

Projeto de Padronização do Processo de Desenvolvimento de Software

Modelo de Gestão e Desenvolvimento de Software

Histórico de Revisões

Data	Versão	Descrição	Autor
01/09/2011	1.0	Criação do Modelo de Gestão e Desenvolvimento – Introdução, Infraestrutura e descrição do modelo.	Rogério Poggio
08/09/2011	2.0	Criação do Modelo de Gestão e Desenvolvimento – Detalhamento das áreas de conhecimentos, processos e artefatos.	Rogério Poggio
23/09/2011	3.0	Finalização do Modelo de Gestão e Desenvolvimento.	Rogério Poggio
26/09/2011	3.0	Revisão do Modelo.	Rogério Poggio
29/09/2011	3.0	Revisão do Modelo.	Alex Nério
29/09/2011	3.0	Revisão do Modelo.	Juliana Rappel
20/11/2011	4.0	Revisão do Modelo e implantação de correções e melhorias	Rogério Poggio
20/12/2011	5.0	Inclusão do Fluxo de Interação da UDS com as Áreas de Negócios do MPBA	Rogério Poggio
15/10/2012	6.0	Revisão Geral do Modelo.	Rogério Poggio, Renato Cardoso e Juliana Rappel
06/06/2013	7.0	Revisão Geral do Modelo.	Rogério Poggio, Davi Nascimento
16/09/2013	7.1	Inclusão do Modelo de Domínio no Quadro de Artefatos da área de conhecimento “Projeto”	Rogério Poggio, Davi Nascimento
17/09/2013	7.2	Correção da responsabilidade pela produção do artefato Modelo de Domínio no Quadro de Artefatos da área de conhecimento “Projeto”.	Rogério Poggio, Davi Nascimento
20/08/2015	7.3	Inclusão de documento Mensagem do Sistema no Quadro de artefatos	Davi Nascimento
19/02/2018	7.4	Conversão; revisão	UDS



Índice Analítico

1 Introdução.....	4
2 Modelo de Gestão e Desenvolvimento de Software.....	5
2.1 Fases do Ciclo de Vida.....	8
2.2 Marcos do Ciclo de Vida.....	10
3 Classificação dos Artefatos e das Revisões.....	11
4 Áreas de conhecimentos.....	12
4.1 Área de conhecimento de Requisitos.....	12
4.2 Área de conhecimento de Projeto.....	16
4.3 Área de conhecimento de Implementação.....	19
4.4 Área de conhecimento de Testes.....	19
4.5 Área de conhecimento de Implantação.....	21
4.6 Área de conhecimento de Gestão de Configuração.....	23
4.7 Área de conhecimento de Gestão de Mudanças.....	25
4.8 Área de conhecimento de Gestão de Projetos.....	30
4.9 Área de conhecimento de Gestão de Qualidade.....	33
5 Conclusão.....	35



1 Introdução

O modelo de gestão e desenvolvimento de software do Ministério Público do Estado da Bahia (MPBA) é baseado nas boas práticas de engenharia de software e no Rational Unified Process (RUP), considerando as boas práticas de gestão de projetos do PMI, gerenciamento de serviços do ITIL e governança de TI do COBIT.

Este projeto proporciona a melhoria do processo de desenvolvimento de software, contribuindo para melhorar a criação e adaptação dos sistemas de informações do órgão. O modelo também contribui para a certificação dos processos de desenvolvimento de software, pois ao seguir as melhores práticas (citadas no parágrafo anterior) os seus processos obtêm maior conformidade com as práticas disponibilizadas pelo CMMI e MPS-BR.

Com a melhoria do processo de desenvolvimento, projetos que visem a criação e adaptação de sistemas, inclusive projetos relacionados com acessibilidade, conseguem atingir com mais facilidade o seu objetivo, em razão da melhor estruturação e visualização das etapas do projeto de software, além da melhor distribuição dos esforços necessários para alcançar um resultado positivo.

Outro ponto importante é que, com o processo de gestão e desenvolvimento baseado nas melhores práticas do mercado, problemas relacionados com levantamentos de requisitos, controle de mudanças e aceitação de produtos desenvolvidos poderão ser mitigados ou eliminados, contribuindo, ainda mais, para o cumprimento das metas estratégicas destinadas à Unidade de Desenvolvimento de Sistemas (UDS) do MPBA.

Desta forma, esta iniciativa, além de melhorar o processo de desenvolvimento de software do órgão, também contribui para a realização de metas estratégicas, relacionadas com o atual planejamento estratégico do MPBA. As metas são: (i) elaborar e implementar projeto para desenvolver e adaptar softwares no site do MPBA, a fim de permitir a utilização por pessoas com deficiência física e sensorial e (II) elaborar e implementar projeto visando garantir a certificação em processos de desenvolvimento de softwares – qualidade de software.

O restante deste documento está organizado da seguinte forma. A Seção 02 apresenta o processo de desenvolvimento com o ciclo de vida adotado. A Seção 03 descreve o padrão de classificação e revisão dos artefatos que serão produzidos. A Seção 04 apresenta as áreas de conhecimentos, com seus processos, atividades e artefatos produzidos. Finalmente a Seção 05 descreve as conclusões e considerações finais deste projeto.




2 Modelo de Gestão e Desenvolvimento de Software

O modelo de desenvolvimento utiliza o **ciclo de vida iterativo com o modelo de desenvolvimento incremental** (que combina elementos do modelo sequencial linear – modelo em cascata – e da filosofia iterativa da prototipagem) e a abordagem evolutiva, objetivando a cada incremento a elaboração de um produto operacional executável.

Este modelo oferece uma abordagem baseada em áreas de conhecimentos para atribuir tarefas e responsabilidades dentro da Unidade de Desenvolvimento de Sistemas do MPBA. Cada área de conhecimento possui um processo associado que, por sua vez, possui **papéis, atividades e artefatos** que têm a finalidade de documentar o processo, fazendo com que as informações necessárias fiquem devidamente registradas para análises futuras ou eventuais necessidades do órgão. O modelo está estruturado em fases que por sua vez possuem iterações que estão associadas a releases distintos. Uma iteração consiste em um pequeno **projeto de curta duração** que percorre as áreas de conhecimentos, possibilitando a continuação do projeto de maneira mais flexível e menos arriscada.

Uma iteração, geralmente, está relacionada a um release externo que pode se desdobrar em releases internos menores. Um release é o termo utilizado para designar a liberação de uma versão operacional estável e executável do sistema, ou parte dele, com todos os artefatos necessários para o seu uso, manutenção e novos desenvolvimentos. Um release poderá ser construído e liberado para o usuário final (release externo) ou para a unidade de desenvolvimento (release interno). Na Figura 2.1 podem ser visualizados todos os elementos discutidos e utilizados para construir o modelo de gestão e desenvolvimento de software.

 MINISTÉRIO PÚBLICO DO ESTADO DA BAHIA	Projeto Projeto de Padronização do Processo de Desenvolvimento de Software		
Documento Modelo de Gestão e Desenvolvimento de Software	Área Emitente DTI / UDS	Versão 7.4	Data de Emissão 19/02/2018

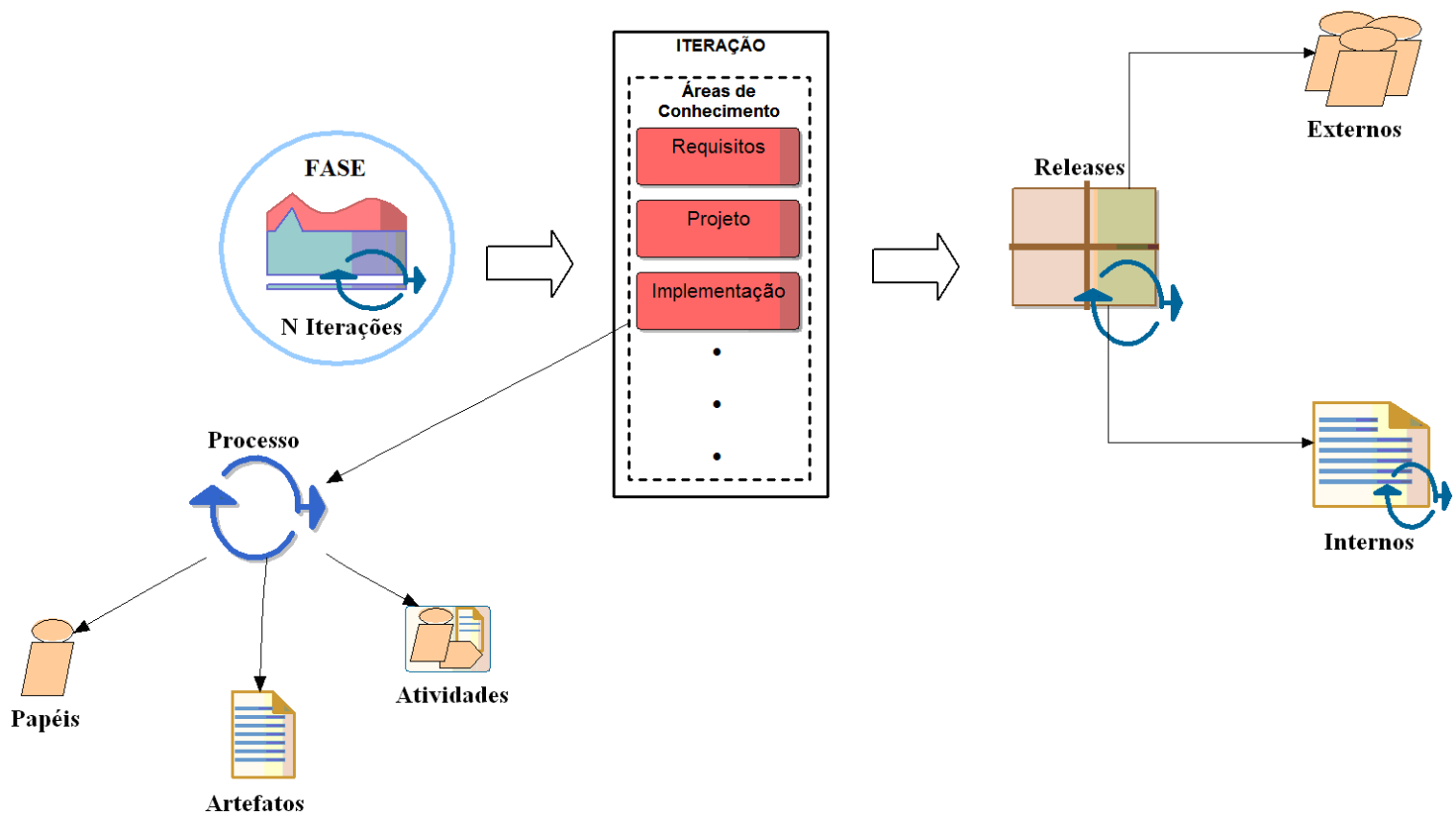


Figura 2.1: Elementos da abordagem utilizada para construção do modelo.

