



Metodologia de Desenvolvimento de Soluções de TIC



Imprensa Nacional

Diretor-Geral

Afonso Oliveira de Almeida

Diretor-Geral Adjunto

Alair José Martins Vargas

Coordenador-Geral de Tecnologia da Informação - CGTI

Marcos Vinicius Garcia Rodrigues Lima

Coordenador de Sistemas, Dados e Inovação - COSIS

Vitor Fonseca Farage

Chefe da Divisão de Dados e Inovação - DINO

Thaís Christina Moreira dos Santos

Chefe da Divisão de Desenvolvimento de Sistemas - DSIST

Marcus Paulo Albanex Andrade

Equipe Técnica da COSIS responsável pela elaboração

Vitor Fonseca Farage

Karolina Martins Milano

Marcus Paulo Albanex Andrade

Thaís Christina Moreira dos Santos

Histórico de Versões do Documento

CONTROLE			
DATA	VERSÃO	DESCRIÇÃO	AUTOR
13/09/2024	5.0A1	Elaboração do Documento	Equipe Técnica da COSIS
14/11/2024	5.0A2	Revisão pela CGTI	Coordenador-Geral
14/02/2025	5.0A3	Adição da seção Termos e Definições	Equipe Técnica da COSIS
07/02/2025	5.0	Correções textuais	Equipe Técnica da COSIS

ÍNDICE

1 APRESENTAÇÃO	6
2 TERMOS E DEFINIÇÕES.....	7
3 INTRODUÇÃO	8
4 METODOLOGIA DE DESENVOLVIMENTO DE SOLUÇÕES DE TIC E O PROCESSO DE SOFTWARE	9
4.1 Modelos de referência para o processo e definição da metodologia	9
5 PESSOAL ENVOLVIDO	10
6 PROCESSO PARA DESENVOLVIMENTO DE SOLUÇÕES DE TIC	12
6.1 Formalização da Demanda e Abertura do Projeto	14
6.2 Planejamento do Desenvolvimento do Produto.....	14
6.2.1 Elaboração da Visão do Produto e <i>Roadmap</i>	15
6.2.2 Concepção da lista de tarefas e funcionalidades do Produto (<i>Product Backlog</i>)	17
6.2.3 Estimativa do tamanho e esforço do <i>backlog</i> do produto	17
6.2.4 Planejamento do desenvolvimento e entrega do Produto por <i>Sprints</i> (<i>Sprint Planning</i>)	17
6.2.4.1 Lista de funcionalidades da <i>Sprint</i> (<i>Sprint Backlog</i>).....	17
6.3 Acompanhamento do desenvolvimento do Produto	18
6.3.1 Realização de Reunião Diária (<i>Scrum Daily</i>)	18
6.3.2 Realização da Reunião para Revisão da Sprint (<i>Sprint Review</i>)	18
6.3.3 Realização da Reunião para a Retrospectiva da <i>Sprint</i> (<i>Sprint Retrospective</i>).....	18
6.3.4 Produção de Relatório da Sprint.....	19
6.3.5 Gestão de Mudanças.....	19
6.4 Entrega dos Produtos	20
6.4.1 Critérios de Aceitação do Produto	20
6.4.2 Definição de Pronto	21
6.4.3 Liberação das Entregas.....	22
7 ARTEFATOS PRODUZIDOS.....	23
8 FERRAMENTAS	25
9 PROCESSO DE SUSTENTAÇÃO E MANUTENÇÃO DAS SOLUÇÕES DE TIC	25
9.1 Processo de Sustentação	26
9.1.1 Fluxo da demanda de Sustentação	26
9.2 Processo de Manutenção.....	27
9.2.1 Fluxo da demanda de Manutenção	27
9.3 Descontinuidade e Obsolescência do Produto	28
10 INSPEÇÃO DE QUALIDADE E TESTES AVANÇADOS	28
10.1 Testes e Controle de Qualidade do Software	29
11 CONSIDERAÇÕES FINAIS	30
12 REFERÊNCIAS BIBLIOGRÁFICAS.....	31

1 APRESENTAÇÃO

No intuito de desenvolver e manter soluções de TIC, é de suma importância descrever e normatizar os processos de gerenciamento, de desenvolvimento e de manutenção de sistemas, a fim de acompanhar e monitorar a execução de projetos para a construção e manutenção de soluções, com a intenção de garantir a entrega de produtos de TIC com eficiência e eficácia.

Sendo assim, foi elaborada a Metodologia de Desenvolvimento de Soluções de TIC, para utilização no âmbito da Imprensa Nacional, com a intenção de nortear o processo de desenvolvimento ou melhoria de soluções digitais que ofereçam serviços públicos de qualidade, no que tange a publicação de atos oficiais no Diário Oficial da União, a preservação da memória institucional e a promoção da transparência das ações do governo para a sociedade.

Destaca-se que foi considerado, fundamentalmente, no processo de elaboração deste documento, a consonância com as melhores práticas relacionadas às metodologias de desenvolvimento/engenharia de software, a fim de aumentar a transparência e a agilidade nas entregas de produtos de software. Para fins de entendimento desta metodologia, o termo produto pode se referir a um produto, serviço ou qualquer outra entrega associada a demandas de TIC, no que envolve *software*.

Com as diretrizes estabelecidas e a abordagem detalhada, orienta-se os passos e as linhas gerais para desenvolvimento ou melhoria de sistemas informatizados, a fim de especificar as tarefas, técnicas e produtos necessários a execução do ciclo de vida do projeto.

Desta forma, é possível padronizar as tarefas executadas em virtude da prestação de serviços especializados na área de tecnologia da informação, para fins de sustentação e desenvolvimento de sistemas. Ressalta-se que tais tarefas, independentemente de suas características, devem ser realizadas de acordo com os critérios e padrões descritos nesta metodologia.

É oportuno destacar que, estão incluídos nesta metodologia processos e atividades para a manutenção/sustentação de soluções em ambiente de produção da Imprensa Nacional. Estão abrangidas a execução de pequenas correções ou evoluções de curto prazo, e que não se caracterizam como projeto.

Esta metodologia pode ser revisada e alterada, a qualquer momento, para que o propósito seja mantido e continue atendendo às necessidades dos envolvidos no desenvolvimento de produtos/serviços de software.

2 TERMOS E DEFINIÇÕES

N.	Termo	Definição
1	Análise de Pontos de Função	Método de medida de tamanho funcional de software definido pela ISO/IEC 14143-1:2007, ISO/IEC 20926:2009, COSMIC (ISO/IEC 19761:2011), ou por métricas derivadas desses padrões internacionais como as contagens da <i>Netherlands Software Metrics Association</i> (NESMA) ou <i>Simple Function Point International Function Point Users Group</i> (SFP) do (IFPUG).
2	Aplicação	É um conjunto coeso de dados e procedimentos automatizados que suportam um objetivo de negócio, podendo consistir em um ou mais componentes, módulos ou subsistemas.
3	<i>Backlog</i> do produto	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.
4	Dívida Técnica	Consiste em decisões de codificação que atendem o projeto a curto prazo, mas que podem comprometer ou encarecer mudanças futuras, ou até mesmo inviabilizá-las.
5	Funcionalidade	Capacidade do produto de <i>software</i> de prover funções que atendam às necessidades explícitas e implícitas, quando o <i>software</i> estiver sendo utilizado sob condições especificadas.
6	História de usuário	Descrições simples, usualmente no padrão “Como um <papel>, eu desejo <objetivo> de modo que <benefício>”, que descrevem objetivos de negócio ou funcionalidades desejadas para um produto, escritas sob o ponto de vista do usuário.
7	Incremento de produto	Versão de um produto que pode ser liberada no final de um período de tempo (<i>time box</i>).
8	Manutenibilidade	Capacidade do produto de software de ser modificado. As modificações podem incluir correções, melhorias ou adaptações do <i>software</i> devido a mudanças no ambiente e nos seus requisitos ou especificações.
9	<i>Minimum Viable Product</i> - MVP	Versão simplificada de um produto, que contém apenas as funcionalidades básicas necessárias para funcionar e ser lançado no ambiente de produção e disponibilizado ao uso do público-alvo.
10	Padrões de Governo	Definições que devem ser cumpridas pelos entes públicos subordinados ao SIS, como Modelo de Acessibilidade em Governo Eletrônico (eMAG), Padrões Web em Governo Eletrônico (ePWG), Padrões de Interoperabilidade de Governo Eletrônico (ePING).
11	Portabilidade	A capacidade de um <i>software</i> apresentar o mesmo funcionamento em diversos ambientes, como sistemas operacionais, provedores de nuvem ou fornecedores.
12	<i>Roadmap</i>	Plano de ação que contempla o planejamento, especificação e implementação dos objetivos de negócio e histórias de usuário contidas nos releases, a serem distribuídas nos ciclos de desenvolvimento (<i>Sprints</i>).
13	Time Ágil	Equipe composta de profissionais multifuncionais que desenvolvem projetos de software, que devem possuir todas as competências necessárias para desenvolver, testar e entregar o produto de valor em um curto espaço de tempo.
14	Usabilidade	Capacidade do produto de <i>software</i> contar com os seguintes atributos: facilidade de aprendizagem, produtividade, facilidade de memorização, previsão de erros, recuperação ao estado anterior e agradável, para que os utilizadores fiquem satisfeitos com a sua utilização.
15	Usuário	É um agente humano (usuário final) ou um agente de <i>software</i> , que usa serviços e produtos de Tecnologia da Informação como um computador, um sistema ou serviços de rede.

3 INTRODUÇÃO

A Metodologia de Desenvolvimento de Soluções da Imprensa Nacional preconiza a disponibilização de um Processo de Desenvolvimento de Software e manutenção de software de referência, que se baseia no *framework* SCRUM, observa as diretrizes de processo de ciclo de vida do software constantes da ABNT NBR ISO IEC IEEE 12.207:2020 e está alinhado à Portaria SGD/MGI n.º 750, de 20 de março de 2023.

Conforme Portaria SGD/MGI n.º 750, de 20 de março de 2023, o processo de desenvolvimento de software deve prever uma fase inicial para o planejamento do projeto, que envolve a captura da visão do usuário, das necessidades e regras negociais, da definição do escopo do projeto e das principais funcionalidades do produto a ser desenvolvido (*backlog* do produto).

É importante frisar que os projetos ágeis devem ser elaborados com a participação de servidor ou profissional contratado com conhecimentos em metodologias ágeis, logo deve-se evitar o início da implementação do software sem o correspondente planejamento do produto a ser desenvolvido, pois a ausência desta ação pode gerar expectativas desalinhadas, além de acarretar retrabalho e aumento de custo na execução do projeto.

