A Systematic Mapping Study on Software Comments Analysis

Amanda F. O. Passos
Federal Institute of Bahia and
Salvador University
Salvador, Brazil
amandainf@gmail.com

Mário André de F. Farias
Federal Institute of Sergipe
Aracajú, Brazil
mario.andre@ifs.edu.br

Crescencio Lima
Federal Institute of Bahia and
Federal University of Bahia
Salvador, Brazil
crescencio@gmail.com

Manoel Mendonça
Department of Computer Science,
Federal University of Bahia
Fraunhofer Project Center @ UFBa
Salvador, Brazil
manoel.mendonca@ufba.br

Rodrigo Oliveira Spínola
PPGCOMP, Salvador University
Fraunhofer Project Center @ UFBa
State University of Bahia
Salvador, Brazil
rodrigo.spinola@unifacs.br

Abstract—***Context*: Mining software repositories has been used as an important tool to support software engineering research. Recent studies indicate that code comments are one of the most explored objects of analysis in the area. *Objective*: This work investigates how analysis of comments has been used to support software engineering activities by identifying its purposes, focuses, techniques, tools, evaluation methods, and the research type performed in the area. *Method*: We performed a systematic mapping study of the literature that considered papers from 1990 to 2016. *Results*: We analyzed 36 primary studies. The collected data pointed out that comment analysis has been used mainly for understanding and identifying the quality of software artifacts. The Dictionary/Vocabulary and Natural Language Processing are among the most used techniques, and most of them are performed in a semiautomatic way. We also organized a set of tools that have been used for mining software comments. Most of the primary studies are a solution proposal paper. Regarding evaluation methods, we found that experiments and case studies are the most considered. *Conclusion*: The results of this mapping study can help to identify points that still require further investigation in comment analysis research.**

Keywords-Code comment analysis; mining software repository; systematic mapping study.

1. Introduction

Software repositories contain large amount of historical data embedded in different artifacts, such as source code, commit data, logs, e-mails, and comments. These data usually have rich cue to support the understanding of code changes, defects, quality issues in the evolution of software projects, and so on [10]. The Mining Software Repositories (MSR) area focuses on analyzing and cross-linking the data available in different types of repositories to discover useful information about software projects [8].

DOI reference number: 10.18293/SEKE2018-013

MSR has been used as an important tool to support research on Software Engineering (SE) with different purposes such as prediction of defects analyzing commit data, identification of defects analyzing bug track system, comprehension of software evolution analyzing e-mail and commit data, and identification of technical debt [1] analyzing code comments. Recently, Farias *et al.* [4] performed a systematic mapping (SM) study to analyze studies on MSR by considering five editions of Working Conference on Mining Software Repositories (MSRConf). They reported that comments analysis is one of the most explored objects of study in the MSR are.

Comments analysis can, for example, reveal information such as the reason for adding new lines of code, knowing the progress of a collective task, or even why relevant changes were performed. Besides, code comments may also describe the developers' point of view about quality issues in the software development [5][6]. Due to such diversity of topics, our research group has faced some difficulty to have a broad view of the area when was starting the development of a new technology to support the identification of technical debt through code comment analysis.

Although some secondary studies have been performed in the MSR area [3][4][9][10], none of them has focused on comments analysis as object of study. It would be beneficial to have a broad view of the current research that has been performed on the area, so new directions of research could be better defined. In this context, this work presents the results of a mapping study performed to investigate the following research question: *"How has comments analysis been explored with the purpose of supporting software engineering activities?"*. By answering this question, we intend to identify which purposes, focuses, techniques, tools, research types and evaluation methods have been considered on the research on comment analysis to support software engineering.

In total, 36 primary studies were selected for data extraction. We identified that analysis of comments has mainly been explored with the purpose of "Comprehension" and "Identification" of software engineering artifacts. Concerning the focuses, we observed that the most considered was "Quality of Software Artifacts" followed by "Technical Debt". We also identified that the most commonly used mining techniques are Vocabulary/Dictionary, and they are usually performed in a semiautomatic way. Our analysis also concluded that the majority of studies can be characterized as "Solution Proposal". However, we identified a rising number of "Evaluation Paper" in the last few years. As for the empirical evaluations, we verified that most of the studies have carried out "Controlled Experiments" followed by "Case Studies".

