

METODOLOGIA DE DESENVOLVIMENTO PROCERGS – MDP

Processo de Desenvolvimento de Sistemas

1. Objetivo

MDP - Metodologia de Desenvolvimento PROCERGS - é uma estrutura básica de definição de processos de desenvolvimento.

Compreende roteiros, modelos, guias e ferramentas que dão suporte a todo o ciclo de vida de desenvolvimento do *software*.

Tem como objetivo garantir:

- a clareza de papéis e responsabilidades;
- atendimento às necessidades do cliente/mercado;
- participação do cliente em todas as fases;
- qualidade do produto;
- planejamento e acompanhamento dos projetos.

2. Ciclos de Desenvolvimento

As etapas do processo de desenvolvimento da **PROCERGS** podem ser gerenciadas utilizando um **Ciclo de Desenvolvimento Preditivo** ou através de um **Ciclo Ágil**.

Todas as etapas são apoiadas pelo **Processo de Gerência de Projetos**, pelo **Processo de Testes** e pelo **Processo de Garantia da Qualidade**.

A cada projeto a equipe de desenvolvimento da **PROCERGS** define o ciclo de desenvolvimento a ser utilizado (Preditivo ou Ágil).

Nos projetos de desenvolvimento, manutenção e migração de sistemas realizados com Fábrica Externa deve ser somente utilizado ciclo ágil.

2.1 Ciclo Preditivo

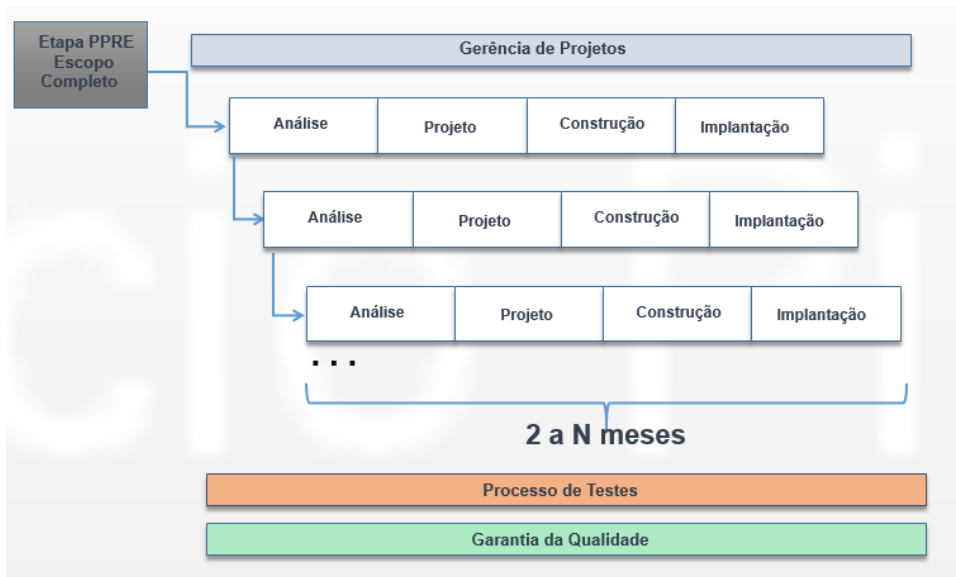


Fig. 1 – Visão do Ciclo Preditivo

O Ciclo Preditivo possui as seguintes características:

- planejamento antecipado;
- entregas por módulos ou pacotes de entregas acordados com o cliente. Cada entrega é um subconjunto do produto final em desenvolvimento e cresce de modo incremental;
- a implementação de um módulo ou pacote passa por todas as etapas do ciclo de desenvolvimento;
- se ocorrerem mudanças o prazo é estendido;
- as prioridades, na maioria das vezes, são definidas pelo Gerente do Projeto baseando-se nas necessidades do cliente.

2.2 Ciclo Ágil

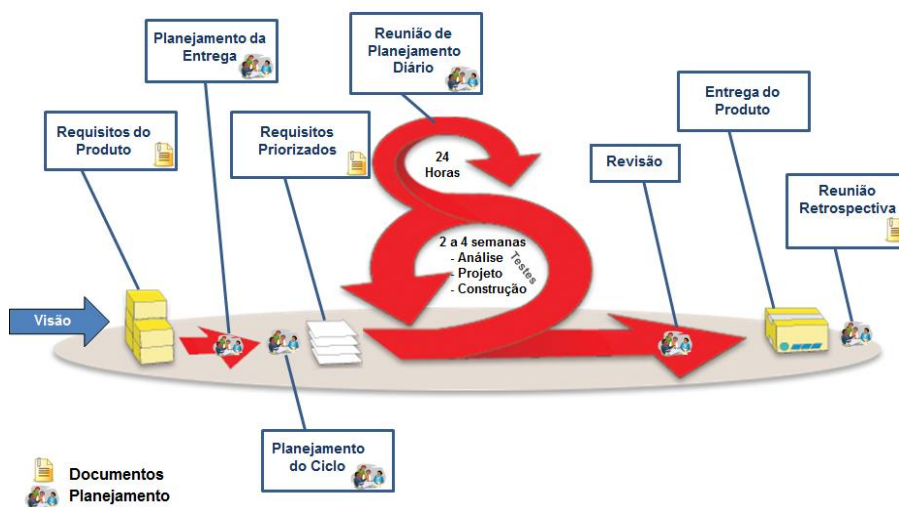


Fig. 2 – Visão do Ciclo Ágil

O Ciclo Ágil tem as seguintes características:

- foco no valor do negócio;
- planejamento contínuo e adaptativo;
- cliente faz parte da equipe e prioridades são definidas pela equipe;
- pequenas entregas com prazos rígidos.

