



MDS – METODOLOGIA DE DESENVOLVIMENTO DE SOFTWARE - CAPES

Versão: 4.2



Metodologia de Desenvolvimento de Sistemas - MDS

Versão: 4.2

Data Emissão:

13/12/2021

Histórico de Revisões

Data	Versão	Descrição	Responsável
21/12/2017	4.0	Criação da metodologia ágil	Grupo de Trabalho instituído pela Portaria DTI Nº 1
04/07/2019	4.0.1	Atualização da metodologia ágil	Grupo de Trabalho da MDS
09/01/2020	4.0.2	Alteração conforme documento Relatório de Revisão da MDS criado pela RNP que trata sobre Segurança da Informação	Raniere Nogueira
02/07/2020	4.0.3	Inclusão de novos perfis e revisão do macroprocesso	Weudes de Souza Evangelista
06/07/2020	4.0.4	Inclusão dos artefatos mínimos de entrega e guias de referências referente a cada etapa do macroprocesso	Weudes de Souza Evangelista
24/07/2020	4.1	Revisão e entrega da versão final	Fabiano Moreira de Aguiar
30/11/2021	4.1.1	Alteração de perfis relacionados ao apoio à Gestão de Software	Aline Lourenconi Gazola
13/12/2021	4.2	Revisão e entrega da versão final	Fabiano Moreira de Aguiar

SUMÁRIO

1. INTRODUÇÃO.....	5
2. METODOLOGIA ÁGIL	5
3. SCRUM.....	5
3.1 Objetivos da Metodologia Scrum	6
4. KANBAN.....	6
5. PAPÉIS.....	7
5.1 Time do Produto.....	7
5.1.1 Product Owner (PO).....	8
5.1.2 Time de Desenvolvimento	8
5.1.2.1 Gerente de Produto	8
5.1.2.2 Analista Programador.....	9
5.1.2.3 Apoio ao PO.....	9
5.2 Designer UX e UI.....	9
5.3 Arquiteto de Soluções	9
5.4 Arquiteto de Automação	9
5.5 Administrador de Dados.....	9
5.6 Analisa de Infraestrutura.....	10
5.7 Gerente PMO	10
6. MACROPROCESSO DE DESENVOLVIMENTO	10
6.1 Solicitar Sustentação	12
6.2 Realizar Sustentação	13
6.3 Realizar reunião de alinhamento.....	14
6.4 Solicitar Novo Produto	14



6.5	Solicitar Internalização	15
6.6	Analisar Solicitação.....	15
6.7	Executar Internalização	16
6.8	Realizar Lean Inception	17
6.9	Definir processo utilizado.....	18
6.10	Executar Kanban	19
6.11	Executar Scrum e Kanban	20
6.12	Executar Ágil em Escala	22
6.13	Preparar Backlog.....	23
6.1	Planejar Release	24
6.2	Homologar Aplicação	25
6.3	Implantar em Produção.....	26
7.	<i>FERRAMENTAS.....</i>	<i>28</i>
8.	<i>REFERÊNCIAS</i>	<i>29</i>
9.	<i>ANEXO I - Definições, Acrônimos e Abreviações</i>	<i>30</i>
10.	<i>ANEXO II – Produtos de Trabalho.....</i>	<i>33</i>



1. INTRODUÇÃO

O objetivo desse documento é definir e detalhar a proposta de metodologia ágil a ser aplicada em desenvolvimento e manutenção de softwares para a CAPES – Coordenação de Aperfeiçoamento de Pessoal de Nível Superior.

Este guia é a principal referência do assunto para Gerentes de Produto, Product Owners, Apoio aos PO's, Analistas Programadores, Arquitetos, Designers UX e UI, Gerentes PMO que atuam na DTI da CAPES. Pode ainda apoiar o trabalho de outros profissionais da DTI que tenham envolvimento com o assunto.

2. METODOLOGIA ÁGIL

A metodologia ágil é uma filosofia que defende a satisfação do cliente e a entrega incremental prévia; equipes de projetos pequenas e altamente motivadas; métodos informais; artefatos de engenharia de software mínimos e, acima de tudo, simplicidade no desenvolvimento geral. Os princípios de desenvolvimento priorizam a entrega mais que a análise e projeto (embora essas atividades não sejam desencorajadas); também priorizam a comunicação ativa e contínua entre desenvolvedores e clientes. (Pressman, 2011).

Dentro desta metodologia existem frameworks que tem por objetivo buscar atingir as metas por meio de processos contínuos e fáceis de medir, integrando os participantes e as ferramentas de forma colaborativa, lembrando-se sempre das necessidades do cliente e trocando feedbacks, para que todos possam andar sempre no mesmo ritmo.

3. SCRUM

Scrum é um *framework* testado e consolidado pelo mercado para estruturar equipes de desenvolvimento Ágil. Suas principais características são:

- Montar times multidisciplinares;
- Trabalhar como equipe;
- Definir o “dono do produto”: o guardião dos interesses do usuário final;
- Criar uma lista de tarefas que devem ser cumpridas (*backlog*);
- Determinar um “*Scrum Master*”;
- Adotar uma rotina de feedbacks regulares e diários; e
- Organizar-se por “*sprints*” de trabalho, com tempo determinado para acabar.

As reuniões diárias são chamadas de “*Daily*” e ao final de cada *Sprint*, é organizada uma reunião de retrospectiva, para se analisar e discutir tudo que foi feito e guardar aprendizados para os próximos projetos.

