

# BPMN, SPEM e Essence no contexto da Modelagem de Processos de Software: uma Revisão Sistemática da Literatura

Igor O. da Fonseca<sup>1</sup>, Luiz Guilherme Fritsch<sup>1</sup>, Maicon Bernardino<sup>1</sup>, Fabio Basso<sup>1</sup>

<sup>1</sup> Universidade Federal do Pampa (UNIPAMPA)  
Código Postal 97.546-550 – Alegrete – RS – Brasil

igorfonsecaigor@gmail.com, fritsch.guilherm3@gmail.com,  
bernardino@acm.org, fabiopbasso@gmail.com

**Abstract.** *A software process model is an abstract representation of a process that details its description in some particular perspective. The literature presents some proposals for process modeling, each offering a different view on the process to be modeled. This study proposes a systematic literature review comparing three process specifications: BPMN, SPEM, and Essence. The study seeks to map their characteristics, understand their advantages and disadvantages, identify research trends, and list the support tools for their implementation. Partial results are presented based on the data extracted from 40 articles selected in the review process.*

**Resumo.** *Um modelo de processo de software é uma representação abstrata de um processo que o descreve em alguma perspectiva particular. A literatura apresenta uma variedade de propostas para a modelagem de processos, cada qual oferecendo uma diferente visão sobre o processo a ser modelado. Este estudo apresenta uma revisão sistemática da literatura com o objetivo de comparar três especificações para modelagem de processos de software: BPMN, SPEM e Essence. O estudo busca mapear suas características, entender as vantagens e desvantagens, identificar as tendências de pesquisa, bem como listar o ferramental de suporte na sua implementação. Resultados parciais são apresentados a partir de 40 artigos selecionados no processo de revisão.*

## 1. Introdução

Um processo de software é um conjunto de atividades relacionadas que levam à produção de um produto de software [Sommerville 2011]. Essas atividades descrevem como os projetos de desenvolvimento de software devem ser executados, com a sequência de fases e marcos expressando um ciclo de vida do produto em desenvolvimento. Um modelo de processo de software é uma representação simplificada de um processo de software que apresenta uma descrição dele em alguma perspectiva particular [Curtis et al. 1992]. A modelagem de processos de software, em geral, é utilizada para descrever essas informações parciais do projeto. Por exemplo, o modelo de atividades de um processo pode exibir as atividades e sua sequência, mas não mostrar os perfis dos atores envolvidos – essas últimas informações podem ser representadas em outro tipo de modelo.

Existe uma variedade de propostas (notações, modelos, linguagens, etc.) para a modelagem de processos de software [Kent 2002], cada qual oferecendo uma diferente visão sobre o processo a ser modelado. O BPMN (*Business Process Model and Notation*) [Chinosi and Trombetta 2012], por exemplo, é uma notação para modelagem de

processos focada no negócio da organização, que pode ser especializada para a modelagem de processos de software. O Essence [Jacobson et al. 2012], por sua vez, é uma especificação que propõe suporte para a definição de práticas ágeis e melhor suporte para a execução de métodos [Elvesæter et al. 2013], por meio de um conceito de estados das entidades. Outra proposta, idealizada e desenvolvida pela OMG (*Object Management Group*), é o metamodelo SPEM (*Software & Systems Process Engineering Metamodel*) [Bendraou et al. 2007, Bezivin 2005]. O SPEM trata a modelagem de processos com a definição de conteúdos que formam uma base de conhecimentos, e a posterior instanciação desses conteúdos de forma estruturada para compor um processo.

A modelagem de processos de software é utilizada na área da engenharia de software para planejar, documentar e acompanhar os processos de um modo mais aprimorado [Portela et al. 2012]. Isso facilita a aplicação de conceitos da gestão de projetos, como o controle de atividades, de prazos das atividades, de utilização de métodos e de alocação de recursos humanos. Esse acompanhamento ajuda o gestor de projetos de software a tomar decisões melhores, visto que ele pode ter acesso, por exemplo, aos números de desempenho das atividades executadas pelos membros de suas equipes, entre outros indicadores. Para isso, é importante conhecer as vantagens e limitações das diferentes opções para a modelagem de processos de software.

