Requisitos de Software

Engenharia de requisitos

- Estabelece os serviços que o cliente requer de um sistema e as restrições sob as quais tal sistema operará e será desenvolvido.
  - Tais serviços e restrições são chamados de **requisitos**

O que é um requisito?

- Pode ser uma descrição abstrata de alto nível de um serviço, uma restrição de sistema ou até uma especificação matemática, entre outras coisas
- O problema cujo desenvolvimento do sistema deve resolver
  - O sistema tem que ser construído de modo a satisfazer **todos** os seus requisitos

Abstração de requisitos (Davis)

"Se uma empresa deseja estabelecer um contrato para um projeto de desenvolvimento de software de grande porte, deve definir suas necessidades de forma suficientemente abstrata, para que uma solução não esteja pré-definida. Os requisitos devem ser escritos de tal forma que vários fornecedores possam apresentar propostas para o contrato, oferecendo, talvez, diferentes formas de atender às necessidades organizacionais do cliente. Uma vez que o contrato for aprovado, o fornecedor deve escrever uma definição de sistema para o cliente, em mais detalhes, tal que o cliente compreenda e possa validar o que o software irá fazer. Ambos os documentos podem ser chamados de documento de requisitos do sistema."

Tipos de requisitos

- Requisitos de usuário
  - Declarações de alto nível escritas em linguagem natural
  - Escritos para os clientes.
- Requisitos de sistema
  - Um documento estruturado estabelecendo descrições detalhadas das funções, serviços e restrições operacionais do sistema.
  - Define o que deve ser implementado e pode até ser parte de um contrato entre o cliente e o desenvolvedor.

Definições e especificações

Quadro 6.1 Requisitos de usuário e de sistema.

Definição de requisitos de usuário

1. LIBSYS deve manter o acompanhamento de todos os dados exigidos pelas agências de licenciamento de direitos autorais no Reino Unido e em outros lugares.

Especificação dos requisitos de sistema

1.1 Ao solicitar um documento ao LIBSYS, deve ser apresentado ao solicitante um formulário que registra os detalhes do usuário e da solicitação feita.
1.2 Os formulários de solicitação do LIBSYS devem ser armazenados no sistema durante cinco anos, a partir da data da solicitação.
1.3 Todos os formulários do LIBSYS devem ser indexados por usuário, nome do material solicitado e fornecedor da solicitação.
1.4 O LIBSYS deve manter um registro de todas as solicitações feitas ao sistema.
1.5 Para materiais aos quais se aplicam os direitos de empréstimo dos autores, os detalhes do empréstimo devem ser enviados mensalmente às agências de licenciamento de direitos autorais que se registraram no LIBSYS.

Requisitos funcionais e não-funcionais

- **Requisitos funcionais**
  - Serviços que o sistema deve fornecer
  - Como o sistema deve reagir a entradas específicas
  - Como o sistema deve se comportar em determinadas situações.
- **Requisitos não-funcionais ou de qualidade**
  - Restrições sobre serviços ou funções oferecidos pelo sistema tais como restrições de *timing*, restrições sobre o processo de desenvolvimento, padrões, etc.

O sistema LIBSYS

- Um sistema de biblioteca que fornece uma interface única para uma série de banco de dados de artigos em bibliotecas diferentes.
- Os usuários podem pesquisar, baixar e imprimir estes artigos para estudo pessoal.

Exemplos de requisitos funcionais

- O usuário deve ser capaz de pesquisar em todo o conjunto inicial de banco de dados ou selecionar um subconjunto a partir dele.
- Para todo pedido deve ser alocado um identificador único (ORDER_ID) que o usuário possa copiar para a área de armazenamento permanente da sua conta.
- O sistema deve fornecer **telas apropriadas** para o usuário ler os documentos no repositório de documentos.

Imprecisão de requisitos

- Problemas surgem quando os requisitos não são precisamente definidos.
- Requisitos ambíguos podem ser interpretados de maneiras diferentes pelos desenvolvedores e usuários.
- Considere o termo 'telas apropriadas'
  - Intenção do usuário – tela de propósito especial para cada tipo diferente de documento;
  - Interpretação do desenvolvedor – fornece uma tela de texto que mostra o conteúdo do documento.

