

ANEXO I - TERMO DE REFERÊNCIA

METODOLOGIA DE DESENVOLVIMENTO DE SOFTWARE - MDS

1. INTRODUÇÃO

1.1. Esta Metodologia de Desenvolvimento de Software (MDS) visa descrever e normatizar os processos de gerenciamento, desenvolvimento e manutenção de softwares da Enap. Ela acompanha o desenvolvimento de produtos de software, ou seja, o fluxo de trabalho e o passo-a-passo de como gerenciar as demandas de desenvolvimento de sistemas junto às empresas contratadas para tal fim.

1.2. Pretende-se com a aplicação desta MDS promover o aumento na eficiência e eficácia da gestão de projetos de software, bem como nortear as atividades de desenvolvimento e manutenção dos sistemas corporativos da Enap.

1.3. Cabe salientar a diferença entre Metodologia de Desenvolvimento de Software e Metodologia de Gerenciamento de Projetos. A Metodologia de Gerenciamento de Projetos - MGP é um conjunto de processos, métodos e ferramentas que visam alcançar os objetivos do projeto, por meio a aplicação de conhecimentos, habilidades e técnicas para projetar atividades que visem atingir as expectativas das partes envolvidas. Já na Metodologia de Desenvolvimento de Software o foco é a construção de um determinado produto de software, com suas características e especificidades.

1.4. Resultados e benefícios a serem alcançados

1.4.1. Aumento da disponibilidade dos sistemas de informação da Enap, garantindo maior continuidade e automação das atividades de ensino desta Escola;

1.4.2. Aperfeiçoamento do atual modelo de gestão e o consequente aumento do nível de maturidade, por meio da governança de TIC;

1.4.3. Redução de custos com desenvolvimento e sustentação de software, propiciando a racionalização e maximização dos recursos orçamentários e financeiros;

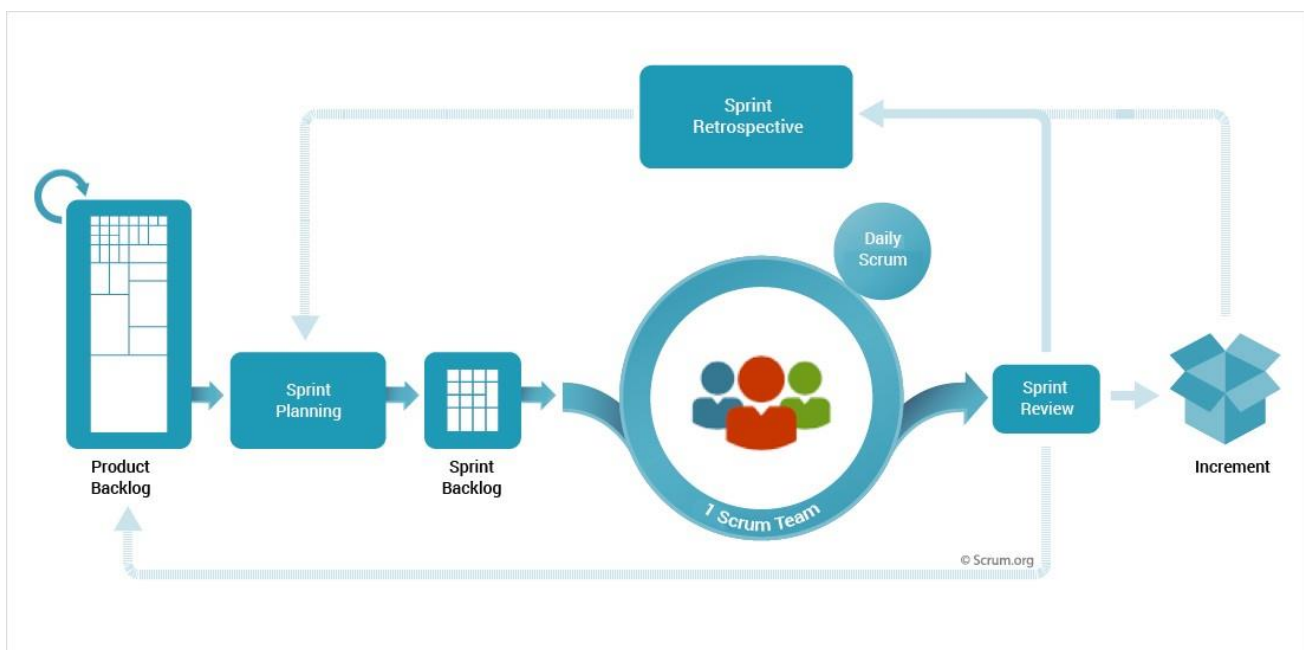
1.4.4. Maior engajamento e satisfação das áreas de ensino da Enap, gerando aumento da produtividade dos servidores;

