



```
for object to mirror...
mirror_mod.mirror_object = ...
operation == "MIRROR_X":
    mirror_mod.use_x = True
    mirror_mod.use_y = False
    mirror_mod.use_z = False
operation == "MIRROR_Y":
    mirror_mod.use_x = False
    mirror_mod.use_y = True
    mirror_mod.use_z = False
operation == "MIRROR_Z":
    mirror_mod.use_x = False
    mirror_mod.use_y = False
    mirror_mod.use_z = True

#selection at the end -add
mirror_ob.select= 1
modifier_ob.select=1
context.scene.objects.active
("Selected" + str(modifier_ob.name))
mirror_ob.select = 0
= bpy.context.selected_objects
data.objects[one.name].select

OPERATOR CLASSES
```

# Aula 03

## Engenharia de requisitos e Análise de Sistema

Profª Otília

Engenharia de Software



# Aula Passada

- Desenvolvimento de software como um projeto e as atividades de gerenciamento
- Planejamento e gerenciamento de riscos.

# Gerenciamento de Riscos

“Que características especiais desse produto podem ameaçar o plano do nosso projeto?”

- **Checklist**

- **Tamanho do produto** – riscos associados ao tamanho geral do software a ser criado ou modificado.
- **Impacto de negócio** – riscos associados a restrições impostas pela gerência ou pelo mercado.
- **Características do cliente** – são riscos associados à sofisticação dos clientes e à habilidade do desenvolvedor em se comunicar com os interessados a tempo.
- **Definição do processo** – riscos associados ao grau em que a gestão de qualidade foi definida e é seguida pela organização de desenvolvimento.
- **Ambiente de desenvolvimento** – riscos associados à disponibilidade e qualidade das ferramentas a ser usadas para criar o produto.
- **Tecnologia a ser criada** – riscos associados à complexidade do sistema a ser criado e com a “novidade” da tecnologia que está embutida no sistema.
- **Quantidade de pessoas e experiência** – riscos associados à experiência técnica em geral e de projeto dos engenheiros de software que farão o trabalho.

Há disponível na Web muitas listas abrangentes para riscos de projeto de software (por exemplo, [Baa07], [NAS07], [Wor04]). Você pode usá-las para ter uma visão dos riscos genéricos para projetos de software.

# Gerenciamento de Riscos

- **Previsão de risco**

**FIGURA 28.1**

Avaliação de impacto  
Fonte: (Boe89)

Componentes		Desempenho	Suporte	Custo	Cronograma
Categoria					
Catastrófico	1	Falha em satisfazer o requisito resultaria em falha da missão		A falha resulta em aumento de custo e atrasos no cronograma com valores previstos que excedem \$ 500 mil	
	2	Degradação significativa até não cumprimento do desempenho técnico	Software que não responde com agilidade ou que é difícil de dar suporte	Dificuldades financeiras significativas, provável estouro no orçamento	Data de entrega não exequível
Crítico	1	Falha em atender o requisito degradará o desempenho do sistema até um ponto no qual o sucesso da missão é questionável		Falha resulta em atrasos operacionais e/ou aumento de custos com valores estimados entre \$ 100 mil e \$ 500mil	
	2	Alguma redução no desempenho técnico	Pequenos atrasos nas modificações de software	Alguma falta de recursos financeiros, possíveis estouros de orçamento	Possível atraso na data de entrega
Marginal	1	Falha em atender o requisito resultaria na degradação de missão secundária		Custos, impactos e/ou atrasos de cronograma recuperáveis com valores estimados de \$ 1 mil a \$ 100 mil	
	2	De mínima a pequena redução no desempenho técnico	Suporte responsivo de software	Recursos financeiros suficientes	Cronograma realístico e possível
Negligenciável	1	Falha em atingir o requisito criaria inconveniência ou impacto não operacional		Erro resulta em pequeno impacto no custo e/ou cronograma com valor esperado de menos de \$ 1 mil	
	2	Nenhuma redução do desempenho técnico	Software facilmente suportável	Possível sobre no orçamento	Data de entrega pode ser antecipada

# Gerenciamento de Riscos

- A previsão de risco, também chamada de **estimativa de risco**, tenta classificar cada risco de duas maneiras – (1) a possibilidade ou probabilidade de que o risco seja real e (2) as consequências dos problemas associados ao risco, caso ele ocorra. Você trabalha com outros gerentes e pessoal técnico para executar quatro etapas de projeção de risco:
  1. Estabeleça uma escala que reflita a possibilidade detectada de um risco.
  2. Esboce as consequências do risco.
  3. Estime o impacto do risco sobre o projeto e o produto.
  4. Avalie a exatidão geral da projeção de risco para que não haja mal-entendidos.

