MDS - Metodologia de Desenvolvimento de *Software Àgil do MME*

Versão 2.2

HISTÓRICO DE REVISÕES

Data	Versão	Descrição	Autor
20/11/2022	2.0	Criação metodologia ágil	Carlos Gomes
13/05/2024	2.1	Atualização metodologia ágil	Carlos Gomes
23/05/2024	2.2	Ajustes finos e revisão final	Luiz Alexandre

Sumário

HISTÓRICO DE REVISÕES 2
METODOLOGIA DE DESENVOLVIMENTO DE SOFTWARE ÁGIL 4
1. *PROPÓSITO* 4
2. *OBJETIVOS* 4
3. *APLICABILIDADE* 4
4. ***POLÍTICA DE REVISÃO*** 5
5. ***DEFINIÇÕES*** 5
6. *PAPÉIS E RESPONSABILIDADES* 6
7. *PROCESSO DE DESENVOLVIMENTO DE SOFTWARE* 11
8. *FLUXOS E PROCESSOS MAPEADOS* 17

METODOLOGIA DE DESENVOLVIMENTO DE *SOFTWARE ÁGIL*

1. PROPÓSITO

A Metodologia de Desenvolvimento de Sistemas – MDS, define os processos relativos ao desenvolvimento, sustentação e implantação de sistemas com aplicação de práticas ágeis, sob gestão e controle da Coordenação Geral de Inovação e Transformação Digital – CGIT, da Subsecretaria de Tecnologia e Inovação – STI, da Secretaria Executiva – SE do Ministério de Minas e Energia.

A presente MDS utiliza como base outras abordagens consolidadas no mercado, integrando valores e métodos ágeis com boas práticas de engenharia de software, gerenciamento de projetos e governança de Tecnologia da Informação. São priorizadas as metodologias e frameworks ágeis, como o Scrum e Lean-IT, permite ainda a utilização de outras metodologias e ferramentas para ótimização do processo, a exemplo do Kanban. Esta MDS foi customizada para melhor atender às necessidades do Ministério de Minas e Energia – MME por meio da CGIT.

O objetivo é estabelecer o Modelo de Execução do objeto com a definição de papéis, artefatos, fluxos de trabalho, esforço por fase e atividades inerentes a prestação de serviços no âmbito do órgão.

2. OBJETIVOS

Padronizar os processos de trabalho, as funções desempenhadas e as tarefas a serem realizadas por todos os envolvidos no processo de desenvolvimento, sustentação e implantação de aplicações de software no âmbito do MME.

a) Definir os papéis e responsabilidades dentro do processo de desenvolvimento de *software*;
b) Descrever o fluxo de atividades a serem executadas durante o desenvolvimento de *software*;
c) Definir os produtos de trabalho gerados durante o processo de desenvolvimento;
d) Padronizar os documentos gerados durante o processo de desenvolvimento de *software*;
e) Definir o ciclo de desenvolvimento adotado pelo órgão;
f) Especificar o modelo de execução das Ordens de Serviços.

3. APLICABILIDADE

As diretrizes, processos e atividades estabelecidos nesta metodologia são aplicáveis a todos os sistemas desenvolvidos e/ou mantidos pela CGIT, promovendo, assim, padronização, eficiência, eficácia e efetividade do processo de desenvolvimento, sustentação e implantação de aplicações de software.

4. POLÍTICA DE REVISÃO

Este documento poderá ser atualizado pela CGIT/CODEV sempre que houver necessidade de atualização, seja por motivo de adequação aos normativos institucionais, diretrizes gerenciais, recomendações de órgãos de controle, revisão do texto para eliminar eventuais ambiguidades, omissões ou contradições ou ainda para o melhor desempenho do objeto. Eventuais atualizações devem ser comunicadas as CONTRATADAS do objeto.

5. DEFINIÇÕES

a) ***Software*** – é uma sequência de instruções escritas para serem interpretadas por um computador com o objetivo de executar tarefas específicas.

b) ***Build*** **-** versão "*compilada*" de um *software* ou parte dele que contém um conjunto de recursos que poderão integrar o produto final;

c) **Iteração** - é o processo chamado na programação de repetição de uma ou mais ações;

d) ***Multi-thread*** - Linha ou encadeamento de execução. É uma forma de um processo dividira si mesmo em duas ou mais tarefas que podem ser executadas concorrentemente;

e) ***Feedback*** - consiste no provimento de informação a uma pessoa sobre o desempenho, conduta, ou ação executada por esta, objetivando reorientar ou estimular comportamentos futuros mais adequados;

f) ***Baseline*** - gerenciamento de configuração que ajuda a controlar as mudanças realizadas nos itens de configuração de hardware ou de software;

g) ***Release*** – consiste em uma liberação pública de uma nova versão de um produto de software;

h) ***Desenvolvimento*** – compreende as atividades de desenvolvimento e evolução (manutenções evolutivas, perfectivas, adaptativas e integrativas) de aplicações de software em plataformas web, desktop e mobile, que englobam a análise e o levantamento de requisitos, a construção e atualização de artefatos de documentação e do código-fonte da aplicação, bem como a execução de testes funcionais;

i) ***Sustentação*** – compreende as atividades realizadas para a manutenção da disponibilidade, estabilidade e desempenho das aplicações de software, tais como manutenções corretivas, investigação de incidentes, resolução de problemas e apoio aos usuários, bem como atividades de suporte à manipulação de dados, monitoramento e operação das aplicações e seus componentes, compreendendo todos os ambientes em que a solução esteja instalada;

j) ***Implantação*** – compreende as atividades de preparação de ambientes de aplicação de software, além das atividades de inserção desse sistema no rol soluções de aplicações, tais como planejamento, parametrizações, configurações, treinamentos técnicos, funcionais e operacionais, além de apoio na intermediação das evoluções, apoio no endomarketing e na elaboração de materiais de apoio à utilização do sistema (manuais, vídeos, orientações etc.).

6. PAPÉIS E RESPONSABILIDADES

Os papeis e responsabilidades a seguir são uma linha base de atuação. Estes não impedem que eventuais condutas proativas sejam realizadas em acordo com as melhores práticas ágeis.

