

CONTINUAÇÃO SOBRE BPMN

Elementos de BPMN para Modelagem de Processos: principais símbolos e regras

Seria extremamente longo apresentar todos os ícones e símbolos BPMN 2.0. Por isso, vamos apresentar apenas alguns deles, divididos em 4 tipos principais que são aplicados no desenho de processos BPMN:

Conectores: elementos de ligação da sequência dos fluxos de trabalho.

Atividades: representam o trabalho que será realizado.

Gateways: mostram a ramificação e a reunião do fluxo de tarefas.

Eventos: indicam eventos exteriores ao processo que o influenciam.

1- Conectores:

Ligam diferentes elementos em um fluxo BPMN.

Fluxo de sequência das atividades:



2- Atividades

São as tarefas que devem ser realizadas. Podem ser discriminadas com pequenos símbolos BPMN no canto superior esquerdo do retângulo que as representam.

Tarefa simples: Uma tarefa representa um trabalho realizado no processo. Pode ter associado um formulário para entrada de dados.



Tarefa de regra de negócio: Aciona uma regra de negócio que retorna um valor para comparação. Pode ser realizado por meio de uma chamada de web service.



3- Eventos

Para todos os tipos de eventos, via de regra, uma linha fina de contorno significa início (pode ser usada também a cor verde), uma linha dupla mostra um evento intermediário (pode ser usada a cor azul) e uma linha grossa indica um evento final (pode ser usada a cor vermelha), com algumas exceções em que não haveria sentido em ter um evento final desse tipo.

É importante lembrar que dependendo do evento, diferentes consequências ocorrerão tanto no início, como na fase intermediária ou final do processo, segundo a notação BPMN 2.0.

Como exemplo, vejamos os eventos referentes a início de processos:

Início de processo simples. Normalmente é utilizado para representar o início manual de um processo.



Fim (genérico): É uma simples marcação do fim do processo e finaliza a “token” do fluxo. Vale lembrar que um processo pode conter 1 ou mais “tokens”, que são caminhos em paralelo.



4- Gateways

Exclusivo: o fluxo segue por apenas um dos fluxos de saída. Pode ser utilizado para representar um desvio no fluxo.



Exclusivo

Paralelo: o fluxo se divide em outros que ocorrem em paralelo.



Paralelo

Inclusivo: o fluxo segue por uma condição inclusiva, ou seja, para cada fluxo de sequência de saída é avaliada uma fórmula e, se for retornado o valor verdadeiro, então o caminho é ativado. Este tipo normalmente demanda o acréscimo de um segundo gateway inclusivo para representar a sincronização. Falamos sobre isso no nosso curso de modelagem BPMN.



Inclusivo

Complexo: Controla condições complexas de divergência e também convergência.



Complexo

Intermediário exclusivo baseado em eventos: usado sempre para dividir o fluxo iniciando um processo devido a ocorrência exclusiva de um de múltiplos eventos. É muito usado para receber uma mensagem a partir de outra piscina.



Evento

Inicial exclusivo baseado em eventos: usado sempre para dividir o fluxo iniciando um processo devido a ocorrência exclusiva de um de múltiplos eventos.

