

IN5140

Smart processes and agile methods in Software Engineering

Group session 11

- Managing Technical Debt

Agenda

First hour:

- Feedback on Deliverable 2
 - by Wed (Nov 6th)
- Information on Presentation
- Repetition of Lecture

Second hour:

- Previous Exam Questions

Information on presentation instruction

- Time slots are already published in this [sheet](#)
 - Enter your group number into a time slot. First come first serve.
- The entire team has to attend all presentations of the bulk you are in
- We recommend that all team members speak during the presentation
- Each time slot is 8 minutes that includes
 - Presentation: 3-6 minutes
 - Feedback/questions: 2 minutes
- Language: English
- No grade, but mandatory exercise

Presentations (Things to consider)

- May include:
 - Description of process
 - Not necessary to go too much into detail
 - Identified problem(s)
 - BPMN diagram of existing process
 - Illustration of what is causing the problem ◦
 - Eg. Fishbone diagram
 - Metrics to measure improvement
 - Possible solutions/ideas
 - Improved BPMN diagram
 - Plan for introducing and measuring solutions/ideas
 - Feel free to structure the presentation differently!

Questions?

Speak out loud, or type in chat

Repetition from lecture

Summary

What is Technical debt (TD), Security Debt

Debt, principal, interest analysis-> decision whether taking the debt or not

Why and how do we manage TD?

- Identification e.g., code smell via automatic tools
- refactoring
- Decrease and encapsulate complex files

TD management

- With different levels of maturity (tracking from L0 to L6)
- agile TD management (i.e., proactive, continuous, reactive)
- maturity ladder
- via visualization with SonaQube (Visma)

