

# A Context-based Recommendation Model for Design Thinking Techniques Selection in Software Development\*

Rafael Parizi (Ph.D. Student)<sup>1</sup>, Sabrina Marczak (supervisor)<sup>1</sup>

<sup>1</sup>Graduate Program in Computer Science

Pontifical Catholic University of Rio Grande do Sul (PUCRS)

Av. Ipiranga, 6681 - Partenon, 90619-900 - Porto Alegre-RS, Brazil

Entry date: March/2019 – Thesis proposal: Expected to March, 2021

Final presentation: Expected to November, 2022

rafael.parizi@edu.pucrs.br, sabrina.marczak@pucrs.br

**Abstract.** *Design Thinking (DT) has been chosen as an approach to support problem-solving by many software development companies. However, there are divergences between the professionals of these companies concerning which techniques are performed, which steps are followed, and the way to implement this approach, as it proposes itself, to be divergent to generate numerous alternatives and, also, convergent, to find a solution. As a practical way to apply DT in software development, a collection of techniques are implemented during different working spaces provided by the DT models. Therefore, selecting the appropriate techniques within the set of the available ones is challenging. Aiming to present a solution for helping software development professionals in how to select the appropriate techniques for DT activities, we propose the development of a theoretical context-based recommendation model for selecting DT techniques for software development. Thus, we aim to develop a recommender and collaborative system of DT techniques and to validate it through industry-based empirical studies. We have already characterized the DT techniques that professionals are using and how they select them, and the benefits and challenges of using DT. These results were used to inform the proposal and the validation of a first version of the DT techniques recommendation tool, which presents the tool first round of core features.*

**Keywords.** Design Thinking, Techniques, Recommendation System, Theoretical Model, Empirical studies

CBSOft Event: SBES

---

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

## 1. Problem Characterization

Design Thinking (DT) is a user-centered design (UCD) approach used to foster innovative software seeking to fit the users' needs. Considering its iterative and creative characteristics, DT is integrated into agile methods to boost software development processes [Gurusamy et al. 2016]. It has been used, for instance, by companies such as IBM [Lucena et al. 2016] and SAP [Jensen et al. 2016].

Brenner and Uebernickel [Brenner and Uebernickel 2016] define DT as a mindset, a process, or a toolbox. As a mindset, DT fosters problem-solving thinking with the user at the center of the solution process. As a process, DT helps to orchestrate the necessary steps to diverge and converge to generate a solution [Luedeke et al. 2018, Berger 2011]. Finally, as a toolbox, DT provides more than 70 techniques to support the understanding of the problem, to ideate over various potential solutions, and to select the most suitable solution for solving the problem [IDEO 2020].

The collection of techniques that can be applied to support DT in software development helps IT professionals to produce better solutions. However, selecting which techniques to use among the set of available techniques is challenging. This selection needs to take into account the project context, the user-centric perspective, the multidisciplinary team's composition, among other aspects. Therefore, we ask *"How can a recommendation system promote the selection of DT techniques in software development?"*.

Aiming to help software development professionals in how to select the techniques for DT activities, this study aims at developing a recommendation and collaborative system of DT techniques to be used in software development. In addition, as a result of the use of the proposed recommender system, we expect to gather information with the professionals, through empirical studies, to create a theoretical context-based recommendation model for selecting DT techniques for software development. This model is an abstract representation of recommendations of DT techniques to be used to qualify the selection of the appropriate DT techniques according to project' context information, such as, for example, considering the experience on the use of DT, project domain, DT techniques characteristics, and involved stakeholders.

The reminder of this paper is organized as follows. Section 2 describes Design Thinking in software development. Section 3 presents the research methodology and the expected contributions of our work. Section 4 presents the current status of our work. Section 5 describes the evaluation of our early results. Finally, Section 6 compares our work with related ones.

## 2. Background

Putting the user at the center of the process is a strategy used by organizations of software development focusing on the improvement of the solution [Hehn et al. 2020]. User-centered Design approaches allow us to collect the users' needs fostering the innovation and the use of creativity to build a solution [Lindberg et al. 2011]. DT has been integrated with requirements engineering to support the development of solutions that understand and collect users' needs efficiently [Hehn et al. 2020, Nedeltcheva and Shoikova 2017].

