Instituto Federal de Educação, Ciência e Tecnologia de São Paulo - IFSP
Câmpus Jacareí

Tecnologia em Análise e Desenvolvimento de Sistemas - ADS

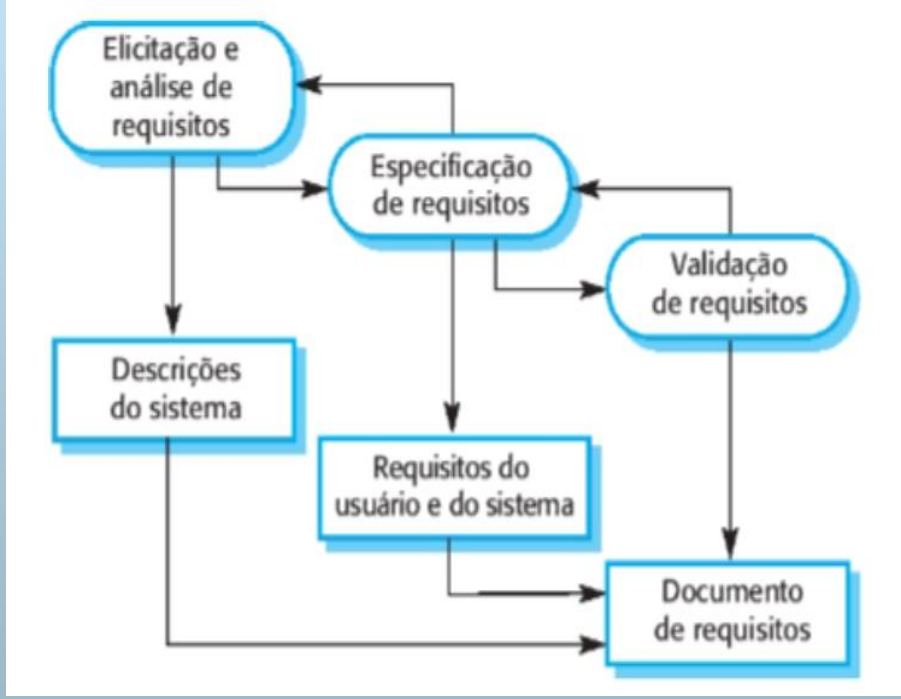
2º Semestre de 2023

Engenharia de Software – JCRESW1

Prof. Lineu Mialaret

Aula 9: Engenharia de Requisitos (2)

Engenharia de Requisitos

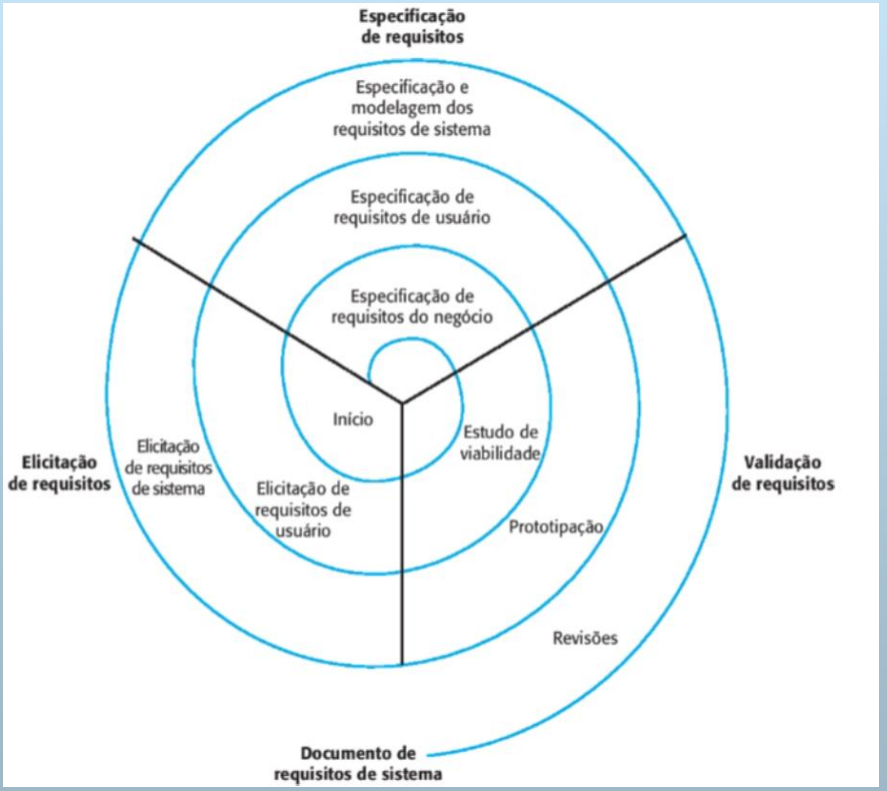


- A Engenharia de Requisitos envolve três atividades fundamentais:
  - Descoberta dos requisitos por meio da interação com *stakeholders* (elicitação e análise).
  - Conversão desses requisitos em uma forma padrão (especificação).
  - Constatação de que os requisitos realmente definem o sistema que o cliente quer (validação).

Engenharia de Requisitos (cont.)

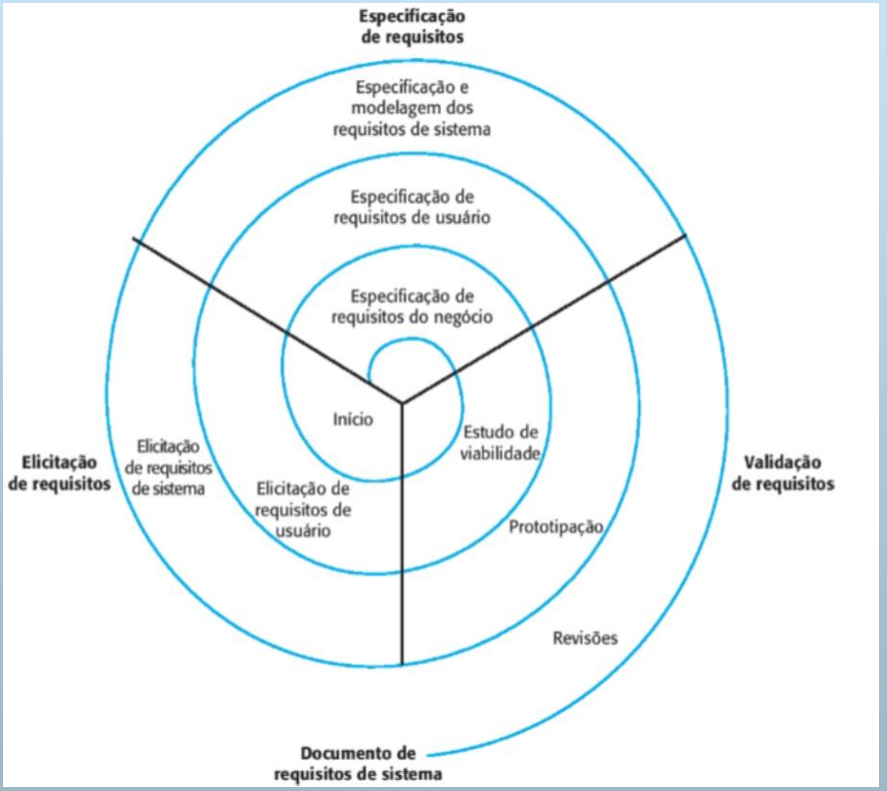
- Na prática, a Engenharia de Requisitos (ER) é um processo iterativo, no qual as atividades são intercaladas, como mostrado a seguir.