1.4.5. Transparência do custo e prazo de entrega da demanda;

1.4.6. Acompanhamento, pelo solicitante, de cada etapa do processo de execução de sua solicitação.

1.5. O framework Scrum

1.5.1. Scrum é um *framework* estrutural que está sendo usado para gerenciar o trabalho em produtos complexos desde o início de 1990. Scrum não é um processo, técnica ou um método definitivo. Em vez disso, é um framework dentro do qual pode-se empregar vários processos ou técnicas. O Scrum deixa clara a eficácia relativa de suas práticas de gerenciamento de produto e técnicas de trabalho, possibilitando continuamente melhorar o produto, o time e o ambiente de trabalho. O framework Scrum consiste de times Scrum associados a papéis, eventos, artefatos e regras. Sendo assim, cada componente dentro do framework serve a um propósito específico e é essencial para o uso e sucesso do Scrum. As regras do Scrum integram os papéis, eventos e artefatos, administrando as relações e interações entre eles.



2. OBJETIVO

2.1. O objetivo desta MDS é definir a forma de trabalho, ou seja, como serão gerenciados o desenvolvimento de produtos de software na Enap. O modelo é contextualmente baseado no framework Scrum e em boas práticas de mercado. Embora Scrum seja o alicerce principal do modelo, outras metodologias (ou parte delas) poderão ser utilizadas pela Enap, tais como Ciclo PDCA, Kanban, Design Thinking e método Lean.

2.2. Cabe destacar que o fluxo descrito nesta MDS serve de apoio aos serviços de desenvolvimento de novos sistemas e manutenções evolutivas complexas, com duração estimada de mais de duas semanas. Os serviços relacionados a sustentação de software e pequenas manutenções poderão seguir um fluxo mais simplificado.

2.3. Seja qual for o tipo de desenvolvimento, o foco deve ser atender a necessidade da área de negócio, objetivando a entrega do produto de forma adequada, simples, com decisões mais rápidas e assertivas. Assim, os clientes (áreas de negócio) definem as necessidades (POR QUÊ e O QUÊ), enquanto a forma de atendê-los, ou seja, o "COMO" será definido pela Coordenação Geral de Tecnologia da Informação - CGTI.

2.4. Os líderes do projeto e stakeholders devem ser solucionadores de problemas e entender sobre o negócio de forma a responder sobre as dúvidas e regras de negócios que poderão surgir pelo time de desenvolvimento.

3. PAPÉIS

3.1. Um determinado papel define o comportamento e responsabilidades de um profissional ou grupo que participam do projeto. A partir dessa visão, abaixo é apresentado os principais papéis envolvidos no desenvolvimento de um projeto de desenvolvimento de software:

3.1.1. **Product Owner (PO):** Líder do produto, analista de negócio, supervisor de entregas e marcos do produto. Trabalha gerenciando o backlog do produto, ou seja, "O QUE" deve ser feito. Deve garantir o valor do produto, definir e esclarecer os itens de backlog do produto (requisitos), bem como suas prioridades. Deve dominar o negócio, o produto e ter o poder decisório. É uma pessoa, nunca pode ser um grupo ou comitê (mas pode se reportar a um). Deve estar sempre disponível para o time de desenvolvimento e deve conhecer as regras, necessidades e expectativas dos usuários e stakeholders. É responsável pela solicitação (fiscal requisitante) e homologação da demanda, acompanhando o processo de desenvolvimento e respondendo a possíveis questionamentos sobre as regras de negócio. É responsável, em conjunto com o Scrum Master e Time de desenvolvimento, pelas etapas de "Definição de Preparado" e "Definição de Pronto", sendo responsável pela homologação final da demanda para aprovação e autorização para disponibilizar as mudanças em ambiente de produção.