When Design Thinking is used in conjunction with Agile methods, an environment prone to both understand the customers and their needs and to produce a solution

exploring constant deliveries, frequent feedback, quick reaction to changes is offered [Nedeltcheva and Shoikova 2017]. Approaches like DT@SCRUM [Vetterli et al. 2015] and DT@XP [Sohaib et al. 2019] are alternatives to integrating DT and Agile methods.

Another research topic is the use of DT techniques in software development. Recent efforts (e.g., [Souza et al. 2017, Hehn et al. 2020]) have aimed to characterize how DT is used in software development, including what and how DT techniques are used in, and how DT activities can be organized for creating an alignment with user's needs, respectively. In addition, our preliminary investigation of literature [Prestes 2020] and of industry practice [Prestes et al. 2020] has suggested that software development professionals still use a limited number of techniques either for not knowing about other options or for not realizing how to select new ones. Therefore, there is still room for recommending DT techniques using collaborative and intelligent mechanisms.

Our long-term goal is to create an abstract model of recommendations of DT techniques for software development. To reach this goal, we aim to develop a recommendation and collaborative system that provides recommendations of DT techniques focusing on software development. This system aims to create a DT users' community, capturing the feedback of the professionals to provide useful information on the selection and use of appropriate DT techniques. In addition, using empirical studies, we expect to gather information with professionals who have used our proposed system to establish an information base on the context of the use of DT Techniques and other relevant criteria to selecting them, still to be defined in our research.

### 3. Methodology and Expected Contributions

This research is classified as qualitative, based on empirical methods, seeking to explore, understand, and explain a certain phenomena [Dybå et al. 2011]. We organized our activities based on the use of empirical methods such as Surveys, Case Studies, and Focus Groups [Easterbrook et al. 2008]. Also, we have used a Systematic Literature Review (SLR) to perform a synthesis of relevant scientific studies about the research topic [Kitchenham et al. 2009]. Our goal is to gather information and to validate our proposals and artifacts with industry professionals who have experience in using DT in software development.

We organized our research methodology into 4 steps: (i) Exploration on Design Thinking in software development; (ii) Construction of a collaborative recommendation system of DT techniques in software development; (iii) Empirical validation of the DT techniques recommendation system, and; (iv) Construction of a theoretical context-based model for selecting DT techniques for software development.

Figure 1 illustrates the methodology of this research. In addition, aiming to describe our research methodology, Table 1 outlines our research steps, our research sub-steps, our research questions, and our research expected contributions.

### 4. Current Status and Future Work

This work seeks to create a theoretical context-based recommendation model for recommendations of DT techniques for software development. As a result, we expect to promote the use of appropriate DT techniques, and contribute to advancing the state-of-the-art and the state-of-practice about the use of DT in software development.

