



PODER JUDICIÁRIO
JUSTIÇA DO TRABALHO

TRIBUNAL REGIONAL DO TRABALHO DA DÉCIMA NONA REGIÃO

ATO Nº 91/2015/GP/TRT 19ª, DE 1º DE JUNHO DE 2015

O DESEMBARGADOR PRESIDENTE DO TRIBUNAL REGIONAL DO TRABALHO DA DÉCIMA NONA REGIÃO, no uso de suas atribuições legais e

Considerando que os sistemas de informação deverão atender a padrões de desenvolvimento, suporte operacional, segurança da informação, gestão documental, interoperabilidade e outros que venham a ser recomendados pelo CNJ;

Considerando a necessidade de formalização e padronização no desenvolvimento de softwares adotados pela Secretaria de Tecnologia da Informação e Comunicações no âmbito do TRT da 19ª Região;

RESOLVE:

Art. 1º. **Aprovar** o Processo de Desenvolvimento de Software, em anexo.

Art. 2º. O Processo de Desenvolvimento de Software utiliza como base práticas de gerenciamento de projeto do PMBoK e metodologia ágil SCRUM, adaptadas às experiências, necessidades e cultura organizacional do Tribunal Regional do Trabalho da 19ª Região.

Art. 3º. O acompanhamento e o controle da execução do Processo de Desenvolvimento de Software ficarão sob a responsabilidade da Secretaria de Tecnologia da Informação e Comunicações.

Dê-se ciência, cumpra-se e
Publique-se.

• **Original Assinado**
PEDRO INÁCIO DA SILVA
Desembargador Presidente

Publicado no DEJT e no BI nº 06, ambos de
03/06/2015

Tribunal Regional do Trabalho da 19ª Região

Processo de Desenvolvimento de Software

1 - Apresentação

O processo Desenvolvimento de Software tem como principal foco a 1ª fase do Ciclo de Vida de Software, que envolve a construção do produto, partindo da aprovação da demanda até sua implementação em produção.

Os principais objetivos do processo são:

- ser aderente aos padrões de desenvolvimento de software em prática no mercado;
- prover clara definição de atividades, em termos de sequência e responsáveis;
- padronizar as etapas e devidas entregas, de forma a estabelecer uma linguagem uniforme, de fácil entendimento e clara documentação;
- promover a entrega do software com qualidade e agilidade;
- garantir que o escopo da demanda seja plenamente compreendido e atendido.

O Processo de Desenvolvimento de Software do TRT da 19ª Região usa como base práticas de gerenciamento de projeto do PMBoK e metodologia ágil SCRUM, adaptadas às experiências, necessidades e cultura organizacional do Tribunal.

2 - Escopo

Este processo engloba as etapas necessárias à entrega de um novo produto de software (sistema completo ou funcionalidade) que requer esforços de desenvolvimento. Tais quais:

- Concepção
- Especificação
- Projeto Arquitetônico
- Construção
- Testes
- Homologação
- Implantação

Não engloba, portanto, atividades e controles relacionados a gestão de portfólio, gerenciamento de mudança, gerenciamento de configuração, entre outros.

3 - Papéis e responsabilidades

Os papéis agrupam atividades e responsabilidades, dentre as especificadas no fluxo do processo desenvolvimento, que devem ser executadas por um colaborador, seja ele servidor da área de TI, servidor de áreas de negócio, magistrado ou funcionário de empresa terceirizada.

A atribuição de papéis a pessoas não implica nenhuma estrutura organizacional.

Uma mesma pessoa poderá assumir mais de um papel.

3.1 - Product Owner (PO)

Responsabilidades:

- representar a área de negócio demandante do produto de software em todas as fases do processo;
- entender e coletar requisitos juntos os usuários finais;
- criar, junto com o Analista de Requisitos, as histórias de usuário e esclarecer dúvidas sobre os requisitos do software;
- criar o backlog do produto;
- detalhar, junto com o Analista de Requisitos, as histórias de usuário;
- validar as histórias de usuário detalhadas;
- homologar o software em todas as suas funcionalidades;
- autorizar e agendar a implantação do software em produção.

3.2 - Scrum Master (SM)

Responsabilidades:

- garantir que a execução do processo seja realizada sem interrupções;
- proteger a equipe de desenvolvimento de interferências externas, garantindo que ela mantenha o foco no desenvolvimento do produto;
- promover e garantir que o processo desenvolvimento seja compreendido e cumprido pelos participantes do projeto;
- convocar e mediar as reuniões de planejamento de sprint;
- convocar e demonstrar o produto nas reuniões de demonstração;
- auxiliar na criação e priorização do backlog.

