

Requisitos de Software

Engenharia de Software

2o. Semestre de 2006

Requisitos de software

- Descrição e especificação de sistemas

Objetivos

- Introduzir os conceitos de requisitos do usuário e requisitos de sistema.
- Descrever requisitos funcionais e não funcionais.
- Explicar duas técnicas para descrever os requisitos de sistema.
- Mostrar como requisitos de software podem ser organizados em um documento de requisitos de software

Tópicos

- Requisitos funcionais e não funcionais
- Requisitos do Usuário
- Requisitos do sistema
- O documento de requisitos de software

Engenharia de Requisitos

- É o processo de estabelecer os serviços que o cliente requer de um sistema e as restrições sob as quais deve ser desenvolvido e operar.
- Requisitos são as descrições das funções e restrições.

O que é um requisito?

- Pode variar de uma declaração abstrata de alto nível, de uma função que o sistema deve fornecer ou de uma restrição do sistema, a uma definição detalhada, matematicamente formal, de uma função do sistema.
- Assim, requisitos possui diferentes níveis de descrição;
 - Pode ser a base para uma licitação de um contrato - deve ser aberto à interpretações.
 - Pode ser a base para o próprio contrato - deve ser definido em detalhes.
 - Ambas declarações podem ser chamada requisitos.

Por que os requisitos são importantes?

Pesquisa em mais de 350 empresas sobre os seus mais de 8.000 Projetos de software – 30 % dos projetos foram cancelados. Dos concluídos, 9% entregues dentro do prazo e do valor estimado(Standish Group –1994).

Fatores principais relatados como causas das falhas:

1. Requisitos incompletos (13.1%)
2. Falta de envolvimento por parte do usuário (12.4%)
3. Falta de recursos (10.6%)
4. Expectativas não realistas (9.9%)
5. Falta de apoio dos executivos (9.3%)
6. Modificações nos requisitos e nas especificações (8.7%)
7. Falta de planejamento (8.1%)
8. O sistema não era mais necessário (7.5%)

Por que os requisitos são importantes?

- Falta de cuidado com os requisitos pode levar a
 - Construção de um sistema que resolve o problema errado;
 - Não funciona como esperado;
 - Difícil para os usuários entenderem e utilizarem;
 - Alto custo.

Vale a pena utilizar algum tempo para entender o problema e seu contexto, e obter os requisitos certos na primeira vez.

Tipos de requisitos

- Requisitos do usuário
 - Declarações em linguagem natural e também em diagramas sobre as funções que o sistema deve fornecer e as restrições sob as quais deve operar.
- Requisitos do sistema
 - Um documento estruturado que estabelece detalhadamente as funções e as restrições de sistema. Escrito como um contrato entre o cliente e o desenvolvedor do software.
- Especificação do software
 - Uma descrição detalhada do software que serve como base para projeto e a implementação. Escrito para os desenvolvedores.

Definições e Especificações

Definição dos requisitos do usuário

1. O software deve oferecer um meio de representar e acessar arquivos externos criados por outra ferramenta

Especificação dos requisitos de sistema

- 1.1. O usuário deve dispor de recursos para definir o tipo dos arquivos externos.
- 1.2. Cada tipo de arquivo externo pode ter uma ferramenta associada que pode ser aplicada a ele.
- 1.3. Cada tipo de arquivo externo pode ser representado como um ícone específico na tela do usuário.
- 1.4 Devem ser fornecidos recursos para o ícone que representa um arquivo externo, a ser definido pelo usuário.
- 1.5. Quando um usuário seleciona um ícone que representa um arquivo externo, o efeito dessa seleção é aplicar a ferramenta associada com o tipo de arquivo externo ao arquivo representado pelo ícone selecionado.

Leitores de diferentes tipos de especificações

Requisitos
do usuário

Gerentes de clientes
Usuários finais de sistemas
Engenheiros do cliente
Gerentes do fornecedor
Arquitetos de sistemas

Requisitos
de sistema

Usuários finais de sistemas
Engenheiros do cliente
Arquitetos de sistemas
Desenvolvedores de software

Especificação de
projeto de software

Engenheiros do cliente (talvez)
Arquitetos de sistemas
Desenvolvedores de software

Requisitos funcionais e não funcionais

- Requisitos funcionais

- Declarações de serviços que o sistema deve fornecer, como o sistema deve reagir a entradas específicas e como deve se comportar em determinadas situações.