Veja detalhes do ciclo ágil PROCERGS com fábrica de software no Anexo 07 deste Edital.

3. Etapas do Processo de Desenvolvimento de Sistemas

Baseado na análise da demanda é elaborado um Documento de Projeto na Etapa de **Projeto Preliminar**. Este documento interpreta as necessidades identificadas de forma a produzir uma Proposta de Projeto, que será utilizada e melhor detalhada nas Etapas de **Análise, Projeto, Construção, Testes (Planejamento Macro de Teste, Planejamento e Especificação de Teste, Execução de Teste e Fechamento) e Documentação de usuário**, bem como apresenta as premissas para **Implantação e Manutenção**.

Os roteiros do Processo de Desenvolvimento definem as atividades que devem ser executadas para realizar cada uma das etapas.

Importante: O quadro resumo, no final deste documento apresenta uma visão geral do **processo, ferramentas, documentos de apoio e produtos de trabalho** que devem ser entregues, ao final de cada etapa.

Todas as etapas são apoiadas pelo **Processo de Gerência de Projetos**, pelo **Processo de Testes** e pelo **Processo de Garantia da Qualidade**.

3.1. Etapa de Projeto Preliminar

Esta etapa busca levantar as necessidades, problemas e requisitos de uma maneira objetiva para construção de uma proposta ao cliente.

Atividades:

- **Abertura do Projeto Preliminar:** visa obter um conjunto de informações para a inicialização da etapa de Projeto Preliminar. Deve-se iniciar o Documento de Projeto com base no modelo disponível na MDP;
- **Definição do escopo:** o objetivo é avaliar a demanda inicial, destacando os principais elementos, tais como: necessidades, agentes envolvidos, requisitos, escopo e outras variáveis relevantes que proporcionem uma visão geral da demanda. Deve-se registrar no Documento de Projeto os principais processos, os requisitos funcionais (os casos de uso/histórias de usuários preliminares), não funcionais (performance, confiabilidade, robustez, interação, usabilidade e aspectos legais) e requisitos de dados (mapeados com um modelo E-R conceitual ou diagrama de classes conceitual). Os requisitos funcionais preliminares devem conter uma descrição breve, especificando e caracterizando o comportamento das principais funções do sistema;
- **Definição da Solução:** tem por objetivo pesquisar, definir e descrever, caso existam, as alternativas de solução baseando-se nos problemas, necessidades, delimitadores, requisitos e riscos levantados durante a atividade de definição do escopo. Caso não

exista a necessidade de apresentar alternativas, deve-se descrever a solução avaliando-a, considerando os riscos, gerando esquemas e desenhos que facilitem o entendimento;

- **Planejamento do Desenvolvimento:** o objetivo é definir a estratégia de desenvolvimento e planejar a próxima etapa do projeto. Neste planejamento deve ser envolvida também a área operacional da **PROCERGS**.

Obs. Preferencialmente, esta etapa será desenvolvida pela PROCERGS.

3.2. Etapa de Análise

Após a etapa de Projeto Preliminar, uma vez aceita a proposta, a equipe inicia a elaboração da Análise do Sistema. O objetivo desta etapa é transformar os requisitos identificados em um conjunto de requisitos técnicos do sistema desejado, que serão a **entrada das Etapas de Planejamento de Testes e Projeto**.

A ordem para execução das atividades deve variar de acordo com a abordagem que será utilizada (Orientado ao Planejamento ou Ágil), contudo todas as tarefas descritas no *Roteiro de Projeto* para essa atividade são executadas para ambas as abordagens.

Durante esta etapa devem ser documentadas as informações e decisões em um documento único de projeto, registrando as evoluções existentes no documento criado na etapa de Projeto Preliminar.

Atividades:

- **Abertura do projeto de sistema:** inicia o projeto do sistema e visa obter um conjunto de informações para a sua inicialização;
- **Análise do sistema:** identifica e detalha os conceitos relativos aos requisitos da aplicação a ser desenvolvida. Nesta atividade as seguintes tarefas devem ser realizadas:
 - **Analisar os requisitos do projeto:** visa obter uma relação dos requisitos mais abrangente e completa. É realizada uma revisão dos requisitos definidos na etapa de Projeto Preliminar e um levantamento mais detalhado a partir de reuniões com o cliente;
 - **Refinar os requisitos:** Os requisitos devem ser detalhados de modo a facilitar o entendimento do time.
 - **Refinar os Casos de Uso:** através do diagrama de casos de uso, deve-se mostrar o que o sistema irá disponibilizar para o usuário. O objetivo é capturar o comportamento desejável do sistema e não como será implementado. O diagrama de casos de uso preliminar (elaborado na etapa de Projeto Preliminar) deve servir como base para o refinamento. Junto com o diagrama, deve ser feito o detalhamento dos casos de uso: passo a passo para realização;
 - **Refinar as Histórias de usuário:** as histórias de usuários são revisadas e complementadas com os critérios de aceitação. Os critérios de aceitação guiam a validação da história de usuário, e o refinamento com base nas características INVEST auxilia na boa escrita das histórias.
 - **Realizar a modelagem lógica:** elaborar o Modelo Lógico de Dados, representando as entidades, relacionamentos, cardinalidades, atributos e restrições do modelo;

- **Elaborar o protótipo:** A construção do protótipo facilita o entendimento do problema e seus requisitos, com a participação do usuário, esclarecendo os aspectos confusos e desconhecidos do sistema. São os desenhos das interfaces visuais e dos relatórios. Para sistemas *Web* as interfaces devem seguir os padrões **PROCERGS** definidas no documento de padrão de interface.
- **Projeto da arquitetura do sistema:** o objetivo é realizar a análise de integrações do sistema, a definição dos requisitos de *software*, *hardware* e de rede e procedimentos necessários para implantação e operação do sistema.

