

Evolução de Software através de Reengenharia: um processo didático

Mateus Henrique Dal Forno

Coorientadora: Andréa Sabedra Bordin

Roteiro

- ▶ Detalhamento do processo;
- ▶ Etapa 1: Avaliação do Software Legado;
- ▶ Etapa 2: Reengenharia;

Detalhamento do processo

- ▶ O processo inicialmente possui uma abordagem baseada no modelo cascata, para a avaliação do software legado sob perspectiva de negócio e técnica;
- ▶ Para a etapa de reengenharia do sistema, optou-se por uma abordagem incremental, a partir da elaboração e priorização da lista de incrementos que serão alocados aos ciclos de desenvolvimento seguintes.

Detalhamento do processo

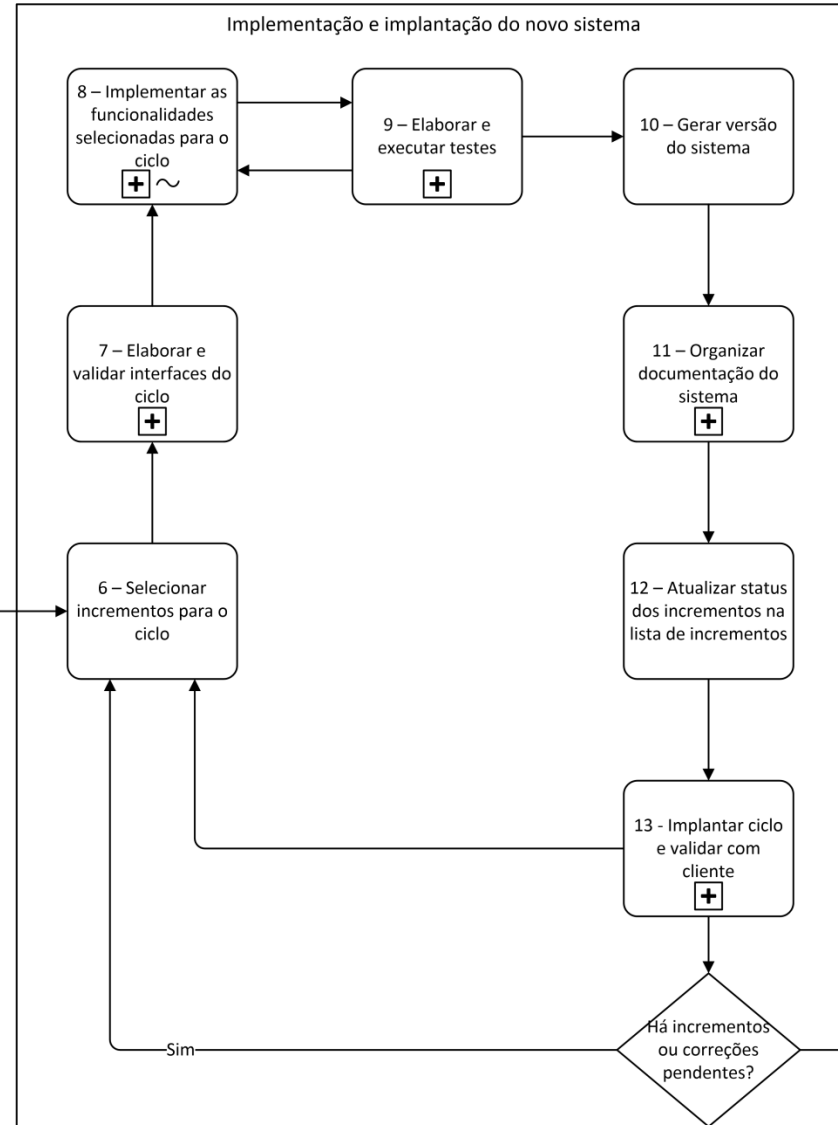
- ▶ A utilização da abordagem incremental justifica-se pela rápida entrega de software funcional ao cliente, permitindo que as funcionalidades sejam migradas gradativamente para o novo sistema, obtendo *feedback* dos usuários, onde as solicitações de melhorias ocorrem ainda na etapa de desenvolvimento.
- ▶ Optou-se pela modelagem do sistema evoluído a partir da abordagem orientada a objetos, onde a realidade a ser modelada está mais próxima do modelo, através do conceito de objetos (BEZERRA,2007) .

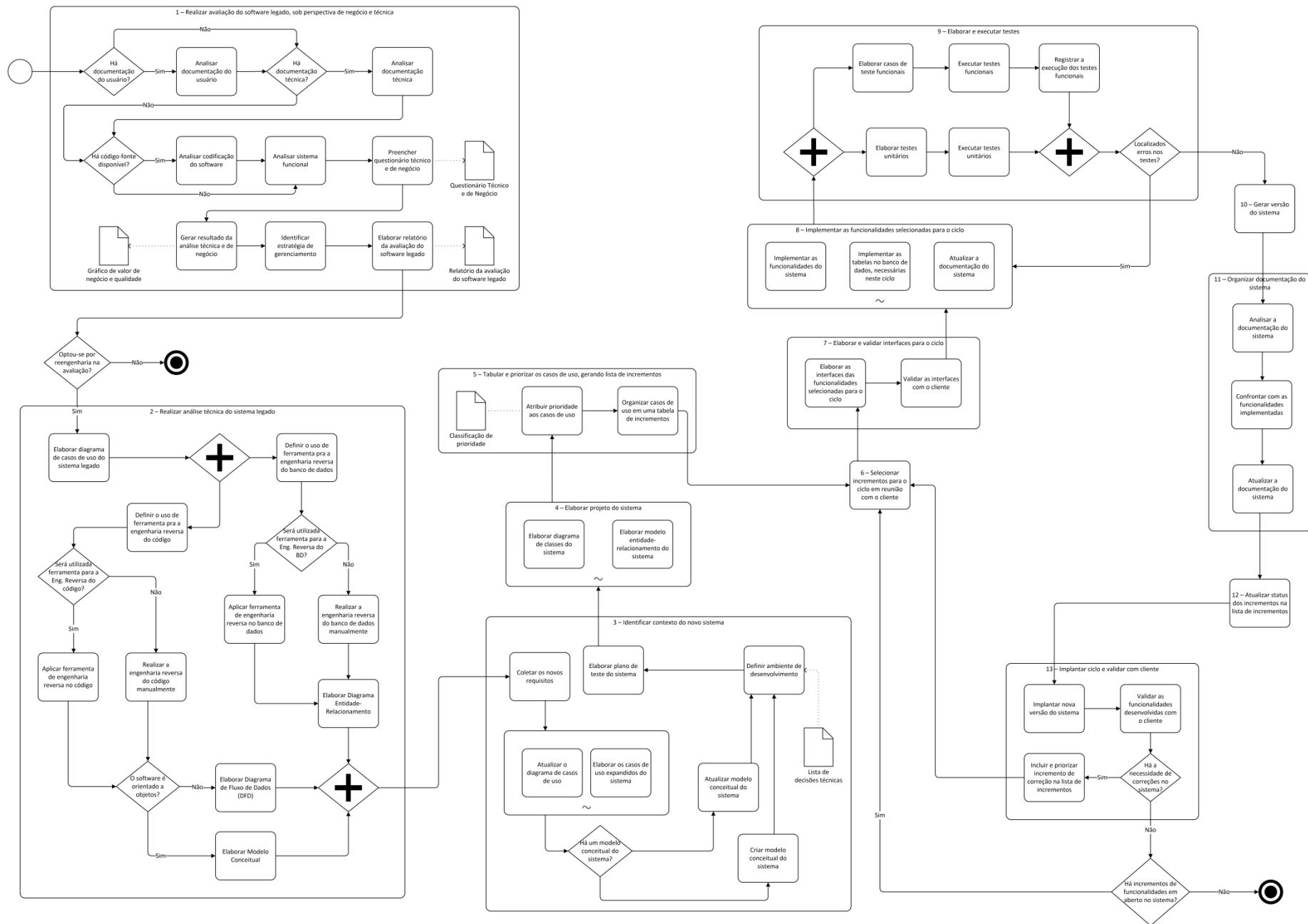
Detalhamento do processo

- ▶ O processo é modelado utilizando-se a notação BPMN;
- ▶ 3 atividades principais e 10 subprocessos;
- ▶ Dividido em 2 etapas
 - 1ª etapa: Avaliação do software legado
 - 2ª etapa: Reengenharia
 - Subdividida em 3 etapas
 - 1ª Análise do sistema legado;
 - 2ª Análise e projeto do novo sistema;
 - 3ª Implementação e implantação do novo sistema;

