



Ministério da Educação
Secretaria de Educação Profissional e Tecnológica
Instituto Federal de Educação, Ciência e Tecnologia do Tocantins
Reitoria

PROCESSO DE SOFTWARE

HISTÓRICO DE VERSÕES

Data	Versão	Descrição
29/02/2023	1	Elaboração do processo de software.

1. INTRODUÇÃO

Processo de *software* é um conjunto de atividades relacionadas que levam à produção de um produto de *software* (SOMMERVILLE, 2019). Esse processo aborda um conjunto de procedimentos estratégicos e organizados com o propósito de arquitetar, adquirir/desenvolver, testar e implantar o *software*.

O processo visa a elaboração de um produto de qualidade, que atenda às necessidades do cliente e esteja dentro do prazo estabelecido. Ele envolve desde a análise e especificação de requisitos até a manutenção do software, passando pelo projeto, implementação, testes e documentação. Como exemplos de processos de software incluem o modelo em cascata, modelo incremental, modelo ágil, entre outros.

Dentro do contexto apresentado, este documento tem como objetivo estabelecer as características do processo de *software* utilizado pelo Instituto Federal de Educação, Ciência e Tecnologia do Tocantins (IFTO). Para isso, este documento está estruturado em uma breve introdução, definições, processo de software, papéis e responsabilidades, matriz RACI, indicador de desempenho, processos relacionados, práticas recomendadas e referências.

1.1. Escopo

O escopo do processo de software refere-se ao conjunto de atividades, tarefas e objetivos definidos para desenvolver, manter ou gerenciar um produto de software. Esse escopo abrange todas as fases do ciclo de vida do desenvolvimento de software, desde a concepção até a entrega e manutenção do produto final.

1.2. Objetivos

O objetivo do processo de software é fornecer uma abordagem sistemática e disciplinada para desenvolver, manter e gerenciar software de forma eficiente. Para que isso seja possível são definidos os seguintes objetivos específicos:

- a) garantir que o produto final atenda aos requisitos do cliente, seja funcionalmente robusto, seguro, confiável, fácil de usar e mantenha um alto nível de qualidade geral;
- b) ajudar as equipes a gerenciar recursos, cronogramas e custos de forma eficiente, garantindo que os projetos sejam concluídos dentro do prazo e do orçamento previstos;
- c) reduzir a probabilidade de problemas e falhas durante o desenvolvimento e a operação do software;
- d) promove uma comunicação clara e colaboração eficaz entre membros da equipe, partes interessadas e clientes, ajudando a garantir que todos tenham uma compreensão comum dos objetivos, requisitos e progresso do projeto;
- e) garantir a satisfação do cliente e fortalecer relacionamentos; e
- f) incentiva a adoção de melhores práticas, tecnologias emergentes e abordagens inovadoras para o desenvolvimento de software, visando melhorar continuamente a qualidade, produtividade e competitividade da equipe de desenvolvimento.

1.3. Abrangência

Este processo abrange o escopo ou amplitude das atividades que são realizadas ao longo do ciclo de vida do software, desde a concepção, desenvolvimento, manutenção, gestão até a descontinuação. Isso inclui desde a definição dos requisitos do software, passando pela elaboração do projeto, implementação, testes, documentação e manutenção.

1.4. Benefícios esperados

O processo de *software* acarreta os seguintes benefícios:

- a) melhoria da qualidade de software;
- b) redução dos custos durante as fases de desenvolvimento e manutenção;
- c) reduz o retrabalho e aumenta a produtividade geral do projeto;
- d) melhoria na comunicação e colaboração entre a equipe de desenvolvimento, clientes e usuários finais;
- e) redução de riscos;
- f) aumento da capacidade de entrega dentro do prazo;
- g) permite que a equipe aprenda com experiências passadas e faça ajustes para aumentar a eficácia e a eficiência ao longo do tempo;
- h) permite uma melhor previsão de custos, prazos e recursos necessários para concluir um projeto, o que reduz o risco de atrasos e estouro de orçamento;
- i) ajuda a equipe a lidar proativamente com potenciais problemas que possam surgir durante o desenvolvimento do software;

- j) facilita a manutenção contínua do produto, permitindo que as equipes realizem atualizações, correções de bugs e adaptações de forma organizada e eficiente;
- k) promove uma comunicação mais clara e eficaz entre os membros da equipe, partes interessadas e clientes, garantindo que todos estejam alinhados quanto aos objetivos e progresso do projeto;
- l) contribui para a satisfação do cliente e fortalece relacionamentos; e
- m) garante a conformidade com regulamentações e padrões específicos, reduzindo assim o risco legal e financeiro para a organização.

