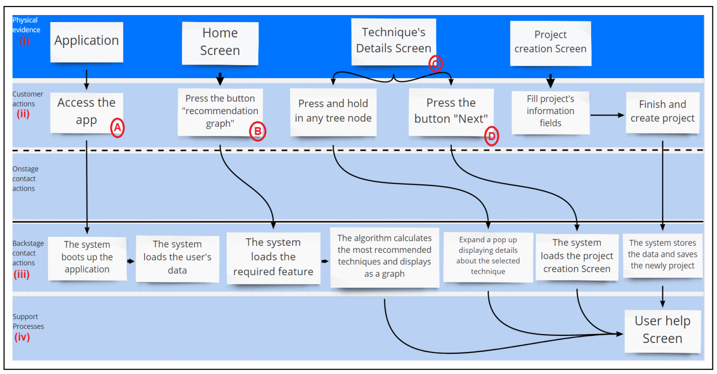
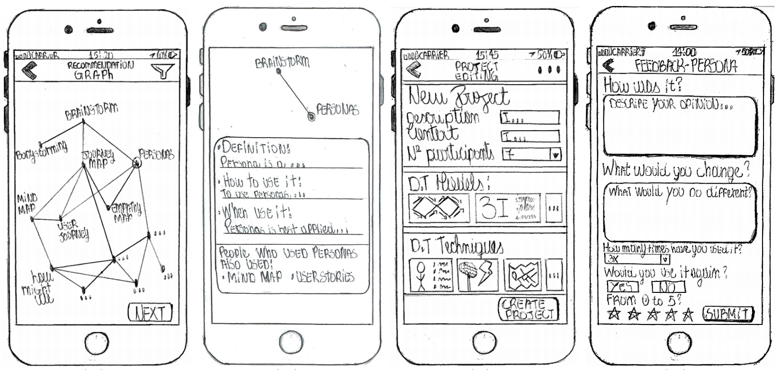


Fig. 6. Service Blueprint - Persona Sindi

Low-Fidelity Prototypes: Considering the insights gathered from the Users Journeys and the Service Blueprints, we further detailed the features through their prototypes. We designed a second round of paper-based prototypes mapping the results from the two referred elicitation techniques to consolidate our understanding as presented here.

Starting from the Techniques Recommendation Graph Screen (Figure 7-(a)), which represents one of the core value of our application, it shows the techniques within a graph according to a specific filter. Filters represent product context (e.g., scenario of use, user profile) and project characteristics (e.g., problem domain, knowledge of the stakeholders, number of participants on DT session, among others). These will be further and better defined later on as part of our future steps. By default, the graph is set to generate the vertices and edges according to the most used technique. Therefore, the starting and central node from this graph presents the most recommended technique, followed by the next most recommended nodes that relate to the first (e.g., Brainstorming followed by Journey Map and Personas). Thus, our tool presents a network of interconnected DT techniques. Yet, this Techniques Recommendation Graph should also consider other aspects, such as (i) DT models and set of techniques previously



(a) (b) (c) (d)

Fig. 7. (a) Graph; (b) Technique detail; (c) Project creation; (d) Feedback

applied by the software professionals community; (ii) feedback from the users about the experience of the use of a technique or a DT model; (iii) experience from the combination of different techniques in specific scenarios of requirements engineering. These are part of our long-term research view and will be discussed only once the first tool version is made available.

In addition, if a user is not familiarized with any of the techniques, each of it contains descriptive information, e.g.: definition, case scenarios to use, and others (see Figure 7-(b)). Exactly as a haptic, when the selected technique is selected, its respective data will pop up inside a little rounded square, delivering the users dexterity and continuous use of the feature, despite having to go back and search manually about the given technique.

Figure 7 illustrates two other features. Figure 7-(c) provides the concept of a project to store related information held together, whereas having to seek them separately all the time. Therefore, the screen exhibits blank fields to be filled about the project's crucial details (e.g.: name, description, purpose and participants), and a set of available DT models to fit in their project and also a list of the previous selected techniques with the graph. This is the foundation for future use as part of the recommendation algorithm previously mentioned.

Finally, Figure 7-(d) illustrates a screen for capturing user feedback from the use of DT techniques in their software development projects, chosen through the recommendations made by the tool we are proposing in our work. In this screen, the user can inform the result of the application of a specific technique (e.g., Persona), filling in fields like what was the experience of use, if it suggested any alteration or if it was done alternatively, how many times already used, would be used again, and how to rate it (on a level of stars ranging from 1 (not suitable) to 5 (very appropriate)).

This last screen represents another core of our tool: exchange of information and experience among DT professionals, creating an effective and consolidated communication channel and establishing a community environment, especially those who apply DT in software engineering. Thus, our proposal is not limited to an information guide but defines it as an collaborative environment that enables the exchange of experiences between DT users. This feature should also be further modeled ahead as per the use of the first tool draft to be made available.

