International Journal on Advanced Science Engineering Information Technology

Vol.7 (2017) No. 4-2
ISSN: 2088-5334

Evaluation of Software Product Line Test Case Prioritization Technique

Muhammad Sahak[#], Shahliza Abd Halim[#,*], Dayang Norhayati Abang Jawawi[#,*], Mohd Adham Isa[#,*]

[#]*Software Engineering Research Group, Universiti Teknologi Malaysia, Skudai 81310 Johor Malaysia*

[*]*Software Engineering Department, Universiti Teknologi Malaysia, Skudai 81310 Johor Malaysia*
E-mail: shahliza@utm.my

***Abstract*— Software product line (SPL) engineering paradigm is commonly used to handle commonalities and variabilities of business applications to satisfy the specific needs or goal of a particular market. However, due to time and space complexities, testing all products is not feasible, and SPL testing is proven to be difficult due to a combinatorial explosion of the number of products to be considered. Combinatorial interaction testing (CIT) is suggested to reduce the size of test suites to overcome budget limitations and deadlines. CIT is conducted to fulfill certain quality attributes. This method can be further improvised through the prioritization of list configuration generated from CIT to gain better results in terms of efficiency and scalability, However, to the best of our knowledge, not much research has been done to evaluate existing Test Case Prioritization (TCP) techniques in SPL. This paper provides a survey of existing works on test case prioritization technique. This study provides classification and compares the best technique, trends, gaps and proposed frameworks based on the literature. The evaluation and discussion are using Normative Information Model-based Systems Analysis and Design (NIMSAD) on aspects that include context, content, and validation. The discussion highlights the lack of technique for scalability issue in SPL with most of the work is on academia setting but not on industrial practices.**

***Keywords*— NIMSAD; software product lines; regression testing**

I. INTRODUCTION

Software Product Line (SPL) is founded on the concept of reusability of products from the same family which is systematically being reused either as common assets or only shared by a subset of the family [1]. Many software organizations changed their development process from a single system to SPL to take advantage of the reduction in time, cost, and effort to market while significantly increase the quality of derived products. Among the common quality assurance methods in SPL is SPL testing. The difference between testing a single system and SPL testing is, in a single system, only one product is tested at a time whereas SPL testing tests a number of products at one time which contributes to the need for systematic testing process due to the commonality and variability of features. The testing process becomes more complicated when the number of configuration grows exponentially with the number of features or known as a combinatorial explosion. In addition, computing these large number of products in the presence of constraints proves to be a tough issue and scalability is considered as an open research area in SPL [3]. Thus, it prompts the need for a new method to overcome these challenges.

Among the promising methods is a regression testing method which is able to reduce the number of test artifacts in a single system through minimization, selection, and prioritization. This method has been adapted into SPL in the works of [1], [2], [3], [6]. Test Case Prioritization (TCP) is known as the best regression technique that can be used to overcome these issues by rearranging the test cases which cover a lot of changed elements of a product variant in order to achieve the desired criteria. A typical TCP phase in SPL is shown in Fig. 1. This study divides the focus of existing works into two quality attributes which are efficiency and scalability.

In terms of efficiency, software tester can benefit the most from prioritization where only a few most important test cases from the test suite will be selected thus helps in increasing the interaction coverage between SPL product under test as fast as possible. Whereas, for scalability issue, the relaxation of the T-wise criterion by using TCP helps contribute to lower testing effort and decent coverage. This is because, in SPL testing, high T-wise criterion requires a lot of budgets. Contributions of this paper are as the following: (i) provide a classification scheme for our own proposed TCP model which is inspired based on three main phases described in Fig. 1; (ii) critically evaluate existing works utilizing TCP technique to determine their strengths and weaknesses; and (iii) Propose a framework for SPL based test case prioritization.

The remainder of the paper is structured as follows: Section II will discuss on material and methodology used. Section III will describe the overview of test case prioritization, and Section IV discusses on our comparison result. Next, Section V will describe our proposed Dissimilarity-Based Prioritization Framework (DBPF).

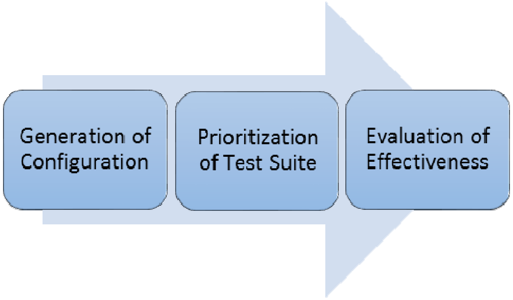
Generation of Configuration
Prioritization of Test Suite
Evaluation of Effectiveness

Fig. 1 An overview of test case prioritization phase

II. Material and Method

In recent years, there has been an increasing number of publications that review SPL testings such as [9], [10], [14], [15], [16]. One of the publications that contribute to SPL testing is a systematic study done by Engstrom and Runeson [14]. Their study focuses on identifying useful approaches and involves six SPL testing key areas namely test organization and process, test management, testability and acceptance testing, integration testing, unit testing and test automation. In their findings, they classify SPL testing as an immature area because of lack research papers published in journals compared to workshops and conferences. In addition, SPL testing involves great effort and evaluations are costly, thus explaining the low number of SPL empirical studies.

