



Universidade Federal de Mato Grosso
Campus Universitário de Várzea Grande
Faculdade de Engenharia



Modelos Conceituais para Implantação da Metodologia Ágil SCRUM em Equipes de Desenvolvimento de Software

Pedro Paulo Santana Costa

Orientadora: Prof^ª. Dr^ª. Joyce Aline de Oliveira Marins

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia de Computação da FAENG/CUVG/UFMT (área de concentração: Engenharia de Computação) como parte dos requisitos para obtenção do título de Bacharel em Engenharia de Computação.

Várzea Grande
2022

Catálogo da publicação na fonte. UFMT / Biblioteca Central

Costa, Pedro Paulo Santana.

Sobre a Preparação de Trabalho de Conclusão de Curso em Engenharia de Computação da FAENG/CUVG/UFMT / Pedro Paulo Santana Costa - Várzea Grande, 2022

23 p.

Orientadora: Joyce Aline de Oliveira Marins

Trabalho de Conclusão de Curso (graduação) - Universidade Federal de Mato Grosso. Campus Universitário de Várzea Grande. Faculdade de Engenharia. Curso de Graduação em Engenharia de Computação.

1. Redação técnica - TCC. 2. \LaTeX - Trabalho de Conclusão de Curso. I. Marins, Joyce Aline de Oliveira. II. Modelos Conceituais para Implantação da Metodologia Ágil SCRUM em Equipes de Desenvolvimento de Software.

MT/UF/BC

CDU 004.932(043.2)

Modelos Conceituais para Implantação da Metodologia Ágil SCRUM em Equipes de Desenvolvimento de Software

Pedro Paulo Santana Costa

Aprovado em 15 de Julho de 2022 pela banca examinadora composta pelos seguintes membros:

Prof^a. Dr^a. Joyce Aline de Oliveira Marins (orientadora)
FAENG/CUVG/UFMT

Prof. Dr. Diogo Henrique Duarte Bezerra FAENG/CUVG/UFMT

Prof^a. Dr^a. Gracyele Santos Souza Guarienti FAENG/CUVG/UFMT

*Aos meus familiares, amigos e a
UFMT.*

Agradecimentos

A todos que participaram, direta ou indiretamente do desenvolvimento deste trabalho de pesquisa.

A minha família pelo apoio durante os anos de faculdade.

A todos os amigos que conheci na UFMT.

A minha orientadora por toda paciência e auxílio no desenvolvimento deste trabalho.

Por fim agradeço a UFMT, essencial no meu processo de formação profissional, assim como por tudo o que aprendi ao longo dos anos do curso.

Resumo

As metodologias ágeis foram desenvolvidas com o principal objetivo de proporcionar ganhos em relação a produtividade, gerando bons resultados e facilitando a realização de tarefas de desenvolvimento de software presentes em empresas/indústrias do setor. Uma das metodologias ágeis existentes denomina-se SCRUM, que tem por princípio gerenciar a equipe a partir de ciclos pré-definidos, tecnicamente denominados por Sprints. Estes ciclos variam a depender do tipo de reunião e entregas a serem realizadas. Este trabalho propõe cinco modelos conceituais para construção de uma metodologia ágil baseada no SCRUM com UML (Linguagem Unificada de Modelagem). O método de Design Science Research foi utilizado na condução da pesquisa, que tem como artefato resultante os modelos conceituais construídos. Para elaboração deste trabalho foram utilizados dados obtidos a partir de uma Revisão Sistemática da Literatura sobre metodologias ágeis baseadas em SCRUM realizada no período de 2018 a 2021, na qual observou-se que uma pequena quantidade de artigos apresenta os passos para construção e implantação destas metodologias. Diante desta lacuna, neste trabalho são apresentados modelos conceituais baseados no SBOK que visam auxiliar o processo de construção e implantação de metodologias ágeis baseadas em SCRUM por equipes de desenvolvimento de software. Os modelos conceituais foram validados por meio de survey com especialistas.

Palavras-chave: Metodologia, Modelos conceituais, SCRUM, UML.

Abstract

Agile methodologies were developed with the main objective of providing gains in terms of productivity, generating good results and facilitating performing software development tasks present in companies/industries in the sector. One of the existing agile methodologies is called SCRUM, whose principle is to manage the team based on pre-defined cycles, technically called Sprints. These cycles vary depending on the type meeting and deliveries to be made. This work proposes five models concepts for building an agile methodology based on SCRUM with UML (Unified Modeling Language). The Design Science Research method was used to conduct the research, which has as a resultant artifact the conceptual models constructed. For the elaboration of this work, data obtained from a Systematic Review of the Literature on agile methodologies based on SCRUM carried out in the period from 2018 to 2021, in which it was observed that a small number of articles present the steps for the construction and implementation of these methodologies. Faced with this gap, this work presents conceptual models based on the SBOK that aim to help the process of construction and implementation of agile methodologies based on SCRUM by software development teams. The conceptual models were validated through a survey with experts.

Keywords: Methodology, Conceptual models, SCRUM, UML.

Sumário

Sumário	i
Lista de Figuras	iii
Lista de Tabelas	v
1 Introdução	1
1.1 Justificativa	4
1.2 Objetivos	5
1.2.1 Objetivos Gerais	5
1.2.2 Objetivos Específicos	5
2 Fundamentação teórica	7
2.1 Engenharia de software	7
2.2 Conceitos do SCRUM	8
2.3 Papéis no SCRUM	11
2.4 Fases do SCRUM	13
2.5 Disciplined Agile Delivery	20
2.6 Os princípios do SCRUM	22
2.7 Os valores do SCRUM	22
2.8 Os aspectos do SCRUM	24
2.9 SBOK	25
3 Metodologia	27

3.1	Design Science Research	27
3.2	Modelo conceitual	30
3.3	Validação dos modelos conceituais	31
3.4	A Revisão Sistemática da Literatura	32
4	Resultados e Discussões	39
4.1	Criação dos modelos conceituais	39
4.1.1	Fase Iniciar	39
4.1.2	Fase Planejar e Estimar	42
4.1.3	Fase Implementar	44
4.1.4	Fase Revisão e Retrospectiva	46
4.1.5	Fase Release	48
4.2	Validação dos modelos conceituais	49
4.2.1	Dados das questões demográficas	49
4.2.2	Dados de validação dos modelos conceituais por fases	50
4.2.3	Dados de validação dos modelos no geral	53
5	Conclusão	57
	Referências Bibliográficas	59
	Apêndice A Informações adicionais	65
A.1	Declaração do Orientador Para a Biblioteca	65
	Apêndice B Survey de validação	67
B.1	Formulário de validação dos modelos	67

Lista de Figuras

2.1	Esqueleto do SCRUM. Fonte: (SCHWABER, 2004)	11
2.2	As Características desejáveis aos papéis essenciais do SCRUM. Fonte: (SATPATHY, 2017)	13
2.3	Estrutura original do SCRUM. Fonte: (SCHWABER; SUTHERLAND, 2013).	14
2.4	Processo de abordagem do ambiente visual. Fonte: (SCHWABER; SUTHERLAND, 2013).	18
2.5	Os 7 Princípios do Disciplined Agile. Fonte: PMI Bahia	21
2.6	Os valores do SCRUM	22
3.1	Ciclo de vida do Design science. Fonte:(SORDI; AZEVEDO; MEI- RELES, 2015)	28
3.2	Processos e suas descrições. Fonte: Autoria Própria.	31
3.3	Número de artigos por fases de busca. Fonte: Autoria Própria.	34
3.4	Dados sobre respostas das questões de pesquisa. Fonte: Autoria Própria.	35
4.1	Fase Iniciar. Fonte: Autoria Própria.	42
4.2	Fase Planejar e Estimar. Fonte: Autoria Própria.	44
4.3	Fase Implementar. Fonte: Autoria Própria	46
4.4	Fase Revisão e Retrospectiva. Fonte: Autoria Própria.	47
4.5	Fase Release. Fonte: Autoria Própria.	49
4.6	Dados da validação, gráfico número 1. Fonte: Autoria Própria.	54

4.7	Dados da validação, gráfico número 2. Fonte: Autoria Própria. . .	54
4.8	Dados da validação, gráfico número 3. Fonte: Autoria Própria. . .	55
4.9	Dados da validação, gráfico número 4. Fonte: Autoria Própria. . .	55

Lista de Tabelas

4.1	Fase Iniciar. Fonte: Adaptado de (SATPATHY, 2017).	40
4.2	Fase Planejar e Estimar. Fonte: Adaptado de (SATPATHY, 2017).	43
4.3	Fase Implementar. Fonte: Adaptado de (SATPATHY, 2017).	44
4.4	Fase Revisão e Restrospectiva. Fonte: Adaptado de (SATPATHY, 2017).	46
4.5	Fase Release. Fonte: Adaptado de (SATPATHY, 2017).	48

Lista de Símbolos e Abreviaturas

UFMT	Universidade Federal de Mato Grosso
FAENG	Faculdade de Engenharia
UML	Linguagem de Modelagem Unificada
SLR	Revisão Sistemática de Literatura
XP	Extreme Programming
OOPSLA	Object-Oriented Programming, Systems, Languages & Applications
WoW	Way to Working
DSDM	Dynamic System Development Model
TI	Tecnologia da informação
TAP	Termo de abertura do projeto
SBOK	SCRUM Body of Knowledge
4SRS	Four Step Rule Set
US	User Stories
UC	Use Cases
QAS	Quality Assurance System
DSR	Design science research

OMT Object Modeling Technique

OOSE Object- Oriented Software Engineering

ISLA Information Systems in Latin America

Capítulo 1

Introdução

Para maximizar o desempenho, as empresas realizam uma variedade de estratégias que estão em constante mudança para obter um produto de alta qualidade. Porém inúmeras vezes o sistema não gera valor para o qual foi criado, ou seja, o produto é desenvolvido, entregue e no final o cliente percebe que de fato o software não segue os critérios que ele solicitou. Observa-se que muitas vezes isso ocorre pela dificuldade de se compreender os reais problemas e criar soluções que efetivamente atendam aos propósitos dos clientes (SILVA, E. C. da; LOVATO, 2016).

As metodologias ágeis cada vez mais são consideradas importantes no contexto atual de desenvolvimento de software devido ao seu foco no benefício do negócio, forte envolvimento das partes interessadas e rápida incorporação de requisitos em mudança (FERRACINI, 2019).

A palavra *ágil* (*Agile*) significa ser rápido, leve, livre e alerta. O desenvolvimento de software ágil é uma maneira de construir software fazendo-o e ajudando outras pessoas a construí-lo de uma só vez. No Desenvolvimento Ágil de Software, as interações e o pessoal são mais importantes do que o processo e as ferramentas. Um software funcional é mais importante do que uma documentação completa, a colaboração com os clientes é mais importante do que a negociação do contrato e ser responsivo às mudanças é mais importante do que

seguir o plano (CARNEIRO; SILVA, A. C. C.; ALENCAR, 2018).

A metodologia SCRUM é uma das metodologias ágeis que se destaca por ter uma forma de trabalhar bem flexível e de fácil adaptação, podendo ser aplicada em diversos ambientes. Além de trabalhar de forma transparente e eficiente, trabalha com constantes alterações no contexto de desenvolvimento (LOPES, 2014).

Além destas vantagens, o SCRUM é uma metodologia de fácil automatização. Isso influencia diretamente nos custos e prazos e colabora de forma efetiva no controle das atividades a serem desenvolvidas (LOPES, 2014).

Definir o escopo do projeto e gerenciá-lo é uma tarefa muito difícil e a área mais importante de se lidar. Sempre há uma mudança no requisito e mudar o requisito também altera o escopo do projeto algumas vezes. Uma vez que o projeto é adquirido, há um acordo assinado entre cliente e empresa de software, então ambos têm que se cumprir e respeitar um ao outro, mudanças são normais em qualquer projeto do início ao fim, estas não podem ser negadas ou recusadas. O SCRUM fornece uma maneira fácil de lidar e gerenciar essas mudanças. A equipe de requisitos primeiro analisa e define o escopo. O arquiteto de soluções define a importância e a urgência desses requisitos que são implementados pela equipe de desenvolvimento conforme instruído pelo líder da equipe (SCHWABER; SUTHERLAND, 2013).

A metodologia ágil acomoda as mudanças de requisitos no futuro, em que pouco tempo é gasto na especificação prévia do sistema e mais ênfase é colocada sobre a rápida entrega de pequenos resultados que apoiam mudanças rápidas nos planos de desenvolvimento, funcionando melhor em meio às incertezas, possuindo tal flexibilidade que pode lidar com custo, escopo e qualidade do software de acordo com as necessidades do cliente. Essas metodologias incentivam a mudança nos requisitos, pois desta forma é possível realmente entregar ao cliente o produto (FÉLIX; TAVARES; CAVALCANTE, 2018).

O termo “metodologias ágeis” tornou-se popular em 2001 quando dezessete especialistas em processos de desenvolvimento de software representando os métodos Extreme Programming (XP), Scrum, DSDM, Crystal e outros, estabeleceram princípios comuns compartilhados por todos esses métodos. O resultado foi a criação da Aliança Ágil e o estabelecimento do “Manifesto Ágil” (Agile Manifesto) (SANTOS SOARES, 2004).

Existem diversas metodologias de projeto e programação de software em uso. Hoje uma das mais comumente utilizadas é o desenvolvimento ágil com SCRUM. O desenvolvimento ágil de software é uma abordagem de desenvolvimento caracterizada pela adaptabilidade, ou seja, pela capacidade de absorver mudanças, sejam elas nas forças de mercado, nos requisitos do sistema, na tecnologia de implementação ou nas equipes de projeto. É por isso que o desenvolvimento ágil é um dos principais paradigmas de desenvolvimento adotados em todo o mundo (MATTIOLI et al., 2009).

O gerenciamento ágil de projetos usando a metodologia SCRUM define-se por ser um processo prescritivo, pois acredita-se que não é possível prever no início de um projeto tudo o que ocorrerá durante o desenvolvimento do mesmo. O SCRUM oferece ferramentas e práticas que permitem maior meta desejada. Usando um processo iterativo de revisão contínua e prazos curtos de *design*, a equipe do projeto aumenta sua capacidade de adaptar rapidamente os projetos a ambientes em rápida evolução nos quais os sistemas serão usados (KAMEI et al., 2011).

Como as metodologias ágeis estão muito preocupadas em produzir produtos de qualidade em tempo hábil que satisfaça as exigências e expectativas tanto do cliente como da equipe de desenvolvimento, elas oferecem os melhores caminhos que podem ser seguidos, e um produto de qualidade é alcançado. Todos os métodos ágeis são bons em questões de qualidade, mas o SCRUM possui o foco no gerenciamento da equipe, preocupada na organização dos processos,

em como as atividades devem ser executadas, deixando a cargo dos participantes do projeto escolher a melhor maneira de concluir com sucesso essas etapas (SAVOINE et al., 2009).

O SCRUM tem sido amplamente utilizado, entretanto, conforme identificado na Revisão Sistemática da Literatura (SLR) sobre metodologias baseadas em SCRUM feita por este autor, poucos trabalhos científicos indicam como o SCRUM pode ser adotado. Este trabalho consiste em apresentar conceitualmente por meio de diagramas de atividades em UML como o SCRUM pode ser implantado. Os modelos conceituais apresentados neste trabalho são inspirado nos conceitos do SBOK e foram validados por meio de um *survey* com especialistas. O SBOK foi escolhido como guia para a construção dos modelos conceituais, pois é um guia de conhecimento em SCRUM. O conteúdo documentado no SBOK foi descrito a partir da experiência de muitos praticantes do SCRUM, sendo apresentados em uma abordagem organizada, servindo de base na fundamentação teórica e para a discussão do presente estudo.

1.1 Justificativa

Durante o ano de 2021 foi dado início ao estudo Metodologias Ágeis Baseadas em SCRUM. A demanda foi solicitada pelo TCE-MT em um convênio firmado entre esta instituição e a UFMT. Para tal foi realizada uma (SLR - Revisão Sistemática da Literatura). O estudo tem conexão com este trabalho de conclusão de curso que possui a finalidade de apresentar o processo de implantação do *framework* SCRUM. Durante a condução da SLR observou-se que poucos trabalhos de fato indicam como o SCRUM foi implantado. Diante desta lacuna observada pelo autor, esta pesquisa apresenta, além de alguns resultados da revisão sistemática, os passos para a implantação do SCRUM em UML (Linguagem de

Modelagem Unificada) por meio de diagrama de atividades, que tem como base o SBOK.

1.2 Objetivos

Nesta seção serão abordados os objetivos que serão abordados neste trabalho.

1.2.1 Objetivos Gerais

O objetivo deste trabalho é apresentar os modelos conceituais para construção e implantação da metodologia ágil SCRUM em UML (Linguagem de Modelagem Unificada).

1.2.2 Objetivos Específicos

A seguir observa-se os objetivos específicos deste trabalho:

- Realizar uma Revisão Sistemática da Literatura para caracterizar e compreender metodologias baseadas em SCRUM;
- Identificar quais são os requisitos necessários para se elaborar os modelos para construção e implantação de uma metodologia baseada em SCRUM;
- Representar através de UML (Linguagem de Modelagem Unificada) os passos de implantação do SCRUM;
- Validar os modelos conceituais por meio de um survey com especialistas.

Capítulo 2

Fundamentação teórica

No presente capítulo serão evidenciados os conceitos referentes aos assuntos tratados no decorrer deste trabalho.

