



0/43

Engenharia de Sistemas de Informação I

Requisitos de Software

Delano M. Beder & Marcos L. Chaim

Escola de Artes, Ciências e Humanidades – EACH
Universidade de São Paulo – USP



Engenharia de Requisitos



1/43

- Os requisitos são descrições dos serviços fornecidos pelo sistema e as suas restrições operacionais
- Esses requisitos refletem as necessidades dos clientes de um sistema que ajuda a resolver algum problema
- O processo de descobrir, analisar, documentar e verificar esses serviços e restrições é chamado de *engenharia de requisitos*



O que é um requisito ?

- o termo *requisito* não é usado pela indústria de software de maneira consistente
- Em alguns casos, um requisito é simplesmente uma declaração abstrata de alto nível de um serviço que o sistema deve fornecer ou uma restrição do sistema
- No outro extremo, é uma definição formal e detalhada de uma função do sistema



Requisitos



3/43

Davis, A.M. (1993). *Software requirements: objects, functions & states*. Englewood Cliffs, NJ: Prentice Hall.

"Se uma empresa deseja estabelecer um contrato para o desenvolvimento de um grande projeto de software, ela precisa definir suas necessidades de maneira suficientemente abstrata, para que uma solução não seja predefinida. Os requisitos devem ser redigidos de modo que os diversos fornecedores possam apresentar propostas, oferecendo, talvez, diferentes maneiras de atender às necessidades organizacionais do cliente. Após a aprovação do contrato, o fornecedor deve redigir uma definição mais detalhada do sistema para o cliente, de modo que o cliente possa compreender e validar o que o software fará. Esses dois documentos podem ser chamados de documentos de requisitos do sistema"



Tipos de Requisitos

- *Requisitos de usuário*

- Declarações, em uma linguagem natural com diagramas, de quais serviços são esperados do sistema e as restrições sob as quais ele deve operar.
- Público alvo: Clientes

- *Requisitos de sistema*

- Definem, detalhadamente, as funções, os serviços e as restrições operacionais do sistema. O documento de requisitos de sistema (às vezes chamado de especificação funcional) deve ser preciso. Ele deve definir exatamente o que será implementado.
- Pode ser parte do contrato entre o comprador e os desenvolvedores de software.



Requisitos funcionais e não funcionais

- *Requisitos funcionais.* São declarações de serviços que o sistema deve fornecer, como o sistema deve reagir a entradas específicas e como o sistema deve se comportar em determinadas situações. Em alguns casos, os requisitos funcionais podem também estabelecer explicitamente o que o sistema não deve fazer.
- *Requisitos não funcionais.* São restrições sobre os serviços ou as funções oferecidos pelo sistema. Eles incluem restrições de tempo, restrições sobre o processo de desenvolvimento e padrões. Os requisitos não funcionais aplicam-se, freqüentemente, ao sistema como um todo.
- *Requisitos de domínio.* São requisitos provenientes do domínio da aplicação do sistema e que refletem as características e as restrições desse domínio. Podem ser requisitos funcionais ou não funcionais.



Requisitos funcionais (RF)

- Os requisitos funcionais de um sistema descrevem o que o sistema deve fazer.
- Esses requisitos dependem do tipo do software que está sendo desenvolvido, dos usuários a que o software se destina e da abordagem geral considerada pela organização ao redigir os requisitos.
- Quando expressos como requisitos de usuário, eles são geralmente descritos de forma bastante abstrata.
- No entanto, os requisitos funcionais de sistema descrevem a função do sistema detalhadamente, suas entradas e saídas, exceções etc.



Exemplo de RFs: sistema LIBSYS



7/43

1. O usuário deve ser capaz de fazer uma busca em todo o conjunto inicial do banco de dados ou selecionar um subconjunto com base nele.
2. O sistema deve fornecer telas apropriadas para o usuário ler os documentos no repositório de documentos.
3. Para cada pedido, deve ser alocado um único identificador (ORDER_ID), o qual o usuário deve ser capaz de copiar para a área de armazenamento permanente da conta.