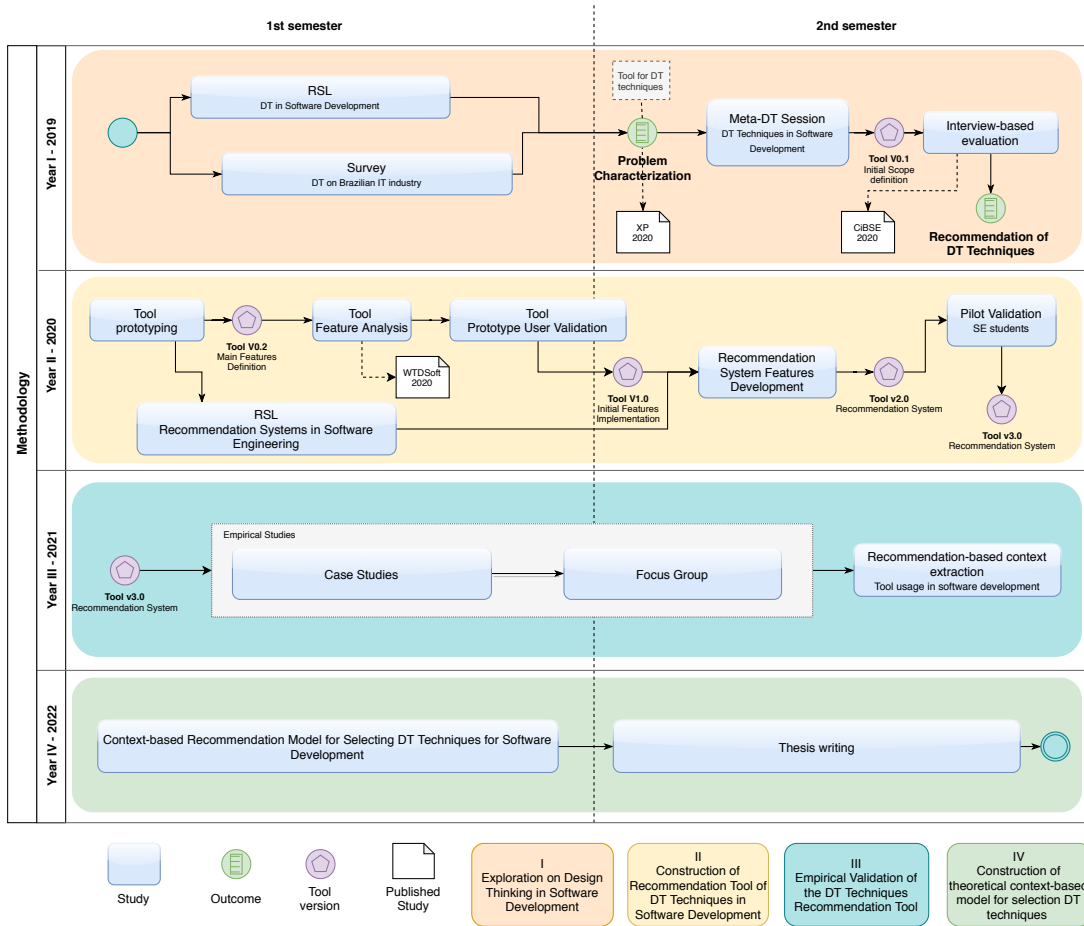


Figure 1. Methodology Steps

We are currently working on activities related to year II (Figure 1). The working-in-progress activities are based on the early outcomes we previously obtained (year I), such as (i) problem characterization: DT in software development and DT techniques in that context, and (ii) Recommendation of DT Techniques: proposal of a tool for DT Techniques recommendation for software development with interview-based validation (version 0.1 of the tool). In the first semester (YEAR II), we created the prototypes of the tool, and performed a tool feature analysis using DESMET method [Kitchenham et al. 1997]. These already obtained results are better described in Section 5.

We are working in the following activities (Figure 1, Year II, 2st semester):

- Tool Prototype user validation – User validation of the tool prototypes.
- RSL: Recommendation Systems in Software Engineering – characterization of how recommendation systems are implemented in SE.
- Tool development – Implementation of the tool using programming language.

In year III, our activities are focused on promoting the use of the tool with industry professionals. For this, we will carry out a case study and, subsequently, a focus group for data collection. After the data collection studies, our goal is to extract information from the context and the impact of the recommendations on the usage of the techniques (year IV). We intend to create a model of DT techniques according to the context of the project with the data collected.

**Table 1. Methodology, Steps, Research Questions, and Contributions**

Methodology				
#	Step	Studies	Research Question	Contributions
I	Exploration on Design Thinking in Software Development	Systematic Literature Review	How DT is used in software development by Industry Professionals?	To characterize how professionals are using DT in software development
		Survey		
		Meta-DT Session and Tool Proposal	How difficult is to choose DT techniques for Software Development?	To propose a tool for support the selection of DT Techniques
		Interview Evaluation		
II	Construction of Recommendation Tool of DT Techniques in Software Development	Systematic Literature Review: Recommendation Systems in Software Engineering	How are Recommendation Systems in Software Engineering implemented?	To understand how to build a recommendation system to help selecting DT Techniques in Software Development
		Recommendation Tool Prototyping and Development	Do recommendations for DT techniques improve the process of selecting techniques and implementing DT in software development?	To develop a recommendation system to help choose DT Techniques in Software Development
		Recommendation Tool Validation: Feature Analysis		
		Pilot Validation		
		Thesis proposal writing		
III	Empirical Validation of the DT Techniques Recommendation Tool	Case Studies in Industry	How software development professionals have used the Recommendations of DT Techniques?	To assess how DT Technique recommendations occur in the software development industry
		Focus Group		
		Recommendation-based context Extraction: tool usage in Software Development	What has been the experience of using the DT technique recommendation system for software development?	To extract usage information of the recommendation system aiming to build a model of DT techniques recommendations
IV	Construction of a theoretical context-based model for selecting DT techniques for software development	Model Construction	How could an abstract model assist in recommending DT techniques for software development?	To develop an abstract model of DT techniques recommendations to be used in software development
		Thesis writing		