Os trabalhos da Coordenação de Desevolvimento de Soluções – CODEV da Coordenação Geral deInovação e Transformação Digital - CGIT estão organizados da seguinte forma:

Responsáveis	Atividades
Área Requisitante	• Informar os requisitos do sistema; • Testar o sistema desenvolvido; • Homologar os produtos do projeto; • Informar eventuais mudanças no negócio que possam impactar no produto desenvolvido; • Participar do ciclo de desenvolvimento do incremento do produto (Planning, Daily ou quando houver eventuais solicitações); • Dar o aceite final ao produto desenvolvido; • Priorizar o desenvolvimento de TI alinhado ao negócio, tanto de forma estática, a partir das estratégias e dos planos de negócio do MME, como dinamicamente, fazendo ajustes contínuos em virtude do surgimento de novas oportunidades de negócio, sendo a TI um ator importante para a geração de valor para o negócio.
Coordenador de Desenvolvimento de Soluções	• Acompanhar e controlar gerencialmente todos os processos da CODEV; • Atender as demandas de sistemas, conforme priorização do Plano Diretor de Tecnologia da Informação e Comunicação - PDTIC; • Captar novos projetos junto às áreas de negócio; • Participar de reuniões de início de Sprints/Projeto; • Manter base de informações administrativas e financeiras da CODEV – Controle de empenho, renovação contratual; • Avaliar os relatórios gerenciais dos processos da CODEV; • Avaliar o relatório de acompanhamento dos projetos da CODEV; • Registrar ocorrências em apoio aos fiscais na execução contratual.

Gestor do Contrato/Fiscais Técnicos	• Acompanhar e controlar gerencialmente todos os projetos da CODEV; • Participar de reuniões de projetos; • Garantir o desenvolvimento de projetos; • Manter os padrões de desenvolvimento de projetos da CODEV; • Manter os relatórios gerenciais dos projetos da CODEV; • Planejar e monitorar as atividades do projeto; • Definir perfil dos recursos necessários para o projeto; • Avaliar o tamanho e a complexidade do projeto e planejar de forma incremental a execução. • Coordenar as entregas com a área de negócio; • Manter a equipe alinhada com os objetivos do projeto.
Qualidade	• Implantar metodologia de testes de *software* com o objetivo de aperfeiçoaros processos tecnológicos no âmbito do Ministério; • Promover o alinhamento da TI ao negócio (suas estratégias e objetivos); • Definir, estabelecer e monitorar o ambiente de gerenciamento de configuração de todos os projetos da CGIT; • Apoiar na elaboração das estimativas de tamanho e complexidade do projeto; • Elaborar, implementar e atualizar o Plano de qualidade dos projetos da CGIT; • Elaborar e manter atualizada a Lista de Verificação dos produtos; • Verificar a aderência dos cronogramas planejados com o que foi executado em cada projeto; • Realizar verificações de atividades e produtos; • Inspecionar o desenvolvimento em frequência suficiente para antecipar problemas de entrega na Sprint; • Reportar resultados; • Monitorar as ações corretivas e preventivas; • Monitorar as atividades do fornecedor; • Implantar e manter ferramentas que auxiliem na execução das atividades de elaboração, acompanhamento, monitoramento e controle dos produtos/serviços inerentes ao desenvolvimento de software; • Aferir a qualidade das entregas de software; • Atuar de maneira proativa em relação a prospecção de problemas no ciclo de desenvolvimento; • Implantar e gerenciar o processo de integração contínua;

	• Aplicar política de administração de dados, visando à integridade, qualidade, segurança, rapidez e disponibilidade das informações dos sistemas corporativos do Ministério; • Validar o Modelo de Dados do projeto; • Apoiar na goverannça de dados corporativos; • Avaliar os procedimentos armazenados; • Verificar se as melhores práticas para o desenvolvimento de solução de BI e suas estruturar foram utilizadas.
Time Scrum	
Product Owner (Dono do Produto)	Responsável por maximizar o valor do produto e do trabalho do Time de Desenvolvimento. É responsável também por gerenciar o Backlog do Produto, são suas tarefas: • Comunicar-se com todas as partes interessadas; • Gerênciar a lista de pendências do produto (backlog do produto), por meio de sistema fornecido pela CONTRATADA; • Priorizar as necessidades de entrega; • Supervisionar as etapas de desenvolvimento; • Antecipar as necessidades da área com o apoio do Analista de requisitos contido no time de desenvolvimento; • Avaliar o andamento do produto.
Scrum Master	Responsável por garantir que o Scrum seja entendido e aplicado. O Scrum Master faz isso para garantir que o Time Scrum possa aderir à teoria, práticas e regras do Scrum, são tarefas do Scrum Master: • Encontrar técnicas para uma gestão eficaz do Product Backlog; • Comunicar claramente a visão, metas e itens do Product Backlog para a equipe de desenvolvimento; • Ensinar a equipe a criar itens de Backlog claros e sucintos; • Abranger o planejamento de longo prazo do produto em um ambiente empírico; • Compreender e praticar a agilidade; • Facilitar eventos Scrum conforme solicitado ou necessário; • Ensinar e liderar a equipe de desenvolvimento para criar produtos de alto valor; • Remover impedimentos para o progresso do desenvolvimento; • Auxiliar a equipe de desenvolvimento em ambientes organizacionais quando o processo Scrum ainda não estiver totalmente aprovado e compreendido; • Treinar a organização na sua adoção ao processo Scrum; • Planejar e apoiar implementações do Scrum dentro da organização;

	• Ajudar os funcionários e as partes interessadas a entender e aprovar o Scrum de forma empírica; • Fomentar transformações que aumentem a produtividade da equipe Scrum; • Aumentar a eficácia da aplicação do Scrum na organização.
Time de Desenvolvimento	Consiste em profissionais que realizam o trabalho de entregar uma versão usável que potencialmente incrementa o produto "Pronto" ao final de cada Sprint, tem como papel: • Programar os componentes; • Realizar testes unitários; • Aplicar política de administração de dados, visando à integridade, qualidade, segurança, rapidez e disponibilidade das informações dos sistemas corporativos do Ministério; • Administrar os bancos de dados e dar suporte (criar scripts) às permissões de acesso dos sistemas corporativos, garantindo o bom desempenho de sua operação; • Validar o Modelo de Dados do projeto; • Avaliar os procedimentos armazenados; • Aplicar as melhores práticas para o desenvolvimento de solução de BI; • Estruturar as soluções de Business Intelligence - BI; • Avaliar e acompanhar requisitos; • Delimitar o sistema e definir suas funcionalidades e regras; • Fornecer suporte ao Product Owner sobre o entendimento dos requisitos • Informar do impacto da ausência ou existência de requesito que seja nocivo a funcionalidade ou não esteja de acordo com as melhores práticas de UX do mercado; • Informar fluxos ou funcionalidades incompletas (ex. Ausência de botão voltar ou outra opção de navegação, ausência de botão de logoff, ausencia de CRUD de dados, dentre outras situações que o usuário não tem experiência); • Definir e avaliar as atividades e produtos técnicos do projeto; • Definir e avaliar a estrutura geral de cada visão da arquitetura do *software*; • Planejar a integração dos componentes de *software* para criar um *build;* • Identificar e definir os testes necessários para o projeto; • Monitorar a abrangência dos testes e avaliar a qualidade geral obtida ao testar o produto de Software; • Desenvolver os roteiros de testes;

	• Conduzir os testes necessários; • Registrar os resultados desses testes; • Executar os testes; • Registrar os resultados dos testes.