2.1 Engenharia de software

A engenharia de software possui desafios, visto que o software e o ciclo de vida do projeto evoluem rapidamente. O reconhecimento deste fato faz com que empresas venham a utilizar Métodos Ágeis, dado que eles abraçam a mudança e gerenciam os riscos relacionados, sendo que nos gerenciamentos de tempo e risco a melhoria é contínua, pois são permitidos acompanhamentos que trazem melhores previsões de negócios e entregas, evitando problemas ou trazendo soluções rápidas e eficazes (BERGAMASCHI; ZUCHI, 2018).

Os sistemas de software estão se tornando cada vez mais complexos devido aos diversos e mutáveis requisitos dos clientes. Ao mesmo tempo, o mercado é muito competitivo. Portanto, as equipes de software devem desenvolver produtos de software para atender a todos os requisitos. A todo o momento os requisitos estão em constante evolução, ou seja, "projetos mutantes", por isso surge à necessidade de metodologias que se adaptem a esse paradigma, visto que possuem um tempo pré-determinado e a um custo definido. O surgimento de mé-

todos ágeis nos últimos anos tem atraído cada vez mais atenção, sendo utilizados com sucesso para melhorar os processos de projetos de software (SAVOINE et al., 2009).

A agilidade, em sua essência, parte da certeza de que não sabemos tudo (ou sabemos quase nada) no início do projeto de um produto inovador. Como são tantas as incertezas envolvidas, cada projeto deve ser tratado como uma experiência de aprendizagem única e estruturado de forma a acelerar a velocidade de aprendizado (CARVALHO, F. H. T. de, 2014).

O conhecimento em um projeto é gerado através de ciclos sucessivos de aprendizado e adaptação, que incorporam práticas e princípios únicos, que são diferenciais do Gerenciamento Ágil de Projetos. Muitas organizações usam a abordagem de desenvolvimento ágil, dado que a medida que o requisito muda, ele se move para um comportamento volátil. O método ágil lida com isso de maneira eficaz para gerenciar esses requisitos de maneira iterativa (CARVALHO, F. H. T. de, 2014).

2.2 Conceitos do SCRUM

O SCRUM é considerado uma metodologia ágil. Ele é um *framework* estrutural dentro do qual é possível empregar vários processos ou técnicas. O SCRUM aplica uma abordagem incremental e iterativa para melhorar a previsibilidade e o controle de risco (TERLIZZI; BIANCOLINO, 2014).

A origem da palavra traz relatos que apontam que a metodologia foi assim nomeada para fazer referência ao ato de substituição da bola e reinício do jogo britânico *Rugby*. Em um momento do jogo há uma interação entre os jogadores que precisam correr em linha reta fazendo o revezamento de um bastão, analogamente à interação dos usuários dentro desta metodologia em que cada membro

desempenha um papel específico e todos se ajudam em busca de um objetivo comum (SCHWABER; SUTHERLAND, 2013).

Schwaber e Sutherland (2013), documentaram o SCRUM junto com sua equipe, o que foi iniciado pelos autores japoneses Hirotaka Takeuchi e Ikujiro Nonaka. Logo foi realizado o desenvolvimento da metodologia no começo dos anos 90 e a propuseram em 1993. Além disso apresentaram sua criação na Conferência OOPSLA em 1995, tendo grande aceitação. A metodologia foi baseada em um artigo dos autores japoneses Hirotaka Takeuchi e Ikujiro Nonaka publicado em 1986 que discutiu as vantagens de pequenas equipes no desenvolvimento de atividades, principalmente na manufatura resumida da Honda e Toyota. Eles chamaram seu novo método de SCRUM, após o termo de *Rugby* que descreve como as equipes formam um círculo e vão para a bola para colocá-la de volta em jogo novamente. Eles aplicaram esse método pela primeira vez na *Easel Corporation* em 1993.

Existem diversas metodologias de projeto e programação de software em uso. Hoje uma das mais comumente utilizadas é o desenvolvimento ágil com SCRUM. Métodos, práticas e técnicas para o desenvolvimento ágil de projetos prometem aumentar a satisfação do cliente para produzir alta qualidade de software, tal como visa acelerar os prazos de desenvolvimento de projetos, diante de bons resultados, o desenvolvimento ágil é um paradigma de desenvolvimento que vem sendo adotado em todo o mundo. (PEREIRA; TORREÃO; MARÇAL, 2007).

Como as metodologias ágeis estão muito preocupadas em produzir produtos de qualidade, elas oferecem bons caminhos que podem ser seguidos, e um produto de qualidade é alcançado. Todos os métodos ágeis são bons em questões de qualidade, todavia o SCRUM se diferencia, visto que atualmente temos cenários onde cada vez mais os desenvolvedores vêm trabalhando com pequenos incrementos e entregas mais curtas, assim como a busca de qualidade dessas

entregas, gerando assim maior segurança e valor para os clientes (SOUZA et al., 2014).

SCRUM pertence à família de métodos ágeis de desenvolvimento de software que atraíram atenção significativa entre os profissionais de software, observado na pesquisa da VersionOne (2013) indica o predomínio do SCRUM em dois terços das organizações usuárias de métodos ágeis, enquanto o XP é usado mais fortemente em conjunto com o SCRUM e, em apenas alguns casos (2%), isoladamente (VALLERÃO; ROSES, 2013)

Enquanto o método (XP - *Extreme Programming*) que tem sido amplamente aceito como uma das abordagens ágeis mais importantes tem o foco para equipes de pequeno a médio porte que desenvolvem software face a exigências vagas ou em rápida mudança, o SCRUM fornece melhor gestão dos processos em comparação com outros modelos ágeis, com foco em mudanças no processo do ambiente de desenvolvimento do sistema, aplicando variáveis técnicas, obtém-se melhor processo no desenvolvimento e um produto adequado ao usuário (SILVA, M. A. A. M. d., 2017)

SCRUM começa com a premissa de que o desenvolvimento de software é muito complexo e imprevisível para ser planejado com antecedência. Em vez disso, o controle de processo empírico deve ser aplicado para garantir visibilidade, inspeção e adaptação. As diferentes variáveis ambientais e técnicas (como prazo, qualidade, requisitos, recursos, tecnologias, ferramentas de implementação e até métodos de desenvolvimento) devem ser controladas constantemente para poder se adaptar às mudanças com flexibilidade. Isso é alcançado por meio de um processo de desenvolvimento iterativo e incremental (KOM; SURYANI, 2019).

O esqueleto do SCRUM é mostrado na (Figura 2.1). O círculo inferior representa uma iteração das atividades de desenvolvimento que ocorrem uma após a outra. A saída de cada iteração é um incremento do produto. O círculo superior

representa a inspeção diária que ocorre durante a iteração, na qual os membros individuais da equipe se reúnem para inspecionar as atividades uns dos outros e fazer as devidas adaptações. Conduzir a iteração é uma lista de requisitos. Este ciclo se repete até que o projeto seja concluído (SCHWABER, 2004).



Figura 2.1: Esqueleto do SCRUM. Fonte: (SCHWABER, 2004)

2.3 Papéis no SCRUM

O SCRUM implementa o esqueleto iterativo e incremental mostrado no tópico anterior por meio de três papéis: o *Product Owner*, o Time e o *SCRUM Master*. O *product owner* é responsável por representar os interesses, ou seja, as necessidades e os valores do produto de todos os interessados no projeto, que podem ser denominado de *Stakeholders*, em que não são considerados papéis essenciais no SCRUM, porém mesmo assim possui grande relevância em todo o projeto. O *product owner* mantém o *product backlog*, ou seja, uma lista priorizada de requisitos do projeto com prazos estimados para transformá-los em funcionalidade completa do produto (CANO et al., 2021).

A Equipe é responsável pelo desenvolvimento das funcionalidades. As equipes são auto-gerenciadas, auto-organizadas e multifuncionais e são responsáveis por descobrir como transformar o *product backlog* em um incremento de fun-

cionalidade dentro de uma iteração e gerenciar seu próprio trabalho para fazê-lo. Os membros da equipe são coletivamente responsáveis pelo sucesso de cada iteração e do projeto como um todo. É Desejável que tenham conhecimentos em mais de uma área, visto que caso alguém atrase o seu trabalho, um outro membro poderá ajudar a finalizar (KOM; SURYANI, 2019).

O *SCRUM Master* ocupa a posição normalmente ocupada pelo gerente de projeto, mas seu papel é um pouco diferente. Enquanto o gerente de projeto tradicional é responsável por definir e gerenciar o trabalho, o *SCRUM Master* é responsável por gerenciar o processo SCRUM, ou seja, por ensinar SCRUM a todos os envolvidos no projeto, por implementar o SCRUM para que ele se encaixe na cultura de uma organização, e ainda entregue os benefícios esperados, tal como garantir que todos sigam as regras e práticas do SCRUM (DE CARVALHO; MELLO, 2011).

No desenvolvimento ágil, as equipes SCRUM são apoiadas por duas funções específicas. O primeiro é um *SCRUM Master*, que pode ser considerado um *coach* para a equipe, ou seja é semelhante a um líder de equipe ou gerente de projeto de uma pequena equipe, ajudando os membros da equipe a usar o processo SCRUM para ter um desempenho de alto nível (PATANAKUL; RUFO-MCCARRON, 2018).

O *SCRUM Master* é o catalisador do grupo SCRUM. Eles garantem que o grupo SCRUM seja eficaz e progressivo. Em caso de algum impedimento, isto é, qualquer entrave ou obstáculo que reduza a produtividade do Time SCRUM, diante disso o *SCRUM Master* acompanha e tenta resolver os problemas para o grupo, tal como é responsável por evitar que seja adicionado novas estórias durante uma *Sprint*, para que de fato não mude o planejamento que já foi definido previamente. O *product owner* é o representante do negócio e responsável por otimizando o valor do trabalho, deve ser uma pessoa que pode se responsabilizar

por garantir que a equipe produza valor a partir do projeto para o negócio, cliente ou quem deseja o projeto (o comprador final). O *product owner* normalmente escreve os requisitos centrados no cliente na forma de estórias, prioriza-os e fornece-os para o *backlog do produto* (BAIJENS; HELMS; IREN, 2020).

As características desejáveis aos papéis essenciais no SCRUM podem ser observadas na (Figura 2.2).

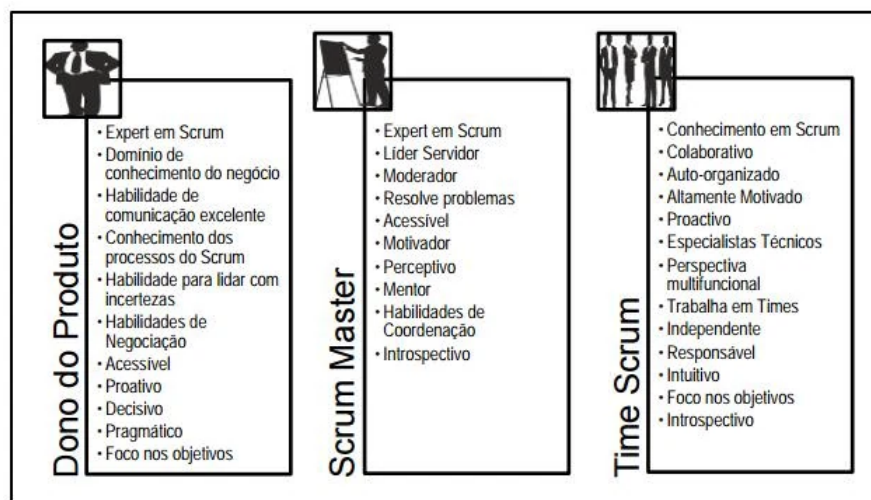


Figura 2.2: As Características desejáveis aos papéis essenciais do SCRUM. Fonte: (SATPATHY, 2017)

2.4 Fases do SCRUM

No SCRUM todo o trabalho é feito em *Sprints*. Cada *Sprint* é uma iteração que pode variar de 7 a 30 dias consecutivos (Figura 2.3). Cada *Sprint* é iniciada com uma reunião de planejamento da *Sprint*, onde o *product owner* e a Equipe se reúnem para colaborar sobre o que será feito para a próxima *Sprint*. Selecionando o *product backlog* de maior prioridade, o *product owner* informa ao Time o que é desejado, e o Time informa ao *product owner* quanto do que é desejado ele acredita que pode se transformar em funcionalidade na próxima *Sprint*. Depois de decidir o que deve ser feito na próxima *Sprint*, o Time desenvolve

o *Sprint backlog*, ou seja, uma lista de tarefas (*SCRUM Board*) que devem ser executadas para entregar um incremento completo da funcionalidade do produto potencialmente entregável até o final da *Sprint* (CANO et al., 2021).

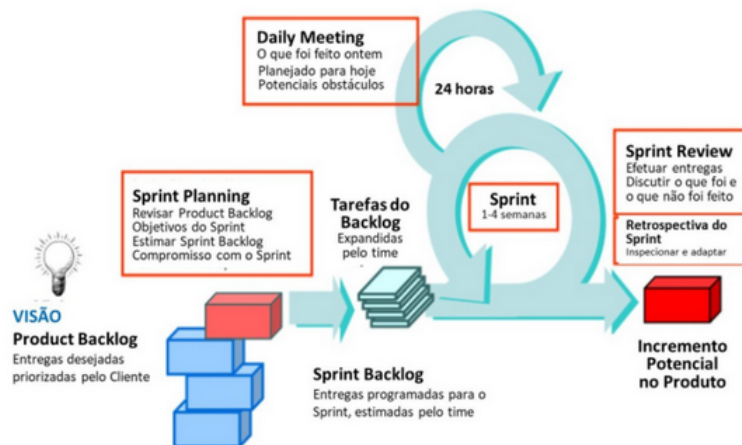


Figura 2.3: Estrutura original do SCRUM. Fonte: (SCHWABER; SUTHERLAND, 2013).

A metodologia SCRUM defende uma reunião de planejamento no início da *Sprint*, onde os membros da equipe de desenvolvimento trabalha para prever qual funcionalidade será desenvolvida ao longo da *Sprint*. A primeira parte desta reunião serve para definir o que será feito ao longo da *Sprint* e a segunda para definir como realizar o trabalho (POSSA, 2013).

De acordo com Schwaber e Sutherland (2013), é uma reunião que tem uma caixa de tempo, também denominado de *time boxing*, dado que para um *Sprint* de 4 semanas, o tempo da reunião será de 8 horas, assim como para *Sprints* menores, será contado 2 horas de reunião para cada semana de desenvolvimento na *Sprint*.

Hoje em dia, tornou-se extremamente importante para uma equipe SCRUM atingir seu propósito e trabalhar para realizá-lo de forma eficaz. O gráfico *burn-down* demonstra o quanto falta para atingir a meta. É sempre aconselhável que a equipe tenha um propósito claro de existência, metas que são traçadas em um

nível razoável que descerá para um nível de *Sprint*. O que vale a pena notar aqui é que os objetivos do grupo não são diferentes dos objetivos individuais, então todo o propósito é trabalhar em direção a um objetivo com total foco e comprometimento (SCHWABER; SUTHERLAND, 2013).

Com o modelo baseado em *Sprint*, a equipe SCRUM normalmente recebe *feedback* das partes interessadas, aonde o *Product Owner*, dado o *feedback* dos *stakeholders*, verifica a melhoria no resultado do trabalho e fornece informações necessárias para atualizar o *Product Backlog* com novas estórias de usuário. Dessa forma, os *stakeholders* ficam mais felizes porque conseguem exatamente o que desejam depois de estarem envolvidos em cada etapa do processo Cano et al. (2021).

As tarefas na lista surgem à medida que a *Sprint* evolui. Essas tarefas devem ser divididas, assim como é sugerido que cada uma não ultrapasse a duração de 16h, caso tenha a necessidade, deve-se quebra-la individualmente, além disso é necessário que todas as tarefas estejam dentro do limite de 16h (PEREIRA; TORREÃO; MARÇAL, 2007).

Todos os dias a equipe se reúne para uma reunião de 15 minutos chamada *Daily SCRUM*. Na *Daily SCRUM*, o intuito é que a equipe de desenvolvimento possa sincronizar o seu trabalho e estabelecer um plano para as próximas 24 horas (POSSA, 2013).

Segundo Schwaber e Sutherland (2013) Ela é uma oportunidade para a equipe inspecionar e adaptar o seu trabalho ao longo da *Sprint*, assim como atualizar o gráfico *burn down*, dado que caso uma tarefa seja concluída, o desenvolvimento real será atualizado, até que de fato chegue ao desenvolvimento planejamento . cada membro da equipe responde a três perguntas:

1. O que você fez neste projeto desde a última *Daily SCRUM Meeting*?

2. O que você vai fazer antes da próxima reunião?

3. Você tem algum obstáculo?

O objetivo da reunião é sincronizar o trabalho de todos os membros da equipe e agendar as reuniões que a equipe precisa para encaminhar seu progresso (POSSA, 2013)

Conforme Schwaber e Sutherland (2013), ao final da *Sprint*, é realizada uma reunião de revisão do *Sprint*, na qual o Time apresenta o que foi desenvolvido durante a *Sprint* ao *product owner* e quaisquer outros *stakeholders*(acionistas), que queiram participar. Após a revisão da *Sprint* e antes da próxima reunião de planejamento da *Sprint*, o *SCRUM Master* também realiza uma outra reunião, denominada de retrospectiva da *Sprint*. Esta reunião tem o intuito de incentivar o Time a revisar, dentro da estrutura do processo SCRUM, seu processo de desenvolvimento para torná-lo mais eficaz e agradável para a próxima *Sprint*.

Os processos do SCRUM envolvem o *SCRUM Master*, a equipe SCRUM e proprietário do produto (*Product Owner*). Dentro de um projeto ou produto, este é dividido em vários pequenos requisitos chamados épicos, visto que estão no *Backlog* do produto em uma visão macro. Esses processos constitui-se também de *Sprints*, e tarefas para a *Sprint*, em que essas tarefas fazem parte da *Sprint backlog*. (KOM; SURYANI, 2019).

