

Repositório de Dinâmicas Reutilizáveis: Promovendo o *Onboarding* Inclusivo para Engenheiros de Software com TEA

Stela Marisco Duarte
Faculdade de Computação (FACOM),
Universidade Federal de Mato Grosso
do Sul (UFMS)
Campo Grande, Brasil
stela.marisco@ufms.br

Daniel Vitoriano Santos
Faculdade de Computação (FACOM),
Universidade Federal de Mato Grosso
do Sul (UFMS)
Campo Grande, Brasil
daniel.vitoriano@ufms.br

Maria Istela Cagnin
Faculdade de Computação (FACOM),
Universidade Federal de Mato Grosso
do Sul (UFMS)
Campo Grande, Brasil
istela.machado@ufms.br

Resumo

A inclusão de engenheiros de software com Transtorno do Espectro Autista (TEA) em equipes de desenvolvimento é uma realidade crescente, porém a fase de *onboarding* permanece um problema estruturalmente não resolvido: práticas eficazes dependem de gestores específicos, não são documentadas e se perdem com a rotatividade organizacional. Para mitigar esses problemas, este artigo propõe a estrutura de um repositório, baseada na instânciação e extensão do metamodelo *Core RAS (Reusable Asset Specification)* da OMG, para catalogar e recuperar dinâmicas inclusivas de *onboarding* como ativos reutilizáveis. Para estruturar a classificação desses ativos, foi definido um modelo de classificação baseado em três eixos ortogonais (habilidades, perfil sensorial e contexto ágil) que viabiliza a busca e a recuperação dos ativos orientada por critérios combinados. Uma Prova de Conceito (PoC) com doze ativos catalogados demonstrou que um gestor pode recuperar a dinâmica que satisfaz simultaneamente critérios de habilidade, perfil sensorial e momento de *sprint*, sem depender de memória institucional. Espera-se que este trabalho, ao aplicar o padrão OMG RAS ao domínio de integração neurodivergente em Engenharia de Software e introduzir pontos de variabilidade comportamental em ativos de processos de *onboarding*, colabore para promover a inclusão de indivíduos com TEA em equipes de desenvolvimento de software.

Palavras-chave

Engenharia de Software, Transtorno do Espectro Autista (TEA), Onboarding, Ativos Reutilizáveis

1 Introdução

A inclusão de engenheiros de software com Transtorno do Espectro Autista (TEA) em equipes de desenvolvimento de software tem sido amplamente discutida na literatura [3, 16] e em iniciativas organizacionais [4, 15, 22]. Estudos apontam o alinhamento entre competências cognitivas frequentemente associadas ao TEA (como pensamento sistemático, atenção sustentada a padrões e memória de trabalho detalhista) e demandas técnicas da Engenharia de Software (particularmente atividades de depuração, revisão de código e modelagem de requisitos) [3, 16]. No entanto, esse potencial cognitivo contrasta com um cenário corporativo que ainda enfrenta dificuldades de adaptação em seus processos, especialmente em contextos ágeis, em que dinâmicas colaborativas intensivas e processos pouco estruturados podem impor barreiras relacionadas à comunicação interpessoal e à previsibilidade das atividades [1, 10].

Nesse cenário, a fase de *onboarding*, que é o período de integração de novos profissionais às equipes, emerge como uma etapa desafiadora e decisiva. A literatura de Engenharia de Software evidencia que a documentação insuficiente de artefatos técnicos e a ausência de mecanismos estruturados de transferência de conhecimento figuram entre os desafios mais recorrentes no processo de integração de desenvolvedores [12, 21]. Para profissionais com TEA, essa lacuna é ainda mais acentuada uma vez que a transição para um ambiente socialmente complexo e com processos pouco explícitos podem comprometer a retenção desses talentos e o desempenho técnico que justificou a contratação desses indivíduos [11].

No contexto ágil, um gestor, quando inicia uma *sprint*, conhece as habilidades técnicas e não técnicas (*hard e soft skills*) que serão demandadas naquele ciclo e dispõe do perfil do profissional com TEA em integração. Contudo, não conta com mecanismo formal que lhe permita selecionar, com base nesses critérios, práticas ou dinâmicas de *onboarding* que desenvolvam exatamente as competências que aquela *sprint* irá exigir. Essa ausência de estruturas padronizadas para seleção de atividades de integração em contextos ágeis é uma lacuna identificada na literatura [7, 10, 21]. A ausência de roteiros padronizados para inclusão neurodivergente, a imprevisibilidade das cerimônias ágeis e a dependência de comunicação síncrona transformam o *onboarding* em um processo reacional e improvisado, dependente da experiência individual dos gestores e não da organização.

Quando gestores experientes deixam a empresa, esse saber se perde. Esse fenômeno é denominado por Hause et al. [13] como o “problema do polvo”. Assim como o polvo, animal inteligente e adaptável, morre sem transmitir o que aprendeu, organizações perdem sistematicamente o conhecimento acumulado [13], por exemplo, sobre a inclusão neurodivergente. Estudos em projetos de software globalmente distribuídos reforçam o impacto dessa perda, demonstrando que, na ausência de estruturas organizacionais formalizadas de integração, o conhecimento das práticas adotadas e o entendimento dos sistemas existentes perdem-se com a rotatividade dos indivíduos que os detêm [5].

