

Software-Based Assets Maturity Level (SML) – Um Modelo de Avaliação de Maturidade para Software-Based Assets

Ivo Simões¹, Maria Bernardo², Pedro Isaías³

¹Universidade Aberta de Portugal, Lisboa, Portugal, 1904175@estudante.uab.pt

²Universidade Aberta de Portugal, Lisboa, Portugal e INESCTEC, maria.bernardo@uab.pt

³Universidade Aberta de Portugal, Lisboa, Portugal e CSG – Investigação em Ciências Sociais e Gestão - ISEG, pedro.isaias@uab.pt

Resumo

Este artigo apresenta a escala *Software-Based Assets Maturity Level (SML)*, concebida para medir a maturidade dos *Software-Based Assets* (SBAs). Esta escala procura preencher a ausência de modelos adequados à sua natureza particular. Metodologicamente, assenta numa revisão crítica da literatura e normas ISO/IEC e IEEE, de que resulta uma escala com três níveis e oito estágios, suportada por critérios verificáveis e descriptores evolutivos. Define-se um plano de validação com consulta a especialistas, estudos de caso múltiplos e avaliação quantitativa da aplicabilidade. O contributo central integra dimensões de prontidão tecnológica, qualidade e governação, permitindo avaliações comparáveis e comunicáveis de SBAs em vários contextos organizacionais. Pretende-se que a adoção do modelo reforce a consistência da medição, a melhoria da comunicação entre equipas e gestores e sustente decisões de investimento alinhadas com o valor organizacional.

Palavras-chave: *Software-Based Assets*, Maturidade de Software, Gestão de Software, Governação de TI, Modelos de Avaliação

Title: *Software-Based Assets Maturity Level (SML) - A Maturity Assessment Model for Software-Based Assets*

Abstract: This article introduces the Software-Based Assets Maturity Level (SML) scale, designed to measure the maturity of Software-Based Assets (SBAs). The scale aims to address the lack of models suited to their nature. Methodologically, it is based on a critical review of the literature and ISO/IEC and IEEE standards, resulting in a three-level, eight-stage scale supported by verifiable criteria and evolutionary descriptors. A validation plan is defined, involving expert consultation, multiple case studies, and quantitative assessment of applicability. The main contribution integrates dimensions of technological readiness, quality, and governance, enabling comparable and communicable evaluations of SBAs across various organisational contexts. The model is intended to enhance measurement consistency, improve communication between teams and managers, and support investment decisions aligned with organisational value.

Keywords: Software-Based Assets, Software Maturity, Software Management, IT Governance, Assessment Models

1. Introdução

A industrialização do software promove uma transformação na forma como o seu valor é criado, medido e sustentado. À medida que os processos de digitalização e automação se intensificam, o software deixou de ser apenas um produto tecnológico para assumir o papel de recurso estratégico, ou um *software asset*, cuja gestão eficiente torna-se determinante para a competitividade e inovação organizacional.

Entretanto, apesar da existência de normas internacionais para gestão, medição e qualidade de recursos tecnológicos, como a ISO 16290:2013 e ISO/IEC 19770-1:2017, subsistem lacunas significativas na forma como se avalia a maturidade e o valor destes recursos.

A ausência de uma taxonomia padronizada e de modelos de medição adaptados à natureza intangível e iterativa do *software asset* compromete a sua governação e dificulta decisões de investimento baseadas em evidência. Neste contexto, o conceito de *Software-Based Asset* (SBA) emerge como uma proposta de categorização e avaliação dentro do universo dos *software assets*, mas carece de escalas de medição e maturidade adaptadas à sua natureza.

O presente artigo propõe, assim, um enquadramento teórico e metodológico para compreender, classificar e medir a maturidade dos SBAs, através da introdução da escala *Software-Based Assets Maturity Level* (SML), concebida para preencher as lacunas identificadas na literatura e promover uma avaliação objetiva, comunicável e alinhada com o valor organizacional.

1.1 Problema

A integração crescente do software na infraestrutura organizacional requer abordagens estruturadas, que possibilitem medir o seu nível de desenvolvimento e a sua capacidade de operação. Contudo, os modelos existentes de avaliação de maturidade, como o *Technology Readiness Level* (TRL) e o *Manufacturing Readiness Level* (MRL), foram concebidos para contextos tangíveis, não captando as particularidades intangíveis, iterativas e evolutivas do software.

Esta lacuna compromete a comparabilidade entre projetos, dificulta decisões de investimento baseadas em evidência e limita a governação tecnológica de SBAs.

