An experimental design on the SPEM 2.0 process model element classification algorithm of the AVISPA tool through ANOVA variance analysis

Un diseño experimental sobre el algoritmo de clasificación de elementos de modelos de proceso SPEM 2.0 de la herramienta AVISPA por medio del análisis de varianza ANOVA

Um delineamento experimental no algoritmo de classificação de Elementos de modelo de processo do SPEM 2.0 da ferramenta AVISO através da análise de variância ANOVA

Jhon Jairo Álvarez Londoño[1]
Julio Ariel Hurtado Alegría[2]

How to cite this article:
J. J. Álvarez Londoño, J. A. Hurtado Alegría, "An experimental design on the SPEM 2.0 process model element classification algorithm of the AVISPA tool through ANOVA variance analysis," *Revista Ingeniería Solidaria*, vol. 16, no. 1, 2020. doi: hhttps://doi.org/10.16925/2357-6014.2020.01.09

Artículo de investigación. https://doi.org/10.16925/2357-6014.2020.01.09

[1] MsC. Resercher. Departamento de Sistemas. Facultad de Ingeniería Electrónica y Telecomunicaciones. Universidad del Cauca, Popayán, Colombia

E-mail: jjalvarezl@unicauca.edu.co

ORCID: https://orcid.org/0000-0002-5088-9229

[2] PhD. Professor. Departamento de Sistemas. Facultad de Ingeniería Electrónica y Telecomunicaciones. Universidad del Cauca, Popayán, Colombia

E-mail: ahurtado@unicauca.edu.co

ORCID: https://orcid.org/0000-0002-2508-0962

Abstract

Introduction: This publication is the product of research: "SPEM 2.0 Process Model Metrics in the Reliability of its Visual Analysis" throughout 2019, which supports the work of a master's degree in Systems Engineering at the University of Cauca.

Objective: Rebase a process model metrics set in order to increase AVISPA reliability to support the visual analysis of SPEM 2.0 software process models.

Methodology: In order to improve AVISPA, a systematic literature review had been performed to find software process model metrics that are potentially apt to be represented in AVISPA. Next, a set of assessments were performed in order to enhance visual analysis tool. Finally, an ANOVA statistical assessment was realized in order to find a variance differential between AVISPA versions by comparing their F1-Score process model elements classification values.

Results: AVISPA significantly improved its general classification algorithm. Most of errors were found in SPEM 2.0 variability resolution feature and collections with duplicated elements. Multiple misclassifications still persists.

Conclusion: General AVISPA process model elements classification is improved. However, some process model samples remain scattered according to ANOVA results.

Originality: AVISPA is a recent solution for SPEM 2.0 software process model assessment. It's recent emergence carried to a lack of articles about software process model metrics and few works about AVISPA improvements. These are the main contributions of this paper.

Limitations: The project has been widely expensive in terms of execution time, traceability with all software process model elements, and mainly to find experts in software process that can meet the research requirements.

Keywords: Software Engineering, Process Validation, Process Documentation, Process Visualization, Process Model Metrics, AVISPA.

Resumen

Introducción: Esta publicación es producto de la investigación: "Métricas de Modelos de Proceso SPEM 2.0 en la Confiabilidad de su Análisis Visual" a lo largo del año 2019, que respalda el trabajo de una maestría en Ingeniería de Sistemas en la Universidad del Cauca.

Objetivo: Redefinir un grupo de métricas de modelos de proceso para incrementar la confiabilidad de AVISPA bajo el análisis visual de modelos de proceso SPEM 2.0.

Metodología: Se realizó una revisión sistemática de la literatura de métricas de modelo de proceso de software potencialmente aptas para ser representadas en AVISPA. Después, se realizó un conjunto de evaluaciones para mejorar la herramienta. Finalmente, se realizó una evaluación con ANOVA para encontrar un diferencial de varianza entre versiones de AVISPA al comparar sus valores F1-Score de clasificación de elementos de proceso.

Resultados: AVISPA mejoró significativamente su algoritmo de clasificación general. La mayoría de los errores se encontraron en la resolución de variabilidad SPEM 2.0 y colecciones con duplicados. Múltiples clasificaciones erróneas aún persisten.

Conclusión: Se ha mejorado la clasificación general de los elementos del modelo de proceso AVISPA. Sin embargo, algunas muestras de modelo de proceso permanecen dispersas según los resultados de ANOVA.

Originalidad: La reciente aparición de AVISPA ha conllevado a la falta de artículos sobre métricas del modelo de proceso de software y pocos trabajos sobre las mejoras de AVISPA. Estas son las principales contribuciones de este trabajo.

Limitaciones: El proyecto ha conllevado un alto costo en tiempo de ejecución, en trazabilidad con todos los elementos de los modelos de proceso, y principalmente en encontrar expertos en proceso software que cumplan con los requisitos de investigación.

Palabras claves: Ingeniería de Software, Validación de Procesos, Documentación de Procesos, Visualización de Procesos, Métricas de Modelos de Proceso, AVISPA.

Resumo

Introdução: Esta publicação é um produto da pesquisa: "Métricas do Modelo de Processo SPEM 2.0 na Confiabilidade de sua Análise Visual" ao longo do ano de 2019, que apóia o trabalho de um Mestrado em Engenharia de Sistemas da Universidade de Cauca.

Objetivo: Redefinir um grupo de métricas de modelo de processo para aumentar a confiabilidade do AVISPA sob a análise visual dos modelos de processo SPEM 2.0.

Metodologia: Foi realizada uma revisão sistemática das métricas do modelo de processo de software potencialmente adequadas para representação na AVISPA. Posteriormente, um conjunto de avaliações foi realizado para aprimorar a ferramenta. Finalmente, foi realizada uma avaliação com a ANOVA para encontrar um diferencial de variância entre as versões do AVISPA ao comparar os valores do F1-Score da classificação dos elementos do processo.

Resultados: AVISPA melhorou significativamente seu algoritmo de classificação geral. A maioria dos erros foi encontrada na resolução da variabilidade do SPEM 2.0 e nas coleções com duplicatas. Múltiplas classificações errôneas ainda persistem.

Conclusão: A classificação geral dos elementos do modelo de processo AVISPA foi aprimorada. No entanto, algumas amostras do modelo de processo permanecem dispersas de acordo com os resultados da ANOVA.

Originalidade: A recente aparição da AVISPA levou à falta de artigos sobre métricas do modelo de processo de software e poucos trabalhos sobre as melhorias da AVISPA. Estas são as principais contribuições deste trabalho.