# Gerenciamento de Riscos

**FIGURA 28.3**

Riscos e preocupação da gerência

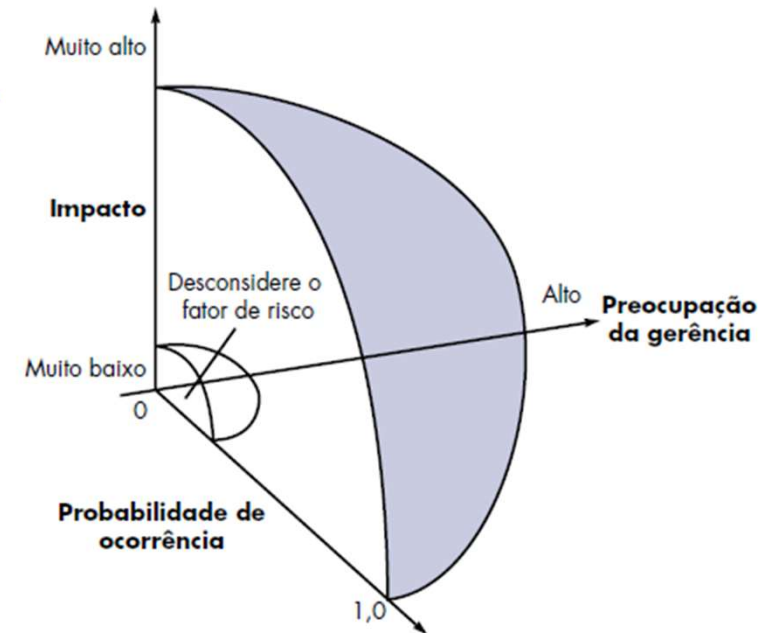
**FIGURA 28.2**

Exemplo de uma tabela de risco antes da ordenação

Riscos	Categoria	Probabilidade	Impacto	RMMM
A estimativa de tamanho pode ser significativamente baixa	PS	60%	2	
Número de usuários maior do que o planejado	PS	30%	3	
Reutilização menor do que a planejada	PS	70%	2	
Os usuários finais resistem ao sistema	BU	40%	3	
O prazo de entrega será apertado	BU	50%	2	
Financiamento será perdido	CU	40%	1	
O cliente mudará os requisitos	PS	80%	2	
A tecnologia não atingirá as expectativas	TE	30%	1	
Falta de treinamento no uso das ferramentas	DE	80%	3	
Pessoal sem experiência	ST	30%	2	
A rotatividade do pessoal será alta	ST	60%	2	
Σ				
Σ				
Σ				

Valores de impacto:

- 1 — catastrófico
- 2 — crítico
- 3 — marginal
- 4 — negligenciável

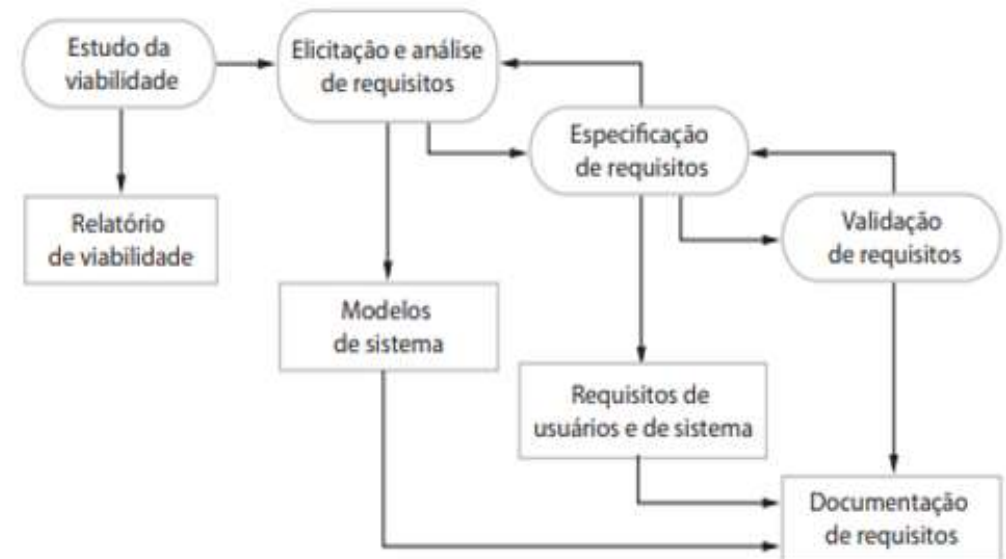


Pressman - pág. 653  
 PS - tamanho do projeto  
 BU - risco de negócio

# Aula 03 - Engenharia de Requisitos e Análise de Sistemas

- **Objetivos:**

- Compreender a importância de o requisito para o desenvolvimento de software e um processo formal para
  - identificar,
  - analisar,
  - validar e
  - gerenciar os requisitos do sistema.





# Situação - problema

- Quando um cliente deseja um software, como nós sabemos quanto tempo demorará para ser desenvolvido?
- Como nós descrevemos para os programadores todas as características que devem ser implementadas?
- De que forma são listadas todas as funcionalidades que o software deve possuir?
- **Resposta: Através dos requisitos!**



# O que é um requisito?

Um sistema sob encomenda

Indagações

O que?

Pra que?

Por que?

Como?

