

# Engenharia de Software II

Documento e Especificação de requisitos de software

Prof. Alexandre Gomes

# Tarefas de ER

- Elicitação
- Especificação
- Validação/Negociação
- Gerenciamento dos requisitos

# Documento de requisitos de software

- O documento de requisitos de software, às vezes chamado Especificação de Requisitos de Software (SRS- do inglês *Software Requirements Specification*), é uma declaração oficial de o que os desenvolvedores do sistema devem implementar.

# Usuários do documento de requisitos

- A diversidade de possíveis usuários é um indicativo de que o documento de requisitos precisa ser um compromisso com a comunicação dos requisitos para os clientes, a definição dos requisitos em detalhes precisos para os desenvolvedores e testadores e a inclusão de informações sobre a possível evolução do sistema.

# Usuários do documento de requisitos

- Clientes do Sistema
  - Especificam e leem os requisitos para verificar se estes satisfazem suas necessidades. Os clientes especificam as alterações nos requisitos.
- Gerentes
  - Usam o documento de requisitos para planejar uma proposta para o sistema e para planejar o processo de desenvolvimento do sistema.
- Engenheiros do sistema
  - Usam os requisitos para entender o sistema que será desenvolvido.
- Engenheiros de teste de sistema
  - Usam os requisitos para desenvolver testes de validação do sistema.
- Engenheiros de manutenção de sistema
  - Usam os requisitos para entender o sistema e os relacionamentos entre suas partes.

# Detalhes do documento de requisitos

- Sistemas críticos precisam ter requisitos detalhados, porque a segurança e a proteção devem ser analisadas em detalhes.
- Quando o sistema está sendo desenvolvido por terceiros, as especificações de sistema devem ser detalhadas e precisas.
- Se um processo de desenvolvimento iterativo (modelo espiral ou métodos ágeis) é usado, o documento de requisitos pode ser muito menos detalhado e quaisquer ambiguidades podem ser resolvidas durante o desenvolvimento do sistema.

# Importância de um documento de requisitos bem elaborado

- Requisitos formam a base para o desenvolvimento do sistema.
- Requisitos têm relevância legal.
- Documento de requisitos são complexos
- Requisitos devem ser acessíveis para todas as partes envolvidas.

# Especificação de Requisitos

- Um especificação de requisitos é uma coleção de requisitos representada de forma sistemática, tipicamente para um sistema ou componente, atendendo a determinados critérios.
- Os requisitos de usuário e de sistema devem ser claros, inequívocos, de fácil compreensão, completos e consistentes.



# Tipos de Documentação

- Documentação de Requisitos Usando Linguagem Natural
- Documentação de Requisitos Usando Modelos Conceituais
- Documentos de Requisitos Híbridos

# Linguagem Natural

- É a forma de documentação de requisitos mais aplicada na prática.
- Possui a vantagem de que nenhum stakeholder precisa aprender uma nova notação.
- Pode ser utilizada para expressão qualquer requisito.
- Pode apresentar requisitos ambíguos.
- Requisitos de diversos tipos e perspectivas correm o risco de serem misturados.

# Linguagem Natural

- Crie um formato-padrão e garanta que todas as definições de requisitos aderem a esse formato.
- Use uma linguagem consistente para distinguir entre os requisitos obrigatórios e os desejáveis. Os obrigatórios são requisitos aos quais o sistema tem de dar suporte e geralmente são escritos usando-se 'deve' Requisitos desejáveis não são essenciais e são escritos usando-se 'pode'.
- Use uma forma de destacar as partes fundamentais do requisito (negrito, itálico ou cores).
- Não assuma que os leitores compreendem a linguagem técnica da engenharia de software.
- Sempre que possível, tente associar uma lógica a cada um dos requisitos de usuário.