3.3 - Gerente de Projeto (GP)

Na maioria das vezes, o GP assumirá também o papel de SM. Portanto, além das responsabilidades de SM, o GP terá as seguintes responsabilidades:

- planejar, acompanhar e controlar o desenvolvimento do software, com ênfase em custos e prazos;
- elaborar o cronograma, orçamento e plano de trabalho do projeto;
- gerenciar e controlar as mudanças do projeto;

- realizar a reunião de kickoff;
- realizar o planejamento preliminar do projeto, definindo equipe do projeto, estimando prazo e custo.

3.4 - Analista de Requisitos (REQ)

Responsabilidades:

- definir, junto com o PO, o escopo do produto e criação do backlog do produto;
- detalhar, junto com o PO, as histórias de usuário;
- aprovar o protótipo;
- receber e dar aprovação técnica da entrega;
- treinar os usuários finais.

3.5 - Arquiteto de software (ARQ)

- definir a arquitetura da solução, incluindo a modelagem dos componentes e plataforma tecnológica;
- prover esclarecimento a eventuais dúvidas dos desenvolvedores sobre a arquitetura definida;
- criar a modelagem de dados;
- criar e manter os scripts de criação e alteração do esquema do banco de dados (tabelas, índices, regras de integridade, *stored procedures*, etc.).

3.6 - Desenvolvedor (DEV)

Responsabilidades:

- validar e esclarecer possíveis dúvidas sobre a arquitetura definida;
- elaborar o código da solução desenhada nas fases de Especificação e Projeto Arquitetônico;
- prover a publicação do software nos ambientes de desenvolvimento, homologação e produção;
- criar e manter os objetos do banco de dados em ambiente de desenvolvimento, homologação e produção.

3.7 - Analista de Infra (INFRA)

Responsabilidades:

- disponibilizar e manter os ambientes de desenvolvimento, homologação e produção;
- implantar o software em produção, executando os scripts fornecidos pelo Desenvolvedor.

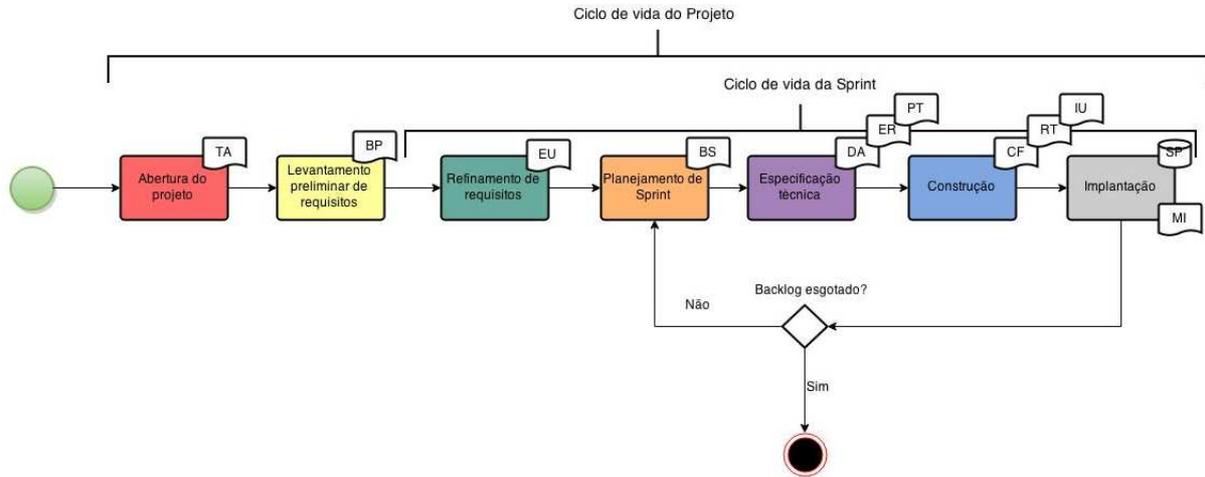
3.8 - Analista de Testes (TEST)

Responsabilidades:

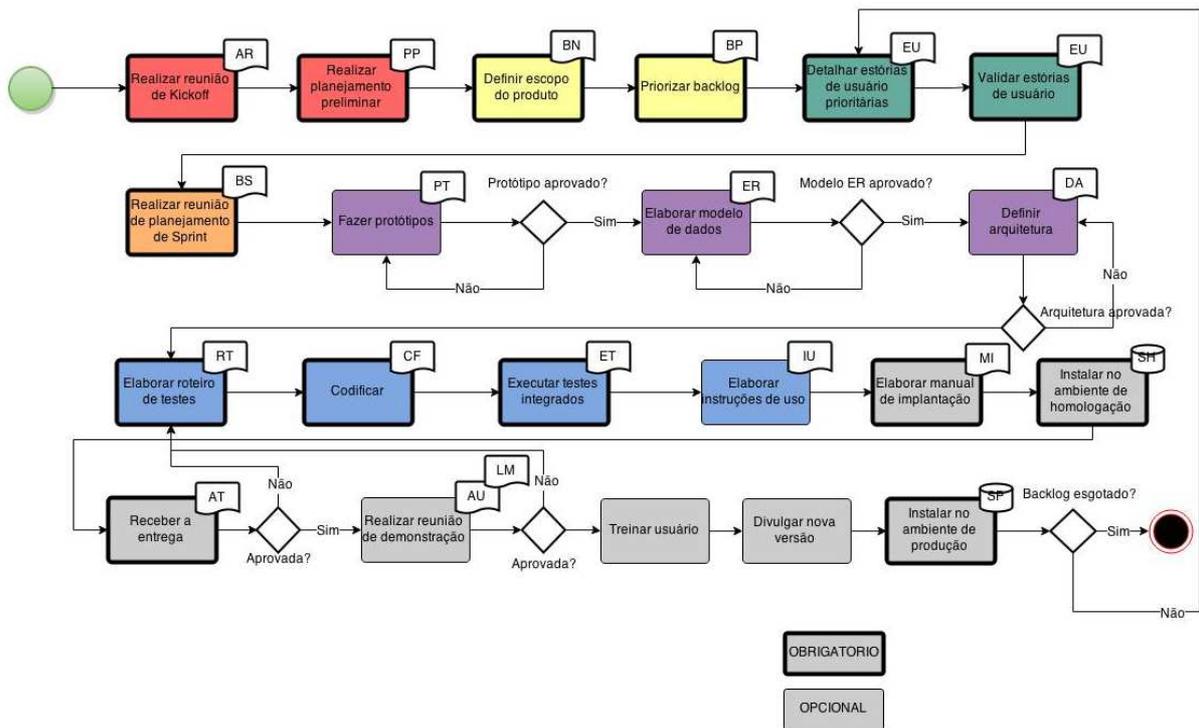
- elaborar roteiro de testes;
- executar os testes integrados para validar o código implementado;
- gerar evidências de testes;
- registrar ocorrência de defeitos encontrados nos testes integrados.

4 - Fluxo

4.1 - Ciclo de vida



4.3 - Fluxo detalhado



4.4 - Artefatos

<p>TA: Termo de Abertura</p> <p>BP: Backlog o produto</p> <p>BS: backlog da sprint</p> <p>EU: Estórias de usuário</p> <p>DA: Documento de Arquitetura</p> <p>ER: Modelo Entidade Relacionamento</p> <p>PT: Protótipo de tela</p> <p>CF: Código Fonte</p> <p>MU: Manual do usuário</p> <p>IU: Instruções de uso</p> <p>RT: Roteiro de testes</p> <p>SP: Software em Produção</p> <p>MI: Manual de Implantação</p> <p>BN: Backlog não priorizado</p> <p>BP: Backlog do produto</p>	<p>ET: Evidências de teste</p> <p>SH: Software em homologação</p> <p>AT: Aprovação técnica</p> <p>LM: Lista de mudanças</p> <p>AU: Aprovação de usuário</p> <p>BN: Backlog não priorizado</p> <p>BP: Backlog do produto</p> <p>ET: Evidências de teste</p> <p>SH: Software em homologação</p> <p>AT: Aprovação técnica</p> <p>AU: Aprovação de usuário</p> <p>AR: Ata de reunião</p> <p>TA: Termo de Abertura</p> <p>BP: Backlog o produto</p> <p>BS: backlog da sprint</p> <p>EU: Estórias de usuário</p>	<p>DA: Documento de Arquitetura</p> <p>ER: Modelo Entidade Relacionamento</p> <p>PT: Protótipo de tela</p> <p>CF: Código Fonte</p> <p>MU: Manual do usuário</p> <p>IU: Instruções de uso</p> <p>RT: Roteiro de testes</p> <p>SP: Software em Produção</p> <p>MI: Manual de Implantação</p> <p>BN: Backlog não priorizado</p> <p>BP: Backlog do produto</p> <p>ET: Evidências de teste</p> <p>SH: Software em homologação</p> <p>AT: Aprovação técnica</p> <p>AU: Aprovação de usuário</p>
--	--	---

4.5 - Matriz de responsabilidade

Atividade	PO	SM	GP	REQ	ARQ	DEV	TEST	INFRA
Realizar reunião de kickoff			■					
Realizar planejamento preliminar			■					
Definir o escopo do produto				■				
Priorizar backlog do produto				■				
Detalhar estórias de usuário				■				
Validar estórias de usuário	■							
Realizar reunião de planejamento de sprint		■						
Fazer protótipos						■		
Aprovar protótipos				■				
Elaborar modelo de dados					■			

Aprovar modelo de dados								
Definir arquitetura								
Aprovar arquitetura								
Elaborar roteiro de testes								
Codificar								
Executar testes integrados								
Elaborar instruções de uso								
Elaborar manual de implantação								
Instalar no ambiente de homologação								
Receber a entrega								
Realizar reunião de demonstração								
Dar aprovação de usuário								
Treinar o usuário final								
Divulgar a nova versão								
Instalar no ambiente de produção								