TDresearchteam
Technical Debt Research Team

Towards an Ontology of Terms on Technical Debt

Nicolli S. R. Alves, Leilane F. Ribeiro, Vivyane Caires,
Thiago S. Mendes, Rodrigo O. Spínola

UNIFACS UNIVERSIDADE SALVADOR LAUREATE INTERNATIONAL UNIVERSITIES

Federal University of Bahia

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA BAHIA

Fraunhofer Project Center
for Software and Systems Engineering
at UFBA

In cooperation with
Fraunhofer

Agenda

- Introduction
  - Technical debt
  - Motivation
  - Research Goal
- Ontology of Terms on Technical Debt
- Initial Evaluation
- Final Remarks

Technical Debt (TD)

Technical Debt includes those **internal tasks you choose not to do now**, but it runs a risk of **causing future problems** if not done.

[Cunningham, 1992]

Technical Debt

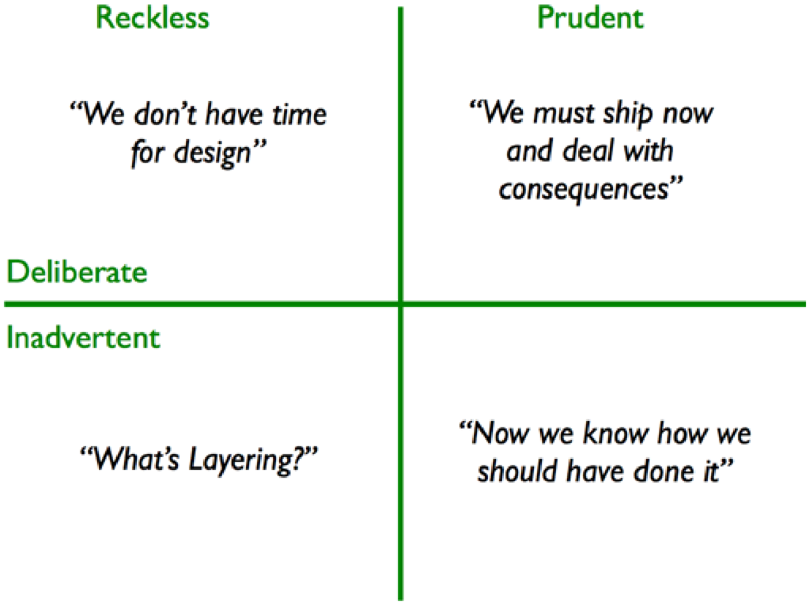
- The acceptance and use of the Technical Debt metaphor is in large part because it is easily understood;

- Technical debt is a new research area, having published papers only from 2010, its different types and indicators are not organized yet;

- Besides, it is difficult to define a common vocabulary for the area and to indicate which directions to follow in order to find out the existing debt on software projects.

Motivation

There are some initiatives on literature in order to organize the different types of Technical Debt.



Reckless

Prudent

"We don't have time for design"

"We must ship now and deal with consequences"

Deliberate

Inadvertent

"What's Layering?"

"Now we know how we should have done it"

Technical Debt Quadrant By Martin Fowler (2009)

Intentional

vs.

Unintentional Debt

By Steve McConnell (2007)

Motivation

- However, those initiatives **do not consider the nature of the debt as a factor to be considered in its classification**.