2. DEFINIÇÕES

Para fins de compreensão dos termos utilizados neste processo serão utilizados os seguintes conceitos e definições:

- a) ciclo de vida do software: estrutura contendo processos, atividades e tarefas envolvidas no desenvolvimento, operação e manutenção de um produto de *software*, abrangendo a vida do sistema, desde a definição de seus requisitos até o término de seu uso;
- b) escopo: refere-se ao somatório dos produtos e das metas contidos na proposta do projeto, bem como às principais atividades necessárias para garantir a entrega desses produtos e o alcance dessas metas;
- c) processo: conjunto de passos realizados para alcançar um certo propósito. Conjunto de atividades inter-relacionadas, que transforma entradas em saídas;
- d) software: sequência de instruções escritas para serem interpretadas por um computador com o objetivo de executar tarefas específicas. Todo programa rodado em computadores, celulares ou outros dispositivos que permite a execução de suas funções.

3. PROCESSO DE SOFTWARE

Processo de *software* consiste em um conjunto de atividades que são realizadas durante a produção de um *software* (SOMMERVILLE, 2019). O objetivo deste processo é garantir a qualidade do software produzido, desde a definição dos requisitos até a entrega do produto final, passando pelas fases de análise, projeto, implementação e testes.

Este processo tem como propósito aumentar a eficiência e eficácia do desenvolvimento, reduzindo custos e prazos e garantindo a satisfação do cliente. Pressman (2021) afirma que o processo abrange um conjunto de três elementos fundamentais: métodos, ferramentas e procedimentos para projetar, construir e manter o *software*.

Para que o software seja desenvolvido com eficiência e qualidade a norma ABNT 12207 define o ciclo de vida de software contendo fases, processos, atividades e tarefas que são aplicáveis durante a aquisição, fornecimento, desenvolvimento, operação, manutenção ou desativação de sistemas, produtos e serviços de *software* (ABNT, 2021). A figura 1 apresenta o ciclo de vida do *software* utilizado pelo IFTO.