We believe that the results of this mapping study will be beneficial for both researchers and practitioners. For the research community, this mapping will provide information about the current state of comments analysis research, as well as topics that require further investigation. For practitioners, the study presents a set of techniques and tools that can be used to improve or develop new approaches to explore comments analysis.

The remainder of this paper is structured as follows: Section 2 presents some related works. Next, Section 3 describes the systematic mapping protocol. Section 4 discusses the main outcomes of the study. Section 5 considers the threats to the validity of this study. Finally, Section 6 presents some concluding remarks.

2. Related Works

To the best of our knowledge, there are four secondary studies on MSR area. Kagdi *et al.* [10] performed a comprehensive literature survey on approaches for MSR in the context of software evolution. As a result, the authors proposed a taxonomy of different terms used by researchers for presenting purpose, focus, and object of analysis into categories. In another work, Hemmati *et al.* [9] analyzed 117 papers published in the MSRConf between 2004 and 2012. They codified a set of guidelines, tips, and recommendations, and provided a set of best practices that can be continuously used and updated as the MSR community matures and advances.

Next, Demeyer *et al.* [3] aimed to identify how the research on MSR evolved in the last decade. They focused on: (i) outdated research topics, (ii) the most (and less) frequently cited cases, (iii) emerging mining infrastructure, and (iv) software engineering state-of-the-practice. Finally, Farias *et al.* [4] investigated recent studies on MSR approaches collecting data about software analysis goals (purpose, focus and object of analysis), data sources, evaluation methods, tools, and how the area is evolving.

The mapping study presented in this paper and the works discussed above are complementary to each other. Different from the others, our mapping study has a more specific focus and intends to investigate how comments analysis have been explored in MSR area with the purpose of supporting software engineering activities.

3. Systematic Mapping Protocol

This work follows a well-organized set of guidelines for carrying out SMs in the context of software engineering [7], and the defined protocol is presented in the next subsections.

A. Definition of Research Questions

For this study, a primary research question (RQ) was defined: *"How has comments analysis been explored with the purpose of supporting software engineering activities?"*. The following complementary research questions were derived from this main one. By answering these questions, we will have a detailed characterization of the identified studies.

RQ1. Which are the main purposes and focus of researches on analysis of comments?

The objective of this question is to identify the goal of MSR approaches in the area of analysis of comments. To perform the classification of the extracted data, we used a taxonomy (available at https://goo.gl/qO6nUa) discussed by Farias *et al.* [4]. By identifying the purpose, we classify the primaries goals of the studies (e.g., identification, characterization, prediction). In complement, by identifying their focuses (e.g., technical debt and defect), we classify the main attributes of interest in the studies between the purpose and the object of analysis (comment analysis). Thus, for example, we could have: to identify (purpose) technical debt (focus) exploring comments analysis (object of analysis).

RQ2. What are the techniques used by researchers to analyze comments?

This question intends to identify what techniques of comments analysis have been used to extract, process, and analyze comments. We also intend to categorize the techniques as manual, semiautomatic and automatic.

RQ3. What are the tools used to extract, process, or analyze comments?

This question aims to identify what tools have been used to extract, preprocess, and analyze comments. The result of this RQ is a set of tools that can be used in comments mining process.

RQ4. Which empirical evaluations have been performed in the area?

This question aims to identify whether the proposed approaches have been evaluated through empirical methods, and if so, which method was used. To classify the types of studies, we considered the empirical evaluation types discussed by Farias *et al.* [4]. The taxonomy is available at https://goo.gl/qO6nUa.

RQ5. What are the identified research types?

This question intends to categorize the studies according to the research type facets defined by Wieringa *et al.* [11]: evaluation research, experience papers, opinion papers, philosophical papers, solution proposal, and validation research. By doing this, we intend to understand the overall contribution provided by the studies. In combination with the answers of RQ4, it also allows the identification of gaps and needs in the area.