O *backlog* do produto é uma lista de desejos de todas as histórias de usuário (requisitos) que se espera que sejam concluídas no projeto. A história mais importante deve estar no topo da lista, de forma que todo o *backlog* seja classificado continuamente em ordem com base na importância da história (BAIJENS; HELMS; IREN, 2020).

O tamanho de um *backlog* deve ser definido de forma que o time consiga concluí-lo em um ciclo chamado de *sprint*, aonde contém todos os requisitos para

a *Sprint* atual. Os requisitos podem mudar durante o de acordo com as necessidades do usuário, porém deve-se evitar inserir novas funcionalidades em uma *Sprint* em andamento. O *backlog* do produto é considerado como a maior parte do requisito e é avaliado pelo proprietário do produto. O *backlog* do produto é dividido em *Sprints* e seguido pela estrutura de planejamento do ciclo de desenvolvimento, que inclui diferentes métodos para concluir uma *Sprint* no prazo (SCHWABER; SUTHERLAND, 2013).

A estrutura do SCRUM é um princípio chave, em que o reconhecimento de que desafios fundamentalmente empíricos não podem ser resolvidos com sucesso utilizando uma abordagem tradicional de "controle". O SCRUM adota uma abordagem empírica acompanhando o progresso e replanejando em reuniões regulares para desenvolver produtos de forma incremental, na qual pode responder de forma ágil aos desafios emergentes (SILVA, R. T., 2011).

O SCRUM é um processo iterativo para o desenvolvimento de processos inovadores de produtos ou serviços. É uma abordagem ágil usada para criar uma lista de desejos prioritários da característica do produto com os recursos necessários para construir um produto com sucesso (NAGARIA; SADATH; AHMED, 2019).

Durante o processo iterativo, o produto é revisado pelo proprietário do produto. O *feedback* de o proprietário do produto ajuda a decidir qual trabalho fazer a seguir e como realizá-los, visto que os itens de maior prioridade do *backlog* são concluídos antes da prioridade mais baixa. A iteração pode ser de uma semana a um calendário mês de duração. Durante o processo iterativo, o produto é revisado pelo proprietário do produto. (NAGARIA; SADATH; AHMED, 2019).

Para Schwaber e Sutherland (2013), as *Sprints* no SCRUM são muito importantes para fornecer as especificações para a realização do trabalho pela equipe de desenvolvimento durante um período de tempo fixo. Ao final da *Sprint*, esse

trabalho deve estar pronto para ser entregue ao cliente atendendo a cada item do *product backlog* (critério de pronto). Finalmente, a *Sprint* termina com uma revisão e retrospectiva da *Sprint* - ou melhor, lições aprendidas. Este ciclo é repetido ao longo do ciclo de vida do projeto até que todo o escopo tenha sido entregue.

Conforme Schwaber e Sutherland (2013), existe um ciclo de vida do ambiente visual, dado que ele é dividido em duas fases principais, que são o *Sprint Planning*, utilizando a técnica de modelagem UML e a *Sprint Execution*. As diretrizes a seguir são fornecidas na metodologia SCRUM: primeiro deve ser criado o *product backlog*, que é uma lista de *User Stories* (pequenas descrições de todas as funcionalidades exigidas no produto de software), o *product owner* é responsável por priorizar o *product backlog*.

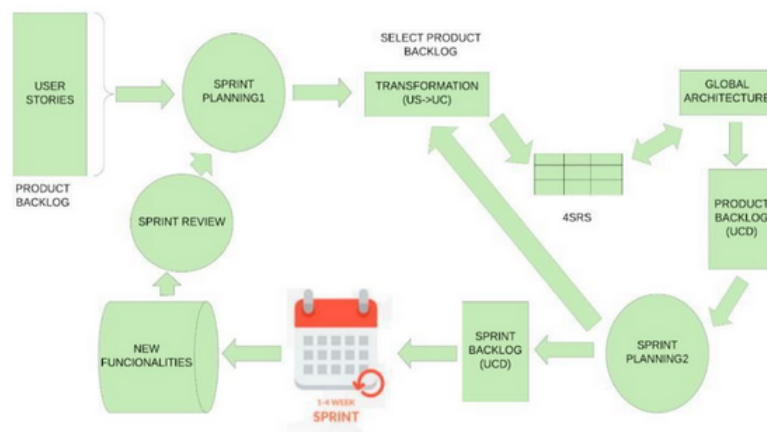


Figura 2.4: Processo de abordagem do ambiente visual. Fonte: (SCHWABER; SUTHERLAND, 2013).

A primeira fase consiste em *product backlog* (lista de *User Stories*), *Sprint Planning*, *Selectproduct backlog (Transformation - US-> UC)*, *4SRS*, *Arquitetura Global*, *product backlog* (com Diagramas de Caso de Uso) e a segunda fase consiste em *Sprint backlog*, *Sprint* e *Revisão de Sprint*(SCHWABER; SUTHERLAND, 2013).

Na (Figura 2.4) estão ilustradas as atividades e artefatos dessas duas eta-

pas. Os principais artefatos da fase de análise de requisitos ágeis são o *product backlog* e os modelos de Caso de Uso resultantes do processo de transformação de *User Stories*. A segunda fase é implementada por meio de uma série de iterações chamadas *Sprints*. Cada *Sprint* entrega um incremento de produto potencialmente utilizável. A abordagem proposta é inspirada em SCRUM, UML e métodos orientados a objetos, por isso é focada no usuário e na arquitetura, bem como orientada iterativa e incremental. O desenvolvimento orientado ao usuário é uma das práticas comumente adotadas no desenvolvimento de software, que fornece uma excelente solução para requisitos imprecisos, incompletos ou inconsistentes (SCHWABER; SUTHERLAND, 2013).

Segundo Schwaber e Sutherland (2013), de muitas maneiras, isso reflete aspectos do gerenciamento de projetos tradicionais. Uma das principais diferenças, no entanto, é como se cria partes “entregáveis” do projeto ao longo do caminho, em vez de entregar tudo no final. Isso permite que o cliente perceba o valor do projeto em todo o processo, em vez de esperar até que o projeto seja fechado para ver os resultados.

De acordo com (SATPATHY, 2017) existem 5 fases no SCRUM, em que dentro destas fases, encontram-se 19 processos envolvidos. As 5 fases são denominadas de:

1. Fase Iniciar
2. Fase Planejar e Estimar
3. Fase Implementar
4. Fase Revisão e Retrospectiva
5. Fase *Release*

2.5 Disciplined Agile Delivery

O *Disciplined Agile Delivery* é uma metodologia ágil que funciona como um kit de ferramentas / estratégias, que auxiliam na tomada de decisões da empresa, levando em consideração o tamanho da equipe e outras especificidades do contexto da organização. Este *framework* organiza, simplifica e agiliza os processos de uma organização de forma personalizada, respeitando o WoW - *Way to Working* de cada empresa. É um método híbrido, ou seja, inclui uma variedade de outras práticas e métodos, como: SCRUM, Kanban, *Extreme Programming*, *Agile Data*, *Agile Modeling* e DSDM (AMBLER; LINES, 2012).

Com o *Disciplined Agile* (Figura 2.5), consegue-se identificar quais ferramentas são necessárias para cada situação específica, e é essa flexibilidade do método que mantém a empresa competitiva e adaptável a qualquer adversidade. Cada função a ser desempenhada recebe funções e responsabilidades específicas para manter o fluxo da organização em movimento (AMBLER; LINES, 2012).

A estrutura de decisão de processo *Disciplined Agile* fornece orientação leve para ajudar as organizações a otimizar seus processos de tecnologia da informação (TI) de uma maneira sensível ao contexto. Ele faz isso mostrando como várias atividades, como entrega de soluções, operações, arquitetura corporativa, gerenciamento de portfólio e muitas outras, funcionam juntas de forma coesa. A estrutura também descreve o que essas atividades devem abordar, fornece uma gama de opções para fazer isso e descreve as compensações associadas a cada opção. Cada pessoa, cada equipe e cada organização são únicas, portanto, as estruturas de processo devem fornecer opções, não prescrever respostas (AMBLER; LINES, 2012).



Figura 2.5: Os 7 Princípios do Disciplined Agile. Fonte: PMI Bahia

Nota-se como sendo um benefício da metodologia *Agile* e SCRUM é sua flexibilidade, dado que torna os métodos ágeis atraentes em situações incertas e mutáveis, por ser flexível as mudanças (CARNEIRO; SILVA, A. C. C.; ALENCAR, 2018).

A estrutura do processo *Disciplined Agile* estende o ciclo de vida de construção do SCRUM para abordar o ciclo de vida de entrega completo enquanto adota estratégias de vários métodos ágeis e enxutos. Muitas das práticas sugeridas pelo *Disciplined Agile* são comumente discutidas na comunidade ágil, como a integração contínua, reuniões diárias de coordenação e refatoração, e algumas são práticas “avançadas” comumente aplicadas, mas, por algum motivo, não são comumente discutidas. Essas práticas avançadas incluem previsão de requisitos iniciais, previsão de arquitetura inicial e teste de ciclo de vida (AMBLER; LINES, 2012).

O foco do SCRUM é a liderança do projeto e alguns aspectos do gerenciamento de requisitos. O *Disciplined Agile* adota e adapta muitas ideias do SCRUM, como trabalhar a partir de uma pilha de itens de trabalho em ordem de prioridade, ter um *product owner* responsável por representar as partes interessadas e produzir uma solução potencialmente consumível a partir de cada iteração (AMBLER;

LINES, 2012).

2.6 Os princípios do SCRUM

Os princípios do SCRUM são uma base ou a base subjacente para a estrutura ágil do SCRUM. Os princípios do SCRUM, quando incorporados aos valores centrais do *Agile*, ajudam a melhorar a transparência e a adaptação no gerenciamento de projetos. Os princípios e valores do SCRUM são as diretrizes que são seguidas para estruturar e fortalecer o *framework* SCRUM (RUBIN, 2018).

2.7 Os valores do SCRUM

A equipe do SCRUM trabalha para um objetivo comum, ou seja, criar valor para o cliente. Mais importante ainda, cada membro da equipe desempenha um papel diferente na criação desse valor (Figura 2.6).



Figura 2.6: Os valores do SCRUM
Fonte: (AMBLER; LINES, 2016)

O primeiro valor do SCRUM é o comprometimento, que pode parecer simples no papel. Basta entrar no trabalho todos os dias com uma atitude positiva, cum-

prir os prazos e seguir as instruções, certo? Nesse contexto, o compromisso é permanecer dedicado aos objetivos desenvolvidos como equipe. Contribuir para o sucesso da equipe e enfrentar os desafios diários é crucial, mas é igualmente importante falar se o projeto estiver fora do caminho. SCRUM ensina a otimizar seus esforços. Portanto, uma parte crucial do compromisso é refletir sobre o objetivo comum e garantir que se priorize tarefas que realmente agreguem valor. Os *SCRUM Masters* podem ajudar a promover o comprometimento por meio de uma boa comunicação e planejamento de *Sprint* adequado (SCHWABER; SUTHERLAND, 2013).

O foco é permanecer no caminho certo e ajudar outros membros da equipe a fazer o mesmo. Uma parte fundamental da metodologia SCRUM é a *Sprint* – um período de trabalho definido e com prazo determinado para concluir uma série de tarefas. Para se concentrar nessas tarefas, a equipe deve eliminar ao máximo as distrações e realizar várias tarefas ao mesmo tempo. As distrações podem causar contratempos que prolongam o tempo necessário para concluir as tarefas. O resultado final é o excesso de horas extras, o que anula o propósito de usar uma metodologia ágil (SCHWABER; SUTHERLAND, 2013).

Reuniões em pé são uma ótima oportunidade para explicar a meta do *Sprint* e planejar o que será concluído dentro desse prazo. Para ajudar os membros da equipe a se concentrarem, os *SCRUM Masters* devem falar abertamente sobre cargas de trabalho individuais para garantir que eles atribuam apenas um número alcançável de tarefas. Assuma muitos objetivos e todos se sentirão sobrecarregados (SCHWABER; SUTHERLAND, 2013).

Abertura/franqueza significa ter a mente aberta em termos de comunicação entre membros de diferentes disciplinas. Promova uma cultura que acolhe novas ideias e estilos de trabalho, que podem ajudar a equipe a avançar (SCHWABER; SUTHERLAND, 2013).

Respeito requer tratar as pessoas como iguais, independente de idade, educação, posição social, etc. Quem são seus colegas de trabalho fora do time SCRUM não importa; tudo o que importa é como eles trabalham juntos nos objetivos da equipe. Trata-se também de respeitar e entender os clientes e as partes interessadas, para que você esteja mais bem equipado para atender às necessidades deles. Respeito significa que os colegas de equipe devem apreciar uns aos outros por seus pontos fortes em termos de *hard* e *soft skills*. Significa também respeitar as decisões e opiniões dos outros, mesmo que você discorde delas (SCHWABER; SUTHERLAND, 2013).

Mostrar coragem, saindo além de sua zona de conforto para alcançar o sucesso. Ao permanecer comprometido com a meta e se concentrar menos em si mesmo, você pode resolver problemas desafiadores e produzir resultados inesperados. Deve-se estar disposto a enfrentar o desconhecido. Se você encontrar coisas que não entende ou identificar um problema, não tenha medo de fazer perguntas difíceis. A capacidade de falar honestamente e questionar o status pode ser a chave para impulsionar a melhoria durante uma *Sprint* em específico (SCHWABER; SUTHERLAND, 2013).

Uma combinação desses valores contribui significativamente para o desempenho geral de uma equipe. Adicione motivação, entusiasmo e motivação, e você terá a fórmula para manter o moral alto o tempo todo.

2.8 Os aspectos do SCRUM

Segundo Satpathy (2017) existem cinco aspectos do SCRUM que devem ser abordados ao longo de qualquer projeto: organização, justificativa do negócio, qualidade, mudança e risco. Organização: Compreender os papéis e responsabilidades definidos em um projeto SCRUM é muito importante para garantir a

implementação bem-sucedida do SCRUM.

1. **Organização** - Este aspecto se concentra em várias facetas de uma organização de projeto SCRUM, bem como papéis principais e não essenciais e como formar equipes SCRUM de alto desempenho.
2. **Justificativa de Negócios** - Este aspecto se concentra no conceito e propósito de Justificativa de Negócios no que se refere aos projetos SCRUM.
3. **Qualidade** - Este aspecto se concentra em definir a qualidade no que se refere aos projetos e apresentar a abordagem SCRUM para atingir os níveis de qualidade exigidos.
4. **Mudança** - Este aspecto se concentra na importância da mudança em qualquer projeto, independentemente de seu método ou estrutura e expande como os processos de desenvolvimento SCRUM são projetados para abraçar a mudança.
5. **Risco** - Este aspecto foca no gerenciamento de riscos em um ambiente SCRUM, considerando várias ferramentas que facilitam o gerenciamento de riscos.

2.9 SBOK

SBOK é um Guia para o Conhecimento em SCRUM (Guia SBOK®) fornece diretrizes para a implementação bem-sucedida do SCRUM - a estrutura de entrega de projetos e desenvolvimento de produtos *Agile*. O SBOK é um guia abrangente para entregar projetos usando SCRUM e fornece diretrizes para a implementação bem-sucedida do SCRUM (SATPATHY, 2017).

O Guia SBOK ® foi desenvolvido para organizações e profissionais que desejam implementar o SCRUM ou que desejam fazer melhorias em seus processos SCRUM. Destina-se a ser usado como referência e guia de conhecimento por SCRUM experiente e outros profissionais de desenvolvimento de produtos ou serviços, bem como por pessoas sem experiência ou conhecimento prévio de SCRUM ou qualquer outra metodologia de entrega de projeto (SATPATHY, 2017).

No passado, a comunidade SCRUM dependia fortemente de tentativa e erro e/ou coaches ou consultores individuais para ajudar a implementar o SCRUM. Com uma infinidade de experiências de muitos praticantes de SCRUM em todo o mundo em diferentes indústrias, essas práticas recomendadas agora foram documentadas em uma abordagem organizada no SBOK ® Guia para beneficiar qualquer praticante interessado. O objetivo é tirar a implementação do SCRUM das mãos de consultores pagos e compartilhar o conhecimento com a comunidade (SATPATHY, 2017).

Capítulo 3

Metodologia

3.1 Design Science Research

Design science research é uma abordagem de pesquisa qualitativa em que o objeto de estudo é o processo de design. O processo de design gera simultaneamente conhecimento sobre o método usado para projetar um artefato e o design ou o próprio artefato (LACERDA et al., 2013).

O principal objetivo da pesquisa em *Design science research* (DSR) é de estudar, pesquisar e investigar o artificial e seu comportamento, tanto do ponto de vista acadêmico quanto da organização para que possa alcançar o conhecimento e a compreensão de um domínio de problema por meio da construção e aplicação de um artefato projetado (LACERDA et al., 2013).

A pesquisa em *Design science research* é motivada pelo desejo de melhorar o meio ambiente pela introdução de artefatos novos e inovadores. Nesse sentido, este método constitui em um processo rigoroso para projetar artefatos para resolver problemas, avaliar o que foi projetado ou o que está funcionando, e comunicar os resultados obtidos (LACERDA et al., 2013).

O ciclo que constitui o método utilizado neste trabalho pode ser observado na (Figura 3.1).

