



MINISTÉRIO DOS TRANSPORTES, PORTOS E AVIAÇÃO CIVIL  
SECRETARIA EXECUTIVA – SE  
COORDENAÇÃO GERAL DE TECNOLOGIA DA INFORMAÇÃO - CGTI  
COORDENAÇÃO DE PROJETOS E SISTEMAS - COPSI



# **METODOLOGIA DE DESENVOLVIMENTO DE SOFTWARE - MDS**

Todos os direitos reservados. É permitida a reprodução parcial ou total desta obra, desde que citada à fonte e que não seja para venda ou qualquer fim comercial.

Tiragem: 1.ª edição – 1ª impressão – 2017

Ministro dos Transportes, Portos e Aviação Civil:

Maurício Quintella Lessa

Secretário Executivo:

Fernando Fortes Melro Filho

Subsecretário de Assuntos Administrativos:

Wallace Moreira Bastos

Coordenador Geral de Tecnologia da Informação:

Kleber Campos Rodrigues Filho

Organização e Elaboração:

André Luiz Teodoro de Amorim

Felipe Torres da Silva e Souza

Valdirene Carneiro de Souza

Revisão, Projeto Gráfico e Capa:

COPSI

## Ficha Catalográfica

Brasil. Ministério dos Transportes. Secretaria Executiva. Produtos e Serviços: metodologia de desenvolvimento de software da Coordenação Geral de Tecnologia da Informação

### HISTÓRICO DO DOCUMENTO

<b>Data</b>	<b>Versão</b>	<b>Autor</b>	<b>Descrição</b>
14/02/2017	1.0	Grupo de Trabalho - MTPA	Elaboração da Metodologia de Desenvolvimento de Software do MTPA.

## ÍNDICE

1 INTRODUÇÃO.....	05
2 COORDENAÇÃO DE PROJETOS E SISTEMAS .....	06
3 SOBRE A METODOLOGIA DE DESENVOLVIMENTO DE SOFTWARE.....	07
3.1 ESTRUTURA GERAL.....	07
3.2 CLASSIFICAÇÃO.....	11
3.3 CRITÉRIO DE CLASSIFICAÇÃO DA DEMANDA.....	13
3.4 AUTORES.....	13
4 ORGANIZAÇÃO DO DOCUMENTO.....	13
5 PROCESSO: PROJETO.....	16
5.1 VISÃO GERAL – PROJETO.....	16
5.2 FASE – PROPOSTA DE PROJETO.....	18
5.3 FASE – INICIAÇÃO.....	22
5.4 FASE – ELABORAÇÃO.....	26
5.5 FASE – CONSTRUÇÃO.....	30
5.6 FASE – TRANSIÇÃO.....	34
5.7 SOLICITAÇÃO DE MUDANÇA.....	36
5.8 HOMOLOGAR PRODUTO.....	38
6 PROCESSO: MANUTENÇÃO.....	40
6.1 VISÃO GERAL – MANUTENÇÃO.....	40
6.2 FASE – INICIAÇÃO.....	41
6.3 FASE – ELABORAÇÃO.....	45
6.4 FASE – CONSTRUÇÃO.....	49
6.5 FASE – TRANSIÇÃO.....	52
7 ARTEFATOS.....	56
8 PAPÉIS.....	58
9 SCRUM.....	63
10 SERVIÇOS DE APOIO AO DESENVOLVIMENTO DE APLICAÇÕES.....	71
11 GLOSSÁRIO.....	77
12 REFERÊNCIAS BIBLIOGRÁFICAS.....	86

# 1 INTRODUÇÃO

A CGTI, na intenção de estar sempre atento a prestar um bom serviço a seus usuários e a investir na sua organização interna, oferece a Metodologia de Desenvolvimento de Software – MDS que tem como objetivo ser um instrumento de auxílio às equipes de desenvolvimento e manutenção de software da Coordenação Geral de Tecnologia da Informação.

Com esta metodologia, a COPSI (Coordenação de Projetos e Sistemas) terá insumos para orientar a equipe técnica a realizar suas atividades de maneira simples, estruturada e padronizada.

Elaborada em consonância com o Project Management Body of Knowledge - PMBOK adotado pela CGTI, a MDS é focada no Processo de Engenharia de Software, é baseada nos conceitos do Processo Unificado da Rational, é integrada com trâmites de gerenciamento de projetos, e inicia-se com conceitos de método ágil.

O nível de utilização da metodologia junto às equipes dependerá de alguns fatores, sendo eles: necessidade do negócio, tamanho do projeto etc. Portanto, alguns processos e artefatos especificados como obrigatórios poderão ser definidos como opcionais, de acordo com realidade de cada projeto.

Lembramos que o trabalho deve ser continuamente evoluído e atualizado e que estará disponível para uso de todos os envolvidos em desenvolvimento de sistemas no MTPA.

## 2 COORDENAÇÃO DE PROJETOS E SISTEMAS - COPSI

À Coordenação de Projetos e Sistemas - **COPSI** compete:

I - planejar, coordenar, assegurar, executar e controlar as atividades relacionadas com o planejamento estratégico e governança de tecnologia da informação e comunicações, o desenvolvimento e a manutenção de sistemas de informação, a administração de dados e de recursos de informática e a auditoria de sistemas no âmbito do Ministério dos Transportes, em consonância com as orientações, normas e diretrizes emanadas do órgão central do Sistema de Administração de Recursos de Informação e Informática - **SISP**;

II - propor diretrizes e normas para orientar o processo de alocação de bens e serviços de tecnologia da informação e comunicações, aquisição de hardware e software e contratação de prestação de serviços especializados em tecnologia da informação e comunicações;

III - gerenciar, em conjunto com a Coordenação-Geral de Recursos Logísticos, a execução dos contratos e convênios de prestação de serviços de informática;

IV - planejar, em conjunto com a Coordenação-Geral de Gestão de Pessoas, o treinamento de recursos humanos envolvidos nos projetos de modernização tecnológica;

V - manter o acervo das bases de dados necessários ao sistema de informações em transportes e aos sistemas internos de gestão institucional;

VI - assegurar aos gestores e aos órgãos congêneres o acesso aos serviços de tecnologia da informação e bases de dados mantidas pelo Ministério;

VII - coordenar e implementar as ações de segurança da informação definidas pelo Comitê Gestor de Segurança da Informação; e

VIII - coordenar a gestão de contratos de tecnologia da informação e comunicações.

## **3 SOBRE A METODOLOGIA DE DESENVOLVIMENTO DE SOFTWARE**

A Metodologia de Desenvolvimento de Software - MDS é um conjunto de boas práticas em desenvolvimento de sistemas que serão utilizadas pelas equipes de desenvolvimento e manutenção de softwares do MTPA.

A elaboração da metodologia é uma iniciativa da Coordenação de Projetos e Sistemas, precedida de um estudo da evolução dos processos, artefatos e orientações existentes no Processo de Gerenciamento e Desenvolvimento de Sistemas (PGDS), nas metodologias apresentadas no mercado e disponível às equipes de desenvolvimento de sistemas do MTPA.

A utilização permitirá às equipes da Coordenação de Projetos e Sistemas padronizar a forma de desenvolver software, alinhando os processos de trabalho e criando a documentação adequada. Com isto, obtém-se a melhoria na governança, o que reflete em aumento da eficácia, eficiência, efetividade, produtividade e menor dependência dos indivíduos, aumentando a retenção do conhecimento na organização.

### **3.1 ESTRUTURA GERAL**

A MDS está baseada no processo RUP (Rational Unified Process), que é um conjunto de boas práticas coletadas de engenharia de software. O RUP utiliza o processo Iterativo, que é a sequência de passos incrementais, isto é, a execução de um ciclo de vida de um projeto de sistemas tem como resultado uma versão do sistema liberada interna ou externamente. Ou seja, cada nova versão de sistema entregue à área requisitante poderá receber melhorias de forma incremental em relação à versão anterior. A MDS está desenhada para adotar padrões próprios, como: artefatos, formulários, modelos e instruções utilizadas ao longo do ciclo de vida dos projetos de desenvolvimento de sistemas.

Ainda está prevista na MDS a migração gradual para a Metodologia Ágil, utilizando SCRUM.

#### **3.1.1 FASES**

Do ponto de vista do gerenciamento, o ciclo de vida do software do RUP é dividido em 4 fases sequenciais, cada uma concluída por um marco principal, ou seja, cada fase é basicamente um intervalo de tempo entre dois marcos principais. A cada final de fase, uma avaliação é executada para determinar se os objetivos da fase foram alcançados. Uma avaliação satisfatória permite que o projeto passe para a próxima fase.

Cada fase possui suas próprias metas, seu próprio estilo de iteração e geralmente customiza suas tarefas e produtos de trabalho de forma diferente.

A fase Proposta de Projeto, que precede a aprovação do projeto, faz parte do processo de Iniciação da PMBOK (Project Management Body of Knowledge). Está demonstrada na MDS devido à participação significativa da área de desenvolvimento. Esta fase será usada apenas para projetos, não para as manutenções.

As demais fases são usadas para as demandas de projeto e para as manutenções. Nas demandas de projeto, as fases 02 a 05 representam cada iteração, portanto a Iniciação define o escopo da iteração; nas demandas de manutenção, a fase Iniciação detalha o escopo da demanda.

## Proposta de Projeto

Esta fase abrange o entendimento do problema e a identificação de suas necessidades. A partir dessas informações, é definido o escopo inicial e é estimado o tamanho funcional do projeto, cujo objetivo é obter aprovação de todos os *Stakeholders* sobre a continuidade do projeto.

Marco: Insumo para análise de viabilidade do projeto.

- Iniciação

Nesta fase o projeto já está aprovado. Os objetivos da iniciação são: definir o escopo e a visão geral da iteração ou demanda; realizar o planejamento do projeto; definir a arquitetura que será adotada com base nas diretrizes do órgão. Os envolvidos sobre os objetivos do ciclo de vida do projeto entram em consenso e a equipe técnica é definida.

A iniciação tem muita importância principalmente para os esforços dos desenvolvimentos novos, nos quais há muitos riscos de negócio e de requisitos que devem ser



tratados para que o projeto possa prosseguir adequadamente. Para objetos que visam manutenções em um sistema, a fase de iniciação é mais rápida, mas ainda se concentra em assegurar que a demanda seja viável.

Marco: Escopo da iteração ou demanda.

- **Elaboração**

Esta fase abrange o detalhamento da solução e a transformação dos requisitos do produto em casos de uso que darão suporte à atualização do documento arquitetural e a estimativa do esforço, de acordo com a métrica contratada. Com base nas informações novas obtidas durante a fase é possível que a visão sistêmica seja refinada, estabelecendo uma compreensão sólida dos casos de uso mais críticos que conduzem as decisões de arquitetura e planejamento da(s) próxima(s) iteração(ões). Toda documentação de análise é gerada nesta fase.

Marco: Detalhamento da solução a ser construída ou alterada.

- **Construção**

Esta fase abrange o desenvolvimento do sistema com base nos requisitos especificados e na arquitetura definida. Ela está essencialmente relacionada a projeto, programação, teste e homologação do sistema. As partes do sistema são desenvolvidas paralelamente e integradas durante esta fase. Diversas atividades de testes também são executadas a fim de se validar o produto de software, testando as funcionalidades de cada módulo. Ao final deve-se ter um produto em funcionamento pronto para ser liberado para o gestor.

Marco: Sistema funcionando em ambiente de homologação.

- **Transição**

Esta fase abrange a execução do plano de implantação, a finalização e disponibilização do material de suporte operacional, o treinamento de usuários e a entrega do produto para os usuários finais.

Marco: Sistema implantado e funcionando em ambiente de produção.

### 3.1.2 ELEMENTOS BÁSICOS

Além das fases, a MDS está estruturada com base em 4 elementos básicos, que representam “quem” faz “o que”, “como” e “quando”:

- Papéis (quem) - Um papel define as responsabilidades de um profissional ou grupo de profissionais que participam do desenvolvimento do projeto. As responsabilidades normalmente estão associadas aos artefatos que cada papel deve produzir e manter ao longo das atividades que realiza. Na prática, um mesmo papel pode ser desempenhado por mais de uma pessoa, assim como uma mesma pessoa pode assumir vários papéis ao longo do projeto.

- Artefatos (o que) - Em sentido amplo, o termo artefato representa um elemento tangível de um projeto. Podem assumir várias formas como: modelo, documento, código fonte ou executável. A MDS não inclui todos os artefatos de um projeto de desenvolvimento, mas todos os artefatos obrigatórios descritos na MDS devem ser elaborados ao longo do projeto. A MDS disponibiliza modelos (*templates*) para os seus artefatos, com o objetivo de orientar e facilitar a sua elaboração.

- Atividades (como) - Uma atividade na MDS representa um conjunto de passos e tarefas que um profissional, que desempenha o papel responsável por aquela atividade, deve executar para gerar algum resultado. As atividades envolvem a produção e modificação de artefatos do projeto.

- Fases (quando) - As fases da MDS apresentam a sequência e a dependência entre as atividades do projeto ao longo do tempo. As atividades no fluxo são divididas em fases do ciclo de vida do projeto e nos papéis responsáveis pela execução de cada uma.

## 3.2 CLASSIFICAÇÃO

A Metodologia deverá ser utilizada para os seguintes casos:



### 3.2.1 PROJETO

Consiste no desenvolvimento de novos sistemas ou módulos de sistemas que darão apoio aos programas do Ministério dos Transportes, Portos e Aviação Civil. Para o desenvolvimento, utiliza-se o processo de engenharia de software de modo a garantir que o produto seja de qualidade e que atenda às necessidades dos usuários. No âmbito do MTPA, consideram-se os seguintes tipos de projetos:

- **Projeto de Desenvolvimento**

Consiste no esforço necessário para o atendimento de uma demanda do tipo “novo sistema”, ou seja, a criação de uma nova aplicação para atender às necessidades de negócio dos gestores.

Os projetos de desenvolvimento podem se enquadrar em várias modalidades, que impactam no fluxo e artefatos utilizados:

- » Desenvolvimento padrão;
- » Desenvolvimento usando ferramentas, como exemplos: SharePoint, IBM Máximo, ferramenta de automação de processos;
- » Desenvolvimento de Aplicação de Tabulação, como exemplos: TabNet, Tabwin.

- **Projeto de Melhoria**

Consiste no esforço necessário para o atendimento de uma demanda de manutenção para evolução de um sistema já existente. Normalmente os projetos de melhoria estão associados a demandas evolutivas ou adaptativas, com tamanho funcional significativo e/ou alta criticidade para o negócio e/ou complexidade de desenvolvimento.

### 3.2.2 MANUTENÇÃO

A manutenção de sistemas ocorre após a entrada do sistema em produção ou, ainda, durante sua homologação, quando houver mudança de escopo. Pode ser um processo que envolve mudanças para corrigir erros, uma melhoria negocial e/ou melhoria técnica para otimização de um sistema. Esta manutenção inclui um grupo de atividades que são executadas durante o ciclo de vida da aplicação.

- **Manutenção Corretiva**

Mudanças no sistema (versão de produção) para corrigir defeitos e/ou deficiências que foram encontrados durante a utilização pelo usuário final. Não envolve mudanças nas funcionalidades de negócio, mas assegura que cada funcionalidade existente seja executada conforme requerido. Existem duas categorias para manutenção corretiva que seguem um fluxo diferenciado dentro do processo. São elas: as manutenções dentro do prazo de garantia e fora do prazo de garantia.

- **Manutenção Evolutiva**

A evolução (melhoria) de sistemas visa implementar novas funcionalidades, adequar funcionalidades existentes ou excluir funcionalidades, buscando melhorar sua aplicabilidade e usabilidade dentro da organização.

Normalmente, manutenções evolutivas têm fases de Iniciação e Elaboração menores, pois a definição e a arquitetura básica do produto foram determinadas pelos ciclos de desenvolvimento anteriores.

- **Manutenção Adaptativa**

Adequação do sistema às mudanças de ambiente operacional e/ou infraestrutura, compreendendo hardware e software básico, mudanças de versão, linguagem, SGBD e ajustes de performance ou de segurança, que não impliquem inserção, alteração ou exclusão de funcionalidades.

### **3.3 CRITÉRIO DE CLASSIFICAÇÃO DA DEMANDA**

A decisão de tratar as demandas como projeto ou manutenção de sistemas se dará de acordo com a necessidade de acompanhamento gerencial. Esta decisão será tomada em conjunto pela Coordenação Geral de Tecnologia da Informação – CGTI/MTPA e a Coordenação de Projetos e Sistemas - COPSI, com referência na metodologia PMBOK.

### **3.4 AUTORES**

A MDS foi elaborada por um Grupo de Trabalho que buscou apoio de Analistas de Negócio, Analistas de Requisito, Analistas de Sistema, Analistas de Métrica, Analistas de Teste, Arquitetos, Desenvolvedores, Administradores de Dados e Gerentes de Projeto e outros profissionais envolvidos nas diversas etapas do desenvolvimento de um software.

## **4 ORGANIZAÇÃO DO DOCUMENTO**

Para melhor leitura e organização, o documento obedece à divisão abaixo.

» Parte I – Processos: Apresenta os grupos de processos e o detalhe de cada tarefa com seu objetivo, entrada, descrição das atividades, saída, ferramenta e ator. Alguns processos possuem seus respectivos guias de referência.

» Parte II – Artefatos: Fornece os modelos de documentos utilizados nos processos da MDS. Esses documentos estão disponíveis no repositório da metodologia.

- » Parte III – Papéis: Descreve as atribuições de cada papel desempenhado no processo.
- » Parte IV – Scrum: Descreve o que é e como utilizar o Scrum.
- » Parte V – Serviços de Apoio ao Desenvolvimento de Aplicações
- » Parte VI – Glossário: Consta o glossário com a lista de termos e explicações relevantes sobre desenvolvimento de sistemas e uma lista de siglas utilizadas na metodologia.
- » Parte VII – Referências Bibliográficas: Indica a origem do conteúdo descrito no documento.

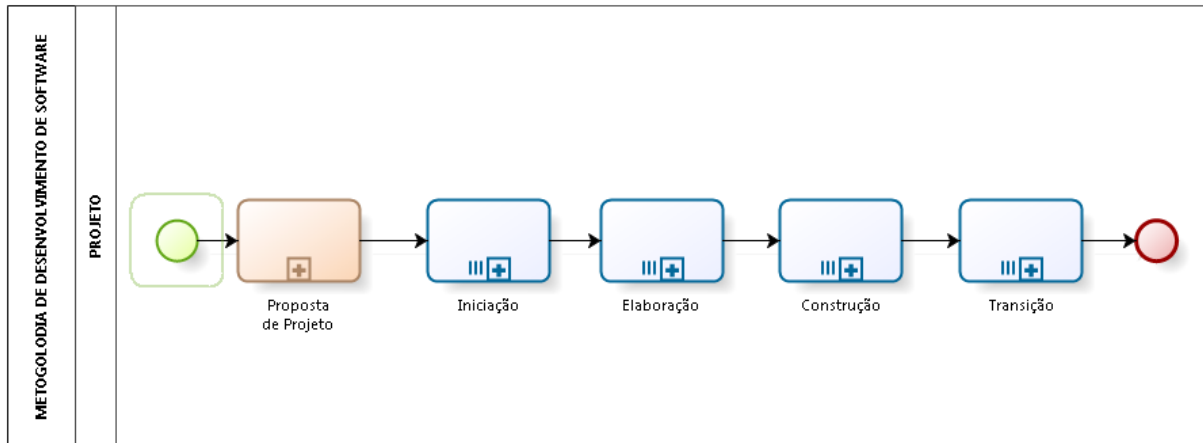
# PARTE I – PROCESSOS



## 5 PROCESSO: PROJETO

### 5.1 VISÃO GERAL – PROJETO

A visão geral das fases da metodologia está descrita a seguir:



Legenda:



- Subprocessos ou tarefas que demonstram processos que são executados por outra Metodologia.