Etapa 1 Avaliação do software legado

Etapa 2 Reengenharia

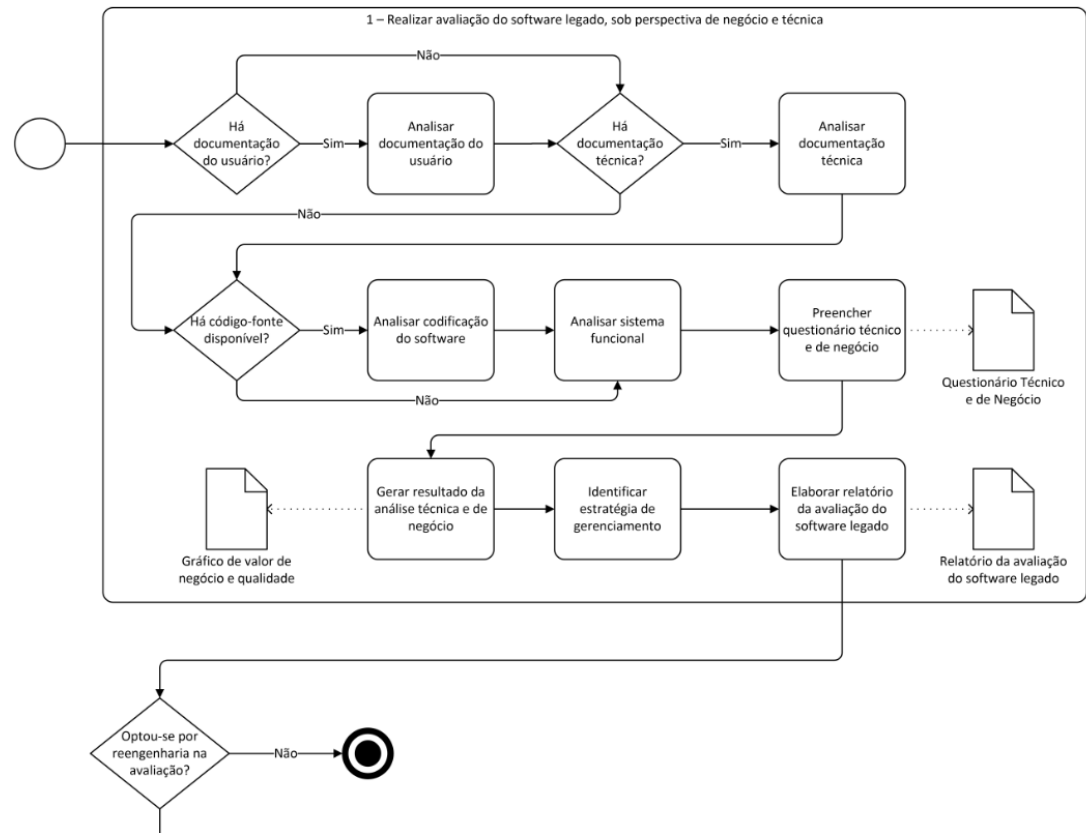




Etapa 1

Avaliação do Software Legado

- Envolve a avaliação do software legado e a definição da estratégia de manutenção para o sistema.



Atividade 1 – Realizar avaliação do software legado, sob perspectiva técnica e de negócio

- ▶ Tem por objetivo avaliar o software legado, sob duas perspectivas:
 - Perspectiva de negócio: avalia-se a real necessidade do sistema para a organização;
 - Perspectiva técnica: examina-se a qualidade do software e sua compatibilidade com as tecnologias mais recentes.
- ▶ A partir do cruzamento das perspectivas de negócio e técnica, obtém-se quatro grupos distintos de sistemas

Atividade 1 – Realizar avaliação do software legado, sob perspectiva técnica e de negócio

- ▶ Cada um desses grupos possui correspondência com as estratégias propostas para a manutenção do software:
 1. Descartar completamente o sistema;
 2. Deixar o sistema inalterado e seguir a manutenção regular;
 3. Reestruturar o sistema;
 4. Substituir o sistema por um novo sistema.

Atividade 1 – Realizar avaliação do software legado, sob perspectiva técnica e de negócio

► Grupos de sistemas

1. Baixa qualidade e baixo valor de negócio.

- O uso do software pelo cliente possui pouco retorno, e sua manutenção é cara;
- Recomenda-se o descarte;

2. Baixa qualidade e alto valor de negócio.

- São importantes para a organização e não devem ser descartados.
- A baixa qualidade indica que os custos de manutenção são altos e que o sistema deve ser reestruturado para ampliar a sua qualidade, ou então ser substituído por uma solução de prateleira.

Atividade 1 – Realizar avaliação do software legado, sob perspectiva técnica e de negócio

3. Alta qualidade e baixo valor de negócio

- Não contribuem de forma efetiva para a organização;
- Podem ser submetidos a manutenções regulares se as mudanças necessárias não forem de alto custo, caso contrário o mesmo deve ser descartado.

4. Alta qualidade e alto valor de negocio.

- Devem permanecer em operação e a manutenção regular deve ser mantida.

Atividade 1 – Realizar avaliação do software legado, sob perspectiva técnica e de negócio

- ▶ É importante salientar que o processo somente prosseguirá caso a opção escolhida seja a **reengenharia** como estratégia de manutenção do sistema legado.

Atividade 1 – Realizar avaliação do software legado, sob perspectiva técnica e de negócio

► Atividades

- Avalia-se o software, a partir de um conjunto de questões previamente definidas, com base na proposta de Sommerville(2011);
- Cada questão está identificada com o responsável ou o grupo de responsáveis por respondê-la;
- De acordo com as respostas das questões inseridas em uma planilha eletrônica, é representado em um gráfico em qual dos grupos que o sistema se enquadra;

Tabela 1 – Questões para avaliação sob perspectiva de valor de negócio

Nº	Questão	Avaliador	Resposta
1	Considerando o contexto onde o software está inserido, o sistema é usado com frequência?	Usuário	
2	Os processos de negócio do sistema estão atualizados?	Gerente	
3	O sistema é confiável (possui pouquíssimos defeitos ou falhas)?	Usuário, gerente	
4	As saídas do sistema são utilizadas? Há importância para o usuário?	Usuário, gerente	

Exemplo de questionário (perspectiva de valor de negócio)

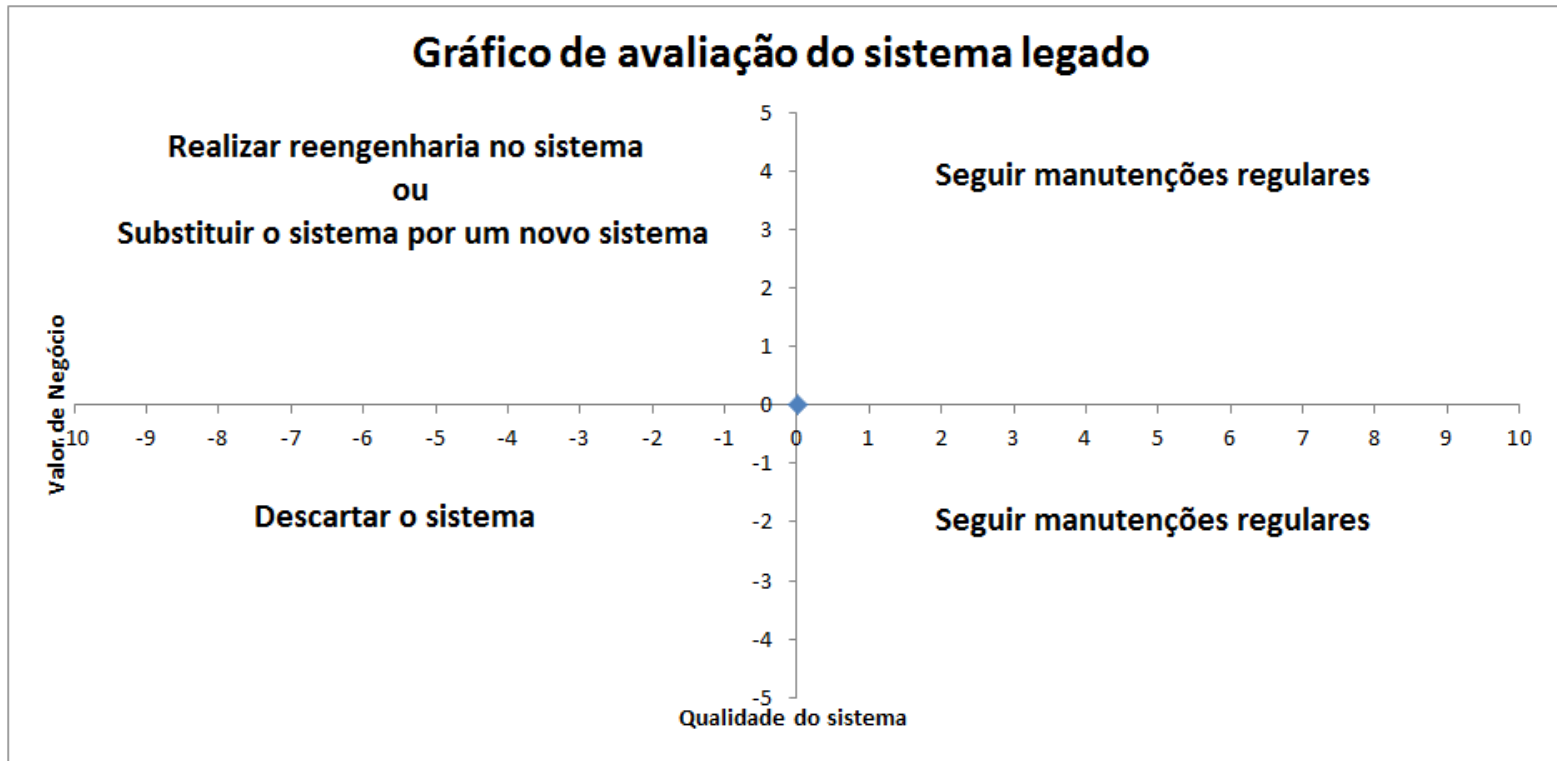
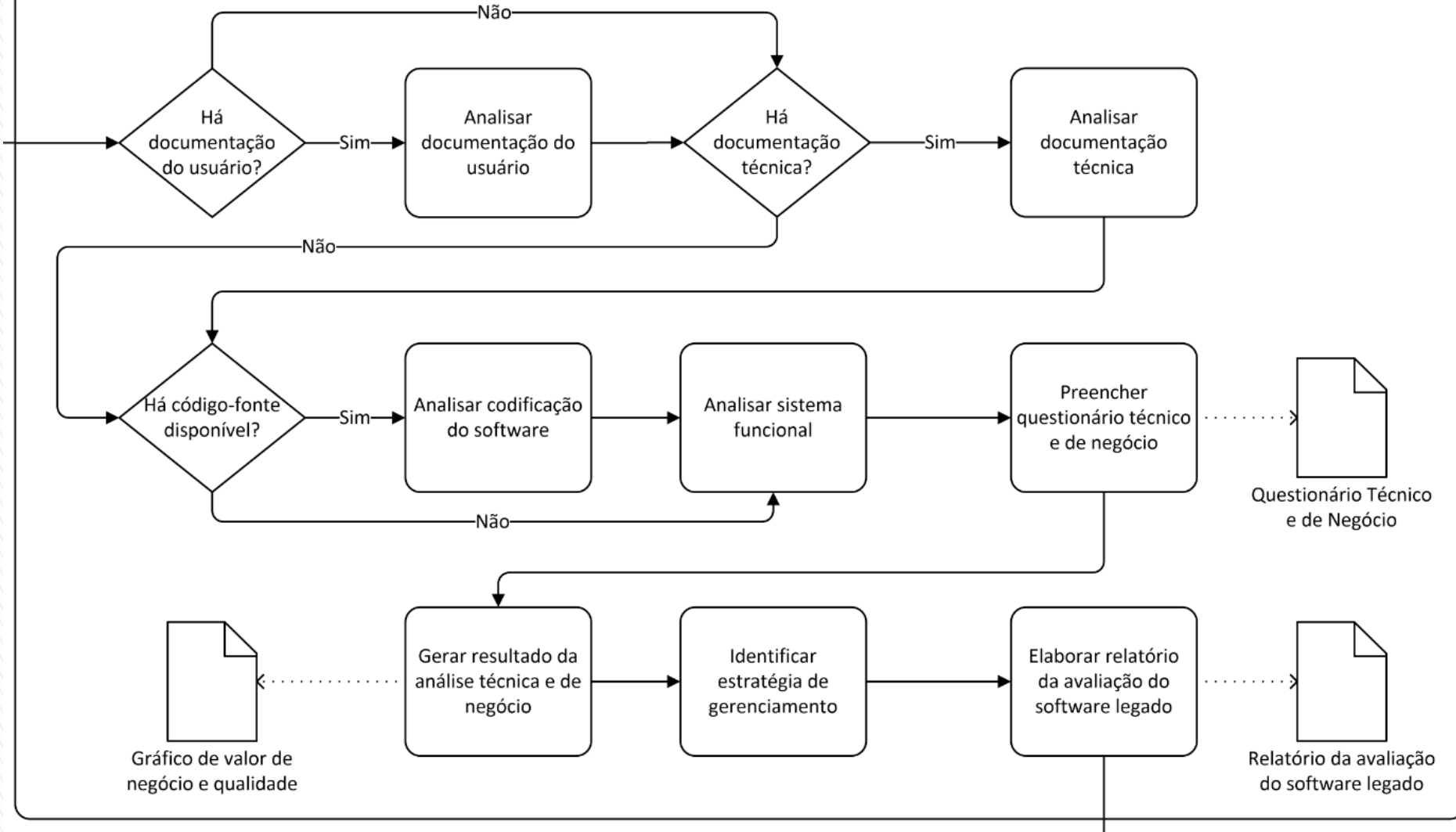


