Software Product Line Regression Testing: A Research Roadmap

Willian D. F. Mendonça[1], Wesley K. G. Assunção[2] and Silvia R. Vergilio[1]
[1]*DInf, Federal University of Paraná, Curitiba, Brazil*
[2]*DI, Pontifical Catholic University of Rio de Janeiro, Rio de Janeiro, Brazil*

Keywords: Software Testing, Software Reuse, Software Evolution, Research Opportunities, Systematic Mapping.

Abstract: Similarly to traditional single-product software, Software Product Lines (SPLs) are constantly maintained and evolved. However, an unrevealed bug in an SPL can be propagated to a wide set of products and impact customers differently, depending on the set of features they are using. In such scenarios, SPL regression testing is paramount to avoid undesired problems and guarantee that the SPL maintenance and evolution are performed accordingly. Although there are several studies on SPL regression testing, the research community lacks a clear set of research opportunities to be addressed in a short and medium term. To fulfill this gap, the goal of this work is to overview the current body of knowledge of SPL regression testing and present a research roadmap for the following years. For this, we conducted a systematic mapping study that found 27 primary studies. We identified techniques used by the approaches, and applied strategies. Test case selection and prioritization techniques are prevalent, as well as fault and coverage based criteria. Furthermore, based on gaps and limitations reported in the studies we distilled a set of future work opportunities that serve as a guide for new research in the field.

1 INTRODUCTION

Software Product Line Engineering is a reuse-oriented approach to systematically develop families of software systems. A Software Product Line (SPL) allows cost-efficiently derivation of tailored products to specific markets, utilizing common and variable assets in a planned manner (Linden et al., 2007). We have seen several pieces of work describing adoption of SPLs in industry in the last years (Grüner et al., 2020; Abbas et al., 2020).

Similarly to traditional single-product software development, SPLs are constantly maintained and evolved (Marques et al., 2019). However, an unrevealed bug in an SPL can be propagated to a wide set of products and impact customers differently, depending on the set of features in the products they use. In such a scenario, SPL testing is paramount to avoid undesired problems (do Carmo Machado et al., 2014; Engström and Runeson, 2011). More specifically, regression testing has the role of guaranteeing that maintenance and evolution of SPLs are performed accordingly (Runeson and Engström, 2012; Engström, 2010b).

Although there are several recent primary studies on SPL regression testing, the research community lacks a clear set of research opportunities to be addressed in a short and medium term. The last secondary pieces of work on this topic were published in 2010 (Engström, 2010b; Engström, 2010a). Therefore, there is a need for an updated and comprehensive study to fulfill this gap (bin Ali et al., 2019; Marques et al., 2019). Based on this, the goal of this work is to overview the current scenario and existing body of knowledge of SPL regression testing to present a research roadmap for the following years. For this, we conducted a systematic mapping study to collect existing studies on SPL regression testing (Petersen et al., 2015). Guided by four research questions that aim to identify existing SPL regression testing approaches and their main characteristics, we identified 27 primary studies published in the period of 2005 to 2020. More than 85% of them were published after 2010, therefore, not discussed in the last literature review on the topic (Engström, 2010a).

The approaches proposed in the collected studies are analyzed considering regression testing technique supported, input and output artifacts used, strategy to apply the technique, and testing criteria adopted. As a result, and main contribution of our work, we distilled a roadmap with research opportunities for future work. This roadmap spans through the whole

Mendonça, W., Assunção, W. and Vergilio, S.
Software Product Line Regression Testing: A Research Roadmap.
DOI: 10.5220/0010959700003179
In *Proceedings of the 24th International Conference on Enterprise Information Systems (ICEIS 2022) - Volume 2*, pages 81-89
ISBN: 978-989-758-569-2; ISSN: 2184-4992

process of regression testing. It serves as a guide to motivate new research in the field and make existing approaches adopted in practice.

2 RELATED WORK

In the literature, we can find pieces of work that report secondary studies for general regression testing (Yoo and Harman, 2012; Minhas et al., 2017). Also, studies that carry out mapping and surveys for SPL testing in general (Lee et al., 2012; do Carmo Machado et al., 2014), and other ones, more related to ours, addressing specifically the topic of SPL regression testing (Engström, 2010b; Engström, 2010a). However, these pieces of work were published in 2010. As a consequence, they do not encompass more than 10 years of research and practice in the field. The analysis of more recent studies, published in the last decade, allows us to derive new opportunities, and to discuss new trends not presented in related work.

Besides the studies on testing and regression testing, some papers published in the last two years have described studies on diverse topics of SPLs. For example, a mapping study on SPL evolution (Marques et al., 2019) reinforces the need for regression testing. We can also mention secondary studies on SPL and variability management in emerging technologies, such as IoT (Geraldi et al., 2020), microservices (Mendonça et al., 2020; Assunção et al., 2020), and composition of ML/AI products with SPLs (Nomme, 2020). Based on that, we can argue the need of focusing on regression testing.

