

DT4RE: Design Thinking for RE

Jennifer Hehn, Falk Uebernickel, Daniel Méndez

20.08.2018

Banff, Alberta, Canada

Jennifer Hehn
www.iwi.unisg.ch

Research areas

- Design Thinking and
- Design Science Research

Falk Uebernickel
goo.gl/WmRyAu

Research areas

- Design Thinking and Human-centered **Design** Requirements Engineering for Information Systems
- Industry 4.0 / Internet of Things
- InsurTech
- Qualitative research

Daniel Méndez
www.mendezfe.org

Research areas

- Empirical software engineering
 - Requirements Engineering
 - Software Process Models
 - Quality Management
- Interdisciplinary qualitative research

This session is based on...

- Previous tutorials given on either Design Thinking or Requirements Engineering
- Experiences made in projects

This tutorial will be about...


Scope

- Introduction into basic principles and methods for Design Thinking (DT)
- Sharing experiences and lessons learnt on using DT in context of RE
- Discuss synergies with RE and open research challenges

Out of Scope

- Out of the box solutions
- Universally applicable “blueprint”

Ground rule

Whenever you have questions / remarks,
please don't ask , but
share them with the whole group.

Introduction - Who are you?

Quick round...

- Who are you?
- What are your experiences in Design Thinking in the context of software development projects/processes?



What do you know?

What is Design Thinking?



Same as with agile methods, there are different perspectives on Design Thinking

Way of Doing

Way of Thinking

Process



Toolbox



Mindset



- (1) Define
- (2) Needfinding
- (3) Synthesis
- (4) Ideate
- (5) Prototype
- (6) Test



Prototype

- » Ioft prototyping
- » wireframing
- » mockups
- » 3D printing
- » CAD Simulations
- » Software programming
- » Mechanical and electrical engineering
- » Service Blueprinting
- » Role plays
- » Stories and Comics



Needfinding
Synthesis

- » Interviews
- » Focus Groups
- » Observation
- » Diary Studies
- » Customer Journeys
- » Re-Framing Techniques
- » Engaging

Bias towards action

Radical Collaboration

Experimentation

Focus on human values

Iteration

What is Design Thinking (not)?

Design Thinking...

- ... is a human-centered problem solving method that applies extensive user-research, rapid prototyping, iterative improvement cycles, and interdisciplinary team work

In contrast, Requirements Engineering

- is a holistic discipline with various principles, approaches and even more methods

Two faces of the same medal?

In Design Thinking, we often pretend that after building a high-resolution prototype, the rest is “just development”.



In RE, we often pretend that requirements are somehow present and “just need to be elicited”.

Issues in scope of current debates

When should we make use of Design Thinking?

How can we make use of Design Thinking?

How can we integrate Design Thinking and RE in a seamless manner?

Outline

1. Design Thinking in a Nutshell



2. Design Thinking for Requirements Engineering

3. Final discussion and closure

Outline

1. Design Thinking in a Nutshell



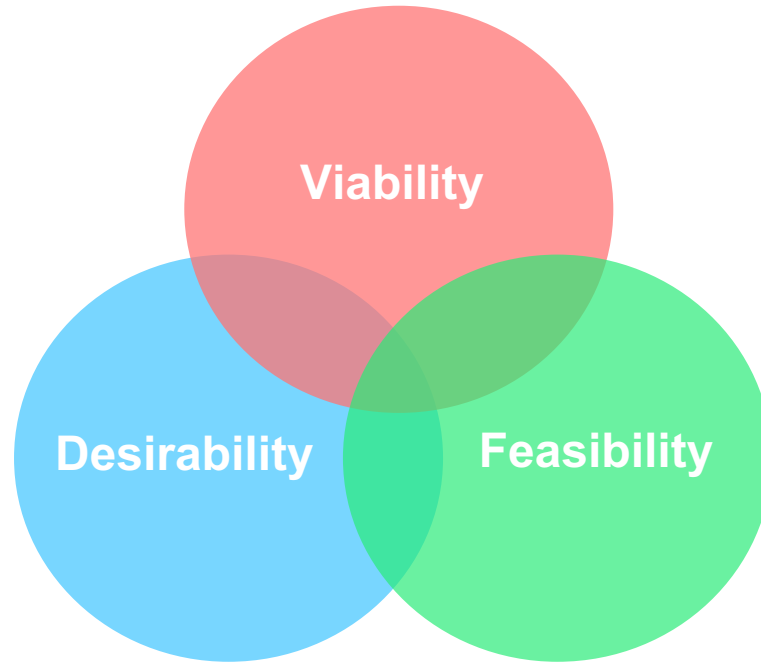
2. Design Thinking for Requirements Engineering

3. Final discussion and closure





Design Thinking is a problem solving approach that starts with the human



Let's try it...



WALLET EXERCISE



Re-design the wallet of your partner.

Your **challenge** for the next 30 minutes.

Find a teammate.

2 people per team

Interview. **Explore.** Understand.

Who? **Why?** Where? What? How? When?

3 minutes each

Point-of-View

Synthesize.

2 minute

Let's hear it.

Share your Point-of-View (POV).

Ideate.

Brainstorm. The more ideas the better.

2 minutes

Prototype.

Express **one idea** through a prototype.

5 minutes

Test it.

Get some **feedback** from your user.

2 minutes each

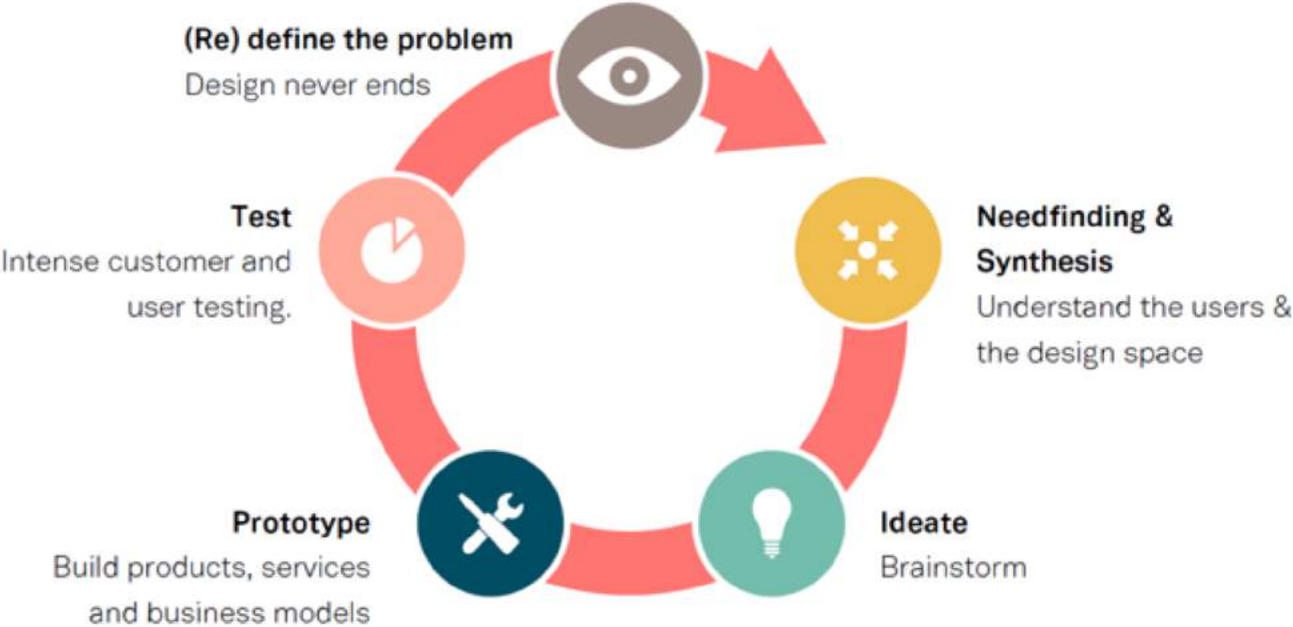
Let's see it.

Share your Prototype.

What was our process?

What did we just do?

Design Thinking is explorative and iterative





Needfinding Empathize.

Interviewing and self-immersion in Kenya (Photographer: Falk Uebernickel)

In Needfinding we apply three methods

Observe



Immerse



Ask



A group of four people (three men and one woman) are seated in a circle in a meeting room. They are surrounded by whiteboards covered in numerous colorful sticky notes and diagrams. One whiteboard in the background shows a flowchart and a line graph. The room is filled with information, suggesting a collaborative brainstorming or synthesis session.

Synthesis
Make sense.