Gráfico de avaliação do sistema legado

Atividade 1 – Realizar avaliação do software legado, sob perspectiva técnica e de negócio

- ▶ Além da avaliação do software legado:
 - Identificar a linguagem de programação em que o software foi desenvolvido;
 - Identificar o propósito do sistema (a que se destina?);
 - Identificar o paradigma do software (estruturado ou orientado a objetos?);
- ▶ Essas informações deverão estar presentes no relatório da avaliação do software legado, juntamente com o questionário técnico e de negócio e o gráfico gerado a partir das respostas dos questionários e a estratégia de manutenção escolhida;

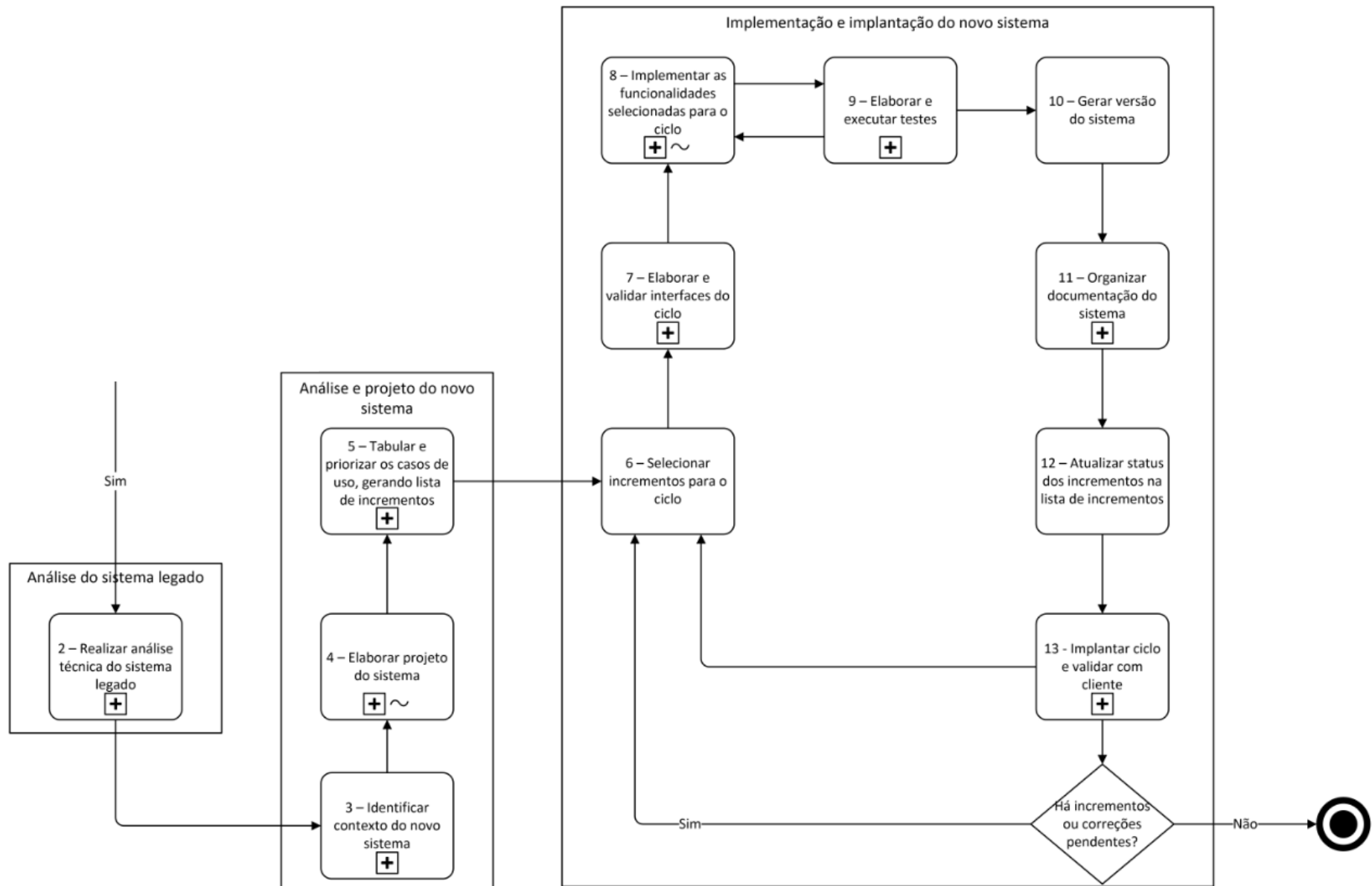
1 – Realizar avaliação do software legado, sob perspectiva de negócio e técnica



Etapa 2

Reengenharia

- ▶ Nesta etapa acontecem as atividades de reengenharia do sistema legado.
- ▶ É composta por 3 sub etapas
 - 1ª Análise do sistema legado;
 - 2ª Análise e projeto do novo sistema;
 - 3ª Implementação e implantação do novo sistema;



Atividade 2– Realizar análise técnica do software legado

- ▶ Tem por objetivo levantar informações relacionadas a estrutura do sistema legado, de forma mais aprofundada, envolvendo a elaboração do diagrama de casos de uso do sistema e a obtenção de diagramas através de engenharia reversa.
- ▶ A engenharia reversa é realizada a partir do uso de ferramentas ou de forma manual, caso não existam ferramentas para a linguagem de programação do sistema legado.

