An Empirical Study of Technical Debt in Open-Source Software Systems

Reem Alfayez, Celia Chen, Pooyan Behnamghader, Kamonphop Srisopha, Barry Boehm

What is Technical Debt

- **In 1992, Ward Cunningham described technical debt as writing immature or "not quite right" code in order to ship a new product to market faster.**
- **Technical Debt consists of:**
  - **Principle:** measures the cost or effort for eliminating technical debt .
  - **Interest:** measures the extra cost or effort over some period of time incurred for NOT eliminating the technical debt.

Technical Debt Examples

"No one understand this code except Tom"

"Let's finish the code now and document later.."

"`// Should be changed before the release`"

"You can copy this code and use it."

"Do not touch this module everything else will break if you do"

"We will test it after the release"

"You work on this part and I will work on the other part and we will integrate later"

Why Do We Take Technical Debt

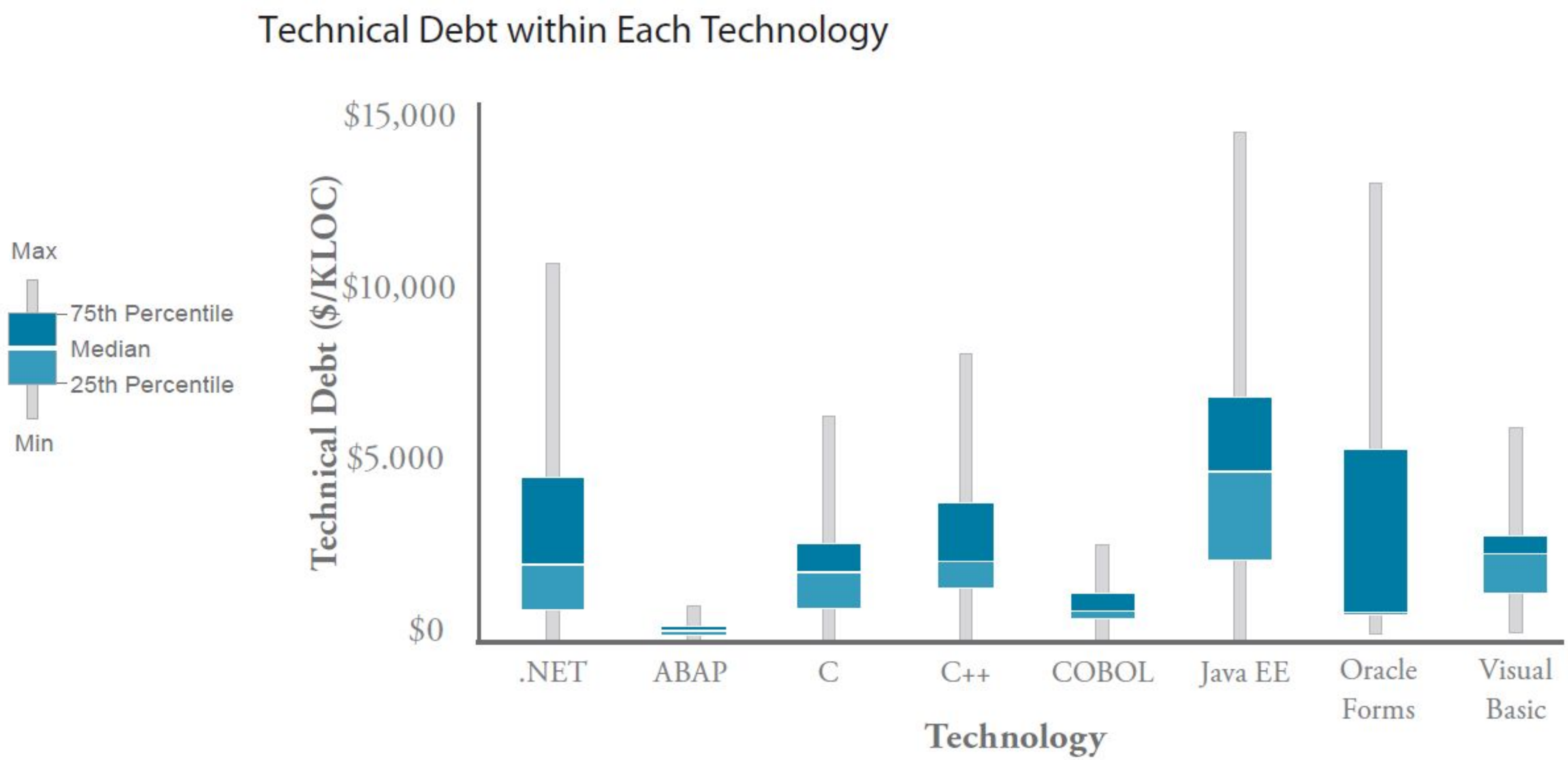
- Release faster.
- Decrease current release cost.
- Gather more information.
- Delay decisions.