**Desenvolvedor**



**Cliente**



Para automatizar o agendamento de consultas e exames

Porque queremos que o próprio paciente escolha o melhor horário, sem necessitar ocupar nossa secretária

Através de uma ferramenta online

Tentativa de solução do problema

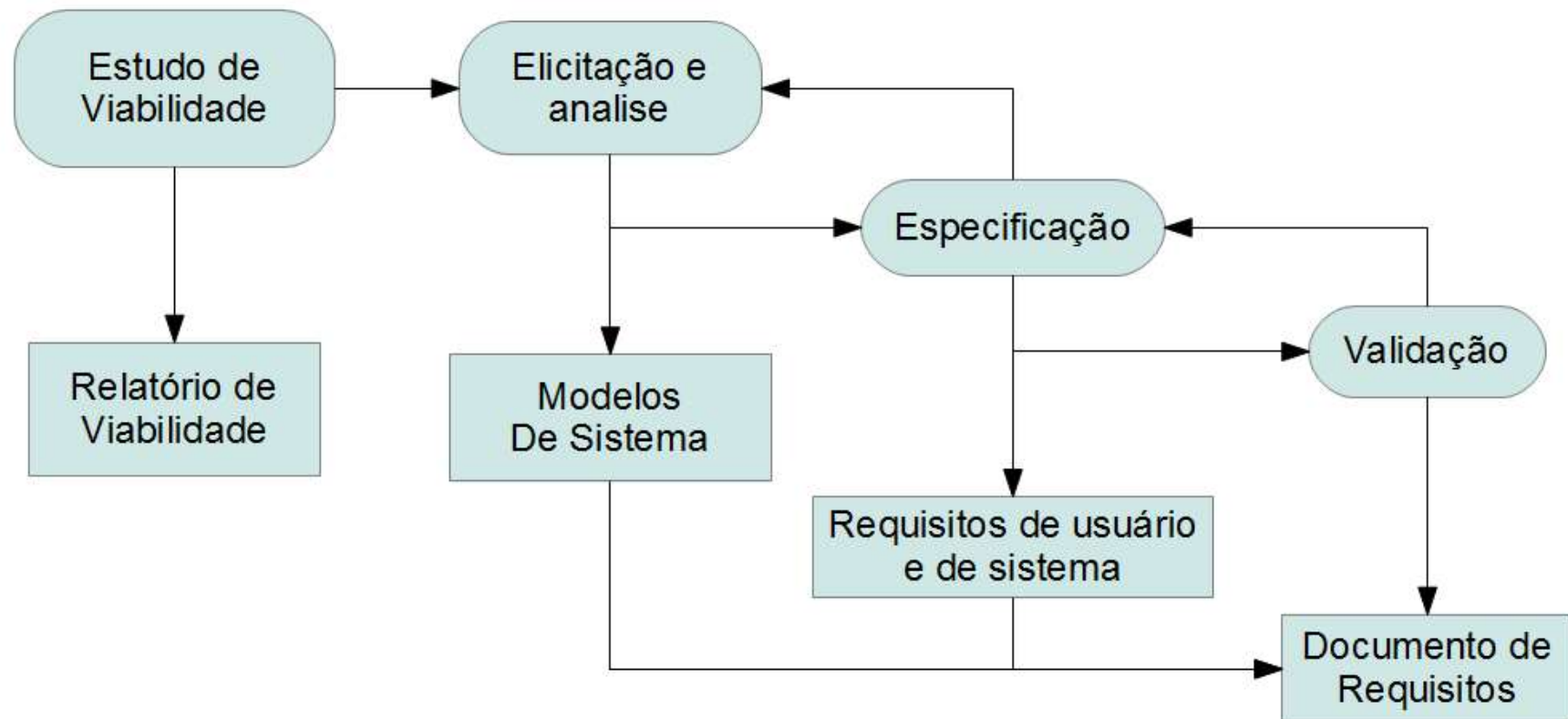
Tentativa de passar informações do negócio