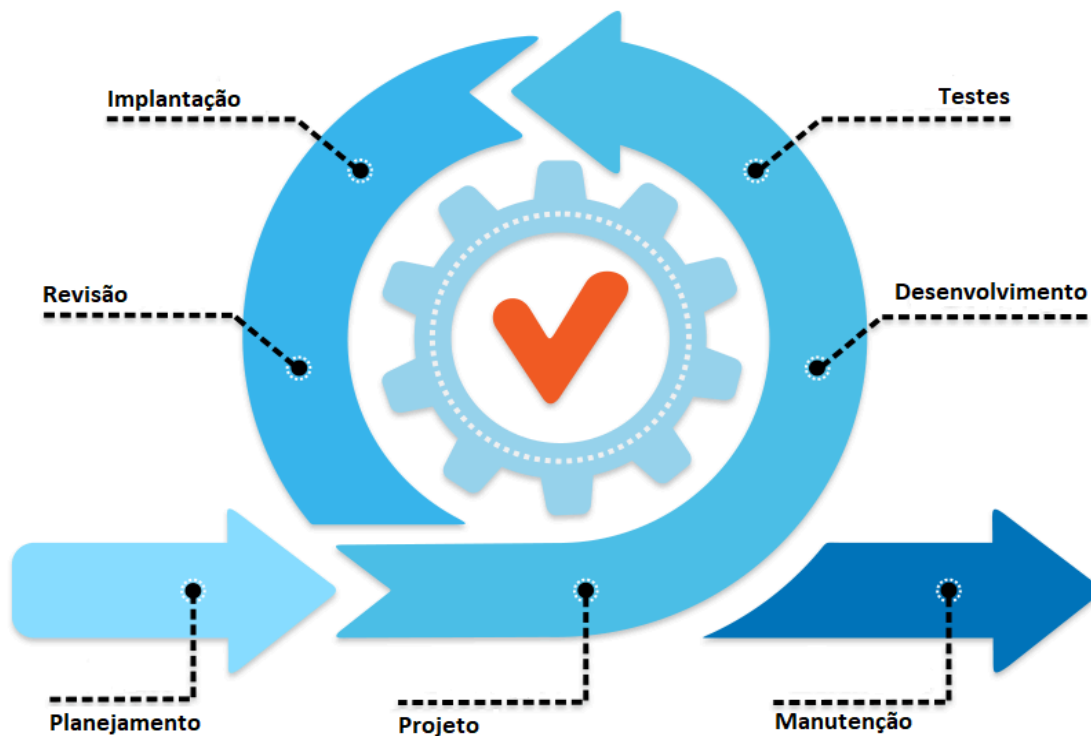


Figura 1 - Ciclo de vida do software

O ciclo de vida do *software* definido pelo IFTO segue as práticas de desenvolvimento ágil. O modelo de processo de *software* é iterativo e incremental. Ao invés de entregar o sistema como uma única entrega, o desenvolvimento e a entrega são separados em incrementos, sendo que cada incremento fornece parte da funcionalidade solicitada. Este processo promove a participação de representante da área de negócio como integrante da equipe de desenvolvimento ou aquisição de software, desde a concepção até a aceitação final. Dentro desta concepção, o ciclo de vida do *software* é composto por 7 (sete) fases.

3.1. Planejamento

Fase que consiste na definição dos requisitos que o sistema deverá atender, ou seja define as funcionalidades e as restrições sobre a operação e desenvolvimento do *software*. Esta fase inicia-se com o recebimento do documento de oficialização de demanda. Nesta fase podem ser realizadas as seguintes atividades:

- a) promover a participação de representante da área de negócio como integrante da equipe de desenvolvimento ou aquisição de software, desde a concepção até a aceitação final;
- b) analisar requisitos funcionais de *software*;
- c) promover a identificação precoce de requisitos de segurança da informação e a gestão permanente desses requisitos durante todo o ciclo de vida do software;
- d) promover a identificação precoce de requisitos de interoperabilidade e a gestão permanente desses requisitos durante todo o ciclo de vida do software;
- e) promover a identificação precoce de acessibilidade e de usabilidade, bem como a gestão permanente desses requisitos durante todo o ciclo de vida do software;
- f) especificar requisitos de usuário, recursos e as restrições;

- g) avaliar as soluções existentes no mercado antes de decidir pelo desenvolvimento de software;
- h) realizar estudos para identificar e mitigar risco de dependência tecnológica, com vista a viabilizar a substituição de fabricante/fornecedor quanto tecnicamente viável e economicamente vantajoso em caso de opção por contratação de terceiros; e
- i) assegurar os direitos autorais, de propriedade e de uso relativamente ao software que desenvolve por meio de contratação.

3.2. Projeto

Fase que traduz os requisitos em uma descrição (especificações) de todos os componentes necessários para o *software*. Nesta fase é definido o funcionamento interno do *software* para que todas as especificações sejam atendidas. A fase envolve a própria estrutura do *software*, além de questões referentes a linguagem de programação, banco de dados, *design* de interface, entre outras. Nesta fase podem ser realizadas as seguintes atividades:

- a) entender e dimensionar a demanda de software;
- b) definir o escopo do produto;
- c) elaborar o modelo de negócio;
- d) levantar requisitos funcionais e não funcionais; e
- e) projetar a estrutura do software, banco de dados e design de interface.