Imprecisão na especificação de requisitos –

- A imprecisão na especificação de requisitos é o motivo de muitos problemas de engenharia de software
- É natural que um desenvolvedor de sistema interprete um requisito ambíguo de modo a simplificar sua implementação.
- Considere o termo "telas apropriadas" (requisito 2 - slide anterior)
 - *Intenção do usuário* - Telas de diferentes formatos de arquivo estejam disponíveis
 - *Interpretação do desenvolvedor (sob pressão do cronograma)* - Simplesmente fornecer uma tela de texto e argumentar que o requisito foi atendido.



Especificação completa e consistente

- Em princípio, a especificação de requisitos funcionais de um sistema deve ser completa e consistente.
- *Completa*
 - Todos os serviços exigidos pelo usuário devem ser definidos
- *Consistente*
 - Os requisitos não devem ter definições contraditórias

Na prática, em sistemas grandes e complexos, é praticamente impossível atingir a consistência e completude de requisitos.



Requisitos não funcionais (RNF)

- São aqueles não diretamente relacionados às funções específicas fornecidas pelo sistema.
 - Eles podem estar relacionados às propriedades emergentes do sistema, como confiabilidade, tempo de resposta e espaço de armazenamento.
 - Como alternativa, eles podem definir restrições, como a capacidade dos dispositivos de E/S (entrada/saída) e as representações de dados usadas nas interfaces do sistema.
- RNFs são geralmente mais importantes do que os RFs individuais
 - Os usuários do sistema em geral encontram meios de contornar uma função do sistema que não atenda às suas necessidades
 - Contudo, uma falha no atendimento de um RNF pode significar que todo o sistema é inútil



Classificação dos requisitos não funcionais

- *Requisitos do produto.* Estes requisitos especificam o comportamento do produto. Entre os exemplos estão requisitos de desempenho, requisitos de confiabilidade, requisitos de portabilidade e requisitos de usabilidade.
- *Requisitos organizacionais.* Estes requisitos são derivados de políticas e procedimentos da organização do cliente e do desenvolvedor. Alguns exemplos incluem padrões de processo que podem ser usados, requisitos de implementação, como a linguagem de programação, e requisitos de entrega.
- *Requisitos externos.* Este título amplo abrange todos os requisitos derivados de fatores externos ao sistema e seu processo de desenvolvimento.



Exemplos de Requisitos não funcionais

- **Requisito de produto.** A interface de usuário para o LIBSYS deve ser implementada como simples HTML, sem frames ou applets Java.
- **Requisito organizacional.** O processo de desenvolvimento do sistema e os documentos entregues devem estar em conformidade com o processo e produtos a serem entregues definidos em XYZCo-SP-STAN-95.
- **Requisito externo.** 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 X Requisitos

- Um problema comum com requisitos não funcionais é que eles podem ser difíceis de verificar.
- Os usuários ou os clientes freqüentemente definem esses requisitos como metas gerais, como usabilidade, capacidade do sistema de se recuperar de falhas ou resposta rápida ao usuário.
- Meta
 - Uma intenção genérica do usuário tal como facilidade de uso.
- Requisito não funcional verificável
 - Descrição usando alguma métrica que pode ser objetivamente testada.



Exemplos

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.



Métricas de requisitos



15/43

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 falha Disponibilidade
Robustez	Tempo de 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



Requisitos

- Os requisitos não funcionais frequentemente entram em conflito e interagem com outros requisitos funcionais e não funcionais.
- **Requisito.** limite da memória usada por um sistema é 4M.
 - Comuns em sistemas embutidos, no qual o espaço ou o peso é limitado e o número de chips ROM que armazenam o software do sistema deve ser mínimo.
- **Requisito.** o sistema deve ser implementado em Ada, uma linguagem de programação para desenvolvimento de software crítico de tempo real.
 - Pode não ser possível compilar um programa em Ada com a funcionalidade necessária em menos de 4 Mbytes
- Compromisso: linguagem de programação alternativa ou mais memória ?



Requisitos de domínio

- Derivados do domínio de aplicação do sistema e descrevem características e funcionalidades do sistema que refletem o domínio.
- Podem ser novos requisitos funcionais em si mesmos, podem restringir os requisitos funcionais existentes ou estabelecer como cálculos específicos devem ser realizados.
- Os requisitos de domínio são importantes porque, com frequência, refletem os fundamentos do domínio da aplicação.
 - Se esses requisitos não forem satisfeitos, pode ser impossível fazer o sistema funcionar satisfatoriamente.