Requisitos completos e consistentes

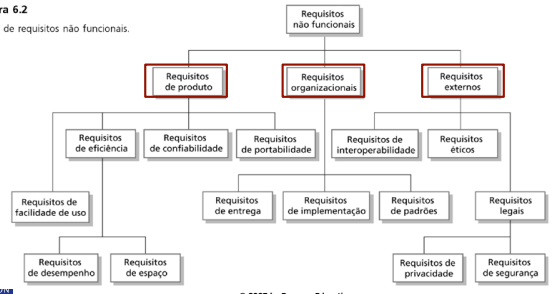
- Em princípio, requisitos devem ser completos e consistentes.
- Completude
  - Eles devem incluir descrições de **todos** os recursos requeridos.
- Consistência
  - Não deve haver **conflitos** ou **contradições** nas descrições dos recursos de sistema.
- Na prática, é impossível produzir um documento de requisitos completo e consistente.

Requisitos não-funcionais

- Definem propriedades e restrições de sistema
- Exemplos incluem confiabilidade, tempo de resposta e requisitos de armazenamento.
- Restrições são capacidade de dispositivos de E/S, representações de sistema, etc.
- Requisitos de processo podem também ser especificados, impondo uma linguagem de programação, IDE ou método de desenvolvimento particular
- Requisitos não-funcionais podem ser mais críticos do que os requisitos funcionais.

Tipos de requisitos não-funcionais

Figura 6.2
Tipos de requisitos não funcionais.



Exemplos de requisitos não- funcionais

Quadro 6.2
Exemplos de requisitos não funcionais.

Requisito de produto
8.1 A interface de usuário para o LIBSYS deve ser implementada como simples HTML, sem frames ou applets de Java.

Requisito organizacional
9.3.2 O processo de desenvolvimento do sistema e os documentos a serem entregues devem estar em conformidade com o processo e produtos a serem entregues definidos em XYZCo-SP-STAN-95.

Requisito externo
10.6 O sistema não deve revelar quaisquer informações pessoais sobre os usuários do sistema ao pessoal da biblioteca que usa o sistema, com exceção do nome e número de referência da biblioteca.

Metas e requisitos

- Requisitos não-funcionais podem ser difíceis de definir precisamente
  - Requisitos imprecisos podem ser difíceis de verificar.
- Meta
  - Uma intenção geral do usuário, tal como facilidade de uso.
- Requisito não-funcional verificável
  - Uma declaração usando alguma medida que pode ser objetivamente testada.
- Metas são úteis para desenvolvedores quando exprimem as intenções dos usuários do sistema.

Exemplo

Quadro 6.3
Metas do sistema e requisitos verificáveis

Meta do sistema
O sistema deve ser fácil de ser usado pelos controladores experientes e ser organizado de modo que os erros dos usuários sejam minimizados.

Requisito não funcional verificável
Os controladores experientes devem ser capazes de usar todas as funções do sistema depois de um treinamento no total de duas horas. Após esse treinamento, o número médio de erros cometidos pelos usuários experientes não deve exceder dois por dia.

Medidas de requisitos

Tabela 6.1 Métricas para especificar requisitos não funcionais

Propriedade	Medida
Velocidade	Transações processadas/segundo Tempo de resposta de usuário/evento Tempo de atualização da tela
Tamanho	Kbytes Número de chips de RAM
Facilidade de uso	Tempo de treinamento Número de frames de ajuda
Confiabilidade	Tempo médio de falha Probabilidade de indisponibilidade Taxa de ocorrência de falhas Disponibilidade
Robustez	Tempo para reiniciar após falha Porcentagem de eventos que causam falhas Probabilidade de corrupção de dados por falhas
Portabilidade	Porcentagem de declarações dependentes do sistema-alvo Número de sistemas-alvo

Interação de requisitos

- Conflitos entre os diferentes requisitos não-funcionais são comuns em sistemas complexos.
- Sistema de aeronave
  - Para minimizar o peso, o número de *chips* separados no sistema deve ser minimizado.
  - Para minimizar o consumo de energia, *chips* de baixa potência devem ser usados.
    - E o desempenho pode ser impactado!
  - Contudo, o uso de *chips* de baixa potência pode significar que mais chips devem ser usados . Qual é o requisito mais crítico?