Diante das lacunas observadas, este trabalho propõe a seguinte questão de pesquisa: *Como favorecer o compartilhamento e a reutilização de dinâmicas adequadas ao perfil de engenheiros de software com TEA durante o processo de onboarding?*

A especificação *Reusable Asset Specification* (RAS) [17], padrão da *Object Management Group* (OMG), emerge como uma resposta arquitetural direta para promover o compartilhamento e a transferência do conhecimento acumulado sobre a integração de indivíduos com TEA em equipes de desenvolvimento de software. Originalmente

concebido para empacotamento de componentes de software, o RAS define como ativos reutilizáveis devem ser descritos por metadados estruturados e indexados de forma a viabilizar a descoberta de ativos orientada por critérios. Sua capacidade de extensão via perfil (*profile*) e pontos de variabilidade (*variability points*) já foi aplicada com sucesso em domínios como processos de software [18], metaprocessos [9] e Sistemas-de-Sistemas (SoS) [14], demonstrando que o RAS é um metamodelo de propósito geral, não restrito a artefatos de código. O esforço recente de atualização para o RAS 3.0, com foco em catálogos pesquisáveis e curadoria estruturada de ativos [13], reforça que a comunidade reconhece a necessidade de evoluir o padrão exatamente na direção de metadados mais ricos e contextualizados.

Para responder a questão de pesquisa supramencionada, este trabalho tem como principal objetivo especificar a estrutura de um repositório de dinâmicas, que podem ser utilizadas durante o *onboarding* de engenheiros de software com TEA, como ativos reutilizáveis e em conformidade com o padrão RAS. Delimita-se, neste trabalho, o público-alvo de engenheiros de software com TEA classificados nos níveis de suporte 1 e 2 [2] por manterem autonomia funcional para participar de dinâmicas colaborativas de *onboarding*, ainda que com necessidades específicas de adaptação sensorial e comunicacional [8].

Para o desenvolvimento deste trabalho, o metamodelo *Core* do RAS (Seção 2) é instanciado e estendido para catalogar e categorizar apropriadamente os ativos, considerando habilidades (técnicas e não técnicas) e perfil sensorial cobertos, momento de adotá-los na *sprint* e como utilizá-los, obtendo-se o perfil RAS-TEA. Uma Prova de Conceito (PoC) foi realizada no âmbito deste trabalho para observar a capacidade da estrutura do repositório definida quanto à catalogação e recuperação de dinâmicas para apoiar o *onboarding* inclusivo.

Diferentemente de repositórios genéricos de ativos reutilizáveis, que catalogam artefatos técnicos sem considerar dimensões de neurodiversidade, e de práticas de *coaching* ágil para profissionais com TEA, dependentes do julgamento situacional de gestores específicos [1, 7], este trabalho formaliza a descoberta orientada por critérios e a execução segura de dinâmicas de *onboarding*, eliminando a dependência de conhecimento tácito e favorecendo a gestão de conhecimento organizacional.

A escrita deste artigo está estruturada em mais seis seções. A Seção 2 apresenta o referencial teórico e trabalhos relacionados. A Seção 3 descreve o método de pesquisa. A Seção 4 apresenta o modelo definido para classificar dinâmicas reutilizáveis para subsidiar a sua catalogação, busca e recuperação. A Seção 5 descreve como o *Core* RAS foi instanciado e estendido neste trabalho para estruturar o repositório de dinâmicas reutilizáveis de *onboarding*. A Seção 6 apresenta a prova de conceito. Por fim, a Seção 7 conclui o trabalho.

2 Referencial Teórico e Trabalhos Relacionados

A literatura sobre *onboarding* de desenvolvedores de software distingue entre abordagens genéricas, focadas em ambientes, ferramentas e fluxos organizacionais [21]; e abordagens diferenciadas, que consideram necessidades individuais de integração [6]. No espectro das abordagens diferenciadas, iniciativas para profissionais com TEA têm se concentrado em adaptações pontuais de ambiente físico

(iluminação, redução de ruído), pareamento com mentores experientes e flexibilização de cerimônias ágeis [1, 7]. Embora efetivas no nível individual, essas iniciativas não são institucionalizadas, pois dependem de gestores com conhecimentos específicos e não são documentadas como padrões replicáveis que permitam a sua adaptação a novos contextos [16].

Em paralelo, a especificação RAS é um padrão da OMG e foi projetada para padronizar o empacotamento, a descrição e a classificação de ativos reutilizáveis [17]. Seu metamodelo central (*Core* RAS) organiza cada ativo em seis metaclasses principais: *Asset*, responsável pela identidade do ativo; *Classification*, voltada à descoberta e ao contexto; *Usage*, que descreve instruções de customização e execução; *Solution*, que representa os artefatos concretos do ativo; *Related Assets*, destinada ao gerenciamento de relacionamentos entre ativos reutilizáveis e *Profile*, que representa extensões semânticas específicas de domínio.

A especificação RAS foi originalmente concebida para artefatos de código, mas pode ser estendida via *Profile*, que permite adicionar semântica específica de domínio à estrutura do ativo, e por meio de *variability points*, que são pontos de adaptação obrigatórios em todo ativo ou em seu contexto de uso, viabilizando sua customização a cenários particulares sem violar a estrutura normativa do padrão [17]. A flexibilidade e a estensibilidade provida pelo RAS permite que esse padrão não seja restrito a artefatos de código, oferecendo um metamodelo de propósito geral para qualquer ativo que precise ser descoberto, configurado e reutilizado de forma sistemática. Isso foi evidenciado por meio do uso com sucesso do RAS em domínios distintos de código-fonte.

Pacini e Braga [18] propuseram uma abordagem para o reúso de elementos de processos de software baseada no RAS, aplicada no contexto de linhas de produto de software. Fernández-Ledesma [9] demonstrou sua viabilidade para metaprocessos e validou estatisticamente seu modelo de metaprocessos orientado a RAS. Hause [14] estendeu o padrão para modelagem de SoS, proporcionando o reúso de modelos arquiteturais em cenários desse tipo de sistemas complexos. No entanto, não foram encontrados trabalhos sobre dinâmicas adotadas em *onboarding* de indivíduos com TEA como ativos reutilizáveis e em conformidade com o RAS.

