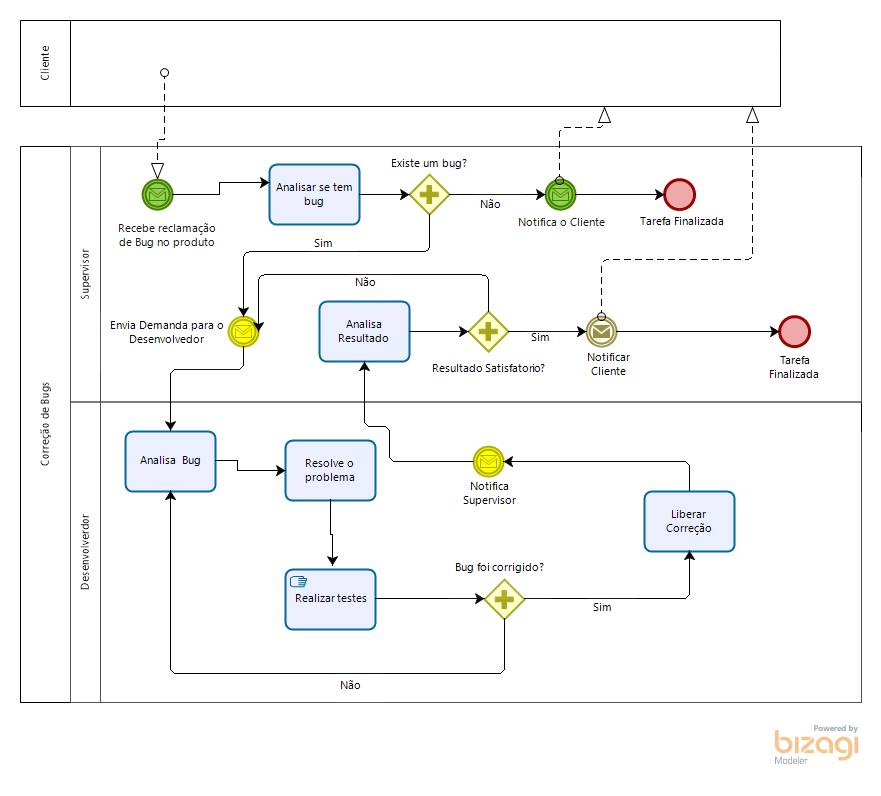
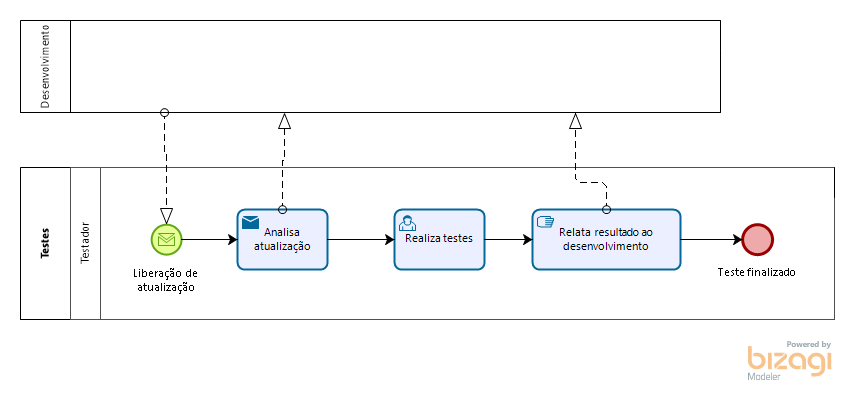
# 4. Modelagem do processo de negócio

## 

## 4.1. Análise da situação atual

O foco desta análise é uma fabrica de desenvolvimento de software que não possui um sistema para organizar suas pendências. Analisamos a situação atual de seus processos e foi encontrado diversos problemas que poderiam facilmente serem resolvidos caso a empresa tivesse maior organização em suas atividades. Segue abaixo quatro processos analisados  
  
**Criação de um projeto de software**:  
  
 No modelo é possível perceber as do processo analisado. A resolução depende muito do conhecimento e memória do supervisor e do desenvolvedor. Não há nenhuma forma de documentação e a análise da demanda é superficial. O principal problema do processo atual é que a tarefa pode ir e voltar entre o supervisor e o desenvolvedor várias vezes até que uma solução seja encontrada, gerando muito retrabalho e prejuízo para a empresa.