- Requisitos não funcionais

- Restrições sobre os serviços ou as funções oferecidas pelo sistema.

- Requisitos de domínio

- Requisitos que se originam do domínio da aplicação do sistema e que refletem características desse domínio (Podem ser requisitos funcionais e não funcionais).

Requisitos funcionais

- Descrevem a funcionalidade ou os serviços do sistema.
- Dependem do tipo de software, das expectativas dos usuários e do tipo de sistema que está sendo desenvolvido.
- Requisitos funcionais do usuário são descritos de forma bem geral, mas os requisitos funcionais de sistema descrevem a função de sistema detalhadamente.

Exemplos de requisitos funcionais

- O usuário deverá ser capaz de buscar todo o conjunto inicial de banco de dados ou selecionar um subconjunto a partir dele.
- O sistema fornecerá telas apropriadas para o usuário ler documentos no repositório de documentos.
- Cada pedido será alocado a um único identificador (ORDER-ID), que o usuário poderá copiar para a área de armazenagem permanente da conta

Requisitos funcionais do usuário

Imprecisão de Requisitos

- Problemas se originam da imprecisão na especificação de requisitos.
- Requisitos ambíguos podem ser interpretados de maneira diferente pelos desenvolvedores e usuário.
- Considere o termo ‘telas apropriadas’
 - Intenção do usuário - telas para cada formato de documento devem ser disponibilizadas.
 - Intenção do desenvolvedor - disponibilizar uma tela de texto que mostra o conteúdo do documento.

Completeza e consistência de requisitos

- Em princípio, a especificação de requisitos funcionais deve ser completa e consistente
- Completeza
 - Todas as funções requeridas pelo usuário devem estar definidas
- Consistência
 - Não devem haver definições contraditórias de requisitos
- Na prática, é quase impossível atingir a consistência e a completeza dos requisitos.

