

Ministério da Educação

Secretaria Executiva

Subsecretaria de Tecnologia da Informação e Comunicação

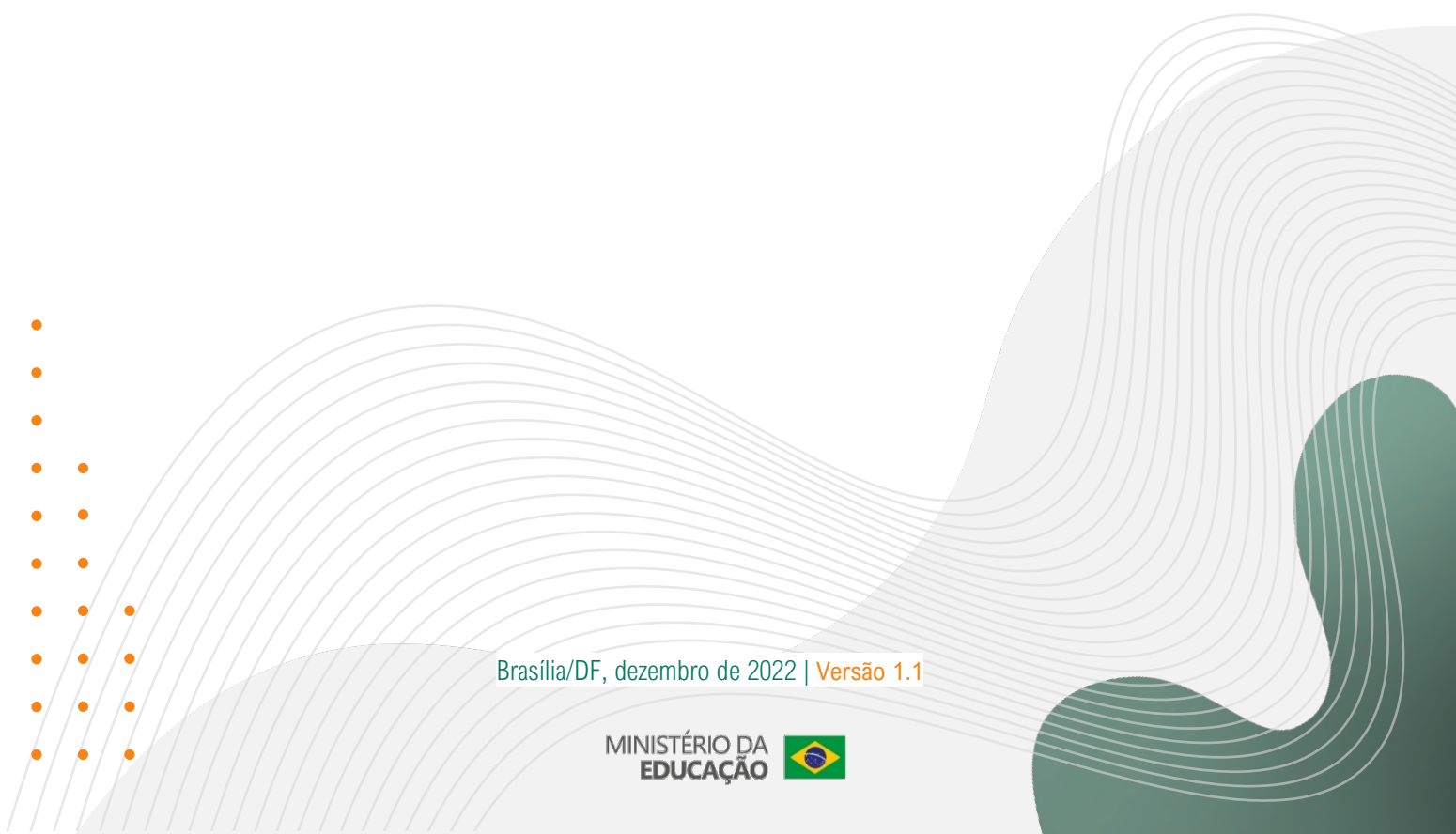


METODOLOGIA ÁGIL

DE GESTÃO DE DESENVOLVIMENTO DE SOFTWARE

D O M I N I S T É R I O D A E D U C A Ç Ã O

Brasília/DF, dezembro de 2022 | Versão 1.1



METODOLOGIA DE GESTÃO DE DESENVOLVIMENTO ÁGIL DE SOFTWARE

Ministério da Educação

Victor Godoy Veiga
Ministro de Estado da Educação

José Barreto de Castro Júnior
Secretário-Executivo

Subsecretaria de Tecnologia da Informação e Comunicação

André Henrique dos Santos Castro
Subsecretário

Isabelle Nascimento França
Coordenadora de Gabinete

Delson Pereira da Silva
Gerente de Governança, Projetos e Aquisições

Edgard Carvalho Ribeiro Neto
Coordenador-Geral de Infraestrutura e Segurança da Informação

Eduardo Conde Pires
Coordenador-Geral de Governança de Dados

Márcio Fernandes Cunha
Coordenador-Geral de Sistemas

Regina Carla Oliveira Frazão
Coordenadora-Geral de Serviços de TI



Sumário

Apresentação	1
Abordagem	3
Valores	5
Papéis	7
Gerente do Produto	8
Scrum Master	9
Time de Desenvolvimento	10
Fases	11
Preparação	11
Concepção	12
Construção	12
Homologação	12
Implantação	13
Artefatos	14
Backlog do Produto	14
Sprint Backlog	15
Incremento	16
Definição de Pronto.....	16
Cerimônias	17
Sprint.....	17
Sprint Planning.....	18
Daily Meeting	19
Sprint Review	20
Sprint Retrospective	21
Práticas	22
Timeboxing.....	22
Método de Priorização de Requisitos MoSCoW	22
Refinamento do Backlog do Produto.....	23
Sprint Burn-down.....	23
Velocity Chart	23
Planning Poker	24

Apêndices

APÊNDICE 01	Matriz de atribuições e responsabilidades	25
APÊNDICE 02	Classificação das demandas	26
APÊNDICE 03	Papéis gerenciais complementares	27
APÊNDICE 04	Formatos e meios padronizados de entrega	28
APÊNDICE 05	Padrões técnicos de desenvolvimento	29
APÊNDICE 06	Ferramentas de apoio ao desenvolvimento	30
APÊNDICE 07	Mapa de processo da Metodologia Ágil MEC	31
APÊNDICE 08	Esteira DevSecOps.....	32



Apresentação

O pensamento ágil é propositalmente disruptivo. Desde o Manifesto Ágil de 2001 podemos dizer que esse movimento reinventou o desenvolvimento de *software*, transformando muitas organizações ao redor do mundo.

Dentre as metodologias ágeis da atualidade, o Scrum é reconhecidamente a mais popular e evoluiu nos últimos 25 anos para se tornar a estrutura de processo mais aplicada em desenvolvimento de *software*. No entanto, sua aparente simplicidade – com seus três papéis, três artefatos e cinco eventos – esconde os grandes desafios de sua implementação.

Na busca por melhorar a satisfação dos clientes e reduzir o tempo das entregas de *software*, a Subsecretaria de Tecnologia da Informação e Comunicação se desafiou a protagonizar a elevação da maturidade em processos ágeis no Ministério da Educação.

Com isso, amparados em um movimento de mudança de paradigmas, apresentamos a primeira versão da Metodologia Ágil de Gestão de Desenvolvimento do Ministério da Educação. Que, essencialmente baseada em Scrum, que tem como principais fundamentos:

- Implementar os valores do modelo ágil;
- Estabelecer papéis e responsabilidades;
- Definir um processo mínimo a partir do qual ingressamos na jornada ágil;
- Explorar processos complementares para produzir uma implementação eficiente; e
- Assentar as bases de uma cultura de acompanhamento, qualidade e melhoria contínua.

Nesse contexto, entendemos que escolhemos usar Scrum como nosso caminho para construir times ágeis e de alta performance, mas não como nossa finalidade. Ou seja, Scrum não será executado como uma “receita” e sim como uma linha guia que auxiliará nossos times a construir maturidade, reagir às mudanças e entregar valor de maneira contínua e incremental em um processo de melhoria contínua.

Dentro desse mindset assumimos que *software* é construído por partes (modelo incremental) em um ciclo iterativo onde o projeto pode ser dividido em pequenos módulos de entrega, abordando um escopo reduzido e adaptativo – no qual, inclusive, o cliente pode ainda não ter a certeza do todo e ir evoluindo de forma gradual e adaptável, dentro de uma visão de futuro. Os projetos são guiados por entrega de valor, onde o cliente pode ver produto funcional desde a primeira entrega e prover feedback contínuo tanto sobre o produto quanto sobre o processo de desenvolvimento.

Outrora, nos resta claro que o processo de transformação digital não se resume à camada de desenvolvimento de *software* – antes, é uma responsabilidade distribuída, tanto na TI quando nas áreas de negócio.





