



**MINISTÉRIO DO DESENVOLVIMENTO, INDÚSTRIA, COMÉRCIO E SERVIÇOS
INSTITUTO NACIONAL DA PROPRIEDADE INDUSTRIAL**

PORTARIA /INPI/DIRAD/CGTI Nº 8, DE 10 DE SETEMBRO DE 2024

Publica a Metodologia de Desenvolvimento
de Software

O COORDENADOR-GERAL DE TECNOLOGIA DA INFORMAÇÃO DO INSTITUTO NACIONAL DA PROPRIEDADE INDUSTRIAL, no uso das atribuições que lhe foram conferidas pelo Decreto nº 8.854, de 22 de setembro de 2016, e pela Portaria INPI PR nº 09, publicada em 13 de março de 2024, e tendo em vista o conteúdo do processo SEI nº 52402.009888/2024-01,

RESOLVE:

Art. 1º Publicar, na forma do anexo a esta Portaria, conforme definido no item 7.4 do Manual do Sistema de Padronização de Documentos GEQU-GDS-MN-0001, rev. 2.0, a Metodologia de Desenvolvimento de Software (1084389) e seus anexos GETI-GST-FP-0008 (1084395); GETI-GST-FP-0009 (1084399); GETI-GST-FP-0010 (1084401); e GETI-GST-FP-0011 (1084404).

Art. 2º Revogar a PORTARIA/INPI/CGTI nº 01, de 10 de novembro de 2023.

Art 3º Esta Portaria entra em vigor na data de sua publicação.

Marcus Vinicius da Motta Vieira
Coordenador-Geral de Tecnologia da Informação



Documento assinado eletronicamente por **MARCUS VINICIUS DA MOTTA VIEIRA, Coordenador(a) Geral**, em 20/09/2024, às 10:10, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site http://sei.inpi.gov.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **1086024** e o código CRC **AEB61014**.

Boletim Pessoal XV
do mês de Setembro
de 2024
Expedido em
20/09/2024

	SISTEMA DE PADRONIZAÇÃO DO INPI	Código	GETI – GST – MN – 0002
		Revisão	0.0
		Aprovação	09/09/2024
	METODOLOGIA DE DESENVOLVIMENTO DE SOFTWARE	Processo	Gestão de Soluções de TIC (Nível 2)

Sumário

1. Introdução	2
2. Objetivo	2
3. Abrangência	2
4. Documentos complementares	2
5. Glossário	2
6. Fundamentos Gerais	4
6.1. Projeto de Software.....	4
6.1.1. Serviços de desenvolvimento, manutenção e qualidade de software.....	4
6.2. Resultados esperados	5
6.3. Princípios Gerais.....	5
6.4. Práticas do método Scrum	6
6.5. Padrões e Requisitos	7
6.6. Papéis e Responsabilidades	8
7. Descrição dos processos ou atividades	10
7.1. Processo de desenvolvimento e manutenção de software.....	10
7.1.1. Planejar a sprint.....	10
7.1.2. Executar a sprint.....	12
7.1.3. Revisar a sprint.....	13
7.1.4. Entregar incremento	15
8. Entradas do processo	17
9. Saídas do processo	17
10. Fluxo do processo	17
11. Indicadores do processo.....	17
11.1. Indicador de Aceitação da Sprint/Entrega.....	18
11.2. Indicador de Produtividade Ágil.....	18
11.3. Indicador de Qualidade de código	19
11.4. Indicador de Conformidades em Homologação.....	20
11.5. Leadtime	20
12. Governança.....	21
13. Dono do documento.....	21
14. Outros elaborador(es) do documento	21
15. Aprovador(es) do documento.....	21
16. Bibliografia.....	21
17. Histórico das alterações	22
18. Anexos	22

1. Introdução

As metodologias de desenvolvimento de software consistem, basicamente, no conjunto de abordagens que podem ser utilizadas para a criação de sistemas de processamento de dados.

Os serviços de desenvolvimento e manutenção de software integram, no ramo da engenharia de software, o processo de ciclo de vida do desenvolvimento de software (*Software Development Life Cycle - SLDC*), ou seja, um processo contínuo e sistemático de construção que garante a qualidade e a correção precisa de falhas do software construído. Neste sentido, a continuidade dos serviços é uma característica inerente ao ciclo de vida do software assegurando a evolução dos requisitos, a disponibilidade e o desempenho das funcionalidades à medida que as necessidades de negócio e os recursos tecnológicos (físicos e lógicos) evoluem.

Neste manual é descrita a metodologia aplicável aos projetos de desenvolvimento e manutenção de software no INPI, contemplando atividades, fluxos, responsáveis e artefatos necessários ao ciclo de vida do projeto.

2. Objetivo

Definir um padrão para a gestão e o desenvolvimento de software no INPI, por meio de uma abordagem iterativa e incremental, adotando práticas ágeis, com o intuito de focar na qualidade da entrega do software e valor agregado para os clientes.

3. Abrangência

Este documento abrange todas as áreas que atuam diretamente no processo de desenvolvimento e manutenção de software no âmbito do INPI, devendo ser amplamente divulgado a todos os servidores, colaboradores e empresas prestadoras de serviços dessas áreas.

Igualmente, todas as áreas demandantes de soluções de tecnologia de informação precisam ter acesso ao documento e conhecimento de sua amplitude, a fim de que estejam cientes das etapas fundamentais para a implementação de um produto de software que atenda às necessidades pretendidas.

4. Documentos complementares

GETI – GST – IT – 0001_Abertura de demandas no Sistema Redmine.

GETI–GST–PP– 0001 – Detalhamento do Backlog do Produto (ou Histórias do Usuário) Ver. 0.0.

5. Glossário

Artefatos – Subprodutos produzidos durante o desenvolvimento de software. Ajudam a descrever a função, arquitetura e o design do software ou estão relacionados com o próprio processo de desenvolvimento.

Backlog da Sprint – conjunto de itens do Backlog do Produto selecionados para a Sprint

Backlog do Produto – representa tudo que é necessário para desenvolver e lançar um produto de valor agregado ao negócio. É uma lista priorizada de todos os requisitos (funcionais e não funcionais), funções, tecnologias, melhorias e correções de defeitos que constituem as mudanças que serão efetuadas no produto para versões futuras.

Bugs – Erros em sistemas.

CGTI – Coordenação Geral de Tecnologia da Informação.

COSIS – Coordenação de Sistemas.

DELPHI – Linguagem de programação para o desenvolvimento de sistemas.

DIPRO – Divisão de Padronização de Software.

DIREX – Diretoria Executiva.

GETI – Gestão de Tecnologia da Informação e Comunicações.

História de usuário - é uma explicação informal e geral sobre um recurso de software escrita a partir da perspectiva do usuário final ou cliente

Incremento de produto – versão de um produto que pode ser liberada no final de um período(timebox).

INPI – Instituto Nacional da Propriedade Industrial.

Metodologias ágeis – são um conjunto de práticas que visam a entrega rápida e de alta qualidade do produto ou serviço e que promovem um processo de gerenciamento de projetos que incentiva a inspeção e adaptação frequente, beneficiando a eficiência e efetividade dos gestores públicos no controle da prestação dos serviços de TI, haja vista que o foco passa a ser realmente nas atividades que entregam valor para as áreas de negócios.

