

Tribunal Regional Eleitoral de Goiás
Secretaria de Tecnologia da Informação
Coordenadoria de Desenvolvimento e Gestão de Sistemas

Guia de Referência



SPRINT7

PROCESSO DE
DESENVOLVIMENTO DE
SOFTWARE

© 2015 – Tribunal Regional Eleitoral de Goiás
SECRETARIA DE TECNOLOGIA DA INFORMAÇÃO - STI
COORDENADORIA DE DESENVOLVIMENTO E GESTÃO DE SISTEMAS - CDGS
Praça Cívica, 300, Centro, Goiânia - GO - 74003-010
(62) 3920-4202

Secretário de Tecnologia da Informação

Dory Gonzaga Rodrigues

Coordenador de Desenvolvimento e Gestão de Sistemas

Roberto Lima Manoel da Costa

Chefe da Seção de Banco de Dados

Leonardo Antônio de Souza

Chefe da Seção de Intranet e Internet

Rafael Dídimo Santos

Chefe da Seção de Sistemas Administrativos

Alexandre Einstein Barcelos Cunha

Adaptação do

. Método de Desenvolvimento com Práticas Ágeis – MAgil

Referências

. Guia do Processo de Desenvolvimento Unificado de *Software* do TSE – PRODUS
. Método ágil de gerenciamento de projetos - SCRUM

Elaboração e Concepção

Roberto Lima Manoel da Costa
Thaís Domingues de Magalhães

Contribuições

Augusto César de Castro Ovelar
Dory Gonzaga Rodrigues
Luís Cláudio Fernandes
Paulo Henrique da Silva
Marcos Mamede Ventura
Marcilio Zaccarelli Bersaneti

Diagramação e arte

Carlos Eduardo Linhares Montenegro

Aprovador por
Roberto Lima Manoel da Costa

Data de aprovação
06/05/2015

Versão
1.0

Alterações

A presente seção possui a finalidade de manter um controle das alterações que serão realizadas neste documento.

Data

Item

Descrição

Sumário

Alterações	2
Sumário	3
Contextualização	5
Situação atual	6
Premissas	7
1. Objetivo	8
2. Papéis e responsabilidades	8
3. Visão geral	15
4. Iniciar o projeto	
Detalhamento do subprocesso	16
Objetivo	17
Descrição das etapas	17
5. Produzir a <i>sprint</i>	
Detalhamento do subprocesso	22
Objetivo	22
Descrição das etapas	23

6. Executar a <i>sprint</i>	
Detalhamento do subprocesso	24
Objetivo	25
Descrição das etapas	25
7. Encerrar a <i>sprint</i>	
Detalhamento do subprocesso	31
Objetivo	32
Descrição das etapas	32
8. Encerrar o projeto	
Detalhamento do subprocesso	34
Objetivo	34
Descrição das etapas	35
9. Artefatos	37
10. Glossário	38

Contextualização

O desenvolvimento de software é uma atividade de natureza intelectual complexa que está em frequente avanço e exige conhecimentos cada vez mais amplos e diversificados. Possui várias etapas como: levantamento de requisitos, análise, projeto, codificação, testes e implantação. A popularização de dispositivos móveis, como *smartphones* e *tablets*, também tem contribuído para o incremento desta complexidade, visto que tendem a aumentar as demandas de soluções para estes aparelhos, os quais devem executar aplicações de forma integrada a sistemas já existentes.

Tendo em vista estes aspectos, um dos principais mecanismos para se obter software de qualidade e cumprir corretamente com os contratos de desenvolvimento de sistemas informatizados é a utilização de um Processo de Desenvolvimento de *Software* (PDS). Estudado dentro da área de

Engenharia de *Software*, um PDS pode ser definido como uma sequência lógica e coordenada de atividades, com a finalidade de obter um produto de *software*.

O Tribunal Regional Eleitoral de Goiás (TRE-GO) possui uma demanda contínua de desenvolvimento de *software* por suas unidades internas e clientes externos (advogados, juristas, magistrados, imprensa e eleitores). A utilização de um processo baseado em metodologias ágeis se revela mais adequada devido aos seguintes aspectos favoráveis: equipe pequena de analistas de sistemas e desenvolvedores na Coordenadoria de Desenvolvimento e Gestão de Sistemas (CDGS) e proximidade entre as áreas de negócio e de tecnologia da informação (TI), dentre outras.

O Tribunal Superior Eleitoral (TSE) elaborou o método de desenvolvimento com práticas ágeis (MAgil) utilizando os conceitos de metodologias ágeis de mercado como *Scrum* e *Extreme Programming*, mas também considerando as boas práticas recomendadas por outras metodologias consagradas como RUP, PMBOK e UML. Assim, levando-se em conta as características da equipe, dos clientes e da maioria dos projetos de desenvolvimento de *software* já realizados no âmbito deste Regional, e buscando estar alinhado à corte superior, optou-se pela adaptação do MAgil para a realidade do TRE-GO.



Figura 1: Ciclo de vida do *Scrum*

Situação atual

Atualmente, a CDGS possui um Processo de Desenvolvimento de *Software* informal, parcialmente definido, e com documentação elaborada em apenas algumas etapas do ciclo de desenvolvimento. A ausência de um padrão e de

documentação em algumas etapas dificulta a transferência e a gestão do conhecimento, gerando retrabalho e resultando em um não aproveitamento dos esforços em projetos futuros.

Premissas

Para a aplicação deste processo, parte-se do pressuposto de que os projetos de *software* repassados a esta Coordenadoria foram previamente priorizados pela administração superior e tiveram seu início autorizado pelo Comitê Gestor de Tecnologia da Informação e Comunicação (CGTIC), preferencialmente tendo sido inseridos no Plano Diretor de Tecnologia da Informação (PDTI). Além disso, é necessário que o cliente possua seu processo de trabalho mapeado, fornecendo-o como insumo para o desenvolvimento do software pretendido.

Por fim, consideram-se aqui os projetos de desenvolvimento de novos sistemas ou manutenção evolutiva em sistemas existentes. As demandas referentes a manutenções adaptativas, perfectivas e corretivas em sistemas legados, além da implantação de softwares de terceiros, poderão utilizar de maneira adaptada este PDS até que sejam tratadas em processos específicos para estas necessidades.

1. Objetivo

Definir o SPRINT7 – Processo de Desenvolvimento de *Software* do TRE-GO – um conjunto de procedimentos interligados, métodos e documentos que se relacionam, a fim de otimizar o controle sobre imprevistos que possam ocorrer durante o ciclo de desenvolvimento de *software*, assim como mensurar sua qualidade e produtividade, além de colaborar com a transferência do conhecimento.

2. Papéis e responsabilidades

Foram definidos os seguintes papéis e responsabilidades neste processo de desenvolvimento de *software*:



Analista de Conformidade

- Verificar se os procedimentos e produtos de trabalho do projeto estão em conformidade com os processos, políticas, e padrões definidos.



Analista de Sistemas

- Auxiliar o Time *Scrum* na elicitação e análise dos requisitos ;
- Auxiliar o Time *Scrum* na documentação do projeto/produto com o apoio do restante do Time.



Analista de Testes

- Auxiliar o Time *Scrum* na elaboração e documentação dos casos de testes;
- Auxiliar o Time *Scrum* na execução dos testes.



