**1. Bibliographic Data**

* **Title:** On Integrating Design Thinking for Human-Centered Requirements Engineering
* **Authors:** Jennifer Hehn, Daniel Mendez, Falk Uebernickel, Walter Brenner and Manfred Broy
* **Year:** 12 February 2020
* **Publication:** IEEE Software

**2. Theme of the Paper**

**Scientific Area:** requirements engineering and design thinking in software engineering

**Specific Topics:**

* human centered requirements discovery and elicitation
* Integration strategies to combine design thinking DT with traditional requirements engineering RE approaches
* establish a shared model connecting some more user focus prototypes and systematic requirement specifications
* meeting the challenges of imprecise or volatile requirements in complex and innovative software projects

**3. Synthesis of the Paper**

**Motivation and Importance of the Research**  
Requirements engineers tend to find difficulties in the requirements gathering and their clarification really often, as they change quite a lot and sometimes aren’t very clear. The DT offers a creative approach, centered on the human being and interactive in order to better understand the users in order to complements the practices used in RE that are more structure like. The authors see a lot of potential in the synergy between RE and DT but think that there’s not a lot of orientation on how to do this in a systematic way.

**Main Points of Background Information and State of the Art**

* **Design Thinking**: presented as a structure process but still being kinda flexible, that emphasizes empathy, a fast prototyping and interactive learning. It is centered on exploring the user’s context and needs, often leading to nontechnical prototype solutions.
* **Requirements Engineering**: a discipline to get, specify and manage the requirements systematically. Traditional RE is often more formal and oriented towards the documenting functional and nonfunctional requirements for software systems.
* Historically, many RE activities assume that requirements exist and need to be elicited. DT assumes a problem space must first be deeply explored before thinking about the solutions.

**Main Findings and Results, and Their Novelty**

**Combined Artifact Model**: the paper presents a blueprint merging artifacts from both DT and RE. It organizes deliverables into three layers: context layer, requirements layer and system layer.

**Three Integration Strategies**:

* **Upfront Design Thinking**: conduct a dedicated DT phase to explore user needs and produce a prototype before traditional RE activities begin
* **Infused Design Thinking**: insert DT into ongoing RE processes, use DT workshops or methods where clarity or creativity is needed without needing to change the entire RE setup
* **Continuous Design Thinking**: continuously merge DT and RE ideas throughout a project, integrating roles and processes across the entire lifecycle

By analyzing these strategies, the paper highlights that the companies can adopt a flexible and oriented way to the context in order to integrate de DT into the RE. This depends on the project scope, the volatility of the requirements and in the availability of the organizations in being more interactive or oriented.

**Main Conclusions and Discussion Points**

* **Complementary Nature**: DT and RE each bring strengths, human-centered exploration vs systematic specification
* **Project Context Matters**: which integration strategy to choose depends on factors like time, resource availability and how complex the problem space is
* **Open Challenges**: align the principles and artifacts of DT with the formal engineering achieve milestones. Define the functions of who can meditate between creative prototypes and documentation specification. Determine on how to keep DT’s user focus once the formal specification and coding starts.

**4. Questions and Reflection**

**Questions Raised by the Reading**

* Whats the most appropriate strategies for various types of projects? Do simpler projects need only an infused DT but bigger projects with more likes of uncertainty need more like of upfront and continuous DT?
* How is it possible to ensure that DT artifacts like empathy maps lead to clear software requirements?
* What skills or roles do we require? Do we require a special helper or design coach and a requirement engineer?

**Opinion About the Paper**  
The paper shows how DT can be used to teach the users about RE and how RE guarantees that the documentation is complete and adapted to the real world softwares. Based on artifacts the paper makes a good explanation on how DT and RE relate to each other. The paper shows how that organizational support is need for DT to be sustainable. The continuous integration requires more than just one or two innovative workshops and actually requires that all of the people involved work together, otherwise it wont work as expected.

**What to Retain for Future Research/Professional Practice**

* **Artifact Focus:** creative models need to be aligned with formal system rules that are more focused on users work and them conscious of engineering issues
* **Flexible Integration:** there is no one-size-fits-all solution, the engineers should adapt their techniques to each project complexity and to the organization readiness
* **Empathy and Iteration:** both are really needed for discovering real user needs. DT promotes empathy and more like dynamic prototypes. RE provides a good mechanism to capture, validate and manage changes correctly