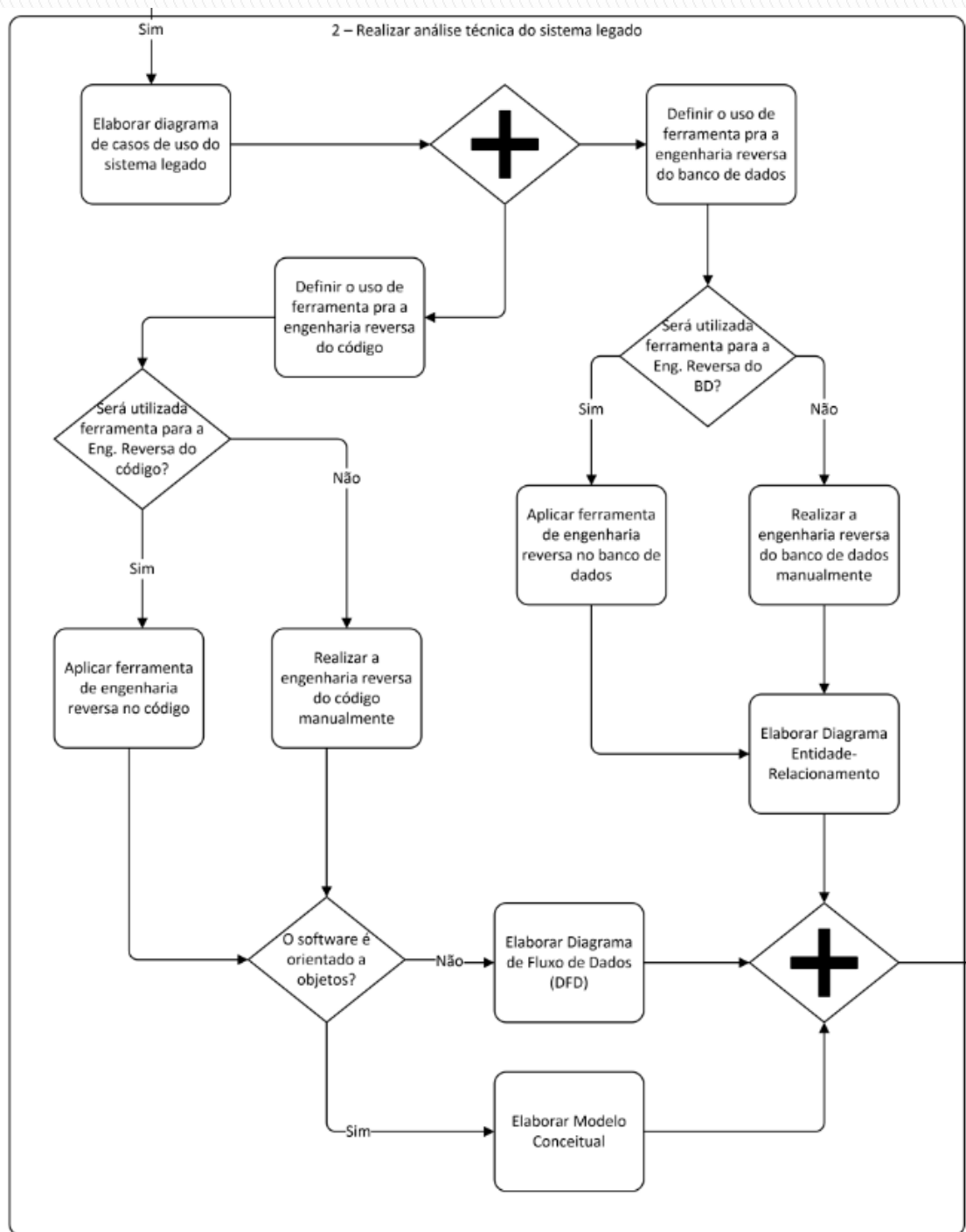
Atividade 2 – Realizar análise técnica do software legado

► Atividades

- Elaboração do diagrama de casos de uso do sistema legado;
- Definir se a engenharia reversa será apoiada por ferramentas ou realizada de forma manual;
- A partir da engenharia reversa, elaboração do modelo conceitual do software ou diagrama de fluxo de dados do software;
- A partir da engenharia reversa do banco de dados, elaboração do modelo entidade–relacionamento do software legado;

Atividade 2 – Realizar análise técnica do software legado

- ▶ Porque Modelo Conceitual **OU** Diagrama de Fluxo de Dados?
 - **Modelo conceitual:** representa as informações que o sistema gerencia e que o usuário identifica, representadas através de notação UML;
 - **Diagrama de Fluxo de Dados:** de uso mais frequente em sistemas que possuem paradigma estruturado;



Atividade 3 – Identificar contexto do novo sistema

- ▶ Envolve a identificação do contexto do novo sistema
 - Elaborado a partir das informações obtidas através da análise do sistema legado e das mudanças solicitadas pelo cliente;

Atividade 3 – Identificar contexto do novo sistema

► Atividades

- Coletar novos requisitos;
- Atualizar diagrama de casos de uso do sistema;
- Elaborar casos de uso expandidos do sistema;
- Atualizar ou elaborar modelo conceitual do sistema;
- Definir ambiente de desenvolvimento;
- Definir do plano de teste;

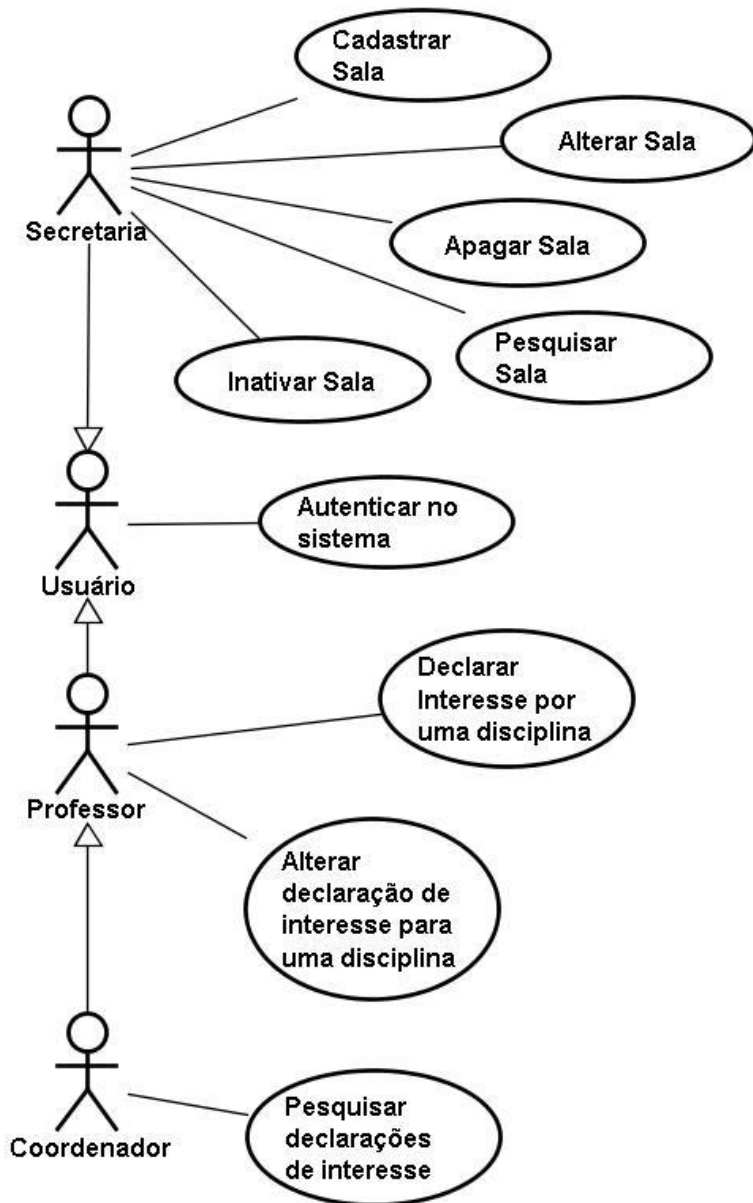
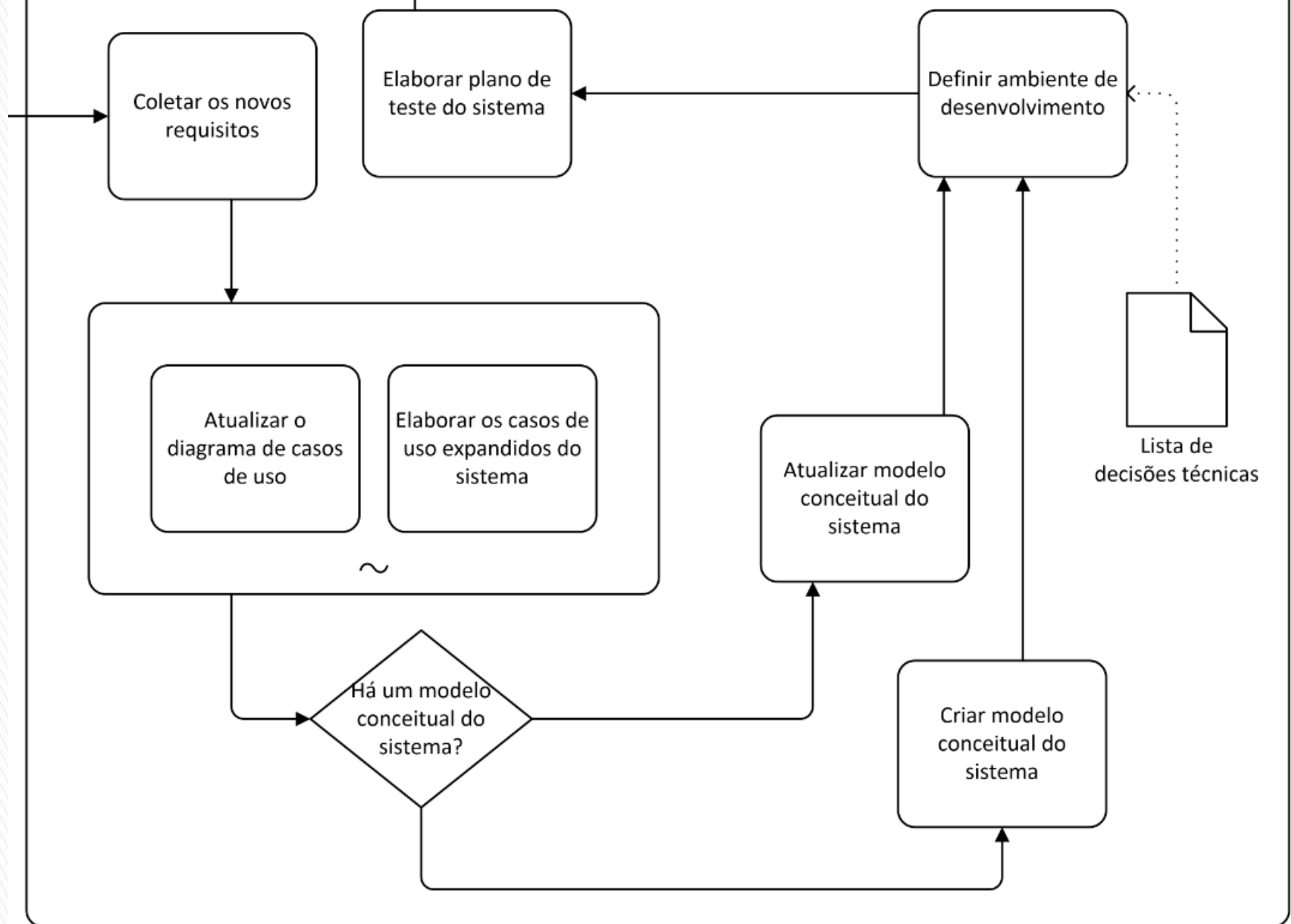