1.2 Objetivos

O objetivo principal deste trabalho é propor o modelo Software-Based Assets Maturity Level (SML), uma escala estruturada para medir a maturidade de SBAs de forma objetiva, verificável e comunicável. Especificamente, pretende-se:

- identificar as limitações dos modelos de maturidade existentes aplicados ao software;
- conceber uma escala adaptada às características dos SBAs, integrando dimensões de maturidade/prontidão, qualidade e governação; e
- delinear um plano de validação que comprove a aplicabilidade e utilidade do modelo em contextos organizacionais reais.

1.3 Estrutura do trabalho

O artigo organiza-se da seguinte forma. Na secção 2, apresenta-se a revisão da literatura sobre *software assets* e modelos de maturidade tecnológica.

Também introduz o conceito de SBA e analisa as limitações dos modelos existentes. Na secção 3, é descrita a proposta da escala SML, com os seus níveis, estágios e critérios de avaliação. A secção 4 detalha a metodologia de validação proposta.

Na secção 5, são discutidas as principais implicações e contributos do modelo, e, por fim, na secção 6, apresentam-se as conclusões e perspetivas de investigação futura.

2. Revisão de Literatura

Os *software assets* são recursos intangíveis que detêm valor económico, operacional e estratégico dentro de uma organização. De acordo com a ISO/IEC 19770-1:2017, revisada em 2024, estes recursos são componentes de um sistema de gestão de recursos de TI (*IT Asset Management*), cuja finalidade é maximizar o valor do software ao longo do seu ciclo de vida, garantindo conformidade, eficiência e governação [ISO/IEC 2017/2024].

A norma ISO 55000:2024 amplia esta visão ao definir “recurso” como qualquer entidade que possua valor real ou potencial, implicando que o software deve ser gerido com o mesmo rigor aplicado aos recursos físicos [ISO 2024]. Por sua vez, a ISO/IEC/IEEE 24765:2017, revisada em 2022, descreve *software asset* como qualquer elemento de software reutilizável, documentado e controlado, incorporando código, bibliotecas, componentes, especificações ou documentação técnica [IEEE 2017/2022].

Assim, estes recursos não se limitam a artefactos técnicos, mas a entidades com valor operacional e estratégico, cuja gestão influencia diretamente a capacidade de inovação e sustentabilidade das organizações [Karimi & Zand 1998; Zabardast 2022].

Complementarmente, a literatura tem sublinhado a necessidade de instrumentos de medição e maturidade específicos para este domínio, uma vez que a avaliação sistemática da prontidão e evolução dos *software assets* é um imperativo para alinhar decisões de investimento, governação e valor organizacional [Héder 2017; Yfanti & Sakkas 2024; Zabardast et al. 2022, 2023].

Modelos canónicos e de referência como os TRLs [ISO, 2013/2024] oferecem uma linguagem comum para maturidade tecnológica, mas foram concebidos para contextos predominantemente tangíveis. Autores como [Çalışkanbaş 2019] e [Chandra 2009] propõem abordagens alternativas orientadas ao software, com a proposta de suprir lacunas neste âmbito, reforçando que a medição da maturidade em software requer critérios e taxonomias adaptados à sua natureza intangível, iterativa e evolutiva [Héder, 2017; Yfanti & Sakkas 2024; Zabardast 2023].

2.1 O que é um Software-Based Asset?

Um SBA corresponde à proposta de uma categoria distinta de *software assets*, definida como qualquer artefacto de software concebido, desenvolvido ou mantido com o propósito de gerar valor organizacional em múltiplas dimensões. Este valor pode

materializar-se em inovação tecnológica, diferenciação estratégica, eficiência operacional, produção de propriedade intelectual ou mesmo na viabilização de novas soluções para problemas identificados [Simões et al. 2025].

Ao contrário de um simples produto de software, o SBA não se limita ao código-fonte ou à sua execução funcional. Inclui também os elementos associados ao seu ciclo de vida, tais como documentação, modelos, protótipos, especificações, registos de conceção e até a própria ideia formalmente registada que deu origem ao artefacto [Simões 2023; Simões et al. 2025].

O seu caráter intangível e evolutivo destaca-se pela capacidade de ser reutilizado, adaptado e integrado em diferentes domínios de conhecimento, ultrapassando a fronteira da engenharia de software e estendendo-se a contextos industriais, científicos e organizacionais.

Importa sublinhar que o conceito de software não se limita ao código executável, mas inclui também os dados utilizados pelo código, a documentação que orienta o seu uso e manutenção, bem como todos os materiais necessários à produção, operação e eventual descontinuação de um sistema [IEEE 2017/2024].