3.3. Desenvolvimento

Fase responsável por escolher a estratégia (interno, produção colaborativa ou contratação) mais adequada para o desenvolvimento e/ou manutenção do *software* (evolutiva, corretiva e adaptativa). Após a escolha a estratégia de desenvolvimento, será avaliada qual a melhor metodologia de desenvolvimento de *software* e qual a infraestrutura e sustentação necessárias para que o *software* funcione corretamente no ambiente de produção. Nesta fase podem ser realizadas as seguintes atividades:

- a) escolher a estratégia para desenvolvimento e/ou manutenção do software;
- b) avaliar a metodologia que será utilizada para o desenvolvimento do software;
- c) avaliar infraestrutura para ambiente de desenvolvimento e produção do software; e
- d) utilizar prioritariamente arquiteturas de software que promovem o desacoplamento de soluções, sistemas e componentes, inclusive nos casos de software adquirido e desenvolvimento realizado mediante contratação, com vistas a facilitar a realização de manutenções e otimizar custos.

3.4. Testes

Fase responsável por verificar a presença de erros e comportamento adequado a nível das funções e módulos básicos do *software*. Nesta fase são definidos o escopo, etapas, recursos (ferramentas, hardware etc), tipos de testes e as demais atividades necessárias à execução, controle e acompanhamento dos testes de *software*. A fase pode ser composta pelas seguintes atividades:

- a) definir tipos de testes a serem realizados;
- b) verificar erros e comportamento adequados para o funcionamento do software; e
- c) acompanhar a execução dos testes de software.

3.5. Implantação

Fase responsável por disponibilizar o *software* para uso por parte do usuário final. Nesta fase podem ser realizadas as seguintes atividades:

- a) disponibilizar software em ambiente de produção;
- b) elaborar manuais de uso do software;
- c) importar dados; e
- d) realizar treinamento de usuários.

3.6. Revisão

Fase responsável por validar se o *software* atende a demanda apresentada pelo *product owner*, ou seja mostra que o mesmo está em conformidade com a sua especificação e que atende aos requisitos do cliente. Nesta fase podem ser realizadas as seguintes atividades:

- a) avaliar se o *software* adquirido ou desenvolvido possui a qualidade desejada;
- b) avaliar se todos os requisitos funcionais e não funcionais foram implementados;
- c) avaliar por meio de mensurações, indicadores e metas, a qualidade do software desenvolvido ou adquirido; e
- d) avaliar periodicamente o desempenho e a conformidade do processo de software e promover eventuais ajustes necessários.

3.7. Manutenção

Fase responsável por realizar as adequações necessárias ao *software* para atendimento de novas demandas apresentadas pelo *product owner*. Esta fase é uma garantia de que o bom funcionamento do *software* será mantido. É uma fase que pode ser realizada de duas formas: corretiva e evolutiva. Nesta fase podem ser realizadas as seguintes atividades:

- a) corrigir erros apresentados pelo software; e
- b) desenvolver novas funcionalidades.

O processo de *software* apresentado na figura 1 é executado por uma equipe multidisciplinar, formada por servidores da área de TI e negócios. A tabela 1 faz o detalhamento da entrada, fases e saídas deste processo.

Tabela 1 - Processo de Software

Processo de Software	
Descrição	Processo responsável pela gestão do processo de <i>software</i> .
Entrada	- Documento de oficialização de demanda.
Fases	<ol style="list-style-type: none"> 1. Planejamento. 2. Projeto. 3. Desenvolvimento. 4. Testes. 5. Implantação. 6. Revisão. 7. Manutenção.
Saídas	<ul style="list-style-type: none"> - Software. - Documentação do <i>software</i>.

Fonte: Diretoria de Tecnologia da Informação

O processo de *software* executado no IFTO é iterativo uma vez que a equipe refina e melhora o *software* a cada entrega. É incremental, pois segundo Sommerville (2019) intercala as atividades de especificação, desenvolvimento e validação. O *software* é desenvolvido como uma série de versões (incrementos), de maneira que cada versão colocada em produção, adiciona funcionalidade à anterior.

Assim, frequentemente os incrementos iniciais incluem a funcionalidade mais importante ou mais urgente. Isso significa que o solicitante pode avaliar o *software* em um estágio relativamente inicial do desenvolvimento para ver se ele oferece o que foi requisitado. Em caso negativo, só o incremento que estiver em desenvolvimento no momento precisará ser alterado e, possivelmente, nova funcionalidade deverá ser definida para incrementos posteriores.

