

# Perfil Operacional do Software: investigando aplicabilidades específicas

Luiz Cavamura Júnior<sup>1</sup>, Sandra Fabbri<sup>1</sup>, and Auri M. R. Vincenzir<sup>1</sup>

Departamento de Computação, Universidade Federal de São Carlos (UFSCar)  
São Carlos, Brasil  
{luiz\_cavamura,sfabbri,auri}@ufscar.br

**Resumo** O Perfil Operacional do Software (*POS*) é uma representação da maneira como os usuários usam o software, na qual as operações, executada pelo software, são associadas a uma probabilidade de ocorrência. Sendo a confiabilidade de um software dependente do contexto no qual ele é usado, o *POS* é empregado na engenharia de confiabilidade de software. Os resultados obtidos pela realização de um Mapeamento Sistemático da Literatura (*MSL*) evidenciaram que a Qualidade de Software é a área mais favorecida pelas contribuições providas pelo uso do *POS* no processo de desenvolvimento de software. Contudo, poucos foram os estudos selecionados associados à essa área, caracterizando-a, junto às demais áreas, como uma área que pode ser explorada. Esses resultados motivaram a realização de uma Revisão Sistemática da Literatura (*RSL*) que investigou o uso do *POS* como um critério de avaliação para casos de uso existentes, ou seja, sob a perspectiva de uso do software, e como um critério de priorização aplicável à artefatos e atividades inerentes ao processo de desenvolvimento de software. Como o *POS* representa a maneira como o software é operado pelos usuários e o teste de regressão é realizado quando há modificações no software, podendo ocorrer enquanto o software está em operação, o uso do *POS* em atividades de teste de regressão também foi investigado. O propósito deste artigo é prover, como contribuições, os resultados obtidos pela *RSL* realizada. Por meio dos resultados obtidos, constatou-se que, sendo um recurso adotado para determinar a confiabilidade do software, o *POS* é usado em técnicas de teste estatístico e de regressão para gerar, selecionar e priorizar os casos de teste. O uso do *POS* em outros contextos de teste e com outros propósitos não têm sido explorado. Constatou-se, também, que por ser um recurso usado em tais técnicas, informações detalhadas sobre o *POS* são escassas. Os resultados obtidos permitiram, também, caracterizar o uso do *POS* nas situações investigadas.

**Keywords:** Perfil Operacional · Revisão Sistemática · Qualidade de Software · Teste de Software · Perfil de Teste.

## 1 Introdução

A necessidade de processos eficientes faz com que as organizações estejam empenhadas na melhoria desses processos. O software é um elemento estratégico nesses processos e, assim, pode ter recursos que permitam ser parametrizado para se adaptar às diversificadas maneiras de operação, atendendo às necessidades específicas de cada organização. Nesse contexto, visando a praticidade e eficiência em suas atividades, usuários experientes conseguem adaptar a maneira como o software é operado [24]. Como a qualidade do software é dependente do

seu uso operacional [8], a maneira como o software é operado torna-se relevante ao processo de desenvolvimento de software. O Perfil Operacional do Software (*POS*) corresponde a uma especificação do software que representa a maneira como o software é operado por seus usuários [8,24]. Musa [16] define o *POS* como uma caracterização quantitativa do software. Essa caracterização é baseada nas operações do software, para as quais uma distribuição probabilística é atribuída para indicar as partes do software mais usadas pelos usuários.

Dada a relevância do *POS* ao processo de desenvolvimento de software, este trabalho relata a realização de uma Revisão Sistemática da Literatura (*RSL*) que investigou o uso do *POS* sob três perspectivas (situações de uso): em atividades associadas ao teste de regressão ( $SU_1$ ); como um critério de priorização aplicável a artefatos e atividades inerentes ao processo de desenvolvimento de software ( $SU_2$ ); e como um critério de avaliação para casos de teste existentes ( $SU_3$ ). As situações de uso investigadas pela *RSL* foram definidas e estão associadas a uma pesquisa de doutoramento que investiga o *POS* em um escopo mais amplo [5]. A investigação do uso do *POS* em técnicas de teste de regressão foi motivada pela natureza do teste de regressão, o qual é executado quando modificações são feitas em um software existente [28]. No momento em que o teste de regressão é realizado, o software pode fornecer dados sobre a maneira como é operado por seus usuários. O *POS*, por representar a maneira como o software é operado pelos usuários, corresponde a um critério de priorização baseado no uso. A investigação do uso do *POS* como critério de priorização busca identificar outras atividades e recursos que podem adotar o *POS* como um critério de priorização. A investigação do uso do *POS* como um critério de avaliação dos testes existentes, gerados por outros critérios, foi definida em decorrência do possível descompasso entre as partes testadas do software e o *POS* [6]. O uso do *POS* como critério de priorização e avaliação são exemplos de uso que podem contribuir com a qualidade do software do ponto de vista dos usuários e, também, ser usado para priorizar atividades do processo de desenvolvimento do software de maior interesse do usuário, tais como manutenção, melhorias e customizações.

Este trabalho provê, como contribuições, a partir dos resultados obtidos, uma base de conhecimento sobre o uso do *POS* nos contextos investigados, fornecendo uma caracterização e entendimento quanto ao uso do *POS* nesses contextos. Este trabalho está organizado da seguinte forma: na Seção 2, os conceitos que envolvem o *POS* e a qualidade de software são abordados. A Seção 3 refere-se aos trabalhos relacionados. A Seção 4 descreve a metodologia adotada para a realização da pesquisa. As Seções 5, 6, 7, 8 e 9 descrevem a *RSL* realizada e os respectivos resultados. As ameaças à validade deste trabalho e as conclusões são apresentadas nas Seções 10 e 11 respectivamente.

