

Estratégia para nortear o processo de engenharia de requisitos aplicada à metodologia ágil Scrum

Matheus Nascimento Sales. *Graduado em Ciências da Computação pela UESB. Membro Projeto de Pesquisa Adoção da ÁGIL-PERSCRUM para direcionar o processo de engenharia de requisitos aplicada à metodologia ágil Scrum no âmbito de sistemas de informação da Universidade Estadual do Sudoeste da Bahia.*

Vitória da Conquista, Bahia, Brasil
fneto@uesb.edu.br
<https://orcid.org/0000-0002-0445-286X>

José Carlos Martins Oliveira. *Professor Adjunto, Departamento de Ciências Exatas e Tecnológicas (DCET). Universidade Estadual do Sudoeste da Bahia (UESB). Membro do Projeto de Pesquisa Adoção da ÁGIL-PERSCRUM para direcionar o processo de engenharia de requisitos aplicada à metodologia ágil Scrum no âmbito de sistemas de informação da Universidade Estadual do Sudoeste da Bahia.*

Vitória da Conquista, Bahia, Brasil
jcarlos@uesb.edu.br
<https://orcid.org/0000-0003-2148-4502>

Francisco dos Santos Carvalho. *Professor Titular, Doutor, Departamento de Ciências Exatas e Tecnológicas (DCET). Universidade Estadual do Sudoeste da Bahia (UESB) Membro do Projeto de Pesquisa Adoção da ÁGIL-PERSCRUM para direcionar o processo de engenharia de requisitos aplicada à metodologia ágil Scrum no âmbito de sistemas de informação da Universidade Estadual do Sudoeste da Bahia.*

Vitória da Conquista, Bahia, Brasil
franciscoscarvalho@uesb.edu.br
<https://orcid.org/0000-0003-3598-329X>

Francisco Ângelo de Almeida Neto. *Graduado em Ciências da Computação. Unidade de Informática (UIFOR). Universidade Estadual do Sudoeste da Bahia (UESB). Pós-graduação em Gestão Universitária. Membro do Projeto de Pesquisa Adoção da ÁGIL-PERSCRUM para direcionar o processo de engenharia de requisitos aplicada à metodologia ágil Scrum no âmbito de sistemas de informação da Universidade Estadual do Sudoeste da Bahia.*

Vitória da Conquista, Bahia, Brasil
fneto@uesb.edu.br
<https://orcid.org/0000-0002-4134-1245>

Cátia Mesquita Brasi Khouri. *Professora Adjunta, Departamento de Ciências Exatas e Tecnológicas (DCET). Universidade Estadual do Sudoeste da Bahia (UESB). Membro do Projeto de Pesquisa Adoção da ÁGIL-PERSCRUM para direcionar o processo de engenharia de requisitos aplicada à metodologia ágil Scrum no âmbito de sistemas de informação da Universidade Estadual do Sudoeste da Bahia.*

Vitória da Conquista, Bahia, Brasil
khouri@uesb.edu.br
<http://orcid.org/0000-0002-5543-7299>

Mauro dos Santos Carvalho. *Mestre em Educação. Técnico da Universidade Estadual do Sudoeste da Bahia (UESB). Membro do Projeto de Pesquisa Adoção da ÁGIL-PERSCRUM para direcionar o processo de engenharia de requisitos aplicada à metodologia ágil Scrum no âmbito de sistemas de informação da Universidade Estadual do Sudoeste da Bahia.*

Vitória da Conquista, Bahia, Brasil
mauro@uesb.edu.br
<https://orcid.org/0000-0001-5130-4168>

Abstract - Agile methodologies have gained space in public and private organizations. It is a way of development that values a wide set of principles and practices that aim, for example, to generate speed, integration between team members, involvement of customers and other stakeholders, seeking to meet a productive context characterized by changes, contained in the product requirements. This research work aimed to present the AGIL-PERSCRUM strategy to guide the requirements engineering process applied to the agile Scrum methodology. The following specific objectives were established: a) Describe the requirements engineering process used in traditional software development models; b) Describe the agile Scrum methodology, with emphasis on its principles and practices, the structuring of Scrum team members, existing events and generated artifacts; c) Create models in Draw.io to describe the proposed activity flows for the Requirements Engineering process applied to Scrum; d) Propose criteria for prioritizing requirements; e) Present a scoring example of the activities and/or artifacts contained in the proposed flows. As for the methodological procedures, basic, exploratory and descriptive research was used, with a qualitative approach. At the end of the work, it was possible to conclude that the general and specific objectives were achieved. A proposal was built that is already being researched in the project Adoption of ÁGIL-PERSCRUM to direct the requirements engineering process applied to the

agile Scrum methodology in the scope of information systems at the Universidade Estadual do Sudoeste da Bahia.

Keywords - Agile development, ÁGIL-PERSCRUM, Requirements Engineering, Scrum, System GED-UESB.

I. INTRODUÇÃO

Atualmente, as empresas e organizações de desenvolvimento de software estão cada vez mais interessadas em acelerar a produção dos projetos sem perder de vista a qualidade dos mesmos. Muitos dos projetos são prejudicados quando a engenharia de requisitos não é cumprida da forma eficiente [1]. A engenharia de requisitos tem forte impacto sobre a qualidade final da construção de softwares. Dessa forma, é importante que os engenheiros de softwares e outros profissionais envolvidos direta e indiretamente no processo de desenvolvimento de software conheçam a real necessidade do cliente e dos usuários diretos e indiretos para que a proposição e desenvolvimento de produtos sejam realizados com metodologias eficientes e eficazes.

Pesquisas recentes indicam que, quanto mais tarde ocorre a detecção de erros no sistema, mais caro e mais

complexo torna-se a resolução desse problema [2]. Por esse motivo, é importante que, durante o processo de desenvolvimento de software, cada fase seja planejada e implementada, pois se os requisitos especificados não forem bem definidos e implementados, poderão ocorrer prejuízos quantitativos e/ou qualitativos para as organizações desenvolvedoras e clientes envolvidas no processo produtivo de software.

Na literatura especializada, constata-se que o processo de engenharia de requisitos envolve fases fundamentais. Na percepção de Roger Pressman [3], estão presentes nesse processo as fases de elicitação, análise e negociação, especificação e validação dos requisitos. Estas fases podem ocorrer de forma sequencial ou simultaneamente.

Geralmente, o processo de engenharia de requisitos, contém um número expressivo de atividades que poderão ser implementadas de diferentes maneiras. Podem ser apresentadas estratégias e/ou metodologias inovadoras que resultem em um melhor retorno para as organizações.

Observa-se, ainda, na literatura relacionada ao objeto de estudo, a necessidade de pesquisas para integrar o processo de engenharia de requisitos às metodologias ágeis para o desenvolvimento de software. Face ao exposto, é importante a proposição de estratégias e metodologias voltados para simplificar o tradicional processo de engenharia de requisitos.

Entre as principais metodologias ágeis que podem ser aplicadas em um projeto de desenvolvimento de software, destacam-se: Programação extrema (XP, *eXtreme Programming*), Scrum, desenvolvimento dirigido a funcionalidades (FDD, *Feature Driven Development*), *Adaptive Software Development* (ASD), Framework de solução da Microsoft (MSF, *Microsoft Solution Framework*), Método de desenvolvimento de sistemas dinâmicos (DSDM, *Dynamic System Development Method*) e a família Crystal.

Assim, não há uma melhor metodologia. A escolha de uma metodologia deverá ser realizada considerando-se o contexto e a problemática de cada organização.

No geral, os princípios fundamentais dessas metodologias ágeis são: economicidade de tempo e recursos, simplicidade, interação com o cliente e projetos enxutos. Os supracitados princípios podem gerar melhorias na participação de organizações nos processos competitivos que requeiram a entrega de produtos em menos tempo.

Desse modo, torna-se pertinente aplicar essas metodologias ágeis na engenharia de requisitos com o propósito de reduzir o tempo para o desenvolvimento do software.

Com o intuito de contribuir para a melhoria da etapa de engenharia de requisitos no desenvolvimento de software, este trabalho propõe uma estratégia para contribuir com o processo de engenharia de requisitos na Metodologia Scrum.

Para tanto, foi estabelecido o seguinte problema da pesquisa: Como propor uma estratégia capaz de representar e pontuar atividades e/ou artefatos dos fluxos de atividades da estratégia ÁGIL-PERSCRUM, de modo a possibilitar uma visualização e diagnóstico da situação atual de uma

organização em relação ao processo de integração da engenharia de requisitos com as atividades da metodologia ágil Scrum? O objetivo geral da presente investigação, por conseguinte, visou propor uma estratégia capaz representar e avaliar esse mencionados fluxos de atividades e/ou artefatos.

Em termos dos objetivos específicos buscou-se: a) Descrever à luz da literatura especializada o processo de engenharia de requisitos; b) Descrever a metodologia ágil Scrum, com ênfase nos seus princípios e práticas, na estruturação dos membros da equipe Scrum, nos eventos existentes e nos artefatos gerados; c) Criar modelos no Draw.io para descrever os fluxos de atividades propostos para o processo de engenharia de requisitos aplicado ao Scrum; d) Propor critérios para priorização dos requisitos; e) Apresentar exemplo de avaliação (pontuação) das atividades nos fluxos de atividades e/ou artefatos propostos na presente estratégia.

Este trabalho tem relevância acadêmica e econômica. No âmbito acadêmico, trata-se de uma proposta inédita que deverá contribuir de algum modo para um melhor entendimento da necessidade de realização de novas pesquisas no sentido de desenvolver estratégias para integrar as atividades de engenharia de requisitos no âmbito da metodologia ágil Scrum.

Há um grupo de pesquisa na Universidade Estadual do Sudoeste da Bahia (UESB) desenvolvendo o projeto de pesquisa Adoção da ÁGIL-PERSCRUM para direcionar o processo de engenharia de requisitos aplicada à metodologia ágil Scrum no âmbito de sistemas de informações da UESB.

A estratégia ÁGIL-PERSCRUM está em fase inicial. Os artefatos estão sendo gerados e, gradativamente, submetidos para à avaliação da equipe de desenvolvimento de software da UESB. A pretensão é desenvolver uma estratégia que possa ser utilizada em organizações públicas e privadas. Busca-se, por conseguinte, gerar um produto de fácil compreensão, aplicabilidade, que incorpore valor para as organizações e contribua para pesquisas futuras na importante tarefa de integrar a engenharia de requisitos ao processo ágil Scrum.

