Tribunal Regional do Trabalho da 24ª Região
Coordenadoria de Tecnologia da Informação e Comunicações

MPS-TRT24 – Metodologia de produção de software

Aguardando aprovação

Histórico do documento

Data	Descrição
28/10/2010	Criação da metodologia baseada no framework *Beware*, através da CI TRT/SINF/NCS Nº 026/2010
06/05/2011	Revisão da metodologia baseada no framework *Beware*
15/02/2016	Baseado no processo de trabalho Gerenciar software, descrito no PGS-TRT24
25/08/2016	Revisão das diretrizes de manutenção de software e inclusão de glossário

Equipe responsável

Nome	Função	Contato
Aise Maria Longhi Canéppele	Chefe do Gabinete de Sistemas de Informação	aise@trt24.jus.br
Alex Sandro Pontes da Silva		assilva@trt24.jus.br
Waldeci Leitun de Almeida	Chefe da Seção de Desenvolvimento de Sistemas	walmeida@trt24.jus.br

Sumário

1 INTRODUÇÃO **3**

2 TERMOS E DEFINIÇÕES **3**

3 PROCESSOS DE TRABALHO **3**

4 DIRETRIZES **4**

4.1 DIRETRIZES APLICADAS AO DESENVOLVIMENTO DE SOFTWARE 4

4.1.1 Modelo desenvolvimento de software- abordagem do Open Up *4*

Fase de Concepção *6*

Fase de Elaboração *7*

Fase de Construção *7*

Fase de Transição *8*

4.1.2 Adaptação do framework Open Up *8*

4.1.3 Documentação *10*

4.2 DIRETRIZES APLICADAS À MANUTENÇÃO DE SOFTWARE 11

4.2.1 Liberação de versões *11*

4.2.2 Documentação *12*

5 PADRONIZAÇÃO **12**

6 REFERÊNCIAS **12**

ANEXO I – LISTA DE ARTEFATOS **13**

ANEXO II – GLOSSÁRIO **16**

Aguardando aprovação

1 Introdução

Este documento apresenta a Metodologia de Produção de Software (MPS-TRT24), aplicável aos projetos de desenvolvimento e a manutenção de software.

A MPS-TRT24 foi elaborada pelo Gabinete de Sistemas de Informação e tem como objetivo padronizar as atividades de desenvolvimento e de manutenção de software. O público alvo deste instrumento é a equipe de desenvolvimento do Gabinete de Sistemas de Informação.

2 Termos e definições

a. **CTIC**: Coordenadoria de Tecnologia da Informação e Comunicações

b. **GSI**: Gabinete de Sistemas de Informação

c. **PGS-TRT24**: Processo Gerenciar Software - Produção e manutenção de software TRT24

d. **MPS-TRT24**: Metodologia que define as diretrizes de produção e manutenção de software

e. **Desenvolvimento de software**: Execução de um processo de trabalho com o objetivo de entregar novo software ou serviço.

f. **Manutenção de software**: Execução de um processo de trabalho com o objetivo de atualizar o software já existente seja através de uma correção ou evolução.

g. **Manutenção evolutiva**: Alteração executada no software, com o objetivo de agregar novas funcionalidades.

h. **Manutenção adaptativa**: Alteração executada no software, com o objetivo de adaptar o software quando existe alteração do ambiente ao qual o mesmo de destina.

i. **Manutenção corretiva**: Alteração executada no software com o objetivo de corrigir falhas de processamento do software.

3 Processos de trabalho

A MPS-TRT24 regulamenta a execução dos processos ***Desenvolver software*** e ***Manter software***, de produção e de manutenção de software descritas no Processo ***Gerenciar software*** – PGS-TRT24.

Aguardando aprovação

4 Diretrizes

4.1 Diretrizes aplicadas ao desenvolvimento de software

a. O desenvolvimento de software deve seguir a execução do processo de trabalho ***Desenvolver software***, descrito no PGS-TRT24;

b. O processo ***Desenvolver software*** é executado em paralelo ao processo ***Gerenciar projeto***, ao qual é subordinado;

c. O trabalho deve ser executado de forma colaborativa e, de preferência, com a disponibilidade dos membros da equipe no mesmo local de trabalho;

d. O PGS-TRT24 descreve o conjunto mínimo de papéis, atividades e artefatos envolvidos no desenvolvimento de software. Os membros da equipe serão alocados a estes papéis específicos e executarão as atividades previstas no processo, gerando os artefatos exigidos;

e. A equipe alocada a um projeto de desenvolvimento deve ser composta, minimamente, por um analista, um arquiteto, um desenvolvedor e um testador, além do gerente do projeto;

f. A definição da equipe de projeto dar-se-á através de ordem de serviço de designação e será registrada na ferramenta de gerenciamento de projetos;

g. A participação dos clientes em todas as fases do projeto é crítica para o sucesso do mesmo;

h. Todas as atividades previstas no processo ***Desenvolver software*** devem ser registradas em ferramenta adequada, definida pelo GSI;

4.1.1 Modelo desenvolvimento de software- abordagem do *Open Up*

O processo ***Desenvolver software***, descrito no PGS-TRT24, é adaptado do *framework Open UP* e aplica a abordagem incremental em um ciclo de vida estruturado. É um processo iterativo realizado em quatro fases onde cada fase, por sua vez, pode ser composta de várias iterações.

Esta abordagem permite que o software seja construído em ciclos de desenvolvimento (iterações), com a vantagem de ser possível entregar valor ao cliente mesmo antes da conclusão do projeto. Os módulos do sistema podem ser entregues em partes, após a conclusão de cada iteração, até que o software esteja totalmente pronto.

A execução do projeto de forma incremental permite avaliar constantemente o valor do que foi entregue, além de permitir a alteração do escopo do projeto para corrigir falhas e omissões ou, ainda, aperfeiçoar os seus resultados.