3.3. Etapa de Projeto

O objetivo desta etapa é fornecer um projeto para o *software* que implemente e possa ser verificado, com base em seus requisitos de *software*.

Durante esta etapa devem ser documentadas as informações e decisões em um documento único de projeto, registrando as evoluções existentes no documento criado na etapa de Projeto Preliminar ou Análise.

Nesta etapa é gerado o Modelo E-R Físico a partir do Modelo Lógico elaborado na Etapa de Análise. Além dos scripts necessários para a criação ou atualização do banco de dados.

Atividades:

- **Projeto da arquitetura do *software*:** envolve um conjunto de decisões estratégicas de projeto. A existência de um padrão de arquitetura, *frameworks* de desenvolvimento e uma aplicação modelo, fornecem suporte ao projetista, agregando qualidade e produtividade ao projeto. Nesta atividade deve ser definida a arquitetura do *software* a ser desenvolvido. Os Projetos que utilizam os frameworks padrões da **PROCERGS** possuem a alternativa de utilizar os geradores disponíveis para definir: as classes de interface e segunda camada necessárias. Fazem parte desta atividade a geração do Modelo ER-Físico e a criação dos scripts necessários para a criação ou atualização do banco de dados;
- **Projeto detalhado do *software*:** O objetivo principal desta atividade é detalhar as unidades de *software* em um nível que seja de fácil entendimento pelos construtores, para se obter qualidade e produtividade nas etapas seguintes. Deve ser especificado os Detalhes Técnicos das histórias de Usuário e Casos de Uso. Os projetos que utilizam os frameworks padrões da **PROCERGS** possuem a alternativa de definir a integração entre os diversos objetos do *software* (classes e métodos), utilizando os geradores disponíveis para estas tecnologias.

3.4. Etapa de Construção

Esta etapa tem como objetivo a construção do produto de *software*.

Atividades:

- **Codificação de *software*:** o objetivo é desenvolver e documentar cada unidade de *software* e banco de dados, bem como os procedimentos de testes individuais das unidades. É muito importante que os padrões definidos pela **PROCERGS** sejam utilizados, bem como a utilização de alguns padrões mundiais;
- **Práticas de Integração Contínua, Entrega Contínua e DevOps:**

Sempre que aplicável, o desenvolvimento deverá adotar práticas de Integração Contínua e Entrega Contínua (CI/CD) e princípios DevOps, visando automação, rastreabilidade, qualidade e padronização dos processos de construção, testes e implantação, em conformidade com as diretrizes técnicas da **PROCERGS**;[AS1]

- **Elaboração dos componentes de comunicação:** O objetivo desta atividade é desenvolver os componentes de comunicação definidos no projeto e complementar as informações para operação do produto (*batch* e *online*). Nesta atividade também está a tarefa de realizar os cadastrados necessários nos Sistemas de Controle de Acessos (SOE ou SCA) da **PROCERGS**.
- **Integração de software:** o objetivo é executar a integração das unidades e componentes de *software*;
- **Testes de Sistema:** O objetivo desta atividade é executar os testes de sistema, após terem sido integradas todas as unidades, componentes e módulos do sistema. Veja no Processo de Testes, detalhes sobre esta atividade.

3.5. Etapa de Implantação

O objetivo desta etapa é garantir que o produto desenvolvido fique disponível para seus usuários finais. Ao final desta etapa, os objetivos planejados para entrega deverão ter sido alcançados e, se for a última entrega, o projeto será finalizado. Faz parte da implantação a **homologação** do produto entregue de acordo com as expectativas dos usuários (cliente).

Objetivos desta etapa:

- Definir critérios de aceitação do produto;
- homologação do produto entregue de acordo com as expectativas dos usuários (cliente);
- operação paralela com os sistemas legados que serão substituídos, quando aplicável;
- conversão de base de dados, quando aplicável;
- treinamento de usuários;
- empacotamento e distribuição do sistema;
- definir critérios de aceitação do produto;
- obter aceite do produto entregue.

Atividades:

- **Planejamento da Implantação:** deve assegurar que haja um planejamento das atividades de implantação e que todas as pessoas-chave sejam envolvidas no processo;
- **Preparação do Ambiente:** deve assegurar que todos os pré-requisitos necessários para a implantação no ambiente de homologação e produção sejam providenciados de forma adequada;
- **Preparação dos Dados:** deve assegurar que a base de dados seja devidamente gerada e populada (incluindo conversão/migração, quando aplicável) e que os controles de acesso estejam adequados;
- **Instalação:** deve garantir que a instalação do *software* ocorra conforme o designado no contrato. Os recursos e informações necessários devem ser definidos e estarem disponíveis;
- **Acompanhamento:** acompanhar o período inicial de utilização do *software* no ambiente de produção;

- **Treinamento:** planejar e realizar o treinamento do *software* nos diferentes grupos de usuários, bem como discipliná-los na correta utilização do *software*;
- **Encerramento da implantação:** marca o final da etapa de implantação e o início da etapa seguinte, a de manutenção.