Este trabalho tem como objetivo realizar uma revisão sistemática da literatura para análise comparativa entre as especificações BPMN, SPEM e Essence para modelagem de processos de software, buscando analisar suas utilizações, vantagens e limitações. A escolha das três especificações de interesse ocorreu pelos seguintes motivos: BPMN, por ser uma notação amplamente adotada para modelagem de processos; SPEM, por ser a notação padrão para a modelagem de processos de software e por ter grande influência no meio acadêmico; e Essence, por ser uma proposta mais contemporânea e possuir um consequente potencial de crescimento e interesse de pesquisa. Vale ressaltar que este é um estudo em andamento, apresentando resultados parciais da pesquisa.

O estudo está organizado conforme segue. A Seção 2 apresenta estudos similares previamente desenvolvidos sobre o tema. A Seção 3 descreve o protocolo da revisão sistemática da literatura. A Seção 4 discute os resultados parciais da análise dos dados extraídos dos estudos selecionados. Por fim, a Seção 5 discorre sobre as conclusões e trabalhos futuros.

## **2. Trabalhos Relacionados**

Ainda que nenhum outro trabalho encontrado na literatura traga uma revisão sistemática da literatura sobre a modelagem de processos de software com BPMN, SPEM e Essence, foram encontrados estudos que trazem uma revisão mais genérica sobre este tipo de ferramenta ou comparações entre duas das notações.

O trabalho de [García-Borgoñon et al. 2014] apresenta um conjunto de linguagens para modelagem de processos definidas entre os anos de 2000 e 2011, identificando os relacionamentos e dependências entre elas por meio de uma revisão sistemática da literatura. Dos 46 estudos incluídos nos resultados do trabalho, foram identificadas 40 linguagens para modelagem de processos de software, cada uma com propósitos específicos. Os autores propõem uma taxonomia para classificação das PMLs (*Process Modeling Languages*) que considera a tecnologia na qual foram baseadas. As três classificações

propostas são linguagens com base em gramáticas, linguagens baseadas em UML e linguagens baseadas em modelos (ou metamodelos). Os resultados do estudo mostram um crescimento mais significativo da utilização de linguagens baseadas em modelo, grupo que inclui o SPEM e suas derivações. Por fim, o estudo aponta três tópicos como direcionamentos para pesquisas futuras: (i) estudos empíricos que coloquem em prática as propostas de PMLs em grandes organizações, (ii) explorar a relação entre as PMLs e os demais campos da engenharia de processos de software (como melhoria de processos de software, simulação e orquestração) e (iii) criar um ambiente integrado no qual linguagens e ferramentas sejam incorporadas, de forma que organizações tenham a capacidade de desenvolver e executar seus processos de software.

Em [Portela et al. 2012], os autores fazem uma análise comparativa entre os padrões BPMN e SPEM quanto às suas notações e elementos no contexto da modelagem de processos de software. Para realizar esta análise, os elementos de cada notação foram mapeados de acordo com sua equivalência com os modelos de qualidade CMMI-DEV (*Capability Maturity Model Integration for Development*) e MR-MPS (*Reference Model for Software Process Improvement*). Isso possibilitou a avaliação por meio de um estudo de caso que modela as melhores práticas desses modelos de qualidade. A avaliação mediou a representatividade alcançada por cada notação, utilizando como critério algumas características consideradas desejadas para a modelagem de processos: expressividade, reuso, gerenciamento, evolução, possibilidade de visão em alto nível e detalhada, facilidade de compreensão e integração organizacional. O estudo concluiu que o SPEM possui vantagem nos critérios de expressividade e gerenciamento, visto que possui uma quantidade maior de elementos básicos, enquanto o BPMN se destaca nos critérios de facilidade de compreensão e integração organizacional.