Diagrama de Caso de Uso

ITEM	VALUE
Caso de Uso	Enviar solicitação de reunião
Propósito	Como professor eu quero enviar uma solicitação de reunião para uma lista de professores para averiguar se eles possuem disponibilidade no horário desejado.
Pré-condição	Estar na tela de cadastro de eventos Ter adicionado uma data inicial
Pós-condição	
Fluxo Principal	<p>1. O usuário digita parte nome de um professor que ele deseja enviar uma solicitação de reunião</p> <p>2. O sistema busca possíveis nomes na base de dados e apresenta alternativas de nomes</p> <p>3. O usuário seleciona um nome apresentado pelo sistema e clica em um botão para adicionar o professor ao evento.</p> <p>4. O sistema axibe uma agenda com os horários livres e ocupados na data do evento do(s) professor(es) adicionado(s)</p> <p>5. O usuário verifica um horário vago na agenda do(s) professor(es) e preenche o formulário com os dados desejados e cadastra o evento.</p> <p>6. O sistema salva os dados, adiciona o evento à agenda do usuário e envia um email de solicitação com os dados do evento para todos os professores que foram adicionados ao evento pelo usuário e o caso de uso termina.</p>
Fluxo Alternativo	<p>4.1 O usuário remove algum professor do evento antes de salvar.</p> <p>4.2 O sistema remove o professor da agenda</p>

Caso de uso Expandido

3 – Identificar contexto do novo sistema



Nº	Questão técnica	Decisão
01	Linguagem de programação	
02	IDE de desenvolvimento	
03	Padrões de projeto	
04	Forma de migração das informações para o novo sistema	
05	SGBD	
06	Ferramentas de teste	
07	<outras questões técnicas>	

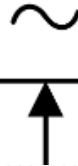
Atividade 4 – Elaborar projeto do sistema

- ▶ O projeto do novo sistema é elaborado inicialmente de forma ampla e será modificado e complementado de acordo com a necessidade durante os ciclos de desenvolvimento.
- ▶ Atividades
 - Elaboração do diagrama de classes;
 - Elaboração do modelo entidade–relacionamento;

4 – Elaborar projeto do sistema

Elaborar diagrama
de classes do
sistema

Elaborar modelo
entidade-
relacionamento do
sistema



Atividade 5 – Tabular e priorizar os casos de uso, gerando lista de incrementos

► Atividades

- A partir das informações das expansões dos casos de uso do sistema e do projeto, a equipe de desenvolvimento do sistema atribui a cada caso de uso sua prioridade, seguindo o método descrito por **Bezerra(2007)**, que relaciona o risco de desenvolvimento e a prioridade do usuário:

Atividade 5 – Tabular e priorizar os casos de uso, gerando lista de incrementos

1. **Risco alto e prioridade alta**, são casos de uso críticos e devem ser implementados o quanto antes;
2. **Risco alto e prioridade baixa**, são casos de uso de alto risco, porém sua necessidade de implementação é baixa e deve-se negociar com o cliente a necessidade de implementação;
3. **Risco baixo e prioridade alta**, possuem risco baixo, porém devem ser priorizados após os de alto risco;
4. **Risco baixo e prioridade baixa**, são os casos de uso de menor importância e os passíveis de corte em caso de atrasos no desenvolvimento.

Atividade 5 – Tabular e priorizar os casos de uso, gerando lista de incrementos

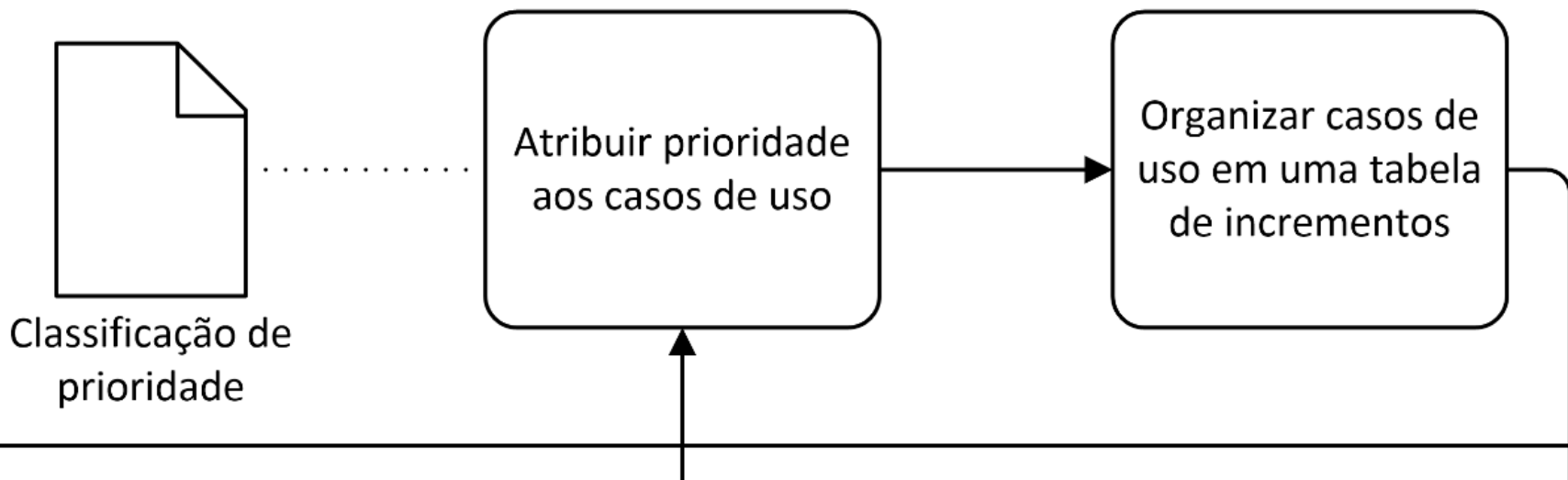
- ▶ Os casos de uso são então organizados em uma tabela de incrementos, que serão posteriormente selecionados para a realização dos ciclos.
- ▶ Nesta tabela também serão registradas as correções necessárias após a conclusão de um ciclo, a partir da verificação do cliente.

Tabela 3 – Modelo de lista de incrementos.

Nº	Descrição	Prioridade	Status
01	Cadastro de Clientes	03	Em aberto
02	Emissão de Nota Fiscal	01	Realizado no ciclo 1
03	Cadastro de Produtos	03	Em implementação