Observações Importantes:

- Na Homologação com o Cliente é de responsabilidade da **PROCERGS** as atividades de: Planejamento da Implantação, Preparação do Ambiente, Preparação dos Dados, Instalação, Acompanhamento e Treinamento.
- Na Homologação com o Cliente é de responsabilidade da **CONTRATADA** o acompanhamento da Homologação pela equipe responsável, juntamente com a **PROCERGS**, visando esclarecer dúvidas e agilizar a correção das não conformidades que surgirem. A **CONTRATADA** deverá apoiar a **PROCERGS** na Preparação do Ambiente e Preparação dos Dados.

3.6. Documentação de Usuário

Esta atividade compreende a execução dos serviços especializados de documentação, a elaboração do Manual do Usuário e *Help On-line*, adequados às necessidades e ao interesse dos usuários dos produtos de *software* desenvolvidos pela **PROCERGS** ou pela **CONTRATADA**.

Atividades:

- Análise dos produtos de trabalho das Etapas de Análise e Projeto;
- Elaboração de documentação detalhada para usuários de sistemas de *software* na forma de Manual do Usuário e *Help On-line*;

A documentação detalhada deve ser baseada nos documentos de especificação do sistema e em sua navegação. Deve conter a descrição das funcionalidades e dos processos especificados pelo Gerente do Projeto ou *Scrum Master* da **PROCERGS**. Também deve conter o significado, finalidade e origem dos campos da tela.

4. MDP com UX –User Experience

A área de UX está integrada em diversos momentos do desenvolvimento, contribuindo para a conexão entre as etapas:

- **Planejamento:** todo desenvolvimento de um novo *software*/nova funcionalidade deve ser planejado para que os profissionais sejam envolvidos desde o começo. O profissional de UX precisa entender as necessidades do *software* para definir onde irá atuar e quais técnicas serão aplicadas.
- **Análise:** o profissional de UX deve participar de todo o entendimento do problema junto aos usuários e não somente com as explicações de analistas de negócios/funcionais/líderes de produtos etc. A perspectiva de UX busca trazer o modelo mental do usuário para o *software*, tornando-o mais intuitivo.
- **Construção / Testes:** o profissional de UX deve acompanhar o desenvolvimento para garantir a integridade e padronização do que foi proposto e redefinir a proposta caso a equipe de desenvolvimento encontre limitações técnicas. Um problema muito frequente é que os desenvolvedores, quando não conseguem desenvolver algo proposto pela área de UX, tendem a substituir elementos da proposta, prejudicando a usabilidade. Por isso, a

área de UX também deve estar bem próxima à área de desenvolvimento para garantir que o trabalho anterior não seja prejudicado.

- **Homologação:** O profissional de UX pode acompanhar os usuários durante a homologação do software, apoiando a identificação e correção de problemas, além de coletar indicadores de satisfação, contribuindo para avaliar a qualidade do produto entregue.

4.1. Analista de Sistemas e a Área de UX

Um analista de sistemas está focado na análise de um sistema como um todo e especificamente em torno do uso de TI para resolver problemas / requisitos de negócios. Um profissional de UX tem os mesmos objetivos, mas seu ponto de vista é a partir das necessidades do usuário e como ele deve interagir com a interface da solução.

Existem semelhanças entre o trabalho do Analista de Sistemas e do UX Designer, ou seja, eles trabalham em conjunto com papéis compatíveis. Os Analistas de Sistemas tendem a ter empatia com os negócios, enquanto o UX tende a ter empatia com os usuários. Em conjunto, eles podem alcançar uma solução que beneficia tanto a empresa quanto o usuário.

Em projetos em que não há a participação do profissional de UX é importante que o Analista de Sistemas tenha conhecimento da área de UX e utilize suas técnicas para garantir que a interface atenda o usuário e que ele tenha satisfação ao interagir com o sistema.

4.2. Etapas do processo contínuo de UX Design

	Projeto Preliminar	Análise	Projeto	Construção	Implantação
Abordagem Preditiva	<ul style="list-style-type: none"> •Objetivos e Hipóteses 	<ul style="list-style-type: none"> •Pesquisa e Análise •Ideias e Protótipos •Testar e Iterar 	<ul style="list-style-type: none"> •Acompanhar Projeto •Realizar ajustes 	<ul style="list-style-type: none"> •Acompanhar desenvolvimento •Realizar ajustes 	<ul style="list-style-type: none"> • Acompanhar Homologação
Abordagem Ágil	Visão	Descoberta		Execução	Entrega
	<ul style="list-style-type: none"> •Objetivos e Hipóteses 	<ul style="list-style-type: none"> •Pesquisa e Análise •Ideias e Protótipos •Testar e Iterar 		<ul style="list-style-type: none"> •Acompanhar desenvolvimento 	<ul style="list-style-type: none"> •Acompanhar Homologação