Abordagem

A abordagem ágil do Ministério da Educação contempla a essência da estrutura Scrum, com processos e práticas complementares, visando a garantir sua adoção bem-sucedida, o que inclui:

Valores	Comprometimento	Coragem	Foco	Abertura	Respeito	
Papéis	Gerente do Produto Product Owner		Scrum Master		Equipe de Desenvolvimento	
Fases	Preparação	Concepção	Construção	Homologação	Implantação	
Artefatos	Backlog do Produto	Sprint Backlog	Incremento		Definição de Pronto	
Cerimônias	Sprint Planning	Sprint	Daily Meeting	Sprint Review	Sprint Retrospective	
Práticas	Timeboxing	MoSCoW	Refinamento do Backlog	Sprint Burn-down	Velocity Chart	Planning Poker



Valores

A proficiência no processo depende de as equipes adotarem e refletirem sobre os valores ágeis para explorar comportamentos e evoluir seu processo de desenvolvimento.

Esses valores fundamentais fornecem clareza à maneira como as equipes devem construir sua identidade e agir para garantir a melhoria contínua e evitar falhas no produto, são eles:



Além desses valores, temos também que o processo ágil é assentado em três pilares:





Papéis

O Time Ágil é composto essencialmente por três papéis, são eles: O Gerente do Produto (originalmente, Product Owner), Scrum Master e Time de Desenvolvimento. Há um conjunto de princípios universais aplicáveis a todos os membros dessa equipe, assim o Time Ágil:

- Trabalha verdadeiramente como um time, além de ser pequeno e auto-organizado;
- Compartilha a responsabilidade, colabora e se comunica clara e constantemente, de forma direta;
- Foca na entrega de valor contínua e incremental, aprendendo a cada entrega através da melhoria contínua;
- Compartilha seus aprendizados dentro do time e com as partes interessadas

Esses três papéis descrevem um conjunto de responsabilidades a serem atribuídas e executadas pelos membros de um Time Ágil, mas não são cargos ou funções. Ou seja: qualquer cargo, mesmo os que já existem na estrutura da organização, podem desempenhar esses papéis.

Considerando que a essência da Metodologia é a auto-organização e a melhoria contínua, os três papéis recebem uma definição mínima de responsabilidades e prestação de contas para permitir que as equipes entreguem seus trabalhos com efetividade. Assim as equipes podem assumir a responsabilidade de como se organizam e continuar melhorando. Nos itens a seguir detalhamos cada um desses papéis e seu respectivo conjunto de responsabilidades.





Gerente do Produto



Papel exercido pelo representante máximo do cliente/produto dentro do Time Ágil. Traduz a visão e as necessidades do negócio em requisitos e funcionalidades, administra o backlog e prioriza os itens para o desenvolvimento. Deve possuir autoridade para tomar as decisões necessárias sobre o produto e disponibilidade para interagir com o time durante as diversas fases do processo.

O Gerente do Produto (originalmente Product Owner)¹ é o representante máximo do cliente/produto dentro do Time Ágil. É ele quem conhece o problema, as "dores", o retorno esperado pelo projeto e é dele a responsabilidade por usar toda sua expertise para guiar o Time de Desenvolvimento para que o seu trabalho seja o mais otimizado possível, do ponto de vista de Retorno Sobre o Investimento (ROI).

O Gerente do Produto atua como a voz do cliente para orientar as equipes de desenvolvimento na criação de recursos de software. Ele representa as necessidades e prioridades do negócio, prioriza e gerencia o backlog do produto e incorpora o feedback das partes interessadas do negócio para orientar a melhoria contínua do produto (Gartner, 2022). Enquanto que nos modelos tradicionais as conversas com o cliente resumem-se a reuniões pontuais em marcos do projeto (ou, por vezes, apenas no início e no fim do projeto), o Gerente do Produto é um membro full-time do Time Ágil, estando sempre à disposição dos desenvolvedores para resolver todo tipo de impedimento de negócio, dúvidas de priorização de atividades e principalmente, para trabalhar constantemente na definição e no refinamento do Backlog do Produto – incluindo seus desdobramentos (sprints, releases e roadmap).

Junto às partes interessadas, o Gerente do Produto transforma a estratégia e os requisitos de negócio (visão do produto) em especificações para os desenvolvedores, como Histórias de Usuário, por exemplo. É ele que faz a ponte entre o Time Ágil e o negócio, sendo a sua palavra soberana neste assunto. Nesse ponto é importante destacar que cada Produto deve possuir um único Gerente de Produto

O Gerente do Produto possui um papel crucial na Sprint Planning, quando deve apresentar ao time quais são os próximos objetivos para o produto e as prioridades do backlog. Além disso, enquanto o Time de Desenvolvimento estiver trabalhando numa Sprint, o Gerente do Produto deve estar sempre planejando e organizando a próxima. Também faz parte do ciclo sua participação na Sprint Review, onde geralmente ele recebe o Incremento de software que o time gerou e o apresenta às demais partes interessadas, caso ele já conheça o resultado da Sprint.

Junto ao Scrum Master, o Gerente do Produto compartilha a servo-liderança do Time, guiando todos para o maior sucesso possível em termos do produto. Embora não defina aspectos tecnológicos do produto, o Gerente do Produto provê a orientação necessária do que deve ser construído primeiro, o que faz mais sentido com a estratégia de "go to market" (lançamento) do produto, dentro do "time to market" (tempo de lançamento).

Dentre os requisitos para exercer esse papel, destacamos que o Gerente do Produto deve ter alto nível de conhecimento do produto e do negócio no qual está inserido – isso é essencial pelo fato de que é ele quem orientará a construção do resultado evitando o risco das entregas sem valor, sendo esse justamente um dos principais problemas que os métodos ágeis buscam resolver. Outro requisito é a disponibilidade tempestiva para o contínuo detalhamento do backlog e esclarecimentos da solução ao time de desenvolvimento ao longo das releases e iterações, sob pena de não cumprimento da meta de negócio definida.

¹ Entendemos que utilizar a designação de "Gerente do Produto" ao invés de Dono do Produto (ou Product Owner), como originalmente denominado em Scrum, é mais adequado ao contexto. No Governo, um servidor atua como gestor de uma determinada área e/ou política pública, sendo que suas decisões, sujeitas aos princípios legais e de accountability, sempre serão uma materialização do interesse público.





Scrum Master



Papel exercido pelo profissional que domina os processos e as regras da Metodologia, com o objetivo de garantir seu cumprimento e atuar na remoção de impedimentos ao trabalho do Time Ágil. Atua como líder-servidor responsável por levar o Time Ágil à eficácia, através da melhoria de suas práticas.

Scrum Master é o papel no Time Ágil que domina os processos e as regras contidas na Metodologia Ágil e tem como missão principal garantir seu cumprimento por todos. O Scrum Master é um líder-servidor que puxa a melhoria contínua e auxilia o time na obtenção dos melhores resultados possíveis através da remoção de impedimentos e promoção de novas práticas e técnicas que aproximem mais o time dos objetivos esperados.

O Scrum Master facilita e defende o processo de desenvolvimento ágil de software, treinando e orientando as equipes de desenvolvimento, identificando e eliminando impedimentos e promovendo o uso de padrões e melhores práticas. Cabe a ele garantir o bom funcionamento dos processos e atividades ágeis (Gartner, 2022).

Cabe ao Scrum Master ser o mentor da Metodologia zelando para que todas as cerimônias, papéis e artefatos estejam presentes e sejam respeitados. Geralmente é tarefa dele também conduzir todas as cerimônias atuando como um árbitro facilitador, até que o time consiga entrar em sincronia para conseguirem conduzirem as mesmas por conta própria.

Deve ser capaz de mediar conflitos dentro e fora do time, para evitar que afetem o cumprimento do objetivo das Sprints. Conflitos estes que podem ter como resolução mudanças comportamentais, culturais e até técnicas, sendo um agente de mudança organizacional.

Também deve acompanhar de perto métricas de produtividade do Time como Lead Time, Cycle Time, Velocity, Story Points, Work in Progress e outros KPIs estabelecidos, para garantir que qualquer desvio nos padrões aceitáveis seja corrigido o mais rápido possível sem afetar o objetivo das Sprints.

Junto ao Time de Desenvolvimento, cabe ao Scrum Master promover uma cultura Lean² de desenvolvimento de software, dentro do que prescreve a Metodologia. Junto ao Gerente do Produto (ou, Product Owner), o Scrum Master compartilha da liderança ágil do time (mas jamais chefia ou gerência), sendo um mentor do PO no que tange métodos ágeis e auxiliando-o na comunicação com o restante da empresa e na tradução da linguagem de negócio com os técnicos e vice-versa. Junto às demais partes interessadas, o Scrum Master é o “embaixador ágil” que circula e comunica como a Metodologia funciona, ajudando a empoderar Gerentes de Produto e Desenvolvedores, função primordial para o sucesso de qualquer transformação ágil.

Além disso, tal como não existem duas pessoas iguais no mundo, não existem dois Scrum Masters iguais. Considerando este básico que permeia a função, uma infinidade de outras competências e responsabilidades podem surgir conforme a necessidade e contexto de cada projeto/organização, cabendo ao Scrum Master ter o discernimento de assumir ou não tais responsabilidades para si, desde que sempre blindando o básico do seu papel tal qual descrito na Metodologia.

Independentemente do que compõe o rol completo de atividades do Scrum Master, o que considera o contexto de cada organização/projeto, sua existência e presença constante de cada Time Ágil não pode, em nenhuma hipótese, ser omitido. Assim, quando não houver possibilidade de ter um profissional dedicado a esta função, em último caso recomenda-se que um membro do time funcione como Scrum Master part-time, além de suas outras atribuições.

² *Lean Thinking* é uma estratégia que se concentra na eliminação do desperdício, variação e tarefas sem valor agregado. É um compromisso cultural de melhoria contínua e foco no cliente através da otimização do fluxo de informações e materiais.





