

Helius: On a Recommendation System of Design Thinking Techniques for Software Development based on Professionals' Collaboration

Rafael Parizi^{1,2}, Igor Couto¹, Lucas Hanauer¹,
Tayana Conte³, Sabrina Marczak¹

¹ MunDDoS Research Group – Escola Politécnica
Pontifícia Universidade Católica do Rio Grande do Sul - PUCRS
Porto Alegre, RS, Brasil

² Instituto Federal Farroupilha – IFFAR
São Borja, RS, Brasil

³ USES Research Group
Universidade Federal do Amazonas – UFAM
Manaus, AM, Brasil

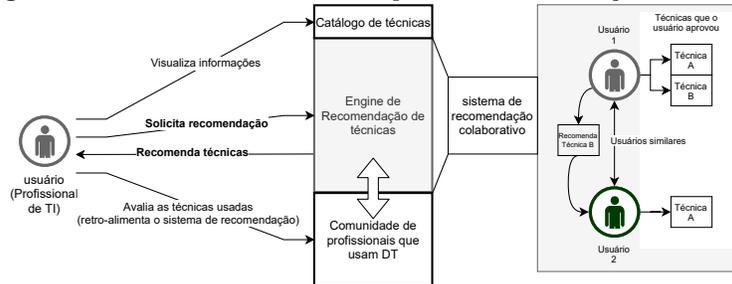
{rafael.parizi,igor.couto,lucas.hanauer}@edu.pucrs.br,
tayana@icomp.ufam.edu.br, sabrina.marczak@pucrs.br

Resumo This paper reports further on Helius, a collaborative recommendation system of Design Thinking techniques for software development. DT has been using as an approach for problem-solving, supporting software development activities such as Requirement Engineering (RE). DT integrated to RE fosters empathy, team collaboration, creativity, and an innovative mindset putting the user needs at the center of the development process. DT provides a large set of techniques, including Interviews, Brainstorming, Personas, User Journeys, Prototyping, etc. Therefore, selecting the DT techniques for RE activities becomes a challenge, mainly for novice professionals. This way, Helius uses professionals' experiences and techniques evaluations as collaborative resources to recommend DT techniques. Helius' proposes resulted from a DT session with software development professionals and researchers. Helius presents innovative features when compared with similar tools. Therefore, we expect to support software development professionals in selecting DT techniques and collaborate with the DT adoption in software development.

Keywords: Design Thinking · Techniques · Recommendation System · Requirements Engineering · Elicitation Support.

1 Introdução

Design Thinking (DT) tem sido utilizado no desenvolvimento de software como uma abordagem que fomenta a exploração do problema de forma iterativa e multidisciplinar, que promove o engajamento dos usuários, e que encoraja a filosofia do aprender pela prática voltada à criação da solução desejada [7]. O DT

Figura 1. Funcionamento do Helius a partir da recomendação colaborativa

tem sido integrado à Engenharia de Requisitos para auxiliar a compreensão do problema, a exploração da solução e a inovação [2], colaborando com a elicitação de requisitos, com o relacionamento com o cliente e em uma maior coesão da equipe de desenvolvimento de software [8].

Neste contexto, a literatura reporta que a experiência e o papel exercido pelo profissional são fatores determinantes para a percepção do DT nas organizações e, conseqüentemente, como as técnicas associadas ao DT são entendidas [4]. Estudos como [3] e Carlgren et al. [1] discutem a importância da seleção das técnicas de DT para alcançar resultados satisfatórios no desenvolvimento de software e, que a seleção das técnicas deve ser realizada como uma atividade prévia ao início do processo de desenvolvimento.

Portanto, para auxiliar os profissionais do desenvolvimento de software a selecionar as técnicas de DT para a Engenharia de Requisitos, este trabalho demonstra Helius, um sistema de recomendação colaborativo que recomenda técnicas de DT. Esta demonstração foca nas funcionalidades que permitem a recomendação das técnicas de DT a partir da experiência de uso de técnicas de DT por profissionais, ou seja, o sistema recomenda técnicas de forma colaborativa, indo além da concepção da ferramenta apresentada em [9].