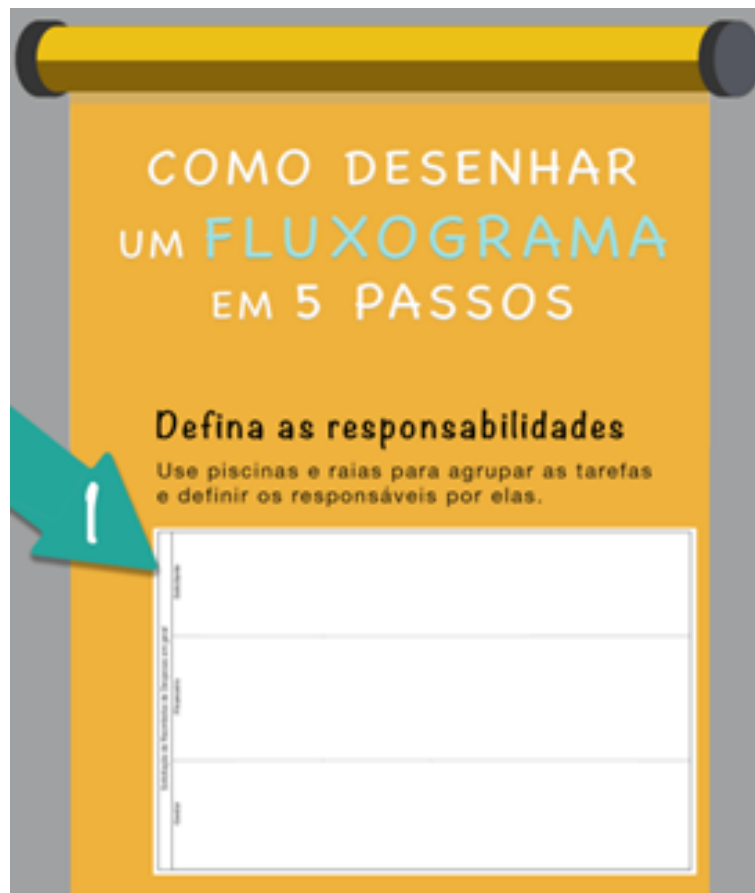


Evento inicial exclusivo

Inicial paralelo: usado para iniciar um processo a partir de vários eventos. Nesse caso todos os eventos devem ocorrer para que seja gerada a instância do processo.

As imagens abaixo mostram o quê devemos observar no momento de desenhar um fluxograma BPMN.

Passo 1



Passo 2



Passo 3



Passo 4 e 5



INTRODUÇÃO À ENGENHARIA DE REQUISITOS

Engenharia de requisitos

Qualquer projeto precisa de requisitos bem definidos para que realmente possa atender as necessidades a que se destina.

No desenvolvimento de softwares, essa realidade não é diferente: todas as partes interessadas devem ter clareza e estar de acordo em relação ao que o sistema deve ter para que seus principais requisitos sejam cumpridos. Em poucas palavras, são os requisitos que guiam o andamento de um projeto.

Por isso, mais que definir os problemas e soluções que irão guiar o processo de criação, conhecer os requisitos garante o entendimento de todos os atores, as funcionalidades e faz com que o processo de desenvolvimento seja mais rápido e econômico.

Sendo assim, podemos dizer que os requisitos servem como base para o planejamento do projeto, bem como para o seu gerenciamento de riscos, controle de mudanças e até testes de aceitação.

Requisitos mal organizados, relatados de forma inadequada e sem expectativas realistas são diretamente responsáveis pelos problemas e falhas nos projetos de softwares.

Em suma, é possível afirmar que o serviço de engenharia de requisitos é um processo que gerencia todas as exigências para que um sistema seja desenvolvido e sempre se mantenha alinhado com as necessidades e demandas dos usuários.

No próximo item, entenda como esse serviço funciona e quais são as suas principais etapas.

Entenda como o serviço funciona

O serviço de engenharia de requisitos inclui algumas atividades básicas, divididas em etapas que visam padronizar o gerenciamento do projeto e garantir o seu sucesso. São elas:

Levantamento dos Requisitos

Esse é o momento inicial, em que são levantadas as necessidades dos usuários do software, além de informações gerais, como as de domínio, sobre sistemas já utilizados na empresa, legislação, entre outras.

O principal objetivo é entender as demandas, processos, restrições e possibilidades dos contratantes e, a partir desses detalhes, elencar os requisitos do sistema.

No levantamento de requisitos, quatro entendimentos básicos devem ser obtidos pela equipe de desenvolvimento:

- Entendimento do Domínio da Aplicação, ou seja, em qual área o software será utilizado;
- Entendimento do Problema, no qual são levantados os detalhes referentes às demandas que o sistema irá sanar;
- Entendimento do Negócio e Atores, que apontam como o software impactará a organização e contribuirá para que ela atinja seus objetivos;
- Entendimento das Necessidades e Restrições dos Interessados, que compreende os processos e funções que serão auxiliados pelo sistema e as demandas para a realização dos trabalhos de seus interessados.

Muitas são as técnicas para que os requisitos sejam levantados. Elas podem incluir entrevistas e questionários entre os interessados, observação do ambiente de trabalho em que ele será aplicado, simulações junto aos usuários finais, etc.

