

Rolling in the Debt: From Technical Debt To Asset Management

Dr. Ehsan Zabardast



redeploy



SERL Sweden

LEADING SOFTWARE ENGINEERING



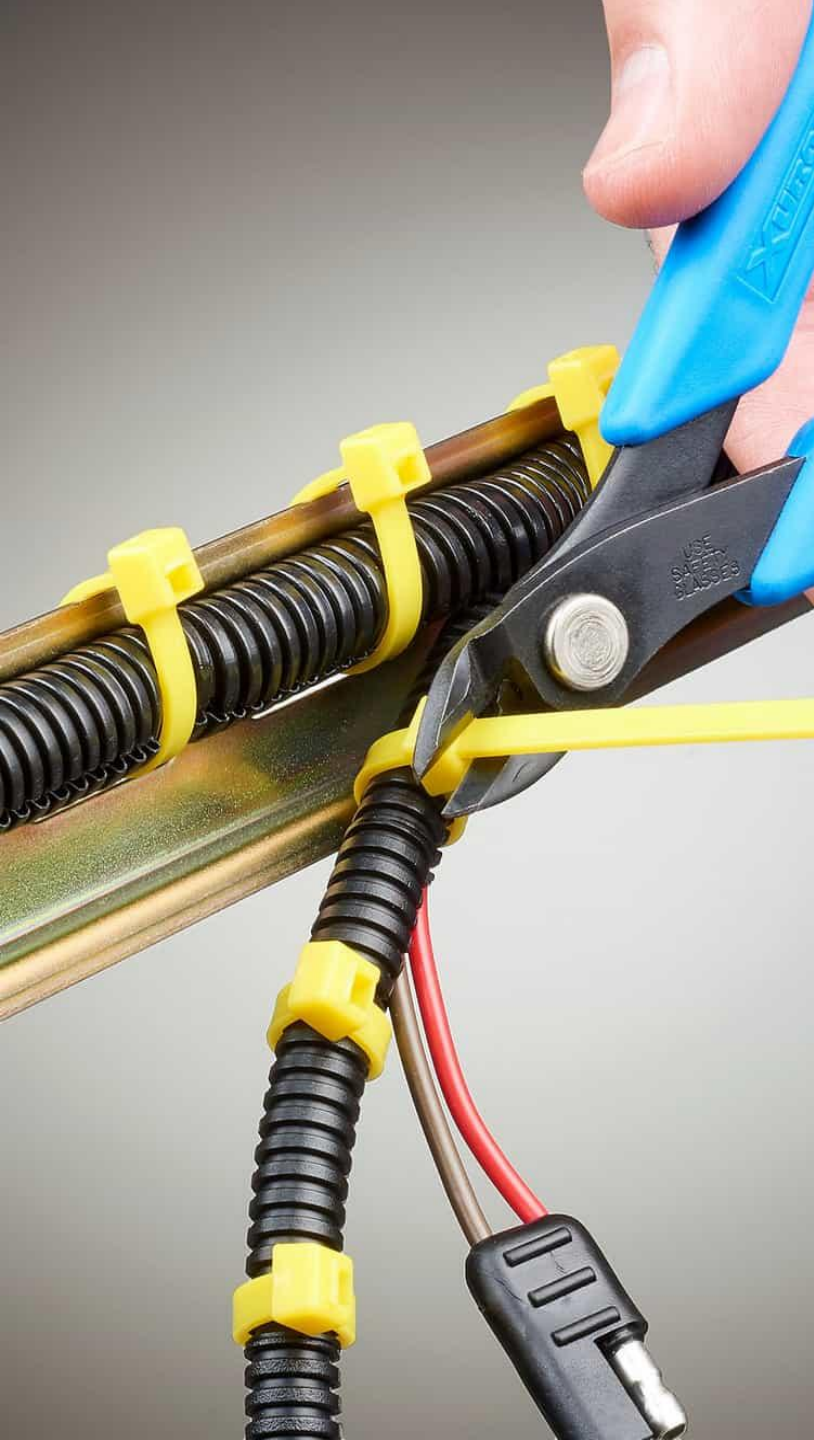
Dr. Javier Gonzalez Huerta



Prof. Dr. Tony Gorschek



SERL Sweden
LEADING SOFTWARE ENGINEERING



MULTI PURPOSE
SUPER
GLUE

IDEAL FOR

- GLASS
- PLASTIC
- METAL
- WOOD
- RUBBER
- LEATHER
- PORCELAIN

Galaxy
MULTI PURPOSE
SUPER
GLUE
20 g
Keep Out Of Reach Of Children

20 g
KEEP OUT OF REACH OF CHILDREN



Sub Optimal

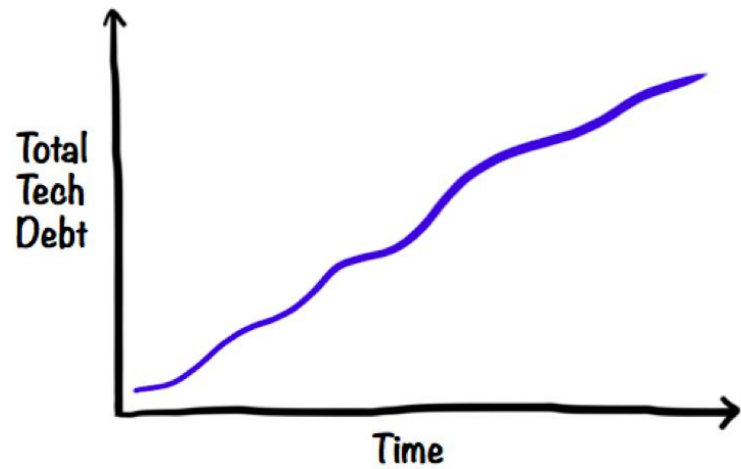
Optimal



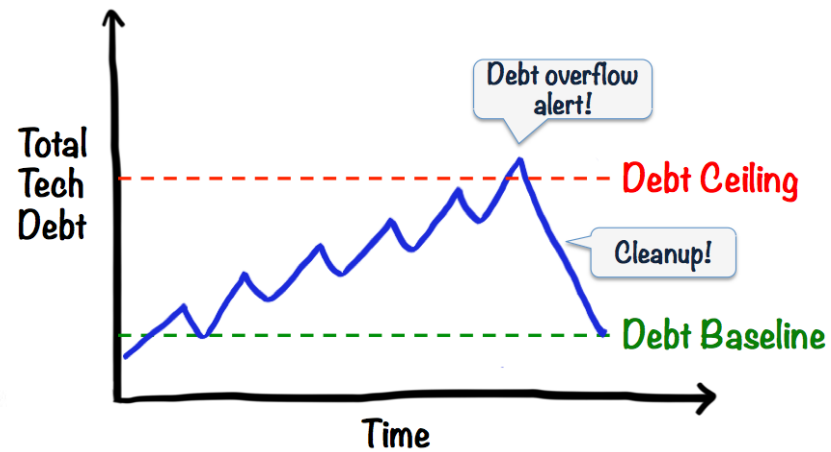
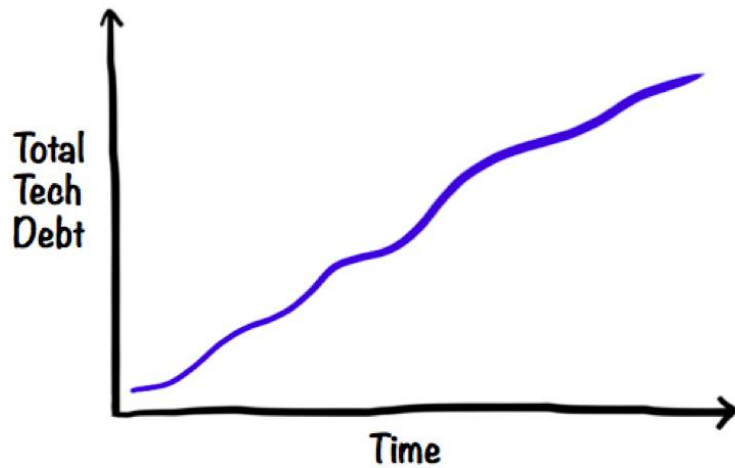


Technical Debt refers to the long-term cost and inefficiencies that result from taking shortcuts or suboptimal solutions in software development, which require future refactoring or maintenance.

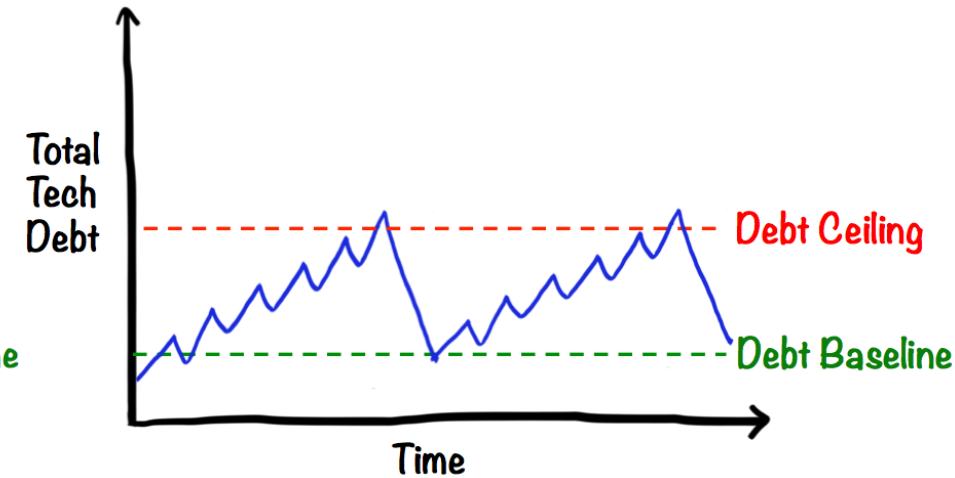
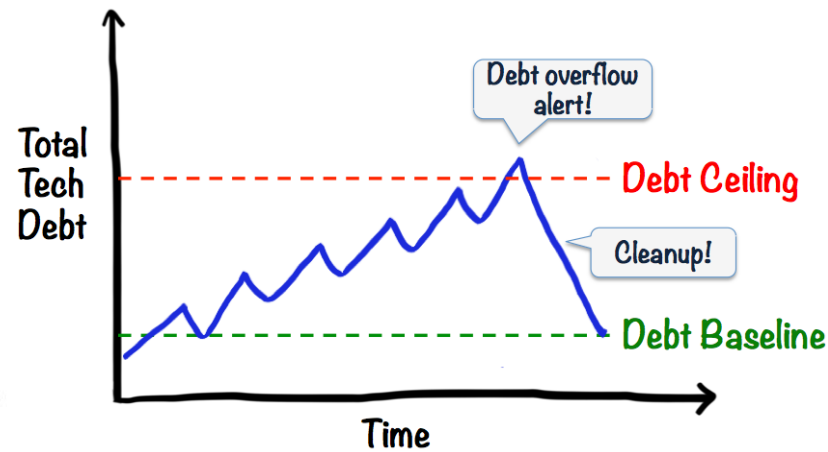
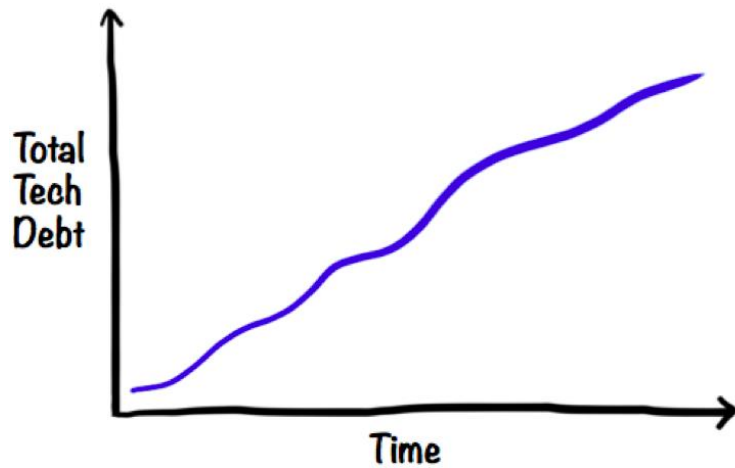
Technical Debt In Theory