## 5. Description and Evaluation of Results

This work started in 2019. In the first year, we performed the activities: (i) problem' characterization, and; (ii) a first solution proposal. We focused on understanding, knowing-

in-depth, and finding research gaps in the use of DT in SD. We based us on the research question: “*How DT is used in software development by industry professionals?*”.

Initially, for the (i) problem characterization, we conducted a systematic literature review [Kitchenham et al. 2009]. For the SLR, our main research question was: “*How Design Thinking is being used in software development, and what are the benefits and challenges of using it?*”. As a result, we have gathered information about the use of DT in SD, including a theoretical overview, a set of models composed by working spaces, a set of techniques and tools, establishing DT as a Mindset, as a process, or as a toolbox [Lindberg et al. 2011]. In addition, we identified the relationship of DT with software engineering, such as its application in software development in industry, for academic purposes, and for requirement engineering.

In parallel with the SLR and based on its results, still considering the (i) problem characterization, we conducted a survey to identify which models, techniques and tools IT professionals use, reasons for choosing DT, usage scenarios, and the benefits and difficulties of applying DT. Our survey was answered by 149 Brazilian IT professionals who were invited using the LinkedIn<sup>1</sup> professional-based social network. As a result, we characterized how DT is used in the Brazilian software development industry. We found that professionals have difficulties in selecting which techniques to apply in their activities, according to the multidisciplinary, innovative, creative and dynamic context fostered by DT and required in today’s SD processes. As an outcome of activity (i), we published a paper in the XP 2020 conference [Prestes et al. 2020].

After the problem characterization, we worked in a first solution proposal (ii). We performed a Meta-Design Thinking Session. We call this session as Meta because it was a DT session focused on a solution for DT. This meta-DT activity was started pointing out the problem: “*How can software development professionals select the most suitable Design Thinking (set of) technique to support software engineering?*”. As a solution, a recommendation system of DT techniques was proposed. It seeks to provide relevant information on DT techniques, going beyond the existing user’s toolkit. In addition, it allows the selection of other similar techniques, using a technique relationship graph that should consider a set of items for the recommendation such as previous use of a precise technique, user feedback, product context, and project characteristics. Through an interview-based study, we validated that the tool is valuable in assisting both on-boarding novice users and expert ones during their DT sessions in SD. Both IT professionals’ profiles found the tool idea useful and helpful. As an outcome for activity (ii), we published a paper in the CIbSE 2020 conference [Parizi et al. 2020].

Next, we prototyped the screens to define the tool’s features and flows. We used the Figma<sup>2</sup> and QuantUX<sup>3</sup> tools to support the refinement of user requirements and to validate them. Using Figma, we created high-level fidelity prototypes, and used these high-level fidelity prototypes to design and run our requirements validation study using Quant-UX. In addition, we compared our solution with the related tools using the feature analysis methodology proposed by Kitchenham [Kitchenham et al. 1997]. As a result, we validated the potential of the proposal for the recommendation of DT techniques.