- Subprocessos ou tarefas que demonstram processos que são executados dentro da MDS.

#### 5.1.1 PROPOSTA DE PROJETO


Proposta de Projeto	
 Proposta de Projeto	Esta fase abrange o entendimento do problema e a identificação de suas necessidades. A partir dessas informações, são definidos o escopo e o tamanho funcional do projeto, cujo objetivo é obter aprovação de todos os Stakeholders sobre a continuidade do projeto. <b>Marco:</b> Insumo para análise de viabilidade do projeto.




## 5.1.2 INICIAÇÃO

Iniciação	
 Iniciação	<p>Nesta fase o projeto já está aprovado. Os objetivos da iniciação são: definir o escopo e a visão geral da iteração ou demanda; realizar o planejamento do projeto; definir a arquitetura que será adotada com base nas diretrizes do órgão. Os envolvidos sobre os objetivos do ciclo de vida do projeto entram em consenso e a equipe técnica é definida. A iniciação tem muita importância principalmente para os esforços dos desenvolvimentos novos, nos quais há muitos riscos de negócio e de requisitos que devem ser tratados para que o projeto possa prosseguir adequadamente. Para objetos que visam manutenções em um sistema, a fase de iniciação é mais rápida, mas ainda se concentra em assegurar que o projeto ou demanda seja viável.</p> <p><b>Marco:</b> Escopo da iteração ou demanda.</p>


## 5.1.3 ELABORAÇÃO

Elaboração	
 Elaboração	<p>Esta fase abrange o detalhamento da solução e a transformação dos requisitos do produto em casos de uso que darão suporte à atualização do documento arquitetural e a estimativa do esforço, de acordo com a métrica contratada. Com base nas informações novas obtidas durante a fase é possível que a visão sistêmica seja refinada, estabelecendo uma compreensão sólida dos casos de uso mais críticos que conduzem as decisões de arquitetura e planejamento da(s) próxima(s) iteração(ões). Toda documentação de análise é gerada nesta fase.</p> <p><b>Marco:</b> Detalhamento da solução a ser construída ou alterada.</p>

## 5.1.4 CONSTRUÇÃO

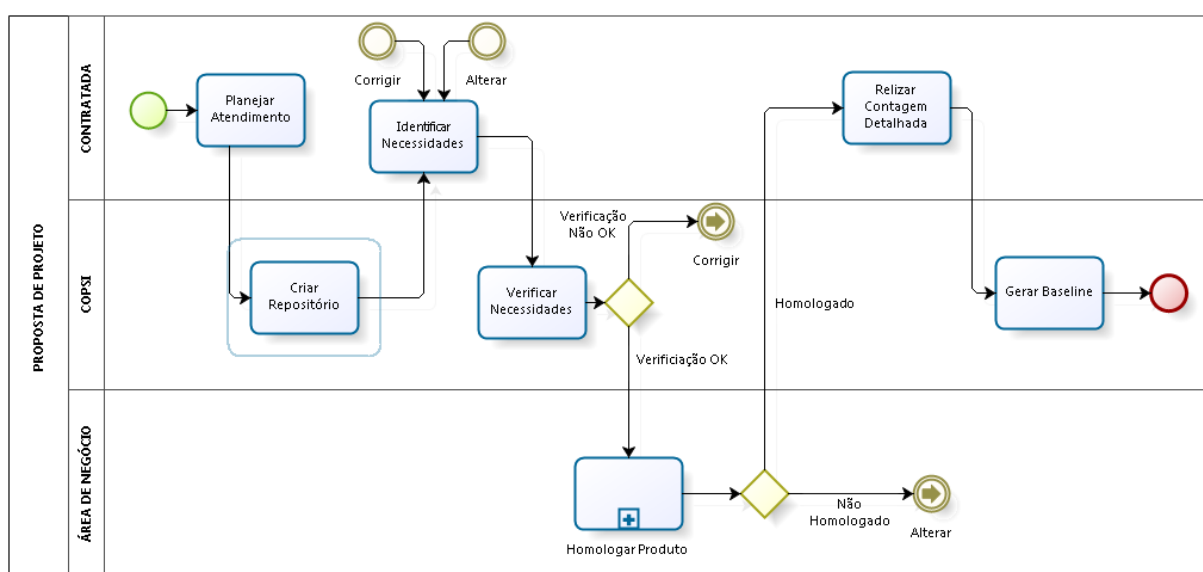
Construção	
 Construção	<p>Esta fase abrange o desenvolvimento do sistema com base nos requisitos especificados e na arquitetura definida. Ela está essencialmente relacionada a projeto, programação, teste e homologação do sistema.</p> <p>As partes do sistema são desenvolvidas paralelamente e integradas durante esta fase. Diversas atividades de testes também são executadas a fim de se validar o produto de software, testando as funcionalidades de cada módulo. Ao final deve-se ter um produto em funcionamento pronto para ser liberado para o gestor.</p> <p><b>Marco:</b> Sistema funcionando em ambiente de homologação.</p>

## 5.1.5 TRANSIÇÃO


Transição	
 <p>Transição</p>	<p>Esta fase abrange a execução do plano de implantação, a finalização e disponibilização do material de suporte operacional, o treinamento de usuários e a entrega do produto para os usuários finais.</p> <p><b>Marco:</b> Sistema implantado e funcionando em ambiente de produção.</p>

## 5.2 FASE – PROPOSTA DE PROJETO

As atividades da proposta de projetos serão descritas a seguir:




### 5.2.1 PLANEJAR ATENDIMENTO

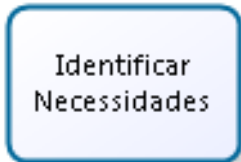
Planejar Atendimento	
 <p>Planejar Atendimento</p>	<p>Planejar na ferramenta de gestão de demanda a quantidade de horas ou o item do catálogo de serviços (UST) que será necessário para a execução das atividades da fase de proposta de projeto e encaminhar para aprovação do fiscal técnico.</p>
<p>Entradas:</p>	<p>Documento de Cadastro de Iniciativa (DCI); Projeto cadastrado no sistema de gestão de demandas.</p>

Descrição das Atividades:	Informar a quantidade de horas ou UST necessárias para execução da demanda de proposta de projeto; Vincular os recursos humanos que irão atuar na demanda; Analisar a necessidade de mapeamento de processo; Encaminhar o planejamento para aprovação do fiscal técnico por meio do sistema de gestão de demanda. O fiscal técnico poderá: aprovar, rejeitar para ajustes ou cancelar a demanda.
Saídas:	Demanda de proposta de projeto planejada.
Ferramentas:	Sistemas de Gestão de Demandas
Ator:	Gerente do Projeto

### 5.2.2 CRIAR REPOSITÓRIO

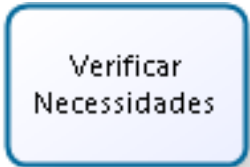
Criar Repositório	
	Disponibilizar o repositório e planejar as atividades de gerência de configuração para manter a integridade dos produtos de trabalho.
Entradas:	Solicitação do Escritório de Projetos para criação do repositório.
Descrição das Atividades:	Realizar reunião com gerente de projeto e consultor de projeto; Elaborar o Plano de Gerenciamento de Configuração; Criar repositório do projeto; Aplicar as regras de acesso de acordo com os padrões definidos no documento de Controle de Baselines e Branches (CBB); Comunicar o solicitante quanto à criação do ambiente.
Saídas:	Plano de Gerenciamento de Configuração; Controle de Baseline e Branches (CBB); Repositório criado.
Ferramentas:	Sistema de Controle de Versão; Sistema de Gestão de Demanda.
Ator:	Gerente de Configuração de Software.

### 5.2.3 IDENTIFICAR NECESSIDADES


Identificar Necessidades	
	Elicitar, documentar e validar as necessidades do negócio, bem como os problemas.
Entradas:	Diagrama de Processo; Documento de Cadastro de Iniciativa (DCI).

Descrição das Atividades:	Reuniões entre o gestor, analista de requisito e consultor de projeto para entendimento do problema e as necessidades; Especificar as características funcionais e não funcionais; Registro das atas de reunião; Reuniões com equipe de arquitetura para alinhamento das características não funcionais; Capturar vocabulário comum.
Saídas:	Documento de Visão de Negócio; Glossário; Ata de Reunião.
Ferramentas:	N/A.
Ator:	Analista de Requisito.

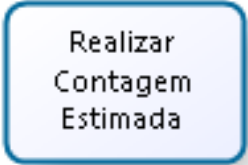
#### 5.2.4 VERIFICAR NECESSIDADES

Verificar Necessidades	
	Verificar e avaliar o documento de visão de negócio de acordo com os critérios de qualidade: Não ambíguo, Correto, Consistente, Verificável, Realizável, Completo e Compreensível.
Entradas:	Documento de Visão de Negócio.
Descrição das Atividades:	Analisar a qualidade do documento levando em consideração as informações de valor para o projeto.
Saídas:	Checklist de Negócio.
Ferramentas:	N/A.
Ator:	Analista de Qualidade.

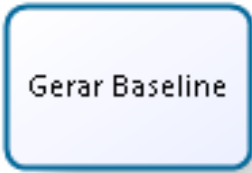
#### 5.2.5 HOMOLOGAR PRODUTO

Homologar Produto	
 Homologar Produto	Obter uma estimativa inicial do tamanho do projeto com base no Documento de Visão de Negócio.

## 5.2.6 REALIZAR CONTAGEM ESTIMADA

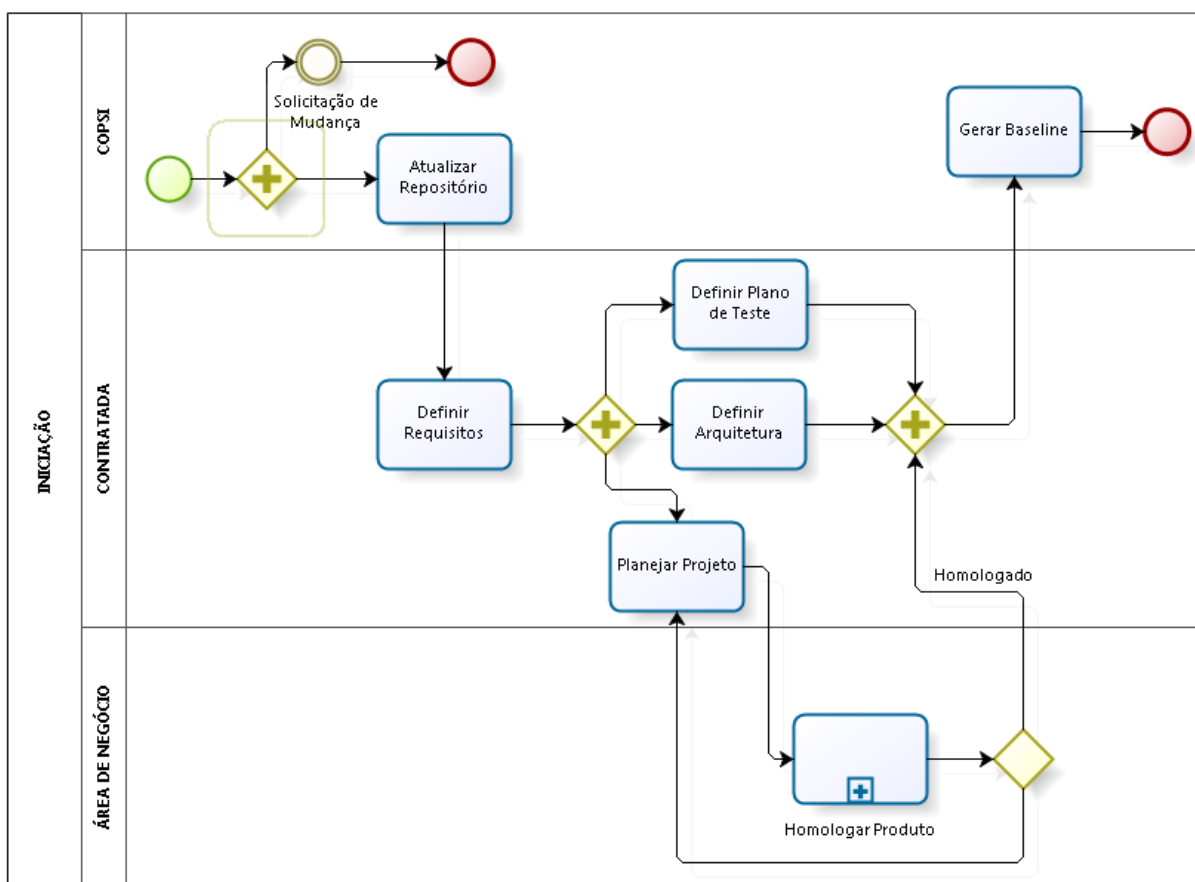
Realizar Contagem Estimada	
	Obter uma estimativa inicial do tamanho do projeto com base no Documento de Visão de Negócio.
Entradas:	Documento de Visão de Negócio.
Descrição das Atividades:	Realizar uma estimativa inicial do tamanho do software, utilizando a métrica adotada no contrato.
Saídas:	Contagem Estimada.
Ferramentas:	N/A.
Ator:	Analista de Métrica.

## 5.2.7 GERAR BASELINE

Gerar Baseline	
	Gerar uma versão estável e congelada dos artefatos quando estes estiverem maduros, mantendo a integridade dos artefatos do projeto permitindo que mudanças sejam aplicadas somente através de um processo formal.
Entradas:	Solicitação para geração da baseline.
Descrição das Atividades:	Criar baseline referente aos artefatos indicados no Plano de Gerenciamento e Configuração.
Saídas:	Baseline gerada.
Ferramentas:	Sistema de controle de versão.
Ator:	Gerente de Configuração de Software.

## 5.3 FASE – INICIAÇÃO

As atividades da fase de iniciação para projeto serão descritas a seguir:



### 5.3.1 SOLICITAÇÃO DE MUDANÇA


A qualquer momento, pode haver necessidade de se realizar alguma alteração nos requisitos. Isso acontecendo, deverá ser seguido o fluxo 6.7 deste documento.

### 5.3.2 ATUALIZAR REPOSITÓRIO

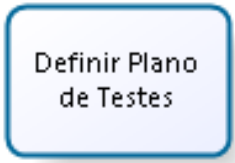
Atualizar Repositório	
	Atualizar o repositório e planejar as atividades de gerencia de configuração para manter a integridade dos produtos de trabalho.
Entradas:	Reunião com Gerente de Projeto.

Descrição das Atividades:	Atualizar o Plano de Gerenciamento de Configuração; Atualizar o repositório do projeto; Aplicar as regras de acesso de acordo com os padrões definidos no documento de Controle de Baselines e branches (CBB); Comunicar o solicitante quanto à criação da Branche.
Saídas:	Plano de Gerenciamento de Configuração atualizado; Controle de Baseline e Branches (CBB) atualizado; Repositório atualizado.
Ferramentas:	Sistema de Controle de Versão; Sistema de Gestão de Demanda.
Ator:	Gerente de Configuração de Software.

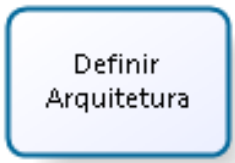
### 5.3.3 DEFINIR REQUISITOS

Definir Requisitos	
	Elicitar, documentar e validar os requisitos funcionais e não funcionais do sistema.
Entradas:	Documento de Visão de Negócio.
Descrição das Atividades:	Reuniões com gestor para definir o escopo e não escopo das iterações; Levantamento dos envolvidos no projeto; Definir os requisitos funcionais e não funcionais; Registro das atas de reunião; Atualizar vocabulário se necessário.
Saídas:	Documento de Visão de Sistema; Glossário atualizado; Ata de Reunião.
Ferramentas:	N/A.
Ator:	Analista de Requisito.


### 5.3.4 DEFINIR PLANO DE TESTE

Definir Plano de Teste	
	Definir uma estratégia de testes para o projeto com base na avaliação das necessidades, objetivos, abordagens e infraestrutura mais adequada para o projeto.
Entradas:	Documento de Visão de Negócio; Documento de Visão de Sistema.
Descrição das Atividades:	Elaborar Plano de Teste.
Saídas:	Plano de Teste.
Ferramentas:	N/A.
Ator:	Gerente de Teste.

### 5.3.5 ARQUITETURA

Definir Arquitetura	
	Registrar a visão geral arquitetural abrangente do sistema. O objetivo deste documento é capturar e comunicar as decisões arquiteturais significativas que foram tomadas em relação ao sistema.
Entradas:	Documento de Visão de Negócio; Documento de Visão de Sistema.
Descrição das Atividades:	Definir uma sugestão de arquitetura com base nas diretrizes do órgão; Elaborar provas de conceito; Avaliar perfil técnico necessário para construção; Avaliar disponibilidade de infra-estrutura.
Saídas:	Documento de Definição Arquitetural.
Ferramentas:	N/A.
Ator:	Arquiteto de Software.


### 5.3.6 PLANEJAR PROJETO

Planejar Projeto	
	Elaborar e consolidar o plano de projeto e seus planos auxiliares a fim de manter a descrição de como o projeto será executado, monitorado e controlado.

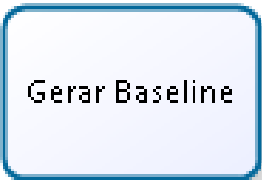


Entradas:	Documento de Visão de Negócio; Documento de Visão de Sistema; Termo de Abertura do Projeto.
Descrição das Atividades:	Desenvolver a EAP para fornecer uma visão estruturada das entregas e dos pacotes de trabalho; Planejar a estratégica do gerenciamento dos riscos; Estimar atividades; Definir a equipe; Planejar comunicação; Desenvolver cronograma; Elaborar o caso de desenvolvimento; Elaborar o plano de iteração.
Saídas:	Plano de Projeto; Caso de Desenvolvimento; Plano de Iteração.
Ferramentas:	Sistema de Gestão de Demanda.
Ator:	Gerente de Projeto.