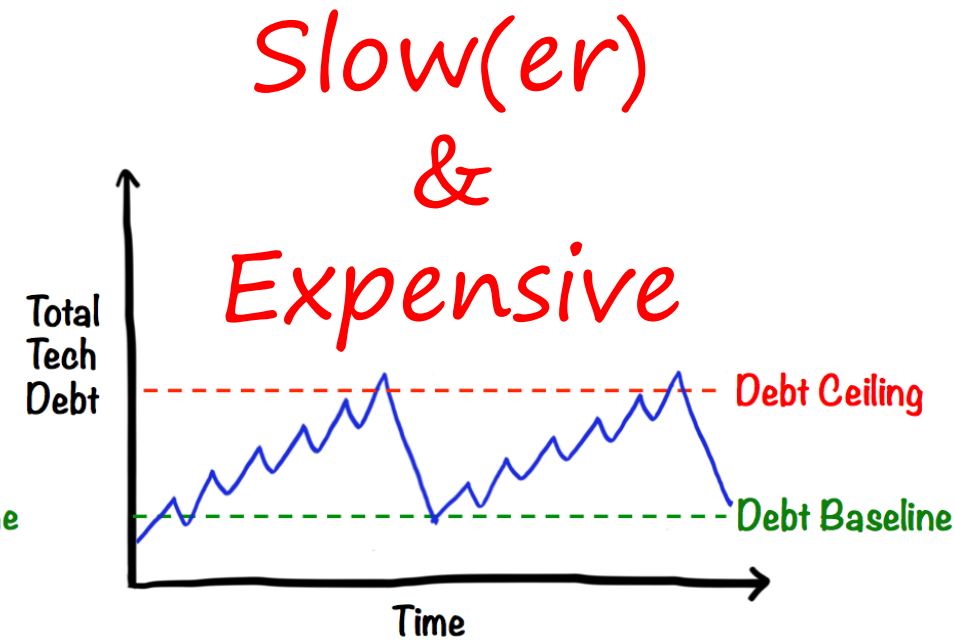
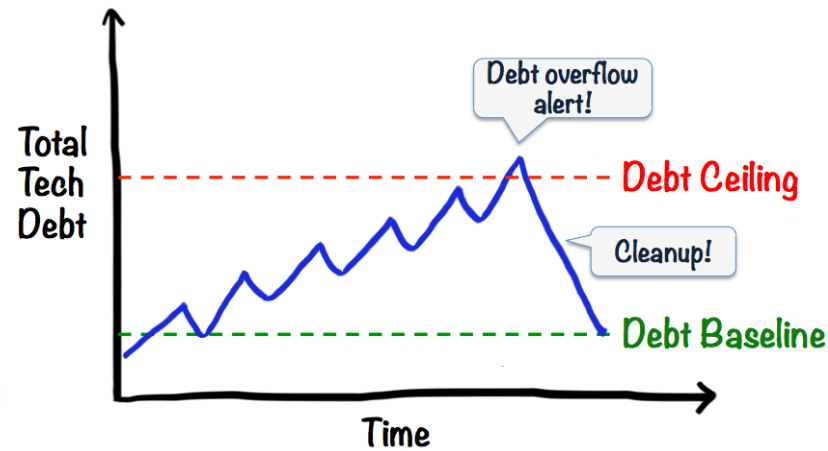
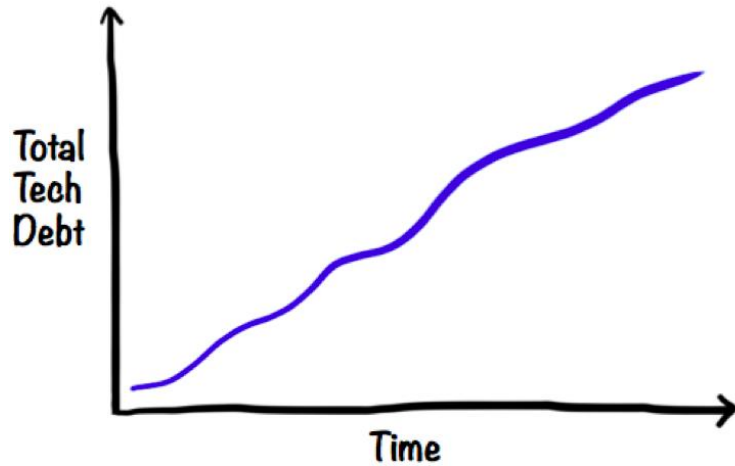
Technical Debt In Theory



Technical Debt In Theory



Technical Debt In Theory



Technical Debt In Practice

sonarqube



Apache Hive

The Apache Hive™ is a distributed, fault-tolerant data warehouse system that enables analytics at a massive scale and facilitates reading, writing, and managing petabytes of data residing in distributed storage using SQL.



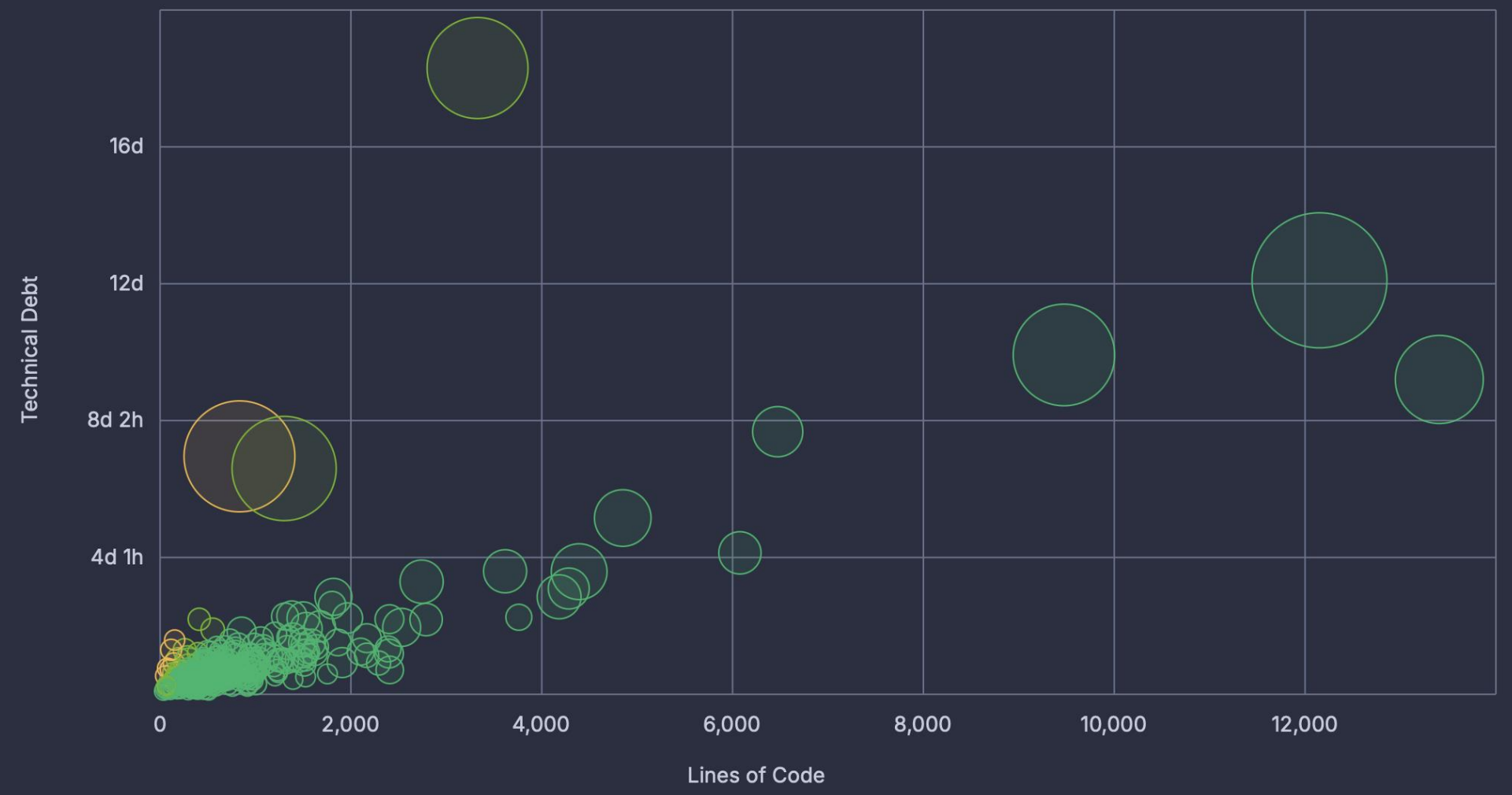
Hive

500/7,728 files

Maintainability Overview ?

