Research Protocol

Di Wang
Lentune Ltd.
Christchurch, New Zealand
di.wang@outlook.co.nz

Matthias Galster
University of Canterbury
Christchurch, New Zealand
matthias.galster@canterbury.ac.nz

I. Research Method

To achieve the research goal outlined in Section **??** ("Present an overview of previous research on bug reproduction and localization"), we conducted a systematic mapping study. Systematic mapping studies identify the gap in previous research and areas which require further attention based on frequently occurring themes in current research using a systematic and objective procedure [Budgen et al.(2008)Budgen, Turner, Brereton and Kitchenham]. Furthermore, the outcome of the mapping study will help plan new research to avoid duplicating previous work [Mota Silveira Neto et al.()Mota Silveira Neto, Carmo MacHado, McGregor, Almeida and Lemos Meira]. Given our study goal which is rather broad, a systematic mapping study is more suitable than other methods, such as a systematic literature review, which require more specific questions and aim at synthesizing existing evidence regarding a phenomenon.

To conduct this study, we adopted the review process suggested by Petersen et al. [Petersen et al.()Petersen, Vakkalanka and Kuzniarz] and followed three generic phases: definition of research directives, data collection, and presentation of results:

- In the research directives stage, we defined our research questions (see Section **??**) and the research protocol. We developed our research protocol based on the template used in [Barros-Justo et al.(2016)Barros-Justo, Pinciroli, Matalonga, Gonzalez and Araujo] and considering the ACM SIGSOFT Empirical Standards for systematic literature studies [Ralph(2021)].
- During data collection we identified studies and extracted relevant data from studies. This required us to define search strings to search electronic data sources for studies (Section I-A). We also selected data sources (Section I-B), defined selection criteria (Section I-C) and planned the search process (Section I-E).
- To present the results, we created a scheme to classify existing papers using the instructions from Petersen et al. [Petersen et al.(2008)Petersen, Feldt, Mujtaba and Mattsson] and performed the actual mapping of current literature (Section I-D).

Below we outline the details of our study design.

A. Search String

After an initial search and based on the research questions, we identified a search string with three main parts:

- The domain in which we operate, i.e., software;
- The "object" in the domain we are interested in, i.e., bugs (including related concepts like faults and errors, see our definition and scope of the concept of "bug" in Section **??**);
- The "treatment" of that object in the domain, e.g., debugging, fixing, reproduction and localization.

The three main parts are connected by a logical `AND`, while the elements within each part are connected with an `OR`. For each part of the search string we included synonyms of related concepts. This resulted in the generic search string: `software AND (bug OR fault OR error OR failure OR defect) AND (debug OR debugging OR localization OR localisation OR locating OR locate OR localize OR localise OR detect OR detection OR reproduce OR reproduced OR reproduction OR reproducing OR fix OR fixing OR remove OR removal).`

B. Data Sources

We identified the following four digital databases from [Diebold and Dahlem(2014)], [Kuhrmann et al.(2017)Kuhrmann, Fernández and Daneva] as our data source for the research: IEEE, ACM, SpringerLink and ScienceDirect. For all databases we limited the search to publications between 2011 and 2021 to obtain more recent studies conducted in the last ten years. The search was conducted in June 2021. During the search we also utilised several search engines' filtering functions:

- For IEEE we searched all metadata and used "program debugging" and "fault diagnosis" in the publication topic filter.
- For ACM we searched titles and abstracts and publications in English only.
- For SpringerLink we set the language of publications to English and limited the search to disciplines Computer Science, Software Engineering, Software Engineering/Programming and Operating Systems.
- For ScienceDirect we searched title, abstract or author-specified keywords. We limited the subject areas to Computer science. Furthermore, due to the limitation of the number of logical operators supported on ScienceDirect, we split the search string into two, and, since ScienceDirect supports "containment"

search, altered the search string: (1) `software AND (bug OR fault OR error OR failure OR defect) AND (reproduc OR fix OR remov)`, and (2) `software AND (bug OR fault OR error OR failure OR defect) AND (debug OR loca OR detect)`.

1) IEEE: 3,718
   - Search area: All metadata
   - Publication Topic filter: program debugging, fault diagnosis
2) ACM: 165
   - Search area: Title and abstract
3) SpringerLink: 1,967
   - Discipline: Computer Science, Software Engineering, Software Engineering/Programming and Operating Systems
   - Language: English
4) Science Direct: 566
   - Search area: Title, abstract or author-specified keywords
   - Subject area: Computer science
   - Due to the limitation of the number of logic operators for this search engine we have to break the search string into two. We also altered the search string since the search engine can do the search based on the "containment" of the string rather than the whole and exact word.
     - software AND (bug OR fault OR error OR failure OR defect) AND (reproduc OR fix OR remov)
     - software AND (bug OR fault OR error OR failure OR defect) AND (debug OR loca OR detect)

C. Inclusion and Exclusion Criteria

We included a paper if it met at least one of the inclusion criteria and excluded a paper if it met at least one of the exclusions criteria.