O modelo possui 04 fases sequenciais, mas que em determinados momentos **podem ocorrer em paralelo**, conforme mostra a Figura 2.2. São elas: (i) a fase de **concepção**; (II) a fase de **elaboração**; (III) a fase de **construção** e (IV) a fase de **transição**. Cada fase pode possuir uma ou mais iterações. As áreas de conhecimentos específicas, relacionadas diretamente com o desenvolvimento, são: (i) requisitos; (II) projeto; (III) implementação; (IV) testes e (v) implantação. Também serão abordadas mais três áreas de conhecimentos de escopo gerencial. São elas: (i) gestão de configuração; (II) gestão de mudanças e (III) gestão de projetos. Adicionalmente, foi criada a área de conhecimento de gestão de qualidade, que fica responsável pela criação e manutenção do processo de desenvolvimento de software (modelos de artefatos, diretrizes de desenvolvimento, dentre outros). Essas áreas de conhecimentos, apesar de produzir diversos artefatos, não possuem processos formais relacionados a elas.

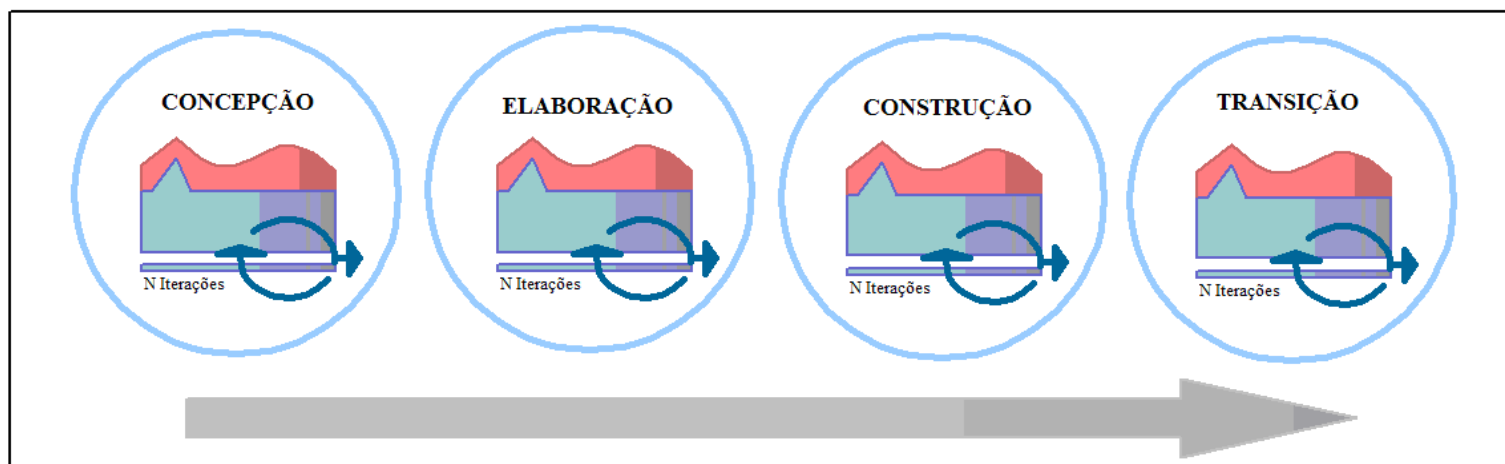


Figura 2.2: Fases do Processo de Gestão e Desenvolvimento de Software.

Uma visão do processo de gestão e desenvolvimento de software pode ser visualizada na Figura 2.3. Nela é possível observar todos os elementos que compõe o modelo.



MODELO DE GESTÃO E DESENVOLVIMENTO DE SOFTWARE

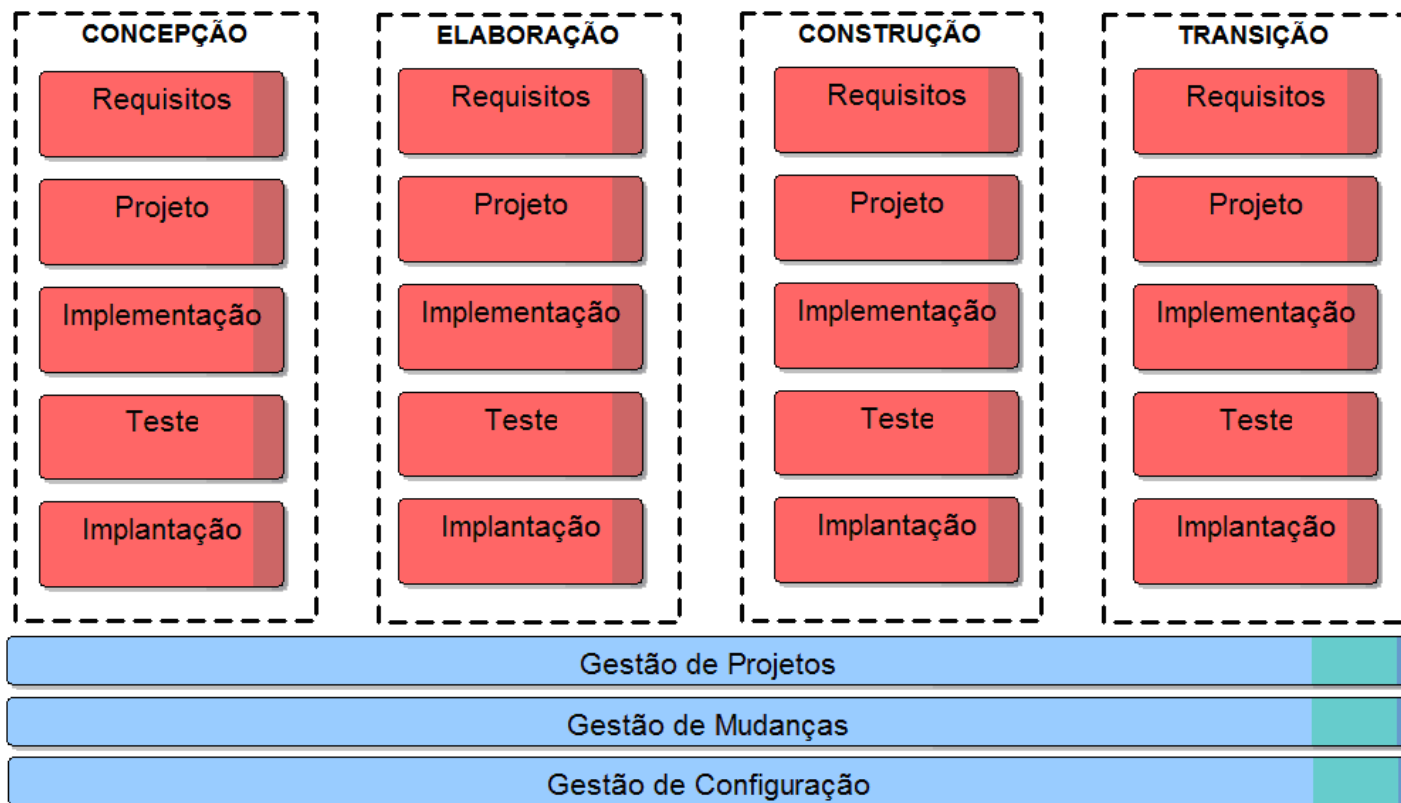


Figura 2.3: Visão do Processo de Desenvolvimento de Software.

2.1 Fases do Ciclo de Vida

Esta subseção discute os objetivos das fases que compõem o ciclo de vida do processo de gestão e desenvolvimento de software. O descritivo de cada uma das fases, com as suas principais metas são mostradas a seguir.

2.1.1 Concepção

Esta fase foca no **escopo do projeto**. A sua principal preocupação é estabelecer uma visão e uma compreensão (ainda que sejam vagas) das exigências gerais de prazo, escopo e riscos do projeto. Os objetivos da fase de concepção incluem:

- Estabelecer o escopo do software e as suas premissas e restrições, incluindo os critérios de aceitação e o que deve, ou não, estar no produto;
- Estimar o prazo geral para o projeto (e estimativas detalhadas para a fase de elaboração imediatamente a seguir);
- Estimar riscos em potencial;
- Preparar o ambiente para o projeto.