Engenharia de Requisitos (cont.)



- As atividades são organizadas como um processo iterativo em torno de uma espiral, e o resultado do processo de ER é um documento de requisitos de sistema.
- A quantidade de tempo e esforço dedicados a cada atividade em uma iteração depende do estágio do processo geral, do tipo de sistema a ser desenvolvido e do orçamento disponível.

Engenharia de Requisitos (cont.)



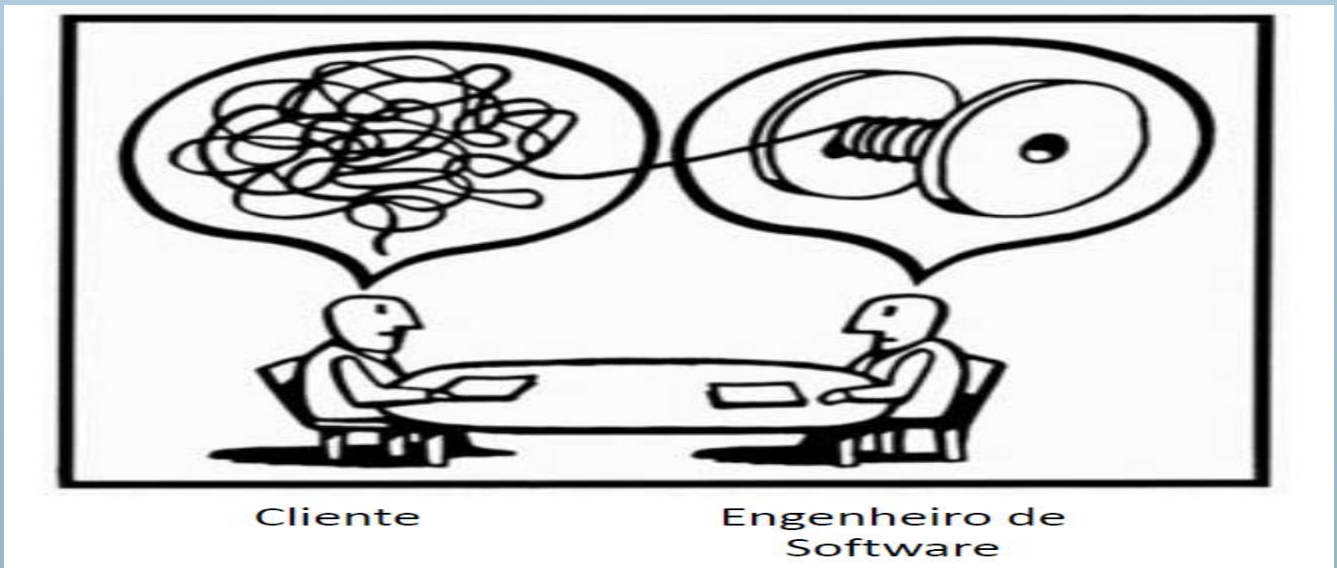
- No início do processo, a maior parte do esforço é dedicada à compreensão do negócio em alto nível e dos requisitos não funcionais, além dos requisitos de usuário e do sistema.
- Numa etapa mais avançada do processo (nos anéis mais externos da espiral), mais esforço será dedicado à elicitação e à compreensão dos requisitos não funcionais e dos requisitos de sistema mais detalhados.

Elicitação de Requisitos

- Os objetivos do processo de elicitação de requisitos são compreender o trabalho que os *stakeholders* realizam e entender como usariam um novo sistema para apoiar o trabalho deles.
  - Durante a elicitação de requisitos, os engenheiros de software trabalham com os *stakeholders* para saber mais sobre o domínio da aplicação, as atividades envolvidas no trabalho, os serviços e as características do sistema que eles querem, o desempenho desejado para o sistema, as limitações de hardware, etc.

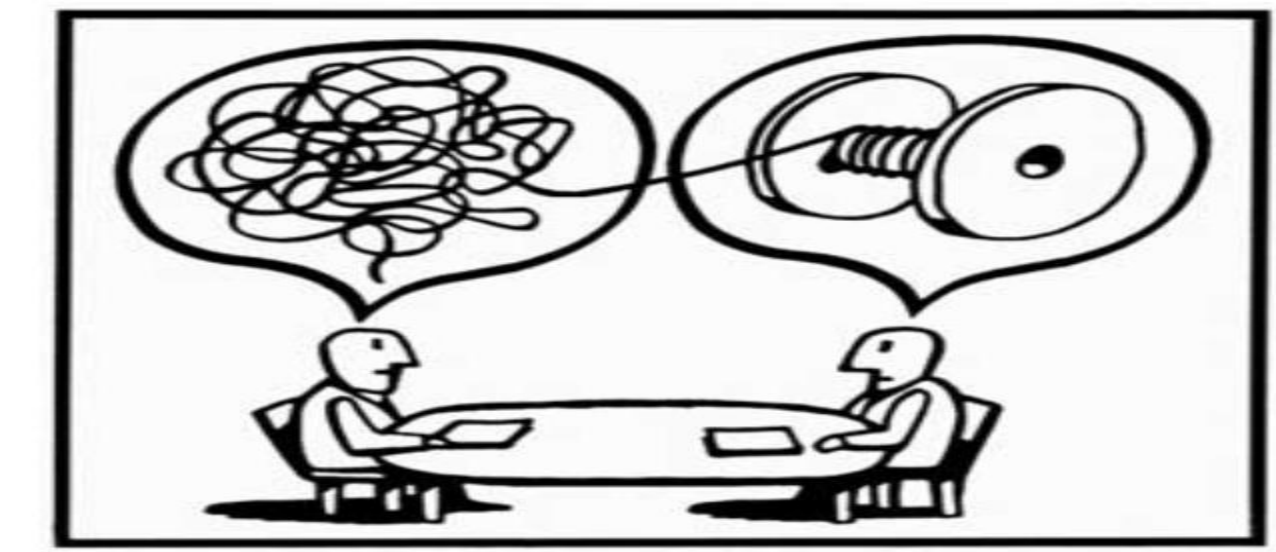
Elicitação de Requisitos (cont.)

- É um processo com uma série de dificuldades:
  - Muitas vezes os *stakeholders* não sabem o que querem de um sistema de computadorizado, exceto em aspectos mais gerais
  - Eles podem achar difícil articular o que querem que o sistema faça.
  - Eles podem fazer exigências irreais porque não sabem o que é viável ou não.