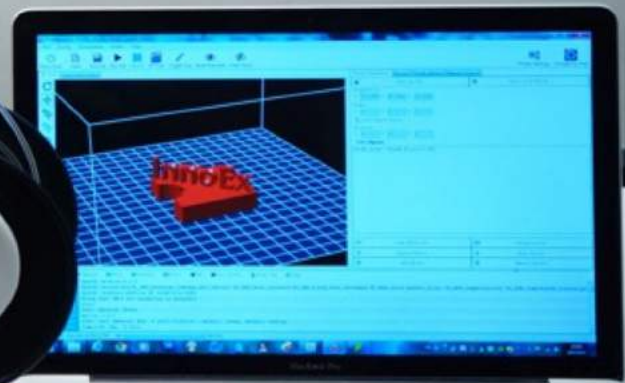
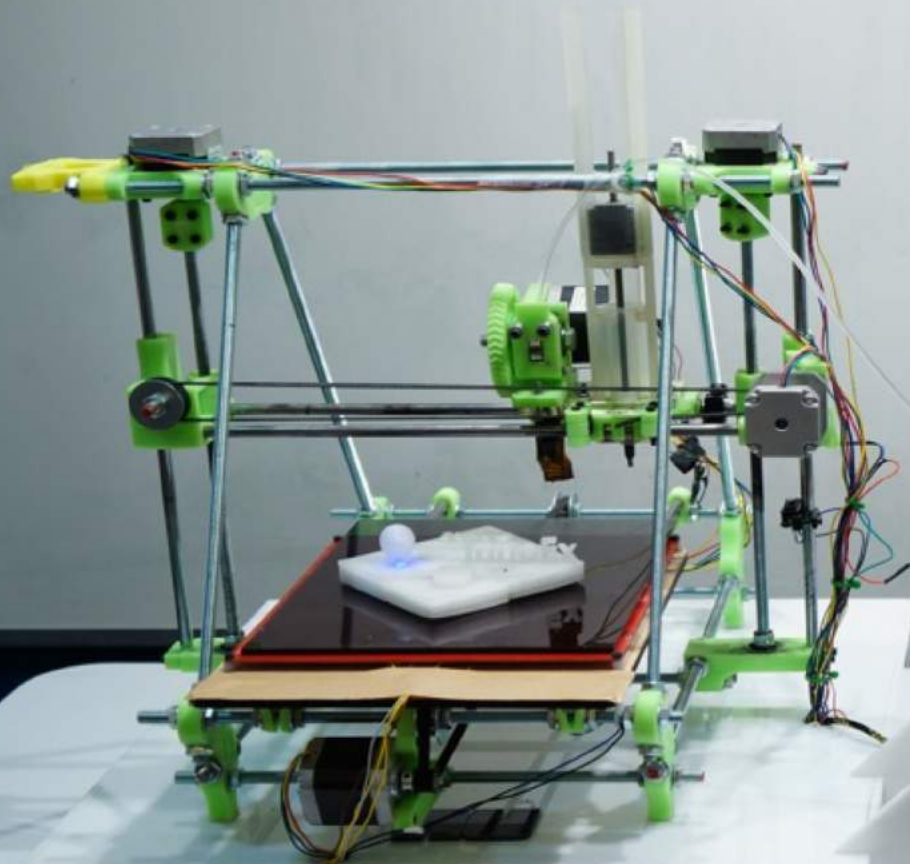


Pictures: project for an insurance company (2017)

Ideate
Generate ideas.



Prototype
Make ideas tangible.





WAPP?

Rise and shine!
09:43

This exercise helps with practically everything, including your core, legs, balance, and strength:

- Sit on the edge of a firm surface like a chair or a bed, with your feet on the ground
- Activate your core and stand up using as little support as you can
- Sit down and repeat 4 times





The outcome of a Design Thinking project is one or more tested prototypes



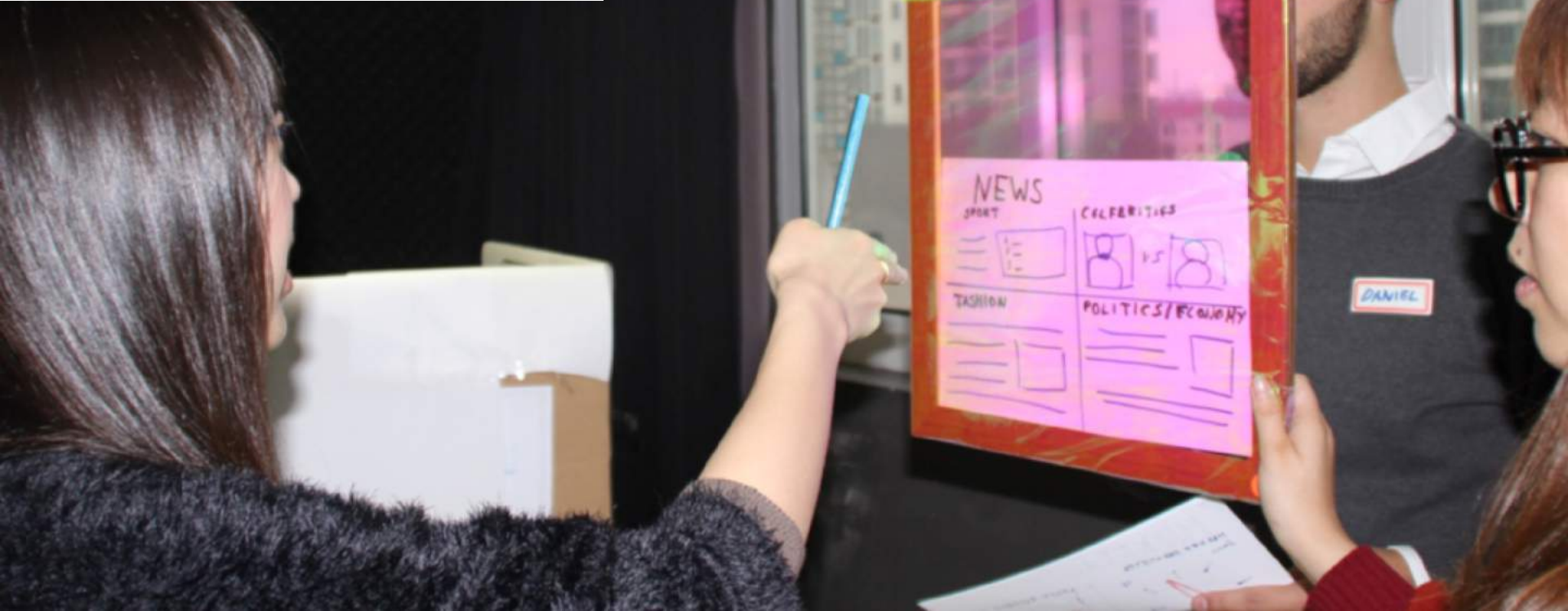
LOW RESOLUTION PROTOTYPE

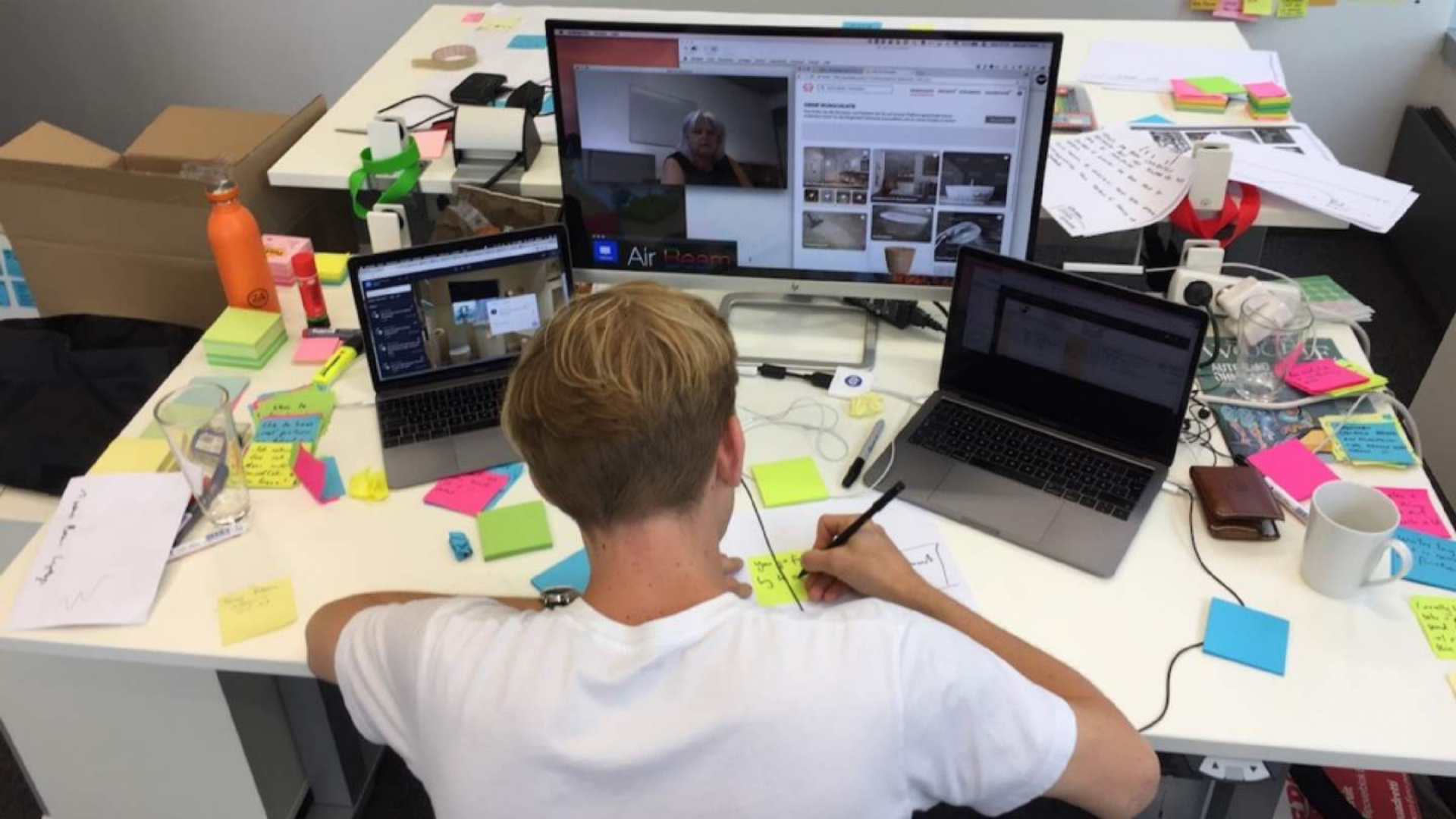
Simple and easy to handle prototypes
Quickly created
Focus is on few features / critical functions
Costs: low