## 2 *POS* e a Qualidade de Software

Por representar a maneira como o software será utilizado por seus usuários e sendo a confiabilidade de um software dependente do contexto em que o software é usado, o *POS* é usado em atividades relacionadas a engenharia de confiabilidade de software. Nessas atividades, o *POS* é usado para gerar dados de teste que reproduzem a maneira com a qual o software é operado pelos usuários, garantindo que os indicadores de confiabilidade do software sejam válidos para a operação do software em um ambiente de produção [17]. A confiabilidade do software é um dos atributos associados à qualidade de software e a representa a partir do ponto de vista do usuário [4,15]. Para o usuário do software, a frequência com que uma falha se torna aparente, durante a operação do software, é mais

significante que as falhas remanescentes [26]. A confiabilidade do software corresponde a probabilidade de operação do software, por um determinado tempo, em um ambiente específico, sem a ocorrência de falhas [8,15]. Sendo a confiabilidade determinada pelo tempo da operação do software sem a ocorrência de falhas, e que uma falha revelada a um usuário pode não ser revelada no modo de trabalho de outro usuário, a atividade de teste está implícita à confiabilidade do software. O *POS* não garante que todos os defeitos sejam detectados, mas garante que as operações mais utilizadas do software sejam testadas.

Um Mapeamento Sistemático da Literatura (*MSL*) investigou como as contribuições providas pelos estudos que abordaram o uso do *POS* estão distribuídas nas áreas da engenharia de software correlatas ao processo de desenvolvimento de software, evidenciando que a qualidade de software é a área mais abordada pelos estudos que usaram o *POS* como um recurso nas estratégias abordadas nesses estudos. A maioria dessas estratégias estão associadas à confiabilidade de software. Embora a qualidade de software seja a área mais abordada, são poucos os estudos relacionados a essa área, situação que a caracteriza, junto às demais áreas identificadas, como uma área que pode ser explorada. A descrição detalhada do *MSL* realizado, e dos respectivos resultados, está disponível em: [lcv.com.br/artigos/anexos/cibse2020/cap-3-rs-ms.pdf](http://lcv.com.br/artigos/anexos/cibse2020/cap-3-rs-ms.pdf).

### 3 Trabalhos Relacionados

Dentre os trabalhos selecionados pelo *MSL*, Smidts et al. [23] forneceram uma caracterização do modelo baseado no *POS* usado na geração de casos de teste, assemelhando-se a este trabalho dada a abordagem investigativa focada diretamente no *POS*. Os demais trabalhos usaram o *POS* como um recurso às estratégias propostas, não fornecendo, detalhadamente, informações sobre o *POS*. A caracterização, provida em [23], foi obtida por meio de mapeamentos sistemáticos e análises qualitativas. Os dados extraídos foram estruturados e classificados. A classificação desses dados forneceu uma taxonomia que diferencia e analisa os modelos baseados no *POS* sob várias perspectivas: características comuns entre os modelos, dependências necessárias à elaboração dos modelos, características do software sob teste usadas pelos modelos, dentre outros.

Constatou-se, por meio dos resultados obtidos em [23], que uma variedade de modelos baseados no *POS* podem ser construídos, sendo a definição e complexidade do modelo dependente do contexto de aplicação do software sob teste. Embora as classificações e a taxonomia desenvolvidas proveem uma compreensão e perspectivas sobre o uso do *POS* [23], por abordar especificamente os modelos de representação aplicados para a geração de casos de teste, o estudo não provê informações detalhadas sobre a obtenção do *POS*. O uso do *POS* associado a outras finalidades também não foi explorado em [23]. A *RSL* descrita neste trabalho investiga o uso do *POS* em situações de uso específicas, fornecendo, por meio da abordagem investigativa adotada, informações não providas pelo estudo realizado por Smidts et al. [23].

### 4 Metodologia

Foram definidas 4 questões de pesquisa cuja consecução das respostas proveem a investigação sobre o uso do *POS* nas situações descritas na seção anterior. A Tabela 1 apresenta as questões de pesquisa que foram definidas, aplicáveis às situações de uso do *POS* investigadas.

Tabela 1. Questões de pesquisa

Q <sub>i</sub>	Descrição
Q <sub>1</sub>	Como o <i>POS</i> foi usado nas técnicas identificadas nos estudos selecionados?
Q <sub>2</sub>	Como o <i>POS</i> foi obtido pelas técnicas identificadas?
Q <sub>3</sub>	Como o <i>POS</i> foi caracterizado pelas técnicas identificadas?
Q <sub>4</sub>	Relacionado ao uso do <i>POS</i> , quais foram as dificuldades reportadas pelos estudos selecionados?

Para obter respostas às questões de pesquisa, a investigação foi feita por meio de uma *RSL* sucedida pela técnica *snowballing* (*backward* e *forward*). Para a realização dessas atividades, a metodologia de pesquisa proposta em [25] foi instanciada. A Figura 1 ilustra essa metodologia.

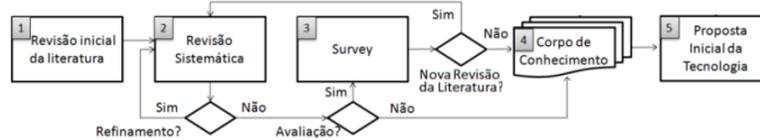


Figura 1. Representação da metodologia adotada. Adaptado de [25]

A etapa inicial (1), denominada "Revisão inicial da literatura", consiste em coletar informações básicas a respeito do tema a ser pesquisado, necessárias para a elaboração do protocolo da *RSL*. A etapa seguinte (2) compreende o planejamento e execução da *RSL* a qual, dependendo da análise dos resultados obtidos, pode ser refinada. Após a realização da *RSL*, opcionalmente, em decorrência de possíveis questionamentos que podem surgir sobre o conhecimento obtido, um *survey* (Etapa 3) pode ser realizado, o qual consiste em submeter o conhecimento adquirido à uma avaliação de especialistas. As duas etapas finais da metodologia (4 e 5) compreendem, respectivamente, à estruturação e consolidação do conhecimento adquirido, tornando-se a base para a idealização de novas tecnologias.

Para este trabalho, a *Etapa 1* usou os resultados do *MSL* citado na Seção 2. A *Etapa 2* refere-se à execução da *RSL* seguida pela execução da técnica *snowballing*. Referente à *Etapa 3*, o conhecimento adquirido na *Etapa 2*, realizada pelo primeiro autor, foi submetido a uma avaliação de outros pesquisadores, demais autores deste trabalho. Na *Etapa 4*, os dados obtidos pela *RSL* foram estruturados, permitindo obter respostas às questões de pesquisa e, assim, prover as contribuições esperadas para este trabalho. Essas contribuições servem como dados de entrada para a *Etapa 5*, a qual não foi abordada neste trabalho.