O cliente participa ativamente do projeto e consegue perceber a sua evolução à medida que vai recebendo partes do produto que podem ser utilizadas mesmo que o software não esteja completo. Neste contexto, o processo de desenvolvimento tem seguintes objetivos:

a. Executar apenas as atividades que agreguem valor ao trabalho;

b. Produzir apenas os artefatos necessários;

c. Permitir a extensão do processo para atender a eventuais necessidades de adaptação às mudanças que surgirem durante a execução do projeto;

d. Garantir a execução completa da construção de um sistema através de um número suficiente de iterações;

e. Garantir a agilidade do processo através da adoção de boas práticas de desenvolvimento de software;

É importante ressaltar que o processo de desenvolvimento segue os fundamentos do *Open Up*, porém a descrição dos ciclos de iterações foi adaptada, pois adotamos processos que executam essas iterações; estes processos se repetem várias vezes dentro de cada fase, pois são executados a cada iteração.

Com esta nova forma de desenhar o ciclo de vida do processo ***Desenvolver software***, cada iteração é controlada dentro de um processo específico e que, por sua vez, se repete quantas vezes quanto forem necessárias dentro de cada uma das fases do *Open Up*.

Esta notação foi utilizada para facilitar a visualização do fluxo do processo segundo a notação do BPMN (*Business Process Model and Notation*), adotada no mapeamento dos demais processos de trabalho da CTIC. Porém, apesar de o desenho do ciclo de vida do processo de desenvolvimento de software ter sido adaptado, na prática, executamos as atividades previstas no *Open Up*, mas restringimos os papéis e os artefatos obrigatórios. As figuras 1 e 2 apresentam a notação das fases do *Open Up* e a notação adotada no PDS-TRT24.

Aguardando aprovação

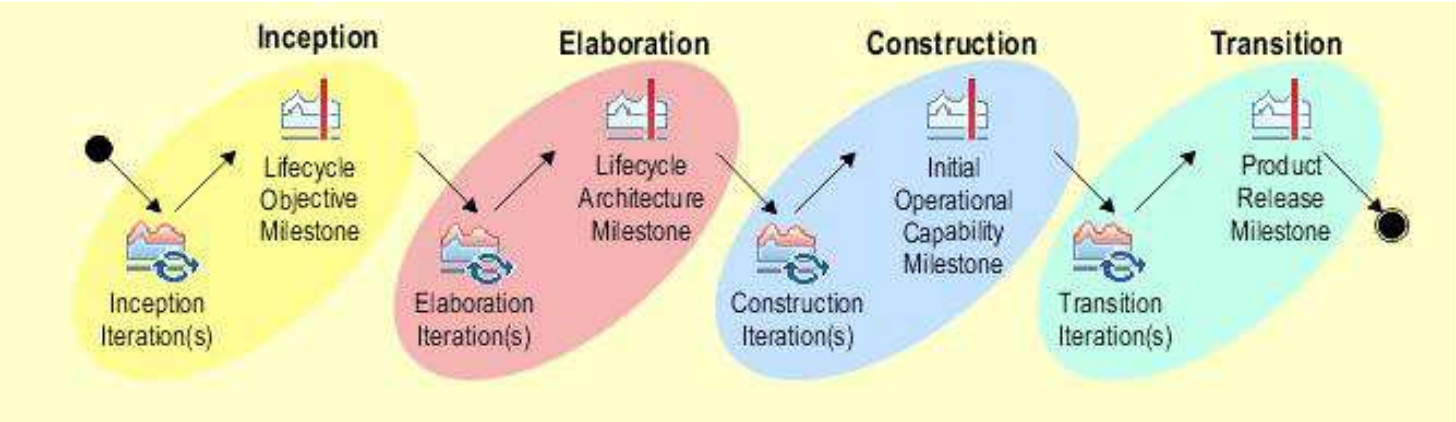

Figura 1: Ciclo de vida do projeto no *Open Up*

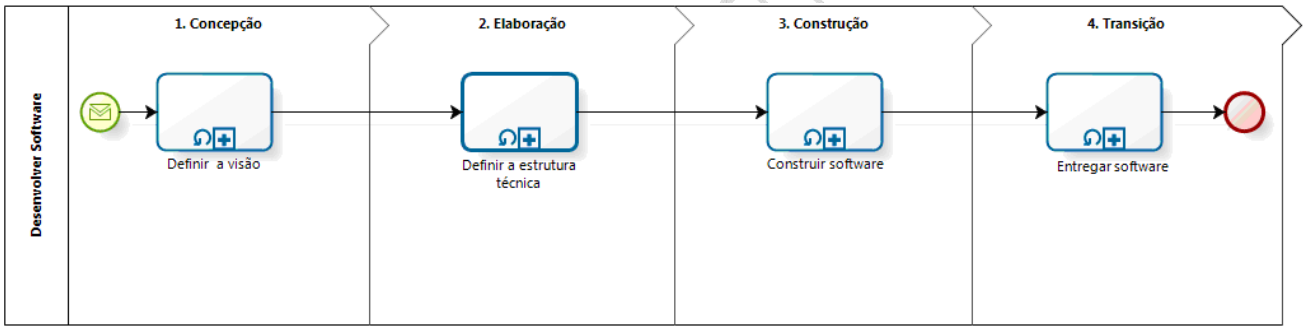

Figura 2: Notação adotada no PDS-TRT24 – Ciclo de vida do projeto

A tabela abaixo apresenta o ciclo de vida do projeto estruturado em quatro fases, cada uma com as suas iterações, que são delimitadas pelos marcos que definem o fim de cada fase.