3 Método de Pesquisa

O método de pesquisa utilizado neste trabalho foi exploratório e descritivo, com instanciação e extensão do metamodelo *Core* do RAS v2.2 [17] para estabelecer a estrutura de um repositório de dinâmicas reutilizáveis no *onboarding* de engenheiros de software com TEA.

A extensão ocorreu em dois níveis: (i) no nível do *Profile*, em que foi definida uma extensão semântica específica do domínio TEA, codificando dimensões de neurodiversidade (sensibilidade sensorial, ansiedade social, literalidade, previsibilidade, habilidades e contexto ágil); e (ii) no nível dos *variability points*, em que foram especificados pontos de adaptação obrigatórios em cada ativo e no seu uso, permitindo a customização do ativo a perfis neurocognitivos particulares sem violar a estrutura normativa do padrão. A extensão realizada é metodologicamente coerente com o histórico de extensões do RAS em outros domínios [9, 14, 18], mas inédita no

contexto de dinâmicas para *onboarding* inclusivo de profissionais neurodivergentes.

Para subsidiar a classificação do ativo reutilizável no contexto deste trabalho, com o intuito de favorecer a catalogação, busca e a recuperação de dinâmicas adequadas para determinado contexto do processo de *onboarding*, foi definido um modelo de classificação de dinâmicas reutilizáveis. A concepção desse modelo foi baseada nos resultados de um mapeamento sistemático multivocal da literatura [8], conforme apresentado na próxima seção.

4 Modelo de Classificação de Dinâmicas Reutilizáveis

Para que os ativos do repositório proposto sejam descobertos, selecionados e recuperados de forma adequada, foi definida uma classificação baseada em três eixos ortogonais: habilidades, perfil sensorial e de neurodiversidade, e contexto ágil. Esses eixos emergiram diretamente a partir de dimensões recorrentes e relevantes da literatura sobre *onboarding* de profissionais com TEA em contextos ágeis, resultantes de um mapeamento sistemático multivocal da literatura [8]. Cada eixo é apresentado a seguir.

(1) Eixo de Habilidades: mapeia cada dinâmica às habilidades técnicas (e.g., metodologias ágeis, controle de versão, revisão de código) e habilidades não técnicas (e.g., comunicação assertiva, gestão de ansiedade, colaboração em pares) que ela exercita ou avalia.

(2) Eixo de Perfil Sensorial e de Neurodiversidade: classifica cada dinâmica segundo as dimensões de neurodiversidade que ela mobiliza (como sensibilidade auditiva e tátil, ansiedade social, literalidade e previsibilidade) e as adaptações necessárias na execução da dinâmica para cada dimensão coberta. Esse eixo garante que a seleção de uma dinâmica leve em conta o perfil do profissional, e que as adaptações necessárias estejam descritas no próprio ativo, não delegadas ao julgamento situacional do gestor.

(3) Eixo de Contexto Ágil: posiciona cada dinâmica em um dos três momentos de uma *sprint* do *onboarding*. No *início* da *sprint*, dinâmicas podem ser selecionadas em função do que será trabalhado naquele ciclo — introduzindo o profissional ao contexto técnico e social que ele encontrará durante a *sprint*. No *meio* da *sprint*, dinâmicas de reforço podem ser selecionadas para serem aplicadas a partir de dificuldades observadas pelo gestor, endereçando de forma estruturada os pontos onde a integração apresentou fricção. Ao *final* da *sprint*, dinâmicas de consolidação podem ser selecionadas para promover a fixação das competências desenvolvidas, funcionando analogamente como uma retrospectiva ágil. Este eixo contribui incrementalmente para a integração do profissional com TEA.

A combinação dos três eixos permite que gestores filtrem dinâmicas no repositório não apenas por habilidade-alvo, mas pelo perfil sensorial do profissional e pelo momento específico da integração em que se encontram, tornando a seleção do ativo tão estruturada quanto a sua execução.

5 Instanciação e Extensão do RAS

O RAS possui três propriedades que o tornam adequado ao problema abordado neste trabalho. Primeiro, a metaclass *Classification* resolve diretamente a principal barreira ao reúso identificada por

Park et al. [19], que se refere à dificuldade de descobrir ativos existentes e saber como utilizá-los. Segundo, a metaclass *Usage* obriga a especificação explícita de roteiros de execução (*scaffolding*) e pontos de variabilidade (*variability points*) dentro da própria estrutura do ativo para torná-lo adequado a cada uso. Terceiro, o padrão é compatível com repositórios baseados em Git, viabilizando versionamento distribuído, colaboração e rastreabilidade da evolução dos ativos [20]. Em conjunto, essas três propriedades (ou seja, descoberta estruturada, execução prescrita e versionamento) fazem do RAS a base para estruturar dinâmicas de *onboarding* como ativos reutilizáveis para que possam ser replicáveis nas organizações.

Portanto, a inovação central deste trabalho consiste em adotar o padrão RAS para estabelecer a estrutura de um repositório de dinâmicas para apoiar o *onboarding* inclusivo de profissionais com TEA. Para isso, adotou-se como premissa que ao iniciar uma *sprint* em um projeto de *onboarding*, o gestor conhece três variáveis: (i) as habilidades técnicas e não técnicas que serão demandadas naquele ciclo; (ii) o momento de utilizar dinâmicas em uma *sprint* (no início, no meio ou no final); e (iii) o perfil sensorial e cognitivo dos profissionais com TEA participantes da integração.