Logo, um SBA pode assumir múltiplas formas, desde um código-fonte, modelos de arquitetura, protótipos em *software*, ou mesmo artigos que fundamentem soluções de base digital [Simões 2023; Simões et al. 2025]. O traço comum é a sua identificação como recurso intangível, concebido para satisfazer um objetivo desejável no âmbito da engenharia de software, podendo ainda ser aplicável a outros domínios científicos ou organizacionais.

Assim, os SBAs distinguem-se de outros *software assets* por não serem apenas ferramentas técnicas, mas instrumentos de criação de valor sustentado, cuja gestão requer modelos próprios de classificação, medição e maturidade [Simões et al. 2025].

2.2 Modelos de maturidade e adaptação ao domínio do software

Os modelos de avaliação da maturidade tecnológica, como os TRLs, definidos na norma ISO 16290:2013, e os MRL, foram originalmente concebidos para contextos de hardware e de manufatura, onde os processos são tangíveis, lineares e relativamente estáveis.

A sua aplicação direta ao domínio do software revela limitações significativas, dado que o software possui natureza intangível, altamente mutável e evolutiva, sendo desenvolvido de forma iterativa e incremental [Department of Defense [DoD] 2025; ISO, 2013/2024; Simões et al. 2025; Wheeler & Ulsh 2010].

2.3 O modelo TRL

O modelo TRL surgiu na NASA, na década de 1970, como instrumento para aferir a maturidade de tecnologias aeroespaciais emergentes, estruturando o progresso em nove níveis que vão desde a observação de princípios básicos (TRL 1) até à comprovação de operação em ambiente real (TRL 9) [Mankins 1995; Yfanti & Sakkas 2024].

A sua adoção disseminou-se rapidamente para outros setores industriais e científicos, tornando-se uma linguagem comum na gestão de R&D, em programas de financiamento e em políticas públicas de inovação tecnológica [Héder 2017]. A padronização

internacional foi consolidada com a ISO 16290:2013, a qual formalizou os critérios de avaliação de cada nível [ISO 2013/2024].

A escala apresentada pelos TRLs destaca-se pela sua simplicidade conceptual, a comparabilidade entre projetos e a utilidade estratégica na redução de riscos de transição tecnológica [Mankins 2009; Olechowski et al. 2020].

Contudo, a literatura reconhece que o modelo possui limitações significativas quando aplicado a domínios imateriais, como software ou serviços digitais. A sua estrutura linear não capta adequadamente os processos iterativos de desenvolvimento, a mutabilidade inerente ao software, nem integra métricas robustas para prototipagem contínua, validação incremental ou operacionalização em ecossistemas digitais complexos [Héder 2017; Yfanti & Sakkas 2024].

Conforme [Lavin et al. 2022], estes limites têm impulsionado propostas de extensões ou adaptações do TRL. No domínio da inteligência artificial, por exemplo, foram sugeridos modelos que incorporam requisitos de *data readiness* (prontidão de dados) e modos de falha específicos.

Do mesmo modo, em contextos colaborativos e de co-criação, têm sido discutidas adaptações que adicionam novos níveis ou caminhos paralelos de maturidade para refletir melhor processos inovadores distribuídos [Yfanti & Sakkas 2024].

Estas abordagens evidenciam que, embora o TRL permaneça uma referência global consolidada, a avaliação da maturidade de SBAs exige escalas complementares capazes de capturar as dimensões dinâmicas e intangíveis do desenvolvimento de software.

2.4 Limitações identificadas na medição de maturidade em Software-Based Assets

A revisão da literatura evidencia que, apesar dos avanços na definição de normas internacionais e de modelos de maturidade tecnológica, subsistem limitações estruturais quando se pretende avaliar a maturidade e prontidão (*readiness*) de SBAs.

Os modelos mais utilizados, como o TRL [ISO 2013/2024] ou o MRL [DoD 2025] foram concebidos nomeadamente para contextos de hardware e manufatura, onde os processos são lineares e estáveis. A transposição direta destes referenciais para o software mostra-se insuficiente, dada a natureza intangível, evolutiva e iterativa deste domínio [Héder 2017; Yfanti & Sakkas 2024].

Uma primeira lacuna para os SBAs refere-se à ausência de um modelo de maturidade específico à sua natureza de software, que seja ao mesmo tempo simples, aplicável e capaz de facilitar a comunicação entre técnicos e gestores.

Embora modelos como o SPL-CMM [Çalışkanbaş 2019] ou o SAMM [Chandra 2009] ofereçam contribuições relevantes, estes não capturam adequadamente o valor estratégico e multidimensional dos SBAs.

Em segundo lugar, destaca-se uma falta de uma taxonomia padronizada de *software assets* que possa servir de base para a definição de critérios de avaliação [Zabardast et al. 2022, 2023].