PDTIC – Plano Diretor de Tecnologia da Informação e Comunicação.

Refatoração (*Refactoring*) – Processo de modificar um sistema de software para melhorar a estrutura interna do código sem alterar seu comportamento externo.

Release – distribuição/liberação de um incremento de produto para um cliente ou usuários. A quantidade de sprints por release deve ser definida previamente à execução dos serviços.

Timebox - Determina um limite de tempo para a conclusão de uma Sprint

Tipo de Tarefa: Sprint – Nesta categoria são classificadas as tarefas que representam as sprints definidas para implementar as histórias de usuário.

Tipo de Tarefa: Item da sprint – Nesta categoria são classificadas as tarefas que representam os itens de backlog da sprint.

Sprint – Corresponde a um grupo de atividades para transformar os itens de backlog em um incremento de software.

6. Fundamentos Gerais

A metodologia está alicerçada em 4 pilares:

1. Priorização balanceada: considera o valor agregado, visando promover práticas que permitam aos participantes do projeto desenvolver uma solução que maximize os benefícios e seja compatível com as restrições existentes.
2. Colaboração: alinhamento de interesses e compartilhamento do entendimento, visando promover um ambiente de equipe saudável que promova a colaboração e a compreensão do projeto.
3. Focar na arquitetura: concentração na arquitetura desde o início para minimizar riscos e organizar o desenvolvimento, além de promover práticas que minimizem os riscos na fase de desenvolvimento.
4. Evolução contínua e obtenção de feedback: promove práticas que permitem a equipe obter feedback de maneira contínua e o mais cedo possível, permitindo agregar valor a cada desenvolvimento e diminuir o risco do projeto conforme a execução.

6.1. Projeto de Software

Projeto de software é um serviço disponibilizado pela área de Tecnologia da Informação para atender às necessidades do Instituto. Como principais serviços nos projetos de desenvolvimento de software temos: desenvolvimento, manutenção de software e serviço de qualidade e teste avançado de software.

6.1.1. Serviços de desenvolvimento, manutenção e qualidade de software

Compreendem o conjunto de atividades executadas com a finalidade de atender às necessidades do órgão ou entidade por meio da implementação de um novo software, de uma nova funcionalidade ou manutenção de funcionalidades já existentes. Suas atividades incluem:

- a) aplicação de técnicas de engenharia de requisitos com vistas a identificar e especificar requisitos funcionais e não funcionais dos produtos a serem entregues;
- b) execução de procedimentos de design / arquitetura de software com vistas a estabelecer os padrões, tecnologias, formas de organização e de componentização dos recursos a serem utilizados na construção e manutenção dos sistemas;
- c) implementação dos códigos, componentes e recursos necessários à materialização do produto de software;
- d) planejamento da execução de testes de software para modelar e elaborar estratégias de testes;
- e) realização de testes de unidade, de integração, funcionais e testes de “fumaça”, com vistas a assegurar a qualidade do software;
- f) criação/manutenção do versionamento dos artefatos no repositório do INPI;
- g) realização da homologação dos produtos junto aos clientes, com vistas a certificar-se que o software atende aos requisitos esperados;
- h) realização da implantação dos produtos junto às áreas de operação e suporte de rede, ou áreas equivalentes, com o objetivo de assegurar a efetiva entrega do software em ambiente de produção;
- i) adoção das medidas necessárias para assegurar a disponibilidade, integridade, confidencialidade e autenticidade das informações a serem tratadas no âmbito da prestação dos serviços de desenvolvimento, manutenção, testes e controle de qualidade de software;

- j) adoção das medidas para garantir a proteção dos dados, antecipando ameaças à privacidade, à segurança e à integridade, prevenindo acesso não autorizado às informações disponibilizadas para prestação dos serviços de desenvolvimento, manutenção, testes e controle de qualidade de software;
- k) adoção de práticas de codificação segura;
- l) criação/atualização da documentação dos sistemas;
- m) mapeamento de problemas, cenários e soluções dos sistemas em produção;
- n) apoio técnico na busca contínua pela melhoria de processos das áreas;
- o) apoio ao desenvolvimento e manutenção de software, atuando para garantir que os softwares entregues tenham sido testados de acordo com as melhores práticas de mercado e os padrões previstos neste MDS, apontando as falhas e/ou oportunidades de melhoria nos processos de desenvolvimento e testes, bem como nos produtos resultantes destes processos;
- p) verificação da atualidade da documentação técnica dos sistemas;
- q) realização de diagnóstico de situações de gargalos e problemas de desempenho nos sistemas;
- r) proposição de melhoria da arquitetura dos sistemas visando garantir a arquitetura mais robusta possível;

6.2. Resultados esperados

Com a implementação desta Metodologia de Desenvolvimento de Software são esperados os seguintes benefícios:

- fornecer novos serviços digitais, relacionados ao escopo de atuação do Instituto;
- manter e aprimorar a disponibilidade, segurança, acessibilidade e usabilidade dos serviços digitais ofertados;
- aprimorar e evoluir os sistemas de informação que apoiam as políticas públicas de governo;
- assegurar a troca eficiente de informações entre os diferentes processos de negócio;
- assegurar a segurança e privacidade das informações mantidas e processadas pelos sistemas;
- manter a resiliência e eficiência dos sistemas de informação do Instituto;
- demonstrar agilidade no atendimento das demandas, mesmo que haja flutuações sazonais na quantidade de ordens de serviço;

6.3. Princípios Gerais

Os princípios elencados na enumeração abaixo devem ser utilizados:

I. PD1 - Primeiro Princípio Geral do Processo de Desenvolvimento de Software do INPI

Haverá apenas um processo de desenvolvimento de software para todo o Órgão, sob a responsabilidade da Divisão de Padronização de Software.

II. PD2 - Segundo Princípio Geral do Processo de Desenvolvimento de Software do INPI

O desenvolvimento de software é atividade exclusiva da CGTI.

III. PD3 - Princípio de Ferramentas e Instrumentos

A Divisão de Padronização de Software definirá as ferramentas e instrumentos padrões da metodologia de desenvolvimento de software, que deverão ser utilizados por todos os agentes.

IV. PD4 - Princípio de Viabilidade Técnica da Demanda

O atendimento da demanda somente poderá ser executado após a análise da viabilidade técnica da solicitação, sendo esta onerosa para o INPI ou não.

V. PD5 - Princípio de Prioridades

As demandas deverão seguir a determinação de prioridades estabelecida pelo Plano Diretor de Tecnologia da Informação e Comunicação (PDTIC vigente).

VI. PD6 – Princípio da Simplicidade

Elimine tudo aquilo que for desnecessário, pois ele onera e sobrecarrega o processo, as pessoas e o produto.

VII. PD7 – Princípio do Foco

Estabeleça um foco nos itens de solução, nas necessidades e nos problemas, para que a energia de trabalho se concentre apenas nos objetivos e acordos firmados.

6.4. Práticas do método Scrum

Para promover a entrega de projetos de software que cumpram as metas de qualidade, prazo e custos, os métodos ágeis preconizam uma série de valores e princípios distintos do desenvolvimento tradicional de software. O entendimento destes valores e princípios, bem como suas práticas, são fatores de sucesso. Todos os envolvidos no projeto devem conhecer princípios e práticas dos métodos ágeis.