# Introdução

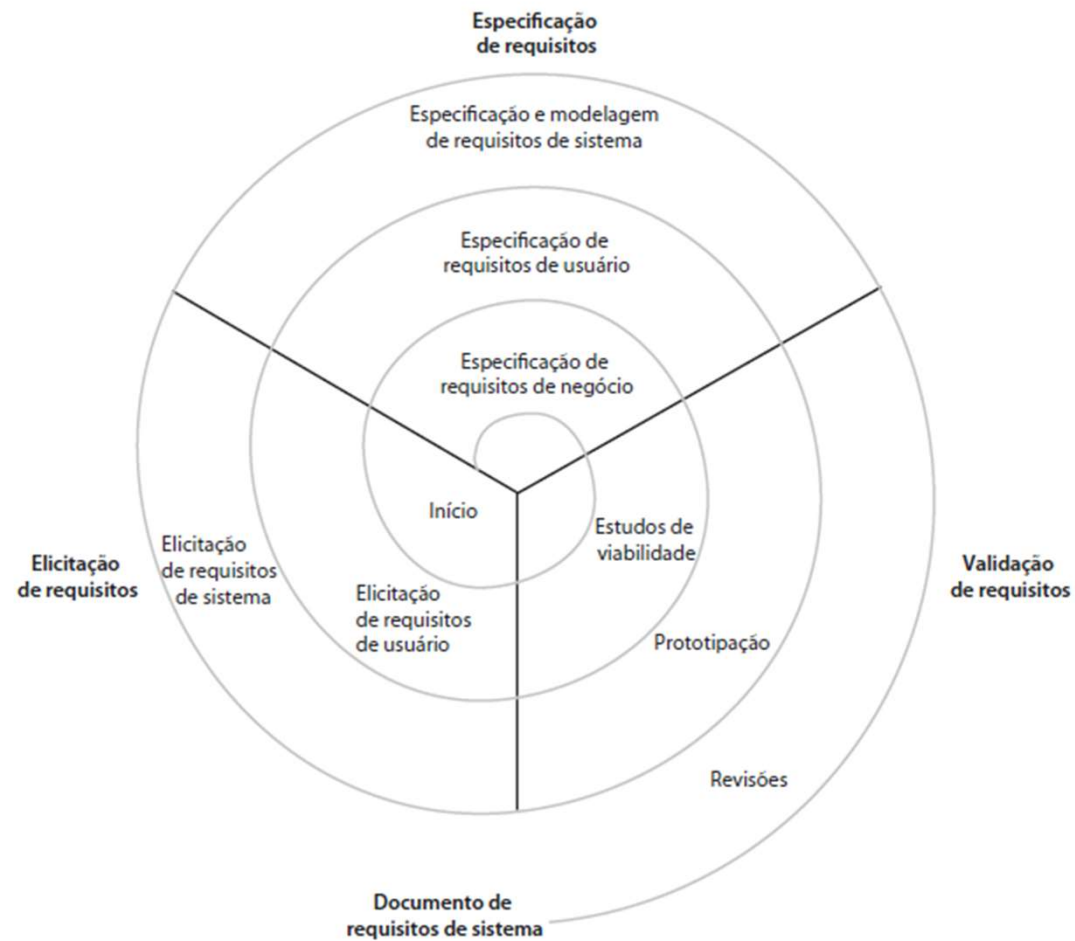
- O processo de Engenharia de Requisitos possui como finalidade a criação e a manutenção do documento de requisitos.
- Foco na qualidade do processo, já que se trata de uma fase importante.
- A engenharia de requisitos possui quatro subprocessos:
  - Estudo de viabilidade
  - Elicitação e análise dos requisitos
  - Especificação
  - Validação.

# Introdução



# Introdução

**Figura 4.5** Uma visão em espiral do processo de engenharia de requisitos.





# Estudo de Viabilidade

- Em novos sistemas, a engenharia de requisitos deve começar com o estudo de viabilidade.
- A entrada para o estudo de viabilidade consiste num conjunto preliminar de requisitos de negócio, um esboço da descrição do sistema e como este pretende apoiar os processos de negócio.
- Os resultados devem estar presentes num relatório de viabilidade, onde recomenda se vale a pena ou não prosseguir como processo.

# Estudo de Viabilidade

- Este estudo é breve e focado em responder uma série de questões, como:
  - O sistema contribui para os objetivos gerais da organização?
  - O sistema pode ser implementado com tecnologia atual e dentro das restrições de prazo e custo?
  - O sistema pode ser integrado a outros sistemas existentes?
  - Como a empresa se comportaria se esse sistema não fosse implementado?
  - Quais são os problemas com os processos atuais e como o novo sistema ajudaria a reduzir esses problemas?

# Estudo de Viabilidade

- Como responder essas questões?
  - Através de conversar com: gerentes de departamento em que o sistema será usado, engenheiros de software familiarizados com o tipo de sistema proposto, especialistas em TI e usuários finais.
- Após obter as informações, a elaboração de um relatório é realizada com a finalidade de recomendar se o desenvolvimento deve prosseguir ou não.



# Elicitação e Análise de Requisitos

- Também podemos chamar de **Levantamento de Requisitos**
  - Nessa atividade, os engenheiros de software trabalham com os clientes e usuários finais do sistema para compreender sobre o domínio da aplicação, os serviços do sistema, desempenho, restrições, etc.
  - Pode envolver vários **stakeholders**.



# Elicitação e Análise de Requisitos

**Figura 4.6**

O processo de elicitação e análise de requisitos.





# Descoberta de requisitos

- Essa é a atividade de interação com os stakeholders do sistema para descobrir seus requisitos.
- Os requisitos de domínio dos stakeholders e da documentação também são descobertos durante essa atividade.
- Os requisitos podem ser obtidos:
  - Entrevistas
  - Cenários
  - Casos de uso



# Entrevistas

- Entrevistas formais ou informais com os stakeholders fazem parte da maioria dos processos de engenharia de requisitos.
- Nessas entrevistas, são formuladas perguntas para os *stakeholders*.
- Os requisitos são derivados das respostas dessas questões.
- As entrevistas podem ser:
  - Fechadas
  - abertas
- Na prática, as entrevistas são uma combinação desses tipos.



# Entrevistas

- As informações obtidas nas entrevistas complementam outras informações obtidas de outras fontes (ex: documentação)
- Se utilizarmos apenas a entrevista como meio de obter requisitos, pode haver perda de informações essenciais.
- Logo, essa técnica deve ser utilizada em conjunto com outras técnicas de elicitação.



# Cenários

- Geralmente as pessoas consideram mais fácil relatar exemplos da vida real do que abstrair descrições.
- Elas podem compreender e criticar um cenário de como seria a interação com o sistema.
- O cenário começa com um esboço da interação e, durante o processo de elicitação, os detalhes são adicionados para uma descrição completa dessa interação.