Although the investigation contributes much more to the discussion, there are many other investigations that have been conducted on this issue. For example, Lee *et al.* [9] presented a survey on SPL testing and discussed the two main activities of SPL testing that are domain and application engineering. The paper also discussed the lack of studies on these separate engineering activities and their relationship with each other. Moreover, this study also provides the context of SPL testing process in a detailed manner. Da Mota *et al.* [15] conducted systematic mapping study of SPL testing to investigate state-of-the-art testing practices, synthesize available evidence, and identify gaps between required techniques and existing approaches that are available in the literature. One of the points highlighted in the study is the lack of studies made by the industry because the majority of existing works is done by the academia. Moreover, this discussion provides some significance inputs for future research, for example, non-functional testing quality attributes such as response time, performance and scalability which might differ in several SPL instances.

Recently, a systematic literature study has been conducted by Machado *et al.* [16] on strategies for testing software products. The paper explained on how products are selected from a very large set of possible products for asset testing, and how each selected product is tested. The results from this study confirmed the increase in SPL testing interest. However there is a lack of effective methods and techniques to solve existing problems, and further investigation is needed because the number of existing general techniques is too low.

Only one researcher focus on the prioritization criteria, a work Sanchez *et al.* [10] proposed five different prioritization criteria on common metrics of feature models in SPL such as Cross-Tree-Constraints Ratio (CTCR), Coefficient of Connectivity-Density (CoC), VC&CC prioritization, commonality, and dissimilarity. Though Sanchez has contributed to a detail classification on the prioritization technique; the paper lacks concentration on SPL based prioritization testing which will be the focus of the paper and elaborated in the following subtopics.

The comparative evaluation framework is used to compare between existing works. The framework is based on Normative Information Model-based Systems Analysis and Design (NIMSAD) framework [7]. The framework suggests that effective application of a method depends on three elements: the method itself, the person who applies the method, and the context in which the method is applied [8]. Four categories suggested by NIMSAD are context, elements, user, and validation. Our framework generally consists of three categories which are context, content, and validation. These three categories are chosen because they suit our purpose. Context is used to describe general information on existing TCP works. Next, content is divided into two namely generation of configuration and prioritization of test suite. These two categories help to further improve the existing techniques.

Lastly, validation category from NIMSAD is extended into our evaluation of effectiveness phase in our framework which is used to determine the application domain the technique is applied and the type of evaluation method used to investigate the effectiveness of the method and lastly their scale of evaluation. Table 1 shows the description of the framework category.

A. Context

The aim of the context category is to discover basic information on TCP techniques which include their goals, domain application, method input, output, elements inside the problem areas that the techniques are trying to solve, and lastly quality attributes achieved. These details should be extracted carefully in order to distinguish it from other works and provide us with the scope of each technique involved.

B. Content

This category focuses on the six elements involved in TCP works. These elements are further divided into two DBPF phases which are a generation of configuration and prioritization of test suite. Each phase contains three elements.

TABLE I
COMPARATIVE EVALUATION FRAMEWORK DESCRIPTION

NIMSAD Category	Phase	Elements	Related Questions
Context		Goal	What is the goal of the approach?
		Domain Application	Which area is the technique applied to?
		Method Input	What are the inputs of the approach?
		Method Output	What are the results of the method?
		Central Element Focus	What are the main problems that the technique is trying to solve?
		Quality Attributes	What kind of qualities does the technique trying to achieve?
Content	Generation of Configuration	Testing level	What type of testing is conducted by the technique?
		Sampling Algorithm	What type of sampling algorithm is used?
		T-wise coverage	What is the number of T-wise coverage used in this method?
	Prioritization of Test Suite	Prioritization criteria	How does this method prioritize test cases?
		Tools used	What kind of tools used?
		Limitations	What kind of limitations exists for the technique?
Validation	Evaluation of Effectiveness	Application Domain	What areas are focused in TCP technique?
		Evaluation method	What types of evaluation method is used?
		Scale of evaluation	What evaluation scale is used by the TCP technique?

The categorization under these two phases helps our study in analysing the issues that the techniques are trying to solve under their own scopes which are mentioned in our sub-criteria. This categorization will surely help in identifying the strengths and weaknesses of existing works.