4.3. Técnicas e Produtos de Trabalho do UX Design

	Objetivos e Hipóteses	Pesquisa e Análise	Ideias e Protótipo	Testar e Iterar
Técnicas	<ul style="list-style-type: none"> •Brainstorming para gerar ideias e soluções •Personas para definir uma hipótese de quem você acha que usaria esse aplicativo •Editor de texto para anotações •Outras 	<ul style="list-style-type: none"> •Benchmarking •Personas •Etnografia (Pesquisas e Comportamentos) •Mapa da empatia •Jornada do Usuário •Outras 	<ul style="list-style-type: none"> •Sessão de brainstorming com colaboradores do projeto •Wireframes •Prototipação •Outras 	<ul style="list-style-type: none"> •Teste de usabilidade qualitativo •Câmera para gravar o teste •Outras
Produtos de Trabalho	<ul style="list-style-type: none"> •Documento de Projeto (Informações para requisitos funcionais). (Obrigatório) 	<ul style="list-style-type: none"> •Documento de Projeto (Informações para requisitos funcionais). (Obrigatório) •Documentos com personas, mapa da empatia e jornada do usuário •Storyboards •Planilha com dados de benchmarking 	<ul style="list-style-type: none"> •Protótipo de alta fidelidade (software) (Obrigatório) •Documento com todas as ideias do processo de brainstorm 	<ul style="list-style-type: none"> •Testes de Usabilidade (Obrigatório) •Protótipo de alta fidelidade Aprovado (software) [Obrigatório]

- **Protótipos de Alta Fidelidade:** são protótipos funcionais com a representação do sistema a ser desenvolvido. Em alguns casos, é possível simular o fluxo completo das funcionalidades, permitindo a interação do usuário como se fosse o produto final. A aparência visual, as formas de navegação e interatividade já são concebidas e aplicadas aos protótipos de alta fidelidade. Seu desenvolvimento é realizado na fase final de definição da interface, utilizando programas de design gráfico.
- **Testes de Usabilidade:** é uma técnica de pesquisa utilizada para avaliar um produto ou serviço. Os testes são realizados com usuários representativos do público-alvo. Cada participante tenta realizar tarefas típicas enquanto o profissional de UX observa, ouve e anota. Utilize o [Modelo UX - Teste de Usabilidade](#) para elaborar o teste e documentar.

Processo de Manutenção de Sistemas

Este processo tem por finalidade auxiliar no desenvolvimento das atividades de manutenção de sistemas de forma a prover mecanismos que auxiliem na avaliação, planejamento, modificações e documentação.

A manutenção corretiva pode ter um caráter urgente, ou seja, deve ser imediatamente solucionada, pois pode comprometer a execução do sistema.

As manutenções adaptativas e evolutivas (melhorias) estão associadas às mudanças em requisitos funcionais (inclusão, alteração ou exclusão de funcionalidades) e não funcionais, em aplicações implantadas.

Abaixo resumo das atividades previstas conforme o tipo de manutenção.

1. Manutenção Corretiva

Atividades:

- **Implementação da Versão Corretiva:** manutenções do tipo corretivas visam corrigir defeitos de funcionalidade, o que inclui acertos emergenciais de código. Quando urgentes são tratadas de maneira diferenciada das demais porque devem solucionar o problema rapidamente.
 - Entradas:
 - ✓ Incidente aberto.
 - Saídas:
 - ✓ Código corrigido no ambiente de produção;
 - ✓ Registro da versão realizado;
 - ✓ Código sincronizado no Gerenciador de Versão;
 - ✓ Incidente encerrado.

Observações:

- para os sistemas desenvolvidos pela **CONTRATADA**, a manutenção corretiva será do tipo Garantia, conforme prazos e demais cláusulas do contrato. Desta forma, estas manutenções corretivas estão compreendidas dentro do custo do desenvolvimento do projeto;
- quando o sistema estiver fora da Garantia, deverá ser estimado e calculado o tamanho do projeto de manutenção corretiva.

2. Manutenções Adaptativas e Evolutivas (Melhorias)

Atividades:

- **Planejamento da Versão:** o objetivo principal desta atividade é planejar as versões conforme a prioridade estabelecida pelo Cliente ou pela **PROCERGS**. Nesta atividade é feita uma verificação detalhada da demanda incluindo a análise do seu impacto e de possíveis soluções para atendê-la.

Esta atividade deve ser realizada para todos os tipos de manutenção, com exceção das corretivas urgentes, que automaticamente possuem prioridade sobre qualquer outro tipo de demanda.

 - Entradas:

- ✓ Demandas registradas;
- ✓ Atas de reunião com cliente (quando houver).
- Saídas:
 - ✓ Registro das versões planejadas;
 - ✓ Mudanças dos requisitos registradas;
- **Implementação:** Esta atividade tem como objetivo realizar as alterações previstas nas versões definidas. Possui as mesmas etapas do Processo de Desenvolvimento de Sistemas: Análise, Projeto, Construção, Testes, Documentação e Implantação.
 - Entradas:
 - ✓ Planejamento das Versões.
 - Saídas:
 - ✓ Produtos de trabalho atualizados;
 - ✓ Registro das Mudanças realizado;
 - ✓ Registro da versão realizado;
 - ✓ Versão liberada no ambiente de produção;
 - ✓ Código sincronizado no Gerenciador de Versão.