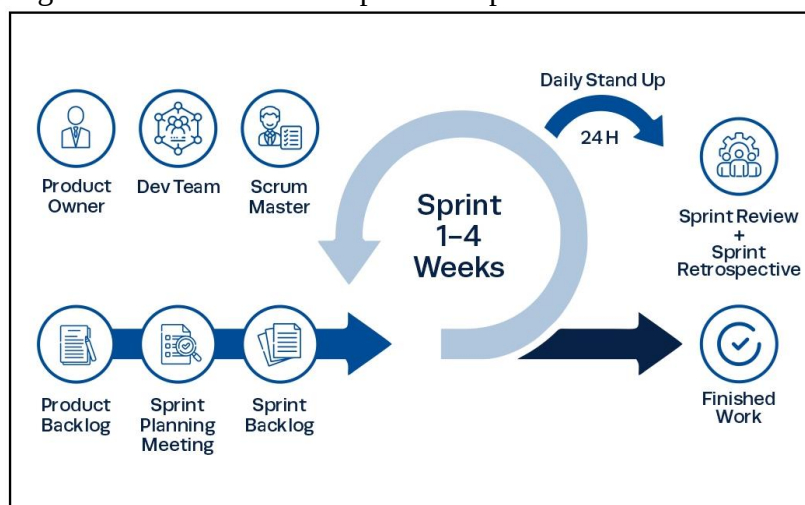
O processo de desenvolvimento ágil de software adotado baseia-se no método Scrum, sendo esta uma metodologia que propõe que um projeto seja dividido em diversos (pequenos) ciclos de atividades, com reuniões frequentes para que a equipe possa alinhar o que vem fazendo e pensar formas de melhorar o processo com agilidade.

Resumindo, o processo Scrum acontece em três etapas:

- O dono do produto analisa as necessidades e faz o backlog do produto;
- As equipes scrum trabalham em uma entrega de valor durante um período denominado sprint;
- As equipes scrum e as partes interessadas inspecionam e ajustam os resultados para o próximo sprint.

E então o processo é repetido, conforme Figura1.

Figura 1 - Fluxo do Scrum para um Sprint



Institute Project Management (2022).

6.5. Padrões e Requisitos

Deve-se observar, no que couber, os seguintes padrões:

- Lei Geral de Proteção de Dados - LGPD - Lei 13.709, de 2018;
- Princípios de *Security by Design*, *Privacy by Design* e *Shift-left testing*, de forma a minimizar os riscos de privacidade e segurança em tempo de concepção dos projetos;
- Padrão Digital de Governo (*Design System*) e suas atualizações com relação ao padrão visual das aplicações, disponível no endereço: <https://www.gov.br/ds/home>;
- Padrões de projeto (*Design Patterns*) ou padrões arquiteturais consolidados no mercado aderentes às necessidades da aplicação, além de métodos de codificação limpa (Clean Code);
- Abordagem TDD (*Test Driven Development*);
- ABNT NBR ISO/IEC/IEEE 12207:2021 - Engenharia de sistemas e software - Processos de ciclo de vida de software;
- ABNT NBR ISO/IEC 25030:2008 - Engenharia de software - Requisitos e Avaliação da Qualidade de Produto de Software (SQuaRE) - Requisitos de qualidade;
- ABNT NBR ISO 22301:2013 - Sistemas de gestão de continuidade de negócios;
- ABNT NBR ISO 22313:2015 - Sistemas de gestão de continuidade de negócios;
- ABNT NBR ISO 27031:2015 - Diretrizes para a prontidão para a continuidade dos negócios da tecnologia da informação e comunicação;
- ABNT NBR ISO 23081-1:2019 - Metadados para documentos de arquivo;
- ABNT NBR 11515:2007 - Guia de práticas para segurança física relativas ao armazenamento de dados;
- ABNT NBR ISO/IEC 27037:2012 - Diretrizes para identificação, coleta, aquisição e preservação de evidência digital;
- ABNT NBR ISO/IEC 27002:2013 - Código de prática para controles de segurança da informação;
- ABNT NBR ISO/IEC 27014:2013 - Governança de segurança da informação;
- ABNT NBR 16167:2013 - Diretrizes para classificação, rotulação e tratamento da informação;
- ABNT NBR ISO/IEC 27017:2016 - Código de prática para controles de segurança da informação com base;
- Guia de Gerenciamento de Vulnerabilidades e Modelo de Política de Gerenciamento de Vulnerabilidades - SGD/ME, disponível em <https://www.gov.br/governodigital/pt-br/seguranca-e-protecao-de-dados/ppsi>;
- Guia de Segurança em Aplicações Web - SGD/ME, disponível em <https://www.gov.br/governodigital/pt-br/seguranca-e-protecao-de-dados/ppsi>;
- Guia de Requisitos Mínimos de Segurança e Privacidade para APIs - SGD/ME, disponível em <https://www.gov.br/governodigital/pt-br/seguranca-e-protecao-de-dados/ppsi>;
- Guia de Requisitos Mínimos de Segurança e Privacidade para Aplicativos Móveis - SGD/ME, disponível em <https://www.gov.br/governodigital/pt-br/seguranca-e-protecao-de-dados/ppsi>;

6.6. Papéis e Responsabilidades

Tabela 1: Descrição de papéis e responsabilidades

Papel	Descrição	Responsabilidade
Cliente (INPI)	Servidor do INPI	• Descrever o projeto de acordo com as necessidades e resultados esperados; • Abrir demandas de manutenção; • Homologar implementação;
Analista TI (INPI)	Servidor da área de TI	• Analisar a solicitação de um novo projeto; criar projeto no sistema de controle de demandas do INPI (Redmine); • Validar a escrita das histórias de usuário; priorizar demandas; analisar e detalhar solicitações de novas funcionalidades; • Validar especificações e testar implementação; • Registrar defeitos relacionados às demandas e concluir demandas.
Scrum Master	Profissional com conhecimento aprofundado em técnicas ágeis.	• Garantir que o <i>Scrum</i> seja entendido e aplicado; • Assegurar que todos os eventos do Scrum têm lugar e são positivos, produtivos e mantidos dentro tempo previsto; • Apoiar o Dono do Produto e a organização na adoção de práticas ágeis; • Buscar melhoria contínua do time; • Facilitar a colaboração dos stakeholders conforme solicitado ou necessário; • Acompanhar e apresentar os indicadores do processo; • Remover impedimentos para a equipe de desenvolvimento durante a execução das Sprints. • Criar o Sprint Backlog;
Dono do Produto (Product Owner - PO)	Servidor representante da área demandante de soluções de software, designado por autoridade competente	• Responsável por ordenar o trabalho a ser realizado pelo time, criando, mantendo e priorizando o(s) backlog(s) do(s) produto(s); • Criar e compartilhar a Visão do Produto; • Planejar o Roadmap; • Construir o Backlog do Produto; • Expressar claramente os itens do Backlog do Produto; • Ordenar e priorizar os itens do