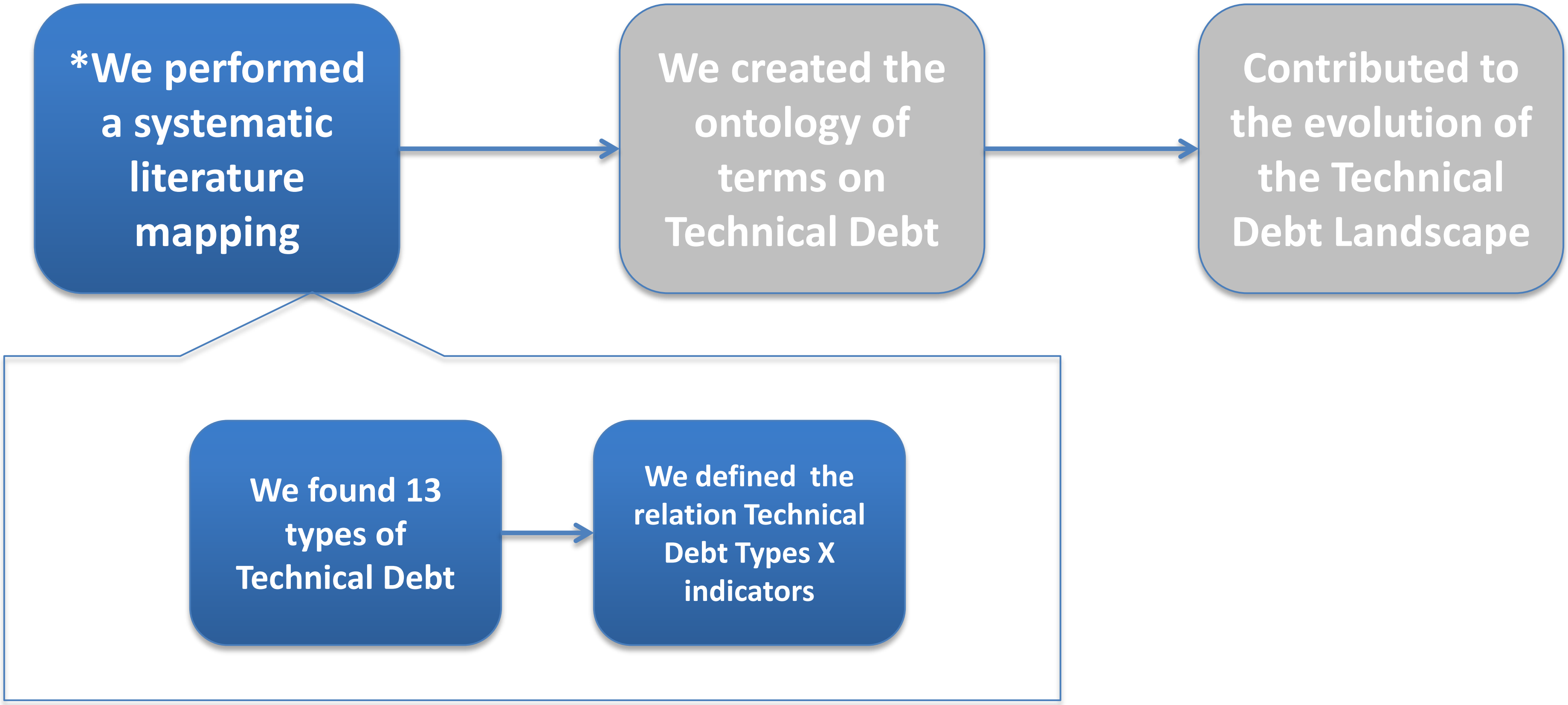
- By nature we mean the activity of the development process execution where the debt was inserted or it is associated with. For example, a debt incurred by a tester who does not execute a set of planned test scenarios can be considered a test debt.

Research Goal

- To propose an ontology to organize the different types of technical debt considering their nature as a classification criterion.

- Based on this organization, it was possible to identify different indicators that have been proposed to find out each type of debt on software projects.

Research Strategy


*We performed a systematic literature mapping
We created the ontology of terms on Technical Debt
Contributed to the evolution of the Technical Debt Landscape
We found 13 types of Technical Debt
We defined the relation Technical Debt Types X indicators

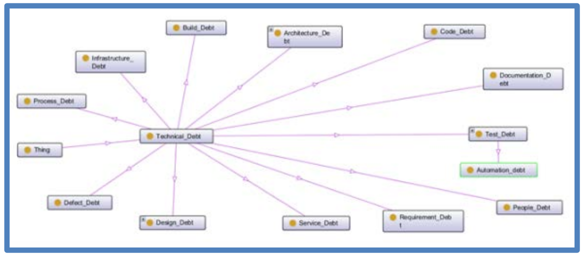
*The results of the systematic literature mapping will be available soon.

Research Strategy

Performed Systematic literature mapping

We Created the ontology of terms on Technical Debt

Contributed to the evolution of the Technical Debt Landscape



Created TD Ontology in OWL Format

Initial Evaluation Regarding Quality

Initial Evaluation by a specialist

TDresearchteam
Technical Debt Research Team

Agenda

- Introduction
  - Technical debt
  - Motivation
  - Research Goal
- Ontology of Terms on Technical Debt
- Initial Evaluation
- Final Remarks

Purpose Identification and Requirements Specification

- The first activity when building an ontology. In this activity the competence of the ontology is identified. Thus, the following competency question was defined [Falbo, 1998]:

What are the types of technical debt and their indicators that can be considered in software projects?