3.1.2. **Scrum Master (SM):** Líder técnico, um supervisor, um coordenador, um gerente de equipe de especialista ou mesmo um gerente de projetos. Trabalha para garantir que a cultura e as práticas do Scrum sejam seguidas, ou seja, "COMO" as coisas devem ser feitas. Devem ser os solucionadores de problemas.

3.1.3. **Time de Desenvolvimento (TD):** É responsável pelo desenvolvimento do produto. O principal objetivo das equipes é ter recursos que cumpram as habilidades necessárias para alcançar os objetivos previamente estipulados, no tempo e com os custos acordados.

4. EVENTOS

4.1. **Sprint:** evento com duração de 01 (um) mês ou menos, durante o qual um incremento de produto potencialmente liberável é criado seguindo as definições estabelecidas na Definição de Pronto. Uma nova Sprint inicia imediatamente após a conclusão da Sprint anterior.

4.1.1. Somente o *Product Owner* tem a autoridade para cancelar a Sprint, embora ele (ou ela) possa fazer isso sob influência das partes interessadas, do Time de Desenvolvimento ou do Scrum Master;

4.1.2. Quando a Sprint é cancelada, qualquer item de Backlog do Produto completado e "Pronto" é revisado. Se uma parte do trabalho estiver potencialmente liberável, tipicamente o Product Owner o aceita. Todos os itens de Backlog do Produto incompletos são reestimados e colocados de volta no Backlog do Produto;

4.1.3. Durante a fase de planejamento do projeto de desenvolvimento de software a equipe Scrum fará a estimativa de quantidade de Sprints necessárias e suas respectivas durações.

4.2. **Sprint Planning:** evento com duração máxima 08 (oito) horas para uma Sprint de um mês. Para Sprints menores, este evento é proporcionalmente menor.

4.2.1. O Sprint Planning responde as seguintes questões:

4.2.2. **O que** pode ser entregue como resultado do incremento da próxima Sprint?

4.2.3. **Como** o trabalho necessário para entregar o incremento será realizado?

4.2.4. Durante o Sprint Planning, é obrigatório o Time Scrum definir a meta da Sprint, a qual é o objetivo que será satisfeito dentro da Sprint através da implementação do Backlog do Produto, e que fornece a orientação para o Time de Desenvolvimento sobre o porquê dele estar construindo o incremento.

4.2.5. **A Definição de Preparado** é um acordo formal entre *Product Owner* e Time de Desenvolvimento sobre o estado em que um item do Product Backlog deve estar para ser qualificado para discussão na reunião de Sprint

Planning. É importante que durante a Sprint Planning os requisitos estejam levantados e as regras de negócios estejam claras.