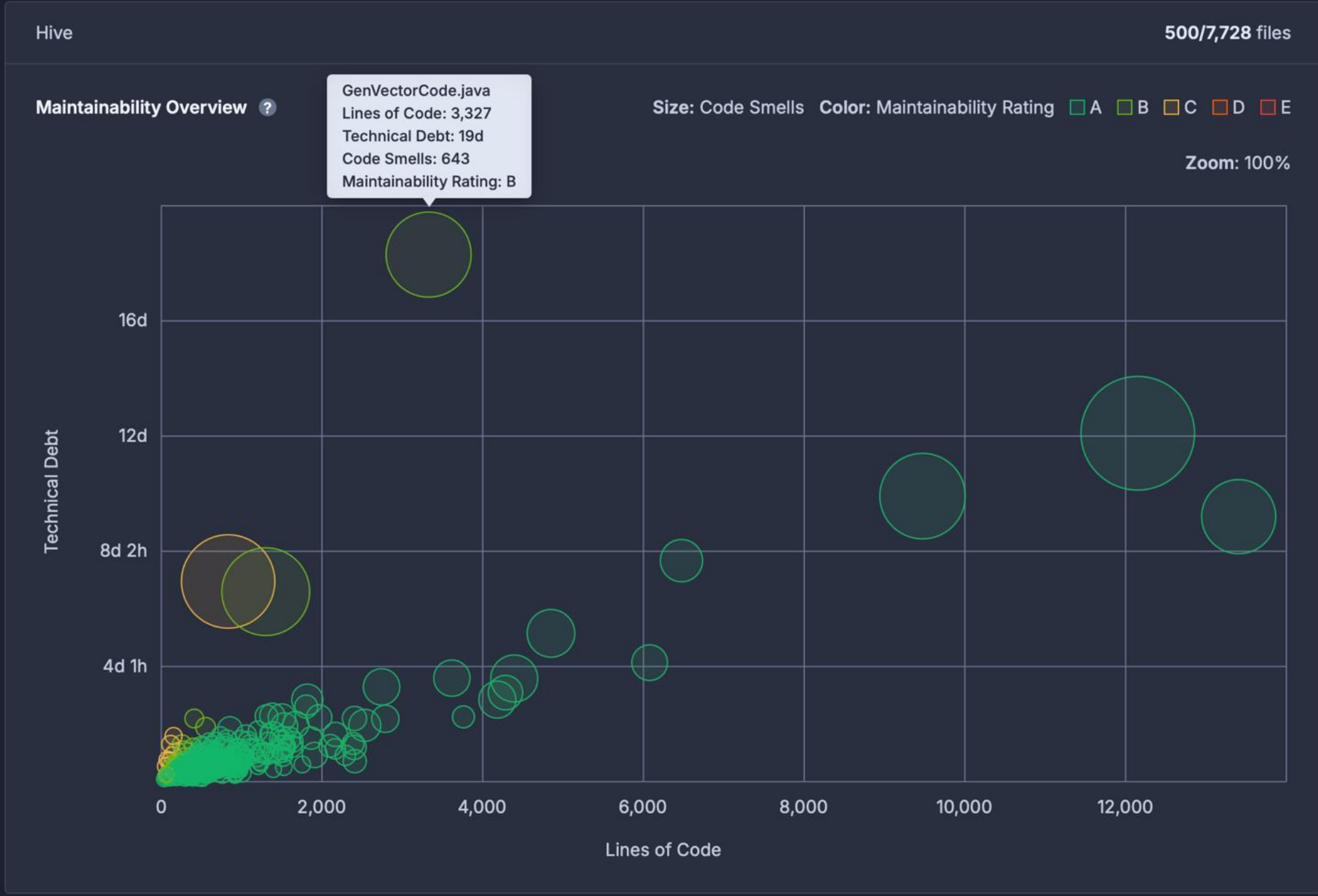
Size: Code Smells Color: Maintainability Rating A B C D E

Zoom: 100%





Technical Debt:
838 Day!

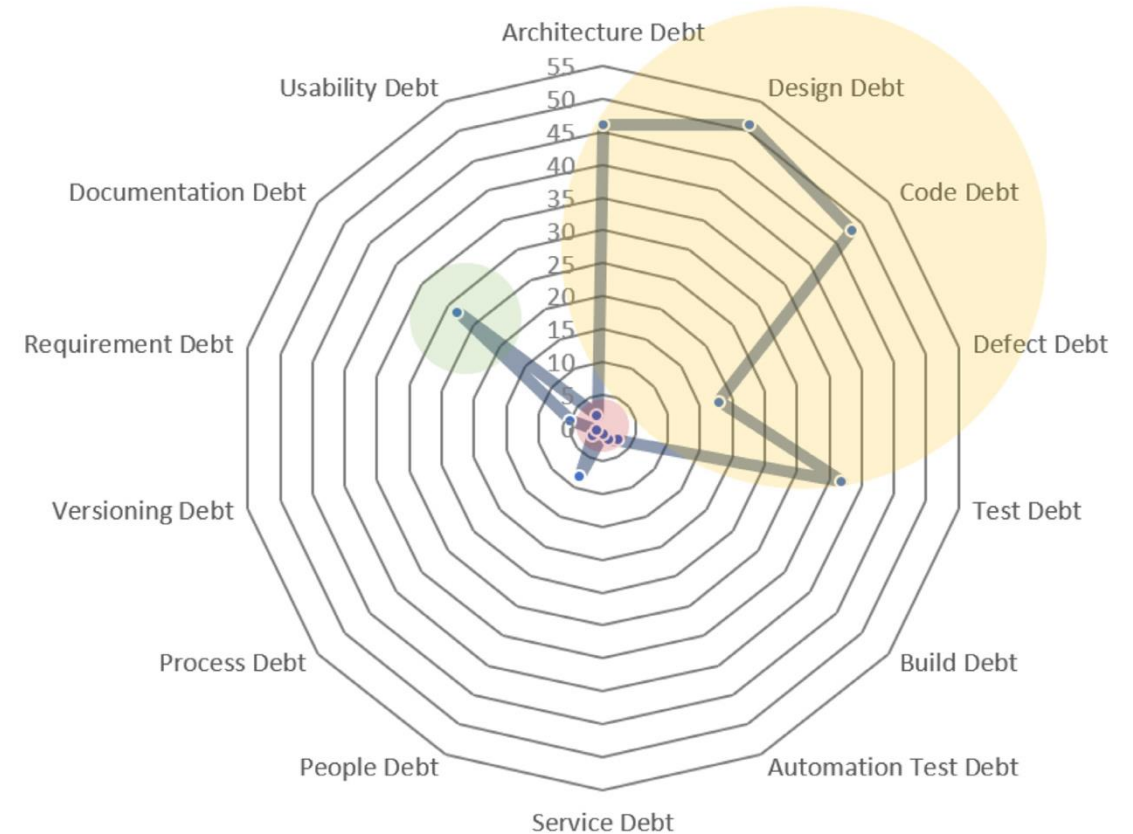


Technical Debt Research

Current Research State:

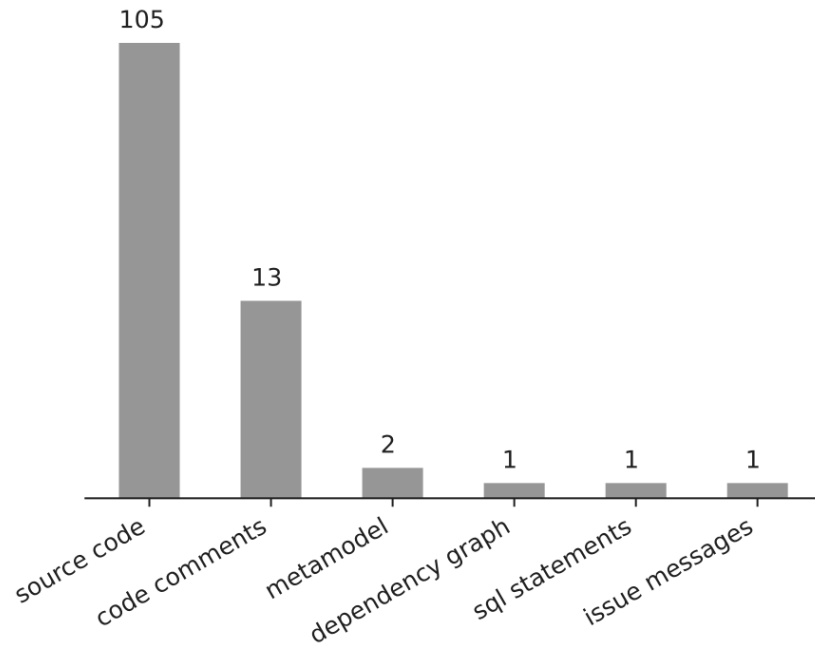
- Value creation perspective
- Focus on architecture over code
- Advanced tools for technical debt management
- Continuous technical debt management
- Data-driven technical debt management
- Socio-technical factors in technical debt
- Cross-disciplinary collaboration
- Proactive over reactive debt management
- Use of standardized metrics for technical debt
- Stakeholder involvement in technical debt decision

Avgeriou, P., Ozkaya, I., Chatzigeorgiou, A., Ciolkowski, M., Ernst, N. A., Koontz, R. J., ... & Shull, F. (2023, May). Technical debt management: The road ahead for successful software delivery. In 2023 IEEE/ACM International Conference on Software Engineering: Future of Software Engineering (ICSE-FoSE) (pp. 15-30). IEEE.

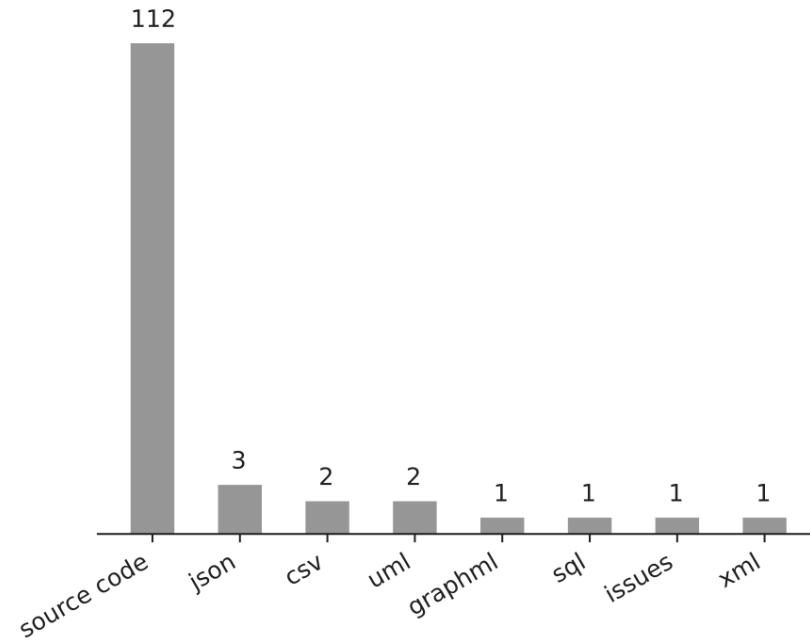


Rios, N., de Mendonça Neto, M. G., & Spínola, R. O. (2018). A tertiary study on technical debt: Types, management strategies, research trends, and base information for practitioners. *Information and Software Technology*, 102, 117-145.