B. Search strategy

In this study, we defined a generic search string:

((Software OR System OR Program OR Application) AND ((Comment analysis) OR (Comment Identification) OR (Code Comments) OR (Comment detection) OR (Examination of comments) OR (Analysis of comment) OR (Comments analysis) OR (Study of comments) OR (Comments study)))

We applied this search string to Titles and Abstracts. We chose not to do full text search because we found that it resulted in a very large number of studies out of scope.

C. Data Source

In choosing data sources, we aimed to include important journals and conferences regarding the research topic. For this, we considered the list recommended by Brereton *et al.* [2]: ACM Digital Library, IEEE Xplorer, Science Direct, Engineering Village, Springer Link, Scopus, and Citeseer.

D. Study Selection

After applying the search strings in the digital libraries, we filtered the relevant primary studies from the search results using the following selection criteria:

Inclusion criteria: (i) Published works that describe how comments analysis are used in software engineering activities; (ii) when several papers reported the same study, only the most recent was included; (iii) the publication date of the article should be between 1990 and 2016; and (iv) papers published in workshops, conferences or peer reviewed journals;

Exclusion criteria: (i) Studies out of the scope of this research; (ii) papers that are only available in the form of workshop/conference reports, abstracts or PowerPoint presentations; (iii) duplicated papers; and (iv) book chapters and articles published without revisions (white papers).

E. Screening of Papers

The screening of papers process to identify the primary studies comprises the following steps: (i) apply the selection criteria by reading the paper title and abstract and select the relevant studies from the search results; and (ii) read introduction and conclusion in case the researcher needs further information to decide on the study selection. A master and a Ph.D. student performed these steps and other two experienced researchers reviewed the results.

F. Data Extraction

Before the data extraction execution, we performed a pilot extraction, aiming to align the researchers' understanding of the research questions. During the extraction process, researchers carefully read the primary studies in a peer-reviewed process. Two researchers extracted data for the same study and a third researcher solved the possible disagreements. All relevant data of each study was registered in a spreadsheet. At the end, one experienced researcher reviewed the extracted data. The complete data are available at https://goo.gl/PGFVim.

G. Data Analysis and Synthesis

We considered a quantitative method to analyze the extracted data. Although we have done an analysis on the results, most of them were summarized to present an overview of the findings. Thus, this work is characterized as a scoping study, which maps the primary studies on mining code comments in the software engineering area.

4. Results and Discussion

We selected the papers following four steps. First, we searched papers in the digital libraries, resulting in 402 studies. Then, we removed the duplicated papers, resulting in 240. Next, we read the title and abstract to remove the studies out of the scope. This activity resulted in 68 studies, which were fully read. During this last step, we removed 32 studies, resulting in 36 papers to perform the data extraction. The complete list of selected studies is available at https://goo.gl/q7nekb.

We have 24 papers published on conferences, 9 on journals, and 3 on workshops. From a temporal point of view (Fig. 1), the studies are more concentrated from 2011. We also observed that 2015 was the year with more publications (6 – 16.7%), followed by 2016 (5 – 13.9%), 2011 and 2014, both with the same number of studies (4 – 11.43%). The increasing number of publications in the last two years (2015 and 2016) is partially justified by the presence of works relating comments analysis and Quality of Software Artifacts/Technical Debt. We can observe this trend in Fig. 2.