First, for testing level, there are four types of SPL testing namely system testing, integration testing, acceptance testing and unit testing. These testing types are performed in two engineering phases, domain and application engineering [9]. It is beneficial to know the specific testing types included in each phase such Domain engineering consists unit testing, and integration testing meanwhile Application engineering consists system and acceptance testing. Secondly, a sampling algorithm is used in TCP works for combinatorial interaction testing in order to reduce the number of test suites. By identifying which method uses sampling algorithm and which method does not, this study will be able to formulate the reason why certain TCP technique utilizes sampling algorithm.

Third, analysis of T-wise coverage of each TCP contributes is important in identifying the problem that the technique's trying to solve. The prioritization of test cases is proposed to overcome scalability issue which is known to affect various current CIT [3]. Testing feature model with $T > 3$ is not cost effective and time-consuming. Thus, the analysis will certainly help to determine the number of coverage that an approach concentrates on. The values of T-wise coverage are divided into two parts with low ($T \leq 3$) and high ($T > 3$).

Fourth, the prioritization criteria: Analysis of prioritization criteria used is needed in order to identify the most suitable method to be used. There are five types of prioritization criteria specified for test case prioritization in SPL [10]. The criteria consist of their own process on how to conduct prioritization of test cases. Our study provides one additional column labelled unspecified for any method that is not listed by the criteria. The prioritization criteria as described in [10] is shown in Table 2.

TABLE II
PRIORITIZATION CRITERIA

No.	Prioritization Criteria	Aim
1.	Cross Tree Constraint Ratio (CTCR)	Prioritize based on complexity of products constraints
2.	Coefficient of Connectivity-Density (CoC)	Prioritize the products according to their Coefficient of Connectivity-Density(CoC).
3.	Variability Coverage & Cyclomatic Complexity (VC&CC)	Prioritize based on variability coverage and cyclomatic complexity
4.	Commonality	Prioritize based on common features between product
5.	Dissimilarity	Prioritize based on different features between product

Moreover, there are various tools used in TCP works. This investigation is beneficial because it helps to determine the best tool to be used in TCP techniques. Lastly, Each technique investigated has their own limitations in conducting the prioritization process. Identifying the limitations of each technique will help to offer insights on rooms for improvements in future research.

C. *Validation*

This category focuses on application domain applied, evaluation method used and scale of evaluation. For the application domain category, it helps in identifying which areas are being focused by SPL researchers. Next, the evaluation method is important to determine the evaluation methods used by each technique. Lastly, for the scale of evaluation is used in this paper for TCP technique's scale of evaluation is adopted from [14]. It is important to know the evaluation scale in order to identify the impact of the evaluation conducted. Value for each scale is shown in Table 3.

TABLE III
PRIORITIZATION CRITERIA

No.	Type	Scale
1.	Toy-example	Small-scale
2.	Down-scaled real world	Medium scale
3.	Industrial	Large-scale

1. OVERVIEW OF TEST CASE PRIORITIZATION TECHNIQUE

For this paper, we only select TCP studies that fulfil our criteria such as the works is published between 2011 to 2016, limited to SPL and focus on the uses of TCP technique. Six papers that fulfilled the criteria stated above are given below:

GOSP - Goal-oriented test case selection & prioritization [2]
WMP - Weight modelling prioritization [4]
SHP - Similarity heuristic prioritization [3]
DOP - Delta-oriented test case prioritization [5]
SP - Statistical prioritization [1]
SBP - Similarity-based prioritization [6]

A. Goal-Oriented Selection & Prioritization

GOSP was proposed to overcome time and space complexity on SPL testing through the selection of most important features that are being used often and important from stakeholders perspective.

1) Context: The main goal of GOSP is to achieve higher error coverage by testing fewer test cases. To achieve this goal, GOSP is applied to both parts of engineering with input from graphical representation (feature model) which shows the inter-relationship between features. The feature model will undergo feature selection (removal of less important features) based on the domain stakeholders' goals and objectives. The output of the prioritization is test cases arranged based on their importance with the most important test cases will be placed highest.

2) Content: GOSP concentrates on system testing level and using the pre-configured algorithm in their technique. GOSP also focuses on small-scale feature model and does not involve T-wise coverage. Prioritization that occurred in this technique is through calculation using weight matrix in order to determine the order of test cases. The prioritization process is highly influenced by the goal of the stakeholders; some features are going to be ranked higher based on the stakeholders' objectives. The tools used by GOSP are fmp2rms plugin tool and Rational Software Modeler (RSM). Limitation of GOSP is it does not be tested in more than one case study, which brings to question the viability of GOSP in different feature models.