A *RSL* foi conduzida por um processo composto por três fases [18]: Planejamento, Condução e Publicação. O paradigma *GQM* (*Goal, Question, Metric*) foi adotado como referência para o planejamento da *RSL*. A ferramenta computacional *Start*, cuja finalidade, é prover apoio, ao pesquisador, na realização de revisões sistemáticas da literatura [9], foi usada para dar suporte à realização da *RSL*.

## 5 Planejamento

A *RSL* foi guiada por um protocolo no qual as diretrizes da pesquisa foram definidas. As subseções seguintes apresentam uma síntese desse protocolo.

### 5.1 Questões de pesquisa

O paradigma *GQM* foi adaptado para, a partir dos propósitos da pesquisa, estruturar as questões de pesquisa, identificando os dados a serem extraídos dos estudos selecionados para prover as repostas às respectivas questões. A Figura 2 ilustra, na estrutura do paradigma *GQM*, o objetivo, as questões de pesquisa e os dados a serem obtidos.

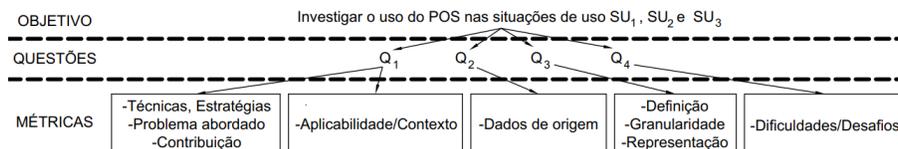


Figura 2. GQM referente ao planejamento da *RSL*

Conforme ilustrado pela Figura 2, o objetivo da *RSL* corresponde ao *Goal*, as questões de pesquisa correspondem ao *Question* e os dados a serem extraídos, dos estudos selecionados, correspondem ao *Metric*.

### 5.2 Grupo de controle e critérios de seleção de estudos

Foram considerados, como grupo de controle, os estudos [2] e [16]. O estudo [2] evidencia, por meio dos resultados apresentados, que informações sobre como o software é operado por seus usuários são relevantes ao processo de desenvolvimento de software. O estudo [16] refere-se a abordagem de Musa sobre o *POS*, a qual é uma das mais relevantes abordagens para a construção do *POS*. A ocorrência desses estudos, como grupo de controle, foi verificada no conjunto de estudos identificados pela *RSL* e, também, nas respectivas referências bibliográficas desses estudos. A Tabela 2 apresenta os critérios de inclusão (I) e exclusão (E) usados para a seleção dos estudos.

Tabela 2. Critérios de inclusão e exclusão

Critério	
I	O estudo usa o <i>POS</i> em teste de regressão
I	O estudo usa o <i>POS</i> como critério de priorização
I	O estudo foca no <i>POS</i> e suas características
I	O estudo usa o <i>POS</i> em técnica/estratégia para avaliar casos de teste
E	O estudo não usa o <i>POS</i> em teste de regressão
E	O estudo não aborda o teste de regressão
E	O estudo não aborda o <i>POS</i> e suas características
E	O estudo não usa o <i>POS</i> como critério de priorização
E	O estudo aborda o <i>POS</i> porém o <i>POS</i> não é usado em técnica/estratégia para avaliar casos de teste
E	O estudo aborda a avaliação de casos de teste porém não usa o <i>POS</i>

### 5.3 String de busca

As *keywords* iniciais, relacionadas ao *POS*, foram obtidas pelo *MSL* citado na Seção 2, definindo uma *string* de busca inicial. A *string* de busca inicial foi usada em um estudo piloto para identificar novas *keywords*. Os estudos, referentes ao resultado retornado pelo estudo piloto, foram registrados na ferramenta computacional *Start*. Dentre suas funcionalidades, a *Start* fornece uma lista das *keywords* detectadas nos estudos registrados, classificando-as como "usadas" ou "não usadas" na *string* de busca, e ordenando-as pela frequência com que cada uma é empregada nos estudos registrados [9]. As *keywords* consideradas relevantes foram identificadas e incorporadas à *string* de busca inicial, dando origem

a uma nova versão da *string* de busca, iniciando-se, assim, uma iteratividade na execução do estudo piloto. Essa iteratividade foi processada enquanto novas *keywords* eram identificadas.

Após a definição das *keywords* relacionadas ao *POS* ( $S_{pos}$ ), novas *keywords* para tratar a especificidade de cada situação de uso do *POS* investigada foram adicionadas. As *strings* de busca, relacionadas às respectivas situações de uso do *POS* a serem investigadas ( $SU_{1,2,3}$ ), foram construídas a partir das *keywords* e do uso dos operadores lógicos “and” ( $\wedge$ ) e “or” ( $\vee$ ). As *strings* de busca são mostradas na Tabela 3.

**Tabela 3.** *Strings* de busca

String de Busca	SU
("operational profile" $\vee$ "usage model" $\vee$ "usage profile" $\vee$ "user profile" $\vee$ "users profile" $\vee$ "profile of use" $\vee$ "usage testing" $\vee$ "input distribution" $\vee$ "usage distribution" $\vee$ "operational distribution" $\vee$ "usage chain")	$S_{pos}$
("regression" $\wedge$ ("software") $\wedge$ ("testing" $\vee$ "test") $\wedge$ $S_{pos}$ )	$SU_1$
("prioritization" $\vee$ "prioritize") $\wedge$ $S_{pos}$	$SU_2$
("effectiveness" $\vee$ "efficiency" $\vee$ "evaluation") $\wedge$ $S_{pos}$	$SU_3$

