Software Maintenance and Technical Debt

Dr. Rodrigo Spínola

Software Maintenance

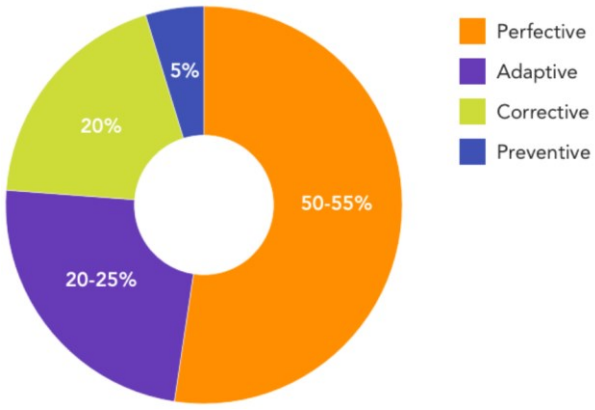
Software Maintenance

- The life of your software does **not begin when coding starts** and **end with the launch**. Instead, it has an **ongoing lifecycle** that stops and starts whenever necessary.
- Software is **always changing** and as long as it is being used, **it has to be monitored and maintained properly**.
- This is partly to adjust for the changes within an organization but also because technology keeps changing.

Why is it necessary?

Corrective	To **fix the bugs and errors** in the software system.
Perfective	To **improve the functionality** of the software to make your product more compatible with the latest marketing and business environments.
Adaptive	To **keep the system functioning effectively** despite changes in its surrounding environment.
Preventive	To **prevent the deterioration** of your software as it continues to adapt and change. These services can include optimizing code and updating documentation as needed.

Types



Technical Debt

Technical Debt

- Outdated documentation;
- Code that do not follow good practices;
- Insufficient test coverage;
- ...

Technical debt (TD) contextualizes the problem of pending development tasks as a type of debt that brings a short-term benefit to the project, but which may have to be paid with interest later in the development process.

- Working on evolving software with internal quality issues tends to be harder, more costly, time-consuming, and demotivating.

- Increased development speed;
- Shortened time to market;
- Reach deadlines;
- ...

(Seaman and Guo, 2011; Kruchten et al., 2012; Avgeriou et al., 2016)