Technical Debt Management Research

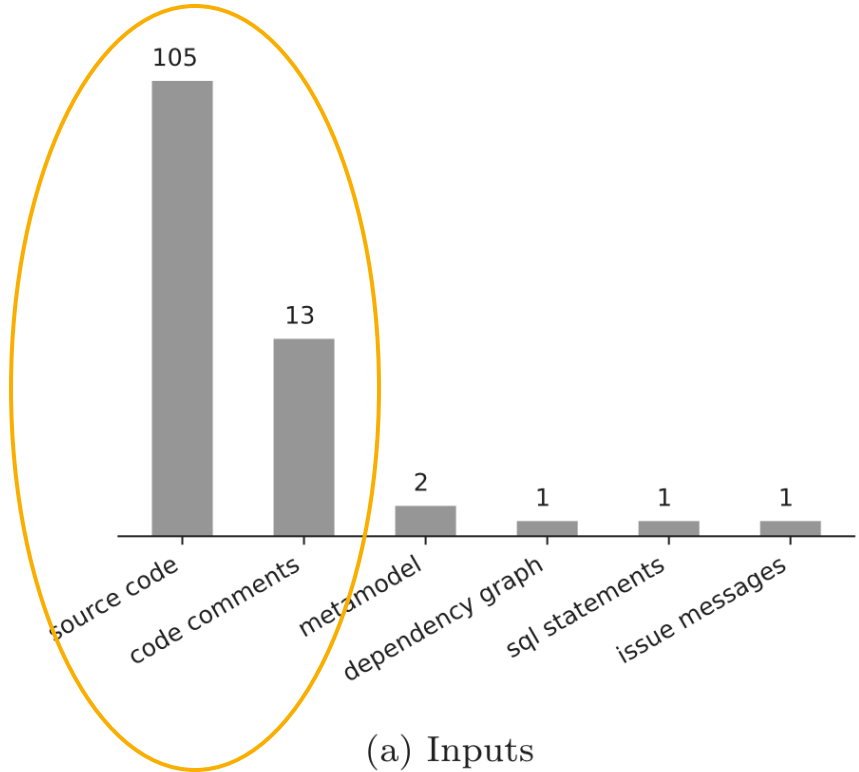


(a) Inputs

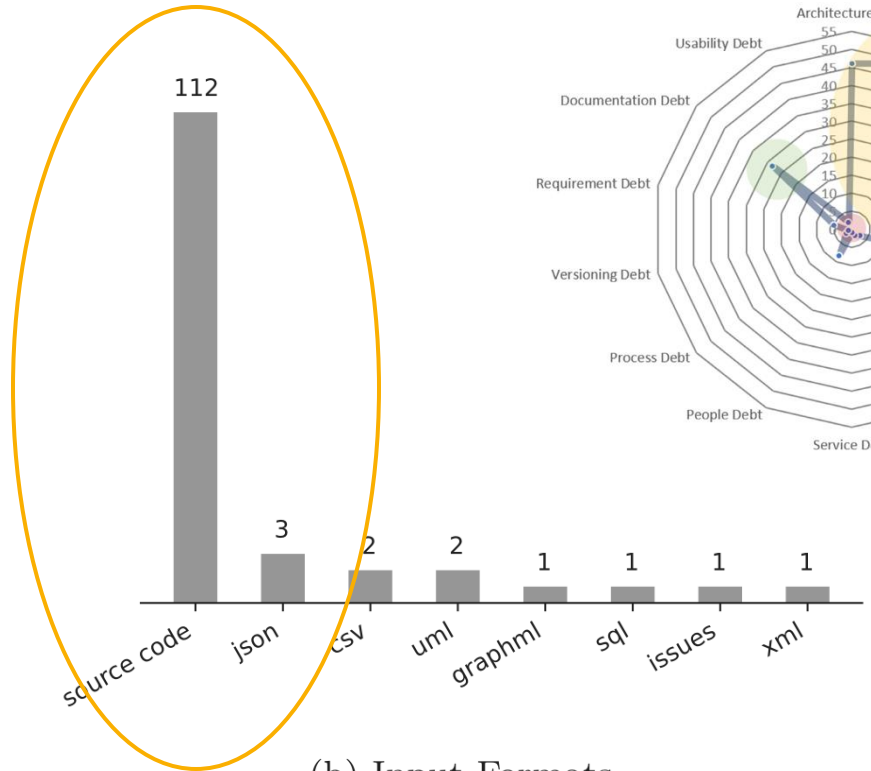


(b) Input Formats

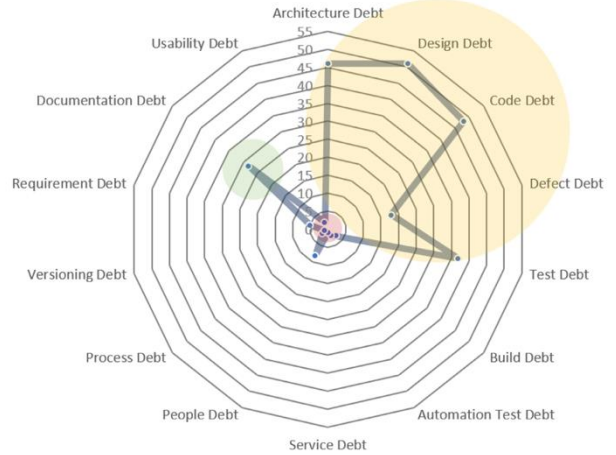
Technical Debt Management Research



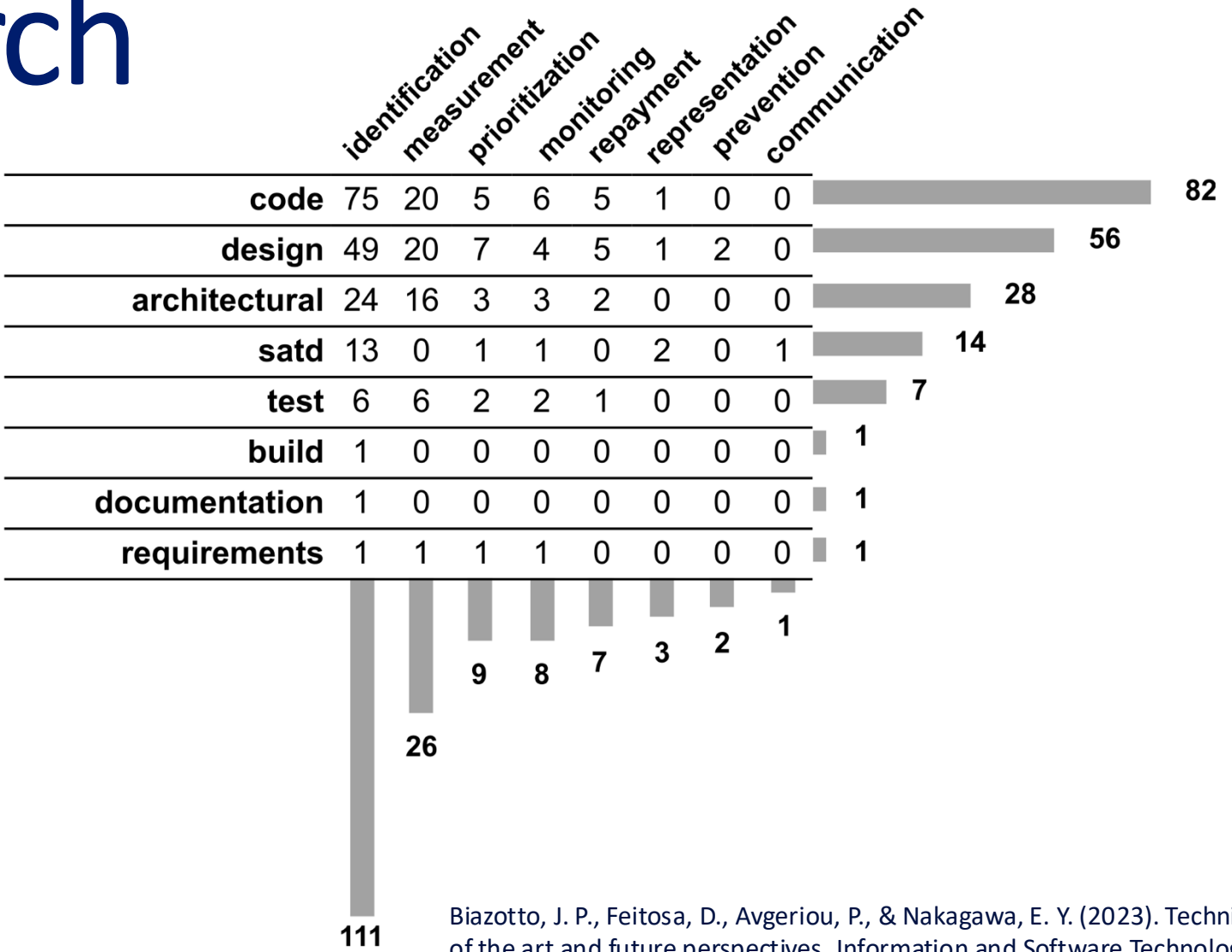
(a) Inputs



(b) Input Formats

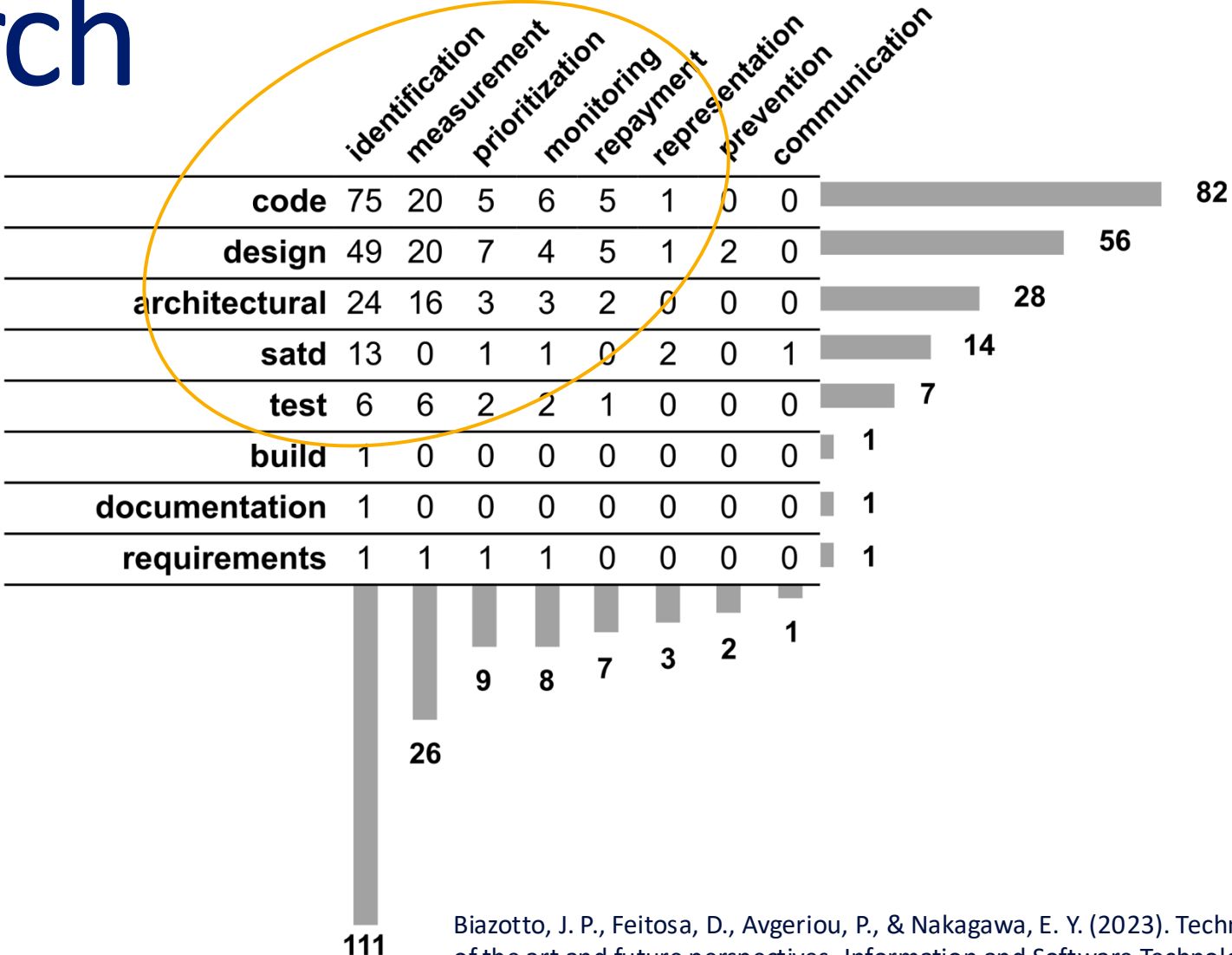


Technical Debt Management Research



Biazotto, J. P., Feitosa, D., Avgeriou, P., & Nakagawa, E. Y. (2023). Technical debt management automation: State of the art and future perspectives. Information and Software Technology, 107375.

