



**PODER JUDICIÁRIO DO ESTADO DO PIAUÍ**  
**SECRETARIA DA PRESIDÊNCIA - SECPRE**  
Pça Des. Edgard Nogueira s/n - Bairro Cabral - Centro Cívico - CEP 64000-830  
Teresina - PI - www.tjpi.jus.br

Portaria (Presidência) N° 911/2021 - PJPI/TJPI/SECPRE, de 09 de abril de 2021

Dispõe sobre a instituição da Metodologia de Desenvolvimento de Sistemas no âmbito do Poder Judiciário do Estado do Piauí.

O **PRESIDENTE DO TRIBUNAL DE JUSTIÇA DO ESTADO DO PIAUÍ**, Excelentíssimo Desembargador José Ribamar Oliveira, no uso de suas atribuições legais e regimentais;

**CONSIDERANDO** a Resolução n° 370, de 28 de janeiro de 2021, do Conselho Nacional de Justiça (CNJ), que institui a Estratégia Nacional de Tecnologia da Informação e Comunicação do Poder Judiciário (ENTICJUD);

**CONSIDERANDO** o disposto no item 2.5, do Levantamento iGovTIC-Jud-2020 do CNJ, referente à formalização e cumprimento do processo de software;

**CONSIDERANDO** a Tecnologia de Informação (TIC) como ferramenta indispensável à realização das funções institucionais do TJPI e como instrumento para viabilizar soluções que conduzam ao alcance dos objetivos estratégicos do Tribunal;

#### **R E S O L V E:**

**Art. 1º** Instituir a Metodologia de Desenvolvimento de Sistemas no âmbito da Secretaria de Tecnologia da Informação e Comunicação (STIC) do PJPI, que passa a vigorar conforme Anexo Único desta Portaria.

**Art. 2º** Esta Portaria entra em vigor na data de sua publicação.

Publique-se. Registre-se. Cumpra-se.

GABINETE DA PRESIDÊNCIA DO EGRÉGIO TRIBUNAL DE JUSTIÇA DO ESTADO DO PIAUÍ, em Teresina, 09 de abril de 2021.

**ANEXO ÚNICO**  
**METODOLOGIA DE DESENVOLVIMENTO DE SISTEMAS - MDS-TJPI**  
**TRIBUNAL DE JUSTIÇA DO ESTADO DO PIAUÍ**  
**TERESINA - 2021**  
**VERSÃO 1.0**

#### **1. INTRODUÇÃO**

Esta Metodologia de Desenvolvimento de Sistemas visa descrever e normatizar os processos de gerenciamento, desenvolvimento e manutenção de sistemas utilizados no Tribunal de Justiça do Piauí.

Em engenharia de software, uma metodologia de desenvolvimento comumente é entendida como um conjunto estruturado de práticas que pode ser repetível durante o processo de produção do sistema ou, ainda, a forma de se utilizar um conjunto de práticas, métodos ou processos para se desenvolver ou manter um produto de software, de modo que se evite subjetividade na execução do trabalho. O uso de metodologias visa à produtividade das equipes e à qualidade do produto.

Atualmente, modelos de melhoria de processos de software, como o CMMI (Capability Maturity Model Integration ou Modelo Integrado de Maturidade em Capacitação) e o MPS.Br (Melhoria do Processo de Software Brasileiro), bem como a jurisprudência do Tribunal de Contas da União (Acórdãos 953/2009, 1.233/2012, 3.132/2012 e 1.167/2013, todos do Plenário do TCU), têm utilizado o termo “processo de software” em detrimento a “metodologia de desenvolvimento de software”, embora as definições de ambos sejam, em essência, idênticas.

As metodologias ágeis são representadas por um conjunto de valores e princípios a ser utilizado no processo de desenvolvimento de sistemas. Esse conjunto de valores e princípios foi externado em 2001, por meio da divulgação do Manifesto para Desenvolvimento Ágil de Software, criado por um grupo de dezessete especialistas em processos de desenvolvimento de software que, individualmente, já utilizavam práticas e teorias adaptadas.

Alguns órgãos da administração pública adotam metodologia ágil no desenvolvimento de software: Tribunal Superior do Trabalho (TST), Banco Central do Brasil (BACEN), Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira (Inep); Supremo Tribunal Federal (STF); Tribunal Superior Eleitoral (TSE), Tribunal de Justiça do Paraná (TJPR), Sistema de Administração de Recursos de Tecnologia da Informação do Governo Federal (SISP) e etc.

Considerando o Indicador de Governança, Gestão e de Infraestrutura de Tecnologia da Informação do Poder Judiciário (iGovTIC-JUD), foi elaborada uma metodologia de desenvolvimento de software, visando atender o item "Processos de Software".

Assim os processos de software do iGovTIC-JUD devem conter os seguintes processos:

- Gerenciamento de escopo e requisitos
- Gerenciamento de Arquitetura
- Processo de Desenvolvimento
- Processo de Sustentação ou Manutenção
- Gerenciamento de solução de software

Esse processo foi elaborado considerando as melhores práticas de desenvolvimento ágeis, na qual encontram-se artefatos das metodologias, métodos e frameworks disponíveis no mercado como UML, XP, Scrum e Kanban.

Scrum e Kanban são frameworks dentro dos quais pessoas podem tratar e resolver problemas complexos e adaptativos, enquanto produtiva e criativamente entregam produtos com o mais alto valor possível.

### 1.1 Objetivo

Implantar uma Metodologia Desenvolvimento de Sistemas visando organizar, padronizar e melhorar os processos de software, considerando as recomendações do Conselho Nacional de Justiça.

Esta metodologia define grupo de processos de desenvolvimento de software baseados em desenvolvimento iterativo, onde os requisitos e as soluções para os problemas evoluem através da colaboração das equipes, visando a utilização de um conjunto de boas práticas de engenharia de software, permitindo entregas frequentes com alto grau de qualidade.

## 2. METODOLOGIAS DE DESENVOLVIMENTO E MANUTENÇÃO DE SISTEMAS

Tendo em conta a grande diversidade da natureza das demandas de TI e a variedade do tamanho e configuração das equipes de trabalho, definir somente uma metodologia de desenvolvimento e manutenção de software pode, dependendo da situação, criar um entrave para o bom andamento das atividades, ainda que se trate de um processo ágil. Isto posto, serão apresentadas duas formas de trabalho possíveis - Scrum e Kanban - de modo de, independente de qual for adotada, o objetivo principal será atendido, que é a entrega rápida e eficaz de resultados, com gerenciamento ágil do andamento dos trabalhos.

### 2.1 SCRUM

O Scrum é recomendado para equipes de três ou mais pessoas, e escopo de trabalho claro o bastante para a definição de iterações de tempo fixo (de duas a quatro semanas cada).

#### 2.1.1 Papéis Envolvidos

Papel	Sigla	Obrigatório	Responsabilidade
Product Owner	PO	Sim	<ul style="list-style-type: none"> <li>Representar o cliente final nas decisões sobre as necessidades e requisitos do sistemas.</li> <li>Homologar os produtos e serviços entregues.</li> </ul>
Scrum Master	SM	Sim	<ul style="list-style-type: none"> <li>Garantir que as técnicas do Scrum sejam utilizadas no projeto, incentivando as práticas ágeis e removendo os impedimentos do time. O Scrum Master pode ser alguém do time Scrum (ou não), o gerente de projeto pode assumir esse papel.</li> <li>Conduzir o planejamento e gerenciamento do projeto.</li> <li>Coordenar as interações com os principais envolvidos.</li> <li>Manter a equipe de projeto focada em alcançar os objetivos.</li> <li>Conduzir o planejamento do projeto.</li> <li>Coordenar as interações com os principais envolvidos.</li> <li>Manter a equipe de projeto focada em alcançar os objetivos.</li> </ul>
Analista de Conformidade	ANC	Não	<ul style="list-style-type: none"> <li>Avaliar os processos e produtos de trabalho do projeto em relação à conformidade com a descrição de processos, políticas, padrões e procedimentos aplicáveis à STIC.</li> </ul>
Equipe de Suporte	SUP	Sim	<ul style="list-style-type: none"> <li>Participar da transferência de conhecimento recebendo as informações sobre a utilização do sistema que auxiliarão a equipe no suporte aos usuários finais.</li> </ul>
Especialista em Ponto de Função	APF	Não	<ul style="list-style-type: none"> <li>Realizar a medição funcional do projeto.</li> <li>Validar as contagens recebidas da contratada.</li> </ul>
Fiscal do Contrato	FCO	Não	<ul style="list-style-type: none"> <li>Autorizar a abertura e o fechamento das ordens de serviço.</li> <li>Aplicar as regras de não conformidade com o contrato.</li> </ul>
Preposto da Fábrica	PFW	Não	<ul style="list-style-type: none"> <li>Participar quando necessário, da formalização de aceites provisórios ou definitivos das entregas dos produtos gerados na sprint, e apurar se ocorreu (ou não) o atendimento de todos requisitos especificados na ordem de serviço.</li> <li>Remover impedimentos que envolvam a equipe da FSW.</li> </ul>

Time Scrum	TS	Sim	<ul style="list-style-type: none"> <li>• Um conjunto de pessoas com habilidades multidisciplinares, capaz de se auto-organizar para produzir o produto com qualidade e valor para o cliente, responsável por: <ul style="list-style-type: none"> <li>◦ Executar as atividades do desenvolvimento do software;</li> <li>◦ Executar a engenharia de requisitos de negócio e de sistema;</li> <li>◦ Produzir e validar a documentação a ser entregue;</li> <li>◦ Definir a arquitetura de software;</li> <li>◦ Tomar decisões técnicas que orientam todo o design e a implementação do projeto;</li> <li>◦ Prezar pela qualidade do código-fonte;</li> <li>◦ Desenvolver as soluções propostas;</li> <li>◦ Analisar e modelar o banco de dados;</li> <li>◦ Criar as interfaces solicitadas pelo usuário;</li> <li>◦ Criar e manter a estrutura do banco de dados;</li> <li>◦ Zelar pela integridade do banco de dados;</li> <li>◦ Melhorar o desempenho do banco de dados.</li> </ul> </li> </ul>
Usuário Final	USU	Sim	<ul style="list-style-type: none"> <li>• Participar da transferência de conhecimento recebendo as informações sobre a utilização do sistema que auxiliarão o usuário nas suas atividades no produto construído.</li> </ul>
Gestor do Sistema	GES	Sim	<ul style="list-style-type: none"> <li>• Será de Responsabilidade do gestor do sistema todas as decisões referentes a ele, desde sua implantação, permissões de acessos, definição de informações restritas;</li> <li>• Autorizar e Priorizar solicitações de manutenção evolutiva e corretiva;</li> <li>• Homologação de Testes;</li> <li>• Definir os perfis de acesso ao sistema;</li> <li>• Estabelecer critérios de aceitação para novas funcionalidades, atualizações e novas versões e acompanhar testes apropriados durante o desenvolvimento;</li> <li>• Prestar suporte de negócio quanto ao seu uso correto;</li> <li>• Realizar treinamento do sistema em parceria com a Diretoria de Tecnologia da Informação.</li> </ul>
Requisitante	Req	Sim	<ul style="list-style-type: none"> <li>• Qualquer unidade do Poder Judiciário do Piauí que demande o desenvolvimento de software.</li> </ul>

### 2.1.2 Definições do Processo

**Backlog:** É uma lista que contém todas as funcionalidades desejadas para um novo produto ou para a evolução de um já existente. O conteúdo dessa lista é definido pelo Product Owner. O backlog do produto é dinâmico e construído ao longo do projeto.