Time de Desenvolvimento



Papel exercido pelos profissionais responsáveis por transformar o Backlog do Produto em incrementos que possam ser entregues, considerando as diversas competências multidisciplinares da engenharia de software.

No Time Ágil, enquanto o Scrum Master é o especialista de processos e o Gerente do Produto o especialista do negócio, cabe aos Desenvolvedores serem os especialistas técnicos do time. O Scrum chama de Desenvolvedor todo profissional que trabalha no desenvolvimento do produto ou sistema que é responsabilidade do time.

É notável que, no Time de Desenvolvimento, a denominação “desenvolvedor” não se restringe à uma função específica, mas sim a todas as disciplinas de Engenharia de Software (requisitos, design construção/codificação, testes etc.). Assim como também não há hierarquia entre estes profissionais, ou seja, cada um contribui com a sua especialidade para o sucesso do Time Ágil e para a entrega de valor.

É importante destacar que, independentemente da especialidade individual de cada membro, a responsabilidade do desenvolvimento do produto recai sobre todo o Time de Desenvolvimento. Assim, o ideal é que o time seja o mais multifuncional quanto possível, de modo a reunir todas as habilidades necessárias à entrega do Incremento.

O Time de Desenvolvimento deve seguir a priorização de Backlog e roadmap definido pelo Gerente do Produto. Uma vez que o Gerente do Produto apresente o que ele espera para a Sprint (durante a Sprint Planning), cabe ao time definir um plano de ação para entregar o esperado, quebrando os itens de backlog em tarefas e estimando o esforço necessário para a entrega. Entendendo que o esforço necessário é superior à capacidade do time, cabe negociar com o Gerente do Produto uma entrega menor ou mudança de prioridades. Somente o Time de Desenvolvimento é capaz de definir o escopo que cabe em uma Sprint e é sua responsabilidade se comprometer apenas com o que possui capacidade de entregar.

Ademais, enquanto existe apenas um Scrum Master e um Product Owner por Time Ágil, o número de Desenvolvedores é bem maior, sendo recomendado entre 3 e 9 recursos. Menos do que isso há risco de overhead de processos e mais do que isso há risco de perda de qualidade na condução do Time, uma vez que a comunicação não fluirá tão bem e as cerimônias ficarão mais complexas. Somando todos os papéis, teremos entre 5 e 11 profissionais por Time Ágil.

Por fim, o Time de Desenvolvimento trabalha seguindo o processo prescrito pelo Scrum Master: as cerimônias e os artefatos da Metodologia devem ser respeitados e qualquer atividade do processo que consuma tempo dos desenvolvedores deve ser levada em consideração nas suas estimativas na definição de escopo das Sprints, mas esse esforço jamais poderá servir como argumento para abandonar ou desviar da Metodologia.

Dependendo da capacidade do Scrum Master e/ou da complexidade/risco do projeto, processos e técnicas poderão ir além da base prevista nesta Metodologia, sendo práticas comuns a adoção de técnicas ágeis de engenharia de software como BDD, TDD, Integração Contínua, Entrega Contínua etc. Estas técnicas podem ser adicionadas ao rol do Time de Desenvolvimento conforme o Scrum Master for entendendo a necessidade e benefício delas para a entrega do resultado. Também cabe ao próprio Time de Desenvolvimento promover melhoria técnica em seus processos de engenharia, uma vez que a qualidade do software entregue é sua responsabilidade.

Em resumo, para executar sua missão o Time de Desenvolvimento deve planejar seu trabalho, definindo com o Gerente do Produto a Meta da Sprint e o que será realizado no decorrer da Sprint, para então detalhar, de forma autônoma, como esse trabalho será feito e executá-lo garantindo a qualidade do que é produzido, reportando os impedimentos ao Scrum Master e buscando entregar valor com frequência, mediante apresentação de resultados e feedback das partes interessadas.





Fases

Nossa Metodologia Ágil fornece uma estrutura base para gestão dos projetos de desenvolvimento organizada em fases claras e processos leves que visam a assegurar a aplicação dos princípios e valores ágeis – incluindo papéis, cerimônias e artefatos.

A distribuição por fases também tem por objetivo correlacionar a Metodologia com as atividades típicas do processo de Engenharia de Software aliadas à cultura integrativa DevOps, com objetivo de eliminar silos e manter o trabalho dentro de um ritmo incremental constante.

Assim, as atividades de desenvolvimento e evolução de soluções de software dentro da Metodologia Ágil do Ministério da Educação serão executadas de acordo com as seguintes fases:



Nos itens a seguir estão descritas as principais características e atividades de cada uma dessas fases.

Preparação



A Fase de Preparação contempla a identificação, o mapeamento e a modelagem das necessidades e oportunidades fornecidas pelo negócio. Nela temos a articulação da Visão do Produto e a definição do Gerente de Produto para a demanda.

A Preparação marca a etapa de materialização da necessidade em termos de identificação, mapeamento e a modelagem estratégica do produto que será detalhada na etapa seguinte para entrar nos ciclos iterativos de desenvolvimento.

Entendemos que nessa fase deverá ser produzida pelo Gerente do Produto a Declaração de Visão do Produto (Vision Statement), articulando os objetivos para o produto como um guia para todo o Time Ágil durante as fases seguintes.

A Visão do Produto é um objetivo ou necessidade de negócios de alto nível que fornece contexto, orientação, motivação e inspiração para o trabalho de desenvolvimento do produto durante todo o projeto. É o objetivo maior a ser alcançado pelo Time de Scrum, que se traduz na satisfação dos clientes. (Pichler, 2010)

A criação da Visão do Produto deve percorrer os seguintes passos básicos:

- Articular o objetivo do produto-chave (Por que o produto é necessário? Como o produto beneficiará o requisitante e o negócio? Quais as estratégias/necessidades que o produto atenderá? Quem serão os usuários?)



- Articular a visão do produto-chave (estado futuro para o produto quando finalizado, sem generalizações e/ou especificidades tecnológicas, projetos longos podem demandar a revisão da declaração ao longo do tempo)
- Validar e revisar a declaração de visão (atestar clareza, objetividade, consistência e o valor esperado)
- Finalizar e compartilhar a declaração de visão (a declaração deve ser compartilhada com todas as partes interessadas e com o Time Ágil).

Concepção



A Fase de Concepção contempla o detalhamento da Visão do Produto em termos de especificações e requisitos, no formato de Histórias de Usuário, gerando o Backlog do Produto – que será continuamente refinado e priorizado para compor os ciclos iterativos de desenvolvimento subsequentes.

A Concepção é a fase onde o Gerente do Produto produzirá o Backlog do Produto contendo uma descrição da arquitetura preliminar da solução a partir das informações fornecidas na Visão do Produto (descrição, necessidades e funcionalidades, requisitos significativos para o negócio, atores, incluindo necessidades de integração, necessidades identificadas em sistemas impactados, requisitos não funcionais inicialmente identificados, identificação das funcionalidades impactadas com necessidade de testes, identificação das funções de conversão, especificação da massa de dados em caso de necessidades de povoamento em homologação, necessidades adicionais) e os requisitos funcionais da solução, na forma de Histórias de Usuário.

Construção



A Fase de Construção se refere ao desenvolvimento do produto propriamente dito, pelo Time de Desenvolvimento, de forma contínua e incremental, a partir da lista de itens priorizados pelo Gerente do Produto (Sprint Backlog). Esses desenvolvimentos são executados em ciclos, aqui denominados Sprints.

A Construção é a fase de criação dos principais ativos e funcionalidades que compõe a solução e/ou Incremento, tais como seus ambientes, requisitos, código, componentes e serviços, testes, pacotes e documentação. As Sprints de desenvolvimento ocorrem nessa fase, contemplando suas cerimônias elementares.

Homologação



A Fase de Homologação envolve a validação pelo Gerente do Produto do resultado do ciclo de desenvolvimento e acontece previamente à implantação. Ao final da homologação deve ser formalizado o Termo de Aceite pelo Gerente de Produto das histórias entregues, de acordo com os critérios de aceitação previamente definidos.

A Homologação será realizada após cada Sprint concluída, sem paralelismo de atividades, sendo que a equipe fica focada em acompanhar a homologação antes de iniciar um novo ciclo de desenvolvimento. Somente o Gerente de Produto tem autoridade para homologar uma entrega, assim é fundamental que ele esteja mobilizado para essa fase uma vez que eventuais atrasos poderão impactar todo planejamento do projeto.





Não é recomendado que sejam acumuladas Sprints sem homologação, pois isso gera estoque de entregas não validadas, dificultando a correção dos defeitos (quanto mais tempo um defeito fica aberto tende a ser mais demorada a sua correção). Também aumenta a probabilidade de mudanças no projeto, aumentando o risco de entregas sem valor.

É importante que, independentemente da forma de homologação a ser escolhida, a Definição de Pronto da Sprint e da Release seja obedecida e esteja alinhada com o Time de Desenvolvimento e o Gerente de Produto, evitando retrabalho e otimizando a formalização do aceite e o subsequente processo de implantação.

Também é recomendável que o ambiente de homologação seja igual ao de produção, visando a evitar o surgimento de defeitos decorridos em diferenças de ambiente. Todos os defeitos levantados nesta fase devem ser registrados para avaliação da causa raiz e acompanhamento das métricas de qualidade. A equipe deve certificar se os registros estão sendo realizados de forma adequada para a devida apuração.