4.3 Step 3. Early Tool Evaluation

To gather the first impressions and to align our research goals in the context of this recommendation tool, we interviewed five industry practitioners. Here, to maintain the confidentiality of both the practitioners and their companies, we identify them as P1, P2, P3, P4, and P5. We asked: i) how do you select DT techniques? ii) how do you deal with changes in the technique selection during a DT session, if any?, and iii) how would you welcome a tool that recommend you techniques considering your product context and feedback from others?

Regarding the question (i) choice of DT techniques, the participants reported that they select considering the client's knowledge and feeling, according to P2; that they learn to choose from the experience gained from years of application, and that ends up creating a particular set of techniques by the results that have already been obtained in previous applications, as mentioned by P3. In addition, the techniques can be chosen and determined according to the DT working space, either for understanding the problem, for ideation or the construction of the solution. P4 unveiled it:

- *"One learn to choose techniques over time because each technique has a result. If one wants to understand the problem, one uses certain techniques. One defines ones' own toolkit overtime. Also, if one has to think about solutions, one has these (certain techniques) here that help one get there"*–P4.

On the need to looking for other techniques while conducting a DT session (question ii), interviewees responded that situations arise when users/participants are not engaged or do not understand how the technique works and need to change or adapt the technique for more meaningful results. This situation is illustrated by the answers of P3 and P4:

- *"There are situations in DT Workshops that the use of some techniques does not work, so the moderator/designer must choose another technique from their experience. There are also instances when participants find it challenging to use a particular technique, so one needs to use others".*–P3
- *"It happens that one has to change in the middle of a workshop because the customer does not respond well to a certain activity. In this situation, empathy must be used to understand what is happening and get around the situation. Changing the technique is often helpful in such situation."*–P4

Still, P5 and P2 suggest making combinations between different techniques, observing that with the applied variations, going beyond the pre-established models, one gets a more efficient result.

- "*Experiment variations between techniques (a mix of techniques), because this ensures the bias of Designer and not just replicating methods [...]*"–P5
- "*Ah, you test if this technique combines this one.*"–P2

Regarding the need for a recommendation tool (question iii), the interviews showed that it will be useful and of great help to professionals who use DT. They suggested that the DT user might input some data such as: *"Do you already have the problem defined?"* to know if the person (client) already has the scope of the problem defined; *"Do you know users?"* – then suggest a particular set of tools.

For P1, the tool should be geared to the DT community by fostering information exchange (feedback, more effective techniques); should have explanations of concepts, such as whether the solution to the problem is developing an mobile app, providing useful MVP tips; and should clearly present the techniques to understanding the problem, for ideation, and for creating the solution. Also, it should establish a sense of community, with a forum for users to interact among themselves, going beyond a simple guide, and provide metrics for evaluation (number of users who accessed in a period, user satisfaction when using).

5 Discussion

The tool proposed in this paper seeks to recommend Design Thinking techniques to users who are integrating them into requirements engineering activities. To discuss our tool proposal, which results from answering our posed research question–*How can software development professionals select the most suitable Design Thinking (set of) technique to support requirement engineering activities?*, we list some relevant takeaways from this study:

- The tool seeks to provide relevant information on DT techniques, going beyond the existing user's toolkit. The tool must allow the selection of other similar techniques through its recommendation system, using a technique relationship graph that should take into account a set of items for the recommendation such as previous use of a certain technique, users feedback, product context, and project characteristics.
- The tool should be valuable and able to assist both on-boarding novice users to expert ones during their DT sessions in software development. Both profiles found the tool idea useful.
- We discussed an innovative solution presenting a recommendation tool associated with a community building environment through feedback. Thus, the industry practitioners might collaborate with their community fostering the improvement of DT in software development.

Finally, as we show in each of the three stages of tool development, it is possible to see that there is a universe not yet explored, that is, we can still consider a lot of other features and technologies to apply in this tool, such as decision support methods, recommendation systems, Artificial Intelligence, multi-platform system, and others. Thus, we seek to ensure that the software industry, from those responsible for applying DT sessions, perceive in this tool a guide of recommendations that enables effective gains in the software development process.

6 Concluding Remarks and Perspectives

This paper proposes a tool for recommendations of Design Thinking Techniques in the context of software requirements. To further the understanding of the need to propose such a tool and identify a solution, we conducted a meta-DT session (Step 1) followed of a requirements specification activity (Step 2), and an interview-based early evaluation with industry practitioners (Step 3).

As a result, we identified four macro features to be present in our recommendation tool, namely: i) qualified decision making, ii) prediction-based recommendation, iii) techniques presentation, and iv) dynamic visualization of recommendations. These macro features were broke-down into smaller features and detailed in their use (User Journeys) and functionality (Service Blueprints). In addition, they were prototyped in a second round to refine their understanding and serve as guidance for the tool early evaluation with industry practitioners.

Results of the early evaluation with 5 professionals revealed that: i) previous experience is key to select techniques, ii) new techniques might be needed during a DT session, and iii) a recommendation tool could be useful to help with the decision process and addressing changes, mainly when considering context and historical use from others.

