

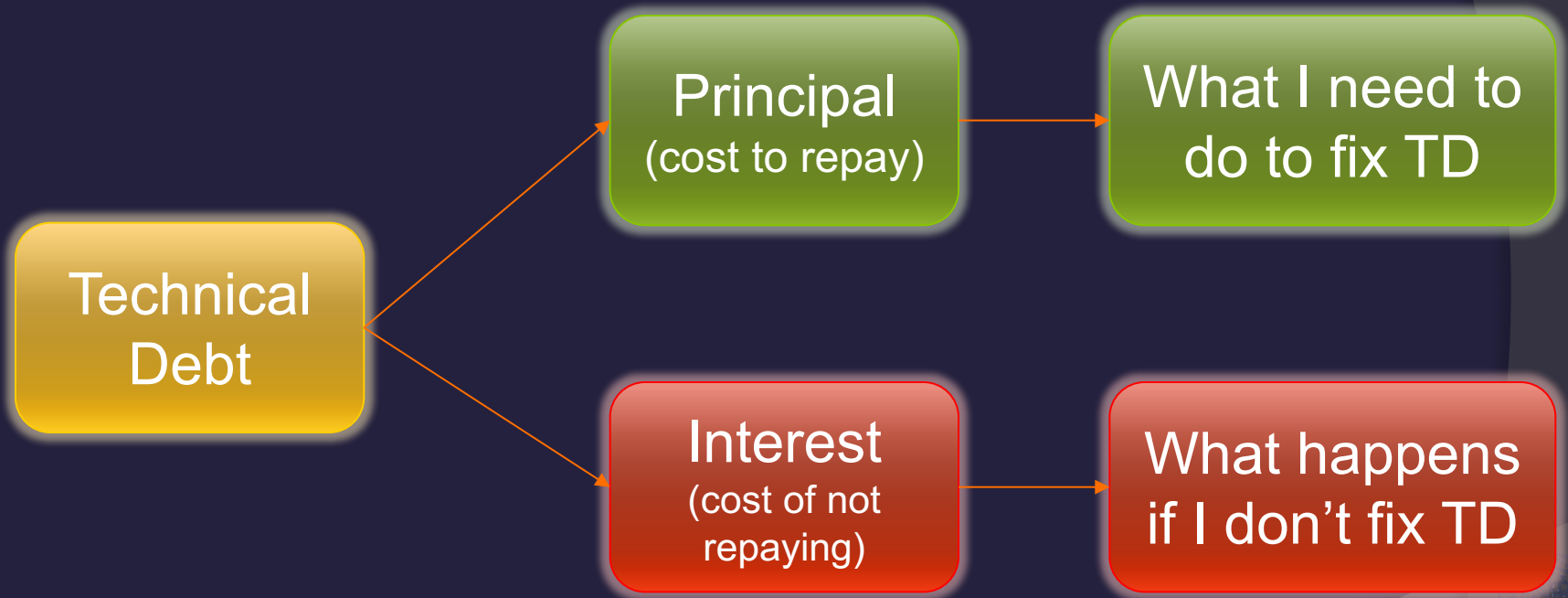
Antonio Martini
Jan Bosch

Chalmers University of Technology
Gothenburg, Sweden

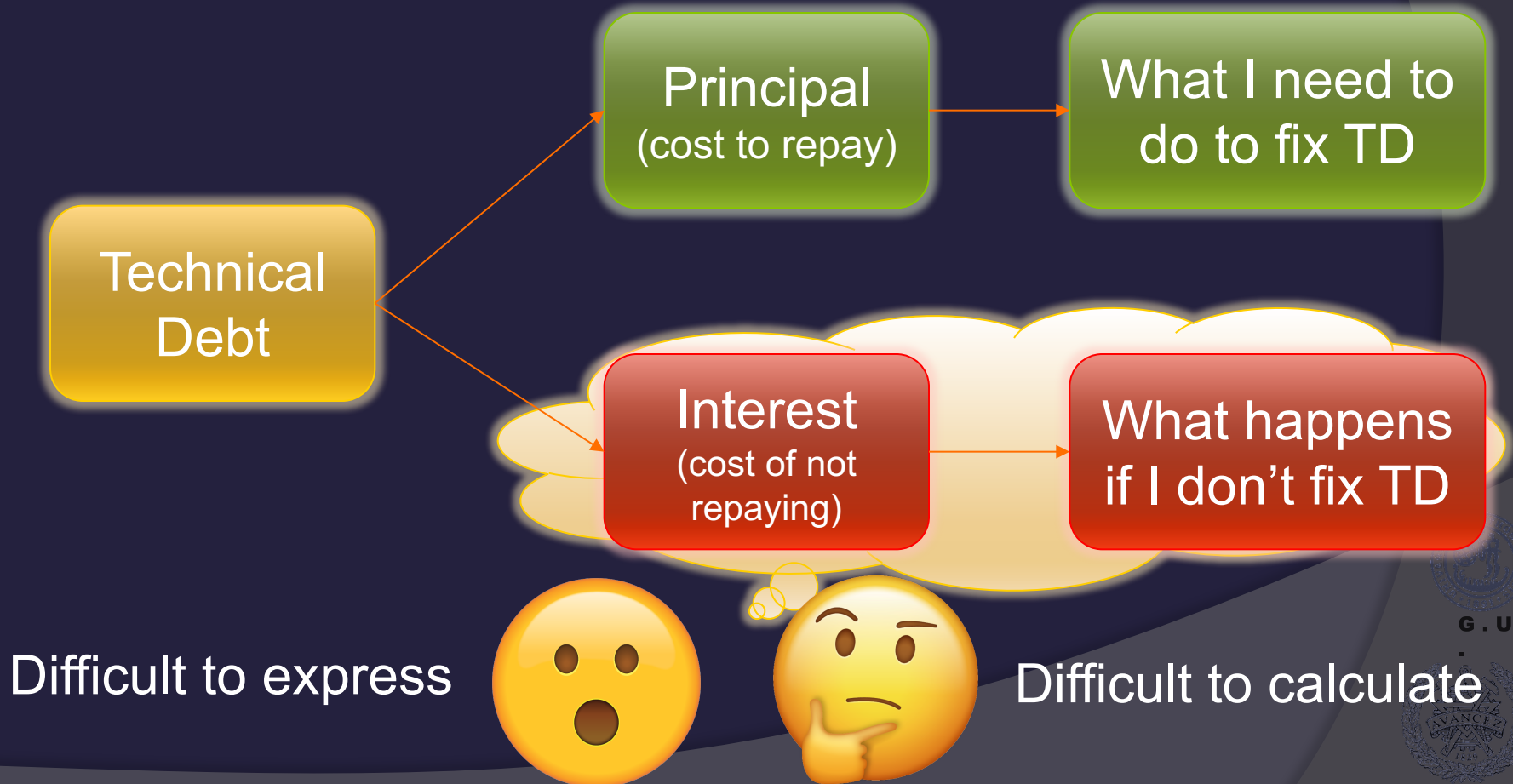
MTD 2017
Cologne, Germany
2017-05-22

THE MAGNIFICENT SEVEN: TOWARDS A SYSTEMATIC ESTIMATION OF TECHNICAL DEBT INTEREST

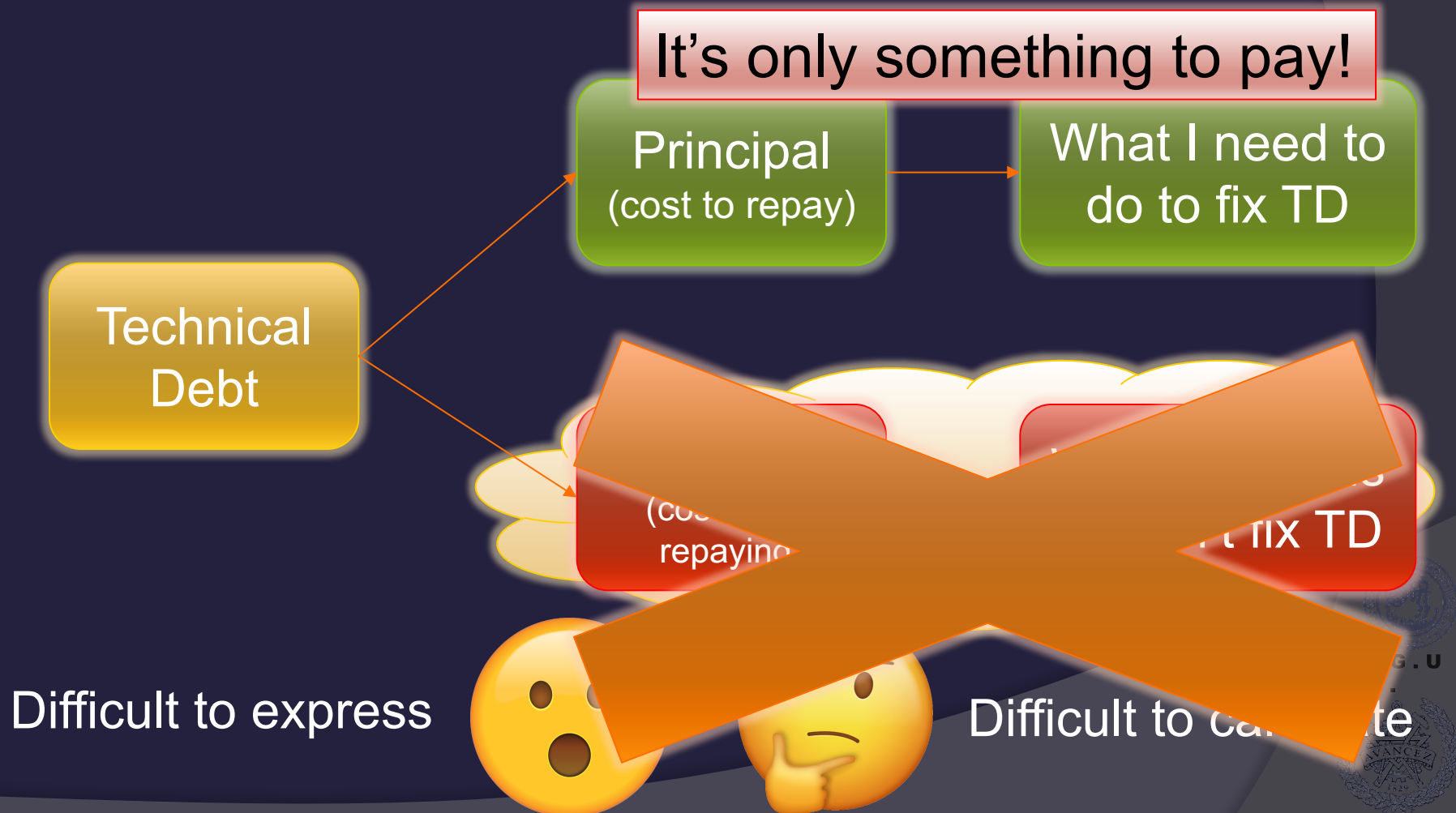
Recap: TD, principal, interest



Current challenges about the interest



Problem: what's the gain of repaying TD?



Challenges with the interest

- ⦿ Difficult to express
 - Business is not the main **focus** of developers
 - Interest **changes** over time
- ⦿ Difficult to calculate
 - **Subjective**
 - Present AND **Future** interest should be calculate
 - Future = not measurable
- ⦿ We need a less subjective and systematic approach
- ⦿ **RQ: how can we calculate interest in a systematic way?**

How to calculate interest?

- ◎ To track TD, teams usually use* **items**:
 - Backlog
 - Static analyzer tool
 - Generic documentation
- ◎ **None of these focus on calculating the interest!**
- ◎ Calculation: Probability*Severity
 - Probability
 - **Severity** = (negative impact)
 - We called it Negative Impact of TD

- Y. Guo *et al.*, “Tracking technical debt—An exploratory case study,” in *Software Maintenance (ICSM), 2011 27th IEEE International Conference on*, 2011, pp. 528–531.
- A. Martini, T. Besker, and J. Bosch, “The Introduction of Technical Debt Tracking in Large Companies,” in *accepted at APSEC 2016*, Hamilton, New Zealand, 2016.

