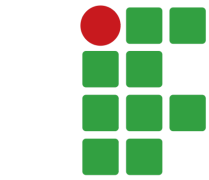

INSTITUTO
FEDERAL
Farroupilha

PDS

Processo de Desenvolvimento de *Software*

Santa Maria, 2023.

Sumário

1. Introdução

Este documento estabelece um processo de engenharia de *software*, com base nos artefatos e eventos Scrum, uma metodologia ágil para desenvolvimento de *software* moderno. Se aplica a todas as unidades do Instituto Federal Farroupilha (IFFar), que idealmente devem utilizar bases tecnológicas equivalentes. A meta é garantir a produção de *software* de qualidade, que atenda aos requisitos e necessidades dos usuários. Para fins deste documento, considera-se:

I - Ambiente de Desenvolvimento Integrado (IDE): Do inglês *Integrated Development Environment*, trata-se de um programa de computador que reúne características e ferramentas de apoio ao desenvolvimento de *software* com o objetivo de agilizar este processo.

II - Banco de dados: coleções organizadas de dados que se relacionam de forma a criar algum sentido (informação) e conferir mais eficiência a uma pesquisa ou a um estudo.

III - Controle de versão (versionamento): processo que tem como finalidade gerenciar diferentes versões no desenvolvimento de um sistema.

IV - Escopo: características e funções que singularizam um produto, serviço ou resultado.

V - *Framework*: conjunto de classes implementadas em uma linguagem de programação específica, usadas para auxiliar o desenvolvimento de *software*.

VI - Linguagem de programação: método padronizado para comunicar instruções para um computador. Pode ser também um conjunto de regras sintáticas e semânticas usadas para definir um programa de computador.

VII - Módulo: uma parte de sistema de informação que utiliza a mesma arquitetura tecnológica do sistema principal. É responsável por atividades que atendem a um assunto bem definido.

VIII - *Open-source*: *software* com o código-fonte aberto e licença gratuita.

IX - PDTI: Plano Diretor de Tecnologia da Informação, instrumento de diagnóstico, planejamento e gestão dos recursos e processos de Tecnologia da Informação (TI).

X - Projeto: esforço temporário empreendido para criar um produto, serviço ou resultado exclusivo, tendo uma data prevista para iniciar e uma data prevista para terminar.

XI - *Sprint*: período/ciclo entre 1 a 4 semanas estabelecido para o desenvolvimento de

um grupo de tarefas do projeto.

XII - *Product Owner* (dono do produto): servidor da área finalística que possui conhecimento de aplicação da área (regra de negócio); responsável pelos requisitos e testes do sistema.

XIII - Proprietário: *software*, geralmente pago, que é licenciado com direitos exclusivos para o produtor/desenvolvedor.

XIV - Protocolo: convenção que controla e possibilita conexão, comunicação e transferência de dados entre dois sistemas computacionais.

XV - Requisito: definição documentada de uma propriedade ou comportamento que um produto ou serviço particular deve atender. Pode ser funcional, indicando uma funcionalidade de um sistema, ou não-funcional, indicando, por exemplo, restrições dentro do sistema.

XVI - Wiki: Termo utilizado para identificar um tipo específico de coleção de documentos em hipertexto, ou o *software* colaborativo usado para criá-lo.

1.1 Gerência de projetos

Para gerência de projetos recomenda-se a utilização das ferramentas Jira ou GitLab. Cada colaborador poderá criar novas tarefas e gerenciá-las dentro de um projeto, bem como atribuir a si mesmo a execução de determinadas tarefas. O gerente de projetos, juntamente com o *Product Owner*, são os responsáveis pelos artefatos e atividades Scrum, devendo acompanhar o andamento global das tarefas, considerando o cronograma e os recursos disponíveis. O código-fonte do projeto deve ser armazenado no GitLab.

1.2 Orientações para documentação

O *dashboard* da ferramenta de gestão de projetos deve, idealmente, ser dividido em seis seções, em que cada uma delas significa uma situação da tarefa.