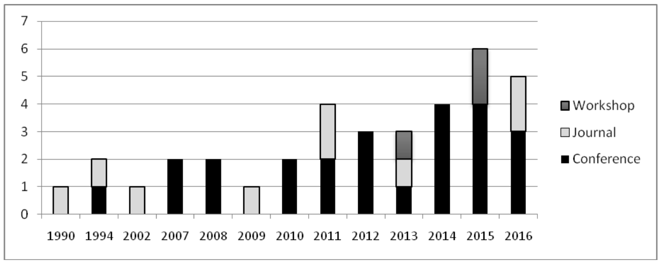

Figure 1. Temporal view of the selected studies

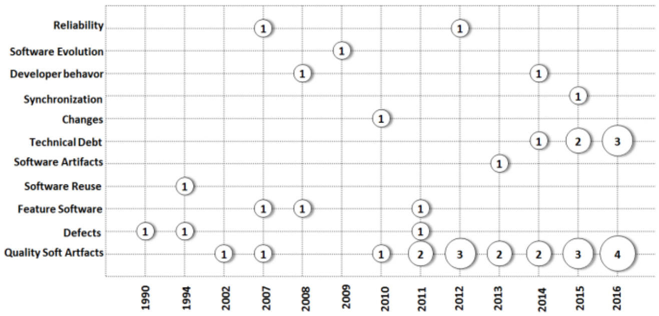

Figure 2. Studies' focus over the years

A. *Purposes and focus of researches (RQ1)*

To analyze RQ1 and classify the studies according their purpose and focus, we considered the taxonomy discussed in [4]. We also considered that each study could be categorized into more than one classification. For example, a study can explore comments to *identify* and *comprehend* (purpose) the quality of software artifacts (focus).

Purpose: "Comprehension" was considered in the majority of the studies (29 – 80.6%), followed by "identification" (10 – 27.8%). Studies related to the purpose comprehension intend to understand the behavior of a specific attribute by analyzing code comments (e.g., comprehension of the quality of software artifacts or comprehension of defect using code comment analysis). Whereas, studies with the purpose identification aim to detect a specific attribute through comment analysis (e.g., to identify defect). Only two studies had "improvement" as main purpose and all other categories ("classification", "evaluation", "localization", "association", and "characterization") were identified in only one study.

Focus: Fig. 2 shows the classification of the focus of the studies over the years. The majority of studies focused on "Quality of Software Artifacts" (19 – 52.8%) followed by "Technical Debt" (6 – 16.7%). "Quality of Software Artifacts" appeared nearly every year, but the number of studies with this focus has increased in the last years. The focus on "Technical Debt" only started to be considered more recently, in 2014, and it has also increased in the last years, revealing a new area of investigation.

Purpose x Focus: Fig. 3 presents a bubble chart showing the relationship between the facets purpose and focus. We can observe that "Comprehension" of "Quality of Software Artifacts" is the most explored purpose and focus in 16 out of 36 studies. The second most identified purpose and focus is "Identification" of "Quality of Software Artifacts", "Comprehension" of "Technical Debt", and "Identification" of "Technical Debt" with 4 studies each.

The results also pointed out that, while the purposes "Evaluation", "Localization", "Improvement", "Classification", "Association", and "Characterization" explore just one focus, the purpose "Comprehension" explores almost all focuses identified in this work.

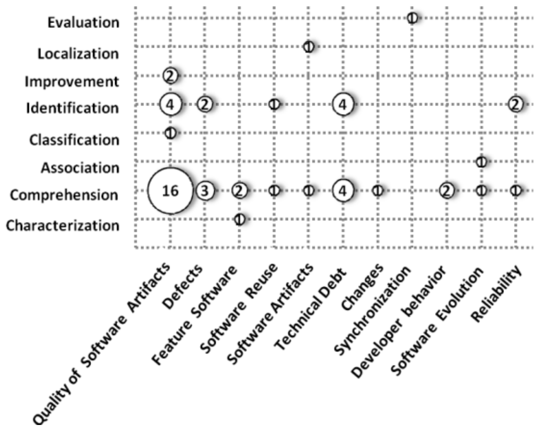

Figure 3. Purpose vs Focus

Regarding technical debt, we can observe that the studies on the area have been performed with the purposes of "Comprehension" or "Identification".

B. *Techniques to analyze comments (RQ2)*

We analyzed the techniques that have been used to mine comments and identified 14 techniques at total. 80.6% of the studies used: Dictionary/Vocabulary (13 studies – 36.1%), NLP (10 – 27.8%), and Statistic/Statistic Analysis/Method Statistic (6 – 16.7%).

Fig. 4 shows the relationship between the study purpose and the technique used to explore comments analysis. Comprehension and Dictionary/Vocabulary were the purpose and technique most used together. Next, NPL was used together with Comprehension and Identification. In the following, we have Identification and Dictionary/Vocabulary. We can also observe that Comprehension and Identification were combined with almost all techniques. On the other hand, some techniques have been used by only one study (e.g. Dynamic analysis and clustering).