7. PROCESSO DE DESENVOLVIMENTO DE *SOFTWARE*

O processo de desenvolvimento de *software* descreve o fluxo de atividades e tarefas a serem executadas, com o objetivo de transformar necessidades de usuários e de sistemas em produto final de *software.* Para projetos de desenvolvimento de aplicações de software, será utilizado o framework Scrum como referência de práticas ágeis.

7.1. Pilares

- **Transparência:** aspectos significativos do processo devem estar visíveis aos responsáveis pelos resultados. Esta transparência requer aspectos definidos por um padrão comum para que os observadores compartilharem um mesmo entendimento do que está sendo visto;
- **Inspeção:** os usuários Scrum devem, frequentemente, inspecionar os artefatos Scrum e o progresso em direção a detectar variações. Esta inspeção não deve, no entanto, ser tão frequente que atrapalhe a própria execução das tarefas. As inspeções são mais benéficas quando realizadas de forma diligente por inspetores especializados no trabalho a se verificar;
- **Adaptação:** se um inspetor determina que um ou mais aspectos de um processo desviou para fora dos limites aceitáveis, e que o produto resultado será inaceitável, o processo ou o material sendo produzido deve ser ajustado. O ajuste deve ser realizado o mais breve possível para minimizar mais desvios.

7.2. Valores

- **Coragem:** os integrantes de um projeto Scrum (Scrum Master + Product Owner + Time de Desenvolvimento) precisam ter coragem para fazer a coisa certa e trabalharem juntos removendo impedimentos, buscando soluções.
- **Foco:** os integrantes de um projeto Scrum precisam focar no trabalho durante o Sprint e nas metas designadas pela empresa. Time disperso perde produtividade e não alcança os objetivos.
- **Comprometimento:** cada integrante de um projeto Scrum se compromete com o trabalho que se responsabilizou em fazer. Se comprometer é se envolver e não abandonar o trabalho pela metade ou entregar sem qualidade.
- **Respeito:** respeitar uns aos outros é o primeiro passo para manter a colaboração, a integração e bom ambiente de trabalho.
- **Abertura:** poder ser franco, expor as ideias e propostas mesmo que elas não sejam aproveitadas. Promover momentos de debates, discussões, ouvir opiniões e

sugestões para gerar novas práticas.

7.3. Time Scrum

- **Product Owner (Dono do Produto):** responsável por maximizar o valor do produto e do trabalho do Time de Desenvolvimento. É responsável também por gerenciar o Backlog do Produto.
- **Scrum Master**: responsável por garantir que o Scrum seja entendido e aplicado. O Scrum Master faz isso para garantir que o Time Scrum possa aderir à teoria, práticas e regras do Scrum.
- **Time de Desenvolvimento:** consiste em profissionais que realizam o trabalho de entregar uma versão usável que potencialmente incrementa o produto "Pronto" ao final de cada Sprint.

O tamanho ideal do Time de Desenvolvimento é ser pequeno o suficiente para se manter ágil e grande o suficiente para completar um trabalho significativo dentro da Sprint. Menos de três integrantes no Time diminuem a interação e resultam em um menor ganho de produtividade. Times menores podem encontrar restrições de habilidades durante a Sprint, sendo incapaz de entregar um incremento potencialmente utilizável. Havendo mais de nove integrantes é exigida muita coordenação. Times grandes geram muita complexidade para que um processo empírico seja útil. Os papéis de Product Owner e de Scrum Master não são incluídos nesta contagem, a menos que eles também executem o trabalho do Backlog da Sprint.

Eventos

- **Sprint:** um *time-boxed* de uma a quatro semanas, durante o qual um "Pronto", versão incremental potencialmente utilizável do produto, é criado. Sprints tem durações coerentes em todo o esforço de desenvolvimento. Uma nova Sprint inicia imediatamente após a conclusão da Sprint anterior.
- **Planejamento de Sprint:** reunião de planejamento da Sprint, na qual são definidos o objetivo, as entregas e o trabalho necessário para entregar o incremento da Sprint. Para cada item do Backlog da Sprint, é necessário obter do Dono do Produto a definição de pronto do item, assim não há dúvidas na hora de receber o produto previsto no item de trabalho planejado. No campo Descrição do item de backlog, deve-se incluir uma pequena descrição do item e a definição de pronto do item (o critério que será usado para aceitar o item ao final da sprint), essa descrição deve estar em linguagem entendida pelo Dono do Produto, que aceitará ou não o incremento entregue. A emissão do Termo de Aceitação da Sprint deve ser realizada imediatamente após a reunião de encerramento e depende grandemente da definição objetiva da descrição e da definição de pronto. O tempo máximo é de 08 horas, no modelo atual indicamos o tempo de 02 horas de planning para cada semana de Sprint.
    - Ex 01. Sprint 04 semanas, deve durar até 08 horas;
    - Ex 02. Sprint 02 semanas, deve durar até 04 horas.
- **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 realizada para inspecionar o trabalho desde a última Reunião

Diária, e prever o trabalho que deverá ser feito antes da próxima Reunião Diária.

- **Revisão da Sprint:** reunião executada ao 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 nos próximos itens que podem ser feitos para otimizar valor. O Tempo máximo da revisão da Sprint é de 04 horas.
- **Retrospectiva da Sprint:** 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 reunião ocorre depois da Revisão da Sprint e antes da reunião de planejamento da próxima Sprint. O tempo máximo da retrospectiva da Sprint é de 03 horas.

Artefatos

- **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.
- **Backlog da Sprint:** 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. O Backlog 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".
- **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. Ao final da Sprint um novo incremento deve estar "Pronto", o que significa que deve estar na condição utilizável e atender a definição de "Pronto" do Time Scrum. Este deve estar na condição utilizável independente do Product Owner decidir por liberá-lo realmente ou não.

NOTA: Quando o Scrum é utilizado, um incremento potencialmente utilizável é entregue ao final de cada Sprint. Isso não significa que a entrega realizada é útil para os usuários. Portanto, muitas vezes há necessidade de lotes de incremento utilizáveis em Sprint subsequentes para que algo potencialmente útil seja entregue aos usuários. Um incremento utilizável é aquele sem nenhum trabalho "não feito" associado a ele no final de um Sprint. Se as equipes não conseguirem criar incrementos realmente utilizáveis, o trabalho restante (teste, documentação, aceitação do usuário etc.) será feito como parte do processo (às vezes chamado de fase de estabilização), potencializando a aceitação das más práticas de desenvolvimento.

7.4. Kanbam

É uma metodologia que serve para definir, gerenciar e melhorar o fluxo de desenvolvimento ou de entrega de serviço que visam disponibilizar o produto de trabalho, como por exemplo um produto ou software. Essa métodologia utiliza o quadro de gerenciamento visual para demonstrar como o serviço/trabalho está sendo realizado atualmente. No MME poderá ser aplicada um

software que auxilie no uso do Kanban, como por exemplo, Trello, Task's to do (planner) ou Tai.go ou outros, devendo esta solução ser definido à critério dos servidores da CGIT.

