

ANEXO I.E
Metodologia de Desenvolvimento de Sistemas – MDS

Sumário

1	Introdução.....	3
2	Modelos de Referência.....	3
2.1	Processo Unificado.....	3
2.2	Metodologias Ágeis.....	4
2.2.1	Praxis.....	4
1.1	Unified Modeling Language (UML).....	6
1.2	PMBOK.....	9
2	Organização da Metodologia.....	10
2.3	Fases.....	11
2.3.1	Iniciação.....	11
2.3.2	Elaboração.....	11
2.3.3	Construção.....	12
2.3.4	Transição.....	12
2.3.5	Encerramento.....	12
2.3.6	Monitoramento e Controle.....	12
2.4	Disciplinas.....	13
2.4.1	Modelagem de Negócios.....	13
2.4.2	Requisitos.....	13
2.4.3	Análise e Design.....	13
2.4.4	Implementação.....	13
2.4.5	Teste.....	13
2.4.6	Implantação.....	14
2.4.7	Gerência de Projetos.....	14
2.5	Papéis.....	14
2.5.1	Analista de Negócio.....	14
2.5.2	Analista de Sistemas.....	14
2.5.3	Analista de Teste.....	14
2.5.4	Designer de Interface.....	15
2.5.5	Designer de Banco de Dados.....	15
2.5.6	Implementador.....	15
2.5.7	Testador.....	15
2.5.8	Gerente de Projetos.....	15
2.5.9	Patrocinador.....	15
2.5.10	Embaixador.....	16
2.5.11	Administrador de Dados.....	16
2.5.12	Arquiteto de Software.....	16
2.5.13	Documentador.....	16
2.5.14	Envolvido.....	16
2.5.15	Analista de Produção.....	17
2.6	Artefatos.....	17
2.6.1	Glossário de Termos Técnicos.....	18
2.6.2	Cronograma.....	18
2.6.3	Proposta de Sistema de Informação.....	18
2.6.4	Documento de Mensuração.....	19
2.6.5	Plano do Projeto.....	19
2.6.6	Especificação de Casos de Uso.....	19
2.6.7	Diagrama de Casos de Uso.....	19
2.6.8	Regras de Negócio.....	19
2.6.9	Especificação de Requisitos Não-Funcionais.....	19

2.6.10 Modelo de Classes.....	20
2.6.11 MER.....	20
2.6.12 Termo de Fim de Fase.....	20
2.6.13 Documento de Implantação.....	20
2.6.14 Plano de Testes.....	20
2.6.15 Formulário de Implantação.....	20
2.6.16 Termo de Fim de Projeto.....	20
2.6.17 Ata de Reunião.....	21
2.6.18 Evidências de Testes.....	21
2.6.19 Cadeia de Valor.....	21
2.6.20 Fluxos atuais	21
2.6.21 Fluxos Redesenhados.....	21
2.6.22 Relatório de Diagnóstico e Prognóstico.....	21
2.7 Roteiro Padrão.....	22
2.7.1 Iniciação.....	24
1.1.1 Elaboração.....	28
1.1.2 Construção.....	38
1.1.3 Transição.....	56
1.1.4 Encerramento.....	64
1.1.5 Monitoramento e Controle.....	66
2 Referências.....	67

1 Introdução

Este documento descreve a primeira versão da Metodologia de Desenvolvimento de Sistemas (MDS) da Agência Nacional de Transportes Aquaviários – ANTAQ.

Uma Metodologia de Desenvolvimento de Sistemas é composta por seqüências de atividades executadas de maneira organizada, que visam a construção de um Sistema de Informação. Como produtos das atividades, são gerados os artefatos, que formalizam o conhecimento produzido durante o processo e que servem como balizadores no processo de construção do software e como uma espécie de contrato entre os usuários demandantes, a área de tecnologia da ANTAQ eventuais subcontratados.

2 Modelos de Referência

2.1 Processo Unificado

O Processo Unificado (PU) de desenvolvimento de Software, concebido por Jacobson, Booch e Rumbaugh em 1999, é um framework que combina os ciclos iterativo e incremental para a construção de softwares. A especialização mais conhecida e utilizada do processo unificado é o RUP (Rational Unified Process).

A combinação dos ciclos iterativo e incremental no Processo Unificado é dada da seguinte forma: Na primeira dimensão do processo (eixo horizontal da figura abaixo), tem-se as macro-divisões que são chamadas de Fases. As fases (Iniciação, Elaboração, Construção e Transição) são subdivididas em uma série de iterações menores (I1, E1, E2, C1, ..., T2). O produto de cada iteração incrementa as funcionalidades contidas nas iterações entregues previamente.

Outra dimensão importante do processo unificado é a relativa às Disciplinas (Eixo Vertical da figura abaixo). Uma disciplina é um conjunto de atividades relacionadas a uma área de interesse importante em todo o projeto. A ênfase e os esforços alocados em cada disciplina variam em função do tempo. Por exemplo, enquanto os esforços na disciplina de Modelagem de Negócios são maiores no início do projeto, os mesmos vão se atenuando até chegarem à fase de Transição. Já a disciplina de Implantação tem comportamento inverso, já que a ênfase dada à mesma vai se acentuando até chegar a seu ápice, próximo do fim do projeto.

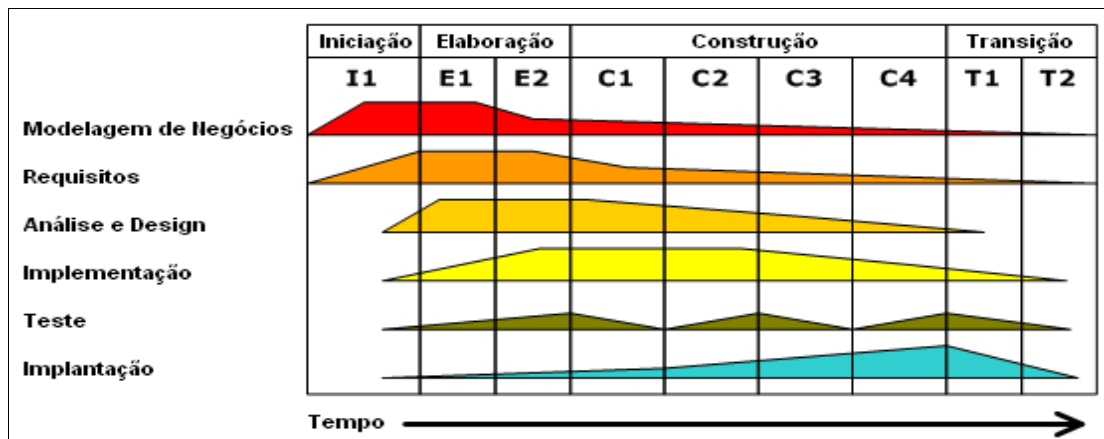


Figura – Processo Unificado: Iterativo e Incremental

Uma característica marcante do Processo Unificado é que o mesmo é adaptável para as necessidades específicas de cada organização onde será empregado. Várias são as adaptações derivadas do PU, por exemplo: Agile Unified Process (AUP), Basic Unified Process (BUP), Enterprise Unified Process (EUP), Essential Unified Process (EssUP), Open Unified Process (OpenUP), Rational Unified Process (RUP) e Oracle Unified Method (OUM). Da mesma forma, a MDS-ANTAQ será uma adaptação do Processo Unificado.

Maiores informações sobre o Processo Unificado podem ser encontradas em Kendall (2003).

2.2 Metodologias Ágeis

A maioria dos métodos ágeis tenta minimizar o risco pelo desenvolvimento do software em curtos períodos, chamados de iteração, os quais gastam tipicamente menos de uma semana a até quatro. Cada iteração é como um projeto de software em miniatura de seu próprio, e inclui todas as tarefas necessárias para implantar o mini-incremento da nova funcionalidade: planejamento, Análise de Requisitos, projeto, codificação, teste e documentação. Enquanto em um processo convencional, cada iteração não está necessariamente focada em adicionar um novo conjunto significativo de funcionalidades, um projeto de software ágil busca a capacidade de implantar uma nova versão do software ao fim de cada iteração, etapa a qual a equipe responsável reavalia as prioridades do projeto.

Métodos ágeis enfatizam comunicações em tempo real, preferencialmente face a face, a documentos escritos. A maioria dos componentes de um grupo ágil devem estar agrupados em uma sala. Isto inclui todas as pessoas necessárias para terminar o software. No mínimo, isto inclui os programadores e seus clientes (clientes são as pessoas que definem o produto, eles podem ser os gerentes, analistas de negócio, ou realmente os clientes). Nesta sala devem também se encontrar os testadores, projetistas de iteração, redatores técnicos e gerentes.

Métodos ágeis também enfatizam trabalho no software como uma medida primária de progresso. Combinado com a comunicação face-a-face, métodos ágeis produzem pouca documentação em relação a outros métodos, sendo este um de seus pontos negativos.

2.2.1 Praxis

O Praxis é uma metodologia ágil derivada do Processo Unificado desenhada para suportar projetos de seis meses a um ano de duração, realizados individualmente ou por pequenas

equipes. Por esse motivo, é aplicável a utilização de algumas práticas do Praxis na ANTAQ.

O processo Praxis é composto pelos seguintes elementos:

- **Passo.** Divisão formal de um processo, com pré-requisitos, entradas, critérios de aprovação e resultados definidos.
- **Fase.** Divisão maior de um processo, para fins gerenciais, que corresponde aos pontos principais de aceitação por parte do cliente.
- **Iteração.** Passo constituinte de uma fase, no qual se atinge um conjunto bem definido de metas parciais de um projeto.
- **Fluxo** Subprocesso caracterizado por um tema técnico.
- **Etapa** Passo constituinte de um fluxo.
- **Atividade** Termo genérico para unidades de trabalho executadas em um passo.

Os elementos descritos acima são utilizados para descrição das unidades de trabalho que compõem o Praxis. No estilo do Processo Unificado, o Praxis abrange tanto fases (subprocessos gerenciais) quanto fluxos (subprocessos técnicos / disciplinas). Uma fase é composta por uma ou mais iterações. Um fluxo é dividido em uma ou mais etapas. Iterações e etapas são exemplos de passos (Figura 13).

A relação entre os referidos elementos se dá da seguinte forma:

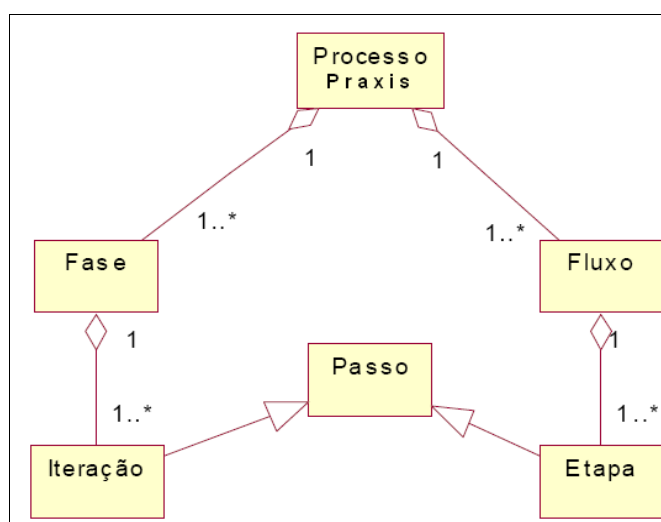


Figura Elementos do Processo Praxis

Os elementos maiores da arquitetura do Praxis, as Fases, são inspirados nos elementos correspondentes do Processo Unificado. Descreve-se abaixo o objetivo de cada fase:

- **Concepção.** Fase na qual necessidades dos usuários e conceitos da aplicação são analisados o suficiente para justificar a especificação de um produto de software, resultando em uma proposta de especificação.
- **Elaboração.** Fase na qual a especificação do produto é detalhada o suficiente para modelar conceitualmente o domínio do problema, validar os requisitos em termos deste modelo conceitual e permitir um planejamento acurado da fase de construção.
- **Construção.** Fase na qual é desenvolvida (desenhada, implementada e testada) uma liberação completamente operacional do produto, que atende aos requisitos especificados.
- **Transição.** Fase na qual o produto é colocado à disposição de uma comunidade de usuários para testes finais, treinamento e uso inicial.

Da mesma maneira, os Fluxos do Praxis são inspirados nas Disciplinas do Processo Unificado.

- **Requisitos.** Fluxo que visa obter um conjunto de requisitos de um produto, acordado entre cliente e fornecedor.
- **Análise.** Fluxo que visa detalhar, estruturar e validar os requisitos, em termos de um modelo conceitual do problema, de forma que estes possam ser usados como base para o planejamento e acompanhamento detalhados da construção do produto.
- **Desenho.** Fluxo que visa formular um modelo estrutural do produto, que sirva de base para a implementação, definindo os componentes a desenvolver e a reutilizar, assim como as interfaces entre si e com o contexto do produto.
- **Implementação.** Fluxo que visa detalhar e implementar o desenho através de componentes de código e de documentação associada.
- **Testes.** Fluxo que visa verificar os resultados da implementação, através do planejamento, desenho e realização de baterias de testes.

O método Praxis consolida os resultados produzidos e insumos consumidos nos passos do processo em artefatos do processo. Os artefatos podem ser documentos e modelos, conforme seus consumidores primários sejam humanos ou ferramentas. Os documentos permanentes oficiais especificados pelo Praxis são apenas sete, o que dá bastante agilidade ao desenvolvimento de sistemas, sem perder, no entanto, o formalismo necessário à documentação do processo. São os documentos:

- **Proposta de Especificação do Software.** Documento que delimita preliminarmente o escopo de um projeto, contendo um plano da fase de Elaboração.
- **Especificação dos Requisitos do Software.** Documento que descreve, de forma detalhada, o conjunto de requisitos especificados para um produto de software.
- **Plano de Desenvolvimento do Software.** Documento que descreve, de forma detalhada, os compromissos que o fornecedor assume em relação ao projeto, quanto a recursos, custos, prazos, riscos e outros aspectos gerenciais.
- **Plano da Qualidade do Software.** Documento que descreve, de forma detalhada, os procedimentos de garantia da qualidade que serão adotados no projeto.
- **Descrição do Desenho do Software.** Documento que descreve, de forma detalhada, os aspectos mais importantes do desenho do software.
- **Descrição dos Testes do Software.** Documento que descreve, de forma detalhada, os planos e especificações dos testes que serão executados
- **Manual do Usuário do Software.** Documento que serve de referência para uso do produto.

Maiores informações sobre o Praxis podem ser encontradas em PAULA FILHO (2003).

1.1 Unified Modeling Language (UML)

A Unified Modeling Language (UML) é uma linguagem de modelagem não proprietária de terceira geração. A UML não é uma metodologia de desenvolvimento, o que significa que ela não diz para você o que fazer primeiro e em seguida ou como projetar seu sistema, mas ela lhe auxilia a visualizar seu desenho e a comunicação entre objetos.

Basicamente, a UML permite que desenvolvedores visualizem os produtos de seu trabalho em diagramas padronizados. Junto com uma notação gráfica, a UML também especifica significados, isto é, semântica. É uma notação independente de processos, embora o RUP (Rational Unified Process) tenha sido especificamente desenvolvido utilizando a UML.

A UML é uma linguagem para especificação, documentação, visualização e desenvolvimento de sistemas orientados a objetos. Sintetiza os principais métodos existentes, sendo considerada uma das linguagens mais expressivas para modelagem de sistemas orientados a objetos. Por meio de seus diagramas é possível representar sistemas de softwares sob diversas perspectivas de visualização. Facilita a comunicação de todas as pessoas envolvidas no processo de desenvolvimento de um sistema - gerentes, coordenadores, analistas, designers e desenvolvedores - por apresentar um vocabulário de fácil entendimento.

Os esforços para a criação da UML tiveram início em outubro de 1994, quando Rumbaugh se juntou a Booch na Rational. Com o objetivo de unificar os métodos Booch e OMT, decorrido um ano de trabalho, foi lançado, em outubro de 1995, o esboço da versão 0.8 do Método Unificado (como era conhecido). Nesta mesma época, Jacobson se associou à Rational e o escopo do projeto da UML foi expandido para incorporar o método OOSE. Nasceu então, em junho de 1996, a versão 0.9 da UML. Atualmente, a UML encontra-se atualmente na versão 2.0 e é regida pelo consórcio OMG (*Object Management Group*).

Os diagramas UML subdividem-se em três grandes grupos: Estruturais, Comportamentais e Interação. Os diagramas de cada um desses grupos são descritos sucintamente abaixo:

- Estruturais

- Diagrama de Objetos - O diagrama de objetos é uma variação do diagrama de classes e utiliza quase a mesma notação. A diferença é que o diagrama de objetos mostra os objetos que foram instanciados das classes. Os diagramas de objetos não são tão importantes como os diagramas de classes, mas eles são muito úteis para exemplificar diagramas complexos de classes ajudando muito em sua compreensão. Diagramas de objetos também são usados como parte dos diagramas de colaboração, onde a colaboração dinâmica entre os objetos do sistema são mostrados.
- Diagrama de classes - O diagrama de classes é uma representação da estrutura e relações das classes que servem de modelo para objetos. É uma modelagem muito útil para o sistema, pois define todas as classes que o sistema necessita possuir sendo base para a construção dos diagramas de comunicação, sequência e estados. Uma classe é um descritor de um conjunto de objetos que partilham as mesmas propriedades (atributos, operações, relações e semântica). São usadas para capturar o vocabulário de um sistema., elementos do domínio do problema, como “Cliente”, “Banco”, “Conta”.
- Diagrama de componentes - O diagrama de componentes ilustra como as classes deverão se encontrar organizadas através da noção de componentes de trabalho. Por exemplo, pode-se explicitar, para cada componente, qual das classes que ele representa. É utilizado para: Modelar os componentes do código-fonte, do código executável do software; Destacar a função de cada módulo para facilitar a sua reutilização e; Auxiliar no processo de engenharia reversa, por meio da organização dos módulos do sistema e seus relacionamentos.
- Diagrama de instalação - O Diagrama de instalação descreve os componentes de hardware e software e sua interação com outros elementos de suporte ao processamento. Representa a configuração e a arquitetura de um sistema em que estarão ligados seus respectivos componentes, sendo representado pela arquitetura física de hardware, processadores, dentre outros.
- Diagrama de pacotes - O Diagrama de pacotes descreve os pacotes ou pedaços do sistema divididos em agrupamentos lógicos mostrando as dependências entre estes,

ou seja, pacotes podem depender de outros pacotes. Este diagrama é muito utilizado para ilustrar a arquitetura de um sistema mostrando o agrupamento de suas classes.

