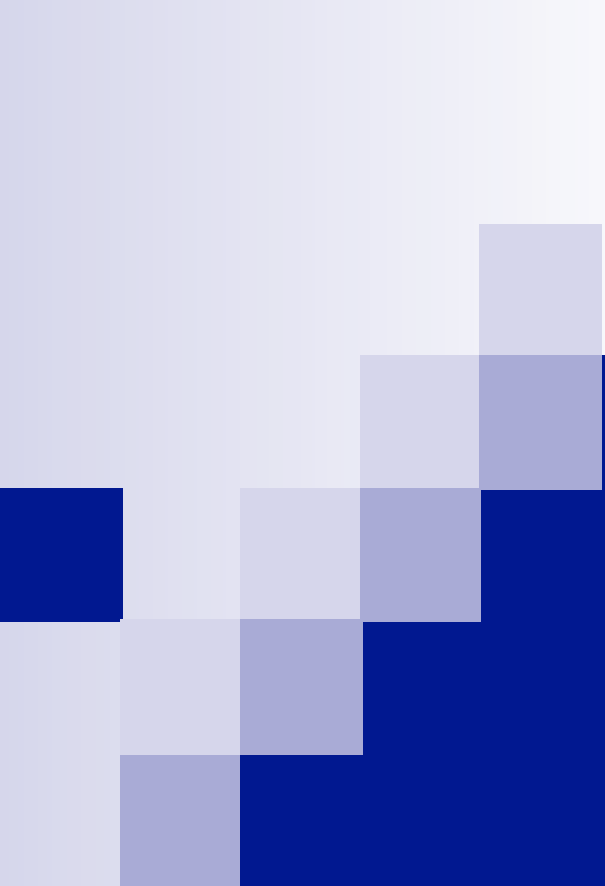




ENGENHARIA DE REQUISITOS

Engenharia de Software
Profa. Dra. Elisa Yumi Nakagawa

1. Semestre de 2016



Conteúdo

1. Engenharia de Requisitos
 - 1.1 Elicitação
 - 1.2 Análise
 - 1.3 Modelagem
2. Documento de Requisitos

1. *ENGENHARIA DE REQUISITOS*

REQUISITO:

- Condição necessária para a obtenção de um certo objetivo, ou para o preenchimento de certo fim.
 - (AURÉLIO)

ESPECIFICAÇÃO:

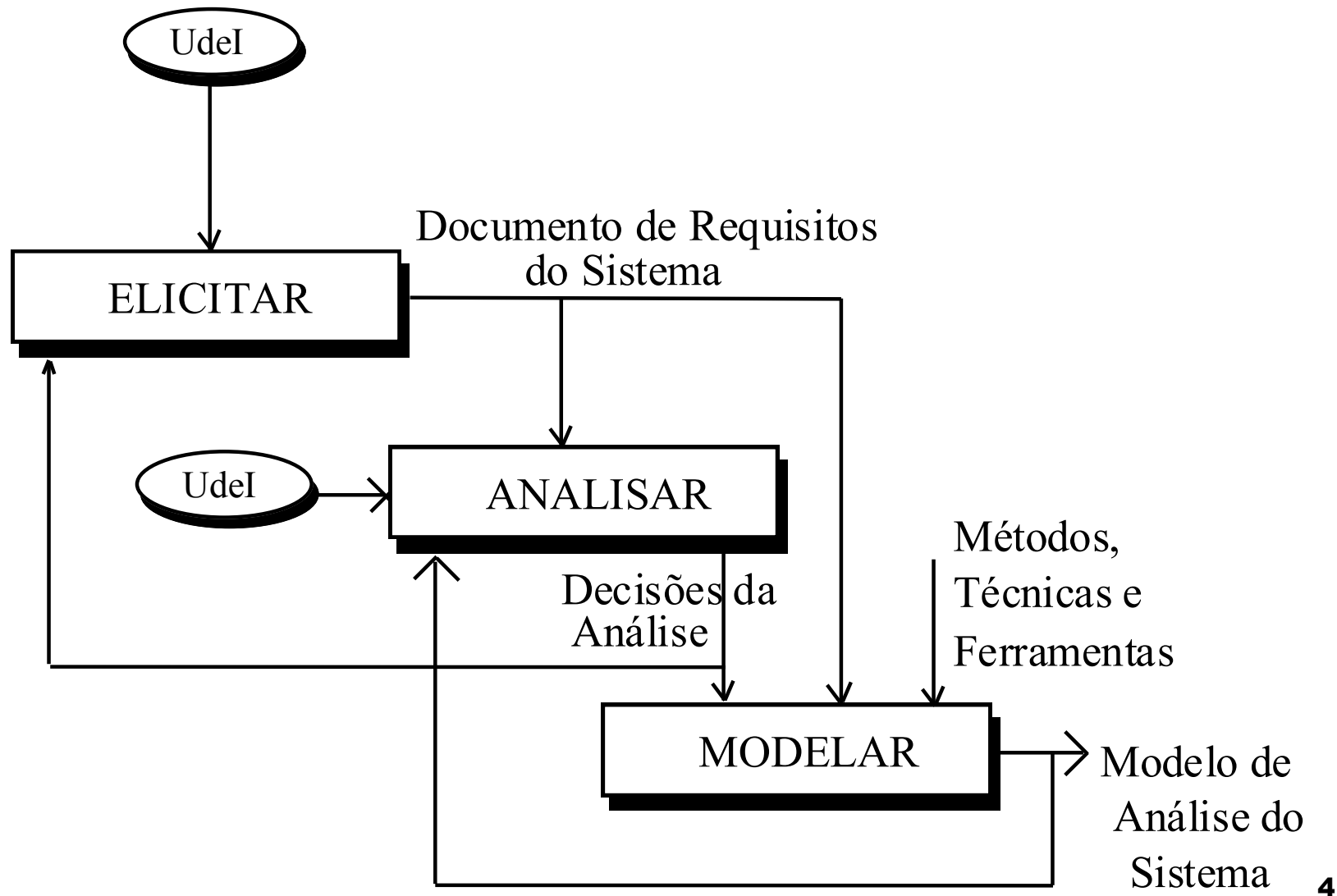
- Descrição minuciosa das características que um material, uma obra, ou um serviço deverão apresentar.
 - (AURÉLIO)

Portanto, Especificação é diferente de Requisitos.

Às vezes, usa-se:

- Especificação de Requisitos
- Especificação de Projeto

ENGENHARIA DE REQUISITOS



Definição

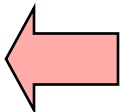
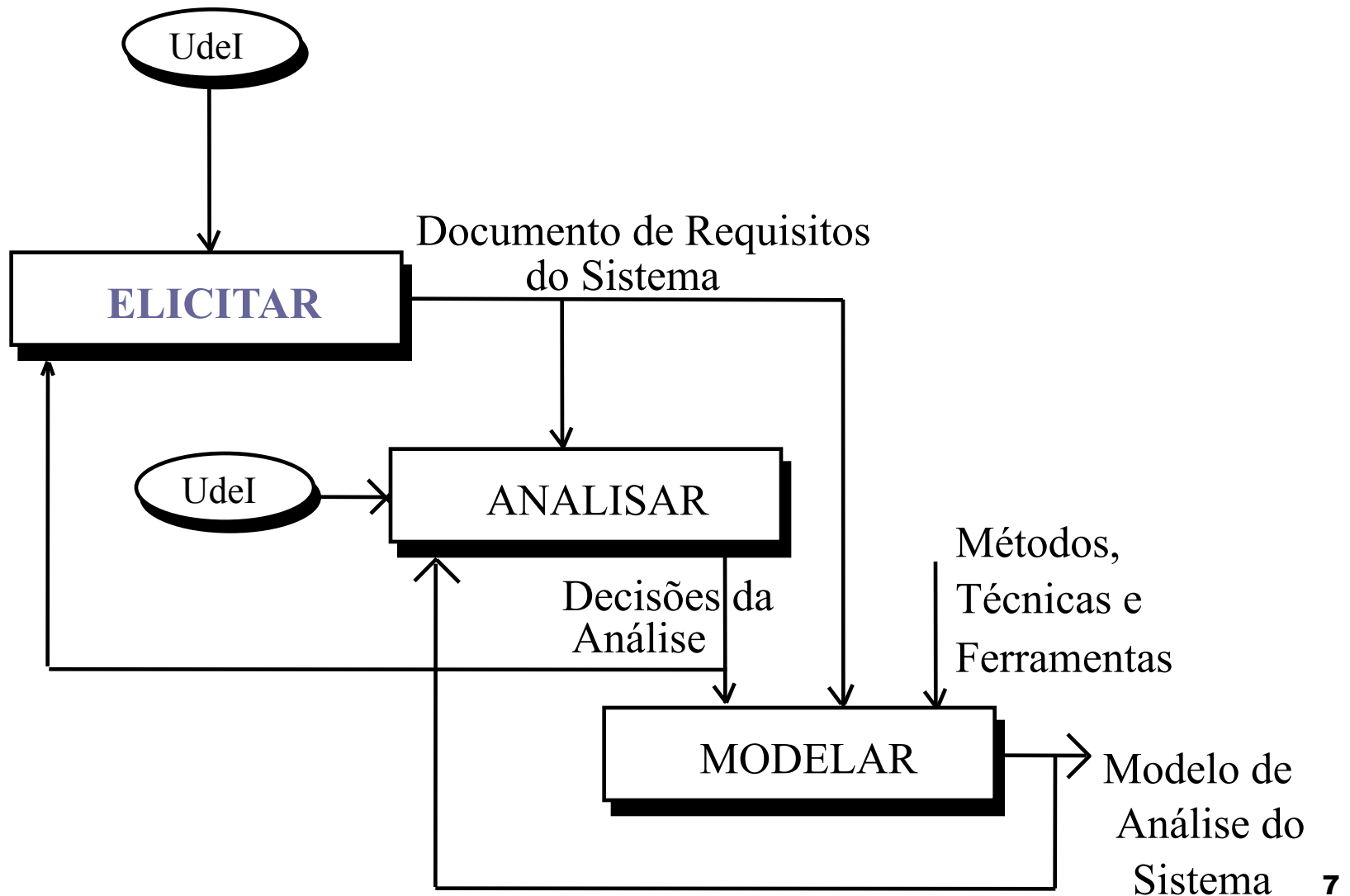
A **E.R.** estabelece o processo de definição de Requisitos como um processo no qual o que deve ser feito é elicitado, modelado e analisado. Este processo deve lidar com diferentes pontos de vista, e usar uma combinação de métodos, ferramentas e pessoal. O produto desse processo é um modelo, do qual um documento de requisitos é produzido. Este processo acontece num contexto previamente definido a que chamamos de **Universo de Informação**.

(Júlio Leite, 1994)

Universo de Informação

É o conjunto geral no qual o software será desenvolvido. Inclui todas as fontes de informação e todas as pessoas relacionadas ao software, às quais denominamos de agentes desse universo. O Udel é a realidade circunstanciada pelo conjunto de objetivos definidos por quem solicitou o software.

ENGENHARIA DE REQUISITOS



1.1 *ELICITAÇÃO*

- **ELICITAR** = Eliciar + Clarear + Extrair + Descobrir ,
tornar explícito, obter o máximo de informação para o
conhecimento do objeto em questão.
 - Eliciar = Fazer sair, extrair, trazer à tona.
- **HÁ TRÊS ATIVIDADES PRINCIPAIS:**
 - Identificação de fontes de informação
 - Coleta de Fatos
 - Comunicação

1.1 *ELICITAÇÃO*

■ NECESSIDADE DA ELICITAÇÃO

- “There is no service in being precise about something when you do not even know what you are talking about” (von Neumann).
- RESOLUÇÃO DE PROBLEMAS:
 - + What is the unknown?
 - + Do you know a related problem?
- CUSTOS CRESCENTES PARA CORREÇÃO DE ERROS!!!