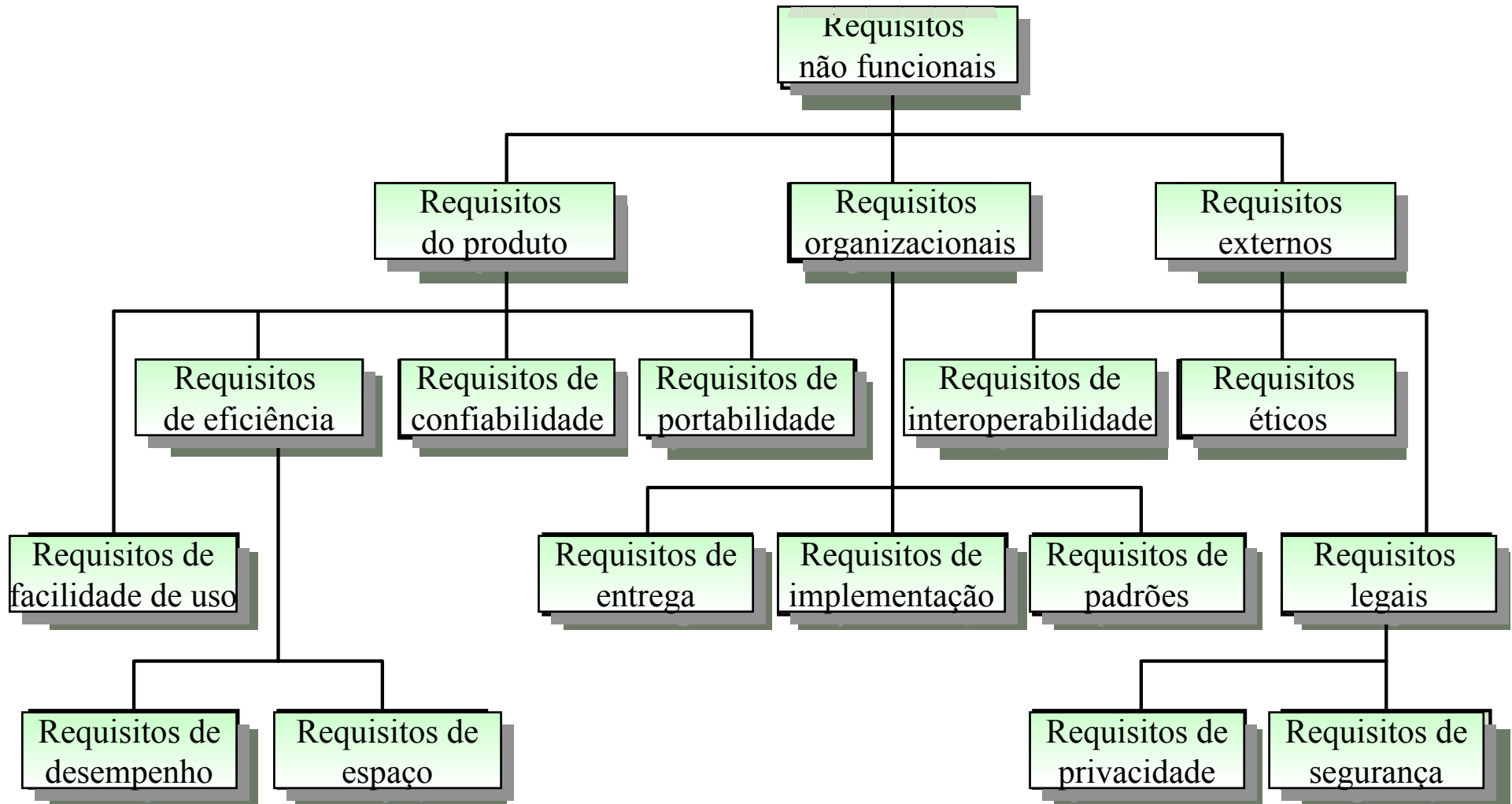
Requisitos não funcionais

- Definem as propriedades de sistemas e restrições, por ex: confiabilidade, tempo de resposta e espaço em disco. Restrições: capacidade dos dispositivos de E/S, representações de dados, etc.
- Requisitos não funcionais dizem respeito ao sistema como um todo. Alguns podem restringir o processo que é utilizado para desenvolver o sistema (ditar um sistema CASE específico, linguagem de programação ou método de desenvolvimento)
- Podem ser mais críticos que requisitos funcionais. A falha em atender um requisito não funcional de sistema pode inutilizar o sistema.

Classificações não funcionais

- Requisitos de produto
 - Requisitos que especificam o comportamento do produto. Ex: velocidade de execução, confiabilidade, portabilidade, facilidade de uso, etc..
- Requisitos organizacionais
 - Requisitos que são consequência de políticas de procedimentos nas organizações do cliente e do desenvolvedor. Ex: padrões de processos que devem ser utilizados, requisitos de implementação, etc.
- Requisitos externos
 - Requisitos procedentes de fatores externos ao sistema e a seu processo de desenvolvimento. Ex: requisitos de interoperabilidade, requisitos legais e os requisitos éticos.

Tipos de requisitos não funcionais



Exemplos de requisitos não funcionais

- Requisitos de produto
 - 4.C.8 Deve ser possível que toda a comunicação necessária entre o Ambiente de apoio à programação Ada (APSE) e o usuário seja expressa no conjunto padrão de caracteres Ada. (restringe a liberdade do projetistas do APSE na sua escolha de símbolos utilizados)
- Requisito Organizacional
 - 9.3.2 O processo de desenvolvimento de sistema e os documentos a serem entregues deverão estar de acordo com o processo e os produtos a serem entregues, definidos em XYZ-SP- STAN95
- Requisitos externos
 - 7.6.5 O sistema não deverá revelar aos operadores nenhuma informação pessoal sobre os clientes, além de seus nomes e o número de referência (legislação de privacidade)

Metas e requisitos

- Requisitos não funcionais muitas vezes podem ser difíceis de serem colocados e verificados.
- Meta do sistema
 - Uma intenção geral do usuário, por ex.: a facilidade de uso.
- Requisito não funcional verificável
 - Uma declaração usando alguma métrica que pode ser objetivamente testada.