Analista de Banco de Dados

- Apoiar o Time *Scrum* na criação e atualização do banco de dados no ambiente de testes;
- Criar, atualizar e gerenciar o controle de acesso no que diz respeito aos bancos de dados de homologação e produção;
- Auxiliar na otimização do código referente aos bancos de dados das aplicações.



Gerente de Projetos

- Definir a equipe do projeto;
- Gerenciar os recursos destinados ao projeto;
- Promover a interação entre os principais envolvidos no projeto;
- Acompanhar o cronograma e o andamento do projeto.



Product Owner (cliente)

- Definir os requisitos do sistema e tomar decisões sobre ele;
- Garantir que todos os *stakeholders* foram consultados e levados em consideração na tomada de decisões;
- Homologar os produtos entregues;
- Elaborar, com o apoio do Time *Scrum*, o manual de operação do produto;
- Estar disponível dedicando-se de forma efetiva ao projeto e atuando como membro da equipe;
- Regulamentar o uso do produto de *software* no âmbito do Tribunal, providenciando quaisquer documentos e aprovações que se fizerem necessárias.



Scrum Master

- Assegurar que a equipe respeite e siga os valores e as práticas do *Scrum*;
- Apoiar o gerente do projeto;
- Remover os impedimentos para a realização da *sprint*, atuando como *facilitador*.



Equipe de Infraestrutura

- Preparar e disponibilizar os ambientes de teste, homologação e produção;
- Implantar em conjunto com o Time Scrum, o produto no ambiente de homologação e produção. Papel a ser assumido por servidor ou colaborador da Coordenadoria de Infraestrutura da STI;
- Participar, quando necessário, da definição de aspectos do ambiente que possam influenciar na definição da solução.



Time Scrum

- Executar as atividades do desenvolvimento do *software*;
- Executar a engenharia de requisitos, de negócio e de sistema;
- Produzir e validar a documentação a ser entregue;
- Definir a arquitetura do *software*;
- Tomar decisões técnicas que orientem todo o design e a implementação do projeto;
- Prezar pela qualidade do código-fonte;

- ⊖ Desenvolver as soluções propostas;
- ⊖ Realizar testes de aceitação, homologação e integração;
- ⊖ Analisar e modelar o banco de dados;
- ⊖ Criar as interfaces solicitadas pelo usuário;
- ⊖ Criar e manter a estrutura do banco de dados;
- ⊖ Zelar pela integridade e melhorar o desempenho do banco de dados.

Por padrão, o Time Scrum é composto pela equipe de desenvolvedores e pelo Scrum Master. Entretanto, qualquer um dos demais papéis citados neste guia pode integrar o Time de acordo com a especificidade de cada projeto.



Titular da CDGS

- ⊖ Receber formalmente a demanda;
- ⊖ Definir o Gerente do Projeto;
- ⊖ Identificar o *Product Owner* em conjunto com o Gerente do Projeto;
- ⊖ Acompanhar a homologação do produto por parte do *Product Owner*;
- ⊖ Encerrar formalmente o projeto em conjunto com o Gerente do Projeto e encaminhá-lo para apreciação da Administração Superior.



Equipe de Suporte

- Receber a capacitação para realizar o suporte ao sistema;
- Prestar suporte de primeiro e segundo níveis quanto ao sistema;
- Filtrar e encaminhar, à unidade responsável, os chamados relacionados a questões para as quais não haja instrução de trabalho, ou que exijam uso de conhecimentos e ferramentas às quais a equipe de suporte não tenha acesso.



Usuário Final

- Ler o manual do produto;
- Participar das capacitações fornecidas para uso do produto.

3. Visão geral

O SPRINT7 é composto por quatro etapas que compreendem todo o ciclo de vida do desenvolvimento de *software* do TRE-GO, e propõe-se a atribuir melhor controle e qualidade no atendimento às demandas deste Tribunal. Tais etapas compreendem: Iniciação do Projeto, Produção da *sprint*, Encerramento da *sprint* e Encerramento do Projeto. Elas são compostas por subprocessos que detalham sua execução.

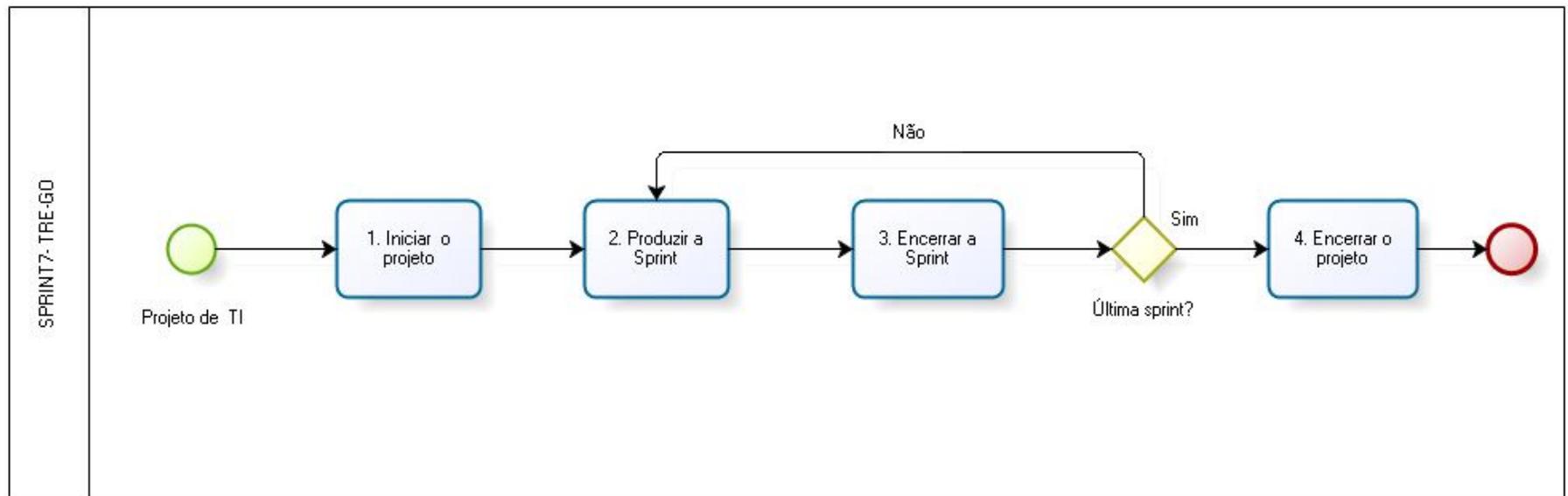


Figura 2: Fluxo principal

As próximas seções trazem o detalhamento de cada um dos subprocessos mostrados na figura 2, informando seus objetivos, papéis envolvidos, entradas e saídas esperadas. Todos os artefatos gerados pelas atividades devem ser registrados na ferramenta de gerenciamento de projetos definida pela equipe.