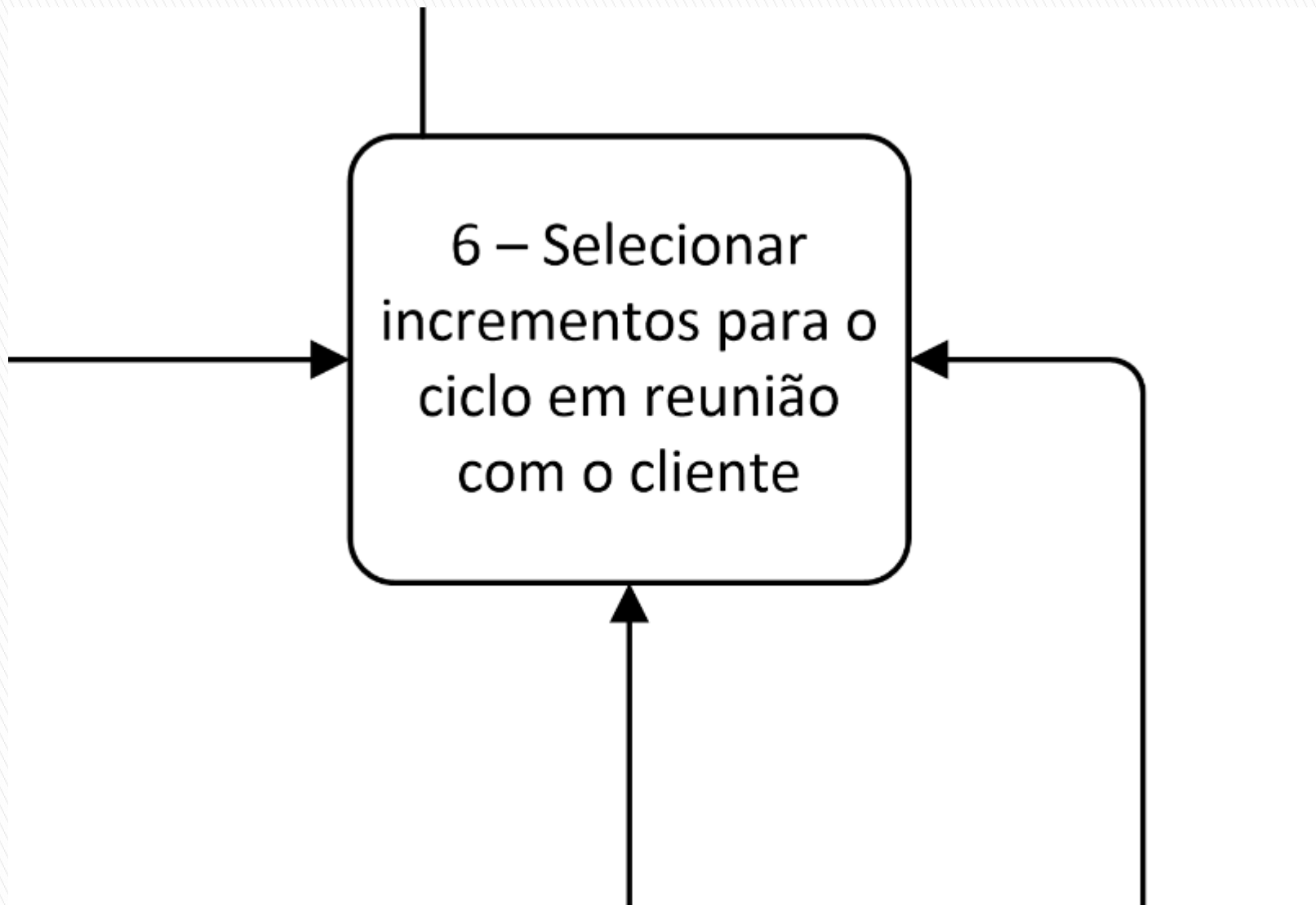
Modelo de lista de incrementos

5 – Tabular e priorizar os casos de uso, gerando lista de incrementos



Atividade 6 – Selecionar incrementos para o ciclo

- ▶ Esta atividade tem por objetivo selecionar os incrementos que serão realizados durante o ciclo de desenvolvimento, através de uma reunião com o cliente.
 - A definição de quais incrementos serão selecionados para o ciclo é determinada em uma reunião da equipe, juntamente com o cliente e levando em consideração:
 - O tempo de duração do ciclo de desenvolvimento;
 - A priorização dos incrementos;
 - A dependência entre os incrementos;
 - As necessidades do cliente;
 - Os incrementos de correção (caso existam) e sua prioridade;



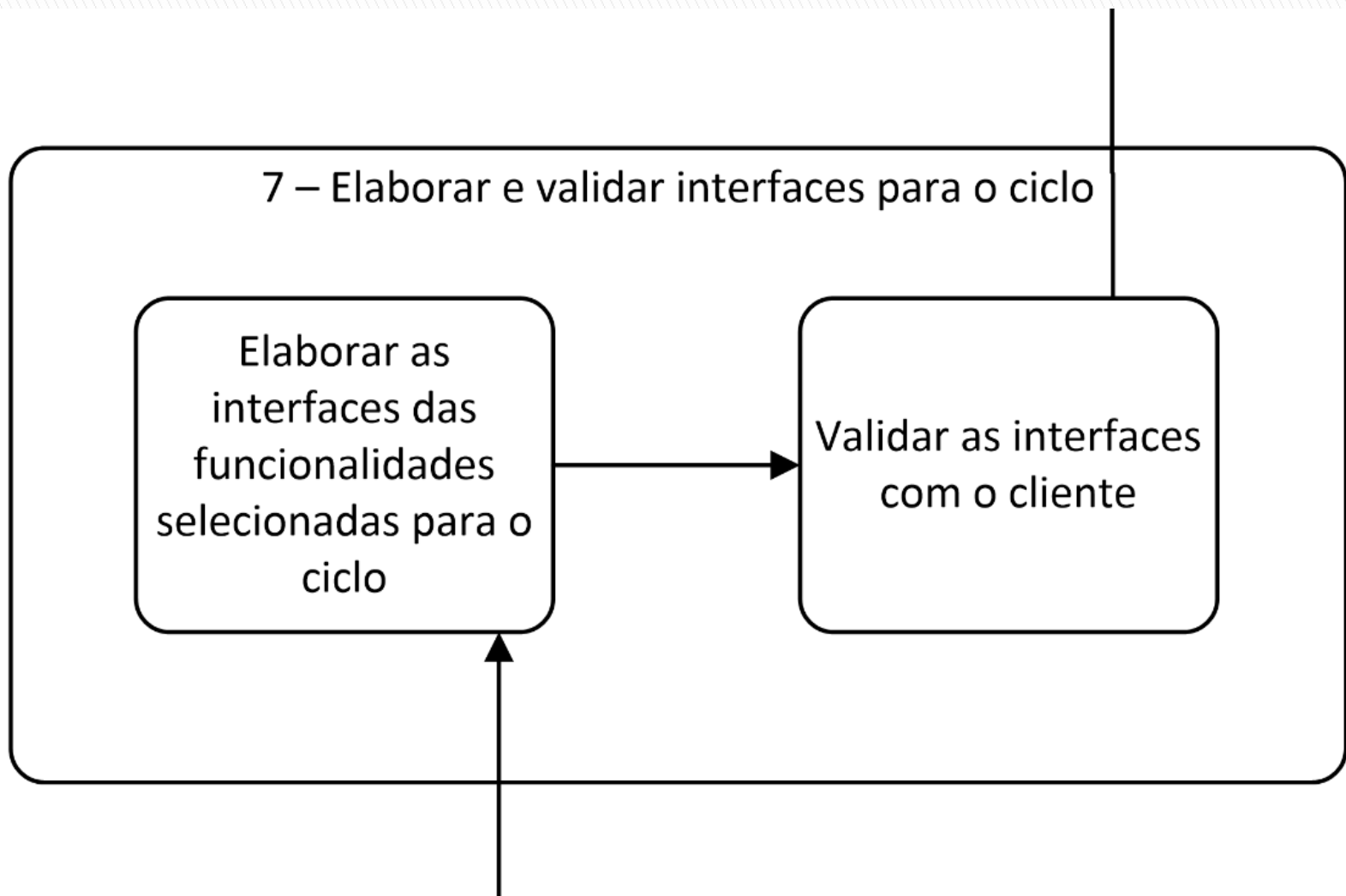
Atividade 7 – Elaborar e validar interfaces do ciclo

- ▶ Nesta etapa do processo são elaboradas as interfaces das funcionalidades selecionadas para o ciclo, observando mudanças solicitadas pelo cliente e também recomendações para a elaboração de interfaces.

Atividade 7 – Elaborar e validar interfaces do ciclo

► Atividades

- A elaboração das interfaces deve utilizar como referência as interfaces antigas
 - Evitar a perda de identidade ao sistema antigo.
- Validar a usabilidade do sistema, utilizando-se ao menos um dos métodos de avaliação de usabilidade;
 - Estabelecer um nível adequado de usabilidade do sistema.
- As interfaces devem ser validadas antes da programação;



Atividade 8 – Implementar as funcionalidades do ciclo

- ▶ Implementação da funcionalidade selecionada para o ciclo de desenvolvimento.
- ▶ Implantação no Banco de Dados das tabelas necessárias ao sistema neste ciclo.
- ▶ Correção de inconsistências identificadas na documentação do sistema.

8 – Implementar as funcionalidades selecionadas para o ciclo

Implementar as
funcionalidades do
sistema

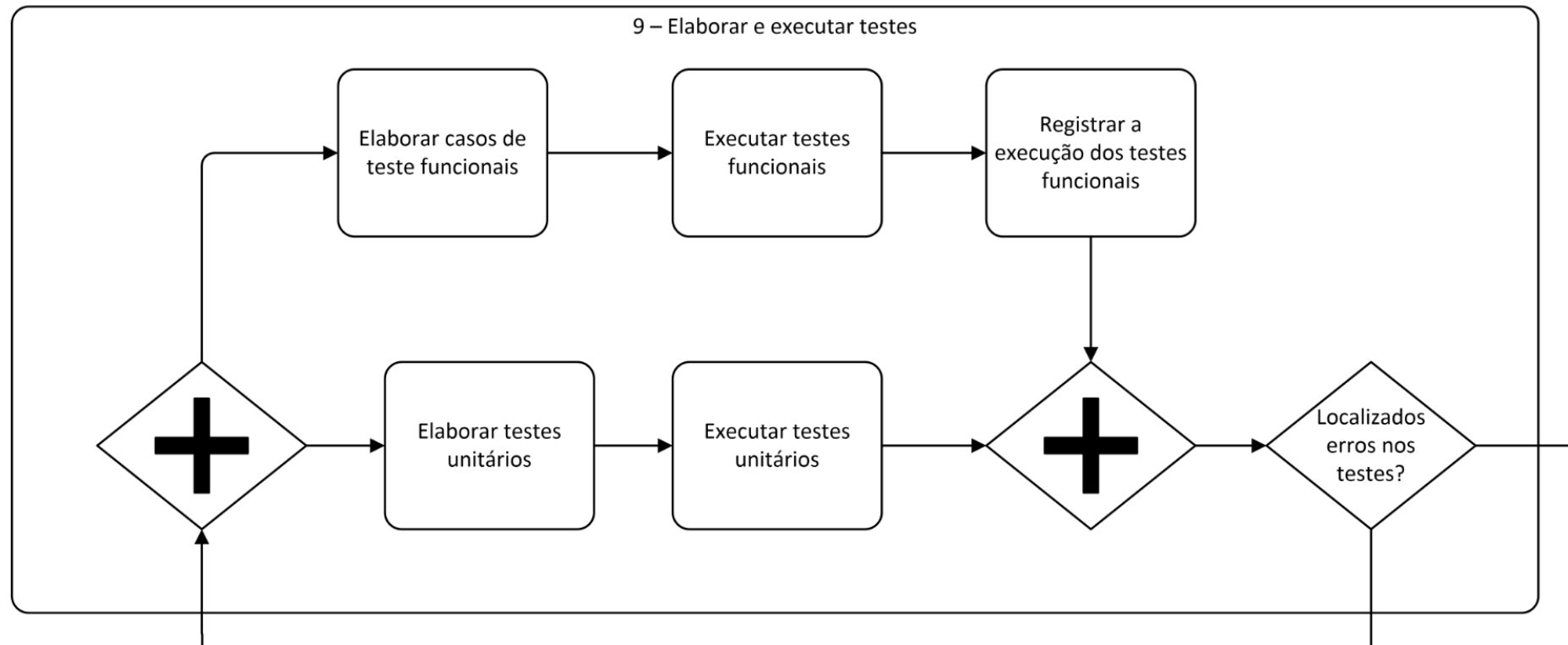
Implementar as
tabelas no banco de
dados, necessárias
neste ciclo