Exemplos

- **Uma meta do sistema**

- O sistema deve ser fácil de utilizar por controladores experientes e deve ser organizado de modo que os erros dos usuários sejam minimizados.

- **Um requisito não funcional verificável**

- Controladores experientes devem ser capazes de utilizar todas as funções do sistema depois de um total de duas horas de treinamento. Depois desse treinamento, o número médio de erros feitos pelo usuário não deve exceder a dois por dia.

Métricas para especificar requisitos não funcionais

Propriedade	Métrica
Velocidade	Transações processadas/segundo Tempo de resposta ao usuário/evento Tempo de refresh da tela
Tamanho	K bytes Números de chips de RAM
Confiabilidade	Tempo médio para falha Probabilidade de indisponibilidade Taxa de ocorrência de falhas Disponibilidade
Robustez	Tempo de reinício depois de uma falha Porcentagem de eventos que causam falhas. Probabilidade de que dados sejam corrompidos por falhas
Portabilidade	Porcentagem de declarações dependentes de sistemas alvo. Número de sistemas alvo
Facilidade de uso	Tempo de treinamento Número de frames de ajuda

Interação de requisitos

- É comum a existência de conflitos entre diferentes requisitos não funcionais em sistemas complexos.
- Sistema para uma Espaçonave
 - Para minimizar peso, o número de chips separado deve ser minimizado.
 - Para minimizar consumo de energia, chips de baixo consumo deve ser usado
 - Contudo, o uso de chips de baixo consumo significa que mais chips devem ser utilizados. Qual é o requisito mais crítico?

Requisitos de Domínio

- Derivados do domínio de aplicação e refletem fundamentos do domínio da aplicação.
- Podem ser novos requisitos funcionais em si, podem restringir os requisitos funcionais existentes, ou estabelecer como realizar cálculos específicos.
- Se não forem satisfeitos, poderá ser impossível fazer o sistema operar satisfatoriamente.

Requisitos de domínio do sistema de biblioteca

- Deve haver uma interface padrão com o usuário para todos os bancos de dados, que terá como base o padrão Z 39.50
- Em razão das restrições referentes a direitos autorais, alguns documentos devem ser excluídos imediatamente ao serem fornecidos. Dependendo dos requisitos dos usuários, esses documentos serão impressos localmente no servidor do sistema para serem encaminhados manualmente ao usuário ou direcionados para uma impressora de rede.

Sistema automatizado de proteção de trens

- A desaceleração do trem será computada como:
 - $D_{\text{trem}} = D_{\text{controle}} + D_{\text{gradiente}}$
onde $D_{\text{gradiente}}$ é 9.81ms^2 * gradiente compensado/alfa e onde os valores de $9.81\text{ms}^2/\text{alfa}$ são conhecidos para diferentes tipos de trens .

Problemas com os requisitos de domínio

- Entendimento
 - Requisitos são expressos com o uso de uma linguagem que é específica do domínio da aplicação
- Declarações implícitas
 - Especialistas em um domínio podem deixar de fornecer informações em um requisito, simplesmente por acharem que essas informações são muito óbvias.

Requisitos de usuário

- Devem descrever os requisitos funcionais e não funcionais de modo compreensível pelos usuários do sistema, que não tem conhecimentos técnicos detalhados.
- Requisitos do usuário são definidos usando linguagem natural, tabelas e diagramas.

Problemas com linguagem natural

- Falta de clareza
 - Difícil usar a linguagem de maneira precisa e sem ambigüidade, sem produzir um documento de difícil leitura
- Confusão de requisitos
 - Os requisitos funcionais e os não funcionais, os objetivos do sistema e as informações sobre o projeto podem não estar claramente definidos.
- Fusão de requisitos
 - Vários requisitos diferentes podem ser expressos juntos.

Ex: Requisitos sobre um banco de dados

4.A.5 O banco de dados deve gerar objetos e fazer o controle de configuração de objetos , que podem ser agrupamentos de outros objetos no banco de dados. Os recursos de controle de configuração devem permitir o acesso aos objetos em um grupo de versão, pelo uso de um nome incompleto.