HIGH RESOLUTION PROTOTYPE

Complex simulations and prototypes of the future product, service or process and business model
All important functions are implemented
Costs: higher

Test
Collect feedback.





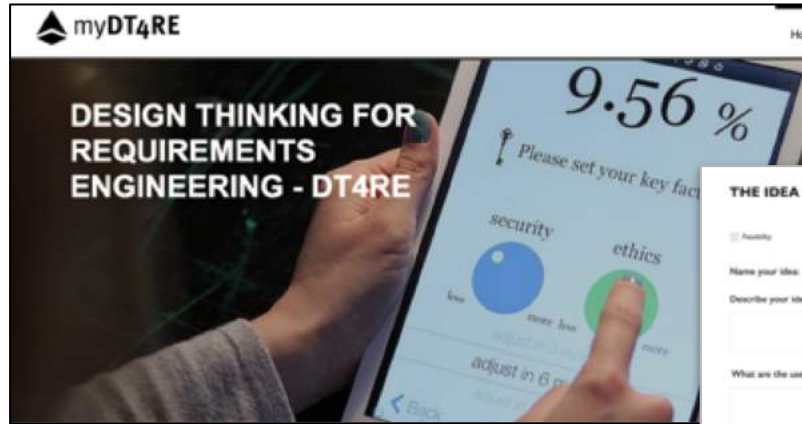
**(Re-)define
Iterate.**



Toolbox



<https://www.dt4re.org/>



IDEATION

The **Ideation** phase represents the process of “going wide” which enables to explore a broad solution space. Brainstorming creativity methods to generate a large quantity of diverse ideas are usually applied. A strong focus lies on listening each other’s ideas to leverage collective thinking. Professional teams can achieve hundreds of diverse ideas which help building rapid prototypes.

L01 // BRAINSTORMING

L02 // IDEA NAPKIN

L03 // PEANUT CHART

PROTOTYPE

Make ideas real and tangible is the motto of the **Prototyping** phase. The goal is to make ideas and concepts as soon as possible testable for customers and other stakeholders. Furthermore, it helps to communicate ideas in the team and the company. In the beginning of a project we use low fidelity prototyping in order to be fast and quick. Later in the process we build high fidelity prototypes that demonstrate the full functionality of a product or service. Typical outcomes are business models, services, products or processes which can be tested with customers and other stakeholders.

THE IDEA NAPKIN

project: _____ date: _____

Available Big Impact High Innovation Factor

Name your idea: _____ Who is your user: _____

Describe your idea: _____ What issue does your idea solve? _____

What are the user profiles? _____ What are the big values from your idea? _____

Sketch how your idea solves the defined _____

102 / IDEA NAPKIN

WHAT?
The idea Napkin service was designed for describing an idea.

WHY?
If there were no examples or absence of time, it is worthwhile spending time not just describing them in more detail. The idea Napkin aims to concrete ideas or idea concepts that were generated in an ideation session and that are usually just a word or short sentence on their own.

HOW?
• The specific user roles necessary. Everybody has the same tools and requirements.

THINK / Infrastructure

- Form
- Processes of the Idea Napkin Template

PROCESS

- 1. Idea Napkin:**
 - 1.1 **Template:** The idea Napkin template is used to describe an idea in order to understand it.
 - 1.2 **Fill & use:** Fill out an idea Napkin for every idea you have previously come up with and want to discuss further. Ideally, stick all the idea Napkins to a thin wall or flat stretched out on the floor. The napkin prompts you to give your idea a name, to describe in one sentence, in detail, the problem solved and the benefits generated by your idea. In addition, it also gives you a think about the user experience by providing free space for a short sketch.
- 2. Presentation:**
 - 2.1 **Presentation:** Every idea Napkin is clearly presented to the group in order to ensure that everybody has an overview and basic understanding of all the ideas.
- 3. Voting:**
 - 3.1 **Rating:** Give a priority to the ideas. Every user describes his vote as short voice or card ("Recommend").
 - 3.2 **Discussion:** Discuss the result of the vote. Usually, the ideas with the most votes are being prioritized in the next step.

People and Making are the heart of
Design Thinking

Principles



See the human
behind the user

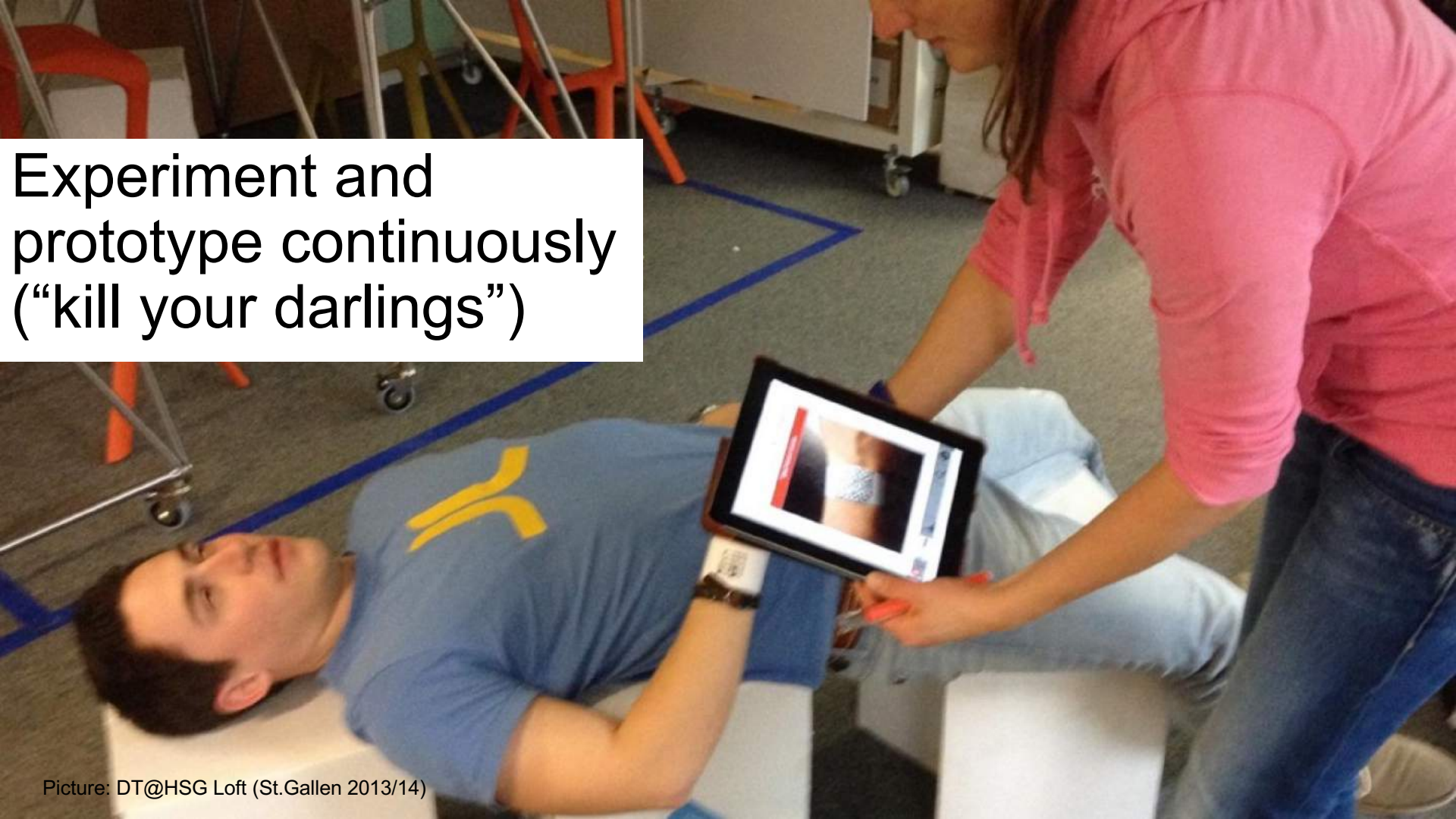


Do not stop at your
corporate doors

Making instead of
over-thinking




Experiment and
prototype continuously
("kill your darlings")