4. Iniciar o Projeto

Detalhamento do subprocesso

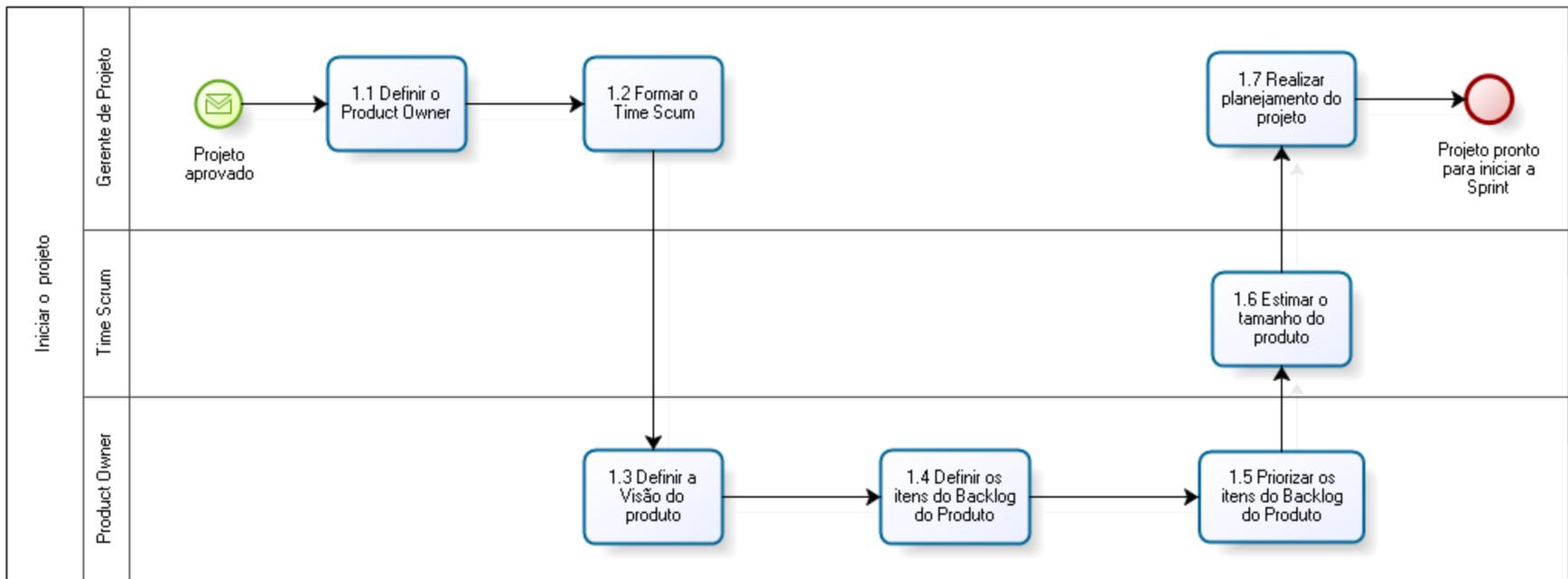


Figura 3: Fluxo do subprocesso iniciar o projeto

Objetivo

Definir a equipe do projeto e planejar o desenvolvimento do sistema, validando algumas práticas ágeis que serão necessárias para as iterações e incrementos do projeto.

Descrição das etapas

Atividades	Papéis	Entradas	Saídas
1.1 Definir o <i>Product Owner</i> (PO)	 	<ul style="list-style-type: none">. Documento formal contendo autorização para o início do projeto (Ex: Termo de Abertura de Projeto, Documento de Oficialização da Demanda, Memorando, etc).	<ul style="list-style-type: none">. <i>Product Owner</i> definido e ciente das práticas ágeis que serão adotadas no projeto;. Projeto criado na ferramenta de gestão de projetos, e <i>Product Owner</i> cadastrado.
<ul style="list-style-type: none">. Definir o responsável pelo produto PO;. Realizar reunião com o PO para contextualizá-lo acerca da utilização do PDS.<ul style="list-style-type: none">■ O representante do cliente deve dedicar-se de forma efetiva ao projeto. Participações esporádicas e superficiais do PO afetarão a qualidade do produto e o resultado final.			

Atividades	Papéis	Entradas	Saídas
<h2>1.2</h2> <h3>Formar o Time <i>Scrum</i></h3>		<ul style="list-style-type: none"> . Documento formal contendo autorização para o início do projeto (Ex: Termo de Abertura de Projeto, Documento de Oficialização da Demanda, Memorando, etc); . <i>Product Owner</i> definido e ciente das práticas ágeis que serão adotadas no projeto; . Projeto criado na ferramenta de gestão de projetos, e <i>Product Owner</i> cadastrado. 	<ul style="list-style-type: none"> . Time <i>Scrum</i> definido para o projeto; . Equipe do projeto registrada na ferramenta de gestão de projetos.
<p>. Definir o Time <i>Scrum</i> que atenderá ao projeto.</p> <p>■ Recomenda-se que o time possua no mínimo 2 integrantes para que seja possível aplicar as técnicas ágeis de desenvolvimento de <i>software</i>.</p>			

<h2>1.3</h2> <h3>Definir a visão do produto</h3>		<ul style="list-style-type: none"> . Documento formal contendo autorização para o início do projeto (Ex: Termo de Abertura de Projeto, Documento de Oficialização da Demanda, Memorando, etc); . Time <i>Scrum</i> definido para o projeto; . Equipe do projeto registrada na ferramenta de gestão de projetos. 	<ul style="list-style-type: none"> . Visão do produto aprovada pelo PO.
<p>. Definir a visão do produto – junto aos clientes, gestores, interessados, entre outros – descrevendo o seu problema atual, as suas necessidades e as expectativas que definem a razão da existência do projeto e as características do produto.</p>			

Atividades	Papéis	Entradas	Saídas
<p>1.4 Definir os itens do <i>Backlog</i> do Produto</p> <ul style="list-style-type: none"> . Identificar as funcionalidades e as condições de desenvolvimento necessárias para atender à visão do <i>PO</i>; . Escrever o <i>backlog</i> do produto; . Elaborar, se necessário, o modelo de casos de uso e outros diagramas UML, para auxiliar na identificação da relação entre as funcionalidades e as interações do sistema; <p>■ Recomenda-se investir na compreensão do sistema como um todo para melhorar as estimativas e garantir um planejamento mais fiel do projeto.</p>		<ul style="list-style-type: none"> . Visão do produto; . Processo de trabalho Mapeado pela unidade solicitante. 	<ul style="list-style-type: none"> . <i>Backlog</i> do produto definido em nível macro.

<p>1.5 Priorizar os itens do <i>Backlog</i> do produto</p> <ul style="list-style-type: none"> . Priorizar os itens do <i>backlog</i> do produto sob a perspectiva do <i>PO</i>, os quais ajudarão na composição das <i>sprints</i> conforme a produtividade do Time <i>Scrum</i>. <p>■ A priorização das funcionalidades deve considerar o impacto dos requisitos não funcionais na arquitetura de tecnologia da informação existente.</p>		<ul style="list-style-type: none"> . Visão do produto; . Processo de trabalho mapeado pela unidade solicitante; . <i>Backlog</i> do produto definido em nível macro. 	<ul style="list-style-type: none"> . <i>Backlog</i> do produto atualizado com a priorização e avaliação de impacto na arquitetura de tecnologia da informação; . Modelo de casos de uso atualizado, se necessário.
---	--	---	--

Atividades	Papéis	Entradas	Saídas
<p>1.6 Estimar o tamanho do produto</p>	 <p>Opcional</p>	<ul style="list-style-type: none"> . Visão do produto; . Processo de trabalho mapeado pela unidade solicitante; . <i>Backlog</i> do produto atualizado com a priorização e avaliação de impacto na arquitetura de tecnologia da informação; 	<ul style="list-style-type: none"> . <i>Backlog</i> do produto atualizado com a estimativa preliminar de tamanho ou esforço necessário para cada item; . Documento de Visão do Produto.
<ul style="list-style-type: none"> . Estimar o tamanho funcional do produto por análise de pontos de função, pontos de história de usuário e/ou através da análise de uma base histórica de projetos semelhantes, caso haja; . É recomendável que seja utilizada também a técnica do <i>Planning Poker</i> para estimativa do esforço de construção das histórias de usuário. 			