1.1 ELICITAÇÃO

- *Faz* coleta de fatos
- *Faz* identificação de fontes de informação
- *Faz* comunicação
- *Faz/Usa* ferramentas
- *Usa* pessoal
- *Usa* métodos
- *Depende de* pontos de vista



COLETA DE FATOS

- Leitura de documentos
- Observação
- Entrevistas
- Questionários
- Análise de Protocolos
- Participação ativa dos agentes do Udel
- Reuniões
- Reutilização
- Recuperação (eng. reversa) do projeto do software

IDENTIFICAÇÃO DAS FONTES DE INFORMAÇÃO

- Udel: Contém toda informação necessária
- Agentes (Atores, Usuários)
- Outras fontes de Informação:
 - Documentação do macrosistema
 - Políticas
 - Manuais
 - Memos, atas, contratos...
 - Livros sobre tema relacionado
 - Outros sistemas da empresa
 - Outros sistemas externos.

IDENTIFICAÇÃO DAS FONTES DE INFORMAÇÃO

- *Importante:*

- Priorizar as Fontes de Informação.

- Heurísticas:

- Atores mais importantes

- Documentos mais mencionados

- Rede de comunicações entre os componentes do macro-sistema

- ...

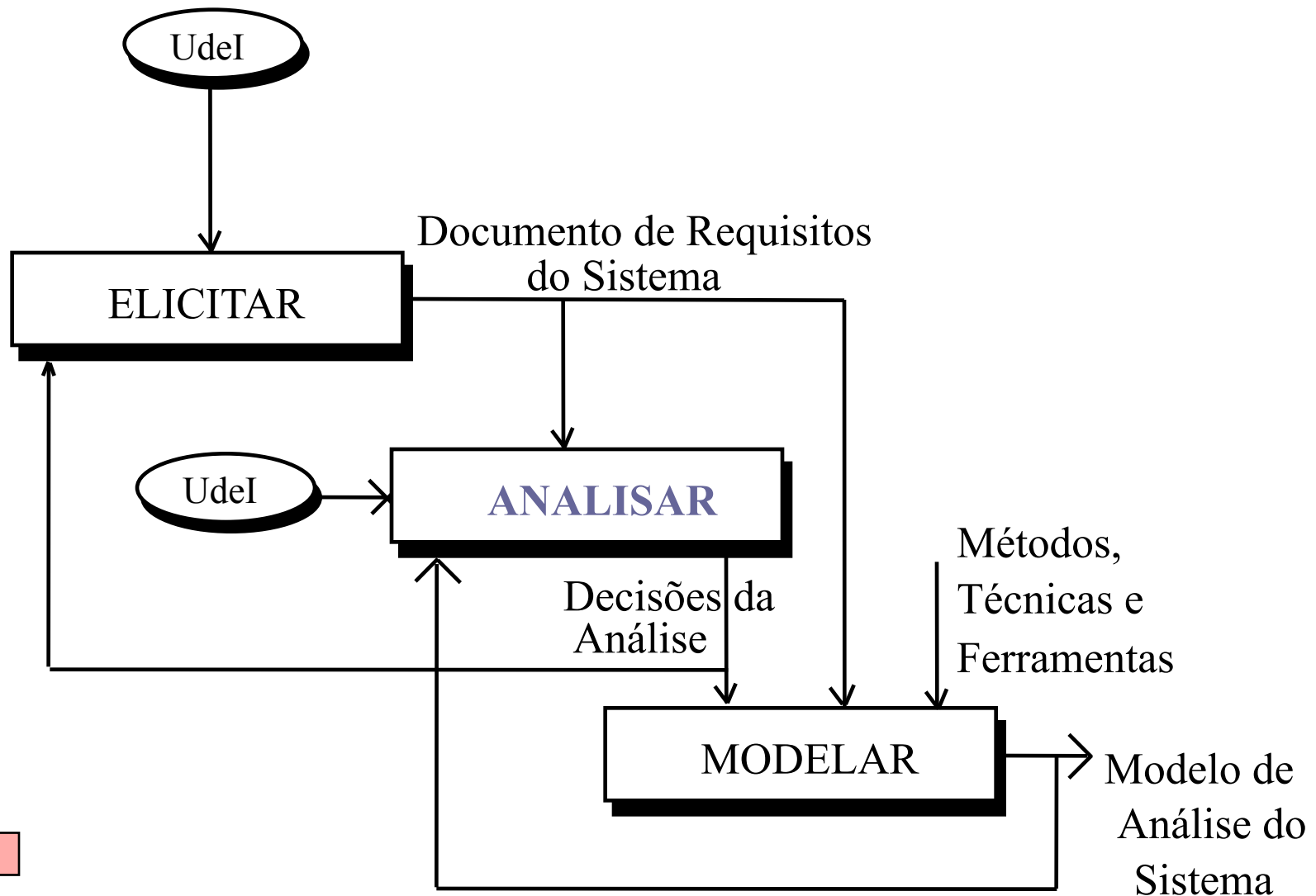
COMUNICAÇÃO

- Atividade fundamental para que a fase de elicitación tenha sucesso.
- Trata-se da comunicação entre clientes/agentes e os engenheiros de software.
 - *Entendimento*: estabelecimento de um contexto comum.
 - Ex. Ordem de 5,10,2,9,8,4,6,...
 - *Linguagem*
 - *Nível de Abstração*
 - *Retro-alimentação*

1.1 ELICITAÇÃO

- *Faz* coleta de fatos
- *Faz* identificação de fontes de informação
- *Faz* comunicação
- *Faz/Usa* ferramentas
- *Usa* pessoal
- *Usa* métodos
- *Depende de* pontos de vista

ENGENHARIA DE REQUISITOS



1.2 ANÁLISE

- Fundamental para o sucesso do processo de desenvolvimento do software.
- Envolve três principais atividades:
 - *Identificação de Partes*
 - *Verificação*
 - *Validação*

1.2 ANÁLISE

FAZ Identificação de Partes

FAZ Verificação

FAZ Validação

USA Pessoal

USA Métodos

USA Ferramentas

DEPENDE DE

Pontos de Vista

Identificação de Partes

Organização

Armazenamento



*É Completo?
É Correto?*

MODELO

É Consistente?