## 6 Condução

As fontes de informação selecionadas foram *Scopus*, *ACM Digital Library*, *IEEE Xplore*, *SpringerLink*, *ScienceDirect*, *Engineering Village* e *Web of Science*. A identificação dos estudos foi feita por meio de busca automática nas fontes de informação usadas, identificando, para as situações de uso investigadas  $SU_1$ ,  $SU_2$  e  $SU_3$ , respectivamente, 130, 86 e 214 estudos. A identificação e seleção de estudos foram realizadas separadamente, não havendo dependência entre as situações de uso investigadas. Na fase de seleção, dos estudos identificados, foram selecionados os estudos em que no título, resumo ou palavras-chaves, foram identificados ao menos um dos critérios de inclusão definidos. Na fase de extração, foram mantidos os estudos que, após a leitura completa e análise desses estudos, mantiverem ao menos um dos critérios de inclusão atribuídos. Os critérios de inclusão sobrepõe os critérios de exclusão. Do total de estudos identificados, 9 estudos foram selecionados para a fase de extração de dados. Esses estudos foram submetidos a técnica *snowballing*, identificando 214 novos estudos para a situação de uso  $SU_1$ . Para as situações de uso  $SU_2$  e  $SU_3$ , a técnica *snowballing* identificou, respectivamente 46 e 184 novos estudos. Dos 444 novos estudos identificados pela técnica *snowballing*, 2 estudos foram adicionados à fase de extração de dados, totalizando, para a extração de dados, 11 estudos. A Figura 3 mostra a quantidade de estudos processados em cada etapa da *RSL*.

## 7 Qualidade dos Estudos Selecionados

A cada estudo selecionado na *RSL*, um indicador, cujo valor varia entre 0 e 1, foi atribuído visando quantificar a relevância do estudo aos propósitos deste trabalho. O indicador corresponde ao valor da média aritmética dos valores atribuídos às questões de avaliação para cada estudo selecionado. Quanto maior o valor do indicador, maior é a relevância do estudo com base na avaliação da qualidade do estudo. Dada a abordagem investigativa e exploratória deste trabalho, não foi definido um valor mínimo a ser obtido pelos estudos selecionados. A Tabela 4 apresenta as questões avaliativas, a motivação que deu origem a questão e os valores que podem ser atribuídos à questão.

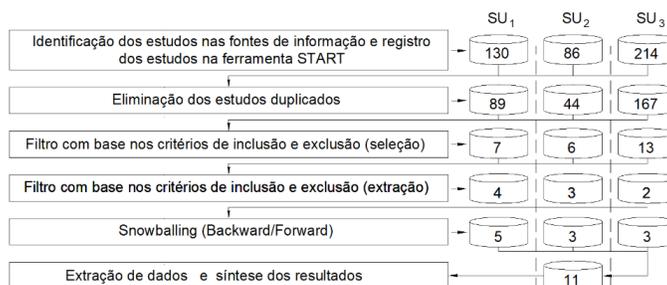


Figura 3. Quantitativo de estudos processados

Tabela 4. Questões avaliativas da qualidade dos estudos

Questão	Motivação	Valor
1-Os estudos selecionados citam estudos publicados pelos autores dos estudos de controle?	Citações de estudos publicados pelos autores dos estudos de controle são indícios de que o estudo selecionado tenha maior relevância em relação aos demais estudos	Entre 0 e 1 (conforme a quantidade de autores, dos estudos de controle, citados)
2-Qual o número de citações, em outros estudos, do estudo selecionado?	O número de citações do estudo selecionado evidencia a abrangência do estudo em contextos relacionados ao abordado neste trabalho	Entre 0 e 1 (conforme o número de citações do estudo selecionado)
3-O estudo selecionado tem o POS como foco principal da pesquisa ou relata o uso do perfil operacional?	Estudos cuja abordagem relata o uso do POS estão diretamente relacionados aos propósitos deste trabalho	0 ou 1 conforme a abordagem dada ao POS (Relata o uso do POS: 1; Foco nos conceitos do POS: 0,5)

## 8 Resultados Obtidos

A RSL, sucedida pela técnica *snowballing*, resultou em 11 estudos selecionados para a fase de extração de dados. A Tabela 5 apresenta esses estudos associados às situações de uso investigadas (SU<sub>1,2,3</sub>).

Tabela 5. Estudos selecionados para a extração de dados

SU	Extração de dados
SU <sub>1</sub>	Whittaker et al. [27], Takagi et al. [26], Kashyap [11], Sarwar et al. [22], Rogstad and Briand [21]
SU <sub>2</sub>	Marijan et al. [13], Miransky et al. [14], Nakornburi and Suwannasart [19]
SU <sub>3</sub>	Amrita and Yadav [1], Bertolino et al. [3], Chen et al. [7], Takagi et al. [26]

Observa-se que o estudo [26] foi identificado e selecionado nas investigações das situações de uso SU<sub>1</sub> e SU<sub>3</sub>. As subseções seguintes apresentam, para cada situação de uso investigada, uma síntese desses estudos e, também, uma caracterização do uso do POS obtida por meio dos dados extraídos.

### 8.1 Uso do POS em técnicas de teste de regressão (SU<sub>1</sub>)

Os estudos [22] e [21] abordam a seleção e priorização de casos de teste. Para Sarwar et al. [22], a execução de todos os planos de teste (manuais e automatizados) é uma atividade cansativa que consome tempo, custo e pode não ser efetiva na descoberta, antecipadamente, de defeitos críticos. Assim, é necessário selecionar poucos, porém relevantes, casos de teste a partir de uma massa de teste. Para isso, é proposta uma técnica para priorizar casos de teste aplicada a processos de desenvolvimento de software gerenciados pelo método organizacional *Scrum*. A técnica representa o POS por meio da frequência com a qual os casos de teste são executados durante o teste de regressão. Os casos de teste

são priorizados por meio de um indicador quantitativo atribuído a cada caso de teste. O indicador é calculado considerando-se a frequência de execução do caso de teste, o número de falhas detectadas pelo caso de teste e a criticidade atribuída ao caso de teste.

Rogstad and Briand [21] realizaram uma investigação empírica sobre estratégias para a seleção e geração de dados de teste para técnicas de teste combinatório em nível de teste de regressão. A investigação foi realizada em um ambiente de desenvolvimento de uma aplicação persistente real e representativa. Essa investigação foi motivada pela demanda de tempo e esforço dos testadores decorrentes do uso de dados obtidos em um ambiente de produção do software (dados reais) nas atividades de teste. O uso de dados obtidos em ambiente de produção é uma estratégia comum em atividades de teste, porém, esses dados nem sempre satisfazem os requisitos do plano de teste, forçando os testadores a manipularem ou estenderem os dados de produção para torna-los adequados. O *POS* do software é usado adicionalmente em uma das técnicas de teste aplicadas na investigação, gerando um conjunto de testes alinhado com o uso do software.