Observações Importantes:

- deve-se seguir as orientações e padrões definidos para as etapas de Análise, Projeto, Construção, Testes, Documentação e Implantação do Processo de Desenvolvimento de Sistemas;
- para cada demanda de manutenção a PROCERGS encaminhará para a CONTRATADA o escopo da demanda com o detalhamento compatível com as etapas que serão contratadas. A CONTRATADA deverá realizar a análise de impacto das alterações e repassar para a PROCERGS a estimativa de esforço (Contagem APF). Deverá ser fornecido documento complementar contendo a relação dos requisitos que foram alterados (Casos de Uso, Regras de Negócio, Tabelas e Interfaces).

Processo de Testes

O processo de teste é um conjunto de atividades complexas, de longa duração e essenciais na garantia da qualidade de *software*.

Este processo deve ocorrer em paralelo com algumas das etapas do Processo de desenvolvimento de Sistemas, com o objetivo de prover o gerenciamento, com a adequada visibilidade, da qualidade dos produtos que estão sendo construídos. O processo de testes está subdividido nas seguintes atividades: Planejamento Macro de Testes, Planejamento e Especificação de Testes, Execução de Testes e Fechamento.

O processo contempla, conforme o escopo do projeto, níveis e tipos de testes como: testes unitários, testes de integração, testes de sistema, testes de regressão e testes de aceitação (homologação). Quando aplicável, também podem ser previstos testes não funcionais, como desempenho, segurança e usabilidade.

1. Atividades do Teste na PROCERGS

As atividades e tarefas que devem ser realizadas estão descritas em cada uma das etapas.

Fig. 3 - Papéis e Atividade do Teste na PROCERGS

Atividades:

- **Planejamento Macro de Testes:** esta atividade deve ser iniciada na Etapa de Projeto Preliminar. O processo prevê um plano macro de testes a ser seguido durante todo o ciclo de desenvolvimento. Durante as demais etapas (Análise, Projeto e Construção), deve-se verificar se o projeto está seguindo o que foi planejado no plano macro de testes, e qualquer desvio deverá ser corrigido para se adequar ao planejado ou o plano macro de testes deverá ser atualizado, junto com a justificativa para as atualizações.

Para determinados níveis ou tipos de testes, podem ser definidos planos de testes específicos derivados do plano macro de testes. Esses planos de testes servem como extensão do plano macro de testes, detalhando situações particulares para um determinado nível ou tipo de teste.

Importante: O Gerente de Projeto da PROCERGS deve definir quais requisitos funcionais devem ter os Casos de Testes especificado.

- **Planejamento e Especificação de Testes:** esta atividade deve ser iniciada junto com a Etapa de Análise e finalizada na Etapa de Construção. Os documentos de projeto, requisitos funcionais e não-funcionais, regras de negócio, modelos de ER e projeto de arquitetura do sistema são utilizados critérios de entrada para essa etapa, uma vez que os testes devem ser usados para validar todas as especificações definidas pelo projeto.

As atividades desse processo preveem o levantamento de cenários de testes e escrita dos casos de testes. Deve-se também definir em quais níveis de testes os casos de testes podem ser executados, prioridade dos casos de testes, condições necessárias para iniciar os testes (ex., conjunto de componentes prontos, ambiente de teste disponível, massa de dados preparada, serviços *online* etc.) e critérios de encerramentos do teste conforme a prioridade (ex., 100% dos casos de testes prioritários executados e todos os defeitos encontrados que são referentes a esses casos de testes foram corrigidos, 75% dos casos de testes não-prioritários executados e 80% dos erros referentes a esses casos de testes corrigidos).

- **Execução de Testes:** deve iniciar na Etapa de Construção, quando são executados todos os casos de testes agrupados em um plano de teste, conforme definido na Etapa de Planejamento e Especificação de Testes. Também são executados testes exploratórios (testes que não possuem casos de testes, sendo derivados de casos de testes existentes, experiência do testador com sistemas semelhantes e conhecimento do testador em defeitos). Caso esteja prevista a automação de testes de sistema, a programação dos testes também ocorre nessa etapa.

Considera-se que a automação de testes nessa atividade é a automação de uma aplicação *web*, *desktop* ou *mobile* (i.e., qualquer aplicação com interface gráfica de usuário) utilizando uma ferramenta dedicada a simular os passos que um usuário desempenharia para utilizar uma determinada funcionalidade. Quando aplicável, a automação também pode abranger testes de API/serviços e outros tipos definidos no plano de testes.

- **Fechamento:** esta atividade inicia-se próximo ao encerramento do projeto ou de um Sprint, quando os critérios de encerramento dos testes são atingidos, conforme foi definido no Plano Macro de Testes. Ela ocorre entre a Etapa de Construção e Etapa de Implantação.

As tarefas da atividade de Fechamento se referem ao armazenamento e entrega dos artefatos gerados pelos processos de testes, como planos de testes, casos de testes, registros de defeitos corrigidos e não-corrigidos, códigos de testes unitários e de integração, código de testes automatizados e um registro de conclusão do teste.