2.1.2 Elaboração

Esta fase visa primariamente: (i) analisar melhor as exigências; (II) capturar os requisitos ainda não identificados; (III) **detalhar os requisitos**; (IV) definir e/ou atualizar a arquitetura do sistema e (v) mitigar os riscos. Dessa forma, consegue-se criar uma base estável para o esforço da fase de construção. Os objetivos da fase de elaboração incluem:

- Assegurar que a arquitetura, os requisitos e os planos estão estáveis e que os riscos estão minimizados a fim de determinar, com segurança, o custo e o prazo para a conclusão do desenvolvimento;
- Tratar todos os riscos significativos, do ponto de vista da arquitetura do sistema;
- Produzir um ou mais protótipos, para diminuir riscos específicos relacionados com a falta de entendimento de requisitos, projeto, reutilização ou até mesmo para possibilitar a produção de demonstrações para os usuários finais.

2.1.3 Construção

A meta da fase de construção é esclarecer os requisitos restantes e concluir o desenvolvimento do sistema. A fase de construção é, de certa forma, um **processo de manufatura**, em que a ênfase está no gerenciamento de recursos e no controle de operações para otimizar custos, prazos e qualidade. Os objetivos da fase de construção incluem:

- Minimizar os custos de desenvolvimento, otimizando recursos e evitando retrabalhos desnecessários;
- Atingir as versões úteis, com qualidade adequada, rapidez e eficiência;
- Desenvolver, de modo iterativo e incremental, um produto completo que esteja pronto para ser disponibilizado à comunidade de usuários;
- Decidir se o software, locais e usuários estão prontos para que o aplicativo seja implantado e
- Atingir, caso seja possível, um paralelismo entre o trabalho das equipes de desenvolvimento, pois normalmente existem componentes que podem ser desenvolvidos de forma independente do outro, permitindo o paralelismo natural entre as equipes.

2.1.4 Transição

Esta fase objetiva **liberar e implantar o software**. O foco da fase de transição é assegurar que o software esteja disponível para seus usuários finais. A fase de transição pode atravessar várias iterações e inclui testar o produto e realizar pequenos ajustes com base no retorno dos usuários. Neste momento do ciclo de vida, o retorno do usuário deve priorizar o



ajuste fino do produto, a configuração, a instalação e os problemas de usabilidade e/ou acessibilidade.

É importante que todos os problemas estruturais mais graves tenham sido trabalhados muito antes no ciclo de vida do projeto. No fim da fase de transição, os objetivos do projeto devem ter sido atendidos e o projeto deve estar em uma posição para fechamento. Em alguns casos, o fim da fase de transição pode coincidir com o início de outro ciclo de vida do mesmo produto, conduzindo à nova geração ou versão do produto. Para outros projetos, o fim da transição pode coincidir com uma liberação total dos artefatos a terceiros que poderão ser responsáveis pela operação, manutenção e melhorias no sistema liberado.

A fase de transição se inicia quando o projeto ou parte dele estiver suficientemente desenvolvido para ser implantado no domínio do usuário final. Isso, normalmente, requer que algum subconjunto usável do sistema tenha sido concluído com um nível de qualidade aceitável, de modo que a transição para o usuário forneça resultados positivos para todas as partes. Os objetivos da fase de transição são:

- Testar o novo sistema para validá-lo em confronto com as expectativas do usuário;
- Migrar bancos de dados operacionais;
- Treinar usuários e equipe de manutenção;
- Realizar atividades de ajuste, como correções de erros, melhorias no desempenho e na usabilidade/acessibilidade;
- Finalizar o material de suporte para o usuário final;
- Criar um release do produto;
- Disponibilizar o produto para os usuários finais.

2.2 Marcos do Ciclo de Vida

Cada fase ou iteração deste modelo de desenvolvimento possui um marco que permite ao seu gestor tomar decisões. Estes marcos permitem avaliar o resultado da iteração ou da fase em relação aos critérios de êxito estabelecidos como objetivos para a iteração ou fase específica. Os marcos que se relacionam com as fases são chamados de marcos de fase. Já os marcos que se relacionam com as iterações são chamados de marcos internos. Os marcos das fases presentes neste modelo são:

2.2.1 Marco de Concepção

O marco da fase de concepção envolve possuir o **documento de visão devidamente aprovado pelo gestor da área de negócio**, além de já ter produzido a lista de riscos do projeto



e disponibilizado os artefatos de referência (guias e diretrizes) que serão utilizados durante o desenvolvimento do projeto. Isto pode envolver:

- Haver um consenso dos envolvidos sobre a definição do escopo e as estimativas de custo, prazo e riscos;
- Haver um consenso de que o conjunto correto de requisitos foi capturado e existe uma compreensão compartilhada desses requisitos;
- Haver um consenso de que as prioridades são adequadas;
- Identificar todos os riscos e criar uma estratégia atenuante para cada risco encontrado.

2.2.2 Marco de Elaboração

O marco da fase de elaboração envolve **possuir o arcabouço de requisitos** (modelo de casos de uso, casos de uso, especificação suplementar, documento de regras de negócios, documento de mensagens do sistema) **aprovados pelo gestor da área de negócios**, com os artefatos obrigatórios para a área de conhecimento de projeto, o protótipo da interface do usuário e o glossário de negócio devidamente produzidos.

2.2.3 Marco de Construção

O marco da fase de construção envolve a construção do **release que deve estar estável para ser implantado na comunidade de usuários**. Isto envolve o código do sistema e scripts de banco necessários ao release.

2.2.4 Marco de Transição

O marco da fase de transição envolve a conclusão de todos os testes formais e inspeções de software, com a **disponibilização do release e seus artefatos à comunidade de usuários**. Todos os artefatos da área de conhecimento de teste e implantação devem estar produzidos e o gestor da área de negócios deve ter aceitado formalmente (documento de aceite) o release que será implantado.

3 Classificação dos Artefatos e das Revisões

Esta seção descreve a estrutura de classificação dos artefatos quanto a sua necessidade e quanto à forma como as revisões serão realizadas nos documentos produzidos ao longo do processo de desenvolvimento.

O principal benefício de adotar este sistema de classificação é que ele permite explicitar a importância e o nível de revisão dos artefatos especificados como saída para as áreas de conhecimentos do modelo.



Outro ponto importante é que, através desta classificação, se torna possível identificar onde existem opções para negociação e tomada de decisão. As tabelas 3.1 e 3.2 mostram a estrutura de classificação, quanto à necessidade dos artefatos e quanto à forma como acontece as suas revisões.

Classificação dos Artefatos	
Classificação	Explicação
Obrigatório (O)	Artefatos essenciais ao desenvolvimento. Devem ser produzidos.
Recomendável (R)	Artefato pode ser feito, mas não é essencial. A produção do artefato pode ajudar ao melhor entendimento do projeto.

Tabela 3.1: Classificação dos artefatos.

Classificação das Revisões	
Nível de Revisão	Explicação
Formal-Externo	O artefato requer alguma forma de aprovação formal do usuário final, cliente, patrocinador ou por algum outro envolvido externo.
Formal-Interno	O artefato é aprovado formalmente, pela equipe interna do projeto.
Informal	O artefato é revisto, mas não é aprovado formalmente.
Nenhum	O artefato não precisa ser revisado ou aprovado.

Tabela 3.2: Classificação das revisões.

4 Áreas de conhecimentos

Esta seção detalha como funciona cada área de conhecimento do modelo de gestão e desenvolvimento de software do MPBA.

4.1 Área de conhecimento de Requisitos

A área de conhecimento de requisitos procura estabelecer e manter a concordância com os clientes e outros envolvidos no projeto sobre **o que o sistema deve realizar**. Ela oferece aos desenvolvedores do sistema uma compreensão melhor dos requisitos, além de definir as fronteiras do sistema, fornecendo uma base para planejar o conteúdo técnico das iterações. A área de conhecimento de requisitos também fornece uma base para estimar o custo e o tempo de desenvolvimento do sistema, além de contribuir para a definição da interface do usuário, focando nas **necessidades e metas dos usuários**. Para isso, é necessário compreender a definição e o escopo do problema, que o sistema deve resolver, além de identificar todos os envolvidos. Uma forma de alcançar isso é através do registro e análise dos formulários de levantamento de informações no projeto.



Os formulários de levantamento de informações devem ser recolhidos para se obter uma listagem do que as diversas pessoas envolvidas no projeto esperam e desejam que o sistema contenha, com informações de como cada solicitação foi considerada pelo projeto.

Um modelo de casos de uso, as especificações dos casos de uso e suplementar, além dos demais documentos da metodologia que complementam os casos de uso, também devem ser desenvolvidos para descrever integralmente o que o sistema fará – num trabalho em que todos os envolvidos são vistos como fontes de informações.

O modelo de casos de uso deve servir como um meio de comunicação com os envolvidos no projeto e pode servir como um contrato entre o cliente, os usuários e os desenvolvedores do sistema sobre a funcionalidade do sistema, permitindo que: (i) clientes e usuários validem que o sistema ficará como eles esperam e (II) desenvolvedores do sistema construam o que é esperado.

Cada item do modelo de casos de uso deve ser descrito detalhadamente, mostrando passo a passo como o sistema interage com os atores, e o que o sistema faz no caso de uso. Os casos de uso funcionam como uma base, que no decorrer do ciclo de vida do processo de desenvolvimento, será utilizada nas atividades de projeto, implementação e teste do sistema.

As especificações suplementares são um complemento importante para os casos de uso, porque juntos eles capturam todos os requisitos (funcionais e não funcionais) que precisam ser descritos, servindo como uma especificação de requisitos de software completa.