Requisitos de usuário

- Requisitos funcionais e não-funcionais descritos de modo a ser compreensíveis por usuários que não têm conhecimento técnico detalhado.
- São definidos usando uma linguagem simples, tabelas e diagramas quando estes podem ser compreendidos por todos os usuários.
- Histórias de usuários são similares a requisitos de usuários

Requisito de grade de editor

Quadro 6.6

Requisito de usuário para uma grade de editor.

2.6 Recursos de grade Para ajudar no posicionamento de entidades sobre um diagrama, o usuário pode ativar uma grade em centímetros ou em polegadas por meio de uma opção no painel de controle. Inicialmente, a grade não estará ativada. A grade poderá ser ativada ou desativada a qualquer instante durante uma sessão de edição, e poderá ser alternada entre polegadas e centímetros, a qualquer instante. Uma opção de grade será fornecida na visão reduzida, mas o número de linhas de grade mostradas será reduzido para evitar o preenchimento do diagrama

Diretrizes para escrever requisitos

- Usar um formato padrão para todos os requisitos.
- Usar a linguagem de uma forma consistente.
  - 'deve' para requisitos obrigatórios, e
  - 'deveria' para requisitos desejáveis.
- ***Realçar*** o texto para identificar as partes principais do requisito.
- Evitar o uso de jargões de computação.

Requisitos de sistema

- Especificações mais detalhadas das funções do sistema, dos serviços e das restrições
- Visam forneer ser uma base para o desenvolvimento do sistema
  - Em XP: **histórias de usuário** + **tarefas de desenv**
  - Nosso caso: casos de uso (estilo casual) + especificação detalhada
- Eles podem ser incorporados no contrato de sistema.
- Requisitos de sistema podem ser definidos ou ilustrados usando **notações gráficas**

Requisitos e Projeto

- Requisitos devem definir o que o sistema deve fazer e o projeto deve descrever como ele faz isto.
  - Requisitos => problema
  - Projeto => solução
- Na prática, requisitos e projeto são inseparáveis
  - Uma arquitetura de sistema pode ser projetada para estruturar os requisitos;
  - O sistema pode ter que interoperar com outros sistemas que geram novos requisitos;
  - O uso de uma solução de projeto específica pode ser um requisito de domínio.

Problemas com linguagem natural

- Falta de clareza
  - É difícil atingir uma precisão sem tornar o documento difícil de ler e ambíguo
- Confusão de requisitos
  - Requisitos funcionais e não-funcionais tendem a estar misturados.
- Fusão de requisitos
  - Vários requisitos diferentes podem ser expressos juntos
- Dificuldade de estruturar a especificação

Alternativas à especificação em linguagem natural

Tabela 6.2 Notações para especificação de requisitos

Notação	Descrição
Linguagem natural estruturada	Esta abordagem depende da definição de formulários ou templates-padrão para expressar a especificação de requisitos.
Linguagens de descrição de projeto	Esta abordagem usa uma linguagem semelhante à linguagem de programação, porém com mais características abstratas, para especificar os requisitos por meio da definição de um modelo operacional do sistema. Essa abordagem não é amplamente usada hoje em dia, embora possa ser útil para especificações de interfaces.
Notações gráficas	Uma linguagem gráfica, complementada com anotações de texto é usada para definir os requisitos funcionais do sistema. Um antigo exemplo dessa linguagem gráfica é SADT (Ross, 1977) (Schoman e Ross, 1977). Atualmente, as descrições de casos de uso (Jacobsen, et al., 1993) e os diagramas de seqüência são comumente usados (Stevens e Pooley, 1999).
Especificações matemáticas	São notações baseadas em conceitos matemáticos, como máquinas de estados finitos ou conjuntos. Essas especificações não ambíguas reduzem discussões entre cliente e fornecedor. No entanto, a maioria dos clientes não compreende as especificações formais e são relutantes em aceitá-las no momento da contratação.