3.1 Objetivos da Metodologia Scrum

Conforme já citado o *Scrum* será o *framework* a ser adotado no projeto de sistemas da CAPES. Desta forma, será definido de forma clara “quem” faz “o que”, “quando”, “como”, e até mesmo “onde”, para todos que estejam envolvidos diretamente ou não com o desenvolvimento dos sistemas. Além disso, um conjunto de padrões já existentes dão apoio nesta disciplina o que de certo modo evita a subjetividade na abordagem garantindo o padrão de qualidade. Portanto alguns critérios importantes devem ser seguidos, dentre eles, podemos citar:

- O desenvolvimento do software deve ser alinhado às necessidades do negócio;
- Haver entrega do maior valor possível para o negócio com qualidade;
- O processo de trabalho da Equipe do Projeto ser adaptado continuamente;
- As entregas de software serem frequentes;
- Os riscos serem identificados e tratados eficientemente;
- A Equipe do Projeto trabalhar colaborativamente;
- Os envolvidos terem visibilidade do progresso do projeto;
- Atingir as ações do PDTIC; e
- A documentação do software ser concisa e atualizada.

4. KANBAN

O *Kanban* é uma forma de organizar os projetos *Ágil*, por meio de melhorias incrementais, isto é, pequenos avanços graduais de performance ou funcionalidades. Para a maioria dos especialistas, o *Kanban* é um método de gestão de mudanças, que dá ênfase nos seguintes princípios:

- Visualizar o trabalho em andamento;
- Visualizar cada passo em sua cadeia de valor, do conceito geral até o software que se possa disponibilizar;
- Limitar o Trabalho em Progresso (WIP – Work in Progress), restringindo o total o trabalho em cada estágio definido no modelo;
- Tornar explícitas e claras as políticas que estão sendo seguidas;
- Medir e gerenciar o fluxo, para tomar decisões fundamentadas, além de visualizar as consequências das mesmas;
- Identificar oportunidades de melhorias, criando uma cultura *Kainzen*, na qual a melhoria contínua é responsabilidade de todos;
- Estabelece a filosofia de que começar com o que está fazendo agora; e



- Respeitar o processo atual, com seus papéis, responsabilidades e cargos.

Segundo (Jesper Boeg) o *Kanban* (com K maiúsculo) é o método de mudança evolucionária que utiliza um sistema kanban (com k minúsculo), além de visualização e outras ferramentas, para catalisar a introdução das ideias *Lean* nas áreas de desenvolvimento e operações de TI.

Todo trabalho no *Kanban* se torna visível. Os limites WIP são estabelecidos para dar transparência na produtividade e nos indicadores. Com isso, as políticas se tornam explícitas e o fluxo passa a ser medido.

A razão pela qual o sistema *Kanban* é referido como “Sistema Puxado Kanban” (*Kanban Pull System*) é que, ao visualizar o fluxo e estabelecer os limites do WIP, garantimos que nunca se pode introduzir mais trabalho no sistema que a capacidade do sistema de processar esse trabalho. É importante frisar que não há regras quanto à aparência específica do quadro.

O *Kanban* é excelente em situações em que estruturas organizacionais inibem mudanças radicais, desta forma, e fundamental o respeito ao processo atual, seus papéis, responsabilidades e cargos. Nenhum dos princípios do *Kanban* restringe o uso do *Scrum*. O *Kanban* funciona como um agente de mudanças, e os princípios do *Scrum* devem, portanto, ser usados apenas nos casos em que ajudam a otimizar o fluxo de trabalho. Desta forma, nada impede de se começar com o *Scrum* e utilizar o *Kanban* para impulsionar futuras mudanças.

O *Kanban* é antes de tudo um catalisador para a condução de mudanças e precisa de um ponto de partida. Assim, apesar de a maioria dos projetos poder se beneficiar do uso do *kanban*, o *Kanban* não é um substituto do *Scrum*, que é, na maioria dos casos, um ponto de partida perfeito para a adoção do *kanban*.

O importante é entender que, independente do seu fluxo de trabalho, sempre haverá uma maneira mais adequada de empregar estas metodologias ou outras ao seu negócio. Desta forma, a maioria dos projetos, ágeis ou não, podem ser beneficiados pelo uso dos princípios do *Kanban*, para impulsionar as mudanças e a melhoria contínua.

5. PAPÉIS

Aqui se elencam os papéis específicos de uma equipe ágil, de parceiros diretos em atividades essenciais e as habilidades necessárias de cada um deles para a MDS Ágil.

5.1 Time do Produto

No time do produto cada um é responsável por uma atividade não existem tarefas definidas por perfil. As pessoas trabalham juntas para completar com sucesso as entregas. É importante para

	Metodologia de Desenvolvimento de Sistemas - MDS	Versão: 4.2
		Data Emissão: 13/12/2021

o time que ele seja Multidisciplinar, Auto Organizado, tendo um tamanho certo e seja colaborativo. Fazem parte deste time os seguintes profissionais:

5.1.1 Product Owner (PO)

Profissional que representante a área de negócio com conhecimento suficiente para definir e priorizar requisitos do negócio e responder aos questionamentos da equipe de desenvolvimento. É o representante do cliente e sua atuação tem por finalidade garantir a entrega do valor esperado através do produto final.

5.1.2 Time de Desenvolvimento

O time de desenvolvimento é composto por todas as pessoas responsáveis por transformar o *Product Backlog* em incrementos de funcionalidades que possam ser entregues ao cliente e que agregam valor.

Os integrantes do time são membros interdisciplinares, o qual devem possuir conhecimentos necessários para desenvolver valor ao trabalho. Estes podem possuir conhecimentos especializados, tais como: controle de qualidade, programação, arquitetura, análise, banco de dados, capacidade de aplicação dos padrões e boas práticas de desenvolvimento seguro de software ou outros, mas o mais importante é a habilidade de pegar uma funcionalidade e transformá-la em um produto viável.