Limitações: O projeto acarretou um alto custo em tempo de execução, rastreabilidade com todos os elementos dos modelos de processos e principalmente na procura de especialistas em processos de software que atendam aos requisitos de pesquisa.

Palavras-chave: Engenharia de Software, Validação de Processo, Documentação de Processo, Visualização de Processo, Métricas de Modelo de Processo, AVISPA

1. INTRODUCTION

"A process is a set of partially ordered steps, with related artifacts, human, economic and computerized resources, organizational structures and restrictions, whose objective is the production and maintenance of software" [1]. A well-defined software process model is a key aspect for achieving quality in products and productivity in teams [2]. SPEM 2.0 is a language for defining software process models accepted by the software industry for defining process models and various configurations for adapting process models to different organizational or project contexts [3]. AVISPA

(Analysis and VIsualization for Software Process Assessment) [4] is a tool that produces visual polymetric views [5] represented by rectangular geometric figures whose dimensions are based on metrics calculated from elements of a software process model. These representations are known as blueprints. AVISPA visually highlights on its blueprints, a subset of process elements with possible errors [6] known as an error patterns. AVISPA supports variability resolution according to the rules set in SPEM 2.0 for process models that contain it [7]. Several reports in the industry have accepted and implemented this approach to visual analysis [4], [6]–[10], according to the results, the tool presents some blueprints and error issues affecting its reliability. The metrics for software process models involve a great effort in their definition and, subsequently, their validation of effectiveness and relevance for a software process model. Metrics also allow AVISPA to improve drastically process elements classification. In this work, we present the AVISPA metrics rebasement based on several research results such as (i) a systematic literature review obtaining the most reliable metrics for software process models / SPEM 2.0 models for AVISPA refactoring in section II (ii) The AVISPA tool and refactoring by the lessons learned from M. Bastarrica et. al. work [9] (iii) an AVISPA understandability experimental and exploratory assessment (iv) some Test Driven Development (TTD) techniques asynchronously done such as unit testing, black and white-box testing and minimal examples assessment. A new version of AVISPA was released and evaluated by a one-way ANOVA test assessment against AVISPA's previous version.

1.1. Systematic review and research background

The systematic review was performed by following the search parameters in table 1.

Table 1. Search parameters

Parameters	Values
Academic databases	- Scopus - Engineering Village - Science Direct - IEEExplore - ACM Digital Lirary - Springer
No academic database	- Google - Academia.edu - ResearchGate - SEI web page.

(continúa)

(viene)

Parameters	Values
Article types	- Journals - Workshops - Conference articles - Conference contributors from industry and professionals - No academic online publications
Search applied to	- Title - Abstract - Keyords
Languages	English and Spanish
Time window	2014 to 2019

Source: Own elaboration

Table 2 illustrates the query string executed to retrieve articles:

Table 2. Query string

Principal terms	Alternative terms
Software Process Metamodels and Tools	(SPEM OR AVISPA OR EPF OR (Eclipse Process Framework) OR (Polimetric View*) OR (Software Process Model*)) AND
Software Process Metrics Validation	(Metri* OR Measur* OR (Metrics Scope) OR (Metrics Validation) OR (Metrics Set) OR Confiab* OR Valid*)

Source: Own elaboration

Two main inclusion criteria for each assessed article had been proposed as follows:

- **Software process model metrics (C1):** The study mentioned any software process model metric that could be potentially represented from SPEM 2.0?

- **Software process model metric assessments (C2):** The study realize an adequate assessment about software process metrics that could be instantiated in SPEM 2.0 context?

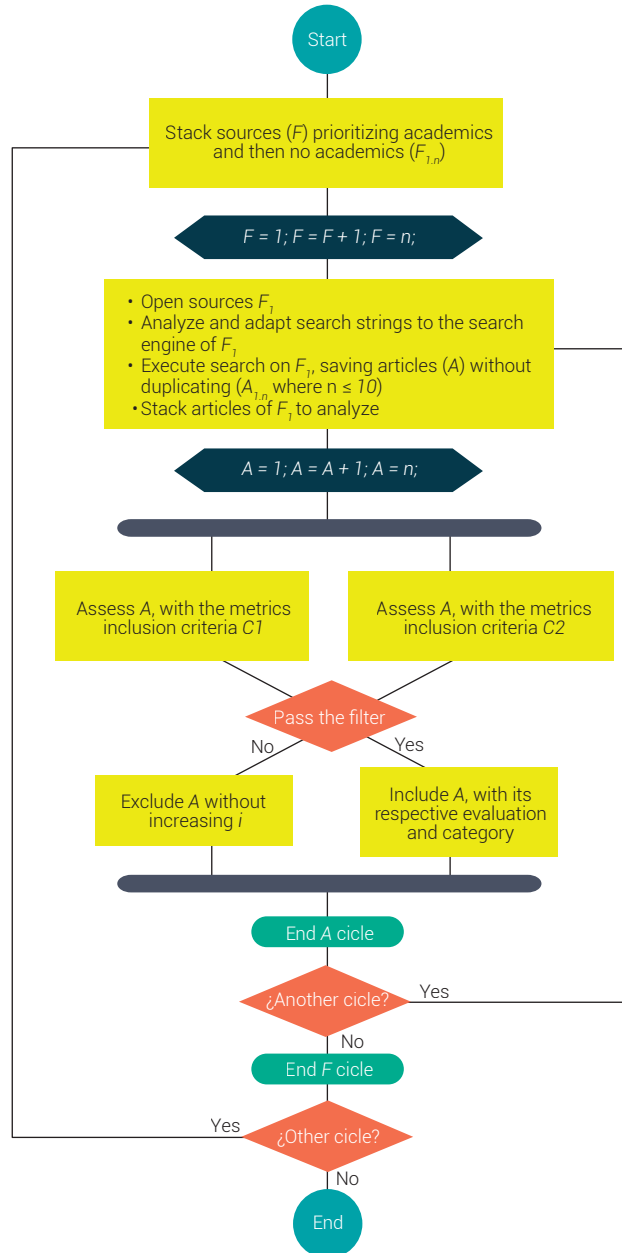Based on above, the literature review process has been stablished in figure 1

Start

Stack sources (F) prioritizing academics and then no academics ($F_{1,n}$)

$F = 1; F = F + 1; F = n;$

- Open sources F_1
- Analyze and adapt search strings to the search engine of F_1
- Execute search on F_1, saving articles (A) without duplicating ($A_{1,n}$ where n ≤ 10)
- Stack articles of F_1 to analyze

$A = 1; A = A + 1; A = n;$

Assess A, with the metrics inclusion criteria $C1$

Assess A, with the metrics inclusion criteria $C2$

Pass the filter

No

Yes

Exclude A without increasing i

Include A, with its respective evaluation and category

End A cicle

¿Another cicle? Yes

No

End F cicle

Yes ¿Other cicle?