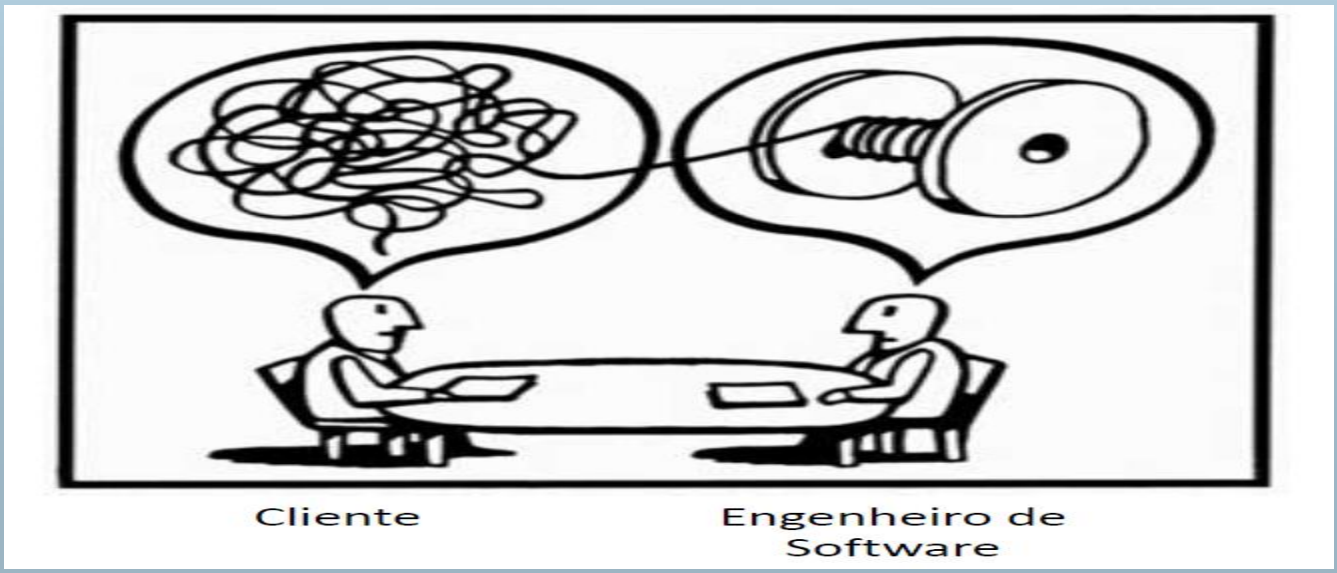
Elicitação de Requisitos (cont.)

- É um processo com uma série de dificuldades:
  - É natural que os *stakeholders* expressem os requisitos em seus próprios termos e com conhecimento implícito de seu próprio trabalho.
  - Os engenheiros de software, sem experiência no domínio do cliente, podem não entender tais requisitos.



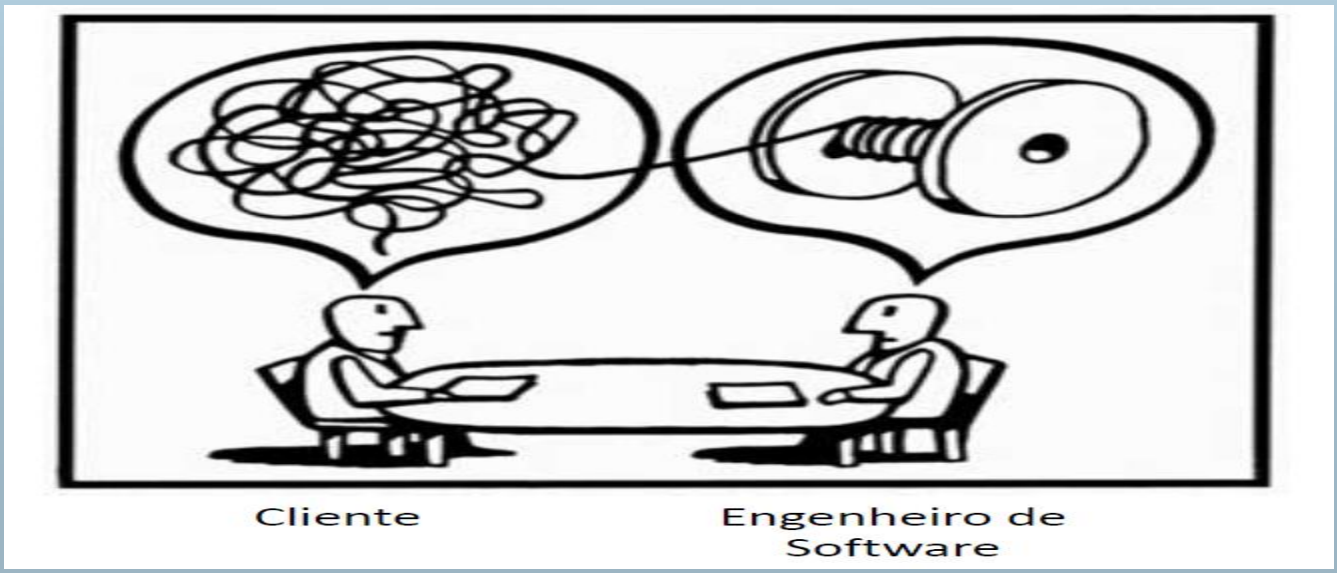
Elicitação de Requisitos (cont.)

- É um processo com uma série de dificuldades:
  - Diferentes *stakeholders,* com requisitos distintos, podem expressá-los de maneiras variadas.
  - Deve-se descobrir todas as possíveis fontes de requisitos, além dos pontos de convergência e de conflito.



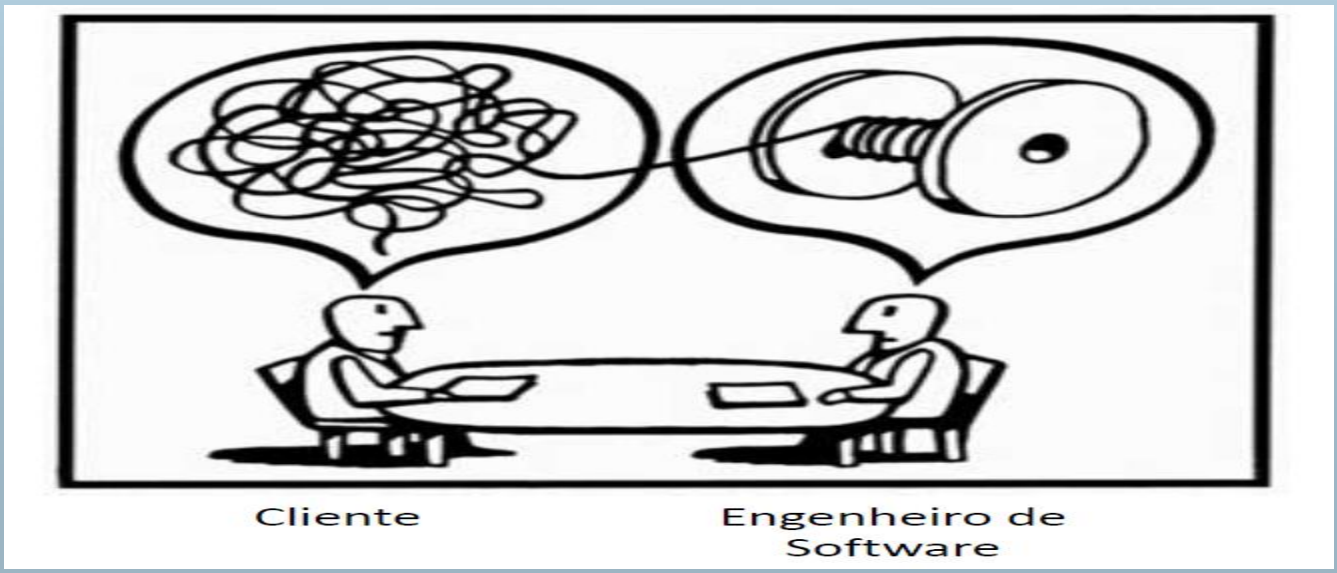
Elicitação de Requisitos (cont.)