Dentre os principais papéis de um Time de Desenvolvimento, devemos citar:

- Fazer as estimativas necessárias;
- Definir as tarefas que serão realizadas;
- Desenvolver o produto;
- Realizar repasses de conhecimento para as equipes de suporte e sustentação;
- Aplicar padrões de boas práticas de desenvolvimento seguro de software;
- Garantir a qualidade do produto; e
- Apresentar o produto ao cliente.

5.1.2.1 Gerente de Produto

Profissional responsável por realizar gestão, conforme preconiza a Metodologia de Gerenciamento de Projetos (MGP), Metodologia de Desenvolvimento de Sistemas (MDS), arquitetura de referência e políticas de segurança da CAPES, de aplicações, times transversais e serviços de TIC. O Gerente de Produto também atuará como Scrum Master dentro do time de desenvolvimento.

	Metodologia de Desenvolvimento de Sistemas - MDS	Versão: 4.2
		Data Emissão: 13/12/2021

5.1.2.2 Analista Programador

Profissional responsável por executar os serviços necessários a codificação de aplicações, sistemas, componentes e/ou serviços (*back-end*), abrangendo o levantamento de requisitos, manipulação de banco de dados, elaboração de documentação técnica e manuais de usuários, além de atividades relacionadas à criação/adaptação de interface do usuário (*front-end*). Dessa forma o profissional deve ter um conhecimento multidisciplinar nas várias áreas exigidas. Tal perfil é comumente denominado “full-stack” e visa valorizar as habilidades e os conhecimentos de computação do desenvolvedor e da equipe, em linha com o que pregam as orientações “ágil”.

5.1.2.3 Apoio ao PO

Profissional que atua na integração entre os times ágeis e área de negócio, apoiando o Product Owner em suas atividades, conforme preconiza a Metodologia de Gerenciamento de Projetos (MGP), Metodologia de Desenvolvimento de Sistemas (MDS), Arquitetura de Referência e Políticas de Segurança da CAPES. Também é responsabilidade deste profissional produzir a documentação dos sistemas, validar os resultados das entregas do desenvolvedor e apoiar nas migrações de dados.

5.2 Designer UX e UI

Profissional responsável pelas atividades relacionadas às interfaces de sistemas e melhoria da experiência dos usuários junto aos sistemas informatizados, realizando estudos e definindo interfaces que propiciem ganho de usabilidade.

5.3 Arquiteto de Soluções

Profissional responsável por definições, análises e resoluções de problemas arquiteturais dos sistemas em uso pela CAPES, que prioritariamente estão nas linguagens Java, PHP e Python.

5.4 Arquiteto de Automação

Responsável pela gestão da configuração, versionamento e processo de deploy, com foco na automação.

5.5 Administrador de Dados

Equipe formada por profissionais da DTI, responsável pela geração e manutenção das estruturas de dados. É composta pelo Administrador de Dados (AD) e pelo Administrador de Banco de Dados (DBA).

5.6 Analisa de Infraestrutura

Profissional com conhecimento sobre o ambiente de TI e responsável por garantir a disponibilidade e performance adequadas para os serviços de TI que são mantidos por recursos de infraestrutura, tais como servidores/*clusters*.

5.7 Gerente PMO

Profissional experiente que coordena esforços para que a estratégia da organização seja atingida e otimizar os recursos compartilhados entre todos os projetos da organização. Desta forma, deve disponibilizar informações consolidadas sobre os projetos que estão sob seu controle e fornecer suporte metodológico e ferramental para os Gerentes de Produto.

6. MACROPROCESSO DE DESENVOLVIMENTO

O modelo de projeto da CAPES traz uma base comum para todos os projetos de desenvolvimento de produtos. Para tanto são conduzidos a partir de um conjunto de etapas básicas independentemente dos objetivos finais. As etapas funcionam de modo a complementar o modelo de projeto, ou seja, o resultado de cada processo contribui para garantir a qualidade do processo como um todo. O presente modelo estabelece a metodologia de desenvolvimento de sistemas ágil da CAPES, a qual estrutura os processos do ciclo de vida de software.

Este guia está estruturado de acordo com os processos redesenhados, começando com a apresentação do macroprocesso (

Figura 1), seguido por uma breve apresentação do escopo e das principais atividades de cada um dos processos.

Os processos terão um breve descritivo, sendo apresentadas as principais atividades em uma tabela contendo os responsáveis pelos mesmos, os profissionais envolvidos, as ferramentas utilizadas e as entradas e saídas dos processos.

Figura 1 – Macroprocesso da MDS

6.1 Solicitar Sustentação

Este processo é referente a criação de uma demanda na ferramenta de ITSM de sustentação.

As demandas de sustentação se encaixam nos seguintes serviços:

- Dúvida;
- Parametrização;
- Erro;
- Evolutiva pontual (a demanda entra no fluxo de sustentação quando o sistema não possui backlog e um fluxo de execução em andamento);
- Solicitação de relatório;
- Solicitação de acesso em sistema;
- Desbloqueio de conta de usuário em sistema;
- Solicitação de conta de usuário em sistema;
- Solicitação de alteração no cadastro de usuário em sistema;
- Correção de cadastro de dados de usuário em sistema.

A área demandante poderá criar uma demanda de 2 (duas) formas: via telefone pela central de atendimento, onde o Suporte Técnico cadastrará a demanda na ferramenta ITSM ou diretamente pela ferramenta ITSM. No segundo caso, o nível 1 (um) do suporte técnico fará uma triagem na demanda. As demandas serão recebidas pelos times dos sistemas.