Technical Debt Management Research



Biazotto, J. P., Feitosa, D., Avgeriou, P., & Nakagawa, E. Y. (2023). Technical debt management automation: State of the art and future perspectives. Information and Software Technology, 107375.

Technical Debt Facts

- Technical debt is unavoidable!
- Virtually, you can't get rid of technical debt.
- Measuring technical debt is not straightforward.
- Technical debt is contagious!



... so what?



AWS Cloud Enterprise Strategy Blog
The CIO-CFO Conversation: Technical Debt—An Apt Term?
 by Mark Schwartz | on 16 DEC 2020 | In Enterprise Strategy, Finance and Investment | Permalink | Comments
 | Share



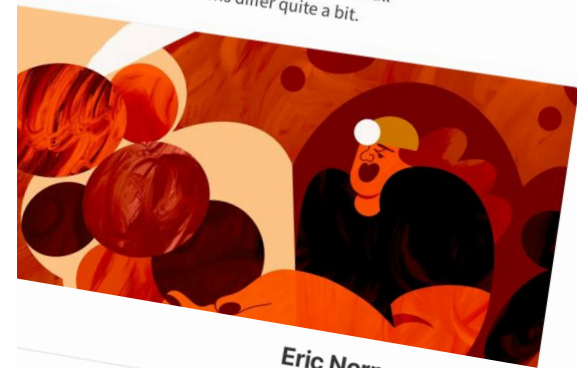
Sometimes we technologists can be a bit too...
 The term *technical debt*, attributed to Ward...
 OOPSLA conference speech, may be an exa...
 often these days, generally in the context of...
 nonfunctional aspects of IT; that is, investm...
 the internals of IT systems, rather than to a...
 capabilities. Such nonfunctional investment...
 difficult to explain and justify. The term te...

a good way to communicate this need "in the language of the business."

DECEMBER 27, 2023

Stop saying "technical debt"

Everyone who says "tech debt" assumes they know what we're all talking about, but their individual definitions differ quite a bit.



AUTHORS
Chelsea Troy
 Chelsea is a Staff Data Engineer at Mozilla, focused on the role of data privacy and scientific rigor in automation. She is a maintainer for the Roc programming language and mentors formerly incarcerated technologists through...

Code for a Living code maintenance technical debt

RECENT ARTICLES
 OCTOBER 22, 2024
CEO Update: Building trust in AI is key to a thriving knowledge

Eric Normand's Newsletter

It's time to stop saying technical debt

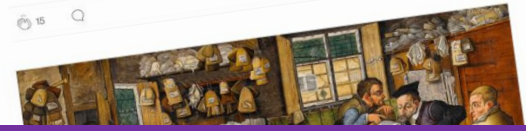
I've never metaphor I didn't like. Until now.

ERIC NORMAND
 DEC 25, 2023

18 2

On Technical Debt and Why We Can't Have Nice Things

Josef (Yossi) Goldstein · Follow
 10 min read · Mar 1, 2024

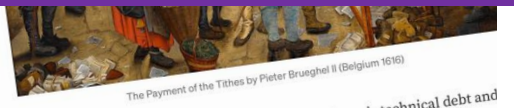


The Controversial Truth about Tech Debt

Raphael Moutard · Follow
 5 min read · May 12, 2024

4.7K 151

Of all the buzzwords invented by the software industry, Technical Debt is the most frustrating. I know this will be controversial, and I can already hear clean architecture zealots fulminating. So let me explain my thoughts.



If there's one thing engineers can't stop talking about is technical debt and how much everyone feels like they are not doing enough to battle it. "We have too much of it!" they say. "We should have a zero tech debt policy in company!" some argue. "So much of our system is legacy code, this is horrible tech debt!" they demand, wishing to rewrite it all. It seems like tech debt is the one and only bane of good software engineers and if we were only to get rid of it we could reach coding utopia.



What the f\$%k is Tech Debt?

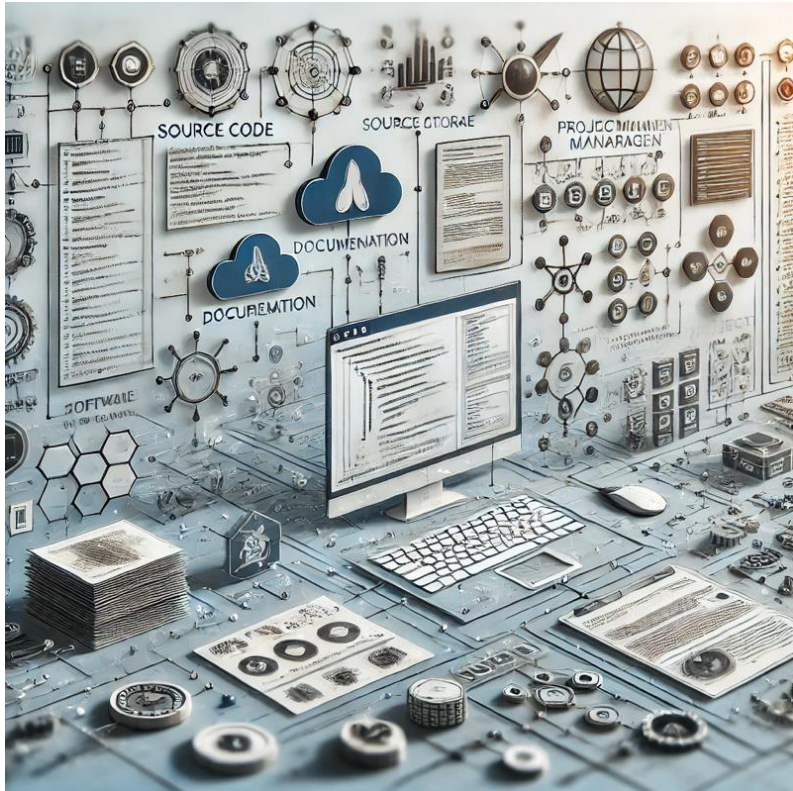
I ask during interviews "What is the definition of tech debt?" surprisingly every candidate has a different answer. It seems like the industry hasn't converged. I classified the responses into the following categories:

People	Best Practices	Unfinished Migration
code nobody wants to modify	code that is not tested	code that is not typed (javascript vs typescript)

"STOP SAYING TECHNICAL DEBT"

...colleagues on the business side. It's to blame for our low morale. And we know the story: If only our managers wouldn't pressure us to deliver the next re, we could do it right. But we never get the chance. We tell them there's too technical debt, but they don't stop pushing us for new features. e all vague terms, scapegoats, and boogeymen, it hides the complexity of reality. rd the term *technical debt* used for so many different things that they've become e messy code, lousy architecture, unfinished database migrations, unused files

Assets in Software Engineering



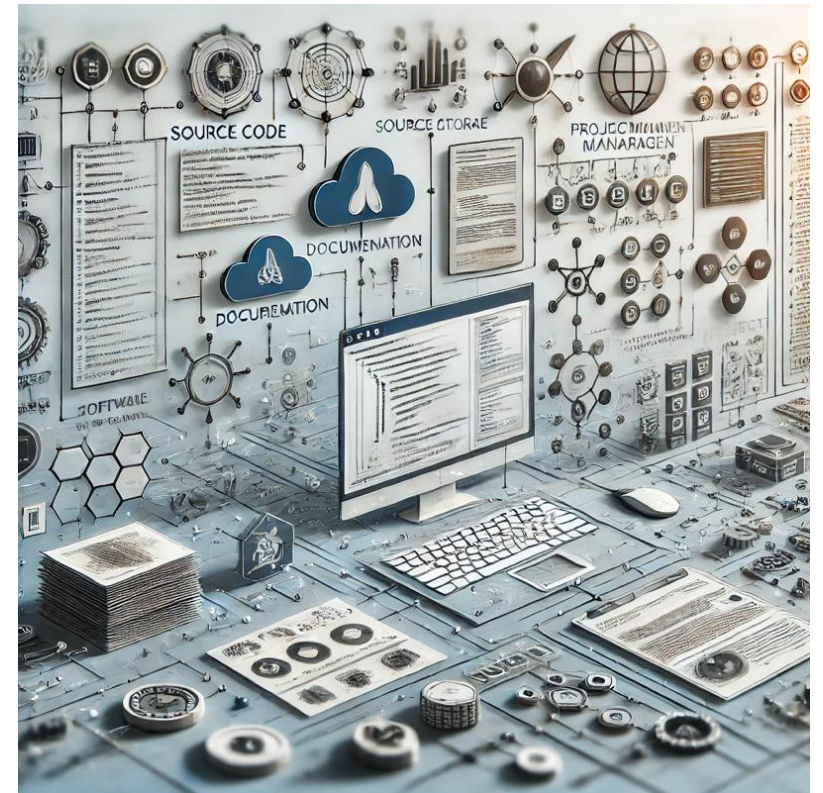
bit.ly/techdebt2025

Assets in Software Engineering



Assets are artefacts that:

- Are used frequently during software's life cycle
- Have potential or actual value for the organisation



Quality Matters



Continuously controlling and ensuring quality is justified by continuous use.

