

Engenharia de Requisitos de Software

Síntese

Este documento sintetiza os conceitos, classificações e processos fundamentais da Engenharia de Requisitos, uma disciplina crítica para o desenvolvimento de software. A fase de requisitos é a mais árdua e de maior impacto no sucesso de um projeto, onde erros são mais difíceis e caros de corrigir posteriormente. A citação do engenheiro da IBM, Fred Brooks, ressalta que "A parte mais árdua na construção de um sistema de software consiste em decidir o que construir!".

Os principais pontos abordados são:

- **Definição de Requisito:** Uma condição ou propriedade indispensável que um produto ou serviço deve atender para satisfazer um objetivo ou ser útil aos seus usuários.
- **Classificação Central:** A distinção mais tradicional e importante é entre **Requisitos Funcionais**, que definem *o que* o sistema deve fazer (suas ações e funcionalidades), e **Requisitos Não-Funcionais**, que definem *como* o sistema deve operar, impondo restrições de qualidade, desempenho, segurança e usabilidade. Frequentemente, os requisitos não-funcionais são mais críticos que os funcionais.
- **O Processo de Engenharia de Requisitos:** Uma abordagem sistemática e formal que engloba todas as atividades para produzir e manter um documento de requisitos. **O modelo de processo de Ian Sommerville, amplamente referenciado, estrutura-se nas seguintes fases:**
 1. **Estudo de Viabilidade:** Uma análise inicial para avaliar se o sistema é viável técnica, financeira e organizacionalmente.
 2. **Elicitação e Análise:** O processo de descobrir, organizar, negociar e documentar os requisitos em colaboração com os stakeholders, utilizando diversas técnicas como entrevistas, prototipação e histórias de usuário.
 3. **Especificação:** A tradução das informações coletadas em um documento formal que serve como contrato entre clientes e desenvolvedores, detalhando os requisitos de usuário (abstratos) e de sistema (técnicos).
 4. **Validação:** A verificação rigorosa do documento de requisitos para garantir que ele seja completo, consistente, realista e que defina o sistema que o cliente realmente deseja.
 5. **Gerenciamento:** O processo contínuo de controlar e rastrear as mudanças nos requisitos ao longo do ciclo de vida do projeto, utilizando mecanismos como matrizes de rastreabilidade.

A complexidade da comunicação entre clientes e desenvolvedores é um desafio central, onde o que é solicitado, entendido e implementado pode variar drasticamente. A Engenharia de Requisitos fornece as ferramentas e processos para mitigar esses riscos, aumentando a probabilidade de sucesso do projeto.

1. Conceitos Fundamentais de Requisitos

1.1. Definição Geral

Um **requisito** é definido como uma condição necessária e indispensável ou uma exigência. No contexto de engenharia, é uma propriedade ou comportamento que um produto ou serviço deve atender. Exemplos práticos incluem pré-requisitos em um currículo acadêmico (ex: cursar "Estrutura de Dados" exige ter cursado "Computação Básica") ou as condições para se candidatar à Presidência da República no Brasil (ex: ser brasileiro nato, ter mais de 35 anos).

1.2. Definição em Engenharia de Software

Em Engenharia de Software, um requisito é uma condição que deve ser satisfeita para alcançar um objetivo ou uma qualidade que o sistema deve prover para ser útil aos seus usuários.

2. Classificação de Requisitos

Os requisitos de software podem ser classificados de diversas formas, sendo a mais importante a distinção entre funcionais e não-funcionais.

2.1. Quanto ao Nível de Abstração

Essa classificação considera o nível de detalhe e o público-alvo do requisito.

- **Requisitos de Usuário:** Descrições de alto nível, em linguagem natural e com diagramas, sobre os serviços que o sistema deve fornecer e suas restrições operacionais. São direcionados a pessoas leigas e possuem poucos detalhes técnicos.
 - *Exemplo:* "O sistema deve gerar um relatório de acompanhamento mensal e enviar para os usuários por e-mail."
- **Requisitos de Sistema:** Descrições detalhadas sobre as funções, operações e restrições que definem exatamente o que deve ser implementado. São direcionados a pessoas experientes (desenvolvedores) e possuem baixo nível de abstração.
 - *Exemplo:* "O sistema deve gerar um relatório com índices a partir de views materializadas geradas a partir de um banco multidimensional."

Ambos os tipos podem ser tanto funcionais quanto não-funcionais.