Processo	Solicitar Sustentação
Responsável	Área demandante
Perfis e áreas envolvidas	Área demandante Suporte técnico
Entradas	Demanda de sustentação solicitada para a DTI
Ferramentas	Ferramenta ITSM
Saídas	Demanda criada na ferramenta de ITSM com a classificação do serviço.

6.2 Realizar Sustentação

Este subprocesso é referente ao fluxo de sustentação. O time de desenvolvimento recebe a demanda após triagem do nível 1 (um) do suporte. Caso a demanda seja referente a dúvidas, o Apoio ao PO recebe a demanda, já para os demais serviços de sustentação, é realizada análise prévia pelo Gerente de Produto.

Em casos de erros graves, o coordenador responsável é informado para realizar o monitoramento. Na solução de erros é necessário realizar primeiramente análise para definir a estratégia e priorização do atendimento. Neste momento poderá ser necessário intervenção em código, onde o PO é envolvido e o item inserido no *Backlog* com a devida prioridade de atendimento.

Nas solicitações de evolutiva pontual o solicitante deverá ser o PO para demanda ser inserida no *Backlog*.

Processo	Realizar Sustentação
Responsável	Time do produto
Perfis e áreas envolvidas	Time de Desenvolvimento Área Demandante Coordenador Responsável Product Owner Time de Arquitetura Analista de Infraestrutura Administrador de Dados
Entradas	Demanda de sustentação solicitada para a DTI
Ferramentas	Ferramenta ITSM Redmine
Saídas	Demanda finalizada

6.3 Realizar reunião de alinhamento

Quando a necessidade da Área Demandante for um novo produto ou a internalização de um sistema já existente, é necessário que haja uma reunião de alinhamento entre solicitante e Coordenador da CGS que atende a área. Este processo é referente a esta reunião de alinhamento.

Processo	Realizar reunião de alinhamento
Responsável	Área Demandante
Perfis e áreas envolvidas	Área Demandante Coordenador Responsável
Entradas	Solicitação de reunião
Ferramentas	Não se aplica
Saídas	Entendimento inicial a respeito da demanda Ata de reunião

6.4 Solicitar Novo Produto

Este processo é referente a área demandante criar uma solicitação de um novo produto. Um novo produto poderá ser o desenvolvimento de um novo sistema, módulo, ou mesmo um conjunto de entregas relevantes. Neste caso, a Área Demandante deverá preencher um Termo de Abertura do Projeto (TAP) Preliminar.

A área demandante deverá criar a demanda via ferramenta ITSM. O nível 1 (um) do suporte técnico fará uma triagem na demanda. A demanda será recebida pela Coordenador da CGS que atende a área.

Processo	Solicitar Novo Produto
Responsável	Área Demandante
Perfis e áreas envolvidas	Área Demandante Suporte técnico

	Coordenador Responsável
Entradas	Demanda de um novo produto solicitada para a DTI
Ferramentas	Ferramenta ITSM
Saídas	Demanda criada na ferramenta de ITSM com a classificação do serviço. Termo de Abertura do Projeto (TAP) Preliminar

6.5 Solicitar Internalização

Este processo é referente a área demandante criar uma solicitação de internalização de sistema. Neste caso, a Área Demandante deverá preencher um Termo de Abertura do Projeto (TAP) Preliminar. A área demandante deverá criar a demanda via ferramenta ITSM. O nível 1 (um) do suporte técnico fará uma triagem na demanda. A demanda será recebida pela Coordenador da CGS que atende a área.

Processo	Solicitar Internalização
Responsável	Área Demandante
Perfis e áreas envolvidas	Área Demandante Suporte técnico Coordenador Responsável
Entradas	Demanda de internalização de sistema solicitada para a DTI
Ferramentas	Ferramenta ITSM
Saídas	Demanda criada na ferramenta de ITSM com a classificação do serviço. Termo de Abertura do Projeto (TAP) Preliminar

6.6 Analisar Solicitação

Subprocesso referente a análise realizada pela Coordenação responsável por atender a demanda, Governança e Escritório de Projetos. Neste momento, o TAP preliminar é validado e caso

	Metodologia de Desenvolvimento de Sistemas - MDS	Versão: 4.1
		Data Emissão: 24/07/2020

seja necessário ajustes a área demandante deverá realizar. O time responsável pela demanda é designado.

Processo	Analisar Solicitação
Responsável	Coordenador Responsável
Perfis e áreas envolvidas	Área Demandante Coordenador Responsável Governança Escritório de Projetos
Entradas	Demanda de novo produto ou de internalização de sistema solicitada para a DTI
Ferramentas	Ferramenta ITSM
Saídas	Termo de Abertura do Projeto (TAP) Preliminar validado. Time responsável designado.

6.7 Executar Internalização

Este subprocesso é referente a execução da internalização de uma aplicação. Se encaixam tanto os casos de migração de um sistema da CAPES para um sistema do Governo Federal ou a implantação de uma ferramenta não desenvolvida pela CAPES. Neste último caso, a ferramenta deverá passar pelo crivo da equipe técnica.

Processo	Executar Internalização
Responsável	Coordenador Responsável Escritório de Projetos Product Owner
Perfis e áreas envolvidas	Product Owner Escritório de Projetos Analista de Infraestrutura Administrador de Dados

	Time de Arquitetura Designer UX e UI Coordenador Responsável
Entradas	Solicitação de internalização de sistema
Ferramentas	Ferramenta ITSM Redmine Sistema de Gestão Integrado de Projetos - GIP
Saídas	Relatório técnico Sistema internalizado Documento de Arquitetura de Sistemas Projeto criado ou atualizado no Sistema de Gestão Integrado de Projetos - GIP (Project PWA)

6.8 Realizar Lean Inception

Processo de *Lean Inception*, envolve a definição do produto; identificação dos objetivos do produto, identificação das personas, identificação e avaliação das macros funcionalidades, priorização e planejamento do *Roadmap de release*.