**Crítérios de aceitação:** O critério de aceite nada mais é que os pontos que os desenvolvedores devem cumprir antes de considerar aquela história como finalizada. O que deve ser feito, o que será testado e quais os resultados esperados são itens que podem estar na lista de critérios para cada uma das histórias. É responsabilidade do Product Owner, juntamente com o Time Scrum, levantar os critérios de aceite para cada história de usuário.

**Daily Scrum:** Breve reunião diária feita pelos membros da Equipe como forma de disseminar o conhecimento sobre o que foi feito no dia anterior, identificar problemas e priorizar o que deve ser feito no dia que se inicia. A reunião é feita "em pé", como forma de salientar seu caráter de duração rápido. Cada membro do time responde as seguintes perguntas: O que você fez ontem? O que você fará hoje? Há algum impedimento em seu caminho? Durante a reunião não se deve efetivamente resolver os problemas/impedimentos detectados, o que faria a reunião se estender, isso deve ser feito em um momento posterior com o apoio do Scrum Master.

**Equipe/Time:** A equipe técnica se organiza para definir a melhor maneira de entregar as funcionalidades de maior prioridade. Uma equipe de Scrum é composta de 3 a 9 indivíduos – requisito fundamental para a boa prática da metodologia. Deve ser multifuncional, ou seja, composta por profissionais de diversas especialidades, desenvolvedores, arquitetos da informação, designers, testadores, etc. É importante que todos os membros se dediquem, em tempo integral, à equipe. Exceções aplicam-se aos profissionais que exercem papéis pontuais: administradores de base de dados, analistas de suporte e equipe de infraestrutura. As equipes devem ser auto-organizáveis, não havendo títulos ou hierarquia entre seus membros.

**História de usuário:** É um texto ou parágrafo de explicação da funcionalidade, que define suas características. O objetivo da história não é definir o escopo global do sistema, mas sim estimar a complexidade de cada parte do sistema visando possibilitar a estimativa do esforço necessário para a sua implementação. Recomenda-se que as histórias mais importantes e/ou mais difíceis tenham prioridade.

**Kanban:** É um termo de origem japonesa e significa literalmente “cartão” ou “sinalização”. Além de representar o quadro de atividades, Kanban é também um método de gestão de mudanças que dá ênfase aos seguintes princípios: visualizar o trabalho em andamento; visualizar cada passo em sua cadeia de valor, do conceito geral até software que se possa lançar; tornar explícitas as políticas sendo seguidas, entre outros princípios. Pode-se utilizar cartões (post-it e outros) para indicar o andamento do desenvolvimento do software, colocando neles indicações sobre uma determinada tarefa, por exemplo: “para executar”, “em andamento” ou “finalizado”.

**MGP:** Metodologia de Gestão de Projetos.

**Planning Poker:** Técnica usada para estimar o tamanho de uma história. Cada membro da equipe recebe um baralho de 13 cartas. Sempre que uma história for estimada, cada membro escolhe uma carta que represente a sua estimativa de tempo (em pontos por história) colocando-a virada para baixo sobre a mesa. Quando todos os membros da equipe tiverem feito sua estimativa, as cartas são reveladas simultaneamente. Dessa forma, cada membro da equipe é forçado a pensar por si próprio ao invés de basear-se na estimativa de outra pessoa. Se houver uma grande divergência entre duas estimativas, a equipe discute as diferenças e tenta chegar a uma visão comum do trabalho envolvido na história. Eles podem fazer algum tipo de decomposição de tarefas. Depois disso, a equipe faz novamente a estimativa. Esse processo é repetido até que as estimativas de tempo cheguem a uma convergência, isto é, até que todas as estimativas sejam aproximadamente a mesma para cada história.

**Product Owner:** É a nomenclatura que identifica o indivíduo no papel de cliente no processo de desenvolvimento de software. O Product Owner é responsável por determinar as prioridades do que deve ser feito no software. Deve ser capaz de sanar as dúvidas de requisitos.

**Scrum:** É um método ágil para gerenciamento de projetos baseado em times pequenos e auto-organizados, forte visibilidade e rápida adaptação e que permite trabalhar de forma iterativa para garantir a inspeção e adaptação do produto de forma incremental.

**Scrum Master:** É um facilitador cuja função primária é remover impedimentos à capacidade da equipe para entregar o objetivo da Sprint. O Scrum Master garante que o processo Scrum seja usado como pretendido. Ele não é líder da equipe, já que as equipes são autogerenciadas, mas atua como um mediador entre a equipe e qualquer influência desestabilizadora. Outra função extremamente importante de um Scrum Master é a de assegurar que a equipe esteja utilizando corretamente as práticas de Scrum, motivando os integrantes, mantendo o foco na meta da Sprint.

**Sprint:** Uma Sprint é a unidade básica de desenvolvimento em Scrum. Tende a durar entre duas semanas e um mês, sendo um esforço em uma "caixa de tempo" (restrito a uma duração específica) de comprimento constante.

**Sprint Backlog:** Subconjunto de funcionalidades do Backlog alocadas para serem implementadas em uma determinada Sprint.

**Sprint Planning:** Reunião de planejamento efetuada antes do início de uma Sprint para priorizar os itens do Product Backlog e selecionar as funcionalidades que entrarão para o Sprint Backlog de acordo com a capacidade da Equipe.

**Sprint Review (Apresentação da Sprint):** É uma reunião que deve ser realizada nos últimos dias da Sprint, com a participação de todos, para apresentar o resultado do trabalho de toda a Sprint e, também, levantar os pontos positivos e negativos. Nessa reunião, podem ser levantadas novas funcionalidades que não foram entregues e a necessidade de novos itens.

**Sprint Retrospective:** Reunião efetuada após a Sprint para identificar o que funcionou bem durante a Sprint e deve ser mantido para as próximas, o que funcionou mal e deve ser melhorado e que ações serão tomadas para melhorar.

**TAP:** É o artefato da MGP por meio do qual a abertura do projeto é formalizada. Nele está contida a descrição dos objetivos macros, requisitos de alto nível e riscos preliminares elaborados com a participação do cliente (ou Product Owner).

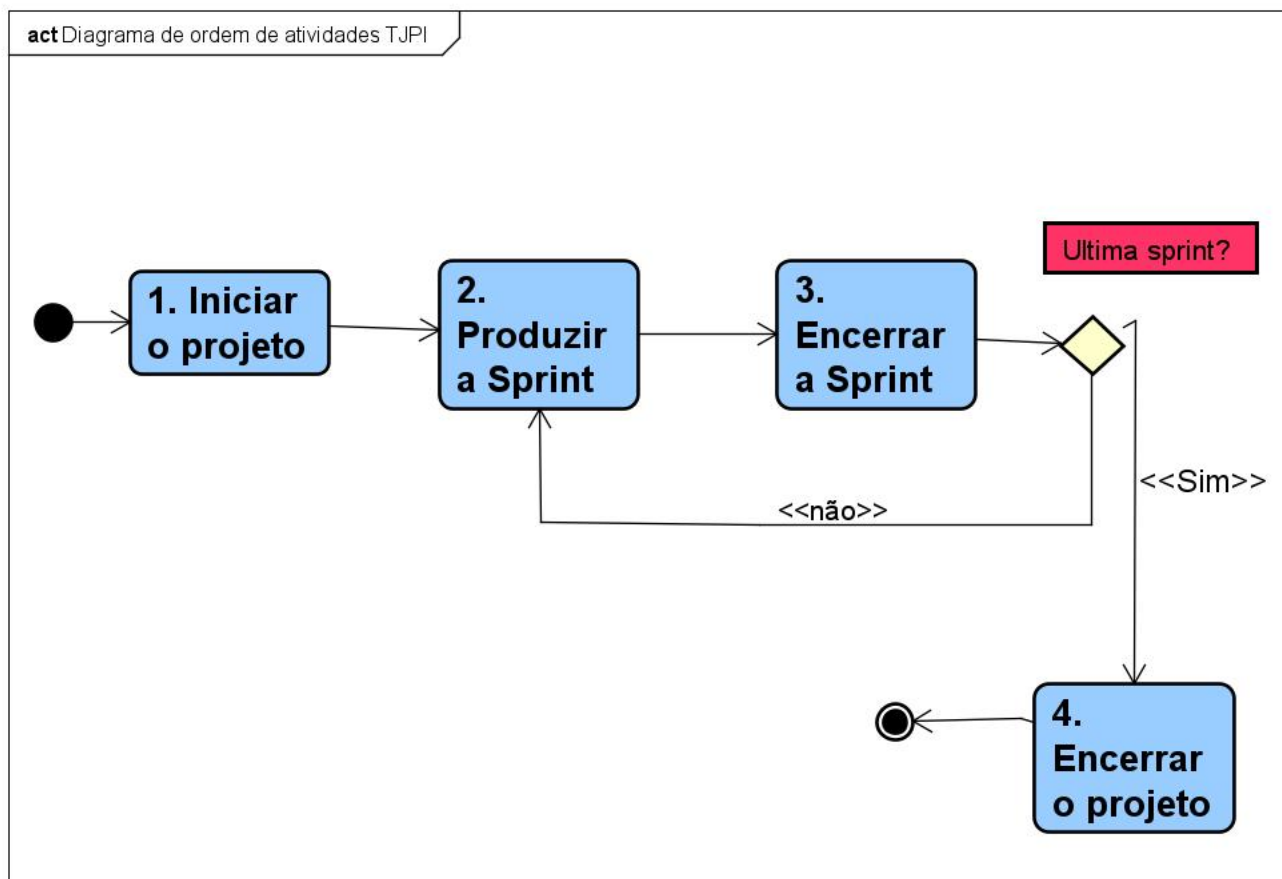
**TEP:** Artefato da MGP responsável por formalizar o encerramento do projeto.