3) Validation: The validation of this proposed technique is done using E-shop case study which is done in academia setting and categorized as toy-example.

B. Weight Modelling Prioritization (WMP)

WMP is proposed to effectively apply combinatorial interaction testing to an industrial product line into the market by modelling weight sub-product lines which are synchronized with domain experts' goals. The technique will generate covering arrays by prioritizing interactions according to their weights then the generation of product will be conducted mimic to the products in the market which covered as many interactions as possible which are assumed as the most relevant with the market release.

1) Context: WMP is applied to both parts of engineering (domain and application engineering). Its specific input method used is a TOMRA's product line of reverse vending machines, and the output from the prioritization technique will be ranked based on a number of important interaction which resembles product in the market. The central element this technique will be focusing on the most important feature interactions for market release.

2) Content: WMP is a system testing that uses covering array generation algorithm for Product Lines (ICPL), a specific algorithm developed for large feature models. T-wise coverage for this technique is done from T1 to T3 using CoC prioritization criteria combined with domain stakeholder's objectives through the incremental evolution of the test products for continually changing market situation. WMP is done using TOMRA Verilab. A limitation of this technique is it might be limited to TOMRA reverse vending machine case study because of the assumption that all market segments will be the same.

3) Validation: WMP technique is validated using TOMRA vending machine case study which is considered as an industrial scale case study. The evaluation is conducted in a test lab provided by TOMRA.

C. Similarity Heuristic Prioritization (SHP)

SHP is proposed to overcome the combinatorial explosion issue which largely comes from the large feature model. The researchers stated that most CIT approaches fail to solve T-wise coverage of more than three and proposed an idea to use TCP to relax the T-wise criterion. They also claimed that their method is effective and scalable.

1) Context: SHP testing is done on both domain and application engineering phase, and their method input starts with feature model. The output of the prioritization will be test cases with the highest summation of distance between test cases. The central element of focus is on the need of practical solutions to test a large number of SPL.

2) Content: SHP testing level is a system testing as it involves a complete test of the system and does not use any sampling algorithm. SHP uses SAT solver to produce valid configurations. The feature model will be converted into a Boolean formula, generating a valid configuration. SHP is also able to solve feature model with T-wise coverage higher than T=3 until T=6, and this technique uses dissimilarity criteria which is similarity heuristic. The tools used in this technique are SPLAR and SAT solver Sat4j. Limitation of this technique is some of the existing tools provide faster speed in configuration generation.

3) Validation: Validation of SHP is done using APFD method using 114 feature models which are divided into two categories, 100 small to medium size and 4 large size feature models. Only 13 of the feature models are real-world feature models with 3 of them are large feature model application.

D. Delta-oriented Test Case Prioritization (DOP)

DOP is proposed to efficiently conduct integration testing approach for SPL based on delta modelling. Delta modelling is used to the model variant rich system and helps to show the differences between a variant of the product in deltas. DOP focuses on structural changes between products and identifies changed parts, where only the changed part will be retested.

1) Context: DOP is done in both SPL phases (domain and architecture engineering), and their method input is architectural models which will be used to produce regression deltas between products under test. This can be achieved by prioritizing the most changed test as the highest priority and considered as the most important test cases in the test suite.

2) Content: DOP is an integration testing since it considers internal part of the product and does not use any sampling algorithm. DOP is based on component weight where the values of the component weight come from the number it has been used. The tool used is MSCDL which is directly embedded in Deltarx. The limitation of this product is it does not consider the influence of different orderings of product variants in its results.

3) V*alidation:* DOP is using Body Comfort System (BCS) case study which is an SPL vehicle. It comprises a total of 11,616 possible variants. The evaluation method used is APCC metric whereas the context of evaluation is done on academia setting and the scale of evaluation is toy-example.

E. Statistical Prioritization (SP)

SP is proposed to improve the prioritization technique taking into consideration behaviour of the product used in Discrete-Time Markov Chain (DTMC) and Feature Transition System (FTS). In this technique, Markov chain is used to extract configurations of interest according to the likelihood of their executions which will be stored in FTS.

1) Context: SP is done on both domain and application engineering. Its method input is a feature model, and output is a reuse test case. Its central element of focus is improving current existing CIT technique by considering the behaviour of the product. The test case will be reordered based on a number of times product behaviour have been used. The product with the most use will be placed highest.

2) Content: SP is a system testing that does not use sampling algorithm in its technique. This technique considers product behaviour in SPL testing, and its T-wise coverage is non-existent. SP prioritizes using FTS by ordering them based on the probability for it to happen. The tool used by this technique is a Phyton bot, a web crawler that systematically browses and records information about a website. This technique is highly dependent on the nature of the case study.