Processo	Realizar <i>Lean Inception</i>
Responsável	Gerente de Produto Escritório de Projetos Product Owner
Perfis e áreas envolvidas	Time de desenvolvimento Product Owner Escritório de Projetos Analista de Infraestrutura Administrador de Dados Time de Arquitetura Designer UX e UI

	Coordenador Responsável
Entradas	Termo de Abertura do Projeto Preliminar
Ferramentas	Ferramenta ITSM Redmine Axure Sistema de Gestão Integrado de Projetos - GIP
Saídas	Produto Planejado Documento de Arquitetura de Sistemas Roadmap de releases Documento de regras de negócio Backlog do Produto Documentação online Projeto criado ou atualizado no Sistema de Gestão Integrado de Projetos - GIP (Project PWA)

6.9 Definir processo utilizado

O time do produto poderá escolher o processo a ser utilizado para atendimento das demandas desde que esteja previsto na MDS. Podendo ser:

- Kanban;
- Scrum e Kanban;
- Ágil em escala customizado para CAPES.

Processo	Definir processo utilizado
Responsável	Time de Desenvolvimento
Perfis e áreas envolvidas	Time de Desenvolvimento Escritório de Projetos Coordenador Responsável

Entradas	Demanda de desenvolvimento
Ferramentas	Ferramenta ITSM Redmine
Saídas	Processo definido

6.10 Executar Kanban

Processo de execução da demanda utilizando o framework Kanban. O Time do Produto deverá executar as seguintes atividades:

- Mapear comportamento das telas e gerar protótipo;
- Gerar HTML e CSS das telas;
- Gerar entrega com funcionalidades passíveis de teste;
- Modelar o banco de dados;
- Realizar teste funcional;
- Implantar em ambiente de homologação;
- Realizar avaliação técnica de qualidade;
- Realizar daily;
- Definir limite da *Wip* e acompanhar os indicadores de desempenho *lead time* e *cycle time*.
- Manter o quadro kanban atualizado.

Processo	Executar Kanban
Responsável	Gerente de Produto
Perfis e áreas envolvidas	Product Owner Time de Desenvolvimento Time de Arquitetura Administrador de Banco de Dados Designer UX e UI
Entradas	Demanda de desenvolvimento

Ferramentas	Redmine Kanban Axure
Saídas	Protótipos de telas navegáveis Regras de negócio atualizadas Modelo de Banco de Dados (conceitual e físico) Scripts de banco de dados Código da aplicação – front-end Código da aplicação – back-end Componentes Serviços e micros serviços Documentação técnica Kanban atualizado

6.11 Executar Scrum e Kanban

Processo dinâmico da execução das demandas utilizando os frameworks Kanban e Scrum aplicado ao desenvolvimento e manutenção de sistemas. O Time do Produto deverá executar as seguintes atividades:

- Planejar a Sprint;
- Mapear comportamento das telas e gerar protótipo;
- Gerar HTML e CSS das telas;
- Gerar release com funcionalidades passíveis de teste;
- Modelar o banco de dados;
- Realizar teste funcional com base no escopo da Sprint;
- Realizar reunião de planejamento da Sprint;
- Realizar daily;
- Realizar reunião de review;
- Realizar reunião de retrospectiva;
- Implementar em ambiente de homologação;
- Realizar avaliação técnica de qualidade;

- Definir limite da *Wip* e acompanhar os indicadores de desempenho *lead time* e *cycle time*; e
- Manter o quadro kanban atualizado.

Processo	Executar Scrum e Kanban
Responsável	Gerente de Produto
Perfis e áreas envolvidas	Product Owner Time de Desenvolvimento Time de Arquitetura Administrador de Banco de Dados Designer UX e UI
Entradas	Histórias de Usuário Critérios de aceitação Regras de negócio atualizadas
Ferramentas	Redmine Repositório – SVN Axure
Saídas	Backlog da Sprint atualizado Protótipos de telas navegáveis Regras de negócio atualizadas Testes de aceitação elaborados Modelo de Banco de Dados (conceitual e físico) Scripts de banco de dados Código da aplicação – front-end Código da aplicação – back-end Componentes Serviços e micros serviços Documento de arquitetura do sistema atualizado Memória de reunião de Review

6.12 Executar Ágil em Escala

Processo dinâmico da execução das demandas utilizando *Scrum* escalado (adaptação do framework Nexus para realidade da CAPES), a ser utilizada em sistemas que possuem múltiplos times de desenvolvimento, aplicado ao desenvolvimento e manutenção de sistemas. O Time do Produto deverá executar as seguintes atividades para cada um dos times de desenvolvimento, em cima de um *backlog* único do produto:

- Planejar a Sprint;
- Mapear comportamento das telas e gerar protótipo;
- Gerar HTML e CSS das telas;
- Gerar release com funcionalidades passíveis de teste;
- Modelar o banco de dados;
- Realizar teste funcional com base no escopo da Sprint;
- Realizar reunião de planejamento da Sprint;
- Realizar daily;
- Realizar reunião de review;
- Realizar reunião de retrospectiva;
- Implementar em ambiente de homologação;
- Realizar avaliação técnica de qualidade;
- Realizar reunião semanal de acompanhamento da integração;
- Definir limite da *Wip* e acompanhar os indicadores de desempenho *lead time* e *cycle time*; e
- Manter o quadro kanban atualizado.