3 STUDY DESIGN AND EXECUTION

The goal of our study is *to overview the current scenario and body of knowledge of SPL regression testing to present a research roadmap for the following years*. Considering this goal, our study was guided by the following Research Question (RQ): "***Which are the existing SPL regression testing approaches and what are their characteristics?***". This question aims to characterize the regression testing approaches that are specific for SPLs. We identify and discuss applied techniques, strategies, input and output artifacts and testing criteria. From this general question, we derived four sub-RQs, as follows: **RQ1.** What are the addressed SPL regression testing techniques? **RQ2.** What kind of strategies are used to apply SPL regression testing techniques? **RQ3.** What are the input and output artifacts used? **RQ4.** Which are the testing criteria adopted?

Next, we present in detail the methodology adopted to conduct our systematic mapping study in order to achieve our goal and answer the posed RQs.

3.1 Primary Sources Selection

For the selection of primary sources, we followed a methodology based on the systematic mapping method, according to the process proposed by Petersen et al. (Petersen et al., 2015). From the goal and RQs of our study, we derived two main keywords, namely "*regression testing*" and "*software product line*". To define the search string[1], we composed these keywords with their lexical and syntactic alternatives (synonym, plural, gerund, etc.).

The string was used for searching studies in five digital libraries, as presented in Table 1. We did not define an initial publication date for the studies, then all the returned papers were considered. Table 1 presents the data sources used and the period covered. As we can see, a total of 2508 studies were found, including the period from 1985 to 2021. The search on these libraries was completed on Jan. 26th, 2021.

Table 1: Number of studies retrieved by each digital library.

Digital Library	Studies	Covered Period
ACM	300	1985-2021
IEEEXplore	24	2013-2021
ScienceDirect	307	1996-2021
Scopus	1106	1996-2021
Springer Link	771	1993-2021
Total	2508	1985-2021

For screening the studies retrieved from the digital libraries, we followed five steps, which are presented in Figure 1. In the first step we applied a filter to keep only studies on the area of computer science, remaining 1428 studies. For managing this set of studies, we used Parsifal[2]. This tool automatically identified 143 duplicated studies (second step). In the third step we read the title, abstract, and keywords of 1272 studies and removed 1064 of them that were out of the scope of our work. The remaining 208 papers went to a full reading, in which we considered one inclusion crite-

[1]The final string was: *("regression testing" **OR** "regression test") **AND** ("product line" **OR** "SPL" **OR** "product-family" **OR** "product family" **OR**"highly-configurable" **OR** "highly configurable" **OR** "feature model" **OR** "feature-model" **OR** "FM" **OR** "variability analysis")*

[2]A web-based tool for planning, conducting and reporting the systematic reviews: https://parsif.al/

ria (IC) and four exclusion criteria (EC), presented in Table 2. Finally, we composed a set of 27 primary sources (see Table 3).

The process was conducted by the two first authors of this paper and validated by the third one. From the 27 primary studies, we extracted pieces of information that are related to five dimensions in order to answer our RQs, shown in Table 3. After this, for each dimension we interactively identified relevant categories used to classify, discuss, analyze the studies, and we provide the complete extraction in a spreadsheet[3].

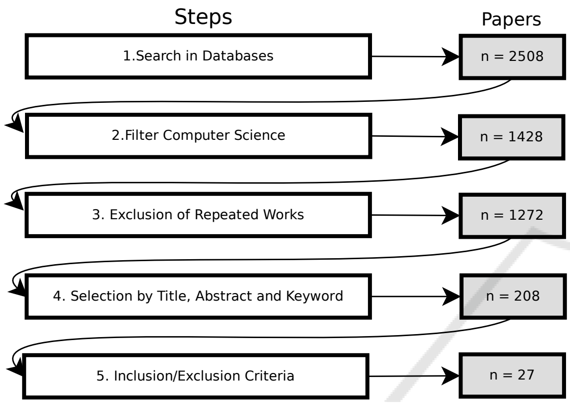


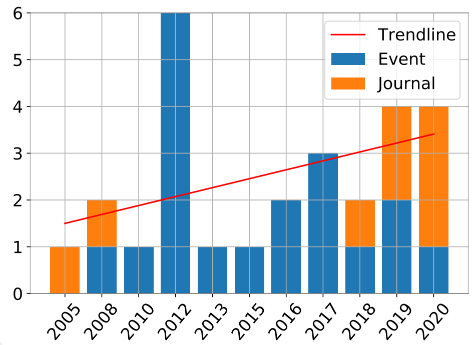
Figure 1: SPL regression testing process.

Table 2: Inclusion and exclusion criteria.

