



**PODER JUDICIÁRIO
TRIBUNAL DE JUSTIÇA DO MARANHÃO
DIRETORIA DE INFORMÁTICA E AUTOMAÇÃO**

**PROCESSO DE TRABALHO
PROCESSO GERENCIAMENTO DE ARQUITETURA DE SOFTWARE**

Sumário

1. Objetivo	2
2. Definições	2
2.1. Cliente.....	2
2.2. Arquiteto de Software	2
2.3. Analista de Requisitos	2
2.4. Gerente de Projetos	2
3. Processo: Cumprimento de Requisição.....	3
3.1. Papéis e Responsabilidades.....	3
3.2. Fluxo do Processo.....	7
3.3. Entradas e Saídas do Processo	8
3.4. Descrição do Fluxo do Processo	8
4. Matriz RECI (RACI)	12
5. Controles do Processo.....	13
5.1. Indicadores do Processo.....	13
6. Histórico de Revisão e Periodicidade	14



**PODER JUDICIÁRIO
TRIBUNAL DE JUSTIÇA DO MARANHÃO
DIRETORIA DE INFORMÁTICA E AUTOMAÇÃO**

**PROCESSO DE TRABALHO
PROCESSO GERENCIAMENTO DE ARQUITETURA DE SOFTWARE**

1. Objetivo

Estabelece um padrão de trabalho para o processo de Processo de Gerenciamento de Arquitetura de Software.

2. Definições

2.1. Cliente

Representa os interesses do negócio. Deve ter conhecimento suficiente para decidir sobre os aspectos negociais, apoiar o Gerente de Projeto e a equipe do projeto.

2.2. Arquiteto de Software

Responsável por estabelecer a estrutura geral de cada visão de arquitetura: a decomposição da visão, o agrupamento dos elementos e as interfaces entre esses principais agrupamentos. Portanto, comparado aos outros papéis, a visão do arquiteto de software é ampla, e não detalhada.

2.3. Analista de Requisitos

Responsável por identificar os problemas e necessidades do cliente referente ao sistema em desenvolvimento ou em manutenção, propondo uma solução sistemática e detalhamento de todos os requisitos do sistema (Funcionais, não funcionais e Restrições do projeto). Conduz o levantamento das informações necessárias para análise e validação do sistema junto ao cliente e a equipe técnica de desenvolvimento do projeto.

2.4. Gerente de Projetos

Profissional experiente na gestão de projetos que zela pela sua correta execução e ajuda a equipe na resolução de possíveis impedimentos, dando apoio para que as atividades sejam executadas dentro do escopo, prazo e custos planejados.

Conhecimentos e habilidades:

- Conhecimento do processo e das práticas de Gestão de Projetos
- Capacidade de disseminação de conhecimento
- Compreensão do processo de mudanças
- Trabalho em equipe



**PODER JUDICIÁRIO
TRIBUNAL DE JUSTIÇA DO MARANHÃO
DIRETORIA DE INFORMÁTICA E AUTOMAÇÃO**

**PROCESSO DE TRABALHO
PROCESSO GERENCIAMENTO DE ARQUITETURA DE SOFTWARE**

3. Processo: Cumprimento de Requisição

3.1. Papéis e Responsabilidades

PAPEL	RESPONSÁVEL	ATIVIDADES
Arquiteto de Software		<ol style="list-style-type: none">1. Elencar possíveis soluções técnicas para atender os requisitos iniciais do projeto, considerando:<ul style="list-style-type: none">- O ambiente tecnológico;- Proposta arquitetural (se houver);- Arquiteturas e soluções publicadas na base de conhecimento;- Padrões de projeto;- Recursos de mercado;- Soluções de referência publicados na comunidade técnica.Se já existir uma arquitetura de projetos, deve-se avaliar se a ela atende as necessidades técnicas dos requisitos apresentados e caso contrário levantar possíveis pontos de melhoria a serem levados ao cliente para aprovação2. Realizar a escolha da arquitetura ideal, levando em consideração as linguagens (ex. PHP, Java, .Net), ambiente de execução (servidor de aplicação, de banco de dados) e integrações com demais sistemas3. Realizar Provas de Conceito para testar a eficácia de uma nova solução técnica, arquitetura proposta ou fatores que apresentem incerteza na sua viabilidade com o objetivo de mitigar riscos arquiteturais.4. Atualizar Arquitetura com base na análise dos requisitos elaborados para a iteração e avaliando o impacto na arquitetura5. Detalhar Arquitetura: descrever, especificar, diagramar e exemplificar como será o funcionamento do sistema. Utilizar, se necessário:<ul style="list-style-type: none">- UML: diagramas de classe, de sequência, de implantação e padrões de projeto