Fase de Concepção	
Finalidade	A **fase de concepção** marca o início do processo de desenvolvimento. A sua finalidade é obter a concordância dos envolvidos quanto ao produto que será desenvolvido. Os principais objetivos da fase são: a. Determinar a visão do produto;

	b. Estabelecer uma visão operacional, critérios de aceitação e o que deve ou não estar no produto (escopo do sistema); c. Identificar e priorizar as principais funcionalidades do sistema; d. Definir pelo menos uma opção de arquitetura candidata para a solução proposta. A conclusão desta fase produz a concordância dos envolvidos sobre a definição do escopo do projeto, a priorização de um conjunto inicial de requisitos a serem atendidos, a identificação dos principais riscos e as estratégias para a sua de atenuação.
Marcos	a. A visão do produto definida; b. O escopo do projeto definido; c. Conjunto inicial de requisitos, devidamente priorizados;
Processo	*Definir a visão* (descrito no PGS-TRT24)
Observação	Geralmente, esta fase é composta por apenas uma iteração, mas pode ter várias iterações, dependendo da complexidade do projeto.
Fase de Elaboração	
Finalidade	A **fase de elaboração** destina-se a prover a estrutura técnica que permita o início das atividades de desenvolvimento do software. Nesta fase busca-se, estabelecer a arquitetura do sistema para fornecer uma base estável ao esforço de desenvolvimento da fase de construção, mitigando os riscos de ordem técnica. Além disso, propõe-se, também a evitar riscos não técnicos, buscando a concordância dos envolvidos com a solução proposta. Os principais objetivos da fase são: a. Assegurar que a arquitetura, os requisitos e os planos sejam estáveis o suficiente e que os riscos sejam minimizados; b. Tratar todos os riscos significativos do ponto de vista da arquitetura do projeto; c. Configurar o ambiente de suporte para o projeto. Isso inclui preparar *templates,* diretrizes e configurar ferramentas.
Marcos	a. Arquitetura definida; b. Ambiente de desenvolvimento definido.
Processo	*Definir a solução* (descrito no PGS-TRT24)
Observação	Esta fase pode ser composta de várias iterações, dependendo da complexidade do projeto.
Fase de Construção	
Finalidade	A **fase de construção** é dedicada ao desenvolvimento propriamente dito, ou seja, quando o software é efetivamente construído. O principal objetivo desta fase é a entrega de uma versão operacional do software que esteja alinhado à visão do produto e que atenda às expectativas dos clientes. Seus objetivos específicos são:

Minuta aguardando aprovação

	a. Desenvolver de forma iterativa um produto completo até que esteja pronto para ser entregue aos clientes. b. Liberar várias versões incrementais do software até que a aplicação possa ser implantada;
Marcos	Incremento de software entregue;
Processo	*Construir software* (descrito no PGS-TRT24)
Observação	Esta fase é composta de várias iterações, até que o software esteja totalmente construído – o processo Construir software é executado até que todas as funcionalidades do software sejam concluídas.
Fase de Transição	
Finalidade	A **fase de transição** é a fase final do processo, quando ocorre a transição do software para o ambiente do cliente e a obtenção do aceite do cliente de que o seu produto está completo. O objetivo desta fase é assegurar que o software esteja pronto para ser entregue. Abrange o planejamento da implantação, o treinamento dos usuários, a importação de bases de dados legadas e, finalmente, a implantação propriamente.
Marcos	Produto implantado;
Processo	Entregar software (descrito no PGS-TRT24)
Observação	Esta fase pode ser composta por uma ou mais iterações, mas geralmente apenas uma é suficiente.

4.1.2 Adaptação do framework *Open Up*

Em razão do tamanho reduzido da equipe e da necessidade de maximizar a produtividade de desenvolvimento, adaptamos o framework para a estratégia em que as atividades de elicitação e análise de requisitos serão sempre iniciadas numa iteração anterior às atividades de codificação do incremento de software.

O objetivo dessa estratégia é que o analista de sistemas esteja uma iteração à frente do desenvolvedor. Dessa forma, o analista e o desenvolvedor atuam em paralelo, mas não sobre os mesmos requisitos. Durante uma iteração, o desenvolvedor consumirá os artefatos que foram produzidos pelo analista na iteração anterior. A figura 3 apresenta esta estratégia.

Aguardando aprovação

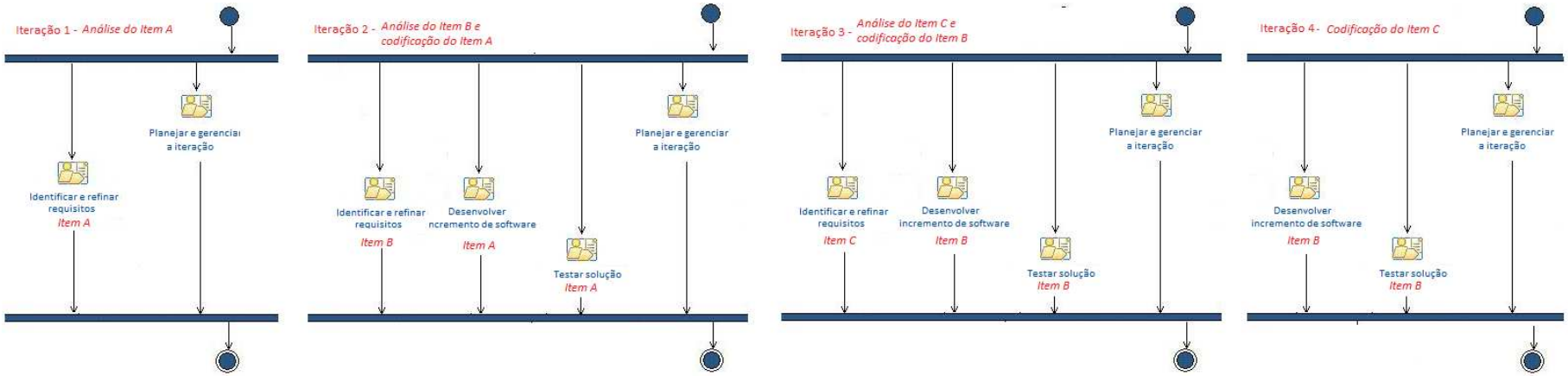

Figura 3: Execução em paralelo

Excepcionalmente, o gerente de projeto poderá adotar como estratégia, dependendo das características do projeto, que o analista e o desenvolvedor atuem de forma seqüencial em uma mesma iteração. Neste caso, o desenvolvedor consumirá os artefatos produzidos pelo analista. A figura 4 apresenta esta estratégia.