Identified Technical Debt Types Definitions

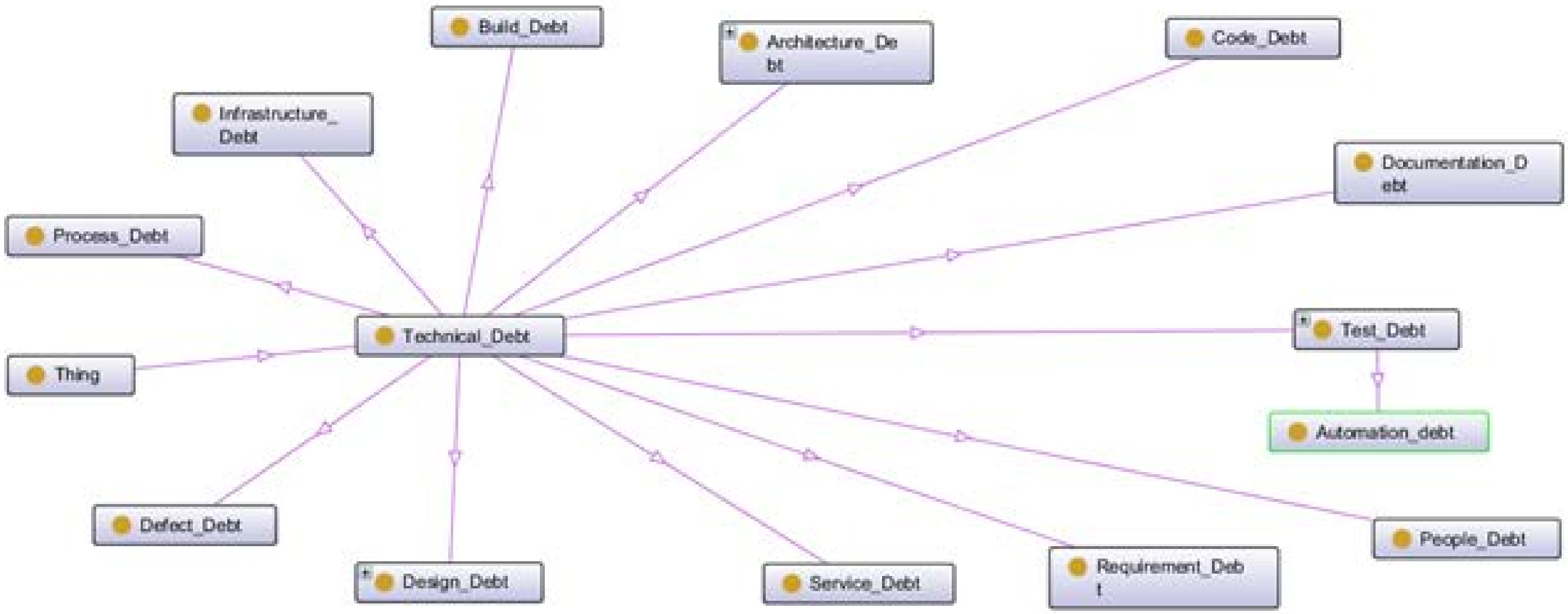
Code Debt	**Refers to the problems found in the source code which can affect negatively the legibility of the code making it more difficult to be maintained.** Usually, this debt can be identified by examining the source code of the project considering issues related to bad coding practices.

Technical Debt Types X Indicators

TD Type	Indicator
Architecture Debt	ACN/PWDR, Betweeness Centrality, Issues in software architecture Structural Analysis Structural Dependencies Violation of Modularity
Build Debt	“Dead Flags” “Zombie Targets” Dependency Visibility
Code Debt	ASA Issues Code Metrics Code outside of standards Duplicated code Multithread correctness (ASA) Slow Algorithm
Defect Debt	Uncorrected known defects
Design Debt	ASA Issues Brain Method Code Metrics Code Smells Data Class Data clumps Dispersed Coupling Duplicated Code God class (or large class) Grime Intensive Coupling Issues in the software design Refused Parent Bequest Schizophrenic Class Structural Analysis
Documentation Debt	Documentation does not exist Incomplete Design Specification Incomplete Documentation Insufficient comments in code Outdated Documentation Test Documentation
Infrastructure Debt	-
People Debt	-
Process Debt	-
Requirement Debt	Requirement Backlog List
Service Debt	Selection/Replacement of web service
Test Automation Debt	-
Test Debt	Incomplete Tests Low coverage

Design Debt	ASA Issues Brain Method Code Metrics Code Smells Data Class Data clumps Dispersed Coupling Duplicated Code God class (or large class) Grime Intensive Coupling Issues in the software design Refused Parent Bequest Schizophrenic Class Structural Analysis

Ontology visual representation in Protegé



Implemented with OWL (Web Ontology Language)

Agenda

- Introduction
  - Technical debt
  - Motivation
  - Research Goal
- Ontology of Terms on Technical Debt
- Initial Evaluation
- Final Remarks

Initial Evaluation

- The proposed ontology was assessed in two steps:

1. The quality criteria defined in Gruber (1995);

2. An expert that did not participate of the ontology development performed an initial evaluation on it.

Initial Evaluation Regarding Quality

- The ontology was sent for two of the researchers involved in the writing of this paper.