O documento de visão e uma matriz de rastreabilidade de requisitos também ajudam a gerenciar os requisitos do projeto. Um glossário do projeto deve ser desenvolvido para definir de forma consistente a terminologia utilizada no projeto.

Um protótipo da interface do usuário pode ser utilizado para descobrir ou elucidar dúvidas relacionadas com os requisitos. Estes protótipos podem ser descartáveis ou podem posteriormente ser aproveitados.

4.1.1 Processo e Atividades Envolvidas

Esta seção descreve as atividades relacionadas com o processo da área de conhecimento de Requisitos. A Figura 4.1 mostra o processo e as atividades relacionadas com a área de conhecimento de Requisitos.

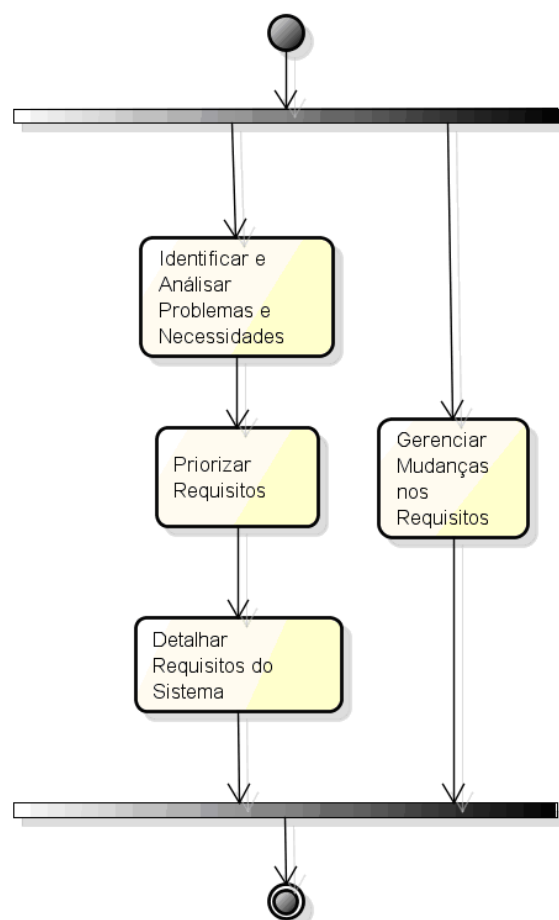


Figura 4.1: Processo da área de conhecimento de Requisitos.

4.1.1.1 Identificar e Analisar Problemas e Necessidades

Esta atividade visa: (i) identificar os envolvidos; (II) identificar o problema a ser resolvido e compreender as necessidades dos envolvidos; (III) definir as fronteiras do sistema e (IV) identificar as restrições impostas ao sistema.

Para realizar esta atividade, deve-se verificar se todas as partes envolvidas estão de acordo com a definição do problema, que será solucionado com a criação ou alteração do sistema. Para evitar ambiguidades, é importante possuir uma terminologia comum que será usada no decorrer do projeto. Portanto, desde o início, deve-se começar a definir os termos do projeto em um glossário que será mantido durante todo o ciclo de vida do projeto. A fim de compreender completamente o(s) problema(s), é muito importante conhecermos os envolvidos e observar que alguns desses envolvidos – os usuários do sistema – serão representados por atores no modelo de casos de usos.

Os formulários de levantamento de informações podem ser usados como forma de identificar as características do sistema que estão presentes na plano de projeto. Para definir o escopo inicial do projeto, é necessário haver um acordo sobre as fronteiras do sistema.



Neste ponto, também é relevante identificar os usuários e os sistemas que interagirão com a solução que será desenvolvida.

Durante esta atividade, já é possível iniciar a discussão sobre os requisitos funcionais do sistema em termos de casos de uso e atores. Os requisitos não funcionais que não se encaixam facilmente no modelo de casos de uso devem ser documentados nas especificações suplementares. Nesta atividade também deve ser feita uma atualização do glossário para facilitar o vocabulário comum entre os membros da equipe.

Esta atividade pode ser revista muitas vezes durante as fases de concepção e elaboração. Após isso, ao longo do ciclo de vida do projeto, esta atividade deve ser revista somente quando necessário, como por exemplo, ao gerenciar mudanças, a fim de assegurar que os problemas corretos não estão sendo desconsiderados.

4.1.1.2 Priorizar Requisitos

Esta atividade visa estabelecer a ordem de implementação dos requisitos do sistema. O gerenciamento do escopo é essencial para fazer com que a implementação destes requisitos se adequeiem aos recursos disponíveis (tempo e pessoas).

Portanto, esta é uma atividade contínua e que requer o desenvolvimento iterativo e incremental, o qual divide o escopo em partes menores e mais simples de gerenciar. A definição dos atributos de requisitos pode ser usada pelo coordenador do projeto para planejar as iterações, permitindo a identificação dos casos de uso mais significativos do sistema.

4.1.1.3 Detalhar Requisitos do Sistema

Esta atividade objetiva descrever os casos de uso do projeto e seus documentos correlatos, além de detalhar as especificações suplementares e desenvolver um protótipo da interface do usuário. A saída dessa atividade é uma compreensão mais aprofundada do sistema expressa em casos de uso e especificações suplementares detalhados, bem como elementos da interface do usuário. É importante trabalhar junto com os usuários ao realizar a modelagem de casos de uso e o protótipo da interface de usuário. Isso possibilita: (i) melhorar o entendimento sobre a usabilidade/acessibilidade do sistema; (II) descobrir qualquer requisito que ainda não tenha sido descoberto e (III) contribuir para o refinamento da definição dos requisitos.

4.1.1.4 Gerenciar Mudanças nos Requisitos

Esta atividade objetiva avaliar solicitações de mudanças relacionadas com os requisitos, formalmente submetidas ao processo de gerenciamento de mudanças, definindo o seu impacto no conjunto de requisitos existente. Esta atividade também procura adequar o



modelo de casos de uso, com os atributos de requisitos e a matriz de rastreabilidade, à nova realidade do projeto.

Os relacionamentos de rastreabilidade identificados fornecem as ligações existentes entre requisitos e outros artefatos. Esses relacionamentos são a chave para a compreensão do impacto da mudança nos requisitos. Análises periódicas, com atualizações dos atributos e dependências, devem ser feitas sempre que as especificações de requisitos forem atualizadas. A Tabela 4.1 mostra os artefatos produzidos nesta área de conhecimento e em que momento (fase) os mesmos se fazem necessários.

Artefatos da área de conhecimento de Requisitos						
Artefatos	Fase de Utilização dos Artefatos				Detalhes da Revisão	Responsável
	Concepção	Elaboração	Construção	Transição		
Documento de Visão	O	O	R	R	Formal-Externo	Analista de Sistemas
Formulário de Levantamento de Informações	R	R	R	R	Formal-Externo	Analista de Sistemas
Planilha de rastreabilidade de requisitos	R	R	R	R	Formal-Interno	Analista de Sistemas
Modelo de Caso de Uso	O	O	R	R	Formal-Externo	Analista de Sistemas
Especificação de Caso de Uso	R	O	O	O	Formal-Externo	Analista de Sistemas
Especificação Suplementar	O	O	O	O	Formal-Externo	Analista de Sistemas
Mensagens do Sistema	R	O	O	O	Formal-Interno	Analista de Sistemas
Definição e Regras de Negócio	R	O	O	O	Formal-Externo	Analista de Sistemas
Glossário de Negócio	O	O	O	O	Formal-Externo	Analista de Sistemas
Protótipo da Interface do Usuário	R	O	O	O	Formal-Externo	Web Design/Analista de Sistemas

Tabela 4.1: Artefatos da área de conhecimento de Requisitos.

4.2 Área de conhecimento de Projeto

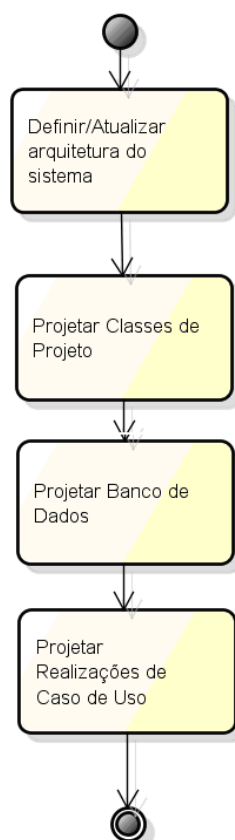
A área de conhecimento de Projeto visa primariamente **transformar os requisitos no projeto do sistema**. Esta área de conhecimento ajuda a desenvolver uma arquitetura sofisticada, além de adaptar o projeto para que ele corresponda às necessidades dos usuários e ao ambiente de implementação.



Na fase de Concepção, a função desta etapa será determinar se o sistema é viável, conforme o previsto, e avaliar as tecnologias possíveis para a solução. Na fase de Elaboração, a área de conhecimento enfatiza a criação de uma arquitetura para o sistema. Se a arquitetura já existir (porque foi produzida nos projetos anteriores, ou porque foi obtida de um framework), o enfoque do trabalho passará ao refinamento da arquitetura. Caso a arquitetura não exista deverá ser feita a análise do comportamento e a criação de um conjunto de elementos que fornecerão o comportamento apropriado para atender aos requisitos do sistema. Junto com essas atividades, as questões de persistência também deverão ser abordadas.

4.2.1 Processo e Atividades Envolvidas

Esta seção descreve as atividades relacionadas com o processo da área de conhecimento de Projeto. A Figura 4.2 mostra o processo e as atividades relacionadas com a área de conhecimento de Projeto.