No

End

Figure 1. Article process filter for literature review
Source: Own work

Table 3. References found in the literature systematic review

Work	Score		Decision	Category
	C1	C2		
First iteration				
K. Kluza et al. [23]	5.0	5.0	Accept	C1
A. Burattin [24]	3.0	5.0	Accept	C1 & C2
K. Kluza [25]	3.0	5.0	Accept	C1 & C2
K. Figl et al. [26]	0.0	5.0	Accept	C2
J. Pavlicek et al. [27]	3.0	5.0	Accept	C1 & C2
J. Pflug et al. [28]	0.0	5.0	Accept	C2
del Rio Ortega et al. [29]	5.0	0.0	Accept	C1
J. Geisel et al. [30]	5.0	0.0	Accept	C1
M. Smatti et al. [31]	2.0	4.0	Accept	C2
G. Kaur et al. [32]	5.0	0.0	Accept	C1
M. Marques et al [33]	2.0	0.0	Reject	No applies
J. Alvarez et al [34]	1.0	2.0	Reject	No applies
J. Kim [35]	0.0	1.0	Reject	No applies
E. Nogueira et. al. [36]	2.0	0.0	Reject	No applies
T. Mens et. al. [37]	1.0	2.0	Reject	No applies

Source: Own work

1.1.1. Systematic review results

Figure 2 illustrates all metrics that accomplish with SPEM 2.0 process models elaboration.

Some AVISPA changes consisted in support and deprecate metrics that does not accomplish with systematic review metric results in order to increase AVISPA reliability.
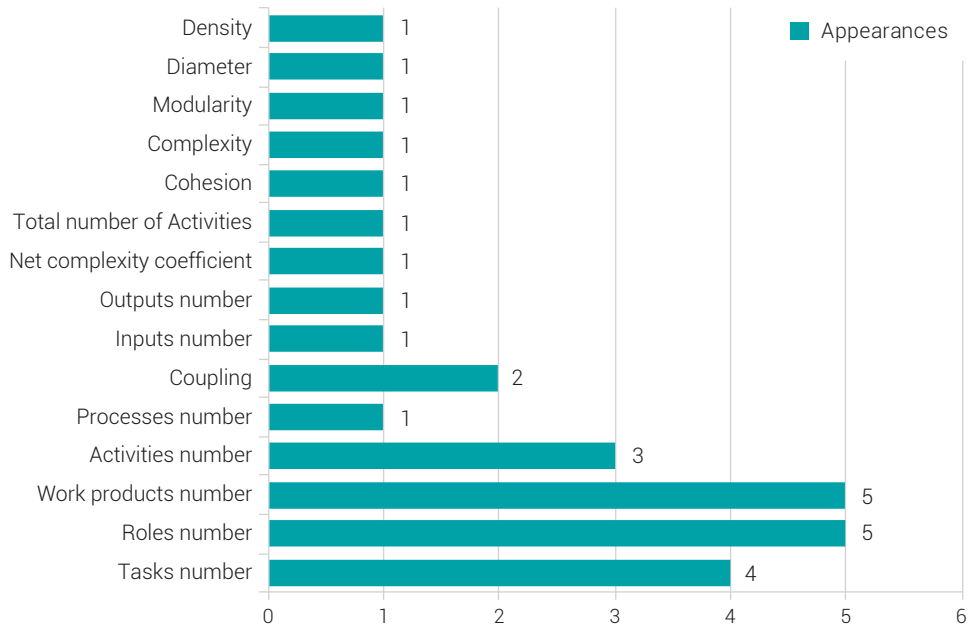
Figure 2. Metrics per article
Source: Own work

1.1.2. Software process models metrics

Multiple reported works establish some process model metrics. By example, Canfora et. al. [11] define a metrics set in order to assess their relation with process models understandability. Also, they assess the process model modification time when metric was understood by the experiment assessed subjects. Results shows that most of assessed software process model metrics had a significant correlation with understandability and modifiability.

Rolón et. al. [12] realizes a parallel between previous software process models metrics of Canfora et. al. [11], and BPMN process models in order to apply software process metrics. Results shows metrics categorized in the model scope and BPMN scope metrics.

These results are pertinent to this work in order to get a software process model metrics set to be assessed accomplishing three main conditions around them: (i) must be supported by SPEM 2.0, (ii) must be supported by EPF Composer and (iii) must be a relevant metric to software organizations.

In order to offer support to the coupling factor new metric as mentioned in Canfora et. al. work [11]. Two new blueprints were added to AVISPA (Activities Coupling Factor by Contained Activities and Activities Coupling Factor by Contained Tasks).

1.1.3. AVISPA

The process model blueprints are "visual representations for analyzing different perspectives of software process model" [6]. AVISPA is a tool that builds software process models blueprints and highlights possible process models recurring errors called error patterns. The AVISPA generation process is shown as figure 3.
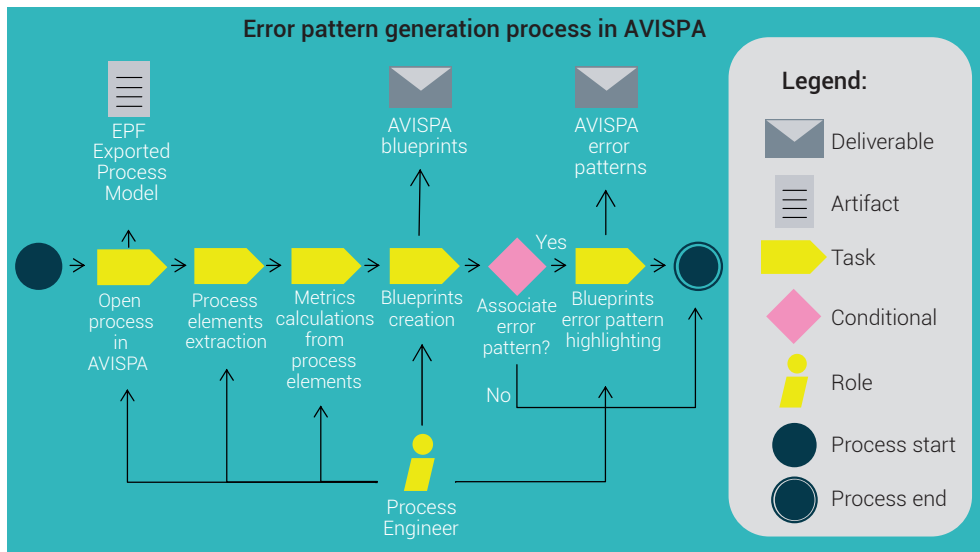

Figure 3. AVISPA blueprints / error patterns generation process
Source: Own work

Anyway some works indicates that AVISPA presents some misclassifications about blueprints generation. M. Bastarrica et. al. [9] realized an AVISPA assessment to a SPEM 2.0 software process model group in order to find problems in software process adoption in relation with software process model formalization level. They define a set of hypothesis to be discarded in order to assess the process model formalization and its relation with software project execution delays. This work established some AVISPA modifications in order to implement improvements and modifications in their error pattern.