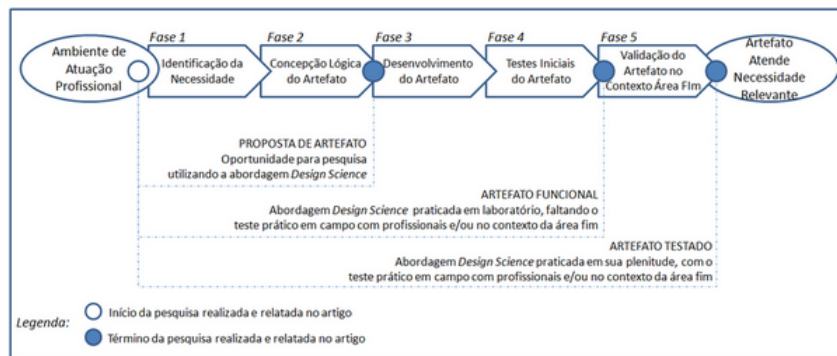


Figura 3.1: Ciclo de vida do Design science. Fonte:(SORDI; AZEVEDO; MEIRELES, 2015)

- **Fase 1 - Identificação da necessidade:** A primeira fase deu início com a realização da Revisão Sistemática da Literatura, em que após a coleta e análise de dados constatou-se uma lacuna em uma das questões de pesquisa, que refere-se a falta de informações sobre o processo de implantação do SCRUM.
- **Fase 2 - Concepção lógica do artefato:** Nesta fase foi dado início ao planejamento do artefato, na qual definiu-se a utilização de UML e o diagrama de atividades para concepção dos modelos conceituais, tal como as fontes bibliográficas.
- **Fase 3 - Desenvolvimento do artefato:** Após o planejamento para elaboração, o artefato que são os modelos conceituais começaram a ser diagramados elencando-se os 19 processos presentes nas 5 fases do SBOK para implantação do *Framework* SCRUM.
- **Fase 4 - Testes iniciais do artefato:** Esta está relacionada a realização de testes pilotos do artefato, podendo ser feito por um ou mais especialistas, todavia para este trabalho o teste piloto não foi realizado devido ao curto tempo para a entrega e defesa do TCC.

- **Fase 5 - Validação do artefato no contexto área fim:** Apesar da fase Fase 4 - Testes iniciais do artefato não ter sido realizada, após a conclusão da elaboração do artefato em questão, foi disponibilizado uma *survey* para validação, em que pessoas ou equipes que já vivenciaram o SCRUM validaram os modelos conceituais construídos.

Neste trabalho é apresentado um total de 5 modelos conceituais referentes a implantação do *framework* SCRUM, baseados no SBOk que é um Guia para o Conhecimento em Scrum. Dentro das fases Iniciar, planejar e estimar, implementar, revisão e retrospectiva e release que compõe os modelos conceituais existem um total de 19 processos. Nestes processos há entradas, ferramentas e saídas obrigatórias e opcionais. Neste trabalho serão apresentados somente as entradas, ferramentas e saídas obrigatórias para um projeto SCRUM, dado que as opcionais ficam a critério do tipo de projeto e questões organizacionais.

Os modelos conceituais são apresentados em Linguagem de Modelagem Unificada (UML) do tipo diagrama de atividades. Para cada modelo foi elaborada uma tabela com as entradas, ferramentas e saídas obrigatórias para cada fase.

O ciclo de relevância neste tipo de método tem o início com a pesquisa em *Design Science* com um contexto de aplicação que não apenas fornece os requisitos para a pesquisa (por exemplo, a oportunidade/problema a ser abordado) como entradas, mas também define critérios de aceitação para a avaliação final dos resultados da pesquisa. O artefato de design melhora o ambiente. Os resultados do teste de campo determinarão se são necessárias iterações adicionais do ciclo de relevância neste projeto de pesquisa em *Design Science*. O novo artefato pode ter deficiências na funcionalidade ou em suas qualidades inerentes (por exemplo, desempenho, usabilidade) que podem limitar sua utilidade na prática. Uma boa pesquisa em ciência do *design* geralmente começa identificando e representando oportunidades e problemas em um ambiente de aplicação real.

O artefato resultante do *Design Science* na pesquisa são os modelos conceituais para implantação do SCRUM (PADUA, 2020).

3.2 Modelo conceitual

Os modelos contendo as instruções para a implantação do SCRUM possuem características conceituais pois apresentam os principais elementos dos modelos, tal como realiza associações entre os elementos em questão, de forma que o usuário consiga visualizar e entender os passos elencados.

A característica básica de um modelo conceitual, como o próprio termo explicita, é que ele é uma abstração da realidade. Um modelo conceitual fornece uma base formal (notacional e semântica) para ferramentas e técnicas usadas para suportar a modelagem de dados. Modelagem de dados é o processo de abstração onde somente os elementos essenciais da realidade observada são enfatizados, descartando-se os elementos não essenciais (FILHO; IOCHPE, 1999).

A visão um pouco menos tradicional da modelagem de dados começa com a modelagem conceitual. A característica básica de um modelo de dados é uma coleção de conceitos que podem ser usados para descrever um conjunto de dados e as operações para manipular esses dados. Modelos conceituais compreendem um formalismo e uma linguagem de descrição, podendo ser uma linguagem léxica e/ou gráfica. Esta linguagem faz a utilização de um sistema padrão de símbolos que formam uma linguagem formal, embora descomplicada, que comunica uma abundância de conhecimento sobre as informações que estão sendo modeladas.

Os modelos conceituais que representam os passos para a implantação do SCRUM neste trabalho foram constituídos por meio de diagramas de atividades, para isso foi utilizado o software ArgoUML que é um ferramenta para elaboração de diversos tipos de diagramas.

Diagramas de atividades são ferramentas visuais poderosas que dividem grandes processos e cadeias de processos em atividades pequenas. Um diagrama de atividades é uma representação visual de um processo ou sistema. Este tipo de diagrama mostra as ações, subprocessos e decisões que constituem uma atividade maior. Um diagrama como este é frequentemente usado para planejar, implementar e otimizar processos em TI e gerenciamento (SPECIFICATION, s.d.).

Na (Figura 3.2) nota-se as simbologias e as suas definições que foram utilizada no diagrama de atividades para elaboração dos modelos conceituais.


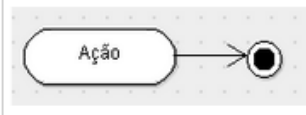
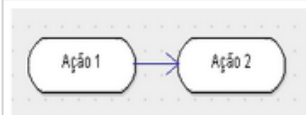

	Símbolo de Início - Utilizado para indicar o início de uma modelagem.
	Símbolo de término - Utilizado para indicar o término da modelagem.
	Símbolo de atividade - Utilizado para indicar um estado de ação ou atividade a ser realizada.
	Símbolo de conector - Utilizado para indicar o fluxo, ou seja, a direção da atividade ou ação.

Figura 3.2: Processos e suas descrições. Fonte: Autoria Própria.

3.3 Validação dos modelos conceituais

Com a conclusão dos modelos conceituais em Linguagem de Modelagem Unificada (UML) do tipo diagrama de atividades, os modelos utilizados foram apresentados para especialistas que já utilizaram o SCRUM por meio de um *survey*. O método de validação consiste em elencar os diagramas com algumas perguntas criadas no google forms para que a equipe responsável verifique o trabalho

realizado.

Para validação dos modelos foram convidados em torno de 65 pessoas. No final obteve-se um total de 23 respostas. O formulário foi enviado para especialistas que trabalham ou já trabalharam em projetos que utilizam o SCRUM. Essas pessoas foram contatadas por meio de mensagens no LinkedIn, tal como por e-mail e mensagens no Whatsapp.

Essas pessoas foram informadas que o formulário em questão trata-se de uma avaliação dos modelos conceituais construídos pelo autor que abordam a implantação do SCRUM e que tem como base o SBOK. Os participantes também foram informadas que o anonimato seria mantido por questões éticas.

3.4 A Revisão Sistemática da Literatura

Foi realizado um estudo de artigos publicados entre os anos de 2018 a 2021 a respeito da metodologia SCRUM, em que teve como base de busca o IEEE e o ScienceDirect. Diante disto foram definidos os critérios de inclusão e exclusão, tal como as questões de pesquisa.

Para a realização das buscas foram realizados testes com chaves de buscas específicas, assim como a utilização da busca avançada e operadores booleanos existentes base do IEEE e ScienceDirect.

Foram utilizadas um total de 8 chaves de buscas que estiveram presentes no estudo, na qual foi utilizado a ferramenta de busca avançada, e em seguida foram agregadas com o conector booleano OR. As chaves de buscas utilizadas na pesquisa estão presentes a seguir: ("SCRUM based methodologies"OR Agile methodology SCRUM OR Results obtained using the agile SCRUM methodology OR Increasing the use of agile SCRUM methodology OR Initiatives using the SCRUM agile methodology OR Projects implementing SCRUM methodology OR SCRUM

Applications OR The advantages of applying the SCRUM methodology).

As questões de pesquisas que compuseram o estudo podem ser observadas a seguir:

1. O que é SCRUM?
2. Quais são os objetivos da metodologia proposta para melhorar o desenvolvimento ágil baseado em SCRUM?
3. Quais foram os passos utilizados para construir e implantar a metodologia proposta?
4. A metodologia proposta foi validada de que forma?

Todas as questões de pesquisa foram respondidas em um artigo submetido no ISLA 2022 (Information Systems in Latin America). Mas o foco desta pesquisa é baseada nas respostas à penúltima questão.

Com a concatenação das duas bases de busca, obteve-se um total de 618 artigos retornados. Após as devidas análises, foram selecionados 74 artigos baseando-se em seus títulos. Em seguida foram analisados os critérios de inclusão e exclusão, em que concluiu-se a pesquisa com um total de 46 artigos incluídos, ou seja, foram excluídos 28 artigos dos selecionados com base nos títulos.

Todos os dados de artigos retornados referente às etapas da pesquisa podem ser observados na (Figura 3.3).

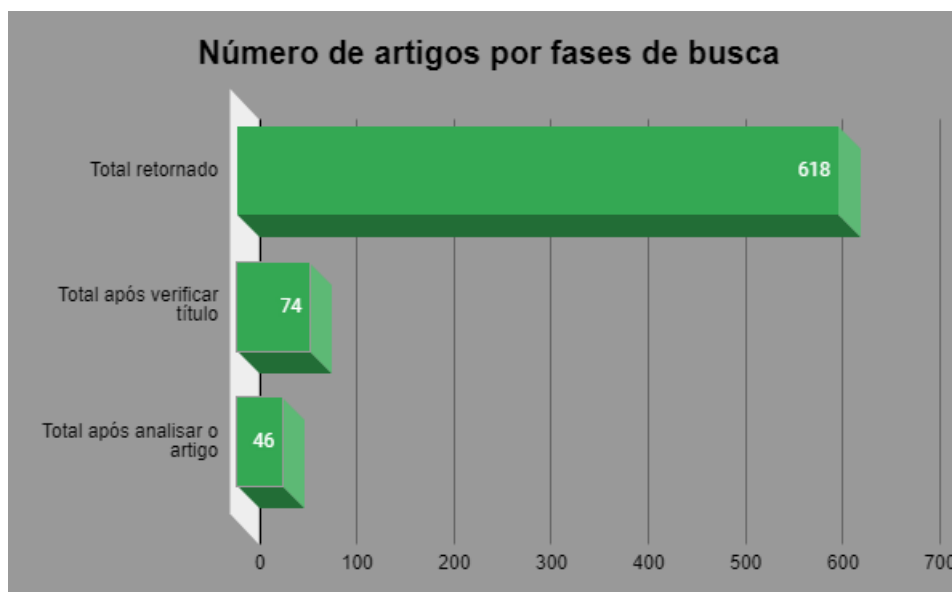


Figura 3.3: Número de artigos por fases de busca. Fonte: Autoria Própria.

Após a conclusão das pesquisas, obteve-se os dados referente ao ano com mais artigos publicados ao estudo proposto, número por locais de publicação e as questões de pesquisa.

Os dados referente as questões de pesquisa da (SLR), estão apresentado na (Figura 3.4).

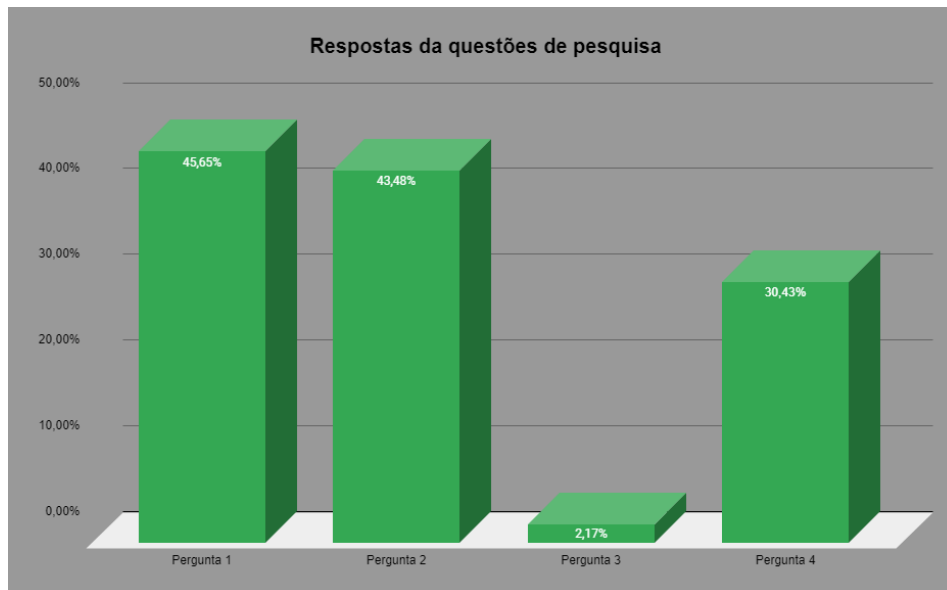


Figura 3.4: Dados sobre respostas das questões de pesquisa. Fonte: Autoria Própria.

Diante do resultado obtido nas questões de pesquisas, obteve-se que apenas um (2.17%) artigo apresentou os aspectos referente aos passos utilizados para construir e implantar a metodologia proposta. O artigo faz menção aos oito estágios de implementação do *framework* SCRUM.

Com o baixo número de artigos contendo o processo de implantação do *framework* SCRUM na pesquisa realizada dos anos de 2018 a 2021, neste trabalho será apresentado um guia conceitual elaborado em UML (Linguagem de Modelagem Unificada), do tipo diagrama de atividades, elencando as 5 fases para se implantar o *framework* SCRUM, na qual dentro dessas fases existem um total de 19 processos.

A elaboração dos modelos conceituais tem como referência principal o GUIA SBOK, que trata-se de um Guia para o Conhecimento em SCRUM (Guia SBOK™) fornece diretrizes para a implementação bem-sucedida do SCRUM, a abordagem mais popular de desenvolvimento de produtos Agile e entrega de projetos (SAT-PATHY, 2017).

Para a construção do guia foi utilizado UML (Linguagem de Modelagem Unificada). Esta linguagem é expressa através de diagramas. Cada diagrama é composto por elementos (formas gráficas usadas para os desenhos) que possuem relação entre si. O UML (*Unified Modeling Language*) surgiu da união de três metodologias de modelagem: o método de Booch, o método OMT (*Object Modeling Technique*) de Jacobson e o método OOSE (*Object-Oriented Software Engineering*) de Rumbaughé um método de modelagem visual como meio de projetar e/ou criar software orientado a objetos. Como a UML é uma linguagem visual para linguagem de modelagem orientada a objetos, todos os elementos e diagramas são baseados no paradigma orientado a objetos (SANTOS; ALVES, 2015).

A U.M.L. não é uma linguagem de programação, mas uma linguagem de modelagem, cujo objetivo é auxiliar os engenheiros de software a definir as características do software, tais como seus requisitos, seu comportamento, sua estrutura lógica, a dinâmica de seus processos e até mesmo suas necessidades físicas em relação ao equipamento sobre o qual o sistema deverá ser implantado (GUEDES, 2007).

Com a utilização do diagramas em UML, eles fornecem inúmeras visões de como modelar, o que facilita para o entendimento e cobertura total dos processos por meio de uma modelagem que contemple todas as áreas, de forma que os diagramas trabalhem em conjunto. Usando UML podemos criar modelos para todos os tipos de aplicativos de software, onde esses aplicativos podem ser executados em qualquer hardware, sistema operacional e rede, e são escritos em qualquer linguagem de programação (SANTOS; ALVES, 2015).

Os diagramas de atividades descrevem quais atividades são realizadas por um sistema do ponto de vista de observações externas. o problema é o que fazer, não como fazer (SPECIFICATION, s.d.).

Os 19 processos presentes para se implantar o SCRUM possuem Entradas, Ferramentas e Saídas obrigatórias e opcionais. As que são consideradas obrigatórias são classificadas com asterisco (*) e devem estar presentes para que de fato siga as exigências do processo específico. Em relação às opcionais estarão diretamente ligadas a questões organizacionais e as variações do projeto a ser elaborado, na qual será definido se entrarão ou não para o processo em questão.

Capítulo 4

Resultados e Discussões

4.1 Criação dos modelos conceituais

Para a criação dos modelos, como mencionado previamente, foi utilizado o software ArgoUML, dado que neste programa é possível realizar diversos tipos de modelagens em diagramas. Para elaboração dos modelos conceituais foram analisados os 19 processos presentes nas 5 fases para se implantar o SCRUM conforme o Satpathy (2017), na qual possuem Entradas, Ferramentas e Saídas obrigatórias e opcionais, em que os processos opcionais ficam a critério do tipo do projeto a ser elaborado, como também das diretrizes organizacionais de cada empresa. Para melhor visualização foi feita uma tabela adaptada, em que elenca somente as Ferramentas e Saídas obrigatórias sinalizadas com (*). Também serão apresentados os modelos elaborados pelo autor e um texto explicando os processos envolvidos em cada fase.

4.1.1 Fase Iniciar

As entradas, ferramentas e saídas obrigatórias desta fase são descritas na (Tabela 4.1).