		Backlog do Produto; • Definir quais itens farão parte da sprint; • Garantir que o time de desenvolvimento entenda os itens do Backlog do Produto no nível necessário; • Apoiar no planejamento da Release; • Validar incremento de Software; • Validar a release ao final de cada sprint; • Reportar a Avaliação de Satisfação do Produto.
Desenvolvedores de software	Desenvolvedores que fazem parte do time ágil	• Construir o(s) produto(s) de software;
Partes interessadas (Stakeholders)	Profissionais impactados pela solução ou que possuam interesse na entrega da solução	• Opinar e contribuir para o planejamento e tomadas de decisão do negócio ou projeto; • Esclarecer dúvidas; • Se necessário, apoiar o PO na validação da sprint ou release.
Analista de Qualidade (INPI)	Servidor da área de TI	• Garantir a qualidade dos sistemas durante todo o ciclo do processo de software até a sua implantação, minimizando a ocorrência de erros no ambiente de produção;
Analistas de Teste e Qualidade	Analistas de Teste e Qualidade que fazem parte do time ágil	• Realizar a revisão de código, realização de testes avançados e revisão da qualidade da documentação produzida; • Aferir os critérios de aceitação da qualidade dos produtos entregues.
Analista de Negócio	Analistas de negócio que fazem parte do time ágil	• Desenhar solução para novos produtos e serviços, para posterior desdobramento em especificações funcionais para desenvolvimento de sistemas, com base em metodologia ágil. • Compreender necessidades de usuários, motivações e comportamentos, transformando informações em insights e features.

7. Descrição dos processos ou atividades

7.1. Processo de desenvolvimento e manutenção de software

7.1.1. Planejar a sprint (vide fluxo GETI-GST-FP-0008_Planejar a sprint_rev0.0)

Atividade: Criar tarefa do tipo Sprint	
Criar no sistema de abertura de demandas uma tarefa do tipo Sprint.	
Entrega: Tarefa Sprint criada no sistema	
Responsável	Participante
Analista TI	Analista TI
Tarefas executadas	Descrição
No sistema Redmine, criar a demanda do tipo Sprint	Criar demanda do tipo Sprint no sistema Redmine contendo informações básicas do sprint em questão.
Observação	

Atividade: Associar itens backlog do produto	
Realizar a associação da Sprint com os itens de backlog priorizados pelo Dono do produto	
Entrega: Itens de backlog associados à Sprint	
Responsável	Participante
Analista TI	Analista TI, Dono do Produto
Tarefas executadas	Descrição
Associar os itens de backlog à sprint que está sendo planejada	No sistema Redmine, realizar a associação dos itens de backlog à sprint planejada
Observação	

Atividade: Fragmentar em itens backlog de Sprint	
Definição da lista de itens backlog de sprint.	
Entrega: itens da sprint	
Responsável	Participante

Analista TI	Analista TI/Desenvolvedor/Analista de negócio
Tarefas executadas	Descrição
Detalhar os itens de backlog do produto, dividindo-os em itens.	Detalhamento dos itens que compõem uma sprint.
Observação	

Atividade: Estimar os itens backlog de sprint	
Estimar esforço dos itens de sprint.	
Entrega: esforço dos itens da sprint	
Responsável	Participante
Desenvolvedor	Analista TI/Desenvolvedor
Tarefas executadas	Descrição
Realizar estimativa do item da sprint	Estimativa do esforço do item da sprint
Observação	

Atividade: Definir itens da sprint	
Definir itens da sprint	
Entrega: sprint definida	
Responsável	Participante
Dono do Produto	Analista TI/Dono do Produto/Scrum Master
Tarefas executadas	Descrição
Definir itens da sprint	Definição de qual itens farão parte da sprint
Observação	

Atividade: Criar tarefa do tipo item da Sprint	
Criar no sistema de abertura de demandas uma tarefa do tipo item da Sprint.	
Entrega: Tarefa item da Sprint criada no sistema	
Responsável	Participante
Analista TI	Analista TI
Tarefas executadas	Descrição
No sistema Redmine, criar a demanda do tipo Sprint	Criar demanda do tipo Sprint no sistema Redmine contendo informações básicas do sprint em questão.
Observação	

7.1.2. Executar a sprint (vide fluxo GETI-GST-FP-0009_Executar a sprint_rev0.0)

Atividade: Codificar item	
Realizar a codificação do item da Sprint	
Entrega: Item da Sprint codificado	
Responsável	Participante
Scrum Master	Scrum Master/Desenvolvedor
Tarefas executadas	Descrição
Codificar o item da Sprint	Codificar o item da Sprint de acordo com as especificações definidas pelos clientes
Observação	

Atividade: Testar item	
Realizar testes relacionados à codificação do item da Sprint	
Entrega:	
Responsável	Participante
Scrum Master	Scrum Master/Desenvolvedor
Tarefas executadas	Descrição

Testes de codificação	Testes associados à codificação do item da Sprint desenvolvido
Observação	

Atividade: Entregar item	
Realizar a entrega do item da Sprint	
Entrega: Itens consolidados e entregues	
Responsável	Participante
Scrum Master/Desenvolvedor	Scrum Master/Desenvolvedor
Tarefas executadas	Descrição
Entregar o item da Sprint	Entrega do item da Sprint
Observação	

7.1.3. Revisar a sprint (vide fluxo GETI-GST-FP-0010_Revisar_sprint_rev0.0)

Atividade: Verificar qualidade	
Verificar a qualidade de software gerado por meio da consolidação dos itens da sprint validando os indicadores definidos.	
Entrega: Qualidade da demanda validada	
Responsável	Participante
Analista de Testes e Qualidade	Analista de Testes e Qualidade
Tarefas executadas	Descrição
Validar os indicadores de qualidade do software gerado na sprint	Teste do software implementado.
Observação	
A validação é realizada na release	

Atividade: Testar itens	
Testar a entrega da Sprint.	
Entrega: Sprint testada	

Responsável	Participante
Analista de Testes e Qualidade	Analista de Testes e Qualidade/Analista TI/PO
Tarefas executadas	Descrição
Testar todos os itens da Sprint	Itens da Sprint testados
Observação	

Atividade: Homologar itens	
Validar os itens da Sprint entregues	
Entrega: incremento aprovado	
Responsável	Participante
Dono do produto	Cliente/ Dono do produto
Tarefas executadas	Descrição
Validar o software como um todo a partir da implementação de cada incremento de software	Validação do produto
Observação	

Atividade: Entregar artefatos	
Entregar artefatos	
Entrega: artefatos	
Responsável	Participante
Scrum Master	Scrum Master/Desenvolvedor
Tarefas executadas	Descrição
Entregar os artefatos na branch Master	Entrega do produto
Observação	

Atividade: Verificar qualidade Master	
Verificar a qualidade de software gerado por meio da consolidação dos itens da sprint validando os indicadores definidos.	
Entrega: Qualidade da demanda validada	
Responsável	Participante
Analista de Testes e Qualidade	Analista de Testes e Qualidade
Tarefas executadas	Descrição
Validar os indicadores de qualidade do software gerado na sprint	Teste do software implementado.
Observação	
A validação é realizada na master	

Atividade: Encerrar Sprint	
Formalizar a finalização da Sprint.	
Entrega: Sprint encerrada	
Responsável	Participante
Analista de TI	Analista de TI
Tarefas executadas	Descrição
Encerrar a sprint informando se a mesma foi aprovada totalmente ou parcialmente	Sprint finalizada
Observação	