As metaclasses do metamodelo *Core* do RAS foram instanciadas como classes e os respectivos metadados foram definidos como atributos da estrutura do repositório de dinâmicas proposto para *onboarding* inclusivo. Sob a perspectiva de extensão, quando necessário, atributos específicos do domínio foram adicionados às classes resultantes da instanciação.

A Figura 1 apresenta o diagrama de classes da estrutura do repositório proposto, contendo as classes resultantes da instanciação do *Core* RAS e os atributos específicos do domínio resultantes da extensão realizada neste trabalho. Para fins de distinção, os atributos incluídos estão no idioma português, enquanto que as classes e os atributos correspondentes, respectivamente, às metaclasses e aos metadados do *Core* RAS, estão no idioma inglês.

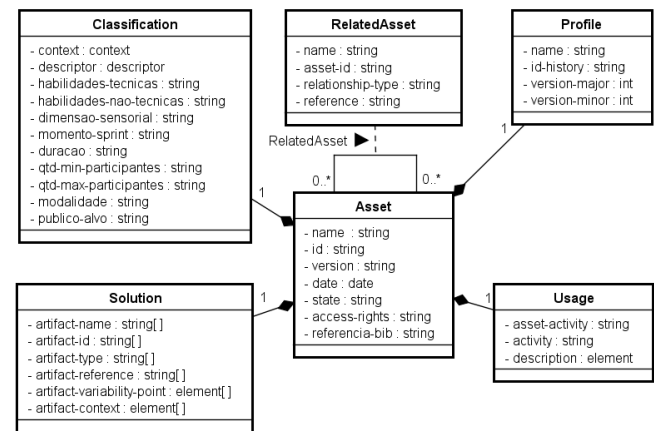


Figura 1: Diagrama de classes da estrutura do repositório de dinâmicas reutilizáveis.

A classe *Asset* representa cada dinâmica de *onboarding* como um ativo reutilizável. Essa classe contém os atributos *name*, *id*, *version*, *date*, *state* e *access-rights* derivados da metaclass *Asset* do *Core* RAS. O atributo *referencia-bib* foi acrescentado e se refere a

referência bibliográfica da dinâmica. A classe *Classification* categoriza a dinâmica reutilizável com base no modelo de classificação definido na Seção 4. Essa classe contém os atributos *context* e *descriptor* resultantes da instanciação da metaclassa *Classification* do Core RAS e novos atributos específicos do domínio para facilitar a busca e a recuperação das dinâmicas para o *onboarding*. Os atributos adicionados habilidades-tecnicas, habilidades-nao-tecnicas, dimensao-sensorial e momento-sprint codificam, respectivamente, as habilidades técnicas e não técnicas que a dinâmica desenvolve, as dimensões de neurodiversidade que endereça e o contexto ágil em que se aplica, com base no modelo de classificação de dinâmicas reutilizáveis apresentado na seção anterior. Outros atributos incluídos foram: duração (tempo previsto de execução da dinâmica), *qtd_min_participantes* e *qtd_max_participantes* (número mínimo e máximo de participantes recomendados para que a dinâmica produza o resultado esperado), modalidade (forma de execução da dinâmica, determinando os recursos físicos e digitais necessários) e publico-alvo (quem pode e está apto a conduzir a dinâmica – por exemplo, gestor, líder técnico, coordenador de equipe, etc.). Os atributos adicionados capturam as variáveis operacionais necessárias para que o gestor avalie a adequação da dinâmica ao contexto específico de *onboarding* antes de selecioná-la. Assim, o gestor não escolhe uma dinâmica qualquer, pois pode selecionar aquela cujas habilidades-alvo são precisamente as que são necessárias para o profissional com TEA em determinado momento do processo de *onboarding*.

A classe *Usage* representa o roteiro de execução completo da dinâmica. Os pontos de variabilidade (*variability points*) de interesse neste domínio estão no contexto de uso do ativo, ou seja, nas atividades que o gestor executa, conforme previsto no metamodelo Core do RAS. Essa classe contém os atributos *asset-activity*, *activity* e *description*, resultantes da instanciação da metaclassa *Usage* do Core RAS. O conteúdo do atributo *asset-activity* representa a adaptação do uso da dinâmica a cada contexto e contém: *scaffolding* (roteiro numerado com passos atômicos), *atencao-sensorial* (mapeamento de estímulos por canal sensorial e respectivas mitigações) e *condicoes-interruptao* (critérios objetivos de pausa e encerramento antecipado da dinâmica). Salienta-se que nas aplicações tradicionais do RAS, a classe resultante da instanciação da metaclassa *Usage* comumente descreve como instalar ou configurar um componente de software. Neste trabalho, ela torna-se o instrumento de eliminação de ambiguidade na execução da dinâmica: cada regra de interação, cada ponto de atenção sensorial e cada restrição vocabular são formalizados como conteúdo do atributo *asset-activity*, garantindo que a dinâmica possa ser executada por qualquer equipe (com ou sem experiência prévia em neurodiversidade) com o mesmo nível de previsibilidade e segurança para o profissional com TEA.