FASE INICIAR	
1. Criar a Visão do Projeto	
ENTRADAS	1. Caso de negócio do projeto*
FERRAMENTAS	1. Reunião de visão do projeto*
SAÍDAS	1. O product Owner é identificado*, 2. Declaração da visão do projeto*
2. Identificar o SCRUM Master e o(s) Stakeholder(s)	
ENTRADAS	1. Product Owner*, 2. Declaração da visão do projeto*
FERRAMENTAS	1. Critérios de seleção*
SAÍDAS	1. SCRUM Master identificado*, 2. Stakeholders identificados*
3. Formar o Time SCRUM	
ENTRADAS	1. Product Owner*, 2. SCRUM Master*, 3. Declaração da visão do projeto*
FERRAMENTAS	1. Seleção do time SCRUM*
SAÍDAS	1. Time SCRUM identificado*
4. Desenvolver os Épicos	
ENTRADAS	1. Time central SCRUM*, 2. Declaração da visão do projeto*
FERRAMENTAS	1. Reuniões dos grupos de usuários*
SAÍDAS	1. Épicos*, 2. Personas*
5. Criar o Backlog Priorizado do Produto	
ENTRADAS	1. Time central SCRUM*, 2. Épicos*, 3. Personas*
FERRAMENTAS	1. Métodos de priorização da história de usuário*
SAÍDAS	1. Backlog do produto priorizado*, 2. Critério de pronto*
6. Conduzir o Planejamento da Release	
ENTRADAS	1. Time central SCRUM*, 2. Stakeholders*, 3. Declaração da visão do projeto*, 4. Backlog priorizado do produto*, 5. Critério de pronto*
FERRAMENTAS	1. Sessões de planejamento da release*, 2. Métodos de priorização da release*
SAÍDAS	1. Cronograma de planejamento da release*, 2. Duração da sprint*

Tabela 4.1: Fase Iniciar. Fonte: Adaptado de (SATPATHY, 2017).

Na (Figura 4.1) da fase Iniciar observa-se que o primeiro passo ocorre quando o usuário possui determinado requisito. Diante disso ele irá solicitar a elaboração de um produto. Para que ocorra esta fase é necessário que se tenha o caso de negócio que é um tipo de documento com motivo para se começar o projeto, ou seja o que se espera sobre o projeto em questão e que geralmente fica nas mãos da diretoria e da presidência. Para sua elaboração nota-se que pode ser um documento estruturado, ou apenas uma conversa mais informal, atrelado a isto é feito o Termo de abertura do projeto (TAP), onde é um complemento para o plano de negócio, este é um tipo de documento que fica para quem irá desenvolver o projeto (SATPATHY, 2017).

A partir do Plano de negócio o *product Owner* realizará a reunião de visão do projeto com todos que possuem ligação com o projeto, em que começará a definir a visão do projeto, ou seja, quais são os valores presentes, vale ressaltar que é necessário que seja flexível para mudanças caso seja necessário (SATPATHY, 2017).

O passo seguinte consiste na identificação dos papéis considerados essenci-

ais e não essenciais no SCRUM, no qual o *SCRUM Master* é um papel essencial e possui critérios para seleção do mesmo. Em relação aos *Stakeholders* são denominados papéis não essenciais, porém mesmo não sendo essencial possui sua relevância no SCRUM. Em relação ao Time SCRUM também é considerado um papel essencial e possui critérios para a seleção, na qual estão presentes no SBOK (SATPATHY, 2017).

A partir dos passos anteriores, seguidamente será realizada a reunião com o grupo de usuários, tendo em vista a elaboração dos épicos, que por sua vez são funcionalidades em nível macro (SATPATHY, 2017).

Como descrito anteriormente são estabelecidos os épicos, e em vista disso define-se priorização de histórias de usuário, visto que o usuário possui requisitos mais importantes que tem a necessidade de estar pronto primeiro, tal como é estabelecido os critérios de pronto que tem o intuito de definir quando determinada história de usuário está pronta, evitando assim o retrabalho, tal como a redução de custos, pois será possível ter a certeza de que determinada história de fato está pronta (SATPATHY, 2017).

Quando os épicos estiverem prontos será realizado sessões de planejamento da *release*, que são reuniões para definir quais são as *releases* mais importantes, tal como a duração das *sprints* que compõem cada *release*, ou seja, as junções de cada *Sprint* formaram uma *release* que é um pedaço do produto em funcionamento (SATPATHY, 2017).

O primeiro modelo em questão, que é o da fase iniciar pode ser observado logo a seguir na (Figura 4.1).

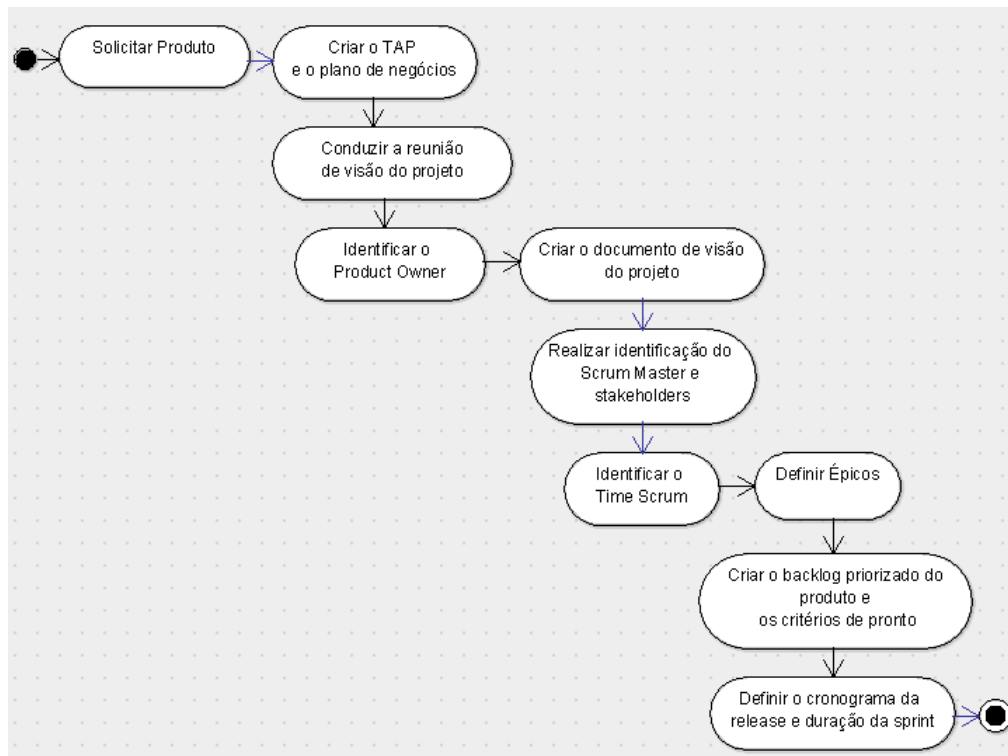


Figura 4.1: Fase Iniciar. Fonte: Autoria Própria.

4.1.2 Fase Planejar e Estimar

As entradas, ferramentas e saídas obrigatórias desta fase são descritas na (Tabela 4.2).

FASE PLANEJAR E ESTIMAR	
7. Criar as Estórias de Usuário	
ENTRADAS	1.Time central SCRUM*, 2.Backlog priorizado do produto*, 3.Critério de pronto*, 4.Personas*
FERRAMENTAS	1.Expertise de escrever a estória de usuário*
SAÍDAS	1.Estórias de usuários*, 2.Critérios de aceitação da Estórias de usuários*
8. Aprovar, Estimar, e Comprometer as Estórias de Usuário	
ENTRADAS	1.Time Central de SCRUM*, 2.Estórias de Usuário*, 3.Critérios de Aceitação da Estória de usuário*
FERRAMENTAS	1.Reuniões do Grupo de Usuários*
SAÍDAS	1.Estórias de Usuário Aprovadas, Estimadas e Comprometidas*
9. Criar as Tarefas	
ENTRADAS	1.Time central do SCRUM*, "2.Estórias de Usuário Aprovadas, Estimadas e Comprometidas"
FERRAMENTAS	1.Reuniões de Planejamento de Tarefas*
SAÍDAS	1.Lista de Tarefas*
10. Estimar as Tarefas	
ENTRADAS	1.Time central SCRUM*, 2.Lista de Tarefas*
FERRAMENTAS	1.Reuniões de Estimativa de Tarefas*, 2.Critério de Seleção*
SAÍDAS	1.Lista de Tarefas de Esforço Estimado*
11. Criar o Backlog do Sprint	
ENTRADAS	1.Time Central do SCRUM*, 2.Lista de Tarefas de Esforço Estimado*, 3.Duração do Sprint*
FERRAMENTAS	1.Reuniões de Planejamento do Sprint*
SAÍDAS	1.Backlog do Sprint*, 2.Gráfico Burndown do Sprint*

Tabela 4.2: Fase Planejar e Estimar. Fonte: Adaptado de (SATPATHY, 2017).

Na (Figura 4.2) da fase Planejar e Estimar nota-se que o primeiro processo consiste em criar estórias de usuário, tendo em vista os épicos, pois estas estórias possuem um maior nível de detalhamento dos épicos. As estórias de usuários possuem critérios de aceitação que por sua vez são limitadores, ou seja, o que é necessário para que o *product owner* aceite determinada estória de usuário no final da *sprint* (SATPATHY, 2017).

Para a elaboração das estórias de usuário é necessário realizar reuniões com os grupos de usuários, em que serão aprovadas pelo *product Owner*, estimadas utilizando determinadas técnicas, tal como comprometer quais estórias de usuário o time acredita que conseguiram concluir na próxima *sprint* (SATPATHY, 2017).

A partir das estórias de usuários, será realizado a reuniões de planejamento e estimativa de tarefas, para que seja feito a identificação e estimativa das tarefas, porém deve-se observar que as tarefas não são obrigatórias no SCRUM, visto que são definidas por serem mais um nível de especificação a partir das estórias de usuário, ficando a critério de cada equipe decidir se irão utilizá-las ou não

(SATPATHY, 2017).

O último processo ocorre com a reunião de planejamento da *sprint*, na qual será criado o *backlog da sprint*, em que consiste em identificar quais estórias de usuário entraram na próxima *sprint*, tal como obter o gráfico *burndown* da *sprint*, visto que nele terá uma ampla visão do desenvolvimento planejado e real referente ao projeto (SATPATHY, 2017).

A seguir nota-se a (Figura 4.2) com o modelo da fase planejar e estimar.

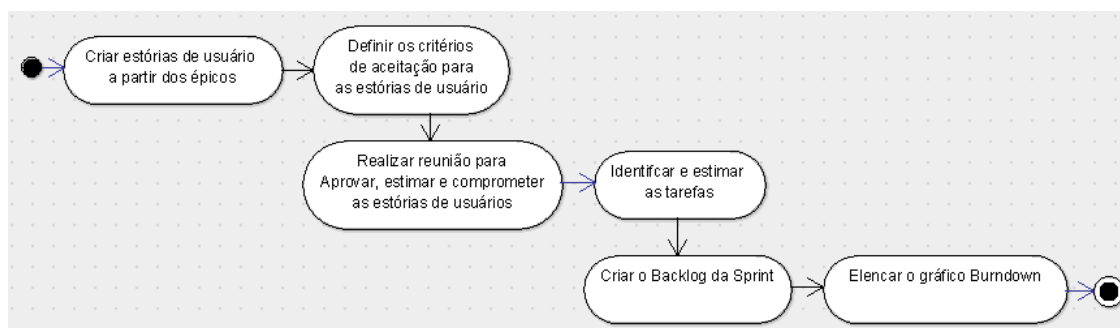


Figura 4.2: Fase Planejar e Estimar. Fonte: Autoria Própria.

4.1.3 Fase Implementar

As entradas, ferramentas e saídas obrigatórias desta fase são descritas na (Tabela 4.3).

FASE IMPLEMENTAR	
12. Criar os Entregáveis	
ENTRADAS	1.Time Central do SCRUM*, 2.Backlog do Sprint*, 3.SCRUMboard*, 4.Registro de Impedimento*
FERRAMENTAS	1.Expertise do Time*
SAÍDAS	1.Entregável do Sprint*, 2.SCRUMboard Atualizado*, 3.Registro de Impedimento Atualizado*
13. Conduzir a Reunião Diária	
ENTRADAS	1.Time SCRUM*, 2.SCRUM Master*, 3.Gráfico Burndown Sprint*, 4.Registro de Impedimento*
FERRAMENTAS	1.Reunião Diária*, 2.Três Perguntas Diárias*
SAÍDAS	1.Gráfico Burndown do Sprint Atualizado*, 2.Registro de Impedimento Atualizado*
14. Refinamento do Backlog Priorizado do Produto	
ENTRADAS	1.Time Central do SCRUM*, 2.Backlog Priorizado do Produto*
FERRAMENTAS	1.Reuniões de Revisão do Backlog Priorizado do Produto*
SAÍDAS	1.Backlog Priorizado do Produto Atualizado*

Tabela 4.3: Fase Implementar. Fonte: Adaptado de (SATPATHY, 2017).

Na (Figura 4.3) da fase implementar apresentada a seguir nota-se que para começar o passo de desenvolvimento, isso irá acontecer com todo o Time SCRUM, pois neste processo as histórias de usuário serão elaboradas, observa-se também que o critério para começar o desenvolvimento irá ficar na responsabilidade do Time, em que apesar de possuírem critérios de priorização, a ordem durante o desenvolvimento não será afetado. Com o término das histórias desenvolvidas o *Product Owner* fará a validação das mesmas, levando em conta que caso alguma não seja concluída ficará a critério do *Product Owner* definir se entrará na próxima *Sprint*, ou se a determinada história de usuário perdeu o valor (SATPATHY, 2017).

No processo de desenvolvimento será criado e atualizado o *SCRUMboard* que será possível visualizar o andamento das histórias de usuário na *Sprint* em questão, onde os seguintes itens estarão presentes no *SCRUMboard*: A fazer, Fazendo e Feito. Também será atualizado o registro de impedimentos, em que caso exista alguma pendência para realização de determinada história de usuário o *SCRUM Master* tentará sanar o impedimento existente (SATPATHY, 2017).

Os impedimentos podem surgir no processo de desenvolvimento e para isso é realizada a reunião diária, com o intuito de que tenha conhecimento a respeito do andamento do projeto, bem como o controle dos impedimentos e atualizar o gráfico *burndown*. Esta reunião é um *time boxing* que deverá respeitar o tempo de no máximo 15 minutos, deve ocorrer todos os dias, no mesmo horário, realizado em pé. Nesta reunião será atualizado diariamente como está o andamento do projeto de modo geral, pois todos devem falar e caso necessário responder a 3 perguntas diárias, tendo como exceção do *Product Owner* e *Stakeholders* que podem participar, porém não irão falar durante a reunião (SATPATHY, 2017).

Com o conclusão de uma *Sprint* um novo ciclo irá iniciar, diante disso será realizado a reunião de revisão do *Backlog* priorizado do produto, onde ficará na responsabilidade do *Product Owner* sempre estar em contato com o usuário para

entender quais são as necessidades, diante disso o *Backlog* do produto estará sempre em movimento, na qual será definido as histórias de usuário mais importantes que estarão presentes na próxima *sprint* (SATPATHY, 2017).

A seguir observa-se a (Figura 4.3) que referencia a fase Implementar.

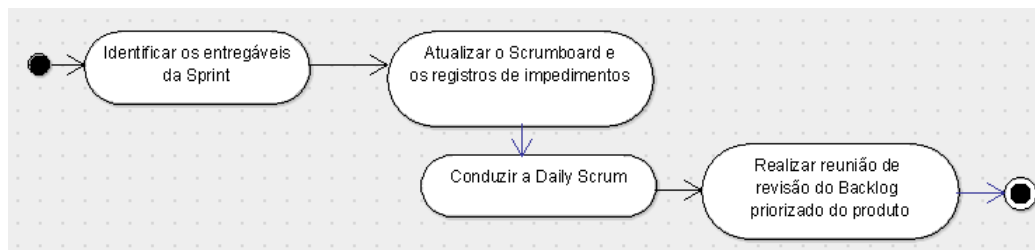


Figura 4.3: Fase Implementar. Fonte: Autoria Própria

4.1.4 Fase Revisão e Retrospectiva

As entradas, ferramentas e saídas obrigatórias desta fase são descritas na (Tabela 4.4).

FASE REVISÃO E RETROSPECTIVA	
15. Convocar o SCRUM de SCRUMs	
ENTRADAS	1. SCRUM Master ou Representantes do Time SCRUM*
FERRAMENTAS	1. Reunião do SCRUM de SCRUMs*, 2. Quatro Perguntas por Time*
SAIDAS	1. Coordenação Melhor do Time*
16. Demonstrar e Validar o Sprint	
ENTRADAS	1.Time Central do SCRUM*, 2.Entregáveis do Sprint*, 3.Backlog do Sprint*, 4.Critério de Pronto*, 5.Critérios de Aceitação da Estória de Usuário*
FERRAMENTAS	1.Reuniões de Revisão do Sprint*
SAIDAS	1.Entregáveis Aceitos*
17. Retrospectiva do Sprint	
ENTRADAS	1.SCRUM Master*, 2.Time SCRUM*
FERRAMENTAS	1.Reunião de Retrospectiva do sprint*
SAIDAS	1.Pontos de Melhoria Acordados*

Tabela 4.4: Fase Revisão e Restrospectiva. Fonte: Adaptado de (SATPATHY, 2017).