No âmbito econômico, a redução de tempo e do custo, por exemplo, ganham relevância em um contexto onde as organizações produtivas buscam contenção das suas despesas relacionadas a criação de produtos de software.

Para facilitar a leitura deste trabalho, o mesmo contém, além desta seção introdutória, as seguintes seções: Referencial teórico, Metodologia, Apresentação da estratégia proposta e Conclusão.

II. TRABALHOS RELACIONADOS

A seguir serão citadas algumas publicações relacionadas como objeto de pesquisa em questão.

Um importante trabalho foi desenvolvido por Santos [4], intitulado “REACT: uma abordagem ágil de apoio ao processo de desenvolvimento de requisitos de software baseada em evidências empíricas”. Considerou-se que a adoção e a criação de novas abordagens ágeis na indústria

de software têm obtido significativo crescimento nas últimas décadas em função da busca por soluções para problemas de uso no contexto atual de abordagens tradicionais para desenvolvimento de software, notadamente no que se refere ao processo de engenharia de requisitos. Nessa perspectiva, Santos [4] apresentou na sua dissertação de Mestrado o método REACT (*Requirements Evolution in Agile Context*), uma proposta de metodologia ágil para o desenvolvimento dos requisitos de um produto de software, considerando os modelos de qualidade MR-MPS-SW e CMMI-DEV.

Daneva et al. (2015 apud SANTOS [4]) anotou que no Agile-RE as atividades do processo de engenharia de requisitos são executadas de modo simultâneo, ou seja, duas ou mais atividades do processo podem ocorrer ao mesmo tempo. A Agile-RE é diferente da engenharia de requisitos tradicional quanto à: aceitação de mudanças nos requisitos durante o ciclo de desenvolvimento; produção simplificada de documentações; evolução e detalhamento dos requisitos ao longo de curtas iterações de desenvolvimento [4] e [5]. O trabalho de Santos [4] apontou ainda a existência de desafios na Agile-RE, a exemplo dos identificados mediante mapeamentos ou revisões sistemáticas da literatura, realizadas por Medeiros et al. [6], Inayat et al [7], Elghariani e Kama [8], Jaqueira et al [9], Keikkila et al. [10], Curcio et al. [11] e Fernández et al. [12].

Exemplos desses desafios: Pouca disponibilidade do cliente; Requisitos não funcionais são negligenciados; Controle de mudanças nos requisitos ineficiente; Rastreabilidade de requisitos ineficaz; User stories são inadequadas para descrever aspectos técnicos; Falta de consenso na negociação ou priorização dos requisitos; Problemas na documentação; Ineficiência na análise e inspeção dos requisitos; Problemas na comunicação entre o cliente e a equipe; Problemas com cálculo de custos e estimativas dos projetos devido às mudanças constantes nos requisitos.

Muitos métodos e práticas ágeis têm requerido pesquisas com o objetivo de gerar outras abordagens adaptadas e fundamentadas. É preciso preencher alguns gaps científicos e resolver alguns desafios existentes nos métodos ágeis atuais [13]. Exemplos desses desafios são encontrados nas abordagens: ScrumS, uma adaptação do Scrum para área de *Safe Agile Development* (Esteves et al.) [14]; SobA (*Story based Agile software development*), uma metodologia de desenvolvimento centrada em user stories (Kumar e Norris) [15]; e a *Brew Model*, uma técnica ágil para a priorização de requisitos [16].

Ademais, é importante lembrar, também, das abordagens criadas para o contexto de organizações de grande porte, principalmente, tendo o Scrum como fundamento, tais como: o SAFe (*Scaled Agile Framework*) [17] e o LeSS (*Large-Scale Scrum*) [18].

O estudo conduzido por [6] ressalta que uma SRS (*Software Requirements Specification*) mal estruturada age como catalizador de outros problemas ao longo do projeto, principalmente, em contextos ágeis. Os autores inferem que em ambientes ágeis as SRS são, geralmente, superficiais, insuficientes e inadequadas. Neste cenário, a autora propôs uma abordagem ágil para SRS, a qual se utiliza de um conjunto de boas práticas da Agile-RE como Agile

Modeling, Protótipos de telas e Cenários de aceitação a fim de fornecer uma SRS ágil com foco na equipe de desenvolvimento. Segundo a autora, a proposta foi aplicada na indústria de software por meio de um estudo de caso.

Similarmente, o estudo conduzido por Boness e Harrison [19] teve como objetivo propor uma técnica ágil para a elicitação de requisitos, a qual foi criada com base nos conceitos e práticas da GORE (*Goal Oriented Requirements Engineering*) e do *Goal Sketching*. A ideia foi gerar um diagrama composto de diferentes tipos de metas, as quais guiassem a elicitação de requisitos de forma iterativa e incremental. A técnica, segundo os autores, foi aplicada e validada em três projetos reais de desenvolvimento de software.

Em pesquisa conduzida por Racheva et al. [20] anotou-se o objetivo de aplicar um modelo conceitual para priorização ágil de requisitos. O modelo foi validado por meio de um estudo de caso, o qual contou com a participação de diversos profissionais de companhias de desenvolvimento de software da Holanda, Itália, Bulgária e Turquia.

III. PROCESSO DE ENGENHARIA DE REQUISITOS

A seguir serão apresentadas informações sobre o processo de engenharia de requisitos. Destaca-se, inicialmente, que mesmo com o avanço da tecnologia nos últimos anos, problemas quantitativos e qualitativos ainda persistem nos projetos de software.

Os principais problemas durante o desenvolvimento de um software estão associados aos requisitos. Cerca de 40% a 60% dos problemas que ocorrem nos projetos de software são causados por falhas na fase de levantamento destes. [21].

Constata-se que quando as etapas que compõem o ciclo de vida do processo de engenharia de requisitos são bem definidas os requisitos podem ser melhor especificados, o que poderá gerar, conseqüentemente, melhorias no processo de desenvolvimento de software [22].

Segundo o Standish Group [23], no *Chaos Report*, os principais fatores de falhas em projetos de software são derivados de erros associados aos requisitos, implicado em última análise na qualidade final do software.

Em pesquisa realizada por Schwaber [24] em mais de cem projetos de desenvolvimento de software, identificou-se que 35% dos requisitos sofreram alterações, 65% das funcionalidades descritas pelos requisitos nunca ou raramente foram implementadas e que em torno de 50% do tempo do projeto foi usado no processo de engenharia de requisitos, definição e implantação de arquitetura de software e especificação do produto.

Quanto mais tarde se descobre um erro em requisitos, maior é o custo para fazer sua correção. Quando o erro é descoberto depois do processo de implementação, seu custo de correção pode ser de até 20 vezes maior do que se fosse descoberto em qualquer outra fase no processo de desenvolvimento do software [25].

Face ao exposto, são plenamente justificáveis pesquisas que associem a área de engenharia de requisitos com os métodos ágeis. No entendimento de Santos [4], essa

associação supracitada, conhecida como Engenharia de Requisitos Ágil (AgileRE - *Agile Requirements Engineering*), tem sido objetivo das investigações de diversos pesquisadores. Ademais, Santos [4] ressalta que nem todos os métodos ágeis aplicados ao desenvolvimento de software possuem foco no processo de engenharia de requisitos.

Geralmente, implementar as atividades de engenharia de requisitos mediante métodos ágeis é um grande desafio [7] e [28]. Na literatura são encontrados vários termos que definem requisitos. Para Young [29], requisito é um atributo necessário em um sistema que ajuda identificar a capacidade, características ou a qualidade deste para ter um valor ou utilidade para o usuário. O conceito de requisitos é importante, porque fornece a base para melhor compreensão do trabalho de desenvolvimento de software.

Mesmo que muitas definições de requisitos destaquem as necessidades dos usuários finais, ao defini-los deve-se levar em consideração todas as pessoas que participam do sistema, ou seja, aqueles que interagem direta ou indiretamente com o projeto de software: clientes, gerente de projeto, analistas de sistema, usuários finais, desenvolvedores, etc. [30]

Para desenvolver um software é necessário um conjunto de ações para compreender as expectativas do cliente, analisar e especificar, gerenciar, verificar e validar os requisitos [29]. Em conformidade com o que foi explicado, Sommerville [31] e [32] afirmaram que essas atividades são chamadas de processo de engenharia de requisitos, no qual, as mesmas, durante o desenvolvimento do software, podem ser realizadas de maneira interativa.

Ressalta-se que há várias propostas para as atividades que podem compor o processo de engenharia de requisitos. Nesta parte deste trabalho, apresenta-se uma visão inicial deste processo.

Por exemplo, a Figura 1 ilustra um dos vários tipos de espirais para o processo de engenharia de requisitos. Ressalta-se que na literatura especializada, há exemplos de espirais com mais de quatro fases.

No caso em questão, da Figura 1, no começo do processo, a equipe responsável pelo desenvolvimento do software trabalha em conjunto com os clientes e usuários finais para descobrirem suas necessidades, na fase denominada de Elicitação de Requisitos.

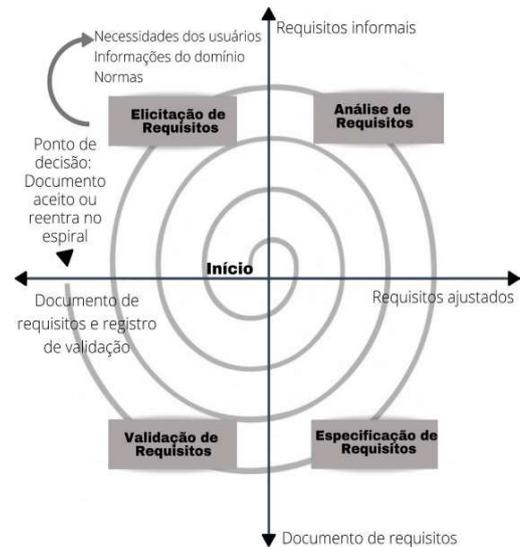
Seguindo a espiral, começa a fase de Análise de Requisitos, que é o processo onde os requisitos serão analisados e ajustados. Geralmente, nesta fase também são realizadas as atividades de Negociação de Requisitos.

Na terceira fase do processo, ocorre a Especificação dos Requisitos, momento no qual os requisitos ajustados passam a compor um documento de requisitos mais elaborado.