- Diagrama de estrutura - O Diagrama de estrutura destina-se à descrição dos relacionamentos entre os elementos. Utilizado para descrever a colaboração interna de classes, interfaces ou componentes para especificar uma funcionalidade.
- Diagramas Comportamentais
 - Diagrama de Caso de Uso - O Diagrama de Caso de Uso descreve a funcionalidade proposta para o novo sistema. Um Caso de Uso representa uma unidade discreta da interação entre um ator (humano ou máquina) e o sistema. Um ator é um humano ou entidade máquina que interage com o sistema para executar um significativo trabalho. Um Caso de Uso é uma unidade de um trabalho significativo. Cada Caso de Uso tem uma descrição o qual descreve a funcionalidade que irá ser construída no sistema proposto. Um Caso de Uso pode "incluir" outra funcionalidade de Caso de Uso ou "estender" outro Caso de Uso com seu próprio comportamento.
 - Diagrama de transição de estados – O diagrama de transição de estados é uma representação do estado ou situação em que um objeto pode se encontrar no decorrer da execução de processos de um sistema. Com isso, o objeto pode passar de um estado A (estado inicial) para um estado B (estado final) através de uma transição.
 - Diagrama de atividade - O Diagrama de atividade representa os fluxos conduzidos por processamentos. É essencialmente um gráfico de fluxo, mostrando o fluxo de controle de uma atividade para outra. Comumente isso envolve a modelagem das etapas seqüenciais em um processo computacional.
- Diagramas de Interação
 - Diagrama de sequência – O Diagrama de sequência representa seqüência de mensagens passadas entre objetos de um software. Como um projeto pode ter uma grande quantidade de métodos em classes diferentes, pode ser difícil determinar a seqüência global do comportamento. O diagrama de seqüência representa essa informação de uma forma simples e lógica, descrevendo a maneira como os grupos de objetos colaboram em algum comportamento ao longo do tempo. Ele registra o comportamento de um único caso de uso e exibe os objetos e as mensagens passadas entre esses objetos no caso de uso.
 - Diagrama de Interatividade – O Diagrama de Interatividade consiste na fusão do diagrama de atividades com o de seqüência. Ele permite que fragmentos das interações sejam facilmente combinados aos pontos e fluxos de decisão.
 - Diagrama de colaboração ou comunicação - O Diagrama de Colaboração exibe uma interação, consistindo de um conjunto de objetos e seus relacionamentos, incluindo as mensagens que podem ser trocadas entre eles. O diagrama mostra de maneira semelhante ao diagrama de seqüência, a colaboração dinâmica entre os objetos. Se a ênfase do diagrama for o decorrer do tempo, é melhor escolher o diagrama de seqüência, mas se a ênfase for o contexto do sistema, é melhor dar prioridade ao diagrama de colaboração. O diagrama de colaboração é desenhado como um diagrama de objeto, onde os diversos objetos são mostrados juntamente com seus relacionamentos.
 - Diagrama de tempo – Com o diagrama de tempo, ou diagrama temporal, é possível visualizar o impacto do tempo em um ou mais objetos e, as condições e os efeitos do

tempo nos estados dos objetos.

Maiores informações sobre a UML podem ser encontradas em <http://www.omg.org>.

1.2 PMBOK

O conjunto de conhecimentos em gerenciamento de projetos (PMBOK - *Project Management Body of Knowledge*) é a soma dos conhecimentos intrínsecos à disciplina de gerenciamento de projetos. O PMBOK inclui práticas tradicionais comprovadas amplamente aplicadas, além de práticas inovadoras que estão surgindo na área de gerenciamento de projetos.

O principal objetivo do PMBOK é identificar o subconjunto do Conjunto de conhecimentos em gerenciamento de projetos que é amplamente reconhecido como boa prática.

O PMBOK entende um Projeto como “um esforço temporário empreendido para criar um produto, serviço ou resultado exclusivo.”. Já o gerenciamento de projetos é entendido como “a aplicação de conhecimento, habilidades, ferramentas e técnicas às atividades do projeto a fim de atender aos seus requisitos.”.

Projetos são diferentes das operações continuadas uma vez que as operações são contínuas e repetitivas, enquanto os projetos são temporários e exclusivos. Portanto, os objetivos dos projetos e das operações são fundamentalmente diferentes: A finalidade de um projeto é atingir seu objetivo e, em seguida, terminar; Por outro lado, o objetivo de uma operação contínua é manter o negócio. Os projetos são diferentes porque o projeto termina quando seus objetivos específicos foram atingidos, enquanto as operações adotam um novo conjunto de objetivos e o trabalho continua.

O gerenciamento de projetos é realizado através de Processos, usando conhecimento, habilidades, ferramentas e técnicas do gerenciamento de projetos que recebem entradas e geram saídas. Um processo é um conjunto de ações e atividades inter-relacionadas realizadas para obter um conjunto pré-especificado de produtos, resultados ou serviços. O PMBOK reúne processos de gerenciamento de projetos reconhecidos como boa prática na maioria dos projetos, durante a maior parte do tempo. Esses processos se aplicam globalmente e também para qualquer setor, ou seja, não apenas à área de Tecnologia da Informação. O PMBOK agrupa os processos em cinco grupos distintos:

- **Grupo de processos de iniciação.** Define e autoriza o projeto ou uma fase do projeto.
- **Grupo de processos de planejamento.** Define e refina os objetivos e planeja a ação necessária para alcançar os objetivos e o escopo para os quais o projeto foi realizado.
- **Grupo de processos de execução.** Integra pessoas e outros recursos para realizar o plano de gerenciamento do projeto para o projeto.
- **Grupo de processos de monitoramento e controle.** Mede e monitora regularmente o progresso para identificar variações em relação ao plano de gerenciamento do projeto, de forma que possam ser tomadas ações corretivas quando necessário para atender aos objetivos do projeto.
- **Grupo de processos de encerramento.** Formaliza a aceitação do produto, serviço ou resultado e conduz o projeto ou uma fase do projeto a um final ordenado.

Uma outra forma de se organizar os quarenta e quatro processos do PMBOK é através das Áreas de Conhecimento. Cada área de conhecimento se refere a um aspecto a ser considerado dentro da gerência de projetos. Existem nove áreas de conhecimento, que são descritas sucintamente abaixo:

- **Gerenciamento de integração do projeto.** Descreve os processos e as atividades que integram os diversos elementos do gerenciamento de projetos, que são identificados, definidos, combinados, unificados e coordenados dentro dos grupos de processos de gerenciamento de projetos. Ele consiste nos processos de gerenciamento de projetos: Desenvolver o termo de abertura do projeto, Desenvolver a declaração do escopo preliminar do projeto, Desenvolver o plano de gerenciamento do projeto, Orientar e gerenciar a execução do projeto, Monitorar e controlar o trabalho do projeto, Controle integrado de mudanças e Encerrar o projeto.
- **Gerenciamento do escopo do projeto.** Descreve os processos envolvidos na verificação de que o projeto inclui todo o trabalho necessário, e apenas o trabalho necessário, para que seja concluído com sucesso. Ele consiste nos processos de gerenciamento de projetos: Planejamento do escopo, Definição do escopo, Criar EAP, Verificação do escopo e Controle do escopo.
- **Gerenciamento de tempo do projeto.** Descreve os processos relativos ao término do projeto no prazo correto. Ele consiste nos processos de gerenciamento de projetos: Definição da atividade, Seqüenciamento de atividades, Estimativa de recursos da atividade, Estimativa de duração da atividade, Desenvolvimento do cronograma e Controle do cronograma.
- **Gerenciamento de custos do projeto.** Descreve os processos envolvidos em planejamento, estimativa, orçamentação e controle de custos, de modo que o projeto termine dentro do orçamento aprovado. Ele consiste nos processos de gerenciamento de projetos: Estimativa de custos, Orçamentação e Controle de custos.
- **Gerenciamento da qualidade do projeto.** Descreve os processos envolvidos na garantia de que o projeto irá satisfazer os objetivos para os quais foi realizado. Ele consiste nos processos de gerenciamento de projetos: Planejamento da qualidade, Realizar a garantia da qualidade e Realizar o controle da qualidade.
- **Gerenciamento de recursos humanos do projeto.** Descreve os processos que organizam e gerenciam a equipe do projeto. Ele consiste nos processos de gerenciamento de projetos: Planejamento de recursos humanos, Contratar ou mobilizar a equipe do projeto, Desenvolver a equipe do projeto e Gerenciar a equipe do projeto.
- **Gerenciamento das comunicações do projeto.** Descreve os processos relativos à geração, coleta, disseminação, armazenamento e destinação final das informações do projeto de forma oportuna e adequada. Ele consiste nos processos de gerenciamento de projetos: Planejamento das comunicações, Distribuição das informações, Relatório de desempenho e Gerenciar as partes interessadas.
- **Gerenciamento de riscos do projeto.** Descreve os processos relativos à realização do gerenciamento de riscos em um projeto. Ele consiste nos processos de gerenciamento de projetos: Planejamento do gerenciamento de riscos, Identificação de riscos, Análise qualitativa de riscos, Análise quantitativa de riscos, Planejamento de respostas a riscos e Monitoramento e controle de riscos.
- **Gerenciamento de aquisições do projeto.** Descreve os processos que compram ou adquirem produtos, serviços ou resultados, além dos processos de gerenciamento de contratos. Ele consiste nos processos de gerenciamento de projetos: Planejar compras e aquisições, Planejar contratações, Solicitar respostas de fornecedores, Selecionar fornecedores, Administração de contrato e Encerramento do contrato.

Maiores informações sobre o PMBOK podem ser encontradas em PMI (2004).

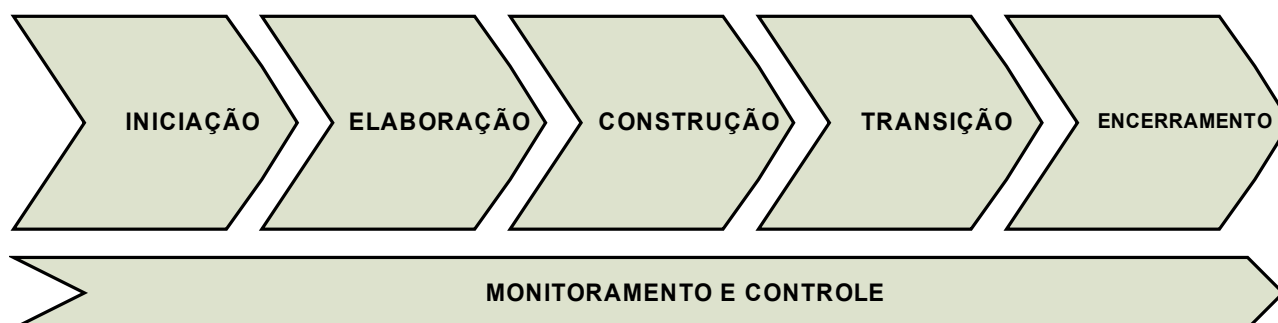
2 Organização da Metodologia

A Metodologia de Desenvolvimento de Sistemas da ANTAQ é estruturada de maneira a contemplar tanto a visão do produto a ser construído (Sistema de Informação), herdando

características do Processo Unificado e das metodologias que dele derivam, quanto a visão de projeto, agregando características do Corpo de Conhecimento em Gerenciamento de Projetos (PMBOK). Esta junção de visões é importante, pois as metodologias tradicionais de desenvolvimento de sistemas não contemplam aspectos importantes de planejamento e controle dos projetos, disciplinas abordadas no PMBOK.

A metodologia é organizada em cinco fases (Iniciação, Elaboração, Construção, Transição e Encerramento) e um processo contínuo de monitoramento e controle de tempo, escopo, recursos e qualidade.

O diagrama abaixo demonstra o encadeamento dos componentes da metodologia:



2.3 Fases

Cada fase terá seus artefatos produzidos (demonstrado no quadro 3.4), e o marco de conclusão das fases será o aceite formal do artefato “Termo de Fim da Fase”, dado pelo responsável da área requisitante e por um servidor da ANTAQ vinculado à STI ou suas coordenadorias.

2.3.1 Iniciação

A meta dominante da fase de iniciação é atingir o consenso entre todos os envolvidos sobre os objetivos do ciclo de vida do projeto. Além disso, os produtos desta fase subsidiam a organização com informações sobre a viabilidade do projeto, permitindo a decisão sobre a continuidade ou não do mesmo. A fase de iniciação tem muita importância principalmente para os esforços dos desenvolvimentos novos, nos quais há muitos riscos de negócios e de requisitos que precisam ser tratados para que o projeto possa prosseguir. Para projetos que visam melhorias em um sistema existente, a fase de iniciação é mais rápida, mas ainda se concentra em assegurar que o projeto seja compensatório e que seja possível fazê-lo.

2.3.2 Elaboração

A meta da fase de elaboração é minimizar as incertezas quanto à arquitetura técnica da solução, criando uma base estável para o esforço da fase de construção. A arquitetura se desenvolve a partir de um exame dos requisitos funcionais e não funcionais e de uma avaliação dos riscos envolvidos no projeto. Nesta fase, incluem-se também atividades de identificação dos macroprocessos e processos de negócios, mapeamento de atividades, identificação de pontos de melhoria e redesenho dos processos de negócios.

2.3.3 Construção

O grande objetivo desta fase é o desenvolvimento completo dos componentes da solução. Além do desenvolvimento, incluem-se também atividades de testes, controle e garantia da qualidade e a conclusão dos itens de requisitos e análise e design restantes.

2.3.4 Transição

A principal meta da Fase de Transição é assegurar que o software esteja disponível para seus usuários finais. A Fase de Transição tem início quando grande parte do software (mínimo em torno de 80%) já estiver com desenvolvimento concluído e pronta para ser implantada em ambiente de produção. Além da implantação do software em ambiente produtivo, outros objetivos desta fase incluem o treinamento ao Usuário e testes em ambiente de produção e/ou homologação (Testes Beta).

2.3.5 Encerramento

A meta da fase de Encerramento é formalizar a aceitação do produto, serviço ou resultado e conduzir o projeto ou uma fase do projeto a um final ordenado. Além da aceitação, outras atividades importantes desta fase são o registro de lições aprendidas durante o projeto e a mensuração final do sistema para registro na base histórica da ANTAQ.

2.3.6 Monitoramento e Controle

A função deste processo contínuo é medir e monitorar regularmente o progresso para identificar variações relativas a escopo, tempo, recursos e qualidade, em relação ao plano do projeto, de forma que possam ser tomadas ações corretivas quando necessário para atender aos objetivos do projeto.

2.4 Disciplinas

2.4.1 Modelagem de Negócios

O objetivo desta disciplina é estabelecer um canal de comunicação e entendimento entre a engenharia de negócios e a engenharia de software. Entender o negócio significa:

- entender a estrutura e a dinâmica dos processos de negócio da organização/área que está sendo analisada (o cliente);
- levantar os problemas (pontos de melhorias), propor alternativas de solução e demonstrar claramente os resultados esperados para cada ação de mudança com redesenho dos processos.
- estabelecer um entendimento comum do negócio entre o cliente, usuários do sistema e desenvolvedores.

2.4.2 Requisitos

O objetivo desta disciplina é descrever o que o sistema de informação deve fazer, ou seja, suas funcionalidades e necessidades não-funcionais que devem ser contempladas para o sucesso do projeto. Esta descrição deve ser acordada entre usuários e desenvolvedores.

2.4.3 Análise e Design

Esta disciplina tem como objetivo converter os requisitos de negócio em um projeto técnico para o sistema a ser construído. A arquitetura necessária para contemplar a solução é criada e adaptada para que corresponda ao ambiente de implementação, projetando-o para fins de desempenho.

2.4.4 Implementação

A principal finalidade da implementação é definir a organização do código em termos de subsistemas de implementação organizados em camadas implementar classes e objetos em termos de componentes (arquivos-fonte, binários, executáveis e outros). Os testes unitários são também alvo desta disciplina.

2.4.5 Teste

A principal finalidade da disciplina de Teste é localizar e expor os pontos fracos do software. Este objetivo é cumprido através da localização e documentação de defeitos na qualidade do software. O conceito de qualidade de software é definido como a conformidade do produto desenvolvido com os requisitos do cliente. As suposições feitas nas especificações técnicas de da disciplina de Análise e Design, e nas especificações negociais providas pela disciplina de requisitos, são validadas ou refutadas.

2.4.6 Implantação

A Disciplina Implantação descreve as atividades que garantem que o produto de software será disponibilizado a seus usuários finais.

2.4.7 Gerência de Projetos

O objetivo desta disciplina é organizar e gerenciar recursos, sejam eles humanos ou não, de forma a garantir que o projeto seja completo com qualidade, dentro das restrições de escopo, tempo e custos.

2.5 Papéis

Os Papéis descrevem como as pessoas se comportam no projeto e quais são as responsabilidades que elas têm. Os papéis não são pessoas, ou seja, um membro da equipe do projeto geralmente desempenha muitos papéis distintos. Da mesma maneira, um único papel pode ser exercido por um grupo de pessoas. Os papéis definidos para a presente metodologia são descritos a seguir.

2.5.1 Analista de Negócio

O papel de analista de processos é responsável por entender o negócio, mapear os processos atuais, identificar os pontos de melhorias e traçar um diagnóstico. Este diagnóstico serve como guia para o desenho de um novo e otimizado processo de negócio, que servirá de base para o novo sistema de informação a ser construído.

2.5.2 Analista de Sistemas

O papel analista de sistemas lidera e coordena a identificação de requisitos e a modelagem de casos de uso, delimitando o sistema e definindo sua funcionalidade; por exemplo, estabelecendo quais são os atores e casos de uso existentes e como eles interagem. Na metodologia da ANTAQ, o Analista de Sistemas é também responsável pela mensuração do sistema em pontos de função, ou na métrica que estiver em uso por determinação da ANTAQ.

2.5.3 Analista de Teste

O papel Analista de Teste é responsável por inicialmente identificar e posteriormente definir os testes necessários, monitorar a abrangência dos testes e avaliar a qualidade geral obtida ao testar os Itens de Teste-alvo. Este papel também envolve a especificação dos Dados de Teste necessários e a avaliação do resultado dos testes conduzidos em cada ciclo de teste.

2.5.4 Designer de Interface

O designer de interface de usuário, também conhecido como Web Designer, lidera e coordena a construção do protótipo e o design da interface do usuário da seguinte forma:

- capturando os requisitos da interface do usuário, incluindo requisitos de usabilidade
- construindo protótipos de interface do usuário
- incluindo outros envolvidos da interface de usuário, como usuários, nas revisões de usabilidade e nas sessões de teste de uso
- revisando e fornecendo o feedback apropriado sobre a implementação final da interface do usuário, se criada por outros desenvolvedores, ou seja, designers e implementadores.

2.5.5 Designer de Banco de Dados

O designer de banco de dados, também conhecido por DBA (Database Administrator) define tabelas, índices, visões, restrições, triggers, procedimentos armazenados, parâmetros de armazenamento, databases e outras construções específicas de um banco de dados necessárias para armazenar, recuperar e excluir objetos persistentes.

2.5.6 Implementador

O papel implementador, também conhecido por Desenvolvedor, é responsável por desenvolver e testar componentes de acordo com os padrões adotados para o projeto, para fins de integração com subsistemas maiores.

2.5.7 Testador

É responsável pelas atividades centrais do esforço de teste, que envolve conduzir os testes necessários e registrar os resultados desses testes. Isso inclui:

- Identificar a abordagem de implementação mais apropriada para um dado teste
- Implementar testes individuais
- Configurar e executar os testes
- Registrar os resultados e verificar a execução dos testes
- Analisar erros de execução e recuperar-se deles

2.5.8 Gerente de Projetos

O papel gerente de projeto aloca recursos, ajusta as prioridades, coordena interações com clientes e usuários e geralmente mantém a equipe do projeto concentrada na meta certa. O gerente de projeto também estabelece um conjunto de práticas que garantem a integridade e a qualidade dos artefatos do projeto.

2.5.9 Patrocinador

Este papel representa o nível mais alto da hierarquia da área de negócio e é responsável por definir as estratégias negociais adotadas durante o projeto. É ele quem define o embaixador

do projeto, dá suporte nas decisões de alto nível e resolve possíveis conflitos relacionados às soluções adotadas no projeto. É o responsável pela homologação do produtos entregues. É também conhecido por Requisitante.

2.5.10 Embaixador

O Embaixador é um membro da equipe da área de negócio que demandou o projeto de sistema de informação. O Embaixador funciona como uma extensão da equipe de desenvolvimento dentro da área de negócios. É a pessoa que deve ser acionada inicialmente no momento de dúvidas sobre requisitos, ou sobre decisões de design que melhor atenderão o cliente final do produto. O recurso que exerce este papel deve ter o desejo, autoridade, responsabilidade e conhecimento para certificar que o sistema que está sendo produzido atenderá os anseios da área demandante.