**Técnicas de Extreme Programming (XP):** O Scrum é focado nas práticas de gerenciamento e organização, enquanto o XP dá mais atenção às tarefas de programação. Muitas das técnicas do XP são utilizadas largamente pelas equipes de programação. Cita-se, como exemplos: programação em par; desenvolvimento orientado a testes (TDD); design incremental; integração contínua; propriedade coletiva do código; ambiente de trabalho informativo; padrão de codificação, entre outros.

### 2.1.3 Visão Geral

A Metodologia de Desenvolvimento de Sistemas - SCRUM do Tribunal de Justiça do Piauí - MDS-SCRUM-TJPI é composta por quatro etapas que abrangem todo o ciclo de vida do desenvolvimento de sistemas e que promovem a agilidade no atendimento às necessidades dos projetos do TJPI.

As etapas são: Iniciar Projeto, Produzir a Sprint, Encerrar a Sprint e Encerrar do Projeto. Essas etapas são compostas por subprocessos que detalham sua execução. Os novos sistemas e as manutenções nos sistemas já implantados devem seguir as fases da Metodologia de Desenvolvimento de Sistemas.



### 2.1.3.1 Projeto de Sistema

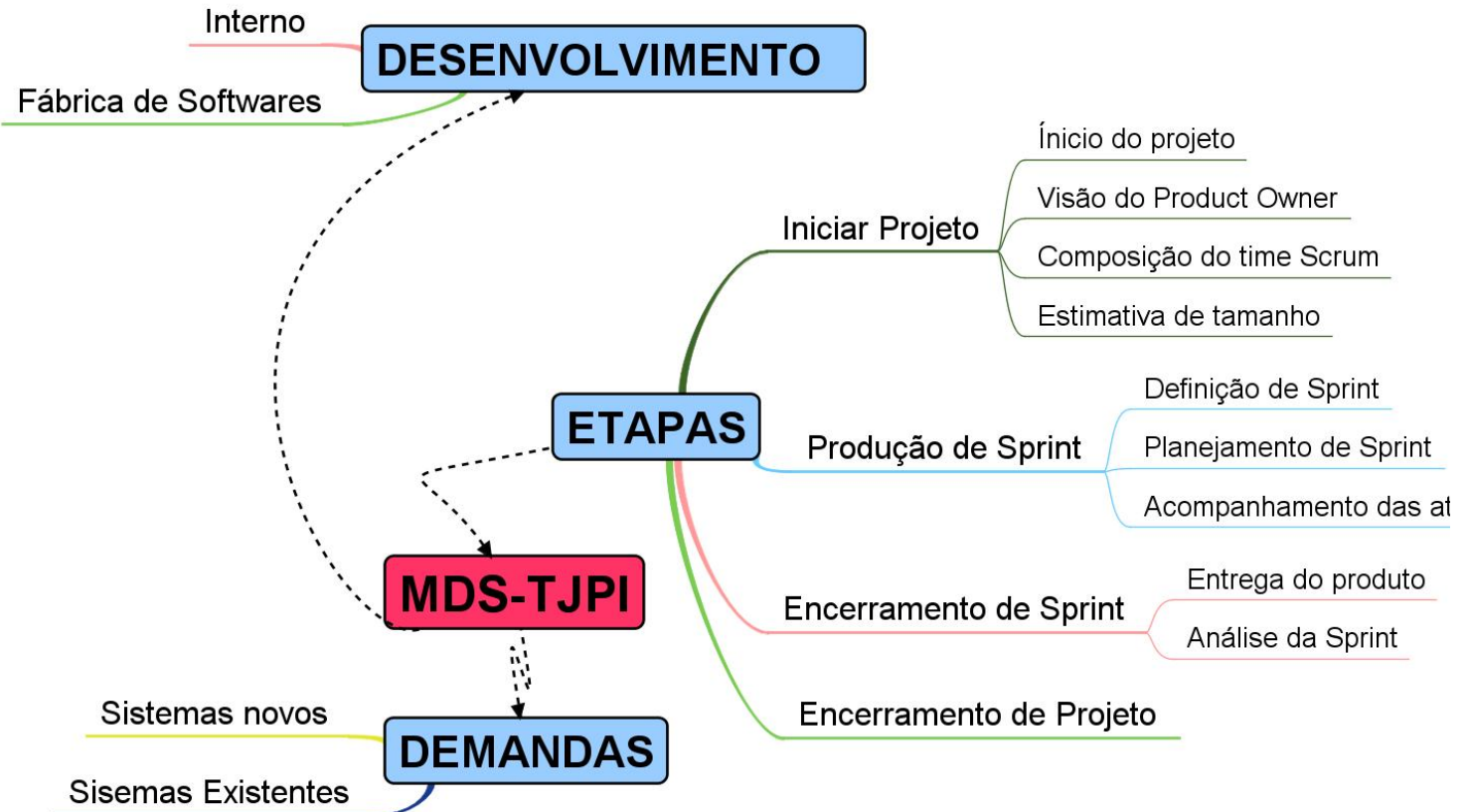
Projeto de Sistema é um serviço disponibilizado pela Área de Tecnologia da Informação para atender várias necessidades do órgão. São exemplos de um projeto de sistema:

- **Novo Desenvolvimento:** Um novo sistema de informação pode ser desenvolvido integralmente ou reconstruído a partir de um legado;
- **Customização e Implantação:** A partir de novas necessidades de negócio, podem ser encontrados sistemas de informação já existentes para estas necessidades, dispensando a construção de um novo. Nestes casos o projeto pode ser a customização e implantação do sistema dentro da estrutura e arquitetura da instituição, utilizando um código-fonte que foi, em todo ou em parte, cedido ou repassado à instituição, ou obtido por outros meios;
- **Manutenções Evolutivas:** Em geral as demandas de evolução de sistemas de informação são de responsabilidade da equipe de sustentação. Estas demandas quando de tamanho ou complexidade maiores poderão ser tratadas como projetos de sistema. Manutenções adaptativas, perfectivas e cosméticas são exemplos de evoluções de sistemas de informação.

### 2.1.3.2 Análise da Demanda

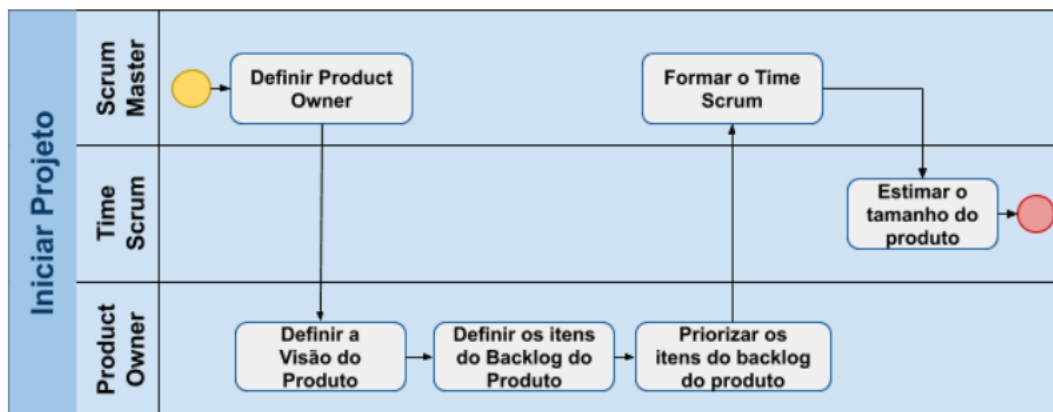
**Formulário de Solicitação** - Contém a Solicitação da Unidade Requisitante.

**Termo de Abertura do Projeto** - Caso a solicitação se desdobre em um projeto, será necessária a elaboração do referido termo que conterá: justificativa, produtos e serviços participantes, cronograma macro, restrições, premissas e assinaturas, conforme definido no Modelo de Gerenciamento de Projetos e Ações de TIC.



2.1.4 Iniciar Projeto

INICIAR PROJETO	
<b>Produtos de entrada</b>	<b>Critérios de entrada</b>
<ul style="list-style-type: none"> <li>Solicitação de Mudança de Iniciativa - SMI ou</li> <li>Termo de Abertura do Projeto - TAP</li> </ul>	<ul style="list-style-type: none"> <li>Solicitação de Mudança de Iniciativa: elaborado durante o processo de execução e controle, utilizado para requerer alterações de impacto na iniciativa. Define-se alteração de impacto, como aquela que irá modificar o custo, prazo ou escopo de maneira relevante. A solicitação deverá conter minimamente as seguintes informações: nome do Projeto; Nome e setor do solicitante; Data da solicitação; Título e descrição da Mudança; Motivo da solicitação (justificativa e benefícios); Impacto (de escopo, prazo e custo). A Seção de Planejamento Estratégico e Gestão de Projetos de TIC - PEGPTIC tem a incumbência de definir a relevância do impacto e a necessidade da emissão da SMI e posterior submissão ao Comitê Gestor de TIC (CGTIC), conforme definido no Processo de Gerenciamento de Projetos e Ações de TIC.</li> <li>Solicitação aprovada pelo Comitê de Governança de TIC segundo o Modelo de Gerenciamento de Projetos e Ações de TIC. O TAP já deve sinalizar se a solução proposta utilizará serviços de fábrica de software, aquisição, termo de cooperação técnica ou desenvolvimento interno.</li> </ul>
<b>Produtos de saída</b>	<b>Critérios de Saída</b>
<ul style="list-style-type: none"> <li>Visão do Produto</li> <li>Backlog do produto</li> </ul>	<ul style="list-style-type: none"> <li>Scrum Master definido;</li> <li>Product Owner definido e ciente das práticas ágeis que serão adotadas no projeto;</li> <li>Necessidade dos interessados x definição da solução alinhada;</li> <li>Time Scrum formado.</li> </ul>



#### 2.1.4.1 Definir Product Owner

Definição da pessoa responsável pelo produto, denominada Product Owner - P.O.

O P.O deve ser uma pessoa capaz de subsidiar as decisões e informações necessárias para a construção da solução a ser proposta e deve conhecer integralmente as necessidades do cliente.

Papéis	Atividades/Recomendações	Entradas	Resultado
Scrum Master	<ul style="list-style-type: none"> <li>É indicado que o P.O seja o usuário responsável pelo sistema, pois o seu comprometimento permitirá que ele entenda o processo e as dificuldades de alcançar o sucesso do projeto;</li> <li>Um Product Owner pode ser o próprio cliente, ou alguém indicado por ele, que tenha autonomia para tomar as decisões em seu nome.</li> </ul>	Termo de Abertura do Projeto - TAP	<ul style="list-style-type: none"> <li>Oficialização do Product Owner do Projeto;</li> <li>Visão do Produto Atualizado com a definição do Product Owner.</li> </ul>

#### 2.1.4.2 Definir a Visão do Produto

Definição da Visão do produto - juntamente com clientes, gestores, interessados, executivos entre outros - descrevendo o seu problema atual, as suas necessidades e as expectativas que subsidiarão a razão da existência do projeto.