Technical Debt Consequences

- Increased time to delivery.
- Increased number of defects.
- Raising maintainability cost.
- Decreased customer satisfaction.

- **Based on the analysis of 1400 applications containing 550 million lines of code submitted by 160 organizations, the average Technical Debt per LOC of $3.61.**

http://www.castsoftware.com/resources/resource/whitepapers/cast-report-on-application-software-health?gad=RLP&EID=e304258b-d795-4c50-93a9-1de71853cd3e



The average cost of Java apps was even higher: $5.42 per line of code.

http://www.castsoftware.com/resources/resource/whitepapers/cast-report-on-application-software-health?gad=RLP&EID=e304258b-d795-4c50-93a9-1de71853cd3e

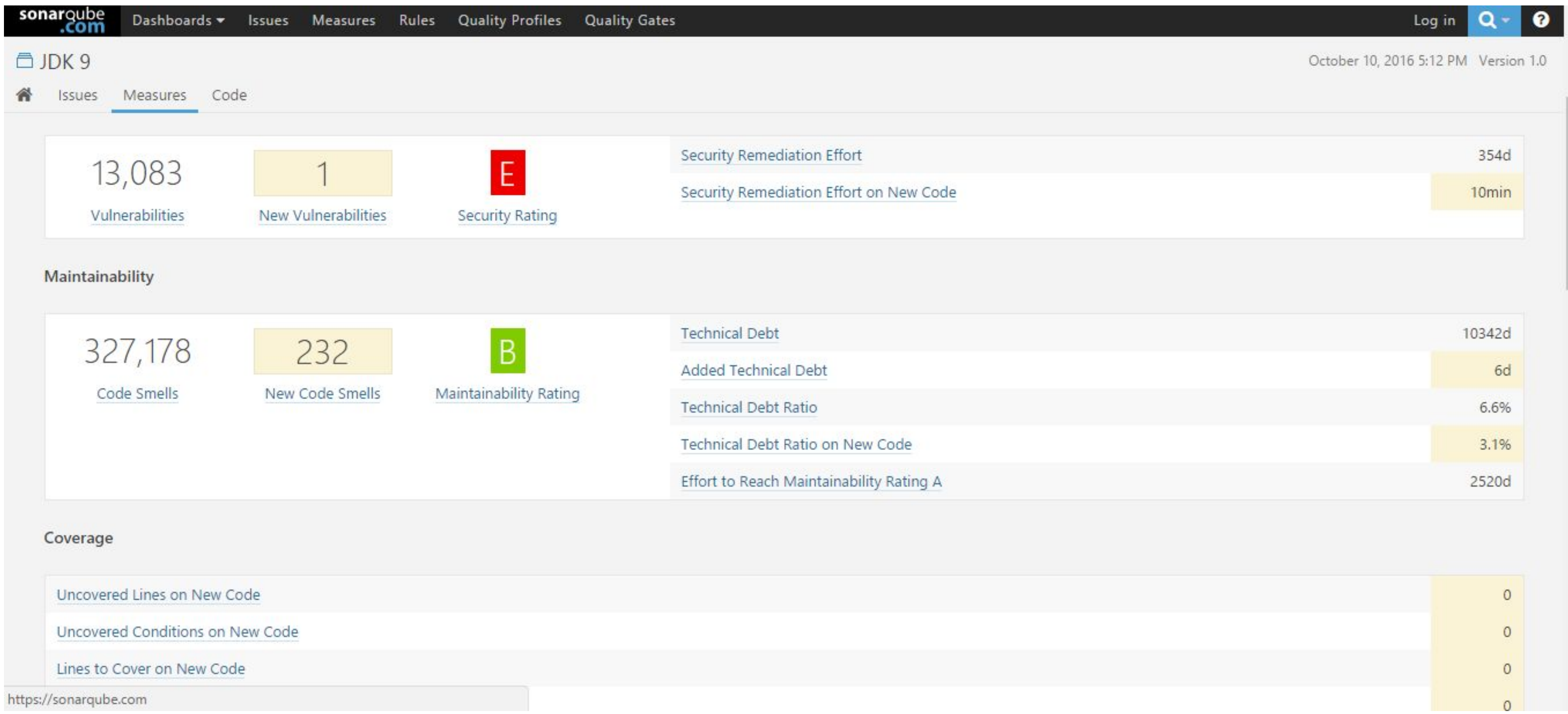
Research Questions

Research Question #1: Does the size of the source code relate to the total technical debt and the technical debt density.

Research Question #2: Do the total technical debt and the technical debt density in a software vary among domains?

Research Question #3: Do system development and management decisions including number of commits, releases, branches, and contributors relate to the total technical debt and the technical debt density?

Technical Debt Calculation



Technical Debt Calculation

Debt(in man days) = cost_to_fix_duplications
+ cost_to_fix_violations
+ cost_to_comment_public_API
+ cost_to_fix_uncovered_complexity
+ cost_to_bring_complexity_below_threshold

Data collection

- More than one official releases.
- Latest stable release source code is available.
- Software system falls under one and only done domain.
- The programming language is only java.
- Well-presented in the community.
- Active Git repository.

Data collection

Apache Software Foundation (Java)

CHARACTERISTICS OF SYSTEM DATA SOURCES

Domains	Number of systems	Average LOC
Big Data	16	44992
Database	13	52610
Library	35	113612
Network Server	9	20624
Web Framework	11	31164
XML	7	51569

Data analysis - (1/3)

- **Evaluation on size hypothesis (RQ#1)**
  - K-means Cluster analysis
    - Help to find pattern in the data based on their similarity.
    - We cluster them based on the size of systems using a clustering algorithm
      - Examine whether total TD and TDD differ significantly among each cluster

Results on size hypothesis

- **Larger systems** have more technical debt in total but less technical debt density while **smaller systems** have less technical debt in total but higher technical debt density