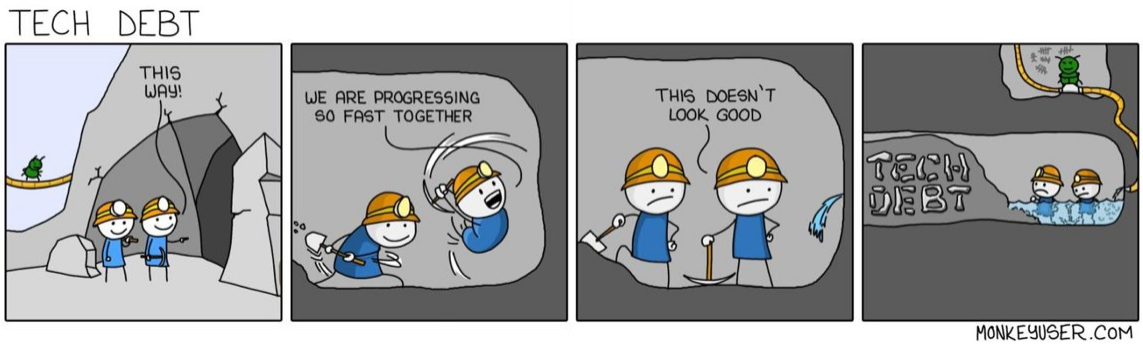
Technical Debt

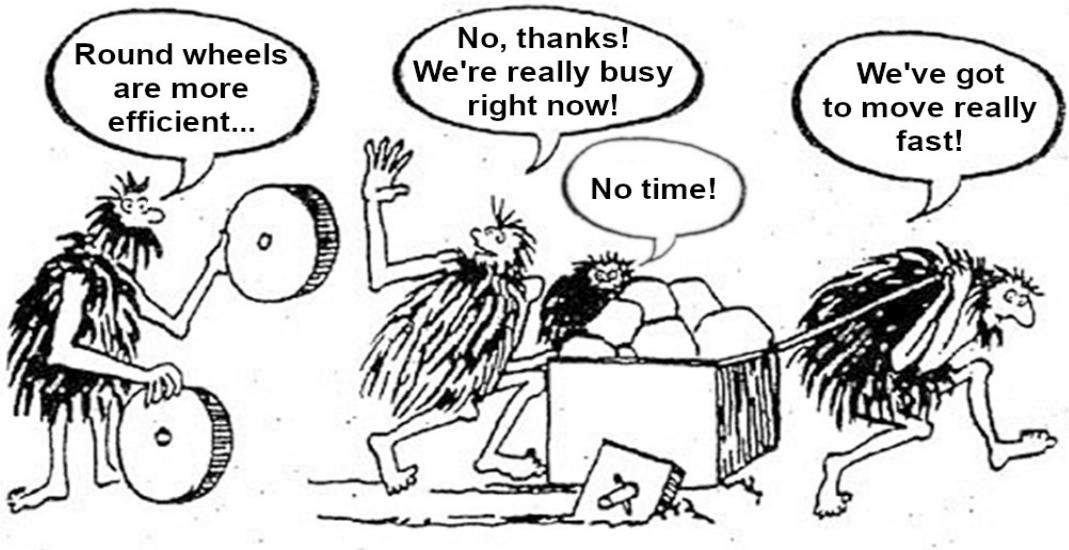
- Technical Debt is the gap between:
  - Developing a software perfectly
    - Preserving architectural design
    - Employ good programming practices and standards
    - Updating the documentation
    - Testing thoroughly
  - And making the software to work
    - As quickly as possible
    - With as few resources as possible

chatGPT

- While technical debt can allow **faster** progress in the short term, it accumulates over time, leading to **increased complexity**, **reduced efficiency**, and **higher costs for future modifications**.
- Managing technical debt requires **balancing short-term gains with long-term sustainability to maintain the health and adaptability of the software system**.

se center
VCU Computer Science College of Engineering


TECH DEBT
THIS WAY!
WE ARE PROGRESSING SO FAST TOGETHER
THIS DOESN'T LOOK GOOD
TECH DEBT
MONKEYUSER.COM


Round wheels are more efficient...
No, thanks! We're really busy right now!
No time!
We've got to move really fast!

13

The Technical Debt Landscape

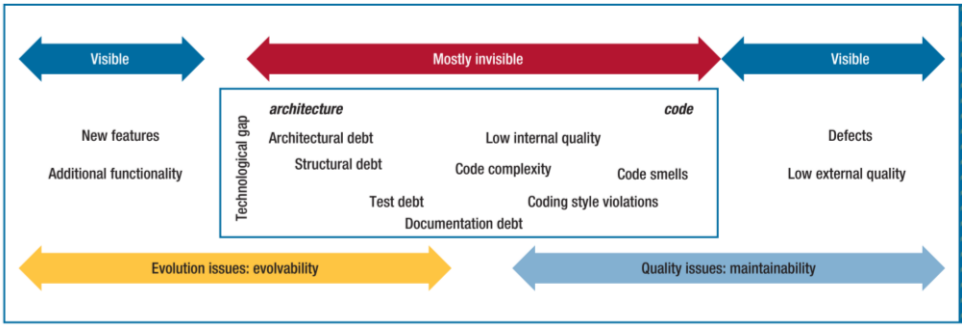


FIGURE 1. The technical debt landscape. On the left, evolution or its challenges; on the right, quality issues, both internal and external.

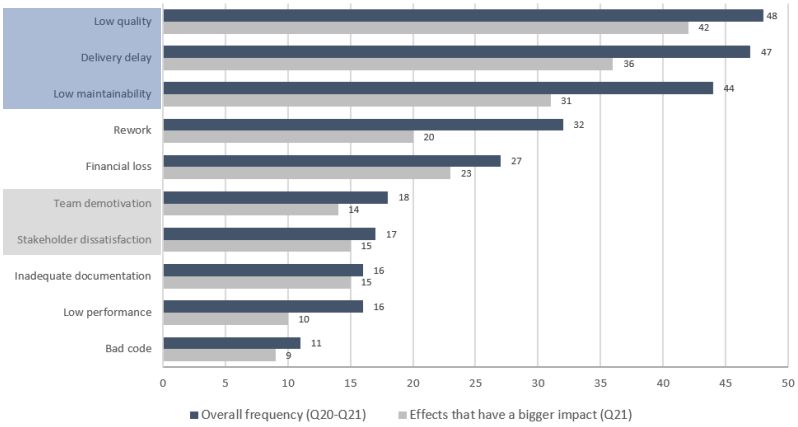
14

What effects does TD have on software projects?