2 Recomendações Colaborativas e Arquitetura do Helius

Helius é um sistema de recomendação que visa auxiliar profissionais de TI na tomada de decisão da seleção das técnicas de DT. Sistemas de recomendação ajudam os usuários a lidar com cenários em que haja uma quantidade de dados que necessite ser processada para a tomada de decisão [6].

Helius é colaborativo por promover a interação dos usuários através da avaliação das técnicas de DT que os mesmos utilizaram para auxílio à ER. A partir destas avaliações das técnicas, o sistema é capaz de recomendar técnicas para novos usuários. Desta forma, Helius configura um ambiente de comunidade em que os profissionais podem colaborar entre si para conhecer novas técnicas de DT a partir da experiência de uso de técnicas e avaliação de outros profissionais.

A Figura 1 mostra um diagrama que representa o funcionamento do Helius. As recomendações no Helius são fornecidas a partir da explícita requisição de um

Tabela 1. Funcionalidades do Heliu

Funcionalidade	Descrição da funcionalidade
Recomendar técnicas de DT	Considerar o uso combinado de técnicas para recomendar técnicas de DT
	Considerar informações dos projetos para recomendar técnicas de DT
	Considerar o <i>feedback</i> de outros usuários para recomendar técnicas de DT
Filtrar técnicas de DT	Considerar características das técnicas para filtrar técnicas de DT similares
	Considerar informações de contexto para filtrar técnicas de DT similares
Avaliar técnicas de DT	Permitir a avaliação de uma técnica utilizada pelo usuário
	Permitir avaliar o uso combinado de técnicas DT em um projeto
Acessar o <i>feedback</i> de outros usuários	Permitir filtrar o <i>feedback</i> dos usuários para as técnicas
	Mostrar o <i>feedback</i> relacionado a projetos com técnicas utilizadas em conjunto
Fornecer informações das técnicas de DT	Mostrar informações sobre cada técnica de DT
	Mostrar informações do combinado de uma técnica com outras técnicas
Gerenciar repositórios de técnicas	Gerenciar repositório com técnicas usadas pelo profissional em projetos
	Compartilhar dados do projeto com outros membros da equipe

usuário. Isto faz do Heliu um sistema de recomendação reativo, diferentemente dos sistemas de recomendação proativos em que as recomendações são fornecidas sem a solicitação do usuário [5]. Heliu caracteriza-se com um sistema de recomendação de filtragem colaborativa por utilizar avaliações de usuários para recomendar técnicas de DT [5]. Isto indica que o sistema compara os usuários para encontrar usuários similares a partir das avaliações das técnicas fornecidas por usuários e, a partir da identificação dos usuários similares, recomenda itens aos usuários que requisitaram. Heliu fornece ainda um catálogo de técnicas, ou seja, disponibiliza aos usuários informações de como aplicar as técnicas de DT.

A Tabela 1 mostra as funcionalidades previstas no Heliu, com destaque às funcionalidades colaborativas⁴. A funcionalidade “Considerar o *feedback* de outros usuários para recomendar técnicas de DT” estabelece que o algoritmo de recomendação considere as avaliações dos profissionais quanto ao uso das técnicas de DT para calcular as recomendações. Esta funcionalidade está vinculada às duas funcionalidades da macro-funcionalidade “Avaliar técnicas de DT”, que sugere a permissão à avaliação de uma técnica de forma individualizada, ou ainda a um conjunto de técnicas. Além destas, a macro-funcionalidade “Acessar o *feedback* de outros usuários” com as funcionalidades “Permitir filtrar o *feedback* dos usuários para as técnicas” e “Mostrar o *feedback* relacionado a projetos com técnicas utilizadas em conjunto” denotam a criação do ambiente de comunidade de usuários de técnicas de DT, onde profissionais podem conhecer as experiências de outros a partir do uso de técnicas de DT no auxílio a atividades da ER.