# Cenário

## Quadro 4.4 Cenário para a coleta do histórico médico em MHC-PMS.

### Suposição inicial:

O paciente é atendido em uma clínica médica por uma recepcionista; ela gera um registro no sistema e coleta suas informações pessoais (nome, endereço, idade etc.). Uma enfermeira é conectada ao sistema e coleta o histórico médico do paciente.

### Normal:

A enfermeira busca o paciente pelo sobrenome. Se houver mais de um paciente com o mesmo sobrenome, o nome e a data de nascimento são usados para identificar o paciente.

A enfermeira escolhe a opção do menu para adicionar o histórico médico.

A enfermeira segue, então, uma série de *prompts* do sistema para inserir informações sobre consultas em outros locais, os problemas de saúde mental (entrada de texto livre), condições médicas (enfermeira seleciona condições do menu), medicação atual (selecionado no menu), alergias (texto livre) e informações da vida doméstica (formulário).

### O que pode dar errado:

O prontuário do paciente não existe ou não pôde ser encontrado. A enfermeira deve criar um novo registro e registrar as informações pessoais.

As condições do paciente ou a medicação em uso não estão inscritas no menu. A enfermeira deve escolher a opção 'outros' e inserir texto livre com descrição da condição/medicação.

O paciente não pode/não fornecerá informações sobre seu histórico médico. A enfermeira deve inserir um texto livre registrando a incapacidade/relutância do paciente em fornecer as informações. O sistema deve imprimir o formulário-padrão de exclusão afirmando que a falta de informação pode significar que o tratamento será limitado ou postergado. Este deverá ser assinado e entregue ao paciente.

### Outras atividades:

Enquanto a informação está sendo inserida, o registro pode ser consultado, mas não editado por outros agentes.

### Estado do sistema na conclusão:

O usuário está conectado. O prontuário do paciente, incluindo seu histórico médico, é inserido no banco de dados e um registro é adicionado ao *log* do sistema, mostrando o tempo de início e fim da sessão e a enfermeira envolvida.





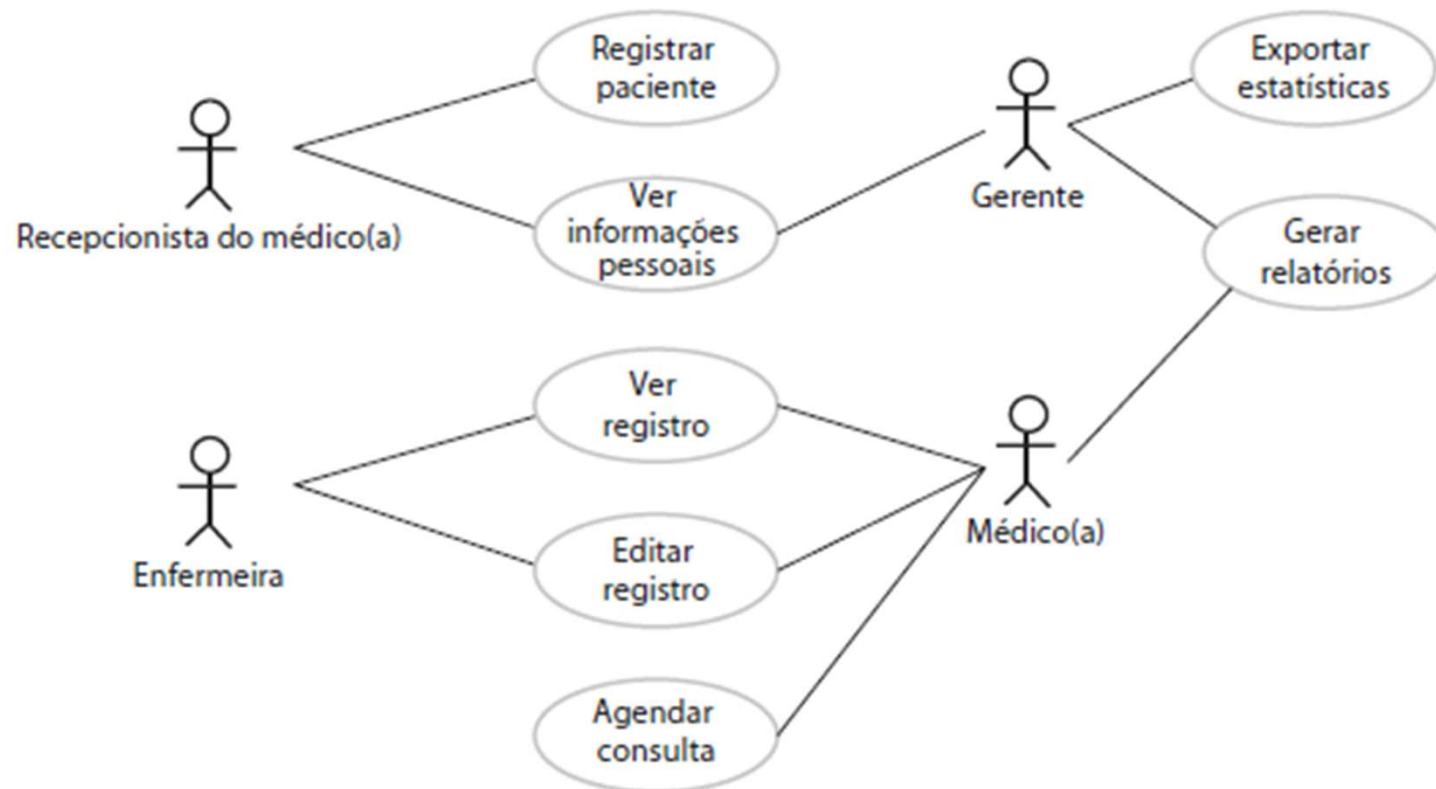
# Casos de Uso

- Utiliza a linguagem de modelagem UML
- É uma técnica baseada em cenários para elicitação de requisitos.
- Os casos de uso identificam interações individuais como sistema.