2.2. Quanto à Funcionalidade (Classificação mais tradicional)

Esta é a classificação mais cobrada e essencial na Engenharia de Requisitos.

- **Requisitos Funcionais:** Descrevem as ações e funcionalidades que o sistema deve executar. Respondem à pergunta "*o que* o sistema deve fazer".
 - **Características:** Declaram serviços, reações a entradas específicas e comportamentos em certas situações. Fazem parte da arquitetura do sistema.
 - *Exemplos:*
 - "O sistema deverá fornecer opção de filtrar e-mails por Assunto e Anexos." (Funcionalidade)

- "O sistema deve calcular a média aritmética das duas maiores dentre três notas de cada disciplina." (Regra de negócio)
- "O sistema deve enviar automaticamente para o e-mail dos gestores autorizados um relatório com o número de alunos inadimplentes." (Processo)
- **Requisitos Não-Funcionais:** Descrevem restrições e condições sob as quais o sistema deve operar. Respondem à pergunta "*como* o sistema deve ser".
 - **Características:** Definem restrições globais sobre qualidade, desempenho, segurança, usabilidade, confiabilidade e custos. São parte da arquitetura técnica e podem ser mais críticos que os requisitos funcionais. Devem ser sempre verificáveis.
 - **Exemplos:**
 - "O sistema deverá fornecer disponibilidade mínima de 99,8%." (Confiabilidade)
 - "O sistema deve responder à solicitação de geração do boletim de um aluno em no máximo 10 segundos." (Desempenho)
 - "O sistema deverá ser capaz de rodar com apenas 1Gb de RAM." (Restrição de hardware)



- **Requisitos de Domínio:** Derivados da área de negócio específica da aplicação. Podem ser funcionais ou não-funcionais e, se não satisfeitos, podem inviabilizar o sistema.
 - **Características:** Frequentemente descritos com jargões do domínio.
 - **Exemplos:**
 - "O sistema deverá calcular o valor do Título Público NTN-B multiplicado pelo valor da Taxa SELIC do mês corrente." (Domínio financeiro)
 - "O sistema deverá funcionar em uma Câmera 4K captando infravermelho e resistente a variações de temperatura." (Domínio aeroespacial)

2.3. Quanto à Qualidade (Baseado em QFD)

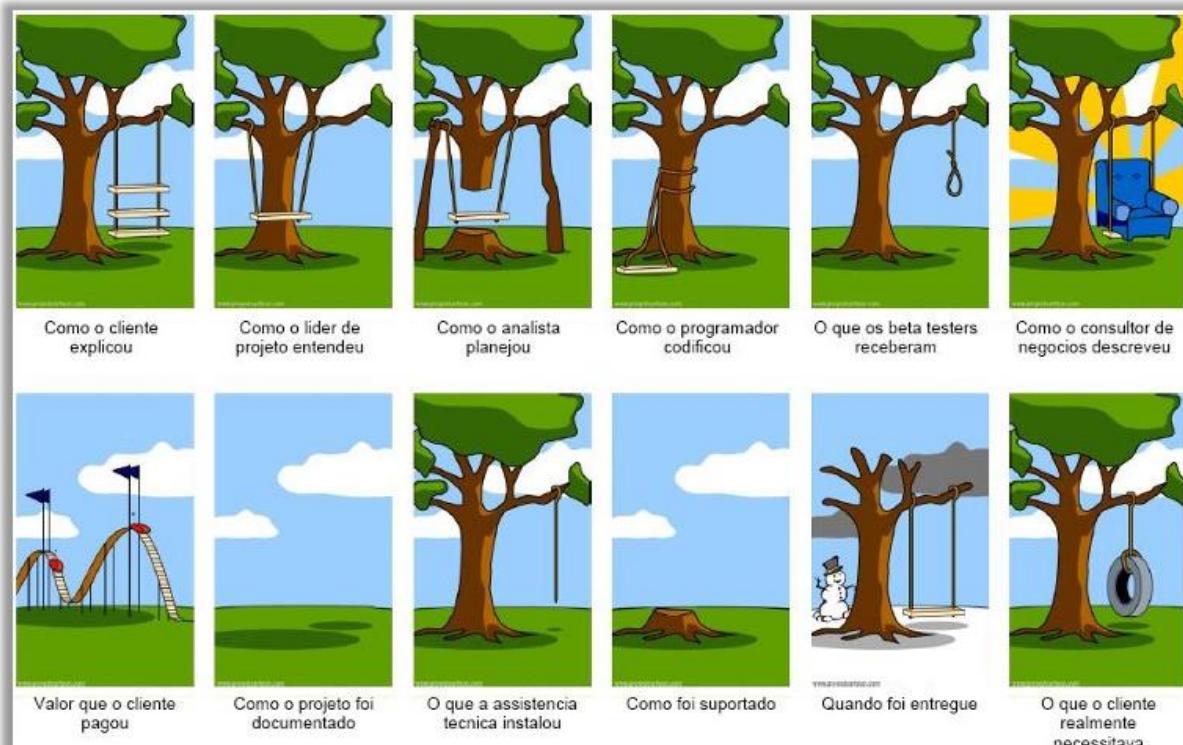
A técnica *Quality Function Deployment* (QFD) classifica os requisitos com base na satisfação do cliente.