A classe *Solution* reúne os artefatos que compõem a dinâmica como ativo entregável. O elemento *artifact* do Core RAS foi instanciado como os seguintes atributos dessa classe: *artifact-name*, *artifact-id*, *artifact-type*, *artifact-reference*, *artifact-variability-point* e *artifact-context*. A variabilidade de cada artefato é declarada por meio do atributo *artifact-variability-point*, especificando o identificador do ponto de variação, o contexto sensorial ao qual se aplica e a descrição da adaptação esperada para cada perfil de neurodiversidade. Os seguintes artefatos obrigatórios são definidos para

eliminar a ambiguidade na condução de cada dinâmica: **cartao-facilitador** (orienta passo a passo a condução da dinâmica pelo gestor); **guia-de-personalizacao** (documenta como adaptar a execução da dinâmica a diferentes perfis sensoriais e de neurodiversidade); **checklist** (verifica pré-requisitos, materiais e condições ambientais necessárias antes da aplicação); e **modelo-debriefing** (estrutura o roteiro da conversa de fechamento após a dinâmica, consolidando o aprendizado obtido).

A classe *RelatedAsset* representa os relacionamentos entre as dinâmicas de *onboarding* modeladas como ativos RAS com os seguintes atributos do Core RAS: *name*, *asset-id*, *relationship-type* e *reference*. Três tipos de relacionamento do vocabulário reservado do Core RAS são reutilizados: *dependency*, para dinâmicas que são pré-requisitos e que devem anteceder o ativo atual; *similar*, para dinâmicas que abordam o mesmo objetivo de aprendizagem sob diferentes perfis sensoriais; e *aggregation*, para sequências de dinâmicas que compõem um fluxo completo de *onboarding* em uma determinada *sprint*. Essa classe suporta diretamente o conceito de progressão de *onboarding*. Por exemplo, um gestor que aplica, no início da *sprint*, uma dinâmica de introdução ao fluxo ágil pode navegar até uma dinâmica de segurança psicológica indicada como pré-requisito, ou, por meio do relacionamento *similar*, até uma dinâmica alternativa de comunicação precisa que atenda o mesmo objetivo sob um perfil sensorial distinto, para atividades de reforço durante o ciclo.

A classe *Profile* define a identidade e a extensão do RAS realizada para estruturar o repositório de dinâmicas reutilizáveis, denominada RAS-TEA neste trabalho. O perfil RAS-TEA deriva do *Default Profile* 2.2 que, por sua vez, deriva do *Core Profile* do RAS. Assim, o atributo *id-history* consiste na concatenação de três identificadores únicos separados por “::”, codificando explicitamente a cadeia completa de derivação do perfil.

O perfil RAS-TEA estende o *Default Profile* 2.2 de forma estritamente aditiva, sem relaxar restrições do perfil pai, em conformidade com as Restrições 14 (um perfil derivado não pode relaxar nenhuma restrição do perfil pai; apenas pode tornar restrições opcionais em obrigatórias) e 15 (um perfil derivado só pode adicionar novos atributos e restrições àqueles herdados, nunca suprimi-los) da especificação RAS [17]. A extensão realizada neste trabalho introduz três modificações principais: (i) torna obrigatória a classe *Classification*, originalmente opcional no perfil base; (ii) define nove atributos obrigatórios de contexto — habilidades-tecnicas, habilidades-nao-tecnicas, dimensao-sensorial, momento-sprint, duracao, qtd-min-participantes, qtd-max-participantes, modalidade e publico-alvo; e (iii) estabelece os artefatos obrigatórios cartao-facilitador, guia-de-personalizacao, checklist e modelo-debriefing que devem compor a solução de todo ativo do perfil RAS-TEA. Salienta-se que a instanciação e extensão do Core RAS foi realizada neste trabalho sob a perspectiva sociotécnica, com o cuidado de não violar a especificação original.

6 Prova de Conceito

Esta seção descreve a Prova de Conceito (PoC) para observar se a estrutura estabelecida para o repositório propicia, na prática, a catalogação, busca e recuperação de dinâmicas reutilizáveis. Para isso, um protótipo funcional do repositório foi construído utilizando

as seguintes tecnologias: (i) **GitHub**, pelo suporte nativo a versionamento, rastreabilidade de mudanças e controle de acesso por *branches*; e (ii) **GitBook**, pela capacidade de renderizar documentação do repositório em um portal web navegável, permitindo que o gestor consuma o conteúdo sem exigir conhecimento técnico sobre Git ou XML.

Para materializar a estrutura do repositório de dinâmicas reutilizáveis definida na Seção 5, a construção do repositório adotou uma hierarquia de diretórios que separa a gestão do perfil RAS-TEA dos ativos criados. A Figura 2 ilustra essa organização estrutural no GitHub.

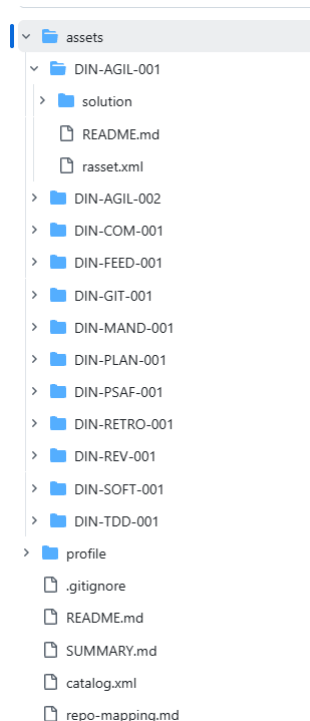


Figura 2: Estrutura de diretórios do protótipo do repositório no GitHub.

A arquitetura do repositório foi construída com base em três componentes físicos principais (o diretório de cada ativo¹, o diretório de perfil *profile* RAS-TEA e o arquivo de catálogo *catalog.xml*), que mapeiam as classes do diagrama de classes (Figura 1) para diretórios e arquivos.

