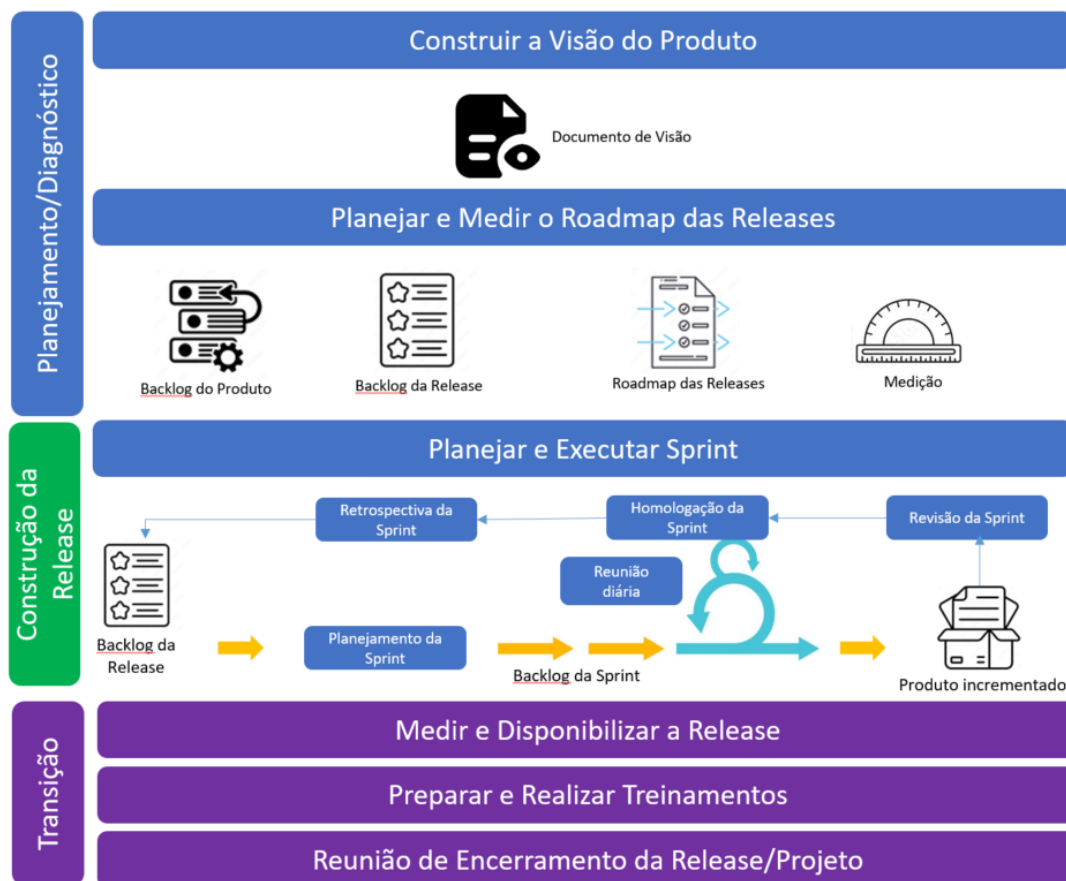


## ANEXO V - PROCESSO DE SOFTWARE

### 1. INTRODUÇÃO

1.1. O Processo de Desenvolvimento de *Software* e manutenção de *software* de referência baseia-se no framework SCRUM e observar as diretrizes de processo de ciclo de vida do *software* constantes da ABNT NBR ISO IEC IEEE12207:2020.

### 2. VISÃO MACRO DO PROCESSO



### 3. CARACTERÍSTICAS GERAIS DO PROCESSO

3.1. O processo de desenvolvimento de software deve levar em consideração o Guia de Desenvolvimento de Sistemas do INMETRO.

3.2. Uma necessidade será classificada como projeto caso seu tamanho funcional seja igual ou superior a 100 PF.

3.3. Uma release corresponderá a uma Ordem de Serviço.

3.4. **Construir a Visão do Produto:** Nesta etapa são descritas as necessidades, expectativas, objetos específicos de negócio e proposta da solução para o projeto.

3.5. **Planejar e medir o roadmap das releases:** Nesta etapa é construído o planejamento de entregas ou plano cronológico de liberação dos releases (versões do produto). Esse plano cronológico é chamado de roadmap. Planejar o roadmap significa dividir objetivos de negócio e as características importantes ou macrofunções do produto em partes entregáveis (releases), por ordem de prioridade. São discutidas as características do backlog do produto e definido o primeiro backlog da release. Nesta fase também são definidos os padrões técnicos centrais como arquitetura (framework, componentes, ferramentas, etc), identidade visual, configurações padrão das máquinas dos times de desenvolvimento, etc.

3.6. As fases **Construir a Visão do Produto** e **Planejar e medir o roadmap das releases** correspondem a etapa de Diagnóstico do Projeto.

3.7. **Planejar e Executar a Sprint:** Nesta etapa é construído e entregue um incremento do produto em um prazo pré-definido, de acordo com o backlog da sprint, com a meta de completar o backlog da release.