- **BACKLOG:** tarefas na fila de espera, a serem desenvolvidas de acordo com as prioridades definidas pelo gerente de projetos em acordo com o *Product Owner* (futuro a médio/longo prazo).
- **TO DO:** tarefas priorizadas pelo gerente de projetos, que devem ser executadas na sequência (futuro a curto prazo).

- **EM ANDAMENTO:** tarefas que estão sendo desenvolvidas no momento atual.
- **EM TESTE:** tarefas que foram desenvolvidas e estão disponíveis para testagem.
- **DONE:** tarefas desenvolvidas, que foram aprovadas pelo testador e já podem passar pelo *merge* (serem carregadas para o ambiente de produção).
- **CLOSED:** tarefas concluídas e finalizadas.

O andamento de cada tarefa deverá ser atualizado regularmente (no mínimo semanalmente), na ferramenta de gestão de projetos, pelo próprio colaborador responsável pela tarefa, para que todos os envolvidos no projeto acompanhem o progresso geral do trabalho.

Todo sistema desenvolvido deve manter publicizada e atualizada toda a documentação necessária ao uso e manutenção do sistema.

1.3 Colaboração externa

Toda a comunidade do IFFar poderá utilizar este PDS para o desenvolvimento de sistemas. Para colaboração externa, a Coordenação de Sistemas (CSI) / Diretoria de Tecnologia da Informação (DTI) fornecerá acesso completo ao PDS, *templates* e manuais.

A documentação realizada dentro dos padrões estabelecidos no PDS poderá se tornar um projeto de *software* homologado pelo IFFar. Para isso, o colaborador deverá fornecer todos os documentos obrigatórios e seguir os procedimentos descritos neste documento.

1.4 Preparando o ambiente de desenvolvimento

Para que todos os projetos desenvolvidos sejam concretizados com maior eficiência e rapidez, e também para uma boa manutenção dos códigos e continuidade dos sistemas institucionais, é importante uma padronização nas tecnologias utilizadas, tais como linguagens de programação, *frameworks*, IDEs e banco de dados.

As ferramentas, *frameworks* e *softwares* listadas a seguir são atualmente utilizadas para desenvolvimento e manutenção de sistemas no IFFar.

- Git (Sistema de versionamento de *software*).
- PHPStorm/VSCode/Eclipse (IDEs para desenvolvimento de *software*).
- Android Studio/VSCode/IntelliJ (IDE para desenvolvimento móvel).
- DataGrip/DBeaver (IDE para acesso a banco de dados).

- MySQL WorkBench (IDE para modelagem de banco de dados).
- Node+ yarn (Gerenciador de pacotes node para *front-end*).
- CakePHP/Laravel/Django/Spring/Hibernate (*Frameworks* de desenvolvimento).
- Vue.js/Angular (*Frameworks* de *front-end*).
- Flutter/Ionic (*Frameworks* para desenvolvimento móvel).
- Docker (Sistema de contêineres de ambientes de desenvolvimento).
- Bootstrap (*Framework* de interface *web*).
- FileZilla (Gerenciador de arquivos FTP e SFTP).
- Pencil 3.0 ou superior (Ferramenta para prototipação de telas).

Dessa forma, deve-se priorizar tais ferramentas, *frameworks* e *softwares*. Na inviabilidade destas, deve-se consultar a DTI.

2. Abertura de Projeto de *Software*

Nesta seção, são descritos os procedimentos para a abertura de um novo projeto de desenvolvimento de *software*. Abrange as etapas de solicitação de *software*, avaliação de viabilidade e necessidade de implementação, definição da equipe de trabalho e abertura do projeto. É importante ressaltar que todo documento gerado deve ser anexado à Wiki do projeto no *GitLab*, criando assim um repositório único de documentos do *software*.

2.1 Solicitação de *software*

Para solicitar um novo projeto de *software*, o responsável por um setor ou departamento de uma unidade institucional (Reitor, Pró-Reitor, Diretor e Coordenador) que possui a demanda deve preencher o Formulário de Solicitação de *Software* (Anexo I). Quando o servidor não possuir cargo de chefia, este poderá se fazer representar por seu chefe imediato.

O Anexo I deverá ser enviado à DTI do IFFar via memorando, seguindo os procedimentos listados no Regulamento de Sistemas Institucionais do IFFar. A DTI terá um prazo de até 60 dias para avaliar e responder à solicitação.