*Figura 4.2:
Processo da
área de
conhecimento
de Projeto.*

4.2.1.1 Definir/Atualizar Arquitetura do Sistema

Esta atividade visa definir ou atualizar a arquitetura do sistema que será construído.

4.2.1.2 Projetar Classes de Projeto

A finalidade desta atividade é identificar e definir o modelo de classes de projeto, a partir dos casos de uso, iniciando pelos mais significativos para o sistema.

4.2.1.3 Projetar Banco de Dados

A finalidade desta atividade é: (i) construir o modelo de dados a partir do modelo de classes de projeto e (II) definir mecanismos e estratégias de armazenamento e recuperação de dados persistentes, de modo a atender aos critérios de desempenho do sistema. Nesta atividade, também é criado o dicionário de dados relacionado com o modelo de dados construído.

4.2.1.4 Projetar Realizações de Caso de Uso

Esta atividade visa definir as realizações de casos de uso que, por sua vez, focam no mapeamento de objetos (fronteira, classes e tabelas) produzidos para cada caso de uso do projeto de sistema. A Tabela 4.2 mostra os artefatos produzidos nestas áreas de conhecimento e em que momento (fase) os mesmos se fazem necessários.

Artefatos da área de conhecimento de Projeto						
Artefatos	Fase de Utilização dos Artefatos				Detalhes da Revisão	Responsável
	Concepção	Elaboração	Construção	Transição		
Modelo de Dados	R	O	O	O	Formal-Interno	Analista de Sistemas
Dicionário de Dados	R	O	O	O	Formal-Interno	Analista de Sistemas / Coordenador de AD
Modelo de Classes de Projeto	R	O	O	O	Formal-Interno	Analista de Sistemas
Diagrama de Sequência	R	O	O	R	Formal-Interno	Analista de Sistemas
Diagrama de Gráfico de Estado	R	R	R	R	Formal-Interno	Analista de Sistemas
Diagrama de Atividades	R	R	R	R	Formal-Interno	Analista de Sistemas
Realização dos Casos de Uso	R	O	O	O	Formal-Interno	Analista de Sistemas

Tabela 4.2: Artefatos da área de conhecimento de Projeto.



4.3 Área de conhecimento de Implementação

A finalidade da área de conhecimento de implementação é definir a organização do código em camadas (subsistemas de implementação) e implementar classes e objetos em termos de componentes (arquivos-fonte e outros), produzindo **uma solução executável**.

4.3.1 Processo e Atividades Envolvidas

Esta seção descreve as atividades relacionadas com o processo da área de conhecimento de Implementação. A Figura 4.3 mostra o processo e as atividades relacionadas com a área de conhecimento de Implementação.

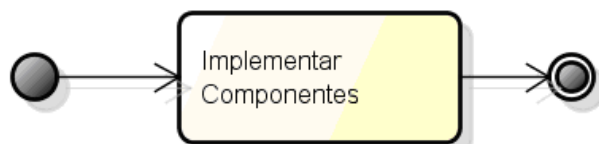


Figura 4.3: Processo da área de conhecimento de Implementação.

4.3.1.1 Implementar componentes

O objetivo desta atividade é realizar a codificação, adaptação e compilação dos componentes, conforme eles implementam as classes de projeto.

A Tabela 4.3 mostra os artefatos produzidos nesta área de conhecimento e em que momento (fase) os mesmos se fazem necessários.

Artefatos da área de conhecimento de Implementação						
Artefatos	Fase de Utilização dos Artefatos				Detalhes da Revisão	Responsável
	Concepção	Elaboração	Construção	Transição		
Codificação do projeto	R	R	O	R	Formal-Interno	Analista de Sistemas
Relatório de Cobertura de Testes	-	-	R	R	Formal-Interno	Analista de Sistemas
Relatório de Segurança de Código	-	-	R	R	Formal-Interno	Analista de Sistemas

Tabela 4.3: Artefatos da área de conhecimento de Implementação.

4.4 Área de conhecimento de Testes

A área de conhecimento de Testes visa primariamente avaliar a **qualidade do produto** através de várias práticas, como (i) localizar e documentar defeitos no software; (II) validar as suposições feitas nas especificações de requisitos e projeto; (III) validar as funções do software conforme projetadas e (IV) verificar se os requisitos foram implementados de forma adequada.



Uma diferença interessante entre a área de conhecimento de teste e as outras deste modelo de desenvolvimento é que a principal finalidade do teste é localizar e expor os pontos fracos do software. O teste desafia as suposições, os riscos e as incertezas inerentes ao trabalho de outras áreas de conhecimentos, tratando essas questões por meio de uma demonstração concreta (execução do sistema ou parte dele) e uma avaliação imparcial.

O desafio é evitar dois extremos potenciais: (i) uma abordagem que não avalie o software de forma adequada e efetiva, deixando de expor seus problemas e pontos fracos e (II) uma abordagem que seja totalmente negativa ou destrutiva, sendo impossível considerar aceitável a qualidade do software.

4.4.1 Processo e Atividades Envolvidas

Esta seção descreve as atividades relacionadas com o processo da área de conhecimento de Teste. A Figura 4.4 mostra o processo e as atividades relacionadas com a área de conhecimento de teste.

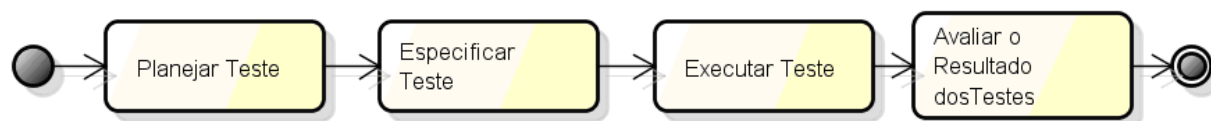


Figura 4.4: Processo da área de conhecimento de Teste.

4.4.1.1 Planejar Teste

Esta atividade visa definir e discriminar os testes que serão executados através de um plano de testes.

4.4.1.2 Especificar Teste

Esta atividade objetiva desenvolver um modelo de teste. O elemento a ser testado deve ser analisado para que esse modelo seja criado. Nesta atividade os casos de uso são transformados em casos de teste. Os passos para esta transformação são: (i) para cada caso de uso, deve-se gerar uma lista de todos os cenários possíveis; (II) para cada cenário, deve-se identificar pelo menos um caso de teste e as condições válidas para execução e (III) para cada caso de teste, deve-se identificar os valores de entrada, para o teste, e o resultado esperado.

4.4.1.3 Executar testes

O objetivo desta atividade é executar os testes especificados na etapa anterior, visando garantir que as funcionalidades do sistema tenham sido executadas conforme foram planejadas. Como o sistema é desenvolvido em incrementos e a cada incremento funcionalidades novas são adicionadas, é importante testar as funcionalidades novas e as mais antigas (que já foram testadas) quanto ao seu funcionamento e desempenho.



4.4.1.4 Avaliar o Resultado dos testes

O objetivo desta atividade é avaliar a qualidade do sistema através dos resultados obtidos pela execução dos testes especificados na etapa anterior. Esta análise sinalizará se o sistema está fornecendo as funcionalidades pretendidas, conforme foi especificado.

A Tabela 4.4 mostra os artefatos produzidos nesta área de conhecimento e em que momento os mesmos se fazem necessários.

Artefatos da área de conhecimento de Testes						
Artefatos	Fase de Utilização dos Artefatos				Detalhes da Revisão	Responsável
	Concepção	Elaboração	Construção	Transição		
Plano de teste	-	R	O	O	Formal-Interno	Analista de Testes
Caso de Teste	-	R	O	O	Formal-Interno	Analista de Testes
Relatório de Avaliação de Testes	-	-	-	O	Formal-Interno	Analista de Testes

Tabela 4.4: Artefatos da área de conhecimento de Testes.

4.5 Área de conhecimento de Implantação

A área de conhecimento de Implantação descreve as atividades que garantem que o **software será disponibilizado a seus usuários finais**. Esta área de conhecimento representa o ápice do esforço do desenvolvimento de software.

O planejamento da implantação pode começar na fase de elaboração e envolve não só colocar o software em produção, mas também o desenvolvimento de material de treinamento e de suporte para garantir que o usuário final possa usar corretamente o software entregue. O material de suporte inclui todo tipo de informações necessárias para que o usuário final instale, opere, use e mantenha o sistema disponibilizado. Esta área de conhecimento também inclui o material de treinamento para que os diversos perfis de usuários consigam utilizar corretamente o novo sistema.

4.5.1 Processo e Atividades Envolvidas

Esta seção descreve as atividades relacionadas com o processo da área de conhecimento de Implantação. A Figura 4.5 mostra o processo e as atividades dessa área de conhecimento.

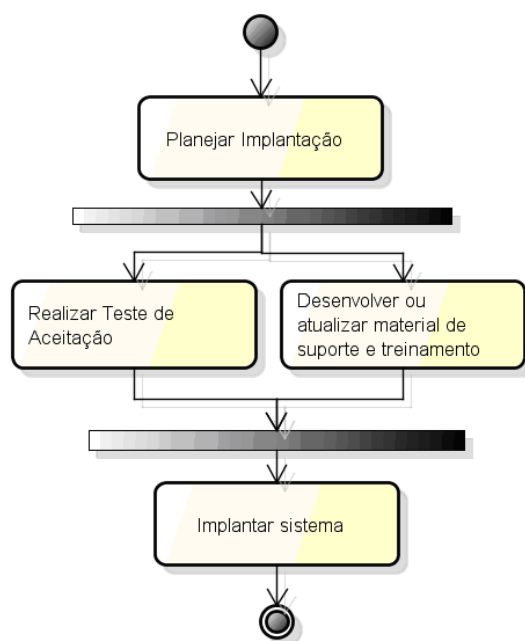


Figura 4.5: Processo da área de conhecimento de Implantação.