Atividades	Papéis	Entradas	Saídas
<p>1.7 Realizar o planejamento do projeto</p>		<ul style="list-style-type: none"> . Documento de Visão do produto; . <i>Backlog</i> do produto atualizado com a estimativa preliminar de tamanho ou esforço necessário para cada item; . Processo de trabalho mapeado pela unidade solicitante. 	<ul style="list-style-type: none"> . Visão do produto atualizada; . Arquitetura do projeto; . Plano de testes; . Estratégias de acompanhamento e monitoramento do projeto; . <i>Backlog</i> do produto atualizado; . Documentação mínima a ser utilizada no projeto definida; . Riscos e seus tratamentos definidos.

- . Realizar reunião com a participação do *SM* e do *GP*, a fim de identificar e analisar os riscos do projeto e a produtividade do *Time Scrum*;
- . Definir a documentação mínima a ser utilizada no projeto;
- . Certificar que o *backlog* do produto esteja pronto para a primeira reunião de planejamento da *sprint*;
- . Definir a arquitetura a ser utilizada na primeira *sprint* do projeto;
- . Definir se é necessário a elaboração de planos e estratégias de teste e de detalhamento de requisitos a serem utilizados no projeto;
- . Preparar o ambiente de desenvolvimento do projeto definindo as ferramentas, os acessos necessários e o repositório de trabalho da equipe;
- . Definir a estratégia de acompanhamento e monitoramento do projeto (tamanho da *sprint*, horário das reuniões diárias, utilização do *Kanban*, atualização de cronograma, *Gráfico Burndown*, etc.);
- . Analisar e definir o tratamento a ser dado aos riscos identificados;
- . Revisar o *backlog* do produto e a priorização de seus itens.

5. Produzir a *sprint*

Detalhamento do subprocesso

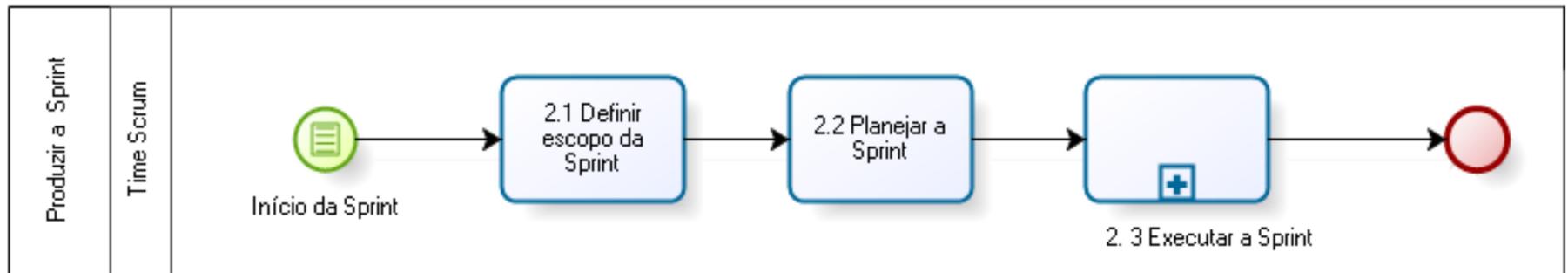
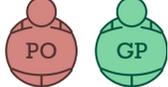


Figura 4: Fluxo do subprocesso produzir a *sprint*

Objetivo

Executar as atividades de geração do produto esperado na meta da *sprint*, com o objetivo de realizar as entregas esperadas pelo *Product Owner*. O acompanhamento destas atividades pode ser feito com o auxílio da metodologia *Kanban*, Gráfico *Burndown* ou do cronograma de atividades.

Descrição das etapas

Atividades	Papéis	Entradas	Saídas
<p>2.1 Definir o escopo da <i>sprint</i></p>	 <p>Opcional</p> 	<ul style="list-style-type: none"> . Documento de visão do produto; . <i>Backlog</i> do produto; . Processo de trabalho mapeado pela unidade solicitante; . Diagramas UML e demais documentos já produzidos; . Arquitetura a ser utilizada no projeto. 	<ul style="list-style-type: none"> . Itens e meta da <i>sprint</i> definida.
<ul style="list-style-type: none"> . Selecionar os itens do <i>backlog</i> do produto que comporão o escopo da entrega ao final da <i>sprint</i>, conforme a prioridade estabelecida pelo <i>PO</i>; . Analisar e entender o objetivo de cada demanda selecionada para a <i>sprint</i>; . Definir meta da <i>sprint</i>; . Caso seja a <i>sprint</i> 0, deverá constar no escopo as tarefas: validação da arquitetura do software, elaboração do modelo de dados e definição do plano de testes. 			
<p>No caso de sistema novo, orienta-se que a primeira <i>sprint</i>, denominada <i>sprint</i> 0, a fim de validar a arquitetura e testar o ambiente de desenvolvimento, tenha como foco a definição da arquitetura do <i>software</i>, a construção do modelo de dados lógico ou conceitual, com base no <i>backlog</i> do produto, e a implementação de ao menos uma funcionalidade básica.</p>			
<p>2.2 Planejar a <i>sprint</i></p>		<ul style="list-style-type: none"> . Documento de visão do produto; . Processo de trabalho mapeado pela unidade solicitante; . Meta e itens da <i>sprint</i>. 	<ul style="list-style-type: none"> . Atualização dos itens da <i>sprint</i>, conforme informações obtidas na reunião de planejamento da <i>sprint</i>.
<ul style="list-style-type: none"> . Decompor os itens da <i>sprint</i> em tarefas que possam ser feitas em até um dia de forma a coincidir com as reuniões diárias; . Definir os responsáveis por executar cada tarefa, registrando essa informação junto à tarefa ou a um cronograma; . Colocar em prática a estratégia definida para acompanhamento do projeto/<i>sprint</i> (montar o <i>Kanban</i>, preparar o gráfico de <i>Burndown</i>, etc). 			