Papéis	Atividades/Recomendações	Entradas	Resultado
<ul style="list-style-type: none"> <li>Product Owner</li> <li>Scrum Master</li> </ul>	Fazer a análise do problema identificando suas causas a fim de garantir o entendimento das necessidades, das expectativas, dos riscos e das características do produto.	Termo de Abertura do Projeto - TAP	Visão do Produto atualizada e aprovada pelo Product Owner.

#### 2.1.4.3 Definir os itens do Backlog do Produto

Identificação das funcionalidades e das condições de desenvolvimento para atender à visão do Product Owner.

Papéis	Atividades/Recomendações	Entradas	Resultado
Product Owner	<ul style="list-style-type: none"> <li>Pode ser envolvido um analista de sistemas para auxiliar na elaboração do modelo sintético de caso de uso, que ajude a identificar a relação entre as funcionalidades e as interações do sistema;</li> <li>As funcionalidades devem ser identificadas de maneira que seja possível realizar uma contagem estimada e uma decomposição funcional;</li> <li>Pode ser gerado um modelo sintético de caso de uso, porém não substitui a utilização da ferramenta de apoio ou artefato;</li> <li>O backlog sofrerá atualizações constantes durante o projeto devido ao amadurecimento dos requisitos do produto que será construído.</li> </ul>	Visão do Produto.	Backlog do produto macro e registrado na ferramenta de <b>apoio ou artefato</b> .

#### 2.1.4.4 Priorizar os itens do backlog do produto

Priorização dos itens do backlog do produto sob a perspectiva do Product Owner, os quais ajudarão na composição das Sprints conforme a produtividade do Time Scrum.

Papéis	Atividades/Recomendações	Entradas	Resultado
<ul style="list-style-type: none"> <li>Product Owner</li> <li>Arquiteto de software (opcional)</li> </ul>	A priorização das funcionalidades deve considerar os requisitos arquiteturalmente significativos para que o conceito da arquitetura seja colocado à prova.	<ul style="list-style-type: none"> <li>Visão do Produto;</li> <li>Backlog do Produto</li> </ul>	Backlog do produto atualizado com a priorização e indicação de impacto (ou não) na arquitetura;

#### 2.1.4.5 Formar o Time Scrum

Definição do Time Scrum que atenderá ao projeto, caso seja desenvolvido por equipe interna do TJPI. Se ele for desenvolvido por fábrica de software, o TJPI definirá quem comporá o time de acompanhamento do projeto.

Papéis	Atividades/Recomendações	Entradas	Resultado
Scrum Master	Definir um time entre 3 e 9 integrantes para que seja possível aplicar as técnicas ágeis de desenvolvimento de software.	<ul style="list-style-type: none"> <li>Visão do Produto;</li> <li>Backlog do Produto.</li> </ul>	Visão do produto atualizada com o Time Scrum que vai atender ao projeto ou acompanhar sua execução pela Fábrica de Software.

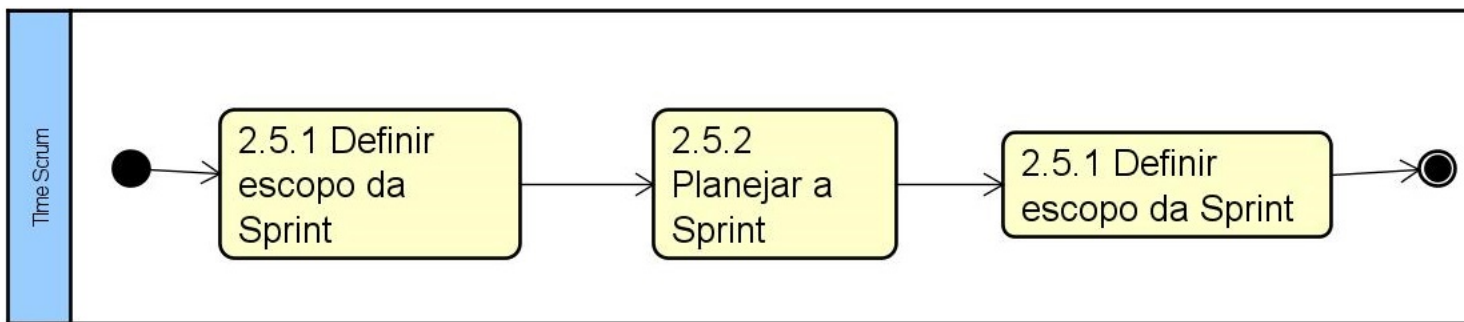
#### 2.1.4.6 Estimar o tamanho do produto

Estimativa do tamanho funcional do produto por pontos de história de usuário ou por análise de ponto de função. Caso o projeto seja atendido por fábrica de software, a métrica por ponto de função é contratualmente obrigatória.

Papéis	Atividades/Recomendações	Entradas	Resultado
<ul style="list-style-type: none"> <li>Product Owner</li> <li>Time Scrum</li> <li>Scrum Master</li> </ul>	<ul style="list-style-type: none"> <li>O esforço estimado para o backlog do produto por ponto de história de usuário deve ser realizado entre o Time Scrum e o Product Owner. O Planning Poker é uma técnica que pode ser utilizada, pois permite a visão de especialista sobre o tamanho da funcionalidade e estimula o diálogo entre os participantes durante as rodadas de estimativa;</li> <li>A estimativa também pode ser feita por meio da técnica de ponto de função.</li> </ul>	<ul style="list-style-type: none"> <li>Visão do Produto;</li> <li>Backlog do Produto.</li> </ul>	Backlog do produto atualizado com tamanho de cada item em ponto por história de usuário ou em ponto de função.

#### 2.1.5 Produzir a Sprint





PRODUZIR A SPRINT	
Executa as atividades de geração do produto esperado na meta da Sprint a fim de que seja possível apresentar resultados para o Product Owner. O acompanhamento das atividades da Sprint pode ser feito com o auxílio do Kanban ou do cronograma de atividades.	
Produtos de entrada	Critérios de entrada
<ul style="list-style-type: none"> <li>• Visão do Produto;</li> <li>• Backlog do Produto.</li> </ul>	<ul style="list-style-type: none"> <li>• Registro da iniciação do produto;</li> <li>• Time Scrum formado.</li> </ul>
Produtos de saída	Critérios de Saída
<ul style="list-style-type: none"> <li>• Visão do Produto Atualizada (Opcional).</li> <li>• Artefatos técnicos e de negócio atualizados: (Opcional)               <ul style="list-style-type: none"> <li>◦ Documento de arquitetura de software;</li> <li>◦ História de usuário detalhada;</li> <li>◦ Critérios de aceite;</li> <li>◦ Regras de negócio;</li> <li>◦ Protótipos;</li> <li>◦ Mensagens do sistema;</li> <li>◦ Glossário;</li> <li>◦ Demais artefatos selecionados pelo time.</li> </ul> </li> <li>• Produto construído, testado, validado e disponível para implantação em ambiente de produção;</li> <li>• Itens e meta da Sprint alcançados;</li> <li>• Defeitos e modificações identificados durante a homologação serão registrados e, se for o caso, como itens, para futuro sprint.</li> </ul>	<ul style="list-style-type: none"> <li>• Produto validado pelo Product Owner de acordo com critérios de aceitação e meta da Sprint;</li> <li>• Lista de defeitos ou adaptações identificadas.</li> </ul>

**2.1.5.1 Definir Escopo da Sprint**

Trata-se de uma das tarefas mais importantes do processo, durante a qual se reúnem o Product Owner e o Time Scrum para selecionar os itens do backlog do produto que comporão o escopo da entrega ao final da Sprint (Sprint Backlog), conforme a prioridade do Product Owner. Ainda, nessa tarefa, deve ser definida a meta da Sprint para orientar as ações do Time Scrum.

No caso de sistema novo, orienta-se que a primeira Sprint, denominada Sprint 0, a fim de validar a arquitetura e testar o ambiente de desenvolvimento, tenha como foco a definição da arquitetura do software, a construção do modelo de dados lógico ou conceitual, com base no backlog do produto, e a implementação de ao menos uma funcionalidade básica.

Papéis	Atividades/Recomendações	Entradas	Resultado
		<ul style="list-style-type: none"> <li>• Visão do Produto;</li> </ul>	