Due to above, six hypotheses were defined and validated with their respective AVISPA refactoring decisions:

- **H1 and H2**: These hypotheses are related to responsible roles in charge of work products. Specifically, no responsible role and two or more responsible roles respectively. However, process execution delays are materialized too few times. So, the responsible roles blueprints were deprecated. However,

responsible roles are supported in union with the overloaded work products new blueprint of H3, in order to highlight the responsible roles in AVISPA's work product nodes.

- **H3**: This hypothesis establish that too many assigned roles to a single work product could represent process execution delays. In order to consider that a work product is overloaded, the threshold consists of 4 or more roles associated in an artifact. As mentioned above, the overloaded work products blueprint was created as consequence of this hypothesis result.
- **H4**: Indicates that an overloaded role responsible for a work product may represent process execution interruptions. The overloaded role blueprint is edited so it also indicates the roles responsible of one or more artifacts.
- **H5**: This hypothesis establish that work products without template could represent process execution delays, but this happens only one time. Thus, the AVISPA refactor decision was the respective error pattern deletion.
- **H6**: The hypothesis establish that waste work products could represent process execution delays. Anyway, most of them are false positives because the waste work product blueprints do not show any difference between an artifact and a deliverable. This hypothesis has a weak relationship with the projects delays, so deliverables are excluded from this error pattern.

J. Salgado et. al. [7] improved AVISPA to support SPEM 2.0 variability resolution. Results show a high improvement in software process models classification in tasks blueprint. Nevertheless, SPEM 2.0 also support process model variability in artifacts and roles. In this work, we perform development in AVISPA for supporting the variability of the three kinds of SPEM 2.0 process model elements. Due to this extension, some AVISPA variability resolution undesired behaviors were identified:

- Contributes variability type was no correctly visualized by the Task Variability blueprint
- The previous AVISPA version had errors related to the resolution order according to the OMG SPEM 2.0 standard

As a result of this research process step the Artifact variability blueprint and the Role Variability blueprint were added and the errors were fixed.

1.1.4. AVISPA code improvement techniques

Several code improvements techniques and protocols have been used for AVISPA refactoring. The think-aloud method [13] has been used in order to retrieve software user experience, AVISPA was assessed with this methodology as an exploratory research of tool inner construction, code concepts and architectural decisions. Also, allows to retrieve the software process models experts' knowledge and opinion.

The Khan et. al. work [14] realizes a comparative between some techniques used in this work, the basis path testing "allows the test case designer to produce a logical complexity measure of procedural design and then uses this measure as an approach for outlining a basic set of execution paths."

The TDD software development approach was selected to realize the new AVISPA functionalities. According to R. Martin et. al. [15] by writing tests first, TDD offered an agile and concrete path to follow, centering codification in the expected functions behaviors with the necessary code to accomplish the objective. This means, in AVISPA's context, that metrics were improved as well as expected behaviors does.

Finally, for the AVISPA variability resolution support, the use cases testing seems to be an ideal technique for assess highly complex business processes such as SPEM 2.0 process model variability rules. The Allala et. al. work [16] was used as a reference framework for transform user requirements to test cases implementing an adapted use case template from Cockburn et. al. work [17].

1.2. AVISPA validation

In the AVISPA context, metrics are highly difficult to understand, modify and control, they do not offer to much information about process model behavior, neither offer a context to be evaluated or an acceptable range as a reference framework.

Figure 4 illustrates all performed assessments distributed between two main AVISPA processing features. Process model consist about the exported XML process model data transformation into data structures ready to be transformed into visualizations, and AVISPA visualizations that realizes metrics, and decisions to draw each process model element and dependency.
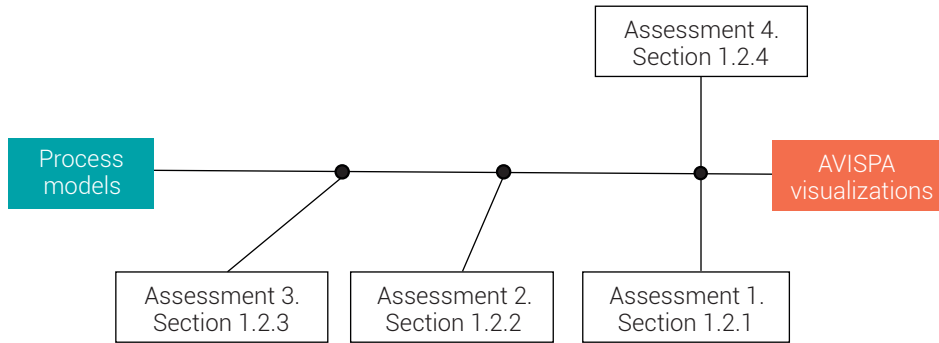
Figure 4. AVISPA blueprints / error patterns generation process
Source: Own work

1.2.1. Assessment 1: AVISPA understandability

The first AVISPA assessment was performed as an exploratory and conceptual tool construction. The research state at this moment consisted in a many visual improvements achieved in error patterns with few modifications about process model processing code.

The exploratory experiment consisted in a think-aloud methodology with surveys to 4 process engineer experts with different skills and knowledge. AVISPA was submitted for the expert assessment. A questions set must be answered in order to gather how easy was for them to interact with the tool and interpret process models possible problems with the generated blueprints / error patterns.

The answers and findings were summarized as follows and used for enhancing AVISPA tool:

- Gradient colors are not friendly for blueprints and error patterns constructions. Using concrete plain colors seems to be a better practice.
- Legends were not well defined at all, some misinterpretations were realized by experts.
- AVISPA tool aims to detect possible process model construction problems. Nevertheless, process engineer experts do not know as process models are organized and constructed.
- Showing process model elements by composition was better in order to understand the process.

1.2.2. Assessment 2: Basis path testing

A formal verification for some AVISPA components were performed in order to achieve an acceptable code behavior. The XML import process had some complex tasks and functions that contains important and long procedures to organize the imported software process model. To ensure its normal operation, a basis path white-box test was performed. Some improvements such as well classified process model elements, duplicated process model elements, and blueprint expected highlighting behavior, were included as a result of this test. This testing results were used for enhancing AVISPA tool.