PODER JUDICIÁRIO
TRIBUNAL DE JUSTIÇA DO MARANHÃO
DIRETORIA DE INFORMÁTICA E AUTOMAÇÃO

PROCESSO DE TRABALHO
PROCESSO GERENCIAMENTO DE ARQUITETURA DE SOFTWARE

PAPEL	RESPONSÁVEL	ATIVIDADES
		<p>- Mapa de Integração: que identifica e detalha as interfaces com os demais sistemas trazendo informações como, protocolo de comunicação e parâmetros de entrada e saída"</p> <p>6. Elaborar os diagramas abaixo, conforme necessidade:</p> <ul style="list-style-type: none">- Diagrama de Classe- Diagrama de Pacote;- Diagrama de Colaboração;- Diagrama de Implantação.- Diagrama de Sequência (*)- Diagrama de Componentes (*) <p>(*) Elaborar se:</p> <ul style="list-style-type: none">- Casos de Uso significativo para o negócio, ou seja, representa ponto crítico para o sistema, pois o mesmo possibilita a interação de outros casos de uso;- Casos de uso possui uma grande quantidade de consultas, integrações, processamento, validações e/ou cálculos <p>7. Planejar a integração do produto, definindo uma estratégia e identificando o ambiente e os procedimentos necessários para a integração</p> <p>8. Alocar servidores e banco de dados, estabelecer conectividades, atribuir licenças e permissões à equipe</p> <p>9. Criar o(s) ambiente(s) de desenvolvimento integrado</p> <p>10. Construir a infraestrutura arquitetural com todos os elementos definidos no documento de arquitetura e guias, tais como os frameworks, padrões de projeto, componentes e configurações de servidores e banco de dados</p> <p>11. Atualizar as estratégias de integração contidas no documento de arquitetura e/ou documento de especificação de integração de produto observando os seguintes itens:</p> <ul style="list-style-type: none">- O sequenciamento de integração está consolidado;- Os padrões para o desenvolvimento foram observados;- A definição do ambiente de integração é viável;- A estratégia e procedimentos de integração estão claros



PODER JUDICIÁRIO
TRIBUNAL DE JUSTIÇA DO MARANHÃO
DIRETORIA DE INFORMÁTICA E AUTOMAÇÃO

PROCESSO DE TRABALHO
PROCESSO GERENCIAMENTO DE ARQUITETURA DE SOFTWARE

PAPEL	RESPONSÁVEL	ATIVIDADES
		<ol style="list-style-type: none">12. Criar/atualizar documento Notas de Release com as informações e restrições da versão.13. Realizar a integração conforme procedimentos e critérios estabelecidos anteriormente a fim de obter um produto executável a partir do código-fonte14. Realizar a liberação da versão integrada
Ferramenta de Integração		<ol style="list-style-type: none">1. O serviço analisa a conformidade das regras cadastradas com o código-fonte e armazena o resultado.2. A ferramenta, após a análise estática do código, emite um feedback com o status da análise. Caso não haja erro será feita a verificação da integração. Caso haja a atividade é encerrada até que seja re-estartada.3. Cada integração é verificada por um build automatizado (incluindo testes que são fundamentais) para detectar erros e devolver um feedback o mais breve possível. Essa abordagem reduz significativamente problemas com integração e permite que a equipe desenvolva software mais coeso de forma mais rápida. No build de integração pode haver várias tarefas como por exemplo: Compilação, Testes Unitários, Cópia de bibliotecas externas, Criação de Banco de Dados, Empacotamento, etc. Não existe um limite estabelecido para o número de tarefas realizadas no build. As tarefas configuradas são:<ul style="list-style-type: none">• Fazer o checkout• Resolver dependências• Compilar• Testar - Testes unitários automatizados. Teste de código (Testa comportamento/estado Ex. Chama o método com alguns parâmetros e verifica se o retorno é o esperado)4. Se a verificação não falhou:<ul style="list-style-type: none">• Empacotar (gerar um EAR - binário implantável com a versão)• Armazenar (no repositório de artefatos)5. Se a verificação falhou a ferramenta emite notificação (e-mail) automaticamente para a



**PODER JUDICIÁRIO
TRIBUNAL DE JUSTIÇA DO MARANHÃO
DIRETORIA DE INFORMÁTICA E AUTOMAÇÃO**

**PROCESSO DE TRABALHO
PROCESSO GERENCIAMENTO DE ARQUITETURA DE SOFTWARE**

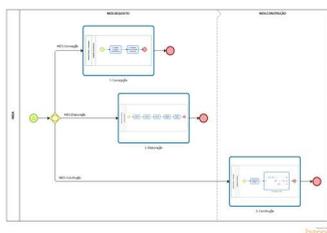
PAPEL	RESPONSÁVEL	ATIVIDADES
		equipe a respeito do problema encontrado na "quebra" do código.