Diante da fase Revisão e Retrospectiva apresentada na (Figura 4.4), o primeiro processo realizado é a reunião SCRUM de SCRUMs (SoS), em que são reuniões mais curtas e não possuem um *time boxing*, como também é realizado 4 perguntas por time. Este tipo de reunião acontece em intervalos pré-definidos, ou quando solicitado pelo time, o intuito é de compartilhar o andamento do projeto por cada time, visando identificar possíveis pendências e riscos que podem

impactar o time, com isto irá possibilitar um *feedback* e a melhor coordenação do time (SATPATHY, 2017).

Seguidamente será realizada a reunião de revisão da *sprint*, no qual ocorre para verificar os entregáveis aceitos pelo *product owner* a partir dos critérios de aceitação definidos previamente. Caso determinada estória de usuário seja aceita pelo *product owner* ela será encerrada, mas se for reprovada, irá retornar para o *Backlog* do produto (SATPATHY, 2017).

Com o término da *Sprint* será executada a reunião de retrospectiva da *Sprint* que é um evento *time boxing*, em que é formada pelo Time SCRUM e o SCRUM Master, porém o *product owner* e *Stakeholders* podem participar, todavia não são obrigatórios. Nesta reunião será possível identificar os pontos de melhorias acordados na última *Sprint*, ou seja, o que deu certo e errado durante a *Sprint* (SATPATHY, 2017).

Na (Figura 4.4) apresentada a seguir nota-se o modelo da fase Revisão e Retrospectiva.

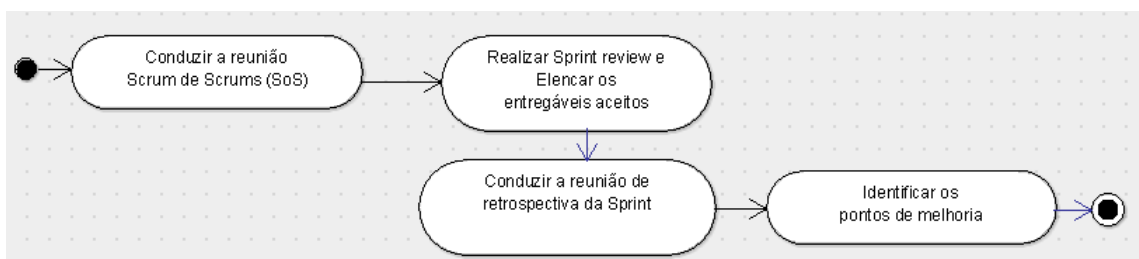


Figura 4.4: Fase Revisão e Retrospectiva. Fonte: Autoria Própria.

Vinte (87%) dos participantes responderam que o modelo e tabela abordam todos os conceitos necessários para a fase revisão e retrospectiva, tal como vinte (87%) e vinte e dois (95.7%) responderam que o modelo é de fácil adoção e facilmente compreensível. Nesta etapa evidenciou-se que seria significativo expressar a razão da reunião (SoS), visto que o intuito é conectar determinadas equipes para melhor gerenciamento de um projeto complexo.

Um outro ponto a ser considerado seria deixar mais claro o objetivo da reunião de revisão da *Sprint*, e reunião de retrospectiva, dado que na reunião de revisão é realizado uma apresentação com as partes interessadas do projeto, na qual obterá um *feedback*, de outra maneira a reunião de retrospectiva identifica os pontos de melhoria. Os critérios de pronto podem ser incluídos para melhor observação, dado que é um ponto importante e que pode reduzir muito o esforço desnecessário em atividades que já foram realizadas, tal como redução de custos.

4.1.5 Fase Release

As entradas, ferramentas e saídas obrigatórias desta fase são descritas na (Tabela 4.5).

FASE RELEASE	
18. Enviar os Entregáveis	
ENTRADAS	1.Dono do Produto*, 2.Stakeholder(s)*, 3.Entregáveis Aceitos*, 4.Cronograma de Planejamento da release*
FERRAMENTAS	1.Métodos de Implantação Organizacional*
SAÍDAS	1.Contrato de Prestação de Trabalho*
19. Retrospectiva do Projeto	
ENTRADAS	1.Time(s) Central(s) do SCRUM*
FERRAMENTAS	1.Reunião de Retrospectiva do Projeto*
SAÍDAS	1.Pontos de Melhoria Acordados*

Tabela 4.5: Fase Release. Fonte: Adaptado de (SATPATHY, 2017).

Ao observar a (Figura 4.5) da fase *Release*, nota-se que o primeiro passo consiste em identificar os métodos de implantação organizacional, ou seja, como a empresa irá colocar em uso o projeto realizado, diante disso a release será entregue e então a empresa em questão gerará o contrato de prestação de serviço. O contrato de prestação de serviço é um documento formal de aceite, isto é, o recebimento de determinada parte do projeto funcionando, com isso também será realizado o pagamento para a equipe envolvida (SATPATHY, 2017).

No final do projeto é realizada a reunião de retrospectiva do projeto, na qual não é um evento *time boxing* e envolve o Time SCRUM, ou seja quem elaborou

o projeto, porém o *Product Owner* também poderá participar. Será tratado tudo que deu certo ou errado no decorrer do projeto, tal como deixar elencado todas as melhorias acordadas para um futuro projeto (SATPATHY, 2017).

Na (Figura 4.5) tem-se o último modelo, denominado de fase *release*.

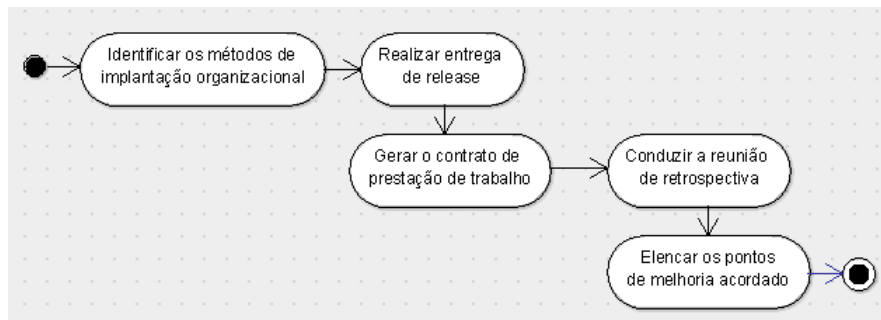


Figura 4.5: Fase Release. Fonte: Autoria Própria.

4.2 Validação dos modelos conceituais

No presente capítulo será apresentado os dados referente ao formulário elaborado pelo autor, na qual para manter a privacidade de cada participante foi atribuído o código P acompanhado de numeração sequencial para cada respondente.

4.2.1 Dados das questões demográficas

Como mencionado no capítulo de metodologia, para a validação dos modelos conceituais elaborado pelo autor, foi realizado um formulário na plataforma google forms, em que foram respondidos por especialistas que trabalham ou já trabalharam com o SCRUM, a seguir nota-se as respostas obtidas das questões demográficas presentes no formulário.

Diante do nível de formação obteve-se que vinte e um (91.3%) dos participantes possuem graduação como nível de formação, assim como a maior parte,

ou seja, sete (30.4%) trabalham em serviço público, seguido por seis (26.1%) no ramo empresarial. Em relação à função exercida nota-se um número maior de analistas, contando com sete (26.09%), sendo eles analistas de TI, analista de sistemas, analista de projetos e analista de negócios.

Em relação ao tempo de experiência em TI dos participantes, retornou-se nove (39.1%) para 0 - 2 anos de experiência, tal como nove (39.1%) para 6 anos ou mais. Acerca de tempo em projetos baseados em SCRUM, dezesseis (69.6%) dos participantes responderam que participam de 1 - 2 anos em projeto SCRUM. No que se refere à quantidade de projetos baseados em SCRUM que os participantes já atuaram, nota-se que doze (52.2%) atuaram em 1 - 2 projetos, tal qual oito (34.8%) responderam que já participaram em mais de 4 projetos baseados em SCRUM.

4.2.2 Dados de validação dos modelos conceituais por fases

Dados de validação da Fase Iniciar

No que se refere a validação da fase iniciar, dezenove (82.6%) dos participantes responderam que o modelo e a tabela abordam os conceitos necessário para a fase iniciar, assim como vinte e dois (95.7%) e vinte (87%) informaram que o modelo é de fácil adoção e facilmente compreensível.

Ao analisar o modelo da fase iniciar, tendo como referências as sugestões elencadas pelos participantes do formulário, conforme os participantes [P13] , [P19] e [P22] "para obter melhor compreensão poderia realizar uma reestruturação do modelo e tabela, pois o SCRUM busca ser um *framework* mais leve e simples, com menos documentação". A reestruturação do modelo e tabela, para a fase iniciar possivelmente foi indicada, visto que é maior fase, diante disso apresentam muitos processos juntos, podendo assim apresentar muitas informações

em uma única etapa ao leitor o que dificulta o entendimento.

Para o participante [P4], seria significativo "Definir que o *Backlog* do produto estará sempre em movimento e poderá acontecer mudanças, pois visa a prioridade do usuário, dado que dependendo de como analisar o modelo, pode-se entender que isso não ocorre". É importante salientar que o *Backlog* do produto sempre está em constante priorização, pois o *Product Owner* estará sempre em contato com o usuário para verificar as suas necessidades.

Dados de validação da Fase Planejar e Estimar

Ao analisar os dados de validação da fase planejar e estimar, dezesseis (69.6%) dos participantes responderam que o modelo e a tabela abordam os conceitos necessários para a fase planejar e estimar, assim como vinte e dois (95.7%) e vinte e um (91.3%) informaram que o modelo é de fácil adoção e facilmente compreensível.

Levando em consideração os dados das respostas obtidas da fase planejar e estimar, para o respondente [P16], "seria interessante adoção do *planning poker* ao modelo". O *planning poker* é um processo voltado a identificação por meio de notas a dificuldade para realização de cada estória de usuário pelo time, porém este é processo que passou a ser opcional, em que não é mais considerado obrigatório, ou seja, ficará a critério de cada projeto SCRUM adoção deste processo.

Um ponto que foi elencado pelo participante [P18] foi em relação ao gráfico *Burndown*, na qual informa que "no primeiro momento, ou seja, durante a primeira *Sprint* ele será criado, porém não terá informações para atualiza-lo". Diante da primeira *Sprint* será evidenciado somente um planejamento estimado no decorrer da *Sprint* em questão, pois somente quando o time começar o desenvolvimento do produto terá as atualizações do planejamento real, ou seja, a situação atual do produto.

Dados de validação da Fase Implementar

Retornou-se que dezenove (82.6%) dos participantes consideram que o modelo e tabela abordam todos os conceitos necessários para a fase implementar, da mesma maneira que dezoito (78.3%) e vinte um (91.3%) responderam que o modelo é de fácil adoção e facilmente compreensível.

Considerando os dados analisado do modelo da fase implementar, para o participante [P13], "a *Daily SCRUM* deverá ocorrer todos os dias para acompanhamento da *Sprint* em questão, porém com a nova versão do *SCRUM Guide*, foi realizada algumas mudanças, em que as três perguntas que faziam parte, atualmente já não são consideradas obrigatórias". Apesar do modelo ser baseado na terceira edição do SBOK e ainda não ter sido atualizado, é importante ressaltar esta modificação.

Dados de validação da Revisão e Retrospectiva

Vinte (87%) dos participantes responderam que o modelo e tabela abordam todos os conceitos necessários para a fase revisão e retrospectiva, tal como vinte (87%) e vinte e dois (95.7%) responderam que o modelo é de fácil adoção e facilmente compreensível.

Nesta etapa para os participantes [P1] e [P4] informam que "seria significativo expressar melhor a razão da reunião (SoS)", na qual o intuito desta reunião é conectar determinadas equipes para melhor gerenciamento de um projeto complexo.

Um outro ponto informado pelo respondente [P2], foi "deixar mais claro o objetivo da reunião de revisão da *Sprint*, e reunião de retrospectiva", dado que na reunião de revisão é realizado uma apresentação com as partes interessadas do projeto, na qual obterá um *feedback*, de outra maneira a reunião de retrospectiva

identifica os pontos de melhoria.

Para o participante [P13] "os critérios de pronto podem ser destacados para melhor observação", dado que é um ponto importante e que pode reduzir muito o esforço desnecessário em atividades que já foram realizadas, tal como redução de custos.

Dados de validação da Fase Release

Diante das respostas obtidas da fase *release*, retornou-se que vinte (87%) responderam que o modelo e tabela abordam todos os conceitos necessários para a fase *release*, da mesma maneira que vinte (87%) e dezenove (82.6%) responderam que o modelo é de fácil adoção e facilmente compreensível.

Diante dos dados da fase *release*, o respondente [P9] "questiona se de fato está reunião de retrospectiva do projeto é necessária dentro do conceito ágil". O objetivo da reunião de retrospectiva de projeto é verificar todos os pontos que poderão ser corrigidos, tal como melhorias para próximos projetos, diante disso esta reunião visa obter um *feedback* de todo projeto, de fato é uma reunião importante para que a equipe envolvida possa sempre estar em constante melhoramento.

4.2.3 Dados de validação dos modelos no geral

Na (Figura 4.6) observa-se que dezenove (82.6%) responderam que os modelos abordam todos os conceitos e características do SCRUM. Diante do que foi apresentado, para o participante [P13] "seria interessante realizar uma abordagem com menos rigidez em processos, e considerar uma mudança de cultura e pensamento de toda equipe", para isso seria relevante realizar uma abordagem com menos rigidez em processos, e considerar uma mudança no *mindset* de toda equipe.

Para o respondente [P2], "nota-se que a teoria está coesa, todavia em ambiente de trabalho muitas vezes será tratado de uma outra maneira", visto em um ambiente de trabalho utilizando o SCRUM, os processos podem variar, na qual irá depender de como o projeto será desenvolvido, assim como quais serão as diretrizes organizacionais da empresa.

28. Os modelos abordam todos os conceitos e características do SCRUM?

23 respostas

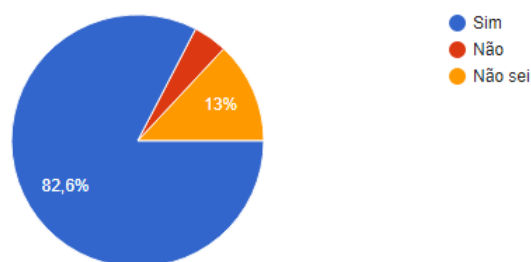


Figura 4.6: Dados da validação, gráfico número 1. Fonte: Autoria Própria.

Dezenove (82.6%) (Figura 4.7) responderam que os modelos facilitam a implantação de uma metodologia ágil, tal qual o participante [P2] indica que "seria significativo tentar deixar as tabelas e modelos mais claros e estruturados, visto que o SCRUM foca em agilidade e simplicidade".

30. Os modelos facilitam a adoção e implantação de uma metodologia ágil?

23 respostas

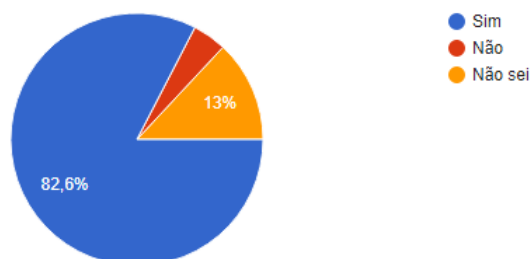


Figura 4.7: Dados da validação, gráfico número 2. Fonte: Autoria Própria.

Na (Figura 4.8) nota-se que quinze (65.2%) responderam que os modelos atendem os requisitos presentes no SBOK.

32. Os modelos atendem os requisitos presentes no SBOK?

23 respostas

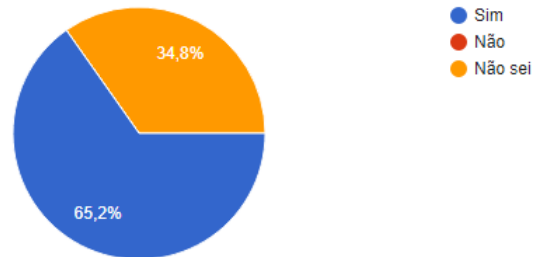


Figura 4.8: Dados da validação, gráfico número 3. Fonte: Autoria Própria.

Na (Figura 4.9) onze (47.8%) participantes informaram que são necessárias modificações para que os modelos de fato atenda o objetivo a que se propõe. Para os participantes [P4] e [P9] "seria considerável realizar uma distinção das atividades por papéis que a executam". O respondente [P13] informa que "poderia ser mais focado no resultado de negócio e na mudança de cultura".

34. Diante do que foi apresentado, você considera que existe algo a ser melhorado para que os modelos de fato atenda o objetivo a que se propõe?

23 respostas

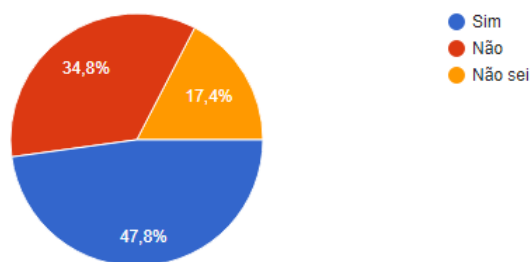


Figura 4.9: Dados da validação, gráfico número 4. Fonte: Autoria Própria.

Capítulo 5

Conclusão

Métodos ágeis estão sendo amplamente adotados para obter melhores resultados em gerenciamento de projetos de software, a palavra ágil significa ser rápido, leve, livre e alerta, a utilização da palavra ágil visa descrever um conceito de modelo de processo. O SCRUM é um entre os métodos ágeis existentes e trata-se de um método que realiza entrega de valor desde o princípio, diante disso são nítidas a redução de custo e do retrabalho durante o desenvolvimento, pois as entregas são feitas em partes, ou seja, em *Sprints* e *releases*.