Na raiz do repositório, o arquivo *catalog.xml* atua como o inventário de todas as dinâmicas. Ele sumariza o contexto das dinâmicas catalogadas no repositório, relacionado a habilidades técnicas e não técnicas, perfil sensorial e momento da *sprint* (início, meio ou fim), refletindo os eixos ortogonais definidos na Seção 4. O arquivo *SUMMARY.md* é utilizado pelo GitBook para gerar o menu lateral de navegação do portal, estruturando a descoberta visual dos ativos. O arquivo *xml profile* RAS-TEA, no diretório de mesmo nome, centraliza as regras do domínio. Ele contém o arquivo *ras-tea-profile.xsd*,

que é o esquema de validação XML responsável por garantir que todos os ativos respeitem as obrigatoriedades do perfil RAS-TEA, e o arquivo *descricao-perfil.md*, que documenta essas regras.

No diretório de ativos (*assets/*), cada dinâmica reutilizável corresponde a um subdiretório isolado, nomeado sob a convenção *DIN-<EIXO>-<SEQ>* (por exemplo, *assets/DIN-AGIL-001/*). *<EIXO>* se refere ao eixo de origem primário da dinâmica, ou seja, sob qual contexto ela foi concebida. Essa denominação foi estabelecida principalmente para fins de versionamento e rastreabilidade. Como definido no modelo de classificação (Seção 4), uma dinâmica pode pertencer a vários eixos e isso está mapeado adequadamente na classe *Classification* da estrutura estabelecida para o repositório de dinâmicas reutilizáveis (Seção 5).

Para garantir a aderência ao padrão RAS, a construção de cada ativo exige a presença de três elementos internos:

- **Manifesto *rasset.xml*:** é o ponto central de definição do ativo e concentra, em formato XML legível por máquina, os atributos das classes da estrutura do repositório de dinâmicas reutilizáveis. A classe *Asset* é mapeada para a tag raiz *<rasset>*. A classe *Classification* e seus atributos estão mapeados nas tags *<classification>* e *<context>*. A classe *Usage* é codificada nas tags *<asset-activity>* dentro da tag *<usage>*. Por fim, este arquivo possui uma tag *<profile>*, que não é uma cópia do perfil central RAS-TEA, mas sim uma declaração formal de vínculo, atestando que aquele ativo específico foi construído e deve ser validado com base no esquema presente na pasta *profile* RAS-TEA.
- **Subdiretório *solution*:** mapeia diretamente a classe *Solution*. É o repositório físico dos artefatos concretos da dinâmica, armazenando arquivos Markdown como *cartao-facilitador.md*, *guia-de-personalizacao.md*, *checklist.md* e *modelo-debriefing.md*. Esses arquivos são referenciados dentro do manifesto *rasset.xml* por meio de tags *<artifact>*.
- **Arquivo *README.md*:** atua como a camada de apresentação do ativo. Enquanto o manifesto *rasset.xml* garante a interoperabilidade estrutural com ferramentas baseadas em RAS, o *README.md* traduz essas informações para o GitBook, permitindo que o gestor leia as instruções da dinâmica de forma amigável no portal web.

Para a integridade e aderência das dinâmicas reutilizáveis ao padrão RAS, necessárias ao longo do tempo, foi operacionalizado o controle de qualidade via Git. A inclusão de uma nova dinâmica no repositório segue o fluxo *feat/<nome-do-ativo> → develop → main*. Esse processo exige uma revisão obrigatória por pares (*pull request*), momento em que se valida se o subdiretório do novo ativo contém todos os artefatos obrigatórios na pasta *solution/* e se o seu *rasset.xml* está perfeitamente em conformidade com o *ras-tea-profile.xsd*. Cada ativo segue integralmente a estrutura estabelecida para o repositório proposto, com aproximadamente 34 atributos distribuídos nas seis classes definidas, em conformidade com o *Core RAS*.

Para observar a capacidade da estrutura definida para o repositório proposto quanto à catalogação, busca e recuperação de dinâmicas reutilizáveis, foram especificados doze ativos e armazenados no protótipo construído². Para fins de exemplificação, quatro deles

¹Cada ativo corresponde a um diretório versionado no repositório.

²https://github.com/Smarisco/dinamicas_onboarding_inclusivo.git

são mencionados a seguir: (1) **DIN-AGIL-001** (*Jogo das Bolinhas — fluxo vs. ocupação em times ágeis*), (2) **DIN-AGIL-002** (*Construção com Blocos — integração modular e comunicação entre equipes*), (3) **DIN-COM-001** (*O Campo Minado — comunicação precisa e confiança em pares Sênior/Júnior*) e (4) **DIN-SOFT-001** (*A Moeda de Duas Faces — empatia técnica entre Dev e QA*).

Cada um dos doze ativos catalogados no repositório foi especificado pelos autores com base em dinâmica proposta na literatura, identificada em um mapeamento sistemático multivocal [8], e estruturada segundo as classes definidas na Seção 5. A referência ao trabalho de origem de cada dinâmica é registrada individualmente no atributo referencia-bib da classe *Asset*, garantindo a rastreabilidade entre o ativo catalogado e a literatura que o fundamenta.

O protótipo do repositório foi utilizado para buscar e recuperar dinâmicas com base no seguinte cenário real, descrito no formato de história de usuário: **Como** gestor responsável pelo *onboarding* de um engenheiro de software júnior recém-contratado com TEA, que apresenta preferência por comunicação escrita em detrimento de fala espontânea e alta necessidade de previsibilidade nas interações sociais e que iniciará sua primeira *sprint* na equipe de desenvolvimento, **quero** recuperar uma dinâmica adequada para introduzi-lo ao conceito de fluxo de trabalho ágil, **a fim de que** ele possa participar ativamente sem sobrecarga sensorial ou situações de comunicação ambígua.