- **Requisitos Normais:** Objetivos e metas explicitamente estabelecidos. Se presentes, o cliente fica satisfeito. Ex: "Permitir buscar um livro pelo seu título."
- **Requisitos Esperados:** Fundamentais e implícitos, o cliente não os declara explicitamente. Sua ausência causa grande insatisfação. Ex: "Ao buscar um livro, o sistema deve listar quais livros foram encontrados."
- **Requisitos Fascinantes:** Vão além das expectativas do cliente e geram grande satisfação quando presentes. Ex: "O sistema da biblioteca possui tecnologia assistiva, como leitor de tela para deficientes visuais."

2.4. Quanto à Evolução

- **Requisitos Permanentes (Estáveis):** Ligados à atividade principal da organização. Mudam lentamente e derivam do modelo de domínio. Ex: um sistema para a Bolsa de Valores sempre terá requisitos relacionados a ações, cotações e índices.
- **Requisitos Voláteis (Instáveis):** Específicos de uma implementação e propensos a mudanças. Subdividem-se em:
 - **Mutáveis:** Mudam devido a alterações no ambiente (ex: leis fiscais).
 - **Emergentes:** Emergem conforme o cliente comprehende melhor o sistema.
 - **Consequentes:** Surgem como consequência do uso do sistema.
 - **De Compatibilidade:** Dependem de outros sistemas ou componentes.

Lembre-se que requisitos são entidades vivas que vão mudando o tempo todo por diversos motivos. Isso ocorre, porque as pessoas envolvidas desenvolvem uma compreensão maior do que desejam que o software faça; a própria organização que está comprando o sistema muda. modificações são feitas no hardware, software e no ambiente organizacional do sistema; entre outros.



Um sólido processo de engenharia de requisitos é capaz de encontrar a melhor solução viável no momento. Na prática, é praticamente impossível satisfazer absolutamente todas as expectativas dos clientes, na medida em que requisitos podem ser bastante complexos e voláteis.

2.5. Quanto à Origem (Subclassificação de Requisitos Não-Funcionais)

- **Requisitos de Produto:** Especificam o comportamento do produto (desempenho, confiabilidade, usabilidade, portabilidade).
- **Requisitos Organizacionais:** Derivados de políticas da organização (padrões de processo, linguagem de programação, prazos de entrega).
- **Requisitos Externos:** Derivados de fatores externos (requisitos legais, éticos, de interoperabilidade).

3. O Processo de Engenharia de Requisitos

3.1. Definição, Importância e Desafios

A Engenharia de Requisitos é uma abordagem sistemática e formal para a formulação, análise, documentação e manutenção de requisitos. É a fase mais crítica do desenvolvimento, pois erros neste estágio levam a problemas graves e custos elevados de correção. Dados indicam que 50% dos principais defeitos de software e 12% dos fracassos em projetos são oriundos de problemas na fase de requisitos.