### 5.3.7 HOMOLOGAR PRODUTO

Homologar Produto	
 <p>Homologar Produto</p>	<p>Realizar a homologação do Plano de Projeto com as parte interessadas para aprovação do cronograma, custo, prazo e entregas.</p> <p>Atividade descrita na Project Management Body of Knowledge (PMBOK).</p>

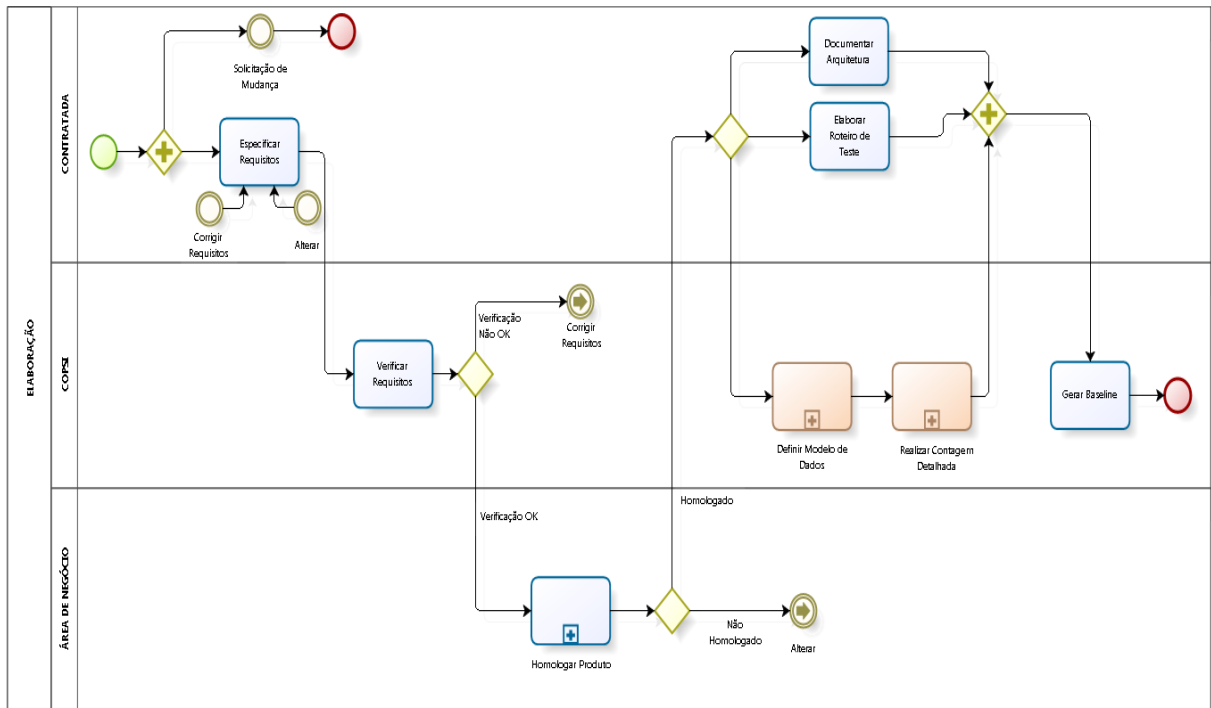
### 5.3.8 GERAR BASELINE

Gerar Baseline	
 <p>Gerar Baseline</p>	<p>Gerar uma versão estável e congelada dos artefatos quando estes estiverem maduros, mantendo a integridade dos artefatos do projeto permitindo que mudanças sejam aplicadas somente através de um processo formal.</p>
Entradas:	Solicitação para geração da baseline.
Descrição das Atividades:	Criar baseline referente aos artefatos indicado no Plano de Gerenciamento e Configuração.

Saídas:	Baseline gerada.
Ferramentas:	Sistema de controle de versão.
Ator:	Gerente de Configuração de Software.

## 5.4 FASE – ELABORAÇÃO

As atividades da fase de elaboração para projeto serão descritas a seguir:



### 5.4.1 SOLICITAÇÃO DE MUDANÇA

A qualquer momento, pode haver necessidade de se realizar alguma alteração nos requisitos. Isso acontecendo, deverá ser seguido o fluxo 6.7 deste documento.

### 5.4.2 ESPECIFICAR REQUISITOS


Especificar Requisitos	
	<p>Descrever um ou mais requisitos (funcionais e não funcionais) com detalhe suficiente para compreensão do requisito, assegurar concorrência com as expectativas dos Stakeholders e permitir o início do desenvolvimento do software.</p> <p>Cada instância dessa atividade representa o detalhamento de um pacote de casos de uso.</p>
Entradas:	<p>Reuniões com o Gestor do Sistema;  Documento de Visão de Negócio;  Documento de Visão de Sistema;  Diagrama de Processo.</p>

Descrição das Atividades:	<p>Especificar os casos de uso de acordo com os requisitos funcionais;</p> <p>Elaborar modelo de casos de uso;</p> <p>Levantar as regras de negócio e apresentação;</p> <p>Definir as mensagens do sistema;</p> <p>Atualizar o Glossário;</p> <p>Reuniões com equipe de arquitetura para definição dos casos de uso mais complexos;</p> <p>Reunião com equipe de Usabilidade para apoio na construção das telas do sistema.</p>
Saídas:	<p>Especificação de Caso de Uso;</p> <p>Modelo de Caso de Uso;</p> <p>Documento de Regras;</p> <p>Lista de Mensagem;</p> <p>Matriz de rastreabilidade;</p> <p>Glossário atualizado.</p>
Ferramentas:	<p>Ferramenta de modelagem UML;</p> <p>Ferramenta de Criação de Tela.</p>
Ator:	Analista de Requisito.


### 5.4.3 VERIFICAR REQUISITOS

Verificar Requisitos	
	<p>Verificar e avaliar o documento de caso de uso de acordo com os critérios de qualidade: Não ambíguo, Correto, Consistente, Verificável, Realizável, Completo e Compreensível.</p>
Entradas:	<p>Especificação de Caso de Uso;</p> <p>Modelo de Caso de Uso;</p> <p>Documento de Regras;</p> <p>Lista de Mensagem.</p>
Descrição das Atividades:	Verificar a conformidade dos requisitos.
Saídas:	Checklist de Requisitos.
Ferramentas:	N/A.
Ator:	Analista de Qualidade.


#### 5.4.4 HOMOLOGAR PRODUTO

Homologar Produto	
 <p>Homologar Produto</p>	<p>Realizar a homologação dos artefatos da fase de elaboração. Assinar os Termos de Recebimento Provisório e Definitivo.</p>

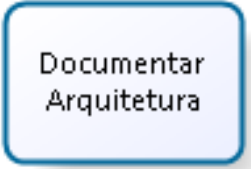
#### 5.4.5 DEFINIR MODELO DE DADOS

Definir Modelo de Dados	
 <p>Definir Modelo de Dados</p>	<p>Elaborar o modelo de dados de um projeto de acordo com a documentação produzida pela equipe de desenvolvimento. Processo descrito na Metodologia de Administração de Dados (MAD).</p>

#### 5.4.6 REALIZAR CONTAGEM DETALHADA

 <p>Realizar Contagem Detalhada</p>	<p>Realizar contagem detalhada em pontos de função após o fechamento da elaboração. Processo descrito na Roteiro de Métricas do MTPA.</p>
--	---

#### 5.4.7 DOCUMENTAR ARQUITETURA

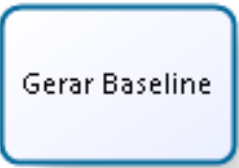
Documentar Arquitetura	
	<p>Definir os componentes de software, suas propriedades e seus relacionamentos com outros softwares. Analisar e priorizar os requisitos funcionais e não funcionais do ponto de vista arquitetural e identificar tecnologias atualmente em uso e novas tecnologias de produto. Analisar os casos de uso do sistema para identificar os elementos estruturais e comportamentais como insumo para a elaboração/definição da arquitetura do sistema.</p>
<p>Entradas:</p>	<p>Documento de Visão de Sistema; Especificação de caso de uso; Modelo de Casos de Uso; Documento de Regras.</p>

Descrição das Atividades:	<p>Descrever requisitos e restrições arquiteturais;</p> <p>Avaliar os componentes públicos quanto a sua possível aplicabilidade na arquitetura do software;</p> <p>Decomposição do sistema de software em subsistemas e componentes;</p> <p>Modelagem arquitetural;</p> <p>Restringir as técnicas de arquitetura a serem utilizadas no projeto, considerando a experiência obtida com projetos de domínios de problemas semelhantes.</p>
Saídas:	<p>Documento de Arquitetura de Software;</p> <p>Plano de Implantação;</p> <p>Mapa de Integração.</p>
Ferramentas:	N / A.
Ator:	Arquiteto de Software.

#### 5.4.8 ELABORAR ROTEIRO DE TESTE

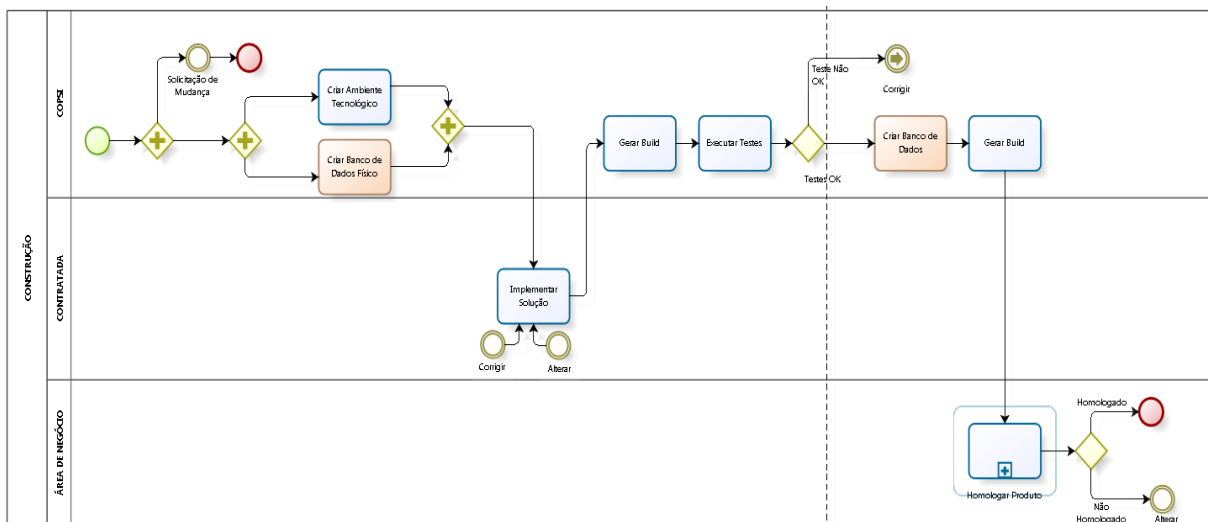
Elaborar Roteiro de Teste	
	Elaborar os roteiros de teste para viabilizar a execução dos testes.
Entradas:	<p>Plano de Teste;</p> <p>Especificação de caso de uso;</p> <p>Modelo de Casos de Uso;</p> <p>Documento de Regras;</p> <p>Lista de Mensagem.</p>
Descrição das Atividades:	Elaborar Roteiro de Teste baseados na especificação dos requisitos.
Saídas:	Roteiro de Teste.
Ferramentas:	Ferramenta de Gestão de Teste.
Ator:	Analista de Teste.

### 5.4.9 GERAR BASELINE

Gerar Baseline	
	Gerar uma versão estável e congelada dos artefatos quando estes estiverem maduros, mantendo a integridade dos artefatos do projeto permitindo que mudanças sejam aplicadas somente através de um processo formal.
Entradas:	Solicitação para geração da baseline.
Descrição das Atividades:	Criar baseline referente aos artefatos indicado no Plano de Gerenciamento e Configuração.
Saídas:	Baseline gerada.
Ferramentas:	Sistema de Controle de Versão.
Ator:	Gerente de Configuração de Software.

## 5.5 FASE – CONSTRUÇÃO

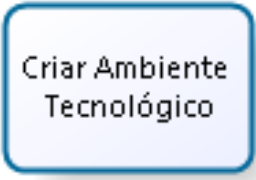
As atividades da fase de construção para projeto serão descritas a seguir:



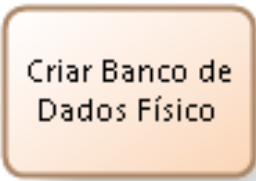
### 5.5.1 SOLICITAÇÃO DE MUDANÇA

A qualquer momento, pode haver necessidade de se realizar alguma alteração nos requisitos. Isso acontecendo, deverá ser seguido o fluxo 6.7 deste documento.

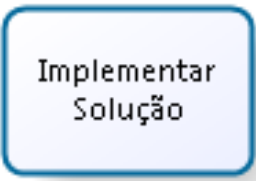
## 5.5.2 CRIAR AMBIENTE TECNOLÓGICO

Criar Ambiente Tecnológico	
	Atualizar as ferramentas de apoio para desenvolvimento ou manutenção dos sistemas.
Entradas:	Documento de Arquitetura de Software; Plano de Implantação.
Descrição das Atividades:	Atualizar o Plano de Gerenciamento de Configuração; Configurar ferramentas de apoio.
Saídas:	Plano de Gerenciamento de Configuração atualizado; Ambiente tecnológico criado.
Ferramentas:	N/A.
Ator:	Gerente de Configuração de Software.

## 5.5.3 CRIAR BANCO DE DADOS FÍSICO


Criar Banco de Dados Físico	
	Especificar o projeto do banco de dados definindo os modelos (conceitual, lógico e físico) do sistema, bem como, os requisitos e componentes arquiteturais de banco de dados de forma a garantir a persistência dos dados com consistência e eficácia. Atividade descrita na Metodologia de Administração de Dados (MAD).

## 5.5.4 IMPLEMENTAR SOLUÇÃO

Implementar Solução	
	Implementar os casos de uso pertencentes ao plano de iteração que está sendo executado, gerando componentes e executando os testes unitários. O Integrador deverá executar os procedimentos planejados para a integração e deploy do sistema de acordo com o ambiente de execução, aplicando os padrões de versionamento definidos no Plano de Gerenciamento de Configuração e as configurações fornecidas pelo Administrador de Banco de Dados.

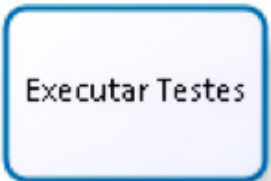
Entradas:	Especificação de Caso de Uso; Modelo de Casos de Uso; Documento de Regras; Lista de Mensagem; Documento de Arquitetura de Software; Modelo de Dados; Protótipo.
Descrição das Atividades:	Produzir código fonte dos casos de uso da iteração, implementando operações, estado, associações e atributos das classes de implementação; Fornecer feedback para o arquiteto se forem encontrados erros de design e execução do passo anterior; Avaliar o código, verificando se está adequado para a sua finalidade; Realizar testes na unicidade, executando os casos de testes unitários; Corrigir erros/defeitos encontrados nos testes.
Saídas:	Componentes implementados e testados.
Ferramentas:	N/A.
Ator:	Desenvolvedor.

### 5.5.5 GERAR BUILD


Gerar Build	
	Gerar a Build no ambiente de desenvolvimento para realização de testes no sistema implementado.
Entradas:	Solicitação para geração da build; Código Fonte.
Descrição das Atividades:	Marcar os códigos gerados que poderão ser testados.
Saídas:	Código fonte marcado.
Ferramentas:	Sistema de Controle de Versão.
Ator:	Analista de Configuração de Software.




### 5.5.6 EXECUTAR TESTES

Executar Testes	
	Realizar os testes da demanda implementada de acordo com a definição no Plano de Teste, registrar o comportamento da build e avaliar o nível de qualidade no projeto desenvolvido.
Entradas:	Roteiro de Teste; Código fonte no ambiente de desenvolvimento.
Descrição das Atividades:	Realizar testes nas funcionalidades.
Saídas:	Planilha de Resultado de Teste.
Ferramentas:	N/A.
Ator:	Analista de Teste.


### 5.5.7 CRIAR BANCO DE DADOS

Criar Banco de Dados	
	Criação do Banco de Dados no ambiente de Homologação. Atividade descrita na Metodologia de Administração de Dados (MAD).

### 5.5.8 GERAR BUILD

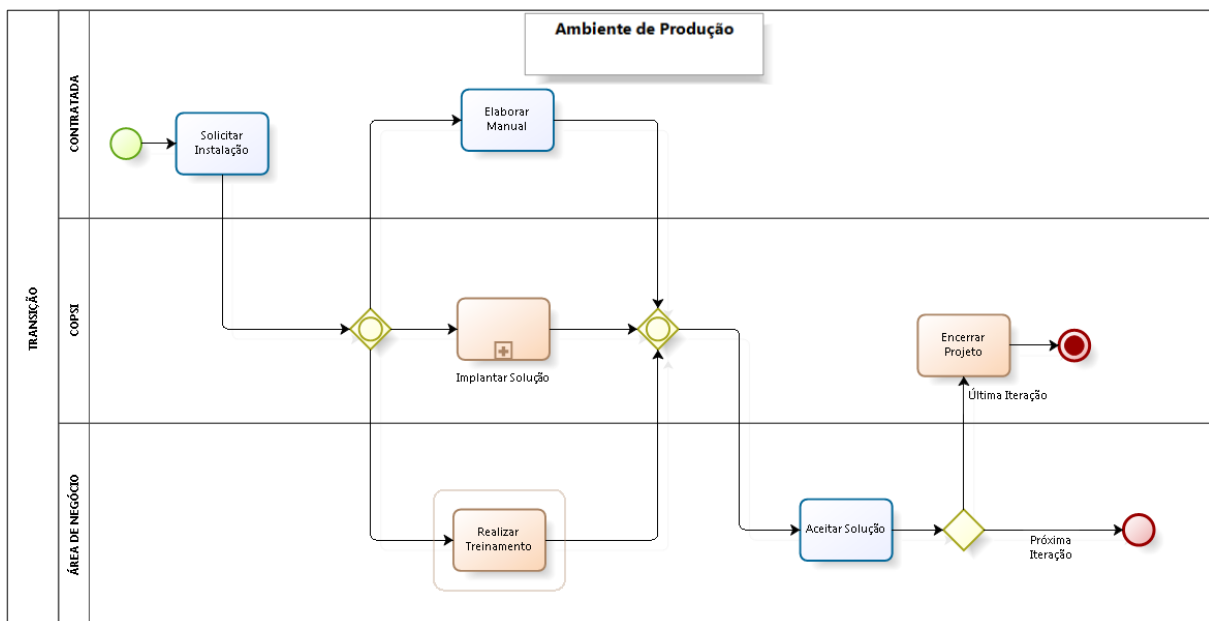
Gerar Build	
	Gerar a Build no ambiente de homologação para homologação da área de negócio.
Entradas:	Solicitação para geração da build; Código fonte testado.
Descrição das Atividades:	Marcar código fonte que foi testado.
Saídas:	Código fonte marcado para homologação.
Ferramentas:	Sistema de Controle de Versão.
Ator:	Analista de Configuração de Software.

## 5.5.9 HOMOLOGAR PRODUTO


Homologar Produto	
 <p>Homologar Produto</p>	<p>Realizar a homologação do produto entregue. Assinar os Termos de Recebimento Provisório e Definitivo.</p>

## 5.6 FASE – TRANSIÇÃO

As atividades da fase de transição para projeto serão descritas a seguir:

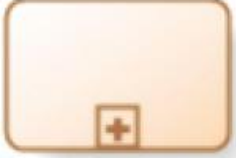


### 5.6.1 SOLICITAR INSTALAÇÃO

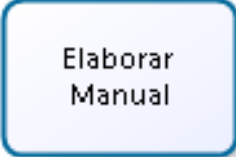
Solicitar Instalação	
 <p>Solicitar Instalação</p>	<p>Solicitação de instalação do sistema no ambiente de produção. Atividade descrita na Metodologia de Gerenciamento de Serviço de Infra-estrutura (MGSI).</p>
Entradas:	Termo de Recebimento Definitivo assinado.
Descrição das Atividades:	Preencher requisição de mudança.
Saídas:	Requisição de Mudança (RDM).
Ferramentas:	Ferramenta de gerenciamento de serviço de TI.

Ator:	Gerente de Projeto.
-------	---------------------


### 5.6.2 IMPLANTAR SOLUÇÃO

Implantar Solução	
 Implantar Solução	Realizar a implantação do produto homologado em ambiente de produção. Processo descrito na Metodologia de Gerenciamento de Serviço de Infra-estrutura (MGSI).

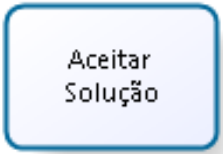
### 5.6.3 ELABORAR MANUAL

Elaborar Manual	
 Elaborar Manual	Elaborar manual de apoio para o uso das funcionalidades do sistema.
Entradas:	Documentação técnica do sistema; Solução no ambiente de homologação.
Descrição das Atividades:	Elaborar manual do usuário.
Saídas:	Manual do Usuário.
Ferramentas:	N/A.
Ator:	Redator Técnico.