Apesar da ampla utilização do SCRUM em projetos de software, com a SLR realizada pelo autor em outro artigo que está em processo de revisão, identificou-se que poucos estudos apresentam como de fato é realizado o processo da implantação do SCRUM. Diante disso, este trabalho apresentou por meio de UML as 5 fases de implantação do SCRUM baseadas no SBOK.

O método *design science research* foi utilizado como tipo de pesquisa e tem como o artefato os modelos conceituais elaborados pelo autor.

Os modelos conceituais criados nestes foram validados através de um *survey* com especialistas que trabalham ou já trabalharam com SCRUM. Pôde-se observar a partir dos dados obtidos que algumas alterações necessárias para melhoria dos modelos conceituais, como (1) realizar uma reestruturação das tabelas e métodos, visto que o SCRUM visa reduzir a documentação para simplificar todo o

processo; (2) apresentar definições de papéis no SCRUM; e (3) elencar mudanças no *mindset*, ou seja, mudança de mentalidade e cultura dos times. Assim como alterações apresentadas anteriormente, mudanças significativas como expor melhor o intuito de reuniões de (SoS) e reuniões diárias também são listadas como trabalhos futuros.

Os trabalhos (I) Implantação de scrum em uma empresa de desenvolvimento de software (LEIDEMER, 2014); (II) Aplicação do método ágil scrum no desenvolvimento de produtos de software em uma pequena empresa de base tecnológica (CARVALHO, B. V. d.; MELLO, 2012); (III) Implantação da metodologia ágil scrum em um ambiente de desenvolvimento (SOUZA et al., 2014); (IV) Implantação e análise do *framework* scrum no desenvolvimento da plataforma aberta também propõe formas de implantação para metodologias baseadas em SCRUM (ELIOTE, 2018). Os estudos realizaram a implantação de SCRUM, nota-se um foco por meio de estudos de casos e realizaram observações in loco para fins de validação. Esta pesquisa se diferencia das demais por apresentar um teor mais teórico com orientações baseadas na modelagem conceitual em UML. Após as devidas análises com outros trabalhos da área, constata-se que os objetivos do trabalho “Modelos Conceituais para Implantação da Metodologia Ágil SCRUM em Equipes de Desenvolvimento de Software” de fato foram atingidos e que este trabalho é relevante por relatar a lacuna existente através da SLR desenvolvida pelo autor, tal como apresentar uma nova abordagem para a implantação do SCRUM através dos modelos conceituais propostos.

Referências Bibliográficas

AMBLER, S. W.; LINES, M. **Disciplined agile delivery: A practitioner's guide to agile software delivery in the enterprise**. IBM press, 2012.

_____. The disciplined agile process decision framework. In: SPRINGER. INTERNATIONAL Conference on Software Quality. 2016. p. 3–14.

BAIJENS, J.; HELMS, R.; IREN, D. Applying scrum in data science projects. In: IEEE. 2020 IEEE 22nd Conference on Business Informatics (CBI). 2020. v. 1, p. 30–38.

BERGAMASCHI, A. A.; ZUCHI, J. D. Gerenciamento de tempo com base em informação e metodologias ágeis. **Revista Interface Tecnológica**, v. 15, n. 1, p. 13–27, 2018.

CANO, E. L. et al. A Scrum-based framework for new product development in the non-software industry. **Journal of Engineering and Technology Management**, Elsevier, v. 61, p. 101634, 2021.

CARNEIRO, L. B.; SILVA, A. C. C.; ALENCAR, L. H. Scrum agile project management methodology application for workflow management: A case study. In: IEEE. 2018 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM). 2018. p. 938–942.

CARVALHO, B. V. d.; MELLO, C. H. P. Aplicação do método ágil scrum no desenvolvimento de produtos de software em uma pequena empresa de base tecnológica. **Gestão & Produção**, SciELO Brasil, v. 19, p. 557–573, 2012.

CARVALHO, F. H. T. de. GERENCIAMENTO ÁGIL DE PROJETOS: UMA NOVA FORMA DE LIDAR COM AS INCERTEZAS. **Anderson Leitoguinho Rossi**, p. 143, 2014.

DE CARVALHO, B. V.; MELLO, C. H. P. Scrum agile product development method-literature review, analysis and classification. **Product: Management and Development**, Instituto de Inovação e Gestão de Desenvolvimento do Produto, v. 9, n. 1, p. 39–49, 2011.

ELIOTE, Y. C. D. Implantação e análise do framework scrum no desenvolvimento da plataforma aberta Nosso Exercício. UFVJM, 2018.

FÉLIX, B. M.; TAVARES, E.; CAVALCANTE, N. W. F. Fatores críticos de sucesso para adoção de Big Data no varejo virtual: estudo de caso do Magazine Luiza. **Revista Brasileira de Gestão de Negócios**, SciELO Brasil, v. 20, p. 112–126, 2018.

FERRACINI, G. d. S. Aplicação da metodologia ágil: estudo de caso na área de Customer Success em uma empresa de tecnologia, 2019.

FILHO, J. L.; IOCHPE, C. Um estudo sobre modelos conceituais de dados para projeto de bancos de dados geográficos. **Revista IP-Informática Pública**, v. 1, n. 2, p. 37–90, 1999.

GUEDES, G. 2–Guia Prático-2ª Edição. **São Paulo: Editora Novatec**, 2007.

KAMEI, F. K. et al. Scrum no Serviço Público: um Relato de Implantação nas Secretarias Estaduais da Fazenda e da Gestão Pública do Estado de Alagoas. In: VIII Excellence in Management and Technology Symposium. 2011.

KOM, E. A. P. S.; SURYANI, E. Designing cost measurement system in a small scrum based software company using activity based costing model (case study: ABC company). In: IEEE. 2019 International Conference on Information and Communications Technology (ICOIACT). 2019. p. 943–947.

LACERDA, D. P. et al. Design Science Research: método de pesquisa para a engenharia de produção. **Gestão & produção**, SciELO Brasil, v. 20, n. 4, p. 741–761, 2013.

LEIDEMER, R. H. **Implantação de Scrum em uma empresa de desenvolvimento de software**. 2014. B.S. thesis.

LOPES, C. S. d. S. **Scrum para ambientes de software distribuído: análise crítica e estudo de casos**. 2014. Tese (Doutorado).

MATTIOLI, F. E. et al. Uma proposta para o desenvolvimento ágil de ambientes virtuais. **SBC. Anais do WRVA**, 2009.

NAGARIA, J.; SADATH, L.; AHMED, S. Agile Implementation-A milestone for Academics using Software Engineering Industry Practices. In: IEEE. 2019 Advances in Science and Engineering Technology International Conferences (ASET). 2019. p. 1–6.

PADUA, M. C. O Design Science Research como ferramenta para a gestão de recursos tecnológicos em Instituições Museológicas. **Mescla**, v. 1, n. 1, 2020.

PATANAKUL, P.; RUFO-MCCARRON, R. Transitioning to agile software development: Lessons learned from a government-contracted program. **The Journal of High Technology Management Research**, Elsevier, v. 29, n. 2, p. 181–192, 2018.

PEREIRA, P.; TORREÃO, P.; MARÇAL, A. S. Entendendo Scrum para gerenciar projetos de forma ágil. **Mundo PM**, v. 1, p. 3–11, 2007.

POSSA, J. P. R. Escalonando o Scrum dentro de uma empresa de desenvolvimento de software com equipes geograficamente distribuídas, 2013.

RUBIN, K. S. **Scrum Essencial: Um guia prático para o mais popular processo ágil**. Alta Books Editora, 2018.

SANTOS, W.; ALVES, L. A aplicação da linguagem de modelagem unificada (UML): novas perspectivas para o desenvolvimento de games educacionais. **Anais do Seminário de Jogos Eletrônicos, Educação e Comunicação**, 2015.

SANTOS SOARES, M. dos. Metodologias ágeis extreme programming e scrum para o desenvolvimento de software. **Revista Eletrônica de Sistemas de Informação**, v. 3, n. 1, 2004.

SATPATHY, T. **Um Guia para o CONHECIMENTO EM SCRUM (GUIA SBOK™)**. third: SCRUMstudy, 2017.

SAVOINE, M. et al. Análise de Gerenciamento de Projeto de Software Utilizando Metodologia Ágil XP e Scrum: Um Estudo de Caso Prático. **XI Encontro de Estudantes de Informática do Tocantins**, p. 93–102, 2009.

SCHWABER, K. **Agile project management with Scrum**. Microsoft press, 2004.

SCHWABER, K.; SUTHERLAND, J. La guía de Scrum. **Scrumguides. Org**, v. 1, p. 21, 2013.

SILVA, E. C. da; LOVATO, L. A. Framework Scrum: eficiência em projetos de software. **Revista de Gestão e Projetos**, v. 7, n. 2, p. 01–15, 2016.

SILVA, M. A. A. M. d. **Aplicação do Earned Value Management na Metodologia Scrum**. 2017. Tese (Doutorado).

SILVA, R. T. Aplicação da metodologia scrum para gestão de projetos na indústria naval, 2011.

SORDI, J. O. D.; AZEVEDO, M. C. d.; MEIRELES, M. A pesquisa design science no Brasil segundo as publicações em administração da informação. **JISTEM- Journal of Information Systems and Technology Management**, SciELO Brasil, v. 12, p. 165–186, 2015.

SOUZA, D. R. d. et al. Implantação da metodologia ágil Scrum em um ambiente de desenvolvimento. Araranguá, SC, 2014.

SPECIFICATION, O. A. OMG Unified Modeling Language (OMG UML), Superstructure, V2. 1.2.

TERLIZZI, M. A.; BIANCOLINO, C. A. Projeto de Software no setor bancário: Scrum ou Modelo V. **TAC, Rio de Janeiro**, v. 4, n. 1, p. 46–58, 2014.

VALLERÃO, A. G.; ROSES, L. K. Monitoramento e controle de projetos de desenvolvimento de Software com o Scrum: avaliação da Produção Científica. **Gestão e Projetos: GeP**, Universidade Nove de Julho, v. 4, n. 2, p. 100–127, 2013.

Apêndice A

Informações adicionais

A.1 Declaração do Orientador Para a Biblioteca

Eu, *Prof(a). Dr(a). Joyce Aline de Oliveira Marins*, na qualidade de orientador(a) do aluno *Pedro Paulo Santana Costa* do Curso de Engenharia de Computação declaro para os devidos fins que o trabalho intitulado **Modelos Conceituais para Implantação da Metodologia Ágil SCRUM em Equipes de Desenvolvimento de Software** RESPEITA TODOS OS DIREITOS AUTORAIS, estando isento de plágio, cópias ilegais ou quaisquer ofensas aos direitos de outros autores, em conformidade com o que rege a Lei nº 9.610/98. Declaro, ainda, que o trabalho em questão passou por uma banca de avaliação, sendo realizadas as devidas correções e, estando assim, apto a ser disponibilizado em texto integral, na Biblioteca Digital da UFMT ou em qualquer outro sistema de automação e gestão de acervos, utilizado pela Instituição, para consulta e acesso livre de modo online.

Apêndice B

Survey de validação

B.1 Formulário de validação dos modelos

Formulário de validação dos modelos

Este formulário tem o intuito de apresentar os processos definidos como sendo essenciais para elaboração de um projeto em SCRUM. Os modelos elaborados são baseado no SBOK e é representado em Linguagem Unificada de Modelagem (UML) por meio de um diagrama de atividades. Ao responder este questionário, você nos ajudará a validar os modelos construído. Manteremos o anonimato dos seus dados. Sua participação é muito importante para a nossa pesquisa.

***Obrigatório**

1. E-mail *

2. 1. Nome

1 ponto

3. 2. Qual o seu nível de formação? *

Marcar apenas uma oval.

- Graduação
- Estudante de mestrado
- Mestrado
- Estudante de Doutorado
- Doutorado
- Estudante de PhD
- Pós-doutorado

4. 3. Em qual organização você trabalha? *

Marcar apenas uma oval.

- Indústria
- Acadêmico
- Serviço Público
- Empresarial
- Industrial/Empresarial
- Outro: _____

5. 4. Qual a sua atual função em seu trabalho? *

6. 5. Qual o seu tempo de experiência em TI? *

Marcar apenas uma oval.

- 0 - 2 Anos
- 2 - 4 Anos
- 4 - 6 Anos
- 6 Anos ou mais

7. 6. Há quanto tempo trabalha em projetos baseados em SCRUM? *

Marcar apenas uma oval.

- 1 - 2 Anos
- 2 - 4 Anos
- Mais de 4 anos

8. 7. Em quantos projetos baseados em SCRUM você já atuou até o momento? *

Marcar apenas uma oval.

- 1 - 2 Projetos
- 3 - 4 Projetos
- mais de 4 projetos

Para responder as questões apresentadas a seguir é sugerido que você siga os seguintes passos:

1°. Para maior detalhamento sobre todo processo, você pode consultar a 3ª edição do SBOOK, apresentado a seguir: https://portaldaoobmep.impa.br/uploads/material_teorico/d9gycv8klfcwc.pdf

2°. Observar as tabelas para cada fase, pois nelas contém as Entradas, Ferramentas e Saídas necessárias para cada fase no SCRUM.


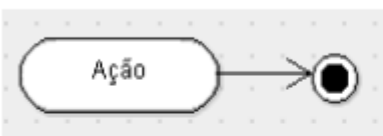
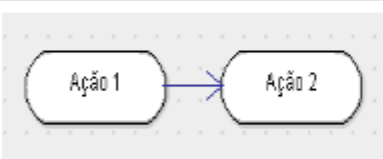

3°. Observe o passo a passo desenvolvido em UML para cada fase do SCRUM.

4° Caso possua alguma dúvida em relação aos termos utilizados, observe o glossário.

5° Após analisar o modelo, responda as questões.

6° Para cada questão, existe um campo para coletar sugestões, por favor, deixe a sua opinião, críticas ou sugestões.

Glossário - Simbologia do diagrama.

	<p>Símbolo de Início - Utilizado para indicar estado inicial da modelagem.</p>
	<p>Símbolo de término - Utilizado para indicar estado final da modelagem.</p>
	<p>Símbolo de atividade - Utilizado para indicar um estado de ação ou atividade a ser realizada.</p>
	<p>Símbolo de conector - Utilizado para indicar o fluxo, ou seja, a direção da atividade ou ação</p>

Glossário - Termos SCRUM.

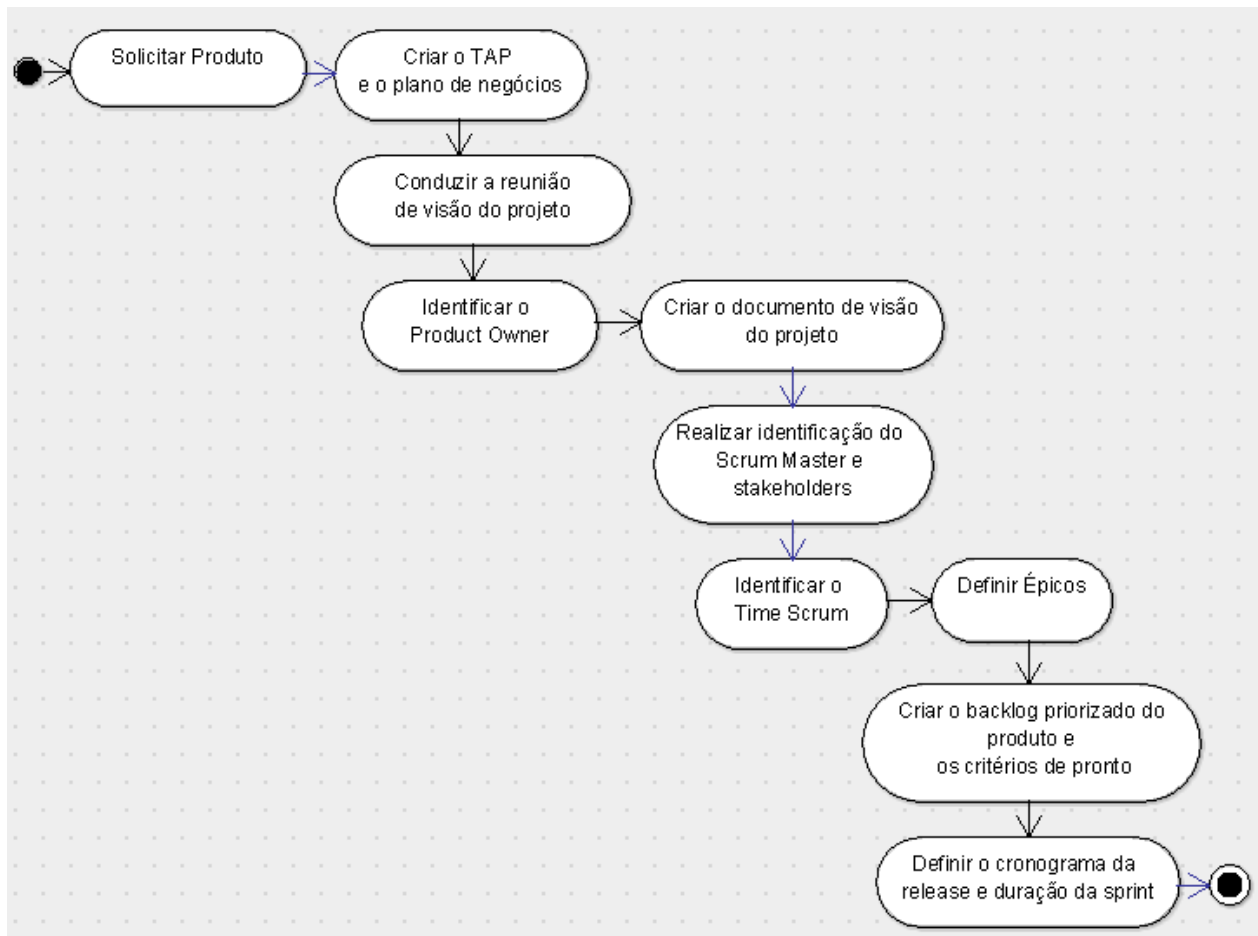
Épicos	São <u>Estórias de Usuário</u> grandes e não refinadas no Backlog Priorizado do Produto.
Stakeholders	Os stakeholders são todos os clientes, usuários e patrocinadores, que interagem frequentemente com o Dono do Produto, Scrum Master e Time Scrum para fornecer inputs e para facilitar a criação de produtos do projeto.
Personas	As Personas são personagens fictícios altamente detalhados, representantes da maioria dos usuários, bem como outros stakeholders, que podem não usar diretamente o produto final.
Release	A release enfatiza a entrega dos Entregáveis Aceitos para o cliente, e a identificação, documentação e internalização das lições aprendidas durante o projeto.
Estória de usuário	As Estórias de Usuário são requisitos específicos descritos por vários stakeholders, no que refere-se ao produto ou serviço proposto.
Gráfico Burndown	O Gráfico Burndown é um gráfico que mostra a quantidade de trabalho restante durante o desenvolvimento do Sprint.
ScrumBoard	O <u>Scrumboard</u> é uma ferramenta utilizada pelo Time Scrum para planejar e acompanhar o progresso durante cada Sprint.
Impedimentos	Um impedimento é qualquer entrave ou obstáculo que reduza a produtividade do Time Scrum.
Critério de pronto	Os Critérios de Pronto são um conjunto de regras aplicáveis a todas as <u>Estórias de Usuário</u> .
TAP	Termo de abertura do projeto

Fase Iniciar

Fase Iniciar - Tabela com Entradas, Ferramentas e Saídas definidas pelo SBOK como sendo obrigatórias para a elaboração de um projeto SCRUM.