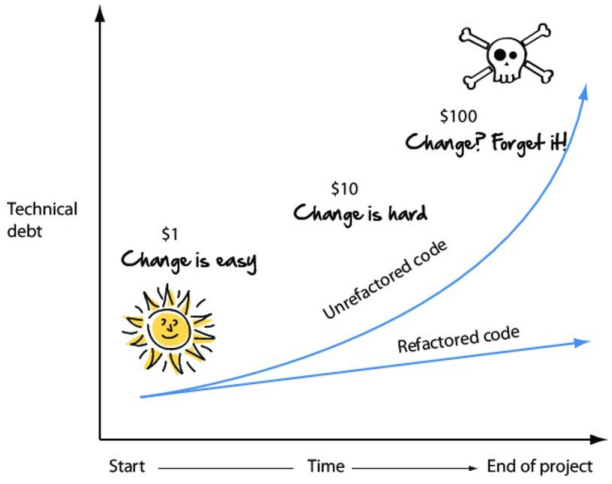
- TD can **affect** projects in different ways. Having this information could aid in **prioritization of TD** items to pay off, by supporting:
  - a more precise **impact analysis**, and;
  - the **identification of corrective actions** to minimize possible negative consequences for the project.

What effects does TD have on software projects?

- 66 effects

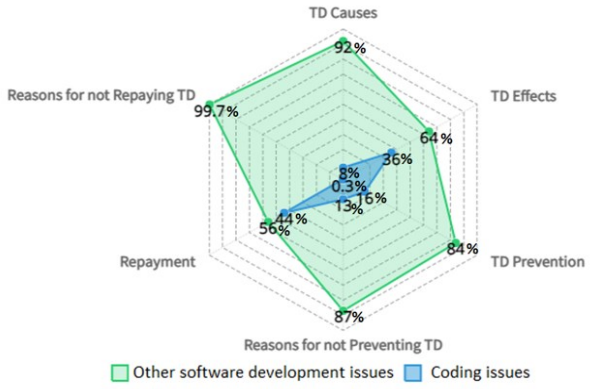


Low Maintainability



17

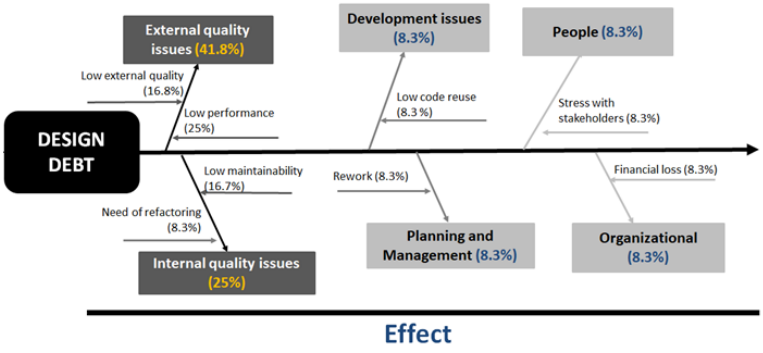
TD is not only about code and we need to be aware of it



18

Probabilistic Diagram of Effects of TD

- Design debt + small development team (< 10 members) + agile process.



Summary

Software Maintenance is an essential part of the software development life cycle; it is **necessary for the success and evolution** of your system. Maintenance on software **goes beyond fixing "bugs"**, which is one of the four types of software change. Updating the software environment, reducing its deterioration over time, and enhancing features to satisfy user needs are all examples of maintenance work.

Summary

- Technical debt

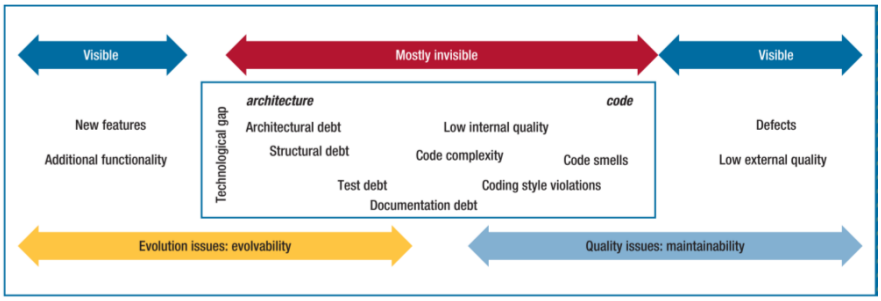


FIGURE 1. The technical debt landscape. On the left, evolution or its challenges; on the right, quality issues, both internal and external.

Software Maintenance and Technical Debt

Dr. Rodrigo Spínola

se center
VCU Computer Science
College of Engineering