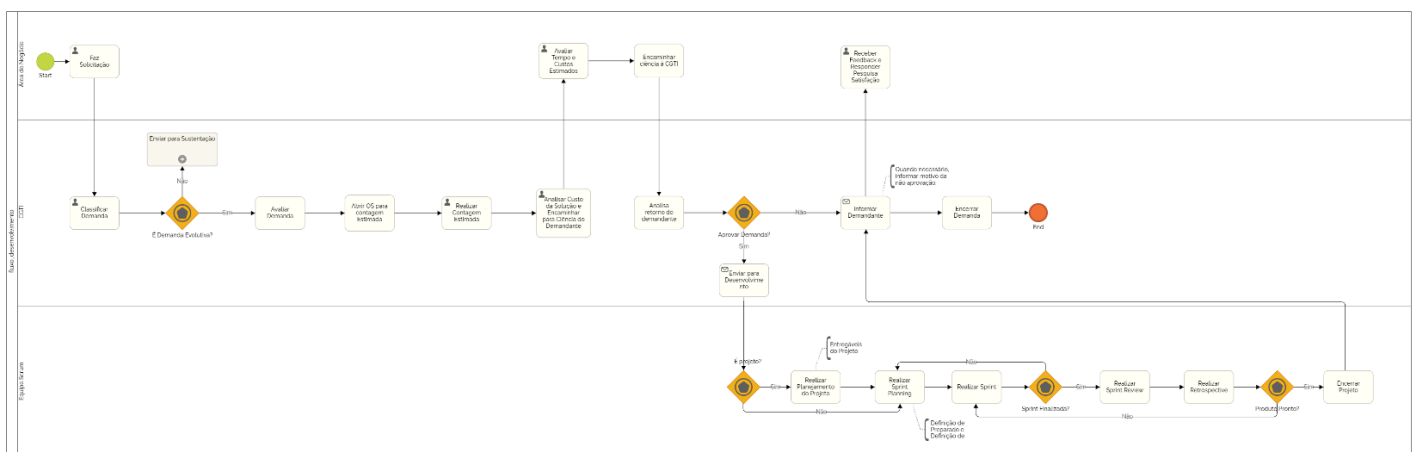
4.2.6. A **Definição de Pronto** é um acordo entre o time de desenvolvimento e o *Product Owner* (PO) para deixar claro o que se espera de uma história de usuário. É através dela que sabemos quais fatores devemos desenvolver em uma funcionalidade para ser entregue como um incremento de qualidade. Portanto, para considerar a demanda homologada, o PO (área de negócio) deve seguir a definição de pronto, de acordo com o que foi acordado no Planejamento da Sprint. É papel do PO a homologação do produto.

4.3. **Daily Meeting:** reunião diária com duração máxima de 15 (quinze) minutos, para colocar em dia o que já foi executado, informar o que está previsto a ser executado e se existe algum impedimento.

4.4. **Sprint Review:** o objetivo da Revisão do Sprint é verificar e adaptar o produto que está sendo construído. É uma reunião para apresentação do incremento desenvolvido, que visa motivar e obter comentários, bem como promover a colaboração dos envolvidos. A Sprint Review possui duração máxima de 04 (quatro) horas para uma Sprint de um mês. É durante a Sprint Review que o resultado da Sprint é apresentado e que deve acontecer a homologação da demanda (produto), de acordo com a definição de pronto, previamente acordada entre o PO e Time Scrum. Caso ocorra uma não entrega ou entrega parcial é importante haver a comunicação dos problemas ocorridos e negociar novos prazos, custo e escopo, e priorizar na próxima iteração a inclusão dos itens não finalizados.

4.5. **Sprint Retrospective:** enquanto o objetivo da Sprint Review é verificar as necessidades de adaptações no produto, o Sprint Retrospective tem o objetivo de verificar necessidades de adaptações no processo de trabalho. É o momento da equipe refletir sobre como as coisas foram feitas e como elas podem melhorar. A Sprint Retrospective possui duração máxima de 03 (três) horas para uma Sprint de um mês.

5. PROCESSO DE DESENVOLVIMENTO DE SOFTWARE



5.1. O processo de desenvolvimento de software abrange as atividades de Gerenciamento do Projeto e Desenvolvimento de Software e, a partir desse fluxo de desenvolvimento, será organizado por fases, contemplando as atividades e artefatos do projeto e do produto, os quais deverão ser gerados pela equipe Scrum.

5.2. Cada projeto de desenvolvimento será executado pela **Equipe Scrum**, composta por Product Owner (PO), Scrum Master (SM) e Time de Desenvolvimento (TD). Assim, o projeto deverá seguir as fases descritas abaixo, facultando a Enap suprimir ou abreviar fases em projetos de menor escala.