4. PAPÉIS E RESPONSABILIDADES

Um papel é um conjunto de responsabilidades, atividades e autoridades definidas em um processo e atribuídas a uma pessoa, equipe ou função. Os papéis e responsabilidades definidos no processo de *software* seguem as definições apresentadas na metodologia SCRUM. São eles:

4.1. Product Owner

Este papel é responsável por decidir o que será feito e em qual a ordem de prioridade. Neste processo este papel é a autoridade que define a priorização do desenvolvimento/manutenção de *software*. Ele colabora ativamente com a equipe de desenvolvimento do *software*. No IFTO é representado pelo Reitor, Pró-Reitorias e Diretorias Sistêmicas, tendo as seguintes responsabilidades:

- a) aprovar as diretrizes estratégicas que norteiam a gestão de *software*;
- b) garantir os recursos necessários para estabelecer, implementar, operar e manter o processo de *software*;
- c) aprovar a estratégia de gestão de *software*;
- d) estabelecer objetivos e prioridades para o desenvolvimento e atualização de *software*; e
- e) aprovar os documentos de processo de *software*.

4.2. Scrum Master

Este papel é responsável por ajudar todos os envolvidos a entender a demanda apresentada pelo *product owner* e também protege a equipe contra interferências externas. Neste processo este papel remove impedimentos que possam atrapalhar a produtividade. A pessoa que exerce este papel sempre estará capacitado para gerir o processo de *software*. No IFTO é representado pela Diretoria de Tecnologia da Informação, tendo as seguintes responsabilidades:

- a) gerir o processo de *software*;
- b) negociar com os clientes e demais partes envolvidas no processo de *software*;
- c) definir os recursos materiais e humanos e os treinamentos necessários para o desenvolvimento/atualização do *software*;
- d) negociar com as áreas de negócio a cessão de recursos humanos para compor a equipe de desenvolvimento/atualização de *software*;
- e) negociar prazos e ações conjuntas com as áreas de negócio;
- f) participar de reunião de análise de viabilidade técnica e mudanças no *software*;
- g) prever aquisições de *software*;
- h) prever capacitações de *software*; e
- i) gerenciar o orçamento do processo de *software*.

4.3. Scrum Team

Este papel é responsável pelo desenvolvimento e entrega do *software*. No IFTO é representado pela equipe que compõem a Coordenação de Sistemas de Informação e representante da área de negócio, tendo as seguintes responsabilidades:

- a) planejar o desenvolvimento do *software*;
- b) elaborar o projeto de *software*;

- c) codificar os requisitos acordados entre o *Product Owner* e *Scrum Master*;
- d) testar *software*;
- e) implantar *software*;
- f) revisar *software*; e
- g) realizar manutenção em *software*.

5. MATRIZ RACI

A matriz RACI apresentada na tabela 2 é um método utilizado para definir com clareza os papéis e responsabilidades de cada ator na execução da atividade relacionada a este processo. A sigla RACI significa, em inglês: *responsible, accountable, consulted e informed*.

- a) **responsible (responsável):** pessoa, função ou unidade organizacional responsável pela execução de uma atividade no âmbito de um processo; Representa quem irá, de fato executar a tarefa. Deve haver ao menos um por tarefa;
- b) **accountable (responsabilizado):** dono da atividade, deverá fornecer os meios para que a atividade possa ser executada, e será responsabilizado caso a atividade não alcance os seus objetivos; cada atividade só pode possuir um *accountable*; Define quem será responsável pelo sucesso da atividade. Fica encarregado de verificar se a atividade foi realizada com sucesso e dentro do prazo. Deve haver um, e apenas um, por atividade;
- c) **consulted (consultado):** pessoa que deve ser consultada durante a execução da atividade; As informações levantadas junto a essas pessoas tornam-se entradas para a execução da atividade; Geralmente exercem papel de conselho na tomada de decisões;
- d) **informed (informado):** pessoa que será informada acerca do progresso da execução da atividade.