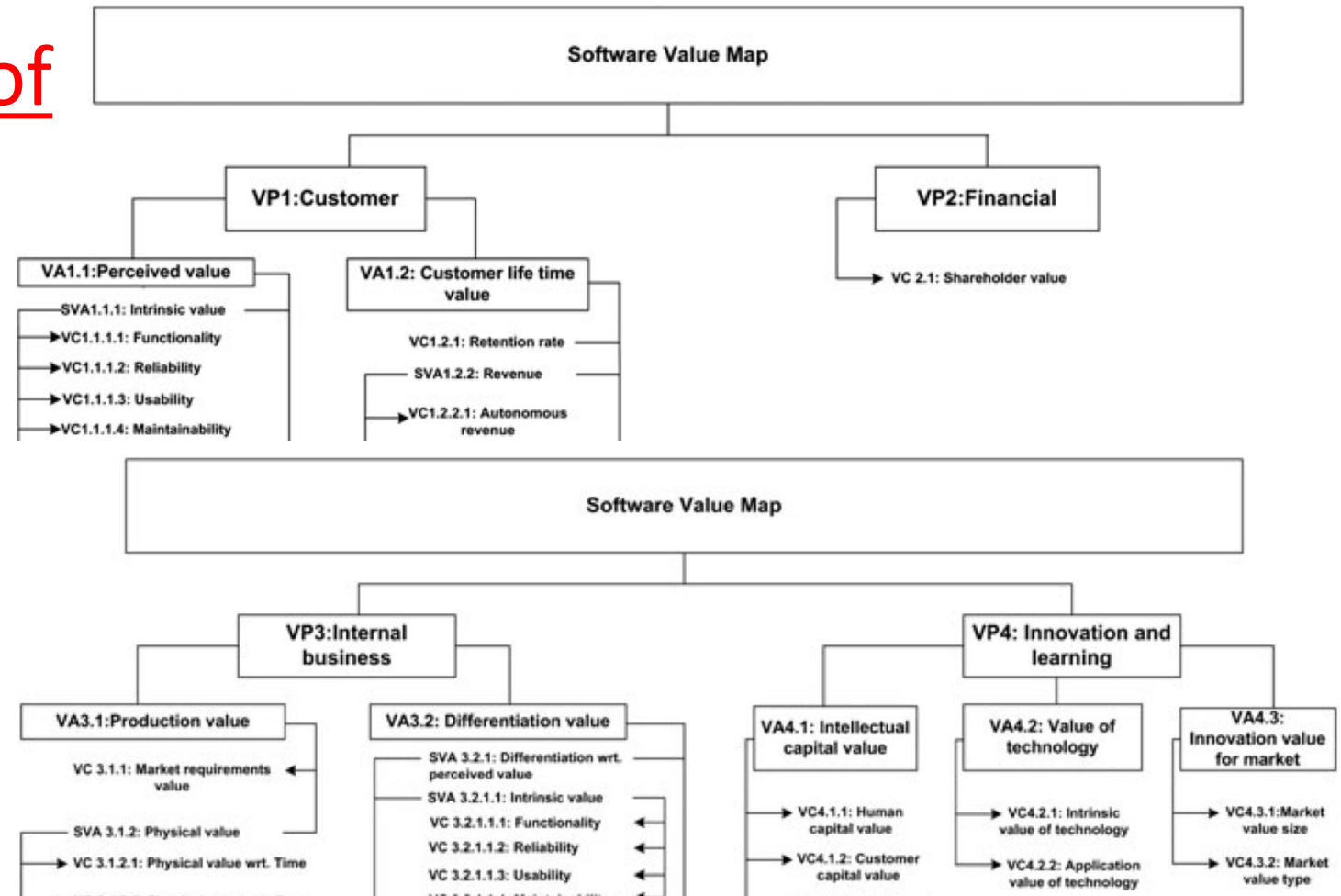


	intended to be used more than once	one time used
Artefacts	Assets <i>e.g., Code</i>	Not Asset <i>e.g., Test Result</i>

Asset Degradation



Degradation is the loss of value that an asset suffers due its manipulation.






Khurum, M., Gorschek, T., & Wilson, M. (2013). The software value map—an exhaustive collection of value aspects for the development of software intensive products. *Journal of software: Evolution and Process*, 25(7), 711-741.

Asset Degradation



Degradation is the **loss of value** that an asset suffers due its **manipulation.**

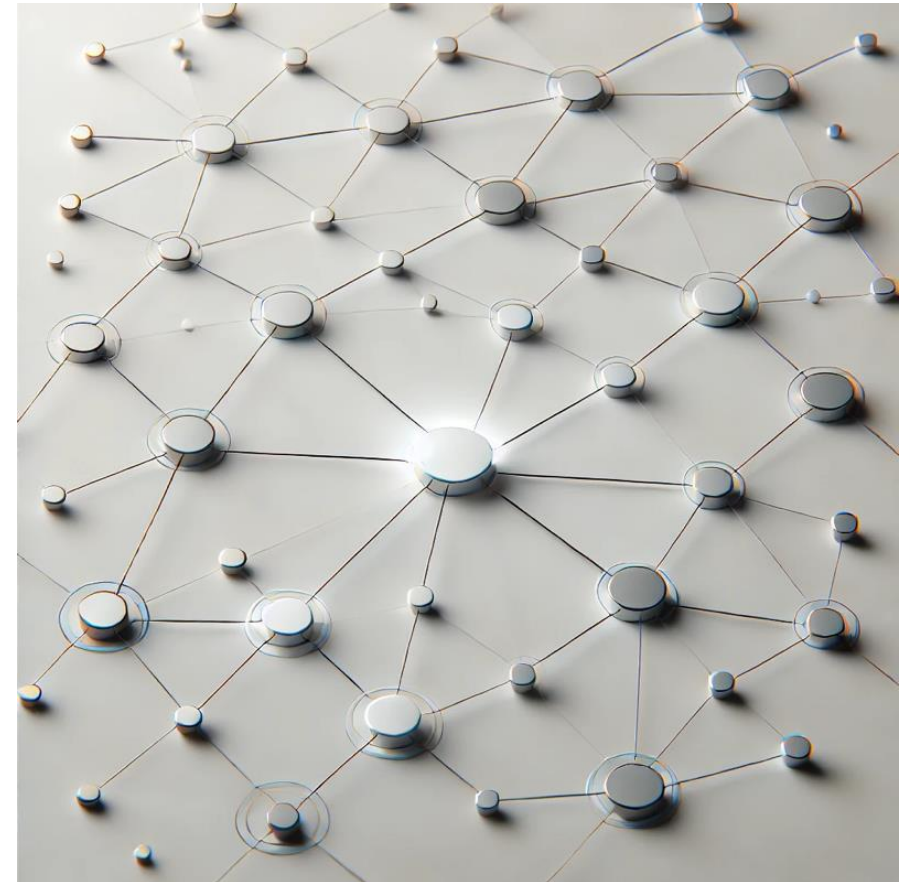
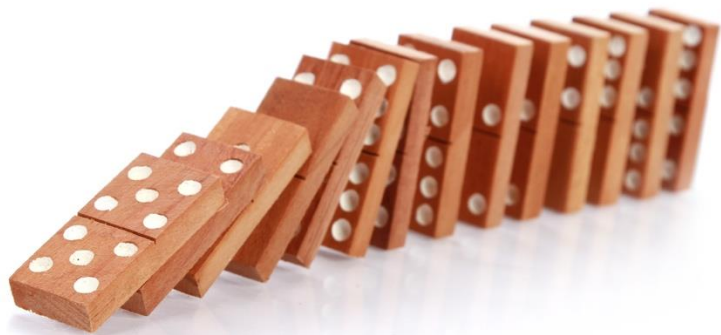
		Degradation		
		Deliberate	Unintentional	Entropy
	Organisation Structure <i>(e.g. Teams' Constellation Document)</i>	1 Over-simplification of role structures	4 Unofficial roles in organisation	7 Outdated document due to company growth
	Activity Diagram	2 Intentionally leaving some details for later	5 Usage of outdated terminology	8 Diagram is non-representative of actual system
	Code	3 Suboptimal solutions to achieve immediate goals (TD)	6 Code clones are introduced unintentionally by an inexperienced developer	9 Technology change and evolution



Propagation of Degradation



When an asset is degraded, it is likely to influence the value of other assets that depend on the degraded asset.



Propagation of Degradation

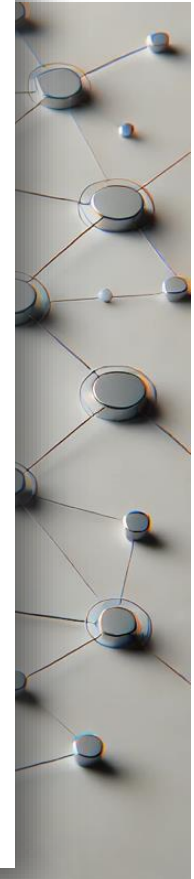
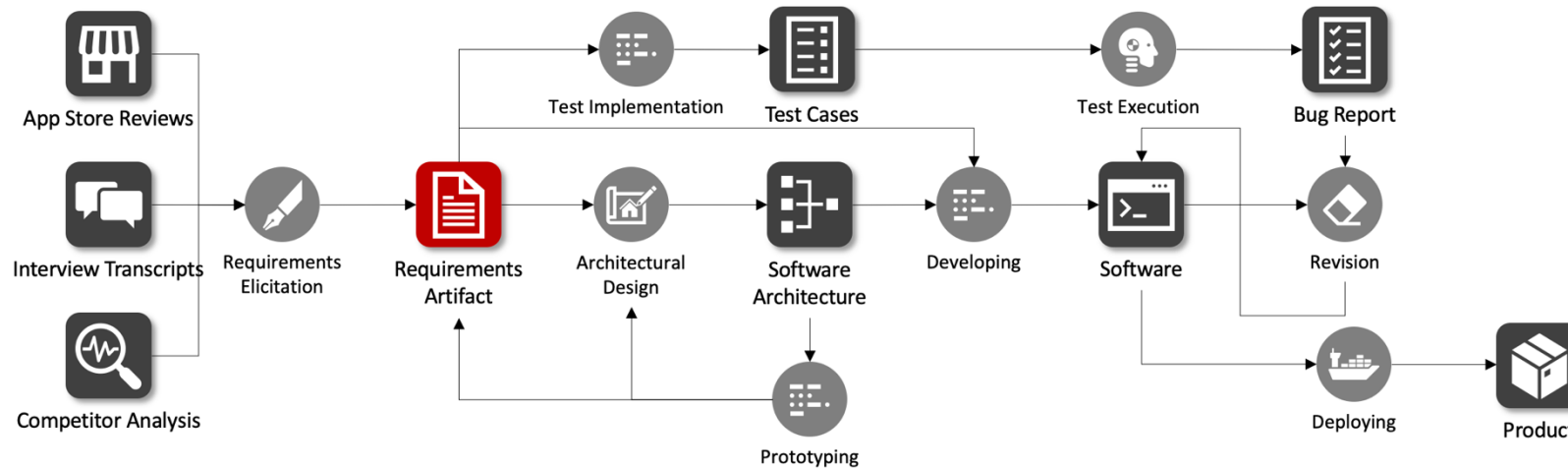