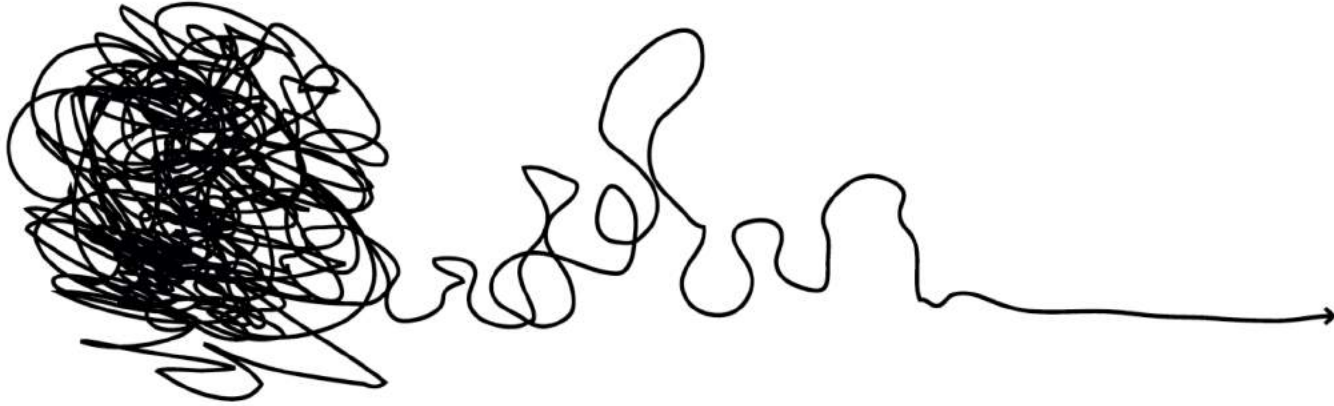
Field testing even in early project phases ("fail forward")



A group of five people are gathered in a living room for a collaborative meeting. Two men are seated on a brown plaid sofa; one is wearing a white t-shirt and the other a black hoodie. Three women are seated on the floor around a red coffee table. The table is cluttered with numerous pink and yellow sticky notes, several tablets, and water bottles. The room features a white bed in the background and a potted plant. The overall atmosphere is one of active collaboration and discussion.

Shape your view with
interdisciplinary teams

Design Thinking transforms wicked into ill- and well-defined problems



Outline

1. Design Thinking in a Nutshell



2. Design Thinking for Requirements Engineering

3. Final discussion and closure

Outline

1. Design Thinking in a Nutshell



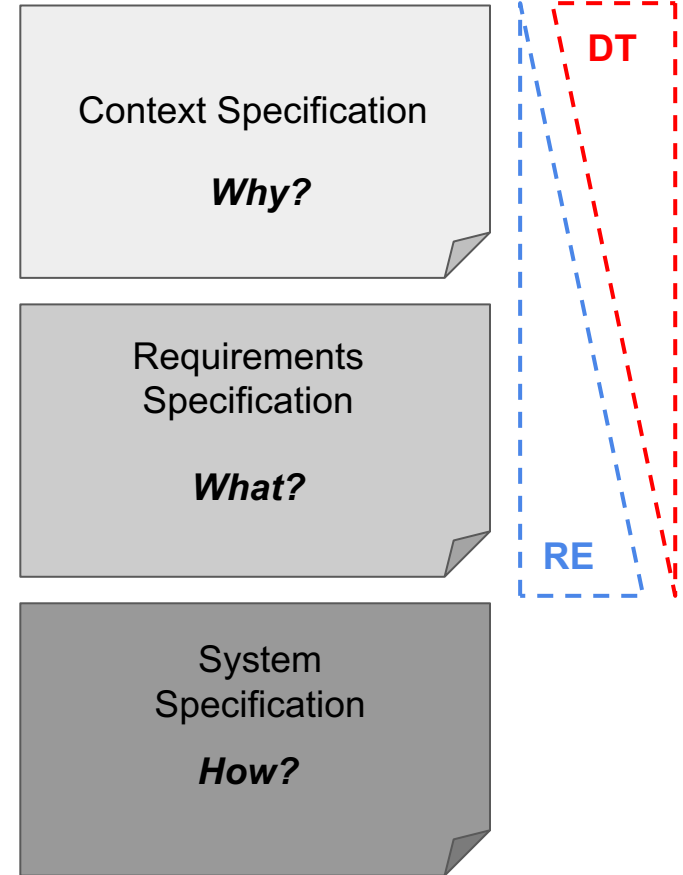
2. Design Thinking for Requirements Engineering

3. Final discussion and closure

Cross Comparison

DT largely concentrates on **identifying/empathising with the stakeholders and end-users**, and understanding the domain and problem space to enable distilling needs and requirements.

RE typically concentrates on **subsequent requirements elicitation, analysis, and documentation**