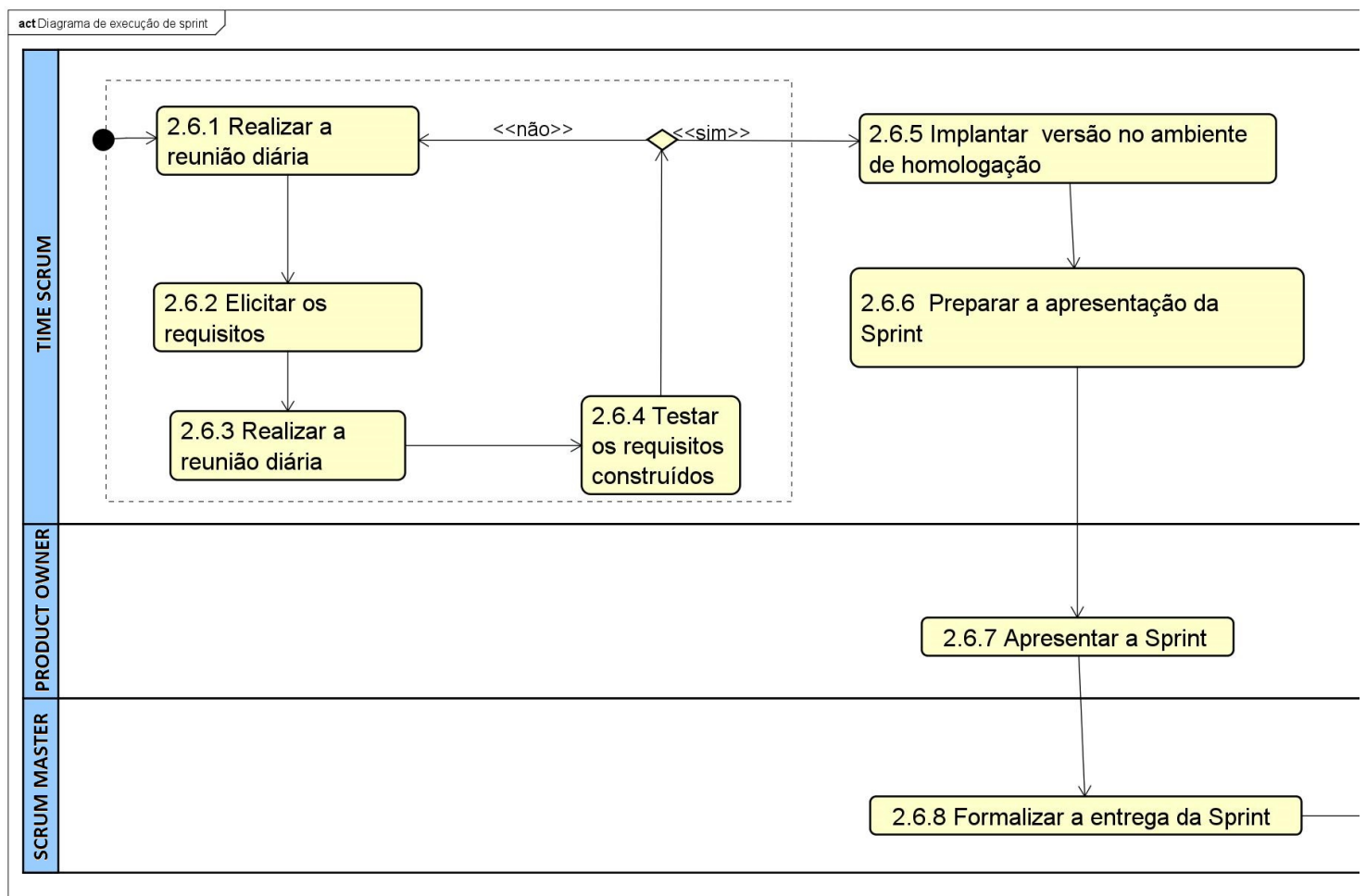
<ul style="list-style-type: none"> <li>Product Owner</li> <li>Time Scrum</li> </ul>	<ul style="list-style-type: none"> <li>A reunião de definição do escopo da Sprint deve levar, no máximo 4 horas, pois o objetivo é selecionar os itens e não planejar a execução;</li> <li>Entender o objetivo de cada item selecionado para a Sprint;</li> <li>Se for a Sprint zero, recomenda-se que conste no escopo: validação de arquitetura do software, elaboração do modelo de dados lógico ou conceitual e definição do plano de testes.</li> </ul>	<ul style="list-style-type: none"> <li>Backlog do Produto com itens priorizados;</li> <li>Documento de arquitetura de referência do TJPI.</li> </ul>	<ul style="list-style-type: none"> <li>Itens e meta da Sprint definidos (Sprint Backlog);</li> <li>Em caso de fábrica, solicitação de serviço encaminhada.</li> </ul>
---	--	--	---

### 2.1.5.2 Planejar a Sprint

Reunião com o objetivo de definir como os itens selecionados para a Sprint serão construídos visando criar um incremento e atingir sua meta. Para isso, o Time Scrum, com a participação do Product Owner, imagina, para cada item do Backlog, uma lista de atividades que possam ser executadas no período de um dia a fim de facilitar o acompanhamento diário.

Papéis	Atividades/Recomendações	Entradas	Resultado
<ul style="list-style-type: none"> <li>Product Owner</li> <li>Time Scrum</li> </ul>	<ul style="list-style-type: none"> <li>Decompor os itens da Sprint em tarefas;</li> <li>Podem ser definidos os responsáveis por executar cada tarefa, registrando essa informação junto à tarefa ou junto a um cronograma;</li> <li>Colocar em prática a estratégia definida para acompanhamento do projeto (montar o kanban).</li> </ul>	<ul style="list-style-type: none"> <li>Itens e Metas da Sprint definidos;</li> <li>Relatório Sintético de Modelo de Caso de Uso;</li> <li>Visão do Produto.</li> </ul>	<ul style="list-style-type: none"> <li>Atualização das informações de planejamento nos itens da Sprint;</li> <li>Em caso de fábrica, minuta da Ordem de Serviço.</li> </ul>

### 2.1.6 Executar a Sprint



Conjunto de tarefas a partir do qual o Time Scrum executa as atividades de construção dos itens da Sprint praticando as técnicas e princípios de métodos ágeis até que seja possível a conclusão do incremento do produto a ser entregue.

Papéis	Atividades/Recomendações	Entradas	Resultado
<ul style="list-style-type: none"> <li>Product Owner</li> <li>Time Scrum</li> </ul>	<ul style="list-style-type: none"> <li>Executar as tarefas conforme os padrões e as políticas definidas pelo TJPI provendo as adequações necessárias para atender às necessidades do projeto e do time.</li> </ul>	<ul style="list-style-type: none"> <li>Visão do Produto;</li> <li>Meta do Sprint;</li> <li>Itens da Sprint;</li> </ul>	<ul style="list-style-type: none"> <li>Produto construído e validado para entrega formal da Sprint.</li> </ul>

### 2.1.6.1 Realizar Reunião Diária

Realização de reuniões diárias com o Time Scrum, preferencialmente de curta duração e que possam ser feitas em pé, nas quais cada membro deve participar ativamente relatando as atividades executadas desde a última reunião, as atividades a serem executadas no dia e os impedimentos encontrados. Dessa maneira é possível ganhar visibilidade sobre o caminho para alcançar a meta da Sprint, atuar nos impedimentos relatados e planejar as atividades do dia seguinte.

Durante as reuniões, as questões que requeiram mais tempo de discussão devem ser tratadas separadamente com os envolvidos para não comprometer todo o time. Os impedimentos que configurarem riscos de caráter mais abrangente ou institucional devem ser registrados na ferramenta de gestão de projetos para monitoramento e, se for o caso, escalonamento.

Papéis	Atividades/Recomendações	Entradas	Resultado
<ul style="list-style-type: none"> <li>Product Owner (opcional)</li> <li>Scrum Master</li> <li>Time Scrum</li> </ul>	<ul style="list-style-type: none"> <li>Monitorar as atividades do dia e atualizar a situação da Sprint por meio do Kanban;</li> <li>Realizar reunião diária geralmente de 15 minutos, para que o time responda às seguintes perguntas: O que foi feito desde a última reunião? O que se pretende fazer até à próxima reunião? Teve ou está tendo algum impedimento?</li> <li>Levantar e atualizar os riscos que devem ser monitorados e/ou escalonados.</li> </ul>	<ul style="list-style-type: none"> <li>Itens da Sprint</li> </ul>	<ul style="list-style-type: none"> <li>Atividades da Sprint atualizadas e meta alinhada;</li> <li>Riscos atualizados na ferramenta de gestão de projetos;</li> </ul>

### 2.1.6.2 Elicitar os requisitos

Identificação, detalhamento, documentação e gerenciamento dos requisitos que compõem os itens da Sprint, conforme padrões de modelagem do projeto ou do Time Scrum.

Papéis	Atividades/Recomendações	Entradas	Resultado
<ul style="list-style-type: none"> <li>Product Owner</li> <li>Scrum Master</li> <li>Time Scrum</li> </ul>	<ul style="list-style-type: none"> <li>Entrevistar o Product Owner para detalhar o requisito a ser construído;</li> <li>Documentar o requisito detalhado nas histórias de usuários ou em outros artefatos de acordo com a necessidade avaliada pelo time;</li> <li>Definir os critérios de aceite para cada funcionalidade;</li> <li>Definir e documentar especificações de tela de interface;</li> <li>Validar, junto ao Product Owner, os requisitos e os critérios de aceite por meio de protótipos;</li> <li>Identificar e documentar os itens do glossário do sistema.</li> </ul>	<ul style="list-style-type: none"> <li>Visão do Produto;</li> <li>Itens da Sprint;</li> <li>Entrevistas;</li> <li>Brainstorming;</li> <li>Questionários;</li> <li>Ferramentas;</li> <li>Modelos de documentos.</li> </ul>	<ul style="list-style-type: none"> <li>Funcionalidade detalhada com histórias de usuário, critérios de aceite,</li> <li>protótipos e outros artefatos definidos para o projeto;</li> <li>Documentação mínima definida para o projeto produzida e/ou atualizada.</li> </ul>

### 2.1.6.3 Construir os itens da Sprint

Implementação dos requisitos com apoio dos padrões de arquitetura, de banco de dados e de design estabelecidos para o projeto, com a intenção de entregar algo que possa gerar resultado observável ao Product Owner.

Papéis	Atividades/Recomendações	Entradas	Resultado
<ul style="list-style-type: none"> <li>Time Scrum</li> <li>Product Owner (opcional)</li> </ul>	<ul style="list-style-type: none"> <li>Modelar Banco de Dados</li> <li>Produzir os modelos UML necessários para a realização das funcionalidades</li> <li>Atender aos padrões de arquitetura</li> <li>Programar os códigos para implementação do produto;</li> <li>Realizar testes unitários</li> <li>Construir as interfaces validadas pelo usuário</li> <li>Utilizar a infraestrutura de integração contínua a fim de possibilitar uma avaliação diária do trecho do software construído.</li> </ul>	<ul style="list-style-type: none"> <li>Itens da Sprint;</li> <li>Histórias de usuário, critérios de aceite e outros artefatos do projeto;</li> <li>Regras de Negócio;</li> <li>Protótipos de tela;</li> <li>Padrões de arquitetura, modelagem de dados e outros pertinentes;</li> <li>Políticas de gerência de configuração da STIC.</li> </ul>	<ul style="list-style-type: none"> <li>Funcionalidade codificada</li> <li>Documentos e artefatos do projeto atualizados</li> <li>Modelos de dados e scripts de banco de dados.</li> </ul>

#### 2.1.6.4 Testar os requisitos construídos

Aplicação das estratégias de teste definidas pelo projeto para minimizar os defeitos e não conformidades do produto no momento de sua verificação no ambiente de homologação, aumentando as chances de sua aprovação pelo Product Owner.

Papéis	Atividades/Recomendações	Entradas	Resultado
<ul style="list-style-type: none"> <li>Time Scrum</li> <li>Product Owner (opcional)</li> </ul>	<ul style="list-style-type: none"> <li>Realizar o teste unitário da funcionalidade;</li> <li>Preparar e realizar testes de integração e de regressão;</li> <li>Realizar teste de verificação e validação da funcionalidade;</li> <li>Os testes unitários, de integração e funcionais devem ser automatizados na medida do possível.</li> <li>Recomenda-se a utilização das ferramentas padrões disponíveis na STI para automatização dos testes, nos respectivos níveis.</li> </ul>	<ul style="list-style-type: none"> <li>Histórias de usuário, critérios de aceite e outros artefatos do projeto;</li> <li>Itens da Sprint aprovados pelo P.O.;</li> <li>Documentação mínima definida para o projeto;</li> <li>Plano e/ou estratégia de teste;</li> <li>Código-fonte, modelo de banco de dados e diagramas UML.</li> </ul>	<ul style="list-style-type: none"> <li>Produto validado com geração de evidências;</li> <li>Testes automatizados construídos e executados.</li> </ul>

#### 2.1.6.5 Implantar versão no ambiente de homologação do TJPI

Implantação, no ambiente de homologação do TJPI, de tudo o que foi construído para que haja validação por parte do Product Owner quanto à qualidade do produto entregue e à satisfação do cliente. Após cada implantação incremental do produto, será necessário realizar os procedimentos da construção contínua quanto à verificação dos testes automatizados (unitário, de integração e funcional), à auditoria do código-fonte e ao incremento da versão.