When
likely
other
degrad

Requirements impact subsequent Software Development Activities



BLEKINGE
INSTITUTE OF
TECHNOLOGY



Propagation of Degradation



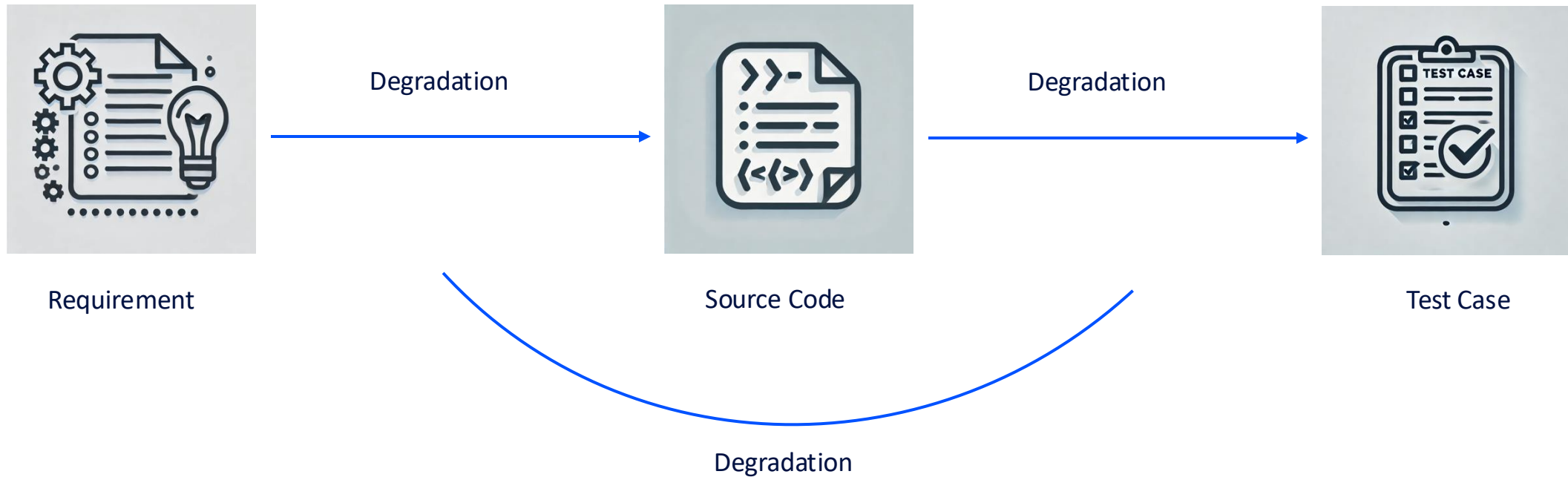
Source Code

Degradation



Test Case

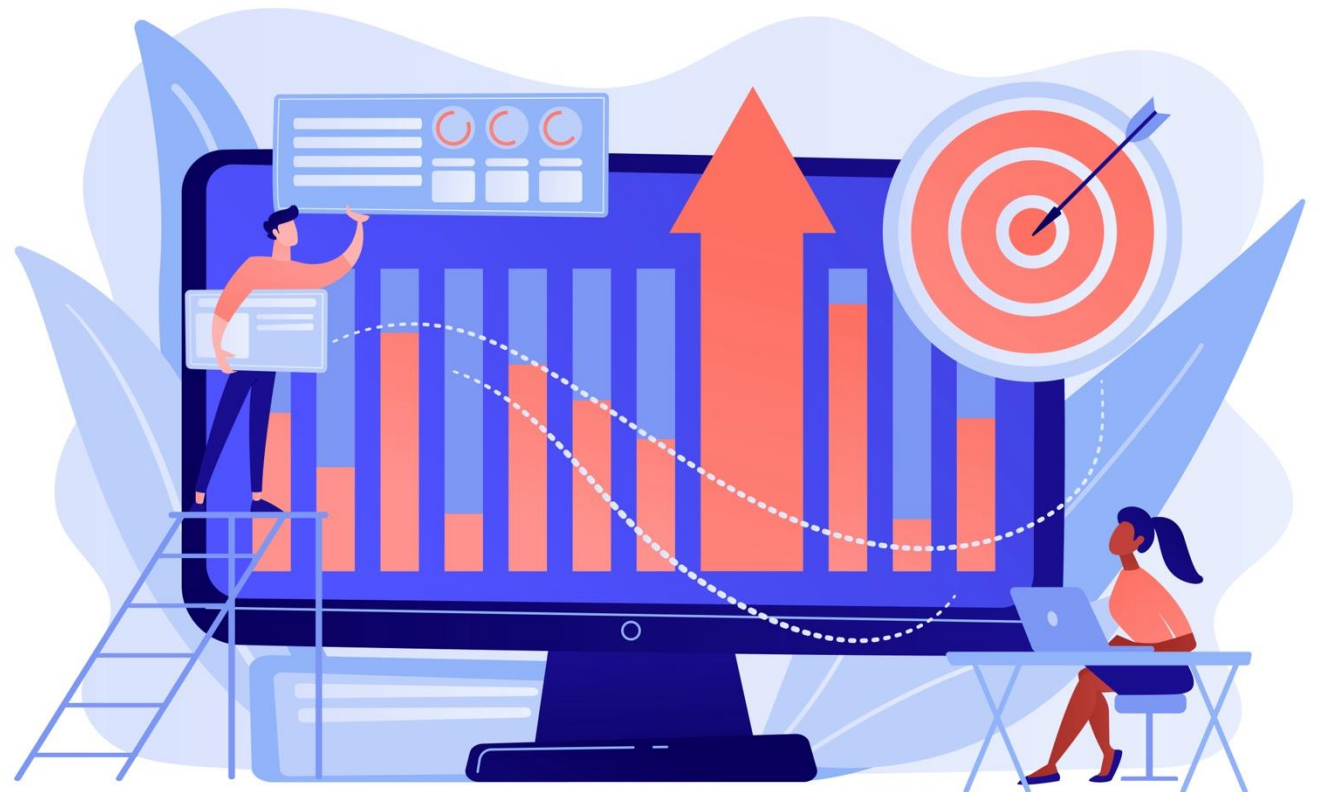
Propagation of Degradation



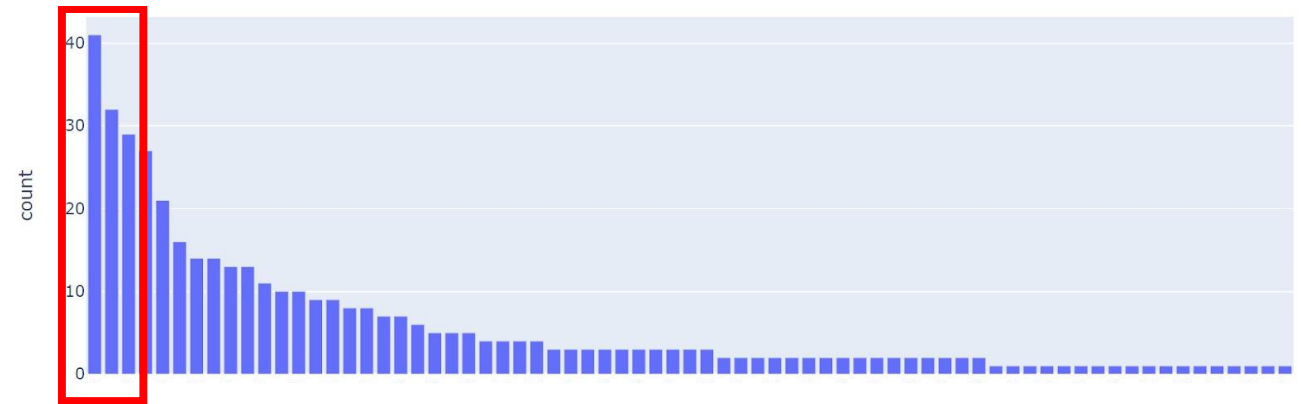
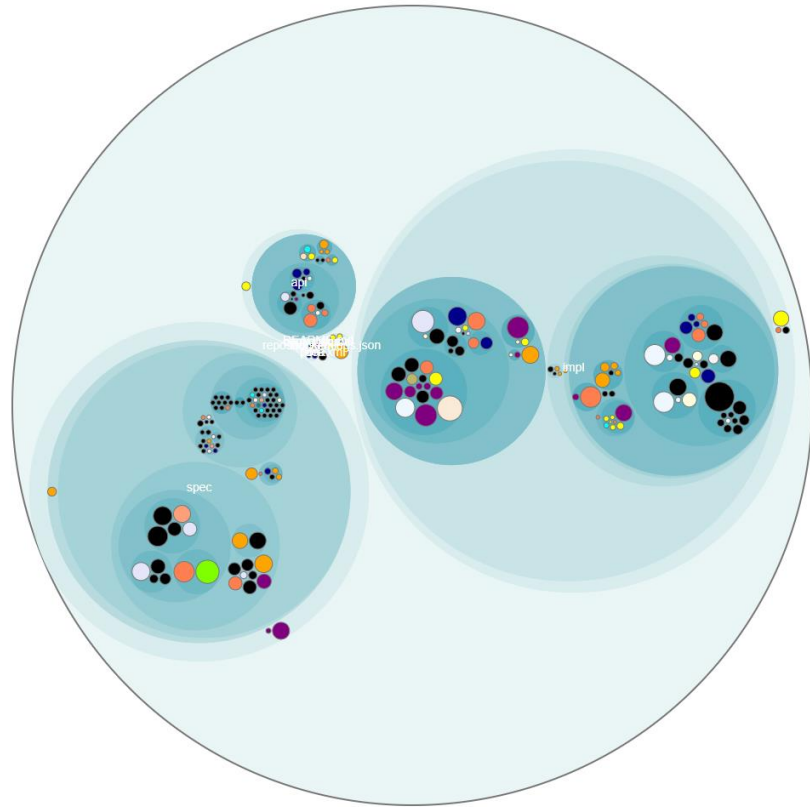
From Technical Debt To Asset Management