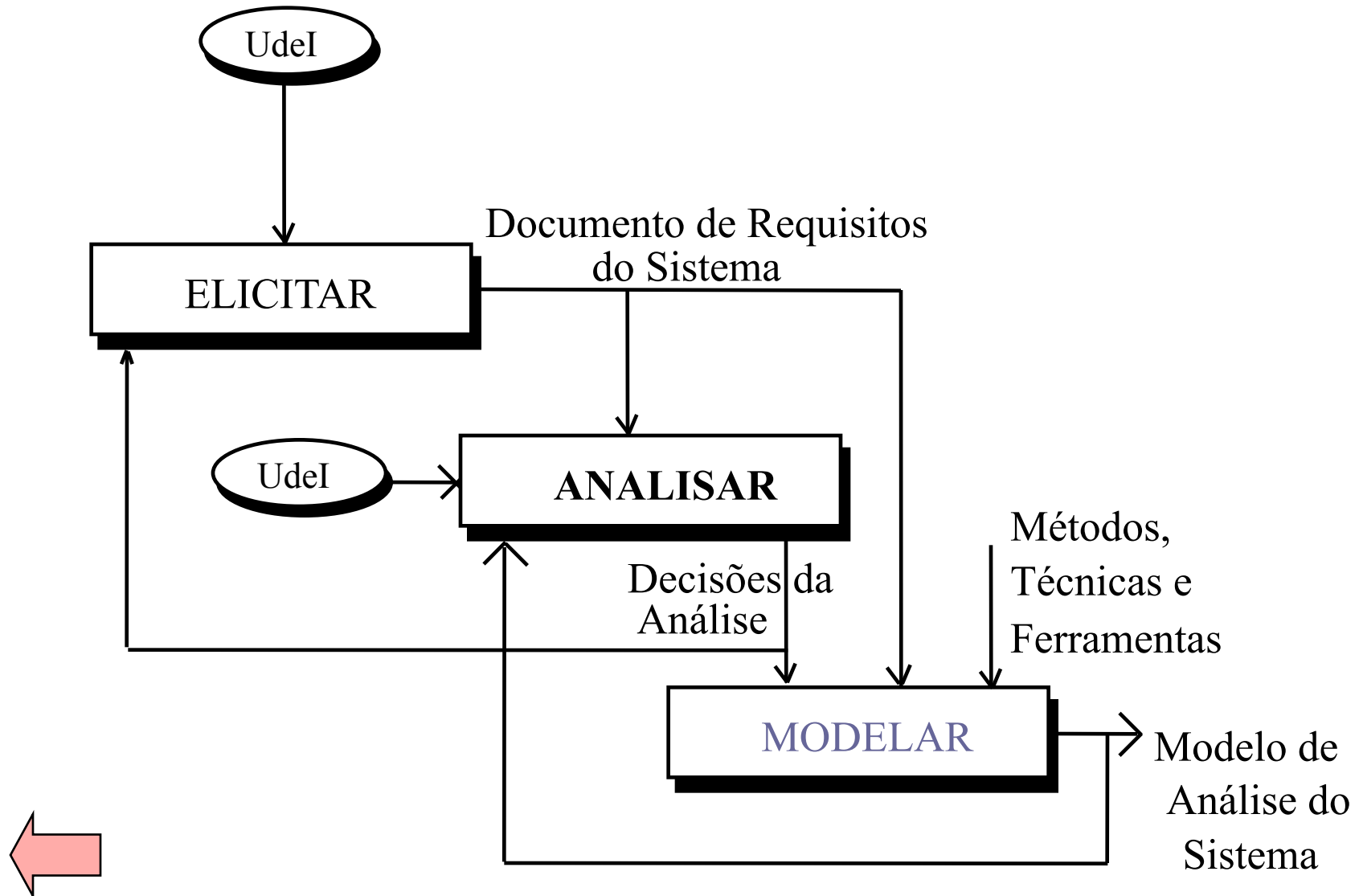
■ **Verificação**

- ☐ Entre níveis diferentes de um modelo (usando métodos formais, como Z e VDM) ou entre modelos de mesmo nível

■ **Validação**

- ☐ Usando comprovação informal
- ☐ Usando protótipos
- ☐ Reusando Domínios
- ☐ Usando pontos de vista

ENGENHARIA DE REQUISITOS



1.3 *MODELAGEM*

- Construção de modelos do sistema utilizado técnicas e métodos.
- Há três atividades:
 - *Representação*
 - *Organização*
 - *Armazenamento*

1.3 MODELAGEM

FAZ Representação
FAZ Organização
FAZ Armazenamento
USA Pessoal

USA Métodos
USA Ferramentas
DEPENDE DE
Pontos de Vista

Representação:

Tipos,
Relações
Operações

Organização:

Níveis de Abstração
Regras de Refinamento
Regras de Consistência Interna

Armazenamento:

Classificação
Indexação
Aspectos Gerais

1.3 MODELAGEM

- Métodos/Técnicas utilizados na modelagem:
 - ☐ Análise Estruturada/Análise Essencial
 - ☐ Fusion
 - ☐ Booch
 - ☐ OMT
 - ☐ JSD
 - ☐ UML
 - ☐ e outros

2. *DOCUMENTO DE REQUISITOS*

- Como resultado do *processo de elicitação* é desenvolvido o documento de requisitos do sistema.
- Este documento contém a especificação de todos os requisitos funcionais e de qualidade do software, incluindo as capacidades do produto, os recursos disponíveis, os benefícios e os critérios de aceitação.