Papéis	Atividades/Recomendações	Entradas	Resultado
<ul style="list-style-type: none"> <li>Time Scrum</li> <li>Equipe de Infraestrutura (opcional)</li> </ul>	<ul style="list-style-type: none"> <li>Realizar a implantação do produto no ambiente de homologação;</li> <li>Realizar a entrega dos códigos-fonte, documentos e artefatos no repositório do projeto;</li> <li>Realizar a auditoria do código.</li> </ul>	<ul style="list-style-type: none"> <li>Requisitos elicitados;</li> <li>Código-fonte, modelo de banco de dados e diagramas da UML.</li> </ul>	<ul style="list-style-type: none"> <li>Produto implantado no ambiente de homologação do TJPI, pronto para a verificação por parte do Product Owner.</li> </ul>

#### 2.1.6.6 Preparar a apresentação da Sprint

Preparação do Time Scrum para a apresentação do resultado da Sprint ao Product Owner, elencando os problemas enfrentados e as soluções adotadas, além do alinhamento do time quanto ao alcance total ou parcial da meta da Sprint.

Papéis	Atividades/Recomendações	Entradas	Resultado

<ul style="list-style-type: none"> <li>• Time Scrum</li> <li>• Equipe de Infraestrutura (opcional)</li> </ul>	<ul style="list-style-type: none"> <li>• Verificar e validar se a meta da Sprint foi alcançada;</li> <li>• Registrar os problemas identificados;</li> <li>• Testar os itens da Sprint em ambiente de homologação.</li> </ul>	<ul style="list-style-type: none"> <li>• Produto implantado no ambiente de homologação do TJPI</li> </ul>	<ul style="list-style-type: none"> <li>• Alinhamento sobre o alcance da meta do Sprint;</li> <li>• Dificuldades listadas;</li> <li>• Lições aprendidas registradas na ferramenta de gestão de projetos.</li> </ul>
---	--	---	--

### 2.1.6.7 Apresentar a Sprint

Apresentação ao Product Owner das funcionalidades concluídas, demonstrando o que foi feito e o que não foi. O Product Owner aceitará (ou não) o produto como entregue, informando, se houver, as não conformidades e sua criticidade.

Se forem apontadas não conformidades críticas que comprometam a meta da Sprint, o Product Owner poderá optar por aceitar a Sprint com ressalvas e definir que as não conformidades sejam corrigidas nas próximas Sprints.

Caso o desenvolvimento seja feito por uma fábrica de software, essa apresentação é premissa para a formalização do aceite provisório dos produtos entregues, e os itens não aprovados serão medidos e não remunerados, voltarão para o Backlog do produto e aguardarão nova priorização.

Papéis	Atividades/Recomendações	Entradas	Resultado
<ul style="list-style-type: none"> <li>• Time Scrum</li> <li>• Product Owner Preposto</li> <li>• Gerente da FSW (opcional)</li> </ul>	<ul style="list-style-type: none"> <li>• Apresentar os resultados da Sprint;</li> <li>• Demonstrar o cumprimento da meta da Sprint;</li> <li>• Registrar as não conformidades ou sugestões verificadas durante a apresentação da Sprint.</li> </ul>	<ul style="list-style-type: none"> <li>• Produto implantado no ambiente de homologação do TJPI;</li> <li>• Itens e metas estabelecidos para a Sprint.</li> </ul>	<ul style="list-style-type: none"> <li>• Aceite total ou com ressalvas do produto pelo Product Owner;</li> <li>• Não conformidades registradas.</li> </ul>

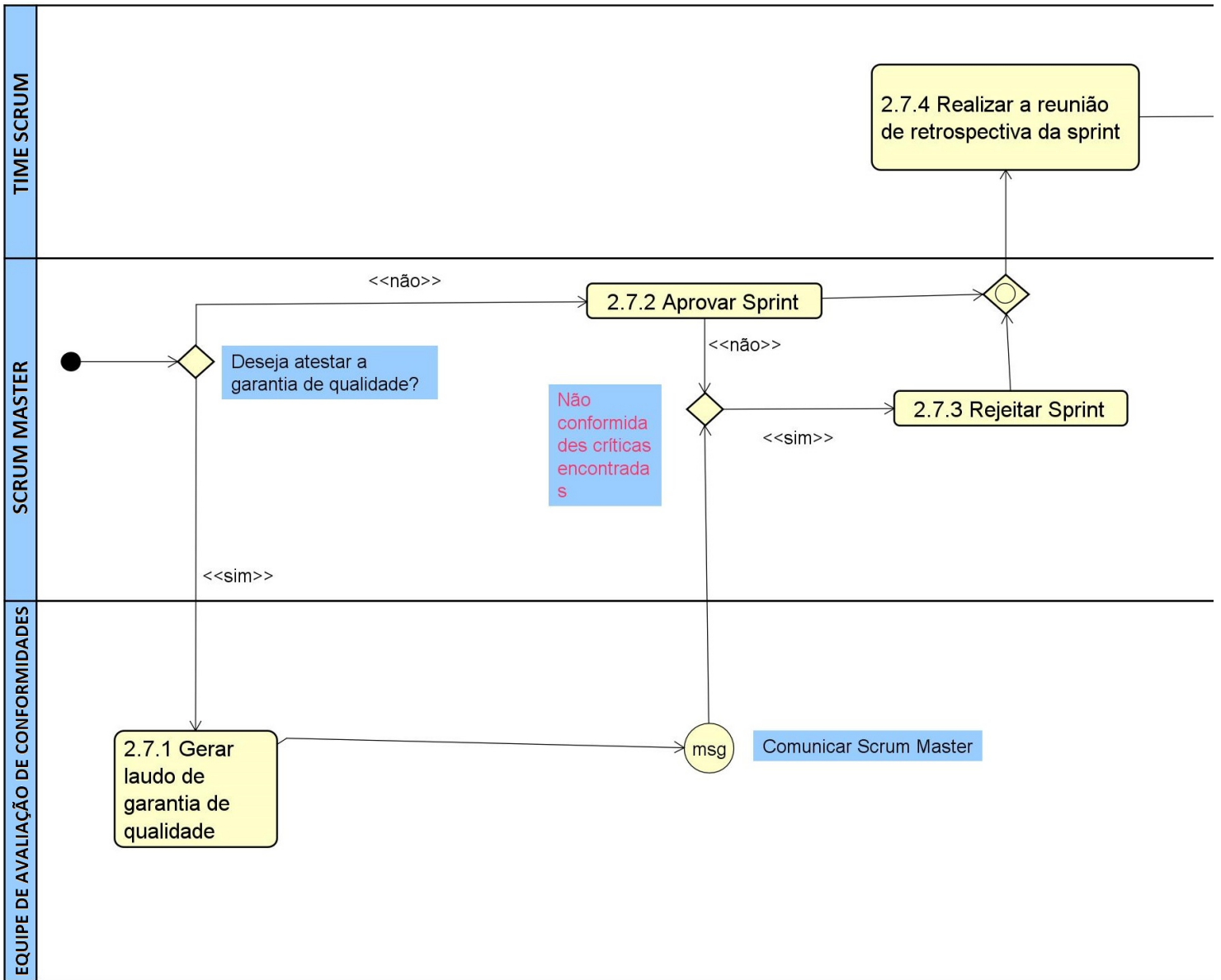
### 2.1.6.8 Formalizar a entrega do Sprint

O gestor do sistema formaliza o aceite da entrega da Sprint, por parte do Product Owner, com ou sem ressalvas quanto a possíveis não conformidades encontradas. Após o aceite, o incremento do produto se torna apto para ser implantado em ambiente de produção, caso esse tenha sido o acordo entre o Time Scrum e Product Owner.

Se o projeto utiliza serviços de fábrica de software, o aceite provisório dos produtos entregues em conformidade pela contratada deve ser formalizado.

Papéis	Atividades/Recomendações	Entradas	Resultado
<ul style="list-style-type: none"> <li>• Gestor do Sistema</li> <li>• Product Owner Preposto</li> <li>• Gerente da FSW (opcional)</li> </ul>	<ul style="list-style-type: none"> <li>• Formalizar o aceite da entrega da Sprint;</li> <li>• Reportar o aceite da apresentação do resultado da Sprint para a equipe de avaliação de conformidade.</li> </ul>	<ul style="list-style-type: none"> <li>• Produto implantado no ambiente de homologação do TJPI;</li> <li>• Repositório do projeto atualizado com todos os documentos, artefatos e código-fonte produzidos durante a Sprint.</li> </ul>	<ul style="list-style-type: none"> <li>• Entrega aceita pelo Product Owner e pronta para validação da qualidade através de conformidade</li> <li>• Termo de aceite provisório formalizado, no caso de fábrica de software.</li> </ul>

### 2.1.7 Encerrar a Sprint



ENCERRAR A SPRINT	
<p>Valida a qualidade do produto entregue, realiza inspeções e adaptações para a melhoria do projeto, dos procedimentos, das técnicas e do método de desenvolvimento. As questões a serem discutidas durante a retrospectiva da Sprint devem ser registradas constantemente para que não fique nada importante de fora.</p> <p>Se o projeto não utiliza fábrica de software, a única atividade obrigatória desta etapa é a 1.6.4 – Realizar reunião de retrospectiva da Sprint. Caso contrário, os laudos de conformidade a serem gerados serão definidos na respectiva ordem de serviço. O Scrum master pode optar por fazer a retrospectiva da Sprint antes da conclusão dos laudos de qualidade, porém esses geram subsídios importantes para a retrospectiva.</p>	
Produtos de entrada	Critérios de entrada
<ul style="list-style-type: none"> <li>• Produto Construído</li> <li>• Documentação gerada na Sprint</li> </ul>	<ul style="list-style-type: none"> <li>• Produto aprovado pelo Product Owner</li> </ul>
Produtos de saída	Critérios de Saída
<ul style="list-style-type: none"> <li>• Visão do Produto Atualizada;</li> <li>• Artefatos técnicos e de negócio atualizados:                             <ul style="list-style-type: none"> <li>◦ Documento de arquitetura de software;</li> <li>◦ História de usuário detalhada;</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• Qualidade do produto atestada pelos analistas de conformidade</li> <li>• Lições aprendidas registradas na ferramenta de gestão de projetos</li> </ul>

