

An investigation of Technical Debt in Automatic Production Systems

Terese Besker*, Antonio Martini*, Jan Bosch* and Matthias Tichy**

*Computer Science and Engineering, Software Engineering
Chalmers University of Technology, Göteborg, Sweden

**Institute of Software Engineering and Programming Languages
Ulm, Germany

MTD 2017

Ninth International Workshop on Managing Technical Debt
In conjunction with XP 2017

Automatic Production Systems

Used in a high variety of industrial sectors such as

- automated packaging
- production of chemicals
- production of transportation vehicles



Contain mechanical, electrical and software parts and are jointly developed by engineers from the different disciplines.

Research Questions

- RQ1: How much effort do companies developing automatic production systems spend on paying the interest (extra-costs)? On which kinds of Technical Debt?
- RQ2: How much effort do companies spend in managing Technical Debt in automatic production systems?
- RQ3: How much is Technical Debt known and tracked in a company developing automatic production systems?
- RQ4: How familiar are practitioners in companies developing automatic production systems with the Technical Debt terminology?

Research Design

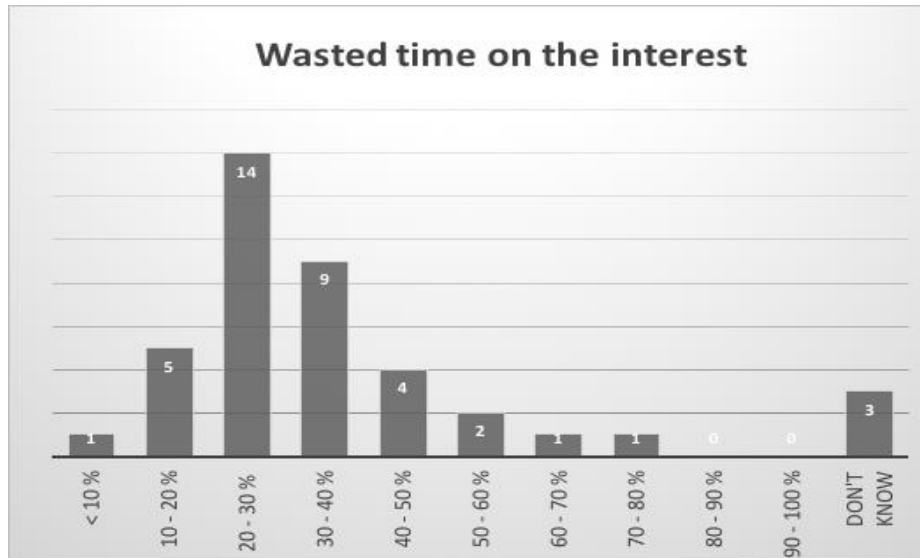
- A combination of quantitative and qualitative research approaches is used.
- The research is conducted by using methodological, source and observer triangulation.
- Survey at an international APS company in Sweden, with several sites t several parts in Europe.
- 40 Complete answers from:
 - 4 Project Managers
 - 3 Product Managers
 - 3 Software Team leaders
 - 3 Line Managers
 - 1 R&D Manager
 - 6 Software Architects
 - 20 Developers
- Qualitative follow-up interviews (8 managers)