Processo	Executar Scrum e Kanban
Responsável	Gerente de Produto
Perfis e áreas envolvidas	Product Owner Time de Desenvolvimento Time de Arquitetura Administrador de Banco de Dados Designer UX e UI

Entradas	Histórias de Usuário Critérios de aceitação Regras de negócio atualizadas
Ferramentas	Redmine Repositório – SVN Axure
Saídas	Backlog da Sprint atualizado Protótipos de telas navegáveis Regras de negócio atualizadas Testes de aceitação elaborados Modelo de Banco de Dados (conceitual e físico) Scripts de banco de dados Código da aplicação – front-end Código da aplicação – back-end Componentes Serviços e micros serviços Documento de arquitetura do sistema atualizado Memória de reunião de Review

6.13 Preparar Backlog

Processo de preparação do *backlog*, que envolve a criação, alteração e exclusão de itens, priorização do backlog do produto e o detalhamento desses itens em Histórias de Usuário. Na reunião de preparação de Backlog, os itens do backlog do projeto devem ser priorizados e detalhados de forma a tornar viável a execução.

Processo	Preparar backlog
Responsável	Gerente de Produto Apoio ao PO Product Owner

Perfis e áreas envolvidas	Time de Desenvolvimento Product Owner
Entradas	Backlog do projeto cadastrado no Redmine Roadmap de Releases Documento de regras e negócio Documentação online
Ferramentas	Redmine Axure
Saídas	Backlog do projeto atualizado Histórias de Usuário criadas Critérios de aceitação criados Memória de reunião

6.1 Planejar Release

Processo de planejamento de uma *release* que envolve a definição dos itens do *backlog* que serão inseridos na release. Sempre que houver necessidade, o Gerente de Produto deve levar em consideração o *backlog* e levantar junto ao demandante quais itens de *backlog* deverão compor a release. O Gerente de Produto deve conduzir a reunião com todas as equipes envolvidas de modo a garantir a disponibilidade adequada dos profissionais ao atendimento das demandas da release.

Processo	Planejar Release
Responsável	Gerente de Produto
Perfis e áreas envolvidas	Product Owner Time de desenvolvimento
Entradas	Backlog do projeto atualizado Documento de regras de negócio atualizado Histórias de usuário criadas Critérios de aceitação criadas

	Roadmap de Releases
Ferramentas	Redmine Gestão Integrada de Projetos (GIP)
Saídas	<i>Backlog da release</i> Roadmap de releases atualizado

6.2 Homologar Aplicação

Processo dinâmico de homologar aplicação como premissa de avaliar uma versão preliminar do sistema. Essa versão deve ser validada conforme a solicitação realizada para verificar a eficácia do software como solução. A homologação é a comprovação, pelo cliente e demais partes interessadas, de que o produto resultante do projeto de software atende aos critérios de aceite previamente estabelecidos com o cliente. Inclui elementos de verificação e de validação do produto todo ou de partes do produto selecionadas em comum acordo com o cliente e tem como meta principal a obtenção do aceite do produto. Tal atividade só deve começar depois que todos os outros testes forem executados, e os erros encontrados forem corrigidos ou aceitos. A homologação da aplicação é acompanhada pelo Gerente do Produto.

Processo	Homologar Aplicação
Responsável	Gerente de Produto
Perfis e áreas envolvidas	Product Owner Time de Desenvolvimento
Entradas	Documentação <i>on-line</i> Histórias de Usuário
Ferramentas	Redmine Axure
Saídas	Demanda pronta para Produção Redmine atualizado

6.3 Implantar em Produção

Processo dinâmico de realização da liberação de sistemas em Produção no contexto de desenvolvimento e manutenção de sistemas da CAPES. Após homologação da entrega, o PO tomará a decisão de quando a versão deverá ser implantada em Produção. Desta forma, o sistema deverá ser liberado em produção pela equipe técnica da CAPES, respeitando o processo vigente definido para mudanças em produção.

Processo	Implantar em Produção
Responsável	Gerente de Produto Product Owner
Perfis e áreas envolvidas	Product Owner Time do Desenvolvimento Analista de infraestrutura Administrador de Dados
Entradas	Backlog da Sprint atualizado Modelo de Banco de Dados (conceitual e físico) Scripts de banco de dados Código da aplicação – front-end Código da aplicação – back-end Componentes Serviços e micros serviços Documento de arquitetura do sistema atualizado Formalização da homologação
Ferramentas	Redmine Ferramenta de ITSM GIT Lab Jenkins
Saídas	GMUD



Metodologia de Desenvolvimento de Sistemas - MDS

Versão: 4.1

Data Emissão:

24/07/2020

Sistema disponível em produção

7. FERRAMENTAS

Este item enumera as ferramentas utilizadas no processo de desenvolvimento e manutenção de sistemas de informação na CGS. Para verificar as ferramentas utilizadas no processo de desenvolvimento e manutenção de sistemas de informação na CGS, acessar o documento “MDS - Ferramentas_Referencia.docx”.

8. REFERÊNCIAS

- <http://www.scrumguides.org;>
- <http://www.manifestoagil.org;>
- Livro “Programação Extrema (XP) Explicada” – Kent Beck;
- PDSBCÁgil - Processo Ágil de Desenvolvimento de SW do BACEN.
- WEBER, K.C., CAVALCANTI DA ROCHA, A.R., MALDONADO, J.C. Qualidade de software: teoria e prática, vol. III, Prentice Hall, 2001.
- PRESSMAN, R. Engenharia de Software: Uma Abordagem Profissional. 7ª Ed., São Paulo: McGraw-Hill, 2011.
- Fonte: <http://tutano.trampos.co/16410-guia-de-profissoes-scrum-master/>
- BOEG, Jesper. Kanban em 10 Passos: Otimizando o fluxo de trabalho em sistemas de entrega de software. Leonardo Galvão. InfoQ Brasil, C4Media Inc., 2014.