A Figura 2 apresenta a arquitetura da versão 1 do Heliu que está sendo desenvolvido em *Flutter* como um aplicativo para dispositivos móveis. Heliu possui conexão com banco de dados NoSQL MongoDB, através de uma comunicação realizada por uma API (*Application Programming Interface*), desenvolvida com o *framework* NestJS.

⁴ A extração, especificação e validação de requisitos pode ser encontrado em [9].

Figura 2. Arquitetura do sistema Heliu

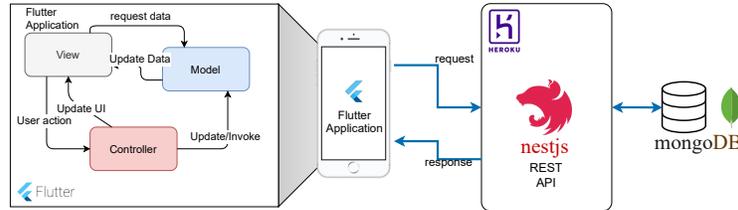


Figura 3. Exemplos de interfaces do Heliu



3 Interfaces de Usuário do Heliu

A Figura 3 mostra exemplos das interfaces de usuários do Heliu. Destaca-se que as interfaces representadas em (a) e em (b) são interfaces que exploram as funcionalidades de colaboração dos profissionais. Em (a), apresenta-se a interface de revisão de uma técnica, na qual o profissional poderá expor sua experiência com a utilização de tal técnica de DT, através de nota (*rate*), e campos de texto. Em (b), apresenta-se um grafo de relacionamento de técnicas, o qual representa as 5 técnicas recomendadas para o usuário. O grafo mostra as 5 técnicas mais utilizadas e seus relacionamentos (projetos que utilizaram as técnicas em conjunto). O tamanho dos vértices indica a quantidade vezes que cada técnica foi usada. As interfaces (c) e (d) mostram informações de uma determinada técnica, a criação do repositório de técnicas, e a requisição de recomendações⁵.

4 Comparação com Ferramentas Similares

Nesta seção apresenta-se uma comparação do Heliu com ferramentas identificadas na literatura: DTA4RE, IDEO DT e DT@IT (veja Tabela 2).

⁵ Para criação de repositório de técnicas, o usuário (profissional) pode selecionar contextos de uso da técnica. Os contextos foram mapeados da literatura em [10].

Tabela 2. Funcionalidades do Heliu

Funcionalidade	DTA4RE	IDEO DT	DT@IT
Recomendar técnicas de DT	●	◐	◐
Filtrar técnicas de DT	●	●	◐
Avaliar técnicas de DT	◐	◐	◐
Acessar o <i>feedback</i> de outros usuários	◐	◐	◐
Fornecer informações das técnicas de DT	●	●	●
Gerenciar portfólio com técnicas de DT	◐	◐	◐

● – Funcionalidade encontrada
 ◐ – funcionalidade parcialmente implementada
 ◐ – funcionalidade não encontrada

DTA4RE⁶ (*Design Thinking Assistant for Requirements Engineering*) é um assistente de seleção de técnicas de DT que fornece ao usuário um formulário com perguntas relacionadas ao contexto do projeto e às necessidades de aplicação de técnicas e, com base nas respostas, lista as técnicas de DT apropriadas. IDEO DT⁷ é um catálogo Web de técnicas de DT em que o usuário pode realizar o acesso a um conjunto de técnicas, conhecer os detalhes das técnicas como tempo necessário de aplicação, materiais necessários, envolvidos, entre outros, e pode filtrar de acordo com o espaço de trabalho de utilização. DT@IT⁸ fornece um catálogo com 12 técnicas de DT, para as quais oferece *templates* de aplicação, ou seja, guias para os usuários utilizarem as técnicas.

A diferença do Heliu para as ferramentas similares em termos de recomendação de técnicas reside no fato do Heliu tratar-se de um sistema que explora a colaboração a partir das experiências de usuários como um mecanismo de retroalimentação para as recomendações de técnicas de DT. Isto significa que as recomendações, por utilizarem de recurso colaborativo e por serem personalizadas, podem ser diferentes para cada usuário e/ou serem diferentes por conta das experiências dos usuários que as avaliaram.

