

# SP3 – Value-Oriented Strategy to Detect and Minimize Waste

Javier Gonzalez-Huerta (leader), Krzysztof Wnuk  
+ the team

# Javier Gonzalez Huerta

(Researcher and practitioner)



- Assistant Professor at BTH in Sweden in Software Architectures and Quality, Software Development (2016-)
- Worked more than 10 years in industry before joining academia and worked with industry the entire academic carrier (Spotify, Ericsson, Trisotech, Rolls Royce....)
- Principal Investigator in the SHADE project (A Value-Oriented Strategy to Handle Software Asset Degradation)
- 10 years in research
  - PhD from Universitat Politècnica de Valencia (Spain) in 2014.
  - Post-Doctoral researcher at the Université du Quebec à Montréal (UQÀM), Canada
  - Assistant Professor at BTH since 2016

# What is the problem?

## Problem



- We work under constant **time pressure**
- Features need to be delivered **on time** to meet the market needs
- Being late means **loosing money**
- So we work **Agile to deliver** fast and frequent
- We make **sub-optimal** decisions that generate waste and overhead and that prevents us from **creating value** (financial, customer, internal business or innovation value)

# What is waste and overhead?



- **Waste** - activities that consume time, money or space without producing relevant value
- **Overhead** - efforts put on activities that:
  - can be avoided by improving the way of performing them,
  - not produce any value per se, but **enables value creation** indirectly
- Waste and overhead should be identified and can be controlled
- **Value Creation** should be the main identification driver