A busca foi conduzida combinando os três eixos ortogonais definidos na Seção 4 como critérios de filtragem no portal GitBook do protótipo: (i) habilidade correspondente à gestão ágil (*gestão ágil*); (ii) perfil sensorial caracterizado por preferência pela comunicação escrita e elevada necessidade de previsibilidade (*comunicação*); e (iii) momento da *sprint* (*início*), coerente com a chegada do profissional à equipe. No portal web do protótipo do repositório, esses três filtros podem ser aplicados em conjunto ou separadamente sobre o menu de navegação: o gestor pode restringir primeiro pela habilidade-alvo, em seguida pelo perfil sensorial e, por fim, pelo momento da *sprint*, sem precisar consultar diretamente o manifesto *rasset.xml*. A combinação simultânea dos três critérios sobre os doze ativos catalogados resultou na recuperação do ativo **DIN-AGIL-001** (*Jogo das Bolinhas*) como o único capaz de satisfazê-los ao mesmo tempo. Seus pontos de variabilidade VP Previsibilidade e VP Comunicação, declarados no atributo *artifact-variability-point* do artefato cartão-facilitador, atendem diretamente as necessidades descritas no cenário.

A prova de conceito comprovou que é possível traduzir a arquitetura conceitual do metamodelo *Core RAS* para uma estrutura de repositório de dinâmicas de *onboarding* reutilizáveis e com facilidade de acesso. Na prática, a união do rigor do XML com a renderização visual do GitBook viabiliza que um gestor busque e recupere a dinâmica adequada que satisfaça os critérios de habilidade, perfil sensorial e momento da *sprint*, dispensando a dependência da memória institucional.

7 Conclusão e Trabalhos Futuros

Este trabalho apresentou a definição e a estrutura de um repositório de dinâmicas reutilizáveis para apoiar o *onboarding* inclusivo de engenheiros de software com TEA. A contribuição central reside na inovação conceitual de transpor o RAS da OMG [17], que é

um padrão concebido originalmente ao reúso de componentes de software, para um domínio sociotécnico. Ao estender o metamodelo *Core* do RAS v2.2, o trabalho demonstrou que as estruturas de restrições e dependências, originalmente projetadas para código e artefatos técnicos, podem ser ressignificadas para modelar dimensões de neurodiversidade, como perfis sensoriais e adaptações comportamentais. Essa formalização propõe uma estrutura inicial para mitigar o problema estrutural da perda de conhecimento (o “problema do polvo”), contribuindo para transformar o *onboarding* inclusivo, hoje refém da experiência isolada e da memória institucional de gestores específicos, em um processo mais sistemático e replicável entre equipes e organizações. A prova de conceito demonstrou a viabilidade do repositório proposto, evidenciando que ele é capaz de catalogar e recuperar dinâmicas que satisfaçam simultaneamente critérios técnicos e não técnicos (habilidades), ágeis (momento da *sprint*) e cognitivos (perfil do profissional com TEA).

Como limitações, reconhece-se que a prova de conceito foi conduzida em cenário controlado, sem avaliação em contextos reais de *onboarding* com profissionais com TEA. Adicionalmente, tanto a catalogação manual dos doze ativos quanto a definição do modelo de classificação em três eixos estão sujeitas ao viés dos pesquisadores. Esse risco de viés foi mitigado pelo fato de o modelo ter sido derivado dos resultados de um mapeamento sistemático multivocal da literatura [8] e cada um dos doze ativos ter sido especificado com base na literatura identificada nesse mapeamento. Como direcionamentos futuros, pretende-se: (i) avaliar a aceitação do repositório por gestores de *onboarding* por meio de um estudo subsidiado pelo Modelo de Aceitação de Tecnologia (TAM), mensurando utilidade percebida, facilidade de uso e intenção de adoção em contextos reais; (ii) conduzir um estudo de caso em um contexto real de *onboarding* ágil de uma organização para verificar se as dinâmicas recuperadas via repositório produzem indicadores mensuráveis de melhoria na integração e no bem-estar de profissionais com TEA; (iii) conduzir um estudo longitudinal com engenheiros de software com TEA que participaram de *onboarding* utilizando as dinâmicas reutilizáveis com o intuito de observar a permanência desses profissionais em equipes ágeis ao longo do tempo; e (iv) investigar a integração do repositório com mecanismos de recomendação semântica de ativos [19], ampliando a descoberta além da busca por palavras-chave.

Disponibilidade de Artefatos

O repositório contendo os doze ativos catalogados (DIN-AGIL-001, DIN-AGIL-002, DIN-GIT-001, DIN-PLAN-001, DIN-TDD-001, DIN-REV-001, DIN-MAND-001, DIN-RETRO-001, DIN-SOFT-001, DIN-COM-001, DIN-FEED-001, DIN-PSAF-001), incluindo os manifestos *rasset.xml*, os artefatos da classe *Solution* (guias, modelos e checklists) e o perfil RAS-TEA, está disponível em: https://github.com/Smarisco/dinamicas_onboarding_inclusivo.git. Tipo de licença: *Creative Commons Attribution 4.0 International*.

Agradecimentos

Este trabalho foi realizado com o apoio da UFMS e da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) – (Código de Financiamento 001).