Valores

- **Transparência:** Compartilhar informações melhora efetivamente o fluxo de valor de negócio;
- **Equilíbrio:** A compreensão de que diferentes aspectos, pontos de vista e habilidades devem ser equilibrados para a eficácia;
- **Colaboração:** Trabalho em conjunto;
- **Foco no Cliente:** Conhecendo o objetivo para o sistema. Os clientes e o valor que recebem são o foco natural no Kanban;
- **Fluxo:** A conscientização que o trabalho é um fluxo de valor, seja ele contínuo ou esporádico;
- **Liderança:** Habilidade de inspirar os outros a agir através do exemplo, palavras e reflexão;
- **Compreensão:** Principalmente o autoconhecimento (tanto do indivíduo como da organização) a fim de avançar;
- **Acordo:** Compromisso de avançar juntos em direção a metas, respeitando sempre que possível, adaptando-as diferenças de opinião ou abordagem;
- **Respeito:** Valorizando, entendendo e mostrando consideração pelas pessoas.

Princípios Fundamentais

- **Gestão de Mudanças:**
  - Comece pelo que você faz agora:
    - Compreender os processos atuais, como eles de fato são praticados e respeitar os papeis, responsabilidades e cargos existentes
  - Concordar em prosseguir a melhoria através da mudança evolutiva;
  - Incentivar atos de liderança a todos os níveis – desde o colaborador até à alta gerência.
- **Entrega de serviços:**
  - Compreender e focar nas necessidades e expectativas dos clientes;
  - Gerenciar o trabalho; deixar que as pessoas se auto organizem em torno dele;
  - Desenvolver políticas para melhorar os resultados.

Sistemas de Fluxo

No Kanban o fluxo é um sistema de fluxo no qual os itens de trabalho fluem através de várias etapas de um processo, ordenados da esquerda para a direita. Para que esse sistema de fluxo funcione corretamente, várias condições devem existir.

Primeiramente, devem existir sinais visuais para limitar o trabalho em progresso (WiP). Esses sinais são combinações das cartas (tarefas), limite do trabalho em progresso e coluna que representa a atividade. Também deve ter identificado ponto de compromisso e entrega, que esses

são itens considerados completos. O compromisso é um acordo explícito ou implícito entre o cliente e o serviço (equipe ou prestador).

Práticas Gerais

Visualizar: O ato de tornar o trabalho e as políticas visíveis – seja em quadro de parede, em displays eletrônicos, ou outros meios – é o resultado do entendimento do sistema atual e a possibilidade de encontrar áreas potenciais para melhoria. Uma boa prática para visualizar as políticas seria por exemplo, colocar resumos entre as colunas do que deve ser feito antes que os itens se movam de uma coluna para a seguinte. O design do quadro visual vária muito entre os sistemas Kanban, cada um deve representar da melhor forma o fluxo de trabalho. O design do cartão ou painel que descreve o item de trabalho (tarefa) é outro aspecto essencial da visualização. Importante destacar visualmente quando o item está bloqueado (impedido) por dependências de outros serviços ou por outras razões.

Limite do Trabalho em Progresso (WiP): Introduzir e respeitar limites no WiP faz que o sistema que esteja com característica de "empurrado" se torne "puxado", fazendo que novos itens não sejam iniciados até que o trabalho seja concluído dos itens que estão em outras colunas seguintes. Ter muito trabalho parcialmente completo é um desperdício, por aumentar o tempo de espera, impedindo a organização de ser sensível aos seus clientes e à mudança de circunstâncias e oportunidades. Observar, limitar e, em seguida, otimizar a quantidade de trabalho em progresso é essencial para o sucesso do Kanban, por melhorar o tempo de espera para serviços, melhor qualidade e uma maior taxa de entregas.

Gerenciar o Fluxo: O fluxo deve maximizar a entrega de valor e minimizar o tempo de espera. Faz necessário um controle empírico através de transparência, inspeção e adaptação. Os gargalos, onde o fluxo é limitado por sub processos específicos, e os bloqueadores, onde existem dependências de outros serviços, são importantes para registrar e gerenciar.

Tornar as Políticas Explícitas: O processo descrito como fluxo de trabalho e políticas cria restrições na ação, potencializa as restrições, e resulta em características emergentes que podem ser ajustadas por testes. As políticas de processo precisam ser raras, simples, bem definidas, visíveis, sempre aplicadas e facilmente modificadas por aqueles que fornecem o serviço. É uma

boa prática, caso for necessário, depois da definição do WiP desafiar, mudar ou quebrar os limites para ver se diferentes limites em diferentes circunstâncias melhoram os resultados.

Uma prática fundamental é tornar explícitas as políticas que se aplicam aos serviços e a existência de um mecanismo viável e simples.

O WiP não é a única política no Kanban, existem outras políticas adicionais como por exemplo: atribuição de capacidade e equilíbrio, “Definição de Feito”, itens de trabalho que saem das etapas de um processo, reposição para selecionar novos trabalhos quando a capacidade tiver disponível, etc.

Implementar ciclos de Feedback: O ciclo de feedback é importante para qualquer processo controlado e servem para mudanças evolutivas. Feedbacks são reuniões cíclicas e as revisões que impulsionam a mudança evolutiva e a entrega efetiva de serviços. O tempo entre os feedbacks (se refere ao período entre revisões) depende do contexto e é crucial para bons resultados. Revisões muito frequentes podem gerar mudanças antes de ver o resultado de mudança antes de ver o resultado de mudanças anteriores, mas se eles não são frequentes o suficiente, um desempenho fraco pode persistir mais do que o necessário.

Melhoria contínua: Kanban é fundamentalmente um método de melhoria. Não há ponto final dos processos de mudança, pelo motivo que a perfeição é inatingível. Por vezes, a utilização de abordagem empíricas e pragmáticas é uma forma apropriada de encontrar a maior adequação para o propósito no ambiente atual.

7.5. Padrão de História de Usuário

Todos os itens do Backlog, independente do tipo da demanda devem seguir o padrão de História de Usuário e Critérios de Aceitação aqui definidos.

- Toda História de Usuário deverá possuir ao menos 1 (um) Critério de Aceitação;
- Se a História do Usuário tiver características de Épico, deve ser dividida em novas histórias;
- Ao escrever a história, deve-se utilizar no mínimo:
  - Enquanto [Papel/função] quero [o que deseja] para que [finalidade/benefício].

- Ao escrever as histórias de usuário, os critérios de aceitação devem ser usados para validá-las e devem seguir o seguinte padrão:
  - [NNN] - Dado que [cenário], quando [ação do usuário] então [comportamento esperado].
- Também poderá ser usado o E/OU/SE para refinar ainda mais os cenários de teste.
- O Exemplo de História de usuário:
  - Enquanto vendedor quero consultar os meus clientes pelo CNPJ para conseguir negociar com ele.
    - Exemplo de Critérios de Aceitação para a história proposta:

✓ 001 - **Dado que** o vendedor consulta o cliente
E insere um CNPJ inválido
quando apertar no botão "Consultar"
OU clicar fora do campo de CNPJ
então o sistema deve apresentar uma mensagem informando "CNPJ Inválido, por favor, coloque um CNPJ válido.".