2.5.11 Administrador de Dados

O papel do Administrador de Dados (AD) é zelar pela integração de dados, conceitos corporativos (Empresa, Embarcações, Mercadorias, etc.), sistemas de informações e padrões de nomenclatura de bases de dados e seus objetos.

Com relação aos padrões de nomenclatura, o documento “Modelagem de Dados: Regras e Padrões” determina as regras e padrões a serem seguidas na construção de modelos físicos de bancos de dados.

2.5.12 Arquiteto de Software

O papel do Arquiteto de Software é zelar pela padronização da implementação dos sistemas, utilização de melhores práticas em desenvolvimento e integração de componentes de software entre os diversos sistemas da Agência.

2.5.13 Documentador

O papel de documentador atua como facilitador em reuniões, redigindo atas e requisitos. Porém, sua atuação mais destacada dentro da metodologia é a relativa à construção do Manual do Usuário, a partir dos insumos gerados pelo Analista de Sistema e Analista de Negócio.

2.5.14 Envolvido

O papel de Envolvido (*stakeholder*) é definido como qualquer pessoa que é afetada em termos materiais pelo resultado do projeto. Resolver um problema complexo com eficiência significa atender às necessidades de um grupo distinto de envolvidos. Normalmente, os envolvidos têm perspectivas distintas sobre o problema e as diferentes necessidades que devem ser atendidas pela solução. Muitos envolvidos são usuários do sistema. Outros envolvidos são apenas usuários indiretos do sistema ou são afetados somente pelos resultados dos negócios influenciados pelo sistema. Muitos são compradores econômicos ou defensores do sistema. Saber quem são os envolvidos e conhecer suas necessidades específicas são aspectos fundamentais para desenvolver uma solução eficaz.

2.5.15 Analista de Produção

A função do Analista de Produção é analisar e publicar em ambiente de Produção os pacotes de implementação enviados pela equipe de desenvolvimento.

2.6 Artefatos

A tabela abaixo relaciona os artefatos contidos na metodologia com as fases da do roteiro padrão.

Fase	Artefato
Iniciação	Cadeia de Valor Diagrama de Casos de Uso Especificação de Requisitos Não-Funcionais Glossário de Termos Técnicos Documento de Mensuração Cronograma Proposta de Sistema de Informação Termo de Fim de Fase Ata de Reunião
Elaboração	Cronograma Plano do Projeto Fluxos Atuais Relatório de Diagnóstico e Prognóstico Cadeia de Valor Fluxos Redesenhados Diagrama de Casos de Uso Regras de Negócio Especificação de Requisitos Não-Funcionais Especificação de Casos de Uso Modelo de Classes MER Glossário de Termos Técnicos Documento de Mensuração Termo de Fim de Fase Ata de Reunião
Construção	Cronograma Plano do Projeto Cadeia de Valor Fluxos redesenhados Diagrama de Casos de Uso Especificação de Casos de Uso Regras de Negócio Especificação de Requisitos Não-Funcionais

	Modelo de Classes MER Documento de Implantação Plano de Testes Evidência de Testes Documento de Mensuração Glossário de Termos Técnicos Termo de Fim de Fase Ata de Reunião
Transição	Cronograma Plano do Projeto Plano de Testes Formulário de Implantação MER Material para Treinamento Documento de Mensuração Evidência de Testes Manual do Usuário Termo de Fim de Fase Ata de Reunião
Encerramento	Documento de Mensuração Cronograma Plano do Projeto Termo de Fim de Projeto Ata de Reunião

A seguir, descreve-se os artefatos da metodologia. Os modelos (*templates*) dos mesmos seguem em documento anexo a este.

2.6.1 Glossário de Termos Técnicos

A finalidade deste documento é descrever os termos técnicos, acrônimos e abreviações, tanto da área de Tecnologia da Informação, quanto da área de Negócio, que são referenciados pelos demais artefatos do projeto do sistema.

2.6.2 Cronograma

O cronograma é a disposição gráfica do tempo que será gasto na realização de um projeto, de acordo com as atividades a serem cumpridas. Serve para auxiliar no gerenciamento e controle do trabalho, permitindo de forma rápida a visualização de seu andamento.

2.6.3 Proposta de Sistema de Informação

A finalidade deste documento é atingir o consenso entre todos os envolvidos sobre os objetivos do ciclo de vida do projeto. Outro objetivo importante é subsidiar a

organização com informações sobre a viabilidade do projeto, permitindo a decisão sobre a continuidade ou não do mesmo.

2.6.4 Documento de Mensuração

Este documento registra as contagens em Pontos de Função do sistema durante o ciclo de vida do projeto.

2.6.5 Plano do Projeto

A finalidade deste documento é explicitar em uma linguagem acessível para o usuário contratante as características relacionadas ao planejamento e ao escopo de necessidades atendidas pelo sistema. O documento fornece informações para o processo de aprovação do projeto, tendo características quase contratuais. Ele comunica os principais questionamentos relacionados ao projeto e funciona como um regulador com base no qual todas as decisões futuras deverão ser validadas.

2.6.6 Especificação de Casos de Uso

Os Casos de Uso representam as funcionalidades do sistema. Casos de Uso são tipicamente relacionados a "atores". Um ator é um humano ou entidade máquina que interage com o sistema para executar um significativo trabalho.

2.6.7 Diagrama de Casos de Uso

O Diagrama de Casos de Uso tem o objetivo de auxiliar a comunicação entre os analistas e o cliente descrevendo um cenário que mostra as funcionalidades do sistema do ponto de vista do usuário.

2.6.8 Regras de Negócio

Este documento visa centralizar todas as regras de negócio do sistema em um único artefato. As regras são descritas, ganham um código identificador, e os Casos de Uso onde são utilizadas. Além disso, é informado também a camada da solução onde a regra é aplicada.

2.6.9 Especificação de Requisitos Não-Funcionais

As Especificações Suplementares capturam os requisitos de sistema que não são capturados imediatamente nos casos de uso do modelo de casos de uso. Dentre tais requisitos estão requisitos legais e de regulamentação e padrões de aplicativo, requisitos de usabilidade, confiabilidade, desempenho e suportabilidade, e outros requisitos, como sistemas operacionais e ambientes, requisitos de compatibilidade e restrições de design.

2.6.10 Modelo de Classes

Este artefato descreve as classes do sistema. Classe é uma descrição de um conjunto de objetos que compartilham as mesmas responsabilidades, relacionamentos, operações, atributos e semântica.

2.6.11 MER

O Modelo Entidade-Relacionamento (MER) descreve a representação lógica e física dos dados persistentes no sistema. Também abrange qualquer comportamento definido no banco de dados, como procedimentos armazenados, triggers, restrições, etc.

2.6.12 Termo de Fim de Fase

Este relatório descreve todos os produtos produzidos durante uma fase da metodologia, para que o usuário contratante possa emitir o aceite de conclusão da fase.

2.6.13 Documento de Implantação

A finalidade deste documento é descrever os requisitos pertinentes à infraestrutura tecnológica necessária à implantação do sistema.

2.6.14 Plano de Testes

Este artefato inclui a definição das metas e dos objetivos dos testes no escopo da iteração (ou projeto), os itens-alvo, a abordagem adotada, os recursos necessários e os produtos que serão liberados.

2.6.15 Formulário de Implantação

As implantações de pacotes de software devem ser feitas através deste artefato. Diversas informações são contempladas como o responsável pela implantação, impacto em outros sistemas, tipo de manutenção e itens de configuração afetados.

2.6.16 Termo de Fim de Projeto

Este relatório descreve todos os produtos produzidos durante o projeto, para que o usuário contratante possa emitir o aceite de conclusão do mesmo.

2.6.17 Ata de Reunião

Este artefato registra informações gerais sobre a reunião, como local, horário e participantes, além de registrar as decisões tomadas durante o encontro e as pendências a serem resolvidas.

2.6.18 Evidências de Testes

Este artefato comprova a execução de testes unitários e integrados, através da reprodução de relatórios gerados por ferramentas de testes. Outra forma de comprovação pode ser a imagem das telas (*print screen*) do sistema respondendo aos testes executados pelos testadores.

2.6.19 Cadeia de Valor

Este artefato representa os macroprocessos, processos ou subprocessos de negócio e sua integração.

2.6.20 Fluxos atuais

Este artefato é o modelo gráfico que representa a visão detalhada do processo em uma sequência lógica e mostra sua integração com os demais processos da Cadeia de Valor Agregado. Representa os processos da organização exatamente como são realizados de maneira detalhada.

2.6.21 Fluxos Redesenhados

Este artefato é o modelo gráfico que representa o resultado do estudo realizado e as melhorias dos processos atuais onde foram utilizadas técnicas específicas para aperfeiçoar a forma de execução do trabalho.

2.6.22 Relatório de Diagnóstico e Prognóstico



Este artefato é o documento que contempla o mapeamento dos processos atuais, explicita os pontos de melhorias, propõem alternativas de solução e demonstra claramente os resultados esperados para cada ação de mudança. Este documento é insumo para o redesenho dos processos.

2.7 Roteiro Padrão


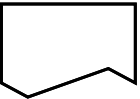


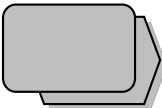

Para cada uma das fases da metodologia, é descrito um fluxo padrão de atividades. O encadeamento das atividades descrito no fluxo padrão não necessariamente precisa ser seguido, uma vez que cada projeto tem suas especificidades. Tal encadeamento deverá ser decidido pela equipe do projeto, através da disciplina de Gerenciamento de Projetos. Entretanto, todos os artefatos previstos devem ser gerados e validados pelo cliente.



A notação utilizada nos fluxos é a seguinte:

- Cores

Cor	Descrição
 Verde	Atividade / Artefato Opcional
 Vermelho	Atividade / Artefato Obrigatório

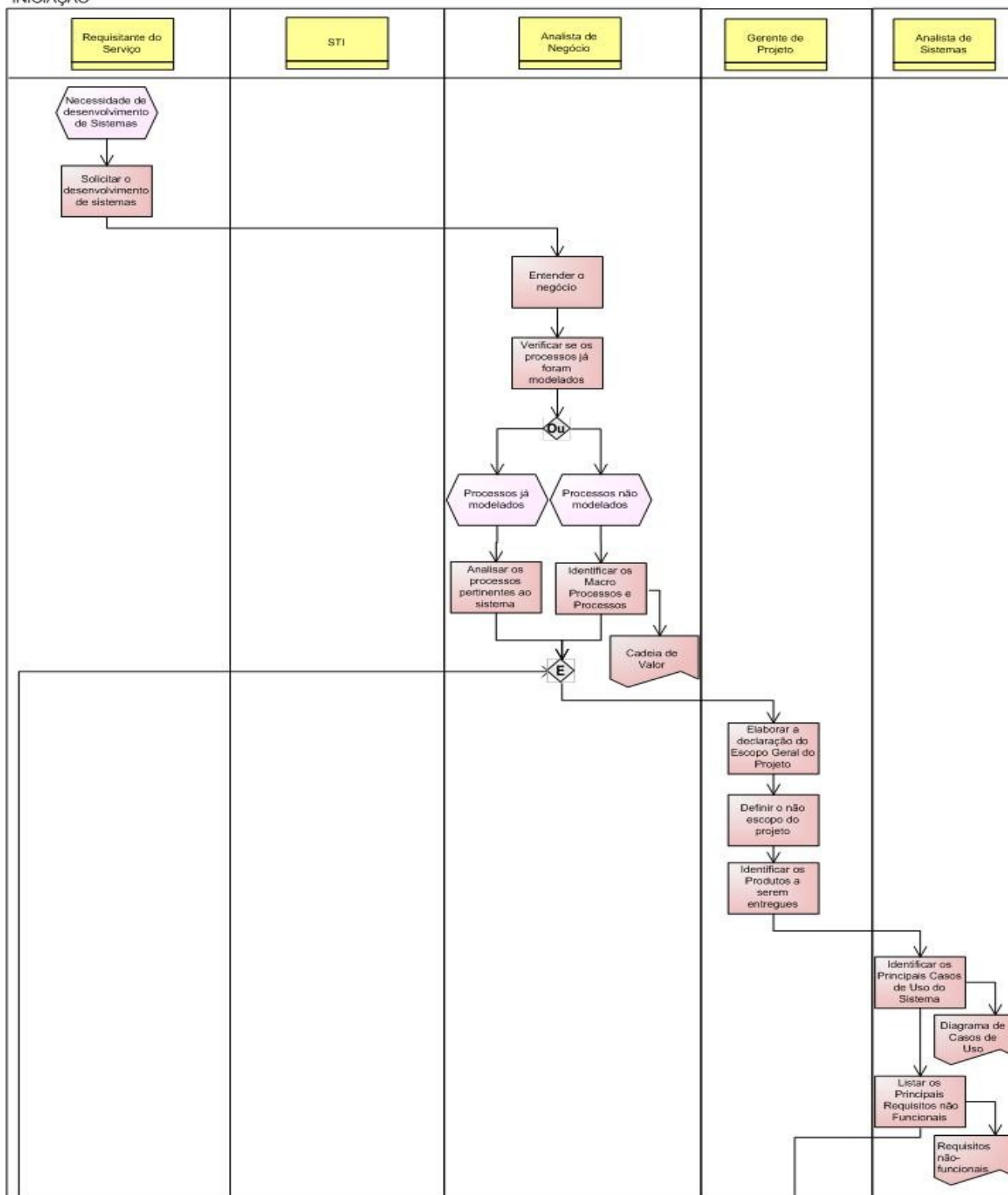
- Símbolos

Símbolo	Descrição
	Atividade
	Artefato
	Resultado de uma atividade ou preparação.
	Indicação de precedência
	Interface com outro fluxo de atividades.
	Conector “E”. Todos os segmentos de fluxo devem ser executados.

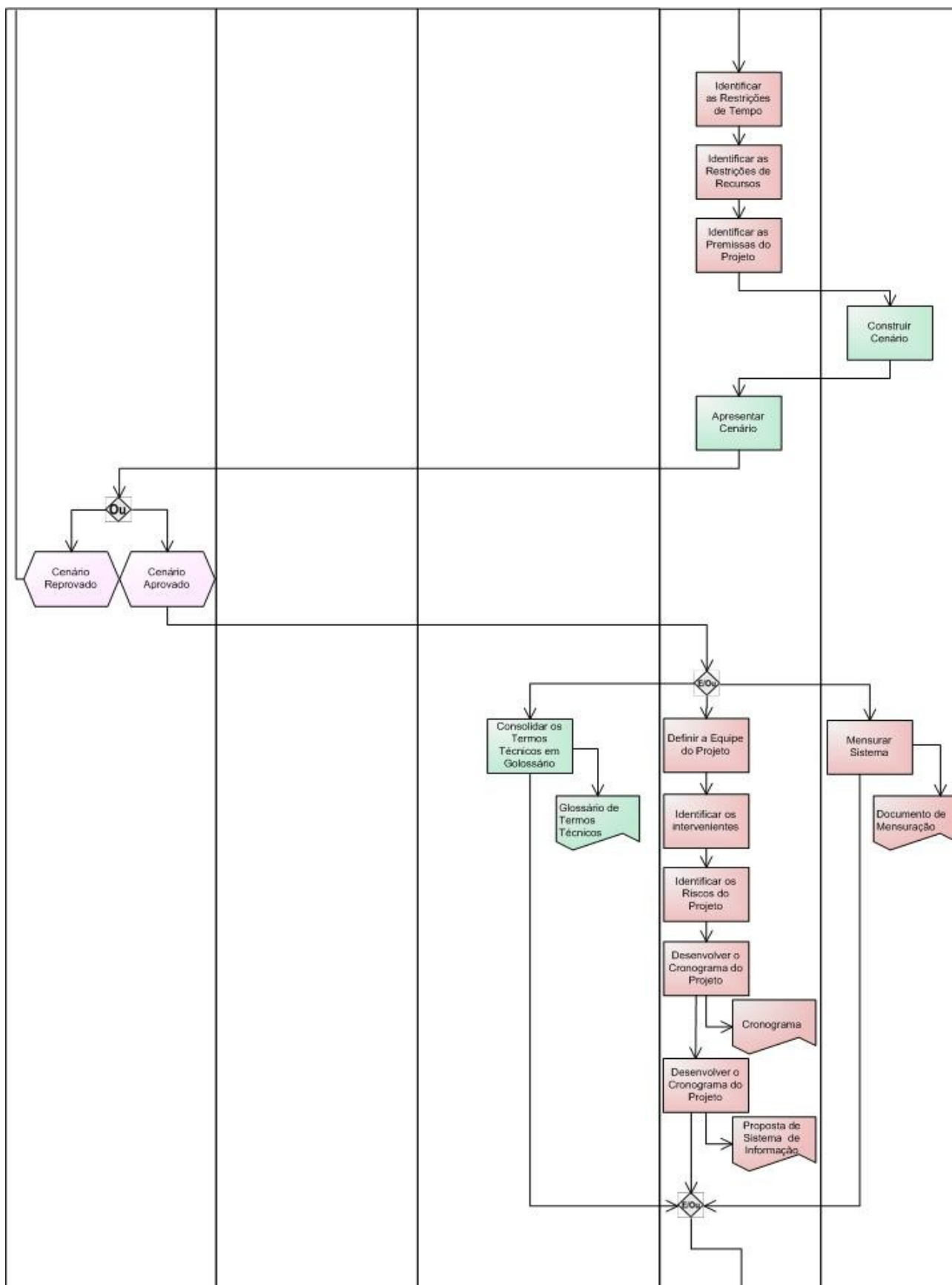
	<p>Conector “OU-Exclusivo”. Apenas um dos segmentos de fluxo deve ser executado.</p>
	<p>Conector “E/OU” Indica que pelo menos um segmento deve ser seguida, cada alternativa tem sua probabilidade de ocorrência.</p>