Referências

- [1] Olga Abramova, Jan Recker, Ursula Schemm, and Lukas-Daniel Barwitzki. 2025. Inclusion of autistic IT workforce in action: An Auticon approach. *Information Systems Journal* 35, 5 (2025), 1439–1459. doi:10.1111/isj.12588
- [2] APA. 2014. *Manual Diagnóstico e Estatístico de Transtornos Mentais (DSM-5)* (5 ed.). Artmed, Porto Alegre.
- [3] Robert D. Austin and Gary P. Pisano. 2017. Neurodiversity as a competitive advantage. *Harvard Business Review* 95, 3 (2017), 96–103.
- [4] Autism Employer Roundtable. 2021. *Autism @ Work Playbook: Finding talent and creating meaningful employment opportunities*. Disability:IN. <https://disabilityin.org/resource/autism-work-playbook/>.
- [5] Ricardo Britto, Daniela S. Cruzes, Darja Smite, and Aivars Sablis. 2018. Onboarding software developers and teams in three globally distributed legacy projects: A multi-case study. *Journal of Software: Evolution and Process* 30, 4 (2018), e1921. doi:10.1002/smr.1921 Cited by: 41.
- [6] Isadora Cardoso-Pereira, Geraldo Gomes, Danilo Monteiro Ribeiro, Alberto de Souza, Danilo Lucena, and Gustavo Pinto. 2023. Supporting the careers of developers with disabilities: Lessons from Zup Innovation. *IEEE Software* 40, 5 (2023), 58–65.
- [7] Nicolay da Silva Menezes, Thayssa Águila da Rocha, Lucas Samuel Santiago Camelo, and Marcelle Pereira Mota. 2025. “I Felt Pressured to Give 100% All the Time”: How are neurodivergent professionals being included in software development teams?. In *XXI Simpósio Brasileiro de Sistemas de Informação (SBSI)*. SBC, Recife-PE, 525–534.
- [8] Stela Marisco Duarte, Daniel Santos Vitoriano, and Maria Istela Cagnin. 2026. Ágil para Quem? Práticas, Desafios e Modelos de Onboarding para Engenheiros de Software com Autismo. In *XI Workshop sobre Aspectos Sociais, Humanos e Econômicos de Software* (Gramado/RS). SBC, Porto Alegre, RS, Brasil, 188–199. doi:10.5753/washes.2026.21856
- [9] Javier Darío Fernández-Ledesma. 2016. Metaprocesses as Software Assets Reuse. In *6th International Symposium on Business Modeling and Software Design (BMSD)*. SciTePress, Rhodes, Greece, 221–224.
- [10] Kiev Gama and Aline Lacerda. 2023. Understanding and supporting neurodiverse software developers in agile teams. In *XXXVII Brazilian Symposium on Software Engineering (SBES)*. SBC, Campo Grande-MS, 1–6. doi:10.1145/3613372.3613384
- [11] Kiev Gama, Grisha Liebel, Miguel Goulão, Aline Lacerda, and Cristiana Lacerda. 2025. A socio-technical grounded theory on the effect of cognitive dysfunctions in the performance of software developers with ADHD and autism. In *47th IEEE/ACM International Conference on Software Engineering: Software Engineering in Society (ICSE-SEIS)*. IEEE, Ottawa, Canada, 1–12.
- [12] Collin Green, Ciera Jaspan, Maggie Hodges, Lanting He, Demei Shen, and Nan Zhang. 2023. Developer Productivity for Humans, Part 5: Onboarding and Ramp-Up. *IEEE Software* 40, 5 (2023), 13–19. doi:10.1109/MS.2023.3291158
- [13] Matthew Hause, Sriram Krishnan, Mark Petrotta, Michael Shearin, and Tomas Vileiniskis. 2025. Solving the Octopus Problem in Digital Engineering – Towards Reusable Asset Specification 3.0. *INSIGHT* 28, 5 (2025), 40–43. doi:10.1002/inst.70013
- [14] Matthew C Hause. 2014. SOS for SoS: A new paradigm for system of systems modeling. In *2014 IEEE Aerospace Conference*. IEEE, Montana, USA, 1–12. doi:10.1109/AERO.2014.6836335
- [15] Microsoft. 2025. Microsoft Neurodiversity Hiring Program. Website Institucional. <https://www.microsoft.com/en-us/diversity/inside-microsoft/cross-disability/neurodiversityhiring>.
- [16] Gastón Márquez, Michelle Pacheco, Hernán Astudillo, Carla Taramasco, and Esteban Calvo. 2024. Inclusion of individuals with autism spectrum disorder in Software Engineering. *Information and Software Technology* 170 (2024), 107434. doi:10.1016/j.infsof.2024.107434
- [17] OMG. 2005. *Reusable Asset Specification, Version 2.2*. Specification formal/2005-11-02. Object Management Group. <https://www.omg.org/spec/RAS/2.2/>
- [18] Karen D. R. Pacini and Rosana T. V. Braga. 2015. An approach for reusing software process elements based on reusable asset specification: a software product line case study. In *10th International Conference on Software Engineering Advances*. IARIA XPS Press, Barcelona, Spain, 200–206.
- [19] Soojin Park, Sooyong Park, and Vijayan Sugumaran. 2007. Extending reusable asset specification to improve software reuse. In *22nd Annual ACM Symposium on Applied Computing (SAC)*. Association for Computing Machinery, Seoul, Korea, 1473–1478. doi:10.1145/1244002.1244317
- [20] Hong-Min Ren, Zhi-Ying Yan, and Jing-Zhou Zhang. 2009. Design and implementation of RAS-based open source software repository. In *6th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*. IEEE, Tianjin, China, 219 – 223.
- [21] Ana Beatriz Cavalcanti Ribeiro and Carina Alves. 2026. Onboarding in software engineering: A multivocal literature review. *Information and Software Technology* 196 (2026), 108120. doi:10.1016/j.infsof.2026.108120
- [22] Ankit Sahu. 2022. Hiring autistic workers: 4 companies embracing neurodiversity in tech. Turing Blog. <https://www.turing.com/blog/hiring-autistic-workers-neurodiversity-in-tech/>.