- É um processo com uma série de dificuldades:
  - Fatores políticos podem influenciar os requisitos de um sistema.
  - Os gerentes podem exigir requisitos de sistema específicos, o que lhes permite aumentar sua influência na organização.



Elicitação de Requisitos (cont.)

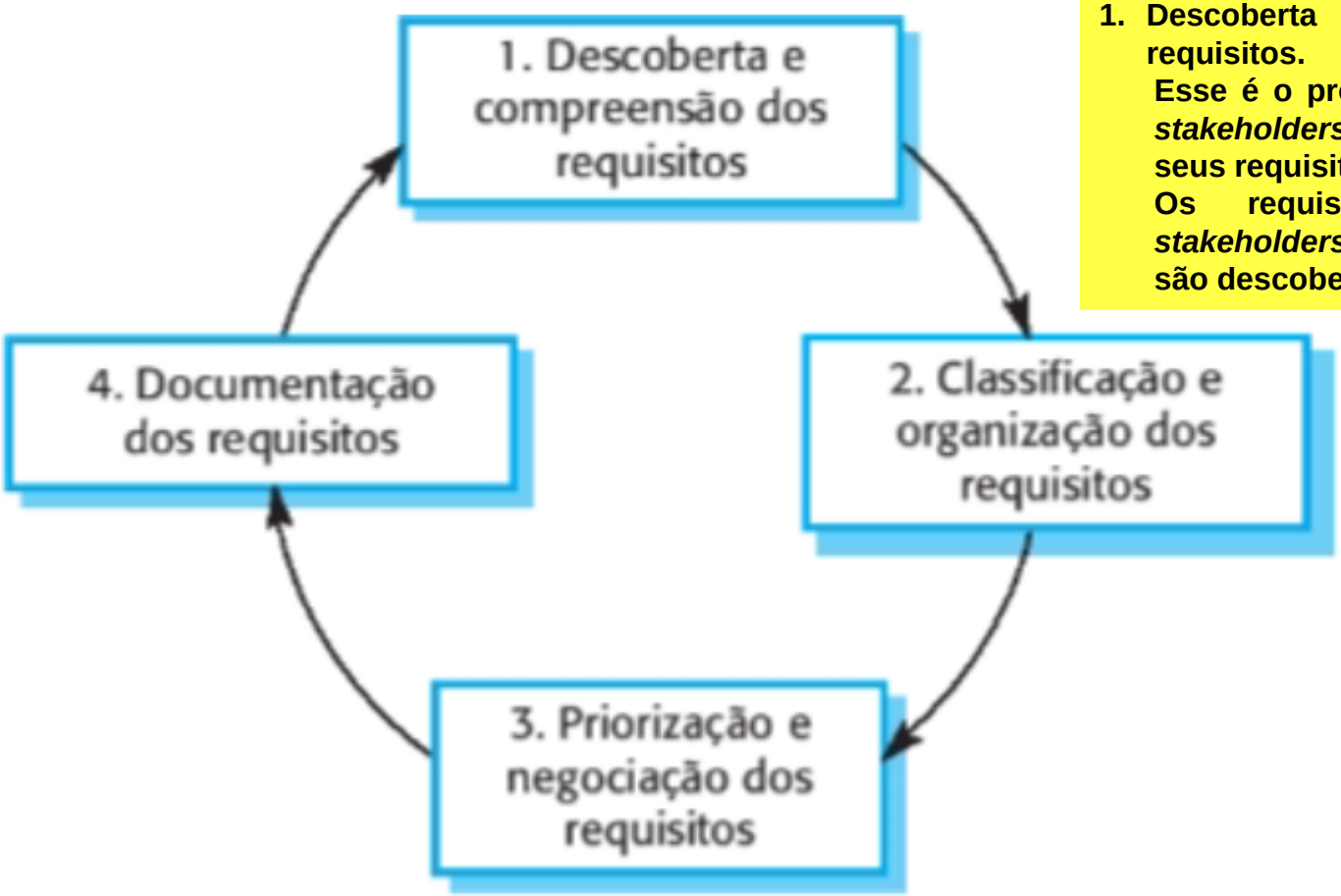
- É um processo com uma série de dificuldades:
  - O ambiente econômico e de negócios no qual a análise ocorre é dinâmico e, inevitavelmente, ele muda durante o processo de análise.
  - A importância de determinados requisitos pode mudar.
  - Novos requisitos podem surgir de novos *stakeholders* que não foram consultados originalmente.



Elicitação de Requisitos (cont.)

- Um modelo do processo de elicitação e análise é exibido a seguir.

Elicitação de Requisitos (cont.)