2. DOCUMENTO DE REQUISITOS

- Erros mais comuns cometidos no desenvolvimento do documento de requisitos:
 - ☐ ignorar um grupo de clientes;
 - ☐ ignorar um único cliente;
 - ☐ omitir um grupo de requisitos;
 - ☐ permitir inconsistências entre grupos de requisitos;
 - ☐ aceitar requisito inadequado;
 - ☐ aceitar requisito incorreto, indefinido, ou impreciso;
 - ☐ aceitar um requisito ambíguo e inconsistente;

2. *DOCUMENTO DE REQUISITOS*

1) O documento de requisitos do sistema deve ser composto por sentenças em linguagem natural, seguindo determinados padrões:

- 1) Iniciar com “***O sistema deve ...***”.
- 2) Usar frases curtas.

Exemplo: “***O sistema deve*** rodar em microcomputadores da linha IBM PC que possuam microprocessador Pentium III ou superior.”

2. DOCUMENTO DE REQUISITOS

2) Os requisitos devem estar organizados logicamente e podem estar organizadas de diversas formas:

- a) Requisitos funcionais e de qualidade.
- b) Sequência de execução:
Entrada, Processamento, Saída.
- c) Todas as entradas, todas as saídas, etc.

Muitas vezes, supõe-se que o usuário elabora este documento.

2. DOCUMENTO DE REQUISITOS

- 3) Cada requisito deve ter um identificador único, por exemplo, um identificador numérico, para posterior referência.
- 4) Os requisitos do software devem estar divididos em requisitos funcionais e não funcionais (de qualidade).
- 5) Os requisitos não devem conter detalhes de implementação, o que não é conveniente nesta fase de desenvolvimento. É importante não utilizar termos relacionados à implementação, tais como “arquivo” e “menu”.



2. DOCUMENTO DE REQUISITOS

- 6) Explicação dos termos do domínio da aplicação não devem estar presentes nos requisitos, devendo aparecer em um vocabulário do domínio da aplicação.
- 7) Manter consistente uso dos termos do domínio de aplicação.

Características de Qualidade

- Devem estar contidas no documento de requisitos
- ISO/IEC 9126 define seis características:
 - ☐ *Funcionalidade,*
 - ☐ *Usabilidade,*
 - ☐ *Confiabilidade,*
 - ☐ *Eficiência,*
 - ☐ *Manutenibilidade e*
 - ☐ *Portabilidade.*

Características de Qualidade

- *Funcionalidade*: “Conjunto de atributos que evidenciam a existência de um conjunto de funções e suas propriedades especificadas. As funções são as que satisfazem as necessidades explícitas e implícitas”.
- *Usabilidade*: “Conjunto de atributos que evidenciam o esforço necessário para se poder utilizar o software, bem como o julgamento individual desse uso, por um conjunto explícito ou implícito de usuários”.

Características de Qualidade

- *Confiabilidade*: “Conjunto de atributos que evidenciam a capacidade do software de manter seu nível de desempenho sob condições estabelecidas durante um período de tempo estabelecido”
- *Eficiência*: “Conjunto de atributos que evidenciam o relacionamento entre o nível de desempenho do software e a quantidade de recursos usados, sob condições estabelecidas”

Características de Qualidade

- *Manutenibilidade*: “Conjunto de atributos que evidenciam o esforço necessário para fazer modificações especificadas no software”
- *Portabilidade*: “Conjunto de atributos que evidenciam a capacidade do software de ser transferido de um ambiente para outro”