Em decorrência dos problemas inerentes às tecnologias existentes, tais como a demanda de esforço e uso de técnicas *ad-hoc*, os quais geram uma alta quantidade de caminhos e cenários de teste que não refletem o uso real do software, Kashyap [11] apresenta uma abordagem para criação de modelo de comportamento que permite a geração e priorização automática dos casos de teste com base em dados de uso do software. Nessa abordagem, é criado um modelo de comportamento que permite a geração e priorização automática dos casos de teste com base em dados de uso do software.

Para Takagi et al. [26], métodos existentes baseados em teste de regressão não são apropriados para avaliar a confiabilidade do software. Takagi et al. [26] apresentam uma técnica que avalia a confiabilidade do software como parte do processo de manutenção, estimando e prevendo falhas que influenciarão na confiabilidade das próximas versões do software. A técnica gera um modelo de teste a partir do *POS*. Os resultados obtidos pela execução dos testes são mapeados no modelo de teste criado. Quando uma falha é detectada, um estado, que representa a falha, é inserido no modelo de teste e associada aos estados que representam as operações nas quais a falha foi detectada. O modelo de teste é usado para determinar o critério de parada dos testes, podendo ser determinado pela cobertura dos estados ou transições que compõe o modelo de teste, pelo número de transições incidentes nos estados de falha ou pelo número de falhas detectadas e mapeadas no modelo de teste.

Segundo Whittaker et al. [27], as estimativas de confiabilidade são obtidas por meio de modelos de probabilidade construídos a partir de dados coletados durante a realização dos testes e, assim, situações de falha são tratadas assumindo-se que obedecem uma regra de probabilidade, permitindo, a partir disso, estimar e prever futuras ocorrências de falhas. Essa abordagem fornece uma análise precisa quando a distribuição probabilística tem suporte empírico e os testes cobriram grande parte do software, porém, quando não há suporte empírico e os recursos de teste são escassos, outras abordagens são necessárias [27]. Whittaker et al. [27] propõem uma estratégia para a detecção de defeitos na manutenção do software. A estratégia proposta combina dados e resultados de testes passados com dados atuais em um único modelo de uso. Com base no modelo de uso, padrões de ocorrência de falhas são identificados para prever os resultados dos testes na próxima compilação do software.

## 8.2 Uso do *POS* como um critério de priorização (SU<sub>2</sub>)

Segundo Marijan et al. [13], à medida que a complexidade de um software aumenta, a qualidade do sistema é submetida a um potencial decréscimo. Para tratar essa situação, é proposta uma metodologia para verificação automática de sistemas multimídia compreendendo a modelagem do comportamento do sistema, a geração de casos de teste baseados no modelo criado e a execução automática dos casos de teste gerados [13]. A modelagem do *POS* é baseada na especificação do sistema e em dados coletados durante a operação do sistema pelos usuários. As sequências de transições do modelo uso dão origem aos casos de teste a serem executados. Os resultados dos testes são analisados para avaliar a confiabilidade do software.

Os dados de uso coletados dos clientes fornecem informações quantitativas valiosas sobre os padrões de uso do software. Para Miransky et al. [14], reunir esses perfis a partir de um grande conjunto de clientes demanda tempo e recursos. Assim, Miransky et al. [14] propõem uma técnica para a seleção e priorização de um conjunto mínimo de clientes a serem considerados na elaboração do *POS*, tendo, como objetivo, prover, na execução dos testes, uma cobertura para determinados conjuntos de defeitos. A técnica proposta, inicialmente, classifica e prioriza os clientes com base no histórico e tipos defeitos reportados por eles. Após a classificação e priorização dos clientes por meio de uma análise qualitativa, a técnica seleciona, para a criação do *POS*, um conjunto mínimo de clientes que tenham descoberto o maior número de defeitos. A seleção manual desse conjunto pode ser inviável. Assim, para se obter o conjunto mínimo de clientes, a priorização dos clientes, feita de maneira qualitativa, é transformada em quantitativa adotando-se o algoritmo de programação inteira binária.

Nakornburi and Suwannasart [19] relatam o desenvolvimento de uma ferramenta computacional para gerar casos de teste usando técnicas de teste combinatório e o *POS*. O *POS* é usado para priorizar e selecionar um conjunto ideal de parâmetros de entrada e os respectivos valores para os casos de teste gerados pela ferramenta proposta.

## 8.3 Uso do *POS* como um critério de avaliação (SU<sub>3</sub>)

Segundo Amrita and Yadav [1], muitos pesquisadores abordam a seleção de casos de teste com base no *POS*, porém, pouco é discutido sobre as partes infrequentes. Amrita and Yadav [1] propõem um modelo que fornece a flexibilidade de alocar casos de teste conforme a prioridade definida pelo *POS* e, também, pela experiência do testador. Assim, o modelo proposto considera tanto a probabilidade de ocorrência quanto a severidade da operação. Com base nessas informações, usando lógica *fuzzy*, o modelo propõe a seleção de casos de teste. O modelo usa a criticidade atribuída às operações do software e o *POS* como variáveis de entrada para um sistema de inferências *fuzzy*. Assim, as operações do software são classificadas em: operação frequente e crítica; operação frequente e não crítica; operação infrequente e crítica; operação infrequente e não crítica. O valor da variável de saída, a ser obtido pelo sistema de inferências *fuzzy*, corresponde ao número de casos de teste a serem alocados para a operação. Fornecidas as variáveis de entrada e definidas as regras *fuzzy*, é obtido um conjunto *fuzzy* resultante que, após a geração, com base nas variações adotadas, é *desfuziado* para se obter os valores equivalentes e, conseqüentemente, a quantidade sugerida de casos de teste a serem alocados em cada operação.