Problemas de requisitos de domínio

- Entendimento
 - Requisitos são redigidos na linguagem do domínio da aplicação;
 - Geralmente os engenheiros de software têm dificuldade em compreendê-los.
- *Implicitness*
 - Os especialistas do domínio podem deixar determinadas informações fora de um requisito, simplesmente por serem muito óbvias para eles.
 - Contudo, pode não ser óbvia para os desenvolvedores do sistema e eles podem, portanto, implementar o requisito de forma equivocada.



Requisitos de usuário

- Devem descrever os requisitos funcionais e não funcionais, de modo que eles sejam compreensíveis pelos usuários do sistema que não possuem conhecimento técnico detalhado
 - Eles devem especificar apenas o comportamento externo do sistema e evitar, sempre que possível, características de projeto do sistema.
 - Conseqüentemente, se você estiver escrevendo requisitos de usuário, você não deve usar jargões de software, notações estruturadas ou formais ou descrever os requisitos por meio da implementação do sistema.
 - Você deve escrever os requisitos de usuário em linguagens simples, com tabelas e formulários simples e diagramas intuitivos.



Problemas com a linguagem natural

- *Falta de clareza.* às vezes, é difícil usar a linguagem de maneira precisa e não ambígua sem tornar o documento prolixo e difícil de ler.
- *Confusão de requisitos.* Requisitos funcionais, requisitos não funcionais, metas do sistema e informações de projeto podem não estar claramente diferenciados.
- *Fusão de requisitos.* Diversos requisitos diferentes podem ser expressos juntos como um único requisito.



Guidelines para escrever requisitos

- Invente um formato padrão e assegure-se de que todas as definições de requisitos aderiram a esse formato. A padronização de formato torna as omissões menos prováveis e os requisitos mais fáceis de serem verificados.
- Use a linguagem de forma consistente. Você deve sempre fazer distinção entre requisitos obrigatórios e desejáveis. Os *Requisitos obrigatórios* são aqueles a que o sistema deve atender e são escritos com o uso da palavra "deve". *Requisitos desejáveis* não são essenciais e são escritos com o uso da palavra "pode".
- Use destaque no texto (negrito, itálico ou cor) para ressaltar as partes principais do requisito.
- Evitar, sempre que possível, o uso de jargões de informática. Contudo, inevitavelmente aparecerão termos técnicos detalhados nos requisitos de usuário.



Requisitos de sistema

- Os requisitos de sistema são versões expandidas dos requisitos de usuário usados pelos engenheiros de software
 - Servem como ponto de partida para o projeto do sistema
- Adicionam detalhes e explicam como os requisitos de usuário devem ser fornecidos pelo sistema.
- Podem ser usados como parte do contrato para a implementação do sistema, e devem, portanto, ser uma especificação completa e consistente de todo o sistema.
- Requisitos de sistema devem ser definidos ou ilustrados usando modelos de sistema (discutido nas próximas aulas).



Requisitos e projeto

- Em princípio, requisitos devem simplesmente descrever o quê o sistema faz e o projeto deve descrever como ele faz. Eles não devem estar relacionados a como o sistema pode ser projetado.
- Entretanto, para especificar completamente um sistema de software complexo no nível de detalhamento necessário, é impossível, na prática, excluir todas as informações de projeto. Existem várias razões para isso:
 - Você pode ter de projetar uma arquitetura inicial do sistema para auxiliar na estruturação da especificação de requisitos.
 - Na maioria dos casos, os sistemas devem interoperar com outros sistemas existentes. Isso restringe o projeto e essas restrições impõem requisitos do sistema.
 - O uso de uma arquitetura específica para satisfazer os requisitos não funcionais pode ser necessário.



Problemas com a especificação em LN

- A compreensão da linguagem natural depende do uso das mesmas palavras para os mesmos conceitos pelos leitores e elaboradores da especificação. Isso leva a mal-entendidos devido à ambigüidade da linguagem natural.
- Uma especificação de requisitos em linguagem natural é flexível demais. É possível dizer a mesma coisa de maneira completamente diferentes. Fica por conta do leitor descobrir quando os requisitos são os mesmos e quando são distintos.
- Não existe uma maneira fácil de padronizar os requisitos em linguagem natural. Pode ser difícil encontrar todos os requisitos relacionados. Para descobrir as conseqüências de uma mudança, pode ser necessário verificar todos os requisitos em vez de apenas um grupo de requisitos relacionados.