Online Web-Survey

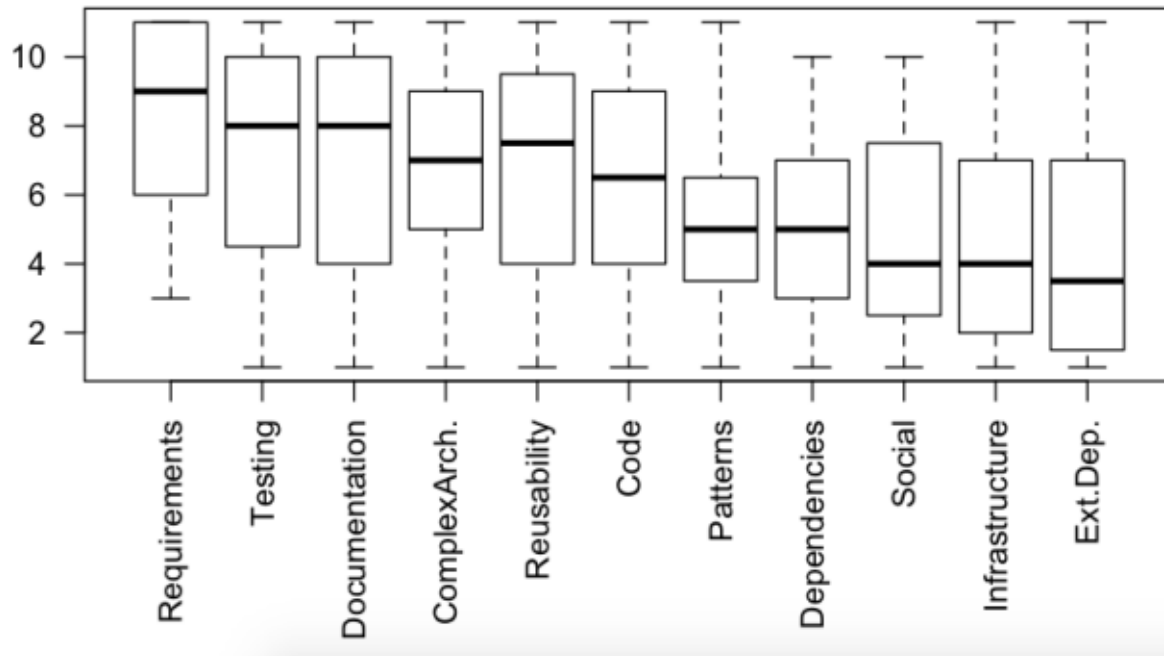
- Q1: “Which of the following challenges generate the most negative impact on your daily software development work? Please rank them from 1 to 11.”
- Q2: “How much of the overall development time is wasted because of these issues?”
- Q3: “How much of the overall development effort is usually spent on Technical Debt management activities?”
- Q4: “I track (using tools, documentation, etc.) Technical Debt in our system.”
- Q5: “If you track Technical Debt in your project, what kind of tool(s) do you use?”
- Q6: “How familiar are you with the term "Technical Debt"?”

Effort spent on paying interest and on different types of TD

The company waste on average **32%** of the development process



Effort spent on *paying interest* and on different types of TD



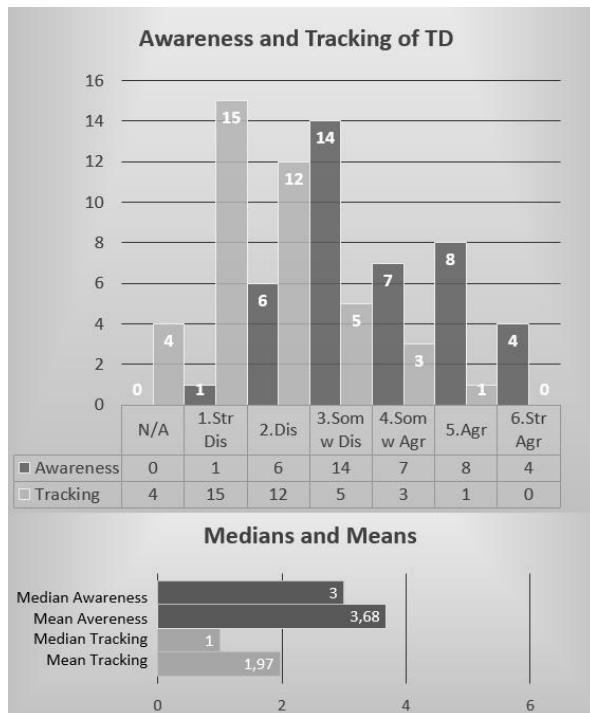
Effort spent on *Managing* TD

Management activities necessary to keep Technical Debt at bay:

- Repaying the debt (refactoring the code),
- Finding the debt
- Measuring it
- Understanding the issues
- Managing, keeping track and communicating