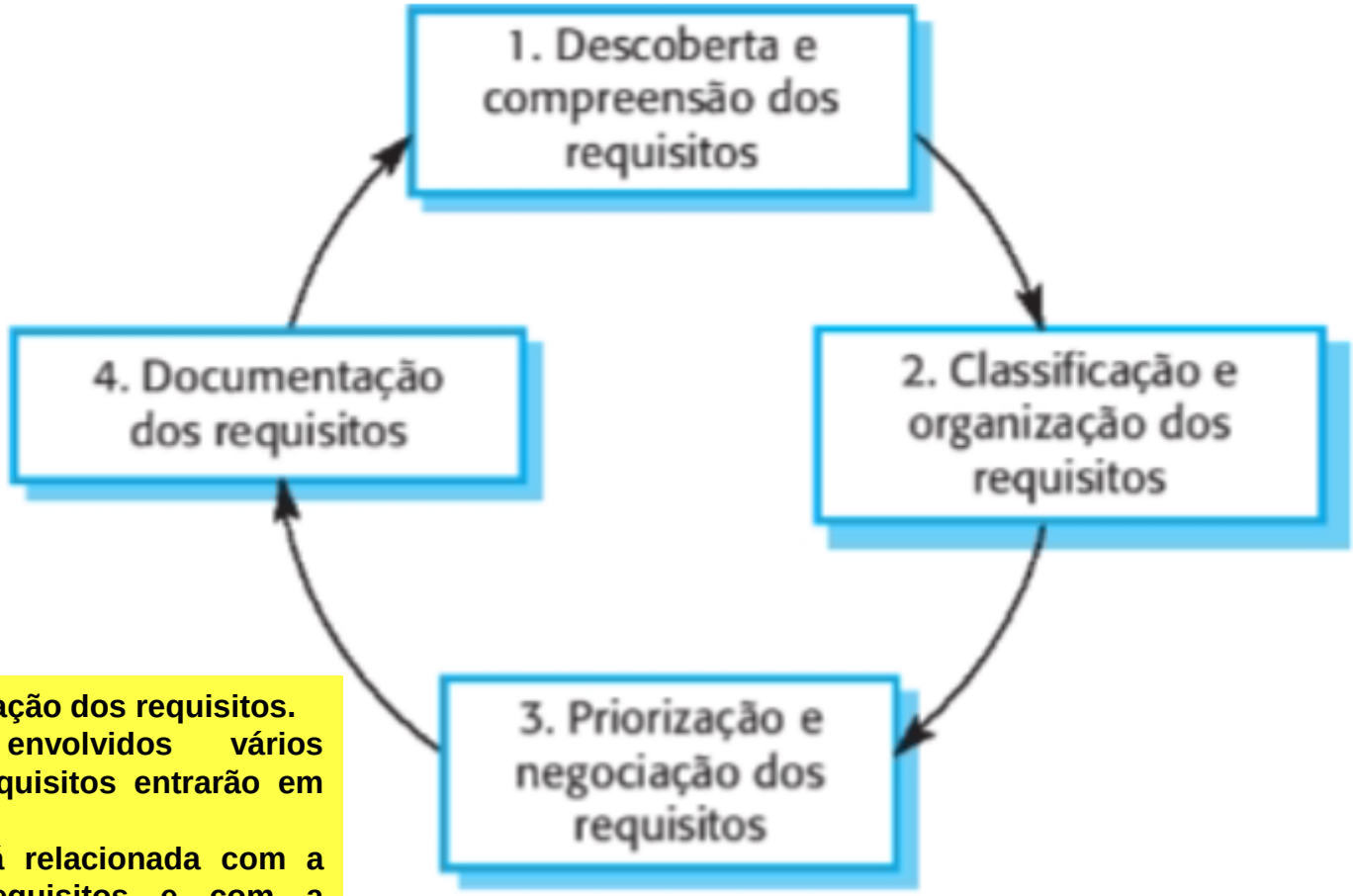


1. **Descoberta e compreensão dos requisitos.**
   Esse é o processo de interagir com os *stakeholders* do sistema para descobrir seus requisitos.
   Os requisitos de domínio dos *stakeholders* e documentação também são descobertos durante essa atividade.

Elicitação de Requisitos (cont.)

2. Classificação e organização dos requisitos.
Essa atividade pega o conjunto não estruturado de requisitos, agrupa os requisitos relacionados e os organiza em grupos coerentes.

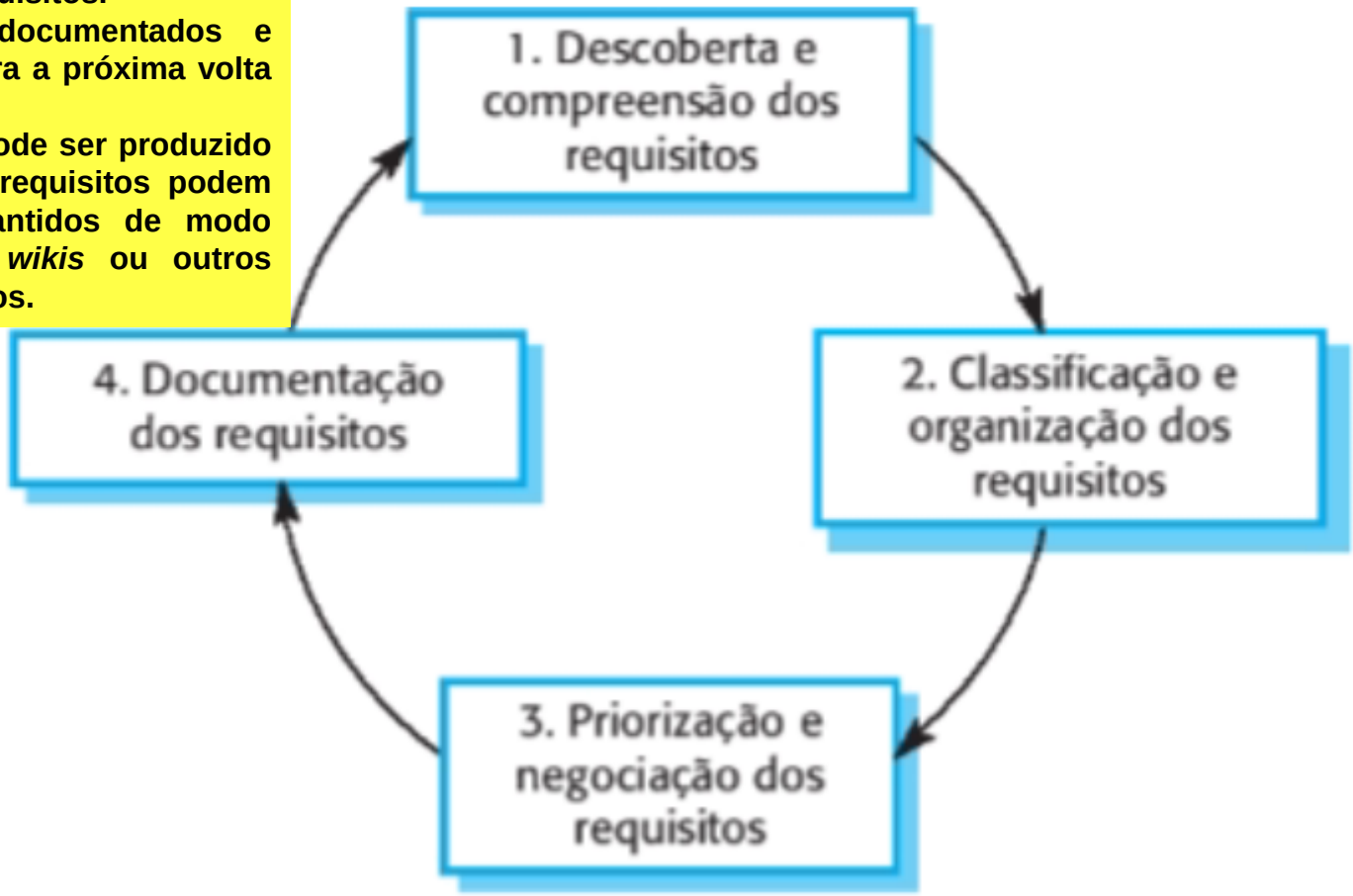
Elicitação de Requisitos (cont.)