### 5.6.4 REALIZAR TREINAMENTO

Realizar Treinamento	
 Realizar Treinamento	Realizar treinamento para os usuários do sistema.

## 5.6.5 ACEITAR SOLUÇÃO

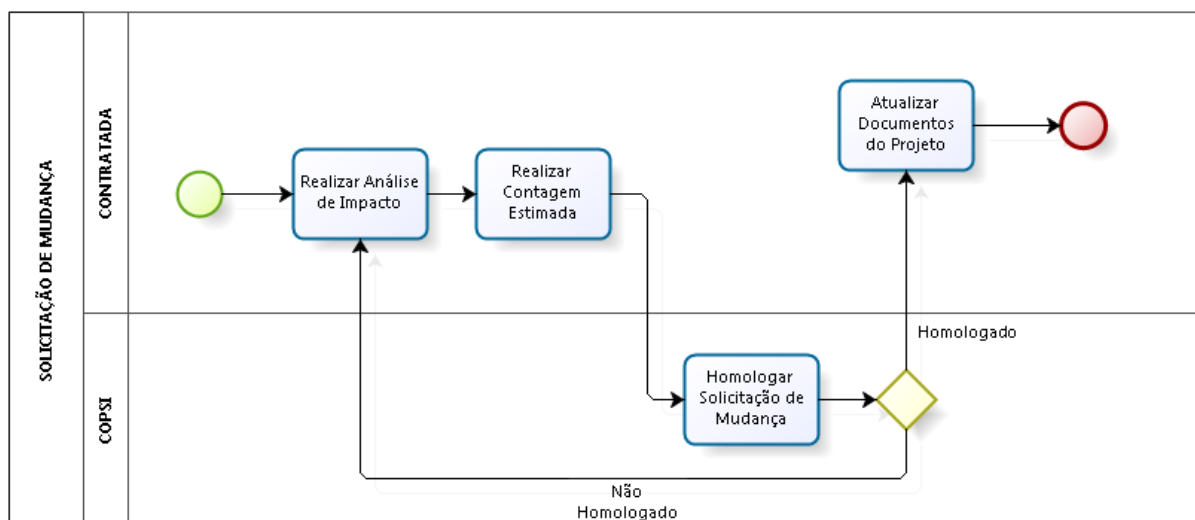
Aceitar Solução	
	Aprovar a solução instalada no ambiente de produção.
Entradas:	Código no ambiente de produção.
Descrição das Atividades:	Emitir termo de recebimento definitivo e aceite do produto.
Saídas:	Termo de Recebimento Definitivo; Termo de Aceite.
Ferramentas:	N/A.
Ator:	Gestor do Sistema.

## 5.6.6 ENCERRAR PROJETO

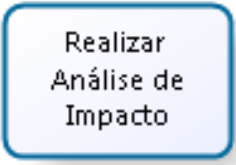
Encerrar Projeto	
	Registrar formalmente o encerramento do projeto após a finalização de todas as iterações. Atividade descrita na PMBOK (Project Management Body of Knowledge).

## 5.7 SOLICITAÇÃO DE MUDANÇA

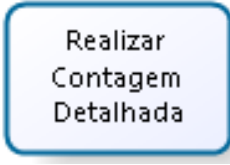
As atividades realizadas para uma solicitação de mudança serão descritas a seguir:



### 5.7.1 REALIZAR ANÁLISE DE IMPACTO

Realizar Análise de Impacto	
	Elicitar, documentar e validar a solicitação de mudança sugerida pelo Gestor ou identificada durante o projeto.
Entradas:	Solicitação de Mudança.
Descrição das Atividades:	Reuniões com gestor para entendimento da solicitação de mudança; Identificar as necessidades; Identificar as funcionalidades impactadas.
Saídas:	Documento de Análise de Impacto.
Ferramentas:	N/A.
Ator:	Analista de Requisito.

### 5.7.2 REALIZAR CONTAGEM ESTIMADA

Realizar Contagem Estimada	
	Obter uma estimativa do tamanho da mudança com base no Documento de Análise de Impacto.
Entradas:	Documento de Análise de Impacto.
Descrição das Atividades:	Realizar uma estimativa inicial do tamanho das funcionalidades impactadas, utilizando a métrica adotada no contrato.
Saídas:	Contagem Estimada.
Ferramentas:	N/A.
Ator:	Analista de Métrica.

### 5.7.3 HOMOLOGAR SOLICITAÇÃO DE MUDANÇA

Homologar Solicitação de Mudança
----------------------------------

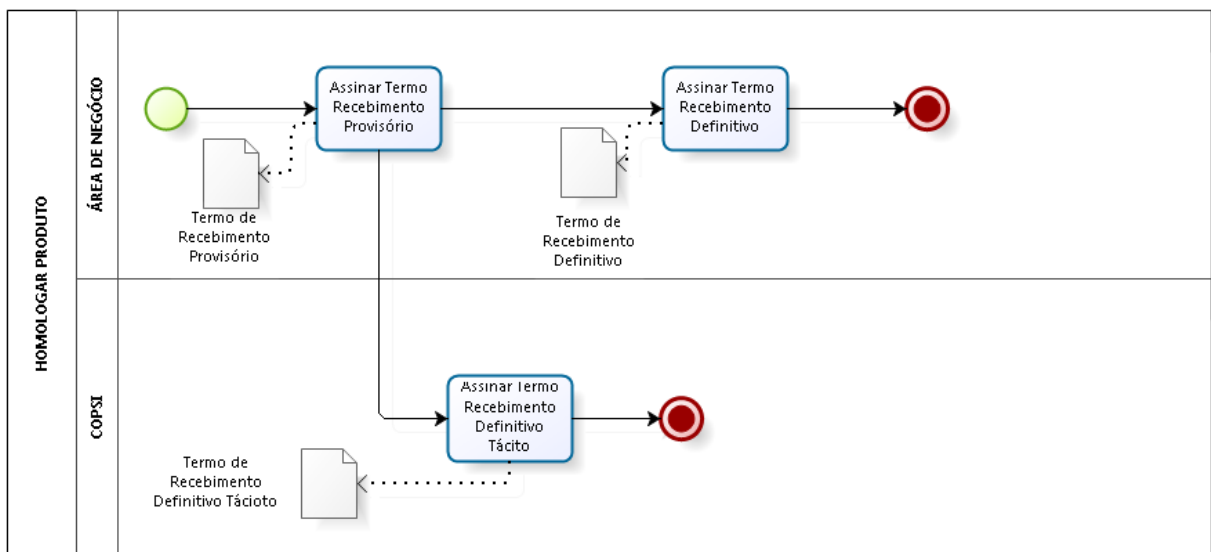
<div style="border: 1px solid blue; border-radius: 15px; padding: 10px; width: fit-content; margin: 0 auto;"> <p><b>Homologar Solicitação de Mudança</b></p> </div>	<p>Realizar a homologação da Solicitação de Mudança. Atividade descrita na PMBOK (Project Management Body of Knowledge).</p>
---	--

### 5.7.4 ATUALIZAR DOCUMENTAÇÃO DO PROJETO

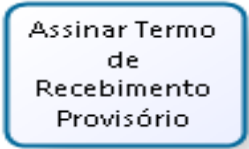
Atualizar Documentação do Projeto	
<div style="border: 1px solid blue; border-radius: 15px; padding: 10px; width: fit-content; margin: 0 auto;"> <p><b>Aualizar Documentos do Projeto</b></p> </div>	<p>Essa atualização deve basear-se na solicitação de mudança feita pelo Gestor o que pode resultar em uma atualização ou criação de novos documentos.</p>
Entradas:	Documento de Análise de Impacto.
Descrição das Atividades:	Atualizar documentos do projeto que tenham sido afetados pela solicitação de mudança.
Saídas:	Documentos atualizados.
Ferramentas:	N/A.
Ator:	Analista de Requisito, Gerente de Projeto, Analista de Teste.

## 5.8 HOMOLOGAR PRODUTO

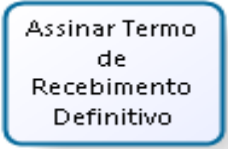
As atividades realizadas para uma homologação de um produto serão descritas a seguir:



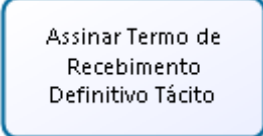
### 5.8.1 ASSINAR TERMO DE RECEBIMENTO PROVISÓRIO

Assinar Termo de Recebimento Provisório	
	<p>O termo de recebimento provisório é uma declaração formal de que os serviços foram prestados ou os bens foram entregues, para posterior análise das conformidades de qualidade baseadas nos critérios de aceitação.</p> <p>Após a entrega dos artefatos/produtos, o gestor terá um tempo determinado para análise e homologação.</p>
Entradas:	Artefatos e/ou produtos para homologação.
Descrição das Atividades:	Receber os artefatos e/ou produtos para a homologação.
Saídas:	Termo de Recebimento Provisório assinado.
Ferramentas:	N/A.
Ator:	Gestor do Sistema.

### 5.8.2 ASSINAR TERMO DE RECEBIMENTO DEFINITIVO

Assinar Termo de Recebimento Definitivo	
	<p>O termo de recebimento definitivo é uma declaração formal de que os serviços foram prestados atendem aos requisitos estabelecidos e definidos.</p> <p>Esse aceite poderá ser feito através de assinatura de documento ou através do sistema de gestão de demanda.</p>
Entradas:	Artefatos e/ou produtos homologados.
Descrição das Atividades:	Aceitar os artefatos e/ou produtos entregues para homologação.
Saídas:	Termo de Recebimento Definitivo assinado.
Ferramentas:	Sistema de Gestão de Demanda.
Ator:	Gestor do Sistema.

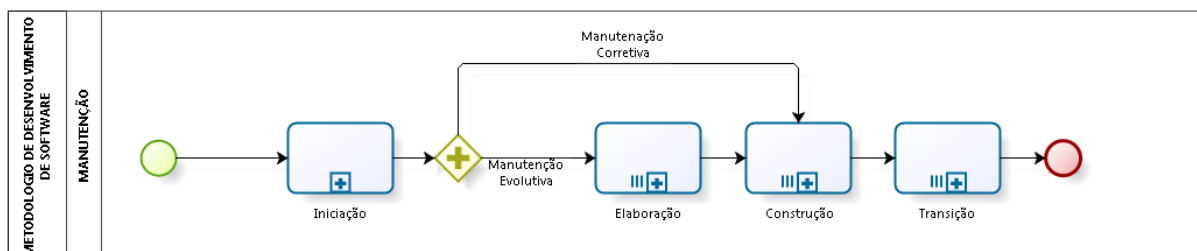
### 5.8.3 ASSINAR TERMO DE RECEBIMENTO DEFINITIVO TÁCITO

Assinar Termo de Recebimento Definitivo Tácito	
	O termo de recebimento definitivo tácito é assinado pelo coordenador de desenvolvimento da COPSI quando o gestor não realiza a homologação dentro do prazo estipulado.
Entradas:	Termo de Recebimento Definitivo não assinado.
Descrição das Atividades:	Realizar a homologação tácita; Comunicar ao Gestor sobre a homologação tácita.
Saídas:	Termo de Recebimento Definitivo Tácito assinado.
Ferramentas:	N/A.
Ator:	Coordenadores de Desenvolvimento.


## 6 PROCESSO – MANUTENÇÃO

### 6.1 VISÃO GERAL – MANUTENÇÃO

A visão geral das fases da metodologia será descrita a seguir:




#### 6.1.1 INICIAÇÃO


Iniciação	
	A fase de Iniciação em tempo de Manutenção é voltada para entender as novas necessidades do cliente. Com base nas informações levantadas nessa fase será possível identificar em qual tipo de Manutenção as necessidades se encaixam.




## 6.1.2 ELABORAÇÃO

Elaboração	
 Elaboração	<p>De acordo com o impacto identificado na fase anterior, a documentação e/ou a arquitetura serão atualizadas. Se na iniciação for identificadas funcionalidades novas nessa fase deverá ser criada toda a documentação para subsidiar a construção delas. É importante ressaltar que a fase de elaboração sempre se guia com base nas informações coletadas na Iniciação já que a fase de elaboração é a consolidação da iniciação.</p>

## 6.1.3 CONSTRUÇÃO

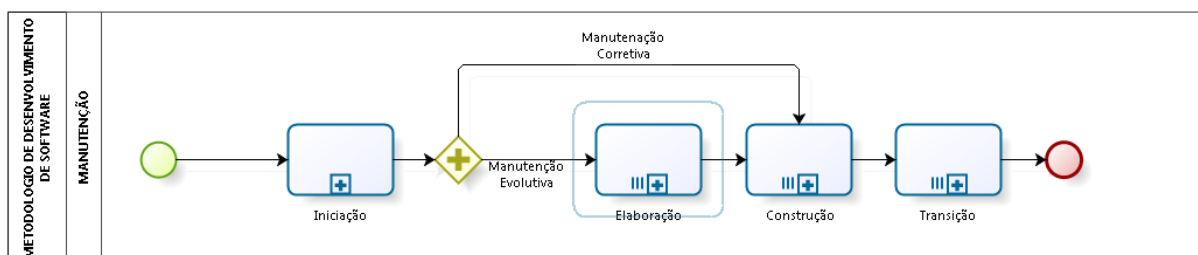
Construção	
 Construção	<p>Essa fase tem o objetivo de codificar e testar as alterações solicitadas pelo usuário. Em tempo de manutenção a construção pode ter duas classificações: corretiva e evolutiva. A construção sendo de uma manutenção corretiva terá como finalidade a correção de eventuais erros identificados pelo usuário final do produto. Quando manutenção evolutiva o aperfeiçoamento do produto com mudanças de funcionalidades ou criação de novas. O teste em ambas as situações ocorre da mesma maneira.</p>

## 6.1.4 TRANSIÇÃO

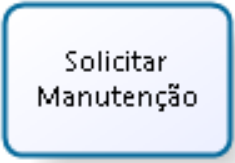
Transição	
 Transição	<p>Esta fase disponibiliza o produto para seus usuários finais. Ao final desta fase, deve-se ter um sistema de software documentado, funcionando corretamente em seu ambiente operacional.</p>

## 6.2 FASE – INICIAÇÃO

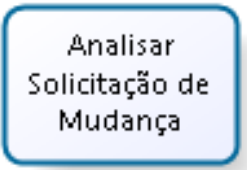
As atividades da fase de iniciação para manutenção serão descritas a seguir:



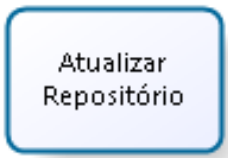
## 6.2.1 SOLICITAR MANUTENÇÃO

Solicitar Manutenção	
	Solicitação de manutenção em um sistema.
Entradas:	Necessidade de evolução ou correção no sistema.
Descrição das Atividades:	Abrir solicitação de manutenção através de ferramenta de gestão de demanda.
Saídas:	Solicitação de Manutenção.
Ferramentas:	Ferramenta de Gestão de Demanda.
Ator:	Gestor do Sistema.

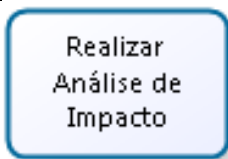
## 6.2.2 ANALISAR SOLICITAÇÃO DE MANUTENÇÃO

Analisar Solicitação de Manutenção	
	Análise sobre o pedido de manutenção para identificação do tipo: Evolutiva; Corretiva (dentro do prazo de garantia); Corretiva (fora do prazo de garantia).
Entradas:	Solicitação de Manutenção.
Descrição das Atividades:	Analisar a solicitação de manutenção e identificar o tipo de solicitação, caso esteja errado corrigir conforme regras contratuais; Planejar o esforço para atendimento da solicitação e enviar para a aprovação do fiscal técnico; Solicitar a preparação do repositório podendo ser pelo andamento da solicitação na ferramenta de gerenciamento de demandas ou conforme estiver explícito no contrato.
Saídas:	Planejamento de Esforço; Solicitação para atualizar repositório.
Ferramentas:	Ferramenta de Gestão de Demanda.
Ator:	Gerente de Projeto.


### 6.2.3 ATUALIZAR REPOSITÓRIO

Atualizar Repositório	
	Disponibilizar o repositório e planejar as atividades do gerenciamento de configuração para manter a integridade dos produtos de trabalho.
Entradas:	Solicitação de Abertura de Branches.
Descrição das Atividades:	Atualizar o Plano de Gerenciamento de Configuração; Criar Branches; Aplicar as regras de acesso de acordo com os padrões definidos no documento de Controle de Baselines e Branches (CBB); Comunicar o solicitante quanto à criação da branche.
Saídas:	Plano de Gerenciamento de Configuração atualizado; Controle de Baseline e Branches (CBB) atualizado; Branches disponíveis para uso.
Ferramentas:	Sistema de Controle de Versão; Sistema de Gestão de Demanda.
Ator:	Gerente de Configuração de Software.


### 6.2.4 REALIZAR ANÁLISE DE IMPACTO

Realizar Análise de Impacto	
	Elicitar, documentar e validar a solicitação de manutenção sugerida pelo Gestor.
Entradas:	Solicitação de Manutenção.
Descrição das Atividades:	Reuniões com gestor para entendimento da solicitação de manutenção; Identificar as necessidades; Identificar as funcionalidades impactadas.
Saídas:	Documento de Análise de Impacto.
Ferramentas:	N/A.
Ator:	Analista de Requisito.

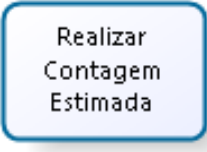
## 6.2.5 VERIFICAR ANÁLISE DE IMPACTO

Verificar Análise de Impacto	
	Verificar e avaliar a análise de impacto de acordo com os critérios de qualidade: Não ambíguo, Correto, Consistente, Verificável, Realizável, Completo e Compreensível.
Entradas:	Documento de Análise de Impacto.
Descrição das Atividades:	Analisar a qualidade do documento levando em consideração as informações de valor para a manutenção.
Saídas:	Checklist do Documento de Análise de Impacto.
Ferramentas:	N/A.
Ator:	Analista de Qualidade.

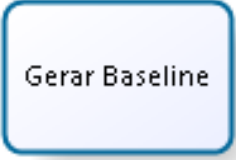
## 6.2.6 HOMOLOGAR PRODUTO

Homologar Produto	
 <p>Homologar Produto</p>	Realizar a homologação do documento de Análise de Impacto. No ato da entrega do artefato é assinado o Termo de Aceite Provisório e depois de homologado é assinado Termo de Aceite Definitivo.