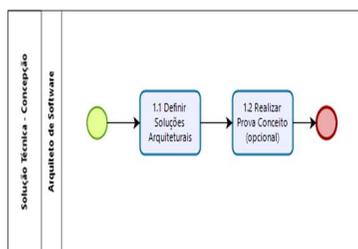
PODER JUDICIÁRIO
TRIBUNAL DE JUSTIÇA DO MARANHÃO
DIRETORIA DE INFORMÁTICA E AUTOMAÇÃO

PROCESSO DE TRABALHO
PROCESSO GERENCIAMENTO DE ARQUITETURA DE SOFTWARE

3.2. Fluxo do Processo

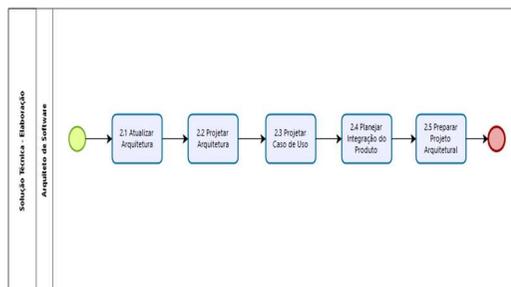


Concepção



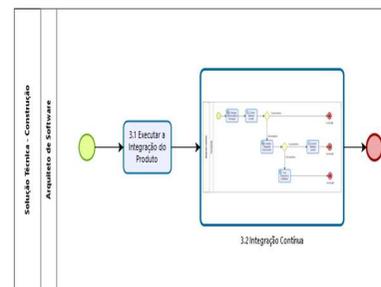
Powered by
bizagi
Modeler

Elaboração



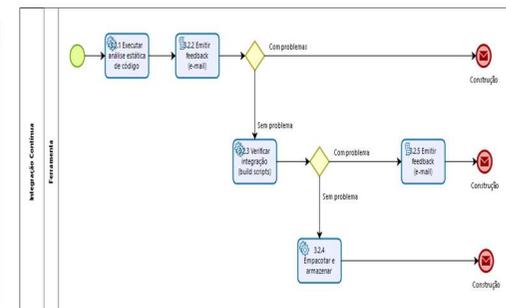
Powered by
bizagi
Modeler

Construção



Powered by
bizagi
Modeler

Integração Contínua



Powered by
bizagi
Modeler



**PODER JUDICIÁRIO
TRIBUNAL DE JUSTIÇA DO MARANHÃO
DIRETORIA DE INFORMÁTICA E AUTOMAÇÃO**

**PROCESSO DE TRABALHO
PROCESSO GERENCIAMENTO DE ARQUITETURA DE SOFTWARE**

3.3. Entradas e Saídas do Processo

ENTRADAS	SAÍDAS
<ul style="list-style-type: none">- Template Documento de Arquitetura- Template Especificação de Integração de Produto- Template Mapa de Integração- Template Plano de Implantação- Template Soluções Candidatas	<ul style="list-style-type: none">- Ambiente(s) criado(s)- Build- Código-Fonte da POC- Diagramas- Documento de Arquitetura- Especificação de Integração de Produto- Infraestrutura criada- Mapa de Integração- Nota de Release- Notificação (e-mail)- Pacote de Sistema- Plano de Implantação- Soluções Candidatas

3.4. Descrição do Fluxo do Processo

#IT	Atividades	Responsável	Procedimentos
	1.1 DEFINIR Soluções Arquiteturais		<ul style="list-style-type: none">- Elencar possíveis soluções técnicas para atender os requisitos iniciais do projeto, considerando:<ul style="list-style-type: none">- O ambiente tecnológico;- Proposta arquitetural (se houver);- Arquiteturas e soluções publicadas na base de conhecimento;- Padrões de projeto;