Processo de Desenvolvimento de Sistemas – Resumo

Projeto Preliminar / Visão

Produtos de trabalho	Atividades	Roteiros / Modelos	Principais documentos	Ferramentas
<ul style="list-style-type: none"> • Documento de Projeto <ul style="list-style-type: none"> - Modelo Conceitual - Especificação dos Requisitos Funcionais (Casos de Uso / Histórias de Usuário) - Estimativa de desenvolvimento • Planilha de Produtos de Trabalho (Planilha ou anexo no Documento do projeto) • Documento LGPD (Informações de Privacidade) 	<ul style="list-style-type: none"> • Abertura do Projeto Preliminar • Definição do Escopo • Definição da solução • Planejamento do Desenvolvimento • Encerramento do Projeto Preliminar 	<ul style="list-style-type: none"> • RPRO – Roteiro de Projeto • MPRO – Modelo de Projeto • Planilha de Contagem APF • Planilha de Estimativa de Prazo • Modelo Documento LGPD (Informações de Privacidade) 	<ul style="list-style-type: none"> • Guia de Contagem (APF) • Guia do Documento LGPD 	<ul style="list-style-type: none"> • RPM • Rational Rose • LibreOffice / MS Office • ERwin

Testes

<ul style="list-style-type: none"> • Plano Macro de Testes <ul style="list-style-type: none"> - Estratégias de Testes planejada / Testes de Segurança - Infraestrutura de Testes planejada 	<ul style="list-style-type: none"> • Plano Macro de Testes 	<ul style="list-style-type: none"> • Plano Macro de Testes 	<ul style="list-style-type: none"> • RTeste - Roteiro de Teste • GTeste – Guia de Testes 	<ul style="list-style-type: none"> • LibreOffice / MS Office • Azure DevOps / Redmine
--	---	---	--	---

GQA

<ul style="list-style-type: none"> • Documento de Evidências e Métricas 	<ul style="list-style-type: none"> • Planejar o GQA • Executar a monitoria 	<ul style="list-style-type: none"> • Checklist para verificação da etapa 	<ul style="list-style-type: none"> • RGQA – Roteiro da Garantia da Qualidade 	<ul style="list-style-type: none"> • RPM e Redmine / Azure DevOps • LibreOffice / MS Office
--	--	---	---	---

Análise / Descoberta				
Produtos de trabalho	Atividades	Roteiros / Modelos	Principais documentos	Ferramentas
<ul style="list-style-type: none"> • Documento de Projeto (Integrações, procedimentos de operação, componentes de documentação) <ul style="list-style-type: none"> - Modelo E-R Lógico (Diagrama) - Especificação dos Requisitos Funcionais (Casos de Uso / Histórias de Usuário) - Protótipo da Interface - Protótipo da Interface de alta fidelidade e Teste de Usabilidade (Analista de UX na equipe) - Estimativa de desenvolvimento • Planilha de Produtos de Trabalho (Planilha ou anexo no Documento do projeto) • Documento LGPD (Informações de Privacidade) 	<ul style="list-style-type: none"> • Abertura do projeto de sistema • Planejamento do projeto de sistema • Análise do sistema • Projeto da arquitetura do sistema 	<ul style="list-style-type: none"> • RPRO - Roteiro de Projeto • MPRO - Modelo de Projeto • Planilha de Contagem APF • Planilha de Estimativa de Prazos • Modelos de Especificação de Casos de Uso • Modelo Teste de Usabilidade (UX) • Especificação e Detalhes Técnicos de História de Usuário • Modelo Documento LGPD (Informações de Privacidade) 	<ul style="list-style-type: none"> • Banco de Dados: - Padrão de Modelagem de Dados; • Interface: - Checklist de utilização do Padrão WEB; - A interface WEB Padrão PROCERGS • Relatórios: - Padrão de Especificação de Relatórios • UML: - Exemplo de Descrição de Casos de Uso; - Orientações para a especificação de RN; • Rose: - Guia do Rational Rose • Guia do Documento LGPD 	<ul style="list-style-type: none"> • RPM • Erwin • Rational Rose • Adobe XD • LibreOffice / MS Office • Redmine / Azure DevOps
GQA				
<ul style="list-style-type: none"> • Documento de Evidências e Métricas 	<ul style="list-style-type: none"> • Planejar o GQA • Executar a monitoria 	<ul style="list-style-type: none"> • Checklist para verificação da etapa 	<ul style="list-style-type: none"> • RGQA – Roteiro da Garantia da Qualidade 	<ul style="list-style-type: none"> • RPM e Redmine / Azure DevOps • LibreOffice / MS Office

Projeto / Descoberta				
Produtos de trabalho	Atividades	Roteiros / Modelos	Principais documentos	Ferramentas
<ul style="list-style-type: none"> • Documento de Projeto <ul style="list-style-type: none"> - Modelos de Classes (Opcional) - Refinamento dos Requisitos Funcionais (Implementação do Requisito: Regras de Interface e Projeto) - Especificação das classes de interface e 2ª e 3ª camadas (Opcional) - Modelo E-R Físico (Diagrama e scripts) • Planilha de Produtos de Trabalho (Planilha ou anexo no Documento do projeto) 	<ul style="list-style-type: none"> • Projeto da arquitetura do software • Projeto detalhado do software 	<ul style="list-style-type: none"> • RPRO - Roteiro de Projeto • MPRO - Modelo de Projeto • Aplicação Modelo (JAVA, .NET, Angular, Ionic, Kotlin) 	<ul style="list-style-type: none"> • Arquitetura: <ul style="list-style-type: none"> - Distribuição do código em camadas; - Documento sobre Arquitetura (conforme tecnologia: JAVA, PHP, .NET, Angular, Android/iOS) • Banco de Dados: <ul style="list-style-type: none"> - Nomenclatura para Objetos de BD; - Nomenclatura de Banco de Dados relacionais e de seus usuários 	<ul style="list-style-type: none"> • Erwin • Rational Rose • LibreOffice / MS Office • Banco de Dados • IDEs • Redmine / Azure DevOps
Testes				
<ul style="list-style-type: none"> • Plano Macro de Testes atualizado • Casos de Testes especificado • Automatização de Testes especificada 	<ul style="list-style-type: none"> • Planejamento e Especificação de Testes 	<ul style="list-style-type: none"> • Plano Macro de Testes • Checklist de Qualificação de Casos de Uso • Templates para testes de RN 	<ul style="list-style-type: none"> • RTeste - Roteiro de Teste • GTeste - Guia de Testes 	<ul style="list-style-type: none"> • Azure DevOps • Specflow / Test Manager (não BDD) • Cucumber