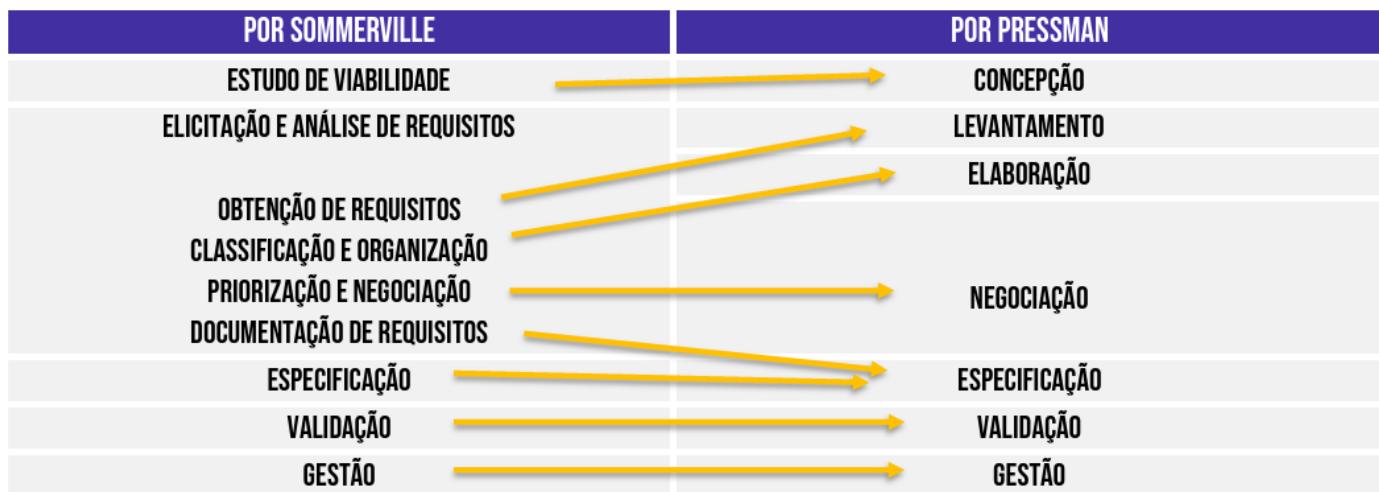
Os desafios são inúmeros:

- Usuários que não sabem expressar suas necessidades.
- Requisitos contraditórios entre diferentes stakeholders.
- Processos de trabalho imaturos na organização cliente.
- Requisitos que mudam constantemente (são "entidades vivas").

3.2. Modelos de Processo: Sommerville vs. Pressman

Existem divergências entre autores sobre as fases do processo. Os dois modelos principais são:

Fases segundo Ian Sommerville	Fases segundo Roger Pressman
Estudo de Viabilidade	Concepção
Elicitação e Análise de Requisitos	Levantamento
Especificação de Requisitos	Elaboração
Validação de Requisitos	Negociação
Gestão de Requisitos	Especificação
	Validação
	Gestão



FASES DE ACORDO COM IAN SOMMERVILLE

ESTÃO VIABILIZANDO ELIANA ESPECIALMENTE NO VASCO DA GAMA

ESTUDO DE VIABILIDADE > ELICITAÇÃO E ANÁLISE DE REQUISITOS > ESPECIFICAÇÃO > VALIDAÇÃO > GESTÃO

FASES DE ACORDO COM ROGER PRESSMAN

CENAS LAMENTÁVEIS EM NOVO EMPATE DO VASCO DA GAMA

CONCEPÇÃO > LEVANTAMENTO > ELABORAÇÃO > NEGOCIAÇÃO > ESPECIFICAÇÃO > VALIDAÇÃO > GESTÃO

O modelo de Sommerville é detalhado a seguir, por ser o mais cobrado em provas.

4. Análise Detalhada das Fases (Modelo de Sommerville)

4.1. Estudo de Viabilidade

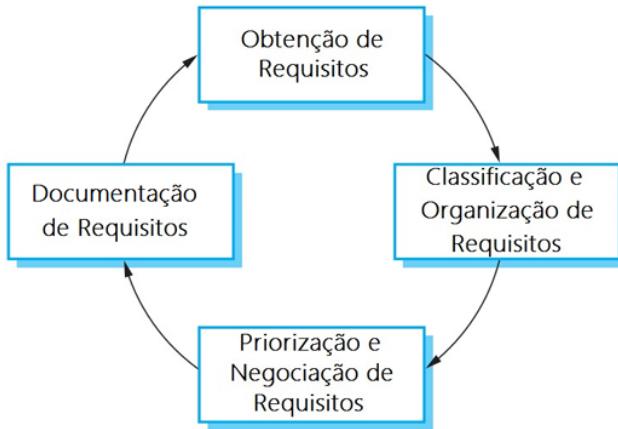
Atividade inicial, rápida e de baixo custo, que avalia se as necessidades podem ser satisfeitas com a tecnologia atual, dentro do custo e prazo. O objetivo é fornecer informações para a decisão de prosseguir (*go/no-go*). **Questões-chave:**

1. O sistema contribui para os objetivos da organização?
2. Pode ser implementado com a tecnologia, custo e prazo disponíveis?
3. Pode ser integrado a outros sistemas existentes? **Resultado:** Relatório de Viabilidade.