PODER JUDICIÁRIO
TRIBUNAL DE JUSTIÇA DO MARANHÃO
DIRETORIA DE INFORMÁTICA E AUTOMAÇÃO

PROCESSO DE TRABALHO
PROCESSO GERENCIAMENTO DE ARQUITETURA DE SOFTWARE

#IT	Atividades	Responsável	Procedimentos
			<ul style="list-style-type: none">- Recursos de mercado;- Soluções de referência publicados na comunidade técnica. <p>Se já existir uma arquitetura de projetos, deve-se avaliar se a ela atende as necessidades técnicas dos requisitos apresentados e caso contrário levantar possíveis pontos de melhoria a serem levados ao cliente para aprovação</p> <ul style="list-style-type: none">- Realizar a escolha da arquitetura ideal, levando em consideração as linguagens (ex. PHP, Java, .Net), ambiente de execução (servidor de aplicação, de banco de dados) e integrações com demais sistemas
	1.2 REALIZAR Prova Conceito (opcional)		<ul style="list-style-type: none">- Realizar Provas de Conceito para testar a eficácia de uma nova solução técnica, arquitetura proposta ou fatores que apresentem incerteza na sua viabilidade com o objetivo de mitigar riscos arquiteturais.
	2.1 ATUALIZAR Arquitetura		<ul style="list-style-type: none">- Atualizar Arquitetura com base na análise dos requisitos elaborados para a iteração e avaliando o impacto na arquitetura
	2.2 PROJETAR Arquitetura		<ul style="list-style-type: none">- Detalhar Arquitetura: descrever, especificar, diagramar e exemplificar como será o funcionamento do sistema. <p>Utilizar, se necessário:</p> <ul style="list-style-type: none">- UML: diagramas de classe, de sequência, de implantação e padrões de projeto- Mapa de Integração: que identifica e detalha as interfaces com os demais sistemas trazendo informações como, protocolo de comunicação e parâmetros de entrada e saída
	2.3 PROJETAR Caso de Uso		<ul style="list-style-type: none">- Elaborar os diagramas abaixo, conforme necessidade:<ul style="list-style-type: none">- Diagrama de Classe- Diagrama de Pacote;- Diagrama de Colaboração;- Diagrama de Implantação.- Diagrama de Sequência (*)



PODER JUDICIÁRIO
TRIBUNAL DE JUSTIÇA DO MARANHÃO
DIRETORIA DE INFORMÁTICA E AUTOMAÇÃO

PROCESSO DE TRABALHO
PROCESSO GERENCIAMENTO DE ARQUITETURA DE SOFTWARE

#IT	Atividades	Responsável	Procedimentos
			<ul style="list-style-type: none">- Diagrama de Componentes (*)(*) Elaborar se:<ul style="list-style-type: none">- Casos de Uso significativo para o negócio, ou seja, representa ponto crítico para o sistema, pois ele possibilita a interação de outros casos de uso;- Casos de uso possui uma grande quantidade de consultas, integrações, processamento, validações e/ou cálculos"
	2.4 PLANEJAR Integração do Produto		<ul style="list-style-type: none">- Planejar a integração do produto, definindo uma estratégia e identificando o ambiente e os procedimentos necessários para a integração
	2.5 PREPARAR Projeto Arquitetural		<ul style="list-style-type: none">- Alocar servidores e banco de dados, estabelecer conectividades, atribuir licenças e permissões à equipe- Criar o(s) ambiente(s) de desenvolvimento integrado- Construir a infraestrutura arquitetural com todos os elementos definidos no documento de arquitetura e guias, tais como os frameworks, padrões de projeto, componentes e configurações de servidores e banco de dados
	3.1 EXECUTAR a Integração do Produto		<ul style="list-style-type: none">- Atualizar as estratégias de integração contidas no documento de arquitetura e/ou documento de especificação de integração de produto observando os seguintes itens:<ul style="list-style-type: none">- O sequenciamento de integração está consolidado;- Os padrões para o desenvolvimento foram observados;- A definição do ambiente de integração é viável;- A estratégia e procedimentos de integração estão claros"- Criar/atualizar documento Notas de Release com as informações e restrições da versão.- Realizar a integração conforme procedimentos e critérios estabelecidos anteriormente a fim de obter um produto executável a partir do código-fonte- Realizar a liberação da versão integrada
	3.2 INTEGRAÇÃO Contínua		