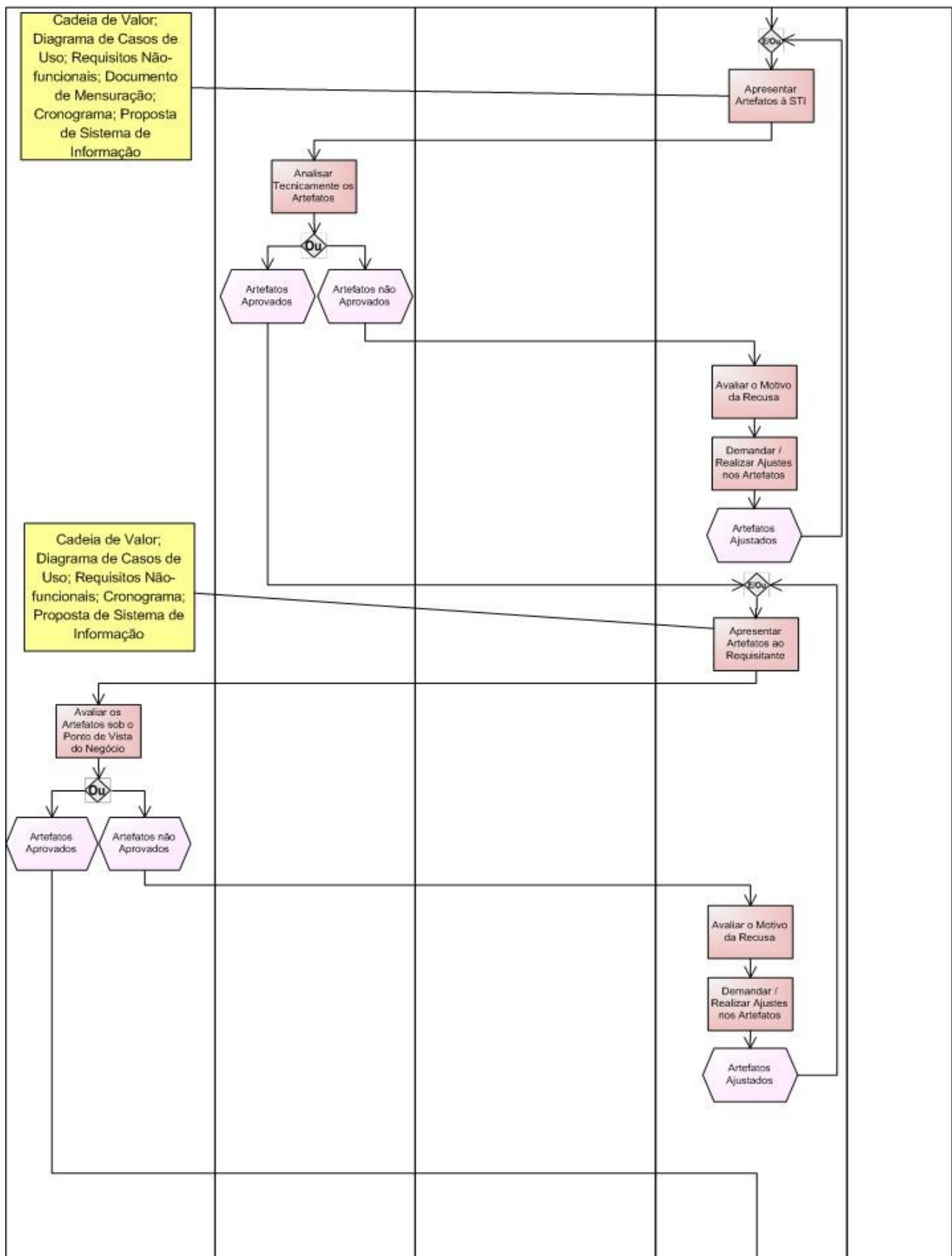
2.7.1 Iniciação

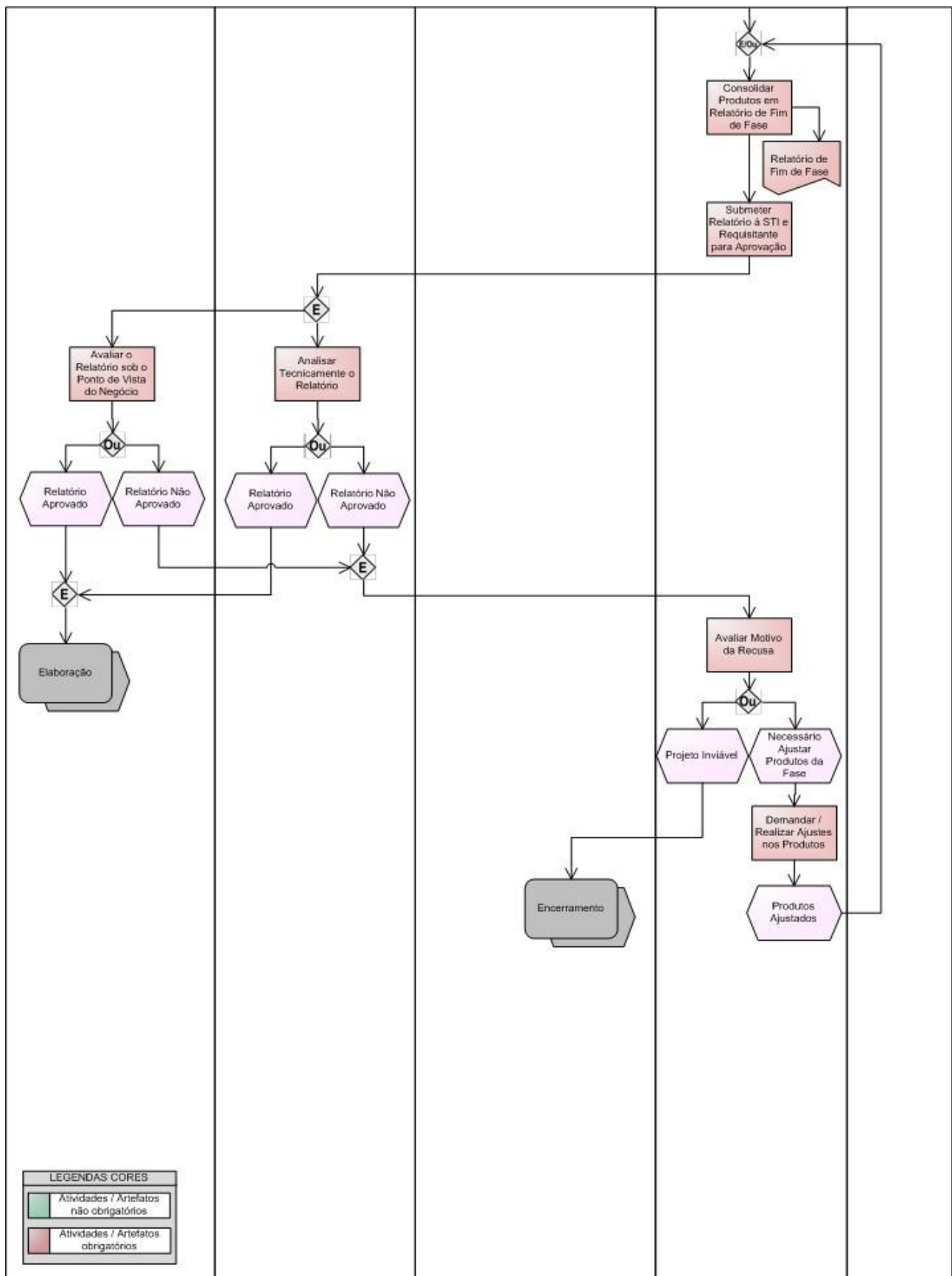
INICIAÇÃO



Fase Iniciação – Página 1A

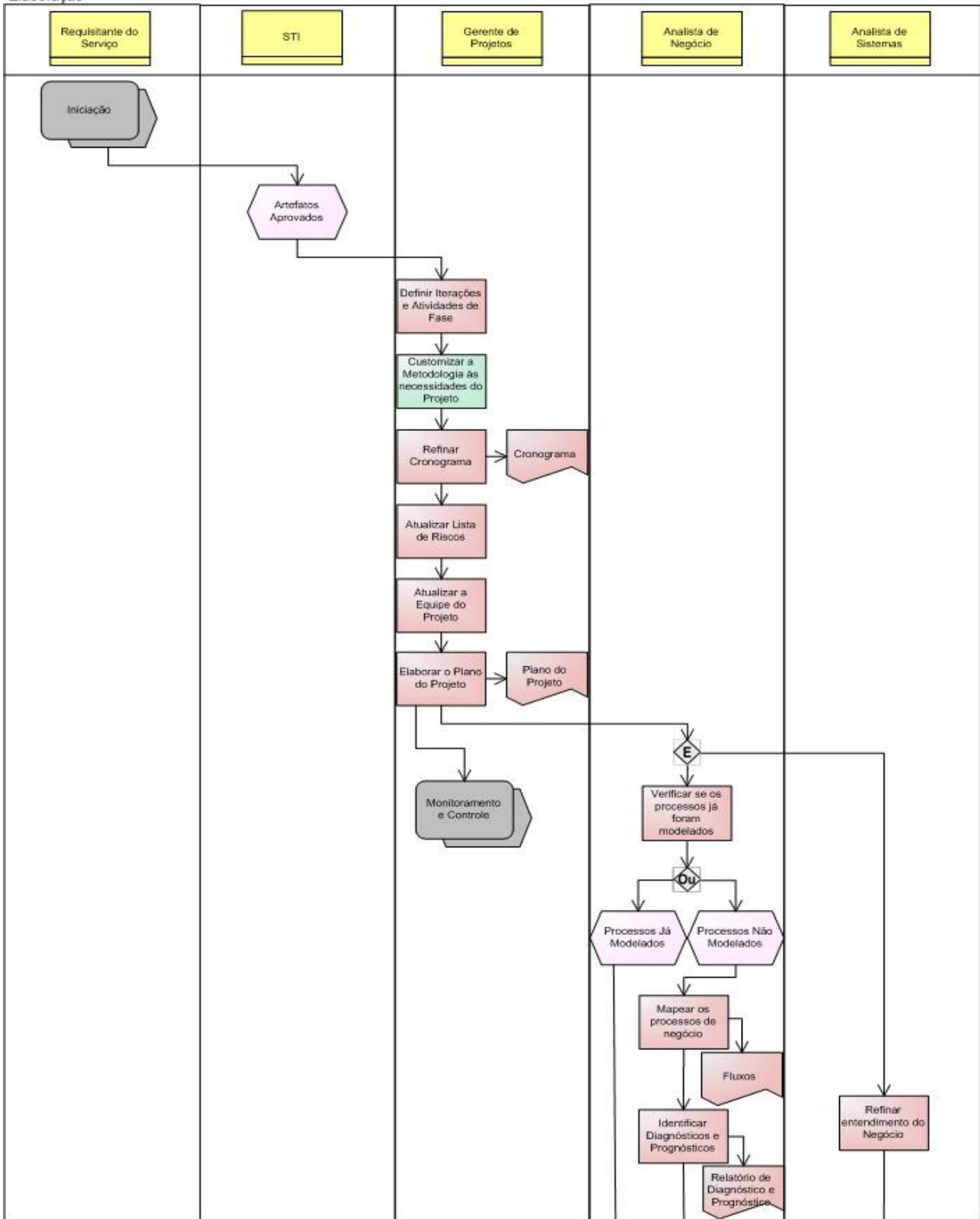






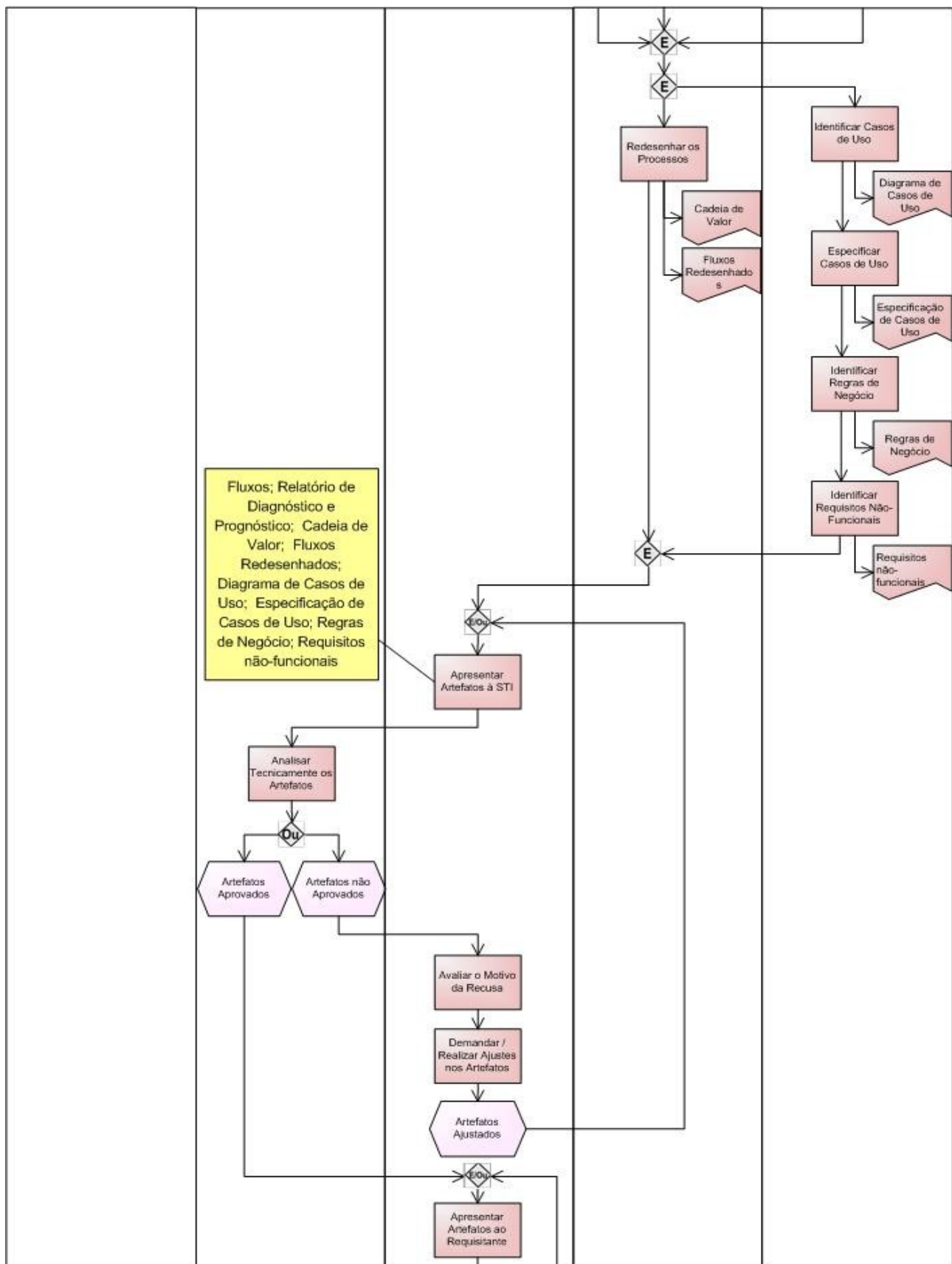
1.1.1 Elaboração

Elaboração

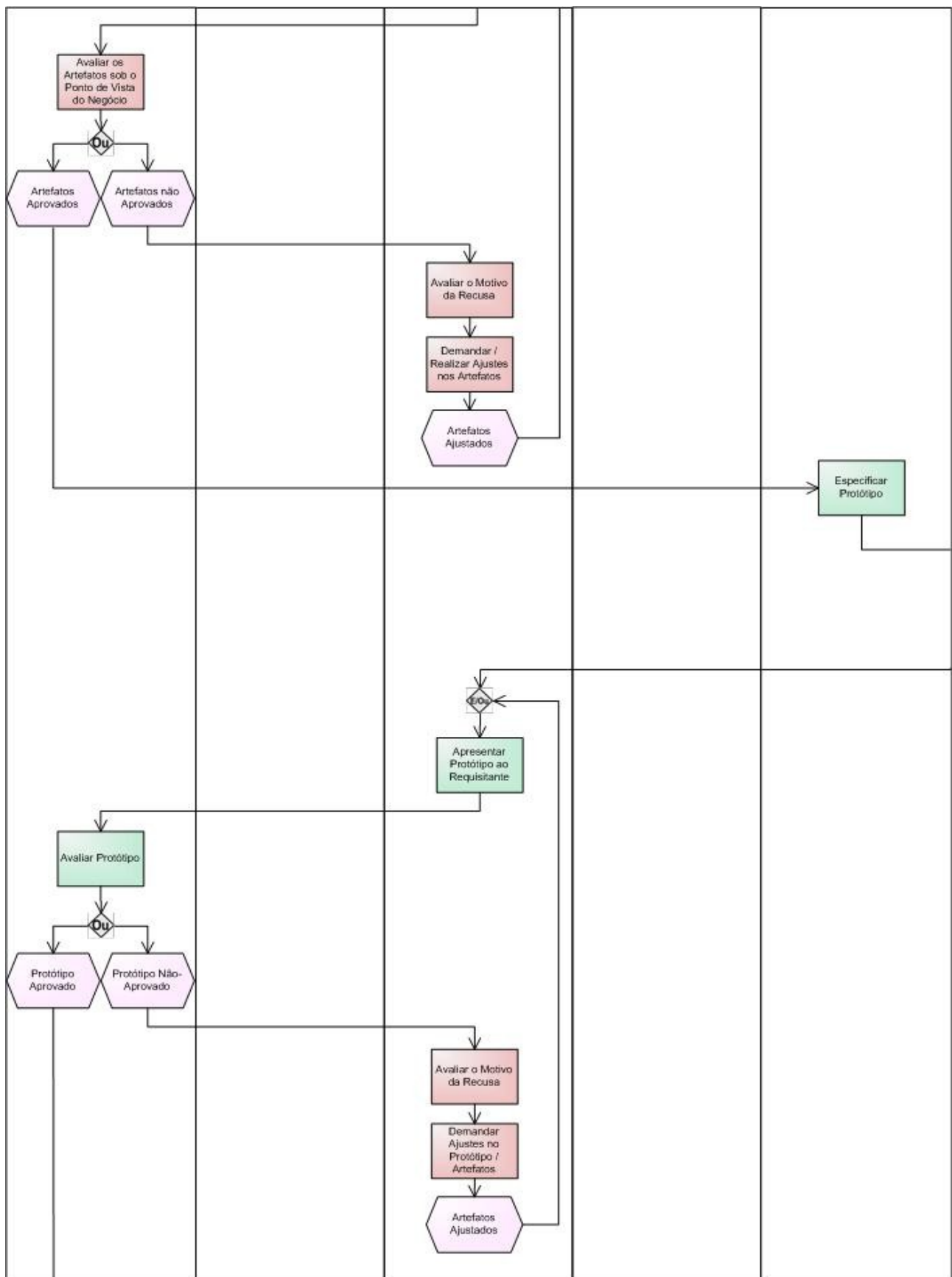


Fase Elaboração – Página 1A

Implementador	Arquiteto de Software	Administrador de Dados	DBA	



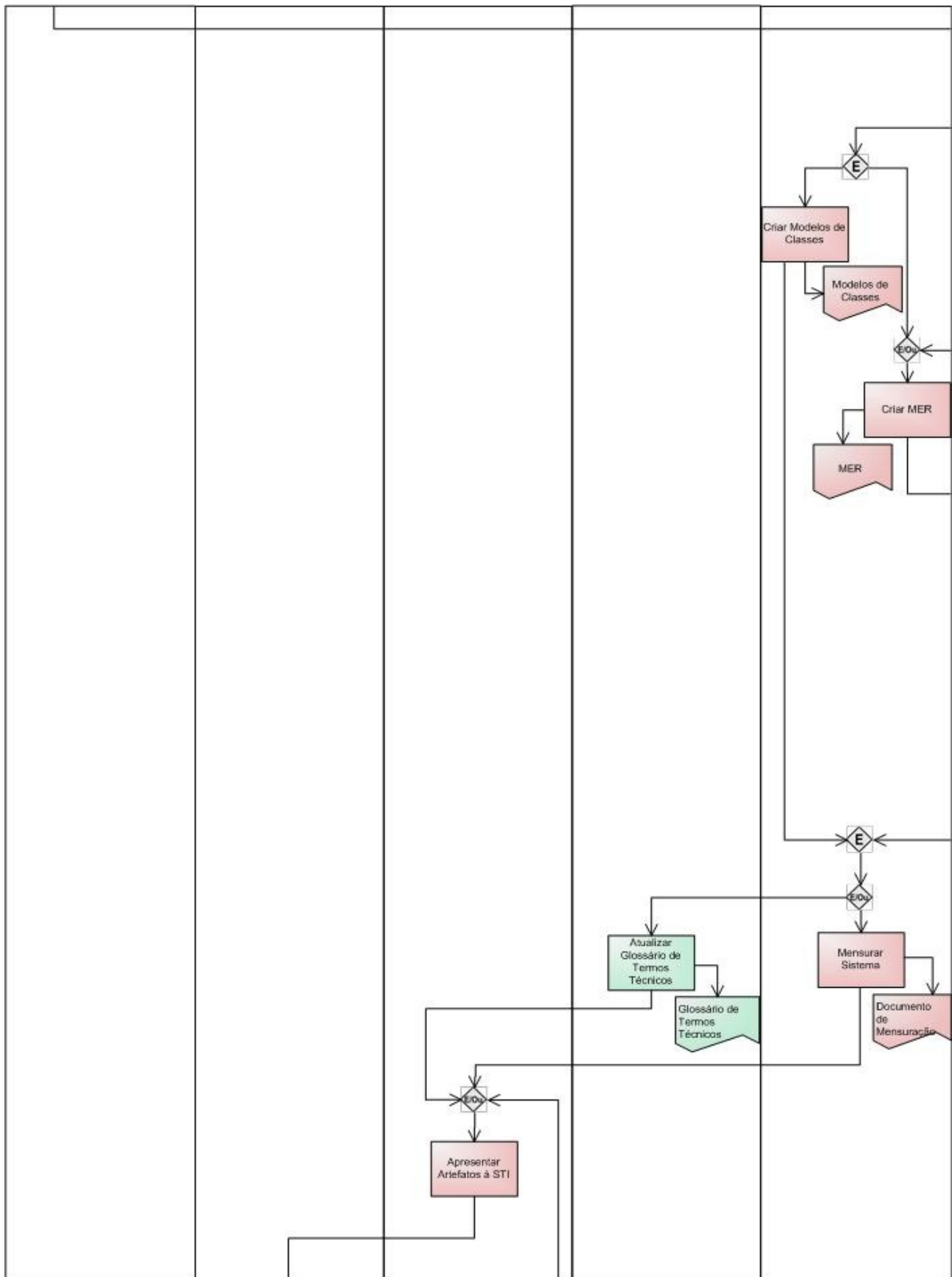
--	--	--	--	--