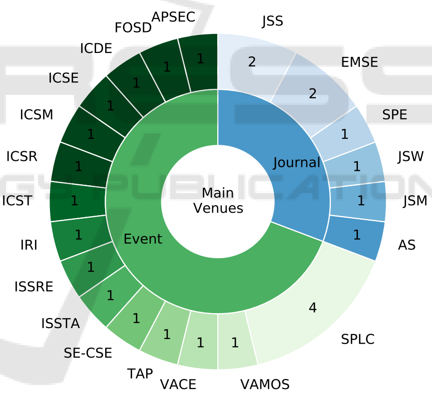
Inclusion Criteria	
IC1	Clear reference to regression testing. The paper refers to one of the regression testing techniques, making clear how is its adoption/use in the regression testing activity.
Exclusion Criteria	
EC1	Out of scope. The paper does not satisfy IC1. It is not clear or presented how the technique is applied in the SPL regression testing.;
EC2	Not available online;
EC3	Not in English;
EC4	Abstracts, posters, reviews, conference reviews, chapters, thesis, keynotes, shorts paper and doctoral symposiums.

4 RESULTS

For an overview of the primary studies demographics, Figure 2 depicts the number of publications over the years and the publication venues. Studies on SPL regression testing have been published since 2005, mainly on conferences, symposiums, and workshops (19 studies, ≈ 70%). Only five studies have been published in journals (≈ 30%). Although the year with most publications was 2012, the trend line shows an increase in the number of publications in the last years (Figure 2(a)). The studies come from 21 different venues, as presented in Figure 2(b). This means that research on SPL regression testing is disseminated in the wide range of venues (events and journals).



(a) Publications per years



(b) Publication venues

Figure 2: Primary sources overview.

To answer the RQs of this study, we refer to Table 3 that chronologically presents each primary study according to the dimensions and categories presented in the previous section. Answers to the RQs are presented in the following subsections.

4.1 RQ1. Regression Testing Techniques

The techniques used in the studies are presented in the 2nd to 5th columns of Table 3. Most of the primary studies apply the *selection* technique (18 out of 27, 67%). This was expected, since the tester would prefer to select test cases and/or products related to

[3]Primary Studies Dataset: https://docs.google.com/spreadsheets/d/1LdrD1h76pTRfIdo4auwtEloca48sZLJnxTUDWuV76Uc/edit?usp=sharing

Table 3: Details of the SPL regression testing approaches found in the primary sources.

Paper	Technique				Input					Output			Strategy			Testing criterion			
	Prioritization	Minimization	Selection	Retest all	Variability model	Test suite	Source code	State machine	Other	List of test cases	List of products	Other	Expert-based	AI-based	Comparison-based	Coverage-based	Combinatorial	Fault-based	Other
(Al Dallal and Sorenson, 2005)				✓				✓	✓	✓			✓			✓		✓	
(Qu et al., 2008)	✓						✓		✓		✓			✓		✓	✓	✓	
(Al-Dallal and Sorenson, 2008)			✓			✓	✓			✓			✓						✓
(Neto et al., 2010)	✓		✓		✓	✓	✓		✓	✓					✓			✓	
(Lochau et al., 2012)			✓		✓			✓		✓	✓				✓	✓			
(Heider et al., 2012)				✓	✓							✓			✓			✓	
(Remmel et al., 2011)			✓		✓							✓			✓			✓	
(Robinson and White, 2012)			✓				✓			✓		✓			✓			✓	
(Qu et al., 2012)			✓				✓				✓				✓	✓		✓	
(Neto et al., 2012)	✓		✓		✓	✓	✓		✓	✓					✓			✓	
(Remmel et al., 2013)			✓		✓					✓		✓			✓		✓		
(Lachmann et al., 2015)	✓							✓	✓	✓					✓	✓		✓	
(Lity et al., 2016)			✓			✓		✓	✓	✓					✓				✓
(Lachmann et al., 2016)	✓					✓		✓		✓					✓	✓			✓
(Al-Hajjaji et al., 2017)	✓							✓	✓		✓				✓			✓	✓
(Marijan et al., 2017)	✓	✓				✓				✓				✓		✓		✓	
(Lachmann et al., 2017)	✓							✓	✓			✓			✓			✓	✓
(Marijan and Liaaen, 2018)			✓			✓			✓	✓					✓	✓	✓	✓	✓
(Souto and d'Amorim, 2018)			✓		✓	✓			✓	✓	✓			✓			✓	✓	
(Jung et al., 2019)			✓			✓	✓			✓		✓			✓	✓			
(Fischer et al., 2019)			✓				✓			✓					✓	✓			
(Marijan et al., 2019)			✓			✓			✓	✓				✓				✓	✓
(Lity et al., 2019)			✓					✓		✓					✓	✓		✓	✓
(Jung et al., 2020)			✓				✓			✓					✓			✓	
(Fischer et al., 2020)			✓				✓			✓					✓	✓			
(Lima et al., 2020a)	✓								✓	✓				✓				✓	✓
(Hajri et al., 2020)	✓		✓						✓	✓					✓	✓		✓	
Total	10	1	18	2	7	10	10	8	13	21	5	6	2	5	20	13	4	19	8

changed features. Among the studies on this category, one deals with the selection of SPL products (Souto and d'Amorim, 2018) and 17 focus on the selection of test cases (Neto et al., 2010; Al-Dallal and Sorenson, 2008; Lochau et al., 2012; Remmel et al., 2011; Robinson and White, 2012; Neto et al., 2012; Remmel et al., 2013; Lity et al., 2016; Marijan and Liaaen, 2018; Souto and d'Amorim, 2018; Jung et al., 2019; Fischer et al., 2019; Marijan et al., 2019; Lity et al., 2019).