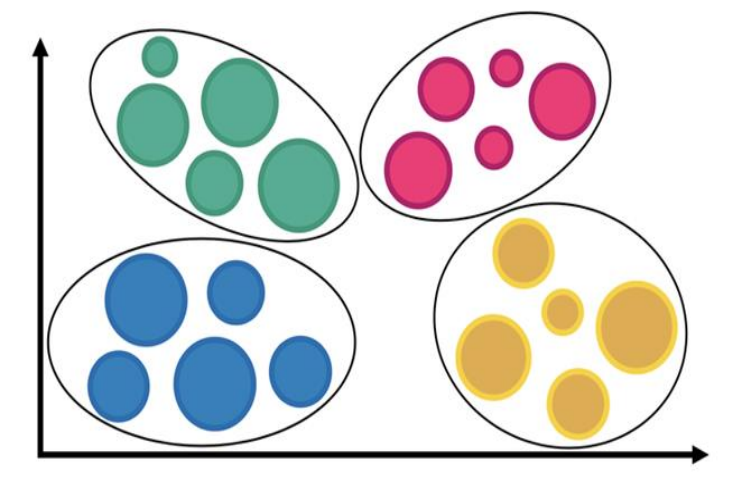
3. **Priorização e negociação dos requisitos.**
   Quando estão envolvidos vários *stakeholders*, os requisitos entrarão em conflito.
   Essa atividade está relacionada com a priorização dos requisitos e com a descoberta e negociação para resolução de conflitos.
   Os *stakeholders* devem se reunir para resolver as diferenças e chegar a um acordo sobre os requisitos.

Elicitação de Requisitos (cont.)

4. **Documentação dos requisitos.**
   Os requisitos são documentados e servem de entrada para a próxima volta da espiral.
   Um rascunho inicial pode ser produzido nesse estágio ou os requisitos podem simplesmente ser mantidos de modo informal em lousas, *wikis* ou outros espaços compartilhados.



Elicitação de Requisitos (cont.)



- Para simplificar a análise de requisitos, pode ser útil organizar e agrupar as informações dos *stackholders*.
  - Pode-se fazer isto considerando cada grupo de *stackholders* como um ponto de vista e coletar os requisitos desse grupo a partir de um determinado ponto de vista.
  - É possível incluir pontos de vista para representar requisitos de domínio e restrições de outros sistemas.
  - Pode-se também utilizar um modelo de arquitetura do sistema para identificar subsistemas e associar requisitos a eles.

Elicitação de Requisitos (cont.)

Pontos de vista

Ponto de vista é uma maneira de coletar e organizar um conjunto de requisitos de um grupo de *stakeholders* que têm algo em comum. Portanto, cada ponto de vista inclui um conjunto de requisitos de sistema. Os pontos de vista poderiam vir de usuários finais, gerentes ou outros, e ajudam a identificar as pessoas que podem fornecer informações sobre seus requisitos e estruturá-los para análise.

Técnicas de Elicitação de Requisitos

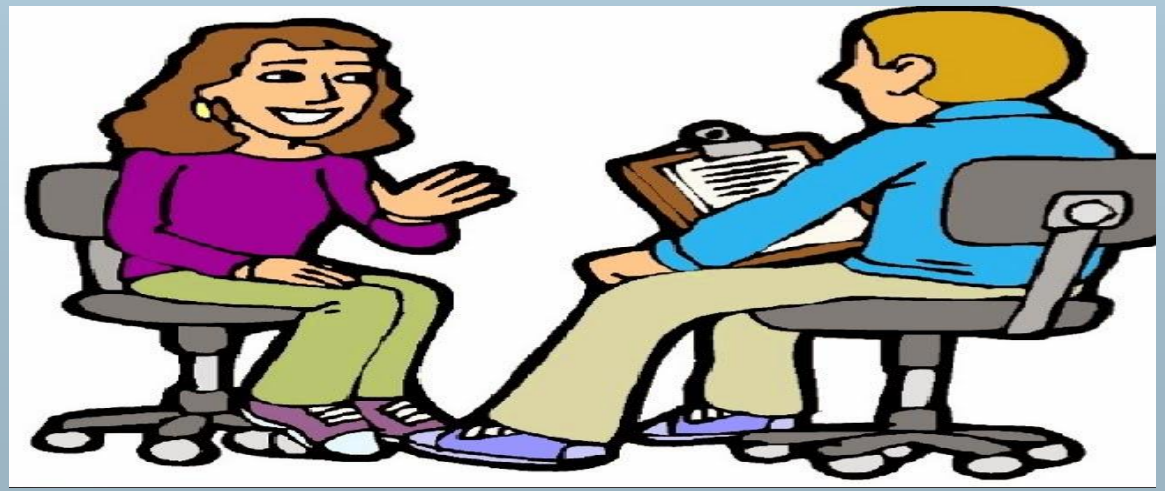
- A elicitação de requisitos envolve encontros com *stackholders* de diferentes tipos para descobrir informações sobre o sistema proposto.
  - Fontes de informação durante a fase de descoberta de requisitos incluem documentação, *stakeholders* do sistema e especificações de sistemas similares.
  - Há interação com os *stakeholders* por meio da observação e de entrevistas
  - Pode usar cenários e protótipos para ajudar os *stakeholders* a compreenderem o que o sistema vai ser.

Entrevistas

- Entrevistas Formais ou Informais com os *stakeholders* do sistema são parte da maioria dos processos de engenharia de requisitos.
  - Nessas entrevistas, a equipe de engenharia de requisitos questiona os *stakeholders* sobre o sistema que usam no momento e sobre o sistema que será desenvolvido.
  - Requisitos surgem a partir das respostas a essas perguntas.

Entrevistas (cont.)

- As entrevistas podem ser de dois tipos:
  - Entrevistas Fechadas, nas quais os *stakeholders* respondem a um conjunto pré-definido de perguntas.
  - Entrevistas Abertas, nas quais não há uma programação predefinida.
    - A equipe de engenharia de requisitos explora uma gama de questões com os *stakeholders* do sistema e, a partir daí, desenvolve uma melhor compreensão de suas necessidades.



Entrevistas (cont.)

Open	Closed
How do you get to work?	Do you get to work by driving, busing, or walking?
Tell me about your relationship with your boss.	Do you get on well with your boss?
What did you manage to accomplish on the trip?	Was your trip successful?
What happened at the meeting?	Did you have a good meeting?

Tipos de Entrevistas.

Entrevistas (cont.)

- Na prática, as entrevistas com os *stakeholders* normalmente são uma mistura dos dois tipos.
  - É possível obter a resposta a determinadas perguntas, mas, normalmente, elas levam a outras questões discutidas de uma maneira menos estruturada.
  - As discussões totalmente abertas raramente funcionam bem.
  - Em geral, deve-se fazer algumas perguntas para começar e manter a entrevista focada no sistema a ser desenvolvido.

Entrevistas (cont.)

- As entrevistas são boas para obter uma compreensão global do que os *stakeholders* fazem, de como irão interagir com o novo sistema e das dificuldades que enfrentam nos sistemas atuais.
  - As pessoas gostam de falar sobre o trabalho delas e, por isso, normalmente ficam felizes em participar de entrevistas.
  - No entanto, a não ser que haja um protótipo do sistema para demonstrar, não se deve esperar que os *stakeholders* sugiram requisitos específicos e detalhados.
  - Todo mundo acha difícil imaginar como um sistema poderia se parecer, portanto é preciso analisar as informações coletadas e gerar os requisitos a partir disso.

Entrevistas (cont.)