O estudo de [Elvesæter et al. 2013], por sua vez, compara as especificações do SPEM 2.0 e da versão de esboço do Essence 1.0 para métodos de engenharia de software. A comparação é realizada por meio de um exemplo ilustrativo baseado na prática Scrum, enfatizando os aspectos de capacidade de autoria de métodos e capacidade de execução (*enactment*) do processo em cada especificação. Os autores concluem que as propostas possuem similaridade quanto às capacidades de autoria de métodos. Enquanto ambas fornecem conceitos básicos de notação que permitem definir práticas com elementos de orientação, o Essence destaca-se por conter alguns conceitos diferentes ou propriedades adicionais que permitem um melhor suporte para a execução (*enactment*) do processo.

### **3. Revisão Sistemática da Literatura**

Para investigação da literatura a respeito do tema deste estudo, conduziu-se uma Revisão Sistemática da Literatura (RSL) utilizando as diretrizes propostas por [Kitchenham and Charters 2007]. Uma RSL é útil para pesquisadores e profissionais por fornecer percepções abrangentes sobre o estado da arte em uma área específica.

#### **3.1. Protocolo da Revisão**

As principais atividades da etapa de planejamento da revisão são a definição das questões de pesquisa, da *string* de busca e da estratégia de busca. Para esta revisão sistemática da literatura, foram estabelecidas as seguintes questões de pesquisa:

**RQ1.** Quais especificações para modelagem de processos de software são caracterizadas como executáveis (“*enactable*”)?

- RQ2.** Como as especificações para modelagem de processos de software são avaliadas?  
**RQ3.** Quais ferramentas de suporte implementam as especificações?  
**RQ4.** Quais as vantagens e desvantagens das especificações indicadas pelos estudos?  
**RQ5.** Qual é a tendência atual de pesquisa na área de especificações?

As bases de dados escolhidas para realização das buscas foram ACM, Engineering Village, IEEE e Scopus.

Para a definição da *string* de busca (Figura 1), foram definidos os termos-chave de relevância para a pesquisa, bem como seus sinônimos. O primeiro termo, “*process modeling*”, e seus sinônimos “*process definition*” e “*process description*”, delimitam o escopo maior da pesquisa, que é a área de modelagem de processos. O termo seguinte – “*language*”, “*notation*” ou “*metamodel*” – visa delimitar a área de modelagem de processos, focando em estudos que discutem as diferentes especificações (que podem ser linguagens, notações e metamodelos). Já o último termo – “*bpmn*”, “*spem*” ou “*essence*” – delimita o escopo das três especificações de interesse para a pesquisa. Assim, a consulta genérica gerada a partir destes termos foi a seguinte:

```
('process modeling' OR 'process definition' OR 'process description') AND (language OR notation OR metamodel) AND (bpmn OR spem OR essence)
```

**Figura 1. String de busca.**

Inicialmente, é realizada a exclusão de artigos duplicados – casos em que o mesmo trabalho foi encontrado em mais de uma das bases de dados. A partir do conjunto de trabalhos únicos obtido com a busca nas bases pesquisadas, é necessário realizar uma filtragem para descartar aqueles que não se referem à área de interesse da pesquisa ou não têm potencial para responder às questões de pesquisa. Para isso, foram definidos critérios de inclusão e exclusão. Para que um trabalho seja selecionado para a próxima etapa, deve cumprir todos os critérios de inclusão e nenhum critério de exclusão. Os três critérios de inclusão estão listados a seguir:

- IC1** O trabalho é um estudo primário ou secundário.
- IC2** O trabalho foi publicado a partir de 2008.
- IC3** O trabalho trata do SPEM 2.0, do BPMN 2.0 ou do Essence 1.0 ou versões posteriores.

Os três critérios de exclusão utilizados nesta revisão foram os seguintes:

- EC1** O trabalho tem menos de 6 páginas.
- EC2** O trabalho está em um idioma diferente do inglês.
- EC3** O trabalho não está disponível para *download*.
- EC4** O trabalho não trata do SPEM 2.0, do BPMN 2.0 ou do Essence 1.0 ou versões posteriores.

Com o conjunto de artigos selecionados a partir dos critérios de seleção, é necessário estabelecer também critérios de qualidade para classificar os melhores artigos com base nos fatores elencados. Para isso, é realizada a leitura completa de cada artigo, avaliando os aspectos mostrados na Tabela 1.