7.1.4. Entregar incremento (vide fluxo GETI-GST-FP-0011_Entregar incremento_rev0.0)

Atividade: Planejar a implantação da Sprint	
Realizar o planejamento da implantação do incremento de software	
Entrega: Plano para implantar incremento em produção	
Responsável	Participante

Dono do produto	Dono do produto /Analista TI
Tarefas executadas	Descrição
Planejar a implantação do incremento de software em produção	Implantação do incremento de software realizada
Observação	

Atividade: Implantar a Sprint	
Realizar a implantação do incremento de software no ambiente de produção	
Entrega: incremento implantado	
Responsável	Participante
Analista TI	Analista TI
Tarefas executadas	Descrição
Implantar incremento de software	Incremento de software implantado no ambiente de produção
Observação	

Atividade: Atualizar o backlog do produto	
Realizar a atualização do backlog do produto após a entrega do incremento de software	
Entrega: backlog do produto atualizado	
Responsável	Participante
PO	Cliente/PO
Tarefas executadas	Descrição
Atualizar backlog do produto	Backlog do produto atualizado com as novas funcionalidades implementadas no incremento de software
Observação	

Atividade: Realizar a retrospectiva da sprint	
Realizar a retrospectiva da sprint para avaliar os pontos positivos e negativos	
Entrega: Plano de melhoria contínua	
Responsável	Participante
Scrum Master	Cliente/Desenvolvedor/ Analista TI/PO/Scrum Master
Tarefas executadas	Descrição
Avaliar a última Sprint e criar um plano de ação para a próxima.	Evento que fecha a Sprint
Observação	

8. Entradas do processo

Itens de Backlog do Produto definidos na saída do processo “GETI–GST–PP– 0001 – Detalhamento do Backlog do Produto (ou Histórias do Usuário) Ver. 0.0”.

9. Saídas do processo

Artefatos produzidos durante o desenvolvimento e manutenção dos sistemas, conforme item 11 deste manual.

10. Fluxo do processo

GETI-GST-FP-0008_Planejar a sprint

GETI-GST-FP-0009_Executar a sprint

GETI-GST-FP-0010_ Revisar a sprint

GETI-GST-FP-0011_Entregar incremento

11. Indicadores do processo

Os seguintes indicadores são previstos neste documento:

- Indicador de Aceitação da Sprint/Entrega (IAS);
- Indicador de Produtividade Ágil (IPA);
- Indicador de qualidade de código (IQC);
- Indicador de Conformidades em Homologação (ICH);
- Leadtime.

11.1. Indicador de Aceitação da Sprint/Entrega

Verificar se as demandas planejadas nas sprints foram executadas no timebox e com qualidade, conforme tabela a seguir:

Tabela 1: IAS

Finalidade	Garantir a qualidade na entrega das sprints .
Meta a cumprir	IAS igual ou superior a 75%
Forma de acompanhamento	São apuradas a quantidade total de sprints entregues no período, a quantidade de sprints que foram aceitas integralmente e a quantidade de sprints aceitas parcialmente.
Periodicidade	Mensal
Mecanismo de cálculo (%)	É feita uma relação de proporção entre a quantidade de sprints aceitas integralmente e parcialmente junto ao total, chegando a um valor percentual: $IAS = \frac{(Q_i + Q_p/3) \times 100}{Q_t}$ Onde: IAS = Indicador de Aceitação da Sprint/Entrega; Q_i = Quantidade de sprints aceitas integralmente; Q_p = Quantidade de sprints aceitas parcialmente; Q_t = Quantidade total de sprints enviadas para aceite.
Observações	O peso das sprints aceitas integralmente deve ser maior que o das aceitas parcialmente. Nessa fórmula específica, o peso das sprints aceitas integralmente é três vezes maior que o das aceitas parcialmente. Para efeitos desse indicador, não são contabilizadas sprints rejeitadas, pois não atendem aos critérios mínimos de aceitação previamente estabelecidos.

11.2. Indicador de Produtividade Ágil

Monitorar o alcance das metas de produtividade.

Tabela 2: IPA

Finalidade	Garantir a produtividade das equipes ágeis, em termos do alcance de metas aferidas por meio de métricas de software , observando os critérios de qualidade e de aceitação definidos, bem como mensuração em termo de produto ou resultado entregue.
Meta a cumprir	IPA igual ou superior a 90%
Forma de acompanhamento	Afere-se a produtividade realizada no período, considerando as metas de produtividade em linhas de código
Periodicidade	Mensal
Mecanismo de cálculo (%)	IPA = 100 * (Pr / Pp) Onde: IPA = Indicador de Produtividade Ágil; Pr = Produtividade realizada no período para os perfis profissionais alocados, em função da métrica de software previamente estabelecida; Pp = Produtividade prevista no período para os perfis profissionais alocados, em função da métrica de software previamente

	estabelecida. Para calcular a Produtividade realizada no período (Pr) em linhas de código deve-se considerar a quantidade de linhas de código alteradas no software no período de aferição, com apoio de ferramenta. É considerada linha de código alterada uma nova linha inserida ou uma linha existente que foi modificada. Não serão contabilizadas linhas removidas, linhas em branco, linhas de comentários e alterações com intuito de somente aumentar artificialmente o número de linhas.
Observações	

11.3. Indicador de Qualidade de código

Assegurar a qualidade técnica das entregas. Utilizando, como por exemplo, SonarQube, é realizada uma análise estática do código para detectar bugs, duplicidade de código e vulnerabilidades de segurança, com exceção da linguagem DELPHI, conforme parâmetros definidos na tabela 3.

Tabela 3: Parâmetros de qualidade para primeira versão do código fonte

Grupo	Indicador	Unidade	Meta
Projeto	Complexity / file ou equivalente	média total	<= 10
	Complexity / class ou equivalente	média total	<= 10
	Complexity /function ou quivalente	média total	<= 3
	Duplications ou equivalente	%	<= 4%
	Security Issue Tags ou equivalente	unidades	=0
	Technical Debt ratio ou equivalente	%	<= 2,5%
	SQALE RATING ou equivalente	Nota	=A
Violações de código (possíveis bugs, estilo de codificação, más práticas de codificação)	Critical Issues ou equivalente	unidades	=0
	Blocker Issues ou equivalente	unidades	=0
Indicadores relacionados a testes	Unit Tests Coverage - camada negócio / Impl ou equivalente	%	>=70%
	Unit Test Success ou equivalente	%	=100%
	Skipped Tests ou equivalente	unidades	=0

O Indicador de qualidade de código (IQC) será calculado dividindo a Quantidade de requisitos de qualidade de código atendidos (ΣQ_{rc}) pela Quantidade total de requisitos de qualidade de código avaliados (ΣQ_{tr}), conforme descrito na tabela 4:

Tabela 4: IQC

Finalidade	Assegurar a qualidade do código em projetos de desenvolvimento, diminuir a ocorrência de defeitos e medir o nível de adequação do código fonte a características de qualidade determinadas pela contratante
Meta a cumprir	>= 90%
Forma de acompanhamento	A aferição será realizada por meio de ferramentas automatizadas
Periodicidade	Por período previamente definido, seja em termos de sprints

	executadas ou releases homologadas.
Mecanismo de cálculo (%)	$IQC = 100 * (Qrc / Qtr)$ Onde: IQC = Indicador de qualidade de código; Qrc = Somatório da Quantidade de requisitos de qualidade de código atendidos; Qtr = Somatório da Quantidade total de requisitos de qualidade de código avaliados.
Observações	A qualidade de código faz parte da visão dos desenvolvedores, engenheiros, arquitetos e, em alguns casos, analistas e gerentes. Indicadores da qualidade de código incluem: complexidade do código, duplicações de código, tamanho do código, entre outros. Vale ressaltar que a menor qualidade no código está relacionada a uma ocorrência maior de defeitos nas aplicações, que afetarão diretamente a produtividade da equipe de desenvolvimento. Esse indicador será utilizado desde o início do projeto, fazendo com que o código seja desenvolvido dentro de padrões aceitáveis de qualidade. Problemas de qualidade no código-fonte do software pré-existentes devem ser desconsiderados na aferição do IQC.