2.2 Reunião com os interessados

No caso do deferimento do projeto, uma reunião deve ser realizada com os seguintes participantes:

- *Product Owner(s)* (dono(s) do produto);
- Diretor(a) de TI;
- Coordenador(a) de Sistemas;
- Equipe de desenvolvimento.

Esta reunião servirá para levantar aspectos funcionais do *software* a ser desenvolvido, assim como elucidar ajustes eventualmente não especificados no Formulário de Solicitação de *Software*.

2.3 Definição da equipe de trabalho

Em cada novo projeto, pelo menos três pessoas deverão estar envolvidas na equipe de desenvolvimento: uma para responder às demandas e coordenar as atividades, e outras duas para atuar no desenvolvimento e codificação dos requisitos. Dessa forma, os papéis mínimos são: gerente de projetos, analista/codificador e codificador/testador. Os demais integrantes, assim como seus respectivos papeis, deverão ser alocados no projeto criado no *GitLab*, juntamente com suas respectivas tarefas. Esta equipe será responsável pela implementação e revisão dos componentes/elementos do sistema e pela realização de testes unitários sobre os mesmos.

Caso exista necessidade ou falta de conhecimento para a criação do banco de dados, poderá ser criado o papel do *Database Administrator (DBA)*, responsável por criar as tabelas e dependências necessárias.

O(s) *Product Owner*(s) deve(m) auxiliar nas definições dos requisitos dos novos sistemas/módulos, sejam eles funcionais ou não funcionais, das regras de negócios, das justificativas legais, e atuar nas atividades de testes, treinamento e implantação.

2.4 Abertura de projeto no GitLab

A DTI/CSI abrirá o novo projeto no *GitLab* para o sistema a ser criado. O gerente do projeto será informado e preencherá as informações necessárias. Durante a criação do projeto, será criado e vinculado um repositório, contendo a estrutura de diretórios.

2.5 Definição da visão e principais envolvidos no projeto

Nesta fase, a partir do Formulário de Solicitação de *Software*, será definido o objetivo geral do projeto, bem como seus participantes e responsabilidades, podendo serem utilizados recursos como questionários e/ou entrevistas, normativas, manuais e demais documentos ligados ao projeto. Enquanto o objetivo e as funcionalidades iniciais pretendidas não estiverem claras e bem definidas, caberá ao gerente do projeto viabilizar meios de defini-las clara e objetivamente.

Todos os envolvidos, seja na definição do *software* (*stakeholders*) ou usuários, serão mapeados nesta fase. A documentação destas atividades será realizada através do artefato Anexo II - Visão e Principais Envolvidos.

3. Execução do Projeto de *Software* com Scrum

Para facilitar a execução das atividades, deve-se utilizar o Guia Scrum. Nesta seção, são brevemente listados os itens e procedimentos inerentes a este *framework*.

3.1 *Backlog* do produto (*product backlog*)

Representa a especificação dos requisitos do *software*, ou seja, as funcionalidades e características do sistema. O envolvimento nesta tarefa é de toda a equipe Scrum, porém o responsável por criar e manter este artefato é o *Product Owner*.

3.2 Planejamento da *sprint* (*sprint planning*)

Reunião realizada com todos os envolvidos, onde o *Product Owner* prioriza os itens a serem desenvolvidos e a equipe estima o esforço para o trabalho.

3.3 Lista de tarefas (*sprint backlog*)

A partir do *backlog* do produto e da priorização do planejamento, a equipe de desenvolvimento define os itens (tarefas) selecionados para a próxima *sprint* e o tempo necessário para o desenvolvimento.

3.4 Codificação

A organização do código-fonte facilita os processos de desenvolvimento, correção de

bugs, atividades de validação e manutenção. O uso de um padrão de codificação também aumenta a produtividade do projeto, uma vez que facilita a comunicação dentro da equipe de desenvolvimento. É importante que os integrantes da equipe sejam sempre fieis aos acordos referentes às nomenclaturas e arquivos, codificação, etc.

Para o desenvolvimento e manutenção de sistemas no âmbito do IFFar, deverão ser seguidos os padrões especificados no Guia do Codificador.