# Exemplo de Requisito em Linguagem Natural

- O sistema deve medir o açúcar no sangue e fornecer insulina, se necessário, a cada dez minutos. (*Mudanças de açúcar no sangue são relativamente lentas, portanto, medições mais frequentes são desnecessárias; medições menos frequentes podem levar a níveis de açúcar desnecessariamente elevados.*)

# Linguagem natural estruturada

- É uma forma de escrever os requisitos do sistema na qual a liberdade do escritor dos requisitos é limitada e todos os requisitos são escritos em uma forma-padrão.
- Essa abordagem mantém grande parte da expressividade e compreensão da linguagem natural, mas garante certa uniformidade imposta sobre a especificação.

# Linguagem natural estruturada

- Pode-se documentar os requisitos de usuário inicialmente escritos em cartões, um requisito por cartão.
- Pode-se colocar um número de campos em cada cartão, algo como a lógica dos requisitos, as dependências de outros requisitos, a origem dos requisitos, materiais de apoio, e assim por diante.

# Exemplo de Especificação Estruturada

Titulo do Requisito	
Função	Ação
Descrição	Requisitos Relacionados
Entradas	Pré-condição
Fonte	Pós-condições
Saídas	Efeitos colaterais
Destino	

# Exemplo de Especificação Estruturada

## Bomba de insulina/Software de Controle/SRS/3.3.2

Função	Calcula doses de insulina: nível seguro de açúcar
Descrição	Calcula a dose de insulina a ser fornecida quando o nível de açúcar está na zona de segurança entre três e sete unidades
Entradas	Leitura atual de açúcar (r2), duas leituras anteriores (r0 e r1)
Fonte	Leitura atual da taxa de açúcar pelo sensor. Outras leituras da memória
Saídas	CompDose – a dose de insulina a ser fornecida
Destino	Loop principal de controle



# Exemplo de Especificação Estruturada

## Bomba de insulina/Software de Controle/SRS/3.3.2

Ação	CompDose é zero se o nível de açúcar está estável ou em queda ou se o nível está aumentando, mas a taxa de aumento está diminuindo. Se o nível está aumentando e a taxa de aumento está aumentando, então CompDose é calculado dividindo-se a diferença entre o nível atual de açúcar e o nível anterior por quatro e arredondando-se o resultado. Se o resultado é arredondado para zero, então CompDose é definida como a dose mínima que pode ser fornecida.
Requisitos	Duas leituras anteriores, de modo que a taxa de variação do nível de açúcar pode ser calculada.
Pré-condição	O reservatório de insulina contém, no mínimo, o máximo de dose única permitida de insulina.
Pós-condições	R0 é substituída por r1 e r1 é substituída por r2.
Efeitos colaterais	Nenhum.

# Modelos Conceituais

- Linguagens especiais de modelagem relacionadas com a perspectiva apropriada devem ser utilizadas.
- O modelo retrata os requisitos de forma muito mais compacta.
- Oferecem um menor grau de ambiguidade.
- Exige conhecimentos específicos de modelagem.

# Modelos Conceituais

- Diagrama de caso de uso:
  - Permite obter uma rápida visão geral das funcionalidades do sistema especificado, e como essas funções se relacionam.
- Diagramas de classes:
  - Documenta requisitos com respeito à estrutura estática dos dados, dependência estruturais entre o sistema e o contexto do sistema.

# Modelos Conceituais

- Diagramas de atividades:
  - Documenta processos de negócio, ou dependências sequenciais do sistema em relação a processos no contexto do sistema. Também podem modelar o caráter sequencial dos casos de uso.
- Diagramas de estados:
  - Documentam comportamentos de um sistema desencadeado por determinados eventos, como foco nos estados individuais em que o sistema pode se encontrar, seus eventos e suas condições que desencadeiam uma transição de estados.

# Documentos Híbridos

- Contêm uma combinação de linguagem natural e modelos conceituais.
- Explora as vantagens de ambos formatos.
- Modelos podem ser complementados por comentários em linguagem natural.