Technical debt and Security debt

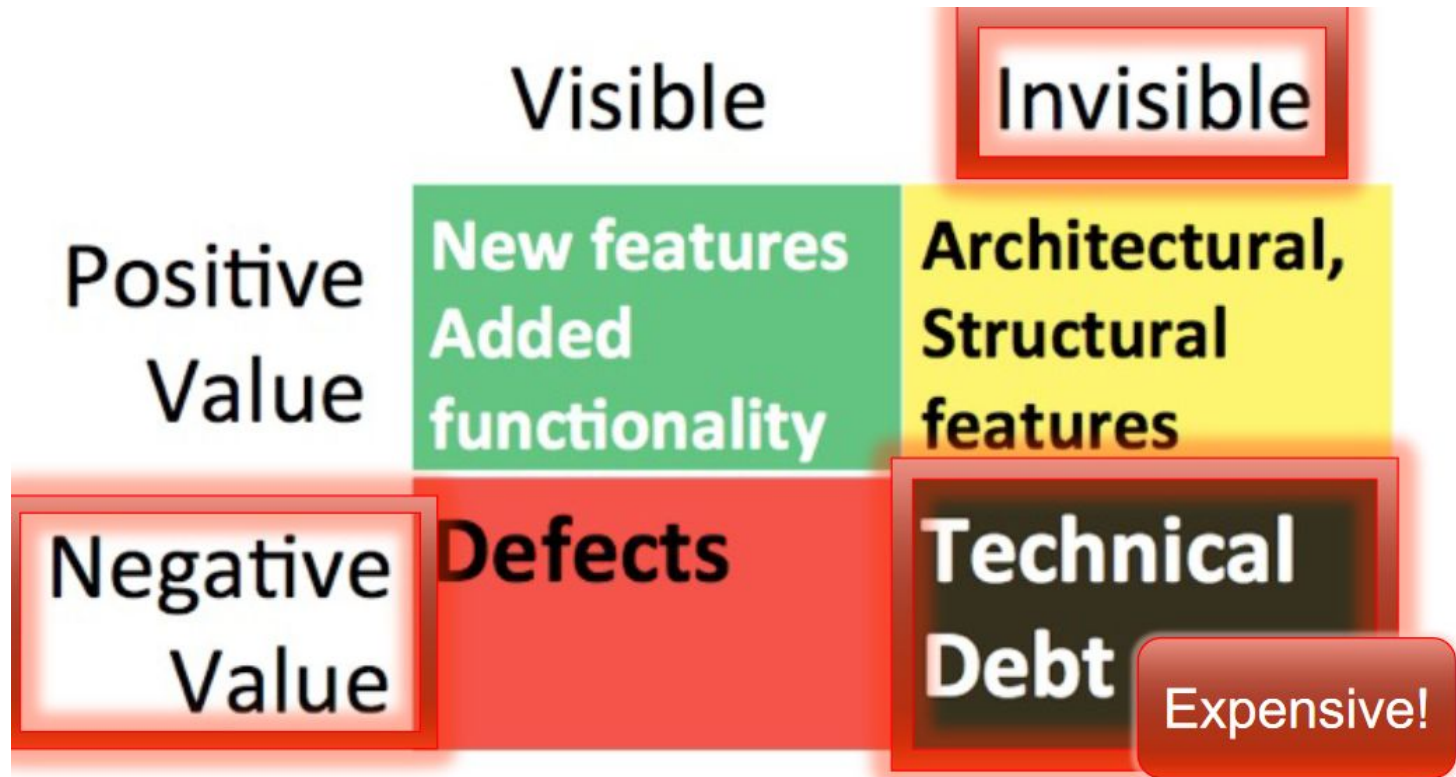
Technical debt definition

"In software-intensive systems, technical debt is a collection of design or implementation constructs that are expedient in the short term, but set up a technical context that can make future changes more costly or impossible. Technical debt presents an actual or contingent liability whose impact is limited to internal system qualities, primarily maintainability and evolvability."

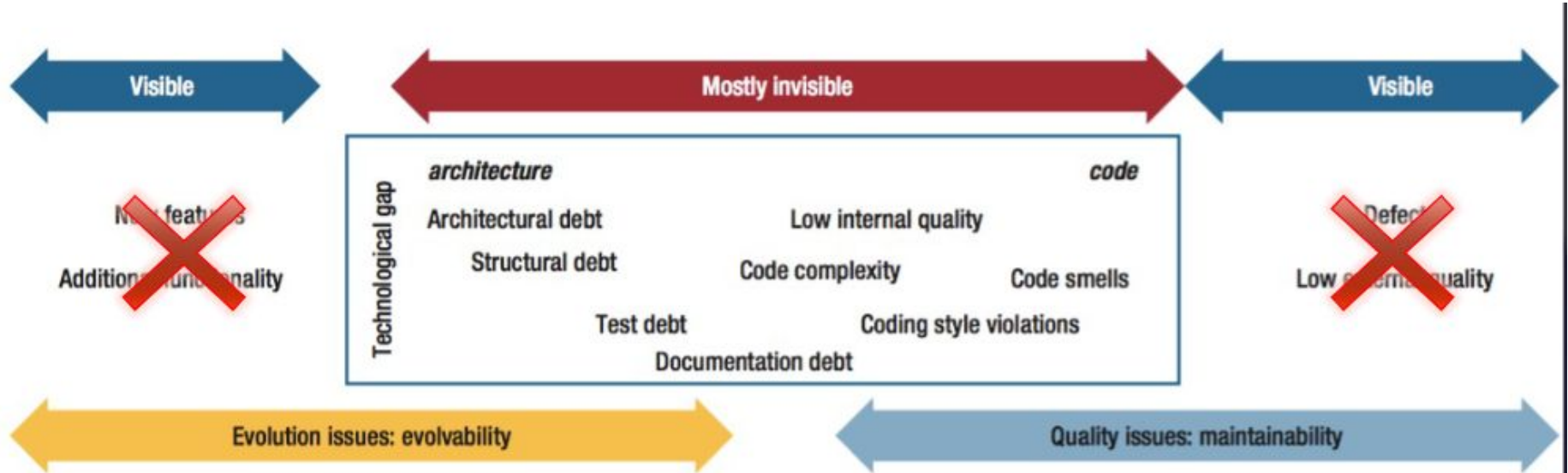
Security debt definition

"Security debt is a set of design or implementation solutions that hinder or has the **potential** to hinder the achievement of a system's security goal"