**Tabela 1. Critérios de Qualidade**

<b>Critério de Qualidade</b>	<b>Valor</b>	<b>Pontuação</b>
<b>QA1</b> O estudo menciona os principais recursos utilizados da especificações para modelagem de processos de software?	Sim	2.0
	Parcialmente	1.0
	Não	0.0
<b>QA2</b> O estudo apresentou um método de avaliação?	Estudo experimental	1.0
	Estudo empírico	0.8
	Industrial	0.6
	Prova de conceito	0.4
	Teórico	0.2
	Nenhum	0.0
<b>QA3</b> O estudo indica pesquisas futuras na área da especificações para modelagem de processos de software?	Sim	1.0
	Parcialmente	0.5
	Não	0.0
<b>QA4</b> O estudo usa uma ferramenta de suporte para modelar seus processos de software?	Sim	1.0
	Parcialmente	0.5
	Não	0.0

Para o critério QA1, é considerado o valor parcial quando o estudo menciona os principais recursos utilizados, e total quando os descreve detalhadamente. Esse critério é considerado o mais relevante, por isso a ele é atribuído o dobro do valor máximo de pontuação. No critério Q3, o valor parcial é atribuído quando a proposta de pesquisas futuras se restringe ao escopo da própria pesquisa, e o valor total representa uma contribuição para a pesquisa futura na área da especificação. Já no critério QA4, o valor parcial refere-se à situação em que a ferramenta de suporte é citada, e o valor total é atribuído quando ela é descrita.

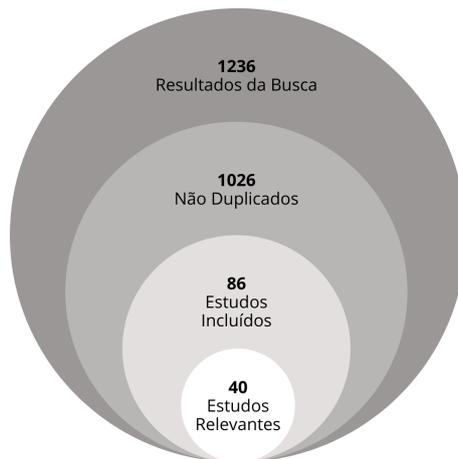
Para que o trabalho seja aceito nos critérios de qualidade, estabelecemos que sua pontuação deve ser maior que 2.0. Finalmente, com o conjunto de trabalhos que atingiram a pontuação mínima, realizamos a leitura completa de cada um deles para a extração dos dados que servirão como base para responder às questões de pesquisa. O conjunto de dados coletados nesta etapa foram: título, autor, ano, tipo de avaliação, detalhes da avaliação, recursos utilizados, pesquisa futura, tipo da ferramenta de suporte, nome da ferramenta de suporte, vantagens reportadas, desvantagens reportadas e descrição da execução (*enactment*).

### **3.2. Condução da Revisão**

A primeira etapa da condução do estudo é a execução da busca em cada uma das bases de dados selecionadas. Para isso, utiliza-se a *string* de busca mostrada na seção anterior, adaptada à sintaxe de cada ferramenta. Nessa etapa, foram obtidos 1236 estudos.

Após a exclusão de 210 artigos duplicados, restaram 1026 trabalhos únicos. Estes trabalhos passaram pela filtragem por meio dos critérios de inclusão e exclusão, em que foram selecionados 86 estudos adequados ao escopo da pesquisa, isto é, que satisfazem todos os critérios de inclusão e nenhum critério de exclusão. Dos trabalhos excluídos nessa etapa, 12 foram removidos devido ao critério de exclusão EC1 (número de páginas), 4 foram removidos devido ao EC2 (idioma), 8 foram removidos devido ao EC3 (disponibilidade para *download*) e 916 devido ao critério EC4 (tema do estudo).