# Quadro de Requisitos Funcionais (TG)

**Quadro 1** – Requisitos Funcionais do sistema

<b>RF001</b> -Cadastro de xxxxxxxxx	Categoria: <input type="checkbox"/> Oculto <input checked="" type="checkbox"/> Evidente	Prioridade: <input checked="" type="checkbox"/> Altíssima <input type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa
<b>Descrição:</b> O sistema deve xxxxxxxxxxxxxxxxxxxx		
<b>RF002</b> -xxxxxxxxxxxxxx	Categoria: <input type="checkbox"/> Oculto <input checked="" type="checkbox"/> Evidente	Prioridade: <input checked="" type="checkbox"/> Altíssima <input type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa
<b>Descrição:</b> O sistema deve xxxxxxxxxxxxxxxxxxxx.		
<b>RF003</b> -Cadastro de x	Categoria: <input type="checkbox"/> Oculto <input checked="" type="checkbox"/> Evidente	Prioridade: <input checked="" type="checkbox"/> Altíssima <input type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa
<b>Descrição:</b> O sistema deve xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx.		

## Categoria

**Oculto:** algo que não é facilmente perceptível. Por exemplo, um bug no código que só ocorre em determinadas condições.

**Evidente:** algo que é facilmente perceptível. Por exemplo, um erro de digitação em um texto.

## Prioridade

**Altíssima:** um problema ou tarefa que deve ser resolvido imediatamente. Por exemplo, um vazamento de dados ou um ataque cibernético.

**Alta:** um problema ou tarefa que deve ser resolvido em breve. Por exemplo, um problema de desempenho ou um bug que está causando problemas aos usuários.

**Média:** um problema ou tarefa que pode ser resolvido em um prazo mais longo. Por exemplo, uma melhoria de desempenho ou uma nova funcionalidade.

**Baixa:** um problema ou tarefa que pode ser resolvido quando houver tempo disponível. Por exemplo, um aprimoramento estético ou uma nova funcionalidade não essencial.

# Quadro de Requisitos Não Funcionais (TG)

**Quadro 2** – Requisitos Não Funcionais do sistema

<b>RNF001-</b> Nome nome <u>nome</u>	O sistema <u>xxxxxxxxxxxxxxxx</u>	Tipo	<u>( )</u> Desejável (X) Obrigatório	(X) Permanente <u>( )</u> Transitório
<b>RNF002-</b> Nome nome <u>nome</u>	<u>xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx</u>	Tipo	<u>( )</u> Desejável (X) Obrigatório	(X) Permanente <u>( )</u> Transitório



## Tipo

**Desejável:** um requisito que não é essencial para o sistema, mas que pode melhorar sua qualidade ou usabilidade. Por exemplo, um requisito que especifica que o sistema deve ter um design intuitivo.

**Obrigatório:** um requisito que é essencial para o sistema e deve ser atendido. Por exemplo, um requisito que especifica que o sistema deve ser seguro.

## Permanência

**Permanente:** um requisito que deve ser atendido pelo sistema durante toda a sua vida útil. Por exemplo, um requisito que especifica que o sistema deve ser acessível a pessoas com deficiência.

**Transitório:** um requisito que só precisa ser atendido por um período de tempo limitado. Por exemplo, um requisito que especifica que o sistema deve ser capaz de suportar um aumento de tráfego.

# Quadro de Regras de Negócio (TG)

*se houver\**

**Quadro 3** – Regras de Negócio do sistema.

<b>RN001 - Nome nome nome</b>
<b>Descrição:</b> Só serão permitidos cadastros xxxxxxxxxxxxxxxx
<b>RN002 - Nome nome nome</b>
<b>Descrição:</b> Apenas serão aceitos xxxxxxxxxxxxxxxx
<b>RN003 - Nome nome nome</b>
<b>Descrição:</b> Serão permitidas xxxxxxxxxxxxxxxxxxxxxxxx

## Exemplos de Regras de Negócio (RN):

As regras de negócio são diretrizes, condições ou restrições que definem como os processos de um negócio devem ser conduzidos e como o sistema deve se comportar em relação a essas diretrizes. Elas garantem que o sistema apoie corretamente as operações e os objetivos da organização.

**Política de Desconto:** O sistema deve aplicar um desconto de 10% para compras acima de R\$ 1.000,00, exceto para produtos já em promoção.

**Limite de Crédito:** Clientes que possuem um limite de crédito maior que R\$ 5.000,00 podem parcelar suas compras em até 12 vezes sem juros.