# Casos de Uso

**Figura 4.6**

Casos de uso para o MHC-PMS.





# Validação de Requisitos

- É a fase que possui a finalidade de mostrar que os requisitos realmente definem o sistema que o usuário deseja.
- Também está relacionada com a descoberta de problemas com os requisitos.
- Erros em um documento de requisitos podem levar a custos excessivos.
- Devem ser realizadas verificações nos requisitos do sistema.

# Validação de Requisitos

- As verificações podem ser:
  - Verificação de validade
  - Verificação de consistência
  - Verificação de clareza
  - Verificação de realismo
  - Facilidade de verificação
- Uma série de técnicas de validação pode ser utilizada:
  - Revisão de requisitos
  - Prototipação
  - Geração de casos de teste

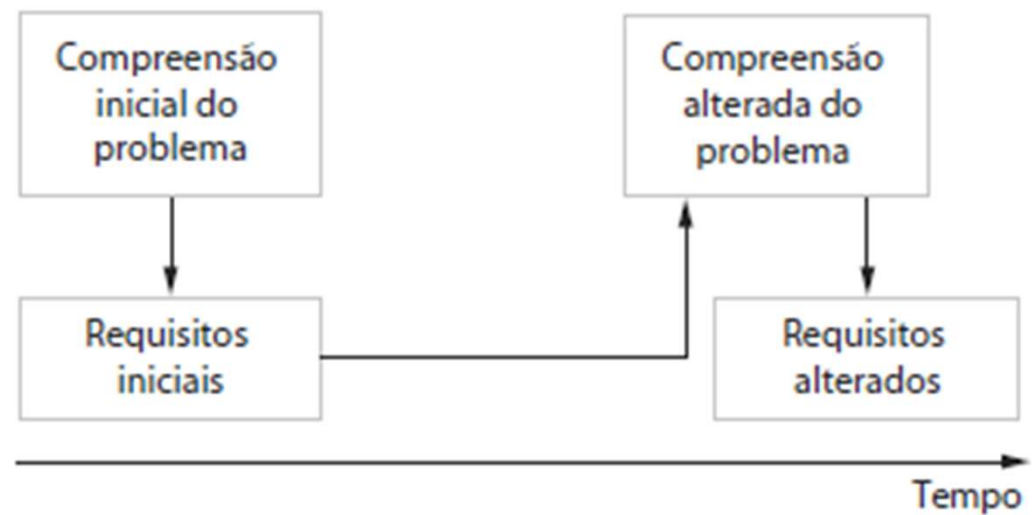
# Gerenciamento de Requisitos

- Os requisitos para sistemas de software de grande porte estão sempre mudando. Uma razão para isso é que esses sistemas geralmente são desenvolvidos para enfrentar os problemas 'maus' – problemas que não podem ser completamente definidos. Porque o problema não pode ser totalmente definido, os requisitos de software são obrigados a ser incompletos.
- Durante o processo de software, o entendimento dos *stakeholders* a respeito do problema está em constante mutação

# Gerenciamento de Requisitos

**Figura 4.8**

Evolução dos requisitos.



# Requisitos

- **Requisitos permanentes.**

- Derivados da atividade principal da organização.
  - Em um hospital sempre haverá requisitos relativos aos pacientes, aos médicos, às enfermeiras e aos tratamentos. Derivados do modelo do domínio.

- **Requisitos voláteis**

- Requisitos que se modificam durante o desenvolvimento ou quando o sistema está em uso. Requisitos resultantes de políticas governamentais.
  - Plano de saúde.



# Classificação dos Requisitos Voláteis

- **Requisitos mutáveis**

- Requisitos que se modificam por causa do ambiente do sistema.

- **Requisitos emergentes**

- Requisitos que surgem à medida que a compreensão do cliente do sistema se desenvolve.

- **Requisitos consequentes**

- Requisitos que resultam da introdução do sistema de computador.

- **Requisitos de compatibilidade**

- Requisitos que dependem de outros sistemas ou processos de negócio específicos dentro da organização.

# Requisitos funcionais e não funcionais

# Requisitos Funcionais e Não-funcionais

- **Requisitos funcionais.**

- São declarações de serviços que o sistema deve fornecer, de como o sistema deve reagir a entradas específicas e de como o sistema deve se comportar em determinadas situações. Em alguns casos, os requisitos funcionais também podem explicitar o que o sistema não deve fazer.

- **Requisitos não funcionais.**

- São restrições aos serviços ou funções oferecidos pelo sistema. Incluem restrições de timing, restrições no processo de desenvolvimento e restrições impostas pelas normas. Ao contrário das características individuais ou serviços do sistema, os requisitos não funcionais, muitas vezes, aplicam-se ao sistema como um todo.