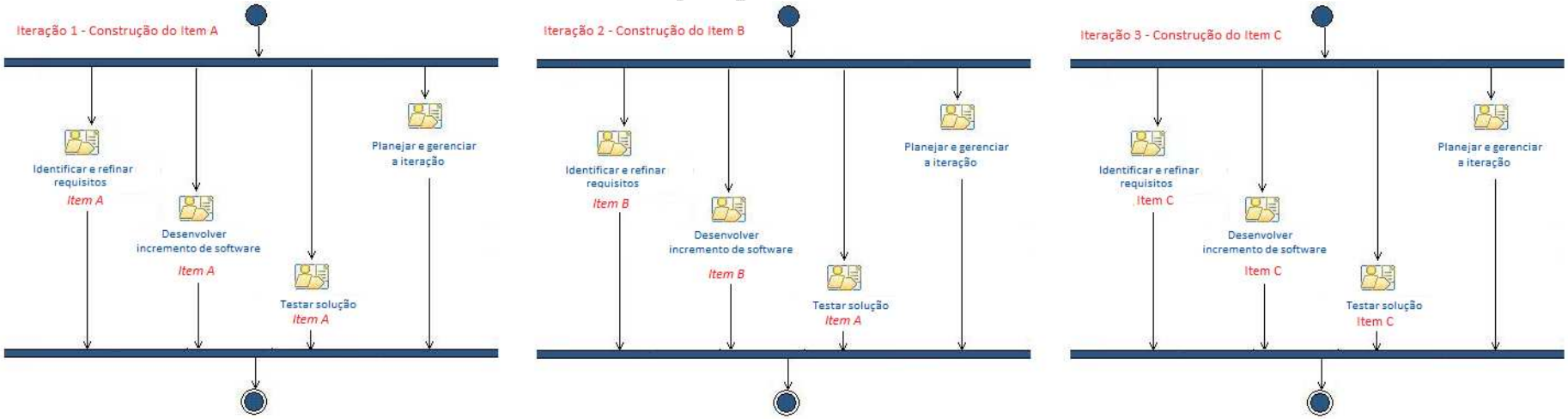

Figura 4: Execução sequencial

4.1.3 Documentação

a. A documentação do software é requisito de QUALIDADE e deve ser registrada no projeto;

b. O gerente do projeto especificará no Plano de Qualidade quais artefatos obrigatórios e opcionais deverão ser produzidos de acordo com as peculiaridades e a complexidade do software a ser desenvolvido no projeto;

c. Os artefatos que documentam o software e que são gerados ao longo do projeto, durante a execução das iterações, devem ser armazenados em repositório específico sustentado por ferramenta apropriada e que contemple o desenvolvimento do ciclo de vida do produto;

d. Os artefatos que documentam a estrutura de banco de dados e que são gerados ao longo do projeto, durante a execução das iterações, devem ser armazenados em repositório específico sustentado por ferramenta apropriada e devidamente integrada ao banco de dados homologado (Oracle);

e. As atividades executadas nas iterações devem ser registradas em ferramenta apropriada a ser definida pelo GSI;

f. Após o início do projeto, o GSI emitirá ordem de serviço com as seguintes definições:

   - Designar os membros da equipe de projeto;
   - Definir o repositório de armazenamento dos artefatos do projeto;
   - Definir o repositório de armazenamento dos artefatos de modelagem de dados;
   - Definir o repositório de armazenamento do registro de atividades das iterações.

g. Após o início do projeto, as ordens de serviço que sustentam o projeto e o documento de visão devem ser disponibilizados para consulta na ferramenta de gerenciamento do projeto;

h. Ao final da execução do projeto os artefatos que documentam o software e que foram gerados durante a execução das iterações, devem ser disponibilizados para consulta na ferramenta de gerenciamento do projeto.

i. Durante a execução do projeto, as informações permanecerão apenas nos repositórios apropriados e não serão apresentados, pois se tratam de documentos muito dinâmicos e que são constantemente atualizados;

j. Os artefatos produzidos pelos analistas de sistemas devem ser repassados aos desenvolvedores em arquivos em formato texto ou PDF.

4.2 Diretrizes aplicadas à manutenção de software

a. As demandas de manutenção adaptativa ou evolutiva de software devem ser executadas somente depois de priorizadas pelo gerente do processo ***Gerenciar software***, descrito no PGS-TRT24, e serão controladas através de ciclos de manutenção;

b. Os ciclos de manutenção serão registrados e controlados em ferramenta apropriada a ser definida pelo GSI;

c. A duração do ciclo de manutenção - de duas até quatro semanas - será definida pelo Gerente do Processo;

d. O Gerente do Processo será responsável pela distribuição das tarefas entre os membros da equipe e por controlar a sua execução;

e. O Gerente do Processo será responsável por definir os itens de *backlog* que serão executados durante o ciclo de manutenção. Os itens serão selecionados das listas de *backlog* de todos os sistemas passíveis de manutenção;

f. As atividades de manutenção de software devem seguir a execução do processo de trabalho ***Manter software***, descrito no PGS-TRT24;

g. O PGS-TRT24 descreve o conjunto mínimo de papéis, atividades e artefatos envolvidos na manutenção de software;

h. Não existe alocação prévia de equipe para as atividades previstas no processo ***Manter software***. As manutenções serão executadas pelos desenvolvedores e analistas designados para atuar na manutenção do software;

i. O servidor designado como responsável pelo software junto ao Catálogo de Serviços do GSI será, preferencialmente, alocado para atuar no processo de manutenção do software;

j. As informações sobre a execução das atividades de manutenção corretiva previstas no processo ***Manter software*** devem ser adicionadas ao registro da abertura do incidente na ferramenta definida pela CTIC;

k. As atividades de manutenção evolutiva ou adaptativa previstas no processo ***Manter software*** devem ser registradas na ferramenta apropriada ao gerenciamento da execução de demandas definida pelo GSI;

l. Quando houver alteração de requisitos, estas serão registradas no repositório específico sustentado por ferramenta apropriada e que já contém as informações sobre o produto;