O processo de desenvolvimento de software descrito neste documento, aborda diferentes dimensões relacionadas ao ciclo de vida de construção e utilização de software, que podem ser adaptadas à realidade operacional da Imprensa Nacional.

Sendo assim, os principais objetivos que devem ser alcançados com a utilização desta metodologia são:

1. Entrega de produtos e serviços no **prazo**, e com a **qualidade** esperada pelo cliente interno;
2. Planejamento das entregas pautado nas necessidades de **cumprimento de objetivos e metas institucionais**;
3. Produtos e serviços frequentemente **testados e melhorados**;
4. Constante **automação e otimização** das funcionalidades implementadas;
5. Aderência aos **padrões de governo**, como Modelo de Acessibilidade em Governo Eletrônico (eMAG), Padrões Web em Governo Eletrônico (ePWG), Padrões de Interoperabilidade de Governo Eletrônico (ePING);
6. Adoção dos requisitos de **usabilidade** dos produtos e serviços;
7. Produtos e serviços pautados nas diretrizes de **segurança da informação**;
8. Produtos e serviços que suportem **portabilidade**.

4 METODOLOGIA DE DESENVOLVIMENTO DE SOLUÇÕES DE TIC E O PROCESSO DE SOFTWARE

Segundo Pressman (2002), o “*processo de software é definido como uma metodologia para as atividades, ações e tarefas necessárias para desenvolver um software de alta qualidade.*”. Adicionalmente destaca que, independentemente do paradigma de engenharia de software adotado, o processo de desenvolvimento de software contém três fases genéricas: definição, desenvolvimento e manutenção.

Na fase de **definição**, é levantada a intenção do que se quer desenvolver. São definidos o **escopo**, **restrições de projeto**, e **critérios de validação que estipulam um produto bem-sucedido**. Essa fase pode ainda ser subdividida em três outros estágios: a análise do problema, o planejamento do projeto e a análise de requisitos.

A fase de **desenvolvimento** o time de desenvolvimento define como a **estrutura de dados** e a **arquitetura de software** devem ser projetadas, envolve a escolha da **linguagem de programação**, **frameworks**, e demais tecnologias associadas à programação, bem como os **testes** que devem ser realizados.

Por fim, a fase de **manutenção** que se concentra nas mudanças associadas às correções de erros, adaptações produzidas por exigências variáveis dos usuários. São reaplicados os estágios das fases de definição e desenvolvimento.

Portanto, pautada por este conceito, a metodologia para o desenvolvimento de soluções de TIC da Imprensa Nacional é resultado da compilação de outras metodologias e boas práticas de desenvolvimento de software atualmente disponíveis na literatura, boas práticas empregadas no mercado privado e necessidades inerentes à atividade pública.

4.1 Modelos de referência para o processo e definição da metodologia

A Metodologia de Desenvolvimento de Soluções de TIC da Imprensa Nacional baseia-se em conceitos do *Lean Inception* e *Scrum*, que contemplam paradigmas do Manifesto Ágil, que, por sua vez, tem foco nas entregas e nos incrementos do produto.

Também incorpora atividades de desenvolvimento e operações (*DevOps*), bem como conceitos largamente documentados na literatura da Engenharia de *Software*.

A seguir são apresentados conceitos utilizados nesta metodologia para melhor compreensão dos processos de trabalho estabelecidos:

- a) **Lean Inception**: consiste na promoção de um *workshop* coletivo que tem por objetivo alinhar o entendimento das áreas de negócio e técnicas sobre um produto em seus aspectos mais fundamentais, a fim de definir o Produto Mínimo Viável (*Minimum Viable Product- MVP*) e seus incrementos.
- b) **Solução de TIC**: denominação genérica para produtos e/ou serviços de software que automatiza, parcial ou total, atividades de coleta, processamento, transmissão, armazenamento, recuperação e disseminação de dados que representam informação para o usuário ou cliente, ou para ambos. São exemplos de soluções de *software*: sistemas de informação, aplicativos, ferramentas, portais, sítios e *blogs*.
- c) **Scrum**: é uma estrutura ágil de colaboração em equipe multidisciplinar e autogerenciada, para entregar valor de forma incremental e colaborativa, de modo que o trabalho é dividido em pequenas metas a serem concluídas, dentro de iterações com limite de tempo (*time-box*), denominadas *Sprints*. Considera a experimentação contínua e ciclos de *feedback* ao longo do caminho para aprender e melhorar à medida que avança nas entregas.
- d) **DevOps**: conjunto de práticas para integração entre equipes de desenvolvimento de *softwares*, operações (infraestrutura) e de apoio envolvidas aliadas com a adoção de processos automatizados para produção rápida e segura de aplicações e serviços.

- e) **Desenvolvimento seguro:** ou SDL (do inglês *Security Development Lifecycle*), consiste na prática orientada à implementação de diversos mecanismos na programação do código-fonte para proteção contra as principais vulnerabilidades. Além disso, inclui mecanismos para evitar os ataques mais críticos a uma aplicação, que geralmente buscam obter acesso irrestrito ou acessar dados sensíveis gerenciados por ela.
- f) **Privacidade e Proteção dos dados:** no que tange a privacidade, é assegurar que o dado esteja protegido de interferência em assuntos pessoais e que o indivíduo possa controlar a exposição e a disponibilidade de informações acerca de si mesmo, o que é chamado de regulação dos limites. E sobre proteção de dados, entende-se o resguardo dessas informações contra ameaças externas e internas.

No desenho da solução tecnológica, deve-se equilibrar os aspectos: Viabilidade, Praticabilidade e Desejabilidade durante todo o processo de especificação.

- **Viabilidade:** significa avaliar se a solução proposta é viável do ponto de vista orçamentário, e se está alinhada à estratégia da Imprensa Nacional, logo será necessário verificar se está previsto ou se será necessário incluir no Plano Diretor de Tecnologia da Informação e Comunicação da Imprensa Nacional.

- **Praticabilidade:** analisar se a área de TIC da Imprensa Nacional possui as condições tecnológicas para desenvolvimento da solução, como contratos vigentes, cujo objeto contemple as tecnologias necessárias, com existência de saldo contratual para a execução do projeto.

- **Desejabilidade:** garantir que a especificação da solução será orientada pelas pessoas envolvidas no contexto, considerando o ponto de vista dos usuários, sejam servidores, colaboradores e outros indivíduos que porventura façam parte do contexto. As inovações propriamente ditas não envolvem necessariamente tecnologias de ponta, ou complexas, mas devem resultar o valor que o usuário da solução percebe ao fazer uso delas no cotidiano.

5 PESSOAL ENVOLVIDO

A Metodologia de Desenvolvimento de Soluções de TIC, que é influenciada pelo *Scrum*, é executada pela equipe denominada **Time Ágil**. A composição do time deve observar e adaptar-se à realidade operacional da Imprensa Nacional.

É composto por, no mínimo, **Scrum Master**, **Dono do Produto (Product Owner – PO)** e **Desenvolvedores**, que devem atender a diversos perfis especializados.

Sendo assim, os membros do Time Ágil devem minimamente:

- a) participar ativamente dos eventos do **Scrum** (tais como: Planejamento da **Sprint**, Reuniões diárias, Revisão da **Sprint**, Retrospectiva da **Sprint**);
- b) executar a **Sprint** e entregar incrementos de **software** ao final de cada **Sprint**;
- c) ser responsáveis por todas as atividades relacionadas com os produtos, desde a colaboração dos **stakeholders**, verificação, manutenção, operação, experimentação, investigação e desenvolvimento, assim como tudo o mais que possa ser necessário para realizar a entrega;
- d) adotar práticas de melhoria contínua;
- e) realizar testes funcionais e não funcionais concomitante ao desenvolvimento de software;
- f) apoiar na homologação das **Sprints** e **Releases**;
- g) elaborar manuais do usuário ou **help** de funcionalidades.

É fundamental que o dono do produto tenha conhecimento aprofundado no negócio, e condições de atuar no processo com representação efetiva das partes interessadas. O Guia SBOK (2017) vem corroborar com essa participação ao afirmar que:

“O Dono do Produto é responsável por garantir uma comunicação clara para o Time Scrum, sobre requisitos de funcionalidade do produto ou serviço, definindo os Critérios de Aceitação, e garantindo o cumprimento desses critérios. Em outras palavras, o Dono do Produto é responsável por garantir que o Time Scrum entregue valor.

A tabela a seguir apresenta os papéis e responsabilidade de todo o pessoal envolvido na consecução do projeto:

Composição do Time Ágil		
Papel	Perfil	Atividades principais e/ou relacionadas
Scrum Master	Profissional com conhecimento aprofundado em técnicas ágeis.	<ul style="list-style-type: none"> Garantir que o Scrum 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 a melhoria contínua do time; Facilitar a colaboração dos stakeholders conforme solicitado ou necessário; Atualizar o Gráfico de Burndown; Remover impedimentos para a equipe de desenvolvimento durante a execução das Sprints.
Dono do Produto (Product Owner – PO)	Servidor lotado nas áreas de negócio, representante da área demandante da solução de software , designado por autoridade competente e apoiado pelo Time Ágil.	<ul style="list-style-type: none"> 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; Garantir que o time de desenvolvimento entenda os itens do Backlog do Produto no nível necessário; Apoiar no planejamento do Release; Validar Incremento de Software; Validar software ao final de cada Sprint e Release; Reportar a avaliação de satisfação do Dono do Produto.
Desenvolvedores de software	Desenvolvedores que fazem parte do Time Ágil.	<ul style="list-style-type: none"> Criar a Sprint Backlog; Construir o(s) produto(s) de software.
Profissionais Especializados	Demais profissionais que integram o time com especialidades definidas a exemplo de Arquitetos, Analistas de Teste/Qualidade, Analista de Dados, etc.	<ul style="list-style-type: none"> Contribuir para a construção dos produtos de software.

Partes interessadas (Stakeholders)	Profissionais impactados pela solução ou que possuam interesse na entrega da solução.	<ul style="list-style-type: none"> • Opinar, influenciar, 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.
Analistas de Teste e Qualidade	Analistas de Teste e Qualidade que fazem parte do Time Ágil.	<ul style="list-style-type: none"> • 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; • Realizar a revisão de código, realização de testes avançados e revisão da qualidade da documentação produzida; • Apoiar a fiscalização técnica dos contratos de desenvolvimento, manutenção e sustentação de <i>software</i> na revisão técnica dos critérios de aceitação e de qualidade dos produtos entregues.