1.2.3. Assessment 3: Unit testing

Unit testing was selected to cover one single error pattern that needs to be improved with all the new features: the Artifact Responsible Roles blueprint.

As said previously, AVISPA has two main process in order to blueprints production, the process elements extraction, and the error patterns visualizations.

These two process were assessed with this code testing technique.

1.2.4. Assessment 4: Use cases testing and variability support refactoring

SPEM 2.0 variability resolution is a highly complex business task. Each resolution type contains a rule set that are difficult to understand and reproduce.

The use case testing provides, in the AVISPA context, a real and concrete knowledge about all rules and the expected blueprint behavior.

The use case template from Cockburn et al. [17], was adapted to the AVISPA context. A use case specification was implemented by each variability type.

Finally, one of most significant AVISPA variability blueprints, the tasks variability blueprint, had been improved as figure 5 illustrates, which the above image is the previously AVISPA version.
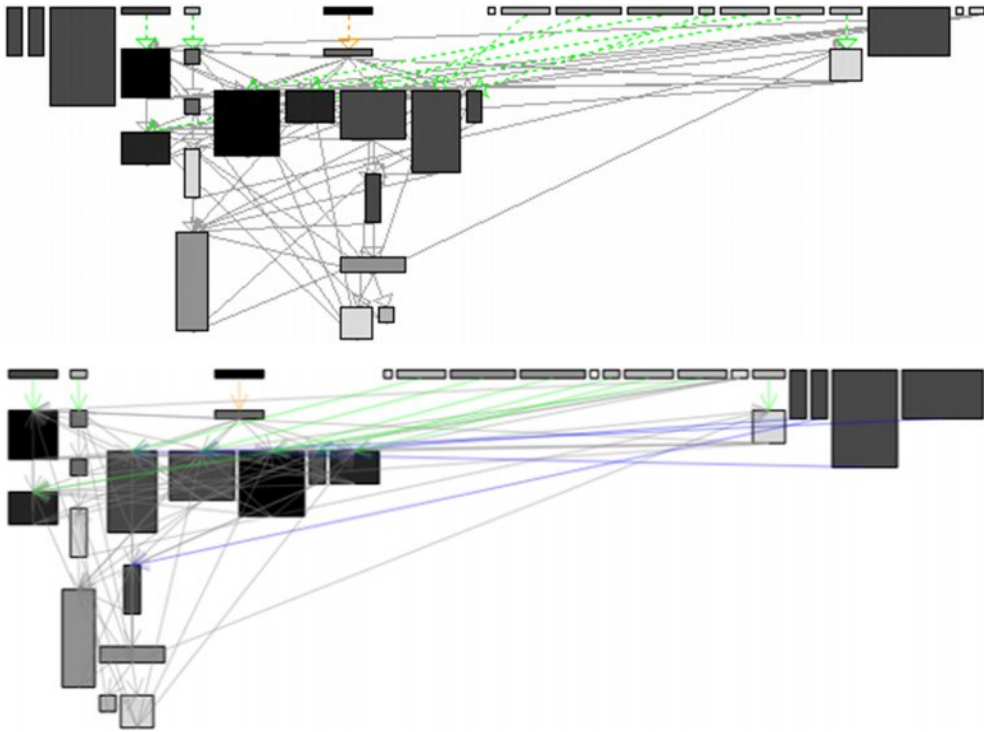
Figure 5. Tasks Variability Blueprint improvement.
Source: Own work

There are several improvements performed in figure 5 comparisons as listed below:

- **Every node is interconnected**. In business logic, this means that all task have or accomplish a business objective. The OpenUp process model is documented by Eclipse Foundation. Hence, is more probably that AVISPA can contain errors in their representations building process than OpenUp in their process model documentations. Thus, the main conclusion was to assume that AVISPA needed to be improved. Finally, these errors were detected and fixed.
- **SPEM 2.0 contribution variability type**. As result of the previous improvement, we detected that contribution variability type was not present inside the first version of this error pattern. We detected that contribution relations were already calculated but were not shown correctly in the error pattern.

2. MATERIALS AND METHOD

An AVISPA evaluation was implemented using an experimental design approach based on Pflenger et. al. work [18] in order to compare the differences between the original AVISPA and the new AVISPA version, which includes the improvements above mentioned. improved version. For this assessment, an ANOVA statistical technique test was performed.

As a binary classifier, AVISPA can organize any kind of process element in four categories: true positive (TP), true negative (TN), false positive (FP) and false negative (FN).

To assess a binary classifier, F_1Score (see equation 3) can measure its classification accuracy and is the harmonic average of the precision (see equation 1) and recall (see equation 2) and, at the same time, these two measures can be explained in terms of the four categories previously mentioned.

$$precision = \frac{|TP|}{|TP| + |FP|} \tag{1}$$

$$recall = \frac{|TP|}{|TP| + |FN|} \tag{2}$$

$$F_1Score = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{3}$$

Finally F1Score is a value between 0 and 1 and is the value that ANOVA consumes to accept or reject the null hypothesis.

All blueprints and error patterns of AVISPA previous version are contrasted against all blueprints and error patterns of AVISPA new version.

The software process models used to this assessment are:

PM-1. OpenUP
PM-2. All SPEM 2.0 variability types process model sample
PM-3. APF
PM-4. UP-VSE [26]
PM-5. Scrum
PM-6. XP
PM-7. Tutelkan
PM-8. Mobius

PM-9. Rhiscom

PM-10. BBR

PM-11. OpenUP - SPL Family

PM-12. EPF Practices

PM-13. DTS Process Model

PM-14. OpenUP / Model Driven Software Development - DSDM 1.0

PM-15. ISPW 6 [27]

PM-16. REuse and Migration of Legacy Applications to Interoperable Cloud Services -REMICS

PM-17. PRODIGIA [28]

PM-18. OpenUP / Basic

2.5.1. Hypothesis

- **Alternative Hypothesis H_1**: The information generated by the AVISPA 2.0 tool represents a significant difference from its previous version, to detect recurring error patterns based on the information obtained from process models developed in SPEM 2.0, through the F-Score and ANOVA.
- **Null Hypothesis H_0**: The information presented by the AVISPA 2.0 tool lacks a significant difference from its previous version. Consequently, it is necessary to reevaluate the interpretations of AVISPA 2.0 users in relation to the process models developed in SPEM 2.0, as well as the efficiency of the AVISPA 2.0 classification.