✓ 002 - Dado que o vendedor consulta o cliente
E insere um CNPJ válido
quando apertar no botão "Consultar"
então o sistema deve apresentar uma mensagem informando os dados do cliente.

O método proposto neste documento foi dividido processos macros que se comunicam para obtenção de melhores resultados no desenvolvimento de software.

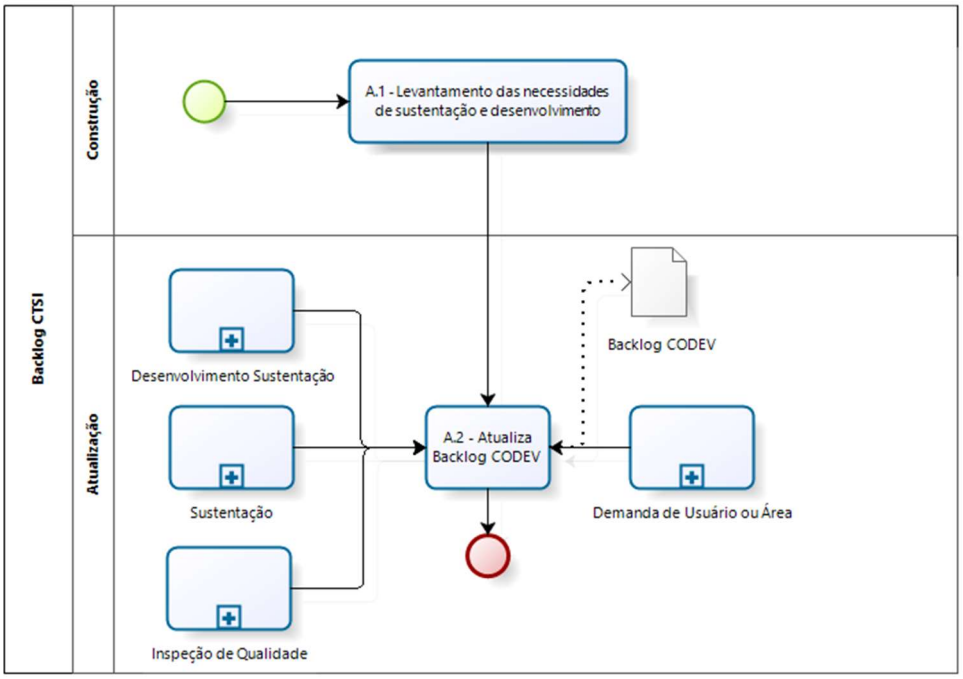
Cada processo pode ser repetido em ciclos de desenvolvimento, é possível que um processo possa iterar até um limite definido no mesmo. De maneira geral o ciclo de desenvolvimento é ordenado por Ordem de Serviço – OS.

8. FLUXOS E PROCESSOS MAPEADOS

A Metodologia de Desenvolvimento de Software é representada a seguir por meio de uma cadeia de processos que incluem todo o ciclo de atividades macro no desenvolvimento e sustentação de software no MME. Cada processo será demonstrado graficamente com a explicação de cada atividade. Considera-se ainda que estes processos podem ser ajustados no decorrer do Contrato, quando justificada a melhoria da eficiência e eficácia das entregas de software.

8.1. Backlog CODEV

O processo Backlog CODEV define as principais atividades de elicitação e demandas de desenvolvimento e sustentação na CGIT/CODEV. Tem como objetivo fornecer uma organização geral dessas demandas e sua respectiva atualização durante a execução dos processos de desenvolvimento de software. Este processo deve ser iniciado antes dos demais processos dessa metodologia.



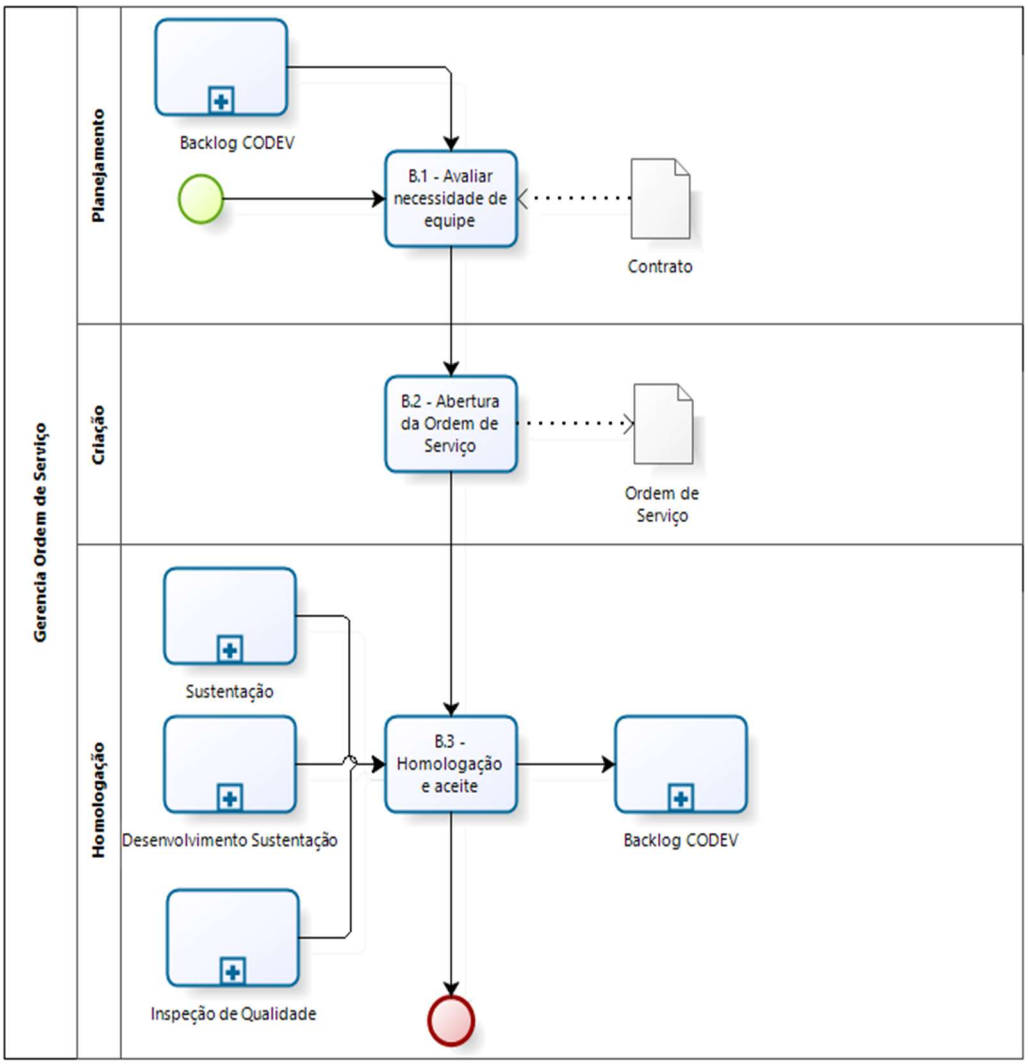
O Backlog CODEV é dividido em duas fases. A fase de Construção, considera a atividade de Levantamento das necessidades de sustentação e desenvolvimento. Nessa fase é realizada toda a avaliação das necessidades de densenvolvimento do ano corrente. É desejável que normativos vigentes e áreas diversas do Ministério de Minas e Energia sejam consultados, para a verificação inicial dessas demandas. A fase de Atualização, corresponde a tarefa recorrente de atualização do Backlog CODEV e considera os resultados dos processos de Desenvolvimento Sustentação, Sustentação e Inspeção de Qualidade. Este processo cria e mantém o Backlog CODEV.