Tabela 1: Papéis e responsabilidades dos membros do Time Ágil

Os diversos profissionais que compõem o Time Ágil possuem vínculo com a IN, por meio de contratos terceirizado e servidores lotados na Coordenação-Geral de Tecnologia da Informação - CGTI:

- 1. Equipe COSIS:** servidores lotados na Coordenação de Sistemas, Dados e Inovação, que geralmente possuem, entre outras atividades previstas no Regimento Interno da Imprensa Nacional, o papel de gerir e fiscalizar os contratos firmados com empresas públicas e privadas, além de acompanhar todas as fases e entregas dos produtos/serviços desenvolvidos.
- 2. Equipe COING:** equipe de servidores lotados na Coordenação de Infraestrutura e Governança de TIC, que geralmente possuem, entre outras atividades previstas no Regimento Interno da Imprensa Nacional, o papel de gerir e fiscalizar os contratos firmados com empresas públicas e privadas, além de acompanhar todas as fases e entregas dos produtos/serviços desenvolvidos.
- 3. Equipe de colaboradores terceirizados:** formado por profissionais de TIC que tem por objetivo o desenvolvimento de soluções. Podem ser formados por profissionais da iniciativa privada, empresas públicas ou por profissionais que possuem vínculo com a Imprensa Nacional por meio de outros instrumentos contratuais.
- 4. Gestores de negócio:** são servidores lotados nas áreas de negócio (unidades organizacionais) da IN que são responsáveis por produtos e serviços de TIC utilizados no cumprimento da missão institucional da Imprensa Nacional.

6 PROCESSO PARA DESENVOLVIMENTO DE SOLUÇÕES DE TIC

Para o desenvolvimento do produto, é fundamental planejar e conhecer o problema a ser resolvido, ou oportunidade a ser aproveitada, para disponibilização de solução de serviços e produtos de TIC. Esse processo será tratado pela CGTI como projeto.

Logo, é necessário inicialmente que a área de negócio encaminhe a solicitação, com a descrição da necessidade de projeto, para a CGTI, que procederá com as ações e atividades para o desenvolvimento e entrega do produto.

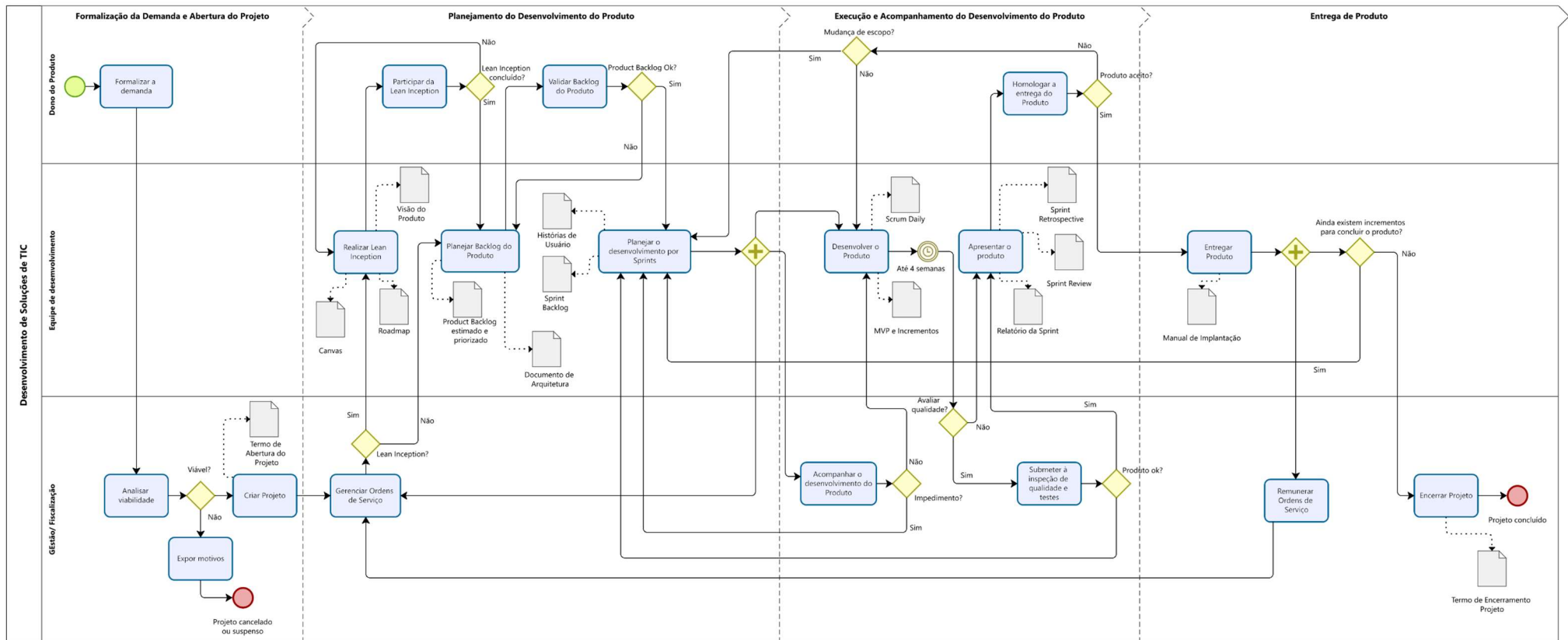


Figura 1: Processo para Desenvolvimento de Produtos

6.1 Formalização da Demanda e Abertura do Projeto

A **área de negócio**, requisitante do projeto de software, que pode ser qualquer unidade organizacional da Imprensa Nacional - IN, é representada pelo dono do produto (*Product Owner – PO*) e partes interessadas (*stakeholders*), e deve fornecer a compreensão do negócio, das necessidades, dos objetivos de negócio e dos requisitos do produto de software de interesse na unidade.

A **solicitação de projeto** deve ser registrada na CGTI, por meio de processo eletrônico, contendo um **ofício** com a solicitação do projeto, contendo:

- Alinhamento estratégico institucional;
- Expectativa de entrega da primeira MVP;
- Indicação do dono do produto e seu eventual substituto.

A COSIS providenciará a confecção do **Termo de Abertura do Projeto**. A partir daí, a equipe técnica, que é composta pelo time fornecido por empresas contratadas e servidores da IN, será responsável pelo desenvolvimento do produto e pelas atividades de planejamento do projeto juntamente com o dono do produto.

6.2 Planejamento do Desenvolvimento do Produto

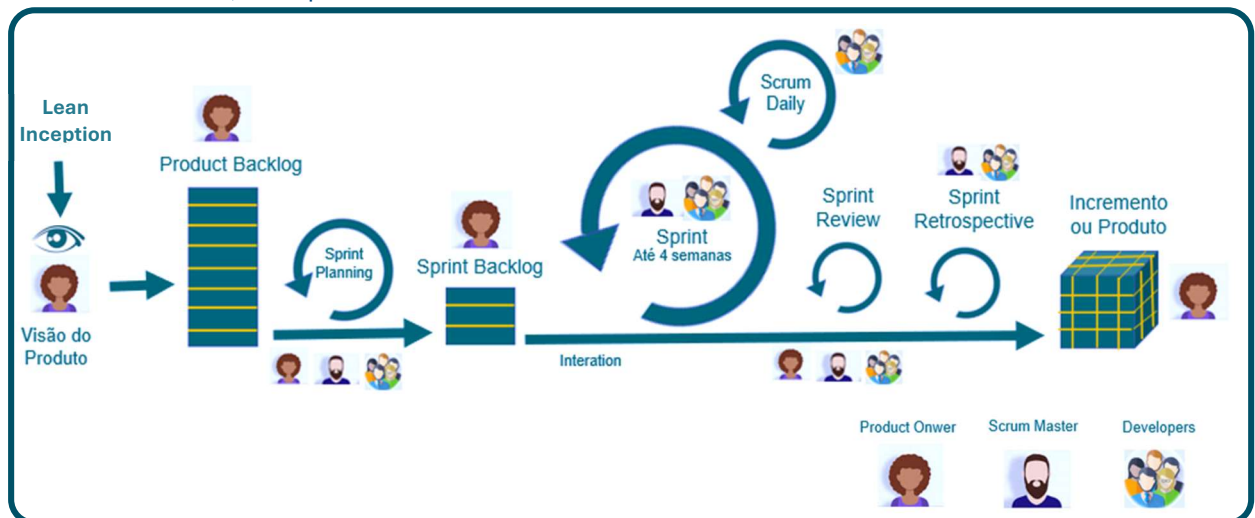
No intuito de viabilizar as entregas do projeto, a COSIS e o time contratado devem iniciar o processo de descoberta com o planejamento do **Workshop Lean Inception**, que deve apresentar como resultados **a Visão do Produto e o Canvas**. Ambos serão essenciais para a fase de concepção do projeto.

Na sequência, deve ser elaborado o **Roadmap do Produto**, a partir os resultados do *workshop*.

Nesta etapa, deve-se fundamentar o problema de negócio que será resolvido, desenhar a solução, que deve ser centrada na experiência do usuário e orientada nas necessidades das pessoas:

- público-alvo das atividades executadas pela Imprensa Nacional;
- servidores e colaboradores que vão utilizar as soluções, e
- atores envolvidos na transformação digital do serviço, que se tornará a solução tecnológica; e
- pessoas envolvidas no contexto que está sendo trabalhado para desenvolvimento da solução.

A seguir serão apresentadas todas as etapas do processo de desenvolvimento de soluções de TIC, baseado no Scrum, na Imprensa Nacional.



Fonte (com adaptações): <https://analistaexpert.com.br/scrum/>

Figura 2: Resumo da estrutura ágil Scrum

É importante que a equipe de desenvolvimento faça uma imersão no contexto de cada persona (usuário) e entenda as necessidades, desejos e expectativas que realmente existem nos processos de trabalhos envolvidos na transformação digital.

Logo, o planejamento compreende atividades anteriores a execução da primeira *Sprint* do projeto, e é composto pelas seguintes atividades, que podem ser adaptadas à realidade operacional da IN:

- a) construir a **Visão do Produto**;
- b) construir o **Roadmap do Produto**.

Ambos os documentos constituem-se base para o planejamento das Sprints.

6.2.1 Elaboração da Visão do Produto e Roadmap

Nesta etapa é importante definir o que o **produto/serviço deve entregar para o usuário final**, e **mapear o que não será entregue**, a fim de alinhar as expectativas de todos os envolvidos no projeto.

