Technical Debt at Scale

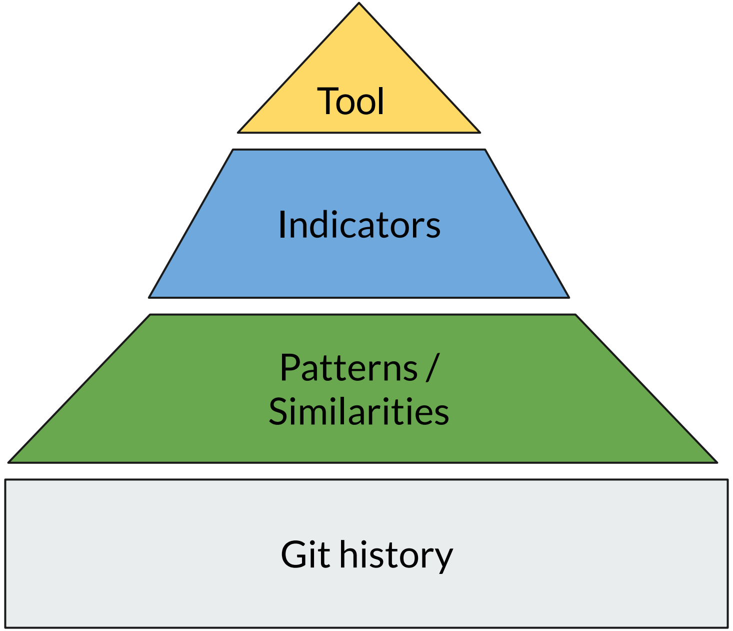
Jonas Grunert - Code Repository Mining - 2.6.2020 - SS 2020

Agenda

1. Motivation
2. Definition
3. Repository Selection
4. Methodology
5. Outlook
6. Literature

Motivation

Building a tool, that warns of tech debt, based upon indicators, which can be found in similarities and patterns in the Git history



Tool

Indicators

Patterns / Similarities

Git history

Definition

Technical Debt

is the **work needed** to take a system from the current to an ideal state. (Cunningham)

generates value in the short term and **interest in the long term**. (Cunningham)

It is environment specific and can be racked up strategically or **recklessly**. (Fowler)

Calculating Technical Debt

SQALE Method

- Breach of Quality Model
- Cost to repair
  Cost not repairing
- Proportion to project budget
- Rating, Pyramid, Heat Map

SIG-MM

- SW Maintainability Metrics
- Unitless risk profile
- Rework Fraction
- Maintenance Fraction

Open Source Delta Maintainability Model

- LOC (Diff) based
- SIG-MM Properties

Repository Selection

Repos with “Technical Debt” mentioned in the commit message

511 Repos

Long running repos between 2015-2019

583.766 Repos

The Technical Debt Dataset
SonarQube and PyDriller
Data from Tampere University (2019)

32 Repos

Methodology

Collect Metainformation

- per Commit
- Use PyDriller
- Histogram of metadata

e.g.

- Test-Coverage
- Change of maintainer
- Change fraction
- Commit size

Calculate TD-Index

- per Commit
- Use SW Metrics
- Histogram of Techdebt

e.g.

- McCabe
- LoC to Comments
- Dead Code Ratio

Only few Repositories

- Just one language
- Long running projects
- High Probability for tech debt

Methodology

Visualization

- Show the metrics
- Comparable for human eye
- Find possible hiccups in the data

Cluster (K-Nearest Neighbor)

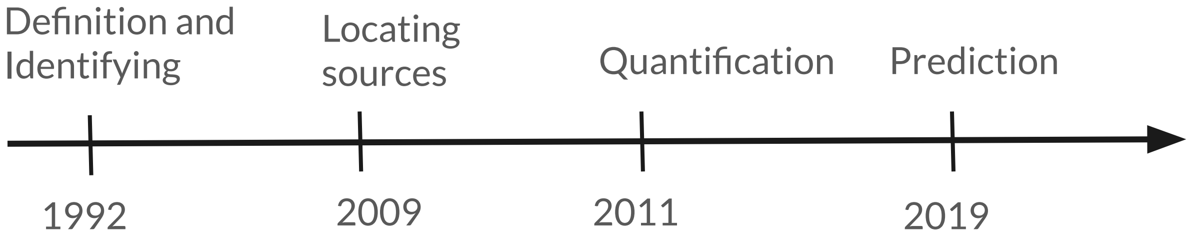
- Cluster over repositories
- Find patterns
- Based on a multidimensional score

Linear Regression

- Predicting metric changes based on meta information
- Warn when metrics get volatile

Outlook

1. Visualization helps every developer
2. Finding patterns, helps in preventing the same errors occuring twice
3. Prediction helps while developing



Definition and Identifying

1992

Locating sources

2009

Quantification

2011

Prediction

2019

Literature

- Cunningham, Ward: Proc. OOPSLA 1992, http://c2.com/doc/oopsla92.html
- Fowler, Martin: Technical Debt Quadrant, https://martinfowler.com/bliki/TechnicalDebtQuadrant.html
- Letouzey, Jean-Louis: Managing Technical debt with the SQALE Method, http://www.sqale.org/wp-content/uploads/2016/04/itj1603_JLL.pdf
- SIG: An Empirical Model of Technical Debt and Interest, https://dl.acm.org/doi/pdf/10.1145/1985362.1985364
- SIG: The Delta Maintainability Model, https://pure.tudelft.nl/portal/files/53625004/deltamaintainability.pdf
- Tampere:The Technical Debt Dataset, https://dl.acm.org/doi/abs/10.1145/3345629.3345630