11.4. Indicador de Conformidades em Homologação

Assegurar o cumprimento dos prazos estabelecidos.

Tabela 5: ICH

Finalidade	Apura a quantidade de conformidades registradas pelo usuário durante a homologação do produto.
Meta a cumprir	ICH igual ou superior a 90%.
Forma de acompanhamento	É apurada a quantidade de itens da sprint entregues em conformidade aos requisitos mínimos de qualidade de código e atendimento aos requisitos funcionais no período de referência.
Periodicidade	Mensal
Mecanismo de cálculo (%)	$ICH = ((Qp - Qpe) / Qp) * 100$ Onde: ICH = Indicador de Conformidades em Homologação. Qpe = Quantidade de itens da sprint entregues com erros de codificação e/ou não implementação adequada dos requisitos funcionais que foram identificados no ambiente de homologação. Qp = Quantidade itens da sprint previstos.
Observações:	Não serão considerados erros identificados e reportados previamente pela equipe de testes e que não foram ajustados pela equipe responsável pelo desenvolvimento.

11.5. Leadtime

Como métrica para controlar quanto tempo uma tarefa leva para ser completada é adotado o Lead Time que é calculado com o número de dias entre o início e o fim de uma entrega. Esse intervalo é calculado conforme a demanda alcança diferentes status no processo de desenvolvimento:

• **Tarefas do tipo Sprint:**

INICIO_LEADTIME – Quando a demanda atinge a situação “Associar itens de backlog”

FIM_LEADTIME – Quando a demanda atinge a situação “Entregar itens da Sprint”

Com o monitoramento do Lead Time é possível detectar pontos de lentidão, antecipando oportunidades de melhorias por meio de ações específicas e direcionadas.

12. Governança

Não aplicável.

13. Dono do documento

Rhodrigo da Venda Santana, Chefe de Divisão, PR/DIRAD/CGTI/DIPRO

14. Outro(s) elaborador(es) do documento

Vladimir Guimarães de Carvalho, colaborador, PR/DIRAD/CGTI

15. Aprovador(es) do documento

Marcus Vinicius da Motta Vieira, Coordenador Geral, PR/DIRAD/CGTI

Pedro Calisto Luppi Monteiro Junior, Coordenador de Sistemas, PR/DIRAD/CGTI/COSIS

16. Bibliografia

IN SGD/ME nº 94, de 2022.

EPL - Empresa de Planejamento e Logística. **Processo de Desenvolvimento de Software da Empresa de Planejamento e Logística PDS – EPL Versão 2.0**. Disponível em: <https://www.epl.gov.br/pds>. Acesso em 26 de jul. 2021.

FINEP - Financiadora de Estudos e Projetos. **Metodologia de desenvolvimento de software da FINEP – MDS FINEP Versão 2.1**. 2017. Disponível em: http://www.finep.gov.br/images/licitacoes/2017/Consulta012017/I_MDS.pdf. Acesso em 26 de jul. 2021.

PJERJ - Poder Judiciário do Estado do Rio de Janeiro. **Metodologia de desenvolvimento de sistemas - Versão 2.0**. Disponível em: <http://www.tjrj.jus.br/documents/10136/6364473/anexo-a.pdf>. Acesso em 26 de jul. 2021.

SCRUM – Conhecimento de Scrum (Guia SBOK) – 3rd Edição. Disponível em: https://portaldaobmep.impa.br/uploads/material_teorico/d9gycv8klfcwc.pdf. 2017.

SUSEP – Superintendência de Seguros Privados. **Metodologia de Gestão e Desenvolvimento de Software**. 2011. Disponível em: http://www.susep.gov.br/setores-susep/noticias/download/tisusep/MGDS_SUSEP_v1.0.pdf. Acesso em 26 de jul. 2021.

TRT7 - Tribunal Regional do Trabalho da 7ª Região. **PDS-TRT7 Processo de Desenvolvimento de Software**. 2016. Disponível em:

https://www.trt7.jus.br/files/institucional/governanca_ti/processos/PDS-TRT7-rev-1_0-semAto.pdf.
Acesso em 26 de jul. 2021.