A seguir, foram aplicados os critérios de qualidade ao conjunto de 86 trabalhos



**Figura 2. Quantidade de estudos obtidos em cada etapa da revisão sistemática**

incluídos na etapa anterior. Quarenta trabalhos atenderam aos critérios mínimos de qualidade, sendo utilizados para o processo de extração dos dados. No contexto deste trabalho, utiliza-se o termo “Estudos Relevantes” para referenciar o conjunto dos 40 trabalhos utilizados na extração dos dados. A Figura 2 mostra a quantidade de estudos obtidos em cada etapa desta revisão. A Tabela 2 apresenta as principais informações (autores, título e ano de publicação) dos Estudos Relevantes. Devido à limitação de espaço, optou-se por apresentá-las em forma de tabela ao invés de utilizar a seção de referências.

#### **4. Resultados Parciais**

Dos 40 estudos relevantes selecionados na revisão, um deles trata ao mesmo tempo das especificações SPEM e BPMN, enquanto 16 apresentam apenas o metamodelo SPEM, 20 são sobre o BPMN e três tratam do Essence. A Figura 3 mostra a distribuição dos estudos de acordo com o ano de publicação e a especificação apresentada. A partir dela, nota-se um maior interesse em estudos sobre o metamodelo SPEM nos primeiros anos após o lançamento da versão 2.0 em 2008, e a diminuição dessa quantidade após 2012. O BPMN, por sua vez, apresenta um maior número de estudos mais recentes, a partir de 2015. O Essence, sendo o padrão mais recente e não tão consolidada, mostra uma pequena quantidade de estudos entre 2015 e 2019.

##### **4.1. RQ1: Quais especificações para modelagem de processos de software são caracterizadas como executáveis (“enactable”)?**

Apenas nove (9) artigos relevantes discutem execução dos processos de software modelados. Um refere-se ao BPMN, seis (6) são sobre o SPEM e dois (2) tratam do Essence.

Segundo o trabalho com ID 26 (Tabela 2), com a publicação do BPMN 2.0 em janeiro de 2011, o suporte para a execução nativa de modelos de processo BPMN e um formato de serialização padronizado foi adicionado à especificação. Os autores ressaltam que a implementação da semântica de execução especificada precisa ser viável, o que infelizmente não pode ser garantido pela discussão puramente teórica e informal da semântica de execução no padrão BPMN 2.0. Isso indica que a maioria das ferramentas atuais que dizem implementar o BPMN 2.0 é limitada à visualização e não implementa realmente a semântica de execução do BPMN 2.0. De acordo com a especificação do