Na última fase, apresentada na Figura 1, o documento de requisitos será verificado e validado. Caso o documento não seja validado, será realizado um novo ciclo no espiral.

Mudanças ou interrupções nos requisitos podem ocorrer desde a atividade de elicitação até durante a operação do sistema. Por isso, a atividade de gerenciamento de requisitos é essencial para integrar as outras quatro fases mencionadas.

Figura 1 - Modelo espiral do processo de engenharia de requisitos



Fonte: Adaptado de Sawyer e Kotonía [33].

Para Sommerville e Sawyer [34] (1997), o termo “engenharia” acarreta na necessidade de utilizar técnicas, práticas e conceitos para garantir a completude, consistência e relevância dos requisitos.

Carvalho e Chiossi [35] definem engenharia de requisitos como um processo que transforma ideias de usuários (entrada) em um documento (saída), que fornece a todos os participantes uma descrição do que o produto deverá fazer sem, no entanto, informar como deve ser feito. Isso pode ser desenvolvido por meio de modelos de análise ou especificação, cenários de usuários ou listas de funções e de características, por exemplo.

A engenharia de requisitos propicia aos engenheiros de software um método para entender e analisar as necessidades dos usuários para que assim possam proporcionar soluções para a construção do software [36].

IV. METODOLOGIA SCRUM

O Scrum é um processo iterativo e incremental cujos princípios são utilizados para orientar atividades com o intuito de produzir um serviço de qualidade em um ambiente passível de modificações. Sommerville [32] anotou três fases para o Scrum. A primeira é uma fase básica de planejamento, na qual se estabelece os objetivos gerais do projeto e se projeta a arquitetura do software. Na sequência, é realizada uma série de ciclos de sprint, em que cada ciclo desenvolve uma funcionalidade do sistema. Finalmente, a fase de encerramento finaliza o projeto, completa a documentação necessária, como quadros de ajuda do sistema e manuais do usuário, e avalia as lições aprendidas com o projeto.

Os três procedimentos principais do processo Scrum são: backlog, sprints e reuniões.

Neste trabalho o termo sprint será usado com o sentido de ciclo e o Backlog como um conjunto de itens.

O recurso inovador da metodologia Scrum é sua fase central, ou seja, os ciclos de sprint. Uma Scrum Sprint é uma unidade de planejamento na qual o trabalho a ser feito é avaliado, os recursos são selecionados para o desenvolvimento e o software é implementado.

No final de um ciclo sprint, a funcionalidade concluída é entregue às partes interessadas [32].

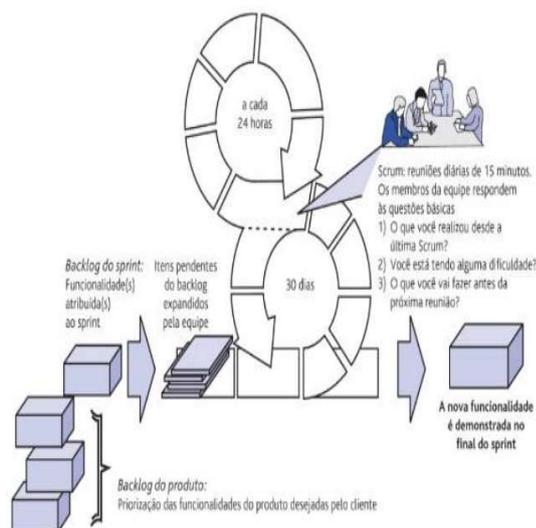
Todos os requisitos necessários são registrados no backlog e contém todas as funcionalidades melhorias e bugs do sistema. Um sprint tem duração aproximada de 2 a 4 semanas de desenvolvimento, e é baseada nas informações do backlog.

Durante um sprint, incluindo a participação do cliente, a equipe passa por várias reuniões que servirão como base para o planejamento do próximo sprint.

Se um sprint relacionado a um backlog está sendo desenvolvido, nenhum requisito desse mesmo backlog poderá ser modificado até que esse sprint acabe. Porém, ainda há flexibilidade para mudanças de requisitos do próximo sprint a ser desenvolvido [37].

O fluxo do processo Scrum é mostrado na Figura 2.

Figura 2 - Fluxo do processo Scrum



Fonte: Pressman e Maxim [38], 2016

V. PROCEDIMENTOS METODOLÓGICOS

Os procedimentos metodológicos escolhidos para nortear a realização da presente investigação foram baseados no paradigma construtivista. Destacar-se-á a seguir, a natureza e o paradigma da pesquisa, os principais instrumentos para construção da estratégia proposta para o processo de engenharia de requisitos (ÁGIL-PERSCRUM).

Em relação aos tipos de pesquisas quanto aos objetivos, em uma primeira etapa da investigação, foi preciso fazer uso da pesquisa exploratória com o propósito

de aproximar os(as) pesquisadores(as) do objeto de estudo, considerando a problemática apontada como desafiadora, posto que se trata de um objeto de estudo bem recente na variante estratégica escolhida para compor a ÁGIL-PERSCRUM. Assim, foram necessárias consultas bibliográficas, em livros, teses, dissertações, monografias, artigos e outras fontes secundárias.

Para descrever as características da estratégia proposta ÁGIL-PERSCRUM, tornou-se necessária uma pesquisa descritiva, modalidade que tem como finalidade descrever as características do objeto de estudo, especificando os atributos das variáveis nele envolvidas [39], [40] e [41].

A conduta baseada em realizar pesquisa exploratória seguida de descritiva confere flexibilidade ao planejamento e é adotada, notadamente, em investigações que se transformam em estudo de caso ou pesquisa bibliográfica. Por isso, segundo Granda [42], é pertinente essa correlação de pesquisas, especialmente para trazer nova luz sobre um tema desafiador e ainda muito recente.

Registra-se que quanto à abordagem, foi escolhida a qualitativa, que serviu para analisar aspectos subjetivos relacionados ao objetivo de estudo. Segundo Chizzotti [41], a abordagem qualitativa parte do fundamento de que há dinâmica entre o mundo real e o sujeito, uma interdependência entre o sujeito e o objeto, um vínculo indissociável entre o mundo objetivo e a subjetividade do sujeito. O conhecimento não se reduz a um rol de dados isolados, conectados por uma teoria explicativa; o sujeito-observador é parte integrante do processo de conhecimento e interpreta os fenômenos, atribuindo-lhes um significado.

Em relação aos tipos de pesquisas quanto aos procedimentos, a construção da estratégia da ÁGIL-PERSCRUM demandou procedimentos que perpassaram pelas pesquisas bibliográfica, de levantamento e pelo estudo de caso com entrevista com membros do projeto GED-UESB (Gerenciamento Eletrônico de Documentos) e integrantes da equipe da Unidade Organizacional de Informática (UINFOR), da UESB.

Quanto às etapas da pesquisa foram estabelecidas para o Estudo de Caso as seguintes:

1 – Seleção do caso a ser estudado (Reflexão) - Escolha do tema - Escolha das questões de pesquisa - Seleção de perspectivas paradigmáticas;

2 – Planejamento do Estudo de Caso - Seleção do contexto - Seleção de estratégia (qualitativa) - Realização de pesquisas preliminares - Escrita preliminar da metodologia (projeto);

3 – Condução do Estudo de Caso - Seleção das fontes de pesquisa - Coleta de dados e informações – Modelagem usando BPMN (*Business Process Model and Notation*) na ferramenta online Draw.io.;

4 – Estabelecimento de conexões de dados e informações - Análise preliminar de dados e informações;

5 – Escrita do caso - Análise intensiva dos dados e informações; e

6 – Apresentação dos resultados - Redação final do caso, adaptado de Carvalho [43]

No que diz respeito a esta pesquisa sobre a ÁGIL-PERSCRUM, utilizou dos seguintes instrumentos: a) Software Draw.io modelagem organizacional; b)

Entrevista estruturada como os desenvolvedores do Sistema GED (Gerenciamento Eletrônico de Documentos) e da Unidade Organizacional de Informática da UESB); c) Uso de abordagem adaptada do SCAMPI (*The Standard CMMI Appraisal Method for Process Improvement*).

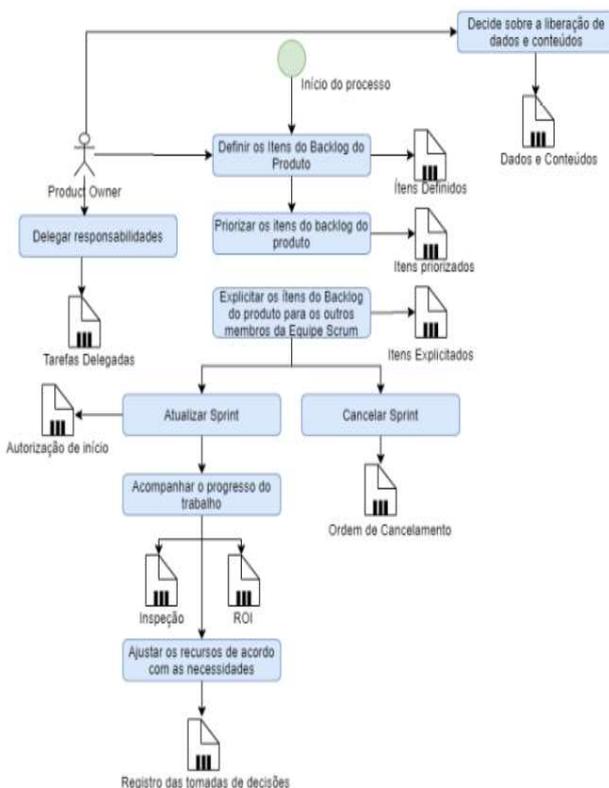
As etapas de coleta e análise de informações seguiram algumas etapas requisitos propostos na Norma IEEE 830/98 que dispõe sobre a Especificação de Requisitos para Desenvolvimento de Software. Considerando-se o escopo do trabalho voltado apenas para a fase inicial de modelagem, as informações coletadas foram baseadas na proposta da ÁGIL-PERSCRUM, visando aprimorar a compreensão, visibilidade e construção do processo de engenharia de requisitos no escolhido método ágil. Para tanto, fez-se uso das técnicas de observação e entrevista utilizadas, que resultaram nos modelos gerados na estratégia ÁGIL-PERSCRUM.