Atualizar a
documentação do
sistema

~

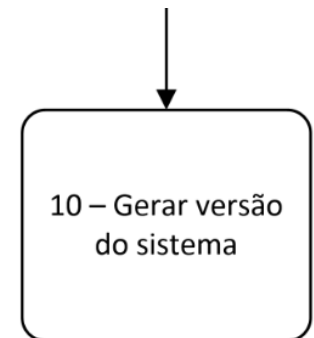
Atividade 9 – Elaborar e executar testes

- ▶ A atividade de elaboração e execução de testes envolve testes funcionais e testes unitários.
 - A realização de testes unitários poderá ser apoiada por ferramentas que automatizem o processo.
- ▶ O controle dos testes funcionais deverá ser realizado através de uma ferramenta que registre a execução dos casos de testes e seus resultados.
- ▶ Verificar a funcionalidade do sistema, levando em consideração a funcionalidade no software legado;



Atividade 10 – Gerar versão do sistema

- ▶ Compilação de uma versão do sistema, incluindo as funcionalidades implementadas no ciclo e nos ciclos anteriores.
- ▶ A versão será identificada pelo número do ciclo que estiver em execução, como forma de manter o histórico das versões do sistema;



Atividade 11 – Organizar documentação do sistema

- ▶ A documentação deve ser revisada,
 - Verificar se as inconsistências identificadas nas etapas anteriores foram corrigidas;
 - Verificar se os diagramas foram complementados com informações omitidas inicialmente.

11 – Organizar documentação do sistema

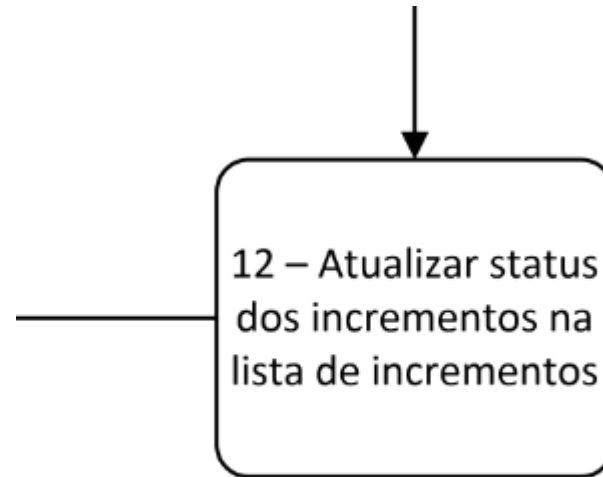
Analisar a documentação do sistema

Confrontar com as funcionalidades implementadas

Atualizar a documentação do sistema

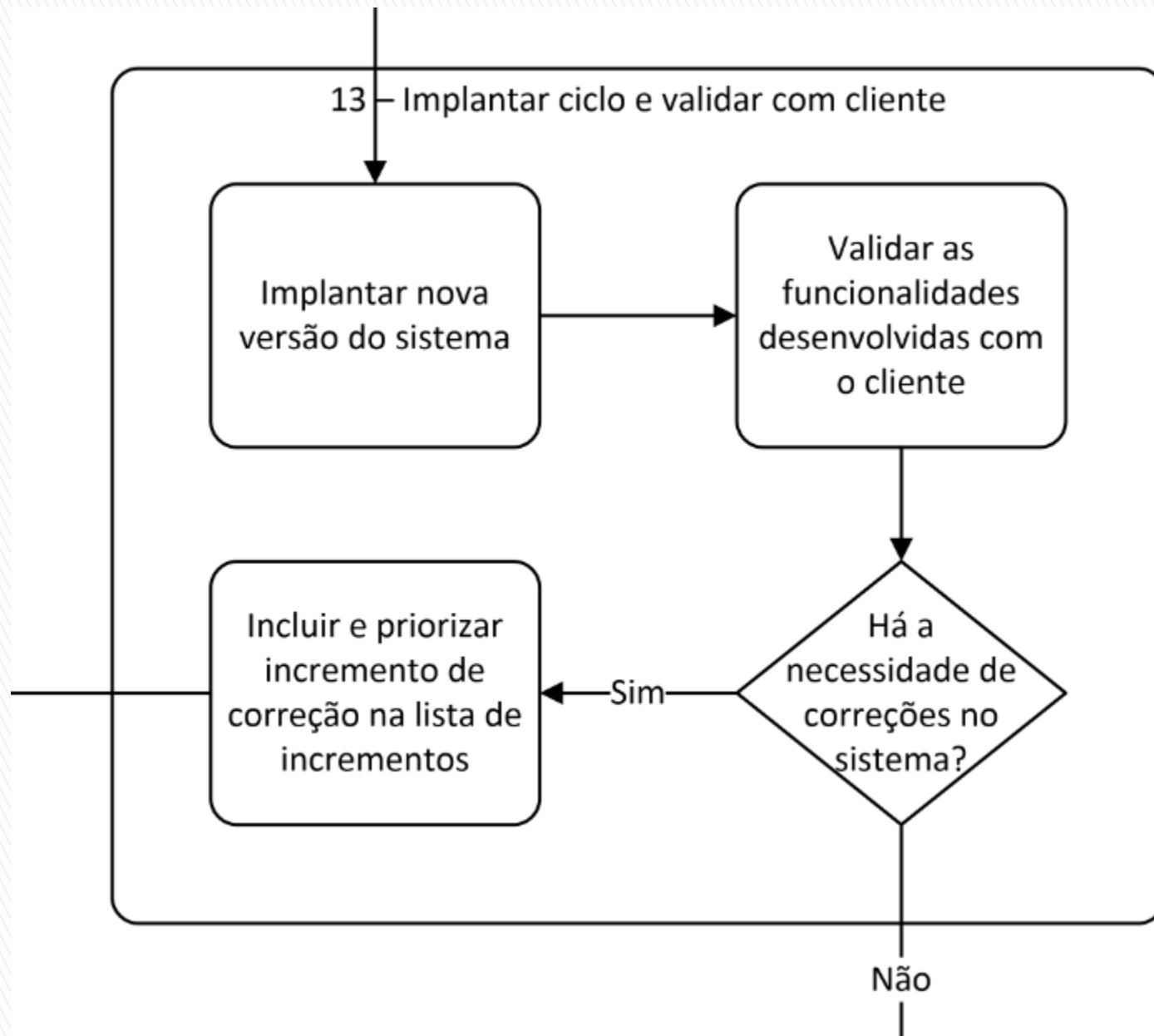
Atividade 12: Atualizar status dos incrementos na lista de incrementos

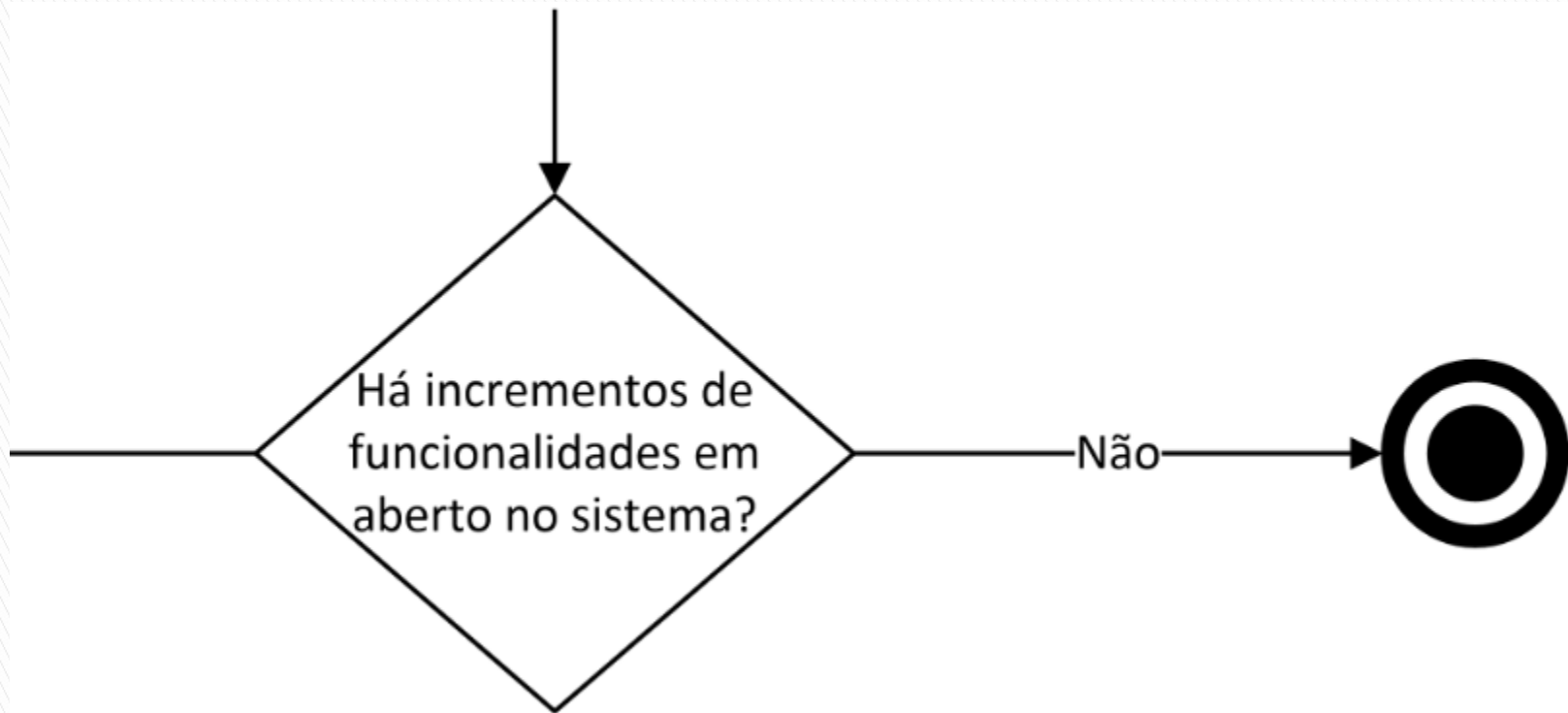
- ▶ A lista de incrementos deve ser atualizada
 - Alterando os status dos incrementos da lista para “Realizado no ciclo X”, referenciando em qual versão do sistema o incremento foi realizado.