Tabela 2 - Matriz de responsabilidades

Fases	PO	SM	ST
Planejamento	A	R	C/I
Projeto	A	R	C/I
Desenvolvimento	A	C/I	R
Testes	A	C/I	R
Implantação	A	C/I	R
Revisão	A	R	C/I
Manutenção	A	C/I	R

Fonte: Diretoria de Tecnologia da Informação

Legenda:

PO: *Product Owner* (Reitor, Pró-Reitorias e Diretorias Sistêmicas).

SM: *Scrum Master* (Diretoria de Tecnologia da Informação).

ST: Scrum Team (Equipe da Coordenação de Sistemas de Informação e Representante da área de negócio).

6. INDICADOR DE DESEMPENHO

O processo de *software* deve ser monitorado periodicamente através de indicador de desempenho. Esse indicador de desempenho tem como objetivo acompanhar a eficácia do processo, identificando tendências, falhas e oportunidades de correções, promovendo sempre a melhoria contínua. A tabela 3 apresenta a métrica operacional que foi definida para o processo.

Tabela 3 - Indicador de desempenho

Indicador	Quantidade de <i>software</i> adquirido/desenvolvido a partir do processo de <i>software</i> .			
Descrição	Índice de desempenho do processo de <i>software</i> .			
Objetivo	Medir o desempenho do processo de <i>software</i> .			
Periodicidade	Anual.			
Fonte	GITLAB.			
Fórmula	Somatório de <i>software</i> adquirido/desenvolvido com base no processo de <i>software</i> .			
Meta	Aumentar a quantidade de <i>software</i> adquirido/desenvolvido a cada ano com base no processo de <i>software</i> .			
2020	2021	2022	2023	2024
1	2	3	4	5

Fonte: PMI (2021)

7. PROCESSOS RELACIONADOS

Para que o processo de software seja executado de forma eficiente ele está interligado à outros processos que compõem o Sistema de Gestão de Segurança da Informação (SGSI-IFTO). A figura 2 apresenta estes processos.



Figura 2 - Processos que compõem o SGSI-IFTO

8. PRÁTICAS RECOMENDADAS

Para que o processo de *software* seja executado de forma satisfatória são recomendadas as seguintes práticas:

1. O processo de *software* utilizado pelo IFTO deve promover a participação de representante da área de negócio como integrante da equipe de desenvolvimento ou aquisição de *software*, desde sua concepção até a aceitação final.
2. O processo de *software* utilizado pelo IFTO deve promover desde a concepção a identificação de requisitos de segurança da informação, bem como a gestão permanente desses requisitos durante todo o ciclo de vida do *software*.
3. O processo de *software* utilizado pelo IFTO deve promover desde a concepção a identificação de requisitos de acessibilidade e de usabilidade, bem como a gestão permanente desses requisitos durante todo o ciclo de vida do *software*.
4. O IFTO deve assegurar os seus direitos autorais, de propriedade e de uso relativamente ao *software* que desenvolve por meio de contratação.
5. A instituição deve ter uma equipe com habilidades e competências específicas para gerir o processo de *software*.
6. Para o desenvolvimento de *software* a instituição deve avaliar se há soluções já existentes na rede federal de ensino.
7. Ao planejar contratações de soluções envolvendo *software* deve-se sempre realizar estudos para identificação e mitigação de risco de dependência tecnológica, com vistas a

viabilizar a substituição de fabricante/fornecedor quando tecnicamente viável e economicamente vantajoso.

8. A instituição deve utilizar prioritariamente arquiteturas de *software* que promovem o desacoplamento de soluções, sistemas e componentes, inclusive nos casos de *software* adquirido e desenvolvimento realizado mediante contratação, com vistas a facilitar a realização de manutenções e otimizar custos.

9. O processo de *software* deve promover a participação de representante da área de negócio como integrante da equipe de desenvolvimento ou aquisição de *software*, desde sua concepção até a aceitação final.

10. O processo de *software* deve promover a identificação precoce de requisitos de segurança da informação e a gestão permanente desses requisitos durante todo o ciclo de vida do *software*.

11. O processo de *software* deve promover a identificação precoce de requisitos de interoperabilidade e a gestão permanente desses requisitos durante todo o ciclo de vida do *software*.