Segundo Bertolino et al. [3], o teste baseado no *POS*, por focar nas falhas com maior probabilidade de ocorrência, pode sofrer saturação e, assim, perder eficácia

na detecção de falhas. Nesse contexto, uma técnica de teste de software adaptativa e iterativa, baseada no *POS*, denominada *covrel*, é proposta por Bertolino et al. [3]. A técnica proposta é executada iterativamente. Na primeira iteração, a técnica usa o *POS* para selecionar e executar os casos de teste randomicamente, com base na probabilidade de ocorrência das partições do domínio de entrada do programa. Nas iterações seguintes, a cada iteração, a técnica calcula, para cada partição: (i) a quantidade de casos de teste ideal a serem selecionados; (ii) seleciona, prioriza e executa a quantidade de casos de teste calculada. Para determinar a quantidade ideal de casos de teste, uma probabilidade é obtida para representar o quanto o teste da partição contribui com a confiabilidade do programa. A seleção e priorização dos casos de teste é feita com base na frequência com que as partes do programa são exercitadas durante a execução dos testes nas iterações anteriores. Como o foco da abordagem é selecionar casos de teste que cobrem partes do programa que são pouco exercitadas, casos de teste associados às partes do programa pouco exercitadas têm maior prioridade.

Quando um conjunto de testes não detecta novas falhas, o uso desses testes, ou similares, pode superestimar a confiabilidade do software. Assim, quanto mais redundantes forem os casos de teste em relação ao código coberto, mais superestimada será a confiabilidade do software. Nesse contexto, Chen et al. [7] propõem uma técnica que determina uma taxa de compressão que ajusta o intervalo de tempo entre as falhas quando são detectados casos de teste redundantes durante a execução do conjunto de teste. Para obter a taxa de compressão são considerados o tempo de execução do conjunto de teste, a cobertura de código provida e, também, a detecção de casos de testes redundantes no conjunto de teste.

#### 8.4 Caracterização do uso do *POS* nos estudos selecionados

A Tabela 6 mostra, a partir dos dados extraídos dos estudos selecionados (Tabela 5), uma caracterização do uso do *POS* nesses estudos.

Tabela 6. Questões avaliativas da qualidade dos estudos

Referência	Granularidade adotada para o POS			Origem do POS				Representação do POS				Aplicabilidade				Atividade-fim que usou o POS		Artefato									
	Dados de Entrada	Operações	Casos de teste	Arquivos de Log	Projetos anteriores	Histórico de uso do software	Conjunto de teste	Observação dos usuários	Não especificado	Registro de frequência de uso da granularidade adotada	Modelo de uso contendo distribuição probabilística	Não se aplica	Teste combinatório	Teste de regressão	Teste randômico	Teste baseado em modelo	Teste de confiabilidade		Não especificado	Geração	Seleção	Priorização	Estimativa	Adequação	Casos de teste	Dados do cliente	
19	X			X					X			X						X							X		
21	X			X						X		X		X				X								X	
26		X		X						X			X	X				X								X	
11		X						X		X						X	X		X							X	
22			X			X			X				X						X			X				X	
13		X			X			X		X			X			X			X							X	
27		X				X				X			X						X			X				X	
14			X					X				X					X		X							X	X
31	X							X		X						X			X	X						X	
1		X						X		X						X			X							X	
7	X							X		X		X							X	X						X	

Por meio da Tabela 6, é possível visualizar a granularidade do *POS*, as fontes de dados usadas para obter o *POS*, a maneira como o *POS* foi representado nos

estudos selecionados, a atividade na qual o *POS* foi usado e os artefatos abordados pelas atividades nas quais o *POS* foi usado. A granularidade, apresentada na Tabela 6, no contexto deste trabalho, corresponde ao nível de fragmentação, conceitual ou estrutural, com a qual o software é abordado para que uma distribuição probabilística seja atribuída aos fragmentos gerados.

## 9 Análise dos Resultados

A análise dos resultados obtidos permitiu responder às questões de pesquisa que foram definidas na fase de planejamento da *RSL*. As questões de pesquisa com as respectivas respostas são apresentadas seguir.

### (a) Como o *POS* foi usado nas técnicas identificadas nos estudos selecionados em cada situação de uso do *POS* investigada?

- *Uso do POS em técnicas de teste de regressão (SU<sub>1</sub>)*: Dado que modelos de uso do software são a base do teste estatístico, as abordagens baseadas em teste estatístico são geralmente referenciadas como teste baseado em uso [11]. Assim, verificou-se, nos estudos aceitos, que o uso do *POS* é associado, geralmente, à técnicas de teste combinatório e estatístico, as quais são empregadas nas estratégias de teste de confiabilidade e de regressão. Esse cenário é entendível dado que o teste estatístico tem, como um de seus objetivos, avaliar a confiabilidade do software e o teste de regressão tem, em sua essência, validar modificações ocorridas no software, as quais podem afetar a confiabilidade do software. Em ambas as situações, dados sobre a maneira com a qual o software é operado por seus usuários são relevantes e podem ser coletados durante a operação do software. Dos 344 estudos identificados e processados para a situação de uso do *POS*  $SU_1$ , 5 estudos foram selecionados para a fase de extração: [21,26,22,27,11]. Esses estudos abordaram o uso do *POS* em sistemáticas para gerar e priorizar casos de teste.
- *Uso POS como critério de priorização (SU<sub>2</sub>)*: Dos 132 estudos identificados e processados para a referida categoria de uso, 3 estudos foram selecionados para a fase de extração: [19,13,14]. Desses estudos, 2 estudos usaram o *POS* em sistemáticas para geração e priorização de casos de teste. Um estudo usou o *POS* em uma sistemática para selecionar e priorizar de um conjunto mínimo de clientes para a obtenção de perfis operacionais específicos para cobrir determinados conjuntos de defeitos.
- *Uso do POS como critério de avaliação de casos de teste (SU<sub>3</sub>)*: Dos 398 estudos identificados e processados para a referida categoria de uso, 3 estudos foram selecionados para a fase de extração: [3,1,7]. Esses estudos usaram o *POS* em técnicas para avaliar os casos de teste tendo, como propósito, selecionar e priorizar casos de teste relevantes às operações a serem testadas. Desses estudos, os estudos [3,7] avaliaram os casos de teste por meio da cobertura provida por eles, sendo que, em um desses estudos, a cobertura era avaliada em relação ao *POS* e, no outro estudo, a cobertura dos casos de teste era usada como um critério de comparação para identificar casos de teste redundantes. O estudo [1] avaliou os casos de teste por meio de critérios definidos na técnica apresentada, tais como criticidade da operação testada, frequência de execução dos casos de teste e número de falhas descobertas.