	Metodologia de Desenvolvimento de Sistemas - MDS	Versão: 4.1
		Data Emissão: 24/07/2020

9. ANEXO I - Definições, Acrônimos e Abreviações

- Backlog – Backlog refere-se a um log de acumulação de trabalho num determinado intervalo de tempo. Backlog é uma espécie de estoque de folhas de requisições/encomendas relativas a produtos ainda não produzidos. Grosso modo, backlog é uma "pilha de pedidos" em espera. O Product Backlog pode crescer muito e com o tempo pode se tornar muito grande e desorganizado. Ele poderá apresentar histórias que já foram finalizadas e não necessitam mais serem exibidas, assim como histórias que, com a evolução do projeto, se tornam obsoletas e devem ser excluídas.
- Benchmarks - É uma análise estratégica das melhores práticas usadas por empresas do mesmo setor que o seu. É uma ferramenta de gestão que objetiva aprimorar processos, produtos e serviços, gerando mais lucro e produtividade.
- CAPES – Coordenação de Aperfeiçoamento de Pessoal de Nível Superior.
- CATI – Central de Atendimento de TI.
- Critérios de aceitação - Lista de itens de negócio que expressam as formas de usar a funcionalidade implementada em uma história, bem como as regras e detalhes que validam se a história foi implementada de acordo com o que o Product Owner esperava e se está aceitável para o uso em produção.
- DTI - Diretoria De Tecnologia Da Informação.
- Deploy – É a instalação da sua aplicação em um servidor de aplicações.
 - Deploy de Release - Corresponde a versão do build gerado na homologação, testado e aprovado pela área gestora que será atualizado em produção. A versão para produção deve estar devidamente aprovada por meio do Aceite Formal.
- Frameworks – Um framework em desenvolvimento de software, é uma abstração que une códigos comuns entre vários projetos de software provendo uma funcionalidade genérica. Um framework pode atingir uma funcionalidade específica, por configuração, durante a programação de uma aplicação.
- Full-stack - Programador full-stack é aquele que atua em várias partes do projeto (back-end, front-end, banco de dados) e para isso pode usar várias tecnologias.
- História – Breve descrição de algo que o sistema deve fazer, com clareza suficiente para ser entendida por usuários e desenvolvedores, escrita de maneira simples, objetiva, eficaz e eficiente. Não deve conter detalhes, apenas informar: o que se deverá realizar, o motivo para a execução e para quem está sendo escrita.

- Lista de necessidades da área solicitante – Lista de itens que a área gestora aponta como necessárias e que gera um projeto de implementação de software. No Workshop de Requisitos ela será tratada, priorizada e a partir dela será gerado o Product Backlog.
- OLAP - É a capacidade para manipular e analisar um grande volume de dados sob múltiplas perspectivas.
- PDTIC - Plano Diretor de Tecnologia da Informação e Comunicação.
- Produto - Conjunto de programas de computador (código fonte, testes automatizados, scripts de banco de dados, testes e operação), procedimentos, documentação e dados associados, que traz o resultado esperado de uma implementação que pode ser parte ou o todo de uma aplicação e atende às necessidades do Product Owner. É o principal objetivo do Projeto.
- PMO - O Escritório de projetos ou Project Management Office é um departamento que tem como responsabilidade definir e zelar pela manutenção dos padrões de gerenciamento de projetos e processos de governança relacionados a programas sob a sua tutela.
- Release – É o intervalo entre dois objetivos primários do desenvolvimento do projeto. O objetivo primário é entender com maior clareza, o escopo e o planejamento do projeto com vistas a produzir o produto. Para cada Release são definidas metas a serem atingidas, artefatos que deverão ser concluídos e tomadas de decisões.
- Roadmap - É uma espécie de "mapa" que visa organizar as metas de desenvolvimento de um software.
- Sprint – Ciclos específicos de tempo, curtos, onde serão executadas as histórias de usuário divididas em tarefas com o objetivo de gerar incremento da aplicação, potencialmente entregável.
- Tarefas – Divisão de uma história em atividades menores que devem ter como características: serem específicas, mensuráveis, capazes de serem executadas pela equipe com sua qualificação, relevantes para o usuário e realizável dentro de um espaço de tempo de no máximo de 8h, preferencialmente.
- Velocidade da Equipe – medida da quantidade de trabalho que uma equipe consegue executar em um período de tempo determinado. É variável e específica para cada equipe de projeto. Só pode ser identificada após algumas semanas de trabalho, quando a equipe se torna mais integrada e inteirada do negócio do sistema e do processo de desenvolvimento.
- Versão do produto – É a versão da aplicação que deverá ser testada, homologada e colocada em produção. Ela deve ser preparada antes da promoção de ambientes ser feita

- Visão do Produto – É a visão do que o produto vai entregar de benefícios para o cliente. Ela deverá ser clara e conter as seguintes informações: nome, categoria, para quem está sendo desenvolvida, a necessidade da área gestora que o produto irá atender, que benefícios trará e, se houver concorrente ou legado, informar o diferencial do já existente.