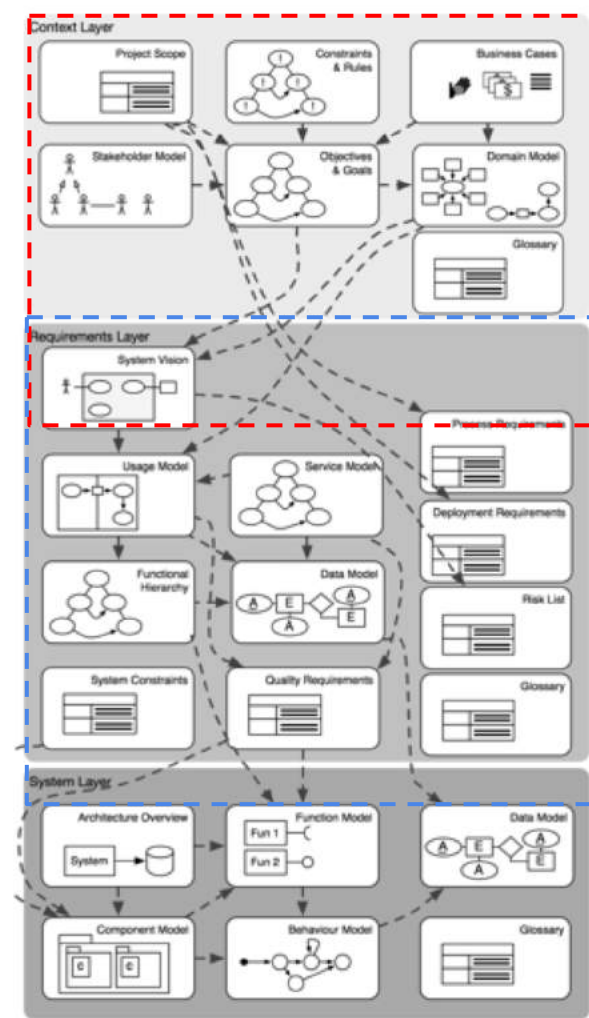
Cross Comparison

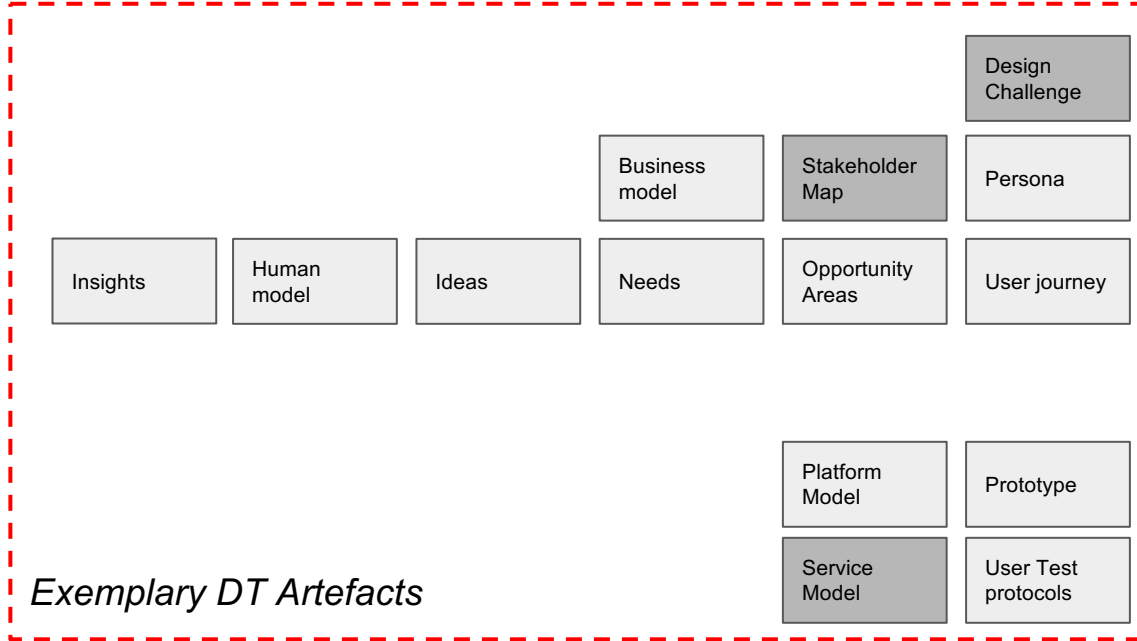
DT largely concentrates on...

- ... better understanding the problem space by identifying and empathising with stakeholders
- ... providing a system vision by defining key (functional) features
- ... the rationale for (“formal”) requirements

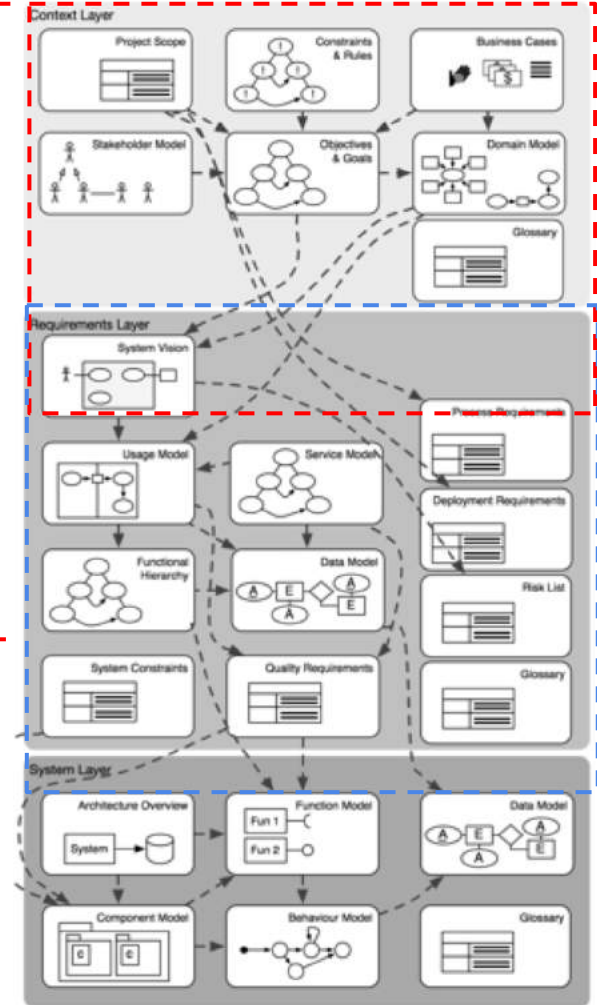
RE largely concentrates on...

- ... identifying, analysing/refining, and specifying/modelling requirements going beyond functional ones