How to calculate the severity of interest?

- Previous study* on the interest suggests that the interest involves **7 factors**
 - Reduced development speed
 - Bugs related to TD items
 - Degraded System Qualities
 - (Other than maintainability and evolvability)
 - Context-related extra-costs
 - E.g. future lack of competences
 - Frequency of the interest
 - Spread in the system
 - Users affected
- ...and their **growth in the future**

* A. Martini and J. Bosch, "An Empirically Developed Method to Aid Decisions on Architectural Technical Debt Refactoring: AnaConDebt," in *Proceedings of the 38th International Conference on Software Engineering Companion*, New York, NY, USA, 2016, pp. 31–40.

Is the overall severity represented by these factors?

- Severity = $f(\text{Factors})$
- Is there a way to calculate $f()$?
- We designed a tool: AnaConDebt*
 - User grades the 7 factors and the outcome is a **severity** value

The screenshot shows the AnaConDebt tool interface with seven factors for grading software debt. Each factor has a set of radio buttons and a corresponding color-coded bar indicating the selected grade.

- Is the development slowed down?** (Green bar, "emerging" selected)
- How many bugs are related to this issue?** (Orange bar, "many" selected)
- Is the issue causing one or more non-functional requirements to be not met?** (Yellow bar, "moderate degradation" selected)
- Is the issue causing other extra-efforts?** (Red bar, "very high extra-effort" selected)
- What is the frequency of the issue?** (Yellow bar, "bi-weekly" selected)
- Is the problem isolated?** (Yellow bar, "affects the whole component" selected)
- Is the issue causing troubles for many users?** (Green bar, "the team is affected" selected)



Automatic
calculation

10



1

* www.anacondebt.com

Does it work?

◎ Evaluation

- Evaluated with
 - 9 TD items
 - Different kind and perceived impact
 - We tried to maximize the variance
 - 3 teams
 - senior developers (Team Architects)
 - 3 product owners
- We especially tried to avoid
 - Cognitive Bias
 - Especially confirmation bias
 - Construct Validity threats

Evaluation

1. TD
training
(researcher)

2. How much is
the severity?

3. Intuitive
estimation

10



1

= ?

4. Grade the
7 factors

5. Automatic
calculation

10



1

= ?

6. Look at the
assessment of
the **7 factors**?

7. Intuitive
assessment of
severity

10



1

Developer

Product O.

Automated

Similarity of interest severity

- ⦿ (details in the paper)
- ⦿ Avg. absolute distance between
 - Tool and Developers: 0.075
 - Tool and Product Owners: 0.121
 - Developers and Product Owners: 0.105
- ⦿ After we asked if the developers wanted to change their value
 - Tool and Developers: 0.028



G . U



CHALMERS

Qualitative sessions

Qualitative sessions

- Developers together with Product Owners
- Results:
 - Stakeholders were **surprised** by the “**precision**” of the tool
 - The tool is useful for **medium-large items**
 - Otherwise overhead
 - **Less subjective** assessment of TD
 - Interest Assessment tool **useful**
 - but **not** necessarily for a **single team** (small projects)
 - when **several teams** work on the same large codebase or project
 - because it gives a more **standardized** way of assessing TD in large organizations
 - **Product owners** were keen in **adjusting** their severity impact after the results were shown

Summary

- ◎ The tool seems quite precise
 - The 7 factors capture the severity of the interest
 - $f()$ seems to have been systematically assessed
- ◎ The tool served as a communication means
 - To reason about TD
 - To assess TD
 - To align stakeholders
- ◎ Agile
 - Tool supports iterative communication on Technical Debt
 - Lightweight: 5-10 minutes to assess an item
 - Assessing TD interest helps understanding:
 - Which upfront design should not be neglected
 - Or it helps refactoring on-the-go
 - When we have more information

Future work

- ◎ More cases
 - Already some new entries
 - Formula not related to single company
- ◎ Analysis of frequency weight of the 7 factors
- ◎ Embed this approach with other SE approaches
 - E.g. project management
 - Agile process
 - ...

...to be continued...

antonio.martini@chalmers.se

Discussion



Debt Free
Zone