3) Validation: The validation process is done on Claroline case study which is an open-source web-based application used for academia setting, and the scale of evaluation is categorized as toy-example. The DFS algorithm is used and tested four times on Claroline DTMC to investigate the most frequent features in valid traces.

F. Similarity-Based Prioritization (SBP)

The goal of SBP is to increase interaction coverage between variants under test as fast as possible. The issue that this method tries to solve is a combinatorial explosion of SPL because exhaustive testing of each product is highly not feasible due to a large number of products.

1) Context: SBP focuses on domain engineering. Its method input is feature model, and output is a prioritized test case based on their dissimilarity value with the highest will be placed on top followed by lower values. The central element of focus is to overcome the combinatorial explosion which causes inefficient use of time and budget.

2) Content: SBP is a system testing and the sampling algorithm used in this technique is CASA, Chavtal, and ICPL. This technique's T-wise coverage is T≤3, and prioritization criteria used by this method is dissimilarity criteria with ratio 0-1 used to determine the similarity between two configurations. SPLCAT tool is used in this technique, and its limitation is assuming that the faults are equally distributed over the features and their interactions in SPL.

3) Validation: This method validation process is done using a mobile phone and smart home feature model because it is widely used in academic setting and its scale of evaluation is toy-example. The evaluation method is done using defect simulation under general idea that defects are generated based on the interaction between several features.

III. Results and Discussion

This discussion provides evaluation for the current approach in TCP. Tables 4, 5 and 6 show the comparative results in terms of context, content, and validation. Our study concludes that majority of the approaches use feature model to show feature and interaction between features in a graphical fashion. In quality attributes category, our study identified that only one existing approach is trying to solve scalability issue that is undeniably known as the most common issue in CIT, as shown by the lack of approach trying to solve T-wise interaction of T > 3 (Table 5). Our study is also able to identify the lack of trend in solving scalability issue and their technique focusing on getting better efficient quality attributes. In Table 5, our study discusses the content of each prioritization technique which is typically found in TCP. Moreover, our study identified the wide use of sampling algorithm in CIT in four of the six techniques, only two of techniques do not use sampling algorithm because their technique focused on integration level and SPL behavior in their TCP. Our study also noted on the wide use of sampling algorithm such as ICPL. Moreover, SHP mentioned about scalability issue in SPL, and how the lack of sampling algorithm in CIT is able to solve large feature model. They even admitted that their own sampling algorithm is not the fastest and suggest ICPL as the fastest sampling algorithm in the market. SHP's suggestion of using TCP as a promising approach to solve high T-wise criterion is because TCP only selects few of the most important test cases in the test suite to achieve desired criteria with low testing effort.

TABLE IV
Comparative Evaluation for Context Criteria

Elements	**GOSP**	**WMP**	**SHP**	**DOP**	**SP**	**SBP**
Goal	Effective testing for immediate market release	Effective testing for industrial line	Effective & scalable technique for large model	Efficient integration testing	Effective testing by considering SPL behaviour	Effective TCP technique for SPL
Domain Application	Both	Both	Both	Both	Both	Domain Engineering
Method Input	Feature model	Feature model	Feature model	Architecture model	Feature model	Feature model
Method Output	Prioritization based on importance of feature	Prioritization based on interaction in the product	Prioritized based on their maximum distance	Test case prioritized based on number of changed parts	Prioritization based on the probability of valid behaviour going to happen	Test case prioritized based on their dissimilarity value
Central Element Focus	Reduce time to market	Reduce time to market	Overcome scalability issue	Reduce redundant testing effort	Behaviour model in product	Reduce testing effort
Quality Attributes	Efficiency	Efficiency	Efficiency & scalability	Efficiency	Efficiency	Efficiency

TABLE V
Comparative Evaluation for Content Criteria

Elements	**GOSP**	**WMP**	**SHP**	**DOP**	**SP**	**SBP**
Testing level	System Testing	System Testing	System Testing	Integration Testing	System Testing	System Testing
Sampling algorithm	Pre-configuration algorithm	ICPL	SAT solver	None	None	CASA, Chvatal, and ICPL
T-wise coverage	No	T ≤ 3	T ≥ 3 to 6	No	No	T ≤ 3
Prioritization criteria	CTCR criteria	VC&CC criteria	Dissimilarity criteria	VC&CC criteria	None	Dissimilariy criteria
Tools used	-fmp2rms plugin -Rational Software Modeler (RSM)	-TOMRA Verilab	-SPLAR -SAT solver Sat4j	-MSCDL into Deltarx	-Phyton Bot	-SPLCATool embedded FeatureIDE
Limitation	Limited to one case study	Limited to one case study	Slow in generation configuration	No consideration on different orderings of product variants	Limited to one case study	Assumption of fault spread equally