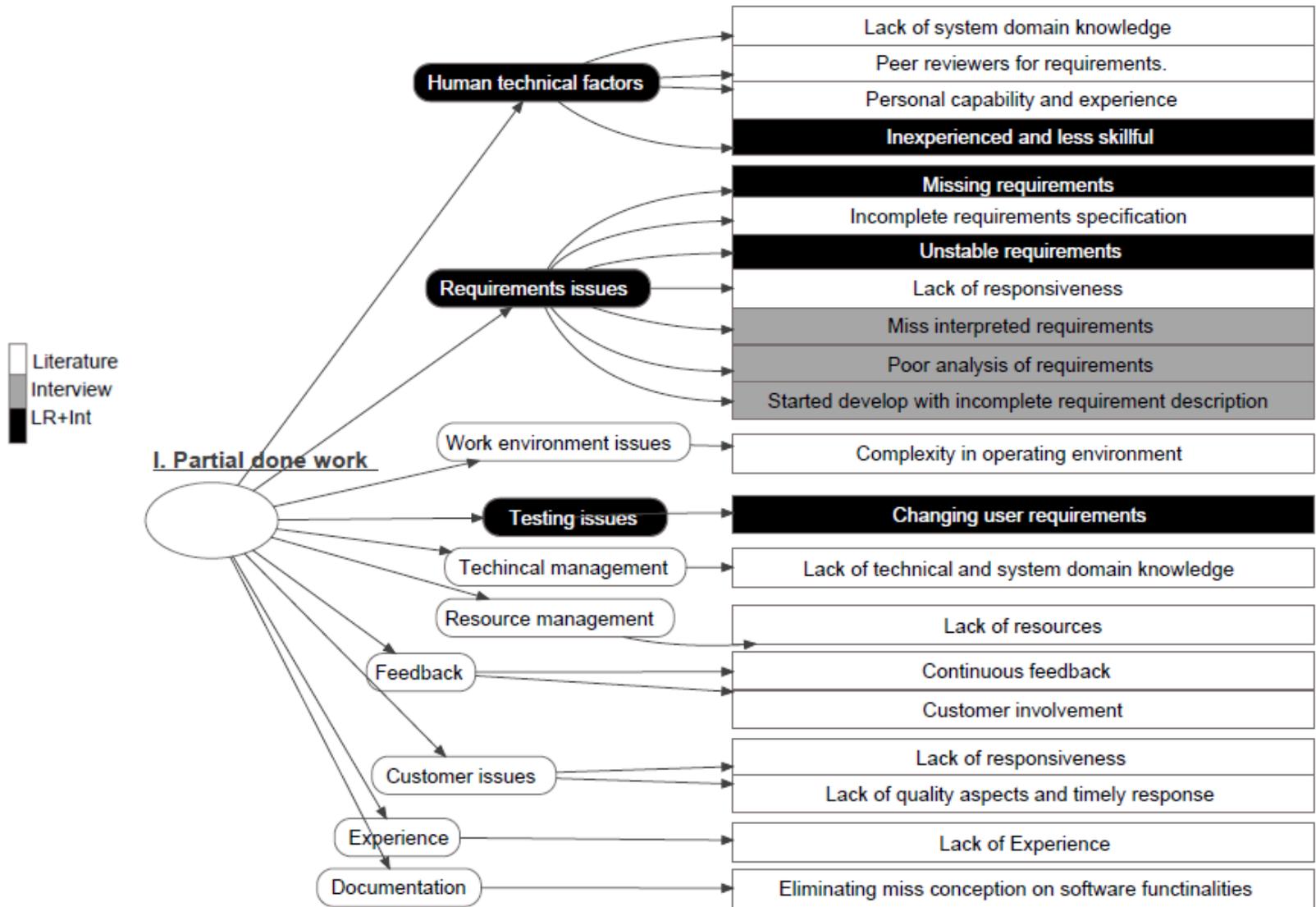
# Some Examples of Waste

- Waste #1 - Partially Done Work
- Waste #2 - Extra Features
- Waste #3 - Relearning
- Waste #4 - Handoffs
- Waste #5 - Delays
- Waste #6 - Task Switching
- Waste #7 - Defects