Exemplary DT Artefacts

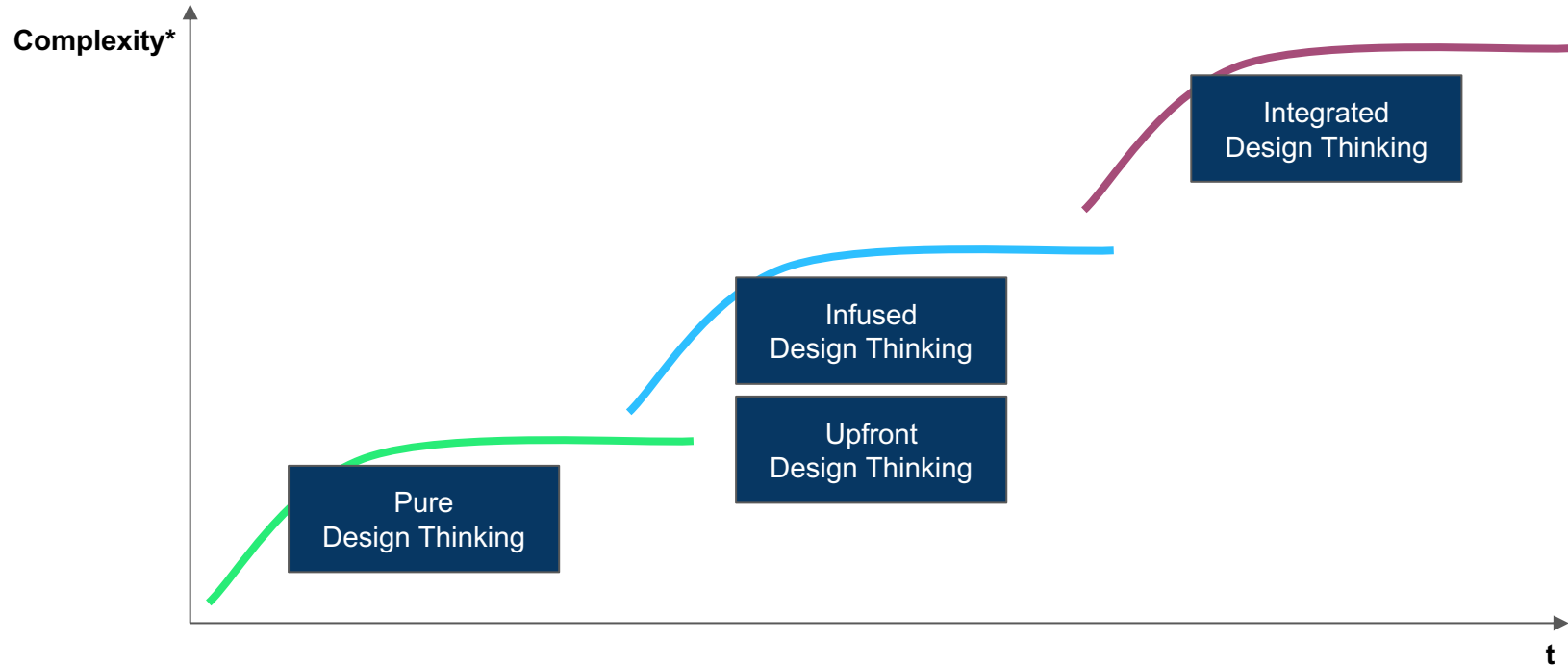


Disclaimer



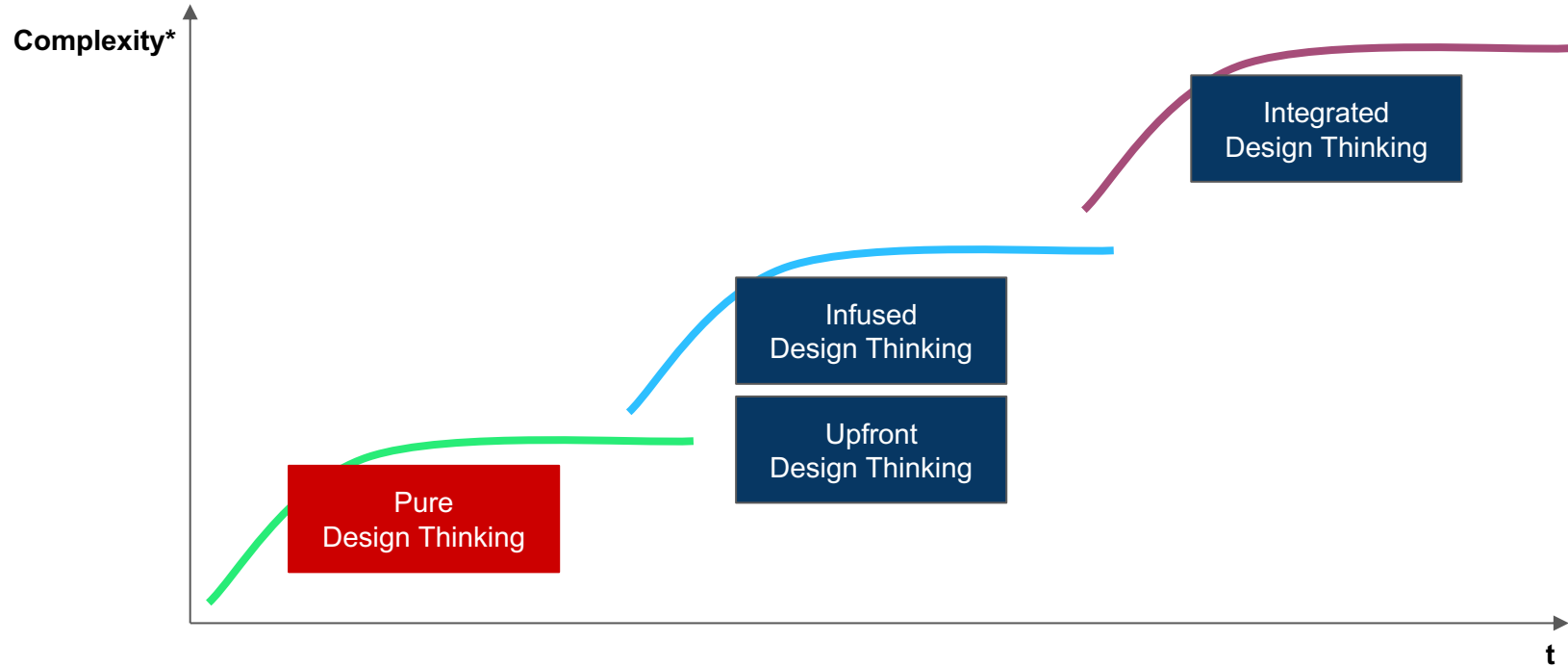
... but many lessons learnt

Evolution of Design Thinking and RE



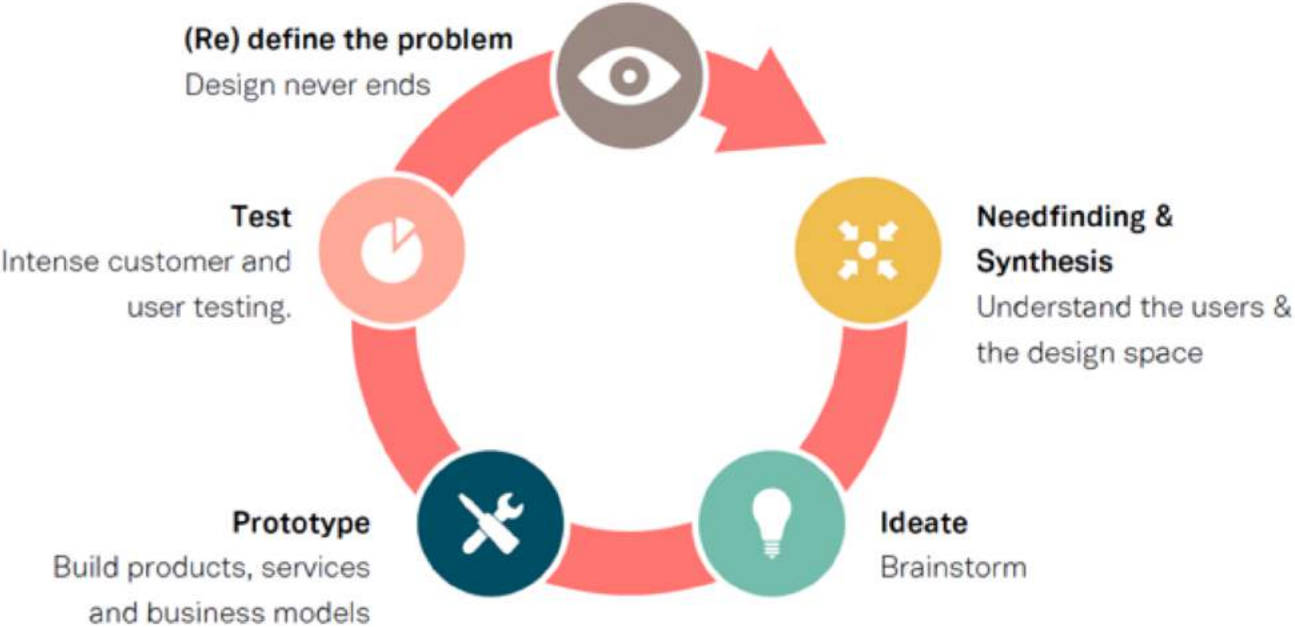
* Note: one is not per-se "better" than the other; everything has its place

Evolution of Design Thinking and RE



* Note: one is not per-se “better” than the other; everything has its place

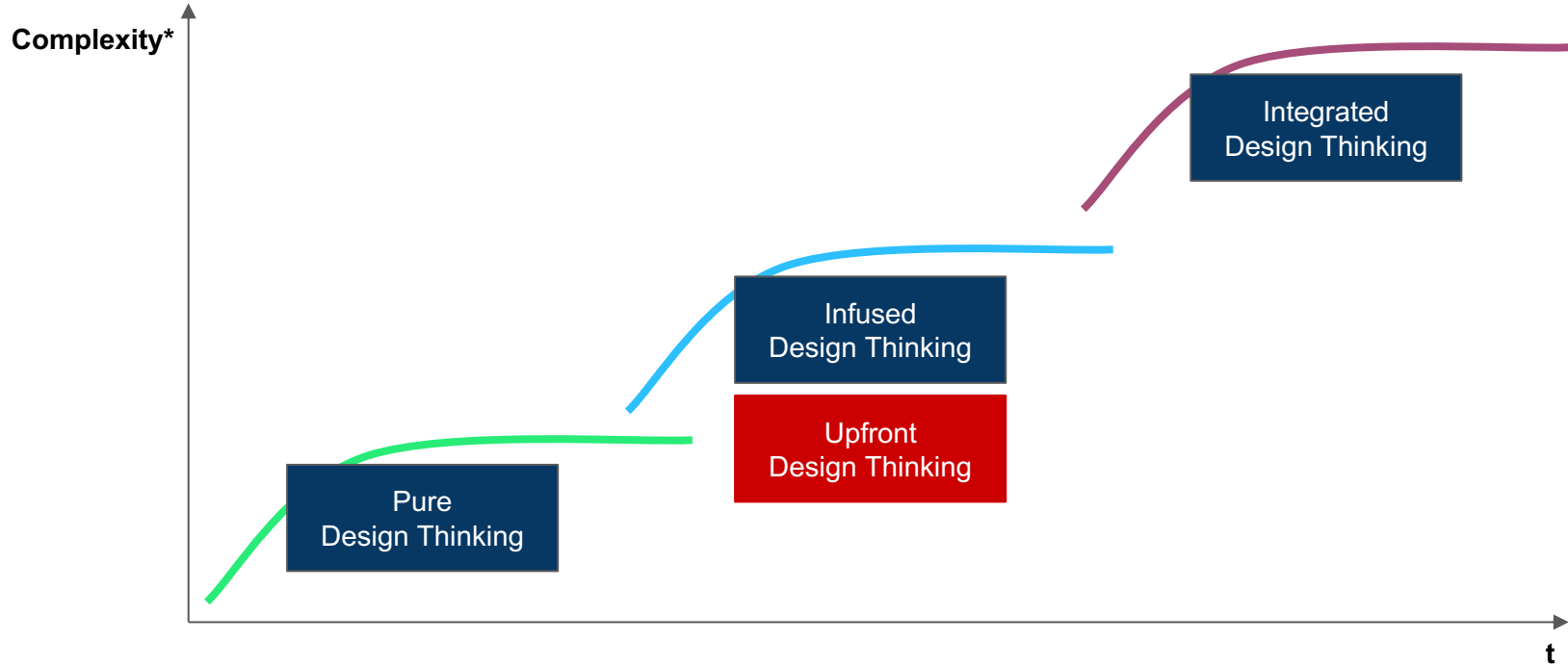
Pure Design Thinking



Take-Aways

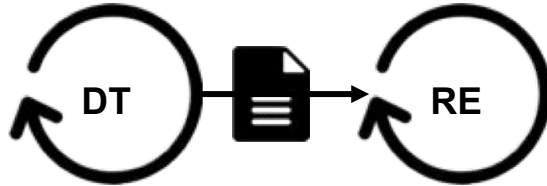
- Much like RE, DT shouldn't suddenly stop
- DT is human-centric, but also team-driven
 - Team members (skills, motivation, participation) are crucial
 - Make explicit implicit assumptions (e.g. to avoid gold plating)
 - Beware dependencies to implicit knowledge
- Potentially working towards the void
 - No immediate counterpart and no institutionalised “hand-shake”
→ Software process model? Needs and team culture?
 - No continuity and potentially no champion
- No guaranteed operationalisation (and feasibility) of prototype