10. ANEXO II – Produtos de Trabalho

- Atas de Reuniões - São documentos utilizados para registrar as ocorrências que ocorram durante o desenvolvimento do projeto, independente da metodologia de desenvolvimento escolhida.
- Cenários de Teste – Roteiro em tópicos que descreve “como” será realizado o teste. Contempla os cenários previstos na história incluindo as validações de regras de negócio e obrigatoriedade. Os cenários devem ser registrados em ferramenta específica considerando:
 - Identificação do cenário;
 - O que deve ser feito;
 - Dados de Entrada;
 - Resultado Esperado.
- Chamado de Atendimento - Chamado registrado pelo usuário gestor para atendimento de demandas da sua área.
- Código implementado - Corresponde à codificação do sistema, ou parte deste, em linguagem de programação específica. O código para estar completo deve exercer a função que fora especificada e atender aos requisitos funcionais e não funcionais descritos na documentação do sistema.
- Cronograma - É o sequenciamento das atividades do projeto conforme acordo firmado com o gestor do sistema. No cronograma é contemplado o marco de entrega, as atividades com a quantidade de horas e início/fim, alocação de recursos e os registros de trabalho em cada uma das tarefas. Para desenvolvimento ágil, os marcos de entrega devem ser registrados ao final dos Releases e os registros das tarefas dentro das Sprints.
- Documento de Arquitetura - Descreve a arquitetura do projeto. Utilizado apenas para projetos que não utilizam integralmente a arquitetura definida pela Equipe de Arquitetura da CAPES. Pode ser utilizado independente da metodologia de desenvolvimento escolhida.
- Documento de Regras de Negócio – Tem o objetivo de documentar os requisitos funcionais e não funcionais sistematizados. Para melhor organização e entendimento, recomenda-se que as regras sejam agrupadas por assunto ou funcionalidade. As regras de negócio que são comuns a mais de um assunto ou funcionalidade devem ter a mesma referência para possibilitar a rastreabilidade.
- Documento de Visão - Este artefato delimita o escopo do projeto a ser realizado. Auxilia na análise da solução, na identificação de requisitos, na expectativa de prazo, custo e recursos

consumidos pelo projeto. O template proposto para o novo Documento de Visão recebe em sessão específica e opcional, o Product Backlog, para os casos de desenvolvimento ágil.

- Evidência de Teste - Artefato utilizado para descrever o resultado de testes conforme os Cenários de Teste. Contém os seguintes atributos:
 - Ambiente de Teste;
 - Técnico da realização dos testes;
 - Data e hora do teste;
 - Cenário de testes envolvidos;
 - Registro de anomalias e/erros do sistema;
 - Parecer do teste.

Este artefato é gerado na ferramenta utilizada para criação de cenários de teste e registro de evidência, que possui integração com a ferramenta de bug tracking adotada pela CAPES.

- Formulário de Solicitação de Mudança - Documento que descreve qual será a alteração, entrada ou saída de um ou mais ativos de informação do ambiente de produção, contendo um planejamento das ações a serem executadas na mudança.
- Kanban – É uma estrutura popular, usada para realizar o desenvolvimento de software ágil e comunica em tempo real e de forma transparente o estado atual do trabalho que está sendo feito. Através de colunas que identificam: o que foi programado para ser feito, o que está em execução, o que foi concluído e o que está pendente de recursos ou de terceiros, pode-se verificar em que situação está cada tarefa de uma história dentro de uma Sprint.
- Product Backlog – Lista priorizada das necessidades do cliente que precisam ser produzidas para que a Visão do Produto em construção seja atingida. É um artefato vivo e deve ser atualizado ao final dos Releases e das Sprints, bem como toda vez que houver mudança relevante que altere seu conteúdo. Deve ser registrado em seção específica no Documento de Visão
- Protótipo – Desenho ou imagem da interface a ser utilizada em uma funcionalidade da aplicação. O objetivo é tornar mais claros, os conceitos que possam ser abstratos se forem apenas registrados por escrito. Pretende também facilitar a internalização de novas ideias e a visão da comunicação entre as telas, de forma que a equipe possa identificar conflitos, caso haja, e tratá-los a tempo. Recomenda-se que o processo seja feito em ferramenta determinada pela equipe ou em prototipagem de papel (descartável e envolve a criação de desenhos simples de uma interface, feitos à mão), sem complexidade de desenho ou gasto de tempo que possa impactar no cronograma.

- Produto de Software – Conjunto de programas de computador (código fonte, testes automatizados, scripts de banco de dados, testes e operação), procedimentos, documentação e dados associados.
- Plano de Implantação - Garantir que o sistema alcance seus usuários com sucesso, fornecendo uma agenda detalhada de eventos, pessoas responsáveis e dependências necessárias para garantir a mudança bem-sucedida para o novo sistema. Este documento é gerado nas atividades da MGP.
- Plano de Migração de Dados - Este documento consiste em descrever a migração de dados que será necessária para o projeto, independente da metodologia de desenvolvimento escolhida.
- Protótipo - Consiste na simulação das telas de um determinado sistema. Sua função principal é auxiliar na elucidação dos requisitos funcionais junto ao Product Owner, uma vez que permite visualizar a aplicação ainda não produzida. A prototipação ajuda na tomada de decisões e consequentemente reduz as mudanças durante o processo de homologação do produto.
- Relatório de Teste - Este documento tem como finalidade mostrar informações resumidas determinadas pela análise de um ou mais registros de teste, que permitem uma avaliação relativamente detalhada da qualidade dos Itens de teste-alvo e do status de cada esforço de teste. Tem como finalidade apresentar os resultados obtidos com a execução das funcionalidades dos sistemas. Organiza e apresenta uma análise resumida dos resultados e as principais medidas do teste para revisão e avaliação, geralmente executadas pelos principais envolvidos na questão de qualidade. Este sumário apresenta indicadores gerais sobre a qualidade dos testes da Sprint e fornece recomendações para um futuro esforço de teste
- Release Backlog – Parte do Product Backlog destacada para ser executada em um determinado Release. É um subconjunto do Product Backlog.
- Sprint Backlog – Parte do Release Backlog destacada para ser executada em uma determinada Sprint. É um subconjunto do Release Backlog.
- Tipos de Teste – Tipos de testes disponíveis que garante uma maior confiabilidade e qualidade aos sistemas alvo dos testes.