**Correção de bugs:**Na correção de bugs vemos que as opções são superficiais, dando muitas brechas para a não correção do bug todo em si. Primeiramente a análise do bug é feita pelo supervisor, que provavelmente já fica sobrecarregado com outras obrigações. O desenvolvedor fica preso tentando resolver o problema e não possui nenhuma ajuda externa caso encontre obstáculos. Posteriormente, o supervisor novamente fica responsável por encaminhar o resultado para o cliente, causando ainda mais sobrecarga em seu trabalho.

**Testes:**

Nesse processo é evidente notar que os testes são rasos, sem opções de maiores questionamentos ou novos destinos da tarefa. Com esse modelo simplificado de testes a probabilidade de realmente encontrar erros é muito baixa.

## 4.2. Descrição Geral da proposta

Nossa proposta é um software capaz de gerenciar as tarefas relacionadas com desenvolvimento de software, garantindo mais precisão em todos esses processos e, consequentemente, melhor qualidade, menos erros e mais satisfação ao cliente. Queremos um ambiente onde os setores de desenvolvimento, testes e supervisão possam colaborar uns com os outros em prol da melhoria e da qualidade, com uma maior facilidade de troca de informações.

Foi acrescentado o setor de atendimento, que será responsável pelo contato direto com o cliente, garantindo diminuição no trabalho do supervisor e garantindo uma melhor comunicação com o cliente.

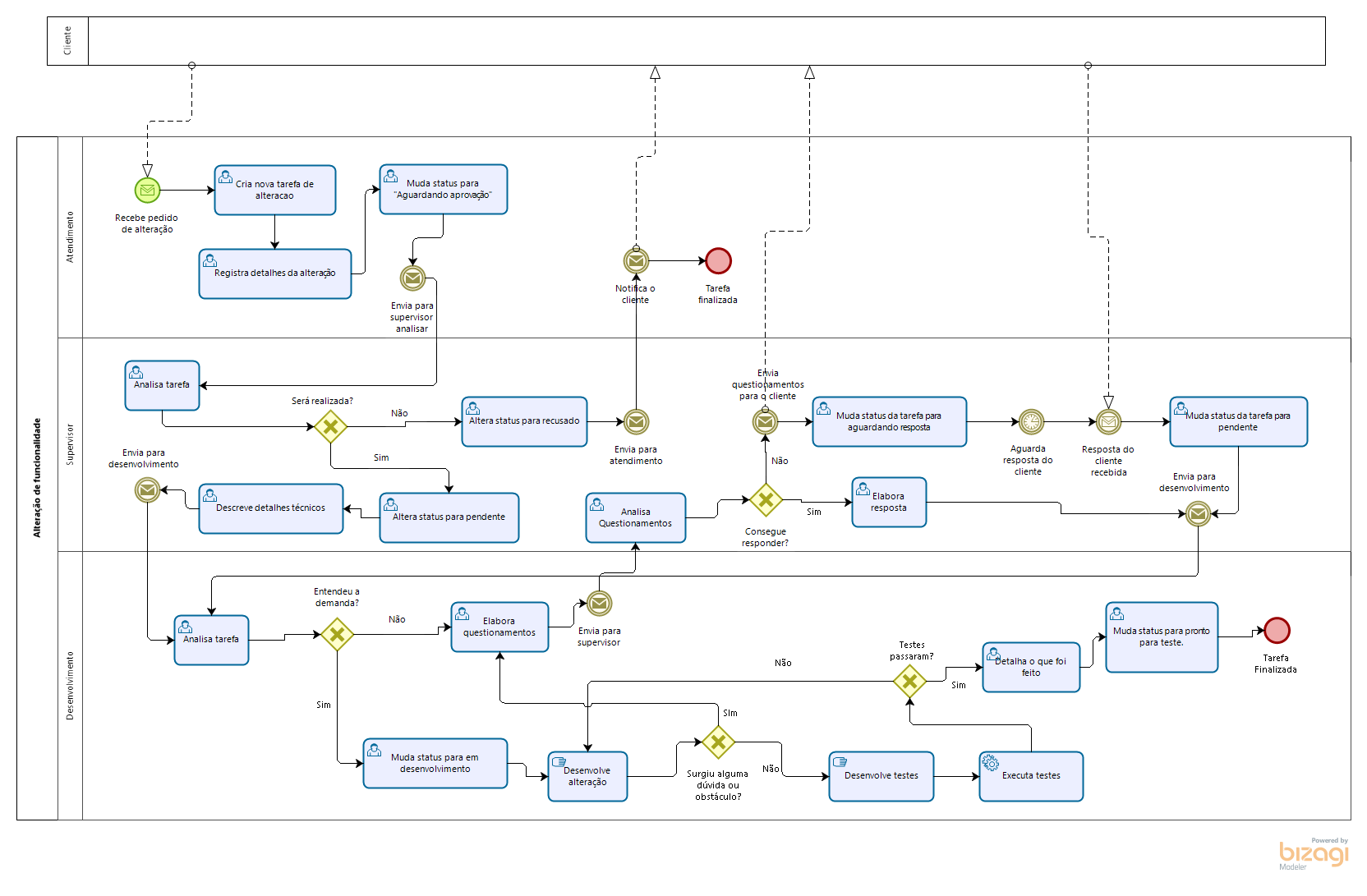
Todas as tarefas serão adicionadas no sistema, assim mantendo organização das tarefas atuais e mantendo um histórico das alterações que foram realizadas no passado. As tarefas vão seguir um fluxo de acordo com sua situação (status) atual. Exemplo: pendente, concluída, aguardando resposta.