Implantação



A Fase de Implantação envolve a disponibilização do Incremento homologado em ambiente de produção, ou seja, a liberação do desenvolvimento para os usuários.

A Implantação encerra o ciclo de desenvolvimento, produtos implantados no ambiente de produção são transferidos para a fase de operação e sustentação. É importante destacar que também aqui somente o Gerente de Produto tem autoridade para decidir quando um Incremento homologado será disponibilizado em produção.

A execução dessa fase demanda interação do Time Ágil com outros times, tais como o de gerenciamento de serviços, garantia da qualidade (QA) e o de operações (infraestrutura).





Artefatos

Backlog do Produto



O Backlog do Produto é uma lista ordenada de tudo que deve ser necessário no produto, e é uma origem única dos requisitos para qualquer mudança a ser feita no produto. O Product Owner é responsável pelo Backlog do Produto, incluindo seu conteúdo, disponibilidade e ordenação.

O Backlog do Produto (ou Product Backlog) é a lista de requisitos que o Gerente do Produto (PO) deseja para o produto. O responsável por esse artefato é o Gerente do Produto – cabe a ele mantê-lo priorizado, visível e refinado, tendo a palavra final sobre ele dentro do Time Scrum.

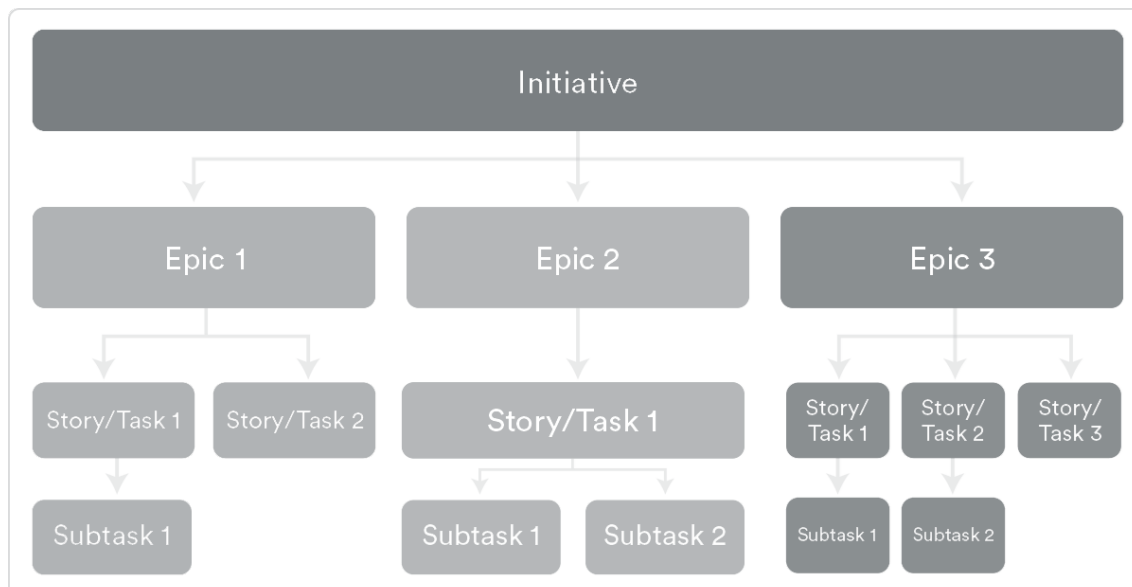
Os itens do Backlog do Produto são escritos como **Histórias de Usuário** (User Stories), que, como técnica de análise de requisitos, são uma especificação em uma ou mais sentenças na linguagem de negócio (ou linguagem cotidiana do usuário final do produto) que captura o que um usuário faz ou necessita fazer como parte de sua função de trabalho.

O Backlog do Produto deve ser visível por todas as partes interessadas e refinado para garantir que está pronto para todas as reuniões de planejamento de Sprints (Sprint Planning).

Histórias de Usuário, Épicos e Temas

Histórias de Usuário são sempre o menor pedaço independente de um produto que entregue valor, elas também podem formar blocos de construção de estruturas ágeis maiores, como épicos e temas.

Épico é a união de Histórias de Usuário que façam parte de um mesmo contexto, formando partes maiores do produto a ser entregue, que podem também ser agrupados em Temas (ou iniciativas) – que, por sua vez, é uma coleção de épicos que tem um objetivo comum.



As Histórias de Usuário devem ser independentes (não depender de outras), negociáveis (os detalhes precisam ser negociáveis, valiosa (deve agregar valor), estimável (ser clara e ao mesmo tempo ter detalhes suficientes), pequena (visando a garantia de entrega, mas ao mesmo tempo garantindo entrega de valor) e testável (capacidade de ser submetida a testes para ser válida).



Sprint Backlog



O Sprint Backlog é um conjunto de itens do Backlog do Produto selecionados para a Sprint, juntamente com o plano para entregar o incremento do produto e atingir o objetivo da Sprint. É a previsão do Time de Desenvolvimento sobre qual funcionalidade estará no próximo incremento e sobre o trabalho necessário para entregar essa funcionalidade em um incremento “Pronto”.

O Sprint Backlog é uma versão do Backlog do Produto com o escopo a ser trabalhado pelo Time de Desenvolvimento na próxima iteração. Ele é criado a partir dos itens priorizados do Backlog do Produto durante as discussões da Sprint Planning, tem seus requisitos quebrados em tarefas realizáveis pelo time e seu esforço estimado para que o time possa ter uma noção da sua velocidade e do tempo de desenvolvimento do roadmap como um todo.

Em casos de demandas concorrentes, no âmbito da mesma solução ou junto ao mesmo Gerente do Produto, o atendimento será planejado conforme a capacidade do time de desenvolvimento. Sendo que o Gerente do Produto decidirá os casos de necessidade de alteração de backlog, porém, em virtude dos custos impactados, não é desejável o cancelamento de iterações.

Uma sugestão de técnica de priorização, apresentada na sessão Práticas, é o **Método de Priorização de Requisitos MoSCoW**, que permite às partes interessadas entender melhor o impacto da definição para o escopo do produto.





Incremento



O Incremento é a soma de todos os itens do Backlog do Produto completados durante a Sprint e o valor dos incrementos de todas as Sprints anteriores. Este deve estar na condição utilizável independente do Product Owner decidir por liberá-lo realmente ou não.

O Incremento é o resultado do trabalho do Time de Desenvolvimento durante a Sprint. Ele geralmente é composto por software com potencial de ser liberado para produção, situação na qual o time terá gerado valor real para o cliente.

A avaliação do Incremento ocorre durante a Sprint Review, a fase de inspeção (homologação) do desenvolvimento da Sprint, sendo o artefato que corrobora com o empirismo do Scrum, onde apenas resultados obtidos pelo time podem ser tomados como base para julgamentos futuros.

Definição de Pronto



Quando o item do Backlog do Produto ou um Incremento é descrito como “Pronto”, todos devem entender o que o “Pronto” significa. Assim, os integrantes devem ter um entendimento compartilhado do que significa o trabalho estar completo no incremento do produto, assegurando a transparência.

A Definição de Pronto (ou *Definition of Done - DoD*), é o artefato que contém uma definição comum e clara do que "pronto" (*done*) significa para o time. Gerente do Produto, Scrum Master, o Time de Desenvolvimento e os demais stakeholders diretamente ligados ao projeto devem ter um entendimento único da situação real de um incremento dito como pronto.

Isso é necessário vez que pessoas diferentes possuem entendimentos diferentes do que “pronto” possa significar, dependendo do seu contexto dentro do projeto. Assim, é importante que o time defina este artefato de transparência logo no início da primeira Sprint, pois geralmente é algo que impacta o Pipeline de desenvolvimento das tarefas, afeta a estimativa e pode gerar conflitos.



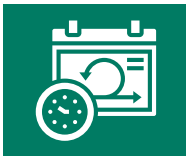


Cerimônias

Essas reuniões são chamadas de cerimônias, pois celebram o início ou o final de um determinado trabalho. Elas servem tanto para manter o controle do projeto em relação aos prazos e entregas, quanto para motivar o time a seguir trabalhando em ritmo acelerado.

No ciclo de vida do processo ágil estão previstas cinco cerimônias: Sprint Planning, Sprint, Daily Meeting, Sprint Review e Sprint Retrospective. A seguir detalhamos cada uma delas.

Sprint



A Sprint é o “coração” do processo ágil, um time-boxed durante o qual uma versão incremental potencialmente utilizável do produto é criada. Sprints tem durações coerentes em todo o esforço de desenvolvimento. Uma nova Sprint inicia imediatamente após a conclusão da Sprint anterior.

As iterações da metodologia ágil Scrum são chamadas de Sprints. Cada Sprint envolve um ciclo completo de desenvolvimento, que percorre do planejamento à entrega do incremento de software – incluindo ainda inspeção e adaptação de processos para garantir a melhoria contínua.

As Sprints podem ter a duração de 2 a 4 semanas, sendo esse o timebox do ciclo. Todas as Sprints de um projeto devem ter a mesma duração, sendo recomendado não variar o seu tamanho visto que isso gera instabilidade na velocidade do projeto (Velocity Chart).