- Pode ser difícil obter conhecimento de uma certa área por meio de entrevistas, devido ao fato de:
  - Todos os especialistas em aplicações usam jargões específicos de sua área de trabalho.
    - É impossível discutir os requisitos sem usar esse tipo de terminologia.
    - Normalmente, usam palavras de maneira precisa e sutil, que os engenheiros de requisitos podem entender erroneamente.

Entrevistas (cont.)

- Pode ser difícil obter conhecimento de uma certa área por meio de entrevistas, devido ao fato de:
  - Certos conhecimentos da área podem ser tão familiares aos *stakeholders* que eles acham difícil explicá-los, ou podem ser tão básicos que eles pensam que não vale a pena mencioná-los.
  - Exemplo:
    - Para um bibliotecário, não é preciso dizer que todas as aquisições são catalogadas antes de serem acrescentadas à biblioteca.
    - No entanto, isso pode não ser óbvio para o entrevistador e, por esse motivo, sequer ser levado em conta entre os requisitos.

Entrevistas (cont.)

- As entrevistas podem não ser uma técnica eficaz para elicitar conhecimento a respeito dos requisitos e das restrições organizacionais porque existem relações de poder sutis entre as pessoas em uma empresa.
  - As estruturas organizacionais divulgadas raramente correspondem à realidade da tomada de decisão em uma empresa, mas os entrevistados podem não querer revelar para estranhos a estrutura real em vez da teórica.
  - Em geral, a maioria das pessoas reluta em discutir questões políticas e organizacionais que possam afetar os requisitos

Entrevistas (cont.)

- Para se fazer entrevistas com eficácia deve-se:
  - Ter a mente aberta, evitar ideias preconcebidas a respeito dos requisitos e ter disposição para ouvir os *stakeholders.*
    - Se ele tiver propostas de requisitos surpreendentes, então é necessário estar disposto a mudar de ideia a respeito do sistema.
  - Incentivar o entrevistado a manter a conversa fazendo uma pergunta que sirva como trampolim ou uma proposta de requisitos; ou então trabalhando juntos em um sistema protótipo.
    - Provavelmente, falar para as pessoas “diga-me o que você quer" não vai resultar em informações úteis, pois é muito mais fácil falar em um contexto definido do que em termos gerais.

Entrevista (cont.)

- As informações das entrevistas são utilizadas junto com outras que dizem respeito ao sistema, como:
  - Documentação que descreve os processos do negócio ou dos sistemas existentes.
  - Observações do usuário.
  - Experiência do desenvolvedor.
- Às vezes, além das informações nos documentos do sistema, as informações da entrevista podem ser a única fonte de informação sobre os requisitos de sistema.
  - Entretanto, a entrevista em si está sujeita à perda de informações essenciais e. portanto, deve ser utilizada em conjunto com outras técnicas de elicitação de requisitos.

Etnografia

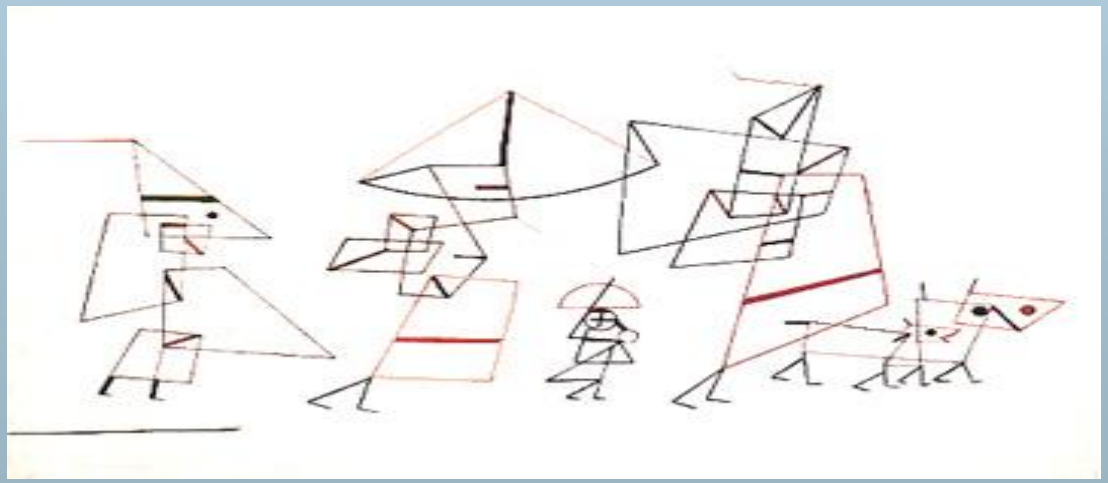
- A Etnografia é uma técnica de observação que pode ser utilizada para entender os processos operacionais e para ajudar a derivar os requisitos do software que apoia esses processos.
  - Um analista deve ficar imerso no ambiente de trabalho em que o sistema será utilizado com o objetivo de observar o dia a dia e tomar nota das tarefas reais nas quais os participantes estão envolvidos.

Etnografia (cont.)

- A vantagem da etnografia é que ela ajuda a descobrir requisitos implícitos do sistema, os quais refletem o verdadeiro modo de trabalho das pessoas, em vez dos processos formais definidos pela organização.
  - É útil para compreender sistemas existentes.

Histórias e Cenários

- As Histórias e Cenários são formas de se capturar as maneiras como as pessoas lidam com determinadas situações ou imaginar coisas que poderiam fazer com uma nova forma de trabalhar, que pode ser usada posteriormente ao entrevistar grupos de *stakeholders* para discutir o sistema com outros grupos e desenvolver requisitos de sistema mais específicos.



Histórias e Cenários (cont.)

- Histórias e cenários são essencialmente a mesma coisa.
- Trata-se de uma descrição de como o sistema pode ser utilizado em alguma tarefa em particular.
  - Histórias e cenários descrevem o que as pessoas fazem, quais informações usam e produzem e quais sistemas podem adotar nesse processo.
  - A diferença está no modo como as descrições são estruturadas e no nível de detalhe apresentado.
- Histórias são escritas como texto narrativo e apresentam uma descrição de alto nível do uso do sistema.
- Cenários normalmente são estruturados com informações especificas coletadas, como entradas e saídas.

Histórias e Cenários (cont.)

Compartilhamento de imagens na sala de aula

Jack é um professor de escola primária em Ullapool (uma vila no norte da Escócia). Ele decidiu que um projeto de sala de aula deveria se concentrar na indústria pesqueira da região, examinando a história, o desenvolvimento e o impacto econômico da pesca. Como parte do projeto, ele pede que os alunos reúnam e compartilhem lembranças dos parentes, usem arquivos de jornal e coletem fotografias antigas relacionadas à pesca e às comunidades pesqueiras da região. Os alunos usam um *wiki* do iLearn para reunir histórias sobre pesca e o SCRAN (um site de recursos de história) para acessar os arquivos do jornal e as fotografias. No entanto, Jack também precisa de um site de compartilhamento de imagens, pois quer que os alunos troquem e comentem as fotos uns dos outros e coloquem no site as imagens escaneadas de fotografias antigas que possam ter em suas famílias.