## 4.3. Modelagem dos processos

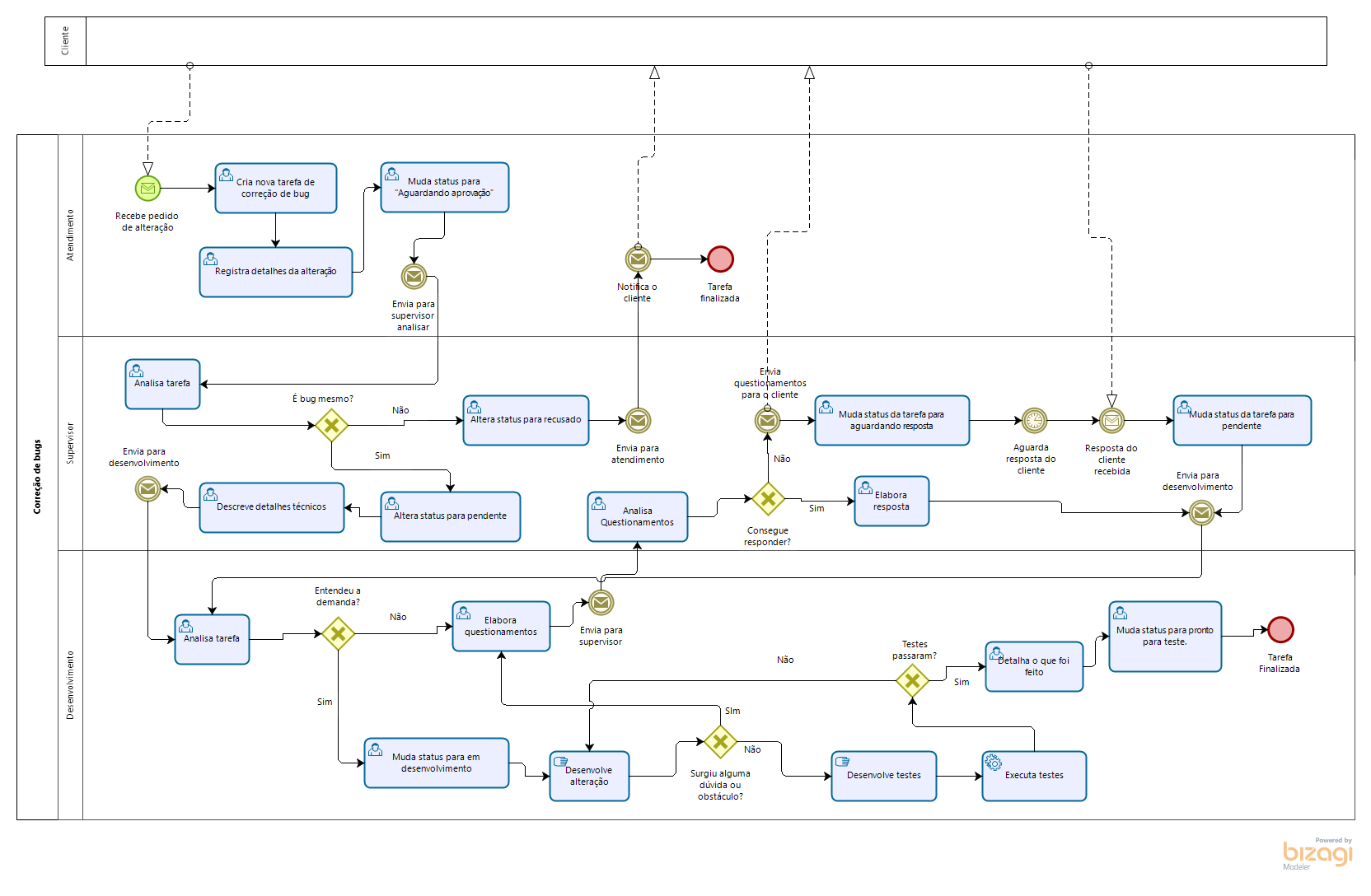
Segue abaixo a proposta de melhoria nos processos mostrados acima.