---

<sup>1</sup><http://www.Linkedin.com/>

<sup>2</sup><http://www.figma.com>

<sup>3</sup><https://www.quant-ux.com>

**Table 2. Related Work**

Work	Classi	DT approach	Techniques Recommendation
DTA4RE	T	DT techniques recommendation for RE	Questionnaire-based recommendation using project context and professional experience
IDEO DT	G/T	DT techniques filtering for general purposes (including SD)	Form-based filtering of techniques for three working spaces: Inspiration, Ideation and Prototyping
DT@SCRUM	G	DT techniques explored in a Design Planning activity. It adapts Scrum's sprint planning sessions to DT activities	DT Techniques suggested to be used in three different operation modes: Design Thinking, initial development, and fully development. Each mode has different levels of DT and SCRUM. First is more focused on DT and third is more focused in SD with Scrum. The model initial development merge both
DT@XP	G	DT practices and techniques adapted into two XP phases called exploration and planning phases	Integrates user stories with personas, uses multidisciplinary teams approaches, develops prototypes and perform user centered design acceptance tests

## 6. Comparison with Related Work

Literature shows some efforts for supporting IT professionals on the use of Design Thinking in software development. We present here 4 related work to our proposed research identified in our Systematic Literature Review (SLR). We have classified the related studies into two classes: tool and guidelines. Tool refers to software that involves features using computational mechanisms, allowing user interactions for providing DT techniques recommendations. The guidelines class represents available resources aiming to help DT users to use DT techniques in software development. Table 2 summarizes the related studies comparing its DT approaches, its classification (T for tools or G for guidelines), if they provide recommendations of DT Techniques, and how they do it.

Design Thinking Assistant for Requirements Elicitation (DTA4RE) [Souza 2019] is a tool for DT Techniques recommendation, based on a questionnaire which requests information from the users for suggesting some techniques related to three phases of DT: inspiration, ideation, and prototyping. DTA4RE provides a collection of techniques of DT to be applied for software development. It also shows detailed information about 27 techniques, providing details such as: what is the technique, how to use it, what are the requisites to apply it, and so on. But, although DTA4RE to be a DT techniques tool, it differs for our approach since it does not provide an iterative way for evaluating DT techniques. Also, DTA4RE does not focus on capturing feedback from the professionals about their experiences.

IDEO DT [IDEO 2020] can be considered as a hybrid approach for DT techniques selection. It focuses on providing information for DT as a problem-solving, and also provides a mechanism for filtering and saving those techniques for a DT application. IDEO DT focus on suggesting techniques for three DT working spaces: Inspiration, Ideation, and Prototyping. Despite being for general purposes, we consider IDEO DT a related work because of its use by software development companies. IDEO DT does not provide techniques evaluations. It also does not allow evaluation and feedback of the technique for the practitioners, which makes our approach different.

DT@SCRUM [Vetterli et al. 2015] and DT@XP [Sohaib et al. 2019] are alternatives aiming to foster DT in software development in conjunction with the agile methods

they represent. DT@SCRUM provides suggestions about what DT technique to using in each step of the Scrum-based development process. Also, it provides the prerequisites and the deliverable expected in each step. In DT@XP is made a combination of DT and XP method. DT activities such as Empathy and Define are used in the exploration phase creating an updated version of the user stories called DT-user stories, combining user stories and personas. However, differently to our purpose, these studies are guidelines and not a tool or a system that provides computational resources for helping how to select a DT technique.

As presented in Table 2, our research introduces some progress to the topic. We are proposing a collection of new aspects related to DT in software development, such as evaluation of the techniques at execution time, using these feedback as data for the recommendation logic; creation of a community, exploring the feedback of the professionals do provide useful information on the selection and use of suitable DT techniques. Also, we are proposing the construction of a theoretical context-based model, an abstract approach for helping DT users how to select these techniques.