17. Histórico das alterações

Data	Nº revisão	Item	Descrição
01/08/2024	0.0	Todo documento	Emissão inicial

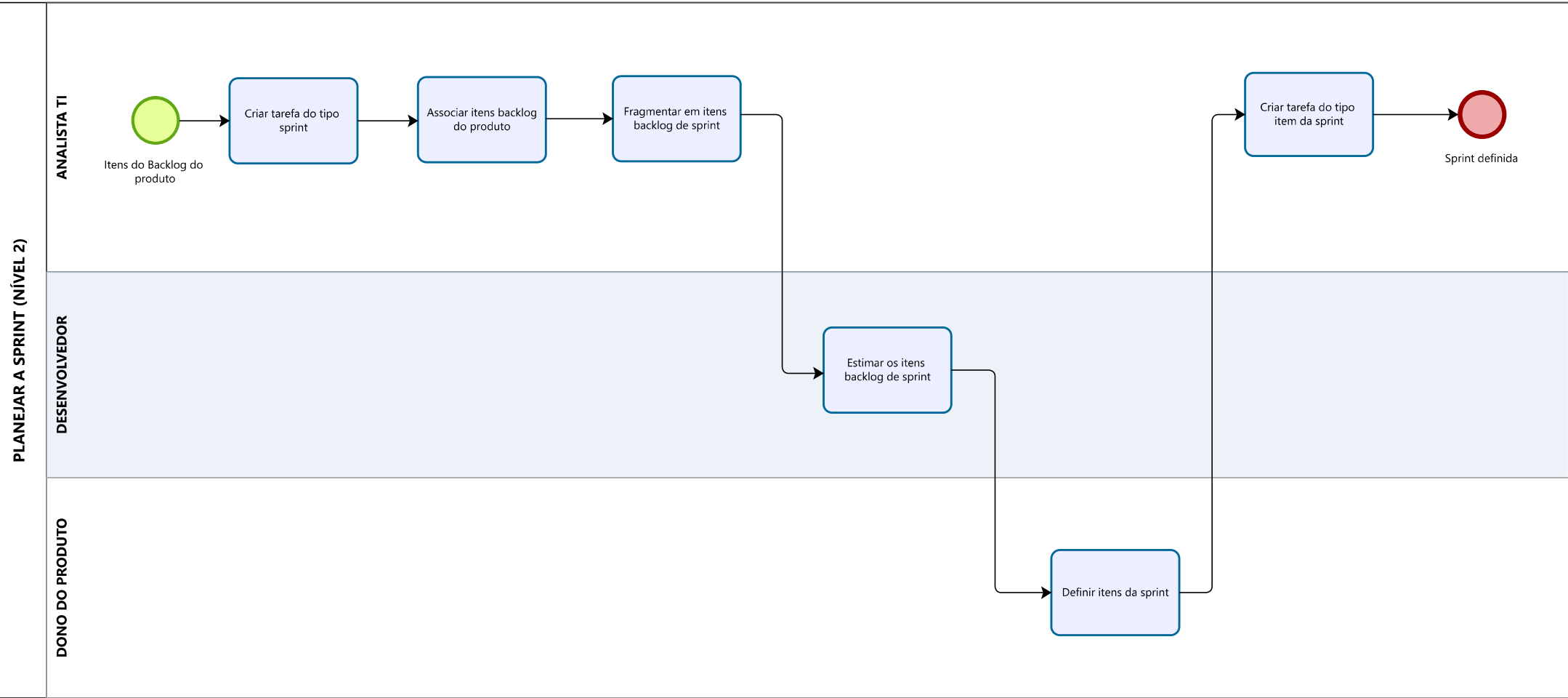
18. Anexos

Não aplicável.

Planejar a sprint (Nível 2 - AS IS)

Autor: Rhodrigo da Venda Santana, Chefe de Divisão, PR/DIRAD/CGTI/DIPRO
Versão: 0.0
Descrição: Código: GETI-GST-FP-0008
Data da aprovação: 09/09/2024
Breve descrição do processo:
Fluxo de planejamento da sprint.

Outro(s) elaborador(es):
Vladimir Guimarães de Carvalho, Colaborador G4F, PR/DIRAD/CGTI
Aprovador(es):
Marcus Vinicius da Motta Vieira, Coordenador Geral, PR/DIRAD/CGTI
Pedro Calisto Luppi Monteiro Junior, Coordenador de Sistemas, PR/DIRAD/CGTI/COSIS



RELAÇÃO DE SIGLAS:

HISTÓRICO DAS ALTERAÇÕES		
Nº da revisão	Data	Breve descrição das alterações
0.0	01/08/2024	Emissão Inicial

Executar a sprint (Nível 2 - AS IS)

Autor: Rhodrigo da Venda Santana, Chefe de Divisão, PR/DIRAD/CGTI/DIPRO

Versão: 0.0

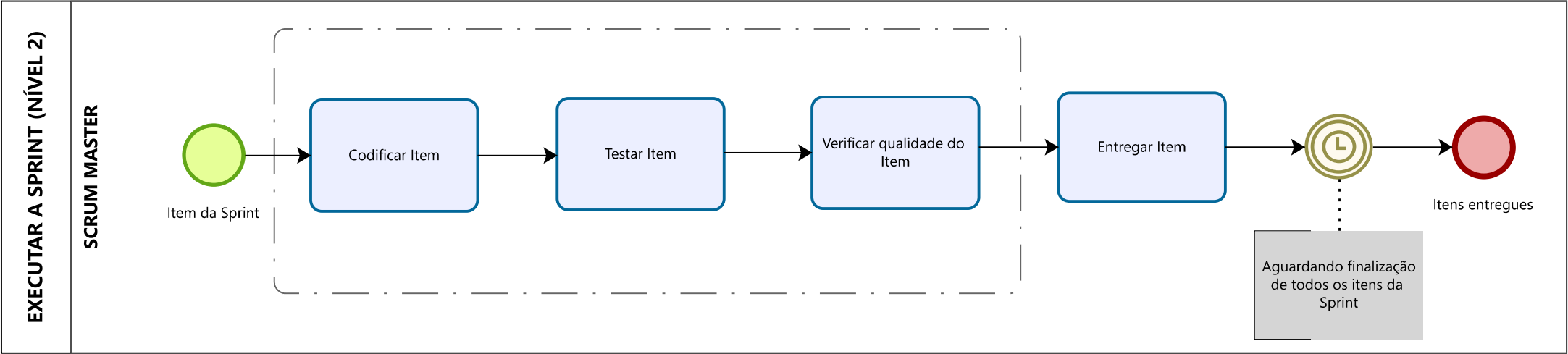
Descrição: Código: GETI-GST-FP-0009

Data da aprovação: 09/09/2024

Breve descrição do processo:
Fluxo de execução da sprint.

Outro(s) elaborador(es):
Vladimir Guimarães de Carvalho, Colaborador G4F, PR/DIRAD/CGTI

Aprovador(es):
Marcus Vinicius da Motta Vieira, Coordenador Geral, PR/DIRAD/CGTI
Pedro Calisto Luppi Monteiro Junior, Coordenador de Sistemas, PR/DIRAD/CGTI/COSIS



RELAÇÃO DE SIGLAS:

HISTÓRICO DAS ALTERAÇÕES		
Nº da revisão	Data	Breve descrição das alterações
0.0	01/08/2024	Emissão Inicial

Revisar a sprint (Nível 2 - AS IS)

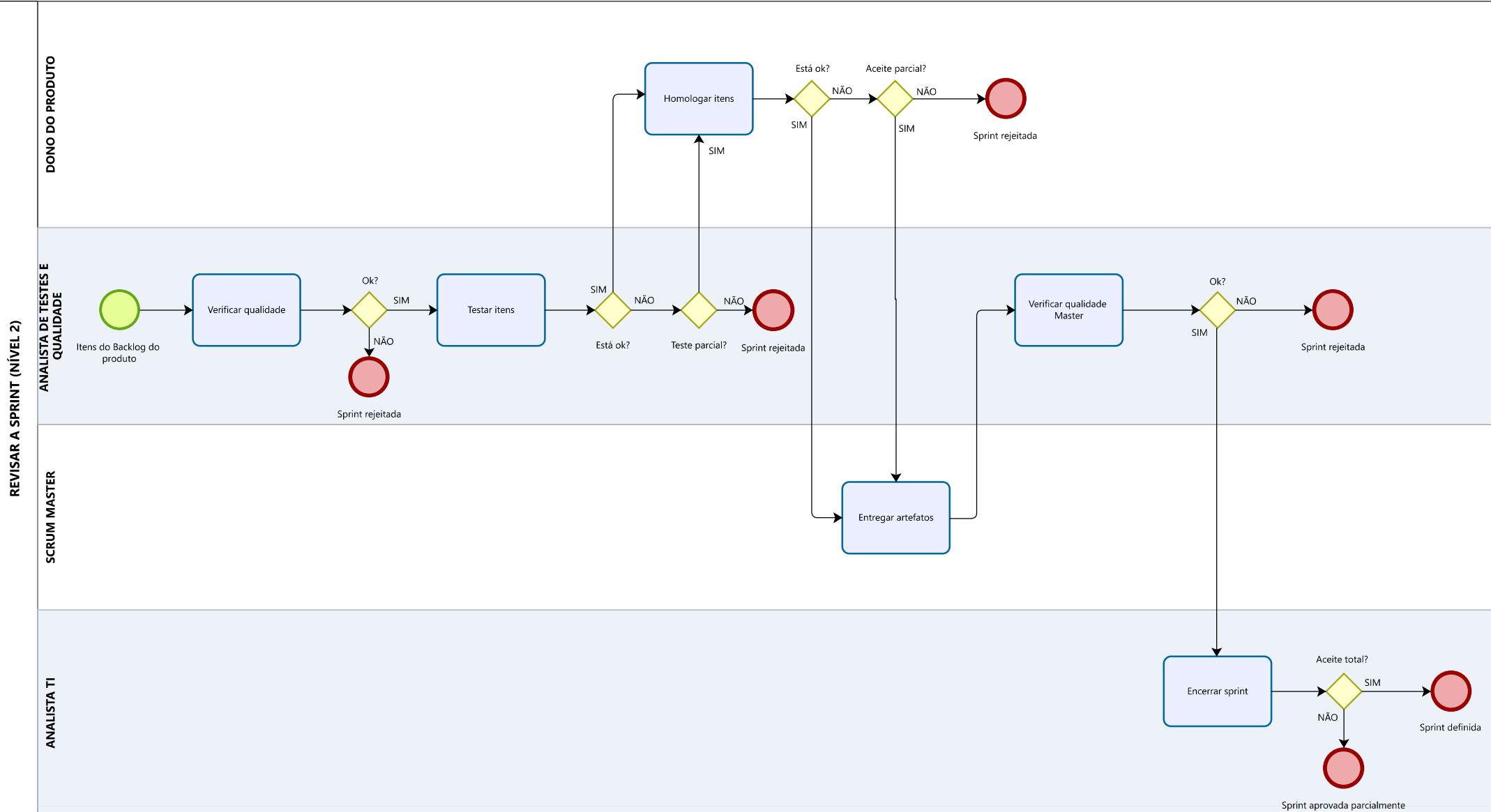
Autor: Rhodrigo da Venda Santana, Chefe de Divisão, PR/DIRAD/CGTI/DIPRO
Versão: 0.0
Descrição: Código: GETI-GST-FP-0010