## 4.3.1 Processo 1 – Alteração de Funcionalidade

Como descrito acima, as tarefas serão registradas no sistema, garantindo que todas as partes fiquem cientes de seu andamento, fazendo com que não dependa mais da memória dos envolvidos. A tarefa terá uma análise muito mais detalhada, e em caso de questionamentos não previstos o cliente pode ser contactado.

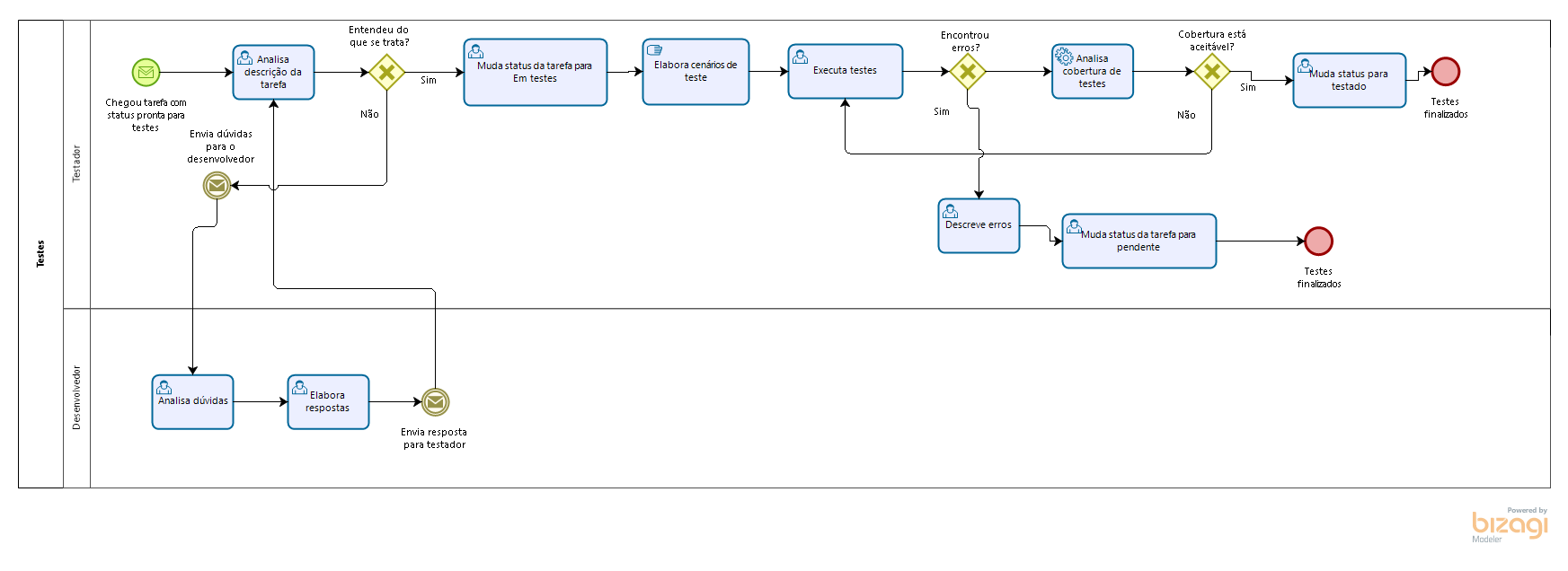


## 4.3.2 Processo 2 – Correção de Bugs Uma breve visualização dos modelos entre AS IS e TO BE, já podemos notar uma drástica mudança no processo de correção de bugs. A análise é muito mais detalhada, gerando uma maior chance de entendimento do problema. O cliente pode ser questionado durante o processo caso necessário. Os testes são bem mais rígidos e o desenvolvedor tem a possibilidade de pedir ajuda caso tenha dificuldades.



## 4.3.3 Processo 3 – Testes

## O processo proposto para testes é quase completamente diferente da situação atual. Primeiramente o setor de testes tem acesso a tarefa registrada no sistema, podendo verificar com facilidade o detalhamento de tudo que foi feito. Caso mesmo com esse registro, tenha dificuldades em entender os detalhes da tarefa que deve testes, existe a opção de enviar dúvidas para o desenvolvedor responsável pela tarefa. Além disso, os cenários de testes são elaborados com cuidado, os erros são detalhados na tarefa em questão e ainda há uma análise para verificar se a cobertura dos testes foi adequada. Com essas mudanças há chance de um erro passar despercebido diminui drasticamente.

****

**4.4. Indicadores de desempenho**

Segue abaixo propostas de métricas que serão utilizadas para medir a eficiência do projeto proposto.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Indicador** | **Objetivos** | **Descrição** | **Cálculo** | **Fonte dados** | **Perspectiva** |