Análise de Requisitos

- Depois que os requisitos são levantados, eles precisam ser analisados para se ter clareza sobre como eles serão utilizados na modelagem do software.

- Além de descrever todos os requisitos em uma linguagem natural, para que todos os envolvidos o compreendam, também é comum a criação de representações gráficas.

- Essas representações servem para melhor demonstrar os processos da organização, os problemas que precisam ser resolvidos e, com base nisso, os melhores meios para o desenvolvimento do sistema.

- Podemos afirmar que a análise de requisitos consiste em uma modelagem conceitual do software, em que meios de análise são desenvolvidos para se obter uma melhor compreensão e especificação do sistema que será criado.

- Essa é uma etapa com menos foco em questões técnicas. Seu centro são questões baseadas em perspectivas conceituais e comportamentais.

- Enquanto a primeira diz respeito aos conceitos e relações de domínio importantes para o desenvolvimento do software, a segunda visa estabelecer como será o comportamento dele e o contexto de suas funcionalidades.

Documentação de Requisitos

- Todos os requisitos e modelos estabelecidos nas fases de levantamento e análise precisam ser descritos e devidamente documentados.

- Isso serve para registrar corretamente e oficializar os resultados do serviço de engenharia de requisitos.

- Com tudo devidamente documentado, os requisitos são mais facilmente comunicados entre os interessados, fornecendo uma fonte realista de estimativas, verificações e validações.

- Descrita em uma linguagem simples e natural, a documentação de requisitos deve ter o propósito do sistema, a descrição dos problemas que ele visa resolver, listas de suas funcionalidades e as regras do negócio.

- Esse tipo de ferramenta é indispensável para garantir pleno alinhamento entre desenvolvedores, clientes, usuários e gerentes.

Verificação, Validação e Garantia da Qualidade

- A documentação de requisitos produzida na última etapa precisa ser **verificada e validada**.

- Enquanto a verificação serve para atestar que o sistema está sendo desenvolvido de maneira adequada, a validação é usada para confirmar que o software que está sendo criado realmente é o certo para os seus objetivos.

- De maneira resumida, é possível afirmar que a verificação confirma que as soluções criadas atendem aos seus requisitos pré-definidos. Já a validação garante que o software criado a partir desses requisitos realmente cumpra suas finalidades.

- A validação demanda uma participação plena dos usuários e do cliente, que são aqueles mais qualificados para confirmar que os propósitos do software realmente estão de acordo com os seus requisitos.

- Assim, é verificado se os requisitos descritos não têm inconsistências, omissões, erros ou conflitos. Os documentos devem estar de acordo com os padrões definidos previamente, trazendo requisitos que satisfaçam as necessidades dos usuários e clientes.

- O principal objetivo dessa etapa é confirmar que os requisitos estão certos, realistas, completos, consistentes e verificáveis.

Gerência de Requisitos

- Desde o levantamento até a própria operação do software, os requisitos podem sofrer mudanças.

- Isso pode ocorrer por conta de erros, omissões, definição de novas prioridades, problemas técnicos, alterações no negócio, no cenário econômico, na concorrência, entre muitos outros fatores relacionados.

- Para que essas alterações não gerem problemas no processo de desenvolvimento, é feito um gerenciamento dos requisitos.

- Esse controle serve para que a equipe responsável possa reconhecer e controlar os requisitos, gerenciando com mais segurança as suas eventuais mudanças em qualquer etapa da criação.

- Nessa fase, são determinados meios para melhor gerenciar alterações nos requisitos, o relacionamento e as dependências entre aqueles que sofreram mudanças, além da atualização de sua documentação.

- Algumas das atividades básicas deste gerenciamento incluem o controle de suas mudanças, controle da versão do sistema, acompanhamento do estado dos requisitos, bem como seu rastreamento.

- Agora que você já conhece as principais etapas que devem ser respeitadas para que o serviço de engenharia de requisitos seja bem-sucedido, descubra no próximo item como a Monitora agrega mais valor aos seus clientes nessa área.