5.3. Tipos de desenvolvimentos:

5.3.1. Novos desenvolvimentos - correspondem ao desenvolvimento de novos sistemas de informação, em alinhamento ao PDTIC. As especificações serão estabelecidas pela CGTI em conjunto com a área de negócio.

5.3.2. Projeto de melhoria, manutenção evolutiva ou manutenção funcional - corresponde à inclusão, alteração e exclusão de características e/ou funcionalidades em aplicações em produção, decorrentes da alteração das regras de negócio ou de demandas legais (requisitos funcionais).

5.3.3. Manutenção corretiva - consiste na correção de defeitos em sistemas em produção e abrange correção de erros que causam problemas de uso e/ou de funcionamento. Se a correção for feita dentro do prazo de garantia do contrato, deve ser feita sem ônus para a CGTI.

5.3.4. Manutenção adaptativa - adequação da aplicação decorrente de mudança de requisitos não funcionais, como ambiente operacional (como hardware e software básico), mudanças de versão, linguagem e SGBD, que não impliquem em inserção, alteração ou exclusão de funcionalidades, ou seja, não há alteração de requisitos funcionais.

5.3.5. Dependendo do tipo de desenvolvimento as fases abaixo podem sofrer alteração. Por exemplo, para manutenções evolutivas ou adaptativas simples, não é necessário realizar as reuniões de revisão.

6. FASES DO PROJETO DE DESENVOLVIMENTO DE SOFTWARE

6.1. Fase de Iniciação

6.1.1. O processo se inicia com a área de negócio realizando a solicitação formal da demanda via chamado CSE (cse@enap.gov.br). A partir dessa solicitação é aberta uma Ordem de Serviço (OS). Nesta OS o Dono do Produto (PO) deverá definir, sempre que possível, a visão do projeto, que deve ser composta, no mínimo de informações como: público-alvo, problema(s) a ser(em) resolvido(s), nome do produto, tipo de produto, principal benefício.

6.1.2. Com base nessas informações e é feita uma avaliação inicial pela **Equipe Scrum**. Esta, então, prepara uma proposta de solução sucinta, contemplando informações *estimadas* sobre o escopo, previsão de custos e prazo de execução.

6.1.3. Essa proposta de solução é avaliada pelo Scrum Master (SM), que deve encaminhar para ciência ao Product Owner (PO).

6.1.4. O Product Owner (PO) ou seu substituto avalia se a solução apresentada, prazos e custos estão de acordo com a sua solicitação. Após esta avaliação do PO, ele deve dar ciência sobre os pontos apresentados na proposta de solução e devolver ao Scrum Master (SM).

6.1.5. De posse dessas informações o Scrum Master (SM) poderá encaminhar a demanda para desenvolvimento, onde se iniciará a fase de Planejamento do Projeto.

6.2. Atividades desenvolvidas

6.2.1. Realizar Solicitação da Demanda (Alterar processo).

6.2.2. Registrar / Classificar Demanda.

6.2.3. Abrir Ordem de Serviço (OS).

6.2.4. Preparar Proposta de Solução.

6.2.5. Analisar Proposta de Solução.

6.2.6. Encaminhar para Ciência do Demandante.

6.2.7. Avaliar Proposta de Solução.

6.2.8. Encaminhar Ciência à CGTI.

6.2.9. Analisar Retorno do Demandante.

6.2.10. Enviar para Desenvolvimento.

6.3. Prazo máximo

6.3.1. Após a solicitação ser realizada pela área de negócio a Fase de Iniciação terá um prazo máximo de cinco dias úteis para a sua conclusão.

6.4. Fase de Planejamento

6.5. É nesta fase que ocorrerá o levantamento de requisitos mais detalhado, observando as regras de negócio que serão necessárias para desenvolvimento da solução, discriminando o que fará parte do escopo do produto, bem como os itens de não escopo, determinando quais os limites do projeto.