Ilustra informações conceituais e detalhadas

Requisitos de usuário para uma grade de editor (Editor de modelos de projeto de software - CASE)

2.6 Recursos de grade Para ajudar no posicionamento de objetos em um digrama, o usuário pode acionar uma grade em centímetros ou em polegadas, por meio de uma opção no painel de controle. Inicialmente, a grade está desativada. Ela pode ser ligada e desligada a qualquer momento durante uma sessão de edição e pode ser alternada entre polegadas e centímetros a qualquer momento. Uma opção de grade será fornecida na visão reduzida do diagrama, mas o número de linhas da grade mostrado diminuirá, para evitar preencher o diagrama menor com linhas de grade.

Ilustra Requisitos do usuário X requisitos do sistema

Problemas com requisitos

- Os requisitos do banco de dados inclui informações conceituais e detalhadas
 - Descreve o conceito de facilidade de controle de configuração
 - Inclui o detalhe de que objetos podem ser acessados usando um nome incompleto.
- Os requisitos para uma grade mistura 3 tipos diferentes de requisitos
 - Requisito funcional conceitual (a necessidade de uma grade)
 - Requisito não funcional (unidades de medida)
 - Requisitos não funcional de interface com o usuário (Como a grade é ligada e desligada pelo usuário)

Apresentação estruturada

Recursos de grade

O editor deverá fornecer um recurso de grade, em que uma matriz de linhas horizontais e verticais constitua um fundo da janela do editor. Essa grade deverá ser uma tela passiva em que o alinhamento de objetos é responsabilidade do usuário. *Lógica:* uma grade ajuda o usuário a criar um diagrama 'limpo', com objetos bem espaçadas. Embora uma grade ativa, em que os objetos saltam as linhas de grade, possa ser útil, o posicionamento é impreciso. O usuário é a melhor pessoa para decidir onde os objetos devem ser posicionadas.

Especificação: Onde se encontra uma descrição mais detalhada: DirProjeto/seção 5.6

Requisitos do usuário detalhado

3.5.1. Adicionando nós a um desenho.

3.5.1.1 O editor deve fornecer um recurso aos usuários para adicionar nós de um tipo especificado a seu desenho.

3.5.1.2 A seqüência de ações para acrescentar um nó deveria ser como se segue:

1. o usuário deve selecionar o tipo de nó a ser acrescentado.
2. O usuário deve mover o cursor para a posição aproximada do nó no diagrama e indicar que o símbolo do nó deve ser adicionado naquele ponto.
3. O usuário deve então arrastar o símbolo do nó para sua posição final.
Lógica: o usuário é a melhor pessoa para decidir onde posicionar um nó no diagrama. Essa abordagem dá ao usuário o controle direto sobre a seleção do tipo de nó e seu posicionamento

Especificação: ECLIPSE/WS/Ferramentas/DE/FS. Seção 3.5.1

Diretrizes para escrever requisitos

- Estabeleça um formato padrão e use-o para todos os requisitos.
- Utilize a linguagem de modo consistente. Faça distinção entre os requisitos obrigatórios e os que são desejáveis (O sistema **deve** ... O sistema **deveria** ...).
- Utilize destaque no texto para ressaltar partes importantes dos requisitos.
- Evite o uso de jargões de informática.

Requisitos do sistema

- Descrições mais detalhadas dos requisitos do usuário.
- Servem como base para um contrato destinado à implementação do sistema.
- Ponto de partida para o projeto do sistema.
- Requisitos do sistema pode ser expresso usando diferentes modelos de especificação de requisitos

Requisitos e projeto

- Em princípio, os requisitos deveriam definir o que o sistema deveria fazer e não como deveria ser implementado.
- Na prática, é difícil conseguir essa separação
 - Uma arquitetura inicial do sistema pode ser definida para ajudar a estruturar os requisitos.
 - O sistema deve inter- operar com outros sistemas que geram requisitos para o novo sistema.
 - O uso de um projeto específico pode ser um requisito externo de sistema

Problemas com especificação em Linguagem Natural

- Ambigüidade

- Leitores e escritores dos requisitos devem interpretar as mesmas palavras da mesma maneira. A linguagem natural é naturalmente ambígua e isso dificulta a interpretação

- Flexibilidade

- A mesma coisa pode ser dita de diferentes maneiras na especificação

- Falta de modularização

- estruturas da linguagem natural não são apropriadas para estruturar os requisitos do sistema.