Ademais, apresentar como as entregas serão distribuídas na linha do tempo, e de que forma agregarão valor ao negócio. Sugere-se o emprego de técnicas de elicitação de requisitos, aliadas ao *Lean Inception*.

Visão do Produto

Para a concepção da **Visão do Produto**, a equipe de desenvolvimento, em conjunto com o dono do produto, deverá planejar o *workshop Lean Inception*, agendando uma sequência de encontros nos quais deverão ocorrer:

1. **Reunião de Kick-off:** momento para alinhar as expectativas entre os *stakeholders* envolvidos (cliente, gestor do projeto e membros da equipe).
2. **Apresentação da Agenda:** na reunião de *Kick-off*, também é o momento de apresentar a agenda que deve contemplar os próximos tópicos. A *Lean Inception* prevê uma semana de imersão para conclusão e apresentação do **Canvas** do projeto, mas dependendo do tamanho do escopo, prazo de entrega, este período pode ser diminuído e ajustado para adaptar as etapas ao projeto.
3. **Visão Geral e Preliminar do Produto:** etapa na qual deve-se definir, de forma colaborativa, o estado futuro do produto que se deseja alcançar. Trata-se de uma descrição dos objetivos e valor do produto, permitindo uma visão de alto nível do que será construído.
4. **Escopo geral:** deve-se nesta etapa, definir o que é o produto, o que não é, o que faz e o que não faz. Seguindo estas diretrizes espera-se que se delimite um escopo inicial do produto.
5. **Objetivos do Produto:** nesta etapa, o mais importante é levantar os objetivos do produto sob a ótica do negócio. Sendo assim, deve-se chegar ao consenso do que o produto deve satisfazer aos seus usuários finais e ao negócio.
6. **Personas:** levantar os usuários do produto e seu papel em relação ao produto. O foco nesta etapa é também capturar características, perfil, comportamento e necessidades, dos diferentes tipos de usuários típicos do produto.
7. **Jornada do Usuário:** aqui, nesta etapa, descrevem-se os percursos traçados pelos usuários do produto. Apresenta-se a interação da persona com o produto, fazendo uma ligação, entre a necessidade do usuário e o cumprimento ou contribuição com um objetivo de negócio, de forma a demonstrar uma interação da persona com o produto.
8. **Brainstorming de Funcionalidades:** neste momento, devem ser levantadas as funcionalidades que o produto deve conter. Cada funcionalidade deve satisfazer um objetivo do negócio total ou parcialmente, contemplar uma necessidade da persona, e/ou um passo da jornada. Esta atividade captura preliminarmente o *Backlog* do produto, pois este pode ser dinâmico e repriorizado de acordo com a necessidade do negócio.

9. **Revisão Técnica, de Negócio e de Experiência do usuário (UX):** na revisão é o momento de realinhamento de expectativas, pois cada funcionalidade deve ser classificada de acordo com o **valor de negócio, esforço e valor para a experiência do usuário**. Esta classificação é subsídio para priorizar, estimar e planejar as entregas. Pode ser revisada durante o andamento do projeto na fase de Planejamento das Sprints, que será tratada em tópico seguinte.
10. **Sequenciamento das funcionalidades:** este é um dos momentos de maior relevância para a definição do Produto Mínimo Viável (MVP) e os incrementos, pois após o sequenciamento é possível identificar a prioridade e a composição das funcionalidades. Também é subsídio para o Planejamento das Sprints, pois o *Lean Inception* prevê regras para a composição do sequenciamento, de considera os parâmetros de **valor de negócio, esforço e valor para a experiência do usuário**, definidos na etapa anterior, de **Revisão Técnica, de Negócio e UX**.
11. **Canvas MVP e Roadmap preliminar:** após passar por todas as etapas, deve-se construir o Canvas com as definições alcançadas em cada etapa. O **Canvas** deve apresentar a versão inicial mais simples do produto, que pode ser utilizada efetivamente pelos usuários finais de forma a agregar valor ao negócio. E também deve ser elaborado o **Roadmap** do projeto, com uma estimativa preliminar de marcos para as entregas.
12. **Showcase:** esta é a última etapa do *Lean Inception*. Após a consolidação realizada em cada etapa anterior, deve ser apresentada a consolidação realizada no Canvas, para todos os envolvidos no projeto.

Geralmente, o *Lean Inception* é executado em uma semana, na qual todas as etapas são distribuídas nos turnos matutino e vespertino em cada dia da semana. Mas esse período pode ser comprimido e adaptado de acordo com as características do projeto.

Roadmap do Produto

Considerando que a etapa de elaboração da Visão do Produto está concluída, a próxima ação é construir o **Roadmap do Produto**, que consiste numa representação gráfica que apresenta a visão, metas e estratégia do produto. Consiste em um guia que auxilia no entendimento sobre a evolução do produto e como as entregas se conectam na estratégia da organização.

A definição do *Roadmap* do Produto deve ser orientada às entregas por Produto Mínimo Viável (*Minimum Viable Product – MVPs*), que consiste em uma versão simplificada de um produto, que contém apenas as funcionalidades básicas necessárias para funcionar e ser lançado no ambiente de produção e disponibilizado ao uso do público-alvo.

Para os demais incrementos, deve-se planejar a entrega de um ou mais *Releases*, conforme o plano cronológico definido no *Roadmap*. Cada *Release* poderá ser construído em uma ou mais *Sprints*. O Time Ágil indica as atividades que deverão compor o planejamento da entrega da *Release*.

O mais importante como resultado da *Lean Inception* e obter ao final os insumos mínimos para a elaboração da Visão do Produto e *Roadmap* do Produto, pois ambos os artefatos são de grande relevância para o acompanhamento e alinhamento de expectativas do projeto.

Com o *Roadmap* do Produto elaborado, inicia-se o desenvolvimento do produto adotando a metodologia *Scrum*, que deve ser seguida até a entrega final do último incremento do produto, encerrando o projeto.

6.2.2 Concepção da lista de tarefas e funcionalidades do Produto (*Product Backlog*)

Obtido a partir da Visão do Produto, o *Product Backlog* é uma lista de funcionalidades do produto, é um controle dinâmico que prioriza as funcionalidades e tarefas necessárias para o desenvolvimento e evolução de um produto.

Uma vez realizado o *Lean Inception*, tem-se como insumo para a elaboração do *Product Backlog*, as atividades realizadas na sessão **10. Sequenciamento das funcionalidades**.

Nessa lista também estão incluídas, além das funcionalidades, melhorias, correção de *bugs* e quaisquer outras tarefas que seja necessária para o desenvolvimento do produto.

Para a produção desta lista, se não for possível realizar o workshop completo, é fundamental a execução mínima da etapa de **8. Brainstorming de Funcionalidades** da *Lean Inception*.

6.2.3 Estimativa do tamanho e esforço do *backlog* do produto

Todo o *backlog* do produto deve ser medido utilizando-se a técnica de Análise de Pontos de Função. Essa mensuração é utilizada para estimar o tamanho do *backlog* do produto e esforço necessário para o desenvolvimento. É uma atividade importante para apoiar no planejamento dos releases e quantidade de iterações e também para dirimir questões contratuais.

A estimativa de tamanho e esforço pode ser feita por meio do método de ponto de função simplificado (SFP) descrito no Roteiro de Métricas do SISP.

6.2.4 Planejamento do desenvolvimento e entrega do Produto por *Sprints* (*Sprint Planning*)

De posse do *Product Backlog*, o time pode planejar a primeira e as demais *Sprints* até a finalização da lista.

A *Sprint* consiste no período de tempo real em que o time trabalhará em conjunto para concluir uma entrega, que varia entre 1 a 4 semanas, de acordo com a característica de cada projeto. Deve-se sempre considerar, para a definição de tempo real, a complexidade e as incertezas em relação ao trabalho a ser realizado, pois em geral, quanto mais complexo for o trabalho e mais dúvidas existirem, mais curto deve ser a *Sprint*.

Na fase de planejamento da *Sprint*, com base no *backlog* do produto priorizado, a equipe elege, a partir do MVP e incrementos resultantes do **10. Sequenciamento das funcionalidades** – realizado na *Lean Inception* – as funcionalidades que devem compor a lista de funcionalidades da *Sprint* (*Sprint Backlog*), e estima o trabalho a ser concluído, no tempo estipulado para a execução daquela *Sprint*.

Nesta etapa, são também descritas as histórias de usuários relacionadas às funcionalidades escolhidas para a *Sprint*. O dono do produto, em conjunto com os demais membros, deve definir os critérios de aceitação das funcionalidades para aprovação da entrega.

Os planejamentos de *Sprints* devem ocorrer enquanto houver itens na lista de funcionalidades e tarefas do produto (*Product Backlog*) que ainda não foram entregues.

6.2.4.1 Lista de funcionalidades da *Sprint* (*Sprint Backlog*)

Esta lista consiste nos itens do *Backlog* do Produto selecionados para serem trabalhados durante a *Sprint*. Representa o trabalho que a equipe de desenvolvimento se compromete a entregar em determinada *Sprint*. A *Sprint backlog* é flexível e pode evoluir durante um *Sprint*, de acordo com o aprendizado sobre o produto, a necessidade de desenvolvimento e do negócio.

Os itens de *Product Backlog* selecionados para a *Sprint* e o plano de entrega do incremento, são agrupados no *Backlog* da *Sprint*. Na definição do *backlog* da *Sprint*, deve-se monitorar a relação quantitativa entre itens planejados e itens não planejados, com vistas a assegurar que o maior esforço esteja sendo empreendido na entrega de valor ao negócio.

Logo, a *Sprint Backlog* deve estar detalhado de forma que o seu progresso possa ser inspecionado nas reuniões diárias (*Scrum Daily*).

6.3 Acompanhamento do desenvolvimento do Produto

Durante a execução das tarefas previstas nas *Sprints*, existem encontros pré-definidos para acompanhamento do progresso do desenvolvimento do produto. Tais encontros são importantes para identificar possíveis impedimentos que possam ser solucionados sem comprometimento das metas e prazos do projeto.

6.3.1 Realização de Reunião Diária (*Scrum Daily*)

Consiste em uma reunião diária, de curta duração, até 15 minutos, realizada pela equipe de desenvolvimento. O principal objetivo desta reunião é sincronizar as atividades da equipe, revisar o progresso em direção ao objetivo da *Sprint* e identificar impedimentos que possam atrapalhar o trabalho.