<ul style="list-style-type: none"> <li>◦ Critérios de aceite;</li> <li>◦ Regras de negócio;</li> <li>◦ Protótipos;</li> <li>◦ Mensagens do sistema;</li> <li>◦ Glossário;</li> <li>◦ Demais artefatos selecionados pelo time;</li> </ul> <ul style="list-style-type: none"> <li>• Produto construído, testado, validado e disponível para implantação em ambiente de produção;</li> <li>• Itens e meta da Sprint alcançados;</li> <li>• Defeitos e modificações identificados durante a homologação e registrados e, se for o caso, como itens, para futura sprint;</li> </ul>	
--	--

### 2.1.7.1 Gerar Laudo de Garantia de Qualidade

Verificação da qualidade dos produtos entregues quanto aos padrões corporativos e do projeto por meio da emissão de laudos de conformidade. Para projetos de fábrica de software, os laudos a serem emitidos serão indicados na ordem de serviço que autorizou a execução da Sprint. Projetos que não utilizam serviços de fábrica de software também podem solicitar emissão de laudos de qualidade, pois eles auxiliam o Scrum Master a atestar a qualidade dos produtos, assim como sua conformidade com os padrões definidos. Nesse caso, a emissão dos laudos deve ser previamente negociada com os analistas de conformidade.

Papéis	Atividades/Recomendações	Entradas	Resultado
<ul style="list-style-type: none"> <li>• Analista de Conformidade</li> <li>• Scrum Master</li> </ul>	<ul style="list-style-type: none"> <li>• Realizar a validação da arquitetura;</li> <li>• Realizar a validação de teste do sistema e a validação negocial;</li> <li>• Realizar a validação do banco de dados;</li> <li>• Realizar a validação de conformidade com processos e padrões;</li> <li>• Cabe ao Scrum Master avaliar os laudos emitidos e decidir pela adequação (ou não) das não conformidades encontradas.</li> </ul>	<ul style="list-style-type: none"> <li>• Produto disponível no ambiente de homologação do TJPI;</li> <li>• Repositório do projeto atualizado com todos os documentos, artefatos e código-fonte produzidos durante a Sprint.</li> </ul>	<ul style="list-style-type: none"> <li>• Laudos gerados, que primam pela qualidade e pelos padrões dos produtos entregues.</li> </ul>

### 2.1.7.2 Aprovar Sprint

Realiza a verificação dos laudos emitidos pelos analistas de conformidade e decide pela aprovação dos produtos entregues na Sprint. A aprovação pode ser integral ou parcial (com ressalvas). A aprovação parcial se dará quando as não conformidades apontadas pelos laudos forem admissíveis, em comum acordo com o Product Owner. Nesse caso, as não conformidades devem ser ajustadas pela contratada de acordo com a definição do Gestor do sistema.

Papéis	Atividades/Recomendações	Entradas	Resultado
<ul style="list-style-type: none"> <li>• Product Owner (opcional)</li> <li>• Scrum Master</li> <li>• Time Scrum</li> </ul>	<ul style="list-style-type: none"> <li>• Aprovar os produtos gerados pela Sprint, após análise dos laudos de conformidades previstos;</li> <li>• Reportar ao Time Scrum e às partes interessadas o resultado da aprovação da Sprint.</li> </ul>	<ul style="list-style-type: none"> <li>• Laudos de conformidade previstos na Sprint ou na ordem de serviço (se fábrica de software);</li> <li>• Laudo consolidado.</li> </ul>	<ul style="list-style-type: none"> <li>• Produtos da Sprint aprovados com ou sem ressalvas e liberados para implantação em produção.</li> </ul>

### 2.1.7.3 Rejeitar Sprint

Realiza a verificação dos laudos emitidos pela equipe de avaliação de conformidade e decide pela rejeição da qualidade do produto entregue na Sprint.

Caso o projeto utilize fábrica de software, o aceite definitivo dos produtos previstos na ordem de serviço não será concedido à contratada, e os itens retornarão ao Backlog do produto, aguardarão nova priorização e autorização para serem executados.

Nesse caso, recomenda-se avaliar a conveniência de abrir uma Sprint específica para entrega desses produtos rejeitados mediante abertura de nova ordem de serviço.

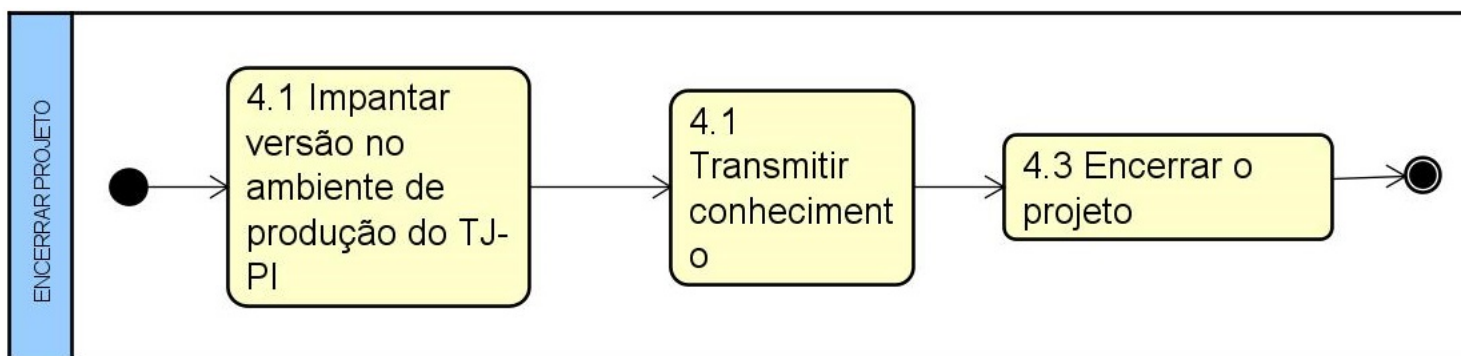
Papéis	Atividades/Recomendações	Entradas	Resultado
<ul style="list-style-type: none"> <li>Analista de Conformidade</li> <li>Product Owner (opcional)</li> <li>Scrum Master</li> <li>Time Scrum</li> </ul>	<ul style="list-style-type: none"> <li>Rejeitar os produtos entregues pela Sprint, caso existam não conformidades críticas e impeditivas, apontadas nos laudos de qualidade;</li> <li>Reportar ao Time Scrum e às partes interessadas o resultado da rejeição da Sprint;</li> </ul>	<ul style="list-style-type: none"> <li>Se opção pela verificação da garantia da qualidade: laudos de conformidade previstos na Sprint ou na ordem de serviço, juntamente com o laudo consolidado.</li> </ul>	<ul style="list-style-type: none"> <li>Produtos da Sprint rejeitados;</li> <li>Sprint não implantada em produção;</li> <li>Riscos do projeto atualizados na ferramenta de gestão.</li> </ul>

**2.1.7.4 Realizar reunião de retrospectiva da Sprint**

Realiza a revisão do processo de trabalho de execução da Sprint para identificar os itens que possam ser melhorados e, assim, tornar o processo mais agradável e eficiente na próxima Sprint.

Papéis	Atividades/Recomendações	Entradas	Resultado
<ul style="list-style-type: none"> <li>Analista de Conformidade (opcional)</li> <li>Time Scrum</li> <li>Gerente da Fábrica de Software (opcional)</li> </ul>	<ul style="list-style-type: none"> <li>Realizar inspeção no processo e propor adaptações para a sua melhoria quanto a atividades, pessoas, ferramentas, técnicas e etc.</li> <li>Discutir os pontos positivos e negativos da Sprint.</li> </ul>	<ul style="list-style-type: none"> <li>Situações ocorridas durante a Sprint.</li> </ul>	<ul style="list-style-type: none"> <li>Lições aprendidas e registradas na ferramenta de gestão de projetos.</li> </ul>

**2.1.8 Encerrar Projeto**



ENCERRAR PROJETO	
Obtém a transferência de conhecimento, garante que os usuários tenham condição de utilizar o produto e verifica se as expectativas do cliente foram atendidas.	
Produtos de entrada	Crítérios de entrada
<ul style="list-style-type: none"> <li>Produto Construído e aprovado pelo Product Owner;</li> <li>Repositório do projeto atualizado com toda a documentação gerada na Sprint.</li> </ul>	<ul style="list-style-type: none"> <li>Produto aprovado pelo Product Owner</li> </ul>



Produtos de saída	Critérios de Saída
<ul style="list-style-type: none"> <li>Sistema em ambiente de produção;</li> <li>Documentos de apoio à utilização do produto;</li> <li>Termo de Encerramento do Projeto;</li> <li>Lições aprendidas atualizadas.</li> </ul>	<ul style="list-style-type: none"> <li>Termo de Encerramento Projeto aprovado;</li> <li>Lições aprendidas atualizadas na ferramenta de gestão de projetos.</li> <li>Garantia das condições mínimas para utilização do sistema.</li> </ul>

### 2.1.8.1 Implantar versão no ambiente de produção do TJPI

Realiza a implantação do produto ou do seu incremento final no seu ambiente de produção.

Papéis	Atividades/Recomendações	Entradas	Resultado
<ul style="list-style-type: none"> <li>Time Scrum</li> <li>Equipe de Infraestrutura</li> </ul>	<ul style="list-style-type: none"> <li>Implantar o produto no ambiente de produção;</li> <li>O P.O. pode liberar módulos sucessivos do produto em produção no decorrer do projeto.</li> </ul>	<ul style="list-style-type: none"> <li>Produto aprovado e verificado no ambiente de homologação do TJPI.</li> </ul>	<ul style="list-style-type: none"> <li>Produto implantado no ambiente de produção, pronto para uso.</li> </ul>

### 2.1.8.2 Transmitir conhecimento

Transmite o conhecimento sobre o produto aos usuários finais, Service Desk e equipes de suporte a fim de proporcionar as condições mínimas necessárias para a sua operação, o seu suporte e o encaminhamento de chamados.