**Tabela 2. Lista de Artigos Relevantes**

ID	Autor — Título (Ano de Publicação)
1	E. Nardini, et al. — SPEM on Test: The SODA Case Study (2008)
2	Samba Diaw, et al. — Using the SPEM 2.0 Kind-Based Extension Mechanism to Define the SPEM4MDE Metamodel (2017)
3	Pascal Poizat and Gwen Salaün — Checking the Realizability of BPMN 2.0 Choreographies (2012)
4	Yousra Odeh — BPMN in Engineering Software Requirements: An Introductory Brief Guide (2017)
5	Vanessa Mendoza, et al. — Verifying BPMN Understandability with Novice Business Managers (2018)
6	Muhammad A. Javed and Barbara Gallina — Safety-Oriented Process Line Engineering via Seamless Integration between EPF Composer and BVR Tool (2018)
7	June Sung Park — Essence-Based, Goal-Driven Adaptive Software Engineering (2015)
8	Sami Alajrami, et al. — EXE-SPEM: Towards cloud-based executable software process models (2016)
9	Richard Braun — Meta model extensibility of BPMN: Current limitations and proposed improvements (2015)
10	Bilal Maqbool, et al. — A comprehensive investigation of BPMN models generation from textual requirements techniques, tools and trends (2019)
11	Dario Campagna, et al. — BPMN 2.0 and the service interaction patterns: Can we support them all? (2015)
12	Giancarlo Guizzardi and Gerd Wagner — Can BPMN be used for making simulation models? (2011)
13	G. Baumgarten, et al. — SPEM 2.0 as process baseline Meta-Model for the development and optimization of complex embedded systems (2015)
14	P. V. Martins and A. R. da Silva — PIT-ProcessM: A Software Process Improvement Meta-Model (2010)
15	N. Kerzazi and P. Robillard — Multi-perspective Software Process Modeling (2010)
16	R. S. P. Maciel, et al. — An Integrated Approach for Model Driven Process Modeling and Enactment (2009)
17	V. Řepa and O. Ždelezník — Methodological limitations of modeling languages BPMN and ARIS (2014)
18	I. Zafar, et al. — A Novel Framework to Automatically Generate Executable Web Services From BPMN Models (2019)
19	K. A. Kedji, et al. — Towards a Tool-Supported Approach for Collaborative Process Modeling and Enactment (2011)
20	F. Aoussat, et al. — SPEM Extension with Software Process Architectural Concepts (2011)
21	Iqra Zafar, et al. — Business Process Models to Web Services Generation: A Systematic Literature Review (2018)
22	Yani Widyani, et al. — Essence-based Method Chunk Metamodel (2019)
23	Christopher C. Andreas and Yani Widyani — Essence-based Method Chunk Composition (2019)
24	P. H. Meland and E. A. Gjære; — Representing Threats in BPMN 2.0 (2012)
25	Matthias Geiger, et al. — BPMN Conformance in Open Source Engines (2015)
26	Matthias Geiger, et al. — BPMN 2.0: The state of support and implementation (2018)
27	Francisco Durán, et al. — Symbolic specification and verification of data-aware BPMN processes using rewriting modulo SMT (2018)
28	Henrik Leopold, et al. — Learning from quality issues of BPMN models from industry (2016)
29	Peter Bollen — Business process model semantics in BPMN (2016)
30	Flavio Corradini, et al. — An operational semantics of BPMN collaboration (2016)
31	Richard Braun — Behind the scenes of the BPMN extension mechanism: Principles, problems and options for improvement (2015)
32	Shan Nan, et al. — DCCSS: A meta-model for dynamic clinical checklist support systems (2015)
33	Raquel M. Pillat, et al. — BPMNt: A BPMN extension for specifying software process tailoring (2015)
34	J.E. Fernandes and R.J. Machado — SPEM 2.0 extension for pervasive information systems (2012)
35	Miroslav Líška and Pavol Navrat — SPEM Ontology as the semantic notation for method and process definition in the context of SWEBOK (2011)
36	D. Rodriguez, et al. — Defining software process model constraints with rules using OWL and SWRL (2010)
37	Oksana Nikiforova and Natalja Pavlova — Application of BPMN instead of GRAPES for two-hemisphere model driven approach (2010)
38	M. Perez Cotac, et al. — Transformations from SPEM work sequences to BPMN sequence flows for the automation of software development process (2010)
39	Daniel Rodríguez and Miguel-Ángel Sicilia — Defining SPEM 2 process constraints with semantic rules using SWRL (2009)
40	O. M. Rodríguez-Elias, et al. — Modeling and analysis of knowledge flows in software processes through the extension of the software process engineering metamodel (2009)

BPMN 2.0, a ferramenta em “conformidade de execução BPMN ” deve suportar e interpretar totalmente a semântica operacional e o ciclo de vida da atividade especificada.

Nativamente, a especificação do SPEM não oferece suporte à execução dos processos modelados. A partir disso, diversos estudos propõem adaptações para torná-lo compatível com esta finalidade. O trabalho com ID 2 indica que a especificação do SPEM sugere duas alternativas para a execução de processos: mapear os modelos de processo em planos de projeto ou vincular elementos de processo com formalizações externas de comportamento. Os autores listam diversas dessas propostas de extensões, com base em



**Figura 3. Distribuição dos estudos relevantes por ano de publicação e especificação apresentada**

máquinas de estados, redes de Petri e fluxos de trabalho. O estudo de ID 8 propõe a extensão EXE-SPEM para execução de processo de software com base em nuvem. O trabalho com ID 15 propõe uma automatização para a modelagem de processos de software com o SPEM, consistindo em uma extensão focada nas relações entre os elementos do modelo para melhorar sua semântica, incluindo o suporte à execução do processo com reordenação das atividades de acordo com o fluxo desejado. Já a pesquisa de ID 16 busca tornar possível a execução de modelos de processo de software para a proposta MDA (*Model Driven Architecture*) usando o SPEM como notação padrão. Para isso, um metamodelo estende a especificação do SPEM especializando-se ao contexto da MDA. O estudo de ID 19 propõe uma ferramenta para gerenciamento de projetos para a execução de processos modelados colaborativamente. A pesquisa com o ID 34 também aponta diversas extensões propostas para habilitar a execução automática de processos com SPEM.