Technical debt



Technical debt

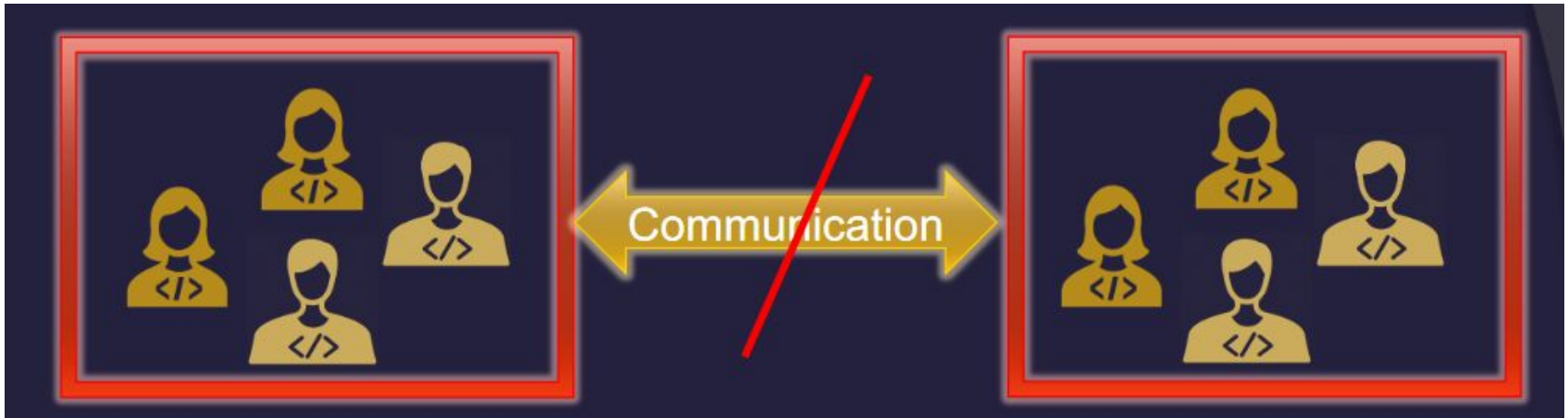


Are there any cases where technical debt can be useful?

- High business risk involved
 - e.g. start-ups
- Prototyping

Social debt

- “The presence of sub-optimality in the development community, which causes a negative effect”



Process debt

- Sub-optimal process design
- Divergence from optimal formal process
- Deficiencies in the infrastructure that might be beneficial in the short term
- Negative long term effects

How do we fix technical debt?

Proactive approaches:

- Education
- Culture
- Organization
- Process
- Guidelines
- Visualization

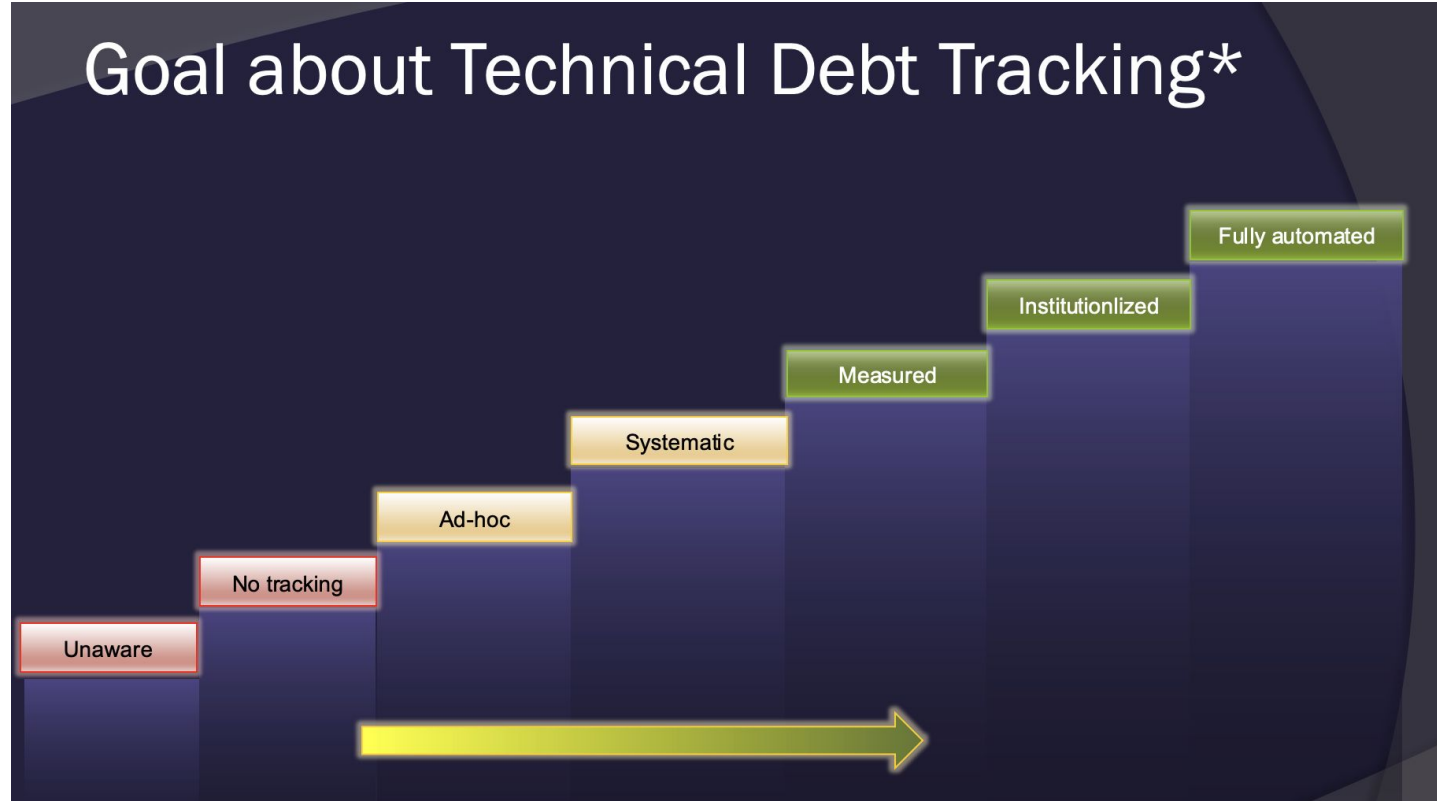
Continuous approaches:

- Semi-automatic identifications
- Code reviews
- Retrospectives
- Technical leadership
- Dedicated refactoring sprints

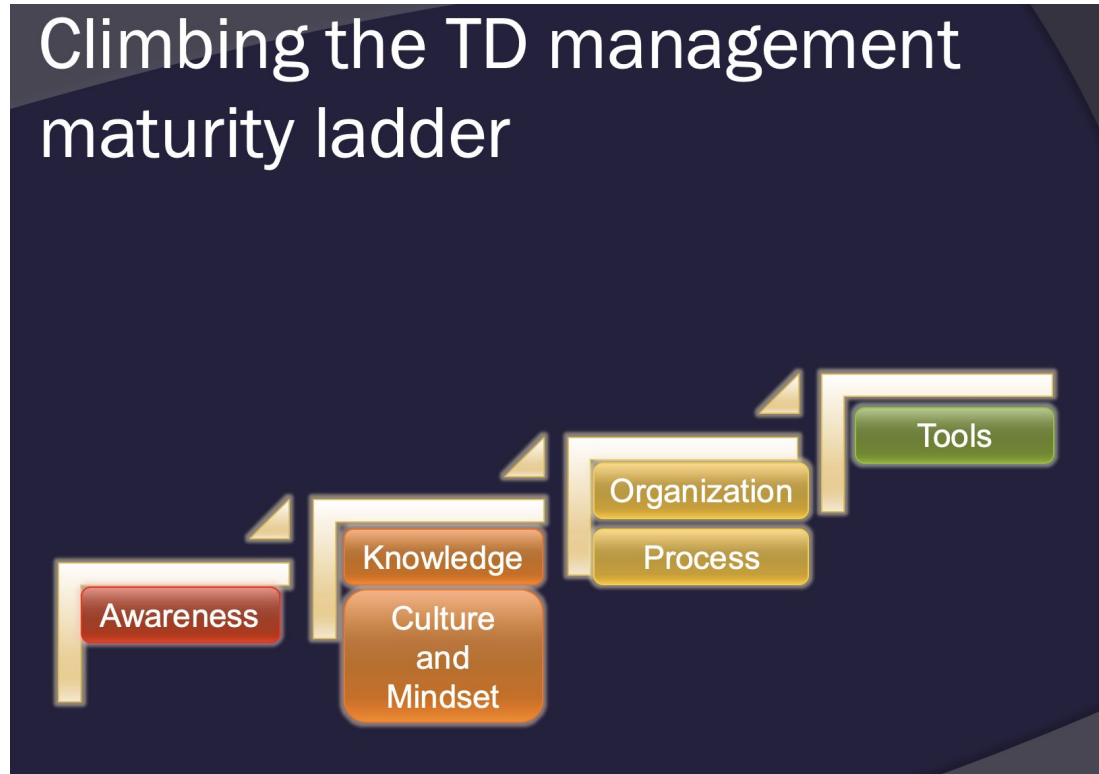
Reactive approaches:

- Impact map
- Roadmap evaluation
- Resources to remove TD
- TD information used in planning and budget

TD management with different levels of maturity



TD management maturity ladder



How can we prevent or fix the other types of debt?

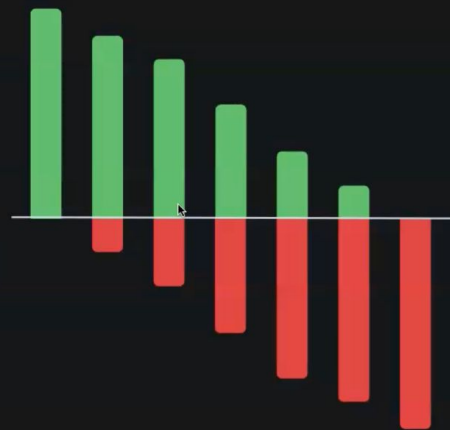
- Social debt
- Process debt

Open question.

Experiences from Visma on TD

Summary

- **Visualise** your technical debt, so you know about it
- Think about **what makes sense** for your service (mature vs startup)
- Having **some technical debt is ok**, keep it under control
- **Never** try to **repay all** technical debt
- Focus on **actual debt first**, then potential debt
- Don't treat the **symptoms**, find the root cause



15 minute Break 

Previous Exam Questions

(2018)

No	Question	Answer A	Answer B	Answer C	Answer D
5	Technical Debt includes only:	Bugs but not missing features	Bugs and missing features	Missing features but not bugs	None of the others

Possible Explanation

Reflect based on the definition of TD

: technical debt is a design or implementation construct that is **expedient in the short term**, but sets up a technical context that can make **a future change more costly or impossible**. Technical debt is a contingent liability whose impact is limited to internal system qualities, primarily maintainability and evolvability

(2018, 2019)

2. Describe the three main concepts of Technical Debt: the debt, the principal and the interest.

Possible Explanation

- The principal (initial cost, cost of replaying on not taking the debt)
- The debt (long term cost, sub-optimal solution)
- The interest (cost of impact)

(2019)

1.3 IN5140 H19 Exam Task 1.3

What is the interest of technical debt?

- ☐ The extra-negative impact paid because of the presence of the technical debt
- ☐ The effort required to fix the technical debt
- ☐ The investment that companies do to manage technical debt
- ☐ The technical issue that needs to be solved

Possible Explanation

TD : design or implementation that is expedient in the short term, but can make a future change most costly or impossible

- invisible, negative value
- Having some TD is ok, keep it under control
- Never try to repay all TD

(2022)

1.4 Technical Debt

Select the correct statement regarding Technical Debt

Select one alternative:

- ☐ The principal of Technical Debt is its the negative impact
- ☐ All Technical Debt should be adressed
- ☐ The interest of Technical Debt is the cost of removing it
- ☐ The average waste generated by Technical Debt is around 30% of the development time

Possible Explanation

TD : design or implementation that is expedient in the short term, but can make a future change most costly or impossible

- invisible, negative value
 - Having some TD is ok, keep it under control
 - Never try to repay all TD
-
- The principal (initial cost, cost of replaying on not taking the debt)
 - The debt (long term cost, sub-optimal solution)
 - The interest (cost of impact)

(2022)

Task B

Technical debt is always negative

Answer here:

Select alternative

(True, False)

Possible Explanation

Define and describe the concept “Technical debt”

Technical debt is a term used to explain the phenomenon where you sacrifice long-term results for short-term results. It arises when you take shortcuts, or develop code in a lazy way to save a small amount of time, resulting in poor code quality that will lead to time wasted in the future. Technical debt can be "fixed" by refactoring the code base, to create better architecture, better design choices, more modular code etc.