- The inclusion criteria are the following:
  - Study investigates software debugging or bug fixing approaches related to bug localization and reproduction.
  - Study investigates system diagnostic techniques that can be used for bug localization and reproduction (e.g., code analysis, system tracing and monitoring).
- The exclusion criteria are the following:
  - Study before 2011: As software development is a fast-changing domain and this research targets the most recent research, studies that are more than ten years old are not considered.
  - Study investigates software usage problems (e.g., users misusing software due to poor usability) rather than software bugs. We focus on the diagnostics of the software itself rather than user behaviour.
  - Study investigates system diagnostics for hardware (e.g., malfunctioning or failing hardware or poorly configured hardware; latency and throughput issues due to outdated hardware).
  - Study investigates bug fixing or debugging approaches but does not discuss bug localization or bug reproduction.
  - Study investigates software quality issues without a clear link to bug localization or reproduction e.g., security vulnerabilities.
  - Study discusses locating the bug in the overall system but not in the software application itself (e.g., a study that identifies the faulty application, rather than the actual bug in a set of multiple applications).
  - Short papers, work-in-progress papers and keynote summaries (we focus on full research research papers with sufficient details).

D. Data Collection and Analysis

We designed a data collection form (shown in Table I). We analyzed data using thematic synthesis and discussions between the researchers [Cruzes and Dyba(2011)]. Following the mapping study guide line [Petersen et al.()Petersen, Vakkalanka and Kuzniarz], we collected the data discussed in the following. To answer **RQ1**, we collected the following:

Following the mapping study guideline [**?**], we collect two types of data, free text and keywording data (referred to as "tags").

- We used free text to extract key pieces of information from the papers about our research questions.
- Keywording data is collected as tags and used to categorize information about the papers. There are two kinds of keyword tags: pre-defined and emerging:
  - Pre-defined keyword tags are keyword tags that are predefined from our initial study or the literature and which do not change during the data analysis (e.g., we use existing categories for research methods).
  - Emerging keyword tags evolved consolidated during data analysis. An example of an emerging keyword tag is "Researched Approach". This tag has some pre-defined options (e.g., "IRBasedBugLocalisation"), but during the analysis more approaches were identified. For example, we identified tags "TextBasedBugLocalisation", but later found that all "TextBasedBugLocalisation" papers are actually "IRBasedBugLocalistion".
- **Type of problem** describes the high-level problem previous research tried to address. We identified three tags based on our initial research "bug localization", "bug reproduction" and "bug fixing", i.e., a paper can address one of these types of problems.
- **Technique/approach** describes what approaches have been studied to address a problem. In contrast to the type of problem, this data item is more about the solution. We identified initial tags such as "IRBasedBugLocalization" and "AutoBugFixing" based on our initial research. New tags were then added when new types of approaches were identified during the analysis.

TABLE I
DATA COLLECTION FORM

ID	Item	Data type and pre-defined tags	RQ
D1	Paper title	Free text	Demographics
D2	Author(s)	Free text	Demographics
D3	Year of publication	Year	Demographics
D4	Type of problem	Tag:#BugLocalization; #BugReproduction; etc.	1
D5	Technique/approach	Tag:#IRBasedBugLocalization;#AutoBugFixing; etc.	1
D6	Data source	Tag:#BugReport;#SourceCode; etc.	1
D7	Research method(s)	Tag:#Survey;#Experiment;#CaseStudy;#Interview; #Fieldstudy;#Observation;#LiteratureReview	2
D8	Research context	Tag:#OpenSourceRepo; #CommercialProject; etc.	2
D9	Target of study	Tag:#Software;#Frameworks; etc.	2
D10	Programming Language	Tag:#Java; C#; etc.	2
D11	Types of contribution	Tag:#AdoptedApproach;#Tool; etc.	3
D12	Main challenges	Free text	4
D13	Future works	Free text	4

- **Data source** gives information about what data sources previous studies used. We identified initial tags such as "BugReport" and "SourceCode" based on our initial research. New tags were then added when new data sources were found in the analyzed papers.