Especificações em linguagem estruturada

- A liberdade do elaborador de requisitos é limitada por um *template* pré-definido para requisitos.
- Todos os requisitos são escritos de maneira padronizada.
- A terminologia usada na descrição pode ser limitada.
- A vantagem é que a maior parte da expressividade da linguagem natural é mantida
  - Mas o grau de uniformidade é imposto na especificação.

Apresentação estruturada

Quadro 6.7

Definição de um recurso de grade de editor.

2.6.1 Recursos de grade

O editor deve fornecer um recurso de grade no qual uma matriz de linhas horizontais e verticais fornece um fundo para a janela do editor. Essa grade deve ser passiva e o alinhamento das entidades é de responsabilidade do usuário.

Justificativa lógica: Uma grade ajuda o usuário a criar um diagrama bem organizado com entidades bem espaçadas. Embora uma grade ativa, na qual as entidades 'saltam' as linhas de grade, possa ser útil, o posicionamento é impreciso. O usuário é a melhor pessoa para decidir onde as entidades devem ser posicionadas.

Especificação: ECLIPSE/WSTools/DE/FS Seção 5.6
Fonte: Ray Wilson, escritório de Glasgow

Especificação baseada em formulário

Tabela 6.3 Especificação de requisitos do sistema com utilização de um formulário-padrão

Bomba de insulina/Software de controle/SRS/3.3.2	
Função	Calcular dose de insulina: nível seguro de açúcar
Descrição	Calcula a dose de insulina a ser liberada quando o nível medido de açúcar atual está na zona segura entre 3 e 7 unidades
Entradas	Leitura atual de açúcar (r2), as duas leituras anteriores (r0 e r1)
Origem	Leitura atual de açúcar do sensor. Outras leituras da memória
Saídas	CompDose — a dose de insulina a ser liberada
Destino	Loop de controle principal
Ação:	CompDose será zero se o nível de açúcar estiver estável ou em queda, ou se o nível estiver aumentando, mas a taxa de aumento estiver diminuindo. Se o nível estiver aumentando e a taxa de aumento estiver aumentando, então CompDose será calculado dividindo-se a diferença entre o nível atual de açúcar e o nível anterior por 4, e arredondando o resultado. Se o resultado do arredondamento for zero, então CompDose será definido como a dose mínima que pode ser liberada.
Requer	Duas leituras anteriores de modo que a taxa de mudança do nível de açúcar possa ser calculada.
Precondição	O reservatório de insulina conter, pelo menos, o máximo de dose única permitida de insulina.
Pós-condição	r0 é substituído por r1, portanto r1 é substituído por r2
Efeitos colaterais	Nenhum

Especificação tabular

- Usada para suplementar a linguagem natural.
- Particularmente útil quando você tem de definir uma série de possíveis cursos alternativos de ação.

Especificação tabular

Tabela 6.4 Especificação tabular de cálculo

Condição	Ação
Nível de açúcar em queda (r2 < r1)	CompDose = 0
Nível de açúcar estável (r2 = r1)	CompDose = 0
Nível de açúcar aumentando e taxa de aumento diminuindo ((r2 – r1) < (r1 – r0))	CompDose = 0
Nível de açúcar aumentando e taxa de aumento estável ou aumentando ((r2 – r1) > (r1 – r0))	CompDose = arredondar ((r2 – r1)/4) Se o resultado do arredondamento for = 0 então CompDose = DoseMínima

O documento de requisitos

- O documento de requisitos é a declaração oficial do que é requisitado pelos desenvolvedores do sistema.
  - Em XP é um pouco **diferente**
- Deve incluir ambos, uma definição dos requisitos de usuário e uma especificação dos requisitos de sistema.
- **NÃO É** um documento de projeto.
  - Logo que possível, será preciso definir **como** o sistema deve fazer, ao invés de **o que** deve ser feito.

Usuários de um documento de requisitos

Figura 6.5
Usuários de um documento de requisitos.