Constatou-se, assim, que as técnicas identificadas trataram a obtenção e o uso do *POS* como "atividades-meio". Entende-se por "atividade-meio" uma atividade cuja execução é necessária para que a técnica proposta pelo estudo possa ser realizada ("atividade-fim"). Por ser uma "atividade-meio", poucos detalhes foram fornecidos quanto a obtenção e representação do *POS*. Verifica-se,

também, pela Tabela 6, que o uso *POS* tem maior evidência nos estudos cujas técnicas apresentadas tratam a seleção e priorização de casos de teste.

(b) **Como o *POS* foi obtido pelas técnicas identificadas?**

Dados obtidos a partir de arquivos de *log*, gerados durante a operação do software, foi a estratégia com maior evidência dentre os estudos selecionados. O uso dessa estratégia foi constatado nos estudos [19,21,26,13]. Informações providas pela expectativa dos desenvolvedores, por softwares similares, pela experiência em projetos anteriores e pela observação do comportamento dos usuários podem, também, contribuir para se obter o *POS* [26,13]. [26] considera, também, na inexistência de outras fontes de informação, uma distribuição probabilística uniforme para todas as transições contidas no modelo de uso que representa o *POS*. O uso da especificação dos software para a estruturação do modelo de uso que representa o *POS* é considerado em [13].

(c) **Como o *POS* é caracterizado pelas técnicas identificadas?**

Os dados extraídos dos estudos selecionados, listados na Tabela 5, permitiram caracterizar o uso do *POS* quanto à granularidade adotada para a concepção do *POS*, origem dos dados que permitiram a concepção do *POS*, a maneira como o *POS* é representado, a estratégia e técnica de teste favorecidas pelo uso do *POS* e a relação entre o uso do *POS* e a abordagem provida pelos estudos aos casos de teste (Tabela 6). Constatou-se que a granularidade adotada para se obter o *POS* não se restringe à dados de entrada, operações e os estados que o software pode assumir. Em [26,13], além das operações do software, os estados de falha assumidos pelo software também foram considerados como um nível de granularidade para a elaboração do *POS*. O *POS*, abordado em [27], foi obtido a partir da frequência de execução e quantidade de falhas descobertas pelos casos de teste do software.

(d) **Relacionado à obtenção e uso do *POS*, quais foram as dificuldades e problemas reportados pelos estudos selecionados?**

Como foi descrito, a identificação do *POS* nas técnicas identificadas é abordada como uma "atividade-meio", não fornecendo detalhes sobre a maneira como o *POS* foi obtido. Apenas Bertolino et al. [3], por se tratar do problema abordado pela técnica apresentada, relata problemas relacionados ao uso do *POS*, porém, ressalta-se que o problema apresentado em [3] não havia sido identificado na revisão informal da literatura, a qual reportou problemas relacionados a: (i) demanda de esforço para se obter o *POS*[20,24];(ii) a dificuldade em prever o uso dos software [8]; (iii) modificações constantes do *POS*[12]; (iv) probabilidade de uso, a qual torna-se pequena quando o modelo de uso possui muitos estados que representam as operações [10].

## 10 Ameaças à Validade

Visando diminuir a subjetividade que poderia ocorrer na avaliação dos estudos durante a realização do *RSL*, os resultados parciais, periodicamente, obtidos por um pesquisador (primeiro autor), eram submetidos a outros pesquisadores (demais autores) para a realização de uma análise informal dos resultados obtidos. A qualidade dos estudos selecionados nessas atividades também foi avaliada, porém, dada a abordagem investigativa deste trabalho, não foi definido um valor mínimo a ser obtido pelos estudos selecionados. Os resultados apresentados fazem parte de uma tese de doutoramento com escopo mais amplo [5]. Assim, a *RSL* abordou as situações de uso do *POS* relacionadas ao escopo da tese de doutoramento citada, não abrangendo o uso do *POS* em situações diferentes das investigadas.

## 11 Conclusões

Este artigo apresentou os resultados obtidos por *RSL* que investigou o uso do *POS* sob três perspectivas: (*i*) em atividades associadas ao teste de regressão, (*ii*) como um critério de priorização aplicável ao processo de desenvolvimento de software e (*iii*) como um critério de avaliação para casos de teste existentes. A *RSL* está associada a uma pesquisa de doutoramento que investiga um possível descompasso entre o *POS* e o Perfil de Teste [6] e as contribuições que o *POS* pode proporcionar ao teste de software, almejando melhorar a qualidade do software do ponto de vista dos usuários. Essa pesquisa propõe, como contribuição à Engenharia de Software, especificamente à área de testes, uma estratégia que que permita (*i*) adequar, dinamicamente, um conjunto de testes existente ao *POS* e, também, (*ii*) usar o *POS* como um critério de priorização que permita, dado um conjunto de defeitos, identificar os defeitos que tem maior impacto na operação do software pelos usuários e, assim, considerar esse impacto, juntamente com outros critérios, na precificação dos defeitos.

Os resultados obtidos pela *RSL* proporcionaram respostas às questões de pesquisa que foram definidas, provendo uma base de conhecimento centralizada sobre o uso do *POS* nos contextos investigados e, assim, uma caracterização e o entendimento quanto ao *POS* nas situações de uso investigadas. Verifica-se que o uso do *POS* como um critério de priorização está associado à atividade de teste, sendo usado na priorização de casos de teste. O uso do *POS* como um critério de priorização associado à outros artefatos do processo de desenvolvimento de software não é explorado. Quanto ao uso do *POS* como um critério de avaliação, verificou-se que as técnicas identificadas abordam a geração, seleção e priorização dos casos de teste por meio da análise de cobertura. O uso do *POS* como um critério de avaliação para adequar um conjunto de teste existente, que não tenha sido criado inicialmente por meio do *POS*, também não foi explorado. Assume-se, assim, que as contribuições providas por este artigo são relevantes e fornecem informações que podem contribuir e favorecer os estudos associados à Engenharia de Software e, especificamente, à área de qualidade de software.