Um conjunto de iterações (sprints) é denominada Release. Cada Release deve ter no máximo 4 (quatro) iterações (Sprints). Cada Release deve ter seu valor traduzido em um objetivo ou meta de negócio (Release Goal), correspondendo a um passo (ou marco no roadmap do produto) em direção à Visão do Produto.

O roadmap do produto corresponde a um plano de alto nível indicando como o produto evoluirá ao longo de cada Sprint até o momento futuro determinado.

O que define o tamanho da Sprint é a estabilidade do escopo, ou seja, se o projeto tem um escopo que sofre muitas alterações ao longo do tempo, é indicado a adoção de Sprints menores, se o escopo é mais estável se pode adotar Sprints maiores.

Em resumo, as principais regras aplicáveis às Sprints são:

- A Sprint possui timebox de 2 a 4 semanas;
- Releases devem ser compostos por, no máximo, 4 Sprints;
- Não deve haver quebra entre Sprints;
- Toda Sprint deve ser precedida de planejamento;
- Cada Sprint deve conter revisão para as partes interessadas e retrospectiva para inspeção e adaptação.

A Sprint contém todas as demais cerimônias do processo ágil (Sprint Planning, Daily Meeting, Sprint Review e Sprint Retrospective).

Uma Sprint pode ser cancelada antes do seu timebox terminar, porém, somente o Gerente de Produto tem a autoridade para isso – embora possa fazê-lo isso sob influência das partes interessadas, do Time de Desenvolvimento ou do Scrum Master.



Sprint Planning



O trabalho a ser realizado na Sprint é planejado na reunião de planejamento da Sprint. Este plano é criado com o trabalho colaborativo de todo o Time Scrum. A Sprint responde, na essência, ao seguinte: O que pode ser entregue como resultado do incremento da próxima Sprint? Como o trabalho necessário para entregar o incremento será realizado?

A Sprint Planning é a cerimônia de planejamento da Sprint, que ocorre no primeiro dia dela. Nesta cerimônia se traça o objetivo da Sprint (Sprint Goal³), o escopo de trabalho (Sprint Backlog) e, em linhas gerais, se discute como o trabalho será realizado. Esta cerimônia possui uma timebox de, no máximo, 1 (um) dia de planejamento para cada 30 (trinta) dias corridos de Sprint (proporção de 30:1).

A entrada desta cerimônia é o Backlog do Produto atualizado e priorizado pelo Gerente do Produto, que utilizará a primeira parte da Sprint Planning para explicar os itens de backlog ao Time de Desenvolvimento – incluindo sua priorização e esclarecimento de eventuais dúvidas.

Na segunda parte da cerimônia, cuja dinâmica e facilitação é conduzida pelo Scrum Master, o Time de Desenvolvimento trabalha na quebra de tarefas e estimativa de esforço para entregar os itens priorizados. Conforme os itens são detalhados em tarefas e as estimativas são realizadas, o time vai compondo o Sprint Backlog, ou seja, o escopo de trabalho da Sprint que é justamente a saída deste processo.

Uma vez fechado o escopo da sprint (Sprint Backlog) é possível definir o objetivo (Sprint Goal) e iniciar o seu desenvolvimento. Em regra, o escopo da Sprint não deve ser alterado sem o consentimento de todos os membros do time, uma vez que, ao defini-lo, o time se comprometeu com a sua entrega.

Mudanças de escopo que podem ser encaixadas na próxima Sprint devem ser postergadas sempre, enquanto mudanças emergenciais/críticas devem ser avaliadas e negociadas com o time caso-a-caso. Os itens de backlog planejados e não atendidos em uma release, deverão ser tratados na próxima.

Em casos de demandas concorrentes, no âmbito de uma mesma solução ou junto ao mesmo Gerente do Produto (PO), o atendimento será planejado conforme a capacidade do Time de Desenvolvimento. O Gerente do Produto (PO) decidirá os casos de necessidade de alteração de backlog, porém, em virtude dos custos impactados, não é desejável o cancelamento de iterações.

Para que essa cerimônia seja produtiva é importante que o Gerente do Produto possua o Backlog do Produto detalhado e claro. O processo de discutir e detalhar melhor o backlog é algo que deve ser constantemente feito, assim, embora esse processo base não prescreva uma cerimônia específica para isso, se entende que esse detalhamento é objeto da atividade de Refinamento de Backlog.

Em projetos complexos e/ou de alto risco é recomendável que haja validação antecipada da viabilidade técnica e da clareza negocial de cada item do Backlog do Produto. Podendo ser adotada uma cerimônia adicional denominada Refinamento de Backlog (ou Grooming) e cujo objetivo é definir provas de conceito (Spikes) que necessitem ser realizadas e apontamento de itens que devam ser mais bem especificados pelo Gerente do Produto.

³ Sprint Goal é a meta responsável por traduzir o valor produzido pela Sprint, definida e formalizada entre o Gerente do Produto e o Time de Desenvolvimento, com mediação do Scrum Master durante a reunião de Sprint Planning de cada Sprint a ser realizada.



Após a Planning, enquanto o Time de Desenvolvimento estiver trabalhando nos itens do Sprint Backlog, o Gerente do Produto estará trabalhando nos itens da próxima Sprint Planning, em um ciclo perpétuo enquanto o time existir.

Daily Meeting



A Reunião Diária é um evento time-boxed de 15 minutos, para que o Time de Desenvolvimento possa sincronizar as atividades e criar um plano para as próximas 24 horas. Esta reunião é feita para inspecionar o trabalho desde a última Reunião Diária, e prever o trabalho que deverá ser feito antes da próxima Reunião Diária.

A Daily é uma cerimônia de alinhamento. Ela possui algumas regras que garantem a sua celeridade, o seu objetivo e o aproveitamento de todos, são elas:

- Como o próprio nome sugere, ela é diária e acontece sempre no mesmo local e no mesmo horário;
- Sua duração é de, no máximo, 15 minutos (se for presencial, é feita em pé);
- Cada pessoa tem a sua vez de falar e fala para os demais membros do time, os demais ouvem;
- Não é uma “prestação de contas” e não se discute detalhes de nenhuma atividade.

Algumas regras da Daily podem soar excêntricas, como o fato dela acontecer em pé ou sempre no mesmo lugar. No entanto, a psicologia de tais regras é muito forte e elas devem ser seguidas. Um roteiro comum para execução da Daily consiste em cada membro do time responder a essas três perguntas:

- O que eu fiz desde a última Daily?
- O que eu vou fazer depois dessa Daily?
- Tem algo impedindo o meu trabalho?

A partir da resposta a essas três perguntas simples e objetivas é possível garantir que todos no time saibam se estão caminhando rumo ao que foi planejado e se algum membro precisa de ajuda (impedimentos).

Duas pessoas possuem um papel especial nesta cerimônia que são o Scrum Master e o Gerente do Produto, uma vez que eles não possuem atividades no Sprint Backlog.

A participação do Scrum Master, embora recomendada, não é obrigatória. Ele deve garantir que ela aconteça e que as regras estão sendo seguidas. Qualquer impedimento levantado nesta cerimônia deve ser comunicado a ele para que possa agir e removê-lo o mais rápido possível.

A participação do Gerente do Produto também não é obrigatória na Daily, embora ela possa ser útil para identificar se o time está seguindo o que foi planejado e corrigir qualquer desvio de rumo depois dela.





Sprint Review



A Revisão da Sprint é executada no final da Sprint para inspecionar o incremento e adaptar o Backlog do Produto se necessário. Durante a reunião de Revisão da Sprint o Time Scrum e as partes interessadas colaboram sobre o que foi feito na Sprint. Com base nisso e em qualquer mudança no Backlog do Produto durante a Sprint, os participantes colaboram nas próximas coisas que podem ser feitas para otimizar valor.

A Sprint Review (Revisão da Sprint) é a cerimônia, que acontece no último ciclo de desenvolvimento, onde é apresentado ao Gerente do Produto e às demais partes interessadas o que foi desenvolvido e se obtém o feedback para possíveis ajustes ou entrega do incremento.

A primeira finalidade da Sprint Review é realizar uma entrega que gere valor ao cliente, vez que o objetivo principal de toda Sprint é gerar um Incremento de software com potencial⁴ de ser colocado em produção. Com esse mindset, a Review é o momento de demonstrar a entrega de valor ao cliente, com um intervalo entre as entregas de, no máximo, um mês.

A segunda finalidade é receber *feedback* das partes interessadas e, caso necessário, ajustar o rumo do projeto, garantindo a comunicação e diminuindo os riscos.

Não existe um formato específico para esta cerimônia, em regra o que não pode faltar em nenhuma Sprint Review é a apresentação do trabalho planejado x trabalho realizado e quais serão os próximos passos do Time de Desenvolvimento. Uma lista do que pode ser demonstrado nesta cerimônia inclui:

- mostrar itens do Sprint Backlog que estão prontos (“done”) e os que não foram finalizados;
- tratar dos impedimentos enfrentados que eventualmente possam ter resultado em não finalização de itens do Sprint Backlog;
- demonstrar o funcionamento e tirar dúvidas sobre itens finalizados;
- tratar do objetivo da próxima sprint, apresentando um esboço de Sprint Backlog;
- rever orçamento, datas e prazos caso necessário;
- tratar de eventuais mudanças e ajustes do roadmap.

Em termos de duração, esta cerimônia não deve possuir mais do que quatro horas para cada trinta dias de Sprint – que será considerada bem-sucedida quando o Gerente do Produto e as demais partes interessadas concordarem que o trabalho entregue foi satisfatório para esta iteração.