Prioritization is the second most common technique. We found 10 studies (37%) applying this regression testing technique, which aims at establishing an order of test cases or products that must be executed firstly. These test cases or products are usually those with high probability of failing. The goal is detecting faults as early as possible, making SPL regression testing more effective and efficient. Three primary sources prioritize SPL products (Qu et al., 2008; Qu et al., 2012; Al-Hajjaji et al., 2017) and six ones the test cases (Neto et al., 2010; Neto et al., 2012; Lachmann et al., 2015; Lachmann et al., 2016; Marijan et al., 2017; Lachmann et al., 2017; Lima et al., 2020a; Hajri et al., 2020).

Retest all and minimization are the techniques less investigated. Two studies apply the *retest all* technique; both focusing on test cases. The first one focuses on the creation of test cases that can be reused as much as possible in different configurations (Al Dallal and Sorenson, 2005). The second one uses the test suite to identify changes in products derived with the same configuration, but from before and after SPL evolution (Heider et al., 2012). Regarding *minimization*, only one paper uses this technique, together with prioritization, to reduce testing execution of continuous integration cycles (Marijan et al., 2017). This study focuses on minimization of test cases. Prioritization and selection are combined in three studies (Neto et al., 2010; Neto et al., 2012; Hajri et al., 2020).

Answer to RQ1: *Most of the primary studies apply the selection technique (67%), followed by prioritization (37%). Minimization is applied only by one study. Only four studies combine techniques.*

4.2 RQ2. Strategies

Most approaches (20 out of 27, 74%) use a *comparison-based* strategy to apply the regression testing techniques. A possible reason for this is the structure of SPL. This strategy relies on comparison of whole SPLs or products versions before and after modifications (change impact analysis) (Neto et al., 2010; Remmel et al., 2011; Qu et al., 2012; Neto et al., 2012; Remmel et al., 2013; Marijan and Liaaen, 2018; Jung et al., 2020; Hajri et al., 2020), comparison of similarities and differences among test cases (overlap analysis) (Jung et al., 2019; Fischer et al., 2019; Fischer et al., 2020), and delta-oriented analysis of products differences (Lochau et al., 2012; Lachmann et al., 2015; Lity et al., 2016; Lachmann et al., 2016; Al-Hajjaji et al., 2017; Lachmann et al., 2017; Lity et al., 2019).

AI-based strategies use optimization or machine learning algorithms. The algorithms are used for selection of SPL products to conduct time-space efficiently regression testing (Souto and d'Amorim, 2018). This allows prioritization and minimization of test cases due to limited time budget in continuous integration (Marijan et al., 2017; Marijan et al., 2019; Lima et al., 2020a), and for test case prioritization in order to maximize coverage of a set of SPL products (Qu et al., 2008).

Two studies apply an *expert-based* strategy in which experts manually design reusable test cases to be used in a greater number of products (Al Dallal and Sorenson, 2005; Al-Dallal and Sorenson, 2008).

Answer to RQ2: *Most strategies (74%) belong to the comparison-based category. AI-based strategies are poorly explored. This category includes only five studies (≈ 18%). The expert-based strategy is explored only in two works that use manual approaches.*

4.3 RQ3. Input and Output Artifacts

There is no predominant type of artifact used as input. Test suite was expected to be widely used as input, as well as source code (10 studies found in each category, 37%). Taking into account the focus of the primary sources on SPLs, the variability model is also a common artifact. Interestingly, state machines are used very often (8 studies, 30%). State machines are used to represent products and analyze differences, allowing the application of regression testing techniques. In the category *other*, we also observed as input: UML models (Al Dallal and Sorenson, 2005), case models (Hajri et al., 2020), configuration options (Qu et al., 2008), feature dependencies (Neto et al., 2012; Neto et al., 2010), software architecture (Lachmann et al., 2015; Lity et al., 2016; Al-Hajjaji et al., 2017; Lachmann et al., 2017; Neto et al., 2012; Neto et al., 2010), history of fault-detection (Marijan and Liaaen, 2018; Lima et al., 2020a), and list of changes (Marijan et al., 2019; Souto and d'Amorim, 2018).

Regarding output, a list of prioritized, minimized, or selected test cases is by far the most common artifact (21 studies, 74%). A list of products is generated in five approaches. Both lists are output in only two studies (Lochau et al., 2012; Souto and d'Amorim, 2018). In the category *other*, the output artifacts are: change impact reports (Heider et al., 2012), failure report (Remmel et al., 2011; Robinson and White, 2012; Lachmann et al., 2017; Remmel et al., 2011), and test cases partition table (Jung et al., 2019).