Data da aprovação: 09/09/2024

Breve descrição do processo:
Fluxo de revisão da sprint.

Dutro(s) elaborador(es):
Vladimir Guimarães de Carvalho, Colaborador G4F, PR/DIRAD/CGTI

Aprovador(es):
Marcus Vinicius da Motta Vieira, Coordenador Geral, PR/DIRAD/CGTI
Pedro Calisto Luppi Monteiro Junior, Coordenador de Sistemas, PR/DIRAD/CGTI/COSIS



RELAÇÃO DE SIGLAS:

HISTÓRICO DAS ALTERAÇÕES		
Nº da revisão	Data	Breve descrição das alterações
0.0	01/08/2024	Emissão Inicial

Entregar incremento (Nível 2 - AS IS)

Autor: Rhodrigo da Venda Santana, Chefe de Divisão, PR/DIRAD/CGTI/DIPRO

Versão: 0.0

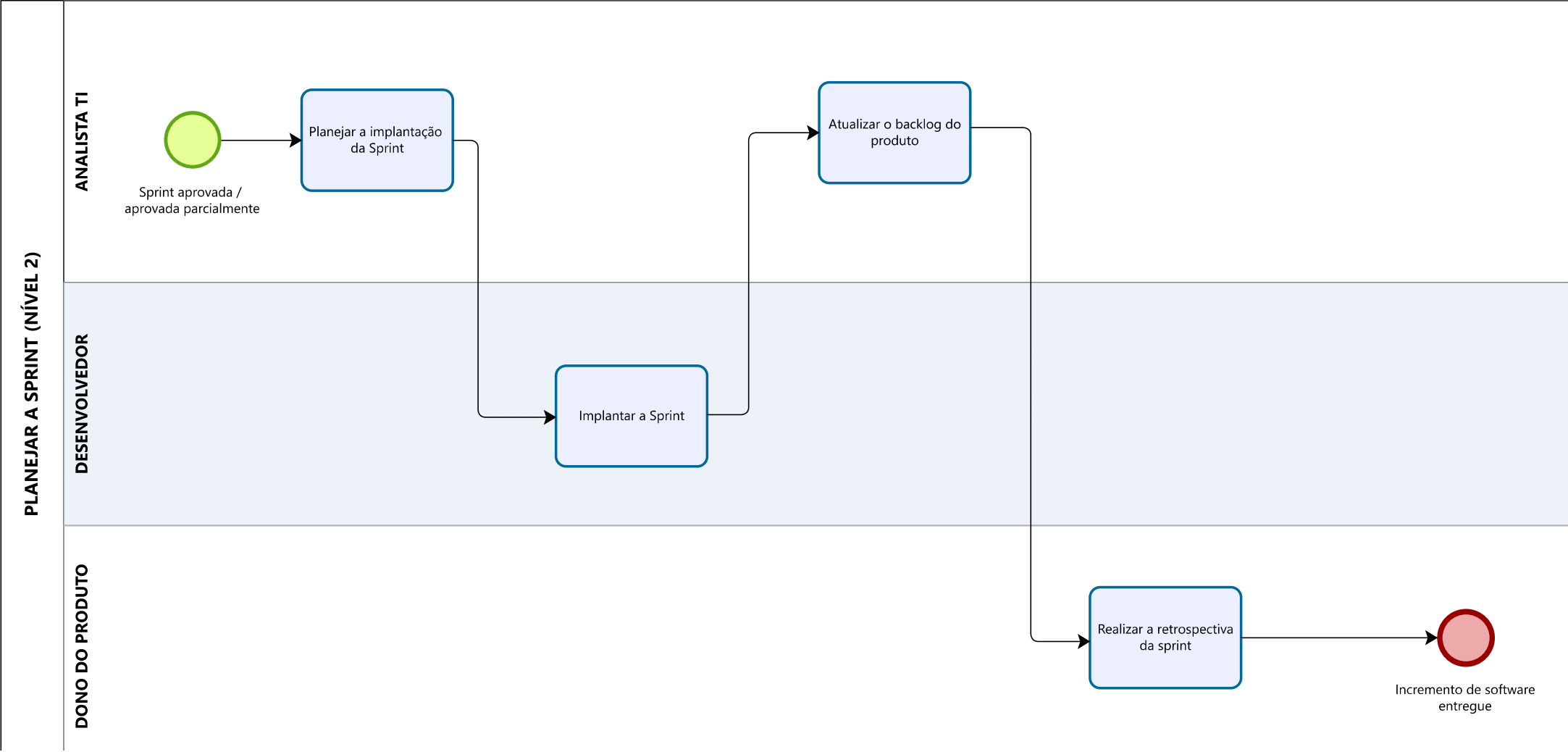
Descrição: **Código:** GETI-GST-FP-0011

Data da aprovação: 09/09/2024

Breve descrição do processo:
Fluxo de entrega de incremeto de software.

Outro(s) elaborador(es):
Vladimir Guimarães de Carvalho, Colaborador G4F, PR/DIRAD/CGTI

Aprovador(es):
Marcus Vinicius da Motta Vieira, Coordenador Geral, PR/DIRAD/CGTI
Pedro Callisto Luppi Monteiro Junior, Coordenador de Sistemas, PR/DIRAD/CGTI/COSIS



RELAÇÃO DE SIGLAS:

HISTÓRICO DAS ALTERAÇÕES		
Nº da revisão	Data	Breve descrição das alterações
0.0	01/08/2024	Emissão Inicial