4.2.1 Liberação de versões

a. Os incidentes serão atendidos através de um planejamento prévio de liberação de versão do software, a cada 15 dias. Soluções de contorno aos incidentes devem ser adotadas até a liberação de nova versão. Esta diretriz não se aplica aos incidentes graves e que causarem paralisação de serviços, pois estes forçarão o lançamento de versões de correção extraordinárias;

b. As demandas serão atendidas através de um planejamento prévio de liberação de versão do software, a cada 60 dias. Soluções de contorno que atendam as necessidades dos usuários devem ser adotadas até a liberação de nova versão. Esta diretriz não se aplica a demandas de caráter emergencial, pois estas forçarão o lançamento de versões extraordinárias;

c. Toda e qualquer liberação de nova versão de software será executada através do processo ***Gerenciar mudança***, da CTIC;

4.2.2 Documentação

a. Os artefatos obrigatórios estão definidos no PGS-TRT24, processo ***Manter software***;

b. Caso a manutenção executada envolva a alteração de requisitos previamente existentes, ou a inclusão de novos requisitos, caberá ao analista designado manter o repositório de informações sobre o software devidamente atualizado;

c. Caso a manutenção executada envolva a alteração da estrutura de banco de dados, caberá ao analista designado manter o repositório de informações sobre o software devidamente atualizado;

d. A documentação da alteração de requisitos e de banco de dados, produzida pelos analistas de sistemas, deve ser repassada aos desenvolvedores em arquivos no formato texto ou PDF.

5 Padronização

a. A padronização das atividades de análise de requisitos será definida através de ordem de serviço a ser emitida pelo GSI;

b. A padronização das atividades de modelagem de dados será definida através de ordem de serviço a ser emitida pelo GSI;

6 Referências

a. Política de Gerenciamento do Desenvolvimento e Manutenção de Software do Tribunal Regional do Trabalho da 24ª Região;

b. Processo *Gerenciar software* – PGS-TRT24;

c. Processo *Gerenciar mudança*;

d. Processo *Gerenciar demandas;*

e. Processo *Gerenciar projeto;*

f. Processo Gerenciar Incidentes;

Aguardando aprovação

ANEXO I – LISTA DE ARTEFATOS

Os seguintes artefatos são produzidos durante a execução dos processos de trabalho executados no GSI. A obrigatoriedade dos mesmos dependerá do tipo de projeto de desenvolvimento e da fase que o projeto se encontra.

Artefato	Descrição	Responsável
Build	Este artefato é a versão operacional do software, o produto desenvolvido.	Desenvolvedor
Caso de teste	Este artefato apresenta a especificação de um conjunto de entradas de teste, condições de execução e resultados previstos com a finalidade de fazer a avaliação de algum aspecto particular de um cenário.	Analista de Sistemas
Caso de uso	Artefato que especifica as funções que devem ser contempladas pelo sistema. É utilizado como fonte de informações para as atividades de análise, design e teste e detalha o que o sistema irá executar. Apresenta os fluxos principais, alternativos e de exceção, além da descrição completa dos atores que interagem com o sistema.	Analista de Sistemas
Diagrama de casos de uso	Diagrama que apresenta os casos de uso a serem implementados. É utilizado para descrever graficamente um subconjunto do modelo de casos de uso para simplificar a comunicação.	Analista de Sistemas
Dicionário e modelo de dados	Esses artefatos descrevem as representações lógicas e físicas dos dados persistentes utilizados pelo aplicativo. Pode incluir, também, os elementos de modelo para procedimentos armazenados, gatilhos e restrições que definem a interação dos componentes de aplicativo com o SGBD.	Analista de Sistemas
Caderno de arquitetura	Documento que fornece uma visão geral de arquitetura abrangente do sistema. Serve como um meio de comunicação entre o arquiteto de software e os demais membros de equipe de projeto, com relação a decisões arquiteturalmente significativas tomadas sobre o projeto.	Arquiteto de software

Fontes	Arquivos de código do software, arquivos de dados e arquivos de suporte tais como os arquivos de ajuda on-line que representam as partes básicas de um sistema que possa ser construído.	Desenvolvedor
Glossário	Lista de termos e definições relevantes ao sistema.	Analista de Sistemas
Material de apoio	Documentação produzida na fase de entrega do software que auxiliar no entendimento do sistema por parte do usuário. Exemplos: Manual do Usuário (impresso ou *WEB*), Guia Rápido, FAQ, Ajuda *Online* e treinamentos.	Analista de Sistemas
Planilha de Contagem de pontos de função	Planilha de contagem do tamanho funcional do software desenvolvido e que utiliza as métricas por ponto de função.	Analista de métricas
Planilha de Estimativa em pontos de função	Planilha estimativa do tamanho o software a ser desenvolvido e que utiliza as métricas por ponto de função.	Analista de métricas
Protótipo	Apresenta as futuras funcionalidades, em formato estático, através dos desenhos das telas do sistema. É utilizado para obter e apresentar informações ao cliente, a fim de retificar ou ratificar requisitos. Este artefato deve ser elaborado, com exceção dos casos em que o projeto não envolve interfaces dos usuários.	Analista de Sistemas
Regra de negócio	Requisito que detalha as regras do negócio do cliente que possuem influência sobre o comportamento do software a ser desenvolvido. As suas sentenças devem possuir uma abordagem restritiva, pois elas restringem os processos operacionais da organização; A lista de regras de negócio não descreve como cada regra será verificada ou como as operações deverão ser realizadas. Descreve apenas o que o processo irá executar. Deve ser percebida pela organização, não deve possuir elementos de soluções de tecnologia e pertence ao domínio do negócio e não do sistema.	Analista de Sistemas
Registro de testes	Este artefato coleta a saída capturada durante uma única execução de um ou mais testes para uma única realização do ciclo de teste.	Testador