6.6. O backlog do produto deverá listar itens que representem os potenciais incrementos do produto e que estejam alinhados aos benefícios definidos na visão. A priorização dos itens deverá considerar o que precisa ser validado mais rapidamente e o que mais se aproxima ao alcance da visão. Apenas os itens que estão no topo do backlog serão refinados, quebrando-os em pedaços menores, mas atômicos. Cada item deverá ser executado durante uma única Sprint e deverá entregar um incremento real. É importante lembrar que é melhor priorizar uma entrega pequena, mas com alto valor de retorno, do que uma grande entrega que contenha funcionalidades periféricas.

6.7. A partir do escopo do produto e determinando o seu backlog, deve-se realizar uma estimativa de custo e prazo para o PO ter ciência e aprovar antes da execução da sprint.

6.8. Nesta fase de Planejamento é importante ressaltar que devem ser descritos os critérios para Definição de Preparado dos itens de Backlog, bem como critérios para a Definição de Pronto.

6.9. A **Definição de Preparado** é um acordo formal entre Product Owner (PO) e Time de Desenvolvimento (TD) sobre o estado em que um item do Product Backlog deve estar para ser qualificado para discussão na reunião de Sprint Planning, ou seja, são critérios mínimos para que um determinado item seja encaminhado para desenvolvimento. Assim, caso um dos itens do backlog do produto não esteja totalmente aderente à Definição de Preparado, o item não poderá ser discutido na Sprint Planning.

6.10. A **Definição de Pronto** é um artefato Scrum usado para garantir a qualidade do produto desenvolvido a cada iteração (Sprint), ou seja, é a formalização entre os membros do Time Scrum e demais envolvidos do significa um produto "pronto" e potencialmente utilizável.

6.11. Atividades desenvolvidas

6.11.1. Realizar Planejamento do Projeto.

6.12. Prazo máximo

6.12.1. Após o projeto ser encaminhado para o desenvolvimento, a Fase de Planejamento terá um prazo máximo de dez dias úteis para a sua conclusão.

6.13. Fase de Execução (Sprints)

6.13.1. Nesta fase de Execução do Projeto acontecerá a implementação da demanda, que deverá seguir de acordo com o Planejamento realizado anteriormente e deverá ser realizada através da execução de Sprints.

6.13.2. As Sprints deverão começar imediatamente após finalizar a Fase de Planejamento.

6.14. Atividades desenvolvidas

6.14.1. Realizar Sprint Planning.

6.14.2. Realizar Sprint.

6.14.3. Realizar Sprint Review.

6.14.4. Realizar Sprint Retrospective.

6.15. Prazo máximo

6.15.1. Cada Sprint terá duração máxima de dez a vinte dias úteis. A equipe Scrum definirá o tamanho de cada Sprint para cada projeto. O tamanho da Sprint será definido com base na necessidade de entregas e/ou de feedback dos stakeholders. Sprints menores aumentam a agilidade do projeto e reduzem o risco de desenvolver um produto atenda adequadamente às necessidades.

6.15.2. Durante a realização de uma Sprint não será possível, em hipótese alguma, alterar o prazo da Sprint. A equipe de desenvolvimento deverá selecionar novos itens do Backlog da Sprint caso termine todo o trabalho antes do tempo.

6.16. Fase de Encerramento

6.16.1. Esta é a fase de revisão e encerramento formal do projeto, onde são colhidas as percepções de todos os envolvidos, visando melhorias para os próximos projetos. É também o momento de encerrar a demanda e colher o feedback por meio da pesquisa de satisfação encaminhada ao solicitante da demanda.

6.17. Atividades desenvolvidas

6.17.1. Encerrar Projeto.

6.17.2. Informar Demandante.

6.17.3. Realizar Pesquisa de Satisfação.

6.17.4. Encerrar Demanda.

6.18. Prazo máximo

6.18.1. Após a conclusão da etapa de execução do projeto de desenvolvimento de software, a Fase de Encerramento deverá ser concluída no prazo máximo de dez dias úteis.