VI. ESTUDO DE CASO: RESULTADOS E DISCUSSÃO

A seguir, serão apresentados os principais resultados do estudo de caso.

a) Atividades propostas para o Proprietário do produto

Figura 3 – Atividades do Proprietário do produto



Fonte: Sales (2021) [44]

Para tornar as informações mais compreensíveis, inicialmente, o ÁGIL-PERSCRUM, exibe fluxos de atividades para os principais participantes do ciclo Scrum,

destacando as atividades propostas para o Proprietário do produto, o Scrum master e Equipe de desenvolvimento.

A Figura 3 mostra que o ator Proprietário do produto desenvolve atividades que geram artefatos.

A ÁGIL-PERSCRUM propõe que o fluxo do ator mencionado será flexível e para que cada organização possa incorporar atividades e artefatos.

As atividades e/ou artefatos presentes nos fluxos mapeados podem ser pontuados. A ÁGIL-PERSCRUM é, portanto, flexível ao contexto, ou seja, a realidade de cada organização.

Ressalta-se, que visando explicar o funcionamento da ÁGIL-PERSCRUM, este trabalho apresentará apenas algumas figuras relacionadas com o Proprietário do produto e Scrum master, por considerar suficientes para o entendimento do funcionamento da estratégia supracitada. Portanto, não serão exibidos fluxos associados com a equipe de desenvolvimento, clientes e outras partes interessadas.

Consta-se na Figura 3 que compete ao Proprietário do produto definir os itens que vão compor o Backlog do Produto, priorizar os itens, explicar para os demais membros do Scrum os itens em questão, atuar ou cancelar o sprint e acompanhar o progresso do trabalho e proceder aos ajustes nos recursos utilizados no sprint, bem como decidir sobre a liberação de dados e conteúdos.

No exemplo do fluxo da Figura 3, foram gerados os seguintes artefatos: itens definidos, itens priorizados, itens explicitados, autorização de início ou ordem de cancelamento do sprint, tarefas delegadas, dados e conteúdos, inspeção, retorno operacional do investimento (ROI) e registro das tomadas de decisões.

Assim, na estratégia ÁGIL-PERSCRUM, são criados os fluxos de atividades e gerados os artefatos.

Como descrito anteriormente, cada organização poderá verificar se as atividades e/ou artefatos dos fluxos aqui explicitados estão presentes nas práticas do Scrum adotadas na organização.

Após o Proprietário do produto definir os itens do Backlog do Produto, é preciso priorizar os itens.

A ÁGIL-PERSCRUM tem uma atenção especial com a atividade de priorização de itens, considerando-a essencial para as atividades seguintes do Scrum.

A estrutura do Scrum possui vários eventos internos (rituais) que buscam garantir a priorização razoável de histórias e tarefas do usuário [45]. A priorização de histórias de usuários e suas tarefas associadas devem ser realizadas de modo contínuo, visando garantir que a equipe Scrum trabalhe corretamente.

A atividade de priorização deve abranger todos os eventos Scrum anteriormente citados. De acordo com Daneva et al. [45] deve-se preparar a lista de pendências geradas no evento de preparação e iniciar o uso da metodologia Scrum, a partir da criação do Backlog do Produto.

Na reunião de planejamento do sprint, verificar-se-á as dependências entre as tarefas, incluindo, evidentemente, a construção e análise das histórias de usuários.

Durante a reunião diária, tanto as tarefas quanto os itens constantes no Backlog do produto devem ser

operacionalizados de acordo com priorização realizada no evento anterior.

Sabe-se que na reunião de revisão do sprint novas análises são realizadas para gerar entradas para a reunião de retrospectiva do sprint, momento no qual a equipe de Scrum poderá priorizar outros aspectos considerados essenciais.

Daneva et al. [45] apresentaram considerações importantes para a priorização dos requisitos de software. Defenderam seu ponto de vista sobre o assunto, afirmando que é preciso escrever os casos de uso essenciais e discutí-los com os arquitetos, clientes, representantes e outras partes interessadas (equipe de desenvolvimento e *stakeholders* diretos e indiretos). Propôs que a estruturação dos casos de uso de alto nível e requisitos sejam escritos durante o Backlog do Produto para que em seguida ocorra a fase de estimativa e as sessões de priorização dos requisitos.

Recomenda-se, no ÁGIL-PERSCRUM, que a equipe de desenvolvimento tenha uma correta compreensão dos novos conceitos associados ao produto que está sendo criado. Nesse sentido, o Proprietário do produto, deverá classificar os requisitos e priorizá-los.

Desse modo, em conformidade com a Figura 3, é válido exibir uma sequência lógica contendo atividades desenvolvidas pelo Proprietário do produto. Entre as mesmas já foram destacadas a definição dos itens e priorização do Backlog do Produto, estas seguidas da explicitação para os membros da Equipe de Desenvolvimento. Ressalta-se, inclusive, a possibilidade do Proprietário do produto delegar atividades, autorizar ou cancelar um ciclo sprint.

A priorização de requisitos, por conseguinte, conforme ressaltado anteriormente, é essencial na Estratégia da proposta do ÁGIL-PERSCRUM. Pesquisadores e pesquisadoras têm tratado da priorização de requisitos, inclusive, na metodologia Ágil Scrum [46], [47], [48], [49], [50], [51], [52], [53] e [54], por exemplo.

b) Atividade do Scrum master

O Scrum master, além de ter o embasamento técnico sobre o processo Scrum, é responsável pela sua correta efetivação e maximização das suas capacidades, ou seja, é ele quem vai fazer o Scrum funcionar. O Scrum define práticas, reuniões, artefatos e terminologias.

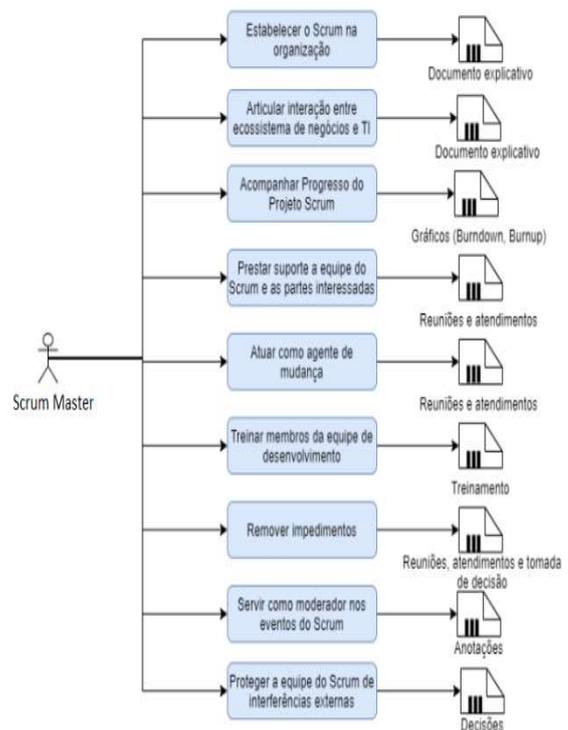
Para Schwaber [55], o Scrum master é responsável por esses conhecimentos e como aplicá-los corretamente. Pelo fato do Scrum ter a possibilidade de gerar falhas, o Scrum master poderá clarear o entendimento sobre seu funcionamento e guiar o projeto Scrum. O Scrum master atende a todos os participantes de um projeto Scrum e as partes interessadas externas para compreender e aplicar o Scrum Framework corretamente.

É um papel que apoia a equipe Scrum para executar o Scrum Framework com sucesso, posto que contribui para melhorar sua produtividade e desempenho continuamente.

O Scrum Master possui uma visão mais detalhada para estabelecer o processo Scrum na organização, reportando a uma nova maneira de pensar e agir; atua como agente de mudanças; tem conhecimento de programação;

treina a equipe para desenvolver novas normas e padrões de equipe; estabelece o Scrum Framework em seu ecossistema de negócios e Tecnologia da Informação; atua como agente de mudança e oferece suporte na adaptação dos processos existentes para maximizar a produtividade da equipe Scrum; treina a equipe Scrum para entender e vivenciar os valores do Scrum Framework; garante colaboração eficiente e estreita entre o Proprietário do produto e o Scrum Team; remove impedimentos que dificultam a continuidade do trabalho; lidera o progresso do trabalho; modera os rituais do Scrum (eventos do Scrum); protege a equipe Scrum de ameaças externas, interferências e interrupções enquanto a equipe funciona.

Figura 4 – Atividades do Scrum master



Fonte: Sales (2021) [44]

Em conformidade com Figura 4, no ÁGIL-PERSCRUM, podem ser atividades do Scrum master: estabelecer o Scrum na organização; promover articulação entre ecossistemas de negócio e a área de tecnologia da informação; acompanhar progresso do projeto Scrum; prestar suporte à equipe do Scrum e partes interessadas, atuar como agente de mudança, treinar os membros da equipe de desenvolvimento, remover impedimentos, atuar como moderador nos eventos do Scrum e proteger a equipe do Scrum de interferências externas.

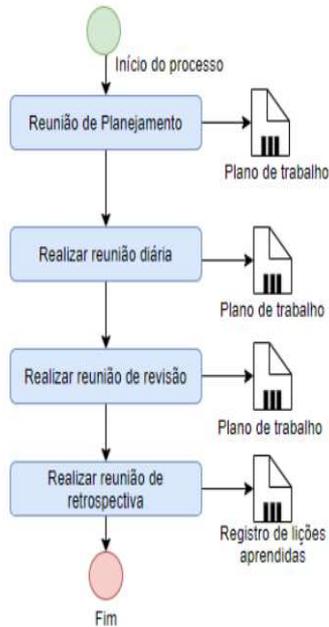
Para cada atividade exibida na Figura 4, há geração de artefatos. Destacam-se: documento explicativo, gráficos (burndown e burnup), reuniões, treinamentos, anotações e decisões.

Em atividades de pesquisa do Projeto de Pesquisa Adoção da ÁGIL-PERSCRUM para direcionar o processo de engenharia de requisitos aplicada à metodologia ágil Scrum no âmbito de sistemas de informações da UESB, busca-se integrar a metodologia Scrum com práticas de gestão de projetos [56], [57], [58], [59], [60], [61], [62],

[63], [64], [65], [66], [67], [68], [69], [70], [71], [72], [73], [74], [75], [76], [77] e [78].

Na estratégia ÁGIL-PERSCRUM, os eventos principais do Scrum geram artefatos, conforme mostra a Figura 5.