To answer **RQ2** we collected the following:

- For the **Research method(s)**, we used pre-defined tags based on the literature ([Shull et al.()Shull, Singer and Sjøberg]) listed in Table I. These help us understand how bug reproduction and localization research has been done and can guide future studies in the area (e.g., if formal experiments dominate to assess theoretical soundness, case studies with developers may help investigate practical issues) [Stol and Fitzgerald(2018)]. Note that when recording the research method we relied on the self-reporting of authors. We did not assess the rigor of the research method, e.g., if a case study is really a case study in the empirical sense or an illustrative example.
- The **research context** describes the characteristics of the context in which previous research has been conducted. For example, from our initial research, we noticed that many studies are done with the open-source projects. We used tags defined from our initial research as well as tags that emerged during analysis.
- The **target of study** aims to understand what target objects have been studied for bug reproduction and localization. For example, some studies have been conducted based on a software project such as Firefox while some studies looked into development frameworks such as Microsoft Windows form. The tags emerged from our initial research and later analyses.
- The **programming language studied** are about the programming languages addressed in bug reproduction and localization studies (note that we did not investigate the technologies used to implement bug reproduction and localization approaches). We used tags, some of which come from our initial research and some that emerged during data analysis.

To answer **RQ3** we collected one type of data:

- The **types of contribution** describe what a paper contributes. For example, from our initial research we identified some types of contributions, e.g., adopted approaches (i.e., a new approach that builds on an existing approach) and tools. Other tags emerged during analysis.

Finally, to answer **RQ4** we collect another two types of data:

- The **main challenges** is collected as free text and aims to find out the main challenges for conducting bug localization and reproduction that were discovered previously. In contrast to data item "research problem", this data item is about the open challenges reported in papers, not the addressed research problem. The outcome should help us better plan future studies and identify gaps in the current body of work. Based on the data collected we grouped similar types of challenges.
- **Future work** is collected as free text by reviewing paper sections (e.g., conclusions, discussion, or dedicated future work sections) to identify gaps acknowledged by previous research. The data collected is evaluated and then categorised by grouping similar types of future work.

E. Search Process

We conducted our research by following the search strategy proposed by Petersen et al. [Petersen et al.()Petersen, Vakkalanka and Kuzniarz]. The overall process is shown in Figure 1:

- **Step 1:** Automated search. We ran the search string mentioned above in all four digital databases and exported the results from each search. This resulted in 6,416 results, (3,718 from IEEE, 165 from ACM, 1,967 from SpringerLink and 566 from ScienceDirect).
- **Step 2:** Exporting data to customized data storage. We developed a database to hold the data exported from each digital library. We used a customized script to import all data into the database.
- **Step 3:** Removing duplicates. We removed duplicates based on the title and abstract of papers. This resulted in another 231 papers being removed. We also removed two duplicates based on the author names and years of

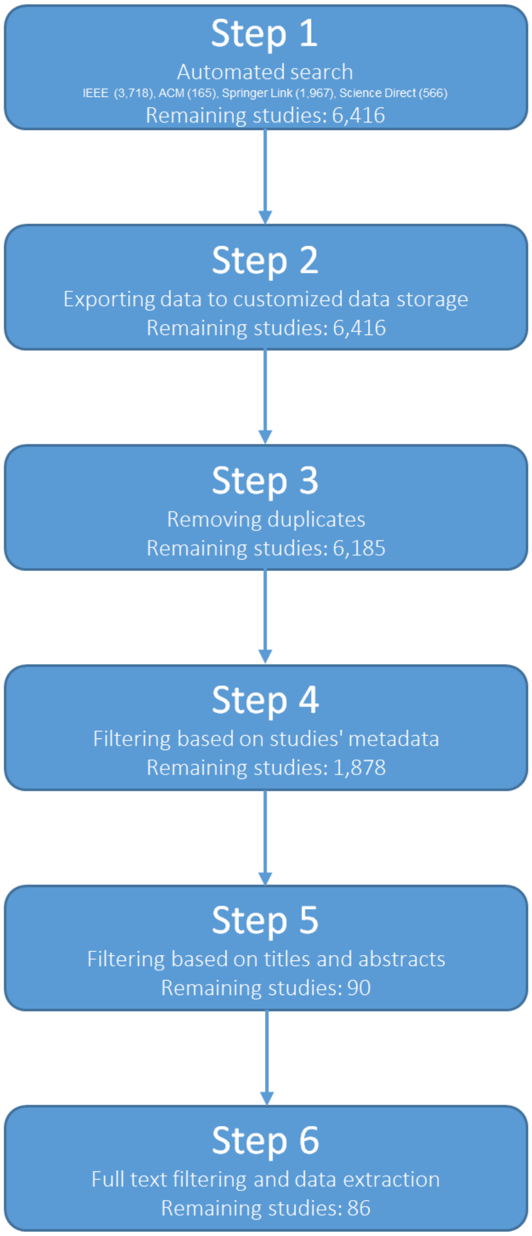

Fig. 1. Search and data extraction process

publication. These duplicates were not caught before, because they are studies with slightly different titles but the same content. This step resulted in a total of 6,185 papers.