6.18.2. O time Scrum poderá realizar, de acordo com a avaliação feita para cada projeto, a Fase de Encerramento por release, módulo ou conjunto de entregas.

7. ARTEFATOS SUGERIDOS

7.1. A utilização dos quadros Canvas como ferramenta de análise e criação de novos modelos de negócio tem com o objetivo criar, capturar e entregar valor. E a partir desta visão, o objetivo é documentar os projetos, considerando o que de fato é importante nesse acompanhamento.

7.2. O Canvas é uma ferramenta estratégica de gestão de negócios e de implantação de novas frentes, desde que conte com um planejamento para evitar desvios no meio do caminho. O principal objetivo do Canvas é fazer com que você planeje um negócio de sucesso de forma pouco burocrática, tendo diversos insights e ideias durante o processo.

7.3. Diante do exposto e visando manter o equilíbrio entre modelo Ágil e a adoção de padrões para o acompanhamento de projetos na CGTI Enap, temos abaixo as questões que devem ser respondidas para o desenvolvimento dos projetos.

7.3.1. O que é o Produto do Projeto?

7.3.2. Por que e Para que preciso desse Produto?

7.3.3. Como será desenvolvido esse Produto?

7.4. Respondendo a essas questões dá-se início a montagem do quadro onde deverão ser especificados os requisitos do Projeto, seguindo o modelo apresentado na figura abaixo:



Projeto:

Data:

| COMO? | | O QUE? | PARA QUEM? |
|---------------------|-------------------|-------------|------------------|
| Parcerias | Atividades Chave: | Entregáveis | Áreas de Negócio |
| Estrutura de Custos | | QUANTO? | |

7.5. Cabe ressaltar que o Product Owner, que é o Dono do Produto, deve especificar a necessidade e descrever de forma clara e objetiva a que se destina o produto e como seu desenvolvimento irá contribuir na melhoria do seu fluxo de trabalho e consequentemente como irá atingir aos clientes da Enap.

7.6. Em casos de mudanças/evoluções de produto ou projetos que serão utilizados apenas por servidores da Enap, descrever de que forma o produto irá contribuir para a melhoria do processo de trabalho da área de negócio.

8. RESPONSABILIDADES

8.1. As responsabilidades das atividades são distribuídas conforme tabela a seguir:

| Papéis | Responsabilidades | | | |
|-----------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------|
| | Iniciação | Planejamento | Execução | Encerramento |
| Product Owner (PO) Responsável: Servidor da Área de Negócio | <ul style="list-style-type: none"> Realizar Solicitação da Demanda; Avaliar Proposta de Solução; Encaminhar Ciência à CGTI. | <ul style="list-style-type: none"> Criar Backlog do Produto; Criar, com o Time de Desenvolvimento, a Definição de Pronto; Criar com o Time de Desenvolvimento, a Definição de Preparado. | <ul style="list-style-type: none"> Participar da Sprint Planning; Participar da Sprint Review; Participar da Sprint Retrospective; Revisar o Detalhamento dos Requisitos (Histórias de usuários, fluxos, protótipos, etc); O PO pode ter o papel de ser o fiscal requisitante dentro do Projeto. Ele deve definir as regras de negócios e apoiar a equipe Scrum; Deve realizar a aprovação da demanda, informando possíveis correções e melhorias; Deve autorizar a entrada da mudança em ambiente de produção, formalizando a solicitação, sugerindo data e horário para a alteração. | <ul style="list-style-type: none"> Responsável pela formalização de encerramento da demanda ou projeto. |
| Scrum Master (SM) Responsável: Servidor da CGTI | <ul style="list-style-type: none"> Abrir Ordem de Serviço; Analisar Proposta de Solução; | <ul style="list-style-type: none"> Realizar Planejamento do Projeto. | <ul style="list-style-type: none"> Participar da Sprint Planning; Participar da Sprint Review; Participar da Sprint Retrospective; Criar o Documento de Revisão da Sprint; | <ul style="list-style-type: none"> Encerrar Projeto; Informar Demandante; Realizar Pesquisa de Satisfação; |