- 3.8. **Medir e disponibilizar a release:** Implantar a release no ambiente de homologação/produção, incluindo a disponibilização de documentações e configurações de acesso. Nesta fase podem ser necessárias outras tarefas como: publicação de documentos, manuais técnicos, de treinamento, códigos-fonte, configuração de acesso, carga de dados, configuração de rotinas automáticas, jobs, email, etc.
- 3.9. **Preparar e realizar treinamentos:** Esta fase compreende o planejamento, a preparação do material e o repasse de todo conhecimento do sistema ou release desenvolvido para as equipes de suporte, sustentação, gestores de negócio e usuários. Esta atividade pode envolver toda a equipe do projeto.
- 3.10. **Reunião de encerramento da release/projeto:** Esta fase compreende da realização de verificações do processo e do produto para confirmar que a release e/ou projeto podem ser formalmente encerrados.
- 3.11. A fase de operação só é iniciada após entrada em produção do produto.

#### 4. TIME ÁGIL

- 4.1. Este Processo é executado pelo Time ágil e deve ser observado e adaptado à realidade operacional da CONTRATANTE.
- 4.2. O Time ágil consiste, no mínimo, em um Scrum Master, um Dono do Produto e Desenvolvedores. Esse time pode ser integrado também por profissionais especializados.
- 4.3. Participam do Time ágil, os seguintes papéis:
- a) Scrum Master: Profissional de conhecimento aprofundado em técnicas ágeis e responsável por promover o ambiente de desenvolvimento ágil resolvendo impedimentos e incentivando o time.
  - b) Dono do Produto (Product Owner – PO): Servidor ou profissional da CONTRATANTE responsável por ordenar o trabalho a ser realizado pelo time, criando, mantendo e priorizando o(s) backlog(s) do(s) produto(s). Papel atribuído ao requisitante do projeto.
  - c) Desenvolvedores de *software*: Profissionais encarregados pela construção do(s) produto(s) de *software*.
  - d) Profissionais Especializados: Demais profissionais que integram o time com especialidades definidas e contribuem para a construção dos produtos de *software*.
  - e) Partes Interessadas (Stakeholder): Profissionais impactados pela solução ou que possuam interesse na entrega da solução.
- 4.4. A formação do time ágil deve considerar os requisitos de formação de equipe estabelecidos no Termo de Referência.

#### 5. BACKLOG DO PRODUTO

- 5.1. Os requisitos do software a ser desenvolvido serão decompostos em histórias de usuários que, por sua vez, poderão ser subdivididas em tarefas. Este refinamento será feito no decorrer do projeto de acordo com a prioridade dos requisitos do software.
- 5.2. Os requisitos do software, as histórias de usuários e as tarefas compõem o backlog do produto.
- 5.3. As eventuais manutenções corretivas e adaptativas que venham a ser necessárias no software podem ser incluídas no backlog do produto.
- 5.4. O backlog do produto será priorizado pelo dono (PO).

#### 6. SPRINTS

- 6.1. As sprints terão duração entre 1 a 4 semanas (independente da quantidade de dias úteis), de acordo com as características de cada projeto.
- 6.2. A duração das sprints e o dia e a duração das reuniões de planejamento, revisão e retrospectiva das sprints serão definidas pela CONTRATANTE no início do projeto. Estas definições poderão ser alteradas posteriormente, a critério da CONTRATANTE, mediante comunicação prévia à equipe de desenvolvimento da CONTRATADA.
- 6.3. Critérios específicos para aceitar cada tarefa como “pronta” (finalizada, done) poderão ser definidos para cada projeto e, quando necessário, de forma particular para cada tarefa. Quando não especificados, serão utilizados os critérios estabelecidos no Termo de Referência.

#### 7. PLANEJAMENTO DA SPRINT

- 7.1. Com base no backlog do produto priorizado, o time realiza o planejamento da sprint, selecionando os itens possíveis de serem desenvolvidos no tempo da sprint.

- 7.2. O dono do produto estabelece os critérios de Definição de Pronto, ou seja, o que deve ser entregue ao final da sprint.
- 7.3. O time, então, se compromete a entregar o incremento de software, ao final da sprint.
- 7.4. Esse planejamento pode levar até 4 horas, dependendo do tamanho da sprint.

## **8. EXECUÇÃO DA SPRINT E REUNIÃO DIÁRIA**

- 8.1. O time de desenvolvimento então inicia a execução da sprint, construindo o incremento de software a partir do backlog da sprint.
- 8.2. Durante esse período, o time também se reúne diariamente, por 15 minutos, para responder a três perguntas:
  - a) O que eu fiz desde a última reunião?
  - b) O que eu vou fazer até a próxima reunião?
  - c) Há algum impedimento para o trabalho ser realizado?