Figura 5 – Eventos do Scrum



Fonte: Sales, 2021 [44]

É necessário verificar se os artefatos exibidos na Figura 6 estão presentes na organização. Ressalta-se, mais uma vez, que a estratégia ÁGIL-PERSCRUM também é flexível em relação aos conteúdos dos artefatos gerados em cada um dos eventos da metodologia Scrum.

Essa flexibilidade está em conformidade com as visões de pesquisadores e pesquisadoras que são favoráveis à implantação de visão moderna e inovadora para os negócios, a exemplo da publicação de Jim Highsmith e Alistair Cockburn [79].

A estratégia ÁGIL-PERSCRUM destaca que é preciso planejar, executar, acompanhar e avaliar cada um dos eventos Scrum (Reunião de Planejamento, Reuniões Diárias, Reuniões de Revisão e de Retrospectiva).

A Figura 6, propõe uma sequência de atividades e seus correspondentes artefatos gerados. Trata-se de um detalhamento em um segundo nível de representação dos fluxos da metodologia Scrum.

Buscou-se integrar as atividades básicas do processo tradicional de engenharia de requisitos [80], [81], [82], [83], [84], [85] e [86] com os eventos e algumas atividades da metodologia Scrum.

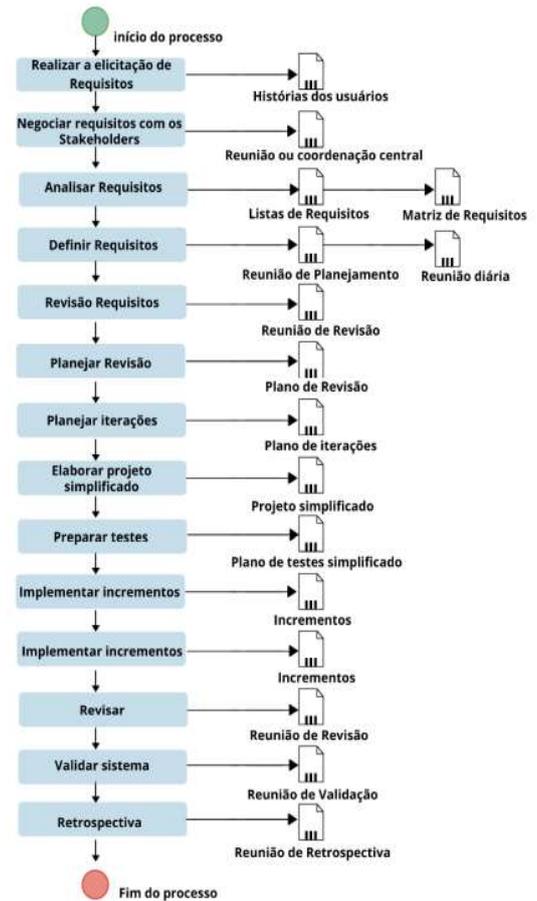
Na fase de elicitação de requisitos, podem ser gerados, por exemplo o artefato histórias dos usuários (*user stories*) ou outros escolhidos pela organização.

Percebe-se, assim, que a integração entre as atividades tradicionais da engenharia de requisitos e as atividades da metodologia Scrum devem primar pelo bom senso, pela flexibilidade no processo de escolha tanto das atividades quanto dos artefatos.

A estratégia ÁGIL-PERSCRUM não é uma estrutura de receita pronta. Pelo contrário, tem o objetivo de representar, tornar visível e mais compreensível o processo de integração da engenharia de requisitos com a metodologia Scrum.

Requer propor, representar fluxos de atividades e artefatos, visualizar o que está sendo realizado, documentar e ajudar no processo de cumprimento dos objetivos de desenvolvimento de um produto.

Figura 6 – Integração das fases da engenharia de requisitos com a metodologia Scrum



Fonte: Sales, 2021 [44]

Verifica-se, portanto, que, de um modo geral, cada organização poderá adequar os fluxos dos eventos às suas necessidades organizacionais, ora acrescentando atividades, ora suprimindo.

Assim, a Figura 6 é um exemplo de um fluxo mais detalhado que poderá ser aplicado às organizações e um segundo nível de aprofundamento de detalhamento.

Todavia, deve-se ainda observar que o gerenciamento de requisitos é uma atividade integrante do processo de desenvolvimento de software. Os requisitos elicitados de software precisam ser tratados por meio de comunicação e integração adequadas entre as partes interessadas, tanto no âmbito do desenvolvimento tradicional quanto no ágil.

As metodologias ágeis são constituídas de processos inovadores e iterativos [79] que suportam requisitos variáveis e ajudam a lidar com as mudanças ao

longo do processo de desenvolvimento de software e que requer priorização.

Os requisitos são elicitados no início de cada processo e projeto de desenvolvimento de software (produto) e estes são priorizados de acordo com sua importância para o mercado e para o próprio produto. Uma das etapas mais importantes e influentes ao criar um produto de software é a priorização de requisitos.

A priorização de requisitos ajuda a equipe de software a entender a existência e a importância de um requisito específico, sua relevância de uso e sua urgência de lançar no mercado.

Existem muitas técnicas de priorização de requisitos com seus pontos fortes e fracos. Muitas delas não consideram todos os fatores que devem ser considerados ao priorizar requisitos, a exemplo do custo, valor, risco, tempo de colocação no mercado, número de requisitos e efeito de requisitos não funcionais nos requisitos funcionais.

Há, portanto, várias metodologias de priorização de requisitos que auxiliam na tomada de decisões. Entretanto, são necessários mais pesquisas para explicar os fatores que exercem influência significativa na priorização de requisitos [87].

Em conformidade com a literatura especializada, a exemplo das recomendações de Sutherland [54], podem estar presentes no processo de desenvolvimento etapas associadas diretamente ou indiretamente ao processo de priorização de requisitos.

Sutherland [54], sugeriu as seguintes etapas:

Etapas 1 - Inspeccionar: fazer o que for preciso para entender o projeto atual;

Etapas 2 - Adaptar: indicar uma direção e visão sobre os próximos passos do projeto e, em seguida, criar estratégias e executar essa visão;

Etapas 3 - Aprenda: observar, aprender e ensinar um ao outro;

Etapas 4 - Reiniciar: desenvolver novas iterações, caso necessárias.

Geralmente, podem ser realizadas atividades de identificação de quais tarefas serão necessárias para cumprir os objetivos de cada sprint, identificação e priorização do que é essencial, desenvolver os melhores artefatos, produtos e serviços de software, em conformidade com as expectativas dos clientes.

Estratégias para priorização de requisitos na ÁGIL-PERSCRUM

Primeira estratégia

Usar pontuação tradicional, a exemplo de prioridade baixa (1 ponto), média (2 pontos) e alta (3 pontos) ou quaisquer outras possibilidades de pontuação com os seguintes critérios: desejável, importante, essencial ou mais de três critérios de pontuação.

Segunda estratégia

Fazer uso de matriz de comparação de pares de requisitos com cálculo do vetor prioritário para geração do resultado, isto é, de maior percentual na soma dos

requisitos priorizados multiplicado pelo vetor prioritário [88].

Terceira estratégia

A equipe de desenvolvimento estimará a velocidade para desenvolver o requisito que será concluído por iteração. Pode adotar pontuações em várias escalas. Por exemplo de 1 a 5 ou de 1 a 10 ou de 1 a 15 e, assim, sucessivamente. Cada equipe poderá escolher e implementar a escala mais apropriada, tanto para a velocidade de desenvolver o requisito ou para avaliar o custo ou a complexidade.

Todavia, deverá compreender que essa pontuação é específica para cada item, diferentemente da pontuação mais geral para critérios de análise geral.

No estudo de caso aplicado na Universidade Estadual do Sudoeste da Bahia, foi realizada a pontuação para as interações nas histórias dos usuários.

A Quadro 1 representa as histórias e as pontuações em uma escala de 1 a 10.

QUADRO 1 – Pontuação para as iterações das histórias

Interações	Histórias	Pontuação
Iteração 1	A e B	3+3=8
Iteração 2	C, D e E	7+5+6=18
Iteração 3	F e G	6+10=16
Iteração 4	H, J e K	5+4=7

Fonte: Autoria dos pesquisadores, 2022

Registrou-se que a iteração 2 (C, D e E) alcançou a maior pontuação com 18 pontos.

A equipe de desenvolvimento procederá à implementação do requisito e realizará as atividades de testes. Para tanto, essa equipe poderá escrever os testes no verso do cartão de histórias, quer sejam testes com ou sem ferramentas automatizadas.

Ressalta-se que na estratégia aqui proposta, um cartão de história deverá conter uma breve descrição da funcionalidade descrita pelos usuários e/ou pelo cliente.

Além disso, devem ser consideradas algumas recomendações nessa etapa do processo de desenvolvimento do produto:

1) Criar um canal de comunicação entre a equipe do Scrum de modo que a equipe de membros indicados pelo cliente descreva corretamente as histórias;

2) As histórias serão priorizadas;

3) Pode-se subdividir uma história de usuário, sempre que necessária;

4) Após, deve-se proceder aos testes de validação das histórias;

5) Cada equipe Scrum deverá escolher a técnica ou as técnicas para obter as informações das histórias dos usuários, como por exemplo: entrevistas, questionários, observação, prototipação, *design thinking*, UCD (Design centrado no usuário), Story card, casos de uso, cenários JRD (*Joint Requirements Document*), *mind mapping*, workshop, GREM (*Gamified Requirements Engineering Model*).

Ou seja, há liberdade para adequar da técnica a cada contexto de desenvolvimento;

6) A equipe de desenvolvimento e equipe do cliente devem atuar em conjunto na fase de testes. Recomenda-se realizar testes de interface do usuário, usabilidade, desempenho, teste de estresse e testes de aceitação e outros que se fizerem necessários.

Quarta estratégia: uso de um conjunto de critérios propostos na ÁGIL-PERSCRUM

A partir da literatura especializada no assunto, a exemplo de considerações de Alkandari e Al-shammeri [87], na ÁGIL-PERSCRUM foi incluída um conjunto de critérios para priorização dos requisitos e tarefas, são eles:

1) Impacto no produto – deve considerar o valor a ser gerado para as organizações externas, sem considerar aqui as novas oportunidades advindas do mercado. Em razão deste condicionante, o impacto do produto para o ambiente externo deve considerar as demandas históricas e as atuais em relação, a partir do fluxo regular de solicitações de desenvolvimento de software.