Answer to RQ3: *A large variety of artifacts are used; 13 studies (≈ 48%) belong to the category others, followed by the category source code with 10 studies (≈ 37%) and the category test cases with 10 studies (≈ 37%). On the other hand, two main categories of outputs were identified, namely lists of test cases (the prevalent) and lists of products.*

4.4 RQ4. Testing Criteria

Fault-based is the most common criterion adopted in nine studies (70%). This criterion uses practical knowledge to select, minimize or prioritize test cases or SPL products. In other words, products or test cases more likely to fail, or that have a history of fails, deserve more attention.

Coverage-based criteria, which is also widely used for single product systems, is the focus of 13 studies (48%). Examples of elements to be covered in this category are code elements (Qu et al., 2008), all-transitions in state machines (Lity et al., 2019), and architecture changes (Lachmann et al., 2017).

Due to the nature of SPLs, i.e., based on combination of features, criteria derived from the *combinatorial* testing is also observed in primary studies, such as pair-wise. This criterion aims to cover variant interaction to select a subset of all possible variant combinations (Remmel et al., 2013; Qu et al., 2008; Marijan and Liaaen, 2018). In the category *other* we observe criteria based on test execution time (Al-Dallal and Sorenson, 2008; Marijan et al., 2019), risk (Lachmann et al., 2017), change impact (Lity et al., 2016; Lity et al., 2019), signals (Lachmann et al., 2016; Al-Hajjaji et al., 2017), dissimilarity (Lachmann et al., 2016), and historical information (Marijan and Liaaen, 2018).

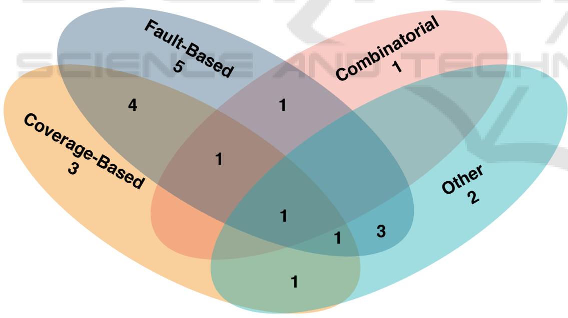


Figure 3: Combination of testing criteria.

Figure 3 presents the number of primary sources per criteria and their intersection. As traditional in software testing, a combination of criteria can be used for evaluating test cases (Melo et al., 2019). For example, fault-based and coverage-based are commonly applied with each other. In three primary sources, fault-based is also applied together with at least two additional criteria (Lity et al., 2019; Qu et al., 2008; Marijan and Liaaen, 2018). One study uses the four types of criteria (Marijan and Liaaen, 2018), two studies use three types (Qu et al., 2008; Lity et al., 2019), and nine a pair of criteria (Al Dallal and Sorenson, 2005; Qu et al., 2012; Lachmann et al., 2015; Lachmann et al., 2016; Al-Hajjaji et al., 2017; Marijan et al., 2017; Lachmann et al., 2017; Souto and d'Amorim, 2018; Marijan et al., 2019). This shows a trend on combining different criteria in the proposed approaches.

Answer to RQ4: *Fault-based and coverage-based criteria are the most applied in, respectively, 70% and 48% of the studies. Many studies (58%) combine more than one criterion.*

5 A ROADMAP FOR FUTURE RESEARCH ON SPL REGRESSION TESTING

As a result of the reading, analysis, and discussion of the primary studies, we defined a roadmap to serve as a guide for researchers and practitioners to contribute to the body of knowledge, both in theory and practice, on SPL regression testing. This roadmap consists of six research opportunities and future directions related to gaps identified in the literature, trends concerned to emerging technologies, and limitations of existing approaches to their application in practice. Each opportunity of the roadmap is described next.

1. To Explore Intelligent and Learning Approaches: As with many other activities of software engineering, software testing is influenced by several factors, criteria, and constraints. This requires proper approaches to satisfactorily aid SPL testing, such as multi-criteria optimization strategies from the field of Search-Based Software Engineering (SBSE) (Colanzi et al., 2020). SBSE uses artificial intelligence to solve software engineering problems. In RQ2 we observed the opportunity of exploring artificial intelligence and machine learning techniques for SPL regression testing. A promising opportunity in this direction is the use of multi/many-objective search techniques, as for example, assessing the performance and scalability (Wang et al., 2014). These algorithms can also be used to deal with the trade-off between test case prioritization compared to prioritization of products (Al-Hajjaji et al., 2017).

2. To Explore Hybrid Approaches: Combining different techniques and strategies to take advantage of the strength of each one can lead to better results of SPL regression testing approaches. As we discussed in the answer of RQ1, minimization techniques are poorly investigated, which could be combined with selection or prioritization techniques. Also, adding semantic impact analysis techniques in combination with syntactic techniques can enable immediate feedback on the change impact (Robinson and White, 2012). Also, the combination of product selection, configuration augmentation, or reduction techniques may further improve the testing effectiveness and efficiency, compared to either approach alone (Qu et al., 2012). Finally, considering both fine and coarse granularity of artifacts might allow more comprehensive decisions during test activity (Lity et al., 2016).