Autenticação

	O registro de teste comprova a execução dos testes de aceitação e registra os defeitos encontrados.	
Requisitos	Artefato que descreve as capacidades necessárias para o software, a fim de satisfazer as necessidades do usuário para resolver um problema e alcançar um objetivo.	Analista de Sistemas
Termo de aceite do produto	Documento assinado pelo cliente e que recebe o software. Este documento descreve o que foi entregue e deve estar alinhado à visão inicial do produto.	Analista de Sistemas
Termo de aceite de requisitos	Documento assinado pelo usuário e que recebe a porção de software entregue após a realização de uma iteração. Este documento descreve todos os requisitos levantados e implementados no release e que devem ter sua funcionalidade confirmada pelo usuário recebedor.	Analista de Sistemas
Visão de produto	Artefato que apresenta claramente a visão do cliente em relação ao negócio e ao sistema de informação que será desenvolvido para suportá-lo; Deve apresentar, além da visão dos envolvidos com o produto a ser desenvolvido, também a especificação das necessidades e recursos mais importantes; Os requisitos centrais pretendidos devem estar descritos, constituindo-se em uma base contratual para o desenvolvimento do produto;	Analista de Sistemas

Aguardando Aprovação

ANEXO II – GLOSSÁRIO

1. ABREVIATURAS E SIGLAS

TERMO	DESCRIÇÃO
APF	Análise de Pontos por Função
BPMN	Notação e Modelo de Processos de Negócio
CTIC	Coordenadoria de Tecnologia da Informação e Comunicações
DBA	*Data Base Administrator* (Administrador de Banco de Dados)
EAP (ou WBS)	Estrutura Analítica do Projeto (ou *Work Breakdown Structure)*
GSI	Gabinete de Sistemas de Informação
UML	Unified Modeling Language - Linguagem Unificada de Modelagem é uma linguagem padrão para modelagem orientada a objetos.

2. TERMOS E DEFINIÇÕES

TERMO	DESCRIÇÃO
Análise de Pontos de Função	Método padrão para medir as funcionalidades de um software sob a perspectiva do usuário. Mede o desenvolvimento e manutenção de software de forma independente da tecnologia utilizada para sua implementação.
Análise de Requisitos	Atividade da engenharia de requisitos responsável pela documentação, organização, modelagem, classificação em grupos coerentes, verificação e validação dos requisitos.
Arquitetura de Software	Descreve a "planta baixa" para desenvolvimento de software, em geral, representada por visões arquiteturais. Também contém o raciocínio, as suposições, as explicações e as implicações das decisões

Aguardando aprovação

	que foram tomadas na formação da arquitetura, bem como o mapeamento global entre as visões.
Aplicação	Conjunto coeso e automatizado de funções, procedimentos e dados que dão sustentação a um objetivo de negócio. Também chamado de sistema de informação.
Artefato	Item criado como parte da definição, manutenção ou utilização de um processo de desenvolvimento/manutenção de sistemas de informação. Inclui, entre outros, procedimentos, especificações, projeto de arquitetura, projeto detalhado, código fonte, documentação para o usuário. Os artefatos documentam o sistema e permitem a comunicação entre os membros da equipe do projeto ou destes como os seus clientes. Além disso, possibilitam um melhor entendimento do sistema para as futuras operações de manutenção.
Ator	Elemento da técnica Caso de Uso que representa uma pessoa (ou representante de um grupo de pessoas) que desempenha um papel, ou "coisa" (outros softwares, dispositivos) que interage com a aplicação com finalidade de cumprir um trabalho significativo.
Backlog	Lista que contém todas as funcionalidades desejadas para o produto em construção ou evolução. O conteúdo é definido pelo Analista de Sistemas e muda à medida que os requisitos vão sendo eliciados e se aprende mais sobre o produto e seus usuários. A lista de *backlog* armazena os requisitos ainda não atendidos em um sistema, além de um conjunto de requisitos ou características que tenham sido identificados como candidatos para uma potencial implementação.
Baseline	1. Uma configuração estável de artefatos de requisitos revisados e aprovados. 2. Uma visão de um ponto no tempo dos requisitos que foi revisada e cujo teor foi acordado de forma que sirva de base para desenvolvimento adicional. 3. A versão aprovada de um produto de trabalho que pode ser modificado apenas por procedimentos formais de mudança e é usado como uma base para comparação.
Bug	Problema na aplicação que, se não corrigido, pode provocar sua falha ou a geração de resultados incorretos. A ausência de funcionalidade que foi especificada e solicitada também é considerada um defeito.