2) Restrições – devem ser avaliadas desde a concepção do projeto no estágio de planejamento da iteração. Estimar as restrições do projeto e da iteração e escolher os requisitos com base na incorporação delas no planejamento da iteração.

3) Escopo – atribuir uma pontuação para a abrangência dos requisitos do projeto.

4) Dependência para realização de outras atividades – Nos projetos de desenvolvimento de requisitos, podem estar presentes as dependências entre eles. Estas têm impacto no ritmo de desenvolvimento e complexidade do projeto. Há dependências cronológicas e arquitetônicas. Caberá à equipe de desenvolvimento e aos membros das demais partes interessadas avaliar a importância das dependências e atribuir uma pontuação para as mesmas.

5) Volatilidade - a tendência dos requisitos de mudar ao longo do tempo em resposta às necessidades em evolução dos clientes, *stakeholders*, organização e ambiente de trabalho deve ser incorporada ao cálculo da priorização dos requisitos. Essa tendência pode levar em consideração aspectos internos e externos. Ressalta-se, assim, que na presente proposta de pontuação cada equipe de projeto poderá incorporar subcritério(s).

6) Risco – é o cálculo do impacto da probabilidade e impacto ou urgência dos requisitos ou tarefas. Os riscos de requisitos estão diretamente associados a requisitos específicos e podem ser riscos de impactos negativos ou positivos.

7) Esforço - deve-se utilizar de estimativas e métricas para calcular o número de membros da equipe que estão responsáveis pelas atividades do processo de engenharia de requisitos.

8) Tempo – é preciso estimar o tempo e atribuir uma pontuação em relação à priorização dos requisitos e tarefas.

9) Débito técnico (*technical debt*) – dívida técnica trata-se de uma metáfora associada às consequências do desenvolvimento de software deficiente, gerando um produto que precisará ser concluído. Descreve as

consequências que os projetos de software enfrentam durante o seu desenvolvimento, ocasionada muitas vezes, quando tarefas não são realizadas adequadamente. Por exemplo, uma elicitação ou análise de requisitos inadequada, ocasiona erros que elevam a incidência de dívida técnica nos projetos de software. O termo foi introduzido em 1992 por Ward Cunningham [89].

10) Oportunidade de uso – se refere às ocorrências advindas do ambiente externo que requerem um reposicionamento do projeto e uma nova priorização de requisitos. Deve-se avaliar ocorrências associadas à concorrência, às mudanças no comportamento do consumidores, alterações legais, etc. Caberá à equipe do Scrum incluir no cálculo de priorização essas ocorrências, pois podem variar no tempo e no espaço.

11) Valor agregado – na literatura especializada há muitas informações sobre o conceito de valor agregado. Neste trabalho, destaca-se apenas que deve ser realizado uma pontuação para o cálculo do valor agregado. Não há uma receita pronta para os critérios que podem ser incorporados a esse critério de priorização. Por exemplo, há casos nos quais a equipe do Scrum poderá incorporar ao cálculo o retorno financeiro, o retorno social, a valorização da marca ou do produto, o posicionamento da organização no mercado, a conquista de novos mercados etc.

12) Custo - trata-se de um dos fatores críticos de priorização, posto que influencia diretamente na seleção dos requisitos acessíveis que poderão ser implementados. Os recursos financeiros devem ser bem dimensionados e incluídos no orçamento do projeto. Geralmente, exercem influência significativa no processo de priorização dos itens integrantes do projeto.

13) Conhecimento da equipe acerca daquilo que será realizado – a experiência associada aos conhecimentos práticos e teóricos são essenciais para priorizar os requisitos. Alguns tipos de requisitos poderão alcançar uma alta pontuação na priorização, caso não requeiram o desenvolvimento de conhecimentos. Porém, para os requisitos que demandem pouco tempo para desenvolver esses conhecimentos é recomendável pontuá-los com média ou baixa prioridade.

14) Gerenciamento de recursos disponíveis – deve-se pontuar o gerenciamento dos recursos do projeto, considerando as fases de identificação, aquisição, gerenciamento da equipe, administração e controle dos recursos necessários para a conclusão bem-sucedida do projeto. Tal atividade englobe, portanto, atividades de planejamento dos recursos necessários (informacionais, computacionais, materiais, equipamentos e humanos), a estimativa de uso de recursos durante o processo de desenvolvimento dos requisitos.

Esta proposta de priorização de requisitos no Scrum abrange:

- 1 - Definição da visão do produto;
- 2 - Elicitação de requisitos com os clientes;
- 3 - Negociar e priorizar requisitos;
- 4 - Definir os requisitos funcionais e não funcionais.

Ressalta-se que em projetos de grande porte, os requisitos precisam ser mais detalhados e contém detalhes que permitam uma devida implementação. Por isso, deve-

se considerar que as histórias dos usuários desempenham um papel muito relevante na priorização dos requisitos.

QUADRO 2 – Exemplo de pontuação dos 14 critérios para priorização

Crítérios	Baixa (1)	Média (2)	Alta (3)
Impacto no produto		2	
Restrições		2	
Escopo		2	
Dependência para realização de outras atividades	1		
Volatilidade	1		
Risco		2	
Esforço (fórmula)		2	
Tempo			3
Débito técnico (<i>technical debt</i>)		2	
Oportunidade de uso			3
Valor agregado			3
Custo			3
Conhecimento da equipe acerca daquilo que será realizado		2	
Gerenciamento de recursos disponíveis	1	2	
Subtotal	3	16	12
Total Geral			31

Fonte: A autoria dos pesquisadores, 2022

Em aplicação inicial em estudo de caso, usando os 14 critérios do Quadro 2, realizado junto a Unidade Organizacional de Informática da UESB, verificou-se em um dos projetos de desenvolvimento de software, que foram alcançados 12 pontos de alta prioridade, 16 de média e 3 de baixa prioridade.

Os itens de maior prioridade devem ser devidamente acompanhados, pois são considerados essenciais para o sucesso do projeto. Essa avaliação deve se adequar a cada projeto e suas especificidades.

Apresenta-se a seguir um exemplo para pontuação da presença de atividades e/ou artefatos nos fluxos, de acordo com a proposta adaptada do SCAMPI (*The Standard CMMI Appraisal Method for Process Improvement*) [76].

Um dos objetivos precípuos da estratégia proposta na ÁGIL-PERSCRUM é pontuar as atividades e/ou os artefatos presentes nos fluxos elaborados no processo de integração entre as atividades do processo de engenharia de requisitos e as atividades da metodologia ágil Scrum.

No estudo de caso na UESB, foi solicitado que os integrantes da equipe Scrum fizessem a pontuação dos artefatos presentes naquela instituição, a partir do fluxo proposto para o Proprietário do produto, conforme Figura 3.

Para tanto, foram consideradas quatro faixas de pontuação, a seguir descritas:

NI – Não Implementado – Pontuação: 0 (0-25%)

PI – Parcialmente Implementado – Pontuação: 1 (26-50%)

LI – Largamente Implementado – Pontuação: 2 (51-75%)

FI – Totalmente Implementado – Pontuação: 3 (76-100%)

QUADRO 3 – Pontuação atribuída aos artefatos gerados na Figura 3.

Artefatos	NI	PI	LI	FI	Total
Dados e conteúdos		1			1
Tarefas delegadas			2		2
Itens definidos			2		2
Itens priorizados			2		2
Itens explicitados			2		2
Autorização/Cancelamento sprint				3	3
Inspeção			2		2
Retorno Operacional do Investimento (ROI)			2		2
Registro das tomadas de decisões		1			1
Total de pontos alcançados					17

Fonte: A autoria dos pesquisadores, 2022.

No exemplo apresentado, foram avaliados apenas 9 artefatos. A pontuação máxima segundo a metodologia adaptada do SCAMPI para esse caso seria: 9 artefatos x 3 (pontuação máxima se todos os itens estivessem totalmente implementados), que resultaria, no máximo, de 27 pontos.

Contudo, no exemplo apresentado, a pontuação máxima alcançada foi de 17 pontos. Usando um cálculo simples de percentual, chegou-se à seguinte conclusão: $(17*100)/27 = 62,96\%$. Vale dizer que, no exemplo utilizado, a organização foi avaliada na faixa de LI – Largamente Implementado – Pontuação: 2 (51-75%), pois alcançou uma pontuação entre 51 a 75%.

De igual modo, poderá cada organização conhecer a situação atual de organização em relação aos fluxos propostos na estratégia ÁGIL-PERSCRUM. Poderá pontuar tanto as atividades quanto os artefatos, em separadamente ou em conjunto.

Os integrantes do Grupo de pesquisa Adoção da ÁGIL-PERSCRUM para direcionar o processo de engenharia de requisitos aplicada à metodologia ágil Scrum no âmbito de sistemas de informação da UESB pretendem, apresentar em publicações futuras, exemplos das variantes de pontuação citadas neste trabalho.

VII. CONSIDERAÇÕES FINAIS

Após as informações até aqui apresentadas, é pertinente citar algumas conclusões relevantes na investigação do grupo de pesquisa responsável por formular a estratégia ÁGIL-PERSCRUM.

Convém reafirmar que as investigações do mencionado grupo de pesquisa em relação aos resultados aqui apresentados buscaram resposta para a seguinte questão de pesquisa: Como propor uma estratégia capaz representar e pontuar atividades e/ou artefatos dos fluxos

de atividades da estratégia ÁGIL-PERSCRUM, de modo a possibilitar uma visualização e diagnóstico da situação atual de uma organização em relação ao processo de integração da engenharia de requisitos com as atividades da metodologia ágil Scrum?

Conforme informações mencionadas, constatou-se que a estratégia proposta conseguiu responder à questão formulada, pois o objetivo geral da investigação foi alcançado. Ou seja, foi possível representar e pontuar os fluxos de atividades e artefatos, incorporar critérios para priorização dos requisitos. Além disso, foi possível também: a) Descrever à luz da literatura especializada o processo de engenharia de requisitos; b) Descrever a metodologia ágil Scrum, com ênfase nos seus princípios e práticas, na estruturação dos membros da equipe Scrum, nos eventos existentes e nos artefatos gerados; c) Criar modelos no Draw.io para descrever os fluxos de atividades propostas para o processo de engenharia de requisitos aplicada ao Scrum; d) Propor critérios para priorização dos requisitos; e) Apresentar exemplo de avaliação (pontuação) das atividades constantes nos fluxos de atividades propostos na presente estratégia.