TABLE VI
Comparative Evaluation for Validation Criteria

Elements	**GOSP**	**WMP**	**SHP**	**DOP**	**SP**	**SBP**
Application Domain	Smart Software factories	Smart Software factories	High Education	Smart Software factories	High Education	High Education
Evaluation method	None	None	APFD metric	APCC metric	None	Defect simulation
Scale of evaluation	Toy-example	Industrial	Down-scaled real world	Toy-example	Toy-example	Toy-example

Next evaluation of TCP in the content category is prioritization of test suite which shows various types of prioritization criteria being used by researchers as classified in [10]. We found out that VC&CC and Dissimilarity are the two most popular TCP techniques being used in the market. However, there is a discussion among researchers on which criteria is preferable to be used in TCP. For example, both SHP & SBP agreed that the most dissimilar test cases would generate more defects and more likely to cover a greater number of t-sets than two similar ones [3], [6]. In contrast, VC&CC focuses on stakeholder desire, which contributes to various testings performed that exhaust the testing effort.

Moreover, VC&CC does not consider the influence of different orderings of product variants. Therefore, minimization of testing effort can be better reduced using dissimilarity technique compared to VC&CC. In the tools used category, most researchers integrate several tools to be used in TCP techniques, and we found that there is no preferable tool for generation and prioritization. Therefore, any suitable tool can be integrated for TCP technique. Finally, it is noted that only one case study is being done to test an SPL model. This is because it is better to test an SPL model using several case studies using different SPL feature models.

SBP also mentioned that their technique's weakness is that they assume that most of the fault is spread equally among configuration which is not the case in real world SPL products. Our validation category consists of the last phase which is an evaluation of effectiveness. This phase helps us to investigate the benchmark of these TCP techniques. In this category, our study identified that researchers used experiments to prove their technique's efficiency. However, some of the researchers only discussed on their results obtained without proving any objective. Therefore, the experiment is perceived as an easy and objective way to evaluate an SPL technique's effectiveness. In the scale of evaluation category, three of the methods used toy-example to validate their techniques. Only one of the techniques evaluate their effectiveness using the industrial environment.

Lastly, based on the comparison of these three tables, our study investigates the maturity of TCP techniques in SPL testing in terms of three typically used phases which is the generation of configuration, prioritization of test suite, and evaluation of the effectiveness. The generation of configuration can be further improved because there are not many techniques utilized to solve scalability issue. In the prioritization of test suite phase, the number of different tools used indicates that there is significant lack of standardized tools used by researchers which is an open area to be investigated. Moreover, to examine the effectiveness of their technique, APFC and APCC are used as benchmark method by most researchers this indicates that this is standardized (matured) way to evaluate TCP technique in the SPL.

IV. CONCLUSION

From the output of the review, our study proposed a complete TCP framework based on three phases Typically TCP in SPL testing is divided into three parts which are generation of configuration, prioritization of test suite and evaluation of effectiveness. The important components of these phases have been analyzed earlier in this paper and based on that; we picked the most suitable methods or tools to be used in our framework.

DBPF starts with generation of configuration that involves extracting valid configuration from a number of possible configuration through CIT using SPLCATool integrated into Eclipse platform. The integrated tool consists of current sampling algorithms in the market such as ICPL, CASA, Chavtal, and SBP. The result of this process will produce a set of valid combination of features from Software Product Lines Online Tools (SPLOT). SPLOT is a repository for hundreds of feature model in SPL. Next, the configuration will be placed in the test suite and undergoes the prioritization of test suite process using existing tool from Sanchez *et al.* [10].

Our approach enhances existing tools, and we also propose our own prioritization technique using dissimilarity criteria. The test suite will be processed, resulting to a ranked test case based on their dissimilarity value. Lastly, the ranked test cases will undergo effectiveness evaluation using Average Percentage Fault Detected (APFD) metric. The APFD metric is often used in regression testing approaches such as test case prioritization to investigate the technique's effectiveness in terms of fault detection rate. As an evaluation method, APFD metric will be carried out on two case studies, Mobile Phone, and Educational Robotics. Mobile Phone is chosen as the benchmark case study because it is commonly used by SPL researchers, while Educational Robotics is used because we need to apply DBPF method into real-world application. The result of APFD metric will help in improving effectiveness testing to SPL products thus contributes to the minimum amount of resources needed in SPL testing phase.