6. Executar a *sprint*

Detalhamento do subprocesso

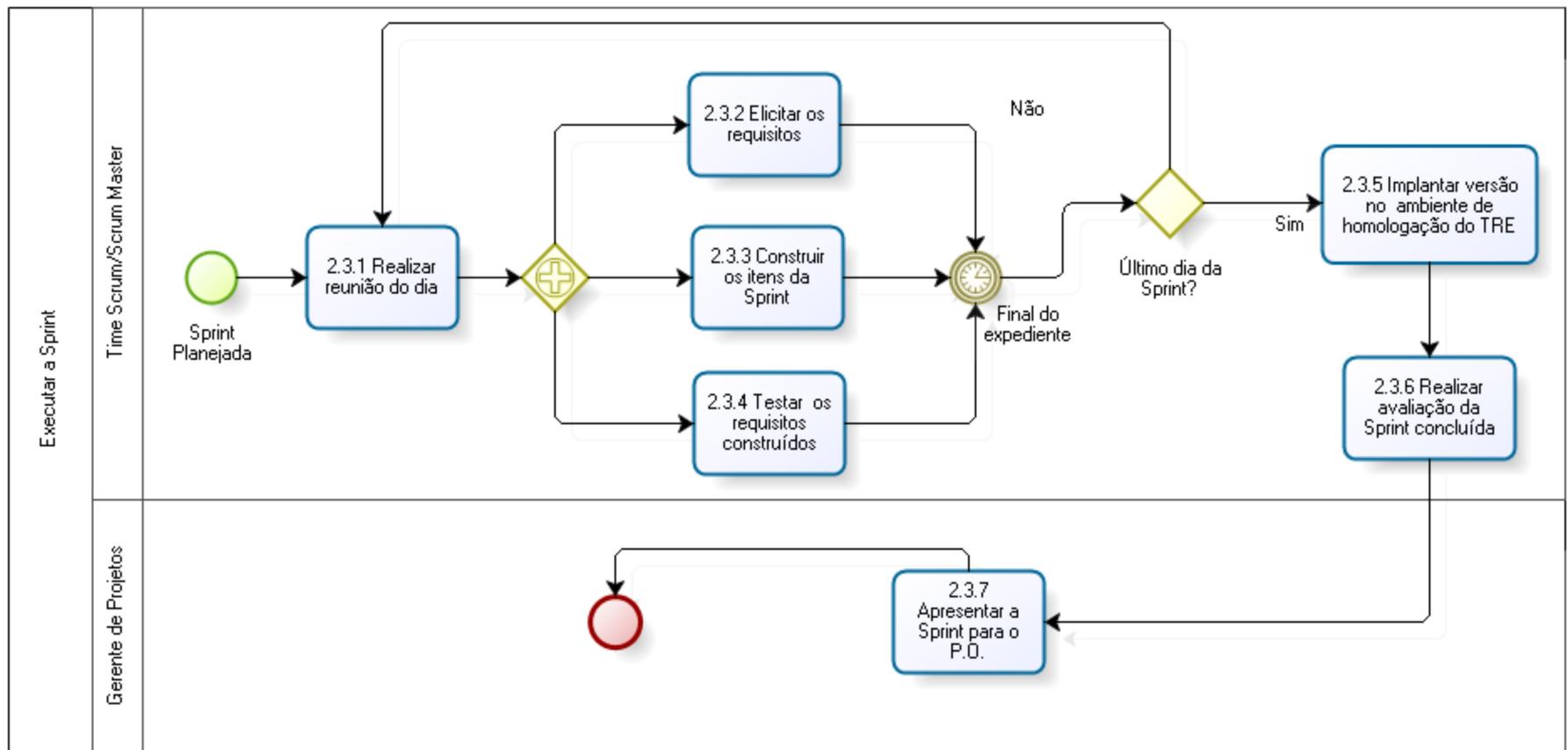


Figura 5: Fluxo do subprocesso executar a *sprint*

Objetivo

Executar as tarefas de execução dos itens da *sprint*, praticando as técnicas e princípios de métodos ágeis conforme os padrões e as políticas definidas pelo TRE, até que seja possível a conclusão do incremento do produto a ser entregue.

Descrição das etapas

Atividades	Papéis	Entradas	Saídas
2.3.1 Realizar reunião do dia	 Opcional Opcional 	. Itens da <i>sprint</i> .	. Atividades da <i>sprint</i> atualizadas e meta alinhada; . Riscos atualizados caso necessário.

- . Monitorar as atividades do dia e atualizar a situação da *sprint* por meio do *Kanban*, do gráfico de *Burndown*, entre outros;
- . Levantar e atualizar os riscos que devem ser monitorados e/ou escalonados.

A reunião diária deve durar aproximadamente 15 minutos, em pé para que o time responda às seguintes perguntas: O que foi feito desde a última reunião? O que se pretende fazer até a próxima reunião? Houve ou está havendo algum impedimento?

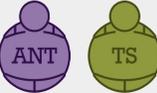
Durante as reuniões, as questões que requeiram mais tempo de discussão devem ser tratadas separadamente com os envolvidos para não comprometer todo o time. Os impedimentos que configurarem riscos de caráter mais abrangente ou institucional devem ser registrados na ferramenta de gestão de projetos para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

Atividades	Papéis	Entradas	Saídas
<p>2.3.2 Elicitar os requisitos</p>	 <p>Opcional</p>	<ul style="list-style-type: none"> . Documento de Visão do produto; . Definição da documentação mínima definida para o projeto; . Documento de planejamento da <i>sprint</i>; . Atas de reuniões, questionários; . Ferramentas e modelos de documentos; . Processo de trabalho mapeado pela unidade solicitante. 	<ul style="list-style-type: none"> . Funcionalidade detalhada com histórias de usuário, regras de negócio, protótipos e outros artefatos definidos para o projeto; . Modelos UML necessários para a implementação das funcionalidades criadas; . Documentação mínima definida para o projeto produzida e/ou atualizada.
<ul style="list-style-type: none"> . Entrevistar o <i>Product Owner</i> para detalhar o requisito a ser construído; . Documentar o requisito detalhado nas histórias de usuário ou em outros artefatos de acordo com a documentação mínima definida para o projeto; . Definir as regras de negócio para cada funcionalidade; . Definir e documentar os protótipos de tela; . Modelar o banco de dados; . Elaborar o dicionário de dados; . Produzir os modelos UML necessários para a implementação das funcionalidades; . Validar, junto ao <i>Product Owner</i>, os requisitos e as regras de negócio por meio dos protótipos de tela; . Identificar e documentar os itens do glossário do sistema. 			

Atividades	Papéis	Entradas	Saídas
<h3>2.3.3</h3> <p>Construir os itens da <i>sprint</i></p>		<ul style="list-style-type: none"> . Itens da <i>sprint</i>; . Processo de trabalho mapeado pela unidade solicitante; . Modelo do banco de dados; . Dicionário de dados; . Histórias de usuário; . Regras de negócio; . Modelos UML necessários para a implementação das funcionalidades criadas; . Protótipos de tela. 	<ul style="list-style-type: none"> . Funcionalidades codificadas; . Documentos e artefatos do projeto atualizados; . <i>Scripts</i> de banco de dados executados; . Todos os documentos e tarefas atualizados na ferramenta de gerenciamento de projetos.

- . Programar os códigos para implementação do produto;
- . Realizar testes unitários;
- . Construir as interfaces conforme a validação dos protótipos de tela feita pelo *Product Owner*;
- . Utilizar a infraestrutura de integração contínua a fim de possibilitar uma avaliação diária do trecho do *software* construído.