Alternativas para especificação em LN



25/43

Limguagem natural estruturada	Esta abordagem depende da definição de formulários ou templates-padrão para expressas 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 especificação de interfaces
Notações gráficas	Uma linguagem gráfica, complementada com anotações de texto é usada para definir os requisitos funcionais do sistema. Atualmente as descrições de casos de uso e os diagramas de seqüência são comumente usados
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.



Descrições estáticas de requisitos

- Para algumas aplicações pode ser interessante expressar os dados manipulados por uma aplicação por meio da descrição da linguagem aceita pelo sistema utilizando a notação de Backus-Naur.
- Porém, é sempre importante descrever as relações entre os objetos que compõem o sistema. Pode-se utilizar um diagrama conceitual, um dicionário de dados ou um diagrama de Warnier.



Notação Backus-Naur

```
<condition> ::= <bool-term> | <bool-term> or <condition>
<bool-term> ::= <bool-factor> | <bool-factor> and <bool-term>
<bool-factor> ::= <expr> <relop> <expr> | (<condition>)
<relop> ::= < | < | = | > | > | < >
<expr> ::= <term> | <expr> <addop> <term> | <addop> <expr>
<term> ::= <factor> | <term> <mpyop> <factor>
<factor> ::= <scaled-expr> | <primary>
<scaled-expr> ::= (<expr>) <scale> | <number> <scale>
<primary> ::= (<expr>) <regname> | <number> | <func> (<expr>)
<number> ::= <integer> | <integer>. | .<integer> | <integer>.<integer>
<regname> ::= $ <regchar> | <regname> <regchar>
<integer> ::= <digit> | <digit> <integer>
<regchar> ::= <digit> | <letter> | <underscore>
<addop> ::= + | -
<digit> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
<func> ::= abs | trunc
<letter> ::= A | a | B | b | C | c | D | d | E | e |
. . . | Y | y | Z | z
<myop> ::= * | / | mod
<scale> ::= c | d | h | i | l | P | p | q | t | v
<underscore> ::= _ (ASCII character 95)
```