12. O processo de *software* da organização deve promover a identificação precoce de requisitos de acessibilidade e de usabilidade, bem como a gestão permanente desses requisitos durante todo o ciclo de vida do *software*.

13. A instituição deve assegurar os seus direitos autorais, de propriedade e de uso relativamente ao *software* que desenvolve por meio de contratação.

14. A instituição deve avaliar por meio de mensurações, indicadores e metas, a qualidade do *software* desenvolvido ou adquirido.

15. A instituição deve definir responsabilidades para a execução do processo de *software*.

16. A instituição deve avaliar periodicamente o desempenho e a conformidade do processo de *software* e promover eventuais ajustes necessários.

9. FERRAMENTAS

Para executar o processo de *software* são utilizadas as seguintes ferramentas:

- a) Gestão de Projetos: GITLAB;
- b) Codificação: *frameworks* JSF, InfraPHP, django, PyCharm, Visual Studio Code (Ionic) e DBeaver; e
- c) Documentação: Google *Workspace for Education Plus*.

10. ARTEFATOS

A tabela 4 apresenta os artefatos utilizados no processo de *software*.

Tabela 4 - Artefatos do processo de software

Fase	Artefato	Atores Envolvidos
Planejamento	- Documento de oficialização de demanda. - Lista de requisitos.	Product Owner Scrum Master Scrum Team
Projeto	- Arquitetura do software.	Scrum Master Scrum Team
Desenvolvimento	- Plano de gerenciamento do projeto.	Scrum Team
Testes	- Plano de Testes.	Scrum Team
Implantação	- Plano de implantação. - Manuais do software.	Scrum Team
Revisão	- Relatório de revisão.	Scrum Master Scrum Team
Manutenção	- Solicitação de mudança.	Scrum Team

11. REFERÊNCIAS

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **ABNT NBR ISO/IEC/IEEE 12207. Engenharia de sistemas e software: processos de ciclo de vida de software.** ABNT, 2021.

PRESSMAN, Roger S.; MAXIM, Bruce R. **Engenharia de software: uma abordagem profissional.** 9ª ed. Mc Graw Hill Education. Bookman, 2021.

SOMMERVILLE, Ian. **Engenharia de Software.** 10ª ed. São Paulo: Pearson Universidades, 2019.

ANEXO I

Gestão de Software

O monitoramento da execução do processo de *software* é realizado através da ferramenta **GITLAB** disponível no endereço eletrônico: <https://gitlab.ifto.edu.br/dashboard/projects>

The screenshot shows the GitLab dashboard interface. At the top, there is a navigation bar with the GitLab logo and a search bar. Below the navigation bar, the main heading is "Projects". Underneath, there are tabs for "Your projects" (12), "Starred projects" (0), "Explore projects", and "Explore topics". A search filter "Filter by name..." is visible. The main content area displays a list of projects, each with a colored icon, a title, a description, and a "Maintainer" badge. The projects listed are:

- IFTO Sistemas / capacitacao-de-servidores-de-ti**: Projeto para planejar, desenvolver e avaliar as competências técnicas necessárias para viabilizar os objetivos do ...
- IFTO Sistemas / comunicacao-e-transparencia**
- IFTO Sistemas / conectividade-ifto**: Conectividade IFTO
- IFTO Sistemas / conformidade-em-ti**
- IFTO Sistemas / desenvolvimento-de-solucoes-de-ti**: Projeto com a finalidade de promover, implantar, aprimorar e manter soluções de software por meio de desenv...



Documento assinado eletronicamente por **Fabiana Ferreira Cardoso, Coordenadora**, em 01/03/2024, às 09:12, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Daniel Felix de Souza, Servidor**, em 01/03/2024, às 09:34, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site http://sei.ifto.edu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **2213646** e o código CRC **F91DD96C**.

Avenida Joaquim Teotônio Segurado, Quadra 202 Sul, ACSU-SE 20, Conjunto 1, Lote 8 - Plano Diretor Sul — CEP 77020-450 Palmas/TO — (63) 3229-2200
portal.ifto.edu.br — reitoria@ifto.edu.br