Although our early evaluation is not generalizable, it confirms that our tool vision –also limited in the sense it was defined–is adequate and serves its purpose. We aim to broaden this evaluation within the own use of the tool. We are currently developing a working prototype that should be soon be made available for experimental use by industry practitioners. We intend to use the tool to collect data on how people are selecting the techniques and, along side with the feedback and the project context features, will retro-feed the recommendation algorithm until its stabilization. Other items might also be taken into account for defining how the recommendation should work. For instance, the work by Ignacio [10] propose a set of characteristics for recommending a requirements elicitation technique. This will complete this first phase of the tool proposal before we move to the refinement of the community building to engage users, which will both compose the tool design second phase.

Acknowledgements

This study was partially financed by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001.

References

1. Brenner, W., Uebernickel, F.: Design Thinking for Innovation: Research and Practice. Springer International Publishing (2016)
2. Brown, T., Katz, B.: Change by design. Journal of Product Innovation Management (3), 381–383 (2011), https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1540-5885.2011.00806.x

3. Docherty, C.: Perspectives on design thinking for social innovation. Design Journal (6), 719–724 (2017)
4. Hasso-Plattner, I.: What is Design Thinking? (Accessed October 20, 2019), https://hpi-academy.de/en/design-thinking/what-is-design-thinking.html
5. Hehn, J., Mendez, D., Uebernickel, F., Brenner, W., Broy, M.: On integrating design thinking for a human-centered requirements engineering. IEEE Software pp. 25–31 (March-April, 2020)
6. Hehn, J., Uebernickel, F.: The use of design thinking for requirements engineering: An ongoing case study in the field of innovative software-intensive systems. In: Proc. of Int'l Requirements Eng. Conf. pp. 400–405. IEEE, Alberta, Canada (2018)
7. Hehn, J., Uebernickel, F., Fernandez, D.M.: Dt4re: Design thinking for requirements engineering: A tutorial on human-centered and structured requirements elicitation. In: Proceedings of International Requirements Engineering Conference. pp. 504–505. IEEE, Alberta, Canada (2018)
8. Hiremath, M., Sathiyam, V.: Fast train to dt: a practical guide to coach design thinking in software industry. In: Proceedings of Conference on Human-Computer Interaction. pp. 780–787. Springer, Las Vegas, USA (2013)
9. IDEO: IDEO Design Kit (Accessed Oct 20, 2019), https://designkit.org/methods
10. Ignacio, R.: Guia facetado de técnicas de elicitação de requisitos. Universidade Federal de Santa Catarina (11 2018)
11. Lindberg, T., Meinel, C., Wagner, R.: Design Thinking: A Fruitful Concept for IT Development?, pp. 3–18. Springer, Berlin, Heidelberg (2011)
12. Lucena, P., Braz, A., Chicoria, A., Tizzei, L.: Ibm design thinking software development framework. In: Proceedings of the Brazilian Workshop on Agile Methods. pp. 98–109. Springer, Curitiba, Brazil (2016)
13. Luma, I.: Innovating for people: Handbook of human-centered design methods. LUMA Institute, LLC (2012)
14. Plattner, H., Meinel, C., Leifer, L.: Design thinking: understand–improve–apply. Springer Science & Business Media (2010)
15. Plautz, M.: Estudo Exploratório sobre Design Thinking no Desenvolvimento de Software. Master's thesis, Escola Politécnica, Pontifícia Universidade Católica do Rio Grande do Sul, Porto Alegre, Brasil (3 2020)
16. Plautz, M., Marczak, S.: Design Thinking no Desenvolvimento de Software: Uma Revisão Sistemática da Literatura (2019) (accessed December 05, 2019)), http://bit.do/CIBSE2020DesignThinking
17. Przybilla, L., Schreieck, M., Klinker, K., Pflügler, C., Wiesche, M., Krcmar, H.: Combining design thinking and aigle development to master highly innovative it projects. Projektmanagement und Vorgehensmodelle 2018-Der Einfluss der Digitalisierung auf Projektmanagementmethoden und Entwicklungsprozesse (2018)
18. Seidel, V.P., Fixson, S.K.: Adopting design thinking in novice multidisciplinary teams: The application and limits of design methods and reflexive practices. Journal of Product Innovation Management (S1), 19–33 (2013)
19. Souza, A., Ferreira, B., Conte, T.: Aplicando design thinking em engenharia de software: um mapeamento sistemático. In: Proceedings of Ibero-American Conference on Software Engineering. Buenos Aires, Argentina (2017)
20. Souza, A.F., Ferreira, B., Valentim, N., Conte, T.: An experience report on teaching multiple design thinking techniques to software engineering students. In: Proc. of the Brazilian Symp. on Software Eng. pp. 220–229. ACM, São Carlos, Brazil (2018)
21. d.School Stanford: Design Thinking - Bootcamp Bootleg Compilation (Accessed October 20, 2019), https://dschool.stanford.edu/resources/the-bootcamp-bootleg
22. Vianna, M.: Design thinking: inovação em negócios. MJV Press (2012)