Modelo Conceitual



28/43

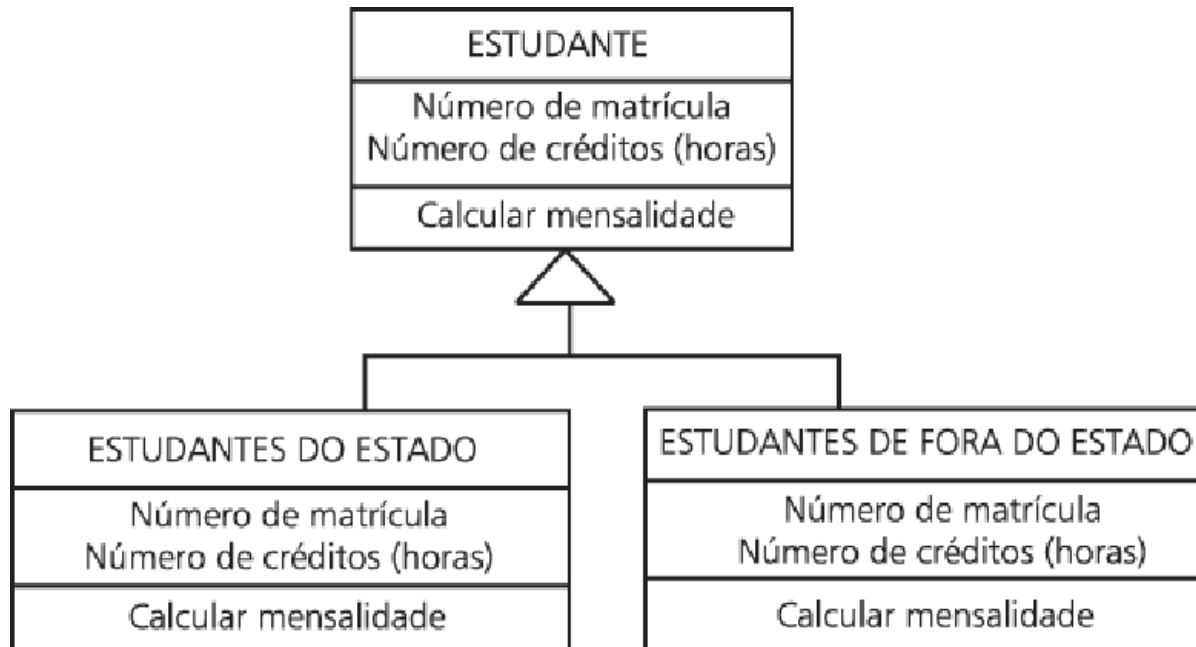
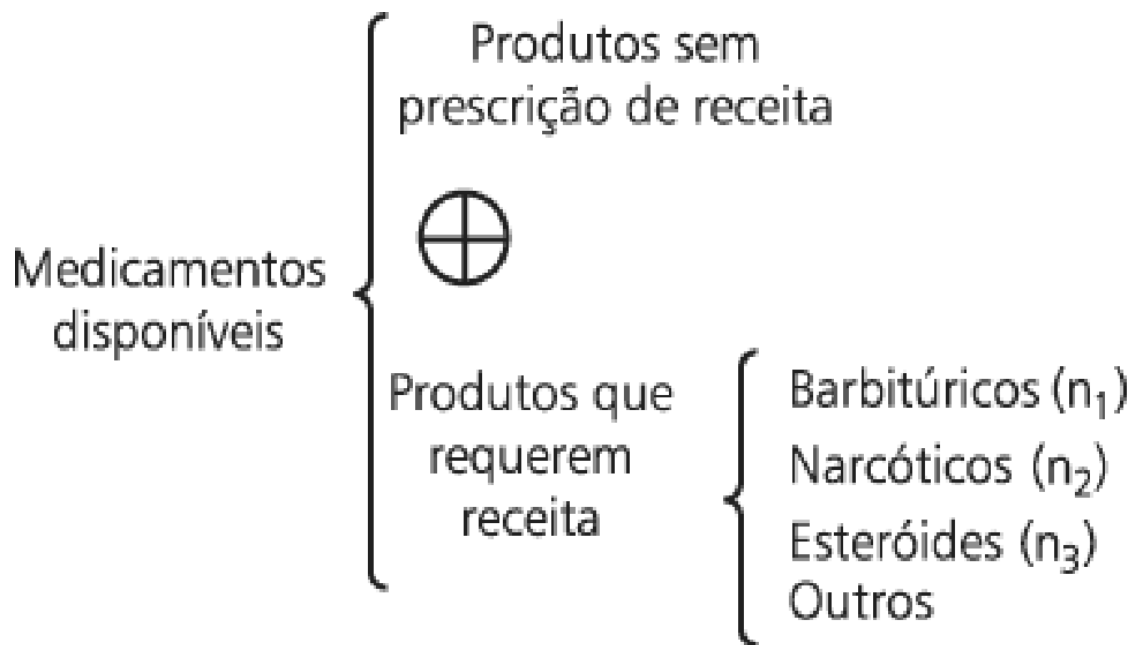


Diagrama de Warnier



29/43



Descrições dinâmicas de requisitos

- Notações puramente estáticas não são suficientes.
- Os sistemas são dinâmicos e requerem que a sua dinâmica seja descrita.
- Exemplos de notação: tabelas de decisão; máquinas de estados finitos; diagrama de estados; diagramas de seqüência, etc.



Tabelas de Decisão

- Usado para suplementar linguagem natural
- As tabelas são particularmente úteis quando há uma série de possíveis situações alternativas e você necessita descrever as ações a serem tomadas a respeito de cada uma delas

TABELA 4.1 Tabela de decisão

	Regra 1	Regra 2	Regra 3	Regra 4	Regra 5
Notas altas nos exames padronizados	V	F	F	F	F
Notas altas (2ª grau)	—	V	F	F	F
Atividades externas	—	—	V	F	F
Boas recomendações	—	—	—	V	F
Enviar carta de não-aceitação			X	X	X
Enviar formulários de admissão	X	X			