3.5 Versionamento

Todos os sistemas desenvolvidos devem ser versionados com controle de versão no *GitLab*.

4. Implantação e Manutenção de *Software*

Nesta seção, são brevemente listados os procedimentos para implantação e manutenção de *software*.

4.1 Homologação

Após executadas as etapas anteriores, é realizada a entrega do protótipo: o sistema é repassado ao *Product Owner* para a realização de testes exaustivos e validações necessárias. Entende-se por teste exaustivo o processo de testar uma funcionalidade por completo, levando em consideração todas as combinações possíveis de entradas, cenários de uso, bem como situações e entradas aleatórias.

Em caso de serem encontrados *bugs* ou inconsistências, o mesmo é devolvido aos desenvolvedores para as devidas correções. Caso contrário, o sistema será considerado homologado, estando, assim, apto para a sua implantação. Esta etapa pode ser iterativa (repetir-se algumas vezes).

4.2 Implantação

Para iniciar a implantação do sistema, o gerente de projetos ou um representante técnico da equipe de desenvolvimento deve enviar os arquivos para o servidor de produção, solicitando ao administrador da rede os acessos e permissões necessárias. Se necessário, também deve

solicitar a criação do domínio para o sistema. Após a publicação dos arquivos, o gerente de projetos deve acessar o domínio da aplicação e verificar a efetiva implantação.

Os *softwares* desenvolvidos serão propriedade intelectual do IFFar e não devem ser repassados a terceiros sem a prévia autorização oficial da DTI.

4.3 Treinamento

Os treinamentos dos usuários são responsabilidade do(s) *Product Owner(s)*, que deve(m) utilizar sempre o servidor de testes (homologação). O responsável por ministrar o treinamento deve consultar a equipe técnica para se certificar de que o servidor de testes está atualizado com a versão do sistema em produção.

4.4 Reportar erros e sugestões

Erros e inconsistências a serem aprimorados no *software* devem ser reportados pelo *Product Owner* via sistema de suporte de TI do IFFar (GLPI). Após receber a notificação, a demanda será encaminhada para a equipe do projeto, que irá definir as correções ou procedimentos necessários. A solicitação pode ser aceita ou rejeitada e o requerente receberá uma notificação com estas informações.

4.5 Fluxo de trabalho

Em síntese, o desenvolvimento de *software* pode ser sintetizado em oito passos principais, conforme elencado a seguir.

1. Memorando Eletrônico com a solicitação de *software* - Anexo I: Formulário de Solicitação de *Software*.
2. Análise e deferimento pela DTI.
3. Reunião com interessados.
    a. Anexo II: Visão e principais envolvidos.
    b. Anexo III: Requisitos de *software* (*backlog* do Produto).
4. Abertura do projeto no *GitLab*.
5. Reunião de planejamento da *sprint*.
    a. Elaboração do quadro de tarefas - *GitLab*.
6. *Sprint* + reuniões periódicas (até o final do projeto).

7. Reunião de revisão da *sprint*.
    a. Retrospectiva da *sprint*.
8. Entrega do incremento utilizável do produto.

ANEXO I

INSTITUTO FEDERAL
Farroupilha

Diretoria de Tecnologia da Informação

Formulário de Solicitação de *Software*

☐ *Software novo* ☐ *Software existente (Módulo)*

Dados do Requisitante

Unidade (Reitoria ou nome do *campus*):	
Pró-Reitoria/ Diretoria ou Coordenação:	
Nome:	
Cargo/ Função:	
E-mail:	
Telefone/ Ramal:	

Dados do Software

Características do *software*: (1)	
Base Legal (Se houver): (2)	
Prazo Legal (Se houver): (3)	
Público-alvo: (4)	

Dados da solicitação

Finalidade/Utilidade: (5)	
Justificativa: (6)	

Termo de compromisso

No prazo de 60 dias a Diretoria de TI (DTI) – Reitoria irá avaliar a procedência da solicitação. Em caso de aprovação, o solicitante tem a responsabilidade de fornecer as informações necessárias para a elaboração da documentação do *software* a ser desenvolvido ou mantido e comparecer a reuniões previamente agendadas. Os *softwares* desenvolvidos não devem ser repassados a terceiros sem a prévia autorização da DTI.