É importante destacar que solicitações da diversas áreas que representem novo sistema ou funcionalidade, devem partir de demanda formal da área assinada por seu representante e os respectivos pontos focais dessa área.

8.2. Gerencia Ordem de Serviço – Desenvolvimento

O processo Gerencia Ordem de Serviço, avalia a necessidade de alocação das equipes. Indica-se que exista uma OS de desenvolvimento para cada período estimado/planejado, a mesma deve especificar o time de desenvovlimento contratado e a demanda estimada daquele período. Quando houver a necessidade de ampliação da equipe em relação a OS anterior ou vigente, pode-se atribuir um prazo (previsto no termo de referência) para que a empresa contratada apresente o novo profissional solicitado. Esta permissão tem por objetivo fornecer

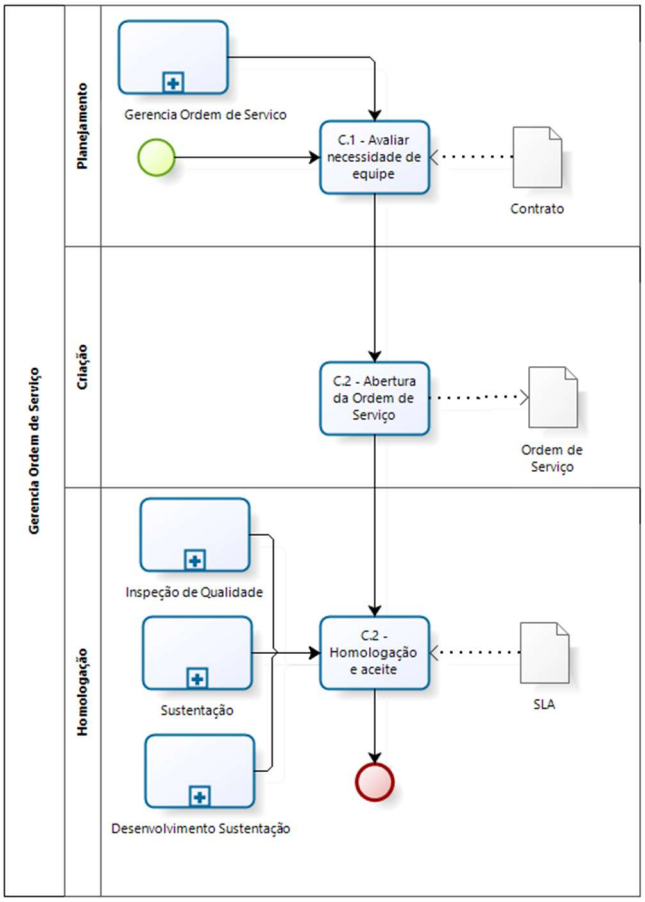
melhor estabilidade e previsibilidade no contrato então firmado.



Este processo consiste nas fases: Planejamento, onde é verificado a necessidade da equipe a ser alocada para as atividades. Esta atividade considera o Backlog CODEV e o contrato; Criação, é a fase seguinte onde de fato é formalizada a Ordem de Serviço, que deve ser assinada por representantes do MME e da Contratada; ao final da OS se executa a Homologação, a atividade de Homologação verifica os processos executados no ciclo da OS e possíveis pendências de ciclos anteriores nos processos de Sustentação, Desenvolvimento Sustentação e Inspeção de Qualidade. Ainda nessa homologação, os indicadores de qualidade e produtividade são avaliados e possíveis sanções previstas podem ser aplicadas em caso de desempenho abaixo dos níveis mínimos. O Backlog CODEV também deve ser atualizado na homologação da OS.

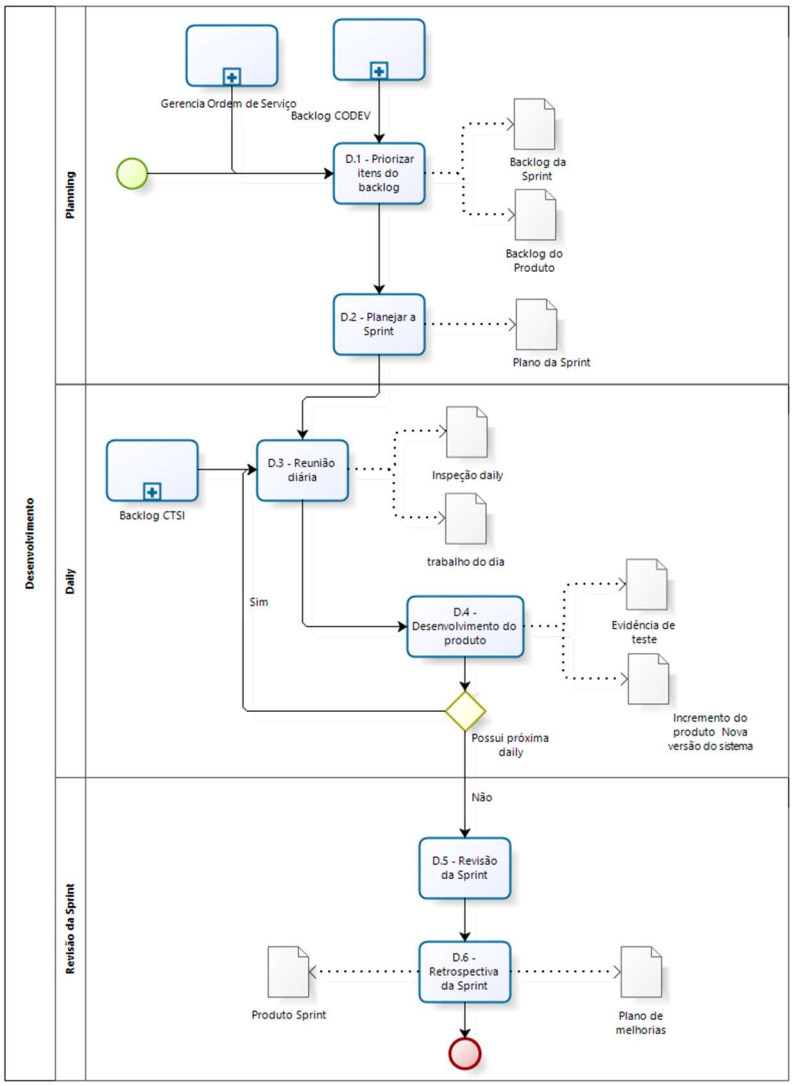
8.3. Gerencia Ordem de Serviço – Qualidade