Diagrama de Seqüência



32/43

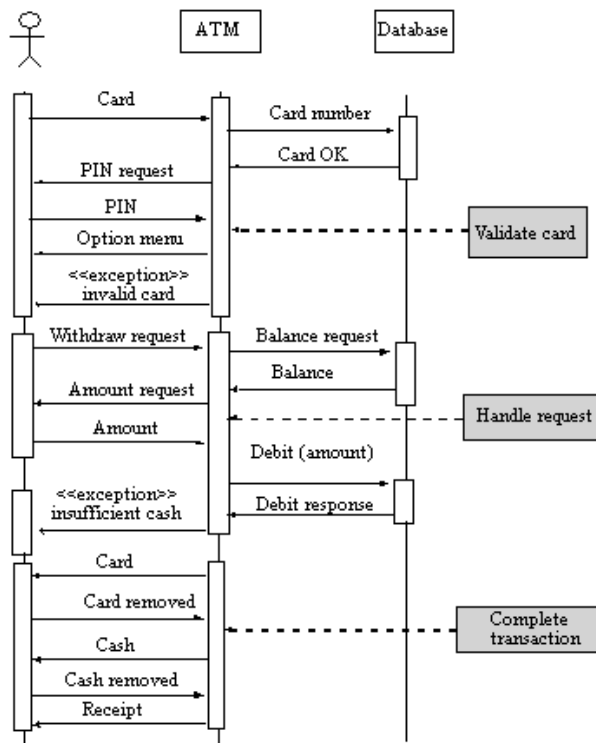
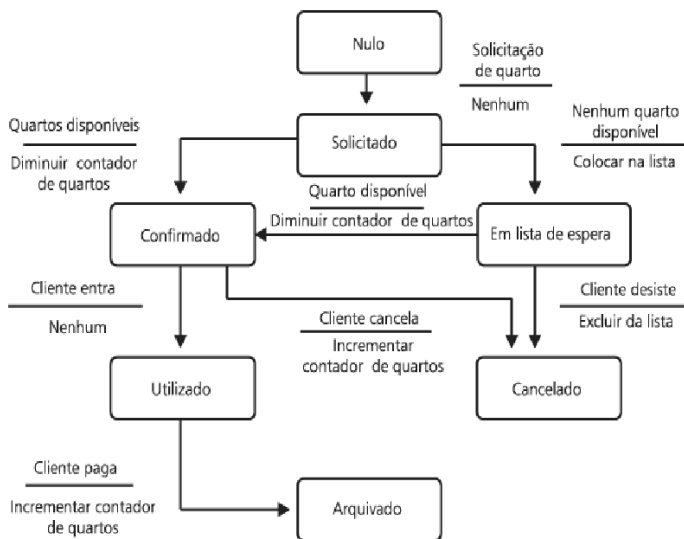


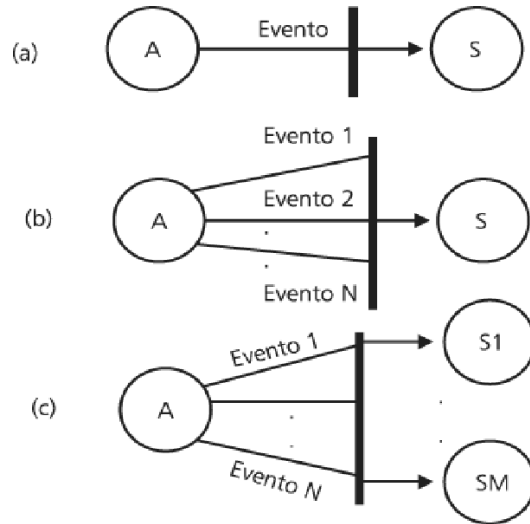
Diagrama de Estados



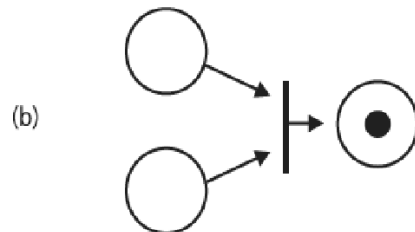
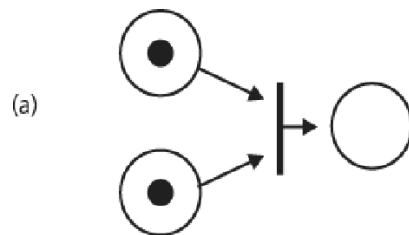
33/43