O Essence fundamentalmente dá suporte à execução de processos modelados de acordo com a especificação. Segundo o autor do trabalho de ID 7, que trás uma proposta de engenharia de software adaptativa orientada a objetivos, o Essence suporta a execução de métodos deste tipo de processo devido à sua definição de elementos do *kernel* e seus estados. Um método adaptativo permite que uma equipe de projeto mude seu plano de trabalho e até mesmo o método de trabalho a qualquer momento com base na avaliação do estado atual do projeto. O trabalho com ID 23 propõe um processo para a composição de métodos de desenvolvimento de software baseados no Essence. Segundo o autor, os métodos gerados podem ser usados para configurar uma ferramenta CASE<sup>1</sup> para dar suporte à execução do método durante o processo de engenharia de software.

#### **4.2. RQ2: Como as especificações para modelagem de processos de software são avaliadas?**

Dos trabalhos selecionados, dez (10) apresentam um estudo experimental. As propostas incluem novas técnicas ou ferramentas aplicadas a estudos de caso/exemplos ou aplicações da especificação e suas extensões a projetos reais. Em relação aos estudos empíricos, doze (12) trabalhos propõem a aplicação da especificação e suas extensões a exemplos ou

<sup>1</sup>Ferramenta CASE (*Computer-Aided Software Engineering*): classificação que abrange todas as ferramentas baseadas em computadores que auxiliam atividades de engenharia de software.

avaliação delas por meio de testes. Os estudos industriais, nos quais a especificação é aplicada a um caso da indústria, são mostrados em dois (2) trabalhos. Quatro (4) trabalhos relevantes são apresentados como provas de conceito, em que novas técnicas/ferramentas ou aplicações da especificação são apresentadas e discutidas. Por fim, onze (11) dos trabalhos apresentam uma avaliação teórica, que incluem revisões sistemáticas da literatura, discussões de funcionalidades ou avaliações qualitativas. A Tabela 3 apresenta um compilado dos métodos de avaliações utilizados por cada estudo.

**Tabela 3. Avaliação das especificações**

ID do Estudo	Tipo de Avaliação
3, 6, 18, 22, 23	Uma nova técnica/ferramenta utilizando a especificação é avaliada por meio de estudo(s) de caso/exemplos.
32, 33, 34	Uma extensão da especificação é avaliada aplicando-a a um projeto real.
16	Uma extensão da especificação é avaliada aplicando-a a um projeto real e um exemplo, além de avaliação qualitativa pelos usuários.
2, 8, 9, 15, 27, 30, 35, 36, 39, 40	Uma extensão da especificação é avaliada aplicando-a a um exemplo.
1, 24	A especificação é avaliada aplicando-a a um ou mais exemplos.
25, 26	As ferramentas de suporte à especificação são avaliadas por meio de testes.
13, 19	Uma extensão da especificação é avaliada por meio de um estudo de caso industrial.
7, 37, 38	Uma nova técnica/ferramenta utilizando a especificação é apresentada, discutindo suas vantagens e aplicações.
20	Uma extensão da especificação é apresentada, discutindo suas vantagens e aplicações.
10, 21	A especificação é avaliada por meio de uma revisão sistemática da literatura.
5	A especificação é avaliada qualitativamente pelos usuários.
4, 11, 12, 14, 17, 29, 28, 31	Uma discussão sobre as características da especificação é apresentada.