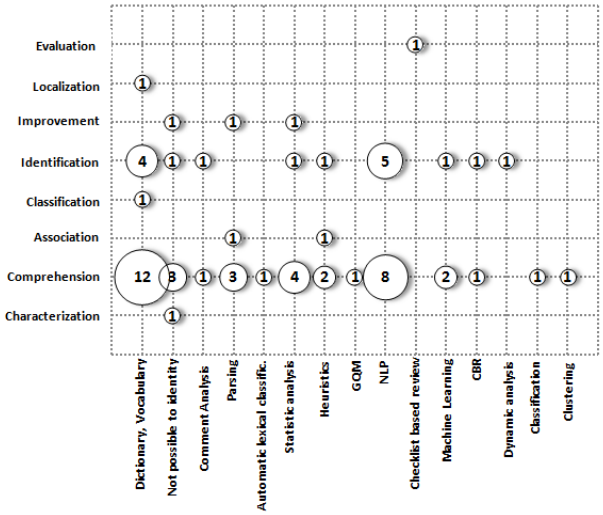

Figure 4. Studies' purpose vs Technique

By analyzing how each technique works, we also identified that 55.6% of them were semiautomatic, followed by automatic (27.8%), manual (8.3%), and the other 8.3% was not determined. A possible reason for the low usage of manual techniques can be the cost to perform a manual analysis in terms of effort and also the fact that the process would be error prone.

C. Tools used to extract, process, or analyze comments (RQ3)

Table 1 presents the tools identified in this work by each step of a mining process (extraction, processing, and analysis). We identified 16 tools used in the extraction step, 14 for the analysis, and 2 for the processing. We did not identify any tool in 50% of the selected studies.

By analyzing Table 1, we can see that most of the tools were presented in recent studies. Another point is that only one tool is used in more than one step (iComment). The others were developed to support only one step of the comment mining process. We also identified that most of the tools are only presented in one study. Therefore, in general, researchers develop new tools as a result of their work. A possible explanation for this is that each study has a specific need (not considered in existing tools) of exploring comments in order to achieve its goals.

This set of information about tools might be useful for researchers and practitioners to develop new approaches or evolve the existing tools with the aim of exploring new perspectives on comment analysis.

D. Empirical evaluations (RQ4)

In this mapping study, we found that 16 papers (44.4%) performed controlled experiments, 8 (22.2%) case studies, 4 (11,1%) exploratory studies, and only 1 paper (2.8%) performed a survey. We also identified 1 ethnographic study (2.8%). The other 6 (16.7%) studies did not present any evaluation. This result indicates that the works in the area of software comments analysis are characterized by the use of empirical methods to assess the proposed approaches.

Fig. 5 represents the evaluation methods performed per purpose. We can observe that controlled experiment and case study are the main research methods used for evaluating “comprehension” and “identification” tasks. “Classification”, “evaluation” and “characterization” were the three categories in which researchers have not used empirical methods to evaluate their approaches.

TABLE 1. LIST OF TOOLS AND MINING STEP

Process Step | Tool	References
Extraction (16) | CLOC, tComments, @Randoop, Prototype Tool, RBG tool, SLOCCount, Jdeodorant, srcML, SSLdoclet, ConQAT, C-REX, Evolizer and ChangeDistiller, eXcomment, iComments, JavaMethodExtractor	S2 (2016), S36 (2012), S14 (2010), S5 (2015), S13 (2015), S1 (2014), S9 (2016), S28 (2013), S3 (2015), S7 (2011)
Processing (2) | iComments, LI Tools	S33(2007), S3(2015
Analysis (14) | tComments,@Randoop, QDA Analysis tool, RBG tool, CommentCounter, LOCCounter, COMTOR, Evolizer and ChangeDistiller, iComments, Javadocminer, MineHEAD , Stanford Parser, next word prediction tool	S36 (2012), S4 (1994), S5 (2015), S32 (2014), S8 (2012), S10 (2009), S33 (2007), S34 (2011), S17 (2015)