| Porcentagem de tarefas reabertas | Avaliar quantitativamente a quantidade de retrabalho. | Percentual de tarefas reabertas em relação ao total de tarefas fechadas. | Tarefas reabertas / (tarefas fechadas \* 100) | Tabelas tarefas e tarefas\_status. | Processos internos |
| Média de tempo por tarefa | Avaliar melhor a velocidade de trabalho da equipe. | Tempo médio de uma tarefa desde sua criação até quando foi finalizada. | Somatória de data final – data inicial divido pelo número de tarefas | Tabelas tarefas e tarefas\_status. | Processos internos |
| Média de horas de trabalho por tarefa | Avaliar melhor a velocidade de trabalho da equipe. | Tempo médio de horas usadas numa tarefa desde sua criação até quando foi finalizada. | Somatória da quantidade de horas gastas nas tarefas divido pelo total de tarefas. | Tabelas tarefas, tarefas\_status e tabela tarefas\_horas. | Processos internos |
| Porcentagem de tarefas recusadas | Avaliar a taxa de aprovação de tarefas. | Percentual de tarefas que foram recusadas. | Quantidade de tarefas recusadas / total de tarefas | Tabelas tarefas e tarefas\_status. | Processos internos |
| Porcentagem de tarefas que voltou para o cliente | Avaliar o entendimento inicial da descrição das tarefas. | Percentual de tarefas que após aprovação retornou para o cliente com questionamentos. | Quantidade de tarefas retornadas / total de tarefas | Tabelas tarefas e tarefas\_status. | Processos internos |