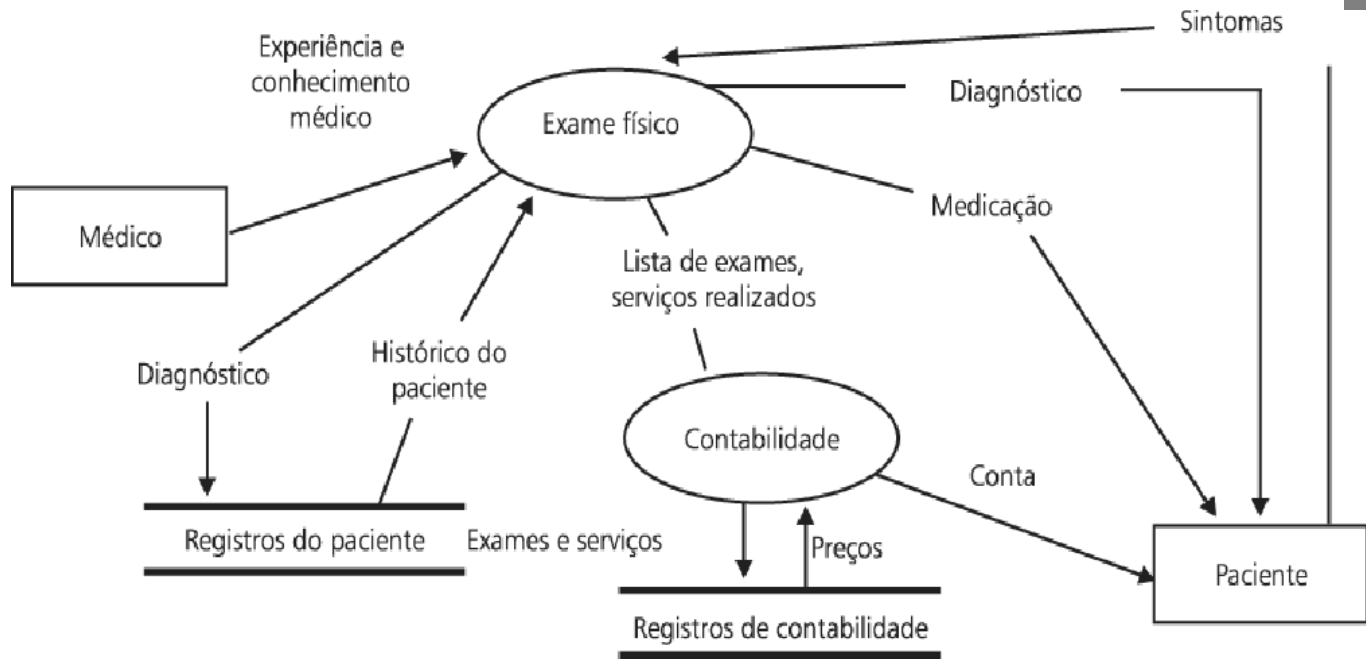
Redes de Petri



Redes de Petri



Outras notações: Diagrama de fluxo de dados



Outras notações: Z



37/43

$ST = \text{Key} \rightarrow \text{VAL}$

INIT

$st' : ST$

$st' = \{\}$

INSERT

$st, st' : ST$

$k : \text{KEY}$

$v : \text{VAL}$

$k \notin \text{dom}(st) \wedge$

$st' = st \cup \{k \mapsto v\}$

LOOKUP

$st, st' : ST$

$k : \text{KEY}$

$v : \text{VAL}$

$k \in \text{dom}(st) \wedge$

$v' = st(k) \wedge$

$st' = st$

DELETE

$st, st' : ST$

$k : \text{KEY}$

$k \in \text{dom}(st) \wedge$

$st' = \{k\} \sqcap st$



O documento de requisitos

- O documento de requisitos de software (algumas vezes chamado de especificação de requisitos de software) é a declaração oficial do que os desenvolvedores de sistema devem implementar.
- Deve incluir os requisitos de usuários de um sistema e uma especificação detalhada dos requisitos de sistema
- O documento de requisitos não é um documento de projeto. Na medida do possível, os requisitos devem simplesmente descrever o QUÊ o sistema faz e não descrever COMO faz.