4.2. Elicitação e Análise de Requisitos

Processo de descobrir, derivar e analisar os requisitos do sistema em colaboração com os stakeholders.

Atividades principais:



- Obtenção de Requisitos:** Interação com stakeholders para coletar os requisitos. Os requisitos de domínio também são descobertos durante essa atividade.
- Classificação e Organização:** envolve a coleção de requisitos não estruturados, agrupa os requisitos relacionados e os organiza em conjuntos coerentes.
- Priorização e Negociação:** inevitavelmente, os requisitos serão conflitantes. Assim, busca-se priorizar os requisitos e resolver conflitos por meio da negociação.
- Documentação de Requisitos:** Os requisitos são documentados e colocados na próxima volta da espiral. Podem ser produzidos documentos de requisitos formais ou informais.

4.3. Principais Técnicas de Elicitação

Técnica	Descrição
Entrevistas	Conversas formais (roteiro definido) ou informais (abertas) com stakeholders para obter um entendimento geral de suas necessidades.
Etnografia	Observação do usuário em seu ambiente de trabalho para descobrir requisitos implícitos e compreender processos reais.
Cenários	Descrições de exemplos de interação do usuário com o sistema, úteis para detalhar requisitos.
Histórias de Usuário	Descrições concisas de uma funcionalidade na perspectiva do usuário, seguindo o formato: "Como um <papel>, eu quero <meta> de modo que <benefício>". Inclui os 3 C's (Cartão, Conversação, Confirmação) e Critérios de Aceitação.
Prototipação	Criação de um modelo executável do sistema para que os usuários possam experimentá-lo e fornecer feedback antecipado.
Workshop de Requisitos	Reunião estruturada e intensiva com analistas e stakeholders, facilitada por um mediador neutro, para definir requisitos em conjunto.
Brainstorming	Sessão de grupo informal para gerar um grande volume de ideias sem críticas, explorando a criatividade coletiva.

JAD (Joint Application Design)	Workshop estruturado, similar ao Workshop de Requisitos, que utiliza dinâmicas de grupo e técnicas visuais para promover a tomada de decisões.
Leitura de Documentos	Análise de documentos existentes (manuais, relatórios, leis) para identificar requisitos, especialmente em sistemas legados.
Reúso de Requisitos	Utilização de especificações de sistemas similares, o que pode economizar tempo, reduzir custos e mitigar riscos.
Questionários	Formulários com perguntas predefinidas, úteis para coletar informações de um grande número de stakeholders.

4.4. Especificação de Requisitos

Atividade de traduzir as informações coletadas em um documento formal, que serve como um contrato entre clientes e desenvolvedores. Este documento contém tanto **Requisitos de Usuário** (abstratos, em linguagem natural) quanto **Requisitos de Sistema** (detalhados, técnicos).

Idealmente, deve ser claro, não-ambíguo, completo e consistente. **Resultado:** Documento de Requisitos (pode ser texto, modelo gráfico, etc.).

4.5. Validação de Requisitos

Processo de verificar se o documento de requisitos define o sistema que o cliente realmente deseja e se os requisitos são de alta qualidade. É uma etapa crítica para evitar custos de retrabalho.

Técnicas de Validação:

- **Revisão de Requisitos:** Análise sistemática dos requisitos por uma equipe de revisores (clientes e desenvolvedores) para identificar erros, omissões e inconsistências.

ATRIBUTOS	DESCRIÇÃO
VALIDADE	Examina se as partes interessadas que contribuíram com o levantamento de requisitos aceitam a especificação final obtida.
CONSISTÊNCIA	Examina se existem conflitos entre os requisitos identificados.
COMPREENSIBILIDADE	Examina se os requisitos são compreendidos de forma inequívoca pelas partes interessadas.
COMPLETUDEN	Examina se todas as funcionalidades pretendidas fazem parte da especificação do sistema.
REALISMO	Examina se, dadas as restrições do projeto (tecnológicas, financeiras e temporais), o sistema especificado é implementável.
VERIFICABILIDADE	Examina se os requisitos foram descritos de forma que seja possível verificar se foram ou não implementados.
RASTREABILIDADE	Examina se a origem de cada requisito está claramente identificada.
ADAPTABILIDADE	Examina se os requisitos podem sofrer alterações sem produzir efeitos em outros requisitos.
CONFORMIDADE COM NORMAS	Examina se a especificação obedece às normas técnicas utilizadas para o desenvolvimento do sistema.