In this paper, evaluation was divided into three parts namely context, content and validation. Furthermore, our survey also analyzed and discussed the comparison to answer our research questions. Our study concludes that TCP techniques in the market currently vary according to their specific contexts, and the most frequent type of prioritization criteria used in the market is VC&CC and Dissimilarity. Between these two criteria, Dissimilarity is more preferred because it offers promising results as shown in the literature, plus it also tackles scalability, a tough issue in TCP for SPL. This issue is also known to be the bottleneck for some techniques. Dissimilarity prioritization is suitable companies with the low testing budget. The discussion also noted that most researchers focused on reducing testing effort, which is also the main reason for the formulation of TCP technique. Moreover, there is an increasing trend in CIT testing using sampling algorithm, but there is a notable lack of sampling algorithm to be used to generate the needed configuration. Some researchers viewed ICPL as the best sampling algorithm can be used in this research.

Finally, our study concludes that most of the existing works are focusing on academic environment and is not fully tested in an industrial environment. For future works, we will conduct our own experiment based on the proposed DBPF framework to provide a structured approach for each TCP phase to gain a better and more effective SPL testing process. As, future work, Our dissimilarity prioritization technique will be applied to the Mobile Phone and Educational Robotic (industrial) case studies and their effectiveness will be evaluated and compared with other works.

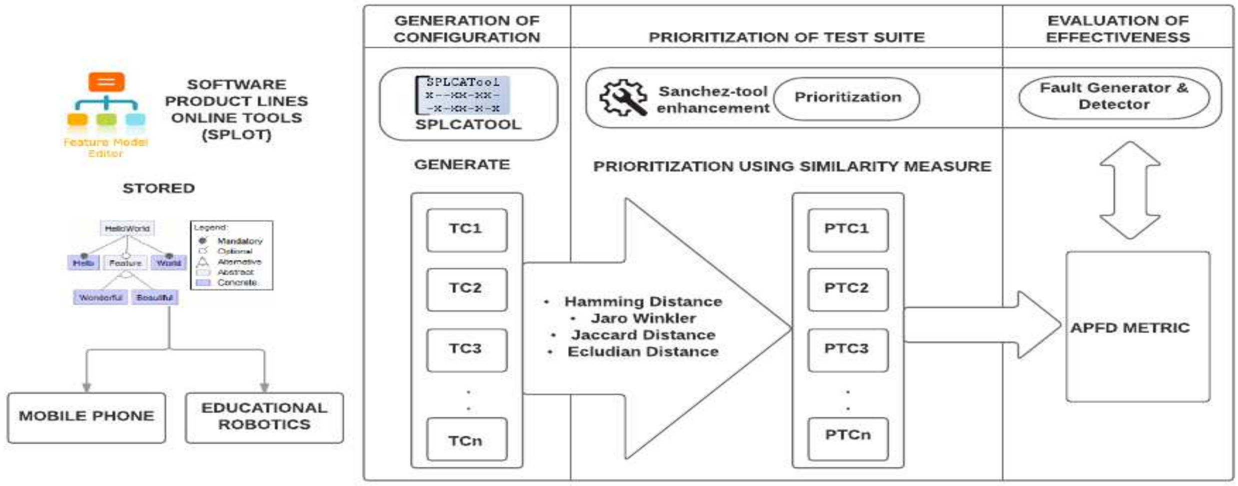

Fig. 2 Overview of DBPF

ACKNOWLEDGEMENT

We are grateful for the UTM scholarship to Author 1. Furthermore, we express our gratitude profoundly to Research University Grant (GUP), Universiti Teknologi Malaysia (UTM) under Cost Centre No Q.J130000.2528.15H44, for their financial support. Our profound appreciation also goes to ERetSEL lab members for their continuous support in the working of this paper.

REFERENCES

[1] Devroey, X., Perrouin, G., Cordy, M., Schobbens, P.Y., Legay, A. and Heymans, P., “Towards statistical prioritization for software product lines testing”. In Proceedings of the Eighth International Workshop on Variability Modelling of Software-Intensive Systems - VaMoS ’14, New York, USA, 2013 ,pp. 1–7. ACM Press.

[2] Ensan, A., Bagheri, E., Asadi, M., Gasevic, D. and Biletskiy, Y., “Goal-oriented test case selection and prioritization for product line feature models” in *Proceedings - 2011 8th International Conference on Information Technology: New Generations, ITNG 2011*, 291–298.

[3] Henard, C., Papadakis, M., Perrouin, G., Klein, J., Heymans, P. and Le Traon, Y., "Bypassing the Combinatorial Explosion: Using Similarity to Generate and Prioritize T-Wise Test Configurations for Software Product Lines," in *IEEE Transactions on Software Engineering*, vol. 40, no. 7, pp. 650-670, July 1 2014.