Alternativas para especificação em Linguagem Natural

Notação	Descrição
Linguagem natural estruturada	Depende da definição de formulários padrão ou <i>templates</i> para expressar a especificação de requisitos.
Linguagem de descrição de projeto	Utiliza uma linguagem parecida com uma linguagem de programação, porém com recursos mais abstratos.
Notações gráficas	Linguagem gráfica, complementada com textos, é utilizada para definir os requisitos funcionais do sistema. Ex: use-case.
Especificações matemáticas	São notações com base em conceitos matemáticos, como por exemplo, máquinas de estados finitos, redes de Petri, statecharts, etc. São especificações não ambíguas, porém usuários relutam em aceitá-las como base para um contrato.

Especificações em linguagem natural estruturada

- É uma forma restrita da linguagem natural, que se destina a escrever requisitos do sistema.
- Essa abordagem elimina problemas de ambigüidade e flexibilidade e garante um certo grau de uniformidade na especificação.
- Em geral, as notações de linguagem estruturada utilizam *templates* e abordagens baseadas em formulários pré-definidos.

Especificações baseadas em formulário padrão

Inclui:

- Descrição da função ou entidade especificada.
- Descrição de suas entradas e origens.
- Descrição de saídas e destinos.
- Indicação de outras entidades utilizadas.
- Pré e pós condição (se apropriado)
- Efeitos colaterais (caso existam)

Especificação utilizando formulário padrão

Função Adicionar nós.

Descrição Adiciona um nó em um desenho existente. O usuário seleciona o tipo de nó e seu posicionamento. Quando Adicionado ao desenho, o nó se torna a seleção atual. O usuário escolhe a posição do nó movimentando o cursor para a área que em que o nó será adicionado.

Entradas Tipo de nó, Posição do nó, identificador do desenho

Origem Tipo de nó e posição do nó são entradas fornecidas pelo usuário; Identificador de desenho se origina da base de dados.

Saídas Identificador do desenho.

Destino O banco de dados do desenho. O desenho é designado para a base de dados, no término da operação.

Requer Gráfico de desenho associado ao identificador de desenho de entrada.

Pré condição O desenho é aberto e exibido na tabela do usuário.

Pós condição O desenho é imutável, a não ser pela adição de um nó do tipo especificado em dada posição.

Efeitos colaterais Nenhum.

Especificação de requisitos com o uso de uma PDL

- Requisitos podem ser definidos operacionalmente com o uso de uma linguagem de descrição de programa (PDL)
- PDL é uma linguagem derivada de uma LP, com maior poder de expressão.
- Apropriada em duas situações
 - Quando uma operação é especificada como uma seqüência de ações e a ordem é importante.
 - Quando interfaces de hardware e software tiverem de ser especificadas.

Parte de uma descrição em PDL de uma ATM

```
class ATM {  
    // declarations here  
    public static void main (String args[]) throws InvalidCard {  
        try {  
            thisCard.read () ; // may throw InvalidCard exception  
            pin = KeyPad.readPin () ; attempts = 1 ;  
            while ( !thisCard.pin.equals (pin) & attempts < 4 )  
                {    pin = KeyPad.readPin () ; attempts = attempts + 1 ;  
                  }  
            if (!thisCard.pin.equals (pin))  
                throw new InvalidCard ("Bad PIN");  
            thisBalance = thisCard.getBalance () ;  
            do { Screen.prompt (" Please select a service ") ;  
                service = Screen.touchKey () ;  
                switch (service) {  
                    case Services.withdrawalWithReceipt:  
                        receiptRequired = true ;  
                }  
            } while (true);  
        }  
    }  
}
```

Vantagens da PDL

- Especificação de requisitos em PDL é menos ambíguo e mais fácil de ser compreendido, se o leitor for familiarizado com a PDL.
- Transição natural dos requisitos para a linguagem de implementação, caso a PDL seja derivada desta.
- Reduz-se a possibilidade de má interpretação.