## 6.2.7 REALIZAR CONTAGEM ESTIMADA

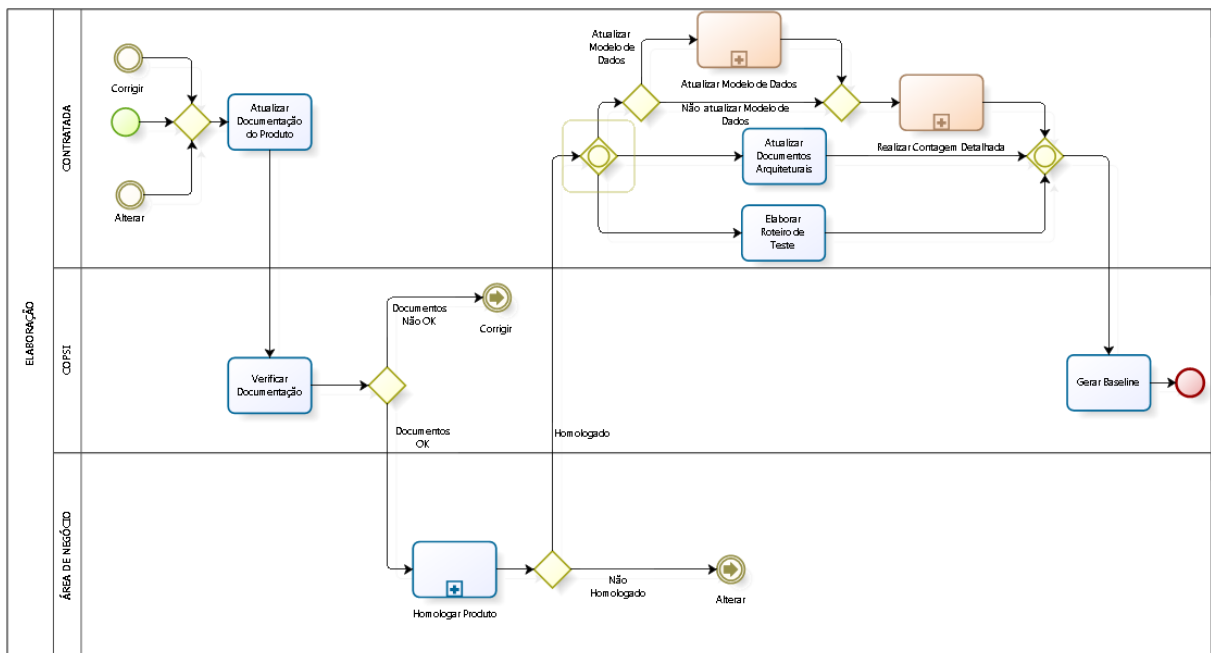
Realizar Contagem Estimada	
	Obter uma estimativa inicial do tamanho da manutenção com base no Documento de Análise de Impacto.
Entradas:	Documento de Análise de Impacto.
Descrição das Atividades:	Realizar uma estimativa inicial do tamanho das funcionalidades impactadas, utilizando a métrica adotada no contrato.
Saídas:	Contagem Estimada.
Ferramentas:	N/A.
Ator:	Analista de Métrica.

## 6.2.8 GERAR BASELINE

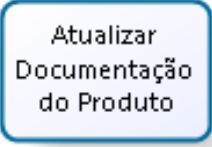
Gerar Baseline	
	Gerar uma versão estável e congelada dos artefatos gerados nessa fase quando estes estiverem homologados, mantendo a integridade dos artefatos do projeto permitindo que mudanças sejam aplicadas somente através de um processo formal.
Entradas:	Solicitação para geração da baseline.
Descrição das Atividades:	Criar baseline referente aos artefatos indicado no Plano de Gerenciamento e Configuração.
Saídas:	Baseline gerada.
Ferramentas:	Sistema de controle de versão.
Ator:	Gerente de Configuração de Software.

## 6.3 FASE – ELABORAÇÃO


As atividades da fase de elaboração para manutenção serão descritas a seguir:



### 6.3.1 ATUALIZAR DOCUMENTAÇÃO DO PRODUTO


Atualizar Documentação do Produto	
	Essa atualização deve basear-se na análise de impacto o que pode resultar em uma atualização ou criação de novos documentos.
Entradas:	Reuniões com o Gestor do Sistema; Documento de Análise de Impacto.
Descrição das Atividades:	Realizar levantamento de requisitos se necessário; Atualizar modelo de casos de uso; Especificar ou atualizar requisitos funcionais e não funcionais; Definir e registrar ou atualizar regras de negócio e apresentação; Definir e registrar ou atualizar mensagens do sistema; Atualizar o Glossário; Reuniões com equipe de arquitetura para definição dos casos de uso mais complexos se houver; Reunião com equipe de Usabilidade para apoio na construção ou alteração das telas do sistema. É imprescindível a produção de atas de reuniões toda vez que houver reuniões de entendimento de qualquer assunto.
Saídas:	Especificação de Caso de Uso novo ou atualizado; Modelo de Caso de Uso atualizado; Documento de Regras atualizado; Lista de Mensagem atualizada; Matriz de rastreabilidade atualizada; Glossário atualizado.
Ferramentas:	Ferramenta de modelagem UML; Ferramenta de Criação de Tela.
Ator:	Analista de Requisito.

### 6.3.2 ATUALIZAR DOCUMENTAÇÃO DO PRODUTO


Verificar Documentação	
	Verificar a consistência, completude e existência de requisitos ambíguos nos documentos incluídos/alterados.

Entradas:	Todos os documentos impactados na atividade anterior.
Descrição das Atividades:	Verificar a conformidade dos requisitos.
Saídas:	Checklist de Requisitos.
Ferramentas:	N/A.
Ator:	Analista de Qualidade.


### 6.3.3 HOMOLOGAR PRODUTO

Homologar Produto	
 Homologar Produto	<p>Realizar a homologação dos artefatos da fase de elaboração. No ato da entrega dos artefatos é assinado o Termo de Aceite Provisório e depois de homologado é assinado Termo de Aceite Definitivo.</p>

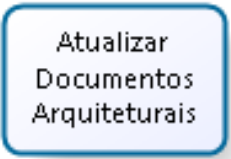
### 6.3.4 ATUALIZAR MODELO DE DADOS

Atualizar Modelo de Dados	
 Atualizar Modelo de Dados	<p>Atualizar o modelo de dados do projeto de acordo com a documentação produzida pela equipe de desenvolvimento e iniciar o processo de execução de script de atualização do Banco de Dados do projeto. Processo descrito na Metodologia de Administração de Dados (MAD).</p>

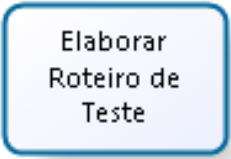
### 6.3.5 REALIZAR CONTAGEM DETALHADA

Realizar Contagem Detalhada	
 Realizar Contagem Detalhada	<p>Realizar contagem detalhada de acordo com a métrica adotada no contrato após o fechamento da elaboração. Processo descrito no Roteiro de Métricas do MTPA.</p>

### 6.3.6 ATUALIZAR DOCUMENTOS ARQUITETURAIS

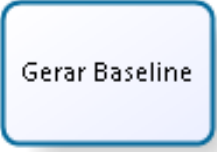
Atualizar Documentos Arquiteturais	
	A atualização do sistema exige a análise técnica para possíveis adaptações arquiteturais.
Entradas:	Especificação de Caso de Uso; Modelo de Casos de Uso; Documento de Regras.
Descrição das Atividades:	Verificar a atualização da documentação do produto e analisar se há necessidade de Atualização Arquitetural. Atualizar Documento de Arquitetura de Software se necessário.
Saídas:	Documento de Arquitetura de Software atualizado; Plano de Implantação atualizado; Mapa de Integração.
Ferramentas:	N/A.
Ator:	Arquiteto de Software.

### 6.3.7 ELABORAR ROTEIRO DE TESTE

Elaborar Roteiro de Teste	
	Elaborar os roteiros de teste para viabilizar a execução dos testes.
Entradas:	Toda documentação que sofreu atualização.
Descrição das Atividades:	Elaborar Roteiro de Teste baseados na especificação dos requisitos.
Saídas:	Roteiro de Teste.
Ferramentas:	Ferramenta de Gestão de Teste.
Ator:	Analista de Teste.

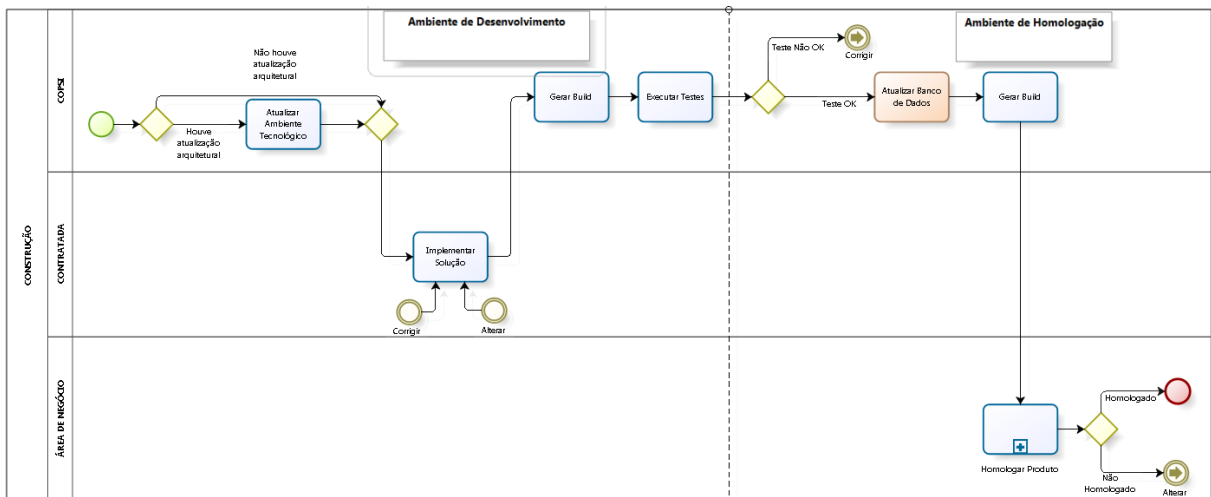


### 6.3.8 GERAR BASELINE

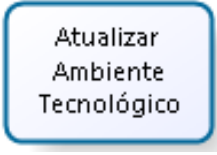
Gerar Baseline	
	Gerar uma versão estável e congelada dos artefatos quando estes estiverem homologados e ou finalizados, mantendo a integridade dos artefatos do projeto permitindo que mudanças sejam aplicadas somente através de um processo formal.
Entradas:	Solicitação para geração da baseline.
Descrição das Atividades:	Criar baseline referente aos artefatos indicado no Plano de Gerenciamento e Configuração.
Saídas:	Baseline gerada.
Ferramentas:	Sistema de Controle de Versão.
Ator:	Gerente de Configuração de Software.

## 6.4 FASE- CONSTRUÇÃO

As atividades da fase de construção para manutenção serão descritas a seguir:

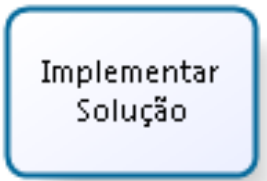


### 6.4.1 ATUALIZAR AMBIENTE TECNOLÓGICO


Atualizar Ambiente Tecnológico	
	Atualizar as ferramentas de apoio para desenvolvimento ou manutenção dos sistemas.
Entradas:	Documento de Arquitetura de Software; Plano de Implantação.
Descrição das Atividades:	Atualizar o Plano de Gerenciamento de Configuração; Configurar ferramentas de apoio.

Saídas:	Plano de Gerenciamento de Configuração atualizado; Ambiente tecnológico criado.
Ferramentas:	N/A.
Ator:	Gerente de Configuração de Software.


#### 6.4.2 IMPLEMENTAR SOLUÇÃO

Implementar Solução	
	Implementar os casos de uso pertencentes ao plano de iteração que está sendo executado, gerando componentes e executando os testes unitários. O Integrador deverá executar os procedimentos planejados para a integração e deploy do sistema de acordo com o ambiente de execução, aplicando os padrões de versionamento definidos no Plano de Gerenciamento de Configuração e as configurações fornecidas pelo Administrador de Banco de Dados.
Entradas:	Especificação de Caso de Uso; Modelo de Casos de Uso; Documento de Regras; Lista de Mensagem; Documento de Arquitetura de Software; Modelo de Dados; Protótipo.
Descrição das Atividades:	Produzir código fonte dos casos de uso da iteração, implementando operações, estado, associações e atributos das classes de implementação; Fornecer feedback para o arquiteto se forem encontrados erros de design e execução do passo anterior; Avaliar o código, verificando se está adequado para a sua finalidade; Realizar testes na unicidade, executando os casos de testes unitários; Corrigir erros/defeitos encontrados nos testes.
Saídas:	Componentes implementados e testados.
Ferramentas:	N/A.
Ator:	Desenvolvedor.


### 6.4.3 GERAR BUILD

Gerar Build	
	Gerar a Build no ambiente de desenvolvimento para realização de testes no sistema implementado.
Entradas:	Solicitação para geração da build; Código Fonte.
Descrição das Atividades:	Marcar os códigos gerados que poderão ser testados.
Saídas:	Código fonte marcado.
Ferramentas:	Sistema de Controle de Versão.
Ator:	Analista de Configuração de Software.

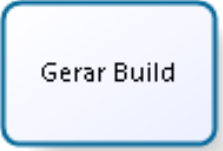
### 6.4.4 EXECUTAR TESTES

Executar Testes	
	Realizar os testes da demanda implementada de acordo com a definição no Plano de Teste, registrar o comportamento da build e avaliar o nível de qualidade no projeto desenvolvido.
Entradas:	Roteiro de Teste; Código fonte no ambiente de desenvolvimento.
Descrição das Atividades:	Realizar testes nas funcionalidades.
Saídas:	Planilha de Resultado de Teste.
Ferramentas:	N/A.
Ator:	Analista de Teste.


### 6.4.5 ATUALIZAR BANCO DE DADOS

Atualizar Banco de Dados	
	Atualização do Banco de Dados no ambiente de Homologação. Atividade descrita na Metodologia de Administração de Dados (MAD).

#### 6.4.6 GERAR BUILD

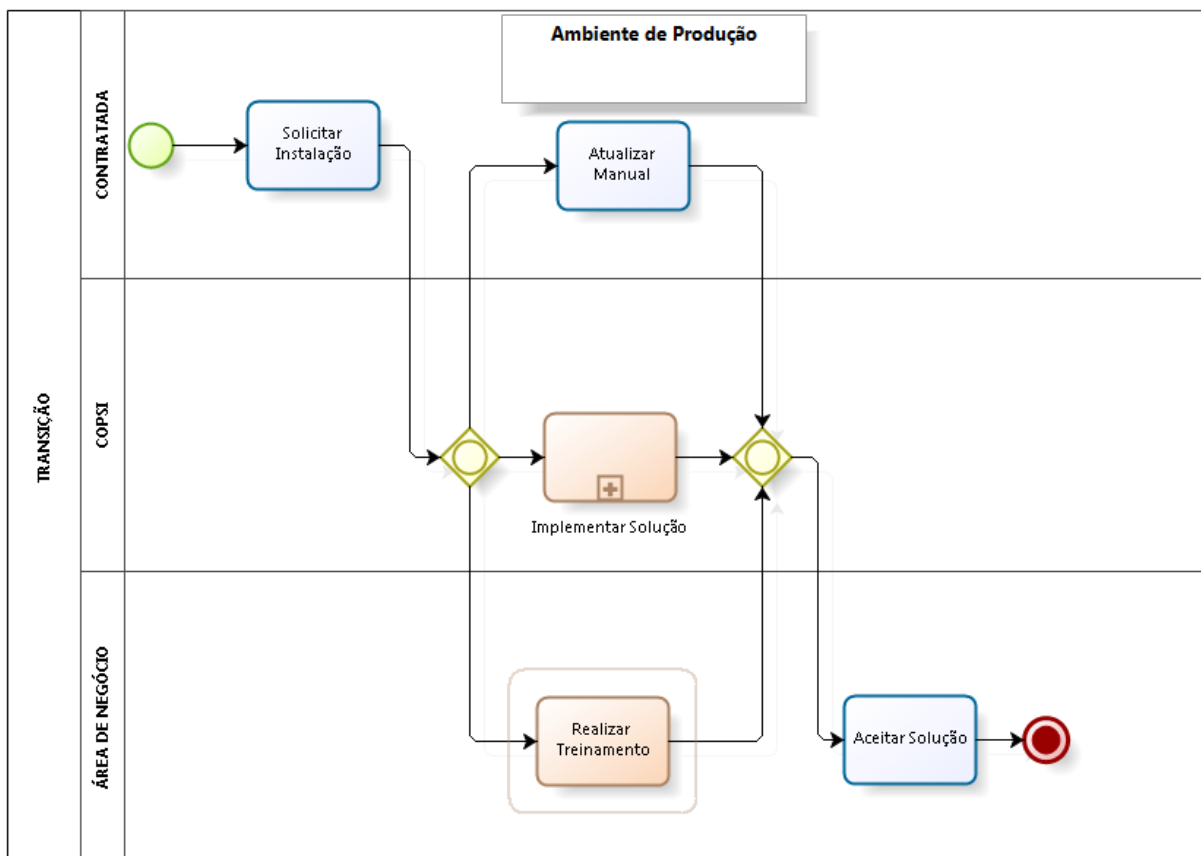
Gerar Build	
	Gerar a Build no ambiente de homologação para homologação da área de negócio.
Entradas:	Solicitação para geração da build; Código fonte testado.
Descrição das Atividades:	Marcar código fonte que foi testado.
Saídas:	Código fonte marcado para homologação.
Ferramentas:	Sistema de Controle de Versão.
Ator:	Analista de Configuração de Software.

#### 6.4.7 HOMOLOGAR PRODUTO

Homologar Produto	
 Homologar Produto	Realizar a homologação do produto entregue. Assinar os Termos de Recebimento Provisório e Definitivo.

### 6.5 FASE – TRANSIÇÃO

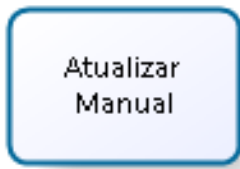
As atividades da fase de transição para manutenção serão descritas a seguir:




### 6.5.1 SOLICITAR INSTALAÇÃO

Solicitar Instalação	
	<p>Solicitação de instalação do sistema no ambiente de produção.</p> <p>Atividade descrita na Metodologia de Gerenciamento de Serviço de Infra-estrutura (MGSI).</p>
Entradas:	Termo de Recebimento Definitivo assinado.
Descrição das Atividades:	Preencher requisição de mudança.
Saídas:	Requisição de Mudança (RDM).
Ferramentas:	Ferramenta de gerenciamento de serviço de TI.
Ator:	Gerente de Projeto.


### 6.5.2 ATUALIZAR MANUAL

Atualizar Manual	
	Atualizar manual de apoio para o uso das funcionalidades do sistema.
Entradas:	Documentação técnica do sistema; Solução no ambiente de homologação.
Descrição das Atividades:	Atualizar manual do usuário.
Saídas:	Manual do Usuário.
Ferramentas:	N/A.
Ator:	Redator Técnico.

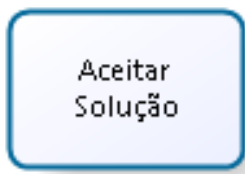
### 6.5.3 IMPLEMENTAR SOLUÇÃO

Implantar Solução	
 Implantar Solução	Realizar a implantação do produto homologado em ambiente de produção. Processo descrito na Metodologia de Gerenciamento de Serviço de Infra-estrutura (MGSI).

### 6.5.4 REALIZAR TREINAMENTO

Realizar Treinamento	
 Realizar Treinamento	Realizar treinamento para os usuários do sistema.

### 6.5.5 ACEITAR SOLUÇÃO

Aceitar Solução	
 Aceitar Solução	Aprovar a solução instalada no ambiente de produção.
Entradas:	Código no ambiente de produção.

Descrição das Atividades:	Assinar Termo de Recebimento Definitivo; Assinar Termo de Aceite.
Saídas:	Termo de Recebimento Definitivo; Termo de Aceite.
Ferramentas:	N/A.
Ator:	Gestor do Sistema.

# PARTE II – ARTEFATOS





## 7 ARTEFATOS

Em sentido amplo, o termo artefato representa um produto concreto produzido, modificado ou utilizado pelas atividades de um processo.

A MDS disponibiliza modelos (templates) para os seus artefatos, com o objetivo de orientar e facilitar a sua elaboração.