⁴ Diz-se “potencial” pois a decisão de colocar ou não em produção pertence exclusivamente ao Gerente do Produto – o que ultrapassa a definição de pronto.





Sprint Retrospective



A Retrospectiva da Sprint é uma oportunidade para o Time Scrum inspecionar a si próprio e criar um plano para melhorias a serem aplicadas na próxima Sprint.

A Sprint Retrospective (ou Retrospectiva da Sprint) é a cerimônia que marca o fechamento de uma Sprint e na qual participa todo o time do projeto. Seu objetivo é analisar tudo o que aconteceu na iteração que está encerrando e propor melhorias para a próxima.

Existem várias formas de conduzir esta dinâmica, mas todas possuem como objetivo levantar uma espécie de matriz SWOT da equipe gerando um plano de ação de melhoria contínua que deverá ser executado concomitantemente à sprint seguinte. Analisam-se processos, ferramentas e relacionamentos, para dizer o mínimo e é importante que exista um ambiente seguro e sincero dentro do time para que de fato possamos resolver todos os problemas encontrados.

Tendo sido elencados pontos de melhoria, eles devem ser o mais concreto possível, ou seja, tenham data para acontecer, responsável por executar e que sejam acionáveis, fugindo do abstrato. Itens abstratos tendem a não gerar melhorias tangíveis.

Outro ponto importante é não definir ações de melhoria que estejam fora de controle do time, que não há como serem realizadas. Nesses casos, o time deve focar em trabalhar contingências para mitigar ocorrências cuja solução não esteja sob seu controle.

Enquanto essa dinâmica produz muitos resultados nas primeiras vezes em que é realizada, com o tempo ela pode tornar-se desgastante e repetitiva, perdendo seu efeito. Sendo assim, encoraja-se os Scrum Masters a buscarem sempre novas e instigantes formas de extrair estas informações do grupo para que eles continuem melhorando sempre, como pregam os métodos ágeis de desenvolvimento.





Práticas

Timeboxing



O timeboxing aloca uma unidade de tempo fixa e máxima para cada atividade/cerimônia dentro do processo ágil (timebox).

O timeboxing é aplicado como técnica para gerenciar escopo e risco em projetos ágeis. Quanto ao escopo, o timeboxing auxilia a definição de quais entregas podem ser feitas dentro do prazo (que é fixo) – assim também contribuiu com a necessidade de priorização das entregas, objetivando o foco em entregar o que é mais importante primeiro.

Em gestão de riscos, o timeboxing permite identificar relações tempo/tarefa incertas, que podem facilmente ultrapassar seu prazo. Assim, contribui no entendimento do impacto dos fatores de risco do projeto orientando ações de ajuste como redução do escopo ou aumento do custo – considerando que o aspecto tempo é uma restrição fixa.

Método de Priorização de Requisitos MoSCoW



Método de priorização de requisitos que permite às partes interessadas entender melhor o impacto da definição para o escopo do produto (M – must have, S – should have, C – could have e W – won't have).

Todos os requisitos são importantes, no entanto, para entregar os maiores e mais imediatos benefícios de negócios com antecedência, os requisitos devem ser priorizados. O significado simples em inglês das categorias de priorização tem valor para fazer com que os clientes entendam melhor o impacto da definição de uma prioridade, em comparação com alternativas como alta, média e baixa. As categorias de priorização são normalmente entendidas como:

- Deve ter (must have): os requisitos rotulados como Must have são críticos para o cronograma de entrega atual para que seja um sucesso. Se um requisito dessa categoria não for incluído, a entrega do projeto deve ser considerada uma falha (observação: os requisitos podem ser rebaixados de Must have, mediante acordo com todas as partes interessadas relevantes; por exemplo, quando novos requisitos são considerados mais importantes). Must também pode ser considerado um acrônimo para o subconjunto mínimo utilizável (ou Produto Mínimo viável – MVP);
- Deveria ter (should have): os requisitos rotulados como Should have são importantes, mas não criticamente necessários para entrega no prazo de entrega atual. Embora esses requisitos possam ser tão importantes quanto aqueles classificados como Must have, eles geralmente não são tão críticos em termos de tempo ou pode haver outra maneira de satisfazer o requisito para que ele possa ser retido até um prazo de entrega futuro (próximas releases);
- Poderia ter (could have): os requisitos rotulados como Could have são desejáveis, mas não são necessários, ou seja, não impactam o objetivo do produto (exemplo: requisitos que podem melhorar a experiência do usuário ou a satisfação do cliente por um pequeno custo de desenvolvimento). Eles normalmente poderão ser incluídos futuramente se o tempo e os recursos permitirem;
- Não terá (won't have): os requisitos rotulados como Won't have são acordados pelas partes interessadas como os itens menos críticos, de menor retorno ou não apropriados no momento. Como resultado, esses requisitos não serão planejados na programação para o próximo prazo de entrega – embora não sejam necessariamente descartados e podem ser reconsiderados para inclusão em um timebox posterior. Ocasionalmente, esses requisitos também podem ser entendidos como “gostaria de ter(Nota:



ocasionalmente o termo. Esses requisitos também podem ser entendidos como 'quero ter, mas não desta vez'.

Aliando essa técnica à do Timeboxing temos que os desenvolvedores tentarão inicialmente entregar todos os requisitos Must have , Should have e Could have , mas os requisitos Should e Could serão os primeiros a serem removidos se a escala de tempo de entrega parecer ameaçada.

Refinamento do Backlog do Produto



O Refinamento do Backlog do Produto (Product Backlog Refinement) é o ato de quebrar e definir ainda mais os elementos do Backlog do Produto em elementos menores e mais precisos.

Essa prática prescreve, basicamente, o constante aperfeiçoamento do entendimento do backlog do produto objetivando organizar, limpar e ordenar o Backlog do Produto – o que inclui:

- Remover itens desnecessários ou que não façam mais sentido;
- Adicionar itens novos, que foram descobertos durante a sprint anterior pelo contato com os stakeholders;
- Quebrar itens épicos em histórias menores;
- Detalhar melhor itens já listados e necessários;
- Priorizar os itens em ordem do que deve ser desenvolvido primeiramente;
- Adaptar estimativas;
- Incluir e revisar critérios de entrega;

Sprint Burn-down



Ferramenta de medição visual dessa metodologia ágil que apresenta o trabalho concluído por dia em relação à taxa de conclusão projetada para o projeto.

O Burndown é uma ferramenta de medição visual dessa metodologia ágil que apresenta o trabalho concluído por dia em relação à taxa de conclusão projetada para o projeto. Ele é considerado um dos mais úteis para monitorar o progresso de um time ágil. O gráfico representa a quantidade de trabalho que falta ser feito no eixo vertical (y) versus o tempo no eixo horizontal (x).

Velocity Chart



O Gráfico de Velocidade é a ferramenta utilizada para medir a quantidade média de trabalho que uma equipe pode concluir em uma Sprint.

Essa métrica é utilizada para medir a capacidade de entrega da equipe. É obtida a partir da entrega dos Sprints, de 1 a 3 ciclos similares e envolve pontos como tecnologia, perfil dos membros, dificuldade do projeto e conhecimento do projeto. Velocidade mede a quantidade média de trabalho que uma equipe pode concluir em uma Sprint. Ela pode ser medida na forma de pontos da história ou horas e é muito útil para a previsão.

Para o cálculo da velocidade, é importante que se utilize o Fator de Foco, que se trata de um parâmetro que é multiplicado pela capacidade de produção esperada da equipe:

Exemplo de Cálculo de Velocity Chart

$$\text{Velocidade} = \text{Capacidade de Produção} \times \text{Fator de Foco}$$



Exemplo de Cálculo de Velocity Chart

- Considerando: time de 4 pessoas em tempo integral com Sprint de 2 semanas
- Capacidade de produção: 10 dias x 4 recursos = 40 hh/dia
- Fator de Foco: de 50% a 70% para equipes novas

$$\text{Velocidade} = 40 \text{ hh/dia} \times 60\% = 24 \text{ story points}$$

Planning Poker



Planning Poker é uma técnica gameficada para definir estimativas de esforço de desenvolvimento de itens do Sprint Backlog de forma consensual dentro do Time de Desenvolvimento.

É uma estratégia gameficada usada em projetos ágeis que busca uma estimativa consensual da equipe. Com essa ferramenta, é possível receber auxílio no planejamento, além disso, ela pode ser usada em um projeto Scrum para estimar os itens do Product Backlog pelo próprio Time de Desenvolvimento.

Acredita-se que as melhores estimativas são feitas pelas pessoas que realmente realizam o trabalho sendo assim, as estimativas no Planning Poker devem ser realizadas somente pela Equipe de Desenvolvimento.

Gestão de Equipes



Gestão de equipes envolve as atividades de requisição, alocação, movimentação e devolução de recursos – assim como o gerenciamento de tarefas e cargas de trabalho dentro do Time.

As requisições de recursos para composição dos times é tarefa do Fiscal Requisitante do Contrato junto ao Gestor do Contrato – que deve considerar a efetiva necessidade do perfil e as disponibilidades contratuais e orçamentárias. A gestão de equipes quanto ao controle de tarefas e cargas de trabalho dentro do próprio Time é de responsabilidade do Scrum Master.