Este é um trabalho inédito que deverá ser continuado com outras pesquisas com base nos autores já citados anteriormente e outros tantos que estão pesquisando o objeto de investigação ora em questão.

A proposta de estratégia ÁGIL-PERSCRUM, conforme destacado em outras seções deste trabalho, se encontra em fase de aperfeiçoamento e aplicabilidade na Universidade Estadual do Sudoeste da Bahia.

Recomenda-se: a) Aplicar a estratégia ÁGIL-PERSCRUM em organizações públicas e privadas; b) Incorporar novas recomendações para tornar a estratégia ÁGIL-PERSCRUM cada vez mais adaptável aos diferentes tipos de organizações e projetos.

REFERÊNCIAS

[1] Ian Sommerville. “Engenharia de software”. 8. ed. São Paulo: Pearson Addison - Wesley, 2007

[2] P. Jalote. “An integrated approach to software engineering”. (D. Gries & F. B. Schneider, Eds.) (Third). New York: Springer, 2005.

[3] Roger S. Pressman S. “Engenharia de software”. 6. ed. São Paulo: McGraw Hill, 2006.

[4] Kleoson B. C. Santos. “React: uma abordagem ágil de apoio ao processo de desenvolvimento de requisitos de software baseada em evidências empíricas”. Universidade Federal do Paraná, Belém, 2018.

[5] A. Qusef; A de Lúcia “A. Requirements engineering in agile software development. Journal of emerging technologies in web intelligence”, v. 2, n. 3, p. 212-220, 2010.

[6] Juliana Medeiros et al. Requirements engineering in agile projects: a systematic mapping based in evidences of industry. In: CibSE. p. 460, 2015.

[7] Irum Inayat, Siti Salwah Salim, Sabrina Marczak, Maya Daneva, Shahaboddin Shamshirband “A systematic literature review on agile requirements engineering practices and challenges”. Computers in Human Behavior, Elsevier BV, v. 51, Part B, October, 2015, p. 915–929. Disponível em: , [4http://dx.doi.org/10.1016/j.chb.2014.10.046](http://dx.doi.org/10.1016/j.chb.2014.10.046) . 2014

[8] K. Elghariani; N. Kama. “Review on agile requirements engineering challenges”. In: 2016 3rd International Conference on Computer and Information Sciences (ICCOINS). IEEE, 2016. Disponível em: <https://ieeexplore.ieee.org/document/7783267>

[9] A Jaqueira et al. “Desafios de requisitos em métodos ágeis: uma revisão sistemática”. In: 3rd Brazilian Workshop on Agile Methods, São Paulo. 2012

[10] V. T. Heikkila; D. Damian; C. Lassenius; M. Paasivaara. “A mapping study on requirements engineering in agile software development”. In: 2015 41st Euromicro Conference on Software Engineering and Advanced Applications. IEEE, 2015. Disponível em: <https://ieeexplore.ieee.org/document/7302452>.

[11] K. Curccio; T. Navarro; A. Malucelli; S. Reinehr. “Requirements engineering: a systematic mapping study in agile software development”. Journal of Systems and Software, Elsevier BV, v. 139, p. 32–50, may 2018. Disponível em: [//doi.org/10.1016/j.jss.2018.01.036](https://doi.org/10.1016/j.jss.2018.01.036)>.

[12] D. M. Fernández,; S. Wagner; M. Kalinowski; M. Felderer; P. Mafra; A. Vetrò; T. Conte; M. T. Christiansson; D. Greer; C. Lassenius; T. Männistö; M. Nayabi; M. Oivo; B. Penzenstadler; D. Pfahl; R. Prikladnicki; G. Ruhe; A. Schekelmann; S. Sen; R. Spinola; A. Tuzcu; J. L. Vara; R. Wieringa. “Naming the pain in requirements engineering”. Empirical Software Engineering, Springer Nature, v. 22, n. 5, p. 2298–2338, oct 2016. Disponível em: <https://link.springer.com/article/10.1007/s10664-016-9451-7>

[13] J.D.R.V. Medeiros. “An approach based on design practices to specify requirements in Agile Software Development”. Tese (Doutorado). 2017. Universidade Federal de Pernambuco. Recife, PE, Brasil, 2017.

[14] R. Esteves; L. A. Rodrigues; N. A. Pinto. “ScrumS: a model for safe agile development.” 7th International Conference on Management of Computational and Collective Intelligence in Digital EcoSystems. DOI: 10.1145/2857218.2857225. Publisher: ACM. Caraguatatuba, Brazil, October 25 - 29, 2015.

[15] Chetan Kumar; John B. Norris. “A new approach for a proxy-level web caching mechanism.” Decision Support Systems, v. 46, n. 1, p. 52-60, 2008.

- [16] E.C. Negrão; E.M. Guerra. "A case study for prioritizing features in environments with multiple stakeholders." OOPSLA'11: ACM international conference companion on Object oriented programming systems languages and applications companion. DOI: 10.1145/2048147.2048187. Publisher: ACM. Portland, Oregon, USA, October 22 - 27, 2011.
- [17] Dean Leffingwell; Richard Knaster. "SAFe 4.5 "Distilled: applying the scaled agile framework for lean enterprises". Addison-Wesley Professional, 2018.
- [18] C. Larman; B. Vodde. "Large-scale Scrum: more with LeSS". Addison-Wesley Professional, 1 edition, ISBN-10: 0321985710. USA, 2016.
- [19] K. Boness; R. Harrison. "Goal sketching: towards agile requirements engineering". International Conference on Software Engineering Advances (ICSEA 2007). DOI: 10.1109/ICSEA.2007.36. Cap Esterel, France, 2007.
- [20] Zornitza Racheva et al. "A conceptual model and process for client-driven agile requirements prioritization". In: 2010 Fourth International Conference on Research Challenges in Information Science (RCIS). IEEE, 2010. p. 287-298.
- [21] D. Leffingwell. "Calculating the return on investment from more effective requirements management." American Programmer v. 10, n. 4, p. 13-16; 1997.
- [22] Wilson de Pádua Paula Filho. "Engenharia de software." v. 2. Rio de Janeiro: LTC, 2003.
- [23] STANDISH GROUP. "Extreme chaos report". The Standish Group International Inc. (2009). Disponível em: <https://www.standishgroup.com/>. Acesso em: jan. 2021.
- [24] Ken Schwaber. "The enterprise and Scrum". Redmond, WA: Microsoft Press, 2007.
- [25] B. W. Boehm et al. "Software engineering economics: background, current practices, and future directions". Proceedings of the 24th International Conference on Software Engineering. ICSE 2002. Florida: IEEE Computer Society, 2002.
- [26] M. Daneva et al. "A reflection on agile requirements engineering: solutions brought and challenges posed". In: XP 2015 Workshops, Helsinki, Finland. 2015.
- [27] Lan Cao; Balasubramaniam Ramesh. "Agile requirements engineering practices: An empirical study". IEEE software, v. 25, n. 1, p. 60-67, 2008.
- [28] E. Bjarnason; K. Wnuk; B. Regnell. "A case study on benefits and side-effects of agile practices in large-scale requirements engineering". In: Proceedings of the 1st Workshop on Agile Requirements Engineering. Publisher: ACM Press. New York, July, 2011.
- [29] Ralph Rowland Young. "The requirements engineering". Handbook Norwood, MA: Artech House, 2004.
- [30] G. Kotonya; I. Sommerville. "Requirements engineering: processes and techniques". Chichester: John Wiley & Sons, 1998.
- [31] Ian Sommerville. "Software engineering". 8th edition, Pearson Education, 2007.
- [32] Ian Sommerville. "Engenharia de software". 9.ed. São Paulo: Pearson Prentice Hall, 2011.
- [33] P. Sawyer; G. Kotonya. "Software requirements". In: Guide to the software engineering body of knowledge, SWEBOK, 2001. Disponível em: http://www.swebok.org/stoneman/trial_1_00.htm. Acesso em: 27 nov. 2021.
- [34] I Sommerville; P. Sawyer. "Requirements engineering: a good practice guide". Chichester: John Wiley, 1997.
- [35] A. M. B. R. Carvalho; T. C. S. Chiossi. "Introdução à engenharia de software". Campinas: Unicamp, 2001.
- [36] R. H. Thayer; M. Doorfman. "Software requirements engineering". 2. ed. IEEE Computer Society Press, 1997. Disponível em: <http://www.sei.cmu.edu/news-atsei/features/1999/mar/Background.mar99.pdf>. Acesso em: 27 nov. 2021.
- [37] L. E. G. Martins. "Uma metodologia de elicitação de requisitos de software baseada na teoria da atividade". Diss. Tese (Doutorado. 2011. Universidade Estadual de Campinas, Faculdade Elétrica e de Computação. Campinas, SP, 2001.
- [38] Roger Pressaman; Bruce Maxim. "Engenharia de software". 8 ed. McGraw Hill Brasil, 2016.
- [39] Antonio Carlos Gil. "Elaboração de projetos de pesquisa". 5. ed. São Paulo: Atlas, 2010.
- [40] Eva Maria Lakatos; Maria de Andrade Marconi. "Técnicas de pesquisa". 7. ed. São Paulo: Atlas, 2012.
- [41] A. Chizzotti. "Pesquisa em ciências humanas e sociais". 2 ed. São Paulo: Cortez, 1995.
- [42] J. B. Granda. "Manual de metodología de la investigación científica". 3. ed. Chimbote: Uladech, 2015.
- [43] Francisco dos Santos Carvalho. "Envolvimento dos moradores no orçamento participativo da cidade de Vitória da Conquista: o caso do bairro Nossa Senhora Aparecida". Cadernos de Ciências Sociais Aplicadas. Vitória da Conquista/BA, n. 19, p. 13-38, 2013.
- [44] Matheus Nascimento Sales. "Estratégia para nortear o processo de engenharia de requisitos aplicada à

metodologia ágil Scrum (Per-Scrum). 2021. Trabalho de Conclusão de Curso (Monografia). Curso de Ciência da Computação. Universidade Estadual do Sudoeste da Bahia (UESB), Vitória da Conquista, 2021. [Orientador.: Prof. Dr. Francisco dos Santos Carvalho].

[45] Maya Daneva et al. “Agile requirements prioritization in large-scale outsourced system projects: An empirical study”. *Journal of systems and software*, v. 86, n. 5, p. 1333-1353, 2013.