Abaixo estão relacionados os artefatos definidos, separados por Fase.

<b>Proposta de Projeto</b>
<ul style="list-style-type: none"><li>• Documento de Visão de Negócio</li><li>• Glossário</li><li>• Diagrama de Processo</li><li>• Contagem Estimada</li><li>• Plano de Gerenciamento de Configuração</li></ul>
<b>Iniciação</b>
<ul style="list-style-type: none"><li>• Documento de Visão de Sistema</li><li>• Plano de Projeto</li><li>• Plano de Iteração</li><li>• Caso de Desenvolvimento</li><li>• Documento de Definição Arquitetural</li><li>• Plano de Teste</li><li>• Documento de Análise de Impacto</li></ul>
<b>Elaboração</b>
<ul style="list-style-type: none"><li>• Especificação de Caso de Uso</li><li>• Modelo de Caso de Uso</li><li>• Protótipo Funcional</li><li>• Documento de Regras</li><li>• Lista de Mensagem</li><li>• Matriz de Rastreabilidade</li><li>• Documento de Arquitetura de Software</li><li>• Plano de Implantação</li><li>• Modelo de Dados</li><li>• Roteiro de Teste</li><li>• Contagem Detalha</li></ul>
<b>Construção</b>
<ul style="list-style-type: none"><li>• Código Fonte</li><li>• Roteiro de Teste</li><li>• Planilha de Resultado de Teste</li><li>• Manual do Usuário</li></ul>
<b>Transição</b>
<ul style="list-style-type: none"><li>• Requisição de Mudança</li><li>• Termo de Encerramento</li><li>• Lições Aprendidas</li></ul>

# PARTE III – PAPÉIS



## 8 PAPÉIS

Um papel define o comportamento e as responsabilidades de um profissional ou grupo de profissionais que participam da elaboração e do desenvolvimento do projeto. O comportamento é representado através das atividades que cada papel deve desempenhar ao longo do projeto.

Normalmente, as responsabilidades estão associadas aos artefatos que cada papel deve produzir e manter ao longo das atividades que realiza. Na prática, um mesmo papel pode ser desempenhado por mais de uma pessoa, assim como, uma mesma pessoa pode assumir vários papéis ao longo do projeto.

Abaixo, estão relacionados os papéis previstos na MDS agrupados em 3 áreas de atuação:

- **ÁREA DE NEGÓCIO:** Representa o demandante e/ou usuário do sistema.
- **CONTRATADA:** Representa a equipe responsável por desenvolver o sistema.
- **COPSI:** Representa a área de gerenciamento, suporte e apoio ao desenvolvimento do sistema.

<b>ÁREA DE NEGÓCIO</b>	
Gestor do Sistema	Responsável por: <ul style="list-style-type: none"><li>• Demandar uma necessidade para solução de tecnologia de informação;</li><li>• Aprovar documentos de planejamento;</li><li>• Acompanhar progresso do projeto;</li><li>• Definir as regras de negócio do sistema, priorizar requisitos;</li><li>• Homologar documentos técnicos;</li><li>• Solicitar mudanças no projeto;</li><li>• Solicitar manutenções nos sistemas em produção;</li><li>• Realizar teste de homologação de cada versão do sistema e autorizar sua entrada em produção;</li><li>• Promover treinamentos sobre a utilização do sistema;</li><li>• Aderir e respeitar o processo de engenharia de software implantado.</li></ul>

<b>CONTRATADA</b>	
Analista de Métrica	Responsável por auxiliar os projetos de desenvolvimento de software quanto a mensuração do tamanho funcional do projeto. Ele é responsável por definir o tipo de contagem e aplicar a técnicas de contagem.

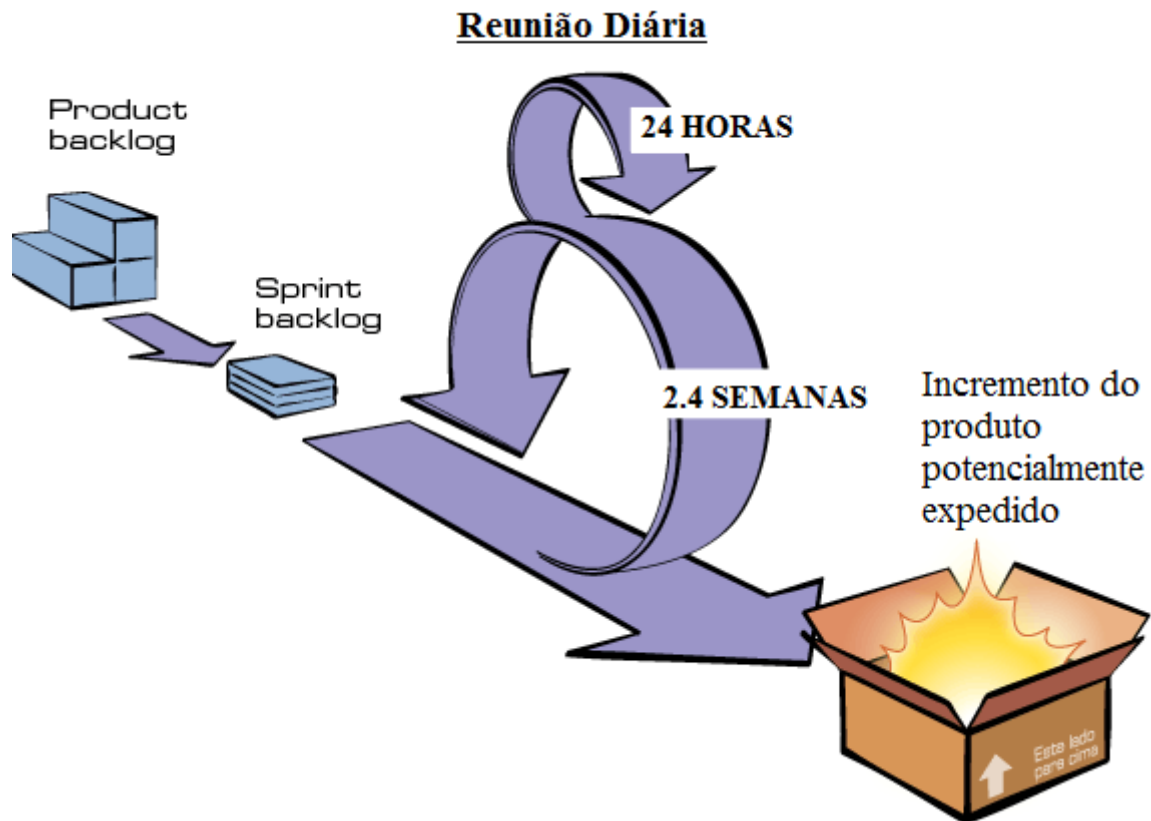
Analista de Processo	<p>Responsável por identificar os atores/donos dos processos de negócios aplicando o ciclo do Gerenciamento de Processo de Negócio que consiste em definir o planejamento e estratégia do processo de negócio, realizar o mapeamento/modelagem do processo para dar base para análises de produtividade, identificação de falhas, desperdícios e gargalos nos processos de negócio com foco no atendimento ao cliente. Com este mapeamento e aplicação de algumas técnicas ele consegue implantar melhorias nos processos para atender a missão, visão e valores da área.</p> <p>Ele ainda é responsável por gerenciar e monitorar os processos de negócios e propor melhorias a fim de refinar o processo.</p>
Analista de Requisito	<p>Responsável por identificar os problemas e necessidades do cliente referente ao sistema em desenvolvimento ou em manutenção, propondo uma solução sistemática e detalhamento de todos os requisitos do sistema (Funcionais, Não funcionais e Restrições do projeto). Conduz o levantamento das informações necessárias para análise e validação do sistema junto ao cliente e a equipe técnica de desenvolvimento do projeto.</p>
Analista de Teste	<p>Responsável por definir a abordagem de teste e assegurar sua correta implementação a fim de encontrar defeitos no software. Isso inclui identificar as técnicas, ferramentas e diretrizes apropriadas para implementar os testes de software para validar a entrega do produto de software a cada iteração.</p>
Arquiteto de Software	<p>Responsável por estabelecer a estrutura geral de cada visão de arquitetura: a decomposição da visão, o agrupamento dos elementos e as interfaces entre esses principais agrupamentos. Portanto, comparado aos outros papéis, a visão do arquiteto de software é ampla, e não detalhada.</p>
Desenvolvedor	<p>Responsável por desenvolver e testar componentes de acordo com os padrões adotados para o projeto, para fins de integração com subsistemas maiores. Quando é necessário criar componentes de teste, como drivers ou stubs, para possibilitar a realização dos testes, o desenvolvedor também é responsável por desenvolver e testar esses componentes e os subsistemas correspondentes.</p>

Designer de Interface	Responsável pelo projeto da interface do usuário e à concepção do protótipo da mesma. É responsável pelo levantamento dos requisitos de interface, incluindo requisitos de usabilidade, pela construção de protótipos, pelo envolvimento de usuários finais nas revisões de usabilidade e pelo fornecimento das informações necessárias para que os implementadores possam realizar o desenvolvimento das interfaces especificadas.
Gerente de Projeto	Responsável por realizar o planejamento do projeto de desenvolvimento de software, a fim de identificar os riscos, traçando planos de ação para cada um deles, realizando o controle de custos do projeto, gerenciando pessoas e tratando de impedimentos para o sucesso e bom andamento do projeto. Ele também é responsável pela gestão do tempo e comunicar a situação do projeto para todos os interessados.
Gerente de Teste	Responsável pelo êxito do esforço de teste. O papel envolve defesa da qualidade e dos testes, planejamento e gerenciamento de recursos e resolução de problemas que representam um obstáculo para o esforço de teste.
Redator Técnico	Responsável por elaborar o material de suporte para o usuário final, como manuais do usuário, textos da ajuda, notas de release e outros.

<b>COPSI</b>	
Administrador de Dados	Responsável pela criação do modelo de banco de dados, seguindo as melhores práticas de normalização do banco, e assim permitindo a definição do projeto de banco de dados, que inclui a criação de tabelas, relacionamentos entre elas, índices, dados e etc, a fim de viabilizar a melhor maneira de armazenar, recuperar e excluir as informações gravadas no banco.
Analista de Configuração de Software	Responsável pela organização de branches, merges e versionamento de código e definição e execução de rotinas de deploy.

<p>Consultor de Projeto</p>	<p>Responsável por assegurar a realização do planejamento, execução e monitoramento do projeto. Especialista na identificação e compreensão de problemas e oportunidades. Isso inclui a capacidade de articular as necessidades que são associadas ao problema-chave a ser resolvido ou a oportunidade a ser realizada.</p>
<p>Gerente de Configuração de Software</p>	<p>Responsável por disponibilizar o ambiente e a infra-estrutura geral de Gerenciamento de Configuração para a equipe de desenvolvimento do produto. A função da gerencia de configuração é oferecer suporte à atividade de desenvolvimento de produtos para que os desenvolvedores e integradores tenham espaços de trabalho adequados para criar e testar seus trabalhos e, dessa forma, permitir que todos os artefatos fiquem disponíveis para inclusão na unidade de implantação, conforme necessário. O gerente de configuração também deve assegurar que o ambiente facilite a revisão do produto e também é responsável por redigir o Plano de Gerenciamento de Configuração.</p>

# PARTE IV – SCRUM



## 9 SCRUM

O Ministério dos Transportes tem utilizado até o momento a Metodologia de Desenvolvimento da Rational – RUP.

Porém, acompanhando o mercado, à CGTI adotará o Processo Ágil de Desenvolvimento (SCRUM), com a capacitação de sua equipe sobre o processo adotado, visando maior leveza, alta produtividade e maiores resultados em suas entregas.

Esta migração de metodologia ocorrerá de forma gradual e será iniciada com projetos menores até que a equipe se sinta madura para adoção integral da nova metodologia.

- **O que é Scrum?**

No Scrum, os projetos são divididos em ciclos (tipicamente mensais) chamados de **Sprints**. O **Sprint** representa um Time Box dentro do qual um conjunto de atividades deve ser executado. **Metodologias ágeis** de desenvolvimento de software são iterativas, ou seja, o trabalho é dividido em iterações, que são chamadas de Sprints no caso do Scrum.

As funcionalidades a serem implementadas em um projeto são mantidas em uma lista que é conhecida como **Product Backlog**. No início de cada Sprint, faz-se um **Sprint Planning Meeting**, ou seja, uma reunião de planejamento na qual o **Dono do Produto (Product Owner)** prioriza os itens do **Product Backlog** e a equipe seleciona as atividades que ela será capaz de implementar durante o Sprint que se inicia. As tarefas alocadas em um Sprint são transferidas do **Product Backlog** para o **Sprint Backlog**.

A cada dia de uma Sprint, a equipe faz uma breve reunião (normalmente de manhã), chamada **Reunião Diária (Daily Scrum)**. O objetivo é disseminar conhecimento sobre o que foi feito no dia anterior, identificar impedimentos e priorizar o trabalho do dia que se inicia.

Ao final de um Sprint, a equipe apresenta as funcionalidades implementadas em uma **Sprint Review Meeting**. Finalmente, faz-se uma **Sprint Retrospective** e a equipe parte para o planejamento do próximo Sprint. Assim reinicia-se o ciclo.

- **Daily Scrum:**

A cada dia do Sprint a equipe faz uma **reunião diária**, chamada **Reunião Diária (Daily Scrum)**. Ela tem como objetivo disseminar conhecimento sobre o que foi feito



no dia anterior, identificar impedimentos e priorizar o trabalho a ser realizado no dia que se inicia.

Os Daily Scrums normalmente são realizadas no mesmo lugar, na mesma hora do dia. Idealmente são realizados na parte da manhã, para ajudar a estabelecer as prioridades do novo dia de trabalho.

Todos os membros da equipe devem participar do Daily Scrum. Outras pessoas também podem estar presentes, mas só poderão escutar. Isso torna os Daily Scrums uma excelente forma para uma equipe disseminar informações sobre o estado do projeto.

O Daily Scrum não deve ser usado como uma reunião para resolução de problemas. Questões levantadas devem ser levadas para fora da reunião e normalmente tratadas por um grupo menor de pessoas que tenham a ver diretamente com o problema ou possam contribuir para solucioná-lo. Durante o Daily Scrum, cada membro da equipe provê respostas para cada uma destas três perguntas:

- O que você fez ontem?
- O que você fará hoje?
- Há algum impedimento no seu caminho?

Concentrando-se no que cada pessoa fez ontem e no que ela irá fazer hoje, a equipe ganha uma excelente compreensão sobre que trabalho foi feito e que trabalho ainda precisa ser feito. O Daily Scrum não é uma reunião de status report na qual um chefe fica coletando informações sobre quem está atrasado. Ao invés disso, é uma reunião na qual membros da equipe assumem compromissos perante os demais.

Os impedimentos identificados no Daily Scrum devem ser tratados pelo **Scrum Master** o mais rapidamente possível.

- **Product Backlog:**

O **Product Backlog** é uma lista contendo todas as funcionalidades desejadas para um produto. O conteúdo desta lista é definido pelo **Dono do Produto (Product Owner)**. O Product Backlog não precisa estar completo no início de um projeto. Pode-se começar com tudo aquilo que é mais óbvio em um primeiro momento. Com o tempo, o Product Backlog cresce e muda à medida que se aprende mais sobre o produto e seus usuários.

Durante o **Sprint Planning Meeting**, o **Dono do Produto (Product Owner)** prioriza os itens do Product Backlog e os descreve para a equipe. A equipe então determina que itens será capaz de completar durante a Sprint que está por começar. Tais itens são, então, transferidos do Product Backlog para o **Sprint Backlog**. Ao fazer isso, a equipe quebra cada item do Product Backlog em uma ou mais tarefas do **Sprint Backlog**. Isso ajuda a dividir o trabalho entre os membros da equipe. Podem fazer parte do Product Backlog tarefas técnicas ou atividades diretamente relacionadas às funcionalidades solicitadas.

- **Product Owner:**

O Product Owner é a pessoa que define os itens que compõem o **Product Backlog** e os prioriza nas **Sprint Planning Meetings**.

O **Scrum Team** olha para o **Product Backlog** priorizado, seleciona os itens mais prioritários e se compromete a entregá-los ao final de um Sprint (iteração). Estes itens transformam-se no **Sprint Backlog**.

A equipe se compromete a executar um conjunto de atividades no Sprint e o Product Owner se compromete a não trazer novos requisitos para a equipe durante o Sprint. Requisitos podem mudar (e mudanças são encorajadas), mas apenas fora do Sprint. Uma vez que a equipe comece a trabalhar em um Sprint, ela permanece concentrada no objetivo traçado para o Sprint e novos requisitos não são aceitos.

- **Release Burndown:**

Em um projeto **Scrum**, a equipe monitora seu progresso em relação a um plano atualizando um Release Burndown Chart ao final de cada Sprint (iteração). O eixo horizontal de um Release Burndown Chart mostra os Sprints; o eixo vertical mostra a quantidade de trabalho que ainda precisa ser feita no início de cada Sprint. O trabalho que ainda resta pode ser mostrado na unidade preferencial da equipe: story points, dias ideais, team days e assim por diante.

- **Scrum Master:**

O Scrum Master procura assegurar que a equipe respeite e siga os valores e as práticas do **Scrum**. Ele também protege a equipe assegurando que ela não se comprometa excessivamente com relação àquilo que é capaz de realizar durante um Sprint.

O papel de Scrum Master é tipicamente exercido por um gerente de projeto ou um líder técnico, mas em princípio pode ser qualquer pessoa da equipe.

- **Scrum Team:**

O Scrum Team é a equipe de desenvolvimento. Nela, não existe necessariamente uma divisão funcional através de papéis tradicionais, tais como programador, designer, analista de testes ou arquiteto. Todos no projeto trabalham juntos para completar o conjunto de trabalho com o qual se comprometeram conjuntamente para um Sprint.

Um Scrum Team típico tem de 6 a 10 pessoas, embora haja relatos de projetos **Scrum** com equipes maiores. A principal abordagem para trabalhar com equipes grandes no **Scrum** é usando o conceito de "Scrum of Scrums". Cada Scrum Team trabalha normalmente, mas cada equipe também contribui com uma pessoa que deverá frequentar o Scrum of Scrums Meeting para coordenar o trabalho de múltiplas equipes **Scrum**. Esses encontros são análogos a **Reunião Diária (Daily Scrums)**, mas não acontecem necessariamente todos os dias. Fazer essa reunião duas ou três vezes por semana tende a ser suficiente na maioria das organizações.

- **Sprint Backlog:**

O Sprint Backlog é uma lista de tarefas que o **Scrum Team** se compromete a fazer em um Sprint. Os itens do Sprint Backlog são extraídos do **Product Backlog**, pela equipe, com base nas prioridades definidas pelo **Product Owner** e a percepção da equipe sobre o tempo que será necessário para completar as várias funcionalidades.

Cabe a equipe determinar a quantidade de itens do **Product Backlog** que serão trazidos para o Sprint Backlog, já que é ela quem irá se comprometer a implementá-los.

Durante um Sprint, o **Scrum Master** mantém o Sprint Backlog atualizando-o para refletir que tarefas são completadas e quanto tempo a equipe acredita que será necessário para completar aquelas que ainda não estão prontas. A estimativa do trabalho que ainda resta a ser feito no Sprint é calculada diariamente e colocada em um gráfico, resultando em um Sprint Burndown Chart.