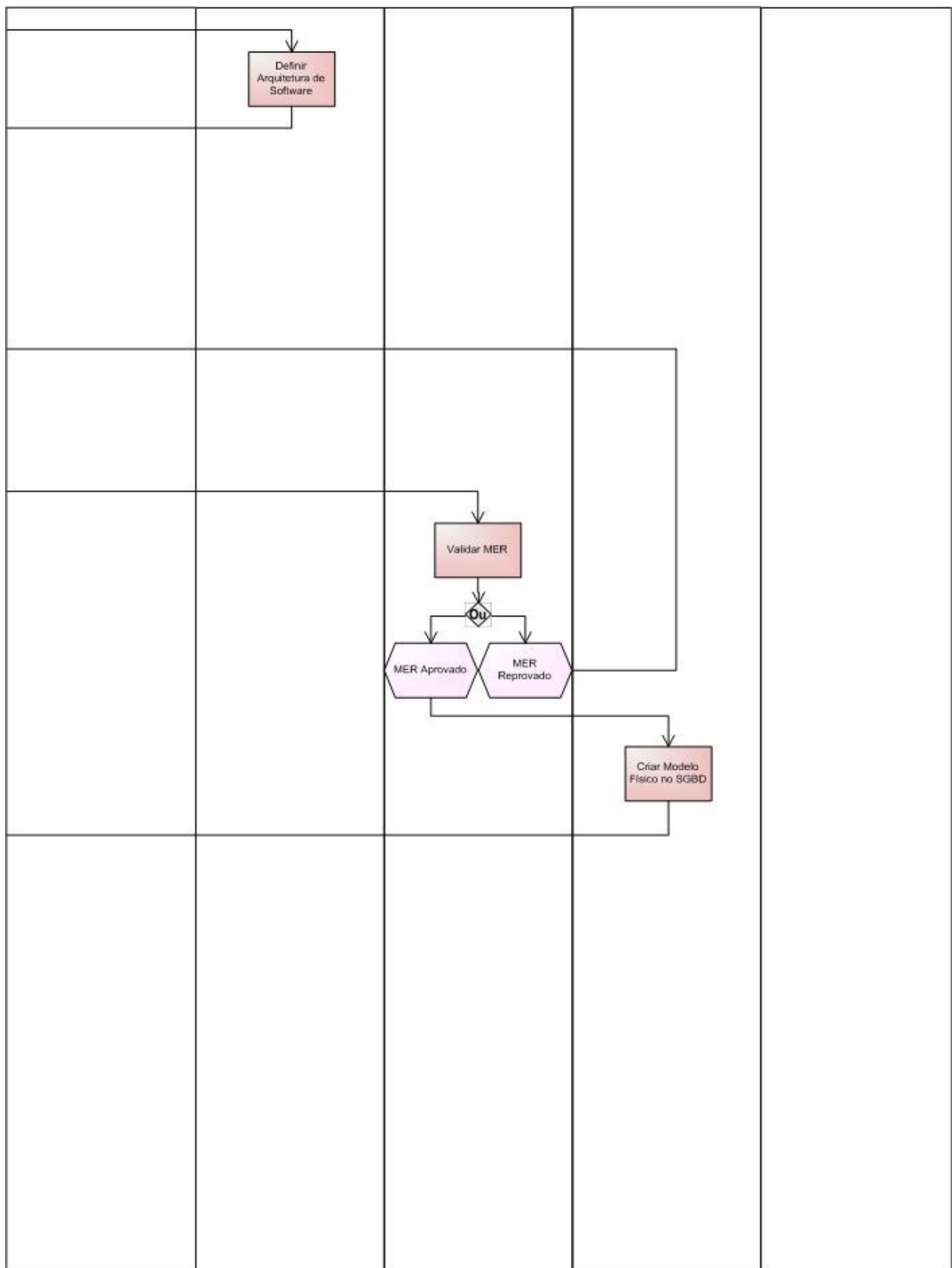


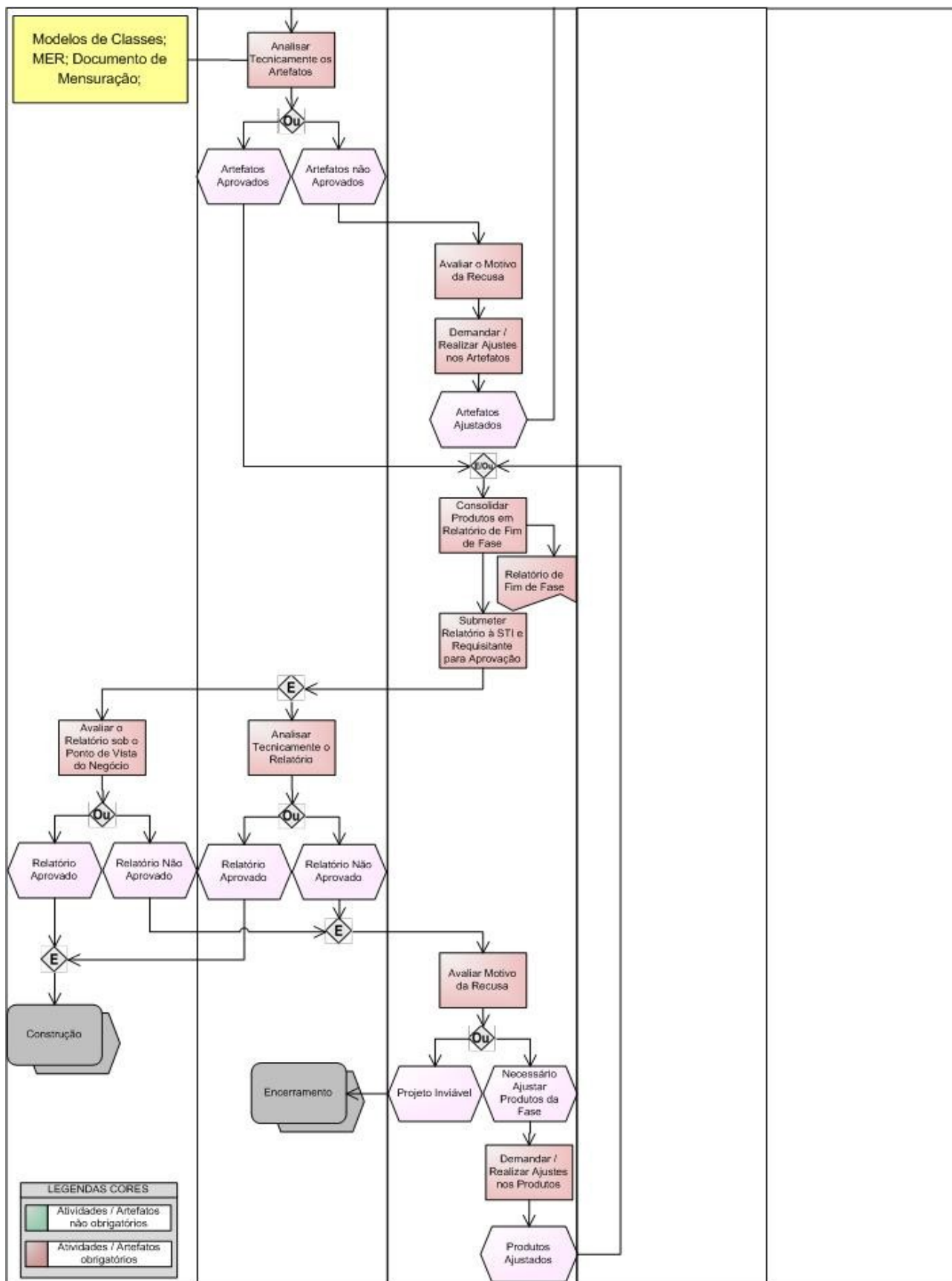
Fase Elaboração – Página 3A

--	--	--	--	--

Construir
Protótipo



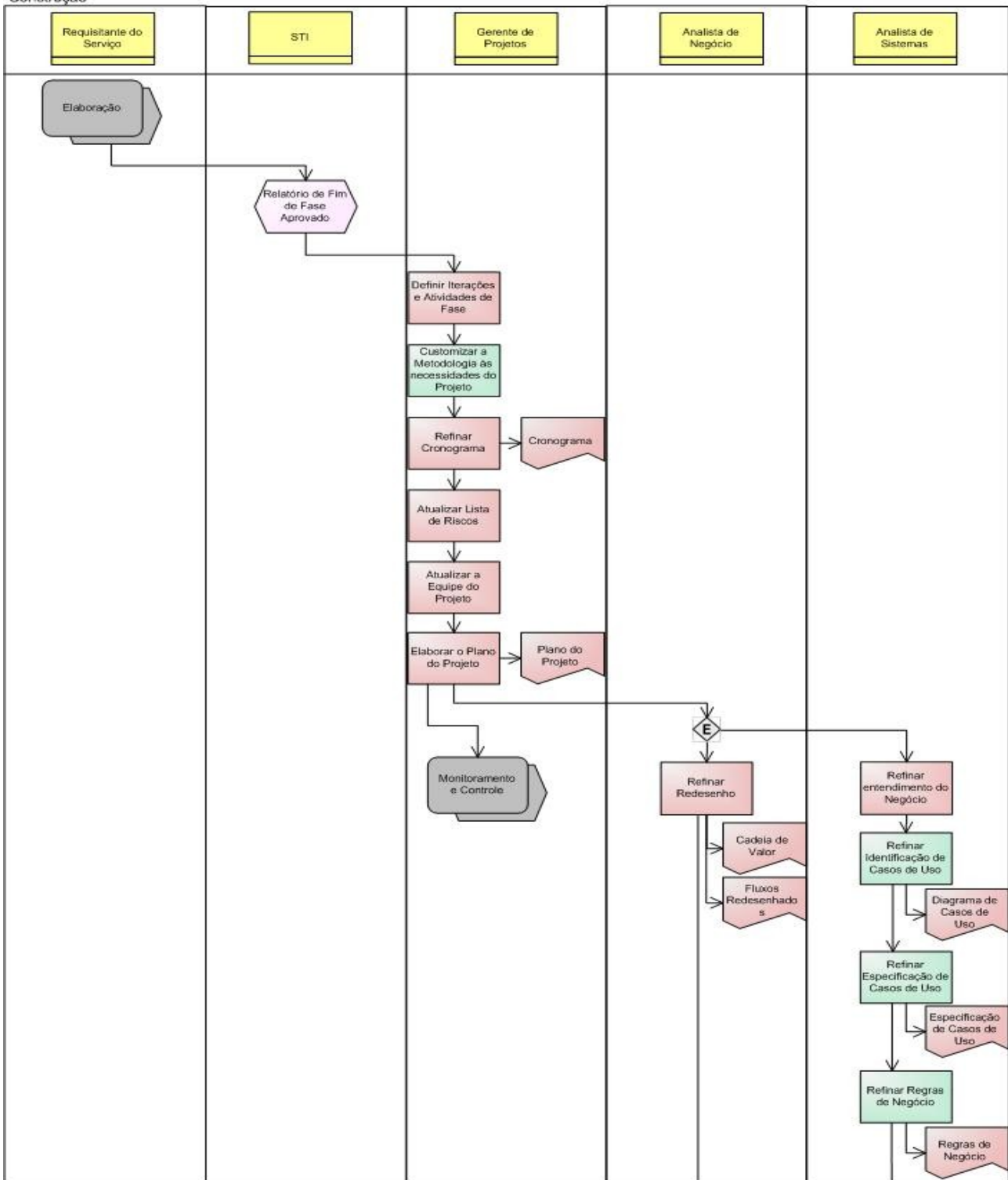




--	--	--	--	--

1.1.2 Construção

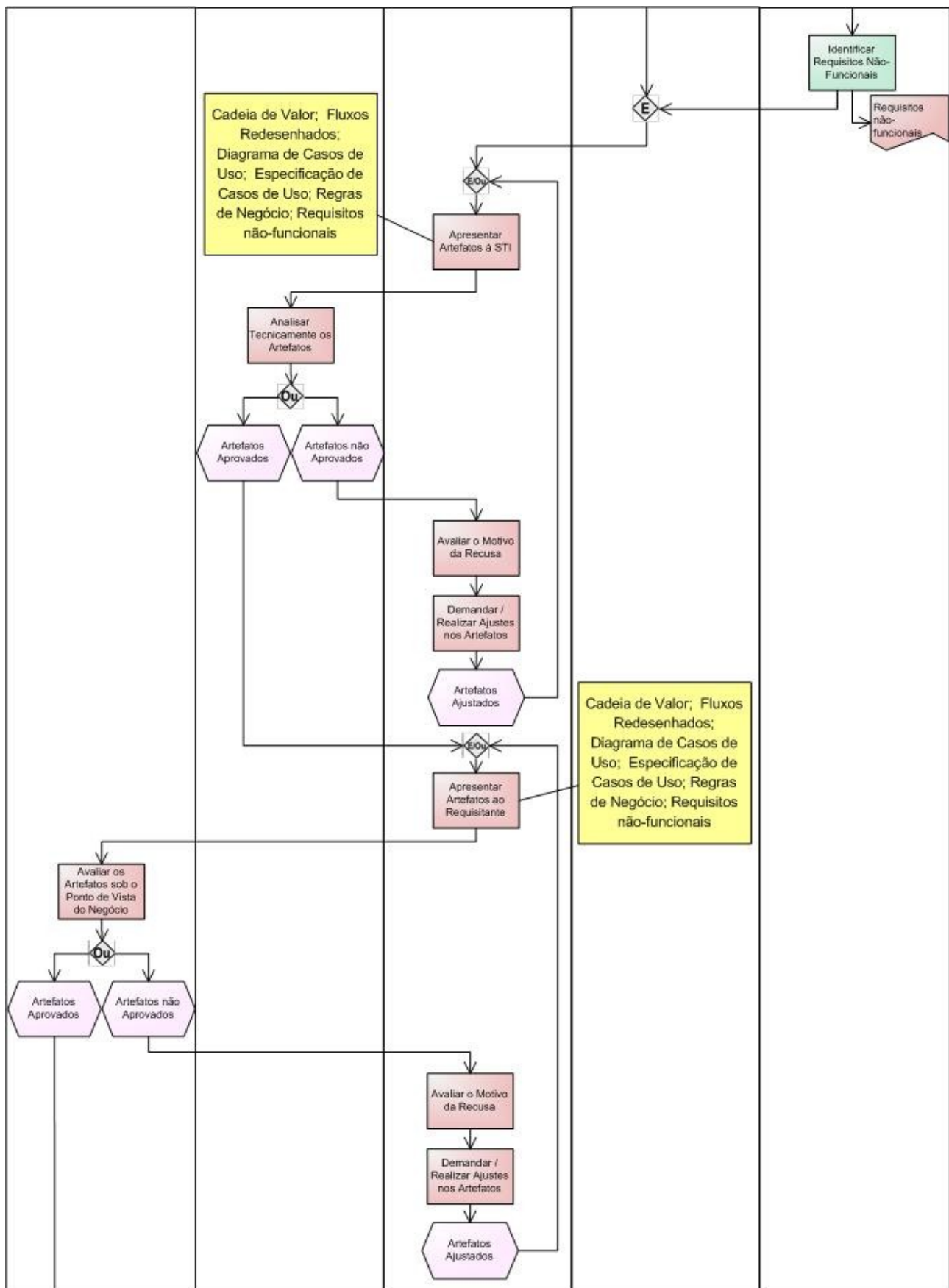
Construção



Fase Construção – Página 1A

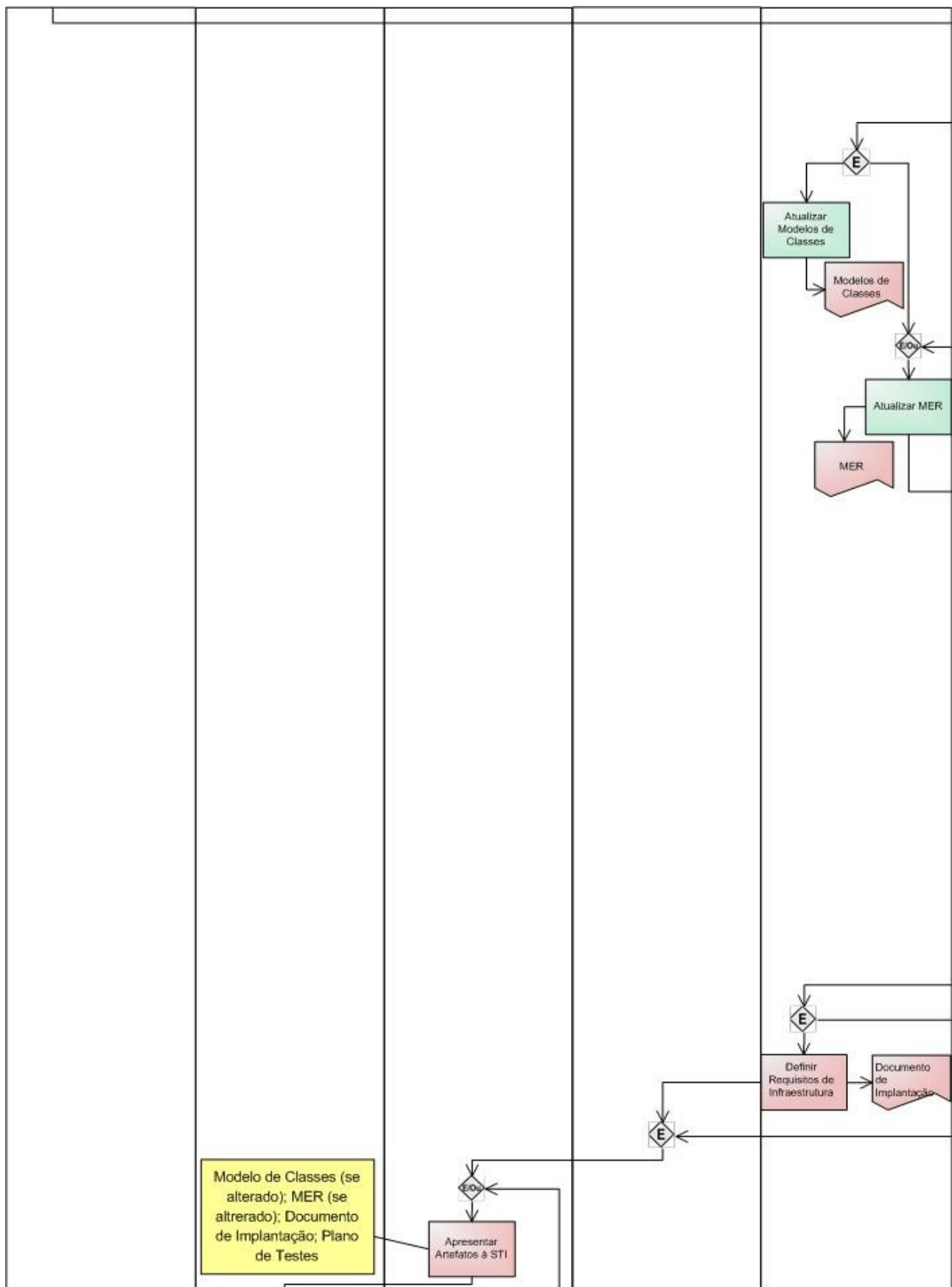
Implementador	Arquiteto de Software	Administrador de Dados	DBA	Analista de Testes