Durante essa agenda, o time é orientado a responder a três perguntas:

- a) O que eu fiz desde a última reunião?
- b) O que eu vou fazer até a próxima reunião?
- c) Há algum impedimento para o trabalho ser realizado?

Dependo das respostas explicitadas pelo time, o *Scrum Master* deve atuar a fim de remover impedimentos, e promover a interação entre os membros do time, partes interessadas (*stakeholders*) para sanar eventuais problemas de entendimento.

6.3.2 Realização da Reunião para Revisão da Sprint (*Sprint Review*)

É uma reunião que ocorre ao final de cada *Sprint*. O principal objetivo da Revisão da *Sprint* é a demonstração do produto incremental, fruto do trabalho daquela *Sprint*, e sua aderência à definição de pronto para cada um dos itens da *Sprint Backlog*.

O Dono do Produto analisará a resolução de cada história apresentada e decidirá se está “pronta” ou não, de acordo com o critério previamente estabelecido. Nesse momento, o *backlog* do produto pode ser atualizado para refletir as novas demandas e oportunidades.

Caso alguma história seja reprovada, ela será reinserida no *Backlog*, ficando disponível para uma próxima *Sprint*. Essa revisão pode levar até 4 horas para *Sprints* de 4 semanas.

Nessa agenda, além de validar as entregas da equipe e verificar se os critérios definidos no planejamento foram atendidos, também é o momento de coletar as opiniões sobre o que a equipe construiu. É uma conversa entre a equipe e as partes interessadas sobre como melhorar o produto.

6.3.3 Realização da Reunião para a Retrospectiva da Sprint (*Sprint Retrospective*)

O último evento antes da conclusão da *Sprint* é chamado de retrospectiva da *Sprint*. É uma reunião que ocorre sempre ao final de cada *Sprint*.

O principal objetivo da Retrospectiva da *Sprint* é permitir que o Time Ágil inspecione e adapte seus processos e práticas. Assim, promovendo melhorias contínuas na eficácia e eficiência da equipe, proporcionando um processo de aprendizagem da equipe por meio do histórico de cada *Sprint*. Além disso, é o momento de planejar maneiras de aumentar a qualidade e a eficácia do trabalho.

Nesta agenda, o time também discute como foi a última *Sprint* em relação a indivíduos, interações, processos, ferramentas e a Definição de Pronto. Responde-se basicamente a dois questionamentos:

- d) O que deu certo?
- e) O que pode ser melhorado?

O time, então, identifica as mudanças mais úteis para melhorar sua eficácia e procura aplicar essas mudanças na *Sprint* seguinte. A retrospectiva pode levar até 4 horas para *Sprints* de 4 semanas.

6.3.4 Produção de Relatório da Sprint

Para manter os envolvidos no projeto atualizados sobre o andamento das *Sprints*, ao final do *time-box*, deve ser gerado um relatório que de forma concisa resume o progresso da equipe durante a *Sprint*, destacando as principais realizações, desafios e métricas relacionadas aos objetivos da *Sprint*.

O documento serve como uma ferramenta de comunicação para as partes interessadas, membros da equipe, donos de produtos, para obter *insights* sobre os resultados da *Sprint* e tomar decisões consciente e informadas.

Em sua estrutura deve contemplar um resumo da *Sprint*, com a duração e os objetivos, listar as histórias de usuários ou tarefas concluídas durante a *Sprint*. Podem ser informados quaisquer obstáculos ou problemas enfrentados pela equipe e como foram resolvidos.

É importante apresentar métricas de desempenho, como velocidade de produção, gráficos de *burndown* e outros indicadores. Também é interessante informar os objetivos da próxima *Sprint*, a fim de alinhar os trabalhos. E por fim, apresentar o resultado geral da *Sprint* e validar o alinhamento com os objetivos do projeto.

6.3.5 Gestão de Mudanças

Os projetos estão sujeitos a mudanças, independente da metodologia empregada, por isso é importante que os membros do Time Ágil entendam que os processos de desenvolvimento *Scrum* são preparados para aceitar estas mudanças.

Por utilizar pequenos ciclos de desenvolvimento, que contemplam o *feedback* dos interessados nas entregas depois de cada *Sprint*, os incrementos do produto, assim que prontos, podem recepcionar mudanças de requisitos durante o ciclo de desenvolvimento, quando interagem regularmente com os membros do Time Ágil.

Os pedidos por mudanças são apresentados na forma de Solicitações de Mudanças, que permanecem em um estado não aprovado até que sejam formalmente aprovadas pela COSIS.

As pequenas mudanças, que não têm impacto significativo sobre o projeto, devem ser aprovadas diretamente pelo Dono do Produto, após acordo com os demais membros do Time Ágil. A tolerância para essas pequenas mudanças pode ser definida em um nível organizacional ou pelo patrocinador do projeto em particular.

Sendo assim, o Dono do Produto desempenha um papel muito importante no gerenciamento de mudanças no projeto. As mudanças que estão além do nível de tolerância do Dono do Produto podem precisar da aprovação dos *stakeholders* que trabalham com o Dono do Produto. Às vezes, a autorização da direção pode ser necessária, se a modificação solicitada tiver um impacto substancial sobre o projeto ou organização.

As Solicitações de Mudança para o projeto são discutidas e aprovadas durante os processos de refinamento do *Backlog* do Produto, após a Revisão da *Sprint*. As Solicitações de Mudança Aprovadas são então priorizadas juntamente com outros requisitos do produto e suas respectivas Histórias de Usuário, e depois são incorporadas no *Backlog* do Produto.

6.4 Entrega dos Produtos

A entrega dos produtos de *software* ocorre após a finalização da *Sprint*, quando o incremento de *software* é revisado e homologado na Reunião de Revisão da *Sprint*. Trata-se de um evento separado da *Sprint*. Logo, refere-se à versão do produto que é disponibilizada aos usuários ou *stakeholders*.

As *Releases* são planejadas para entregar valor de negócio incremental, de maneira contínua e regular, alinhando-se com a filosofia ágil de entregar *software* funcional em curtos períodos de tempo.

O incremento é um passo em direção a uma meta ou visão. É o produto final utilizável de uma *Sprint*. As equipes podem adotar diferentes métodos para determinar e demonstrar suas metas de *Sprint*. Apesar da flexibilidade, a meta fundamental da *Sprint*, o que a equipe deseja alcançar na *Sprint* atual, não pode ser afetada.

Por exemplo, algumas equipes escolhem disponibilizar algo aos clientes no final da *Sprint*, atingindo a meta de *Sprint* assim que a mudança de *software* seja disponibilizada. Outras equipes podem trabalhar para concluir um conjunto de recursos que serão disponibilizados juntos. Nesse caso, a meta da *Sprint* seria concluída quando um recurso fosse testado com sucesso.

Sendo assim, na Revisão da *Sprint*, a opinião de todos os *stakeholders* é coletada, porém somente o Dono do Produto tem o poder de aceitar ou rejeitar uma História de Usuário em particular, fundamentando-se no que foi acordado nos Critérios de Aceitação. Logo, as Histórias de Usuário, parcialmente concluídas são rejeitadas, e classificadas como não Prontas, e retornam para o *Backlog* do Produto.

É oportuno destacar que a Dívida Técnica (dívida de *design* ou dívida de código) é o trabalho que o time entende que possui prioridade inferior, omissa ou incompleta, uma vez que a intenção é desenvolver os principais entregáveis do produto. É imperativo ressaltar que essa dívida vai se acumulando ao longo do projeto, e deve ser paga no futuro. Logo, a dívida técnica tem que constar como item no *Backlog* do produto.

Portanto, com o intuito de manter uma quantidade mínima de dívida técnica, é importante definir o produto necessário de uma *Sprint* e do projeto, juntamente com os Critérios de Aceitação, com os métodos de desenvolvimento a serem seguidos, e com as principais responsabilidades dos membros do Time Ágil em relação à qualidade.

O time deve focar no planejamento contínuo de modo que possa fornecer valor de negócio em intervalos frequentes, para isso o time deve continuamente desenvolver, testar e integrar as funcionalidades de cada Item do Backlog Priorizado do Produto, em cada *Sprint*, com o uso de técnicas como a integração contínua e testes de produtos automatizados.

6.4.1 Critérios de Aceitação do Produto

Cada História de Usuário terá respectivamente os Critérios de Aceitação do Usuário associados, que são os objetivos pelos quais a funcionalidade de uma História de Usuário é avaliada.

Os Critérios de Aceitação são elaborados pelo Dono do Produto, de acordo com seu conhecimento sobre os requisitos dos usuários do produto. Os critérios são repassados durante a elaboração das Histórias de Usuário do *Backlog* do Produto e registrados após acordo entre os envolvidos. Devem explicitar claramente as condições que as Histórias de Usuário devem satisfazer, pois é determinante para o sucesso da entrega.

Ao final de cada *Sprint*, o Dono do Produto utiliza esses critérios durante a homologação do produto, para verificar as entregas concluídas, podendo **aceitar** ou **rejeitar** os itens individualmente e suas respectivas Histórias de Usuário, conforme as opções a seguir:

- a) **rejeitar**: caso sejam percebidos defeitos de natureza impeditiva em alguma história implementada ou não tenha coberto o escopo planejado de tal forma que a entrega não seja passível de aceitação;

- b) **aceitar parcialmente:** caso a demanda possua alguns defeitos significativos de natureza não impeditiva, ou não tenha coberto o escopo planejado de tal forma que ainda seja passível de aceitação; ou
- c) **aceitar integralmente:** caso a demanda esteja em nível de qualidade tal que não sejam percebidos defeitos significativos, bem como envolva cumprimento do escopo planejado.

Se as entregas forem aceitas pelo Dono do Produto, a História de Usuário será considerada pronta. Por isso é fundamental que o Time Ágil tenha uma definição clara de Pronto, para ajudar a esclarecer os requisitos e aderir a normas de qualidade. O Time deve registrar todos os defeitos percebidos nos casos de rejeição ou aceitação parcial da *Sprint*. Tais registros devem fazer parte de um item de *backlog* da próxima *Sprint*.

6.4.2 Definição de Pronto

A definição de pronto é uma descrição formal do estado do incremento, quando este cumpre as medidas de qualidade exigidas para o produto. É importante ressaltar que os **Critérios de Aceitação** são exclusivos para Histórias de Usuário individualmente, enquanto os **Critérios de Pronto** são um conjunto de regras que se aplicam a todas as Histórias de Usuário da *Sprint*.