Apesar de existirem esforços normativos, como a ISO/IEC/IEEE 24765:2017 (vocabulário) ou a ISO/IEC 19770-1:2017 (gestão de *ativos de TI*), ainda não há consenso sobre uma classificação aplicável aos SBAs, em particular no que toca às suas características diferenciadas, como reutilização, documentação evolutiva e intangibilidade [IEEE 2017/2022; ISO/IEC 2017/2024].

Uma terceira limitação prende-se com a escassez de validação empírica em ambientes industriais, necessária para comprovar a aplicabilidade e relevância de escalas de maturidade propostas. Como mostram [Cortiñas et al. 2023], apenas estudos de caso industriais conseguem revelar o impacto real das metodologias na continuidade de *software assets*, sobretudo em cenários de substituição ou obsolescência de linhas de produto.

Para além destas lacunas centrais, a literatura identifica ainda outros desafios como a linearidade excessiva dos modelos herdados, que não refletem ciclos iterativos de desenvolvimento de software [Héder 2017].

A ausência de métricas objetivas e quantificáveis para atributos como prototipagem rápida, documentação dinâmica ou operacionalização em ecossistemas digitais [Lavin et al. 2022]; e as dificuldades na modelação de variabilidades complexas em SPLs, que comprometem a escalabilidade de avaliações de maturidade [Horcas et al. 2023]. A Tabela 1 sintetiza e agrupa estas lacunas verificadas na revisão da literatura:

Tabela 1. Principais limitações na avaliação de maturidade para SBAs

Limitação	Descrição	Referências
Ausência de modelo específico para SBAs	A inexistência de uma escala simples, dedicada a SBAs, que seja compreensível para técnicos e gestores.	[Çalışkanbaş, 2019]; [Chandra, 2009]
Falta de taxonomia padronizada	A ausência de classificação uniforme que sirva de base a critérios de avaliação e métricas consistentes.	[IEEE, 2017/2022]; [ISO/IEC, 2017/2024]; [Zabardast et al., 2022, 2023]
Escassez de validação empírica	Poucos estudos industriais testam modelos de maturidade em cenários reais, limitando sua aplicabilidade.	[Cortiñas et al., 2023]
Linearidade excessiva dos modelos	Modelos como o TRL assumem progressão linear, desajustada ao caráter iterativo e evolutivo do software.	[Héder, 2017]; [Yfanti e Sakkas, 2024]
Falta de indicadores objetivos de análise	A inexistência de indicadores robustos para avaliar prototipagem, validação incremental ou documentação dinâmica.	[Lavin et al., 2022]
Gestão de variabilidade complexa	Os SPLs enfrentam dificuldades em modelar e avaliar variabilidade avançada e atributos não funcionais.	[Horcas et al., 2023]

Neste cenário, emerge a proposta de um modelo alternativo e complementar, o SML, que responda a estas lacunas, integrando critérios normativos, métricas objetivas e validação prática. O SML procura, assim, posicionar-se como proposta inovadora para medir a maturidade dos SBAs, oferecendo uma escala capaz de articular eficiência técnica, valor organizacional e governação tecnológica [Simões 2023; Simões et al. 2025].

Deste modo, a secção estabeleceu as bases conceptuais e metodológicas que sustentam a formulação do modelo SML, preparando o terreno para a sua apresentação formal e discussão nas secções seguintes.

3. Proposta: escala SML para SBAs

A escala SML tem como propósito tornar o processo de avaliação da maturidade mais objetivo e transparente, ao disponibilizar um método estruturado e normalizado para medir SBAs segundo critérios verificáveis e adaptados às condições reais de desenvolvimento e uso.

Esta sistematização procura assegurar maior consistência na avaliação e promover uma linguagem comum entre diferentes *stakeholders*, facilitando a comunicação e reduzindo interpretações divergentes. A escala, ao fornecer orientações claras e comparáveis, fortalece a governação tecnológica, apoia decisões estratégicas fundamentadas, contribuindo para a governação de TI [ISO/IEC 2024] e para que a gestão de SBAs seja mais eficiente, alinhada e orientada para a geração de valor organizacional.

Em última análise, uma classificação clara pode contribuir para operações mais eficientes, mitigação de riscos e melhor alinhamento dos SBAs com os objetivos organizacionais [ISO/IEC 2024]. A escala SML apresenta um modelo estruturado em níveis sequenciais, com critérios, descritores e limiares de cobertura.

A conceção do SML inspira-se em definições já consolidadas por normas internacionais [IEEE 2017; ISO 2013/2024], mas introduz adaptações fundamentais para refletir as características específicas dos SBAs.