O processo Gerencia Ordem de Serviço – Qualidade, consiste nas fases: Planejamento, onde é verificado a necessidade da equipe ser alocada para as atividades. Esta atividade considera o Gerencia Ordem de Serviço Desenvolvimento e o Contrato; Criação é a fase seguinte onde de fato é formalizada a Ordem de Serviço, que deve ser assinada por representantes do MME e da Contratada; ao final da OS executa-se a Homologação, a atividade de Homologação verifica os processos executados no ciclo da OS. Ainda nessa homologação, os indicadores de qualidade e produtividade são avaliados e possíveis sanções previstas podem ser aplicadas em caso de desempenho abaixo dos níveis mínimos.



8.4. Desenvolvimento Sustentação

O processo Desenvolvimento Sustentação, orienta a metodologia de desenvolvimento de software considerando práticas ágeis. As fases são separadas considerando as cerimônias Scrum. O processo prevê além da utilização de Sprint em desenvolvimento, a utilização de Sprint em Sustentação de Software quando o esforço for justificado, respeitando-se o máximo de 2 desenvolvedores dedicados a Sustentação na OS vigente. Pode-se para uma mesma OS existirem Sprints simultâneas respeitando os critérios do termo de referência e a possibilidade máxima de compartilhamento de profissionais. Os desenvolvedores não podem atuar em Sprints simultâneas, sem a autorização do MME. As Sprints podem ter uma, duas, três ou quatro semanas de duração. É imperativo que todos os profissionais previstos na OS estejam alocados em atividades, sejam elas Sprint de desenvolvimento ou sustentação, ou ainda atendendo a sustentação não realizada em Sprint.



O processo de Desenvolvimento Sustentação inicia com a fase Planning, que representa o planejamento da Sprint. Essa fase será realizada em até 08 horas conforme item 7.3 desse MDS. Possuindo como entrada os processos Gerencia Ordem de Serviço e Backlog CODEV, a atividade Priorizar itens do backlog, elenca quais itens serão priorizados na Sprint considerando o que mais agrega valor naquele momento. O PO, ou seja, o Dono do Produto, pode priorizar os itens, que em discussão com o ScrumMaster e o time de desenvolvimento verificará a viabilidade da escolha naquele momento. A equipe de qualidade também participará desse momento e poderá sugerir com o objetivo de melhoria da entrega. Eventualmente outro representante da CODEV poderá participar e opinar nessa priorização. O Backlog da Sprint e Backlog do Produto, deve conter a definição de pronto em linguagem simples de cada item elencado na Sprint. Após a Priorização dos itens de backlog, a atividade de Planejamento da Sprint é realizada. Nesta atividade as solicitações diversas devem ser registradas e comunicadas aos interessados para a boa fluidez da Sprint. Ainda no planejamento técnicas diversas podem ser utilizadas para mitigação de eventuais riscos. O Plano da Sprint deve ser simples o suficiente para manter a característica ágil do projeto e ao mesmo tempo completo o suficiente para a máxima redução de imprevistos.

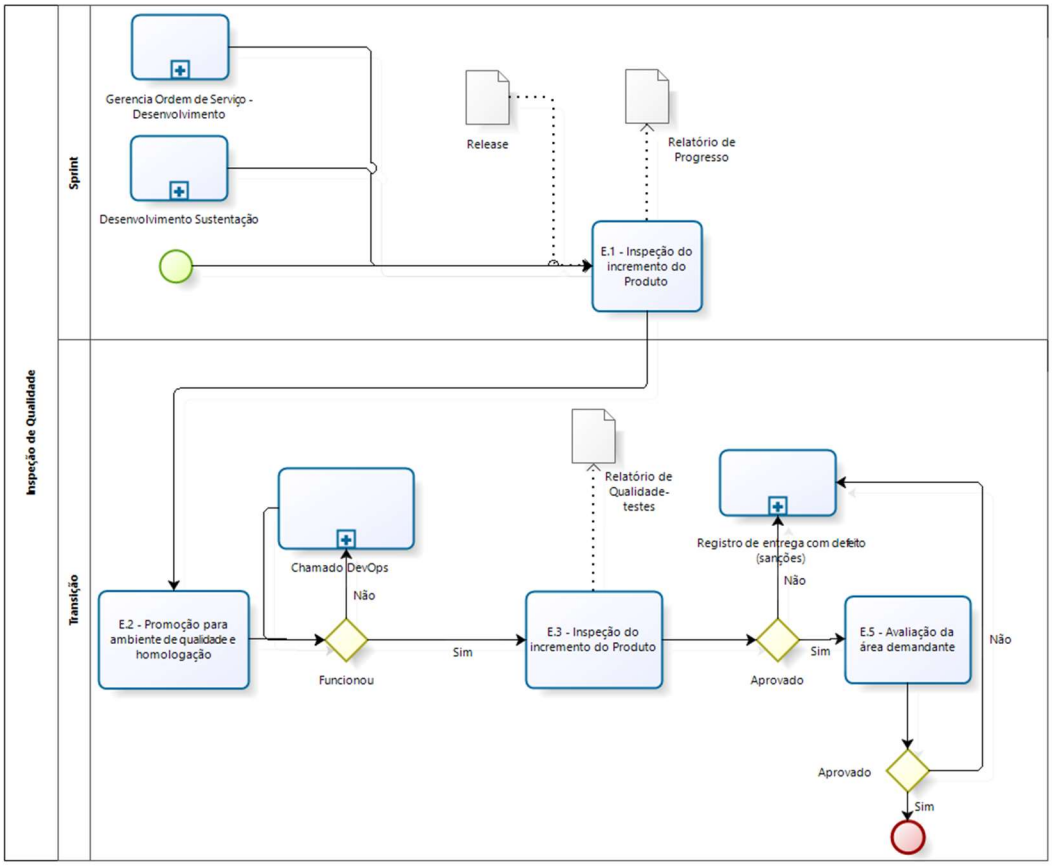
A fase Daily, compreende a rotina diária de desenvolvimento, todos os dias o Backlog CODEV será revisitado para a execução de possíveis demandas de sustentação que não devem onerar a Sprint de projetos específicos. Casos excepcionais podem ser solicitados pela CGIT/CODEV. Na atividade de Reunião diária, pratica-se o princípio de inspeção, onde verifica-se o que foi realizado no dia anterior e o efeito no produto atual. Os profissionais da qualidade podem participar dessas reuniões de forma a prevenir problemas futuros e verificar se os padrões adotados nesse Ministério estão sendo seguidos, cabe ainda a esses profissionais de qualidade informar de imediato eventuais desvios. O trabalho a ser realizado no dia também deve ser registrado. Cabe ressaltar que o desenvolvimento deve considerar as melhores práticas de códificação e padrões. Os encartes anexos no Termo de Referência do Contrato são imperativos e apenas podem ser desconsiderados com justificativa registradas pela CGIT/CODEV. A atividade de desenvolvimento permite autonomia para os times, no entanto, exigem-se as evidências de testes dos incrementos, assim como o incremento do produto ou nova versão do sistema.