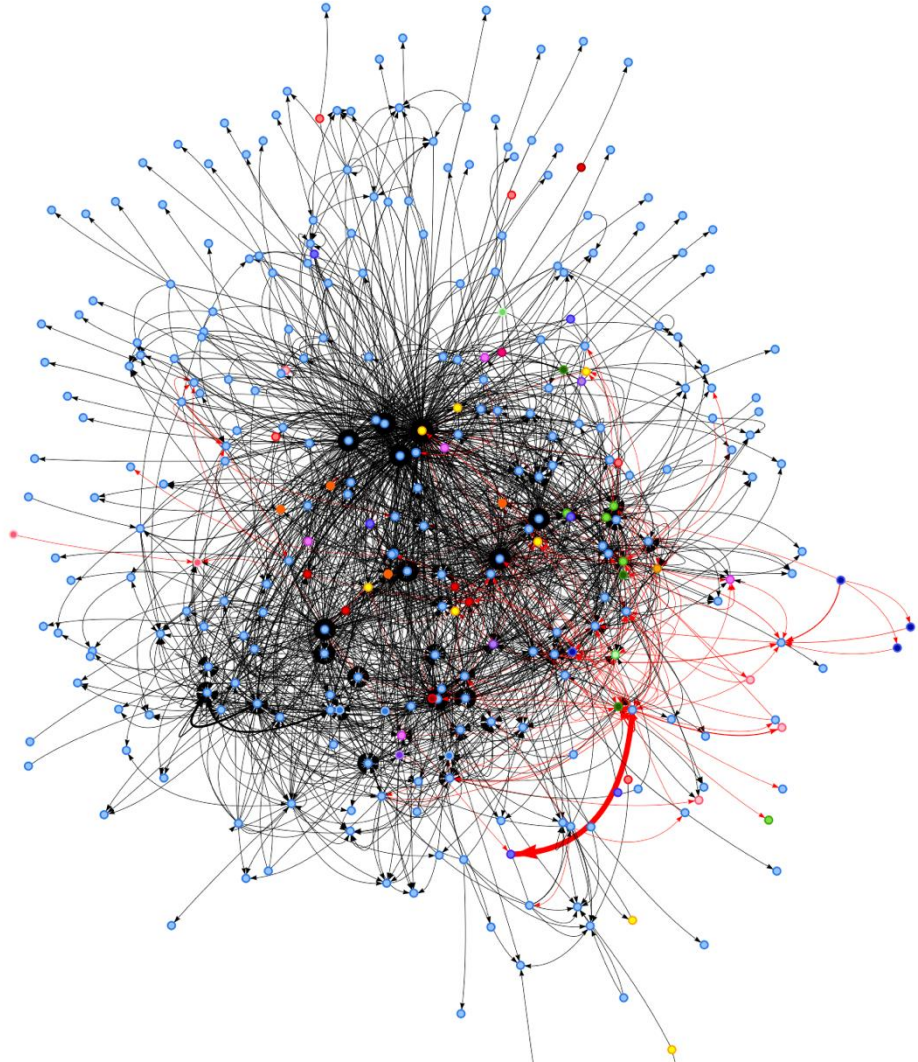
- Asset Management is the **administration of assets** and the activities that are related to creating and maintaining them as well as controlling their quality.
- TD implies the “debt” you owe given the degradation.
- We want to move from Technical Debt to Asset Management.



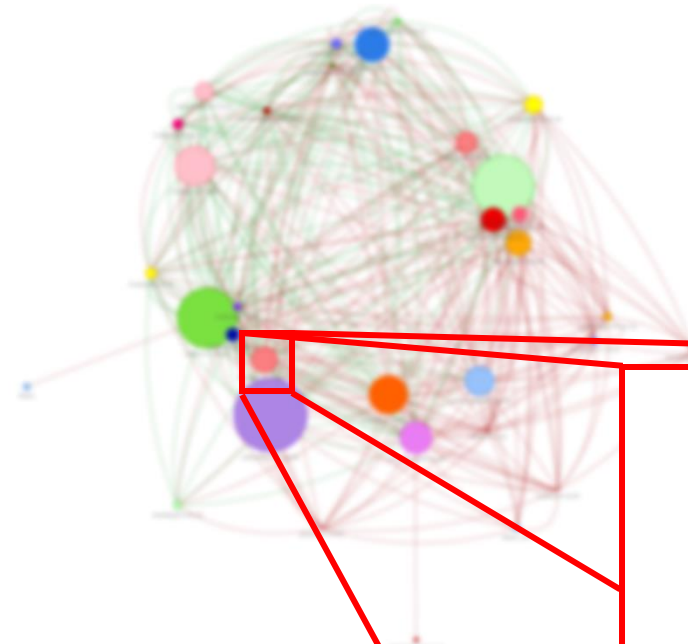
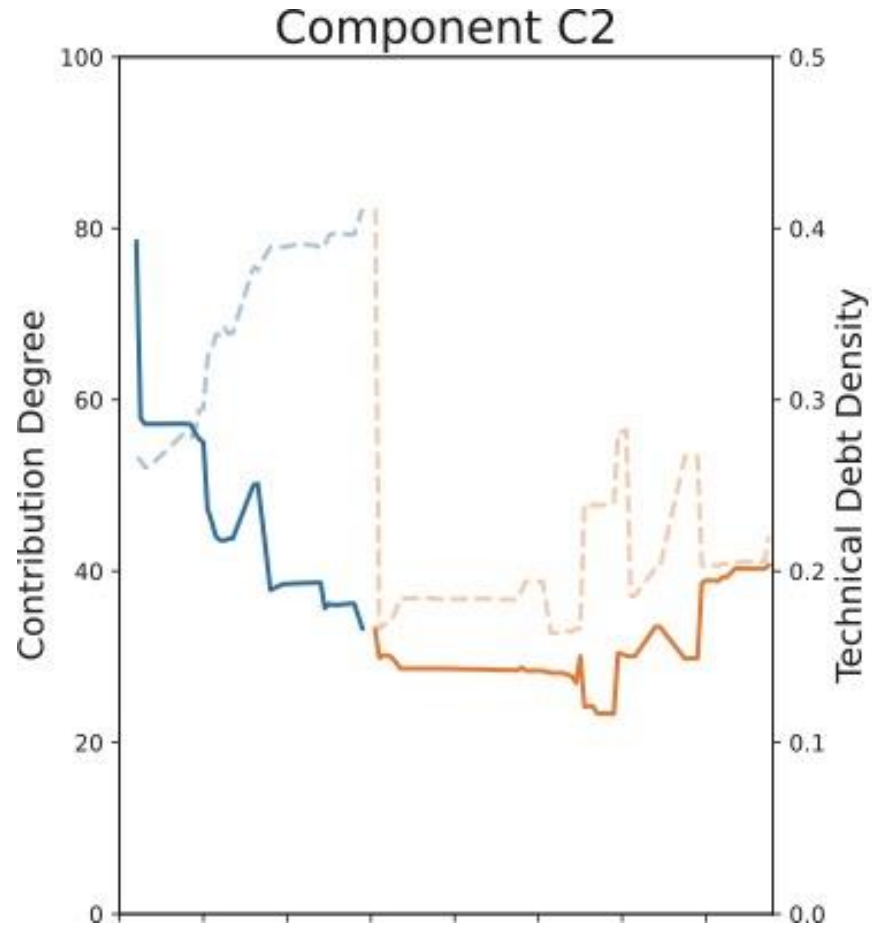
Puzzling it all together...



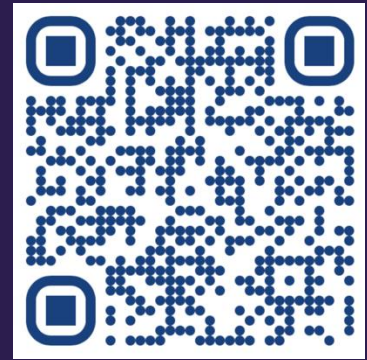
Puzzling it all together...



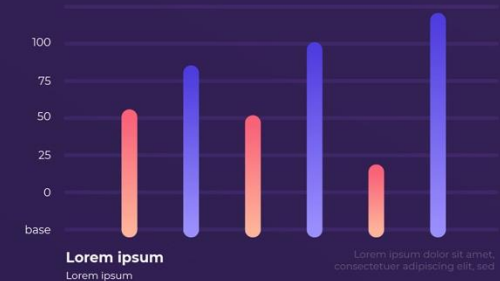
Puzzling it all together...



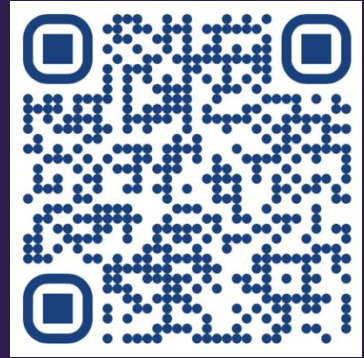
Puzzling it all together in one place



- Asset telemetry dashboard.
 - Monitor assets based on different metrics important for each organisation.
- Better decision-making with AI
 - Create projections of how things are going - better estimation based on assets' history and other organisational inputs.



Where to Begin?



Identify Assets



Identify Metrics
& Measurements



Create A Map of
Assets for Organisations

Other implications



Taxing Collaborative Software Engineering

The Challenges for Tax Compliance in Software Engineering

Michael Dorner¹, Blekinge Institute of Technology

Maximilian Capraro², Oliver Treidler, and Tom-Eric Kunz, Kolabri

Darja Šmite³ and Ehsan Zabardast³, Blekinge Institute of Technology

Daniel Mendez⁴, Blekinge Institute of Technology and fortiss

Krzysztof Wnuk⁵, Blekinge Institute of Technology



Digital Object Identifier 10.1109/MS.2023.3346646
Date of publication 25 December 2023; date of current version 12 June 2024.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see <https://creativecommons.org/licenses/by/4.0/>.

// The engineering of complex software systems is often the result of a highly collaborative effort. However, collaboration within a multinational enterprise has an overlooked legal implication when developers collaborate across national borders: It is taxable. In this article, we discuss the unsolved problem of taxing collaborative software engineering across borders. //

He's spending a year dead for tax reasons.
—Douglas Adams, *The Hitchhiker's Guide to the Galaxy*

MODERN SOFTWARE SYSTEMS are often too large, too complex, and evolving too fast for single developers to oversee. Therefore, software engineering has become highly collaborative. Further, software development is often a joint effort of individuals and teams collaborating across borders, especially in multinational companies with their subsidiaries spread around the globe.¹ However, collaboration has a legal implication if individuals collaborate across borders: *The profits from those cross-border collaborations become taxable.*

Introduction

In this article, we describe the complexity of applying the established international taxation standards required and

Exploring the Factors that Impact The Half-life of Software

Krzysztof Wnuk¹, Theresia Harrer³, Piotr Tomaszewski², and Ehsan Zabardast¹

¹ Blekinge Institute of Technology, Sweden
krw@bth.se, ehsan.zabardast@bth.se,

² RISE Research Institutes of Sweden,
piotr.tomaszewski@ri.se,

³ Hanken School of Economics, Helsinki, Finland
theresia.harrer@hanken.fi,

Abstract. This vision paper explores the factors that impact the aging and depreciation of software. Based on the exploration of related work in software aging, software anti-aging, the financial aspect of technical debt and accounting of intangible assets, we postulate that a more holistic approach towards obsolescence should be taken as most research focuses solely on the technical aspects of software aging, leaving the business and accounting aspects greatly unexplored.

Keywords: software aging, software half-life, software technical debt

1 Introduction

Software business is fiercely competitive with rapidly changing market trends, customer needs and technologies [7]. The intangible and flexible nature of software makes it a suitable mechanism to respond to these changes, however at the risk and cost of rapid obsolescence and aging of produced software artifacts. Software aging is not a new concept, it was discussed in 1994 by Parnas, who claimed that programs like people get old despite software programs being math-



Technology | Government | Corporate Governance | Tax

Microsoft says US has asked for \$28.9 billion in audit dispute

By Stephen Nellis

October 12, 2023 12:55 AM GMT+2 · Updated a year ago

🔖 Aa 🔄



A man stands inside the Microsoft Experience Center in New York City, U.S., January 18, 2023. REUTERS/Shannon Stapleton [Purchase Licensing Rights](#)

Rolling in the Debt: From Technical Debt To Asset Management

Dr. Ehsan Zabardast

More info and Survey on Technical Debt



bit.ly/techdebt2025