Desvantagens da PDL

- A PDL pode não ser suficientemente expressiva para definir a funcionalidade do sistema.
- A notação é difícil de entender para pessoas que não tenham conhecimento de ling. de programação.
- A especificação é mais um esboço de projeto ao invés de um modelo para auxiliar o usuário a compreender o sistema.
- Especificações muito detalhadas nem sempre é desejável em um documento de requisitos.

Especificação de Interface

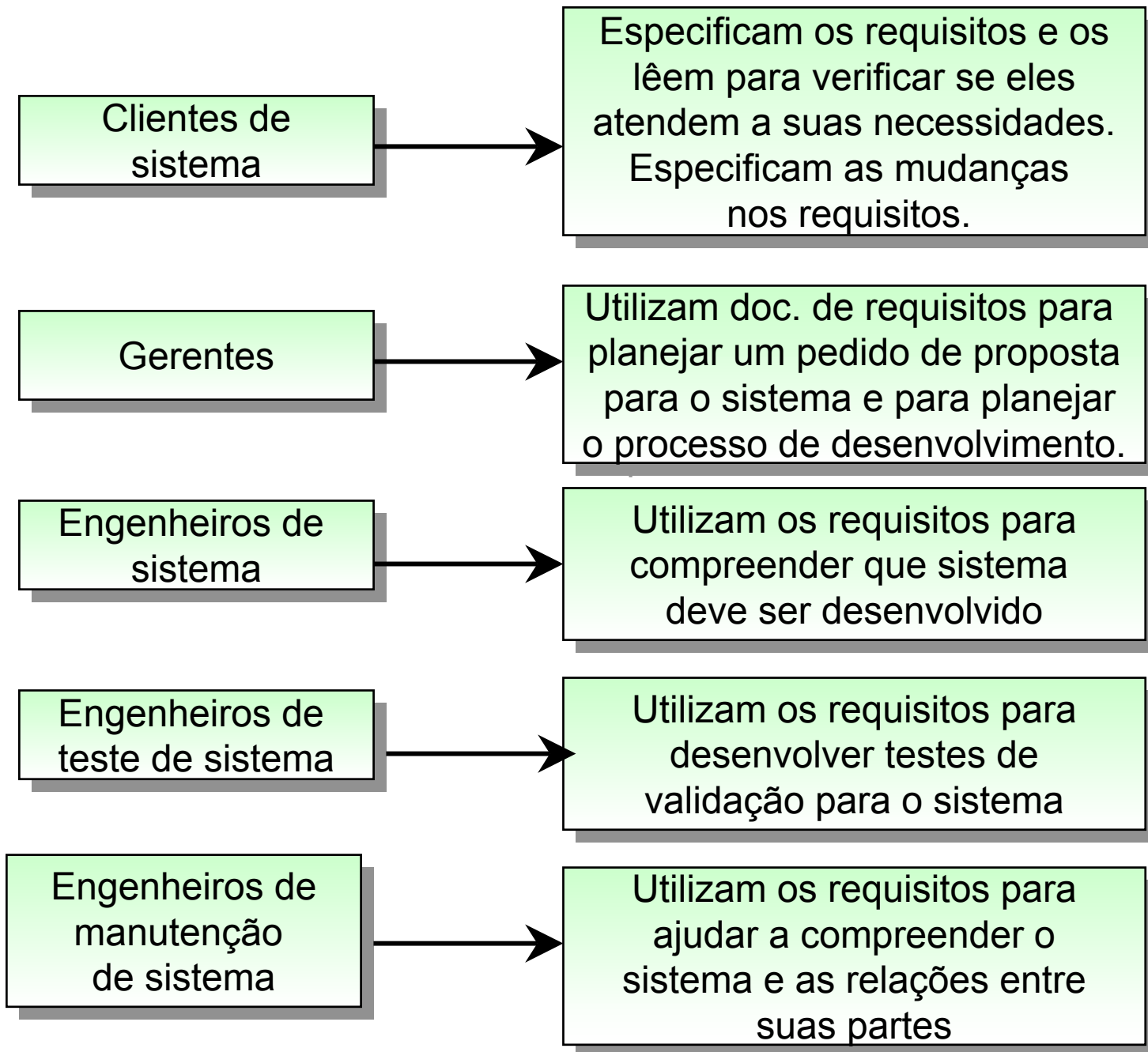
- A maioria dos sistemas deve operar com outros sistemas e as interfaces de sistemas existentes devem ser especificadas como parte dos requisitos.
- Existem três tipos de interfaces que podem precisar ser definidas:
 - Interfaces de procedimentos
 - Estruturas de dados transmitidas de um subsistema para outro
 - Representações de dados.
- Notações formais são técnicas adequadas para a especificação de interfaces.
- Descrições em PDL combinam a facilidade de compreensão e a precisão, mais informais do que as técnicas formais.