3. To Derive Test Cases Automatically based on SPL Modifications: Ideally, test cases might be derived in the same way products variants are (Fischer et al., 2019; Fischer et al., 2020). However, quality of initial product tests is particularly crucial for the subsequent iterations (Lochau et al., 2012). Another challenge to achieve automatic derivation of test cases is that building and maintaining traceability links between features and test cases can be complex and time-consuming. For SPLs that evolve at a moderate rate, building a feature model and traceability links requires high upfront costs (Marijan and Sen, 2017).

4. To Properly Test Feature Interactions: Two studies highlight the need of investigating feature interactions in depth. One study suggests considering feature interaction failures across historical executions (Marijan and Liaaen, 2018). Another study mentions the use of results from testing different configuration combinations to discover interactions among configuration options (Fischer et al., 2019).

5. To Support SPL Regression Testing in Continuous Integration: Continuous Integration has become a *de facto* practice in software development, even in the context of SPLs (Lima et al., 2020b). This also affects regression testing, as there usually exist time budgets (Marijan and Liaaen, 2018). For example, one study suggests splitting test activity in two phases, namely one for the nightly runs and another more complex one for release-acceptance testing (Remmel et al., 2011). And finally, two primary sources mention investigating trade-off between early fault detection and efficient regression testing (Lity et al., 2019) and experimenting variable time budgets with the test prioritization and reduction approaches (Marijan et al., 2019).

6. To Manage Regression Testing for SPL Evolution in Space and Time: It has been acknowledged that SPLs evolve in space (introduction or exclusion of features) and in time (version of a same feature) (Berger et al., 2019; Michelon et al., 2020b; Michelon et al., 2020a). However, in the literature of SPL regression testing, we have not found approaches with such characteristics. For example, SPLs evolving in space and time require lifting of traditional analyses to consider both dimensions of variation (Berger et al., 2019). However, we argue that there is a potential for reusing existing regression analyses for testing purposes, some of which has been already exploited (Heider et al., 2012; Lochau et al., 2012).

6 CONCLUDING REMARKS

This work overviews existing literature on SPL regression testing, a fundamental activity to reveal problems and guarantee that SPL maintenance and evolution were performed accordingly. Test case selection and prioritization are the most addressed techniques. A wide range of input artifacts are used, and the main output is the list of selected/prioritized/ minimized test cases. Comparison-based strategy is by far the most applied strategy, together with fault-based and coverage-based criteria.

Based on the results, we defined a research roadmap for short and medium term. This roadmap describes some research opportunities. Among them, we can mention to introduce multi-criteria approaches, to explore the advantages of hybrid approaches, automatic derivation of test cases based on SPL modifications, proper testing of feature interactions, to support the SPL regression testing in continuous integration environments, and management of regression testing for SPLs evolving in space and time. In addition to this, we observe some opportunities based on limitations and trends related to the investigation of SPL regression testing in the context of emerging technologies and practical needs, such as to properly test feature interactions.

REFERENCES

Abbas, M., Jongeling, R., Lindskog, C., Enoiu, E. P., Saadatmand, M., and Sundmark, D. (2020). Product line adoption in industry: An experience report from the railway domain. In *24th ACM Conference on Systems and Software Product Line-Volume A*, SPLC '20, New York, NY, USA. ACM.

Al Dallal, J. and Sorenson, P. (2005). Reusing class-based test cases for testing object-oriented framework interface classes. *Journal of Software Maintenance and Evolution: Research and Practice*, 17(3):169–196.

Al-Dallal, J. and Sorenson, P. G. (2008). Testing software assets of framework-based product families during application engineering stage. *JSW*, 3(5):11–25.

Al-Hajjaji, M., Lity, S., Lachmann, R., Thüm, T., Schaefer, I., and Saake, G. (2017). Delta-oriented product prioritization for similarity-based product-line testing. In *2017 IEEE/ACM 2nd International Workshop on Variability and Complexity in Software Design (VACE)*, pages 34–40. IEEE.

Assunção, W. K. G., Krüger, J., and Mendonça, W. D. F. (2020). Variability management meets microservices: Six challenges of re-engineering microservice-based webshops. In *24th ACM Conference on Systems and Software Product Line-Volume A*. ACM.

Berger, T., Chechik, M., Kehrer, T., and Wimmer, M. (2019). Software Evolution in Time and Space: Unifying Version and Variability Management (Seminar 19191). *Dagstuhl Reports*, 9(5):1–30.

bin Ali, N., Engström, E., Taromirad, M., Mousavi, M. R., Minhas, N. M., Helgesson, D., Kunze, S., and Varshosaz, M. (2019). On the search for industry-relevant regression testing research. *Empirical Software Engineering*, 24(4):2020–2055.