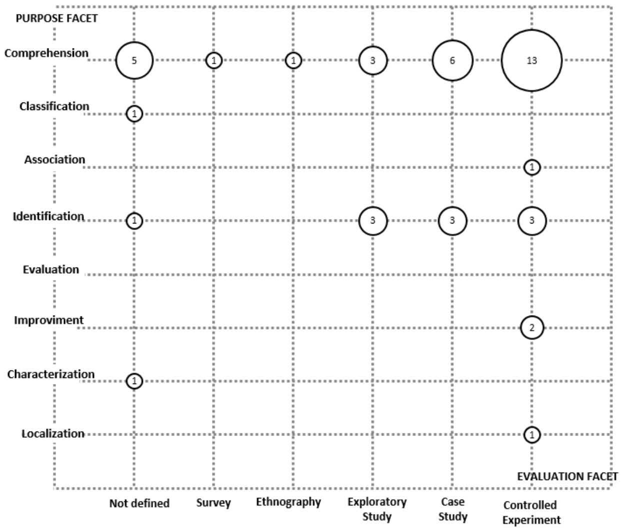

Figure 5. Purpose vs Evaluation methods

E. Research types (RQ5)

Considering the taxonomy of research types presented in [11], we found that the majority of studies were a “Solution Proposal” paper (19 – 52.8%), 12 were a “Evaluation Research” (33.3%), and 3 were a combination of “Solution Proposal” with “Evaluation Research” (8.3%). Only 1 study performed a “Validation Research” (2.8%) and 1 study used “Opinion Research” (2.8%). Solution proposal is paper where a solution for a problem is proposed. The potential benefits and the applicability of the solution are shown by a small example or a good line of argumentation. On the other side, evaluation research is a type of paper in which techniques are implemented in practice and an evaluation of the technique is conducted.

Fig. 6 presents the distribution of the performed research types over the years. We can observe that while the number of “Solution Proposals” published during the years is stable, there is a rising number of “Evaluation Research” in the last few years (2014, 2015 and 2016) indicating a tendency in the area to perform and report more empirical studies.

Fig. 7 presents the relationship between research type and focus. It shows that the types “Solution Proposal” and “Evaluation Research” were widely adopted by researchers to investigate "Quality of Software Artifacts". We can also see that the focus "Technical Debt" is the second most explored considering these same types. The other types of research appear only as isolated initiatives.

5. THREATS TO VALIDITY

The results of this systematic mapping may have been affected by some threats to validity, such as:

Search string: Even though our search string is broad, it is possible that it did not address some studies. Our search string was designed to find the maximum number of works

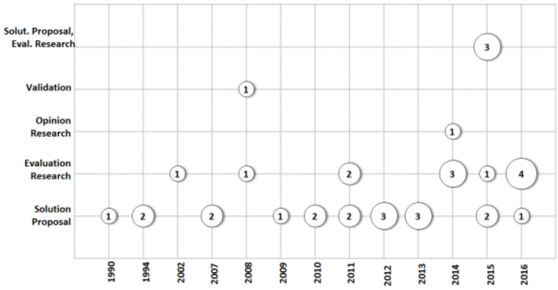

Figure 6. Evolution of the research type over the years

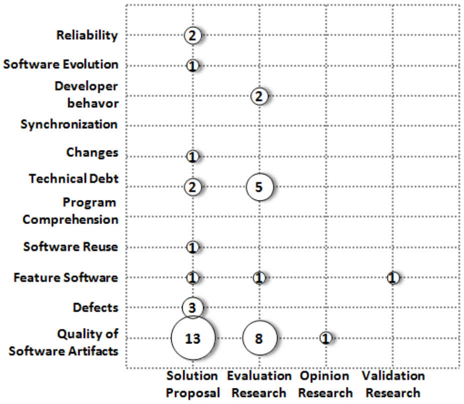

Figure 7. Focus vs Research type

on code comments, but it is possible that it missed studies that did not used the term "comment" in their text. We tried to mitigate this threat by a process of string calibration.

Selection Bias: We cannot ensure that all relevant primary studies were selected for this mapping. We addressed this threat during the selection step. We selected each study based on the judgment of the inclusion and exclusion criteria by more than one researcher. However, some studies could still have been categorized incorrectly. To mitigate this, we discussed the study protocol among the researchers to guarantee a common understanding. Moreover, this step was performed by two researchers and, when both disagreed, we considered a third opinion.