A fase Revisão da Sprint, possui as atividades que representam a revisão e a retrospectiva da Sprint, que respectivamente avaliam a qualidade da entrega e o processo de desenvolvimento aplicado. A equipe de qualidade deve participar e verificar os artefatos confeccionados nessas cerimônias.

Toda a formalização das demandas, entregas e aceites podem ser formalizadas por sistema fornecido pela contratada, atas ou demais artefatos assinados pelos envolvidos.

8.5. Inspeção de Qualidade

O processo Inspeção de Qualidade, orienta a verificação da qualidade dos produtos e incrementos entregues pelas equipes de desenvolvimento.



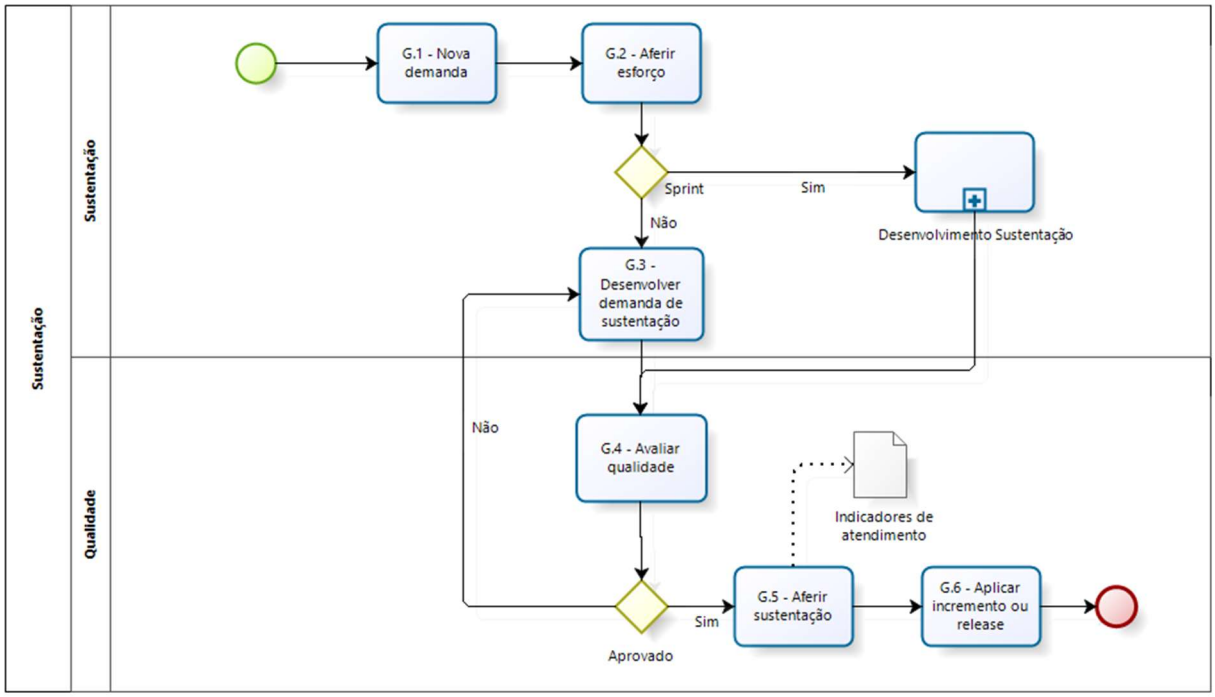
A fase Sprint tem como entrada os processos Gerencia Ordem de Serviço – Desenvolvimento e Desenvolvimento Sustentação quando houver processo executado. A atividade inicial Inspeção do incremento do Produto está relacionada com as fases Planning e Daily do processo Desenvolvimento Sustentação, onde em frequência suficiente QA deverá inspecionar os itens para antecipar eventuais desvios na entrega. Sugere-se que QA participe das reuniões diárias, e um relatório de Progresso da Sprint deve ser criado e atualizado periodicamente dentro de cada Sprint. Esse período é discricionário à equipe de QA em função da complexidade do produto.

A atividade de Promoção para ambiente de qualidade e homologação é realizada imediatamente após a última execução de daily, pode ocorrer de forma concomitante a revisão e retrospectiva da Sprint. Em caso de problemas de infraestrutura um chamado será aberto com a equipe de suporte da esteira DevOps, QA deve possuir profissional gabaritado para apoiar na resolução do problema. Quando solucionado o fluxo será continuado.

Após a Promoção do ambiente é realizada a inspeção profunda do incremento do Produto (qualidade de código e padrões). Os repectivos relatórios de qualidade de cada Sprint devem ser entregues. Quando existir um incremento ou produto testável por usuário, caso o produto seja aprovado seguirá para avaliação da área demandante. Reprovações nessa atividade e atividade anterior iniciaizarão o processo de Registro de entrega com defeito e suas respectivas sanções. Caso aprovado, a avaliação é finalizada até o novo ciclo de desenvolvimento.

8.6. Sustentação

Este Processo representa o fluxo da Sustentação de Software na CGIT/CODEV. As demandas serão registradas em sistema a ser definido pelos servidores do MME, e esta demanda poderá gerar itens de backlog no processo Backlog CODEV.



A fase de Sustentação possui a atividade inicial Nova demanda, que representa a recepção das demandas por meio de sistema específico. Qualquer integrante da CGIT/CODEV pode registrar uma demanda.

Em seguida avalia-se (MME e Contratada) se a demanda será executada pelos profissionais de sustentação ou Times ágeis.

Após o atendimento da demanda de Sustentação, a mesma é avaliada pela fábrica de qualidade (qualidade da entrega), com seu respectivo SLA. Em seguida é realizada a aferição do esforço de Sustentação e Aplicação do incremento ou release conforme fluxo da esteira DevOps.

É importante destacar que conforme os princípios ágeis, necessariamente os processos estarão em evolução contínua, podendo ser ajustados em comum acordo entre o Ministério e as empresas contratadas.