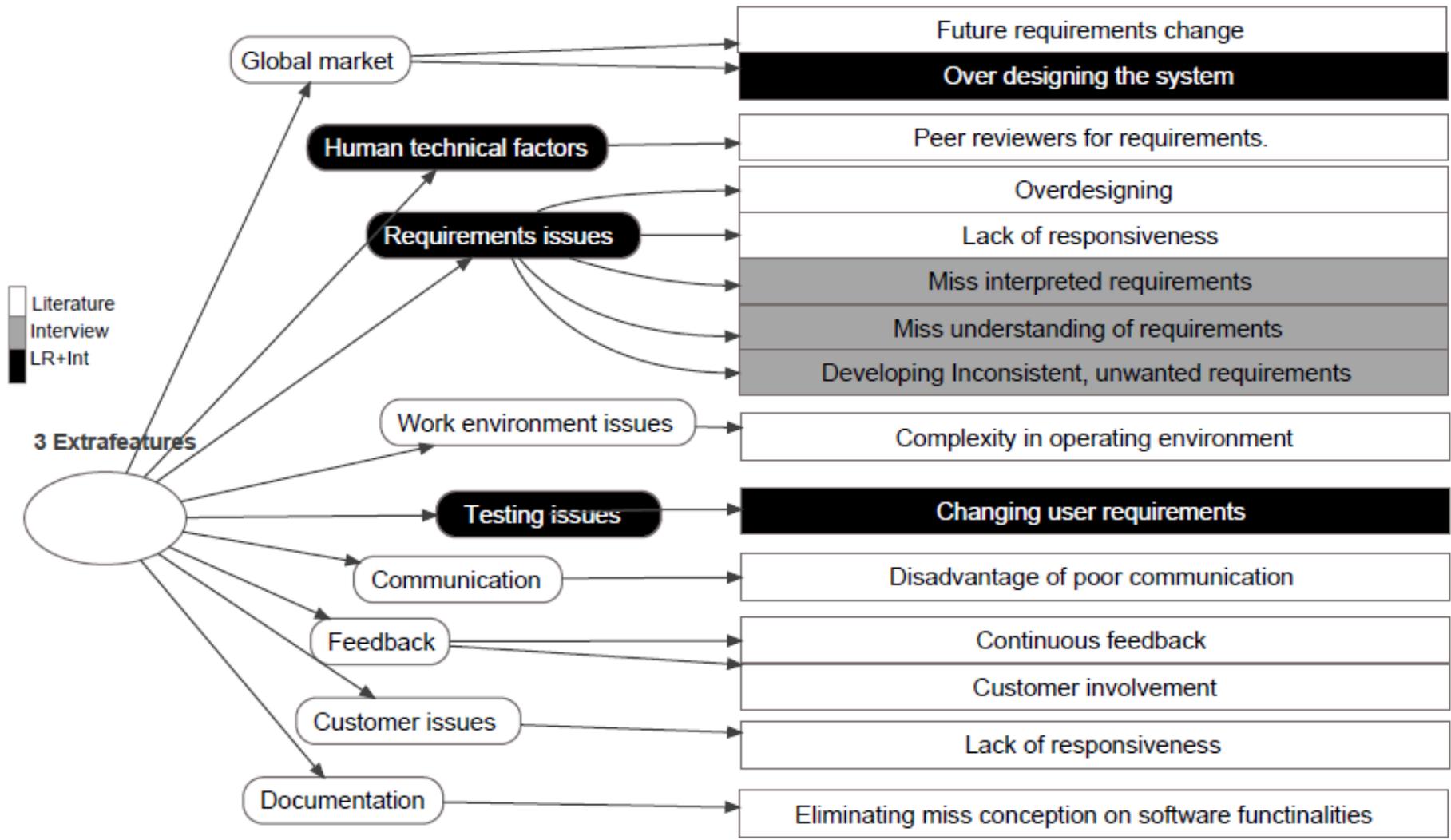
# Waste #1: Partially Done Work

# Problem

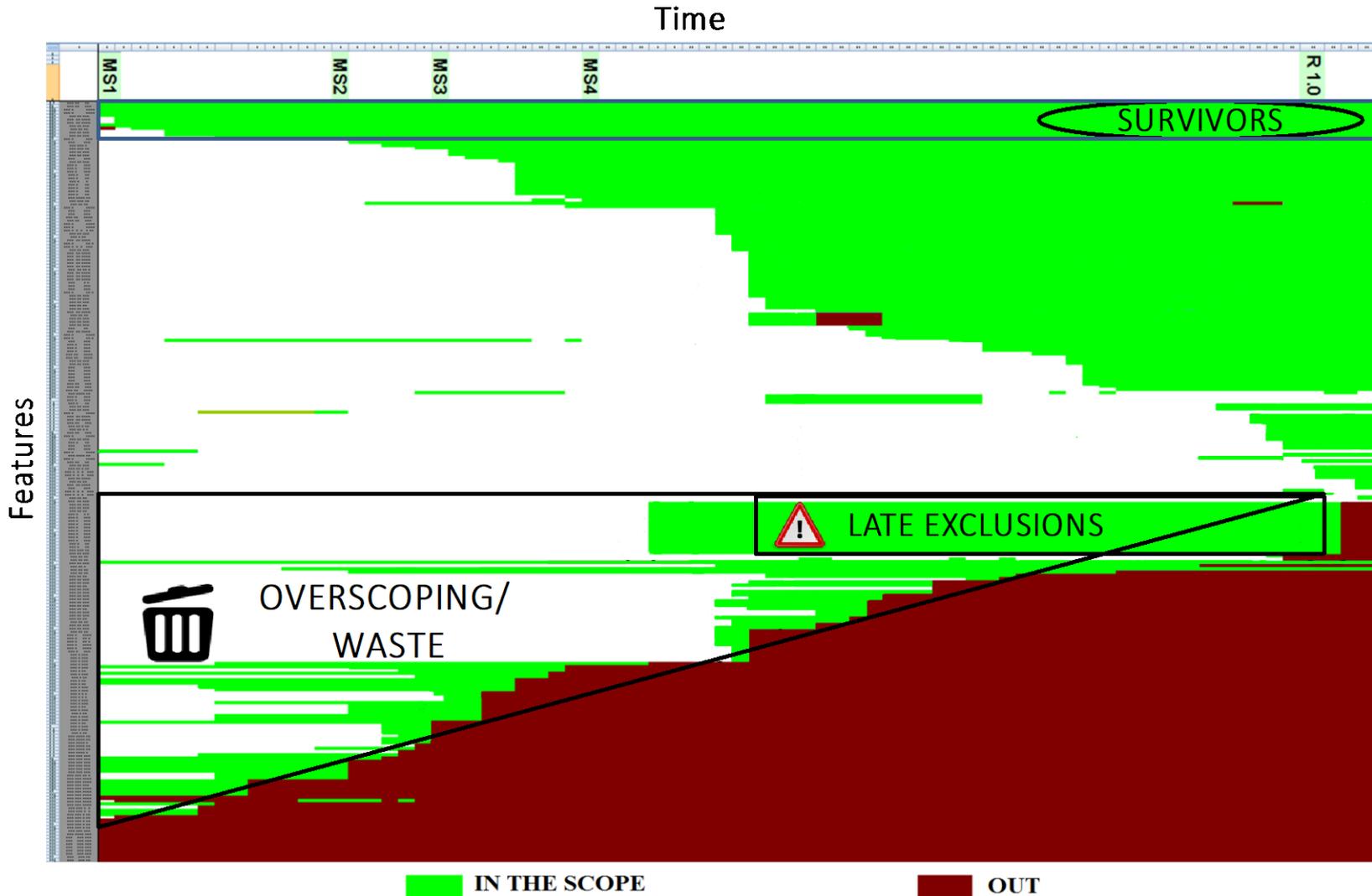


# Waste #2: Extra Features

# Problem

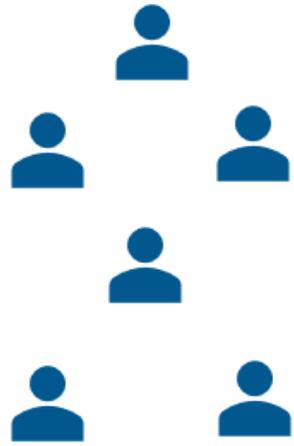


## Example with ~600 features

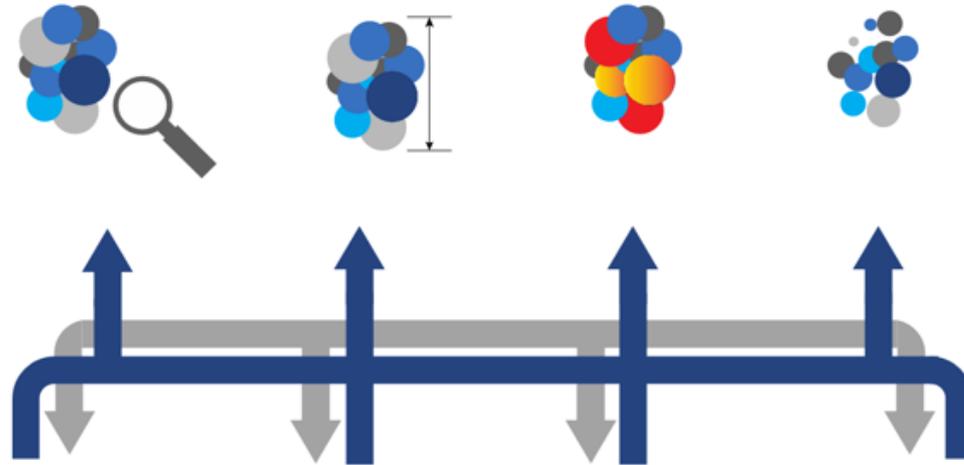


# Overview

# Envisioned Solution



Identification    Measurement    Prioritization    Mitigation

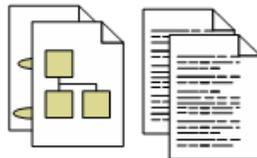


Value

Overhead

Waste

Requirements and design documentation



Software Code



Test Cases



CVS



Issue-tracking systems



- Create and empirically evaluate a catalogue of types of waste for inception, realization and evolution of software intensive products as well as associated metrics
  - 1) Identify what waste category is the most critical for the company
  - 2) Study the associated factors (from our experience)
  - 3) Identify the metrics that can be used and possible automation
  - 4) Identify roles responsible for taking actions to minimize waste