Tal adaptação implica segmentar os níveis de forma mais apropriada à natureza intangível e iterativa do software, clarificando a finalidade distinta de cada nível e a progressão lógica entre eles.

O SML foi projetado com os seguintes princípios:

- a) Simplicidade: visa assegurar que os níveis da escala sejam facilmente compreensíveis por *stakeholders* técnicos e não-técnicos;
- b) Especificidade: estrutura os seus critérios adaptados à natureza dos SBAs;
- c) Granularidade: assenta numa combinação de níveis e estágios intermédios para distinguir progressões relevantes.

3.1 Estrutura da escala

A escala SML organiza-se em três níveis principais chamados Criação, Evolução e Finalizado, que refletem a progressão natural de um SBA desde a sua conceção inicial até à operacionalização plena. Cada nível é subdividido em estágios, permitindo capturar nuances do processo de desenvolvimento. Para fins de notação, adota-se o formato “N.S”, em que “N” representa o nível e “S” o estágio (por exemplo, 1.3 = Criação. Materialização). Esta granularidade facilita a avaliação precisa e comparável da maturidade de um SBA [Simões 2023].

A estrutura sequencial da escala assegura a progressão lógica entre estágios, evitando saltos arbitrários de maturidade e reforçando a rastreabilidade do desenvolvimento. Além disso, a subdivisão em três níveis e oito estágios, apresentada na Tabela 2, procura acomodar as especificidades intangíveis, iterativas e adaptativas dos SBAs. Assim, o SML estrutura-se para fornecer uma métrica clara de maturidade, e também pode funcionar como instrumento de comunicação entre equipas técnicas, gestão e decisores

estratégicos, promovendo alinhamento organizacional e apoio a decisões de investimento e governação tecnológica [ISO/IEC 2024].

Tabela 2. Estrutura da Escala SML

Nível	Nome do nível	Estágio (Nº e Nome)	Critérios de Definição	Descrição Complementar	Âmbito	
1	Criação	1 Conceção	Pedaço ou código simples para testar ou observar resultados. Documentação destinada a expressar a ideia.	Estrutura de Programa de Solução de Software inexistente ou muito reduzida.	Idea, Conceção, Experimentação, Entendimento e Compreensão	
			Software escrito, ou conceito de software, sem uma estrutura necessariamente formal.			
		2 Investigação e Experimentação	Conjunto de códigos/documentos relacionados e/ou que utilizam os resultados de outros para gerar as saídas pretendidas.			
			Software escrito ou conceito de software já melhorado, mas sem necessariamente possuir uma estrutura completa			
		3 Materialização	Pequeno bloco (ou vários) de código a funcionar, mesmo com intervenção do programador, que fornece o resultado pretendido.			
			Software escrito, ou conceito de software, com alguma estrutura formal ou com descrição/comportamento de código melhorado.			
		1 Pequena Escala	Um primeiro software "apresentável" está desenvolvido, podendo ainda necessitar de mais esforços de R&D ou de mais experimentação.	Software protótipo ou solução de código incipiente.	Evolução, Melhoria, Validação e Aplicação em Análise	
		2 Expansão	Software funcional, mas ainda não completo, com apenas algumas partes ou funcionalidades finalizadas.	Software incompleto, mas parcialmente funcional.		
		3 Protótipo	Versão demonstrável. Continua a ser um software incompleto, mas totalmente funcional em alguns módulos ou funcionalidades.			
2	Evolução	1 Versão <i>beta</i>	Software quase concluído ou em processo de aperfeiçoamento/melhoria de funcionalidades.	Testado/experimentado num ambiente semelhante ao de Implementação/Lançamento no ambiente-alvo final.	Utilização em ambiente controlado ou em pequena escala	
			Pode ser utilizado para testes ou em ambiente "Family & Friends"			
		2 Pronto	Versão de software concluída e testada, com a documentação adequada ou desejada.	Pronto para produção ou para Implementação/Lançamento no ambiente-alvo final.	Totalmente implementado e pronto a usar	
3	Finalizado		Software comprovado em todos os aspetos-alvo			

4. Metodologia de validação proposta

Para transformar a proposta em contribuição científica validada, propõe-se um plano de investigação a ser realizado em três fases:

- 1) Validação por especialistas: construir painéis e *workshops* com profissionais especialistas e académicos para aferir validade de conteúdo e clareza dos critérios [Dalkey & Helmer 1963];
- 2) Estudo de caso múltiplo: aplicar a escala SML num conjunto não inferior a quatro SBAs reais, numa ou mais organizações (empresas de software, *startups* de tecnologia ou centros de investigação académica) para testar aplicabilidade, usabilidade e sensibilidade do modelo.
- 3) Avaliação quantitativa: registar medidas relevantes de aceitação, facilidade de uso, e vantagens obtidas.