- **Sprint Planning Meeting:**

O Sprint Planning Meeting é uma reunião na qual estão presentes o **Dono do Produto (Product Owner)**, o **Scrum Master** e todo o **Scrum Team**, bem como qualquer pessoa interessada que esteja representando a gerência ou o cliente.

Durante o Sprint Planning Meeting, o **Dono do Produto (Product Owner)** descreve as funcionalidades de maior prioridade para a equipe. A equipe faz perguntas durante a reunião de modo que seja capaz de quebrar as funcionalidades em tarefas técnicas, após a reunião. Essas tarefas irão dar origem ao **Sprint Backlog**.

O **Dono do Produto (Product Owner)** não precisa descrever todos os itens que estão no **Product Backlog**. Dependendo do tamanho do **Product Backlog** e da velocidade da equipe, pode ser suficiente descrever apenas os itens de maior prioridade, deixando a discussão dos itens de menor prioridade para o próximo Sprint Planning Meeting.

Coletivamente, o **Scrum Team** e o **Dono do Produto (Product Owner)** definem um objetivo para o Sprint, que é uma breve descrição daquilo que se tentará alcançar no Sprint. O sucesso do Sprint será avaliado mais adiante no **Sprint Review Meeting** em relação ao objetivo traçado para o Sprint.

Depois do Sprint Planning Meeting, a equipe **Scrum** se encontra separadamente para conversar sobre o que eles escutaram e decidir quanto eles podem se comprometer a fazer no Sprint que será iniciado. Em alguns casos, haverá negociação com o **Dono do Produto (Product Owner)**, mas será sempre responsabilidade da equipe determinar o quanto ela será capaz de se comprometer a fazer.

- **Sprint Retrospective:**

O Sprint Retrospective ocorre ao final de um Sprint e serve para identificar o que funcionou bem, o que pode ser melhorado e que ações serão tomadas para melhorar.

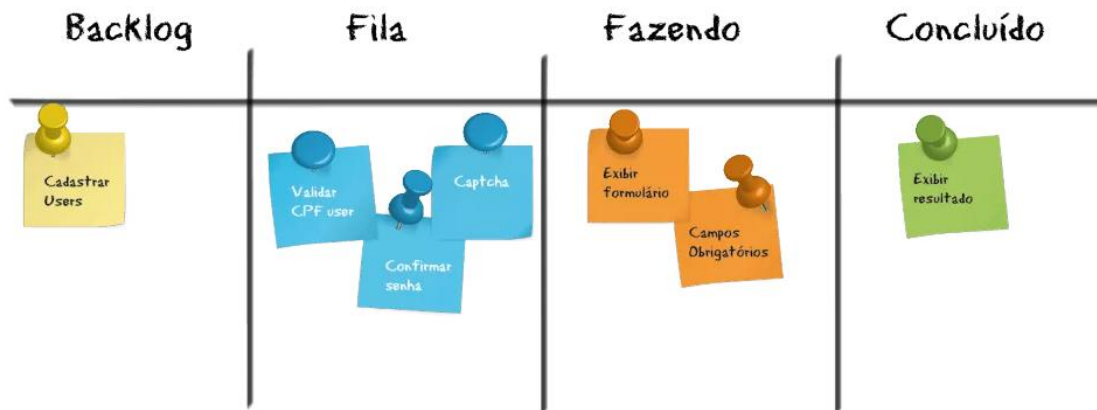
- **Sprint Review Meeting:**

Ao final de cada Sprint é feito um Sprint Review Meeting. Durante esta reunião, o **Scrum Team** mostra o que foi alcançado durante o Sprint. Tipicamente, isso tem o formato de um demo das novas funcionalidades.

Os participantes do Sprint Review tipicamente incluem o **Dono do Produto (Product Owner)**, o **Scrum Team**, o **Scrum Master**, gerência, clientes e engenheiros de outros projetos.

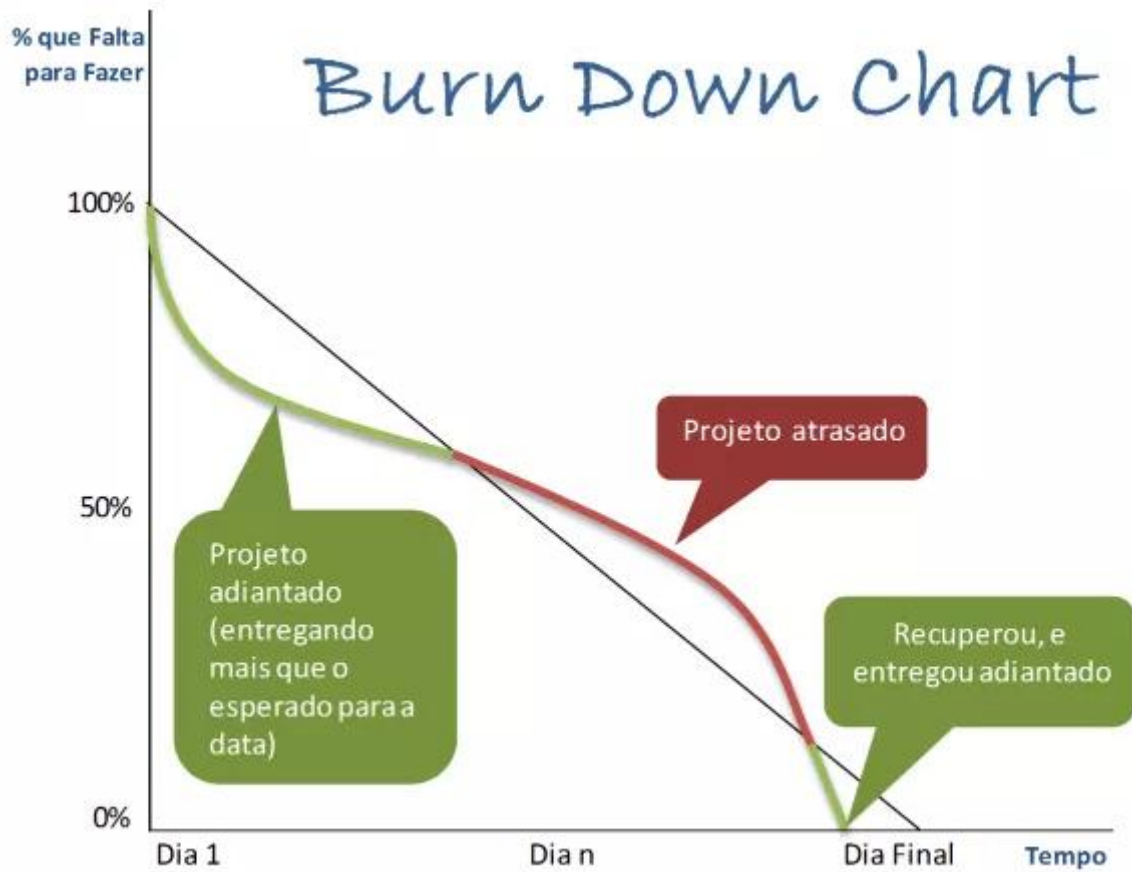
Durante o Sprint Review, o projeto é avaliado em relação aos objetivos do Sprint, determinados durante o **Sprint Planning Meeting**. Idealmente, a equipe completou cada um dos itens do **Product Backlog** trazidos para fazer parte do Sprint, mas o importante mesmo é que a equipe atinja o objetivo geral do Sprint.

Para gerenciar os projetos ágeis, será utilizado o Kanban. O Kanban (Quadro de Trabalho) é utilizado para organizar as atividades dos itens de Backlog da Sprint, separando-as em basicamente em quatro estados (isso pode variar de projeto a projeto): A fazer, Em andamento, Em Testes e Concluído. Esse “quadro” é muito produtivo, pois basta olhar para ele para realizar a leitura do progresso da Sprint. Segue abaixo uma figura que representa este quadro de trabalho:



- **Burn Down Chart:**

O Burndown é um simples gráfico, com dois eixos X e Y, baseado nas atividades que não ultrapassem um dia de trabalho. O eixo X indica o número de tarefas existentes no Sprint e o eixo Y os dias que representam o tamanho do Sprint.



O Scrum possui 5 valores, segue abaixo a imagem que representa eles:



# PARTE V – SERVIÇOS DE APOIO

## 10 SERVIÇOS DE APOIO AO DESENVOLVIMENTO DE APLICAÇÕES

PORTFÓLIO DE SERVIÇOS DE APOIO AO DESENVOLVIMENTO DE APLICAÇÕES - MTPA		
SERVIÇO MACRO	DEFINIÇÃO	SERVIÇOS
<b>1. APOIO À GESTÃO DA COORDENAÇÃO DE PROJETOS E SISTEMAS</b>	Os serviços de Apoio à Gestão da Coordenação de Projetos e sistemas visam a apoiar na melhoria contínua dos processos de trabalho, priorização de projetos de software, gestão de conflitos, alocação e treinamento das pessoas, análise e implementação de boas práticas no mercado para desenvolvimento de software, definição de padrões de artefatos e análise e implementação das melhores ferramentas de trabalho para aumento da produtividade e entrega de resultado.	1.1 Apoiar na priorização dos projetos de software
		1.2 Apoiar na Gestão do Portfólio de Projetos de Software
		1.3 Apoiar na Liderança da equipe de novos projetos de software
		1.4 Apoiar na Criação/Atualização dos fluxos de Gestão de Projetos de Software
		1.5 Apoiar na Gestão de Conflitos
		1.6 Apoiar no acompanhamento dos cronogramas dos projetos
		1.7 Apoiar os Gerentes de Projetos na elaboração dos Cronogramas
		1.8 Apoiar na Criação e Evolução de Artefatos
		1.9 Apoiar na atualização da Metodologia de Desenvolvimento de Software
		1.10 Apoiar na customização da Ferramenta de Gestão de Demandas de sistemas (RTC da IBM)
		1.11 Propor melhorias nos Processo, Pessoas e Ferramentas
<b>2. APOIO À GESTÃO DE PROJETOS DE SOFTWARE</b>	Os serviços de Apoio à Gestão de Projetos de Software visam a apoiar no estudo da proposta de projetos, planejamento, monitoramento e controle e encerramento dos projetos de software visando a entrega dos mesmos com qualidade, no prazo e a satisfação do usuário.	2.1 Apoiar no Estudo da Proposta de Projeto e verificar sua viabilidade
		2.2 Apoiar no Desenvolvimento do Termo de Abertura do Projeto
		2.3 Apoiar no Desenvolvimento do Plano de Gerenciamento do Projeto
		2.4 Apoiar na Orientação e Gerenciamento do trabalho do projeto
		2.5 Apoiar no Monitoramento e controle do trabalho do projeto
		2.6 Apoiar no controle integrado de mudança
		2.7 Apoiar no Encerramento do projeto ou fase
		2.8 Apoiar na pesquisa de satisfação do projeto
		2.9 Apoiar na apresentação final do projeto
		2.10 Apoiar no Aceite Definitivo do Projeto

<b>3. APOIO NA ANÁLISE DE NEGÓCIOS</b>	Os serviços de Apoio na Análise de Negócios visam ao entendimento e desenho do Negócio do Ministério e suas vinculadas e à proposta de melhor solução tecnológica.	3.1 Apoiar no preparo da elicitação MACRO de requisitos
		3.2 Apoiar na condução da elicitação MACRO junto às partes interessadas
		3.3 Apoiar no desenho dos Processos de Negócio
		3.4 Apoiar no desenho dos Processos do Sistema
		3.5 Apoiar na definição e lista das principais funcionalidades dos sistemas
		3.6 Apoiar na definição da necessidade do negócio
		3.7 Apoiar na definição da abordagem da solução
		3.8 Apoiar na definição do escopo da solução
		3.9 Apoiar na elaboração do Documento de Visão
		3.10 Apoiar no gerenciamento do Escopo e dos Requisitos da Solução
		3.11 Apoiar no gerenciamento da rastreabilidade de requisitos
		3.12 Apoiar na manutenção dos requisitos para reutilização
		3.13 Apoiar no preparo do pacote de requisitos
		3.14 Apoiar na comunicação dos requisitos as partes interessadas
		3.15 Apoiar na validação e verificação dos requisitos
		3.16 Apoiar na Apresentação da solução
<b>4. APOIO NA ANÁLISE DE REQUISITOS</b>	Os serviços de Apoio a Análise de Requisitos visam o levantamento, documentação, prototipação, mudanças e apresentação de requisitos.	4.1 Apoiar no Levantamento e Análise de Requisitos
		4.2 Apoiar na Prototipação de novas aplicações
		4.3 Apoiar na Prototipação de Solicitações de Mudanças
		4.4 Apoiar na Modelagem de dados
		4.5 Apoiar na Análise e modelagem de dados utilizando UML
		4.6 Apoiar na Apresentação e Aprovação de Requisitos junto aos usuários
		4.7 Apoiar no Repasse dos Requisitos à Fábrica de Software
		4.8 Apoiar no Repasse dos Requisitos à Gestão da Qualidade de Software
<b>5. APOIO NA GESTÃO DE ADMINISTRAÇÃO DE DADOS</b>	Os serviços de Apoio na Gestão de Administração de Dados visam apoiar na garantia da qualidade dos dados.	5.1 Apoiar na Gestão da Arquitetura de Dados
		5.2 Auxiliar no Desenvolvimento e Implantação de sistemas
		5.3 Apoiar na Criação, Análise e Validação de Artefatos de Bases de Dados
		5.4 Apoiar na eliminação da redundância e manutenção da qualidade dos dados
		5.5 Apoiar na Criação e Evolução de soluções de integração e transformação de dados
		5.6 Suporte na análise e solução de problemas
		5.7 Analisar informações e auxiliar na tomada de decisões



		<p>5.8 Multiplicar Conhecimento</p> <p>5.9 Apoiar na migração de dados</p> <p>5.10 Realizar propostas de melhoria na Arquitetura de Dados do MTPA</p> <p>5.11 Apoiar na validação de Manuais de Implantação de Sistemas, no que concerne à criação das Bases de Dados</p> <p>5.12 Apoiar na validação de Scripts SQL</p> <p>5.13 Apoiar na validação e verificação dos Modelos de Dados</p> <p>5.14 Realizar treinamentos para disseminação do conhecimento</p> <p>5.15 Apoiar no desenvolvimento e manutenção de rotinas de exportação, transformação e carregamento de dados (ETL)</p> <p>5.16 Apoiar na Elaboração e Gerência do Ambiente de Inteligência de Negócio (BI)</p>
<p><b>6. APOIO À GESTÃO DE INFRAESTRUTURA E AMBIENTE</b></p>	<p>Os serviços de Apoio à Gestão de Infraestrutura e ambiente visam adequar, manter e melhor o ambiente tecnológico para o bom funcionamento das aplicações.</p>	<p>6.1 Apoiar na criação do Manual de Implantação do Sistema nos servidores de Homologação e Produção</p> <p>6.2 Apoiar na validação e verificação do pacote de implantação dos sistemas</p> <p>6.3 Apoiar na implantação do sistema em ambiente de desenvolvimento</p> <p>6.4 Apoiar na implantação do sistema em ambiente de homologação</p> <p>6.5 Apoiar na implantação do sistema em ambiente de produção</p> <p>6.6 Apoiar nos testes em sistemas recém-implantados</p> <p>6.7 Apoiar no preenchimento do checklist validando versões homologadas dos sistemas</p> <p>6.8 Apoiar no versionamento dos sistemas</p> <p>6.9 Apoiar na preparação do ambiente</p> <p>6.10 Apoiar na análise e solução de problemas de ambiente</p> <p>6.11 Esclarecer dúvidas sobre infraestrutura e ambiente à equipe da Coordenação de Projetos e sistemas</p>
<p><b>7. APOIO NA ARQUITETURA DE SOFTWARE JAVA, PHP, DELPHI E MOBILE</b></p>	<p>Os serviços de Apoio a Arquitetura de Software JAVA, PHP e DELPHI visam atividades relacionadas à arquitetura de software, desde a definição de padrões e modelos de arquitetura, a auditoria e validação do uso</p>	<p>7.1 Apoiar na elaboração/revisão da proposta de arquitetura JAVA, PHP, DELPHI E MOBILE</p> <p>7.2 Apoiar na Análise e Projeto de Sistemas Orientados à Objetos</p> <p>7.3 Apoiar na Análise e Projetos de Sistemas Mobile</p> <p>7.4 Apoiar na Integração Contínua da Arquitetura</p> <p>7.5 Apoiar na Instalação e Configuração do Java Enterprise Edition 5, 6 e suas atualizações</p> <p>7.6 Apoiar na Instalação e Configuração do Framework de Desenvolvimento</p> <p>7.7 Apoiar na Instalação e Configuração do Hibernate</p>

	adequado dos padrões adotados, bem como, o desenvolvimento de componentes de softwares para uso nas aplicações e sistemas do MTPA.	<p>7.8 Apoiar na Instalação e Configuração do Subversion ou outro equivalente</p> <p>7.9 Apoiar na Instalação e Configuração do Jboss Application Server</p> <p>7.10 Apoiar na Instalação e Configuração do PHP 5 e atualizações</p> <p>7.11 Apoiar na Definição de Framework PHP e suas atualizações</p> <p>7.11 Apoiar na Modelagem e Implementação de Soluções Integradoras</p> <p>7.12 Apoiar na Construção de Componentes DELPHI</p> <p>7.13 Apoiar na Instalação e Configuração da linguagem DELPHI</p>
<b>8. APOIO NA GESTÃO DE CONTEÚDO</b>	Os serviços de Apoio a Gestão de Conteúdo consistem no desenvolvimento de novos sítios, portais, web sites ou hotspots, em plataformas tecnológicas de CMS (Content Management System – Sistema de Gerenciamento de Conteúdo), ou melhorias naqueles em produção a partir de especificações estabelecidas e validadas pelo MTPA.	<p>8.1 Apoiar na Definição do Escopo</p> <p>8.2 Apoiar na Especificação de Requisitos</p> <p>8.3 Apoiar na Especificação da Arquitetura da Informação</p> <p>8.4 Apoiar na Especificação da Identidade Visual</p> <p>8.5 Apoiar na Especificação de Perfis e Permissões de Usuários</p> <p>8.6 Apoiar na Especificação da Solução Técnica</p> <p>8.7 Apoiar na Especificação da Arquitetura de Software</p> <p>8.8 Apoiar no Projeto de Banco de Dados</p> <p>8.9 Apoiar na Integração e Implantação</p> <p>8.10 Apoiar na Especificação de Migração dos Dados</p> <p>8.11 Apoiar na Implementação da Arquitetura de Software</p> <p>8.12 Apoiar na Implementação da Interface</p> <p>8.13 Apoiar na Implementação da Customização</p> <p>8.14 Apoiar na Execução da Migração de Dados</p> <p>8.15 Apoiar na Execução do Treinamento e Material de Suporte</p>
<b>9. APOIO À GESTÃO DE OPERAÇÕES</b>	Os serviços de Apoio à Gestão de Operações incluem tarefas relacionadas ao desenvolvimento e manutenção de Sistemas. Cabe à Gestão de Operações garantir a adoção aos padrões de Desenvolvimento de Software pela equipe de Sustentação e Manutenção de Sistemas. O serviço de Apoio à Gestão de Operações lidera a equipe	<p>9.1 Apoiar na Priorização de Demandas Evolutivas e Corretivas</p> <p>9.2 Apoiar na Elaboração do Escopo Preliminar</p> <p>9.3 Apoiar na Especificação do Serviço</p> <p>9.4 Apoiar na Elaboração do Pré Projeto</p> <p>9.5 Apoiar na Elaboração do Parecer e Análise de Viabilidade</p> <p>9.6 Apoiar na Elaboração do Modelo Conceitual de Dados e Dicionário Conceitual</p>

	de sustentação apoiando na priorização e execução e acompanhamento das demandas.	
<b>10. APOIO NO DESIGN DE INTERFACE</b>	Os serviços de Apoio a Design de Interface incluem tarefas relacionadas ao desenvolvimento e manutenção de padrões de designer de software, garantia do uso adequado dos padrões definidos e adotados pelo MTPA.	10.1 Apoiar na Elaboração de Interfaces Web e Mobile
		10.2 Apoiar na Elaboração de Componentes de Design
		10.3 Apoiar na Manutenção de Padrões de Design

# PARTE VI – GLOSSÁRIO



## 11 GLOSSÁRIO

Para auxiliar o entendimento dos envoltimentos em desenvolvimento de sistemas, apresentamos alguns conceitos:

### A

**Análise:** (UML) Parte do processo de desenvolvimento de software em que o principal objetivo é construir um modelo do domínio do problema. A análise foca no o que fazer. O projeto foca no como fazer.