FASE INICIAR	
1. Criar a Visão do Projeto	
ENTRADAS	1.Caso de negócio do projeto*
FERRAMENTAS	1.Reunião de visão do projeto*
SAÍDAS	1.O product Owner é identificado*, 2.Declaração da visão do projeto*
2. Identificar o Scrum Master e o(s) Stakeholder(s)	
ENTRADAS	1.Product Owner*, 2.Declaração da visão do projeto*
FERRAMENTAS	1.Critérios de seleção*
SAÍDAS	1.Scrum Master identificado*, 2.Stakeholders identificados*
3. Formar o Time Scrum	
ENTRADAS	1.Product Owner*, 2.Scrum Master*, 3.Declaração da visão do projeto*
FERRAMENTAS	1.Seleção do time Scrum*
SAÍDAS	1.Time Scrum identificado*
4. Desenvolver os Épicas	
ENTRADAS	1.Time central Scrum*, 2.Declaração da visão do projeto*
FERRAMENTAS	1.Reuniões dos grupos de usuários*
SAÍDAS	1.Épicos*, 2.Personas*
5. Criar o Backlog Priorizado do Produto	
ENTRADAS	1.Time central Scrum*, 2.Épicos*, 3.Personas*
FERRAMENTAS	1.Métodos de priorização da história de usuário*
SAÍDAS	1.Backlog do produto priorizado*, 2.Critério de pronto*
6. Conduzir o Planejamento da Release	
ENTRADAS	1.Time central Scrum*, 2.Stakeholders*, 3.Declaração da visão do projeto*, 4.Backlog priorizado do produto*, 5.Critério de pronto*
FERRAMENTAS	1.sessões de planejamento da release*, 2.Métodos de priorização da release*
SAÍDAS	1.Cronograma de planejamento da release*, 2.Duração da sprint*

Fase Iniciar - A fase iniciar elenca processos importantes para todo o projeto, visto que aqui será definido a visão de todo o projeto, definição dos papéis essenciais e não essenciais dentro do SCRUM, tal como as entregas de valor no decorrer do projeto, visando as necessidades do usuário.



9. 8. O modelo e a tabela abordam todos os conceitos necessários para a fase Iniciar? *

Marcar apenas uma oval.

- Sim
- Não
- Não sei

10. 9. O modelo da fase Iniciar é de fácil adoção, ou seja, olhando o modelo, você conseguiria executar a fase Iniciar? * 1 ponto

Marcar apenas uma oval.

- Sim
- Não
- Não sei

11. 10. O modelo referente a fase iniciar é facilmente compreensível? * 1 ponto

Marcar apenas uma oval.

- Sim
- Não
- Não sei

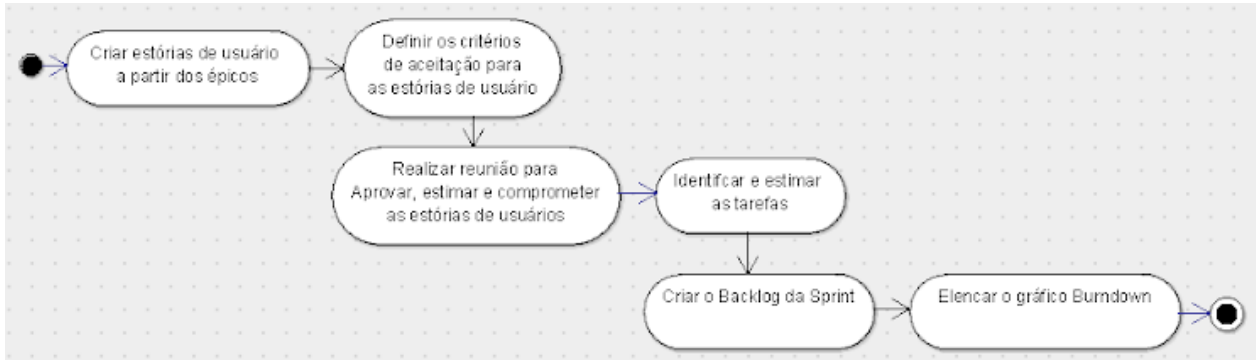
12. 11. Comentários/ Sugestões/ Exemplos

Fase Planejar e Estimar

Fase Planejar e Estimar - Tabela com Entradas, Ferramentas e Saídas definidas pelo SBOK como sendo obrigatórias para a elaboração de um projeto SCRUM.

FASE PLANEJAR E ESTIMAR	
7. Criar as Estórias de Usuário	
ENTRADAS	1.Time central Scrum*, 2.Backlog priorizado do produto*, 3.Critério de pronto*, 4.Personas*
FERRAMENTAS	1.Expertise de escrever a estória de usuário*
SAÍDAS	1.Estórias de usuários*, 2.Critérios de aceitação da Estórias de usuários*
8. Aprovar, Estimar, e Comprometer as Estórias de Usuário	
ENTRADAS	1.Time Central de Scrum*, 2.Estórias de Usuário*, 3.Critérios de Aceitação da Estória de usuário*
FERRAMENTAS	1.Reuniões do Grupo de Usuários*
SAÍDAS	1.Estórias de Usuário Aprovadas, Estimadas e Comprometidas*
9. Criar as Tarefas	
ENTRADAS	1.Time central do Scrum*, "2.Estórias de Usuário Aprovadas, Estimadas e Comprometidas*
FERRAMENTAS	1.Reuniões de Planejamento de Tarefas*
SAÍDAS	1.Lista de Tarefas*
10. Estimar as Tarefas	
ENTRADAS	1.Time central Scrum*, 2.Lista de Tarefas*
FERRAMENTAS	1.Reuniões de Estimativa de Tarefas*, 2.Critério de Seleção*
SAÍDAS	1.Lista de Tarefas de Esforço Estimado*
11. Criar o Backlog do Sprint	
ENTRADAS	1.Time Central do Scrum*, 2.Lista de Tarefas de Esforço Estimado*, 3.Duração do Sprint*
FERRAMENTAS	1.Reuniões de Planejamento do Sprint*
SAÍDAS	1.Backlog do Sprint*, 2.Gráfico Burndown do Sprint*

Fase Planejar e Estimar - Nesta fase nota-se o empenho em definir o que será trabalhado durante a Sprint, visto que as estórias de usuário terão critérios de aceitação definidos, como também serão aprovadas, estimadas e comprometidas. Observa-se que em um projeto SCRUM poderá ou não ser definido tarefas dentro das estórias de usuário. Com o planejamento da Sprint realizado, terá então o gráfico burndown criado.



13. 12. O modelo e a tabela abordam todos os conceitos necessários para a fase Planejar e Estimar? *

Marcar apenas uma oval.

- Sim
- Não
- Não sei

14. 13. O modelo da fase planejar e estimar é de fácil adoção, ou seja, olhando o modelo, você conseguiria executar a fase Planejar e Estimar? *

Marcar apenas uma oval.

- Sim
- Não
- Não sei

15. 14. O modelo referente a fase Planejar e Estimar é facilmente compreensível? *

Marcar apenas uma oval.

- Sim
 Não
 Não sei

16. 15. Comentários/ Sugestões/ Exemplos

Fase Implementar

Fase Implementar - Tabela com Entradas, Ferramentas e Saídas definidas pelo SBOK como sendo obrigatórias para a elaboração de um projeto SCRUM.

FASE IMPLEMENTAR	
12. Criar os Entregáveis	
ENTRADAS	1.Time Central do Scrum*, 2.Backlog do Sprint*, 3.Scrumboard*, 4.Registro de Impedimento*
FERRAMENTAS	1.Expertise do Time*
SAÍDAS	1.Entregável do Sprint*, 2.Scrumboard Atualizado*, 3.Registro de Impedimento Atualizado*
13. Conduzir a Reunião Diária	
ENTRADAS	1.Time Scrum*, 2.Scrum Master*, 3.Gráfico Burndown Sprint*, 4.Registro de Impedimento*
FERRAMENTAS	1.Reunião Diária*, 2.Três Perguntas Diárias*
SAÍDAS	1.Gráfico Burndown do Sprint Atualizado*, 2.Registro de Impedimento Atualizado*
14. Refinamento do Backlog Priorizado do Produto	
ENTRADAS	1.Time Central do Scrum*, 2.Backlog Priorizado do Produto*
FERRAMENTAS	1.Reuniões de Revisão do Backlog Priorizado do Produto*
SAÍDAS	1.Backlog Priorizado do Produto Atualizado*

Fase Implementar - A fase Implementar consiste em desenvolver o produto em questão, visto que com a definição dos entregáveis da Sprint terá então Scrumboard de desenvolvimento atualizado, bem como impedimentos caso existam. Um ponto de suma importância em projeto SCRUM é a realização da reunião diária. Nesta fase, também acontecerá a reunião para ter o Backlog do produto priorizado atualizado de acordo com critérios do usuário.



17. 16. O modelo e a tabela abordam todos os conceitos necessários para a fase Implementar? *

Marcar apenas uma oval.

- Sim
- Não
- Não sei

18. 17. O modelo da fase Implementar é de fácil adoção, ou seja, olhando o modelo, você conseguiria executar a fase Implementar? *

Marcar apenas uma oval.

- Sim
- Não
- Não sei

19. 18. O modelo referente a fase Implementar é facilmente compreensível? *

Marcar apenas uma oval.

- Sim
- Não
- Não sei

20. 19. Comentários/ Sugestões/ Exemplos

Fase Revisão e Restrospectiva

Fase Revisão e Restrospectiva - Tabela com Entradas, Ferramentas e Saídas definidas pelo SBOK como sendo obrigatórias para a elaboração de um projeto SCRUM.

FASE REVISÃO E RETROSPECTIVA	
15. Convocar o Scrum de Scrums	
ENTRADAS	1. Scrum Master ou Representantes do Time Scrum*
FERRAMENTAS	1. Reunião do Scrum de Scrums*, 2. Quatro Perguntas por Time*
SAÍDAS	1. Coordenação Melhor do Time*
16. Demonstrar e Validar o Sprint	
ENTRADAS	1.Time Central do Scrum*, 2.Entregáveis do Sprint*, 3.Backlog do Sprint*, 4.Critério de Pronto*, 5.Critérios de Aceitação da Estória de Usuário*
FERRAMENTAS	1.Reuniões de Revisão do Sprint*
SAÍDAS	1.Entregáveis Aceitos*
17. Retrospectiva do Sprint	
ENTRADAS	1.Scrum Master*, 2.Time Scrum*
FERRAMENTAS	1.Reunião de Retrospectiva do sprint*
SAÍDAS	1.Pontos de Melhoria Acordados*

Fase Revisão e Retrospectiva - Na fase revisão e retrospectiva observa-se que é realizado três reuniões, sendo a primeira a reunião (SoS), para obter melhor coordenação entre times, seguida da revisão da Sprint, com intuito de elencar os entregáveis aceitos, tal como a reunião de retrospectiva da Sprint, com foco em identificar pontos de melhoria diante dos processos ocorridos anteriormente.



21. 20. O modelo e a tabela abordam todos os conceitos necessários para a fase Revisão e Retrospectiva? *

Marcar apenas uma oval.

- Sim
- Não
- Não sei

22. 21. O modelo da fase revisão e retrospectiva é de fácil adoção, ou seja, olhando o modelo, você conseguiria executar a fase Revisão e Retrospectiva? *

Marcar apenas uma oval.

- Sim
- Não
- Não sei

23. 22. O modelo referente a fase Revisão e Retrospectiva é facilmente compreensível? *

Marcar apenas uma oval.

- Sim
 Não
 Não sei

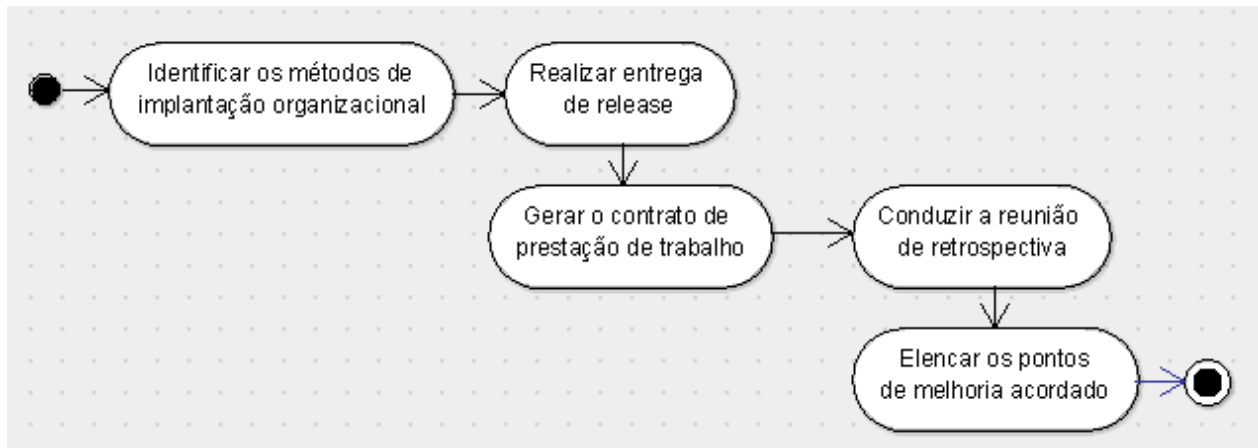
24. 23. Comentários/ Sugestões/ Exemplos

Fase Release

Fase Release - Tabela com Entradas, Ferramentas e Saídas definidas pelo SBOK como sendo obrigatórias para a elaboração de um projeto SCRUM.

FASE RELEASE	
18. Enviar os Entregáveis	
ENTRADAS	1.Dono do Produto*, 2.Stakeholder(s)*, 3.Entregáveis Aceitos*, 4.Cronograma de Planejamento da release*
FERRAMENTAS	"1.Métodos de Implantação Organizacional**"
SAÍDAS	1.Contrato de Prestação de Trabalho*
19. Retrospectiva do Projeto	
ENTRADAS	1.Time(s) Central(s) do Scrum*
FERRAMENTAS	1.Reunião de Retrospectiva do Projeto*
SAÍDAS	1.Pontos de Melhoria Acordados*

Fase Release - Na ultima fase, primeiramente é feita a identificação dos métodos de implantação do produto, ou seja, como será colocado em uso produto em questão. Diante disso a release será entregue e será gerado o contrato de prestação de trabalho, sendo este um documento formal de aceite. Com estes processos finalizados será realizado a reunião de retrospectiva, na qual serão identificados pontos de melhoria para futuros projetos.



25. 24. O modelo e a tabela abordam todos os conceitos necessários para a fase Release? *

Marcar apenas uma oval.

- Sim
- Não
- Não sei

26. 25. O modelo da fase release é de fácil adoção, ou seja, olhando o modelo, você conseguiria executar a fase Release? *

Marcar apenas uma oval.

- Sim
- Não
- Não sei

27. 26. O modelo referente a fase Release é facilmente compreensível? *

Marcar apenas uma oval.

Sim

Não

Não sei

28. 27. Comentários/ Sugestões/ Exemplos

29. 28. Os modelos abordam todos os conceitos e características do SCRUM? *

Marcar apenas uma oval.

Sim

Não

Não sei

30. 29. Comentários/ Sugestões/ Exemplos

31. 30. Os modelos facilitam a adoção e implantação de uma metodologia ágil? *

Marcar apenas uma oval.

- Sim
- Não
- Não sei

32. 31. Comentários/ Sugestões/ Exemplos

33. 32. Os modelos atendem os requisitos presentes no SBOK? *

Marcar apenas uma oval.

- Sim
- Não
- Não sei

34. 33. Comentários/ Sugestões/ Exemplos

35. 34. Diante do que foi apresentado, você considera que existe algo a ser melhorado para que os modelos de fato atenda o objetivo a que se propõe? *

Marcar apenas uma oval.

- Sim
- Não
- Não sei

36. 35. Comentários/ Sugestões/ Exemplos

Este conteúdo não foi criado nem aprovado pelo Google.

Google Formulários