## References

- Berger, A. (2011). Design Thinking for Search User Interface Design. In *Proceedings of the European Human-Computer Interface Conference*, pages 1–4, New York, USA. ACM.
- Brenner, W. and Uebernickel, F. (2016). *Design Thinking for Innovation: Research and Practice*. Springer, Basel, Switzerland.
- Dybå, T., Prikładnicki, R., Rönkkö, K., Seaman, C., and Sillito, J. (2011). Qualitative Research in SE. *Empirical Software Engineering*, 16(4):425–429. Springer.
- Easterbrook, S., Singer, J., Storey, M.-A., and Damian, D. (2008). Selecting Empirical Methods for Software Engineering Research. *Guide to Advanced Empirical Software Engineering*, pages 285–311.
- Gurusamy, K., Srinivasaraghavan, N., and Adikari, S. (2016). An integrated framework for design thinking and agile methods for digital transformation. In *Proceedings of International Conference of Design, User Experience, and Usability*, pages 34–42, Toronto, Canada. Springer.
- Hehn, J., Mendez, D., Uebernickel, F., Brenner, W., and Broy, M. (2020). On integrating design thinking for a human-centered requirements engineering. *IEEE Software*, 37(2):25–31.
- IDEO (2020 (accessed May 20, 2020)). *IDEO Design Kit*. Available in: <https://designkit.org/methods>.
- Jensen, M. B., Lozano, F., and Steinert, M. (2016). The Origins of Design Thinking and the Relevance in Software Innovations. In *Proceedings of the International Conference on Product-Focused Software Process Improvement*, pages 675–678, Trondheim, Norway. Springer.
- Kitchenham, B., Brereton, P., Budgen, D., Turner, M., Bailey, J., and Linkman, S. (2009). Systematic Literature Reviews in Software Engineering – A Systematic Literature Review. *Information and Software Technology*, 51(1):7 – 15.



- Kitchenham, B., Linkman, S., and Law, D. (1997). DESMET: a Methodology for Evaluating Software Engineering Methods and Tools. *Computing & Control Engineering Journal*, 8(3):120–126.
- Lindberg, T., Meinel, C., and Wagner, R. (2011). *Design Thinking: A Fruitful Concept for IT Development?*, pages 3–18. Springer, Berlin, Germany.
- Lucena, P., Braz, A., Chicoria, A., and Tizzei, L. (2016). IBM Design Thinking Software Development Framework. In *Proceedings of the Brazilian Workshop on Agile Methods*, pages 98–109, Curitiba, Brazil. Springer.
- Luedeke, T., Köhler, C., Conrad, J., Grashiller, M., Sailer, A., and Vielhaber, M. (2018). CPM/PDD in the Context of Design Thinking and Agile Development of Cyber-Physical Systems: Use Cases and Methodology. In *Proceedings of the NordDesign Conference*, pages 1–24, Linköping, Sweden.
- Nedeltcheva, G. N. and Shoikova, E. (2017). Coupling Design Thinking, User Experience Design and Agile: Towards Cooperation Framework. In *Proceedings of the International Conference on Big Data and Internet of Things*, pages 225–229, London, UK. ACM.
- Parizi, R., da Silva, M., Couto, I., Trindade, K., Plautz, M., Marczak, S., Conte, T., and Candelero, H. (2020). Design thinking in software requirements: What techniques to use? a proposal for a recommendation tool. In *Proceedings of the Ibero-American Conference on Software Engineering*, Curitiba, Brazil. Curran Associates.
- Prestes, M. (2020). Estudo exploratório sobre design thinking no desenvolvimento de software. Master’s thesis, Escola Politécnica, Pontifícia Universidade Católica do Rio Grande do Sul, Porto Alegre, Brazil.
- Prestes, M., Parizi, R., Marczak, S., and Conte, T. (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*, pages 73–86, Copenhagen, Denmark. Springer.
- Sohaib, O., Solanki, H., Dhaliwa, N., Hussain, W., and Asif, M. (2019). Integrating design thinking into extreme programming. *Journal of Ambient Intelligence and Humanized Computing*, 10(6):2485–2492.
- Souza, A., Ferreira, B., and Conte, T. (2017). Aplicando design thinking em engenharia de software: um mapeamento sistemático. In *Proc. of the Ibero-American Conference on SE*, pages 719–732, Buenos Aires, Argentina. Curran Associates.
- Souza, A. F. B. d. (2019). Dta4re: um assistente de apoio ao design thinking para elicitação de requisitos. Master’s thesis, Univ. Fed. do Amazonas, Manaus, Brazil.
- Vetterli, C., Uebernickel, F., Brenner, W., Häger, F., Kowark, T., Krüger, J., Müller, J., Plattner, H., Stortz, B., and Sikkha, V. (2015). *DT@SCRUM: Design Thinking Research*, volume 2 of *Understanding Innovation*. Springer, Cham.