**Análise de Impacto:** Uma análise de impacto avalia os efeitos que uma mudança proposta terá sobre um sistema ou parte interessada.

**Arquitetura:** Descreve os subsistemas e componentes que compõem o sistema e relacionamento entre eles, incluindo a definição dos mecanismos fundamentais (linguagens, plataformas, protocolos, padrões etc.) que serão utilizados para seu desenvolvimento.

**Artefato:** Representa um produto concreto produzido, modificado ou utilizado pelas atividades de um processo. Um artefato pode ser um modelo, um componente de um modelo ou um documento. Um artefato pode conter outros artefatos.

**Atividade:** Representa um conjunto de passos e tarefas que um profissional, que desempenha o papel responsável por aquela atividade, deve executar para gerar algum resultado.

**Ator:** (RUP) Alguém ou algo externo ao sistema ou negócio que interage com o sistema ou negócio. (UML) Um conjunto coerente de papéis que usuários de casos de uso exercem quando interagem com ele. Um ator tem um papel para cada caso de uso com o qual interage.

### B

**Build:** Versão operacional de um sistema ou parte dele que demonstra um conjunto das funcionalidades a serem oferecidas pelo produto final.

**Business Case:** É uma forma de justificar o investimento para aprovar um projeto estratégico que agrega valor ao negócio da empresa (Business Case, 2011).

### C

**Caso de uso:** (RUP) Uma sequência de ações que um sistema deve realizar para apresentar um resultado de valor mensurável para o usuário.

(UML) Especificação de uma sequência de ações, incluindo suas variações, que um sistema (ou outra entidade) pode realizar, interagindo com seus atores.

**Cenário:** Instância de um caso de uso. Um dos fluxos possíveis de passos em um caso de uso.

**Classe:** (UML) Descrição de um conjunto de objetos que compartilham os mesmos atributos, operações, relacionamentos e semântica.

**Código-fonte:** Artefato da MDS que representa qualquer trecho de código implementado ao longo do desenvolvimento do sistema. Não possui modelo.

**Compilação do sistema (Build):** Artefato da MDS que representa o sistema executável compilado. Não possui modelo.

**Componente:** (UML) Parte não trivial de um sistema, quase independente e substituível que realiza uma função clara no contexto de uma arquitetura bem definida.

**Configuração:** Requisitos, projeto e implementação que definem uma versão específica de um sistema ou componente.

**Controle de mudanças:** Atividade de controlar e rastrear as modificações.

## D

**Diagrama de caso de uso:** Um tipo de diagrama definido pela UML que captura todos os atores e casos e uso envolvidos com um sistema ou produto.

**Diagrama:** Representação gráfica de todo, ou parte de um modelo.

**Dicionário de Dados:** Um modelo de análise descrevendo as estruturas e dados e atributos necessários para o sistema.

**Documento de Arquitetura de Software:** Artefato da MDS que descreve as principais decisões de projeto tomadas pela equipe de desenvolvimento e os critérios considerados durante a tomada destas decisões.

**Documento de Cadastro de Iniciativa (DCI):** Este documento formaliza a solicitação da área gestora para a execução de algum serviço, no que tange à Tecnologia da Informação.

**Documento de Visão de Negócio:** Artefato da MDS que estabelece as características do produto com base nas necessidades da área de negócio e fornece informações para o Termo de Abertura do Projeto.

## E

**Entrega:** Qualquer produto ou serviço de trabalho único e verificável que uma parte concordou em entregar.

**Escopo do produto:** (PMBOK 96) Aspectos e funções que devam ser incluídos no produto ou serviço.

**Escopo do projeto:** (PMBOK 96) Trabalho que deve ser feito com a finalidade de entregar um produto de acordo com os aspectos e as funções especificadas.

**Estrutura Analítica do Projeto (EAP):** Uma decomposição hierárquica orientada às entregas do trabalho a ser executado pela equipe do projeto para atingir os objetivos do projeto e criar as entregas requeridas. Ela organiza e define o escopo total do projeto.

## F

**Fase:** Espaço de tempo entre dois marcos significativos do projeto, durante o qual objetivos são atingidos, artefatos elaborados e decisões sobre passar ou não para a próxima fase são tomadas.

**Fluxo de Atividades:** Apresenta a sequência e a dependência entre as atividades do projeto ao longo do tempo.

**Framework:** É uma estrutura genérica em um domínio específico que pode formar a base de uma família de aplicações. Os frameworks são geralmente implementados como um conjunto de classes concretas e abstratas, especializadas e instanciadas para criar uma aplicação.

**Funcionalidade:** (RUP) Serviço oferecido por um sistema, observável externamente, que satisfaz uma necessidade do *stakeholder*.

## G

**Gerência de configuração:** Processo de apoio cujo objetivo é identificar, definir e estabelecer baselines de itens de configuração.

**Gerência de requisitos:** Abordagem sistemática para levantar, organizar e documentar os requisitos de um sistema, e estabelecer e manter um acordo entre o cliente e a equipe do projeto sobre as alterações nestes requisitos.

**Glossário:** Artefato da MDS que registra termos e definições específicos ao domínio do sistema e do projeto.

**Guia:** Descrição de como realizar determinada atividade ou como trabalhar com determinado artefato, descrevendo, inclusive, seu processo de criação e revisão.

## I

**Implantação:** Atividade da MDS representada pela área de Implantação da COPSI e está fora do escopo da equipe de desenvolvimento do sistema.

**Incremental:** Característica de uma estratégia de desenvolvimento iterativa em que o sistema é desenvolvido com a adição de novas funcionalidades a cada iteração.

**Interface:** (UML) Conceito abstrato para uma coleção de operações utilizada para especificar o serviço de uma classe ou componente.

**Item de configuração:** Artefatos do sistema, arquivos ou conjunto de arquivos, produzidos ou utilizados como insumos em suas atividades.

**Iteração:** Sequência distinta de atividades, com planejamento e critérios de avaliação estabelecidos, que resulta em uma versão (interna ou externa) do produto.

**Iterativo:** Processo que se repete diversas vezes para se chegar a um resultado e a cada vez gera um resultado parcial que será usado na vez seguinte.

## J

**Jazz Team Server – IBM:** Aplicativo de gerenciamento de demandas e de configuração.

## M

**Marco:** São pontos significativos de um projeto. Pontos em que uma iteração é formalmente encerrada.

**Matriz de Rastreabilidade:** Artefato da MDS que registra todos os requisitos do sistema levantados junto aos usuários gestores e finais.

**Método:** Maneira de dizer, de fazer, de ensinar uma coisa, segundo certos princípios e em determinada ordem.

**Métrica:** É um nível quantificável de um indicador que uma organização deseja alcançar em um ponto específico do tempo.

**Modelo de Casos de Uso:** Artefato da MDS que descreve toda a visão funcional do sistema através de seus atores e casos de uso. Modelo: Descrição completa de um sistema sob determinada perspectiva (por completa entenda-se que não é necessária nenhuma outra informação para a compreensão daquela perspectiva do sistema).

## N



**Necessidade do Negócio:** Um tipo de requisito de negócio de alto nível que é uma declaração de um objetivo do negócio ou um impacto que a solução deverá trazer ao ambiente do negócio.

## P

**Pacote de Distribuição:** Empacotamento físico de uma versão do sistema, incluindo o sistema executável, kits de instalação, manuais do sistema, documentação do projeto, bases de dados para carga etc.

**Pacote:** Mecanismo de propósito genérico para organizar elementos em grupos. Pacotes podem ser aninhados em outros pacotes.

**Papéis:** Definem o comportamento e responsabilidades de profissionais que participam do desenvolvimento do projeto.

**Pós-condição:** Descrição textual de uma restrição no sistema quando um caso de uso ou um caso de teste termina.

**Pré-condição:** Descrição textual de uma restrição no sistema para que um caso de uso ou um caso de teste possa ser iniciado.

**PGDS:** Processo de Gerenciamento e Desenvolvimento de Software.

**Processo de desenvolvimento:** Conjunto de passos relativamente ordenados executados com um determinado propósito durante o desenvolvimento de um sistema.

**Processo de Engenharia de Software:** Um processo de Engenharia de Software é formado por um conjunto de passos de processo parcialmente ordenados, relacionados com artefatos, pessoas, recursos, estruturas organizacionais e restrições, tendo como objetivo produzir e manter os produtos de software finais requeridos.

**Processo Unificado:** O processo unificado (*Unified Process* - UP) de desenvolvimento de software é o conjunto de atividades necessárias para transformar requisitos do usuário em um sistema de software.

**Processo:** Conjunto de passos relativamente ordenados executados com a intenção de se atingir um objetivo.

**Produto:** Software resultante do desenvolvimento e alguns dos artefatos relacionados (documentação, material de treinamento, etc.).

**Projeto:** (UML) Parte do processo de desenvolvimento de software em que o principal objetivo é decidir como o sistema será implementado.

(PMBOK) Um esforço temporário com a finalidade de criar um produto/serviço único.

**Proposta de Projeto:** Fase da MGP que abrange o entendimento do problema e suas necessidades. A partir dessas informações é definido o escopo e o tamanho funcional do projeto, cujo objetivo é conseguir concordância de todos os *stakeholders* sobre a continuidade do projeto.

## Q

**Qualidade:** Grau com que um conjunto de características inerentes (produto, sistema ou processo) satisfaz a requisitos.

## R

**Regra de negócio:** (RUP) Declaração de uma política ou condição que deve ser satisfeita no contexto do negócio. (BRG) Instrução que define ou restringe algum aspecto do negócio. Uma diretiva intencionada a influenciar ou guiar o comportamento de um negócio, para suportar políticas formuladas em resposta a oportunidades, ameaças e pontos fortes e fracos.

**Relacionamento:** (UML) Conexão semântica entre elementos de um modelo.

**Release:** Todo, ou parte do produto final, objeto de avaliação ao final de uma fase do processo de desenvolvimento. É composta por uma versão estável e executável do produto junto com os artefatos necessários a sua utilização, podendo ser interna ou externa.

**Repositório:** (UML) Local de armazenamento de documentos, modelos, interfaces e implementações.

**Requisito:** (IEEE 83) Condição ou capacitação que um sistema precisa atender ou ter para satisfazer um contrato, padrão, especificação, ou outro documento formalmente estabelecido.

(RUP) Condição ou capacidade a qual o sistema deve estar em conformidade, derivada diretamente das necessidades dos stakeholders ou definida em um contrato, padrão, especificação ou outro tipo de documento formal. (UML) Funcionalidade, propriedade ou comportamento desejado para um sistema. (Abbot86) Função, restrição, ou outra propriedade que precisa ser fornecida, encontrada, ou atendida para satisfazer as necessidades do usuário do futuro sistema.

**Requisito de sistema:** (IEEE 90) Condição ou capacitação que deve ser atingida ou possuída por um sistema ou componente de sistema para satisfazer uma condição ou capacitação requerida por um cliente ou usuário final. (RUP) Especificação de um comportamento do

sistema ou do seu ambiente, observável externamente, por exemplo: entradas, saídas, funções, atributos etc.

**Requisito de software:** (IEEE 90) Condição ou capacitação que precisa ser contemplada pelo software, necessitada pelo usuário ou cliente para resolver um problema ou alcançar um objetivo.

**Revisão:** Grupo de atividades executadas com o propósito de localizar defeitos potenciais e verificar a qualidade de um conjunto de artefatos.

**Risco:** (PMBOK 96) Possibilidade de sofrer perdas.

(RUP) Preocupação progressiva ou iminente que tem grande probabilidade de afetar adversamente o sucesso do projeto.

**RUP:** *Rational Unified Process* (Processo Unificado Rational)

## S

**Scrum:** Scrum é uma metodologia ágil para gestão e planejamento de projetos de software.

**Scrum Master:** atua como facilitador do Daily Scrum e torna-se responsável por remover quaisquer obstáculos que sejam levantados pela equipe durante essas reuniões.

**Scrum Team:** O Scrum Team é a equipe de desenvolvimento.

**Sprint Backlog:** O Sprint Backlog é uma lista de tarefas que o Scrum Team se compromete a fazer em um Sprint.

**Sistema:** Uma coleção de elementos inter-relacionados que interagem para atingir um objetivo. Elementos do sistema podem incluir hardware, software e pessoas. Um sistema pode ser um sub elemento (ou subsistema) de outro sistema.

**Software:** (Pressman) Instruções (programas de computador) que, quando executadas, produzem a função e o desempenho desejados. (ISO 9001) Criação intelectual compreendendo os programas, procedimentos, regras e qualquer documentação correlata à operação de um sistema de processamento de dados.

## T

**Template:** Gabarito. Estrutura pré-definida para um artefato.

## U

**UML:** *Unified Modeling Language* - Linguagem de Modelagem Unificada. Linguagem para visualizar, especificar, construir e documentar artefatos de um sistema de software.

**UST:** (TCU) Unidade de Serviços Técnicos ou denominações correlatas. Essa técnica consiste em listar uma série de serviços na forma, por exemplo, de um catálogo e valorá-los a fim de pagar mediante a conclusão.

**Usuário Final:** Representa o usuário do sistema propriamente dito. Este será o usuário que utilizará o sistema em seu dia-a-dia e sentirá na prática os benefícios (ou prejuízos) operacionais consequentes da implantação do sistema.

## V

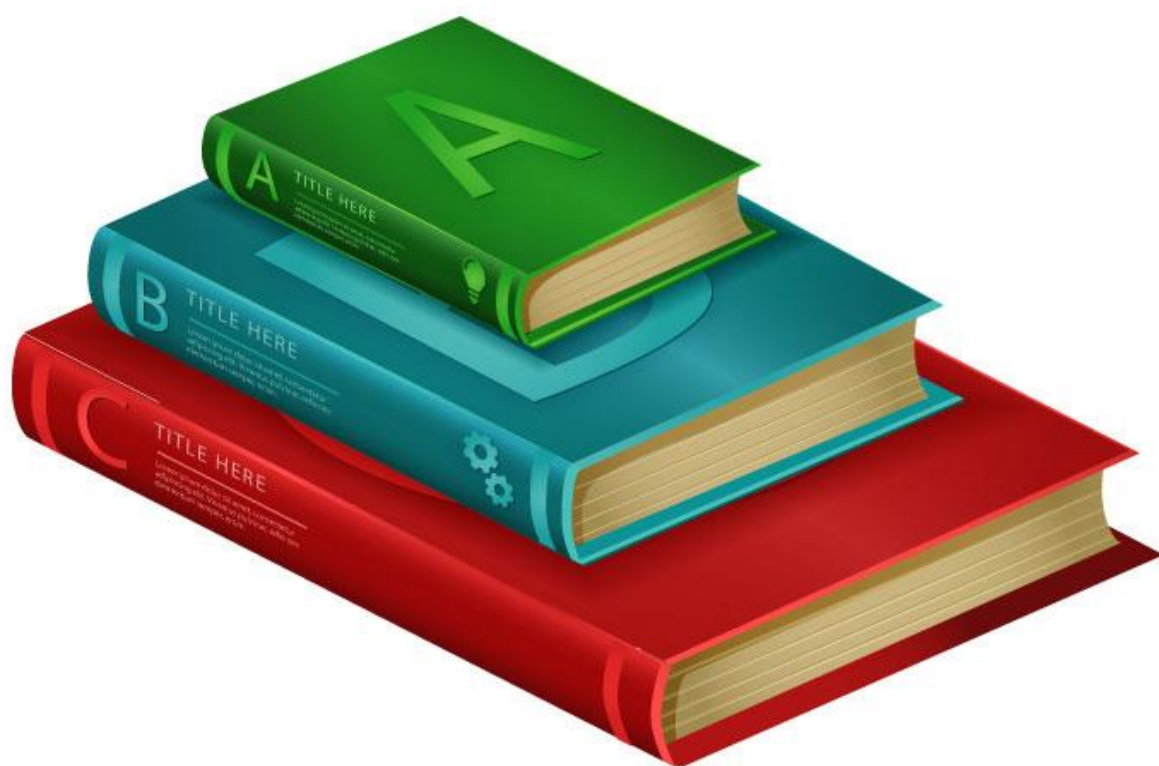
**Verificação:** (CMMI) Assegurar que os produtos de trabalho selecionados satisfazem seus requisitos especificados.

**Validação:** (CMMI) Demonstrar que o produto ou componente do produto satisfazem seu uso pretendido quando colocado no ambiente pretendido.

**Versão do Sistema (Release):** Artefato da MDS que representa o sistema executável ao final de uma fase do projeto, representando um dos grandes marcos do ciclo de vida do projeto. Não possui modelo.

**Visão:** Ponto de vista do cliente ou usuário do produto a ser desenvolvido, especificada no nível de necessidades do usuário e funcionalidades do sistema.

# PARTE VII – REFERÊNCIAS BIBLIOGRÁFICAS



## 12 REFERÊNCIAS BIBLIOGRÁFICAS

SECRETARIA DE TECNOLOGIA DA INFORMAÇÃO – Ministério do Planejamento, Orçamento e Gestão (STI/MP). MGP: Metodologia de Gerenciamento de Projetos do SISP. Versão 1.0 2011.

Disponível em:

<[http://www.planejamento.gov.br/secretarias/upload/Arquivos/servidor/publicacoes/publicacao\\_slti\\_mgp-sisp\\_versao\\_1.pdf](http://www.planejamento.gov.br/secretarias/upload/Arquivos/servidor/publicacoes/publicacao_slti_mgp-sisp_versao_1.pdf)>

COPSI – Ministério dos Transportes. MGP: Metodologia de Gerenciamento de Projetos.

RATIONAL UNIFIED PROCESS

Best Practices for Software Development Teams. Disponível em:

[http://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251\\_bestpractices\\_TP026B.pdf](http://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251_bestpractices_TP026B.pdf)

IBM RATIONAL UNIFIED PROCESS (RUP)

Disponível em:

<<http://www-01.ibm.com/software/rational/rup/>>

ENGENHARIA DE SOFTWARE

PRESSMAN, Roger S. Engenharia de software. 6ª ed. Porto Alegre: Bookman, 2006.

ENGENHARIA DE SOFTWARE

SOMMERVILLE, Ian. Engenharia de Software. 8ª ed. São Paulo: Pearson Addison-Wesley, 2007.

BIZAGI PROCESS MODELER – Software utilizado no desenvolvimento dos fluxos.

Disponível em:

<[http://www.bizagi.com/index.php?option=com\\_content&view=article&id=112&Itemid=122](http://www.bizagi.com/index.php?option=com_content&view=article&id=112&Itemid=122)>

Definições do Scrum

Disponível em:

<<http://www.brq.com/metodologias-ageis/http://www.desenvolvimentoagil.com.br/scrum/>>

