university of groningen

Are developers explicitly documenting technical debt? An exploratory study on open-source projects

Candidate:	João Paulo Biazotto, MSc.
Email:	j.p.biazotto@rug.nl

1 Introduction

2 Background

3 Research Method

3.1 Objective definition

In this study, we aim to understand if, and in what extension, developers explicitly report TD on issues and code comments. To do that, we used **"smell"** as a keyword to filter issues and commits in the projects. Code smells and architecture smells are a well-know type of TD. In this scenario, if this word is used in in commit messages or in issues comments, for instance, probably developers are documenting TD.

3.2 Projects' selection

Selecting which projects will be analyzed is a crucial decision to be made during an exploratory study. This decision is directly related to the number of studies and the quality of the data, thus, it impacts the external validity of the study.

In our study, to select the projects we proceeded with a manual search using Github API[1]. In this step, we collected 300 issues that mention the keyword *"smell"* from repositories of some users, listed in Table 1.

After analyze these projects, we selected those with at least five issues in our sample (300 issues), and then conducted the search again, now inside each project. The list of projects that we selected and the number of issues in each one is in Table 3.

SELF-FIX

We only considered in the analysis of Table 4 that issues which:

1. Contains the keyword smell; and
2. Were solved by a commit.

Thus, even if the issue was fixed by its author, the analysis would not cover it unless there was a commit closing the issue.

[1] https://docs.github.com/en/rest

#	User	Projects
U1	Apache	25
U2	GitHub	3
U3	IBM	2
U4	Oracle	1
U5	Microsoft	48
U6	Google	24
U7	Eclipse	19
U8	Netflix	3
U9	Shopify	5
U10	Atom	3
	Total	**133**

Table 1: Projects per user

#	User	Project	Issues
P1	Eclipse	jkube	24
P2	Facebook	flow	14
P3	Facebook	jest	7
P4	Facebook	react	28
P5	Microsoft	microsoft-ui-xaml	5
P6	Microsoft	pylance-release	5
P7	Microsoft	terminal	10
P8	Microsoft	typescript	79
P9	Microsoft	vscode	59
P10	Microsoft	wsl	10
		Total	**241**

Table 2: Issues per project

References

#	Project	LoC	Files
P1	jkube	495	14
P2	flow	192	3
P3	jest	112	4
P4	react	158	2
P5	microsoft-ui-xaml	113	3
P7	terminal	166	5
P8	typescript	4532	11
P9	vscode	81	2
	Means	731*	5

Table 3: Modification by commits (means)

#	Project	Self-Fixed	Non Self-fixed
P1	jkube	9 (75%)	3 (25%)
P4	react	2 (10%)	18 (90%)
	Total	11 (35%)	21 (65%)

Table 4: Issues solved by commits