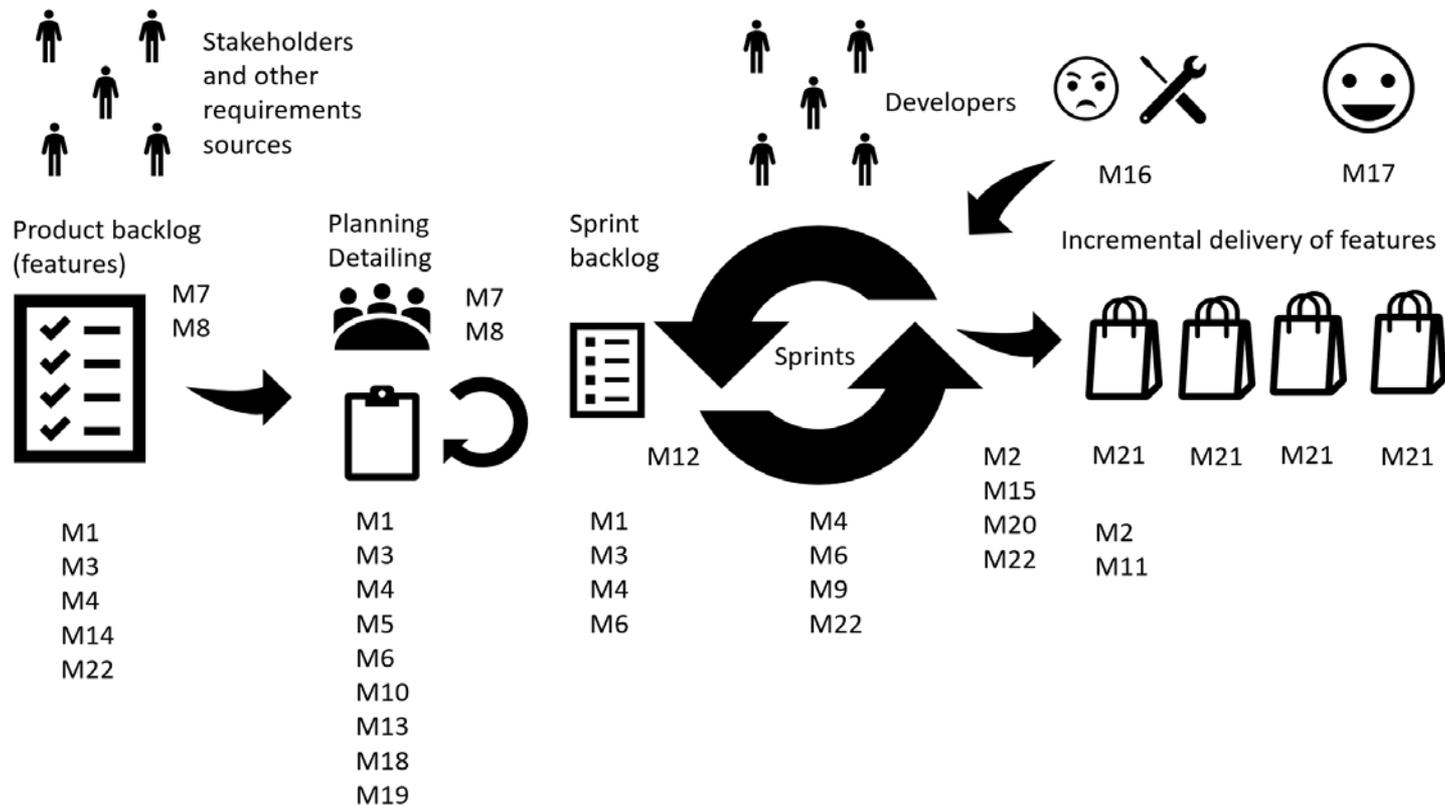


- Introduce waste and overhead detection solutions
  - Waste indicators are introduced and continuously collect and analyse data
  - Inception, realization and evolution support
  - Alerting (notification) system is introduced to notify relevant stakeholders and decision makers



# Remove organizational bottlenecks

- Explore software process and organizational bottlenecks that substantially contribute to waste aspects.