4.5.1.1 Planejar Implantação

Esta atividade visa criar o plano de implantação, definindo a lista de elementos que serão entregues, além de garantir que os usuários estejam cientes e comprometidos com as atividades de implantação.

4.5.1.2 Desenvolver ou atualizar materiais de suporte e treinamento

Esta atividade visa desenvolver ou atualizar os artefatos que serão utilizados pelos usuários no momento da instalação, operação, uso e manutenção do sistema. Esses materiais incluem a documentação do sistema, manuais do usuário, materiais de treinamento, dentre outros que se tornem necessários.

4.5.1.3 Realizar Teste de Aceitação

Esta atividade procura disponibilizar o release no ambiente de homologação para proporcionar os testes de aceitação com os usuários finais e permite a verificação, por parte do usuário, de que o release fornece as funcionalidades que foram encomendadas.

4.5.1.4 Implantar Sistema

Esta atividade visa implantar o sistema no ambiente de produção.

A Tabela 4.5 mostra os artefatos produzidos nesta área de conhecimento e em que momento os mesmos se fazem necessários.



Artefatos da área de conhecimento de Implantação

Artefatos	Fase de Utilização dos Artefatos				Detalhes da Revisão	Responsável
	Concepção	Elaboração	Construção	Transição		
Plano de Implantação	-	R	O	O	Formal-Interno	Analista de Sistemas
Notas de Release	-	R	R	O	Formal-Interno	Analista de Sistemas
Material de Suporte e Treinamento	-	R	R	R	Formal-Interno	Analista de Sistemas

Tabela 4.5: Artefatos da área de conhecimento de Implantação.

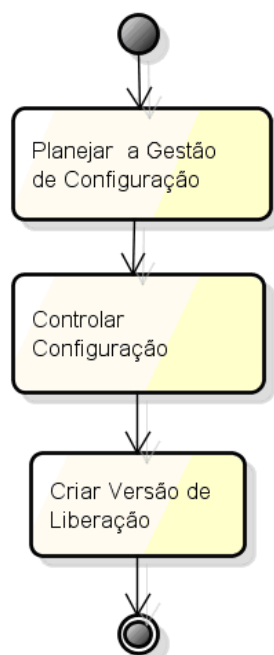
4.6 Área de conhecimento de Gestão de Configuração

A Gestão de Configuração visa **documentar, armazenar, controlar as modificações e verificar a integridade dos artefatos** ao longo do projeto. A correta gestão de configuração busca evitar problemas de perdas de versões, falta de controle de alterações e do estado do projeto, retrabalho e perda de produtividade.

4.6.1 Processo e Atividades Envolvidas

Esta seção descreve as atividades relacionadas com o processo da área de conhecimento de Gestão de Configuração. A Figura 4.6 mostra o processo e as atividades relacionadas com a área de conhecimento de Gestão de Configuração.





*Figura 4.6:
Processo da área
de conhecimento
de Gestão de
Configuração.*

4.6.1.1 Planejar a gestão de configuração

Os objetivos desta atividade são (i) identificar os itens de configuração que serão produzidos durante o processo de desenvolvimento e que deverão ser devidamente gerenciados; (II) definir o analista de configuração e (III) criar o repositório e a estrutura de pastas conforme o guia de configuração do projeto.

Esta atividade ainda objetiva gerenciar a baseline que é composta pelas versões dos artefatos que compõem o produto em dado momento. Normalmente, o baseline pode ser criado: (i) no final de cada passagem pelas áreas de conhecimento (dentro da iteração); (II) no final de cada iteração; (III) nos marcos das fases do projeto e/ou (IV) sempre que o produto for disponibilizado para o cliente.

4.6.1.2 Controlar configuração

Essa atividade objetiva garantir o controle adequado da evolução dos itens de configuração ao longo do projeto. Para isso deve-se: (i) verificar se a configuração do projeto está adequada com o que foi previamente estabelecido; (II) verificar se os backups periódicos estão sendo realizados pela unidade de infraestrutura tecnológica e (III) seguir as orientações técnicas do guia de configuração de projeto. Trata-se de uma atividade que será executada continuamente durante quase todo o ciclo de vida do release.



4.6.1.3 Criar versão de liberação

Esta atividade cria o pacote contendo a versão de software que será implantado e garante que este pacote contenha todos os artefatos necessários à implantação.

A Tabela 4.6 mostra os artefatos produzidos nesta área de conhecimento e em que momento (fase) os mesmos se fazem necessários.

Artefatos da área de conhecimento de Gestão de Configuração						
Artefatos	Fase de Utilização dos Artefatos				Detalhes da Revisão	Responsável
	Concepção	Elaboração	Construção	Transição		
Repositório do Projeto	O	O	O	O	Informal	Analista de Sistemas

Tabela 4.6: Artefatos da área de conhecimento de Gestão de Configuração.

4.7 Área de conhecimento de Gestão de Mudanças

Esta área de conhecimento visa garantir que as **solicitações de mudanças sejam processadas de modo padronizado e passem por um processo de avaliação de impacto e aprovação**, além de fazer com que todos os envolvidos no projeto sempre estejam bem informados. Esta área de conhecimento possibilita mitigar os riscos da implantação das mudanças, minimizar interrupções nos sistemas, evitar mudanças não autorizadas e balancear necessidades e impactos.

O seu objetivo é realizar estruturadamente o controle das mudanças relacionadas com os artefatos do projeto (decorrentes de correções de falhas), melhorias na qualidade, inclusão de novos requisitos (os quais não fazem parte do escopo) etc. É essencial possuir um processo definido de gerenciamento de mudanças para assegurar que as mudanças feitas no produto em desenvolvimento sejam consistentes.

O gerenciamento de mudanças engloba todos os artefatos listados no modelo de processo. Esse controle é feito através das solicitações de mudanças, que devem ser registradas. As solicitações de mudanças são artefatos usados para documentar e controlar qualquer tipo de alteração realizada nos artefatos de um projeto. A vantagem das solicitações de mudança é que elas fornecem um registro das decisões e, devido ao seu processo de avaliação, garantem que os riscos e impactos das mudanças sejam entendidos.

É importante ressaltar que as solicitações de mudanças relacionadas com os artefatos que já possuem uma versão estável (compõem a baseline) deverão obrigatoriamente passar por este processo formal de gestão de mudanças. A Figura 4.7 mostra o processo e as atividades de gestão de mudanças, incluído na área de conhecimento de gestão de mudanças.

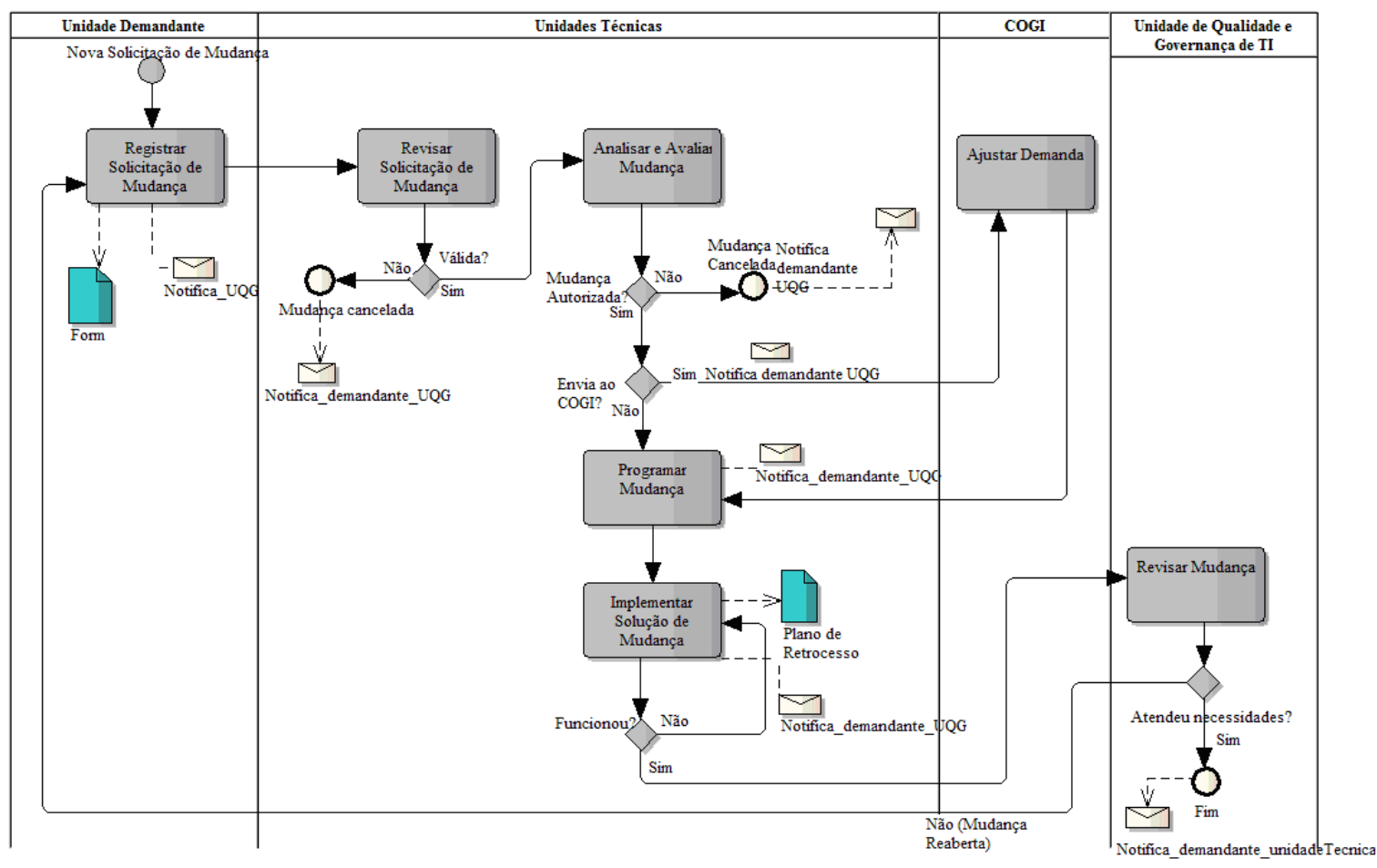


Figura 4.7: Processo de Gestão de Mudanças.