# Requisitos Funcionais

- Requisitos funcionais do sistema variam de requisitos gerais, que abrangem o que o sistema deve fazer, até requisitos muito específicos, que refletem os sistemas e as formas de trabalho em uma organização.
- Por exemplo, aqui estão os exemplos de requisitos funcionais para o sistema MHC-PMS, usados para manter informações sobre os pacientes em tratamento por problemas de saúde mental:
  1. Um usuário deve ser capaz de pesquisar as listas de agendamentos para todas as clínicas.
  2. O sistema deve gerar a cada dia, para cada clínica, a lista dos pacientes para as consultas daquele dia.
  3. Cada membro da equipe que usa o sistema deve ser identificado apenas por seu número de oito dígitos.

# Requisitos Não-Funcionais

- Os requisitos não funcionais, como desempenho, proteção ou disponibilidade, normalmente especificam ou restringem as características do sistema como um todo.
- Requisitos não funcionais são frequentemente mais críticos que requisitos funcionais individuais. Os usuários do sistema podem, geralmente, encontrar maneiras de contornar uma função do sistema que realmente não atenda a suas necessidades. No entanto, deixar de atender a um requisito não funcional pode significar a inutilização de todo o sistema.
- Por exemplo, se um sistema de aeronaves não cumprir seus requisitos de confiabilidade, não será certificado como um sistema seguro para operar; se um sistema de controle embutido não atender aos requisitos de desempenho, as funções de controle não funcionarão corretamente.

# Requisitos Não-Funcionais

## Quadro 4.1 Exemplos de requisitos não funcionais no MHC-PMS.

### Requisito de produto

O MHC-PMS deve estar disponível para todas as clínicas durante as horas normais de trabalho (segunda a sexta-feira, 8h30 às 17h30). Períodos de não operação dentro do horário normal de trabalho não podem exceder cinco segundos em um dia.

### Requisito organizacional

Usuários do sistema MHC-PMS devem se autenticar com seus cartões de identificação da autoridade da saúde.

### Requisito externo

O sistema deve implementar as disposições de privacidade dos pacientes, tal como estabelecido no HStan-03-2006-priv.

# Requisitos Não-Funcionais

**Tabela 4.1** Métricas para especificar requisitos não funcionais.

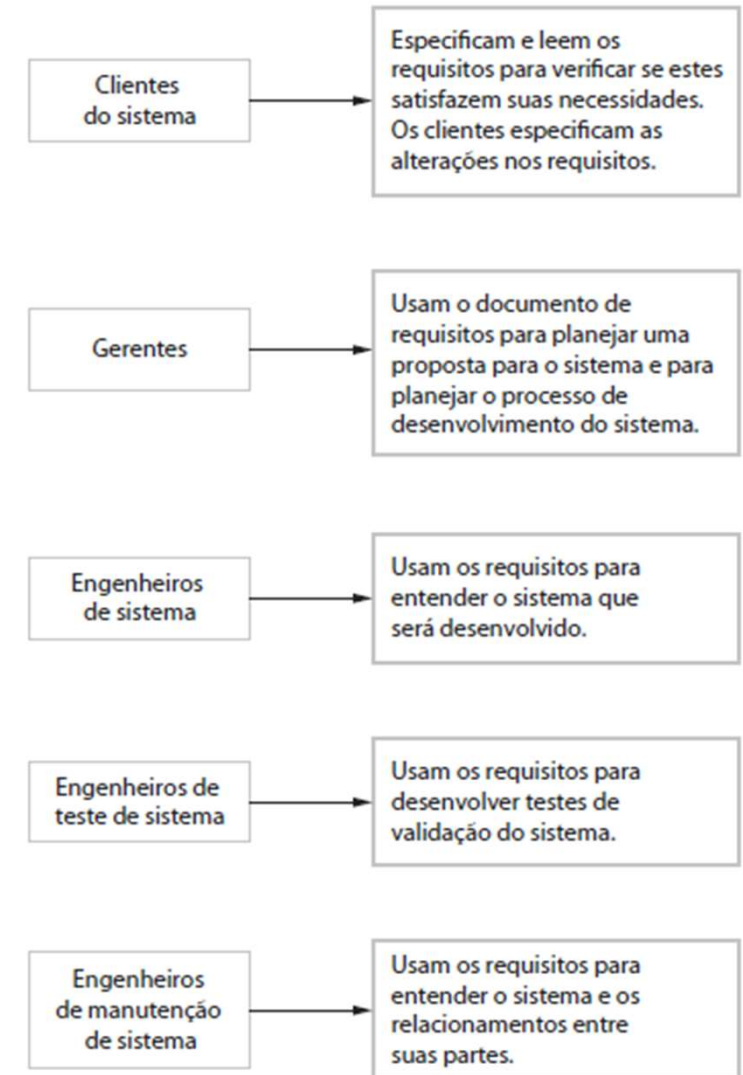
Propriedade	Medida
Velocidade	Transações processadas/segundo Tempo de resposta de usuário/evento Tempo de atualização de tela
Tamanho	Megabytes Número de chips de memória ROM
Facilidade de uso	Tempo de treinamento Número de <i>frames</i> de ajuda
Confiabilidade	Tempo médio para falha Probabilidade de indisponibilidade Taxa de ocorrência de falhas Disponibilidade
Robustez	Tempo de reinício após falha Percentual de eventos que causam falhas Probabilidade de corrupção de dados em caso de falha
Portabilidade	Percentual de declarações dependentes do sistema-alvo Número de sistemas-alvo



# O documento de requisitos de software

Figura 4.4

Usuários de um documento de engenharia de requisitos.