Declaro estar ciente dos termos de compromisso

Data de solicitação: ____/____/_____

Requisitante: _____

Responsável setor requisitante: _____

(1) Descreva, resumidamente, as características do software desejado. (2) Indique a Base Legal que criou a necessidade do software. (3) Indique o prazo estabelecido legalmente para o início da utilização do software. (4) Especifique quem utilizará o software. (5) Indique o objetivo específico a ser alcançado com o software. (6) Escreva os motivos para o desenvolvimento e utilização do software desejado.

ANEXO II

INSTITUTO FEDERAL Farroupilha | Diretoria de Tecnologia da Informação

Visão e Principais Envolvidos

1. Objetivo do sistema

[Descrever, de forma resumida, os objetivos a serem atingidos pelo produto que será desenvolvido.]

2. Necessidades e restrições (Regras de Negócio)

[Refere-se às diretrizes que definem ou restringem ações, mostrando como as operações devem ser conduzidas e se há algum limite nessa aplicação. Essas regras são importantes para que a organização tenha uma visão clara do que deve ser feito, como e por qual razão.]

ID	Descrição	Prioridade
RN-01		
RN-02		

- ID – Informar um número para ordenar a regra de negócio, que será utilizado como identificador único do mesmo ao longo do projeto.
- Descrição – Descrever o requisito em detalhes suficientes para que o mesmo possa ser devidamente implementado. Se existirem, incluir referências a documentos e outras fontes externas de informação.
- Prioridade – MoSCoW (Deve ter/ Deveria ter/Poderia ter/Não terá)

3. Estrutura da Equipe Scrum

Dono(s) do produto (*Product Owner(s)*)	[preencher o nome / e-mail / telefone-ramal]
Gerente de projetos (*Scrum Master*)	[preencher o nome / e-mail / telefone-ramal]
Equipe de desenvolvimento (*Development Team*)	[preencher o nome / e-mail / telefone-ramal]

4. Cronograma Inicial

[Cronograma inicial, proposto pelo solicitante de forma macro.]

5. Perfil do Sistema

☐ **Acadêmico** ☐ **Administrativo** ☐ **Específico**

[Acadêmico: sistema que será utilizado pelos alunos ou servidores da Instituição para atividades acadêmicas. Administrativo: sistema que será utilizado para tarefas administrativas da Instituição. Específico: sistema que será desenvolvido para atender uma demanda específica de uma pró-reitora, diretoria ou coordenação.]

ANEXO III

INSTITUTO FEDERAL
Farroupilha | Diretoria de Tecnologia da Informação

Requisitos de *Software* – *Backlog* do Produto

1. Escopo

[Breve descrição do software ao qual se aplica esta especificação de requisitos.]

2. Requisitos Funcionais

[Representam as funções do sistema (condição ou capacidade) que o sistema precisa atender ou ter, sob o ponto de vista do usuário.]

ID	Descrição	Prioridade	Estimativa	Valor
RQF-01				
RQF-02				
RQF-03				
RQF-04				
RQF-05				
RQF-n				

3. Requisitos Não Funcionais

[Descreve como o sistema deve realizar as funcionalidades dos requisitos funcionais.]

ID	Descrição	Prioridade	Estimativa	Valor
RQNF-01				
RQNF-02				
RQNF-03				
RQNF-n				

- ID – Informar um número para ordenar o requisito, que será utilizado como identificador único do mesmo ao longo do projeto.
- Descrição – Descrever o requisito em detalhes suficientes para que o mesmo possa ser devidamente implementado. Se existirem, incluir referências a documentos e outras fontes externas de informação.
- Prioridade - MoSCoW (Deve ter/ Deveria ter/ Poderia ter/ Não terá).
- Estimativa – Atribuir uma estimativa, em horas ou dias, considerando os recursos existentes e a produtividade da equipe, para concluir um ou mais requisitos.
- Valor – Atribuir um valor, de 0 (zero) a 100 (cem), de modo que os itens de alto valor devem aparecer no topo da lista e os itens de menor valor devem aparecer ao final da lista.