- For each Technical Debt type, the researcher was asked to evaluate it considering the following possible results:
  - **Fully Compliance (FC):** the ontology complies with all variables considered in the criterion being evaluated;
  - **Partially Compliance (PC):** the ontology is partially consistent with the variables considered in the criterion being evaluated;
  - **Not Compliance (NC):** the ontology does not comply with all variables considered in the criterion.

Results

Criteria	Evaluation		Remarks
	1	2	
Clarity: An ontology should effectively communicate the intended meaning of defined terms. Definitions should be objective and, when a definition can be stated in logical axioms, it should be. All definitions should be documented with natural language [13].	PC	PC	Logical axioms were not used in the formalization of the ontology because it is a lightweight ontology and the formalized knowledge does not require. On the other side, the TD types were defined and documented. Some minor adjustments were required in order to clarify the difference between some TD types.
Coherence: an ontology should sanction inferences that are consistent with the definitions. At the least, the defining axioms should be logically consistent. Coherence should also apply to the concepts that are defined informally [13].	PC	PC	The defined ontology did not use axioms. Thus, this criterion was used to evaluate the documented definitions. These criteria received PC result from both evaluators just because it was not possible to evaluate axioms (that does not exist in lightweight ontologies).
Extendibility: An ontology should be designed to anticipate the uses of the shared vocabulary. In other words, one should be able to define new terms form special uses based on the existing vocabulary, in a way that does not require the revision of the existing definitions [13].	FC	PC	The defined ontology is the first step towards a more comprehensive knowledge organization in TD considering its types, causes and indicators. Thus, it is expected that it will be evolved from the definitions here formalized. In this context, one of evaluators requested that some definitions were improved to facilitate its extension.
Minimal encoding bias: The conceptualization should be specified at the knowledge level without depending on a particular symbol-level encoding [13].	FC	FC	The defined and documented concepts in the ontology were not influenced by the restrictions of the chosen language (OWL) for their representation.
Minimal ontological commitment: An ontology should require the minimal ontological commitment sufficient to support the intended knowledge sharing activities [13].	FC	FC	The defined ontology organizes a common vocabulary for the DT area, is extensible and does not make use of a very extensive formalism making it use easier to the research community.

The results showed the need to make some adjustments to improve the description of some technical debt types..

Initial Evaluation by a Specialist

- The defined ontology on Technical Debt was evaluated by a specialist in the area who was not involved on its definition.

- As result, some improvements were suggested in order to clarify some definitions that, in some cases, were described from different points of view.

- The specialist considered it important to keep the concepts in the proposed ontology because this can be a good material for discussion by the research community.

Agenda

- Introduction
  - Technical debt
  - Motivation
  - Study Goal
- Ontology of Terms on Technical Debt
- Initial Evaluation
- Final Remarks

Contributions of work

- This work contributes to the **Technical Debt Landscape**.
- The identified technical debt types were organized using an ontology, which will allow the **sharing of a common vocabulary for the research community** on Technical debt.
- Furthermore, we organized the **indicators of debt for each type of technical debt**.

Future works

- We are working on a web based infrastructure (wrapping the defined ontology) to allow the sharing and the collaborative maintenance and evolvement of this knowledge on Technical debt.

- Besides that, our research group is also working on a set of tools to support the visualization of Technical debts on software projects.

References

- W. Cunningham, “The Wycash Portfolio Management System,” in ACM SIGPLAN OOPS Messenger (Vol. 4, No. 2). ACM. December 1992, pp. 29-30.

- M. Fowler, “Technical Debt Quadrant,” Bliki [Blog]. Available from: http://www.martinfowler.com/bliki/TechnicalDebtQuadrant.html, 2009.

- S. McConnell, “Technical Debt. 10x Software Development,” [Blog]. Available at: http://blogs.construx.com/blogs/stevemcc/archive/2007/11/01/technical-debt-2.aspx, 2007.

- R. A. Falbo, C.S. Menezes, A.R.C. Rocha, “A Systematic Approach for Building Ontologies,” Proc. of the 6th Ibero-American Conference on Artificial Intelligence, Portugal, Lecture Notes in Computer Science, vol. 1484, 1998.

- T.R. Gruber, “Toward Principles For The Design Of Ontologies Used For Knowledge Sharing,” Int. Journal Human-Computer Studies, 43(5/6), p. 907-928, 1995.

Questions

?