Evolution of Design Thinking and RE



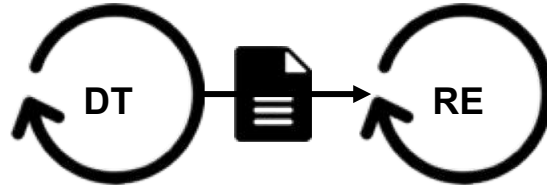
* Note: one is not per-se “better” than the other; everything has its place

Upfront Design Thinking

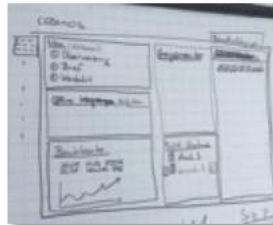


- German Software Company (SME)
- Problem Statement: Development of an offering for a new target group (private landlords) in real estate management
- Team: Requirements Engineer, Product Manager, IT-Architect, Designer, Hotline Support, Project Lead, Design Thinking Coach
- Design Thinking Project: 4 months

Upfront Design Thinking



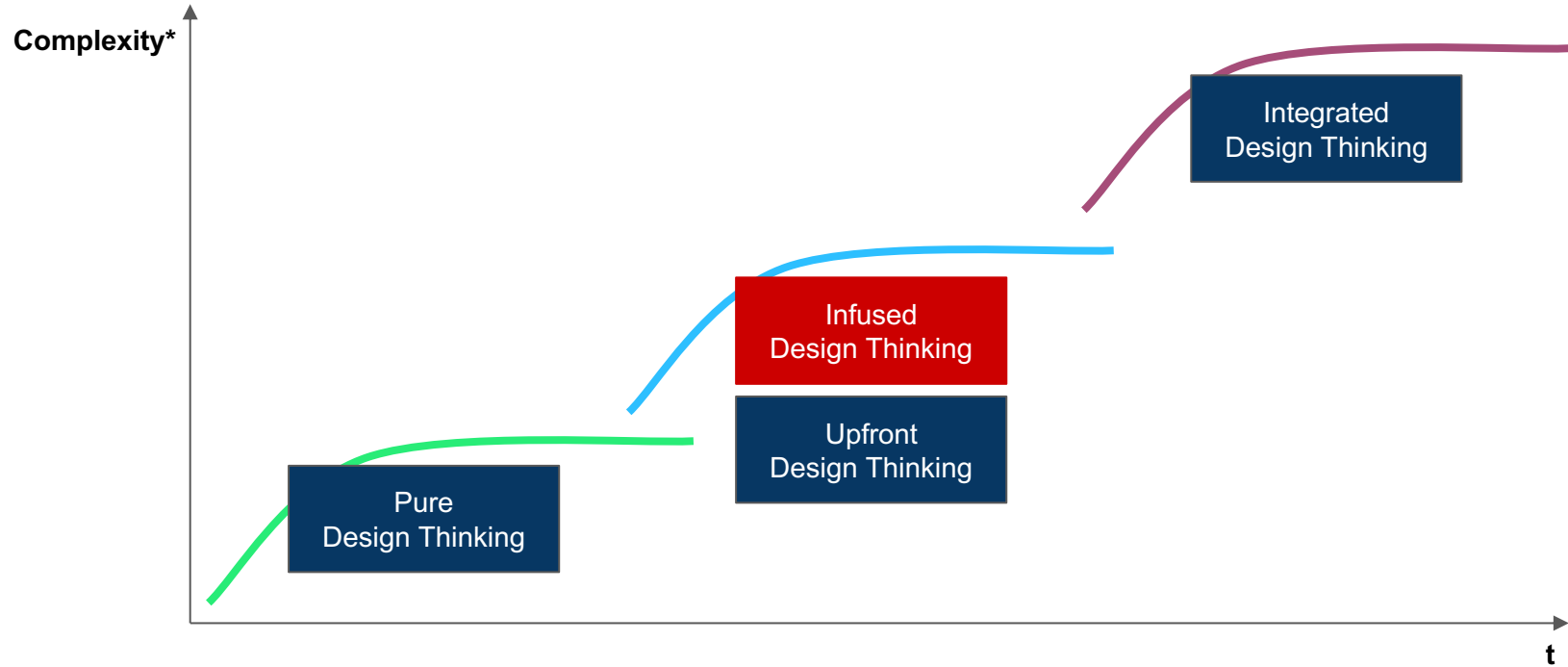
- 12 qualitative interviews
 - 1 quantitative questionnaire
 - 2 Personas
 - 4 prototypes
- User story definition via project team
 - User stories and high resolution prototypes are handed over to implementation



Take-Aways

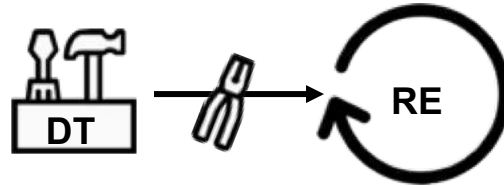
- What works:
 - Fostering a collaborative working environment
 - Fostering a failure tolerant culture through rapid prototyping and continuous experimentation
 - Broadly validated key features / user stories
- Open challenges:
 - Final deliverable via user stories and HighRes prototype
 - No further feedback cycles
 - Potential starvation of results with no implementation (or control over it)

Evolution of Design Thinking and RE

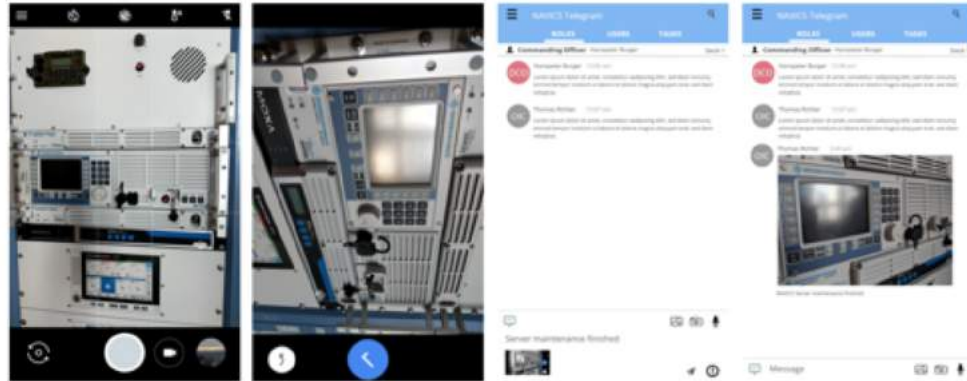


* Note: one is not per-se "better" than the other; everything has its place

Infused Design Thinking



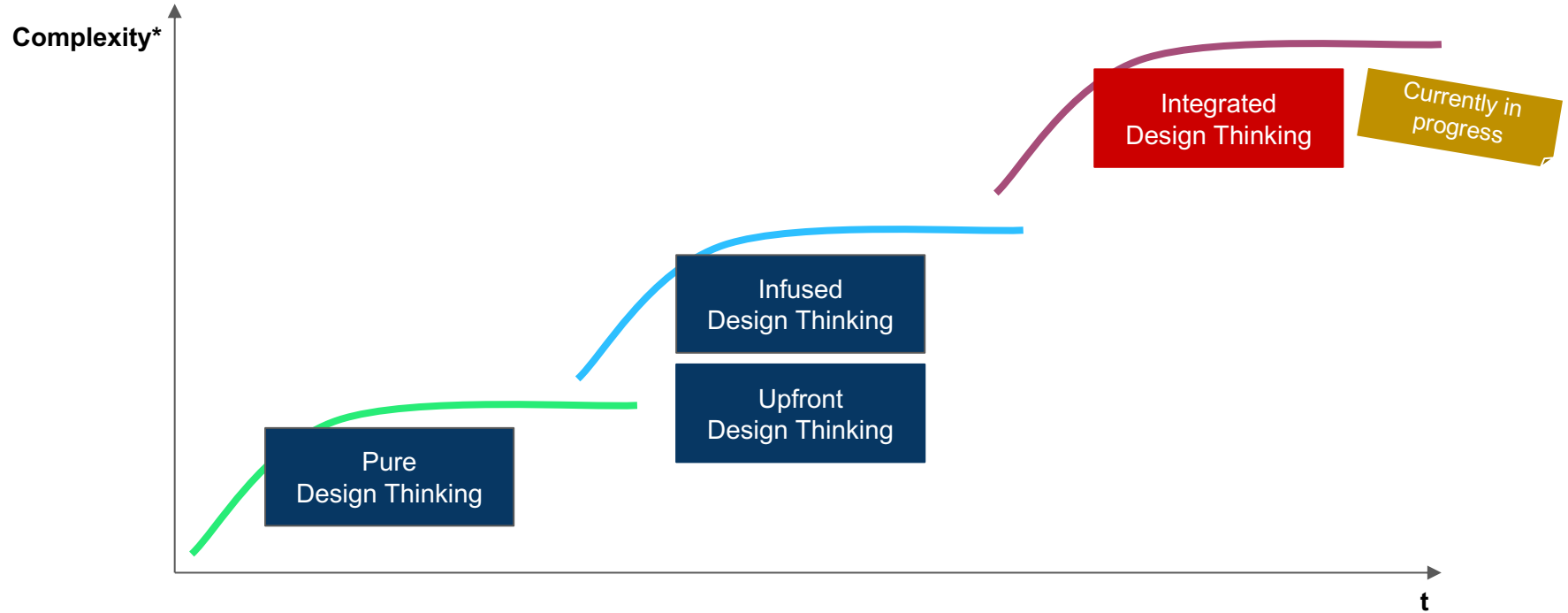
- International Electronics group
- Headquarter in Germany, 10.500 employees
- Needfinding and Prototyping Infusion