No processo de gerenciamento de mudanças é importante a criação do comitê consultivo de mudanças (CCM), que deve ser formado por uma equipe multidisciplinar: usuários; programadores; analistas de sistema e de negócio; coordenadores de projetos; gerente de sistema; diretor de TI e pelos membros do COGI. Este comitê é criado, sempre que necessário, com a finalidade de auxiliar a avaliação e aprovação das mudanças. As atividades necessárias para concluir o processo de gerenciamento de mudanças são:

4.7.1 Registrar solicitação de mudança

Esta atividade é o ponto de partida para o início do processo e objetiva realizar o registro da solicitação de mudança, incluindo todas as informações necessárias para identificação, o aceite e posterior avaliação. Este registro será feito inicialmente por e-mail com o formulário de solicitação de mudança em anexo, que será direcionado para a unidade técnica responsável e com notificação para a unidade de qualidade de software. É importante ressaltar que toda solicitação de mudança deve possuir um status. Este status mostra o

andamento da solicitação dentro do processo de gerenciamento de mudanças. Os status válidos podem ser vistos através da Tabela 4.7.

Código	Status	Descrição
01	Nova	Solicitação de mudança foi especificada e submetida.
02	Aceita	Solicitação de mudança aceita e passará pelas outras etapas do processo de gerenciamento de mudanças.
03	Rejeitada	Solicitação de mudança sem informações suficientes.
04	Duplicada	A solicitação de mudança é uma duplicata de outra já existente.
05	Autorizada	Solicitação de mudança autorizada para ser implementada.
06	Aguardando Parecer do COGI	A solicitação de mudança está sendo avaliada pelo COGI.
07	Aguardando Programação	A implementação da solicitação de mudança foi autorizada, mas está aguardando para entrar em implementação.
08	Planejando	A solução para mudança está sendo planejada.
09	Implementando	Solicitação de mudança sendo implementada.
10	Em Teste	Solicitação de mudança sendo testada.
11	Reprovada	O teste comprovou que a implementação realizada continha defeitos e a mudança não poderia ser implantada.
12	Aprovada	Mudança pronta para ser implantada.
13	Aguardando Implantação	A solicitação de mudança está implementada, mas está aguardando para ser implantada.
14	Implantando	A mudança está sendo implantada.
15	Implantada	A Mudança foi implantada.
16	Em Revisão	A mudança está sendo revisada.
17	Fechada	A mudança foi revisada, atingiu o retorno esperado e encontra-se fechada.
18	Cancelada	A mudança foi rejeitada, não autorizada ou estava duplicada, portanto foi cancelada.
19	Reaberta	A mudança foi revisada, mas não atingiu o retorno esperado e foi novamente aberta.

Tabela 4.7: Status de uma solicitação de mudança.

4.7.2 Revisar solicitação de mudança

Esta atividade objetiva aceitar ou rejeitar as solicitações de mudanças que foram registradas na etapa anterior. Para isso, realiza-se um pequeno filtro que avalia se a requisição de mudança não se encontra confusa, ilógica ou se a mesma é desnecessária ou impraticável, motivos que podem levar a sua rejeição. Quando a rejeição acontece, a solicitação retorna (junto com a sua documentação, incluindo os motivos da rejeição) para o seu solicitante, que tem a oportunidade de defender a sua requisição e a UQG deve ser notificada. No caso da solicitação ser aprovada, a UQG e o solicitante devem ser notificados.



4.7.3 Analisar e Avaliar Mudança

Esta atividade consiste em analisar e avaliar os impactos causados por uma mudança. É necessário realizar uma análise de riscos para identificar fatores que possam impedir a entrega do sistema ou corromper o negócio, impactando, desta forma, os objetivos estratégicos do órgão. Esta análise de riscos é necessária para que as ações corretas sejam tomadas a tempo e da melhor maneira possível.

Ainda nesta atividade, é necessário estabelecer a priorização das mudanças, através da matriz de urgência e impacto, considerando os riscos, os custos e os benefícios das mudanças. Esta priorização se faz necessária para determinar o quanto uma mudança é importante em relação às outras.

Inicialmente, o solicitante sugere o impacto e urgência da mudança, mas é responsabilidade do gerenciamento de mudanças estabelecer a prioridade final das mudanças, depois que o mesmo leva em consideração as prioridades das outras mudanças que estão sendo processadas.

As tabelas 4.8, 4.9, 4.10 e 4.11 mostram sugestões de como estabelecer a prioridade de uma mudança. A Tabela 4.8 apresenta a matriz de urgência e impacto. Através dela é possível estabelecer o nível de prioridade da mudança. As tabelas 4.9 e 4.10 estabelecem as definições dos níveis de impacto e urgência (para auxiliar nas suas atribuições) que devem sempre considerar a análise de risco, custo e benefício da mudança. A Tabela 4.11 mostra os níveis de prioridade com as suas respectivas descrições.

Esta atividade também é responsável pela autorização da mudança, uma vez que isso é necessário para que a mudança possa ser implementada. Caso seja autorizada, a mudança é encaminhada para a equipe técnica adequada e notificada ao solicitante e à UQG. Caso não seja, a solicitação retorna para o solicitante e a UQG é notificada.

Urgência	Impacto			Níveis de Prioridade
	Alto	Médio	Baixo	
Alta	01	02	03	
Média	02	03	04	
Baixa	03	04	04	

Tabela 4.8: Matriz de urgência e impacto de uma mudança.



Nível de Urgência	Descrição
Alta	A mudança deve ser realizada em caráter de urgência, pois a probabilidade de ocorrência dos riscos associados, em algum momento do projeto é igual ou maior que 50%.
Média	A mudança não pode esperar muito, pois a probabilidade de ocorrência dos riscos associados, em algum momento do projeto é igual ou maior que 10% e menor que 50%.
Baixa	A mudança pode esperar, pois a probabilidade de ocorrência dos riscos associados, em algum momento do projeto é menor que 10%.

Tabela 4.9: Descrição dos níveis de urgência.

Nível de Impacto	Descrição
Alto	A mudança acarreta perdas severas no projeto. O impacto dos riscos associados no escopo, tempo, custo ou qualidade do projeto é igual ou maior a 10% do total previsto.
Médio	A mudança causa um impacto significativo (mas não severo como o nível alto), e não pode demorar muito para acontecer. O impacto dos riscos associados no escopo, tempo, custo ou qualidade do projeto é igual ou maior que 5% e menor que 10% do total previsto.
Baixo	A mudança é justificada e necessária, mas pode esperar. O cujo impacto no escopo, tempo, custo ou qualidade do projeto é menor que 5% do total previsto.

Tabela 4.10: Descrição dos níveis de impacto.

Nível	Prioridades das Mudanças
01	Imediata: acarreta perdas severas no projeto;
02	Alta: afeta com severidade o projeto, mas não acarreta as perdas que a prioridade imediata causa;
03	Média: possui um impacto significativo para projetos e não pode demorar muito para acontecer;
04	Baixa: a mudança é justificada e necessária, mas pode esperar.

Tabela 4.11: Prioridades de uma mudança.

4.7.4 Ajustar Demanda

Esta atividade objetiva ajustar as divergências de prioridade que venham a ocorrer. Este ajuste será realizado pelo COGI que, de acordo com as suas necessidades, mas considerando as análises e prioridades definidas pelo corpo técnico da DTI, realizará o ajuste na prioridade das mudanças autorizadas.

4.7.5 Programar Mudança

Esta atividade objetiva programar a execução da mudança para que sejam realizadas no melhor momento.

4.7.6 Implementar Solução Mudança

Esta atividade objetiva executar as atividades de planejamento da solução (incluindo o plano de retrocesso), implementação, testes (que toda mudança deve passar antes de ser implantada) e implantação (colocar a solução em ambiente de produção) da mudança. Esta atividade deve garantir que a mudança foi devidamente testada. É importante que haja um



plano de retrocesso, para permitir o retorno ao estágio imediatamente anterior ao da implantação da mudança, antes que ela seja disponibilizada no ambiente de produção.

4.7.7 Revisar Mudança

Esta atividade objetiva revisar a mudança implantada, após um determinado período (05 dias úteis), para ver se a mudança atingiu o resultado desejado e se os usuários estão satisfeitos. Caso a mudança não tenha atingido seu objetivo, a UQG junto com o gerente de sistemas e caso necessário o CCM devem decidir os procedimentos a serem realizados, o que pode envolver retornar ao início do fluxo. Esta revisão deve focar na satisfação dos usuários à que a solicitação da mudança esta relacionada. Tanto o solicitante quanto a unidade técnica devem ser notificados do resultado da revisão.