Colanzi, T. E., Assunção, W. K., Vergilio, S. R., Farah, P. R., and Guizzo, G. (2020). The symposium on search-based software engineering: Past, present and future. *Information and Software Technology*, 127:106372.

do Carmo Machado, I., McGregor, J. D., Cavalcanti, Y. C., and de Almeida, E. S. (2014). On strategies for testing software product lines: A systematic literature review. *Information and Software Technology*, 56(10):1183–1199.

Engström, E. (2010a). Exploring regression testing and software product line testing: Research and state of practice. Licentiate Thesis, Department of Computer Science, Lund University.

Engström, E. (2010b). Regression test selection and product line system testing. In *3rd International Conference on Software Testing, Verification and Validation*, pages 512–515. IEEE.

Engström, E. and Runeson, P. (2011). Software product line testing–a systematic mapping study. *Information and Software Technology*, 53(1):2–13.

Fischer, S., Michelon, G. K., Ramler, R., Linsbauer, L., and Egyed, A. (2020). Automated test reuse for highly configurable software. *Empirical Software Engineering*, 25(6):5295–5332.

Fischer, S., Ramler, R., Linsbauer, L., and Egyed, A. (2019). Automating test reuse for highly configurable software. In *Proceedings of the 23rd International Systems and Software Product Line Conference-Volume A*, pages 1–11.

Geraldi, R. T., Reinehr, S., and Malucelli, A. (2020). Software product line applied to the internet of things: A systematic literature review. *Information and Software Technology*, 124:106293.

Grüner, S., Burger, A., Kantonen, T., and Rückert, J. (2020). Incremental migration to software product line engineering. In *24th ACM Conference on Systems and Software Product Line-Volume A*. ACM.

Hajri, I., Goknil, A., Pastore, F., and Briand, L. C. (2020). Automating system test case classification and prioritization for use case-driven testing in product lines. *Empirical Software Engineering*, 25(5):3711–3769.

Heider, W., Rabiser, R., Grünbacher, P., and Lettner, D. (2012). Using regression testing to analyze the impact of changes to variability models on products. In *Proceedings of the 16th International Software Product Line Conference-Volume 1*, pages 196–205.

Jung, P., Kang, S., and Lee, J. (2019). Automated code-based test selection for software product line regression testing. *Journal of Systems and Software*, 158:110419.

Jung, P., Kang, S., and Lee, J. (2020). Efficient regression testing of software product lines by reducing redundant test executions. *Applied Sciences*, 10(23):8686.

Lachmann, R., Beddig, S., Lity, S., Schulze, S., and Schaefer, I. (2017). Risk-based integration testing of software product lines. In *Proceedings of the Eleventh International Workshop on Variability Modelling of Software-intensive Systems*, pages 52–59.

Lachmann, R., Lity, S., Al-Hajjaji, M., Fürchtegott, F., and Schaefer, I. (2016). Fine-grained test case prioritization for integration testing of delta-oriented software product lines. In *Proceedings of the 7th International Workshop on Feature-Oriented Software Development*, pages 1–10.

Lachmann, R., Lity, S., Lischke, S., Beddig, S., Schulze, S., and Schaefer, I. (2015). Delta-oriented test case prioritization for integration testing of software product lines. In *Proceedings of the 19th International Conference on Software Product Line*, pages 81–90.

Lee, J., Kang, S., and Lee, D. (2012). A survey on software product line testing. In *16th International Software Product Line Conference-Volume 1*, pages 31–40.

Lima, J. A. P., Mendonça, W. D., Vergilio, S. R., and Assunção, W. K. (2020a). Learning-based prioritization of test cases in continuous integration of highly-configurable software. In *Proceedings of the 24th ACM Conference on Systems and Software Product Line: Volume A-Volume A*, pages 1–11.

Lima, J. A. P., Mendonça, W. D. F., Vergilio, S. R., and Assunção, W. K. G. (2020b). Learning-based prioritization of test cases in continuous integration of highly-configurable software. In *24th ACM Conference on Systems and Software Product Line-Volume A*. ACM.

Linden, F. J. v. d., Schmid, K., and Rommes, E. (2007). *Software Product Lines in Action: The Best Industrial Practice in Product Line Engineering*. Springer.

Lity, S., Morbach, T., Thüm, T., and Schaefer, I. (2016). Applying incremental model slicing to product-line regression testing. In *International Conference on Software Reuse*, pages 3–19. Springer.

Lity, S., Nieke, M., Thüm, T., and Schaefer, I. (2019). Retest test selection for product-line regression testing of variants and versions of variants. *Journal of Systems and Software*, 147:46–63.

Lochau, M., Schaefer, I., Kamischke, J., and Lity, S. (2012). Incremental model-based testing of delta-oriented software product lines. In *International Conference on Tests and Proofs*, pages 67–82. Springer.

Marijan, D., Gotlieb, A., and Liaaen, M. (2019). A learning algorithm for optimizing continuous integration development and testing practice. *Software: Practice and Experience*, 49(2):192–213.