5. Discussão

A análise realizada sobre a literatura observa que os modelos de maturidade mais difundidos foram concebidos para contextos predominantemente tangíveis e processuais, o que os torna insuficientes para captar a natureza intangível, iterativa e altamente variável dos SBAs. A Tabela 3 mostra uma síntese comparativa entre escalas de maturidade citadas neste trabalho:

Tabela 3. Síntese comparativa entre escalas de maturidade

Modelo	Objetivo principal	Estrutura	Limitações principais	Diferencial proposto pela SML
TRL- <i>Technology Readiness Levels</i> [ISO 2013/2024]	Avaliar a maturidade tecnológica até à sua operacionalização.	9 níveis, progressão linear.	Linearidade excessiva, ausência de métricas para software, foco em tangíveis.	A SML adapta a noção de prontidão ao software, com progressão iterativa e associada a qualidade.
MRL - <i>Manufacturing Readiness Levels</i> [DoD, 2025; Wheeler & Ulsh, 2010]	Avaliar a maturidade e prontidão de processos de manufatura.	10 níveis, alinhados com TRL.	Pouca aplicabilidade em software, foco em manufatura física.	A SML centra-se em <i>software assets</i> intangíveis, não em manufatura.
SPL-CMM- <i>Software Product Line Capability Maturity Model</i> [Çalışkanbaş, 2019]	Avaliar a capacidade organizacional para gerir variabilidade em <i>software product lines</i> .	5 níveis de maturidade organizacional.	Restrito aos SPLs, não aplicável a todos os tipos de <i>software assets</i> .	A SML é transversal a qualquer SBA, ou <i>software asset</i> , indo além de linhas de produto.
SAMM – <i>Software Assurance Maturity Model</i> [Chandra, 2009]	Avaliar práticas de segurança em software.	4 domínios, 3 níveis de maturidade por domínio.	Foco apenas em segurança, não cobre qualidade ou reuso.	A SML integra segurança como um critério, mas com visão holística na maturidade de SBAs.
SML – <i>Software-Based Assets Maturity Level</i> [Simões, 2023; Simões et al., 2025]	Avaliar a maturidade de SBAs em todo o ciclo de vida.	3 níveis, subdivididos em 8 estágios.	Ainda carece de validação empírica alargada.	Modelo dedicado a SBAs, integra maturidade, prontidão e governação.

Do ponto de vista prático, o SML procura aportar três vantagens potenciais. Primeiro, a comunicabilidade simplificada, pois a estrutura em níveis e estágios oferece uma semântica simples e partilhável por equipas técnicas e gestão, reduzindo ambiguidades típicas quando se discutem estados intermédios.

Segundo, a mensurabilidade, que, ao ancorar critérios de passagem de estágio em subcaracterísticas, reforça a objetividade da avaliação e favorece a comparabilidade entre SBAs ao longo do tempo. Terceiro, a governação e decisão, ao propor uma ligação explícita à compreensão dos *stakeholders* e às atividades do ciclo de vida.

Em termos de posicionamento, o SML não pretende substituir de forma disruptiva outras escalas consolidadas. Antes, oferece um modelo adaptado aos SBAs e complementa-as, trazendo uma preocupação com consistência e repetibilidade de avaliação.

Em síntese, a análise comparativa sugere que o SML responde às lacunas identificadas, justificando a sua avaliação empírica em contextos organizacionais distintos.

6. Conclusão

Este trabalho propõe o SML como uma escala de maturidade desenhada para SBAs. A sua principal contribuição é oferecer uma linguagem comum e critérios orientadores para avaliar a prontidão, evolução e operacionalização de SBAs, e, possivelmente, para *software assets*.

O SML procura equilibrar utilidade prática e rigor académico. A sua simplicidade facilita adoção por equipas multifuncionais e órgãos de decisão (gestão, TI e inovação) enquanto os estágios permitem rastreio fino da evolução tecnológica. As limitações incluem a validação da necessidade de calibragem por domínio e uma eventual dependência de mais indicadores qualitativos.

A revisão da literatura permitiu consolidar um enquadramento crítico sobre a evolução do conceito de *software asset* e a sua aplicação ao domínio do software, destacando a potencial relevância dos SBAs, enquanto recursos intangíveis de elevado valor estratégico.

Observou-se que, embora existam normas internacionais e modelos consolidados como a ISO 16290:2013 (rev. 2024) e outras escalas específicas, subsistem limitações estruturais na avaliação de maturidade e prontidão de *software assets*, como, nomeadamente, a ausência de taxonomias unificadas e a escassez de validação empírica em contextos industriais.