## **9. REVISÃO DA SPRINT**

- 9.1. Após o último dia da execução da sprint, a equipe se reúne para realizar a revisão do produto.
- 9.2. Nesse evento, o time de desenvolvimento apresenta o incremento de software construído para o Dono do Produto e os stakeholders.
- 9.3. O time revisa o que foi realizado na sprint e avalia o que fazer a seguir.
- 9.4. O Dono do Produto analisará a resolução de cada história apresentada e decidirá se está “Pronta” ou não, de acordo com o critério previamente estabelecido.
- 9.5. Nesse momento, o backlog do produto pode ser atualizado para refletir as novas demandas e oportunidades.
- 9.6. Caso alguma história seja reprovada, ela será reinserida no Backlog ficando disponível para uma próxima sprint.
- 9.7. Essa revisão pode levar até 4 horas para sprints de 4 semanas.

## **10. RETROSPECTIVA DA SPRINT**

- 10.1. O último evento antes da conclusão da sprint é chamado de retrospectiva da sprint.
- 10.2. O propósito da retrospectiva da sprint é planejar maneiras de aumentar a qualidade e a eficácia do trabalho.
- 10.3. O time discute como foi a última sprint em relação a indivíduos, interações, processos, ferramentas e a Definição de Pronto.
- 10.4. Nesse evento responde-se basicamente a dois questionamentos:
  - a) O que deu certo?
  - b) O que pode ser melhorado?
- 10.5. O time, então, identifica as mudanças mais úteis para melhorar sua eficácia e procura aplicar essas mudanças na sprint seguinte.
- 10.6. A retrospectiva pode levar até 4 horas para sprints de 4 semanas.
- 10.7. Finalmente, o time inicia o planejamento da próxima sprint e o ciclo se repete quantas vezes forem necessárias.

## **11. LIBERAÇÃO E ENTREGAS**

- 11.1. A entrega dos produtos de *software* ocorre após a finalização da sprint, quando o incremento de *software* é revisado e homologado. É um evento separado da sprint.

## **12. MEDIÇÃO**

- 12.1. A validação das mensurações realizadas será realizada por empresa contratada especializada ou pelo INMETRO.
- 12.2. A CONTRATADA deverá apresentar os devidos artefatos para a correta validação da contagem (ex: fundamentação da contagem e planilha de contagem)

12.3. A CONTRATADA se obriga a executar o serviço de medição de tamanho funcional segundo o Manual de Práticas de Contagem (CPM) versão 4.3.1 (ou superior) do IFPUG, acrescidas do Roteiro de Métricas de Software do SISP (na versão 2.3 ou superior que pode ser obtido no endereço <http://www.sisp.gov.br/metricas/wiki/roteirometricas>) bem como possíveis roteiros complementares elaborados pelo INMETRO.

12.4. As contagens de tamanho funcional dos tipos poderão ser realizadas conforme as técnicas: NESMA (Netherlands Software Metrics Users Association) ou SIMPLE FUNCTION POINT (SFP), a critério do INMETRO.

12.5. Medição no Planejamento: No planejamento as medições de tamanho e esforço do projeto podem ser realizadas com base nos objetivos de negócio, características-chaves do produto, requisitos ou histórias definidas.

12.6. Medição na Construção do Release: As medições **podem** ser realizadas no início de cada release, baseado no backlog do release e nas especificações dos requisitos definidos. No final do release deve ser realizada uma contagem com base no trabalho realizado, incluindo refinamento de requisitos, alterações e exclusões nas funcionalidades entregues e homologadas pelo dono do produto.

12.7. Medição na Transição: No final de cada release do projeto as medições podem ser realizadas com base nas funcionalidades implementadas, homologadas e disponibilizadas em produção (valor agregado ao produto) preferencialmente aplicando o tipo de contagem detalhada.

12.8. O teto de alterações de refinamento da release é de **30% (fator Delta)**. Será objeto de faturamento somente refinamentos que ultrapassem o teto.

12.9. As alterações que ocorrerem em funcionalidades existentes após o término da release devem ser tratadas como projeto de melhoria.

#### 12.10. Cálculo do Fator Evolutivo Ágil e do PF Excedente

12.10.1. O fator evolutivo ágil é o volume de alterações realizadas ao longo do ciclo da release de acordo com a fórmula abaixo:

$$\text{Fator Evolutivo Ágil} = ((\text{soma PF Sprints} / \text{PF Releases}) - 1) * 100$$

*Soma (PF Sprints) = Soma de PF de todas as sprints*

*PF Release = contagem detalhada da release*

12.10.2. Após o cálculo do Fator Evolutivo Ágil, compara-se o resultado com o Delta%. Se o Fator Evolutivo for menor que o Delta%, então não haverá remuneração adicional além do valor referente a contagem da release. Se o Fator Evolutivo Ágil for maior que o Delta%, então deve ser calculado o quantitativo de pontos excedentes a Delta%.

$$\text{PF Excedente} = (\text{Soma(PF Sprints)} - \text{PF Release}) - (\text{PF Release} * \text{Delta\%})$$

*PF Excedente = Quantidade de pontos de função excedentes ao Delta %.*

*Soma (PF Sprints) = somatório das contagens de pontos de função das sprints.*

*PF Release = contagem final detalhada da release.*

*Delta% = Fator Delta definido pelo órgão.*