Marijan, D. and Liaaen, M. (2018). Practical selective regression testing with effective redundancy in interleaved tests. In *Proceedings of the 40th International*

Conference on Software Engineering: Software Engineering in Practice, pages 153–162.

Marijan, D., Liaaen, M., Gotlieb, A., Sen, S., and Ieva, C. (2017). Titan: Test suite optimization for highly configurable software. In *2017 IEEE International Conference on Software Testing, Verification and Validation (ICST)*, pages 524–531. IEEE.

Marijan, D. and Sen, S. (2017). Detecting and reducing redundancy in software testing for highly configurable systems. In *2017 IEEE 18th International Symposium on High Assurance Systems Engineering (HASE)*, pages 96–99. IEEE.

Marques, M., Simmonds, J., Rossel, P. O., and Bastarrica, M. C. (2019). Software product line evolution: A systematic literature review. *Information and Software Technology*, 105:190 – 208.

Melo, S. M., Carver, J. C., Souza, P. S., and Souza, S. R. (2019). Empirical research on concurrent software testing: A systematic mapping study. *Information and Software Technology*, 105:226–251.

Mendonça, W. D. F., Assunção, W. K. G., Estanislau, L. V., Vergilio, S. R., and Garcia, A. (2020). Towards a microservices-based product line with multi-objective evolutionary algorithms. In *IEEE Congress on Evolutionary Computation*. IEEE.

Michelon, G. K., Obermann, D., Assunção, W. K. G., Linsbauer, L., Grünbacher, P., and Egyed, A. (2020a). Mining feature revisions in highly-configurable software systems. In *24th ACM International Systems and Software Product Line Conference - Volume B*, page 74–78. ACM.

Michelon, G. K., Obermann, D., Linsbauer, L., Assunção, W. K. G., Grünbacher, P., and Egyed, A. (2020b). Locating feature revisions in software systems evolving in space and time. In *24th ACM Conference on Systems and Software Product Line-Volume A*. ACM.

Minhas, N. M., Petersen, K., Ali, N. B., and Wnuk, K. (2017). Regression testing goals-view of practitioners and researchers. In *2017 24th Asia-Pacific Software Engineering Conference Workshops (APSECW)*, pages 25–31. IEEE.

Neto, P. A. d. M. S., do Carmo Machado, I., Cavalcanti, Y. C., De Almeida, E. S., Garcia, V. C., and de Lemos Meira, S. R. (2010). A regression testing approach for software product lines architectures. In *2010 Fourth Brazilian Symposium on Software Components, Architectures and Reuse*, pages 41–50. IEEE.

Neto, P. A. d. M. S., do Carmo Machado, I., Cavalcanti, Y. C., de Almeida, E. S., Garcia, V. C., and de Lemos Meira, S. R. (2012). An experimental study to evaluate a spl architecture regression testing approach. In *2012 IEEE 13th International Conference on Information Reuse & Integration (IRI)*, pages 608–615. IEEE.

Nomme, S. S. (2020). Composing software product lines with machine learning components. Master's thesis, University of Oslo.

Petersen, K., Vakkalanka, S., and Kuzniarz, L. (2015). Guidelines for conducting systematic mapping studies in software engineering: An update. *Information and Software Technology*, 64:1–18.

Qu, X., Acharya, M., and Robinson, B. (2012). Configuration selection using code change impact analysis for regression testing. In *2012 28th IEEE International Conference on Software Maintenance (ICSM)*, pages 129–138. IEEE.

Qu, X., Cohen, M. B., and Rothermel, G. (2008). Configuration-aware regression testing: an empirical study of sampling and prioritization. In *Proceedings of the 2008 international symposium on Software testing and analysis*, pages 75–86.

Remmel, H., Paech, B., Bastian, P., and Engwer, C. (2011). System testing a scientific framework using a regression-test environment. *Computing in Science & Engineering*, 14(2):38–45.

Remmel, H., Paech, B., Engwer, C., and Bastian, P. (2013). Design and rationale of a quality assurance process for a scientific framework. In *2013 5th International Workshop on Software Engineering for Computational Science and Engineering (SE-CSE)*, pages 58–67. IEEE.

Robinson, B. and White, L. (2012). On the testing of user-configurable software systems using firewalls. *Software Testing, Verification and Reliability*, 22(1):3–31.

Runeson, P. and Engström, E. (2012). Regression testing in software product line engineering. In *Advances in computers*, volume 86, pages 223–263. Elsevier.

Souto, S. and d'Amorim, M. (2018). Time-space efficient regression testing for configurable systems. *Journal of Systems and Software*, 137:733–746.

Wang, S., Buchmann, D., Ali, S., Gotlieb, A., Pradhan, D., and Liaaen, M. (2014). Multi-objective test prioritization in software product line testing: an industrial case study. In *18th International Software Product Line Conference-Volume 1*, pages 32–41.

Yoo, S. and Harman, M. (2012). Regression testing minimization, selection and prioritization: a survey. *Software testing, verification and reliability*, 22(2):67–120.