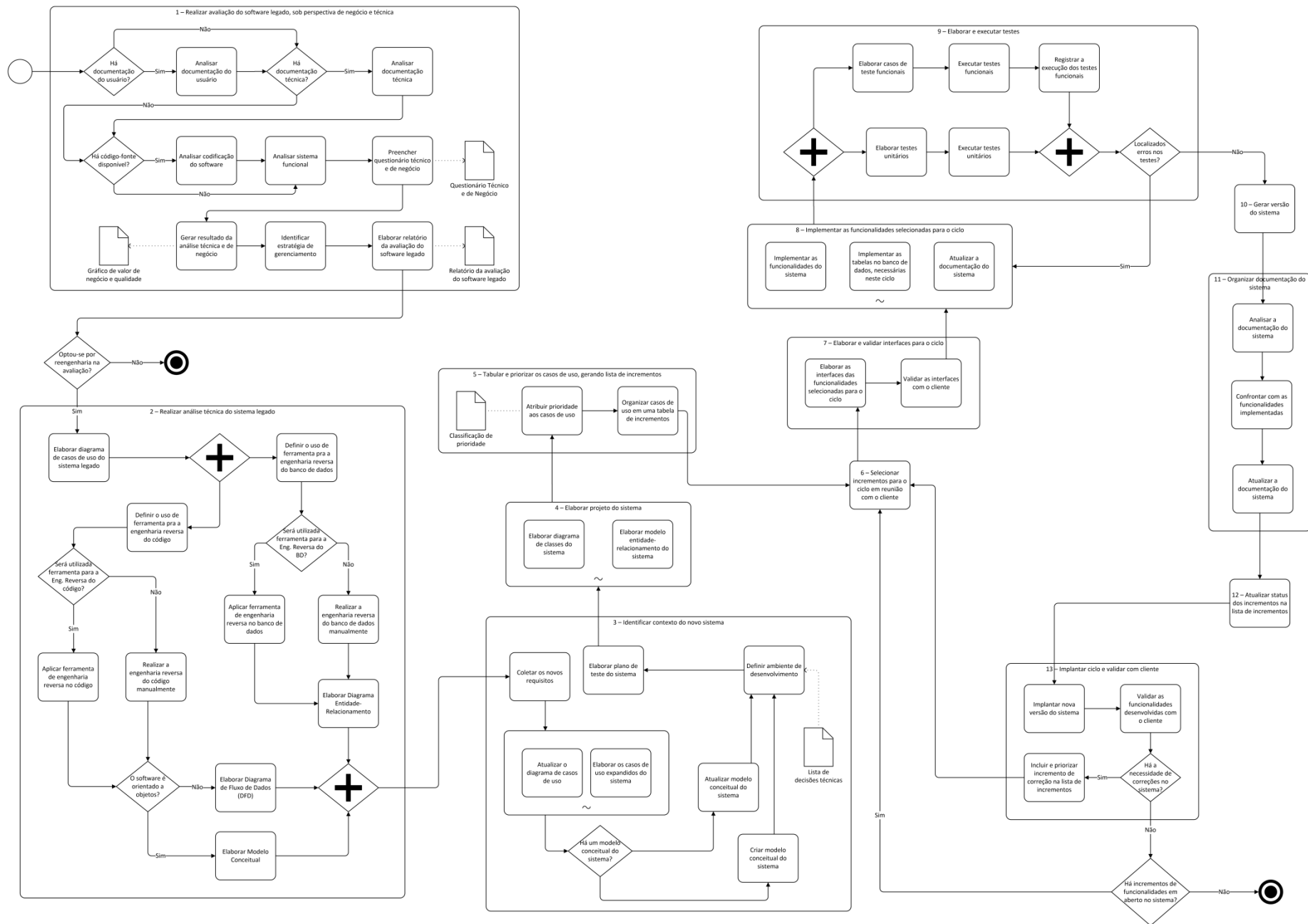


Atividade 13 – Implantar ciclo e validar

- ▶ A nova versão do sistema é implantada no cliente;
 - Migração das funcionalidades do sistema legado para o novo sistema.
- ▶ O cliente valida as novas funcionalidades implementadas no ciclo e avalia a necessidade de correções no sistema,
 - Inclusão de incrementos de correção na lista de incrementos do sistema.
 - O incremento é priorizado, seguindo o método proposto por Bezerra(2007);
 - A correção será selecionada e realizada em um ciclo posterior, de acordo com a prioridade definida para o mesmo.



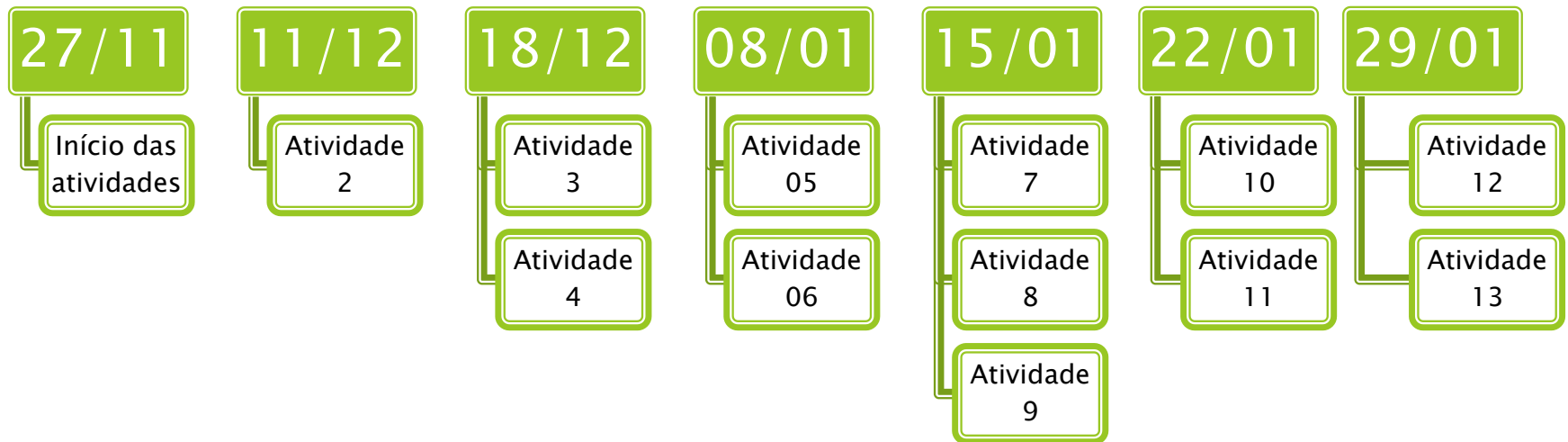




Momentos de contato com usuário

1. Atividade 3 – Coletar novos requisitos;
2. Atividade 5 – Definir a lista de incrementos;
3. Atividade 6 – Selecionar os incrementos;
4. Atividade 7 – Validar as interfaces;
5. Atividade 13 – Validação do sistema;

Cronograma



Link para download do software

- ▶ **Gerenciador de Concursos Públicos para Docentes**
- ▶ <http://online.unipampa.edu.br/concursos/gcp/>

Critérios de avaliação

- ▶ Grupo (60%)
 - Entrega e adequação das atividades dentro do cronograma.
 - Completude do produto final.
- ▶ Individual (40%)
 - Apresentação da atividade sob responsabilidade.

Grupos

- ▶ Grupo 1
 - Douglas
 - Alex
 - Miguel
 - Gean
 - Vinicius
 - Pedro
 - Edson
 - Gustavo
- ▶ Grupo 2
 - Gabriel
 - Eduardo
 - Kezia
 - Jean
 - Rodrigo
 - Tiago
 - Giovanni
 - Anderson

Referências

- ▶ BEZERRA, E. Princípios de análise e projeto de sistemas com UML. 2. ed. Rio de Janeiro:Elsevier, 2007.