Take-Aways

- What works:
 - Fostering a broader collaborative working environment
 - Integrating creative idea generation in context of a software development life cycle
- Open challenges:
 - No further development-critical artefacts, e.g. NFRs, technical constraints, or data models
 - Still no seamless and sustainable integration of DT methods into software development activities
 - Limited learning curve for reuse in further projects

Evolution of Design Thinking and RE



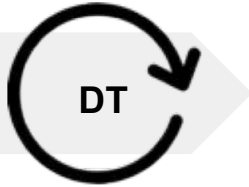
* Note: one is not per-se "better" than the other; everything has its place

Integrated Design Thinking approach



- German Utility Company
- Problem statement: Development of an offering to boost photovoltaik sales
- Team: multidisciplinary
- Design Thinking process: 3 months
- Integrated approach: 12+ months

Design Thinking 3 months



Full Design Thinking
Approach

- 10 expert interviews
- 22 interviews with possible users (homeowners and craftsmen)
- 40 insights collected
- 50 ideas generated
- 12 value propositions for both craftsmen and customers
- 3 Personas
- 12 low resolution prototypes tested with both stakeholder groups
- 1 final high resolution prototype (not yet tested)

**Final
(non-tech.)
prototype**

Revised vision:
Home Improvement
Platform

Design Thinking
3 months



DT@Scrum
12-x months

Full Design Thinking
Approach

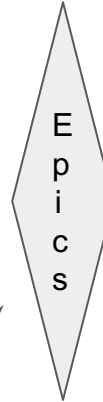
Design Thinking Toolbox: User Testing & Prototyping;
Product Owner Role is inhabited by Design Thinking Team

- 10 expert interviews
- 22 interviews with possible users (homeowners and craftsmen)
- 40 insights collected
- 50 ideas generated
- 12 value propositions for both craftsmen and customers
- 3 Personas
- 12 low resolution prototypes tested with both stakeholder groups
- 1 final high resolution prototype (not yet tested)

**Final
(non-tech.)
prototype**

Revised vision:
Home Improvement
Platform

SCRUM
Sprint 0
*Sprint
backlog*



SCRUM
Sprint 1
*Sprint
backlog*

User
stories

Flow
Charts

Non-tech
Prototype

SCRUM
Sprint 2
....
*Sprint
backlog*

SCRUM
Sprint n
.....

MVP1

Take-Aways

- What works:
 - DT as a structured, domain-agnostic approach to requirements elicitation
 - Extended arm into wicked problems and re-define actual problems and SW system context
 - Sufficiently correct and complete key features / user stories via continuous experimentation and testing of non-technical and technical prototypes
 - Clear roles and responsibilities
- Currently open challenges:
 - Difficulty in integrating further RE-specific artefacts, e.g. NFRs, technical constraints, or data models

How can we efficiently integrate DT and RE?

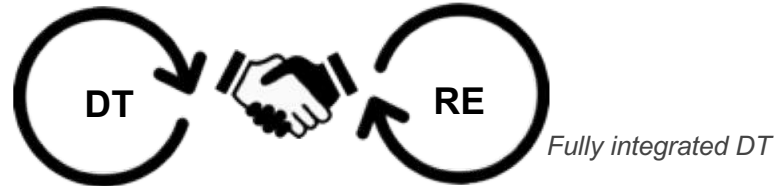
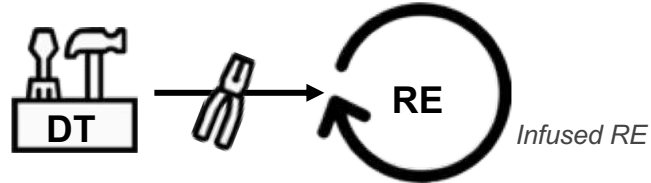
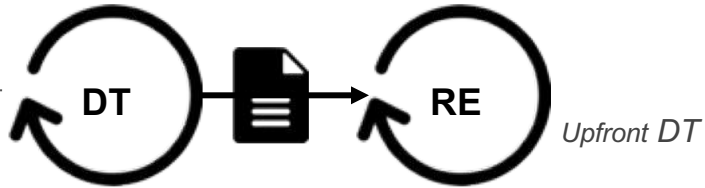
Reminder



Towards a pragmatic approach to human-centric RE



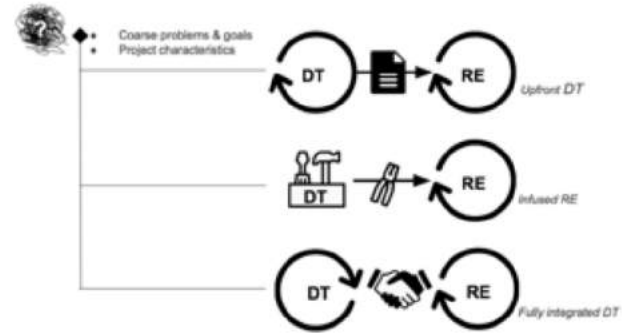
- Coarse problems & goals
- Project characteristics



Open research challenges

General Challenges

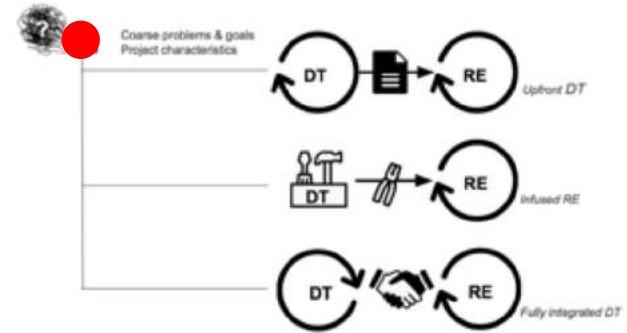
- Principles: Which principles and approaches in DT can be found in more holistic human-centred software development approaches and how do they differ?
- Boundary objects: How can artefacts with similar purposes, but different forms, be integrated?



Open research challenges

Project Influences

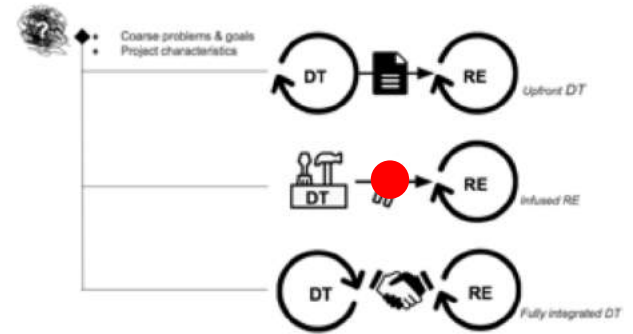
- How can problems be efficiently classified?
- What are typical project situations which influence the choice of a strategy?
- How do these situations and the class of systems influence the choice of a strategy and single methods?
- How can these situations be characterised and assessed in early stages of a project (with which confidence)?



Open research challenges

Method adoption

- Which methods in DT can be found in / reused for other software engineering disciplines (e.g. HCI, TDD)?
- How do these methods differ? How can they be integrated?



Open research challenges

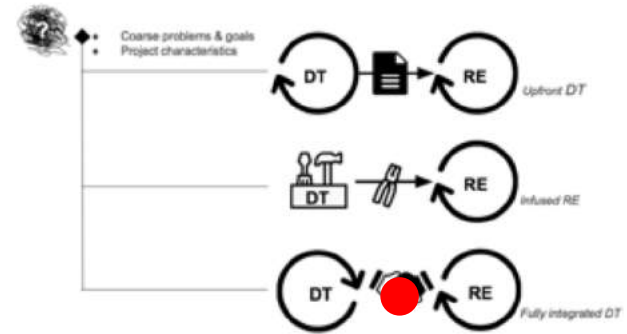
Interface and Operationalisation

Interfaces

- How can artefacts, roles, and methods be seamlessly integrated?
- Which artefacts do overlap? Are shifts in roles and responsibilities necessary?
- How can milestones be efficiently defined?

Operationalisation

- How can resulting processes be integrated (into the overall life cycle) - for instance SCRUM?
- How can resulting processes be tailored?



Outline

1. Design Thinking in a Nutshell



2. Design Thinking for Requirements Engineering

3. Final discussion and closure