A partir das avaliações realizadas nos estudos, cabe destacar que dezessete (17) delas não referem-se apenas às especificações originais, mas a uma extensões delas. Dessas, cinco (5) são extensões para o BPMN e doze (12) para o SPEM. As extensões, em geral, são utilizadas como recurso para adicionar as especificidades de um domínio à notação original, mais generalista. Em especial, nota-se que várias extensões adicionam elementos de estruturação semântica a partir das especificações do BPMN (IDs 27 e 30) e do SPEM (IDs 8, 15, 19, 35, 36 e 39). Isso reforça as informações encontradas na questão anterior (RQ1), demonstrando que a validação e execução dos processos modelados é uma característica de grande importância em uma parte significativa dos estudos e que esse aspecto é uma limitação nas especificações originais.

## 5. Considerações Finais

Este trabalho teve como objetivo realizar uma revisão sistemática da literatura sobre as especificações de modelagem de processos de software BPMN, SPEM e Essence. Este objetivo foi parcialmente atingido, visto que os resultados ainda são parciais e a revisão está em andamento. Foram elaboradas as atividades de planejamento, pesquisa, seleção, avaliação de qualidade e extração dos dados, e a partir dos dados extraídos foram respondidas duas de cinco questões de pesquisa como resultados parciais.

A partir dos resultados obtidos, é possível identificar algumas vantagens e limitações das especificações, como a capacidade de validação e execução das especificações,

relevante para contextos onde a automatização dos processos é importante. Nota-se que a versão atual da especificação BPMN suporta a execução nativa de processos, porém nem todas as ferramentas a implementam corretamente, enquanto o SPEM não oferece esse suporte nativo, apesar de algumas extensões terem sido desenvolvidas para auxiliar nesse sentido. O Essence oferece suporte a esse tipo de execução e acompanhamento, visto que é uma de suas características fundamentais. Além disso, foram levantados dados quanto ao modo como as especificações são utilizadas e avaliadas, muitas vezes por meio de extensões à especificação original.

Com a continuação da pesquisa, em trabalhos futuros será ampliada a análise sobre os dados extraídos para responder às demais questões de pesquisa. Assim, realizaremos um levantamento mais amplo das vantagens e limitações de cada especificação, conforme apontado pelos autores dos estudos, incluindo outros aspectos como as ferramentas de suporte utilizadas e a identificação de pontos de interesse para pesquisas futuras na área.

## Referências

- Bendraou, R., Combemale, B., Cregut, X., and Gervais, M.-P. (2007). Definition of an executable SPEM 2.0. In *14th Asia-Pacific Software Engineering Conference (APSEC'07)*, pages 390–397.
- Bezivin, J. (2005). On the unification power of models. *Softw. Syst. Model.*, 4(2):171–188.
- Chinosi, M. and Trombetta, A. (2012). Bpmn: An introduction to the standard. *Computer Standards & Interfaces*, 34(1):124–134.
- Curtis, B., Kellner, M. I., and Over, J. (1992). Process modeling. *Communication ACM*, 35(9):75—90.
- Elvesæter, B., Benguria, G., and Ilieva, S. (2013). A comparison of the essence 1.0 and SPEM 2.0 specifications for software engineering methods. In *Proceedings of the Third Workshop on Process-Based Approaches for Model-Driven Engineering*, pages 1–10.
- García-Borgoñon, L., Barcelona, M. A., García-García, J. A., Alba, M., and Escalona, M. J. (2014). Software process modeling languages: A systematic literature review. *Information and Software Technology*, 56(2):103–116.
- Jacobson, I., Ng, P.-W., McMahan, P., Spence, I., and Lidman, S. (2012). The essence of software engineering: The semat kernel: A thinking framework in the form of an actionable kernel. *Queue*, 10(10):40—51.
- Kent, S. (2002). Model driven engineering. In *Integrated Formal Methods*, pages 286–298, Berlin, Heidelberg. Springer.
- Kitchenham, B. and Charters, S. (2007). Guidelines for performing systematic literature reviews in software engineering. Technical report, Technical Report EBSE 2007-001, Keele University and Durham University Joint Report.
- Portela, C., Vasconcelos, A., Sinimbú, A., Silva, E., Ronny, M., Lira, W., Oliveira, S., et al. (2012). A comparative analysis between BPMN and SPEM modeling standards in the software processes context. *Journal of Software Engineering and Applications*, 5(5).
- Sommerville, I. (2011). *Engenharia de software*. Pearson Prentice Hall.