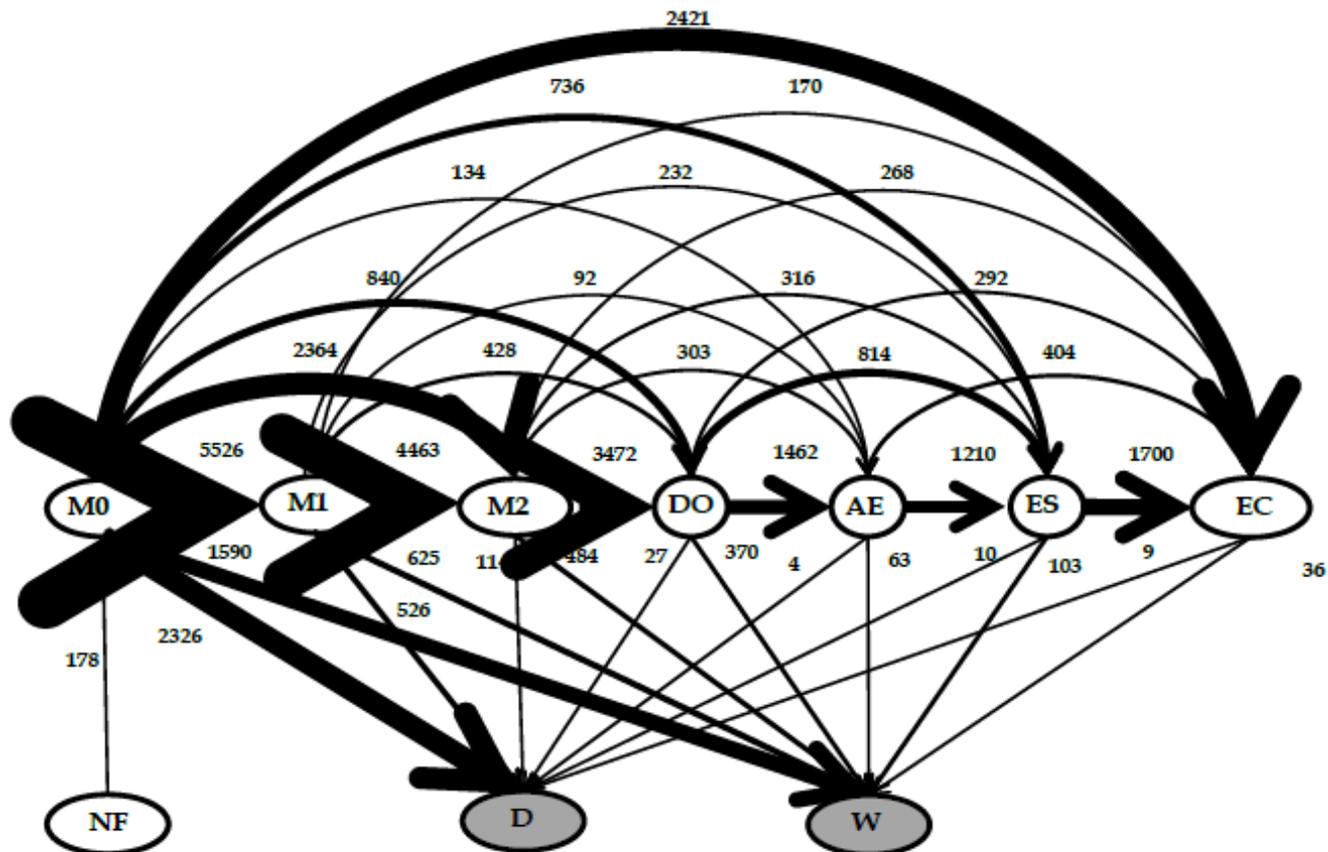
# Create and enforce process routines

- Create and enforce process routines that help identify, prioritize and handle waste based on value and managerial strategy
  - Alerting (notification) system is the first step towards introducing process routines (e.g. regular review meetings)
  - Engineering practices should also be revisited and updated based on waste indicators (e.g. craftsmanship principles could be introduced to enrich Agile)



# Remove organizational bottlenecks

- Explore software process and organizational bottlenecks that substantially contribute to waste aspects.



Questions?

SOFTWARE ENGINEERING  
**R@THOUGHT**



MaxKompetens



**ERICSSON**  
E OF TECHNOLOGY



Handelsbanken



QTEMA



**FORTNOX**  
Småföretagens bästa vän



**CGI**

TimePeopleGroup