A escala SML apresenta-se, assim, como uma proposta de resposta estruturada ao universo dos SBAs e às limitações identificadas, funcionando como ponte entre a normatividade e as necessidades práticas de avaliação de maturidade no domínio do software. O seu objetivo não é substituir modelos consolidados, mas suprir uma necessidade para a medição de maturidade de SBAs e complementar estes modelos, permitindo às organizações medir a maturidade de *software assets* de forma objetiva, comunicável e empiricamente validada.

REFERÊNCIAS

- Çalışkanbaş, R. B. (2019). *SPL-CMM: Software Product Line Capability Maturity Model* [Dissertação de mestrado, Middle East Technical University]. <https://etd.lib.metu.edu.tr/upload/12623414/index.pdf>
- Chandra, P. (2009). *Software Assurance Maturity Model: A guide to building security into software development* (Version 1.0). OWASP. https://opensamm.org/downloads/SAMM-1.0-en_US.pdf
- Cortiñas, A., Krüger, J., Lamas, V., Luaces, M. R., & Pedreira, O. (2023). How to retire and replace a software product line. In P. Arcaini, M. H. ter Beek, G. Perrouin, I. Reinhartz-Berger, M. R. Luaces, C. Schwanninger, S. Ali, M. Varshosaz, A. Gargantini, S. Gnesi, M. Lochau, L. Semini, & H. Washizaki (Eds.), *SPLC '23: Proceedings of the 27th ACM International Systems and Software Product Line Conference – Volume A* (pp. 275–286). Association for Computing Machinery. <https://doi.org/10.1145/3579027.3609004>
- Dalkey, N. C., & Helmer, O. (1963). *An experimental application of the DELPHI method to the use of experts*. Management Science, 9(3), 458–467. <https://doi.org/10.1287/mnsc.9.3.458>
- DoD. Department of Defense. (2025). *Manufacturing readiness level body of knowledge: MRL deskbook*. https://www.dodmrl.com/MRL_Deskbook_2025.pdf
- Héder, M. (2017). *From NASA to EU: The evolution of the TRL scale in Public Sector Innovation*. Innovation Journal, 22(2), 1–23. https://www.researchgate.net/publication/350942366_From_NASA_to_EU_the_evolution_of_the_TRL_scale_in_Public_Sector_Innovation
- Horcas, J. M., Pinto, M., & Fuentes, L. (2023). Empirical analysis of the tool support for software product lines. *Software and Systems Modeling*, 22, 377–414. <https://doi.org/10.1007/s10270-022-01011-2>.
- IEEE. (2017/2022). ISO/IEC/IEEE 24765:2017. *Systems and software engineering—Vocabulary* (2nd ed.). <https://www.iso.org/standard/71952.html>
- ISO. (2013/2024). ISO 16290:2013-Space systems — *Definition of the Technology Readiness Levels (TRLs) and their criteria of assessment*. <https://www.iso.org/standard/56064.html>
- ISO/IEC. (2017/2024). ISO/IEC 19770-1:2017. *Information technology - IT asset management. Part1: IT asset management systems – Requirements*. <https://www.iso.org/standard/68531.html>
- ISO/IEC 38500:2024. (2024). *Information technology - Governance of IT for the organization*. <https://www.iso.org/standard/81684.html>
- ISO 55000:2024. (2024). *Asset management: Overview, principles and terminology*. <https://www.iso.org/standard/83053.html>

Karimi, J., & Zand, M. K. (1998). *Asset-based system and software system development: A frame-based approach*. *Information and Software Technology*, 40(2), 69-78. [https://doi.org/10.1016/S0950-5849\(98\)00034-2](https://doi.org/10.1016/S0950-5849(98)00034-2)

Lavin, A., Gilligan-Lee, C. M., Visnjic, A., Ganju, S., Newman, D., Ganguly, S., Zheng, S., Xing, E. P., Parr, J., & Gal, Y. (2022). *Technology readiness levels for machine learning systems*. *Nature Communications*, 13(1), 6039. <https://doi.org/10.1038/s41467-022-33128-9>

Mankins, J. C. (1995). *Technology readiness levels: A white paper*. NASA. https://www.researchgate.net/publication/247705707_Technology_Readiness_Level_-_A_White_Paper

Mankins, J. C. (2009). *Technology readiness assessments: A retrospective*. *Acta Astronautica*, 65(9–10), 1216–1223. <https://doi.org/10.1016/j.actaastro.2009.03.058>

Olechowski, A., Eppinger, S. D., & Joglekar, N. (2020). *Technology readiness levels in engineering systems*. *Systems Engineering*, 23(1), 13–26. <https://doi.org/10.1002/sys.21533>