Descrição em PDL da interface procedimental de um servidor de impressão

```
interface PrintServer {  
  
    // define um servidor de impressora abstrato  
    // requer: interface Printer, interface PrintDoc  
    // fornece: initialize, print, displayPrintQueue, cancelPrintJob, switchPrinter  
  
    void initialize ( Printer p ) ;  
    void print ( Printer p, PrintDoc d ) ;  
    void displayPrintQueue ( Printer p ) ;  
    void cancelPrintJob (Printer p, PrintDoc d) ;  
    void switchPrinter (Printer p1, Printer p2, PrintDoc d) ;  
} //PrintServer
```

O documento de requisitos

- O documento de requisitos é a declaração oficial do que é exigido dos desenvolvedores de sistema.
- Deve incluir os requisitos de usuário e uma especificação detalhada dos requisitos de sistema.
- NÃO é um documento de projeto. Deve esclarecer o que o sistema deve fazer e não como deve ser feito.



**Usuários de
um
documento de
requisitos**

Requisitos de um documento de requisitos

Deve:

- Especificar o comportamento externo do sistema.
- Especificar as restrições à implementação.
- Ser fácil de ser modificado.
- Servir como ferramenta de referência para a manutenção.
- Registrar a estratégia sobre o ciclo de vida do sistema.
- Caracterizar respostas aceitáveis para eventos indesejáveis.

Padrão IEEE/ANSI 830 - 1993

para o documento de requisitos

- Introdução

- Propósito do documento de requisitos
- Escopo do produto
- Definições, abreviações e acrônimos
- Referências
- Visão geral do restante do documento

- Descrição Geral

- Perspectiva do produto
- Funções do Produto
- Características do usuário
- Restrições gerais
- Suposições e dependências

Padrão IEEE/ANSI 830 - 1993

para o documento de requisitos

- Requisitos específicos (funcionais e não funcionais)
 - Parte principal do documento de requisitos que descreve funções e desempenho do sistema, especifica requisitos de banco de dados, restrições de projeto, propriedades emergentes do sistema e características de qualidade
- Apêndices
- Índice

Documento geral, que deve ser adaptado às necessidades de uma organização particular

A estrutura de um documento de requisitos

- Prefácio
- Introdução
- Glossário
- Definição de requisitos de usuário
- Arquitetura de sistemas
- Especificação de requisitos do sistema
- Modelos do sistema
- Evolução do sistema
- Apêndices
- Índice

Pontos Chave

- Os **requisitos** estabelecem o que o sistema deve fazer e definem restrições sobre sua operação e implementação.
- **Requisitos funcionais** são declarações de funções que o sistema deve fornecer.
- **Requisitos não funcionais** se relacionam às propriedades emergentes do sistema e, portanto, se aplicam ao sistema como um todo.
- **Requisitos de usuário** são declarações de alto nível do que o sistema deve fazer.

Pontos chave

- Requisitos de usuário devem ser escritos em linguagem natural, tabelas e diagramas.
- **Requisitos de sistema** se destinam a comunicar, de modo preciso, as funções que o sistema tem de fornecer.
- Requisitos de sistema podem ser escritos em linguagem natural estruturada, uma PDL ou em linguagem formal.
- O **documento de requisitos de software** é a declaração estabelecida dos requisitos do sistema.

The word "film" is rendered in a 3D, blocky font. The letters are primarily yellow with a gradient that transitions to orange and then to a darker brown at the base, giving it a sense of depth and shadow. The letters are slightly tilted and appear to be floating in the center of the slide.