## Referências

1. Amrita, Yadav, D.K.: A novel method for allocating software test cases. In: 3rd International Conference on Recent Trends in Computing 2015 (ICRTC-2015). vol. 57, pp. 131 – 138. Elsevier, Delhi, India (2015)
2. Begel, A., Zimmermann, T.: Analyze this! 145 questions for data scientists in software engineering. In: Proceedings of the 36th International Conference on Software Engineering. pp. 12–23. ICSE 2014, ACM, New York, NY, USA (2014)
3. Bertolino, A., Miranda, B., Pietrantuono, R., Russo, S.: Adaptive coverage and operational profile-based testing for reliability improvement. In: Proceedings of the 39th International Conference on Software Engineering. pp. 541–551. ICSE '17, IEEE, Piscataway, NJ, USA (2017)
4. Bittanti, S., Bolzern, P., Scattolini, R.: An introduction to software reliability modelling, chap. 12, pp. 43–67. Springer Berlin Heidelberg, Berlin, Heidelberg (1988)
5. Cavamura Jr, L.: Operational profile and software testing: Aligning user interest and test strategy. In: 2019 12th IEEE Conference on Software Testing, Validation and Verification (ICST). pp. 492–494 (April 2019)
6. Cavamura Jr, L., Morimotto, R., Vincenzi, A.R.M., Fabbri, S.: Operational profile vs. test profile (October 2019)
7. Chen, M., Lyu, M.R., Wong, W.E.: Effect of code coverage on software reliability measurement. IEEE Transactions on Reliability **50**(2), 165–170 (June 2001)
8. Cukic, B., Bastani, F.B.: On reducing the sensitivity of software reliability to variations in the operational profile. pp. 45–54. IEEE, White Plains, NY, USA (1996)

9. Fabbri, S., Silva, C., Hernandez, E., Octaviano, F., Di Thommazo, A., Belgamo, A.: Improvements in the start tool to better support the systematic review process. In: Proc. of the 20th International Conference on Evaluation and Assessment in Software Engineering. EASE '16, ACM, New York, NY, USA (2016)
10. Fukutake, H., Xu, L., Takagi, T., Watanabe, R., Yaegashi, R.: The method to create test suite based on operational profiles for combination test of status. pp. 1–4. Institute of Electrical and Electronics Engineers Inc., White Plains, NY, USA (2015)
11. Kashyap, A.: A Markov Chain and Likelihood-Based Model Approach for Automated Test Case Generation, Validation and Prioritization: Theory and Application. Ph.D. thesis (2013)
12. Leung, Y.W.: Software reliability allocation under an uncertain operational profile. *Journal of the Operational Research Society* **48**(4), 401 – 411 (1997)
13. Marijan, D., Teslic, N., Tekcan, T., Pekovic, V.: Multimedia system verification through a usage model and a black test box. In: The 2010 International Conference on Computer Engineering Systems. pp. 178–182. IEEE, Pasadena, CA, United states (Nov 2010)
14. Miranskyy, A., Cialini, E., Godwin, D.: Selection of customers for operational and usage profiling. Providence, RI (2009)
15. Musa, J.D.: Software reliability measures applied to systems engineering. In: Managing Requirements Knowledge, International Workshop on(AFIPS). vol. 00, p. 941. IEEE, S.I. (12 1979)
16. Musa, J.: Operational profiles in software-reliability engineering. *IEEE* **10**(2), 14–32 (1993)
17. Musa, J.D., Ehrlich, W.: Advances in software reliability engineering. *Advances in Computers*, vol. 42, pp. 77 – 117. Elsevier (1996)
18. Nakagawa, E.Y., Scannavino, K.R.F., Fabbri, S., Ferrari, F.C.: *Revisão Sistemática da Literatura em Engenharia de Software: Teoria e Prática*. Elsevier, Rio de Janeiro, 1rd edn. (2017)
19. Nakornburi, S., Suwannasart, T.: A tool for constrained pairwise test case generation using statistical user profile based prioritization. pp. 1–6. Institute of Electrical and Electronics Engineers Inc., White Plains, NY, USA (2016)
20. Namba, Y., Akimoto, S., Takagi, T.: Overview of graphical operational profiles for generating test cases of gui software. pp. 1–3. Institute of Electrical and Electronics Engineers Inc., White Plains, NY, USA (2015)
21. Rogstad, E., Briand, L.: Cost-effective strategies for the regression testing of database applications: Case study and lessons learned. *Journal of Systems and Software* **113**, 257 – 274 (2016)
22. Sarwar, S., Mahmood, Y., Qayyum, Z.U., Shafi, I.: Test case prioritization for nunit based test plans in agile environment. In: Artificial Intelligence: Methodology, Systems, and Applications. pp. 246–253. AIMS 2014, Cham (2014)
23. Smidts, C., Mutha, C., Rodríguez, M., Gerber, M.J.: Software testing with an operational profile: Op definition. *ACM Comput. Surv.* **46**(3), 39:1–39:39 (Feb 2014)
24. Sommerville, I.: *Software Engineering*. Addison-Wesley, Harlow, England, 9 edn. (2011)
25. Spínola, R.O., Dias-neto, A.C., Travassos, G.H.: Abordagem para desenvolver tecnologia de software com apoio de estudos secundários e primários. In: 5th Experimental Software Engineering Latin American Workshop. ICMC/USP, São Carlos, SP, Brasil (2008)
26. Takagi, T., Furukawa, Z., Yamasaki, T.: An overview and case study of a statistical regression testing method for software maintenance. *Electronics and Communications in Japan (Part II: Electronics)* **90**(12), 23–34 (2007)
27. Whittaker, J., Rekab, K., Thomason, M.: A markov chain model for predicting the reliability of multi-build software. *Information and Software Technology* **42**(12), 889 – 894 (2000)
28. Yoo, S., Harman, M.: Regression testing minimization, selection and prioritization: A survey. *Softw. Test. Verif. Reliab.* **22**(2), 67–120 (Mar 2012)