| | | | | |
|------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | <ul style="list-style-type: none"> • Encaminhar para Ciência do Demandante; • Analisar Retorno do Demandante; • Enviar para Desenvolvimento. | <ul style="list-style-type: none"> • Criar Estrutura Analítica do Projeto - EAP; • Criar Contagem Estimada (PF ou UST); • Criar Cronograma de Sprints; • Criar Plano de Gestão de Mudanças; • Criar Plano de Comunicação. | <ul style="list-style-type: none"> • Criar o Documento de Retrospectiva da Sprint. | <ul style="list-style-type: none"> • Encerrar Demanda; • Criar Documento de Lições Aprendidas; • Criar Documento de Transferência de Conhecimento. |
| Time de Desenvolvimento (TD) Responsável: Definido pela CGTI | <ul style="list-style-type: none"> • Preparar Proposta de Solução. | <ul style="list-style-type: none"> • Criar Documento de Visão; • Criar Especificação Técnica • Criar Desenho da Arquitetura da Solução; • Criar Plano de Liberação de Releases; | <ul style="list-style-type: none"> • Participar da Sprint Planning; • Participar da Sprint Review; • Participar da Sprint Retrospective; • Criar Backlog da Sprint; • Realizar Sprint; • Criar o Detalhamento dos Requisitos (Histórias de usuários, fluxos, protótipos, etc). | |

8.2. Importante ressaltar que eventualmente o Scrum Master poderá ser o fiscal técnico do contrato e o Product Owner poderá ser o fiscal requisitante.

8.3. O time de desenvolvimento (Time Scrum) pode ser composto pelos papéis descritos abaixo. Para cada tipo de projeto será definida a equipe que deverá participar do Time Scrum.

- 8.3.1. Gerente de Projetos.
- 8.3.2. Desenvolvedores.
- 8.3.3. Arquiteto de UX.
- 8.3.4. Arquiteto de Dados.
- 8.3.5. Engenheiro de Dados.
- 8.3.6. Cientista de Dados.
- 8.3.7. Administrador de Banco de Dados.
- 8.3.8. Designer.
- 8.3.9. Analista de Teste.

9. REVISÃO DA METODOLOGIA DE DESENVOLVIMENTO DE SOFTWARE

9.1. Prazo para revisão da Metodologia

9.1.1. A revisão desta Metodologia de Desenvolvimento de Software seguirá o mesmo fluxo ágil do Scrum. Portanto, após aprovação deste documento, será realizada uma primeira Sprint de 60 dias, seguindo todo o processo de Scrum.

9.1.2. As Sprints de revisão poderão ser realizadas a cada 60 dias ou quando houver necessidade.

9.2. Processo de Revisão da Metodologia

9.2.1. Durante a primeira Sprint de 60 dias será disponibilizado um quadro na ferramenta de gestão de projetos ágeis da Enap, onde toda equipe CGTI poderá sugerir ações de melhoria da Metodologia.

9.2.2. O comitê responsável pela revisão da Metodologia deverá avaliar cada sugestão de melhoria disponibilizada no quadro e, caso a ação seja aprovada, ela poderá percorrer as listas do trello até a inclusão do item na próxima versão da Metodologia.

9.2.3. Importante ressaltar que, após o prazo da Sprint, serão realizadas as reuniões de Sprint Review e Retrospective, visando o feedback de todos os que se envolveram em projetos utilizando a metodologia para que, a partir dessas informações, a Metodologia possa ser revisada e adequada ao melhor modelo para Enap.

9.2.4. Após o fluxo da Sprint e com a Metodologia revisada e aprovada, esta poderá ser disponibilizada para novo ciclo de utilização.

10. BIBLIOGRAFIA DE REFERÊNCIA

- 10.1. SCHWABER, Ken; SUTHERLAND, Jeff. Guia SCRUM. SCRUM.org, outubro de 2017.
- 10.2. MAGNO, Alexandre. Tire seu projeto do papel com Scrum. Editora Leya, 2019.