APÊNDICE 01 Matriz de atribuições e responsabilidades

Legenda

■ Responsável ■ Apoiar/participa ■ Informado

Atividade	Fase	Gerente do Produto (PO)	Scrum Master	Time de Desenvolvimento	Time de Infraestrutura
Definir a necessidade/opportunidade	Preparação	■			
Definir a visão do produto	Preparação	■	■		
↳ Declaração de visão do produto	Preparação	■	■		
Definir e gerir o backlog do produto	Preparação	■	■		
↳ Priorizar o backlog	Preparação	■	■		
Especificar os requisitos da solução	Concepção	■	■	■	
↳ Aprovar os requisitos da solução	Concepção	■	■	■	
Planejar desenvolvimento	Construção	■	■	■	
↳ Selecionar os itens de backlog da iteração	Construção	■	■	■	
↳ Definir meta de negócio da interação	Construção	■	■	■	
↳ Estimar o esforço da iteração	Construção	■	■	■	
↳ Distribuir os itens de backlog da iteração	Construção	■	■	■	
↳ Definir o time box (objetivo/duração)	Construção	■	■	■	
↳ Construir o incremento	Construção	■	■	■	
Planejar homologação	Homologação	■	■	■	■
↳ Preparar o ambiente para homologação	Homologação	■	■	■	■
↳ Disponibilizar o incremento em ambiente de validação	Homologação	■	■	■	
↳ Demonstrar os itens trabalhados na iteração	Homologação	■	■	■	
↳ Homologar o produto/incremento	Homologação	■	■	■	
↳ Corrigir os erros encontrados em homologação	Homologação	■	■	■	
Planejar implantação	Implantação	■	■	■	■
↳ Preparar ambiente de produção	Implantação	■	■	■	■
↳ Preparar dados para produção	Implantação	■	■	■	■
↳ Validar ambiente de produção	Implantação	■	■	■	■
↳ Autorizar disponibilização do produto/incremento em produção	Implantação	■	■	■	■
↳ Disponibilizar produto/incremento em produção	Implantação	■	■	■	■
↳ Registrar ativos no ciclo de gerenciamento de configuração	Implantação	■	■	■	■



APÊNDICE 02 Classificação das demandas

Demanda	Descrição	Aplicabilidade Processo Ágil
Desenvolvimento de software	Compreende a criação de um novo produto de software a partir de uma nova necessidade do negócio ou a reconstrução (reengenharia) de soluções a partir de um produto legado	■
Evolução de software	Compreende as demandas de incremento de funcionalidades a um sistema. Reflete mudanças em requisitos funcionais da aplicação, ou seja, inclusão, alteração ou exclusão de regras ou funcionalidades em aplicações já existentes	■
Adaptação de software	Compreende as demandas de adaptação de um sistema a novas regras de negócio, melhorias de segurança e/ou a novas tecnologias, visando manter os sistemas alinhados aos processos de negócio e ao ambiente tecnológico seguro e disponível.	■





APÊNDICE 03 Papéis gerenciais complementares

Papel	Função	Atribuições
Gestão de Desenvolvimento de Soluções de Software	Coordenador(a)-Geral	Coordenar o desenvolvimento e a sustentação de soluções de software, de forma aderente à estratégia da unidade de TIC e de acordo com as necessidades do negócio.
	Coordenador(a)	Gerenciamento de manutenção de softwares (Sustentação e Evolução), desenvolvimento de soluções, integração de dados e tratamento de dados corporativos e compartilhados, monitoramento de serviços e de contratos para prestação de serviço, gerenciamento de sites, portais e mobile – de forma segmentada.
	Equipe de Fiscalização de Contrato	Execução de atividades privativas de servidores, onde devem ser estabelecidos e executados de modo consistente os seguintes processos: Planejamento Técnico e Operacional; Execução da Supervisão de Sistemas e Serviços; Gestão de Desempenho de Sistemas, Aplicações, Sites e Portais; Controle de Integração de Dados; Fiscalização Técnica de Contratos.
	Equipe de Arquitetura de Soluções de Software	Definir e gerenciar padrões de engenharia, arquitetura, segurança e configuração de softwares e demais soluções de TIC, incluindo os seguintes processos: Gerenciamento de Padrões de Engenharia e Arquitetura, Gerenciamento de Segurança e Configuração, Operações de Teste e Medição de Qualidade de Softwares.
Relacionamento com o Negócio	Especialistas em Relacionamento com o Negócio	Executa as atividades rotineiras de apoio de nível gerencial à execução de atividades rotineiras de relacionamento com o negócio descritas no Catálogo de Serviços do Contrato ao qual estão vinculados. No contexto do Processo Ágil atuam também no papel de Scrum Master .
	Analista de Negócio	Executa as atividades rotineiras de apoio de nível técnico à execução de atividades rotineiras de relacionamento com o negócio descritas no Catálogo de Serviços do Contrato ao qual estão vinculados.
Gestão de Infraestrutura de TIC	Coordenador(a)-Geral	Orientar, acompanhar e avaliar a execução das atividades de gestão de infraestrutura de TIC; coordenar a execução da política de segurança de Tecnologia da Informação, no âmbito do Ministério; coordenar, propor e manter políticas, diretrizes e critérios referentes a banco de dados e ambientes tecnológicos; gerir a esteira de integração e entrega contínua; e avaliar e propor a aquisição de novas tecnologias, novos produtos e serviços de TIC.
Escritório de Projeto	Project Management Officer	Coordena a equipe de gerenciamento de projetos, acompanha e reporta progressos, riscos, custos e impedimentos em projetos. Implementa mudanças na equipe e nos processos quando necessário. Atua com base nas definições da Metodologia de Gestão de Projetos (MGP-Mec). Projetos de alto risco, de alto impacto, estratégicos e/ou prioritários são elegíveis para gerenciamento pelo Escritório de Projetos.
	Gerentes de Projeto	Atua como interface entre o Gerente do Produto, o Time Ágil e a STIC, abordando aspectos que não são tratados pelo Scrum Master, tais como: orçamento do projeto, gestão de mudanças, monitoramento de status, governança, comunicação e gestão de riscos. Projetos de alto risco, alto impacto (entregas sensíveis), estratégicos e/ou prioritários são elegíveis para acompanhamento por um Gerente de Projeto.



APÊNDICE 04 Formatos e meios padronizados de entrega

Produto / Artefato	Meio de Entrega				Formato
	Ferramenta	Meio digital	Templates	Outro	
Visão do Produto	■	■			GitLab MEC
Termo de Abertura do Projeto	■		■		GitLab MEC
Backlog do Produto	■	■			GitLab MEC
Histórias de Usuário	■	■			HU[n° sequencial] – [Nome HU]
Sprint Backlog	■	■			GitLab MEC
Critérios de Aceitação (DOD)	■	■			GitLab MEC
Documento de Arquitetura de Software		■			MDS_Modelo_Arquitetura_Sigla
Código-fonte e/ou scripts	■	■			GitLab MEC
Modelo de dados com dicionário de dados	■	■	■		GitLab MEC
Regras de negócio e mensagens	■	■			RN[n° sequencial] – [Nome RN] MSG[n° sequencial] – [nome MSG]
Ajuda do Sistema		■			Em definição
Documento de Entrega da Sprint/Release	■	■			MDS_Sprint[n°]_Entrega_Sigla
Termo de Aceite do Incremento	■	■	■		GitLab MEC
Requisitos de implantação	■	■	■		Readme.md



APÊNDICE 05 Padrões técnicos de desenvolvimento

Tipo de Padrão	Descrição	Acesso
Padrões de Interface	Para frontend será utilizado o padrão de design do Governo Federal (Design System - https://gov.br/design).	https://gov.br/design
Padrões de Administração de Dados	A Metodologia de Administração de Dados define padrões de administração de dados são que devem ser aplicados em todos os projetos de desenvolvimento de software [https://portalstic.mec.gov.br/manuais-tutoriais]	Portal STIC
Padrões de Gerenciamento de Configuração	Os padrões de gerenciamento de configuração estão definidos no Processo de Gerenciamento de Configuração	Portal STIC
Modelo de Arquitetura de Software – JAVA	Define um modelo de arquitetura para soluções de software em linguagem JAVA	Portal STIC
Modelo de Arquitetura de Software - PHP	Define um modelo de arquitetura para soluções de software em linguagem PHP	Portal STIC
Aplicativos Móveis	Guia de Requisitos de Segurança e Privacidade para Aplicativos Móveis	Portal STIC
Padrões gerais de arquitetura e ambientes	Conjunto de padrões e recomendações para arquitetura e ambientes de desenvolvimento de software [https://portalstic.mec.gov.br/informacoes-tecnicas]	Portal STIC
Projetos de Business Intelligence	A Metodologia de Projetos de Business Intelligence descreve um conjunto de práticas, padrões e recomendações [https://portalstic.mec.gov.br/manuais-tutoriais]	Portal STIC
Esteira DevSecOps	Descreve procedimentos, padrões e recomendações para operação da esteira DevSecOps no ambiente de TIC do MEC [https://portalstic.mec.gov.br/metodologias-e-processos]	Portal STIC
Segurança da Informação	A Política de Segurança da Informação e Comunicações - POSIC objetiva instituir diretrizes estratégicas, responsabilidades e competências, visando assegurar a integridade, confidencialidade, disponibilidade e autenticidade das informações custodiadas e de propriedade do MEC	Portal STIC