■ **Recomenda-se a utilização de práticas como programação em pares e outras *técnicas do XP*.**

Atividades	Papéis	Entradas	Saídas
<p>2.3.4 Testar os requisitos implementados</p>		<ul style="list-style-type: none"> . Plano e/ou estratégia de teste; . Itens da <i>sprint</i>; . Histórias de usuário; . Funcionalidades codificadas; . Regras de negócio; . <i>Código-fonte</i> das funcionalidades definidas na <i>sprint</i> respectiva; . Modelo de banco de dados; . Diagramas da UML e outros artefatos do projeto definidos pelo time. 	<ul style="list-style-type: none"> . Testes automatizados construídos e executados no ambiente de testes; . Documento de estratégia de testes atualizado; . Produto validado com geração de evidências.
<ul style="list-style-type: none"> . Aplicar as estratégias de teste definidas para o projeto a fim de minimizar os defeitos e não conformidades do produto no momento de sua verificação no ambiente de homologação, aumentando as chances de sua aprovação pelo <i>Product Owner</i>. . Realizar testes unitários das funcionalidades; . Preparar e realizar os testes de integração e de regressão (impacto de uma nova funcionalidade em outras partes funcionais do <i>software</i>); . Realizar verificação e validação do <i>software</i> construído até o momento na <i>sprint</i>. <p>Os testes unitários, de integração e funcionais devem ser automatizados na medida do possível. Recomenda-se a utilização das ferramentas definidas pela equipe para automatização dos testes, nos respectivos níveis.</p> <p>Todo o <i>software</i> deverá ser verificado e validado ao menos uma vez antes de ser disponibilizado no ambiente de produção.</p>			

Atividades	Papéis	Entradas	Saídas
<p>2.3.5 Implantar versão no ambiente de homologação do TRE</p> <ul style="list-style-type: none"> . Criar ou atualizar o banco de dados, caso necessário; . Realizar o <i>deploy</i> do código-fonte , dos documentos e dos artefatos no repositório do projeto; . Realizar auditoria do código-fonte; . Implantar a versão do produto desenvolvido na respectiva <i>sprint</i> no ambiente de homologação; . Testar os itens da <i>sprint</i> em ambiente de homologação. 		<ul style="list-style-type: none"> . Testes automatizados construídos e executados no ambiente de testes; . Documento de estratégia de testes atualizado; . Código-fonte produzido na respectiva <i>sprint</i>; . Produto desenvolvido na respectiva <i>sprint</i>; . Modelo de Banco de Dados; . Modelos UML. 	<ul style="list-style-type: none"> . Código-fonte, documentos e artefatos disponibilizados no repositório do projeto; . Produto implantado no ambiente de homologação TRE, para verificação por parte do <i>Product Owner</i>.
<p>2.3.6 Realizar avaliação da <i>sprint</i> concluída</p> <ul style="list-style-type: none"> . Verificar se a meta da <i>sprint</i> foi alcançada; . Registrar os problemas identificados e as lições aprendidas; . Preencher <i>checklist</i> de final de <i>sprint</i>; . Revisar e atualizar documentação do projeto e o repositório de gestão do conhecimento. 		<ul style="list-style-type: none"> . Produto da <i>sprint</i>, implantado no ambiente de homologação do TRE. 	<ul style="list-style-type: none"> . Dificuldades e lições aprendidas registradas; . <i>Checklist</i> de final de <i>sprint</i> preenchido; . Documentação do projeto e repositório de gestão do conhecimento atualizados.

Atividades	Papéis	Entradas	Saídas
------------	--------	----------	--------

2.3.7

Apresentar a *sprint* ao PO



- . Produto da *sprint*, implantado no ambiente de homologação do TRE-GO;
- . Itens e metas estabelecidos para a *sprint*;
- . *Checklist* de final de *sprint*.

- . Apresentação da *sprint* ao PO realizada.

- . Apresentar os resultados da *sprint* ao *Product Owner*;
- . Registrar as não conformidades ou sugestões verificadas durante a apresentação da *sprint*.

▮ Recomenda-se que a validação das regras de negócio pelo PO seja feita preferencialmente no próprio sistema no ambiente de homologação.

7. Encerrar a *sprint*

Detalhamento do subprocesso

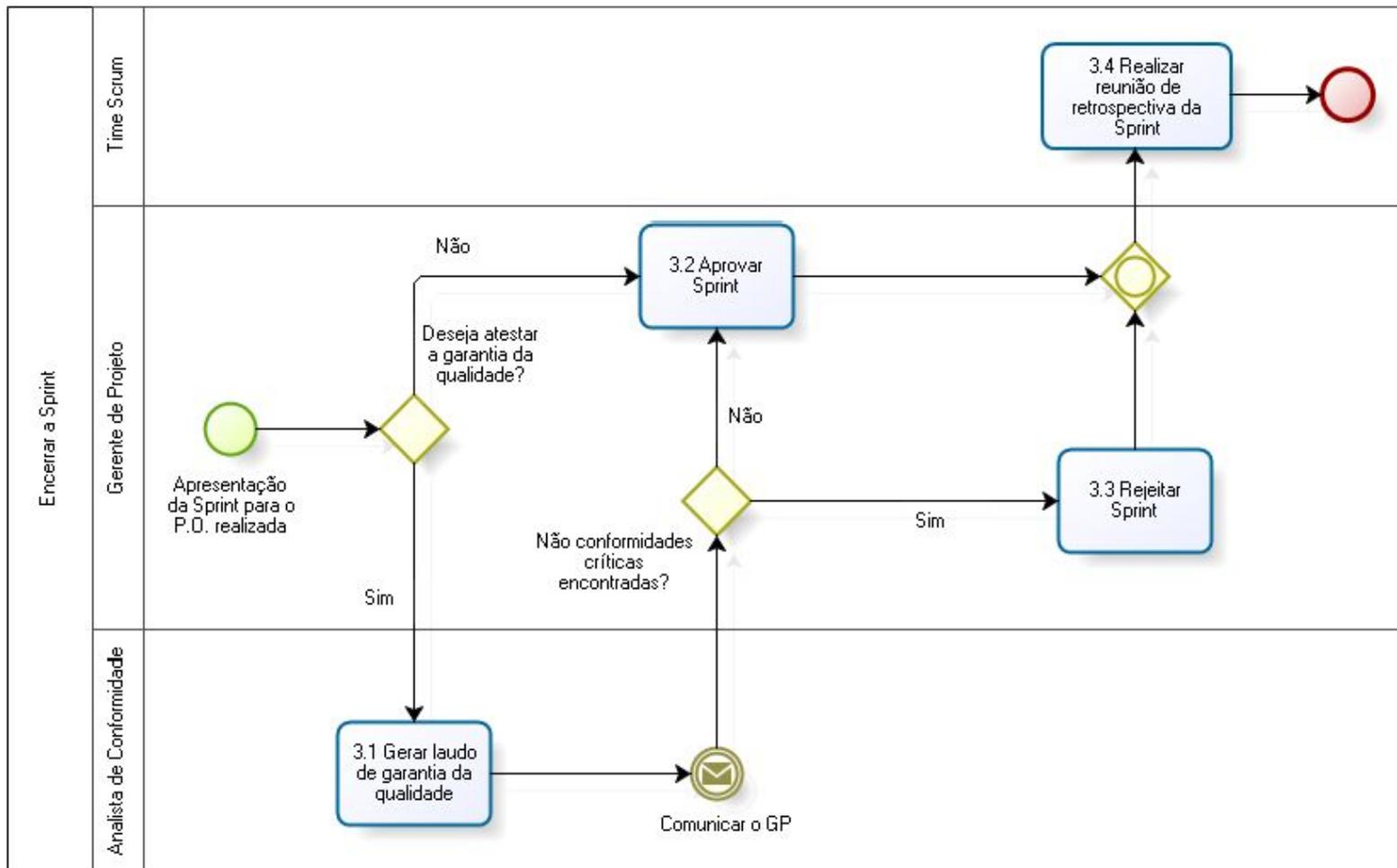


Figura 6: Fluxo do subprocesso encerrar a *sprint*

Objetivo

Atestar a qualidade do produto da *sprint* entregue, realizar inspeções e adaptações visando a melhoria do processo, dos procedimentos, das técnicas e do método de desenvolvimento.