Construção / Execução				
Produtos de trabalho	Atividades	Roteiros / Modelos	Principais documentos	Ferramentas
<ul style="list-style-type: none"> •Objetos construídos •Componentes de documentação •Cadastro nos Sistemas de Controle de Acessos (SOE e LDAP) 	<ul style="list-style-type: none"> •Codificação de software •Elaboração dos componentes de comunicação •Integração de software •Testes de Sistema (Testes unitários e de integração) 	<ul style="list-style-type: none"> •RPRO - Roteiro de Projeto 	<ul style="list-style-type: none"> •Guia do Desenvolvedor WEB •Guia dos Componentes do SOE •Padrão de msg erro – Trigger •Padrão PROCERGS para construção conforme tecnologia •Padrões de Codificação 	<ul style="list-style-type: none"> •SOE •Doc-to-Help •LibreOffice / MS Office •IDEs •Redmine e SVN / Azure DevOps
Testes				
<ul style="list-style-type: none"> •Casos de Testes executados •Automatização de testes executada 	<ul style="list-style-type: none"> •Execução de Testes 	<ul style="list-style-type: none"> •Checklist de Qualificação de Casos de Uso •Templates para testes de RN 	<ul style="list-style-type: none"> •RTeste - Roteiro de Teste •GTeste – Guia de Testes 	<ul style="list-style-type: none"> •SOAPUI, Karate DSL, Postman •VS Enterprise, Coded UI •Selenium, Arquillian, Sonar, PMD, Jacoco, Jmeter, Protractor
GQA				
<ul style="list-style-type: none"> •Documento de Evidências e Métricas 	<ul style="list-style-type: none"> •Planejar o GQA •Executar a monitoria 	<ul style="list-style-type: none"> •Checklist para verificação da etapa 	<ul style="list-style-type: none"> •RGQA – Roteiro da Garantia da Qualidade 	<ul style="list-style-type: none"> •RPM e Redmine / Azure DevOps •LibreOffice / MS Office

Documentação				
Produtos de trabalho	Atividades	Roteiros / Modelos	Principais documentos	Ferramentas
<ul style="list-style-type: none"> •Manual do Usuário •Help On-line 	<ul style="list-style-type: none"> •Análise de UC, Protótipos, Sistema e Modelos •Geração da Documentação 	<ul style="list-style-type: none"> •RDUS - Roteiro de Documentação de Usuário •MDUS - Modelo de Documentação de Usuário 	<ul style="list-style-type: none"> •RED - Manual de Redação e Estilo PROCERGS •Dicas de Ortografia •Exemplo de Manual de Usuário 	<ul style="list-style-type: none"> •LibreOffice / MS Office •Doc-to-Help •Wink •Camtasia
GQA				
<ul style="list-style-type: none"> •Documento de Evidências e Métricas 	<ul style="list-style-type: none"> •Planejar o GQA •Executar a monitoria 	<ul style="list-style-type: none"> •Checklist para verificação da etapa 	<ul style="list-style-type: none"> •RGQA – Roteiro da Garantia da Qualidade 	<ul style="list-style-type: none"> •RPM e Redmine / Azure DevOps •LibreOffice / MS Office
Implantação / Entrega				
Produtos de trabalho	Atividades	Roteiros / Modelos	Principais documentos	Ferramentas
<ul style="list-style-type: none"> •Código dos objetos implantados (Homologação e Produção) •Software homologado •Documentação para a Operação concluída 	<ul style="list-style-type: none"> •Integração e Passagem de Produtos grados para o ambiente adequado 	<ul style="list-style-type: none"> •RPRO - Roteiro de Projeto 		<ul style="list-style-type: none"> •Conforme Tecnologia •LVP
Testes				
<ul style="list-style-type: none"> •Relatório com parecer da qualidade do projeto 	<ul style="list-style-type: none"> •Fechamento 		<ul style="list-style-type: none"> •RTeste - Roteiro de Teste •GTeste – Guia de Testes 	<ul style="list-style-type: none"> •Redmine / Azure DevOps
GQA				
<ul style="list-style-type: none"> •Planilhas de métricas por etapa 	<ul style="list-style-type: none"> •Publicar métricas 	<ul style="list-style-type: none"> •Planilhas de métricas por etapa •Planilha de métricas 		<ul style="list-style-type: none"> •RPM e Redmine / Azure DevOps •LibreOffice / MS Office

Processo de Desenvolvimento de Sistemas:

- **Gerência de Projeto, Etapa Preliminar, Processo de GQA e Documentação de Usuário não fazem parte dos 100% do Projeto, são estimados separadamente.**