Sendo assim, quando um item do *Backlog* do Produto satisfaz os Critérios de Pronto e os Critérios de Aceitação do Produto, constitui-se em um incremento. **Se um item de Product Backlog não cumpre os critérios de pronto, não pode ser lançado, ou mesmo apresentado na Revisão da Sprint.** Em vez disso, volta ao *Backlog* do Produto para consideração futura.

Para **admissibilidade** e cumprimento dos **critérios de pronto**, é necessário:

- a) código-fonte submetido ao controle de versões do repositório da Imprensa Nacional;
- b) existência de testes unitários e do Relatório de Testes;
- c) existência de *scripts* de banco de dados com dicionário de dados embutido nos metadados (ausência apenas quando não houver mudança no modelo de dados);
- d) existência de arquivo para geração de *Build*;
- e) disponibilização de processos prontos para execução na ferramenta de CI/CD (pipeline para Integração Contínua - *Continuous Integration* - CI/*Continuous Delivery* - CD e *Continuous Deployment*) adotada, juntamente com a entrega e configuração de containers configurados pela ferramenta de orquestração adotada;
- f) existência de manual de implantação, conforme modelo disponibilizado pela Imprensa Nacional;
- g) existência documentação concluída, de acordo com os padrões de qualidade definidos pela Imprensa Nacional e validadas pelo Dono do Produto.
- h) resultado da execução de teste de segurança em aplicações, tipo SAST (*Static Application Security Testing*), indicando ausência de vulnerabilidades de nível *HIGH* (Alta) ou *CRITICAL* (Crítica), ou equivalente.

Antes da submissão do produto à apreciação e **aceitação** do dono do produto, e após realizar a inspeção do produto quanto à sua admissibilidade, os desenvolvedores deverão:

- a) executar testes funcionais automatizados que tenham sido solicitados e, conseqüentemente, verificar se estão corretamente implementados ou mesmo se existem, além de observar os resultados da execução;
- b) executar testes unitários ou verificar relatórios de execução destes que possam envolver partes críticas do produto;
- c) realizar alguns testes funcionais, pelo menos nos principais fluxos do produto entregue.

Por fim, **após a realização dos testes**, o Dono do Produto, em conjunto com a equipe da COSIS, deve proceder com a homologação do produto, conforme os critérios de aceitação elaborados de acordo com a seção 5.4.1. Critérios de Aceitação do Produto.

6.4.3 Liberação das Entregas

A entrega dos produtos de *software* ocorre após a finalização da *Sprint*, quando o incremento de *software* é revisado e homologado. É um evento separado da *Sprint*, e só ocorre após o incremento ter atendido os **critérios de aceitação do produto** (ver seção 6.4.1. Critérios de Aceitação do Produto) e a **definição de pronto** (ver seção 6.4.2. Definição de Pronto).

A liberação é caracterizada pela disponibilização do incremento no ambiente de TIC, de produção, da Imprensa Nacional.

7 ARTEFATOS PRODUZIDOS

Os artefatos são os documentos que serão gerados durante as atividades de desenvolvimento de software. Esses documentos devem ser atualizados na medida que forem ocorrendo alterações nas situações que justificaram a sua confecção.

Esses documentos têm como objetivo registrar as características das ações que envolvem o desenvolvimento de software, como a finalidade do desenvolvimento do software, as entregas dos produtos que são gerados no processo de desenvolvimento do software, as métricas utilizadas para fundamentar o pagamento dos serviços, os prazos para a conclusão dos produtos de software, as especificações técnicas, além de outros que a COSIS julgar como necessárias.

Em cada etapa do processo de desenvolvimento, o Dono do Produto, em conjunto com Time Ágil e a equipe da COSIS, produzem os artefatos necessários para materializar o produto.

São considerados artefatos a documentação produzida ao longo do desenvolvimento e todo o código-fonte produzido para a entrega do *software*.

Os artefatos a serem produzidos por orientação desta metodologia estão descritos na tabela a seguir:

N.º	Nome do Artefato	Finalidade	Responsáveis pela elaboração	Etapa do Processo
1	Termo de Abertura do Projeto	Formalizar a abertura do projeto junto à área demandante.	Equipe da COSIS e Dono do Produto.	Antes de iniciar o Planejamento do Produto.
2	Visão do Produto	Apresentar em linhas gerais as características do produto.	Dono do Produto, Equipe da COSIS, Time Ágil.	6.2. Planejamento do Desenvolvimento do Produto
3	<i>Canvas</i>	Panorama do projeto, contendo principais marcos, cronograma e custo preliminares.	Dono do Produto, Equipe da COSIS, Time Ágil.	6.2. Planejamento do Desenvolvimento do Produto
4	<i>Product Backlog</i>	Listar as funcionalidades do produto.	Dono do Produto	6.2. Planejamento do Desenvolvimento do Produto.
5	<i>Roadmap</i>	Apresentar os marcos de entrega do projeto na linha do tempo.	Dono do Produto, Equipe da COSIS, Time Ágil.	6.2. Planejamento do Desenvolvimento do Produto.
6	História de usuário (HU)	Descrever o comportamento da funcionalidade, contendo as regras de negócio, características da interface e interoperabilidades internas e externas.	Dono do Produto, Equipe da COSIS, Time Ágil.	6.2. Planejamento do Desenvolvimento do Produto. 6.2.4 Planejamento do desenvolvimento e entrega do Produto por <i>Sprints (Sprint Planning)</i>
7	Regras de Negócio (RN)	Descrever as diretrizes específicas que compõem processos e operações do produto, que apliquem políticas, objetivos e requisitos da IN.	Dono do Produto, Equipe da COSIS, Time Ágil.	6.2. Planejamento do Desenvolvimento do Produto. 6.2.4 Planejamento do desenvolvimento e entrega do Produto por <i>Sprints (Sprint Planning)</i>
8	Especificações de Serviço (ES)	Descrever detalhadamente as integrações para	Dono do Produto, Equipe da COSIS, Time Ágil.	6.2. Planejamento do Desenvolvimento do Produto. 6.2.4 Planejamento do desenvolvimento e

		além das fronteiras do sistema, com linguagem mais técnica, retratando o funcionamento das integrações e como foram implementadas.		entrega do Produto por <i>Sprints (Sprint Planning)</i>
9	Documento de Arquitetura	Descrever a estrutura técnica e as principais decisões de design que guiarão o desenvolvimento do produto.	Equipe da COSIS, Time Ágil.	6.2. Planejamento do Desenvolvimento do Produto. 6.2.4 Planejamento do desenvolvimento e entrega do Produto por <i>Sprints (Sprint Planning)</i> (para revisão do documento)
10	Contagem de Pontos de Função (PF)	Deve conter o resultado da análise e contagem dos Pontos de Função (PF) contidos na <i>Sprint</i> .	Equipe da COSIS, Time Ágil.	6.2.4 Planejamento do desenvolvimento e entrega do Produto por <i>Sprints (Sprint Planning)</i>
11	<i>Sprint Backlog</i>	Listar as funcionalidades contidas no Product Backlog que vão compor a <i>Sprint</i> .	Dono do Produto, Equipe da COSIS, Time Ágil.	6.2.4 Planejamento do desenvolvimento e entrega do Produto por <i>Sprints (Sprint Planning)</i>
12	MVP e Incrementos do Produto (<i>Releases</i>)	Entregar no repositório da IN, código-fonte, documentação, e orientações de implantação.	Dono do Produto, Equipe da COSIS, Time Ágil.	6.2.4 Planejamento do desenvolvimento e entrega do Produto por <i>Sprints (Sprint Planning)</i>
13	Memória de Reunião da Revisão da <i>Sprint (Sprint Review)</i>	Registrar os principais pontos levantados pelo dono do produto sobre o desenvolvimento do produto.	Dono do Produto, Equipe da COSIS, Time Ágil.	A cada reunião de Revisão da <i>Sprint (Sprint Review)</i> realizada.
14	Memória de Reunião da Retrospectiva da <i>Sprint (Sprint Retrospective)</i>	Registrar as principais lições aprendidas na execução da <i>Sprint</i> .	Dono do Produto, Equipe da COSIS, Time Ágil.	A cada reunião de Retrospectiva da <i>Sprint (Sprint Retrospective)</i> realizada.
15	Relatório da <i>Sprint</i>	Servir como ferramenta de comunicação para as partes interessadas, membros da equipe, dono de produto e CGTI para obter <i>insights</i> relacionados aos resultados da <i>Sprint</i> e tomar decisões informadas.	Dono do Produto, Equipe da COSIS, <i>Scrum Master</i> .	Apresentar a cada <i>Sprint</i> concluída.
16	Manual de Implantação	Descrever como o MVP e incrementos (<i>Sprints</i> e <i>Releases</i>) serão implantados e publicados em ambiente produtivo.	Time Ágil.	6.4 Entrega dos Produtos
17	Termo de Encerramento do Projeto	Formalizar o encerramento ou	Equipe da COSIS e Dono do Produto.	Após a entrega dos artefatos finais do projeto.

		cancelamento do projeto junto à área demandante.		
--	--	--	--	--

Tabela 2: Artefatos produzidos

A produção dos artefatos será orientada em guia específico, pois será disponibilizada ferramenta própria para preenchimento e posterior acompanhamento e atualização da documentação.

8 FERRAMENTAS

Com a finalidade de facilitar a execução e acompanhamento das atividades previstas nesta metodologia, serão disponibilizadas ferramentas próprias para:

- a) Gestão da informação e pagamentos dos serviços realizados por terceiros;
- b) Gestão de demandas;
- c) Repositório para controle de revisão de código;
- d) Apresentação do andamento do projeto;
- e) Documentação complementar;
- f) Qualidade de Código-Fonte;
- g) DevOps - Integração e Entrega Contínuas (Pipeline CI/CD);
- h) Teste de Segurança de Aplicações (SAST, DAST, IAST);

Para a utilização de cada ferramenta, serão disponibilizadas informações adicionais por meio de guias e informações na Wiki (<http://wiki.in.local>), pois não serão objeto desta metodologia.

9 PROCESSO DE SUSTENTAÇÃO E MANUTENÇÃO DAS SOLUÇÕES DE TIC

Os produtos e serviços de *software* tem um ciclo de vida, e o tempo para encerramento deste ciclo depende de várias características que variam dependendo do negócio e das tecnologias empregadas.