- **Step 4:** Filtering based on studies' metadata. In this step, we filtered papers based on the inclusion and exclusion criteria applied to the metadata. We excluded all the studies published in journals not relevant for this study (e.g., Computers and Electronics in Agriculture). The number of papers went down to 4,375. We then removed papers with fewer than five pages in IEEE and ACM, and papers with fewer than nine pages in SpringerLink and ScienceDirect as we only consider full papers (see exclusion criteria above) as full papers provide more details for the study and gennerally with better quality. This resulted in 3,161 papers. Then, we re-run the search string against all paper titles and abstracts in our customized database as some digital library's search engines only search metadata and full text. This resulted in 1,878 papers.
- **Step 5:** Filtering based on titles and abstracts. In this step we first read through all 1,878 papers' titles and removed papers based on the exclusion criteria mentioned above, resulting in 172 papers. Then we read through all papers' abstracts and reduced the number of papers to 90.
- **Step 6:** Full text filtering and data extraction. In this step, we read the full text of each paper and used a custom-made web application, which fits into the need of our study better than existing tools, to record the extracted data for each paper (according to Table I). We used a tagging system to tag the key information. For example, if the research goal is about understanding bug localisation, then tags such as "#BugLocalization" were assigned. Later, we conducted an analysis based on the frequency of the tags. We also recorded additional information in a notes field of our customized tool. This information was used to do some qualitative evaluation. When reading the full texts of papers, another four papers were removed based on the exclusion criteria. This resulted in a final set of 86 papers.

REFERENCES

[Barros-Justo et al.(2016)Barros-Justo, Pinciroli, Matalonga, Gonzalez and Araujo] Barros-Justo, J.L., Pinciroli, F., Matalonga, S., Gonzalez, M.A.P., Araujo, N.M., 2016. Systematic mapping protocol: Have systematic reuse benefits been transferred to real-world settings? arXiv:1609.06553.

[Budgen et al.(2008)Budgen, Turner, Brereton and Kitchenham] Budgen, D., Turner, M., Brereton, P., Kitchenham, B.A., 2008. Using mapping studies in software engineering. 8, 195--204.

[Cruzes and Dyba(2011)] Cruzes, D.S., Dyba, T., 2011. Recommended steps for thematic synthesis in software engineering, in: 2011 International Symposium on Empirical Software Engineering and Measurement, pp. 275--284. doi:10.1109/ESEM.2011.36.

[Diebold and Dahlem(2014)] Diebold, P., Dahlem, M., 2014. Agile practices in practice: A mapping study, in: Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering, Association for Computing Machinery, New York, NY, USA. URL: https://doi.org/10.1145/2601248.2601254, doi:10.1145/2601248.2601254.

[Kuhrmann et al.(2017)Kuhrmann, Fernández and Daneva] Kuhrmann, M., Fernández, D.M., Daneva, M., 2017. On the pragmatic design of literature studies in software engineering: an experience-based guideline. Empirical Software Engineering 22, 2852--2891. URL: https://doi.org/10.1007/s10664-016-9492-y, doi:10.1007/s10664-016-9492-y.

[Mota Silveira Neto et al.()Mota Silveira Neto, Carmo MacHado, Mota Silveira Neto, P., Carmo MacHado, I., McGregor, J., Almeida, E., Lemos Meira, S., . A systematic mapping study of software product lines testing. Information

and Software Technology 53, 407{423,. doi:10.1016/j.infsof.2010.12.003.
[Petersen et al.(2008)Petersen, Feldt, Mujtaba and Mattsson] Petersen, K., Feldt, R., Mujtaba, S., Mattsson, M., 2008. Systematic mapping studies in software engineering, in: Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering, p. 68{77.
[Petersen et al.()Petersen, Vakkalanka and Kuzniarz] Petersen, K., Vakkalanka, S., Kuzniarz, L., . Guidelines for conducting systematic mapping studies in software engineering: An update. Information and Software Technology 64, 1{18,. URL: doi:, doi:10.1016/j.infsof.2015.03.007.
[Ralph(2021)] Ralph, P., 2021. Acm sigsoft empirical standards released. SIGSOFT Softw. Eng. Notes 46, 19. URL: https://doi.org/10.1145/3437479.3437483, doi:10.1145/3437479.3437483.
[Shull et al.()Shull, Singer and Sjøberg] Shull, F., Singer, J., Sjøberg, D., . Guide to advanced empirical software engineering. Guide to Advanced Empirical Software Engineering , 1{388,doi:10.1007/978-1-84800-044-5.
[Stol and Fitzgerald(2018)] Stol, K.J., Fitzgerald, B., 2018. The ABC of Software Engineering Research. ACM Transactions on Software Engineering and Methodology 27, 11--51. URL: https://doi.org/10.1145/3241743, doi:10.1145/3241743.