- **Prototipação:** Uso de um protótipo para validar se os requisitos elicitados correspondem às expectativas do usuário.
- **Geração de Casos de Teste:** O desenvolvimento de testes para cada requisito pode revelar ambiguidades e problemas de verificabilidade.

Atributos verificados na validação:

TIPO DE VERIFICAÇÃO	DESCRIÇÃO
VERIFICAÇÃO DE VALIDADE	Estudos e análises podem identificar que funções adicionais e diferentes daquelas levantadas pelos usuários são necessárias. Os sistemas têm diversos stakeholders com necessidades diferentes e qualquer conjunto de requisitos é, inevitavelmente, um compromisso.
VERIFICAÇÃO DE CONSISTÊNCIA	Os requisitos em um documento não devem ser conflitantes e contraditórios. Isso significa que não devem existir restrições ou descrições contraditórias para a mesma função do sistema, porque isso gera problemas sérios de inconsistência.
VERIFICAÇÃO DE COMPLETEZA	O documento de requisitos deve incluir requisitos que definam todas as funções e as restrições desejadas pelo usuário do sistema. Isso torna o documento de requisitos bastante verboso e pesado, no entanto é um requisito para alcançar a completeza do requisito.
VERIFICAÇÃO DE REALISMO	Usando o conhecimento da tecnologia existente, os requisitos devem ser verificados quanto a se realmente podem ser implementados. Essas verificações também devem levar em consideração o orçamento e o prazo para o desenvolvimento do sistema – caiu questão de prova recente sobre isso!
FACILIDADE DE VERIFICAÇÃO	Para reduzir o potencial de divergências entre cliente e fornecedor, os requisitos do sistema devem sempre ser escritos de modo que sejam verificáveis. Isso significa que você deve ser capaz de escrever um conjunto de testes que possa demonstrar que o sistema entregue atende a cada requisito especificado.

Verificação X Validação de Requisitos:

A diferença entre Verificação e Validação de Requisitos é, em geral, ignorada em prova – ambos são chamados apenas de Validação de Requisitos.

Caso cobrem, eis a diferença:

Verificação de Requisitos: tem o objetivo de descobrir se os requisitos são claros, precisos, completos e consistentes, e analisar se os modelos construídos estão de acordo com os requisitos definidos.

Validação de Requisitos: Ela se ocupa de mostrar que os requisitos realmente definem o sistema que o cliente deseja, isto é, visa assegurar que as necessidades do cliente estão sendo atendidas por tais requisitos.

4.6. Gerenciamento de Requisitos

Processo contínuo que envolve todas as outras fases, responsável por compreender, acompanhar e controlar as mudanças nos requisitos ao longo do tempo. Como os requisitos são voláteis, é essencial ter um processo formal para gerenciar propostas de mudança e avaliar seu impacto.

Ferramenta Central:

Rastreabilidade A rastreabilidade é a capacidade de acompanhar as ligações entre requisitos e outros artefatos do projeto. É frequentemente gerenciada por meio de **matrizes de rastreabilidade**, que registram dependências.

Tipos de Informação de Rastreabilidade:

- **De Origem:** Liga o requisito ao stakeholder que o propôs e à sua justificativa.

- **De Requisitos:** Liga requisitos dependentes entre si.
- **De Projeto:** Liga o requisito aos módulos do sistema onde ele é implementado.

Sommerville afirma que o objetivo da engenharia de requisitos é criar e manter um documento de requisitos de sistema. Assim, note que o resultado do *Estudo de Viabilidade* é o *Relatório de Viabilidade*; o resultado da *Elicitação e Análise de Requisitos* é um conjunto de *Modelos do Sistema*; o resultado da *Especificação de Requisitos* são os *Requisitos de Usuário e de Sistema*; e o resultado da *Validação de Requisitos* é o famoso *Documento de Requisitos*.