Além de manter o produto/serviço aderente aos requisitos de negócio, é necessário garantir a disponibilidade e a adequada operação no ambiente produtivo. Este processo é chamado de **Sustentação**. Neste processo, o time responsável pelo ambiente produtivo monitora a disponibilidade, uso e o desempenho do produto. A Portaria SGD n.º 750 de 20 de maio de 2023, no ANEXO I - MODELO DE CONTRATAÇÃO DE SERVIÇOS DE DESENVOLVIMENTO E SUSTENTAÇÃO DE SOFTWARE define que:

“2.1.2. O serviço de sustentação de software corresponde ao conjunto de atividades necessárias para manter a disponibilidade, estabilidade e desempenho do software em produção, dentro dos níveis de serviço estabelecidos pelo órgão ou entidade. Admite-se, no escopo desse serviço, a previsão de manutenções de pequeno porte, cujos limites, baseados em métricas de software, devem estar previamente definidos.”

Logo, caso ocorram problemas no ambiente de TIC que impactem no funcionamento do produto, o time de desenvolvimento e de infraestrutura atuam para sanar e recuperar a saúde do ambiente e a disponibilidade do produto na performance esperada.

Além da Sustentação, existem outros tipos de atendimentos, pois logo após a conclusão de um projeto, ainda no seu ciclo de vida, o produto/serviço continua em um processo de mudanças e afeiçoamento para garantir a maior longevidade e adaptabilidade da solução de TIC.

Este processo, denominado **Manutenção**, mantém o produto/serviço em funcionamento no ambiente produtivo, garante que os problemas encontrados, durante seu funcionamento, sejam sanados, e que novos comportamentos sejam incorporados, a fim de manter o seu propósito negocial.

Baseado na norma ISO/IEC 14764 (2006), que define um processo de manutenção e descreve as tarefas e atividades necessárias para modificar um produto de software mantendo sua integridade, o processo de manutenção orienta as atividades de identificação da mudança, da análise e da aceitação das mudanças, da eventual migração ou da retirada do software de produção.

Os tipos básicos de mudanças que podem ser encontrados durante o período de manutenção são:

- d) **Corretivas:** que é realizada para sanar erros de desenvolvimento;
- e) **Adaptativas:** que são feitas em função de mudanças no ambiente de TIC;
- f) **Evolutivas:** que correspondem às melhorias funcionais, que resultam de alterações nos comportamentos funcionais, a fim de adaptar-se às mudanças do negócio;
- g) **Consultoria técnica com intervenção:** são demandas que geram custos a partir de um catálogo de serviços pré-definido em contratos com prestadores de serviços.
- h) **Preventivas:** são realizadas a fim de preparar o produto para que o impacto causado por alterações ambientais e funcionais não afetem a operação do *software*.
- i) **Perfectiva:** visa melhorar o desempenho ou funcionalidade do *software*, mesmo quando ele está funcionando de forma adequada, com a finalidade de aumentar o valor agregado ao negócio.

9.1 Processo de Sustentação

Na Imprensa Nacional, o processo de Sustentação se inicia a partir do momento que, em parte ou na totalidade, o produto/serviço se encontra em operação no ambiente de produção.

Sendo assim, sempre que detectados problemas relacionados a potenciais falhas técnicas e de segurança da informação, que afetem a disponibilidade do produto/serviço, o Dono do Produto deve registrar o incidente na ferramenta de ITSM (atualmente a ferramenta utilizada é o Citsmart).

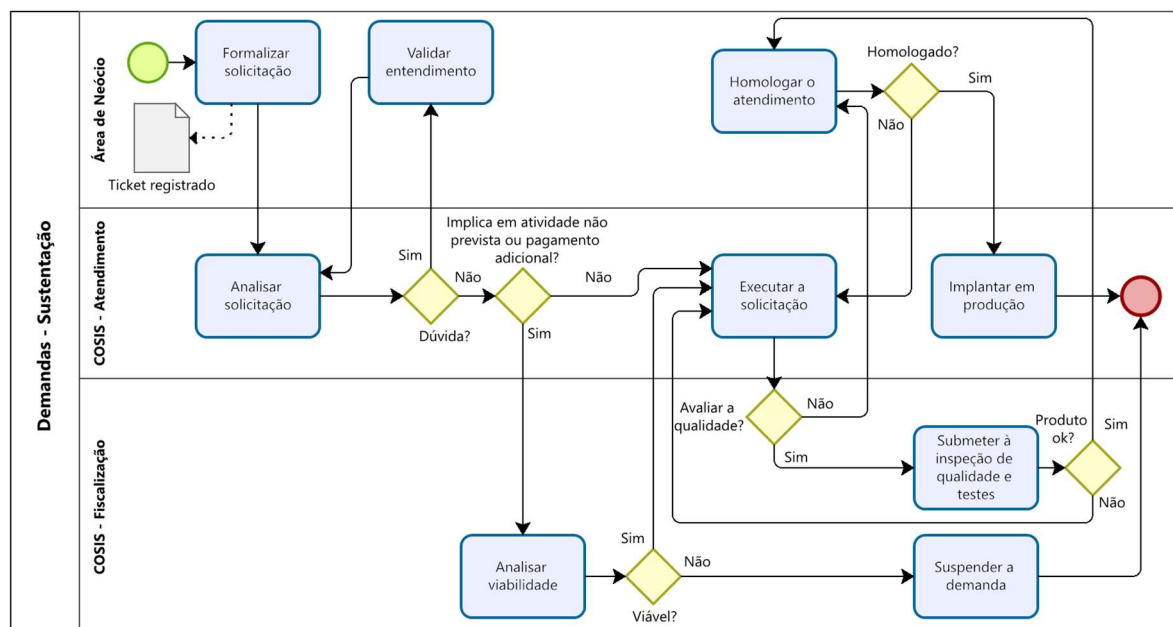


Figura 3: Processo para a gestão da demanda de Sustentação

9.1.1 Fluxo da demanda de Sustentação

A demanda é registrada pelo gestor de negócio (solicitante) no Citsmart. O Nível 1 - N1 de atendimento realiza a triagem e encaminha para a equipe técnica da COSIS. O analista técnico,

responsável pelo atendimento, recebe a demanda, e encaminha para o desenvolvedor, que após entender o escopo e elaborar a solução, implanta no ambiente de homologação.

Após testes preliminares da equipe de desenvolvimento (composta por colaboradores de empresa contratada), a demanda poderá também ser submetida à inspeção de qualidade e testes adicionais, antes de ser liberada para homologação do gestor demandante. Uma vez que a demanda é aceita, o gestor registra na ferramenta e o desenvolvedor solicita a disponibilização da *build* em produção e conclui o atendimento.

Demandas que geram custo adicional para a Imprensa Nacional devem ser submetidas para aprovação da COSIS (Fiscalização), a fim de avaliar a viabilidade, a disponibilidade orçamentária, entre outros aspectos, para liberar o desenvolvimento.

9.2 Processo de Manutenção

Durante a execução do projeto, as demandas de manutenção são incorporadas como itens no *Product Backlog*, para posteriormente serem selecionados para compor *Sprints* futuras. Caso a demanda de manutenção seja imprescindível para a continuidade do uso do produto, pode ser atendida em demanda própria e aplicada em ambiente produtivo, seguindo o rito próprio.

Após a conclusão do projeto, não se trabalha mais com o conceito de projeto, que tem como principais características esforço temporário, resultado único e recursos delimitados. Logo, se inicia o processo de Manutenção do Produto.

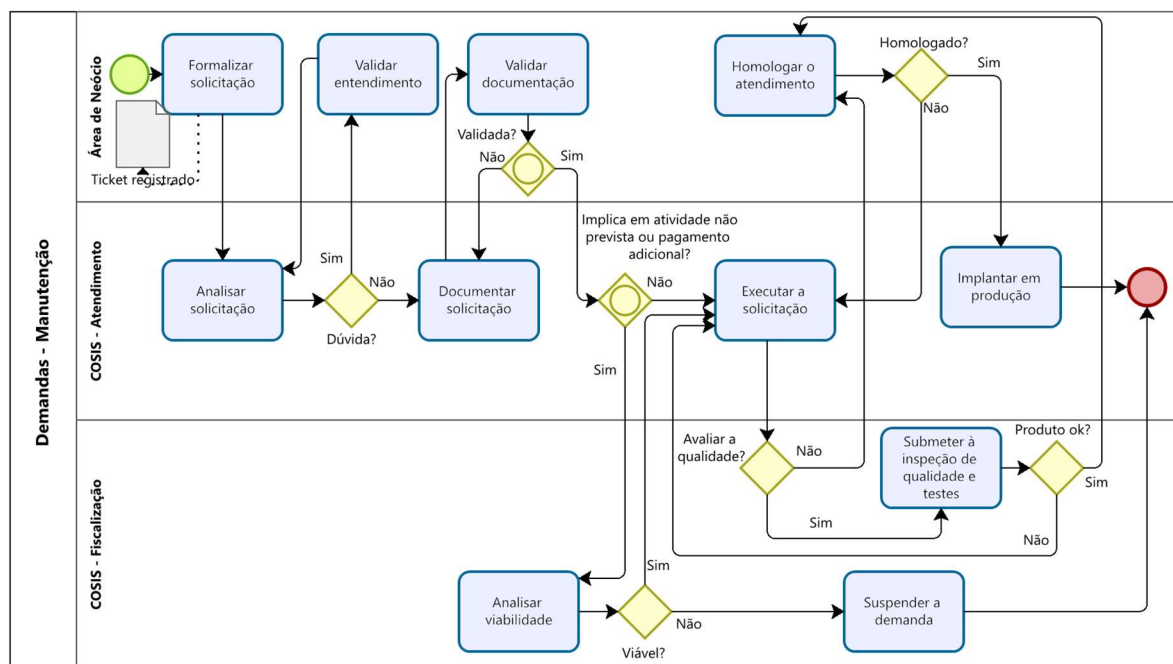


Figura 4: Processo para a gestão da demanda de Manutenção

9.2.1 Fluxo da demanda de Manutenção

As demandas de Manutenção, semelhante ao que ocorre com as demandas de Sustentação, são formalizadas ao serem registradas, pelo gestor de negócio (ou Dono do Produto), no Citsmart. O N1 realiza a triagem e encaminha para a equipe técnica da COSIS.

O analista técnico (servidor ou colaborador de empresa contratada), responsável pelo atendimento, recebe a demanda, realiza as estimativas, encaminha para a COSIS avaliar e conceder a autorização para atendimento. Após este procedimento, nos casos de autorização concedida, a

demanda é encaminhada para o desenvolvedor, que após entender o escopo e elaborar a solução, implanta no ambiente de homologação.