Requisitos funcionais

O que são Requisitos Funcionais: Exemplos, Definição, Guia Completo

O que são requisitos funcionais? Esta é uma pergunta que muitas vezes confunde empresários e desenvolvedores. Um requisito funcional pode ser pensado como um recurso do produto que o usuário detecta. Pode ser um recurso óbvio, como um grande botão Adicionar ao carrinho. Mas também pode ser um recurso menos óbvio,

como calcular corretamente o imposto sobre vendas para a compra online do usuário. Neste guia completo, vamos dividir os requisitos funcionais em suas formas mais simples e dar exemplos de cada tipo. Também definiremos o que cada tipo de requisito significa para o seu negócio e como proceder para criá-los.

O que são Requisitos Funcionais?

Um requisito funcional é uma declaração de como um sistema deve se comportar. Define o que o sistema deve fazer para atender às necessidades ou expectativas do usuário. Os requisitos funcionais podem ser pensados como recursos que o usuário detecta. Eles são diferentes dos requisitos não funcionais, que definem como o sistema deve funcionar internamente (por exemplo, desempenho, segurança, etc.).

Os requisitos funcionais são compostos de duas partes: função e comportamento. A função é o que o sistema faz (por exemplo, “calcular imposto sobre vendas”). O comportamento é como o sistema faz isso (por exemplo, “O sistema deve calcular o imposto sobre vendas multiplicando o preço de compra pela alíquota do imposto.”).

Tipos de Requisitos Funcionais

Aqui estão os tipos de requisitos funcionais mais comuns:

- Regulamentos de Negócios
- Requisitos de Certificação
- Requisitos de relatório
- Funções Administrativas
- Níveis de autorização
- Rastreamento de auditoria
- Interfaces Externas
- Gestão de dados
- Requisitos Legais e Regulamentares

Criando Requisitos Funcionais:

Ao criar requisitos funcionais, é importante ter em mente que eles devem ser específicos, mensuráveis, alcançáveis, relevantes e com prazo determinado (SMART).

Em outras palavras, seus requisitos funcionais devem:

- Seja específico sobre o que o sistema deve fazer
- Seja mensurável para que você possa dizer se o sistema está fazendo isso
- Seja alcançável dentro do prazo que você definiu
- Seja relevante para seus objetivos de negócios
- Seja limitado no tempo para que você possa acompanhar o progresso

Ao seguir essas diretrizes, você pode ter certeza de que seus requisitos funcionais são claros e ajudarão sua equipe de desenvolvimento a criar o produto certo.

Exemplos:

Para lhe dar uma melhor compreensão dos requisitos funcionais, vamos dar uma olhada em alguns exemplos.

Exemplo # 1

:Um usuário deve poder fazer login no sistema usando seu nome de usuário e senha.

Neste exemplo, a função é “login” e o comportamento é “O sistema deve permitir que um usuário faça login usando seu nome de usuário e senha”.

Exemplo # 2

:O sistema deve calcular o imposto sobre vendas para a compra do usuário.

Neste exemplo, a função é “calcular imposto sobre vendas” e o comportamento é “O sistema deve calcular o imposto sobre vendas multiplicando o preço de compra pela alíquota”.

Exemplo # 3

:O sistema enviará um e-mail de confirmação ao usuário após a realização do pedido com sucesso.

Neste exemplo, a função é “enviar e-mail de confirmação” e o comportamento é “O sistema enviará um e-mail de confirmação ao usuário após ele ter feito um pedido com sucesso”.

Como você pode ver, os requisitos funcionais são declarações específicas sobre o que o sistema deve fazer. Eles são diferentes dos requisitos não funcionais, que definem como o sistema funciona internamente (por exemplo, desempenho, segurança, etc.).

Ao criar requisitos funcionais, é importante ter em mente que eles devem ser específicos, mensuráveis, alcançáveis, relevantes e com prazo determinado (SMART). Ao seguir essas diretrizes, você pode ter certeza de que seus requisitos funcionais são claros e ajudarão sua equipe de desenvolvimento a criar o produto certo.

Requisitos não-funcionais

Requisitos não-funcionais são os requisitos relacionados ao uso da aplicação em termos de desempenho, usabilidade, confiabilidade, segurança, disponibilidade, manutenibilidade e tecnologias envolvidas. Não é preciso o cliente dizer sobre eles, pois eles são características mínimas de um software de qualidade, ficando a cargo do desenvolvedor optar por atender esses requisitos ou não.