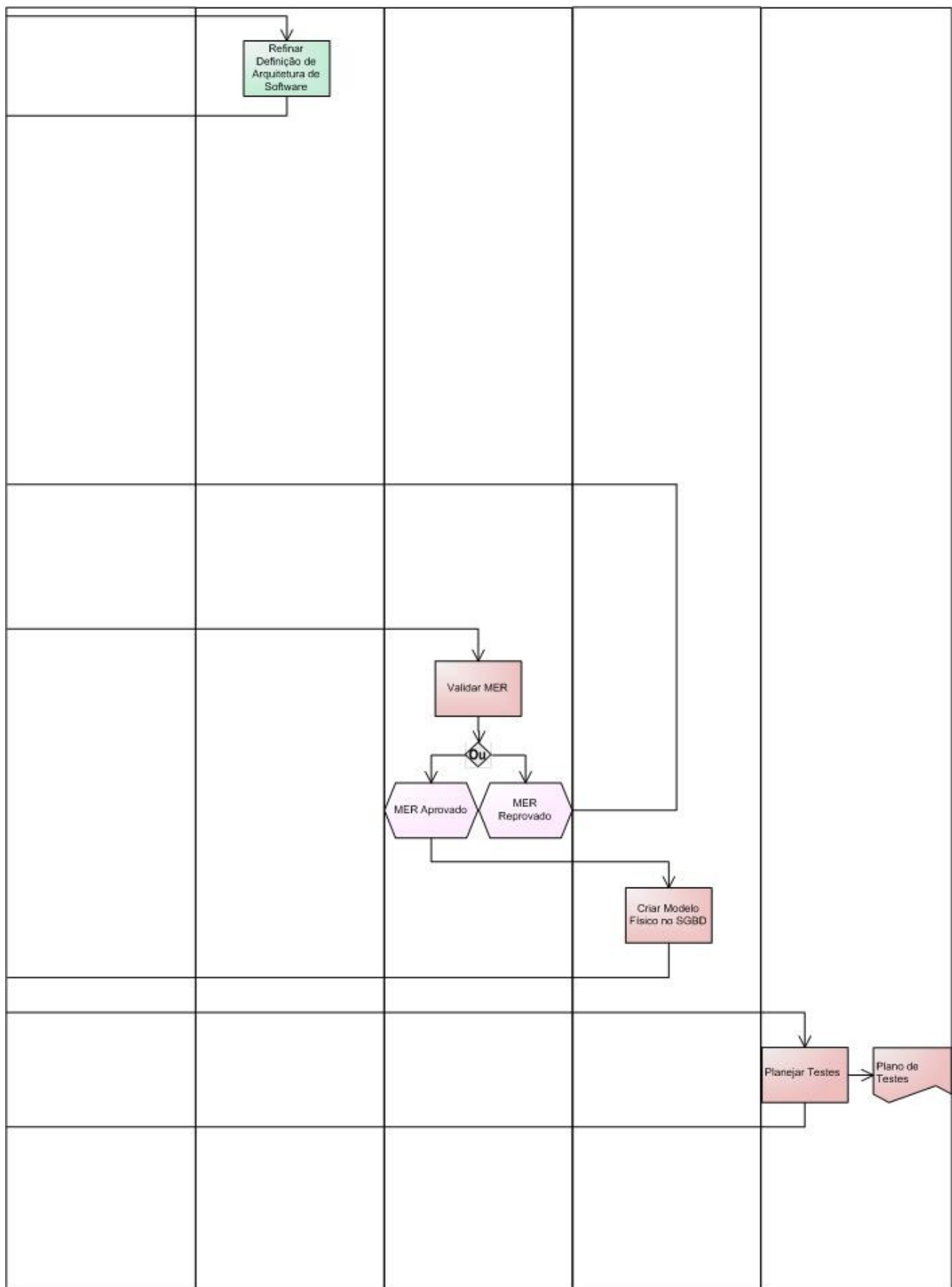
Designer de Interface	Testador			



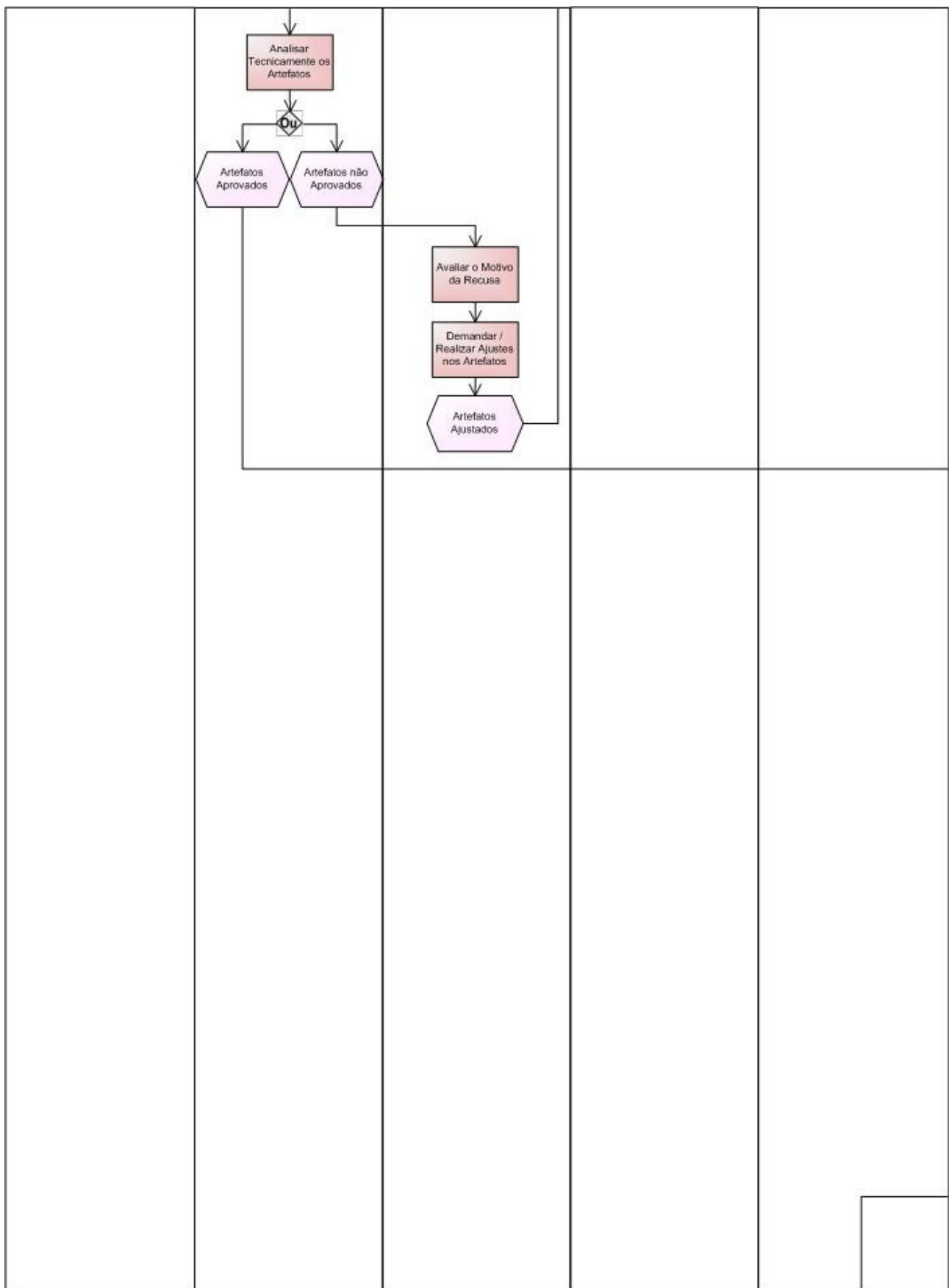
--	--	--	--	--

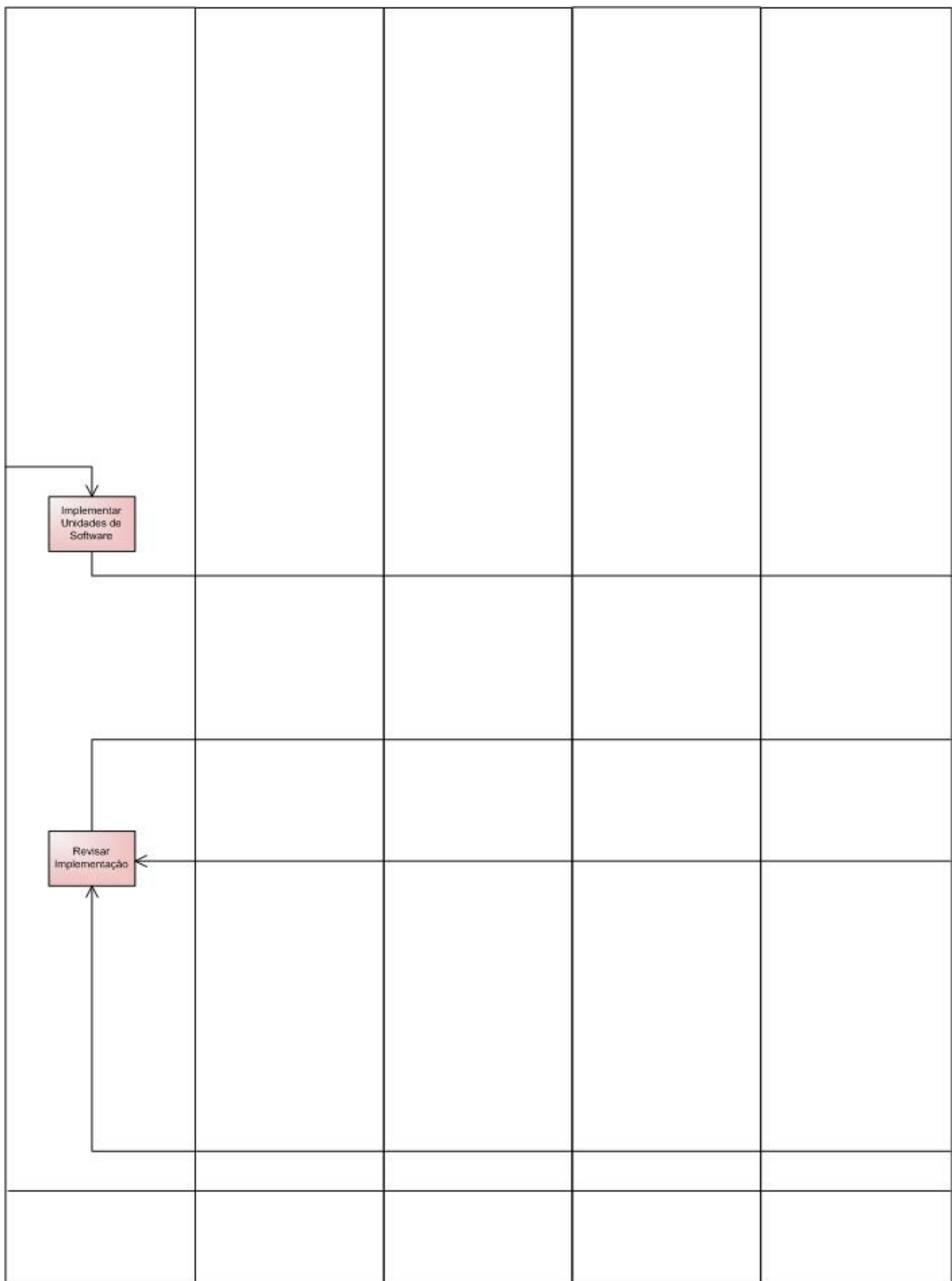
--	--	--	--	--

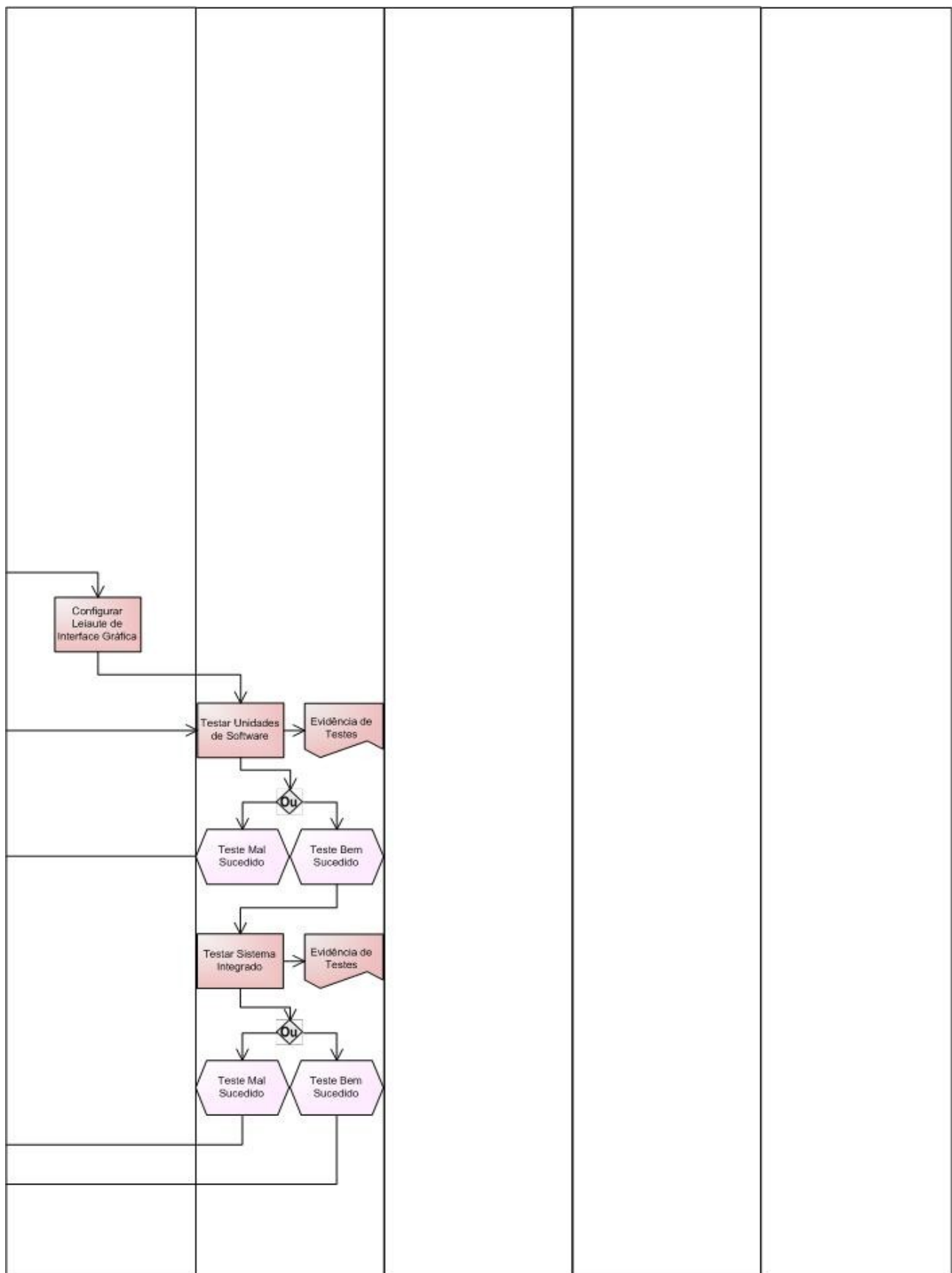


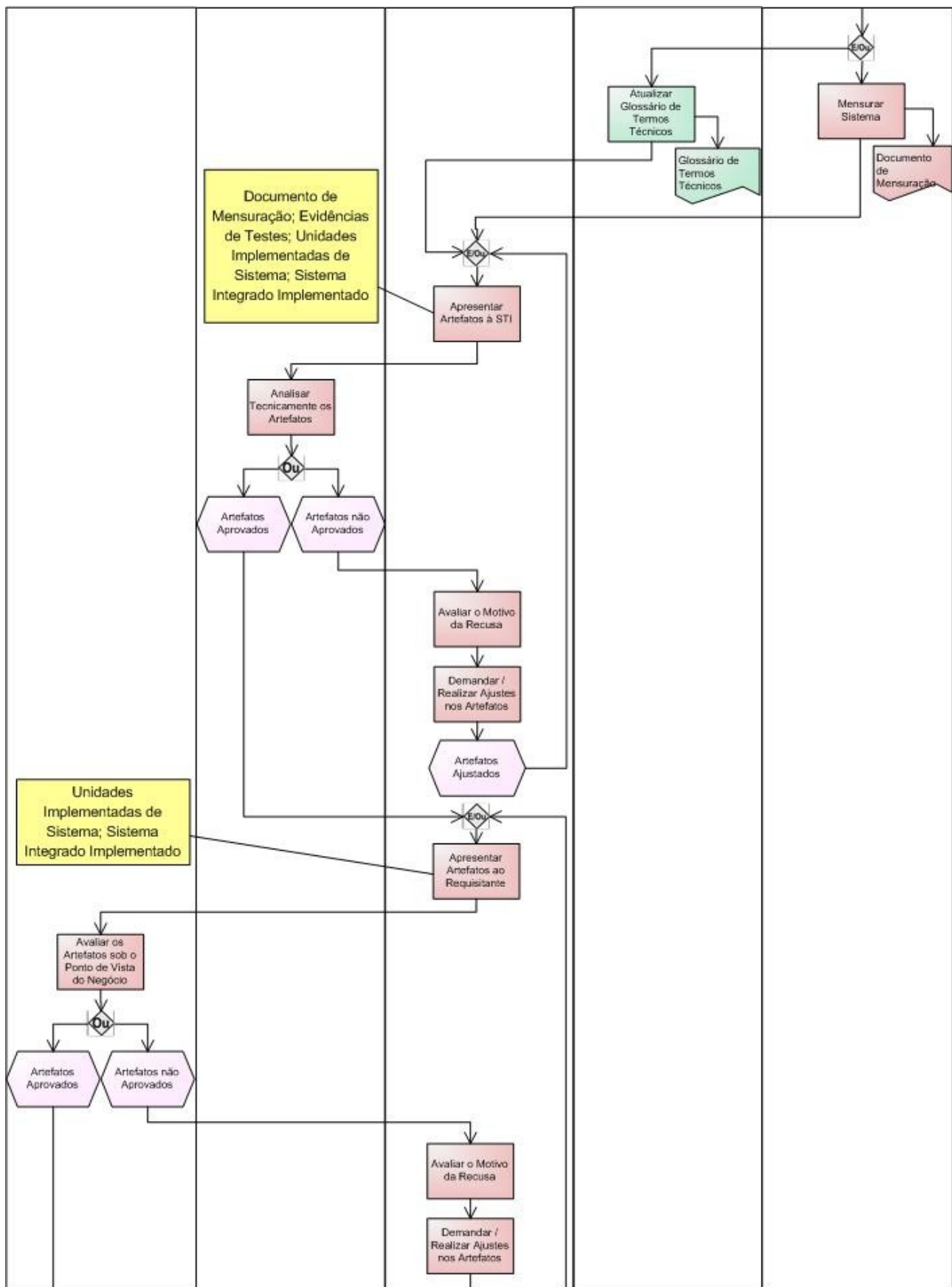


--	--	--	--	--



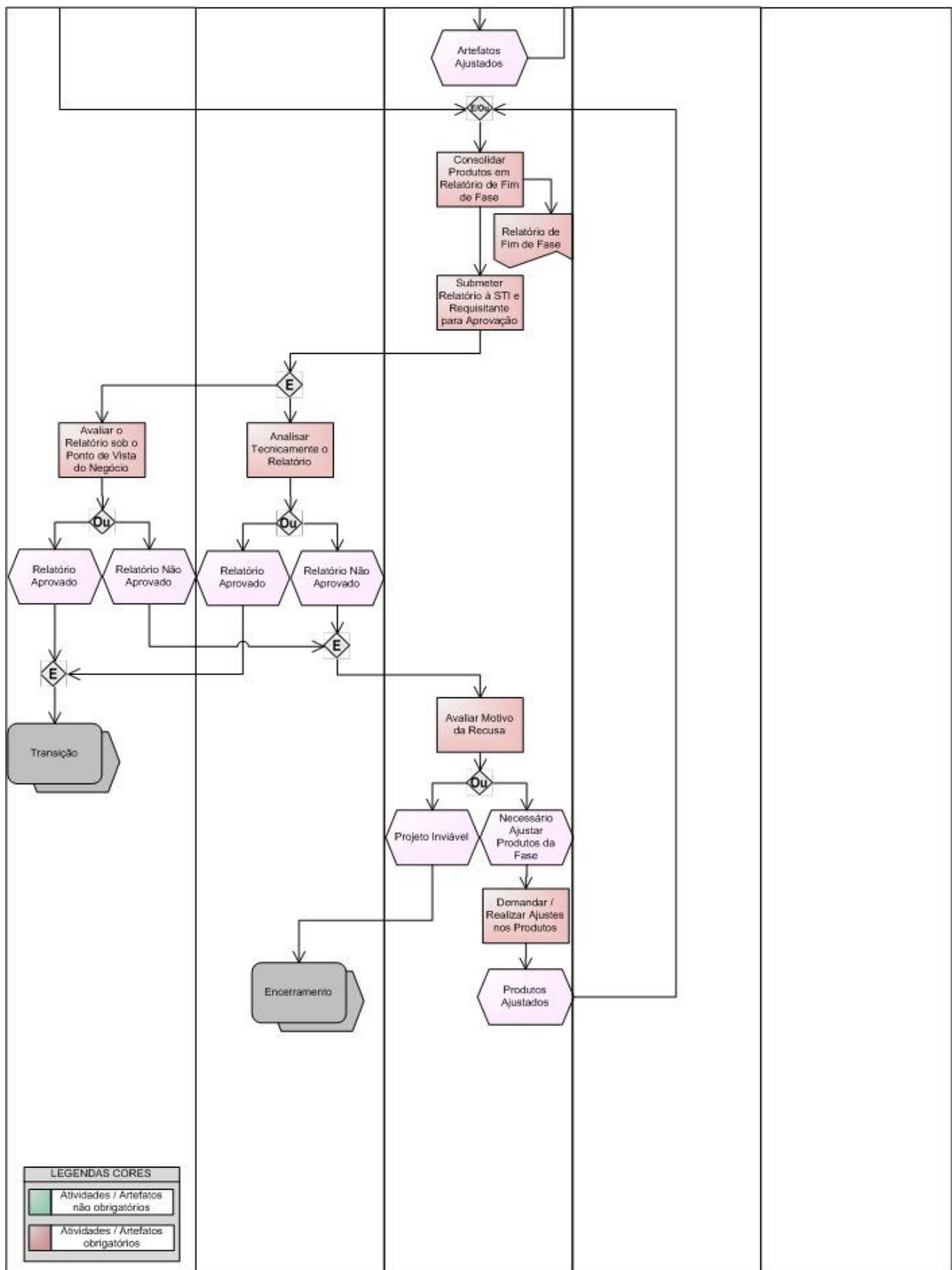






--	--	--	--	--

--	--	--	--	--

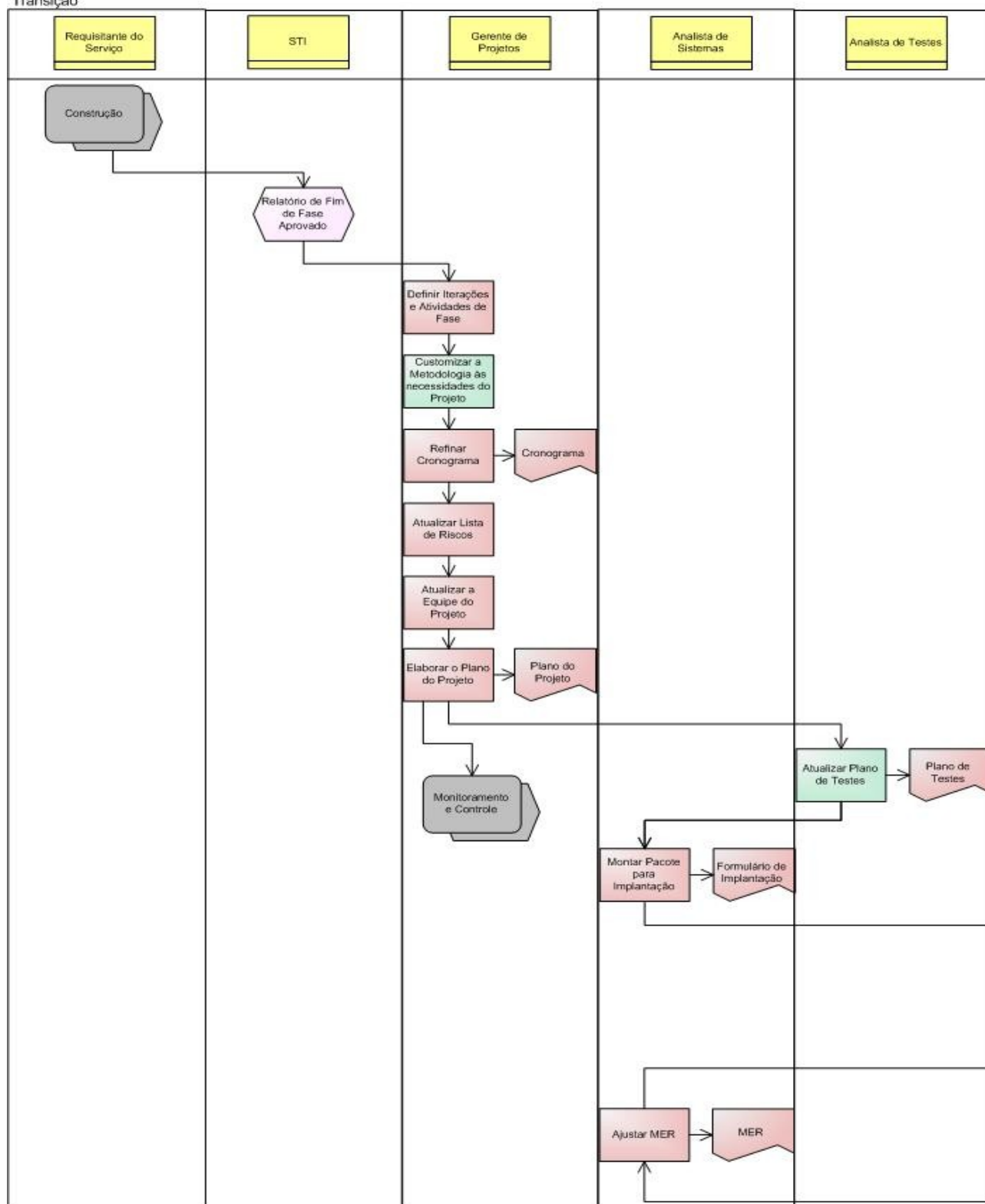


--	--	--	--	--

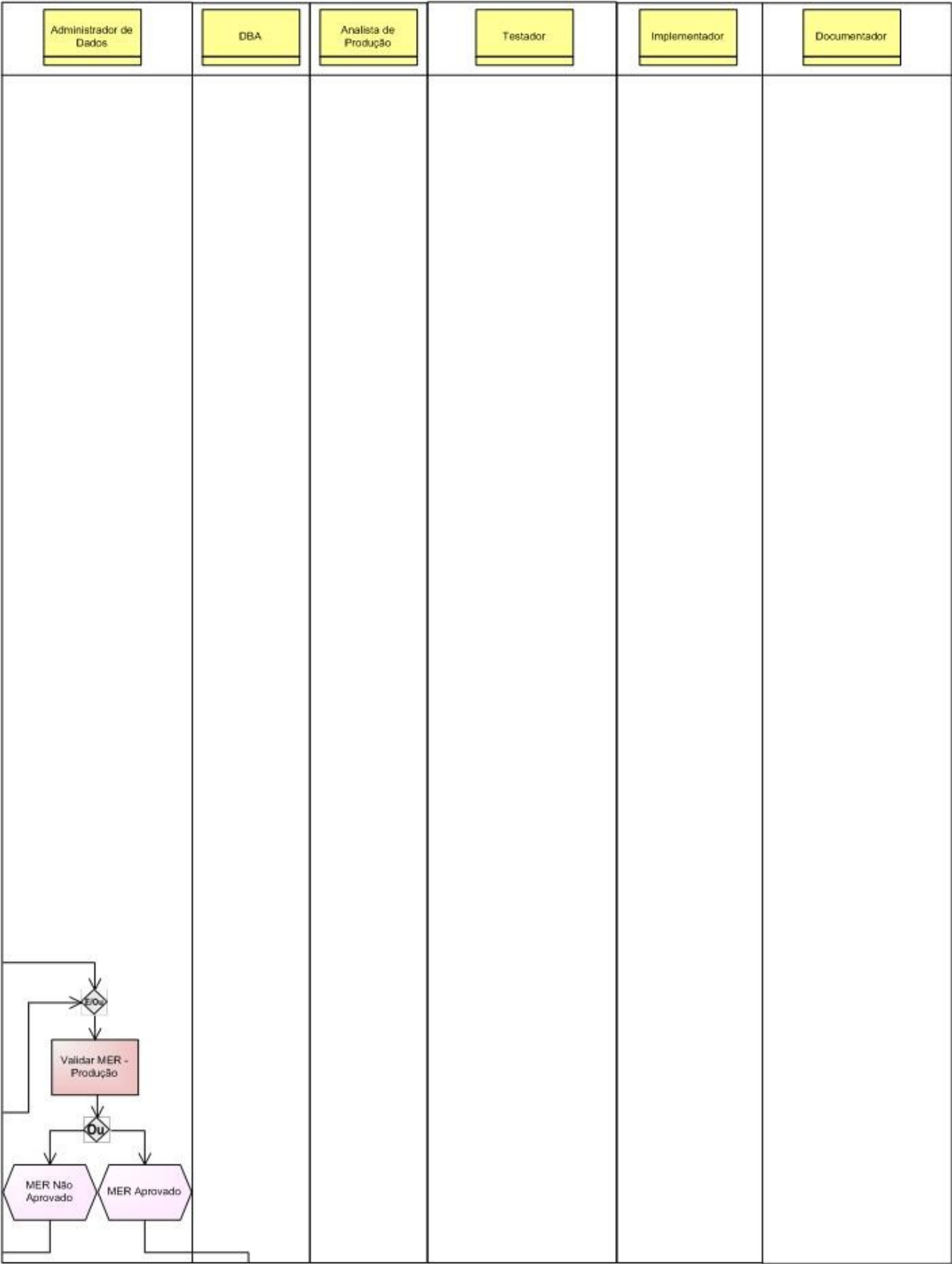
--	--	--	--	--

1.1.3 Transição