Usuários de um documento de requisitos _



39/43

Clientes de sistema	Especificam e lêem os requisitos para verificar se eles atendem às suas necessidades. Os clientes especificam as mudanças no requisitos.
Gerentes	Usam o documento de requisitos para planejar um pedido de proposta para o sistema e planejar o processo de desenvolvimento do sistema.
Engenheiros de sistema	Usam os requisitos para compreender qual sistema será desenvolvido.
Engenheiros de testes de sistema	Usam os requisitos para desenvolver testes de validação para o sistema.
Engenheiros de manutenção de sistemas	Usam os requisitos para compreender o sistema e os relacionamentos entre suas partes.



Estrutura do documento de requisitos (1)

Prefácio	Deve definir o público-alvo do documento e descrever seu histórico de versões, incluindo uma justificativa lógica para a criação da nova versão e um resumo das mudanças feitas em cada versão.
Introdução	Deve descrever a necessidade do sistema. Deve descrever brevemente suas funções e explicar como o sistema irá funcionar com outros sistemas. Deve descrever como o sistema atende aos objetivos gerais de negócios e estratégicos da organização que encomendou o software.
Glossário	Deve definir os termos técnicos usados no documento. Você deve fazer suposições sobre a experiência ou as habilidades do leitor
Definição de requisitos de usuário	Os serviços fornecidos ao usuário e os requisitos não funcionais do sistema devem ser descritos nesta seção. Essa descrição pode usar linguagem natural, diagramas e outras notações compreensíveis pelos clientes. Padrões de produto e de processo a serem seguidos devem ser especificados.
Arquitetura do sistema	Esta seção deve apresentar uma visão de alto nível da arquitetura prevista do sistema, mostrando a distribuição das funções nos módulos do sistema.
Especificação de requisitos de sistema	Deve descrever os requisitos funcionais e não funcionais mais detalhadamente. Caso necessário, mais detalhes podem ser também adicionados aos requisitos não funcionais; por exemplo, interfaces com outros sistemas devem ser definidas.



Estrutura do documento de requisitos (2)



41/43

Modelos de sistema	Deve estabelecer um ou mais modelos de sistema, mostrando os relacionamentos entre os componentes e o sistema e seu ambiente. Podem ser modelos de objetos, modelos de fluxo de dados e modelo semântico de dados.
Evolução de sistema	Deve descrever as hipóteses fundamentais sobre as quais o sistema está baseado, além de mudanças previstas devido à evolução do hardware, mudança das necessidades do usuário etc
Apêndices	Deve fornecer informações detalhadas e específicas relacionadas à aplicação que está sendo desenvolvida. Exemplos de apêndices que podem ser incluídos são descrições de hardware e banco de dados. Os requisitos de hardware definem as configurações mínima e ideal para o sistema. Os requisitos de banco de dados definem a organização lógica dos dados usados pelo sistema e os relacionamento entre os dados.
Índice	Podem ser incluídos diversos índices para o documento. Assim como um índice alfabético normal, pode haver um índice dos diagramas, índice das funções etc.



Resumo

- Engenharia de requisitos tem como objetivos identificar as funções e as restrições associadas a um sistema de informação.
- Requisitos: funcionais e não-funcionais.
- Documento de especificação de requisitos:
 - Requisitos de usuário: descritos em linguagem que o cliente/usuário pode entender.
 - Requisitos de projeto: descritos em linguagem que subsidie o desenvolvimento do projeto. Pode ser utilizado como parte de um contrato entre cliente e desenvolvedores. Podem ser descritos com notações adicionais como casos de usos, diagramas de fluxo de dados e especificações formais.
- Referências utilizadas: [?, ?].



Exercícios



43/43

Elabore o documento de requisitos do sistema que o seu grupo está desenvolvendo. Este documento deverá:

1. ser elaborado utilizando o padrão IEEE (incluído no módulo *Requisitos* que consta do CoL);
2. conter o diagrama de casos de uso, bem os casos de uso elaborados de acordo com o padrão que consta do CoL;
3. conter o modelo conceitual do sistema;
4. conter os diagramas de seqüência de sistema.

Utilize seus conhecimentos de ACH 2003 – Computação Orientada a Objetos.

A propósito, este exercício vale 50% da nota do EP1.





Referências

- [1] Shari Lawrence Pfleeger. *Engenharia de Software: Teoria e prática*. Prentice-Hall, São Paulo, 2a. edition, 2004.
- [2] Ian Sommerville. *Engenharia de Software*. Pearson Addison-Wesley, São Paulo, 8a. edition, 2007.