[46] I. Vacari. “Um estudo empírico sobre a adoção de métodos ágeis para desenvolvimento de software em organizações públicas”. 2015. Disponível em: <http://tede2.pucrs.br/tede2/handle/tede/6189>. Acesso em: 20 nov. 2021.

[47] A. S. da Costa Macuzo. “Percepções de práticas ágeis em desenvolvimento de software”: benefícios e desafios. Dissertação (Mestrado Profissional em Computação Aplicada). 2017. 194 f. Universidade de Brasília Instituto de Ciências Exatas, Departamento de Ciência da Computação, Brasília, 2017.

[48] PROJECT MANAGEMENT INSTITUTE (PMI). “Agile practice guide”. Pennsylvania - EUA: PMI, 2017b.

[49] PROJECT BUILDER. “Gerenciamento ágil de projetos com Scrum + pmbok. 2014”. Disponível em: <<<https://www.projectbuilder.com.br/Downloads/ebook-gratuito-Scrum-pmbok.pdf>>>. Acesso em: 09 dez. 2021.

[50] AGILEMANIFESTO. “Manifesto for agile software development”. Disponível em: <http://agilemanifesto.org/>. Acesso em: 5 out. 2019.

[51] K. Beck, et al. “The agile manifesto: agile alliance”. 2001. Disponível em: <http://agilemanifesto.org/>. Acesso em: 19 set. 2021.

[52] ken Schwaber; Jeff Sutherland. “Um guia definitivo para o Scrum: as regras do jogo”. 2017. Disponível em: <<<https://www.Scrumguides.org/docs/Scrumguide/v1/Scrum-Guide-Portuguese-BR.pdf>>>. Acesso em: 13 maio. 2021

[53] Andrea Herrmann; Maya Daneva. “Requirements prioritization based on benefit and cost prediction: an agenda for future research”. In: 2008 16th IEEE International Requirements Engineering Conference. IEEE, 2008. p. 125-134.

[54] Jeff Sutherland. “Scrum: a arte de fazer o dobro do trabalho na metade do tempo”. São Paulo: Leya, 2016.

[55] Ken Schwaber. “Agile project management with Scrum”. Microsoft press, Redmond, WA, 2004.

[56] A. P. P. Assis. “Práticas ágeis do PMBOK: uma análise da aplicabilidade da metodologia Scrum”. 2018. Monografia (Graduação em Engenharia de Computação).

Faculdade Independente do Nordeste, Vitória da Conquista, 2018.

[57] Sonja Koppensteiner; Nathalie Udo. “Will agile development change the way we manage software project’s? Agile from a PMBOK guide perspective”. 2003. Disponível em: <<<https://www.pmi.org/learning/library/agile-development-change-manage-software-7717>>>. Acesso em: 3 ago. 2021.

[58] Rafael Serra. “Gestão ágil de projetos de software versus PMBOK.” 2012. Disponível em: <<<http://pgsskroton.com.br/seer/index.php/rcext/article/viewFile/2270/2170>>>. Acesso em: 13 set. 2021.

[59] Kathy Schwalbe. “Managing a project using an agile approach and the PMBOK® guide”. Disponível em: <https://kathyschwalbe.files.wordpress.com/2013/06/managing-a-project-using-an-agile-approach-and-the-pmbok2ae-guide.pdf>. Acesso em: 27 out. 2021.

[60] Alexandre Boeing e Dieimes Nunes de Souza. “Gerenciamento de projetos de software com Scrum e pmbok.” 2013. Disponível em: <<http://www.univale.com.br/unisite/documentos/publicacoes/gerenciamento_de_projetos_de_software_com_Scrum_e_pmbok.pdf>>. Acesso em: 17 jun. 2021.

[61] Mauro Sotille. “Gerenciamento de projetos na engenharia de software”. 2014. Disponível em: <https://www.pmttech.com.br/artigos/Gerenciamento_Projetos_Software.pdf>. Acesso em: 8 jul. 2021.

[61] C. A. F. Machado; R. C. Burnett. “Gerência de processos na engenharia de software em relação às práticas do PMBOK”, Pontifícia Universidade Católica (PUC-PR), Paraná, Curitiba, 1993.

[62] Lucas Soares de Araujo. “Gerenciamento de projetos de software com PMBOK e Scrum: um estudo e análise comparativa do método tradicional e ágil”. 2017. Disponível em: <<https://app.uff.br/riuff/bitstream/1/5668/1/TCC_LUCAS_SOARES_DE_ARAUJO.pdf>>. Acesso em: 26 nov. 2018

[63] PROJECT MANAGEMENT INSTITUTE (PMI). “Guia PMBOK®”: um guia para o conjunto de conhecimentos em gerenciamento de projetos. 6. ed. Pennsylvania - EUA: PMI, 2017a

[64] PROJECT MANAGEMENT INSTITUTE (PMI). “Sobre o PMI”. 2018a. Disponível em: <<<http://brasil.pmi.org/brazil/AboutUS.aspx>>>. Acesso em: 18 out. 2021.

[65] PROJECT MANAGEMENT INSTITUTE (PMI). “O que é PMI?” 2018b. Disponível em: <<<https://brasil.pmi.org/brazil/AboutUS/WhatIsPMI.aspx>>>. Acesso em: 7 abr. 2021.

- [66] PROJECT MANAGEMENT INSTITUTE (PMI). “Capítulos brasileiros do PMI”. 2018c. Disponível em: <<<https://brasil.pmi.org/brazil/PMIChapters/BrasilianChapters.aspx>>>. Acesso em: 8 ago. 2021.
- [67] PROJECT MANAGEMENT INSTITUTE (PMI). “PMBOK Guide 6 edição: o que vem por aí?”. 2017c. Disponível em: <<<https://pmipe.org.br/noticia/149/?pmbok-guide-6a-edicao--o-que-vem-por-ai.html>>>. Acesso em: 04 dez. 2021.
- [68] M. Machado; S. G. Medina. “SCRUM - método ágil: uma mudança cultural na gestão de projetos de desenvolvimento de software. 2009”. Revista Científica Intraciência, Faculdade do Guarujá – UNIEESP, 1(1), 58-71.
- [69] Paulo Cesar de Macedo; José Henrique Teixeira de Carvalho Sbrocco. “Metodologias ágeis: engenharia de software sob medida”. 2012. Disponível em: <<<https://integrada.minhabiblioteca.com.br/#/books/9788536519418/>>>. Acesso em: 29 mar. 2021.
- [70] K.. Schwaber; J. Suterland .“The Scrum guide”. Tradução de Fábio Cruz. Disponível em: <http://fabiocruz.com.br/livros/2017-Scrum-Guide-PtBR-v1.pdf>. Acesso em: 18 set. 2021.
- [71] Fernando Salztrager Benzecry. “Metodologias ágeis para gerenciamento de projetos de inovação e pesquisa e desenvolvimento”. 2017. 51 f. Monografia (MBA em Gestão de Projetos). Fundação Getúlio Vargas, Rio de Janeiro, 2017.
- [72] MANIFESTO ÁGIL. “Princípios por trás do manifesto ágil”. 2001. Disponível em: <<<http://www.manifestoagil.com.br/principios.html>>>. Acesso em: 8 out. 2021.
- [73] SCRUMSTUDY. “Um guia para o conhecimento Scrum: guia sbok”. 2016. Disponível em: <<<https://sitecampus.com.br/wp-content/uploads/2015/05/SCRUMstudy-SBOK-Guide-2016-Portuguese.pdf>>>. Acesso em: 17 dez. 2021.
- [74] P. Abrahamsson. “Measuring the success of software process improvement: the dimensions. “EUROSPI’00”, Copenhagen, Dinamarca, 2000.
- [75] R. Black; J. Mitchel “Advanced software testing”, v. 3, ed. Rockynook, 2008.
- [76] Reinaldo de Oliveira Castro; Fabiano Cutigi Ferrari; Nil Magnus Lasgaard; Luciana Silva Aguenta; Rosely Sanches. “CMMI e SCAMPI: uma visão geral de modelos de qualidade e de um método formal para sua avaliação”. 2015. Disponível em: <<<http://pgsskroton.com.br/seer/index.php/rcext/article/view/2392/2296>>>. Acesso em: 16 abr. 2021.
- [77] Fábio Cruz. “Scrum e PMBOK unidos no gerenciamento de projetos”. Rio de Janeiro: Brasport, 2013.
- [78] Vitor Massaari. “Agile Scrum master no gerenciamento avançado de projetos”. Rio de Janeiro, RJ: Brasport, 2016.
- [79] Jim Highsmith; Alistair Cockburn. “Agile software development: the business of innovation”. Computer, v. 34, n. 9, p. 120-127, 2001.
- [80] D.E.H. Damian. “Challenges in requirements engineering”. Department of Computer Science, University of Calgary, 2000.
- [81] B. Alenjung; A. Person. “Portraying the practice of decision-making in requirements engineering: a case study of large scale bespoke development”. Requirements Engineering Journal, 13, pp. 257-279, 2008.
- [82] M.A.S. Turine; P.C. Masiero. “Especificação de requisitos: uma introdução”. Instituto de Ciências Matemáticas de São Carlos, USP. 1996.
- [83]Nagy Ramadan Darwish; Salwa Megahed. “Requirements engineering in Scrum framework”. International Journal of Computer Applications. v. 149, n. 8, set 2016.
- [84]Alan M. Davis. “Software requirements: objects, functions, and states”. Prentice Hall, Inc., 1993.
- [85] Ricardo Augusto Ribeiro de Mendonça. “Levantamento de requisitos no desenvolvimento ágil de software”. Semana da Ciência e Tecnologia da PUC Goiás, p.12. Goiás, 2014.
- [86] Luis Augusto Moretto. “Mecanismos para validação de requisitos”. Morettic, 2018. Disponível em: <https://morettic.com.br/wp2/mecanismos-validacao-de-requisitos/>. Acesso em: 30 abr. de 2021.
- [87] Mohammad Alkandari; Asma Al-Shammeri. “Enhancing the process of requirements prioritization in agile software development - a proposed model”. Journal of Software, v. 12, n. 6, June 2017.
- [88] J. Karlsson; K. Ryan. “A cost-value approach for prioritizing requirements”. IEEE software, IEEE, v. 14, n. 5, p. 67-74, 1997.
- [89] W. Cunningham. “The wycash portfolio management system”. Proc. OOPSLA, ACM, 1992.