TABLE III
CLUSTER MEANS

Cluster	Technical Debt	LOC
1	2074.1818	120362.64
2	346.2475	18070.42

TABLE IV
CLUSTER MEANS

Cluster	Technical Debt Density	LOC
1	19.48511	120362.64
2	20.06992	18070.42

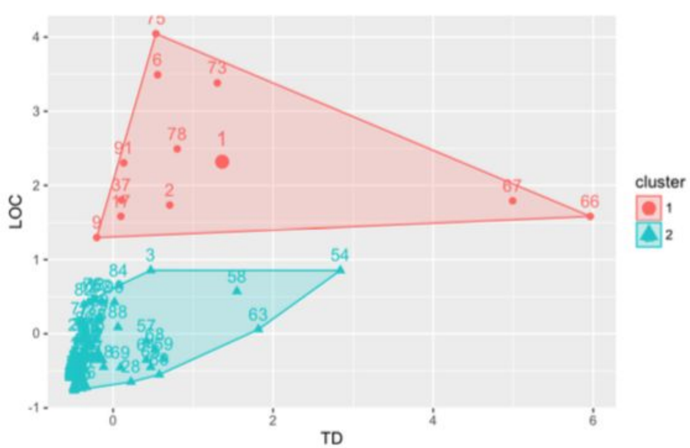


Fig. 1. 2D representation of the cluster solution of Technical Debt

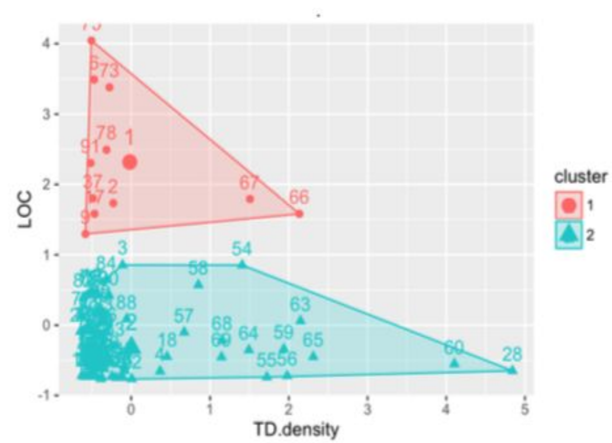


Fig. 2. 2D representation of the cluster solution of Technical Debt Density

Data analysis - (2/3)

- Evaluation on domain hypothesis (RQ#2)
  - We perform various statistical analysis
    - Levene's Test.
    - Welch ANOVA.
    - Games-Howell Test.

Results of Levene's test

- Unequal variances (F = 6.117, p = 6.912e-05) for TD

and unequal variances (F = 4.9892, p = 4.695e-04)for TDD.

- Since the p-value of Levene's test is less than 0.05, we concluded that the variances of the six domains are significantly different.
- These unequal variances suggest that we cannot use one-way ANOVA (violate one of the assumptions for one-way ANOVA)

Welch ANOVA is used in this case

Results of Welch ANOVA

1. **An alpha level of 0.05 is used.**
2. **Total technical debt of six domains**
   - ***Welch's F*(5, 23.508) = 4.2964 with *P*= 0.006408**
     - **(Not all domains have the same TD)**
3. **Total technical debt density of six domains**
   - **Welch's F(5, 25.47) = 5.2781 with *P* = 0.001848**
     - **(Not all domains have the same TDD)**

Games-Howell Test

What about unique pairwise comparison? => Games-Howell Test

Results

Library and **Big Data** have significantly different TD at 0.1 level of significance, **not significant for other comparisons**

Many pairwise appear to have significantly different TDD at the 0.1 level of significance.

- **XML with Network**
- **Web Framework with Network**
- **Web Framework with Big Data**
- **Database with Big Data**

Data analysis - (3/3)

- Evaluation on system development and management decisions hypothesis (RQ#3)

Pearson Correlation Test

- Significance level is set to 0.05 Confidence level=95%

Results of Pearson Correlation

Technical Debt

CORRELATION COEFFICIENTS MATRIX BETWEEN TOTAL TECHNICAL DEBT AND SYSTEM DEVELOPMENT AND MANAGEMENT DECISION FACTORS

	N	R-value	p-value
Number of Branches	91	.05071	.63308
Number of Releases	91	.26114	.01241
Number of Commits	91	.51619	1.63135e-7
Number of Contributors	91	.12618	.23335

Results of Pearson Correlation

Technical Debt Density

CORRELATION COEFFICIENTS MATRIX BETWEEN TECHNICAL DEBT DENSITY AND RESEARCH QUESTIONS FACTORS

	N	R-value	p-value
Number of Branches	91	-0.04658	.66108
Number of Releases	91	.12826	.22567
Number of Commits	91	.17108	.10493
Number of Contributors	91	-.02188	.83688

Conclusion

- We examined 91 Apache Java OSS projects.
- We employed various statistical methods to investigate how TD and TDD relate to different system characteristics, development, and management decisions.
- The size of software system and its domain can impact its TD and TDD significantly.
- Number of system releases and commits have a significant positive relationship with TD.
- Results show no significant relationship between TD and the number of contributors and branches.
- No significant relationship between TDD and any of the system development and management decisions.

Future work

- Further the study to understand the reasons behind these relations.
- Conducting a biopsy analysis on these systems.
- Goal: provide guidelines for decision makers to help them study the tradespace by providing what factor(s) introduce more TD to the system and the quality per capita in the systems.