Após testes preliminares da equipe de desenvolvimento (composta por colaboradores de empresa contratada), a demanda poderá ser submetida à inspeção de qualidade e testes adicionais, antes de ser liberada para homologação do gestor demandante. Uma vez que a demanda é aceita, o gestor registra na ferramenta e o desenvolvedor solicita a disponibilização da *build* em produção e conclui o atendimento.

Demandas que geram custo adicional para a Imprensa Nacional devem ser submetidas para aprovação da COSIS, a fim de avaliar a viabilidade, a disponibilidade orçamentária, entre outros aspectos, para liberar o desenvolvimento.

9.3 Descontinuidade e Obsolescência do Produto

O ciclo de vida é a estrutura contendo processos, atividades e tarefas envolvidas no desenvolvimento, operação e manutenção de um produto de *software*, no que tange aspectos da vida do produto, que compete desde a definição de seus requisitos até o término de seu uso em definitivo.

A complexidade do negócio, o tempo disponível, o custo, a equipe e o ambiente operacional são fatores que influenciarão diretamente na duração deste ciclo.

Com a finalidade de apoiar o Dono do Produto, na análise da possibilidade de descontinuidade, são apontadas as principais características, que podem indicar o fim do ciclo de vida do produto:

- a) Os custos de desenvolvimento e adequação a novos cenários de negócio inviabilizam a continuidade do Produto, no que diz respeito a manutenibilidade e seu custo-benefício.
- b) O *software* pode ser substituído por outros serviços e/ou outras aplicações mais completas;
- c) A quantidade de usuários que adotaram o software ficou abaixo da expectativa da área de negócio, mesmo tendo sido utilizado em maior escala em algum momento;

Caso seja identificado que o produto é um candidato a descontinuidade, a área de negócio responsável pelo produto, deve encaminhar um ofício para a Coordenação-Geral de Tecnologia da Informação – CGTI, e informar sobre a necessidade de descontinuidade do produto. Após o encaminhamento, a Coordenação de Sistemas, Dados e Inovação - COSIS, realizará o levantamento de quais procedimentos devem ser adotados, o que inclui necessidade de transição para outra solução, backup dos dados gerados a partir do produto em obsolescência, entre outros possíveis procedimentos

O produto de *software* que está no fim da vida útil pode representar grandes ameaças de segurança. Isso porque, em geral, os fabricantes não fornecem mais *patches*, correções de *bugs* ou atualizações de segurança para antigas versões de linguagens de programação, frameworks e demais tecnologias que podem ter sido utilizadas no desenvolvimento da solução, e que não tem mais suporte.

10 INSPEÇÃO DE QUALIDADE E TESTES AVANÇADOS

A qualidade é entendida como a capacidade dos produtos ou entregas concluídas atenderem aos Critérios de Aceitação e em alcançar o valor de negócio esperado pelo demandante. Por isso, as atividades relacionadas à qualidade são realizadas pelo mesmo time, como parte do mesmo *Sprint*, quando projeto, ou da mesma demanda, quando estiver relacionada à Sustentação ou Manutenção.

Isso garante que a qualidade seja inerente a qualquer entregável pronto, seja criado como parte de um *Sprint* ou de uma demanda de Sustentação ou Manutenção. Logo, o compromisso do time de

desenvolvimento com a melhoria contínua otimiza a probabilidade de atingir os níveis de qualidade esperados nas entregas.

Sendo assim, mesmo com a atuação do time de desenvolvimento, em prol da qualidade, para assegurar que as entregas atendam aos requisitos de qualidade, os produtos gerados a partir de projetos de desenvolvimento, sustentação e manutenções podem passar por avaliações mais criteriosas, a fim de verificar o atendimento aos requisitos solicitados na demanda, disponibilidade do produto de software, qualidade de código, segurança da informação, performance, fidelidade da documentação ao que foi implementado, entre outros aspectos inerentes aos produtos de *software*.

Para isso, podem ser acionados serviços para avaliação da qualidade e realização de testes avançados, durante todo o ciclo do processo de software até a sua implantação, a fim de minimizar a ocorrência de erros no ambiente de produção.

10.1 Testes e Controle de Qualidade do Software

Os testes e controle da qualidade do produto que serão entregues devem permear todo o projeto e execução das *Sprints*, e também entregas relacionadas às demandas de Sustentação e Manutenção de Softwares.

Esta metodologia orienta que, sempre que possível, deve-se promover as seguintes atividades, que podem ser adaptadas à realidade operacional da demanda e do ambiente de TIC da IN:

- a) analisar riscos e a conformidade de processo, projetos, técnicas, práticas e ferramentas de desenvolvimento e testes das empresas que prestam serviços de desenvolvimento de sistemas, em relação às normas e padrões da organização, e às melhores práticas de mercado, no que diz respeito ao versionamento, arquitetura, padrão visual, segurança, testes e qualidade de *software*;
- b) verificar a atualidade e conformidade da documentação técnica dos sistemas de informação da organização em relação ao *software* que está operando;
- c) promover o diagnóstico de problemas de desempenho nos sistemas;
- d) verificar se os padrões de arquitetura de *software* foram obedecidos, e se os padrões de organização de códigos, bibliotecas, componentes foram observados;
- e) detectar falhas e propor correções em processos de testes implantados na organização;
- f) emitir pareceres técnicos relacionados ao ambiente de sistemas da organização;
- g) elaborar Relatório Técnico de Análise de Qualidade.

Os testes devem ser **planejados** conforme práticas a seguir, que podem ser adotadas conforme a realidade do projeto e das demandas:

- a) realizar reuniões com os usuários e/ou times de desenvolvimento para modelar e elaborar estratégias de testes;
- b) planejar testes funcionais e não funcionais de softwares;
- c) elaborar artefatos como Roteiros/Casos de Teste, Listas de Verificação, Critérios de Aceite.

Um dos principais benefícios do planejamento de testes, para assegurar a qualidade, é a redução da dívida técnica. O objetivo é que nenhuma dívida técnica deve ser transferida para outro Sprint, porque os Critérios de Aceitação e de Pronto devem existir, claramente definidos.

A **execução** dos testes deve abordar as seguintes atividades, que podem ser adotadas conforme a realidade do projeto e das demandas:

- a) elaborar artefatos de apoio a testes, como roteiros, *scripts* de testes, relatório de evidências de testes;
- b) executar testes automáticos e/ou manuais em ambiente de testes e de homologação;
- c) executar testes funcionais e não funcionais;
- d) dar suporte aos testes realizados pelo usuário.

É sabido que a qualidade é uma variável complexa e pode ter um caráter subjetivo. Por este motivo, esta metodologia apresentou as diretrizes para definição de qualidade nas entregas de produtos de software.

Ademais, é oportuno frisar que um aumento no escopo, sem o respectivo aumento de tempo ou de recursos, tende a diminuir a qualidade. Assim como, uma redução de tempo ou de recursos, sem redução de escopo, geralmente culmina na diminuição de qualidade. Logo, para evitar problemas relacionados à qualidade é recomendada a manutenção de um “ritmo sustentável” de trabalho, adequando as variáveis escopo, tempo, custo, expectativa do demandante, o que ajuda a melhorar a qualidade a longo prazo.

11 CONSIDERAÇÕES FINAIS

A Metodologia de Desenvolvimento de Soluções de TIC da Imprensa Nacional consolida práticas que promovem o desenvolvimento de produtos e serviços manuteníveis e seguros. Prevê que todo o ciclo do *software* seja acompanhado a fim de assegurar o cumprimento dos requisitos de qualidade, desde sua concepção até a sua descontinuidade.

É importante frisar que os todos os profissionais envolvidos no desenvolvimento de *software* devem seguir as orientações contidas neste documento, e guias complementares bem como propor melhorias e ajustes necessários para complementar ou até mesmo contemplar novas práticas que viabilizem entregas de maior valor para o negócio da Imprensa Nacional.

Casos omissos neste documento devem ser consultados, na sequência, os guias disponibilizados na Wiki da COSIS, o Guia de Projetos de Software com Práticas de Métodos Ágeis para o SISP, e, caso ainda não solucionados, devem ser tratados pela Coordenação de Sistemas, Dados e Inovação - COSIS.

12 REFERÊNCIAS BIBLIOGRÁFICAS

BRASIL. Departamento de Polícia Federal. **Manual de Desenvolvimento de Sistemas (MDS)**. Brasília, DF: Polícia Federal, 2023.

BRASIL. Ministério da Justiça e Segurança Pública. **Manual de Desenvolvimento de Sistemas (MDS)**. Brasília, DF: Ministério da Justiça e Segurança Pública, 2023.

BRASIL. Procuradoria da República. **Manual de Desenvolvimento de Sistemas (MDS)**. Brasília, DF: Procuradoria Da República, 2023.

BRASIL. Imprensa Nacional. **Manual de Desenvolvimento de Sistemas (MDS)**. Brasília, DF: Imprensa Nacional, 2023. Disponível em: <https://www.gov.br/impresnanacional/pt-br/arquivos/arquivos-acoes-e-programas/in_mdss_v4_2021_.pdf>. Acesso em: 13 set. 2024.

BRASIL. **Portaria SGD/MGI nº 750, de 20 de março de 2023**. Estabelece modelo para a contratação de serviços de desenvolvimento, manutenção e sustentação de software, no âmbito dos órgãos e entidades integrantes do Sistema de Administração dos Recursos de Tecnologia da Informação - SISP do Poder Executivo Federal. Ministério da Gestão e da Inovação em Serviços Públicos. Brasília, DF. 2023.

BRASIL. Ministério do Planejamento, Orçamento e Gestão. **Guia de Projetos de Software com práticas de métodos ágeis para o SISP: versão 1.0** / Ministério do Planejamento, Orçamento e Gestão, Secretaria de Logística e Tecnologia da Informação. Brasília: MP, 2015. 90 p.: il.

PRESSMAN, S. R. **Engenharia de Software**. Rio de Janeiro, RJ: Makron Books, 2002.

SCRUMstudy™. **Guia SBOK: Um Guia para o Conhecimento em Scrum**. Arizona, USA: Avondale, 2017.

SCHWABER, KEN; SUTHERLAND, JEFF. **The Scrum Guide**. 2020. Disponível em: <<https://scrumguides.org/scrum-guide.html>>. Acesso em: 15 mai. 2024.

norma ISO/IEC 14764 (2006

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **ABNT NBR 14724: informação e documentação: trabalhos acadêmicos: apresentação**. Rio de Janeiro: ABNT, 2011.