PODER JUDICIÁRIO
TRIBUNAL DE JUSTIÇA DO MARANHÃO
DIRETORIA DE INFORMÁTICA E AUTOMAÇÃO

PROCESSO DE TRABALHO
PROCESSO GERENCIAMENTO DE ARQUITETURA DE SOFTWARE

#IT	Atividades	Responsável	Procedimentos
	3.2.1 EXECUTAR análise estática de código		– O serviço analisa a conformidade das regras cadastradas com o código-fonte e armazena o resultado.
	3.2.2 EMITIR feedback		– A ferramenta, após a análise estática do código, emite um feedback com o status da análise. Caso não haja erro será feita a verificação da integração. Caso haja a atividade é encerrada até que seja re-estartada.
	3.2.3 VERIFICAR integração (build scripts)		– Cada integração é verificada por um build automatizado (incluindo testes que são fundamentais) para detectar erros e devolver um feedback o mais breve possível. Essa abordagem reduz significativamente problemas com integração e permite que a equipe desenvolva software mais coeso de forma mais rápida. No build de integração pode haver várias tarefas como por exemplo: Compilação, Testes Unitários, Cópia de bibliotecas externas, Criação de Banco de Dados, Empacotamento, etc. Não existe um limite estabelecido para o número de tarefas realizadas no build. As tarefas configuradas são: <ul style="list-style-type: none">• Fazer o checkout• Resolver dependências• Compilar• Testar - Testes unitários automatizados. Teste de código (Testa comportamento/estado Ex. Chama o método com alguns parâmetros e verifica se o retorno é o esperado)
	3.2.4 EMPACOTAR e armazenar		– Se a verificação não falhou: <ul style="list-style-type: none">• Empacotar (gerar um EAR - binário implantável com a versão)• Armazenar (no repositório de artefatos)
	3.2.5 EMITIR feedback		– Se a verificação falhou a ferramenta emite notificação (e-mail) automaticamente para a equipe a respeito do problema encontrado na "quebra" do código



PODER JUDICIÁRIO
TRIBUNAL DE JUSTIÇA DO MARANHÃO
DIRETORIA DE INFORMÁTICA E AUTOMAÇÃO

PROCESSO DE TRABALHO
PROCESSO GERENCIAMENTO DE ARQUITETURA DE SOFTWARE

4. Matriz RECI (RACI)

Atividades	Arquiteto de Software	Gerente de Projetos	Equipe Técnica	Cliente	Ferramenta de Integração	Analista de Requisitos	-
1.1 DEFINIR Soluções Arquiteturais	E	R/C	C	I	-	C	-
1.2 REALIZAR Prova Conceito (opcional)	E	R/C	C	I	-	C	-
2.1 ATUALIZAR Arquitetura	E	R/C	C	I	-	C	-
2.2 PROJETAR Arquitetura	E	R/C	C	I	-	C	-
2.3 PROJETAR Caso de Uso	E	R/C	C	I	-	C	-
2.4 PLANEJAR Integração do Produto	E	R/C	C	I	-	C	-
2.5 PREPARAR Projeto Arquitetural	E	R/C	C	I	-	C	-
3.1 EXECUTAR a Integração do Produto	E	R/C	C	I	-	C	-
3.2 INTEGRAÇÃO Contínua	-	R	-	-	-	-	-
3.2.1 EXECUTAR análise estática de código	E/I	R	C	-	E	-	-
3.2.2 EMITIR feedback	E/I	R	C	-	E	-	-
3.2.3 VERIFICAR integração (build scripts)	E/I	R	C	-	E	-	-
3.2.4 EMPACOTAR e armazenar	E/I	R	C	-	E	-	-
3.2.5 EMITIR feedback	E/I	R	C	-	E	-	-

LEGENDA: R – Responsável (*Accountable*) | E – Executor (*Responsible*) | C – Consultado (*Consulted*) | I – Informado (*Informed*)



**PODER JUDICIÁRIO
TRIBUNAL DE JUSTIÇA DO MARANHÃO
DIRETORIA DE INFORMÁTICA E AUTOMAÇÃO**

**PROCESSO DE TRABALHO
PROCESSO GERENCIAMENTO DE ARQUITETURA DE SOFTWARE**

5. Controles do Processo

5.1. Indicadores do Processo

Descrição	Definir com o cliente
Periodicidade	
Meta	
Cálculo	



**PODER JUDICIÁRIO
TRIBUNAL DE JUSTIÇA DO MARANHÃO
DIRETORIA DE INFORMÁTICA E AUTOMAÇÃO**

**PROCESSO DE TRABALHO
PROCESSO GERENCIAMENTO DE ARQUITETURA DE SOFTWARE**

6. Histórico de Revisão e Periodicidade

Número	Data	Versão	Responsável	Descrição
001	07/05/2020	1.0	CTIS/SONDA: Erick Saraiva	Elaboração do documento.