[4] Johansen, M.F., Haugen, Ø., Fleurey, F., Eldegard, A.G. and Syversen, T., “Generating better partial covering arrays by modeling weights on sub-product lines” Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 7590 LNCS, 269–284. http://doi.org/10.1007/978-3-642-33666-9_18

[5] Lachmann, R., Lity, S., Lischke, S., Beddig, S., Schulze, S. and Schaefer, I., “Delta-oriented test case prioritization for integration testing of software product lines”. In *Proceedings of the 19th International Conference on Software Product Line - SPLC ’15*, New York, USA 2014, pp. 81–90. ACM Press.

[6] Al-Hajjaji, M., Thüm, T., Meinicke, J., Lochau, M. and Saake, G., "Similarity-based prioritization in software product-line testing." In *Proceedings of the 18th International Software Product Line Conference-Volume 1*, pp. 197-206. ACM, 2014.

[7] Nimal Jayaratna. *Understanding and Evaluating Methodologies: Nimsad, a Systematic Framework*. New York, NY, USA, McGraw-Hill, Inc., 1994

[8] Bielkowicz, P., Patel, P. and Tun, “Evaluating Information Systems Development Methods: A New Framework,” In *Proceedings of the 8th International Conference on Object-Oriented. Information Systems* (OOIS '02), Zohra Bellahsene, Dilip Patel, and Colette Rolland (Eds.). Springer-Verlag, London, UK, 2002, pp. 311-322.

[9] Lee, J., Kang, S. and Lee, D., "A survey on software product line testing." In *Proceedings of the 16th International Software Product Line Conference-Volume 1*, pp. 31-40. ACM, 2012.

[10] Sánchez, A.B., Segura, S. and Ruiz-Cortés, A., "A Comparison of Test Case Prioritization Criteria for Software Product Lines," *2014 IEEE Seventh International Conference on Software Testing, Verification and Validation*, Cleveland, OH, 2014, pp. 41-50.

[11] Yoo, Shin, and Mark Harman. "Regression testing minimization, selection and prioritization: a survey." *Software Testing, Verification and Reliability* 22, no. 2 (2012): 67-120.

[12] Ahnassay, Alvin, Ebrahim Bagheri, and Dragan Gasevic. Empirical evaluation in software product line engineering. Tech. Rep. TR-LS3-130084R4T, Laboratory for Systems, Software and Semantics, Ryerson University, 2013

[13] Ahnassay, Alvin, Ebrahim Bagheri, and Dragan Gasevic. *Empirical evaluation in software product line engineering*. Tech. Rep. TR-LS3-130084R4T, Laboratory for Systems, Software and Semantics, Ryerson University, 2013.

[14] Engström, E. and Runeson, P., "Software product line testing–a systematic mapping study." *Information and Software Technology* 53, no. 1, 2011, pp. 2-13.

[15] Neto, P.A.D.M.S., do Carmo Machado, I., McGregor, J.D., De Almeida, E.S. and de Lemos Meira, S.R., "A systematic mapping study of software product lines testing." *Information and Software Technology* 53, no. 5, 2011, pp. 407-423.

[16] Machado *et al.*, "On strategies for testing software product lines: A systematic literature review." *Information and Software Technology* 56, no. 10, 2014, pp. 1183-1199.

[17] Lamancha *et al.*, "Systematic review on software product line testing." In *International Conference on Software and Data Technologies*, Springer Berlin Heidelberg, pp. 58-71. 2010.

[18] do Carmo Machado, I., McGregor, J.D. and Santana de Almeida, E., "Strategies for testing products in software product lines." *ACM SIGSOFT Software Engineering Notes* 37, no. 6 2012, pp. 1-8.

[19] Jin-Hua, L., Qiong, L. and Jing, L., "The w-model for testing software product lines." In *Computer Science and Computational Technology, 2008. ISCSCT'08. International Symposium on*, vol. 1, 2008, pp. 690-693.

[20] Thüm, T., Apel, S., Kästner, C., Schaefer, I. and Saake, G., "A classification and survey of analysis strategies for software product lines." *ACM Computing Surveys (CSUR)* 47, no. 1, 2014, pp 6.

[21] Lopez-Herrejon, R.E., Fischer, S., Ramler, R. and Egyed, A., 2015"A first systematic mapping study on combinatorial interaction testing for software product lines," *2015 IEEE Eighth International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*, Graz, 2015, pp. 1-10.

[22] Mansyur and - Muliana,"Detecting Differential Item Functioning and Differential Test Functioning on Math School Final-exam," *International Journal on Advanced Science, Engineering and Information Technology*, vol. 6, no. 4, pp. 437-440, 2016. [Online].

[23] Rami Hasan Al-Ta’ani and Rozilawati Razali,"A Framework for Requirements Prioritisation Process in an Agile Software Development Environment: Empirical Study," *International Journal on Advanced Science, Engineering and Information Technology*, vol. 6, no. 6, pp. 846-856, 2016. [Online].