Aguardando aprovação

Build	1. Versão operacional de um sistema ou de uma parte do sistema que demonstre um subconjunto das capacidades a serem fornecidas no produto final. 2. Construção de algo que tem um resultado observável e tangível, como o processo de conversão de arquivos de código fonte em um artefato de software que pode ser executado no computador (executável).
Caso de Teste	É um conjunto de condições usadas para teste de software. Deve especificar a saída esperada e os resultados esperados do processamento. Especifica e comunica as condições que necessitam ser validadas para permitir uma avaliação de alguns aspectos em particular dos itens de teste alvo. Casos de teste podem ser motivados por vários fatores, mas geralmente, incluem um subconjunto dos requisitos como casos de uso, características de desempenho e riscos com os quais o projeto está preocupado. Como uma regra geral, especificações de casos de teste são mais úteis onde a implementação do teste será complexa demais para se entender por si só sem o suporte de uma explicação menos detalhada fornecida pelo caso de teste.
Caso de uso	1. Artefato que define um conjunto de cenários, especificação de uma sequência de ações que um sistema pode executar, interagindo com os atores deste sistema. Um caso de uso define uma sequência de ações executadas por um sistema e que leva a um resultado de valor observável para um ator. 2. Artefato que especifica um diálogo entre um ator e o sistema que deve entregar valor. Um ator inicia um caso de uso, que invoca certas funcionalidades no sistema. Os casos de uso devem habilitar o sistema a entregar toda a funcionalidade esperada. 3. Artefato de análise que descreve as tarefas que o sistema deve executar para atores. 4. Artefato que representa os aspectos funcionais de um sistema ou operação. 5. Artefato que modela um requisito funcional do ponto de vista da interação entre usuário (ou ator) e o sistema. Descreve um conjunto de passos organizados em cenários que visam alcançar um objetivo do usuário. Não deve conter termos técnicos da área de desenvolvimento ou descrever como o sistema será construído, deve conter apenas a linguagem do usuário. Tipicamente um sistema terá vários casos de uso, cada um abordando objetivos individuais distintos a serem atingidos. Definido como parte da UML e componente central do RUP.

Aguardando aprovação

Ciclo de vida do projeto	É o conjunto das fases ou atividades que compõem um projeto de software.
Cliente	Tipo em especial de parte interessada com autoridade para tomar decisões sobre requisitos: solicitar, modificar, aprovar ou rejeitar requisitos.
Dicionário de dados	Descreve as representações lógicas e físicas dos dados persistentes utilizados pelo aplicativo.
Documento de aceite de requisitos	Documento emitido pelo Analista de Sistemas responsável pelo sistema e assinado pelo usuário que recebe o software. Este documento descreve todos os requisitos levantados e implementados no release e que devem ter sua funcionalidade confirmada pelo usuário recebedor.
Documento de visão	1. Artefato que apresenta claramente a visão do cliente em relação ao negócio e ao sistema de informação que será desenvolvido para suportá-lo. Deve apresentar, além da visão dos envolvidos com o produto a ser desenvolvido, também a especificação das necessidades e recursos mais importantes. Os requisitos centrais pretendidos devem estar descritos, constituindo-se em uma base contratual para o desenvolvimento do produto. 2. Contém a visão que as partes interessadas têm do sistema a ser desenvolvido, em termos das necessidades e características mais importantes. Por conter uma descrição dos requisitos centrais pretendidos, proporciona a base para requisitos mais detalhados. O documento de visão captura restrições de design e requisitos de alto nível para que o cliente possa compreender o sistema que será desenvolvido.
Elicitação de Requisitos	1. Atividade da engenharia de requisitos que seleciona e executa atividades para identificar e entender os domínios envolvidos, os requisitos de negócio, as partes interessadas e seus requisitos, além da solução e requisitos de transição. 2. Atividade parte do desenvolvimento de requisitos que identifica fontes para requisitos utilizando técnicas como entrevistas, protótipos, facilitação de *workshop* e análise de documentos, com o objetivo de coletar requisitos a partir dessas fontes.
Entregável	Uma saída de um processo que tem um valor para um cliente ou outro *Stakeholder*.
Especificações de Requisitos do Software	Documentação dos requisitos com a declaração oficial das necessidades de desenvolvimento para o software. É um contrato entre usuários e desenvolvedores. Especifica o que o

Apostila - Aprovação

	software deve fazer; contudo, não diz como deve ser feito.
Fontes	Arquivos de código do software, arquivos de dados e arquivos de suporte tais como os arquivos de ajuda on-line que representam as partes básicas de um sistema que possa ser construído.
Framework	Um *framework* ou arcabouço de software compreende de um conjunto de classes implementadas em uma linguagem de programação específica, usadas para auxiliar o desenvolvimento de software.
Integração Contínua	A integração contínua é uma prática de implementação onde membros de equipe integram seu trabalho com conjunto de mudanças realizadas por outros desenvolvedores e testam a aplicação antes de tornar seu trabalho disponível para os demais. Isto permite a detecção de erros de integração mais prematuramente possível, tais como erros de compilação, notificações do sistema de gerenciamento de configuração e erros relatados pela ferramenta de teste.
Manutenção Adaptativa	Modificação de um produto de software, realizada após sua entrega, para mantê-lo funcional em um ambiente modificado ou em mudança. A manutenção adaptativa fornece as melhorias necessárias para acomodar mudanças no ambiente no qual o produto de software deve operar. Essas mudanças são aquelas que devem ser feitas para manter o ritmo com o ambiente em mudança. Por exemplo, o sistema operacional poderia ser atualizado e algumas mudanças podem ser feitas para acomodar o novo sistema operacional.
Manutenção Corretiva	A modificação reativa de um produto de software executada depois da entrega para corrigir problemas identificados. A modificação corrige os produtos de software para satisfazer os requisitos.
Manutenção Perfectiva	Modificação de um produto de software depois da entrega para detectar e corrigir falhas latentes antes que elas se manifestem como falhas. Manutenção perspectiva provê melhorias para usuários, melhorias de documentação de programas e registros para melhorar o desempenho, a facilidade de manutenção ou outros atributos do software.
Modelo de Dados	É uma representação gráfica dos dados persistentes do sistema e o relacionamento entre eles. Pode ter um ponto de vista conceitual, lógico ou físico.
OPEN UP	Framework de processo de desenvolvimento de software. O Open UP prevê desenvolvimento iterativo e incremental. É uma metodologia livre de ferramentas e de baixo formalismo que