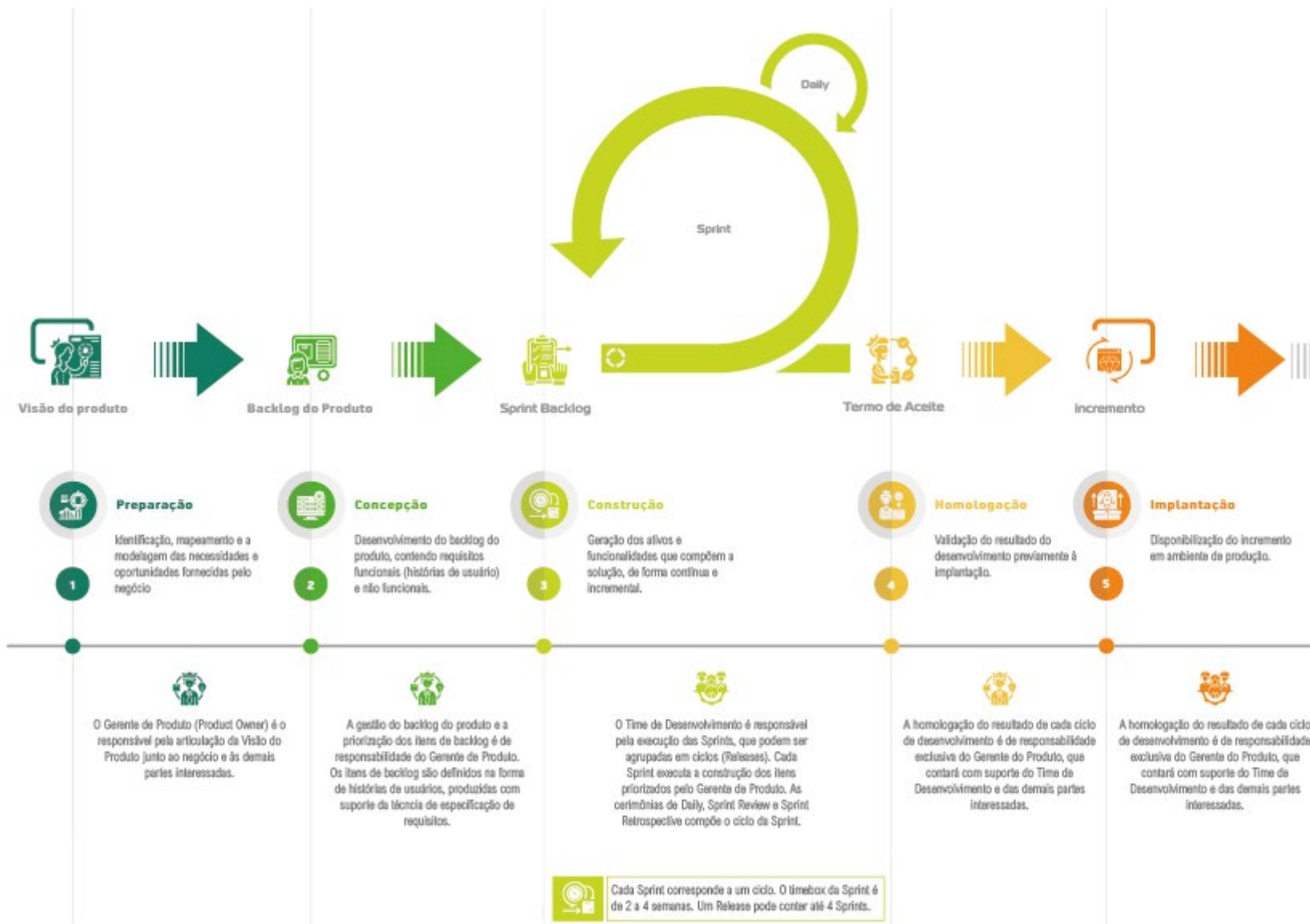


APÊNDICE 06 Ferramentas de apoio ao desenvolvimento

Escopo	Descrição	Acesso
Gestão de demandas / necessidades das áreas de negócio	As necessidades de desenvolvimento de soluções de software devem ser registradas pela equipe de Apoio ao Relacionamento com o Negócio no Funil de Demandas	Em definição
Versionamento e controle de versão	A ferramenta padrão para controle de versionamento de código é o GIT. Em projetos legados é utilizada a ferramenta SVN.	Em definição
Testes automatizados e avaliação de qualidade de código	O sistema automatizado de teste e avaliação da qualidade do código tem como finalidade a detecção de erros, vulnerabilidades e indícios de problemas mais profundo no código através da inspeção contínua de código no repositório do projeto.	Em definição
Avaliação de segurança do código	O sistema automatizado de teste e avaliação da segurança do código tem como finalidade a verificação de possíveis falhas na segurança do código através da análise de vulnerabilidade de aplicação	Em definição

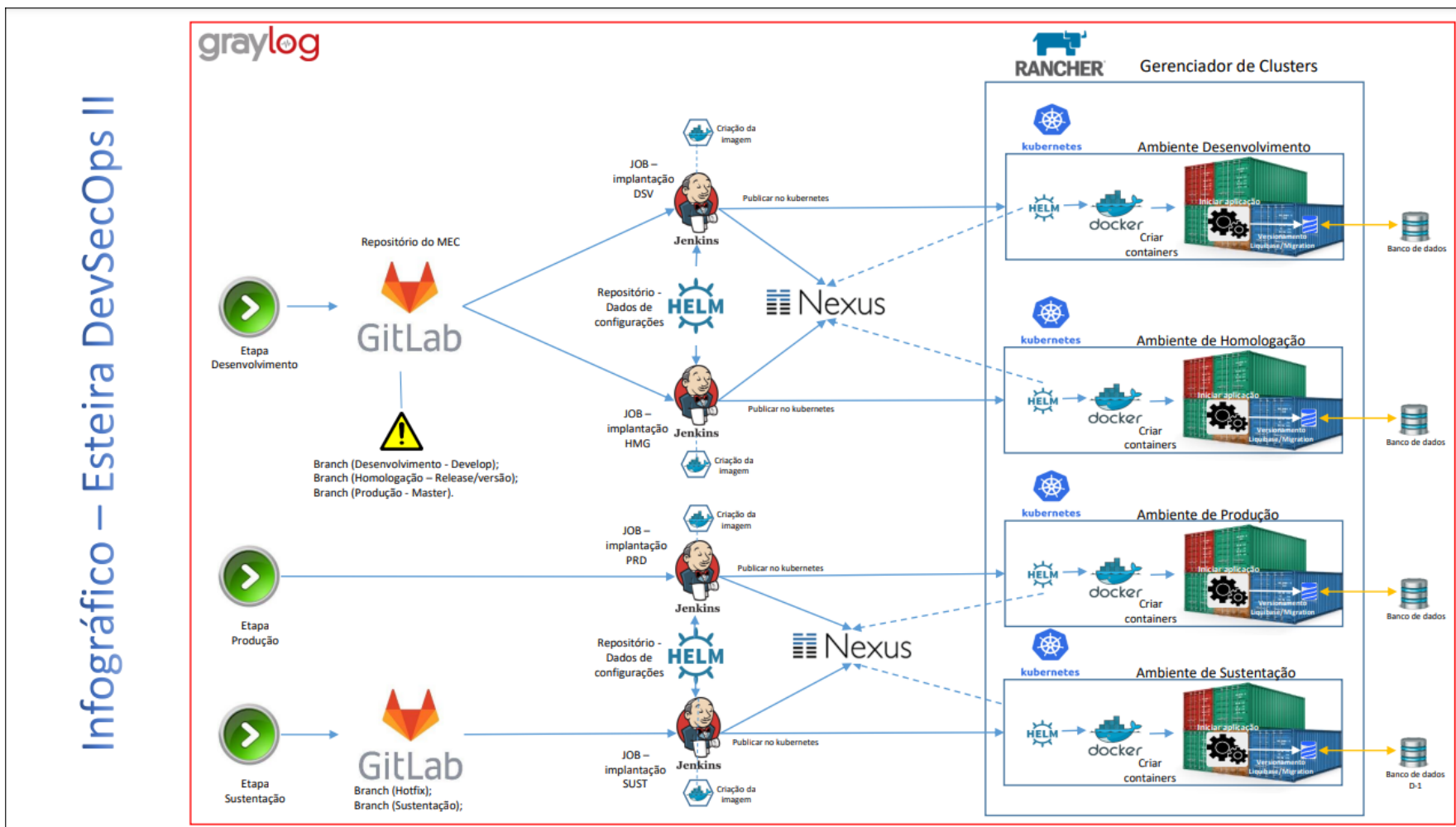


APÊNDICE 07 Mapa de processo da Metodologia Ágil MEC





APÊNDICE 08 Esteira DevSecOps



Detalhamento: <https://portalstic.mec.gov.br/metodologias-e-processos/46-infraestrutura-e-seguranca-da-informacao/129-devsecops>

VERSIONAMENTO:

Metodologia Ágil de Desenvolvimento e Sustentação de Software do Ministério da Educação

VERSÃO:

1.0

1.1

DATA DE APROVAÇÃO:

Maio/2022

dezembro/2022

MINISTÉRIO DA
EDUCAÇÃO



Secretaria Executiva
Subsecretaria de Tecnologia da Informação e Comunicação

Esplanada dos Ministérios, Bloco L, Anexo II

Brasília/DF

portalstic.mec.gov.br