**Cálculo de Juros:** Para pagamentos com atraso, o sistema deve aplicar uma taxa de 2% de multa mais 0,5% de juros por dia de atraso.

**Horário de Funcionamento:** O sistema só pode registrar pedidos entre 8h e 18h, de segunda a sexta-feira. Pedidos feitos fora desse horário serão processados no próximo dia útil.

**Regras de Frete:** O frete será gratuito para compras acima de R\$ 500,00 para entregas dentro do estado de São Paulo.

## Exemplos de Regras de Negócio (RN):

**Aprovação de Cadastro**: Novos cadastros de fornecedores devem ser aprovados por um gerente antes de serem ativados no sistema.

**Prioridade de Atendimento**: Clientes com status "Premium" têm prioridade no atendimento e devem ser atendidos em até 5 minutos após o contato.

**Validade de Produtos**: O sistema deve bloquear a venda de produtos com data de validade inferior a 30 dias a partir da data atual.

**Controle de Estoque**: Não é permitido que o sistema aceite pedidos de produtos cujo estoque esteja esgotado, a menos que o cliente opte por uma pré-venda.

**Níveis de Acesso**: Apenas usuários com o perfil "Administrador" podem acessar a área de relatórios financeiros e realizar alterações nas configurações do sistema.

**Regras Fiscais**: O sistema deve calcular automaticamente os impostos aplicáveis com base na localização do cliente e na categoria do produto (ex.: ICMS, ISS).

**Política de Cancelamento**: Pedidos podem ser cancelados sem custo até 24 horas após a confirmação. Após esse período, o cancelamento será sujeito a uma taxa de 10% do valor do pedido.

As regras de negócio garantem que o sistema opere de acordo com as necessidades e os processos do negócio, fornecendo uma base sólida para tomadas de decisão automáticas ou manuais.

## Exemplos de Requisitos Não Funcionais (RNF):

**Desempenho:** o sistema deve ser capaz de processar uma determinada quantidade de dados em um determinado período de tempo.

**Segurança:** o sistema deve ser protegido contra acesso não autorizado, uso indevido e ataques cibernéticos.

**Usabilidade:** o sistema deve ser fácil de aprender e usar.

**Acessibilidade:** o sistema deve ser acessível a pessoas com deficiência.

**Mantenabilidade:** o sistema deve ser fácil de manter e atualizar.

**Portabilidade:** o sistema deve ser capaz de ser executado em diferentes plataformas.

**Interoperabilidade:** o sistema deve ser capaz de se comunicar com outros sistemas.

**Escalabilidade:** O sistema deve ser capaz de crescer e lidar com um aumento na carga (usuários, dados, transações) sem comprometer o desempenho.

## Outros exemplos de RNF:

**Confiabilidade**: O sistema deve ser capaz de funcionar de maneira consistente e previsível, minimizando falhas e interrupções.

**Disponibilidade**: O sistema deve estar disponível para uso em uma alta porcentagem do tempo, como 99,9% de disponibilidade (alta disponibilidade).

**Eficiência Energética**: O sistema deve consumir a menor quantidade de recursos de hardware e energia possível.

**Auditabilidade**: O sistema deve permitir auditorias internas e externas para verificar se está em conformidade com normas e regulamentos.

**Latência**: O sistema deve garantir tempos de resposta dentro de limites aceitáveis (tempo entre o envio de uma solicitação e a resposta).

## Outros exemplos de RNF:

**Capacidade de Recuperação:** O sistema deve ser capaz de se recuperar rapidamente de falhas ou interrupções, restaurando dados e operações.

**Modularidade:** O sistema deve ser composto de módulos independentes, facilitando a substituição ou atualização de componentes específicos sem impactar o todo.

**Legalidade/Conformidade:** O sistema deve cumprir todas as regulamentações e leis aplicáveis, como proteção de dados e requisitos de privacidade (por exemplo, GDPR).

**Reusabilidade:** O sistema deve ser projetado de forma que partes dele possam ser reutilizadas em outros sistemas ou projetos.