2.5.2. Variables

- **Independent**: The 18 software process models to be assessed.
- **Dependent**: The AVISPA versions

3. RESULTS

For more information about results of assessments 1 to 4 can be accessed from the AVISPA's wiki [19].

Respect to the experiment design, table 4 illustrates the final F_1 Score results by every process model for AVISPA previous and new versions. Each process model had been measured with AVISPA software process model classification level, in this case, F_1 Score. The closer to 1, the better AVISPA performs classifications.

Table 4. F_1 *Score* Results.

Process Model	F$_1$ Score	
	AVISPA 1.0	**AVISPA 2.0**
PM-1	0.587155963302752	0.947289532850362
PM-2	0.602294455066922	0.893564356435644
PM-3	0.646193771626298	0.886506935687263
PM-4	0.643320363164721	0.857142857142857
PM-5	0.837092731829574	0.913319238900634
PM-6	0.743155149934811	0.893877551020408
PM-7	0.63677130044843	0.884404474665497
PM-8	0.621739130434783	0.874669487043892
PM-9	0.636609558160505	0.875834445927904
PM-10	0.637647058823529	0.872088271352677
PM-11	0.59709911361805	0.942387840156901
PM-12	0.636048526863085	0.883747831116252
PM-13	0.633555420956996	0.880942364324779
PM-14	0.587155963302752	0.947289532850362
PM-15	0.675409836065574	0.854320987654321
PM-16	0.630728513720574	0.889279094722466
PM-17	0.632575757575758	0.884287454323995
PM-18	0.610619469026549	0.846481876332623

Source: Own work

Finally, table 5 illustrates the final ANOVA score.

Table 5. AVOVA results.

Font	SS	Degrees	Square average	F
F$_1$ Score	0,54569	1	0,54569	244,695
Error / Residual	0,07582	34	0,00223	
Total	0,62151	35	0,01775	

Source: Own work

3.1. Analysis

F - Critical could be founded by using the *F* distribution with the numerator degrees of freedom (1) and denominator (34) with a 0.05 area to the right (α value). *F − Critical* which is: 4.13001774565202.

F value is greater than *F - Critical* so *H0* is rejected.

Finally, error bars are generated in order to show where AVISPA probably could classify software process models by their elements (see figure 6):

- The error bars in the figure expose the ANOVA improvements related to two main data distribution characteristics from AVISPA: the data dispersion and data classification scores for 18 software process models samples.
- Q2 and Q3 exposes the main data distribution range (gray boxes). For AVISPA 1.0 the most significant data dispersion range is higher than AVISPA 2.0. This means that AVISPA 2.0 performs a more accurate (F1-Score precision metric) data classification compared to their previous version.
- Q1 to Q2 and Q3 to Q4, exposes some special cases in which AVISPA lacks about precision (parallel lines surrounding the gray boxes). In AVISPA 2.0, Q1 and Q2 the distance are longer than AVISPA 1.0 in same quartiles, which means an AVISPA data classification improvements, but still persist too high dispersion in these cases. A more exhaustive study need to be performed to found this fact causes.
- Finally, the F1-Score metric (classification score), have been highly improved from around 63% media score to around 88% media score about the data classification of the AVISPA binary classifier.
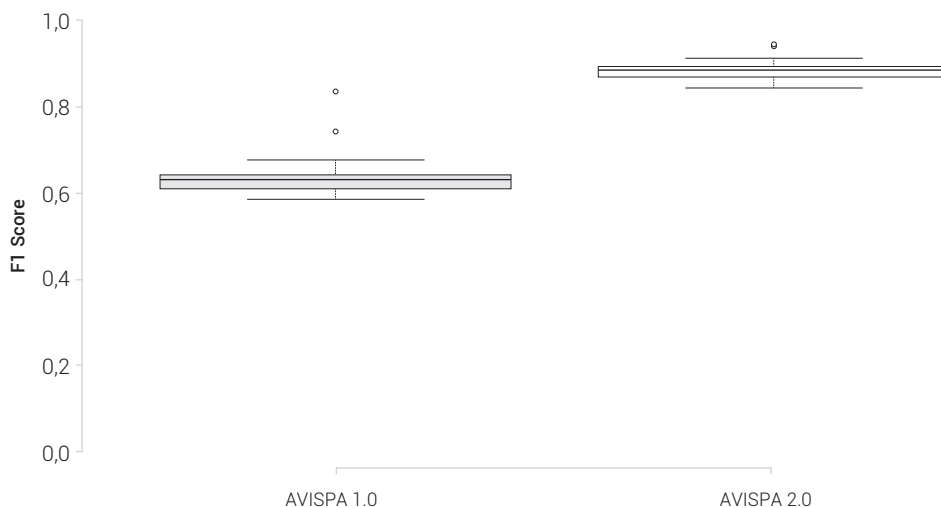

Figure 6. *F₁ Score* error bars.
Source: Own work

To create a classification about the main AVISPA code functions, figure 7 illustrates the relationship between accuracy and recall of each of the most relevant AVISPA functions and how they influence the rating of these metrics.
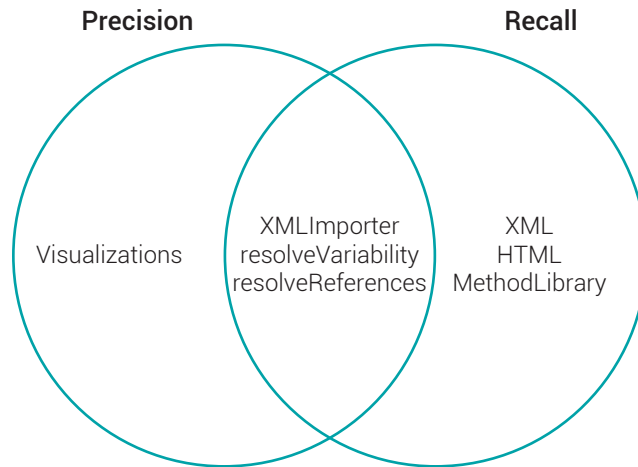
Precision   **Recall**

Visualizations   XMLImporter   XML
                  resolveVariability HTML
                  resolveReferences MethodLibrary

Figure 7. AVISPA elements correlation with Precision and Recall.
Source: Own work

4. DISCUSSION AND CONCLUSIONS

The systematic review of the literature allowed finding crucial elements of research:

- Define the subset of process model metrics in SPEM 2.0 that AVISPA could incorporate respect to their literature importance.
- Define new blueprints and / or error patterns that better illustrate their construction due to research reports that suggest the use of metrics that contribute to the visual assessment of software process elements.
- Remove existing blueprints and / or error patterns that, given the metric inclusion filter due to the number of matches in the articles found are not relevant in the construction of AVISPA visualizations.
    - The organization of the process elements in the imported XML is mainly affected by the various versions of the XML, therefore, of the EPF versions. Therefore, in the evaluation with ANOVA, the process elements that are not correctly classified, are mainly due to this versioning problem of the software process model documentation tool.