Publication Bias: It is difficult to ensure that all relevant work was returned as results in the performed searches. To minimize this threat, the main digital libraries in computing were considered.

Research Questions: The research questions investigated in this study may not cover all software comments analysis area. To address this risk, the defined questions were analyzed by at least two researchers, one of who acted as an external reviewer of the protocol.

Data Extraction: this threat can affect the analysis of selected studies. To reduce this risk, initially, we performed a pilot extraction, aiming to align the researchers' understanding of the research questions. Next, two researchers analyzed each paper to perform the data extraction. A third researcher analyzed the issues on each classification or extracted information to make sure that the extracted data were valid and clear for further analysis.

6. CONCLUDING REMARKS

In this paper, we performed a systematic mapping study on software comments analysis. We have extracted and analyzed data from 36 papers. The results can guide researchers in further studies in MSR area focused on code comments. For practitioners, we catalogued a set of tools and techniques to analyze comments with several purposes. This information can help them avoiding reinventing the wheel when developing approaches to extract, processing or analyzing comments.

In our future research agenda, we intend to combine the evidence identified in this work with new theories and empirical studies developed by our group to create new methods and tools to support comments analysis with focus on technical debt identification and management activities.

ACKNOWLEDGMENT

This work was partially supported by the CNPq Universal grant 458261/2014-9, by the State of Bahia's SECTI-Fraunhofer-UFBa cooperation agreement 2012-1, and by the RESCUER project Grant: 490084/2013- 3.

REFERENCES

[1] N.S.R. Alves, T.S. Mendes, M.G. Mendonça, R.O. Spínola, F. Shull, and C. Seaman, Identification and management of technical debt: A systematic mapping study. Information and Software Technology v. 70, p.100-121. 2016.

[2] P. Brereton, B.A. Kitchenham, D. Budgen, M. Turner, and M. Khalil, Lessons from applying the systematic literature review process within the software engineering domain", J. Syst. Softw. 80, 2007.

[3] S. Demeyer, A. Murgia, K. Wyckmans, and A. Lamkanfi, Happy birthday! A trend analysis on past MSR papers, IEEE Int. Work. Conf. Min. Softw. Repos., p. 353–362, 2013.

[4] M.A.F. Farias, R. Novais, M. Colaço, L.P.S Carvalho, M. Mendonça, and R.O. Spínola, A systematic mapping study on mining software repositories," in 31st ACM/SIGAPP, 2016.

[5] M.A F. Farias, A.B. Silva, M.G. Mendonça, Spínola, R.O., and M. Kalinowski, "Investigating the use of a contextualized vocabulary in the identification of technical debt : A controlled experiment," in 18Th Int. Conf. on Enterprise Information System, vol. 1, pp. 369–378, 2016.

[6] M.A.F Farias, M.G. Mendonça, A.B.D. Silva, and R.O. Spinola, A contextualized vocabulary model for identifying technical debt on code comments," in IEEE 7th Int. Work. on Managing Technical Debt, MTD 2015.

[7] B. Kitchenham, Guidelines for performing systematic literature reviews in software engineering. EBSE Technical Report, Software Engineering Group, School of Computer Science and Mathematics, vol. 3 (2). Keele University, Keele, UK. 2007.

[8] E. Hassan, "The road ahead for mining software repositories," Front. Softw. Maintenance. FoSM 2008., pp. 48–57, 2008.

[9] H. Hemmati, S Nadi, O. Baysal, O. Kononenko, W. Wang, R. Holmes, M. W. Godfrey, "The MSR cookbook: Mining a decade of research," 10th Conference Mining Software. Repository., pp. 343–352, May 2013.

[10] H. Kagdi, M.L. Collard, and J.I. Maletic, "A survey and taxonomy of approaches for mining software repositories in the context of software evolution," Journal of Software Maintenance and Evolution: research and practice pp. 77–131, 2007.

[11] R. Wieringa, N. Maiden, N. Mead, and C. Rolland, Requirements engineering paper classification and evaluation criteria: a proposal and a discussion. Requirements Engineering, pp. 102–107, 2005.