

# Value and waste in software engineering

Matthias Galster

University of Canterbury

Christchurch, New Zealand

[mgalster@ieee.org](mailto:mgalster@ieee.org)



**INFORMATION  
AND  
SOFTWARE  
TECHNOLOGY**

Information and Software Technology  
Supports *open access*

Articles & Issues ▾ About ▾ Publish ▾ Order journal ▾

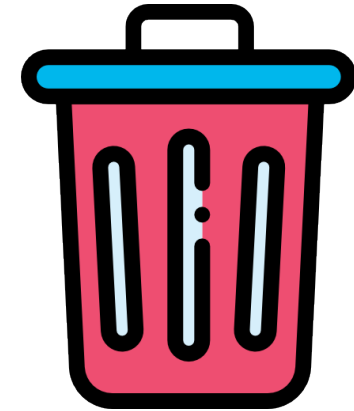
# Waste and Value in Software Engineering

Edited by Michael Felderer, Matthias Galster, Clemente Izurieta, Carolyn Seaman

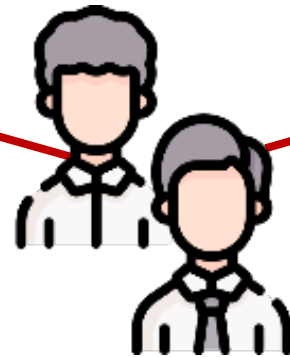
Links to studies mentioned in talk: [shorturl.at/lyHNR](https://shorturl.at/lyHNR)



Value



Waste

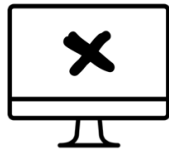


Engineers



# What is waste?

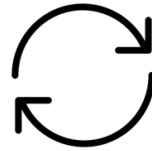
Activities that consume resources but create no “value”



Building the wrong feature or product



Mismanaging the backlog



Rework



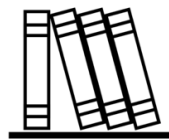
Unnecessarily complex solutions



Extraneous cognitive load



Psychological distress



Knowledge loss



Waiting



Ineffective communication

Understanding waste + causes empowers us to address it explicitly



2-year 5-month participant observation, interviews at software consultancy [Sedano et al.]

# Waste - bad code reviews



What are bad code review practices and resulting waste?

Lack of review  
Looks good to me



Rework

Ping-pong



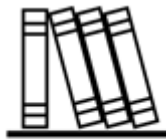
Ineffective communication

Missing context  
Large change sets



Extraneous cognitive  
load

Review buddies



Knowledge loss

Sleeping



Waiting

➔ Plan and communicate expectations for review



Literature review, survey, interviews, ~226k code reviews from 8 OSS [Doğan & Tüzün]

# Waste - technical debt



What to fix? Who should fix it?

Focus dedicated TD removal on types less likely to be fixed, e.g., design debt

- Other types, e.g., code debt, defect debt, are self-fixed

Code debt, defect debt: if urgent, ask those who introduced it

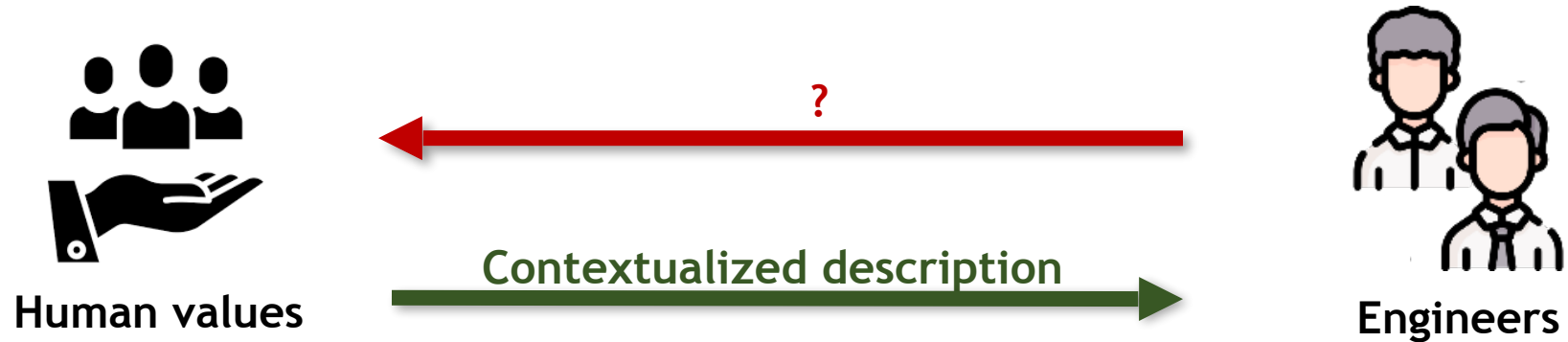
- Design debt: ask others (not self-fixed, survive longer)



44k commits from 20 Python and 16 Java projects [Tan et al.]

# Value - human values?

How can we integrate human values in software?



Dignity

“Maintain honour + respect for users”

(e.g., what user information is kept, how is it shared; how can users unsubscribe)

Inclusiveness

“Facilitate different origins, languages, cultures, knowledge”

(e.g., what languages are supported in a multi-cultural usage environment)

Etc.

Etc.



1,097 issue discussions of Firefox, K-9 Mail, Signal Private Messenger [Nurwidyanto et al.]

# Summary: create value + reduce waste



We know what review smells, how they cause waste and why.  
Use review smells to plan + communicate expectations for reviews (and training).



We know what kinds of TD get self-fixed, e.g., code debt.  
Focus dedicated activities on those that don't, e.g., design debt.



We know how to conceptualize human values for software engineering.  
Use them to identify product requirements and build responsible software.



Want to know more? Want to collaborate? Email [mgalster@ieee.org](mailto:mgalster@ieee.org)



**Acknowledgments:** Contributors to Special Issue, co-guest editors; Greg Wilson, Michael Hoyer