Papéis	Atividades/Recomendações	Entradas	Resultado
<ul style="list-style-type: none"> <li>Product Owner</li> <li>Time Scrum</li> <li>Usuário Final</li> <li>Equipes de suporte (opcional)</li> </ul>	<ul style="list-style-type: none"> <li>Transferir o conhecimento aos usuários finais;</li> <li>Transferir o conhecimento mínimo às equipes de suporte.</li> </ul>	<ul style="list-style-type: none"> <li>Produto implantado no ambiente de produção;</li> <li>Repositório do projeto atualizado.</li> </ul>	<ul style="list-style-type: none"> <li>Usuários aptos para operar o sistema;</li> <li>Equipes de suporte aptas a prestar suporte aos usuários do sistema ou a direcionar chamados.</li> </ul>

### 2.1.8.3 Encerrar o projeto

Esta Tarefa consiste em realizar o encerramento formal do projeto conforme o processo da Metodologia de Gestão de Projetos do Escritório de Projetos do Tribunal de Justiça do Piauí.

Papéis	Atividades/Recomendações	Entradas	Resultado
<ul style="list-style-type: none"> <li>Product Owner</li> <li>Gestor do sistema</li> <li>Scrum Master</li> </ul>	Realizar o encerramento do projeto mediante Termo de Encerramento - TEP.	Lições aprendidas	<ul style="list-style-type: none"> <li>TEP - Termo de Encerramento do Projeto aprovado na reunião com os interessados;</li> <li>Lições aprendidas atualizadas na ferramenta de gestão de demandas.</li> </ul>

### 2.1.9 Conjunto de Artefatos

Os modelos dos documentos serão disponibilizados na intranet e o seu preenchimento é de responsabilidade dos envolvidos no processo, conforme definido em cada atividade. A Seção de Planejamento Estratégico e Gestão de Projetos de TIC pode auxiliar na definição dos artefatos indicados para cada disciplina da engenharia de software, assim como, se for o caso, produzir novos artefatos ou versões atualizadas de produtos de trabalho existentes a fim de evoluir o processo e torná-lo cada vez mais útil para as equipes de projetos de desenvolvimento.

Alguns produtos de trabalho devem ser elaborados obrigatoriamente para o sistema desenvolvido ter o mínimo de documentação necessária para sua manutenção. Já os produtos de trabalho opcionais poderão ser utilizados conforme a definição do responsável do projeto. Os artefatos de um processo podem e devem ser utilizados de forma complementar a outros.

Abaixo, segue uma relação dos principais artefatos disponibilizados pelas metodologias vigentes no TJPI.

Nome do artefato	Gestão de Projetos	MDS	Testes
<b>ACOMPANHAMENTO GERENCIAL</b>			
Termo de Abertura do Projeto	X		
Termo de Encerramento do Projeto	X		
Reunião de apresentação da Sprint		X	
<b>REQUISITOS</b>			
Visão do Produto		X	
História de Usuário com critérios de aceite		X	
Regra de Negócio		X	
<b>ANÁLISE E DESIGN</b>			
Documento de arquitetura de software e infraestrutura (DASI)		X	
<b>TESTES</b>			
Solicitação de Testes			X
Liberação de Aplicativos para testes			X
Resultado de Teste			X
Plano de Testes iterativos			X

## 2.2 KANBAN

Kanban é uma abordagem ainda mais enxuta para o desenvolvimento ágil de software, e mostra-se mais adequada para atividades que não possibilitam planejamento por um tempo fixo, e nos casos em que atividades de planejamento e estimativas tornem o método moroso. É útil também para pequenas equipes, com fluxo de trabalho sob demanda, como manutenção e suporte de sistemas. Pode ser adotada para projetos de toda natureza, não apenas de desenvolvimento de software.

### 2.2.1 Premissas

2.2.1.1. Visualize o fluxo de trabalho: divida o trabalho em tarefas, para cada item gere um cartão e coloque no quadro de tarefas (o quadro Kanban), e use colunas nomeadas para ilustrar onde cada item está no fluxo de trabalho (backlog, fazendo, concluído, implantado, etc).

2.2.1.2. Defina o limite do trabalho em progresso (WIP - work in progress): associe limites explícitos para quantos itens podem estar em progresso em cada estado do fluxo de trabalho. A princípio esses limites são definidos, mas devem ser ajustados à medida em que se identifica gargalos ou folgas em cada etapa do fluxo. Aos poucos o WIP passa a ser reconhecido, em vez de definido, e assim passa a representar a realidade da equipe. A partir disso, pode-se implantar melhorias no processo a fim de otimizar o resultado do trabalho.

2.2.1.3. Identifique o tempo de execução da tarefa: identificar o tempo médio para completar um item, o chamado de "tempo de ciclo". Não se define o tempo para a tarefa, mas se percebe o tempo médio delas, com base no histórico de tarefas. Esse indicador é útil para monitoramento do processo e como feedback

das ações para melhoria do mesmo, devendo buscar o menor e mais previsível possível tempo de ciclo. Além disso, rever o tempo gasto em todo o fluxo nos permite uma maior fidelidade aos ANS - Acordos de Nível de Serviço (ou do inglês SLAs – Service-Level Agreements) e fazer planos de entrega mais realistas.

### 2.2.2 Artefatos

Dessa forma, os artefatos básicos no MDS-Kanban são o próprio quadro Kanban, com a organização visual das atividades, e as entregas propriamente ditas (artefatos entregues, que, em se tratando de software, é o software implantado). Porém não se restringe a isso, podendo utilizar alguns dos artefatos definidos para o Scrum.

### 2.2.3 Papéis

No contexto do TJPI, o Kanban necessita apenas do requisitante, de um líder de equipe (que pode ser o chefe de seção, coordenador, ou um membro designado por estes) e a equipe de trabalho. Esses papéis também existem no Scrum (respectivamente, PO, Scrum Master e time), e têm basicamente as mesmas funções.

### 2.2.4 Comparação entre SCRUM e KANBAN

Percebe-se que os dois métodos são ágeis, têm alguns aspectos em comum, e ainda a possibilidade de adotar práticas um do outro de modo a tornar o processo ainda mais adequado à realidade do projeto e cada vez mais eficaz. Suas diferenças de maior destaque são explanadas a seguir:

- Scrum é baseado em iterações de tempo fixo. No Kanban, iterações de duração fixa não são prescritas. É livre a escolha sobre quando planejar, melhorar processo e entregar. Pode-se escolher, por exemplo, fazer essas atividades numa periodicidade regular (“release toda segunda”), ou por demanda (“release sempre que tivermos algo útil a entregar”). Uma equipe orientada para eventos pode definir seu processo de trabalho da seguinte forma: "Nós Iniciamos uma reunião de planejamento sempre que esgotamos nosso trabalho a fazer. Iniciamos uma release sempre que temos um grupo Mínimo Aceitável de Funcionalidades (MAF's) pronto para entrega. Iniciamos um círculo de qualidade espontâneo sempre que nos deparamos com o mesmo problema pela segunda vez. Também fazemos uma retrospectiva mais profunda toda quarta semana."
- Kanban limita WIP por estado de fluxo de trabalho, Scrum por iteração. Então, tanto Scrum quanto Kanban limitam as atividades em andamento, mas de formas diferentes. Equipes Scrum geralmente medem a velocidade por iteração (quantos itens serão feitos ao longo do tempo). Uma vez que a equipe sabe sua velocidade, esta torna-se seu limite de atividades em andamento. Uma equipe que tem velocidade média de 10, geralmente, não irá colocar mais de 10 itens (ou “pontos da estória”) em uma iteração. Desta forma, em Scrum as atividades em andamento são limitadas por unidade de tempo. Em Kanban as atividades em andamento são limitadas pelo fluxo de trabalho (quantos itens podem ser feitos ao mesmo tempo, independente da cadência).
- Scrum resiste a mudanças dentro de uma iteração, enquanto o Kanban segue o princípio geral de “um item fora = um item dentro” (controlado pelos limites de trabalho em andamento).
- Em Scrum, o Product Owner não pode alterar o quadro Scrum quando a equipe já estiver comprometida com determinados de itens na iteração. Já em Kanban, você precisa definir suas próprias regras básicas sobre quem tem permissão de modificar o que no quadro, e normalmente o Product Owner pode fazer quaisquer modificações nos itens “A Fazer” ou “Backlog”.

### 2.3 Critérios para definição do método a ser adotado

Para facilitar a compreensão de qual método adotar, segue algumas questões a ser respondidas a fim de que se perceba qual a escolha mais adequada.

1. Trata-se de projeto ou ação priorizado segundo o Processo de Gerenciamento de Projetos e Ações de TIC? Scrum.
2. Trata-se de equipe de manutenção ou suporte a sistemas, que realiza tarefas sob demanda, pequenas, e atribuídas de forma pontual pelo coordenador da equipe? Kanban.
3. A equipe de trabalho é composta por três ou mais pessoas? Scrum.
4. A equipe de trabalho é composta por até três pessoas? Kanban.
5. O escopo do trabalho solicitado é definido e tem tamanho que permite um planejamento de trabalho com tempo fixo (duas a quatro semanas)? Scrum.
6. As prioridades mudam diariamente, é complicado passar alguns dias trabalhando num projeto sem ter de parar a todo instante para atender demandas aleatórias? Kanban.
7. Não há conhecimento de planejamento e gerenciamento de projetos e atividades? Kanban.

Tanto Scrum quanto Kanban são empíricos no sentido que se espera que o processo seja experimentado e personalizado ao ambiente. Nem um nem outro fornecem todos os mecanismos necessários para a realização do trabalho – eles apenas fornecem um conjunto básico de restrições para conduzir o seu próprio processo de melhoria. Então, é preciso adequar a escolha a cada contexto, desde que não se deturpe o processo, tornando-o tão personalizado que deixe de apresentar o mínimo definido para o TJPI (que é o Kanban).

## 3. DOCUMENTOS DE REFERÊNCIA

ID	Documento	Descrição
DR1	Acórdão TCU, 010.663/2013-4	Levantamento de Auditoria acerca da utilização de métodos ágeis nas contratações para desenvolvimento de Software pela administração pública.
DR2	Metodologia de Desenvolvimento Ágil - TSE	Método de Desenvolvimento com práticas Ágeis do Tribunal Superior Eleitoral.

DR3	Guia de Software Ágil - SISP	Guia de Projetos de Software com Práticas Ágeis da Secretaria de Tecnologia da Informação do Ministério do Planejamento, Orçamento e Gestão do Governo Federal.
DR4	Scrum Guide	Guia do Scrum
DR5	Kanban e Scrum - obtendo o melhor de ambos	Uma orientação para conhecimento e adoção de cada um dos métodos apresentados, ou ambos



Documento assinado eletronicamente por **José Ribamar Oliveira, Presidente**, em 09/04/2021, às 16:11, conforme art. 1º, III, "b", da Lei 11.419/2006.



A autenticidade do documento pode ser conferida no site <http://sei.tjpi.jus.br/verificar.php> informando o código verificador **2315091** e o código CRC **D1C4D471**.