Simões, I. F. P. (2023). *A proposed maturity assessment scale for software-based assets* [Unpublished working paper]. https://www.researchgate.net/publication/375000558_A_proposed_maturity_assessment_scale_for_Software-Based_Assets

Simões, I. F. P., Bernardo, M. do R. M., & Isaías, P. T. (2025). Software-based assets: A new perspective on the concept of software assets. *Comunicação apresentada na 25.ª Conferência da Associação Portuguesa de Sistemas de Informação (CAPSI'2025)*, São Miguel, Açores, Portugal. <https://aisel.aisnet.org/capsi2025/>

Wheeler, D., & Ulsh, M. (2010). *Manufacturing readiness assessment for fuel cell stacks and systems for the back-up power and material handling equipment emerging markets* (NREL/TP-560-45406, Relatório técnico rev.). National Renewable Energy Laboratory. <https://docs.nrel.gov/docs/fy10osti/45406.pdf>

Yfanti, S., & Sakkas, N. (2024). *Technology Readiness Levels (TRLs) in the era of co-creation*. *Applied System Innovation*, 7(2), 32. <https://doi.org/10.3390/asi7020032>

Zabardast, E., Frattini, J., Gonzalez-Huerta, J., Mendez, D., Gorschek, T., & Wnuk, K. (2022). *Assets in software engineering: What are they after all?* *Journal of Systems and Software*, 193, 111485. <https://doi.org/10.1016/j.jss.2022.111485>

Zabardast, E., Gonzalez-Huerta, J., Gorschek, T., Šmite, D., Alégroth, E., & Fagerholm, F. (2023). *A taxonomy of assets for the development of software-intensive products and services*. *Journal of Systems and Software*, 202, 111701. <https://doi.org/10.1016/j.jss.2023.111701>



Ivo Fabiano Pereira Simões, é licenciado em Informática (2004), e cursou graduação em Eng. Elétrica e de Computação (2009), além do bacharelado em Ciência da Computação. Possui pós-graduações em Análise, Projeto e Gerência de Sistemas, Ativos Intangíveis, Cálculo e Matemática Aplicada, além de outras diversas nas áreas da Engenharia. Cursou mestrado em Sistemas e Computação (2009). Foi professor no Centro Universitário UniCarioca (2013 - 2015) e na pós-graduação em Engenharia de Software com Java na Faculdade SENAC-RJ (2009 - 2010), tendo igualmente lecionado disciplinas na licenciatura do CEFET-RJ (2009 - 2010). É atualmente mestrando em Gestão pela UAb, e mestrando em Engenharia Informática e Tecnologia Web pela UAb e UTAD. É membro da IEEE *Computer Society* e faz parte do grupo docente da Pós-graduação em Inovação Digital e Business Analytics. Os seus interesses de investigação são na área dos *Software-Based Assets*, inovação, Engenharia de Software e em Vírus de Computador.



Maria do Rosário de Abreu de Matos Bernardo, é doutorada em Gestão pelo Instituto Superior de Economia e Gestão da Universidade de Lisboa (ISEG-UL). Docente da Universidade Aberta (UAb) desde 1997 nas áreas de Economia, Gestão e Sistemas de Informação. Os seus interesses de investigação incluem a tomada de decisão, *e-government*, *chatbots*, business intelligence e *smart cities*. É investigadora sénior do INESC TEC, núcleo da UAb, integrando o grupo ISAC (Information Systems and Applied Computing) do centro HumanISE. Tem integrado comissões científicas de conferências nacionais e internacionais, bem como de revistas científicas na área dos sistemas de informação.



Pedro Isaías, é professor associado com agregação na Universidade Aberta em Lisboa. Foi professor associado na *The University of New South Wales (UNSW – Sydney)* e na *The University of Queensland*, Austrália. Leciona unidades curriculares em Gestão de Sistemas de Informação. É doutor em Gestão de Informação (ramo de Sistemas de Informação e Decisão) pela Universidade Nova de Lisboa. Na Universidade Aberta é responsável por diversas unidades curriculares, tendo sido coordenador do mestrado em Gestão/MBA e coordenador do mestrado em Comércio Eletrónico e Internet durante cerca de 10 anos. Autor de diversos livros, capítulos de livros, artigos e relatórios de investigação liderou diversas conferências e workshops na área dos Sistemas de Informação. Foi também responsável pela coordenação científica e avaliador de diversos projetos de investigação financiados pela União Europeia. É editor e membro do comité editorial de diversas revistas científicas e *chair* e membro do comité de diversas conferências e workshops.

(esta página par está propositadamente em branco)