- Quartiles 1, 2 and 3 in each of the bar charts (see figure 6) indicate that the qualifications of process models in industry allow us to indicate that, with other process models, exists a 75% of probability that F1Score, could be in a 6% or less to be in their respective average, also applies to Precision and Recall. This implies an improvement in the classification by means of a significant reduction in the dispersion of the ratings and, therefore, of the process model elements in AVISPA.

- An important detail to highlight is the area between quartiles from Q1 to Q2 and from Q4 to Q5. It still remains without significant changes between the two versions of AVISPA for the 3 metrics involved in the ANOVA evaluation.

5. REFERENCES

[1]   J. Lonchamp, "A structured conceptual and terminological framework for software process engineering," in [1993] Proceedings of the Second International Conference on the Software Process-Continuous Software Process Improvement, Feb 1993, pp. 41–53. [Online]. doi: http://dx.doi.org/10.1109/SPCON.1993.236823

[2]   W. Humphrey, Managing the Software Process. Ed. Boston: Addison-Wesley Professional, 1989. Accessed on: Nov. 26, 2019. [Online]. Available: https://dl.acm.org/citation.cfm?id=64795.

[3]   "Software & systems process engineering metamodel (SPEM) 2.0,"2008, version 2.0. Accessed on: Nov. 26, 2019. [Online]. Available: https://www.omg.org/spec/SPEM/2.0/PDF.

[4]   J. A. Hurtado Alegría, M. C. Bastarrica, and A. Bergel, "Analyzing software process models with AVISPA," in Proceedings of the 2011 International Conference on Software and Systems Process, ser. ICSSP'11. New York, NY, USA: ACM, 2011, pp. 23–32. [Online]. doi: http://dx.doi.org/10.1145/1987875.1987882

[5]   J. A. H. Alegría, A. Lagos, A. Bergel, and M. C. Bastarrica, "Software process model blueprints," in New Modeling Concepts for Today's Software Processes, J. Münch, Y. Yang, and W. Schäfer, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 273–284. [Online]. doi: http://dx.doi.org/10.1007/978-3-642-14347-2_24

[6]   J. A. H. Alegría, M. C. Bastarrica, and A. Bergel, "AVISPA: a tool for analyzing software process models" Journal of Software: Evolution and Process, vol. 26, pp. 434–450, 2014. [Online]. doi: http://dx.doi.org/10.1145/1987875.1987882

[7]   J. Salgado, J. Alvarez, and J. Alegría, "Análisis visual de la variabilidad de modelos de procesos de software," in Colombian Conference on Computing, vol. 11, 2016. Accessed on: Nov. 26, 2019. [Online]. Available: https://www.researchgate.net/publication/337547656_Analisis_Visual_de_la_Variabilidad_en_Modelos_de_Procesos_Software.

[8]   J. Simmonds, M. Bastarrica, L. Silvestre, and A. Quispe, "Analyzing methodologies and tools for specifying variability in software processes" Universidad de Chile, 2011. pp. 3–5. Accessed on: Nov. 26, 2019. [Online]. Available: https://pdfs.semanticscholar.org/31f9/6465e0a-55d57a97e7b4c0aa2e41d50617788.pdf

[9]   M. C. Bastarrica, G. Matturro, R. Robbes, L. Silvestre, and R. Vidal, "How does quality of formalized software processes affect adoption?" in Advanced Information Systems Engineering, M. Jarke, J. Mylopoulos,C. Quix, C. Rolland, Y. Manolopoulos, H. Mouratidis, and J. Horkoff,Eds. Cham: Springer International Publishing, 2014, pp. 226–240.[Online]. doi: http://dx.doi.org/10.1007/978-3-319-07881-6_16

[10]   F. B. Cardenas, J. S. Martinez, M. C. Camacho, and J. A. Hurtado, "Análisis visual de la modularidad de modelos de procesos de software avimo-ps," in Computing Colombian Conference, vol. 8, 2013. pp. 2–43. Accessed on: Nov. 26, 2019. [Online]. Available: https://www.slideserve.com/cathal/an-lisis-visual-de-la-modularidad-de-modelos-de-procesos-de-software-avimo-ps

[11]   G. Canfora, F. García, M. Piattini, F. Ruiz, and C. Visaggio, "A family of experiments to validate metrics for software process models," Journal of Systems and Software, vol. 77, no. 2, pp. 113–129, aug 2005. [Online]. doi: http://dx.doi.org/10.1016/j.jss.2004.11.007

[12]   E. Rolón, F. Ruiz, F. O. G. Rubio, and M. G. P. Velthuis, "Applying software process metrics in business process models." Revista de Procesos y Métricas de las Tecnologías de la Información, vol. 3, no. 2, pp. 45–61, 2006. Accessed on: Nov. 26, 2019. [Online]. Available: https://www.aemes.org/biblioteca/revistas-de-procesos-y-metricas/numeros-publicados/category/25-numero-8.html?download=32:applying-software-process-metrics-in-business-process-models

[13]   M. Nørgaard and K. Hornbæk, "What do usability evaluators do in practice?: An explorative study of think-aloud testing," in Proceedings of the 6th Conference on Designing Interactive Systems, ser. DIS '06. New York, NY, USA: ACM, 2006, pp. 209–218. [Online]. doi: http://dx.doi.org/10.1145/1142405.1142439

[14]   M. E. Khan, F. Khan et al., "A comparative study of white box, black box and grey box testing techniques," Int. J. Adv. Comput. Sci. Appl, vol. 3, no. 6, 2012. Accessed on: Nov. 26, 2019. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.261.1758&rep=rep1&type=pdf

[15] R. C. Martin and M. Martin, Agile principles, patterns, and practices in C# (Robert C. Martin). Prentice Hall PTR, 2006. Accessed on: Nov. 26, 2019. [Online]. Available: https://ivandere-vianko.com/wp-content/uploads/2013/10/Agile-Principles-Patterns-and-Practices-in-C.pdf

[16] S. C. Allala, J. P. Sotomayor, D. Santiago, T. M. King, and P. J. Clarke, "Towards transforming user requirements to test cases using MDE and NPL," in 2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC), vol. 2. IEEE, 2019, pp. 350–355. [Online]. doi: http://dx.doi.org/10.1109/COMPSAC.2019.10231

[17] A. Cockburn, Writing Effective Use Cases, 1st ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2000. pp. 133–192. Accessed on: Nov. 26, 2019. [Online]. Available: https://dl.acm.org/citation.cfm?id=517669

[18] S. L. Pfleeger, "Experimental design and analysis in software engineering," Annals of Software Engineering, vol. 1, no. 1, pp. 219–253, Dec 1995. [Online]. doi: http://dx.doi.org/10.1007/BF02249052

[19] "AVISPA wiki". [Online]. Avaliable: https://bit.ly/2OVO3yd, accessed: 2019-08-13.