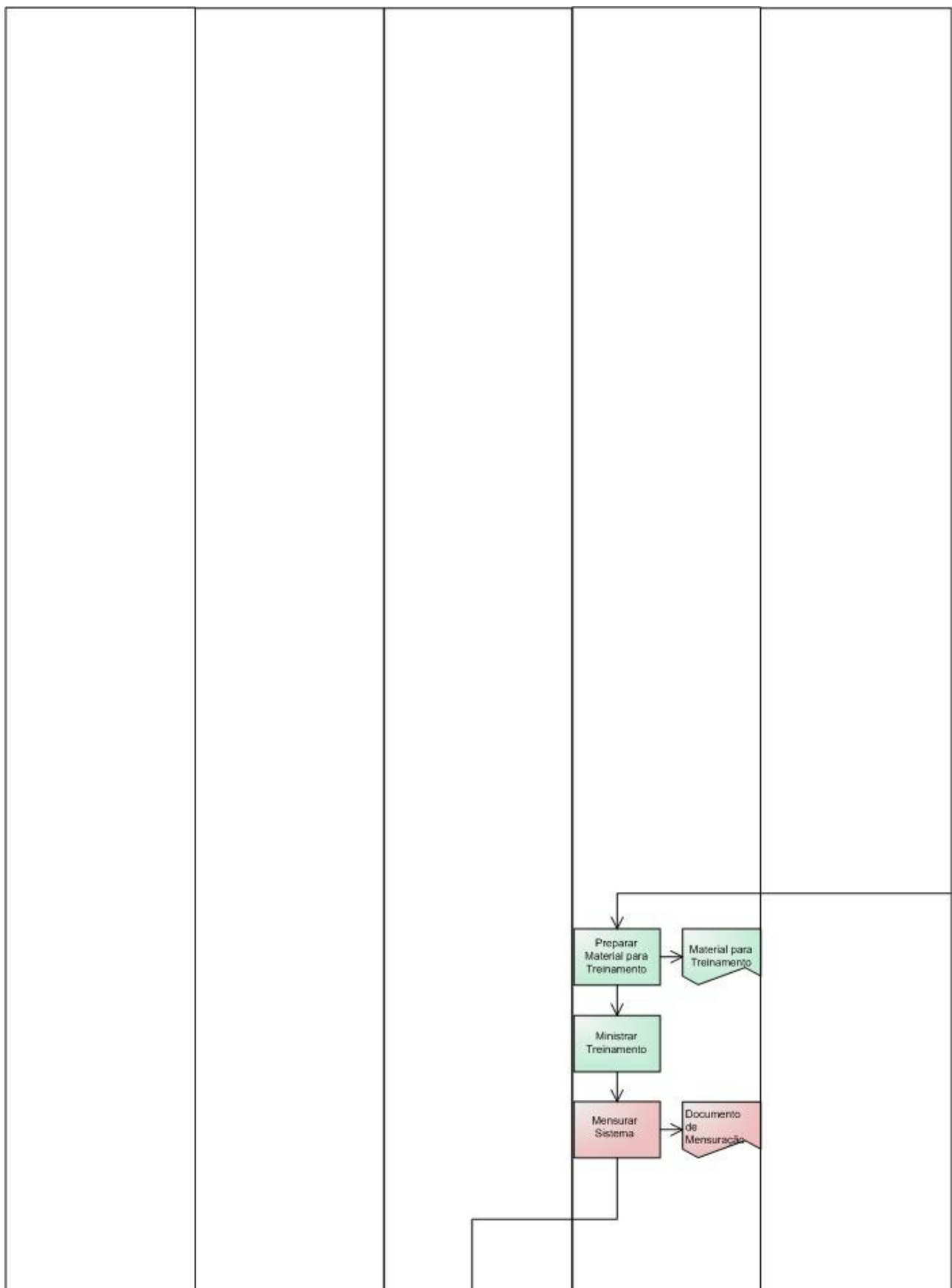
Transição

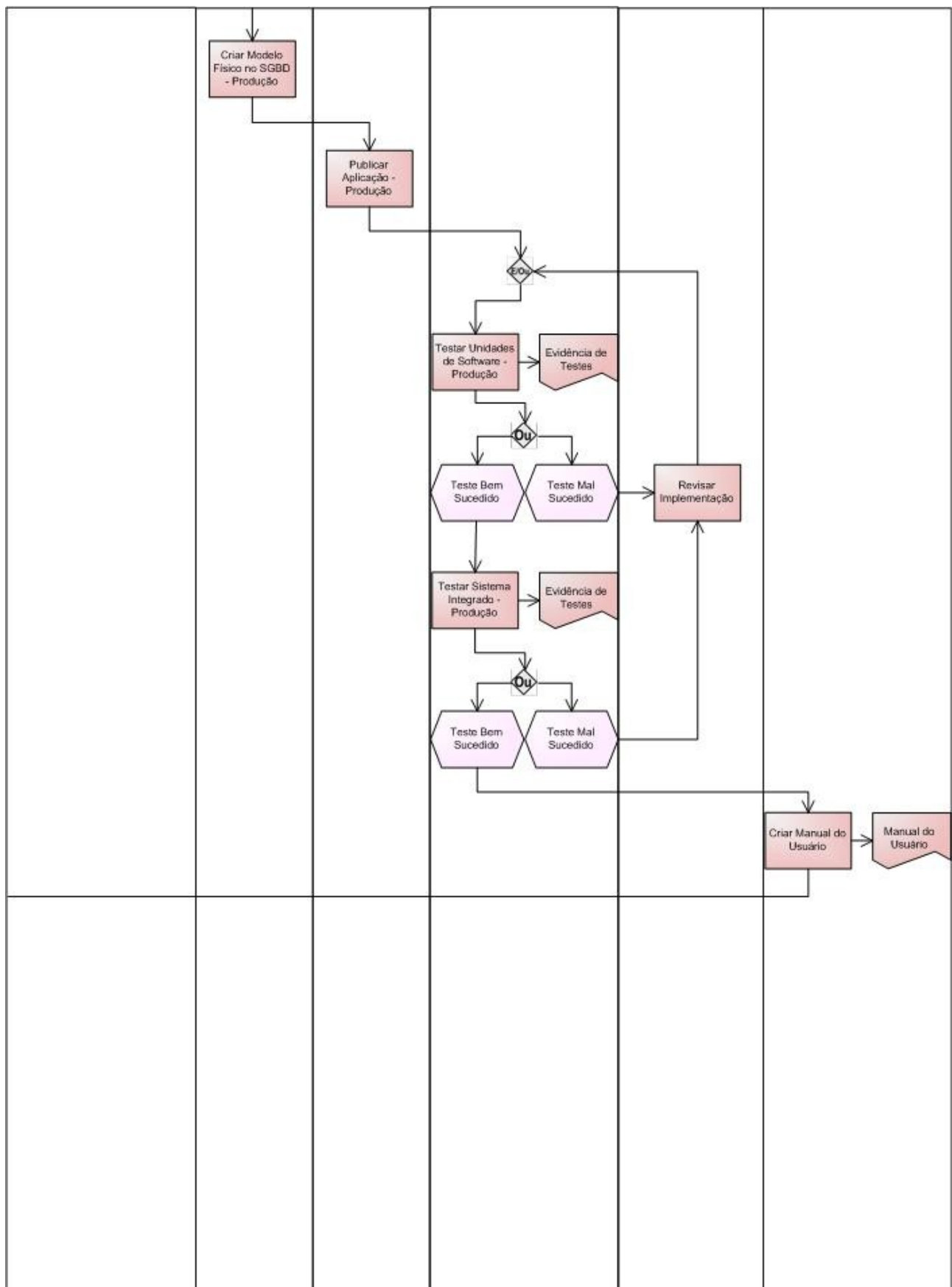


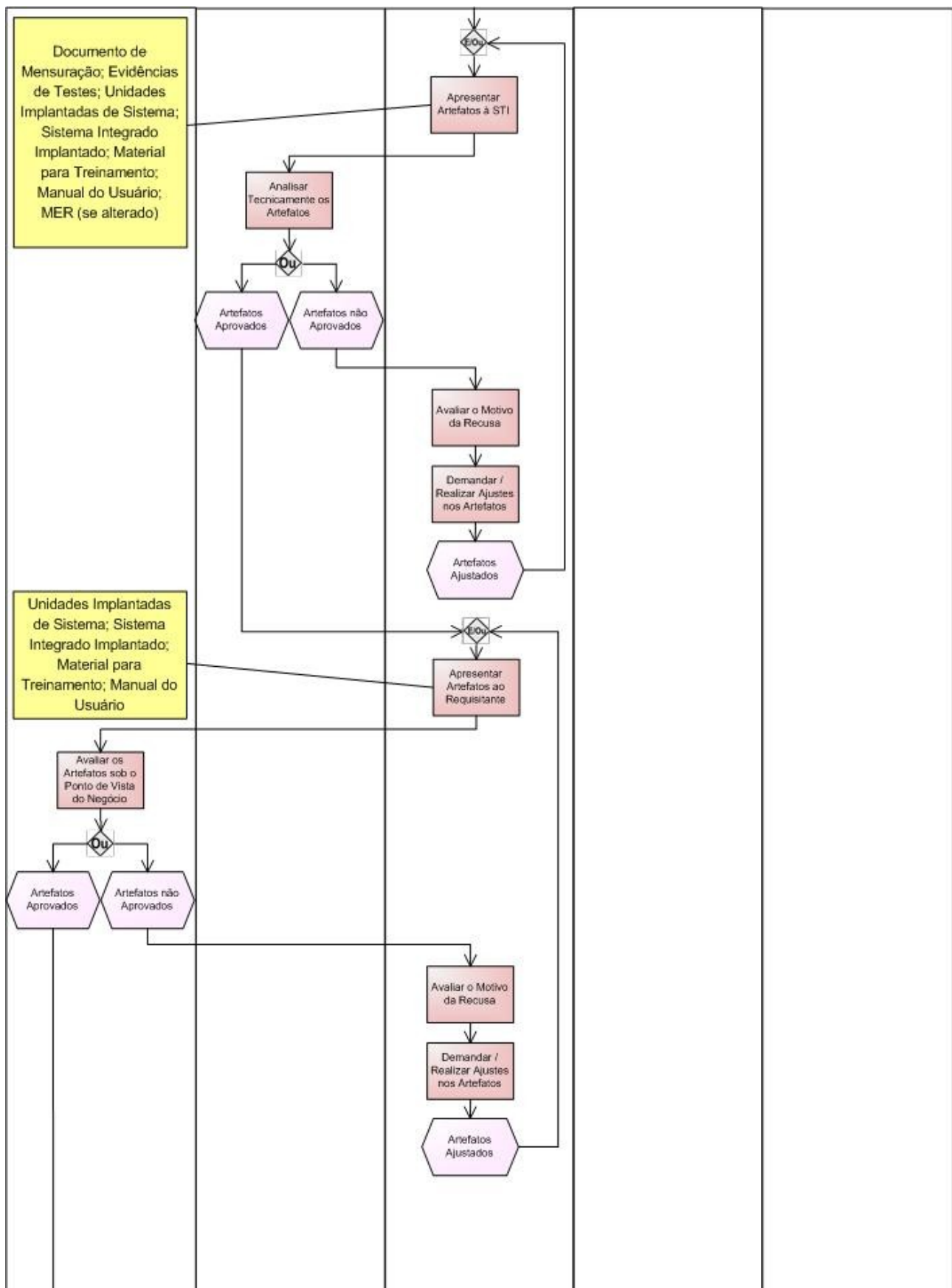
Fase Transição – Página 1A



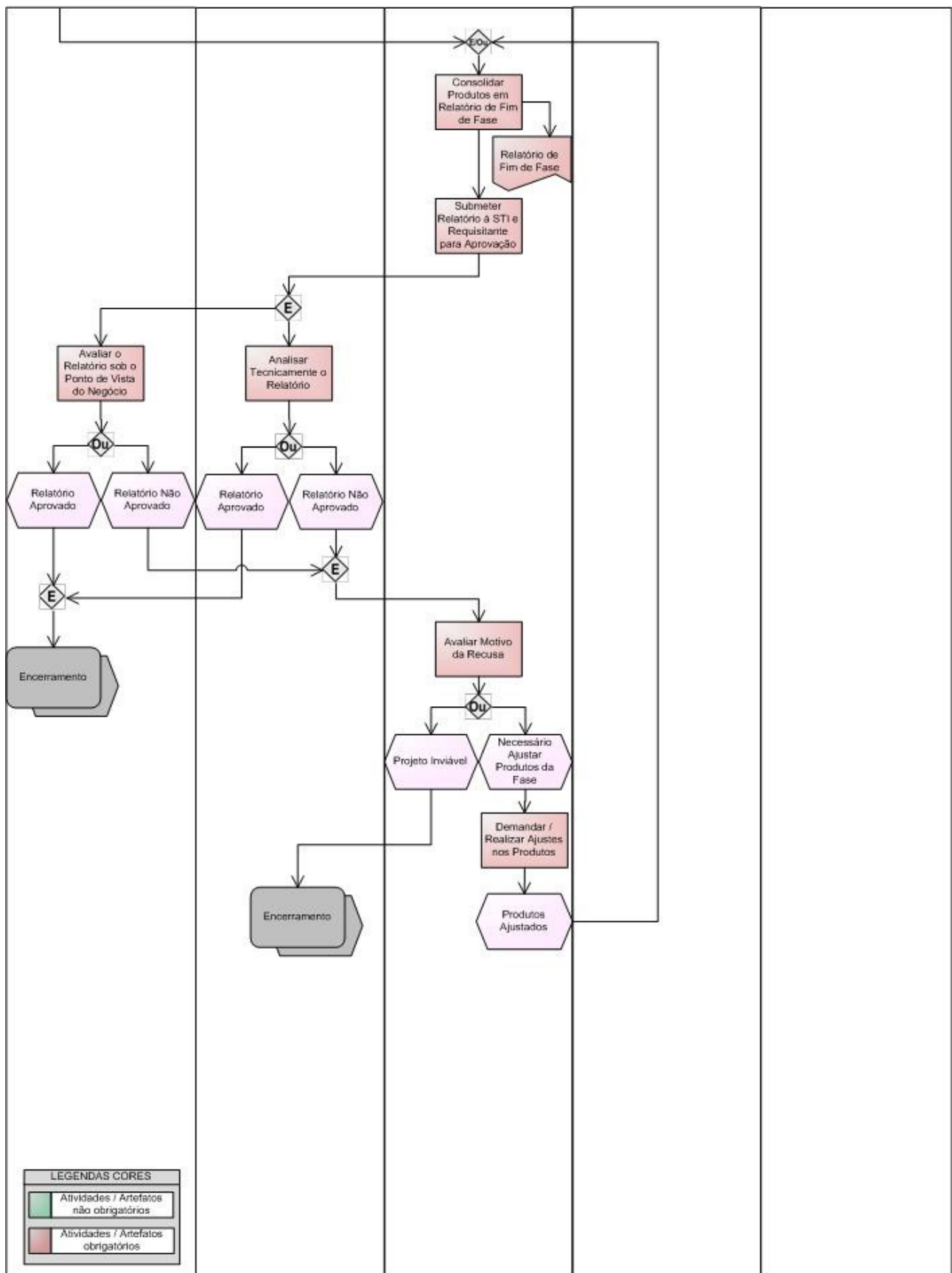
Fase Transição – Página 1B







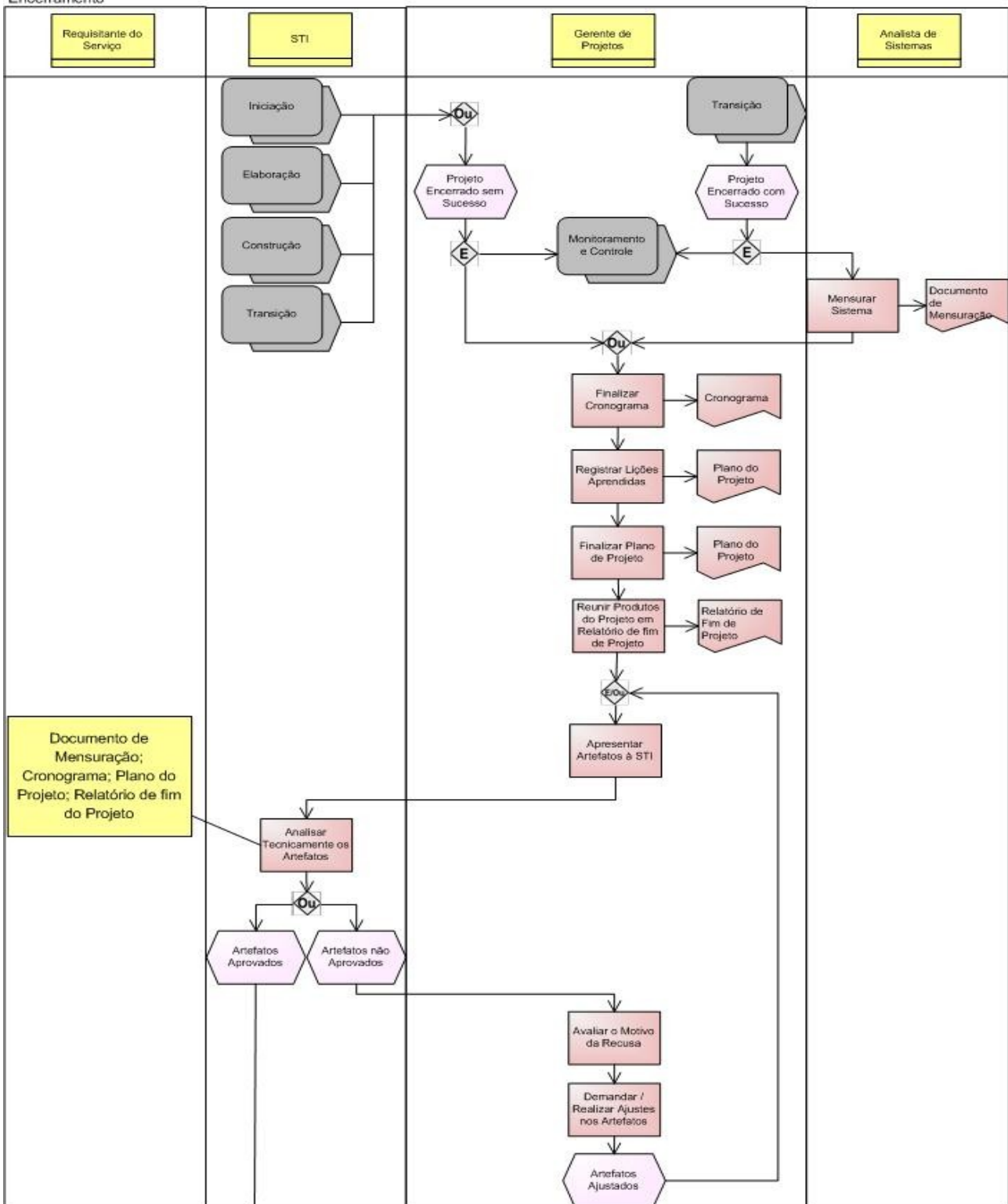
--	--	--	--	--	--

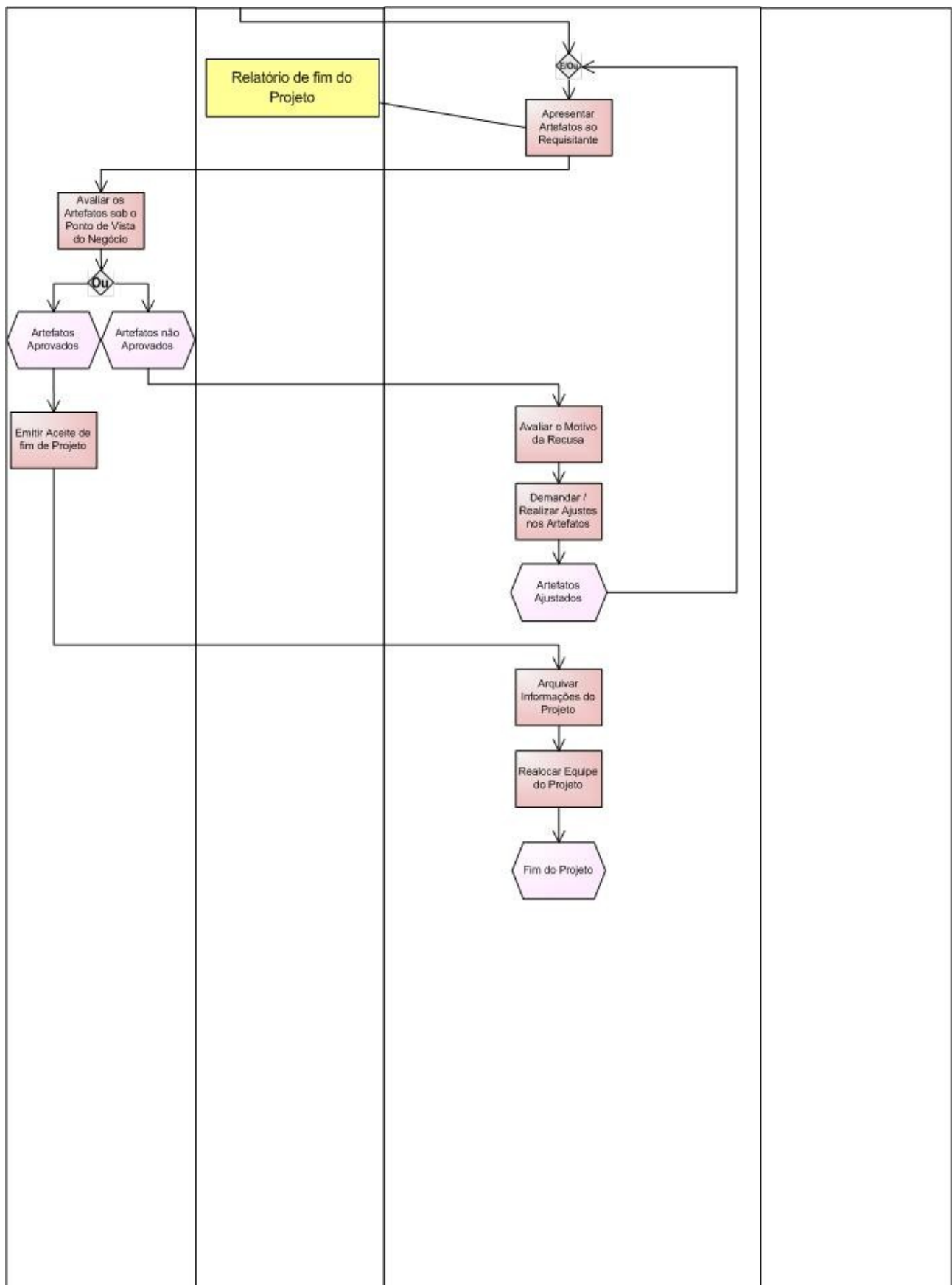


--	--	--	--	--	--

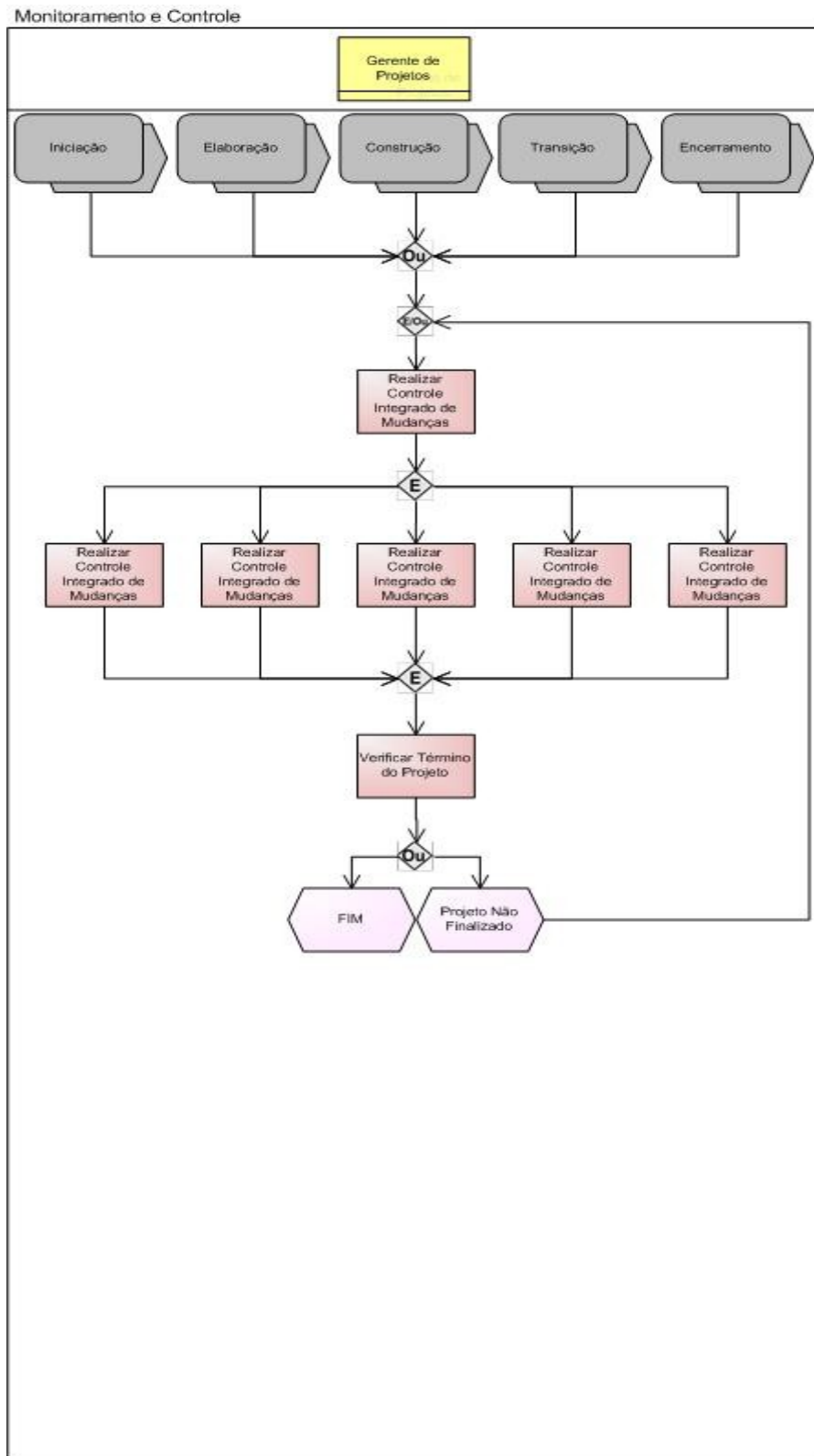
1.1.4 Encerramento

Encerramento





1.1.5 Monitoramento e Controle



2 Referências

Ivar Jacobson, Grady Booch, and James Rumbaugh. The Unified Software Development Process. Addison-Wesley, 1999.

[Processo Praxis. Disponível em <http://www.wppf.uaivip.com.br/praxis/>.](http://www.wppf.uaivip.com.br/praxis/)

PAULA FILHO, Wilson de Pádua. Engenharia de Software fundamentos, métodos e padrões. Editora: LTC, 2003.

SCOTT, Kendall. PROCESSO UNIFICADO EXPLICADO. Editora: ARTMED – BOOKMAN, 2003.

PMI, Project Management Institute. A Guide to the Project Management Body of Knowledge (PMBok). 3rd edition. Project Management Institute Inc., 2004.