5 Considerações Finais e Trabalhos Futuros

Neste artigo demonstrou-se Heliu, um sistema de recomendação para auxílio à tomada de decisão de profissionais de TI na seleção técnicas de DT para o desenvolvimento de software. O foco da demonstração voltou-se para as funcionalidades que configuram o Heliu como um sistema de recomendação colaborativo, ou seja, que consideram as experiências de profissionais na utilização das técnicas de DT, as quais são coletadas por Heliu no formato de avaliações de técnicas.

A partir das funcionalidades, das interfaces de usuário e da breve comparação com outras ferramentas relacionadas a técnicas de DT, para o desenvolvimento de software ou de maneira geral, aponta-se que Heliu inova ao considerar a

⁶ <https://sites.google.com/site/dta4re/>

⁷ <https://www.designkit.org/>

⁸ <https://espspinix.github.io/dt-at-it-toolbox/>

colaboração dos usuários (profissionais de TI) como mecanismo para a retroalimentação dos algoritmos de recomendação. Heliu realiza recomendações de técnicas de DT considerando as técnicas que foram mais utilizadas ou ainda considerando as técnicas que foram melhor avaliadas pelos usuários. Como trabalhos futuros pretende-se incluir algoritmos de recomendação que considerem o perfil do usuário para a recomendação.

Vídeo de demonstração do Heliu: <https://youtu.be/8ypRN5TZ2Sg>.

Acknowledgements

Este projeto é apoiado pelo Programa de Bolsas Pesquisa Alunos da PUCRS Chamada Geral 1/2019 e 1/2020, CNPq 314174/2020-6, CNPq 307177/2018-1 e CNPq PIBIC 2019/2020. Este estudo foi parcialmente financiado pela CAPES - Código de Financiamento 001.

Referências

1. Carlgren, L., Rauth, I., Elmquist, M.: Framing Design Thinking: The Concept in Idea and Enactment. *Creativity and Innovation Management* **25**, 38–57 (2016)
2. de Paula, T.R., Santana Amancio, T., Nonato Flores, J.A.: Design Thinking in Industry. *IEEE Software* **37**(2), 49–51 (2020)
3. Dobrigkeit, F., Pajak, P., de Paula, D.: DT@IT Toolbox: Design Thinking Tools to Support Everyday Software Development, pp. 201–227. Springer (2020)
4. Dobrigkeit, F., de Paula, D.: Design Thinking in Practice: Understanding Manifestations of Design Thinking in Software Engineering. In: Proceedings of the European Software Engineering Conference and Symposium on the Foundations of Software Engineering. p. 1059–1069. ACM, Tallinn, Estonia (2019)
5. Gasparic, M., Janes, A.: What Recommendation Systems for Software Engineering Recommend: A Systematic Literature Review. *Journal of Systems and Software* **113**, 101–113 (2016)
6. Isinkaye, F., Folajimi, Y., Ojokoh, B.: Recommendation Systems: Principles, Methods and Evaluation. *Egyptian Informatics Journal* **16**(3), 261–273 (2015)
7. Kuula, S., Haapasalo, H., Kosonen, J.: Three Phases of Transforming a Project-Based IT Company Into a Lean and Design-Led Digital Service Provider. *IEEE Software* **37**(2), 41–48 (2020)
8. Mahe, N., Adams, B., Marsan, J., Templier, M., Bissonnette, S.: Migrating a Software Factory to Design Thinking: Paying Attention to People and Mind-Sets. *IEEE Software* **37**(2), 32–40 (2020)
9. Parizi, R., Moreira da Silva, M., de Souza Couto, I., Pavin Trindade, K., Plautz, M., Marczak, S., Conte, T., Candello, H.: Design Thinking in Software Requirements: What Techniques to Use? A Proposal for a Recommendation Tool. In: Proceedings of the Ibero-American Conference-American on Software-American Engineering. pp. 1–14. Curran Associates, Curitiba, Brazil (2020)
10. Prestes, 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)