Descrição das etapas

Atividades	Papéis	Entradas	Saídas
3.1 Gerar laudo de garantia da qualidade	 	<ul style="list-style-type: none">. Produto da <i>sprint</i>, disponível no ambiente de homologação do TRE;. Repositório do projeto atualizado com todos os documentos, artefatos e <i>código-fonte</i> produzidos durante a <i>sprint</i>.	<ul style="list-style-type: none">. Laudos gerados a fim de atestar a qualidade do produto da <i>sprint</i> e a conformidade com os padrões estabelecidos.

- . Validar a qualidade do produto entregue;
- . Realizar inspeções e adaptações caso necessário, nos procedimentos, técnicas e no processo de desenvolvimento de software;
- . Validar a arquitetura, o plano de testes, a conformidade das entregas com os requisitos definidos, o banco de dados e gerar os respectivos laudos.
 - ▮ Cabe ao gerente do projeto avaliar os laudos emitidos e decidir pela adequação ou não das não conformidades encontradas, por meio de um laudo consolidado.

Atividades	Papéis	Entradas	Saídas
<p>3.2 Aprovar a <i>sprint</i></p> <ul style="list-style-type: none"> . Aprovar os produtos gerados pela <i>sprint</i>, após análise dos laudos de conformidade previstos; . Reportar ao Time <i>Scrum</i> e às partes interessadas o resultado da aprovação da <i>sprint</i>. <p>A aprovação pode ser integral ou parcial (com ressalvas). A aprovação parcial se dará quando as não conformidades apontadas pelos laudos forem admissíveis, em comum acordo com o <i>Product Owner</i>.</p>		<ul style="list-style-type: none"> . Laudos de conformidade previstos na <i>sprint</i>. 	<ul style="list-style-type: none"> . Produtos da <i>sprint</i> aprovados com ou sem ressalvas e liberados para implantação em produção, registrados em ata de reunião.

<p>3.3 Rejeitar a <i>sprint</i></p> <ul style="list-style-type: none"> . Verificar os laudos emitidos pelo Analista de Conformidade; . Rejeitar os produtos entregues pela <i>sprint</i>, caso existam não conformidades críticas e impeditivas, apontadas no laudo de conformidade; . Reportar ao Time <i>Scrum</i> e às partes interessadas o resultado da rejeição da <i>sprint</i>. 		<ul style="list-style-type: none"> . Laudos de conformidade previstos na <i>sprint</i>. 	<ul style="list-style-type: none"> . Produtos da <i>sprint</i> rejeitados; . Riscos do projeto atualizados.
--	--	--	---

<p>3.4 Realizar reunião de retrospectiva da <i>sprint</i></p> <ul style="list-style-type: none"> . Realizar a inspeção no subprocesso de “produção da <i>sprint</i>”, e propor adaptações para a sua melhoria quanto a atividades, pessoas, ferramentas, técnicas etc; . Discutir e registrar os pontos positivos e negativos da <i>sprint</i>. 		<ul style="list-style-type: none"> . Situações relevantes ocorridas durante <i>sprint</i>. 	<ul style="list-style-type: none"> . Lições aprendidas registradas.
---	--	---	--

8. Encerrar o projeto

Detalhamento do subprocesso

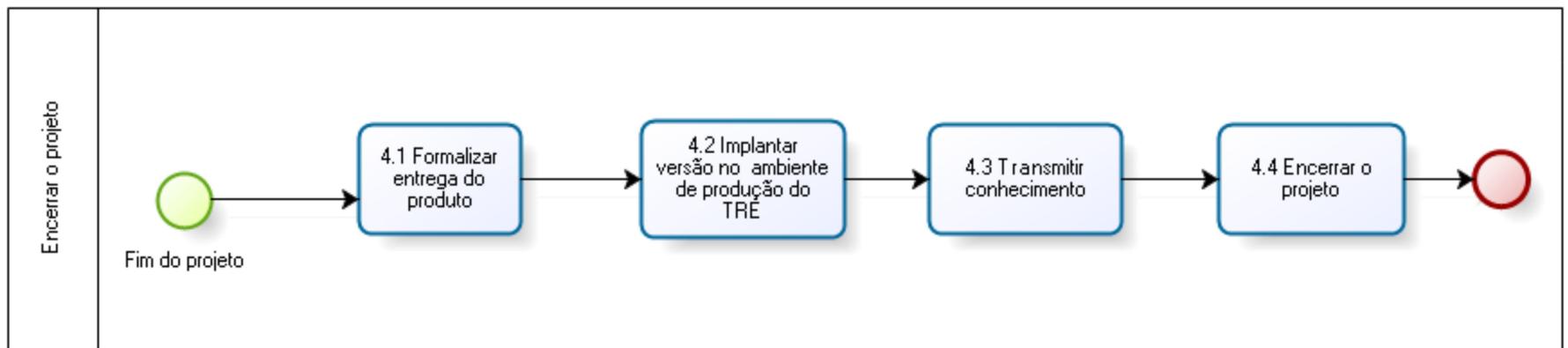


Figura 7: Fluxo do subprocesso encerrar o projeto

Objetivo

Realizar a transferência de conhecimento e verificar se as expectativas do cliente foram atendidas.

Descrição das etapas

Atividades	Papéis	Entradas	Saídas
4.1 Formalizar a entrega do produto . Homologar a entrega do produto; . Elaborar o Termo de Homologação do Sistema e colher as assinaturas do PO e do Titular da Unidade no documento.		<ul style="list-style-type: none">. Produto disponível no ambiente de homologação do TRE;. Repositório do projeto atualizado com todos os documentos, artefatos e código-fonte produzidos durante as <i>sprints</i>.	<ul style="list-style-type: none">. Termo de Homologação do Sistema preenchido e assinado pelo PO.
4.2 Implantar versão no ambiente de produção do TRE . Implantar o produto no ambiente de produção do TRE.		<ul style="list-style-type: none">. Termo de Homologação do Sistema preenchido e assinado pelo PO;. Produto aprovado e verificado no ambiente de homologação do TRE.	<ul style="list-style-type: none">. Sistema regulamentado no âmbito do Tribunal;. Produto implantado no ambiente de produção.

Atividades	Papéis	Entradas	Saídas
<p>4.3 Transmitir conhecimento</p>		<ul style="list-style-type: none"> . Produto implantado no ambiente de produção; . Sistema regulamentado no âmbito do Tribunal; . Termo de Homologação do Sistema preenchido e assinado pelo PO; . Repositório do projeto atualizado. 	<ul style="list-style-type: none"> . Usuários aptos a operar o sistema; . Equipes de suporte aptas a prestar suporte aos usuários do sistema e/ou a direcionar os chamados; . Ferramenta de gestão do conhecimento atualizada com as informações necessárias para operação, suporte e manutenção do sistema.

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

O PO deverá participar da elaboração do manual de operação do produto e regulamentar seu uso no âmbito do Tribunal, providenciando quaisquer documentos e aprovações que se fizerem necessárias.

<p>4.4 Encerrar o projeto</p>		<ul style="list-style-type: none"> . Produto implantado no ambiente de produção; . Termo de Homologação do Sistema preenchido e assinado pelo <i>Product Owner</i>. 	<ul style="list-style-type: none"> . Pesquisa de satisfação do produto final; . Documento formal que ateste o encerramento do Projeto e entrega do produto.
--	--	---	---

. Realizar pesquisa de satisfação com o usuário solicitante e com o PO;
 . Realizar o encerramento do projeto.