Demonstram qualidade acerca dos serviços ou funções disponibilizadas pelo sistema. Ex.: tempo, o processo de desenvolvimento, padrões, etc.

Surgem conforme a necessidade dos usuários, em razão de orçamento e outros fatores.

Podem estar relacionados à confiabilidade, tempo de resposta e espaço nas mídias de armazenamento disponíveis.

Caso ocorra falha do não atendimento a um requisito não funcional, poderá tornar todo o sistema ineficaz. Ex.: requisito confiabilidade em um sistema de controle de voos.

Classificação dos Requisitos Não-Funcionais

- Requisitos de produtos: Requisitos que especificam o comportamento do produto.
Ex. portabilidade; tempo na execução; confiabilidade, mobilidade, etc.

- Requisitos da organização: Requisitos decorrentes de políticas e procedimentos corporativos.
Ex. padrões, infraestrutura, etc.

- Requisitos externos: Requisitos decorrentes de fatores externos ao sistema e ao processo de desenvolvimento.
Ex. requisitos de interoperabilidade, legislação, localização geográfica etc.

- Requisitos de facilidade de uso.
Ex.: usuários deverão operar o sistema após um determinado tempo de treinamento.

- Requisitos de eficiência.
Ex.: o sistema deverá processar n requisições por um determinado tempo.

- Requisitos de confiabilidade.
Ex.: o sistema deverá ter alta disponibilidade, por exemplo, 99% do tempo.

- Requisitos de portabilidade.
Ex.: o sistema deverá rodar em qualquer plataforma.

- Requisitos de entrega.
Ex.: um relatório de acompanhamento deverá ser fornecido toda segunda-feira.

- Requisitos de implementação.:
Ex.: o sistema deverá ser desenvolvido na linguagem Java.

- Requisitos de padrões.:
Ex. uso de programação orientada a objeto sob a plataforma A.

- Requisitos de interoperabilidade.:
Ex. o sistema deverá se comunicar com o SQL Server.

- Requisitos éticos.

Ex.: o sistema não apresentará aos usuários quaisquer dados de cunho privativo.

- Requisitos legais.

Ex.: o sistema deverá atender às normas legais, tais como padrões, leis, etc.

- Requisitos de Integração.

Ex.: o sistema integra com outra aplicação.>

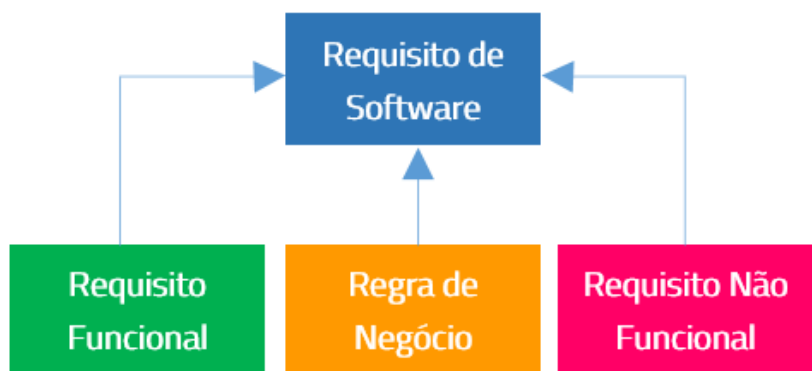
Regra de negócio

Podemos dizer que antes do lançamento do microcomputador o termo regra de negócio era algo interpretado totalmente isolado dos softwares empresariais, ou talvez nem fosse um termo conhecido pelas pessoas.

Nos tempos atuais é difícil encontrar alguém que entende regra de negócio como algo isolado do software. Quando se fala “regra de negócio”, praticamente “sempre” é no contexto de um sistema.

É possível uma empresa mais arcaica viver sem software, mas não consegue viver sem regras de negócio.

Uma RN (Regra de Negócio), no contexto da Engenharia de Software, é tratada como um Requisito de Software, por ser algo que sem ela, o software não existe.



Para ilustrar isso, imaginemos uma empresa que possui um departamento de expedição de materiais.

Este departamento que não possui software para automatizar as atividades deste departamento.

Vejamos a seguir, um pouco sobre este cenário.