Aguardando aprovação

	pode ser estendido a uma variada gama de tipos de projetos e não apenas desenvolvimento de software.
Parte Interessada	Grupo ou pessoa que tem interesse que podem ser afetados por uma iniciativa de mudança ou influência sobre ela.
Plano de Iteração	Um conjunto de atividades e tarefas divididas por sequências de tempo, com recursos atribuídos e dependências de tarefas, para a iteração.
Protótipo	Artefato construído para simular alguns aspectos e recursos do software a fim de ser apresentado ao usuário, permitindo ao analista retificar ou ratificar requisitos.
Prototipação	É um processo interativo de gerar versões iniciais de um sistema futuro com o qual o usuário pode realizar verificações e experimentos, com intuito de avaliar algumas de suas características antes que o sistema seja construído. Sua finalidade é reduzir os riscos do projeto, permitindo a descoberta de falhas nos requisitos em suas etapas iniciais.
Regra de negócio	1. Declaração que controla ou define alguns aspectos de um negócio. O RUP define Regra de Negócio como “declarações sobre políticas ou condições que devem ser satisfeitas”. 2. Trata-se de uma restrição imposta pelo negócio e que regulamenta o comportamento de um procedimento operacional do negócio. É criada por políticas definidas pela organização. Podem ser originárias de leis, portarias e outros regulamentos. 3. São leis que regem o negócio, incluem: políticas corporativas, legislação pertinente, regulamentações governamentais, padrões de mercado. Não estão ligadas à solução (software), mas ao domínio de negócio onde a solução será utilizada. Existe independente do software, do processo de negócio estar automatizado ou não. 4. Regras de negócio têm vida totalmente independente de sistemas. Podem ser criadas e obedecidas sem uso de sistemas, pois pertence somente ao domínio do negócio. Entretanto, nada impede que regras de negócio venham a ser automatizadas, isto é, seja realizada por um sistema.
Release	É uma distribuição de uma primeira ou nova versão atualizada do software que está sendo construído. Pode ou não ser disponibilizada para o usuário para teste e homologação.
Repositório	É um local de armazenamento, na rede de computadores (servidores), das partes do software

Aguardando aprovação

	que está sendo desenvolvido.
Requisito	1. Uma condição ou capacidade do sistema, solicitada por um usuário, para resolver um problema ou atingir um objetivo. 2. Uma condição ou capacidade que deve ser atendida por uma solução para satisfazer um contrato, especificação, padrão ou quaisquer outros documentos formais impostos.
Requisito de Negócio	São declarações de mais alto nível de objetivos, metas ou necessidades da organização. Eles descrevem as razões pelas quais um projeto foi iniciado e as metas que o projeto vai atingir. Requisitos de negócio descrevem as necessidades da organização como um todo e não de grupos ou partes interessadas.
Requisito de Transição	1. Descrevem como realizar a transição do estado atual da organização para um estado futuro desejado, e que não serão mais necessários depois de concluída a transição. São diferenciados dos outros tipos de requisitos, pois são sempre temporários por natureza e não podem ser desenvolvidos até a solução existente e nova solução, estejam definidas. Tipicamente cobrem conversão de dados a partir de sistemas existentes, lacunas (*gaps*) em habilidades que devem ser abordadas e outras mudanças relacionadas para alcançar o estado futuro desejado. 2. Descrevem as capacidades temporárias, tais como a conversão de dados e de treinamento necessários à transição do estado atual como está (*as-is*) ao estado futuro (*to-be*).
Requisito não funcional	1. Capturam as condições que não se relacionam diretamente com o comportamento ou funcionalidade da solução, mas descrevem as condições ambientais sob as quais a solução se manter efetiva. 2. Completam os requisitos funcionais e descrevem as condições de ambiente ou qualidades exigidas para o produto ser efetivo. Exemplos incluem: confiabilidade, segurança de acesso, desempenho, segurança de sigilo, nível de serviço, capacidade de suporte, prazo de retenção/expurgo, etc. 3. Requisitos não funcionais do usuário incluem, mas não estão limitados a: - Restrições de qualidade (por exemplo, usabilidade, confiabilidade, eficiência e portabilidade);

Aguardando aprovação

	- Restrições organizacionais (por exemplo, locais de operação, hardware alvo e conformidade com normas); - Restrições ambientais (por exemplo, interoperabilidade, segurança, privacidade e segurança); - Restrições de entrega (por exemplo, linguagem de desenvolvimento, cronograma de entrega).
Requisito Funcional	1. Um subconjunto dos requisitos do usuário que descreve o que o software faz em termos de tarefas e serviços. 2. Parte dos requisitos da solução que descrevem o comportamento e as informações que a solução vai gerenciar. Eles descrevem as capacidades que o sistema será capaz de executar em termos de comportamento e operações – ações ou operações específicas da aplicação.
Restrição de Negócio	Limitação imposta ao projeto da solução pela própria organização. Descreve limitações às soluções disponíveis ou a aspectos da situação atual que não podem ser modificadas pela entrega da nova solução. Por exemplo, restrições orçamentárias, de prazo, limitações quanto ao número de recursos disponíveis, restrições baseadas nas habilidades da equipe de projeto e das demais partes interessadas, etc.
Restrição Técnica	Limitações ao projeto de uma solução, que derivam da tecnologia usada em sua implementação. Exemplos de restrições técnicas: Linguagens de desenvolvimento, plataformas de hardware/software e software de aplicação de deve ser usado, limites quanto à utilização de recursos, tamanho de mensagens e sincronização, espaço exigido para armazenamento do software, quantidade e tamanho para arquivos, registros e elementos de dados; ou padrão de arquitetura previamente existente.
Solução	Conjunto de mudanças à situação atual de uma organização para atender uma necessidade de negócio, resolver um problema ou aproveitar uma oportunidade.
Stakeholder	Parte interessada ou interveniente. Qualquer pessoa ou entidade que afeta ou é afetada pelas atividades do projeto.
Usuário	Qualquer pessoa ou coisa que interage com o software a qualquer momento.

Aguardando aprovação