# Estrutura de um documento de requisitos

**Tabela 4.2** A estrutura de um documento de requisitos.

Capítulo	Descrição
Prefácio	Deve definir os possíveis leitores do documento e descrever seu histórico de versões, incluindo uma justificativa para a criação de uma nova versão e um resumo das mudanças feitas em cada versão.
Introdução	Deve descrever a necessidade para o sistema. Deve descrever brevemente as funções do sistema e explicar como ele vai funcionar com outros sistemas. Também deve descrever como o sistema atende aos objetivos globais de negócio ou estratégicos da organização que encomendou o software.
Glossário	Deve definir os termos técnicos usados no documento. Você não deve fazer suposições sobre a experiência ou o conhecimento do leitor.
Definição de requisitos de usuário	Deve descrever os serviços fornecidos ao usuário. Os requisitos não funcionais de sistema também devem ser descritos nessa seção. Essa descrição pode usar a linguagem natural, diagramas ou outras notações compreensíveis para os clientes. Normas de produto e processos que devem ser seguidos devem ser especificados.
Arquitetura do sistema	Deve apresentar uma visão geral em alto nível da arquitetura do sistema previsto, mostrando a distribuição de funções entre os módulos do sistema. Componentes de arquitetura que são reusados devem ser destacados.
Especificação de requisitos do sistema	Deve descrever em detalhes os requisitos funcionais e não funcionais. Se necessário, também podem ser adicionados mais detalhes aos requisitos não funcionais. Interfaces com outros sistemas podem ser definidas.
Modelos do sistema	Pode incluir modelos gráficos do sistema que mostram os relacionamentos entre os componentes do sistema, o sistema e seu ambiente. Exemplos de possíveis modelos são modelos de objetos, modelos de fluxo de dados ou modelos semânticos de dados.
Evolução do sistema	Deve descrever os pressupostos fundamentais em que o sistema se baseia, bem como quaisquer mudanças previstas, em decorrência da evolução de hardware, de mudanças nas necessidades do usuário etc. Essa seção é útil para projetistas de sistema, pois pode ajudá-los a evitar decisões capazes de restringir possíveis mudanças futuras no sistema.
Apêndices	Deve fornecer informações detalhadas e específicas relacionadas à aplicação em desenvolvimento, além de descrições de hardware e banco de dados, por exemplo. Os requisitos de hardware definem as configurações mínimas ideais para o sistema. Requisitos de banco de dados definem a organização lógica dos dados usados pelo sistema e os relacionamentos entre esses dados.
Índice	Vários índices podem ser incluídos no documento. Pode haver, além de um índice alfabético normal, um índice de diagramas, de funções, entre outros pertinentes.

# Leitura específica

- [1] Vídeo "**Requisito Funcional e Não Funcional de Software: entenda a diferença.**"  
<https://www.youtube.com/watch?v=YLd6AWKVyas>
- [2] SOMMERVILLE, Ian. **Engenharia de Software**. 10ª Ed. São Paulo: Pearson Prentice Hall, 2011. Páginas 85 até 110.  
Disponível em: <https://plataforma.bvirtual.com.br/Leitor/Loader/168127/pdf>
- [3] Vídeo "**Esqueça isso e seu projeto estará condenado. (Análise de Requisitos)**".  
Disponível em: <https://www.youtube.com/watch?v=rVbJ7ykuLig>



# Aprenda +

- Trabalhando com Engenharia de Requisitos disponível em: <https://www.devmedia.com.br/trabalhando-com-engenharia-de-requisitos/30207>
- O que são Requisitos Funcionais e Requisitos Não Funcionais?. Disponível em: <https://codificar.com.br/requisitos-funcionais-nao-funcionais/>

# Próxima Aula

- **Projeto do Sistema**

- Leitura específica

- [1] Vídeo "Tutorial de Diagramas de Classes UML". Disponível em: <https://www.youtube.com/watch?v=rDidOn6KN9k>
- [2] Vídeo "Tutorial de Caso de Uso UML". Disponível em: <https://www.youtube.com/watch?v=ab6eDdwS3rA>
- [3] SOMMERVILLE, Ian. Engenharia de Software. 10ª Ed. São Paulo: Pearson Prentice Hall, 2011. Páginas 147 até 154.  
Disponível em: <https://plataforma.bvirtual.com.br/Leitor/Loader/168127/pdf>

- Aprenda +

- PINTO, Hudson. **Atividades básicas ao processo de desenvolvimento de Software.**  
Disponível em: <https://www.devmedia.com.br/atividades-basicas-ao-processo-dedesenvolvimento-de-software/5413>
- **UML: Diagrama de Casos de Uso.**  
Disponível em: <https://medium.com/operacionalti/uml-diagrama-de-casos-de-uso-29f4358ce4d5>