9. Artefatos

Os artefatos listados a seguir poderão ser gerados no decorrer do processo e são compostos por um ou mais elementos. Eles serão detalhados na próxima versão deste guia e os seus modelos disponibilizados em documentos anexos.

- Termo de abertura de projeto;
- Documento de oficialização da demanda;
- Termo de encerramento do projeto;
- Ata de reunião;
- Documento de Visão;
- Documento de arquitetura;
- Documento de Planejamento da *sprint*;
- Documento de Planejamento dos Testes;
- Questionário;
- História de usuário;
- Regra de negócio;
- *Sprint review*;
- Manual de operação;
- Documento descritivo do banco de dados;
- Documento de modelagem do produto;
- Documento de conclusão da *sprint*;
- Laudo da qualidade do produto;
- Termo de homologação do sistema;
- Pesquisa de satisfação a respeito do produto entregue.

10. Glossário

Ambiente de produção. Ambiente por meio do qual o cliente aceita a propriedade de produtos de *software* como cumprimento parcial ou completo de um projeto/contrato.

Arquitetura. O conceito de nível mais elevado de um sistema em seu ambiente, de acordo com o IEEE (*Institute of Electrical and Electronics Engineers*). A arquitetura de um sistema de *software* (em um determinado ponto) é a organização ou a estrutura dos componentes significativos interagindo por meio de interfaces, aqueles componentes sendo constituídos de componentes e interfaces sucessivamente menores.

Artefato. Um produto de trabalho formal que: **1.** é produzido, modificado ou utilizado por uma tarefa; **2.** define uma área de responsabilidade; **3.** está sujeito ao controle de versão. • Um Artefato pode ter múltiplos formulários, incluindo um modelo, um elemento de modelo ou um documento.

Backlog. É uma lista que contém todas as funcionalidades desejadas para um novo produto ou para a evolução de um já existente. O conteúdo dessa lista é definido pelo *Product Owner*. O *backlog* do produto é dinâmico e pode ser modificado ao longo do projeto. Refere-se aos serviços, tarefas ou funções que podem ser utilizados de alguma maneira pelos usuários do sistema. Os casos de uso são utilizados para expressar e documentar os comportamentos pretendidos para as funções do sistema.

Código-fonte. é o conjunto de palavras ou símbolos escritos de forma ordenada, contendo instruções em uma das linguagens de programação existentes, de maneira lógica.

Comitê Gestor de Tecnologia da Informação e Comunicação (CGTIC). Comitê diretivo de tecnologia da informação constituído no âmbito do TRE-GO através da Portaria nº 147/2012 – PRES, para deliberar sobre estratégias, investimentos, e contratações de tecnologia da informação.

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

Extreme Programming ou Metodologia XP. Metodologia ágil para equipes pequenas e médias que desenvolvem *software* com requisitos vagos e em constante mudança. Para isso, adota a estratégia de constante acompanhamento e realização de pequenos ajustes durante o desenvolvimento do *software*. Os cinco valores fundamentais da XP são: **1.** comunicação; **2.** simplicidade; **3.** *feedback*; **4.** coragem e **5.** respeito.

Garantia de qualidade. Todas as ações planejadas e sistemáticas necessárias ao fornecimento da confiança adequada que um produto ou serviço satisfará, dados os requisitos de qualidade.

Gráfico Burndown. Ferramenta que permite o acompanhamento da evolução do serviço executado na *sprint*. O gráfico representa, no eixo vertical (y), a quantidade de trabalho que falta ser feito *versus* o tempo, no eixo horizontal (x).

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

Iteração. Uma sequência distinta de atividades com um plano criado através de linha de base e critérios de avaliação que resultam em uma liberação (*release*) do produto, interna ou externa.

Kanban. É um termo de origem japonesa e significa literalmente “cartão” ou “sinalização”. É um método de gestão de mudanças que dá ênfase aos seguintes princípios: **1.** visualizar o trabalho em andamento; **2.** visualizar cada passo em sua cadeia de valor, do conceito geral até *software* que se possa lançar; **3.** tornar explícitas as políticas sendo seguidas, entre outros princípios. • Pode-se utilizar cartões (*post-it* e outros) para indicar o andamento do desenvolvimento do *software*, colocando neles indicações sobre uma determinada tarefa, por exemplo: “para executar”, “em andamento” ou “finalizado”.

Modelo de dados. Descreve as representações lógicas e físicas dos dados persistentes utilizados pelo aplicativo. Nos casos em que o aplicativo utilizará um Sistema Gerenciador de Banco de Dados (SGBD), o modelo de dados poderá incluir também elementos de modelo para procedimentos armazenados, disparos, restrições etc. que definem a interação dos componentes de aplicativo com o SGBD.

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

Papel (função). Uma definição do comportamento e das responsabilidades de um indivíduo - ou conjunto de indivíduos trabalhando em equipe - dentro do contexto de uma organização de engenharia de *software*.

Pontos de função. É a unidade de medida que representa o tamanho funcional de um *software*.

Product Owner. É a nomenclatura que identifica o indivíduo no papel de cliente no processo de desenvolvimento de *software*. O *Product Owner* é responsável por determinar as prioridades do que deve ser feito no *software*. Deve ser uma pessoa capaz de sanar as dúvidas de requisitos, subsidiar as decisões e informações necessárias para a construção da solução a ser proposta, e deve conhecer integralmente as necessidades do cliente. Pode ser o próprio cliente, ou alguém indicado por ele, que tenha autonomia para tomar decisões em seu nome.

Regras de negócio. Assertiva que define ou limita algum aspecto do negócio. • Por exemplo, “O intervalo mínimo para o almoço dos servidores é de uma hora”.

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

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

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

Sprint Review (Apresentação da *sprint*). É uma reunião que deve ser realizada nos últimos dias da *sprint*, com a participação de todos, para apresentar o resultado do trabalho de toda a *sprint* e, também, levantar

os pontos positivos e negativos. Nessa reunião, podem ser levantadas novas funcionalidades que não foram entregues e a necessidade de novos itens.

Testes de aceitação. Uma instância específica de uma configuração de *hardware* e *software* criada com a finalidade de instalar e disponibilizar, para os usuários, o *software* desenvolvido para o uso pretendido.

Técnicas do XP. O *Scrum* é focado nas práticas de gerenciamento e organização, enquanto o XP dá mais atenção às tarefas de programação. Muitas das Técnicas do XP são utilizadas largamente pelas equipes de programação. Cita-se, como exemplos: **1.** programação em par; **2.** desenvolvimento orientado a testes (TDD); **3.** *design* incremental; **4.** integração contínua; **5.** propriedade coletiva do código; **6.** ambiente de trabalho informativo; **7.** padrão de codificação, entre outros.

Time Scrum. É um conjunto de pessoas com habilidades multidisciplinares, capaz de se auto-organizar para produzir o produto com qualidade e valor para o cliente.

Validação. é uma atividade que tem como objetivo assegurar que o produto final corresponda aos requisitos do *software* (Estamos construindo o produto certo?)

Verificação. é uma atividade que tem como objetivo assegurar consistência, completitude e corretitude do produto em cada fase e entre fases consecutivas do ciclo de vida do *software* (Estamos construindo corretamente o produto?)