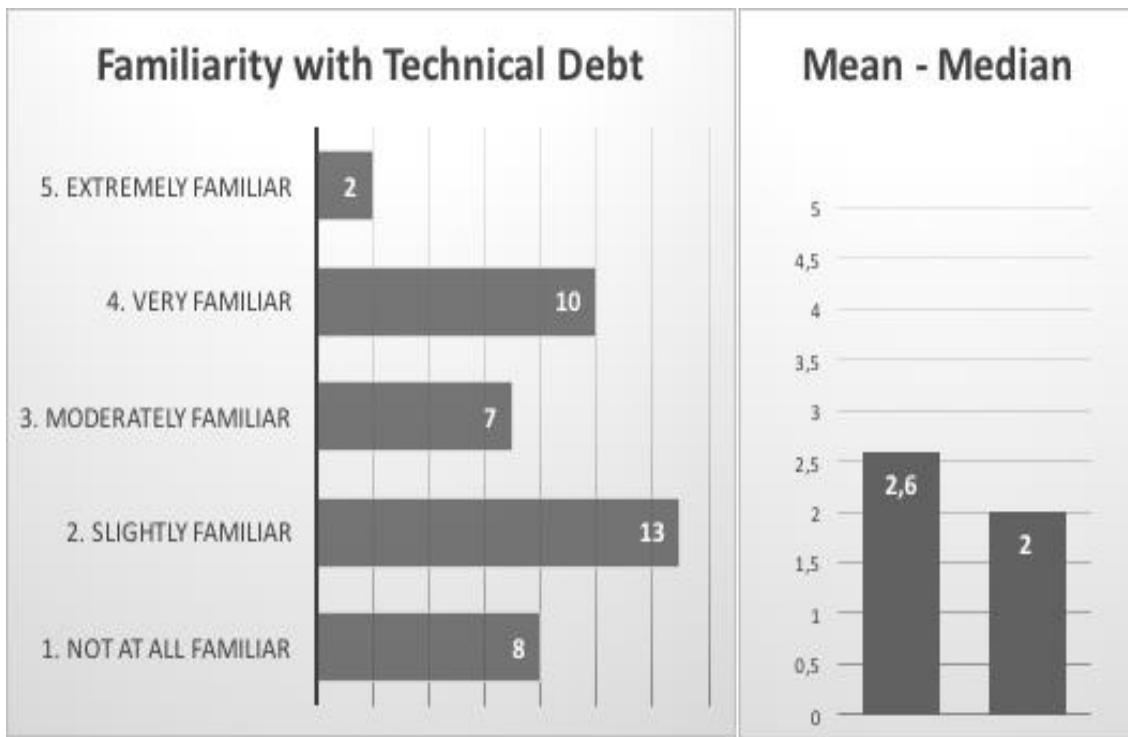
This additional managerial extra-cost is, on average,
24 % of the development time

Tracking and Knowledge about TD



- The practitioners are in general moderately aware of how much Technical Debt they have in their systems.
- As for the systematic tracking, this does not happen a lot.
- This shows that Technical Debt is moderately known even without tracking it.
- The few tools used for tracking Technical Debt, was Jira and Excel.
- The current tools and practices don't support a better strategic management of Technical Debt.

Familiarity with the Technical Debt terminology



Implication for industrial practices

- We see a need for practical improvement of Technical Debt management consisting of:
 - Raising the awareness of Technical Debt
 - Increasing its tracking
 - A better visualization and strategic management
 - Need for introduction of Technical Debt -specific tools



Theoretical implications

- Technical Debt is a concept applicable to companies developing automatic production systems, and not only to software companies
- These results reveal that Technical Debt in automatic production systems companies is a big challenge
- In comparison with earlier studies, we can see generic software companies and companies developing automatic production systems is quite similar
- It seems that the worse extra costs paid because of Technical Debt are at *requirement* and *test* level, while in the more generic software development the first choice as the most hurtful is *architecture*

Thank You

Terese Besker*	besker@chalmers.se
Antonio Martini*	antonio.martini@chalmers.se
Jan Bosch*	jan.bosch@chalmers.se
Mattias Tichy**	matthias.tichy@uni-ulm.de

*Computer Science and Engineering, Software Engineering
Chalmers University of Technology, Göteborg, Sweden

**Institute of Software Engineering and Programming Languages
Ulm, Germany