[20] J. Alvarez and J. Hurtado, "UP-VSE: A unified process - based lifecycle model for very small entities," in Advances in Computing, J. E. Serrano C. and J. C. Martínez-Santos, Eds. Cham: Springer International Publishing, 2018, pp. 260–275. [Online]. doi: http://dx.doi.org/10.1007/978-3-319-98998-3_21

[21] M. I. Kellner, P. H. Feiler, A. Finkelstein, T. Katayama, L. J. Osterweil, M. H. Penedo, and H. D. Rombach, "Ispw-6 software process example," in Proceedings. First International Conference on the Software Process, Oct 1991, pp. 176–186. [Online]. doi: http://dx.doi.org/10.1109/ICSP.1991.664348

[22] "Proceso de Desarrollo Industrial del Grupo de Ingeniería Automática (PRODIGIA).". [Online]. Avaliable: http://artemisa.unicauca.edu.co/~jjalvarezl/PRODIGIA/

[23] K. Kluza, G. J. Nalepa, and J. Lisiecki, Square Complexity Metrics for Business Process Models. Cham: Springer International Publishing, 2014, pp. 89–107. [Online]. doi: http://dx.doi.org/978-3-319-03677-9_6

[24] A. Burattin, V. Bernstein, M. Neurauter, P. Soffer, and B. Weber, "Detection and quantification of flow consistency in business process models," Software & Systems Modeling, vol. 17, no. 2, pp. 633–654, May 2018. [Online]. doi: 10.1007/s10270-017-0576-y

[25] K. Kluza, "Measuring complexity of business process models integrated with rules" in Artificial Intelligence and Soft Computing, L. Rutkowski, M. Korytkowski, R. Scherer, R. Tadeusiewicz, L. A. Zadeh, and J. M. Zurada, Eds. Cham: Springer International Publishing, 2015, pp. 649–659. [Online]. doi: http://dx.doi.org/10.1007/978-3-319-19369-4_57

[26] K. Figl, "Comprehension of procedural visual business process models," Business & Information Systems Engineering, vol. 59, no. 1, pp. 41–67, Feb 2017. [Online]. doi: http://dx.doi.org/10.1007/s12599-016-0460-2

[27] J. Pavlicek, R. Hronza, P. Pavlickova, and K. Jelinkova, "The business process model quality metrics," in Enterprise and Organizational Modeling and Simulation, R. Pergl, R. Lock, E. Babkin, and M. Molhanec, Eds. Cham: Springer International Publishing, 2017, pp. 134–148. [Online]. doi: http://dx.doi.org/10.1007/978-3-319-68185-6_10

[28] J. Pflug and S. Rinderle-Ma, "Process instance similarity: Potentials, metrics, applications," in On the Move to Meaningful Internet Systems: OTM 2016 Conferences, C. Debruyne, H. Panetto, R. Meersman, T. Dillon, e. Kühn, D. O'Sullivan, and C. A. Ardagna, Eds. Cham: Springer International Publishing, 2016, pp. 136–154. [Online]. doi: http://dx.doi.org/10.1007/978-3-319-48472-3_8

[29] A. del Río-Ortega, M. Resinas, A. Durán, B. Bernárdez, A. Ruiz-Cortés, and M. Toro, "Visual ppinot: A graphical notation for process performance indicators," Business & Information Systems Engineering, Jun 2017. [Online]. doi: http://dx.doi.org/10.1007/s12599-017-0483-3

[30] J. Geisel, B. Hamid, D. Gonzales, and J.-M. Bruel, "Towards a methodological tool support for modeling security-oriented processes," in Model and Data Engineering, L. Bellatreche, Ó. Pastor, J. M. Almendros Jiménez, and Y. Aït-Ameur, Eds. Cham: Springer International Publishing, 2016, pp. 31–41. [Online]. doi: http://dx.doi.org/10.1007/978-3-319-45547-1_3

[31] M. Smatti, M. Oussalah, and M. Ahmed Nacer, "Supporting deviations on software processes: A literature overview," in Software Technologies, P. Lorenz, J. Cardoso, L. A. Maciaszek, and M. van Sinderen, Eds. Cham: Springer International Publishing, 2016, pp. 191–209. [Online]. doi: http://dx.doi.org/10.1007/978-3-319-30142-6_11

[32] G. Kaur and K. B. Bahl, "Software reliability, metrics, reliability improvement using agile process," 2014. Accessed on: Nov. 26, 2019. [Online]. Available: http://ijiset.com/v1s3/IJISET_V1_I3_24.pdf

[33] M. Marques, J. Simmonds, P. O. Rossel and M. Bastarrica, "Software product line evolution: A systematic literature review", in Information and Software Technology, 2019, pp. 190–208. [Online]. doi: http://dx.doi.org/10.1016/j.infsof.2018.08.014.

[34] J. Alvarez and J. Hurtado, "UP-VSE: A Unified Process - Based Lifecycle Model for Very Small Entities" in Software Technologies, C. Serrano, E. Jairo and J. Martínez-Santos, Eds. Cham: Springer International Publishing, 2018, pp. 260–275. [Online]. doi: http://dx.doi.org/10.1007/978-3-319-98998-3_21

[35] J. Kim, "Execution Environment for Process Defined in EPF" in Big Data, Cloud Computing, Data Science & Engineering, R. Lee, 2019, pp. 117–129. [Online]. doi: http://dx.doi.org/10.1007/978-3-319-96803-2_10

[36] E. Nogueira, F. Araújo, F. Dione, E. Oliveira Jr, U. Kulesza and C. Werner, "Software process line as an approach to support software process reuse: A systematic literature review", in Information and Software Technology, vol 116, 2019. [Online]. doi: http://dx.doi.org/10.1016/j.infsof.2019.08.007.

[37] T. Mens, A. Decan, and N. Spanoudakis, "A method for testing and validating executable state-chart models", in Software & Systems Modeling, vol. 18, no. 2, pp. 837–863, Apr. 2019, [Online]. doi: http://dx.doi.org/10.1007/s10270-018-0676-3