A Tabela 4.12 mostra os artefatos produzidos nesta área de conhecimento e em que momento (fase) os mesmos se fazem necessários.

Artefatos da área de conhecimento de Gestão de Mudanças						
Artefatos	Fase de Utilização dos Artefatos				Detalhes da Revisão	Responsável
	Concepção	Elaboração	Construção	Transição		
Solicitação de Mudança	O	O	O	O	Informal	Envolvidos (Projeto/Sistema)
Avaliação de Mudança	O	O	O	O	Informal	Analista de Sistema
Revisão de Mudança	O	O	O	O	Informal	Coordenador de Qualidade de Software

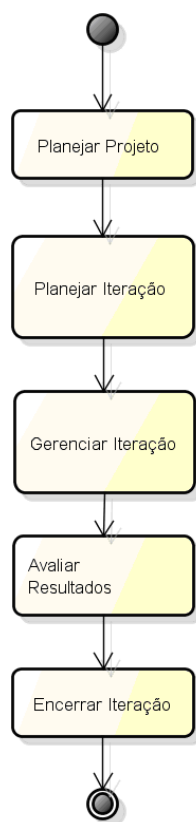
Tabela 4.12: Artefatos da área de conhecimento de Gestão de Mudanças.

4.8 Área de conhecimento de Gestão de Projetos

A finalidade da área de conhecimento de gestão de projetos é fornecer uma estrutura para gerenciar projetos de software, proporcionando diretrizes para **planejar, executar e controlar os projetos**, enquanto gerencia risco. No entanto, esta área de conhecimento não cobre o gerenciamento de pessoal, aquisição ou comunicação. Esta área de conhecimento foca principalmente nos aspectos importantes do processo de desenvolvimento iterativo como, por exemplo: o gerenciamento de risco, o planejamento iterativo do projeto, além do monitoramento do progresso do projeto, através de métricas e indicadores preestabelecidos.

4.8.1 Processo e Atividades Envolvidas

Esta seção descreve as atividades relacionadas com o processo da área de conhecimento de Gestão de Projetos. A Figura 4.8 mostra o processo e as atividades relacionadas com a área de conhecimento de Gestão de Projetos.



*Figura 4.8:
Processo da
área de
conhecimento
de Gestão de
Projetos.*

4.8.1.1 Planejar Projeto

Esta atividade envolve: (i) a identificação dos envolvidos no projeto; (II) a definição da equipe de trabalho; (III) a estimativa prazo e custo do projeto; (IV) identificação e avaliação de riscos; (v) a priorização dos requisitos, premissas e restrições do projeto. As principais saídas desta atividade são o plano de projeto e a lista de riscos.

4.8.1.2 Planejar Iteração

Esta atividade objetiva detalhar o trabalho da iteração, baseado na priorização de requisitos obtida na atividade anterior. Após o detalhamento do trabalho, os riscos identificados devem ser revisados. Esta atividade resulta na criação do cronograma do projeto.

4.8.1.3 Gerenciar Execução da Iteração

Esta atividade abrange o trabalho diário e contínuo do coordenador de projetos. Esta atividade aborda: (i) o tratamento das solicitações de mudanças e o respectivo planejamento



das suas implementações; (II) o monitoramento contínuo de riscos ativos e medições objetivas do andamento do projeto e da qualidade do produto e (III) o desenvolvimento da avaliação do andamento do projeto, com o direcionamento dos problemas descobertos para uma solução.

4.8.1.4 Avaliar resultados

Essa atividade visa avaliar com a equipe de trabalho os resultados obtidos na iteração, identificando os pontos positivos e negativos para registrá-los nas lições aprendidas. Para isso, deve-se realizar uma reunião com a equipe do projeto, registrar e divulgar os dados obtidos na avaliação. A saída desta atividade é o documento de lições aprendidas.

4.8.1.5 Encerrar Iteração

Esta atividade objetiva realizar o encerramento da iteração do projeto, através da aceitação formal do release e da execução dos procedimentos administrativos como redistribuição da equipe, pagamento de custos e atualização do status do projeto para toda equipe. Ainda nesta atividade deve ser realizada a medição final do tamanho software, atualizando os valores obtidos no início do projeto com os valores reais da entrega.

A Tabela 4.13 mostra os artefatos produzidos nesta área de conhecimento e em que momento (fase) os mesmos se fazem necessários.



Artefatos da área de conhecimento de Gestão de Projetos

Artefatos	Fase de Utilização dos Artefatos				Detalhes da Revisão	Responsável
	Concepção	Elaboração	Construção	Transição		
Plano de Projeto	O	O	O	O	Formal-Interno	Coordenador de Projetos
Cronograma do Projeto	O	O	O	O	Formal-Interno	Coordenador de Projetos
Avaliação de Andamento do Projeto	P	O	O	O	Informal	Coordenador de Projetos
Lista de Riscos	O	O	O	O	Formal-Interno	Coordenador de Projetos
Lições aprendidas	O	O	O	O	Formal-Interno	Coordenador de Projetos
Documento de Aceite	O	O	O	O	Formal-Externo	Coordenador de Projetos
Documento de Entrega/Recebimento	-	R	R	R	Formal-Interno	Coordenador de Projetos
Registro de Revisão	R	R	R	R	Informal	Coordenador de Projetos

Tabela 4.13: Artefatos da área de conhecimento de Gestão de Projetos.

4.9 Área de conhecimento de Gestão de Qualidade

A área de conhecimento de Gestão de Qualidade é uma área de conhecimento adicional, mas fundamental para o modelo de gestão e desenvolvimento de software, apesar de tratar de assuntos assessoriais. Esta área de conhecimento concentra-se nas atividades normativas e direcionadoras do processo de gestão e desenvolvimento de software. Ela descreve os artefatos que serão utilizados como **diretrizes para o processo de gestão e desenvolvimento**. A meta desta área de conhecimento é oferecer ao MPBA o ambiente que dará suporte às equipes dos projetos e ao processo de desenvolvimento. Esta área de conhecimento não possui um fluxo de trabalho definido, como as outras, devendo apenas, em períodos de tempo pré-definidos, rever os seus artefatos com a finalidade de aumentar a sua eficiência e melhorar os seus resultados. A Tabela 4.14 mostra os artefatos produzidos nesta área de conhecimento e em que momento (fase) os mesmos se fazem necessários.



Artefatos da área de conhecimento de Gestão de Qualidade

Artefatos	Fase de Utilização dos Artefatos				Detalhes da Revisão	Responsável
	Concepção	Elaboração	Construção	Transição		
Guias de Especificação de Testes Formais	O	O	O	O	Formal-Interno	Analista de Testes
Guias e Formulários de Banco de Dados	O	O	O	O	Formal-Interno	Coordenador de AD
Guia de Desenvolvimento	O	O	O	O	Formal-Interno	Analista de Sistemas
Guia de Projeto de Software	O	O	O	O	Formal-Interno	Analista de Sistemas
Guia de Acessibilidade	O	O	O	O	Formal-Interno	Analista de Sistemas
Guia de Usabilidade	O	O	O	O	Formal-Interno	Analista de Sistemas
Templates Específicos	O	O	O	O	Formal-Interno	Analista de Sistemas
Lista de Ferramentas	O	O	O	O	Formal-Interno	Analista de Sistemas
Modelo de Gestão e Desenvolvimento	O	O	O	O	Formal-Interno	Analista de Sistemas
Modelo de Arquitetura de Referência	O	O	O	O	Formal-Interno	Analista de Sistemas
Guia de Configuração	O	O	O	O	Formal-Interno	Analista de Sistemas
Guia de Gerenciamento de Risco	O	O	O	O	Formal-Interno	Analista de Sistemas
Guia de Verificações de Qualidade	O	O	O	O	Formal-Interno	Analista de Sistemas
Guia de Configuração	O	O	O	O	Formal-Interno	Analista de Sistemas
Plano de Metas e Métricas de Qualidade	O	O	O	O	Formal-Interno	Analista de Sistemas

Tabela 4.14: Artefatos da área de conhecimento de Gestão de Qualidade.

5 Papéis principais

Analista de Sistemas – líder e coordenador da análise de requisitos; delimita o escopo e descreve as funcionalidades do sistema; projeta e desenvolve a solução de software; planeja e implanta a solução desenvolvida; mantém e controla os artefatos do projeto em repositório para acesso compartilhado, e controlado; avalia mudanças e seus impactos.

Coordenador de AD (Administração de Dados) – mantém salvaguarda dos modelos e dicionários de dados; auxilia na construção e evolução dos referidos modelos.



Analista de Testes – planeja, especifica e sumariza os resultados dos testes de sistema; revisa as mudanças realizadas no projeto; responde por todas as atividades da área de conhecimento de Gestão de Qualidade.

Coordenador de Projetos – responde por todas as atividades da Gestão de Projetos.

6 Conclusão

Este modelo visa melhorar o processo de gestão e desenvolvimento de software do MPBA e beneficiar toda a Unidade de Desenvolvimento de Sistemas da Diretoria de Tecnologia da Informação, pois contribuirá não só para alcançar metas estratégicas, mas também para mitigar incidentes relacionados com o levantamento de requisitos, resolução de problemas, controle de mudanças e aceitação dos componentes e produtos disponibilizados.

Desta forma, este modelo procura padronizar a forma como os softwares serão desenvolvidos e alterados dentro do órgão, criando, com isso, um vocabulário único entre toda a Unidade de Desenvolvimento de Sistemas da Diretoria de TI, incluindo a fábrica de software, com a finalidade de aumentar a qualidade dos produtos e a satisfação de seus usuários.