Jack envia um e-mail para um grupo de professores de escola primária, do qual é membro, para ver se alguém pode recomendar um sistema adequado. Dois professores respondem e ambos sugerem que ele use o KidsTakePics, um site de compartilhamento de imagens que permite aos professores conferirem e moderarem o conteúdo. Como o KidsTakePics não é integrado ao serviço de autenticação do iLearn, ele cria uma conta de professor e uma conta de turma. Ele utiliza o serviço de configuração do iLearn para adicionar o KidsTakePics aos serviços visualizados pelos alunos em sua turma para que, quando fizerem o *login*, possam usar imediatamente o sistema para enviar fotos de seus celulares, tablets e computadores da sala de aula.

Exemplo de uma história.

Histórias e Cenários (cont.)

- Um cenário começa com uma descrição da interação e durante o processo de elicitação, são acrescentados detalhes para criar uma descrição completa dessa interação.
- De modo geral, um cenário pode incluir:
  - Uma descrição do que o sistema e os usuários esperam quando o cenário se inicia.
  - Uma descrição do fluxo normal dos eventos no cenário.
  - Uma descrição do que pode dar errado e de como esses problemas podem ser enfrentados.
  - Informações sobre outras atividades que poderiam ocorrer ao mesmo tempo.
  - Uma descrição do estado do sistema quando o cenário termina.

Histórias e Cenários (cont.)

Enviar fotos para o KidsTakePics

Pressuposto inicial: Um usuário ou grupo de usuários tem uma ou mais fotografias digitais para serem enviadas para o site de compartilhamento de imagens. Essas fotos estão salvas em um tablet ou notebook. Eles fizeram o *login* no site KidsTakePics.

Normal: O usuário opta por enviar as fotos e é solicitado a ele que selecione as fotos no computador a serem enviadas e escolha o nome do projeto sob o qual as fotos serão armazenadas. Os usuários também devem ter a opção de digitar palavras-chave que deverão ser associadas a cada foto enviada. Essas fotos recebem um nome criado pela conjunção do nome do usuário com o nome do arquivo da foto no computador local.

No final do envio, o sistema manda automaticamente um e-mail para o moderador do projeto, pedindo-lhe que verifique o novo conteúdo, e gera uma mensagem na tela para o usuário dizendo que essa verificação foi feita.

O que pode dar errado: Nenhum moderador está associado ao projeto selecionado. Um e-mail é gerado automaticamente para o administrador da escola pedindo para nomear um moderador do projeto. Os usuários devem ser informados de um possível atraso no procedimento para tornar suas fotos visíveis.

Fotos com o mesmo nome já foram enviadas pelo mesmo usuário. O usuário deve ser questionado se deseja enviar novamente as fotos, renomeá-las ou cancelar seu envio. Se os usuários escolherem reenviar, os originais serão sobrescritos. Se optarem por renomear, um novo nome será gerado automaticamente acrescentando um número ao nome de arquivo existente.

Outras atividades: O moderador pode estar logado no sistema e aprovar as fotos à medida que forem enviadas.

Estado do sistema ao terminar: O usuário está logado. As fotos escolhidas foram enviadas e receberam o status de 'aguardando moderação'. As fotos estarão visíveis para o moderador e para o usuário que as enviou.

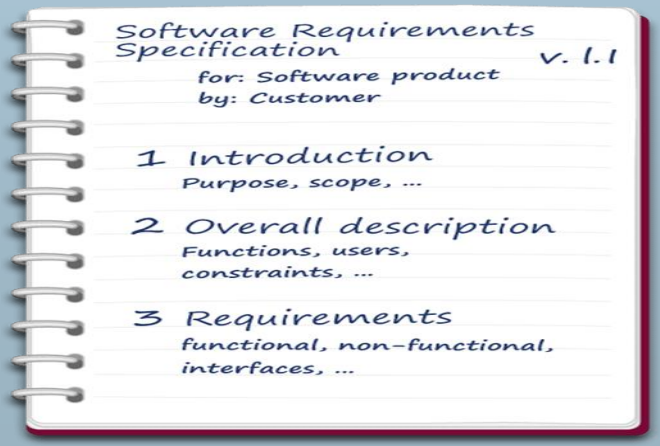
Exemplo de um cenário.

Outras Técnicas

- *Workshops*.
- Reuniões de *Brainstorming.*
- Prototipação.
- Maquetes.
- Análise de documentação existente.
- Análise de sistemas existentes.
- Observação de pessoas trabalhando.
- Pesquisa de mercado.
- Etc.

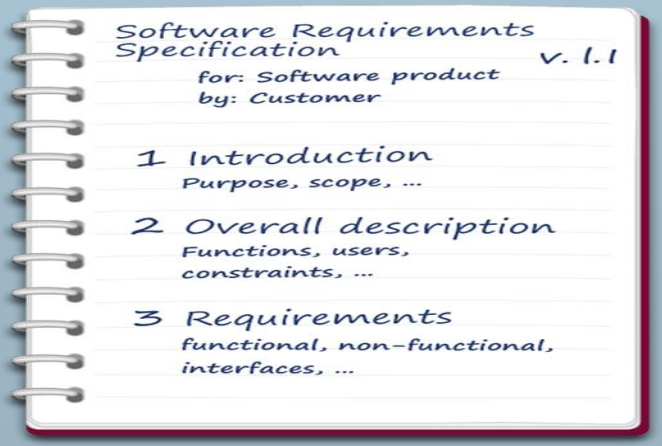
Especificação de Requisitos

- A Especificação de Requisitos é o processo de escrever detalhadamente os requisitos de usuário e de sistema em um documento de requisitos.
  - Em condições ideais, esses requisitos devem ser claros, inequívocos, fáceis de entender, completos e consistentes.
  - Na prática, isso é quase impossível de alcançar.
  - Os *stakeholders* interpretam os requisitos de maneiras diferentes e muitas vezes há conflitos e incoerências inerentes a eles.



Especificação de Requisitos (cont.)

- Os requisitos de usuário quase sempre são escritos em linguagem natural, complementada por diagramas e tabelas apropriados no documento de requisitos.
  - Os requisitos de sistema também podem ser escritos em linguagem natural, mas outras notações baseadas em formulários, gráficos ou modelos matemáticos do sistema também podem ser utilizadas.



Especificação de Requisitos (cont.)

Notação	Descrição
Sentenças em linguagem natural	Os requisitos são escritos usando frases numeradas em linguagem natural. Cada frase deve expressar um requisito.
Linguagem natural estruturada	Os requisitos são escritos em linguagem natural em um formulário ou *template*. Cada campo fornece informações sobre um aspecto do requisito.
Notações gráficas	Modelos gráficos, suplementados por anotações em texto, são utilizados para definir os requisitos funcionais do sistema. São utilizados com frequência os diagramas de casos de uso e de sequência da UML.
Especificações matemáticas	Essas notações se baseiam em conceitos matemáticos como as máquinas de estados finitos ou conjuntos. Embora essas especificações inequívocas possam reduzir a ambiguidade em um documento de requisitos, a maioria dos clientes não compreende uma especificação formal. Eles não conseguem averiguar se ela representa o que desejam e relutam em aceitar essa especificação como um contrato do sistema (discutirei essa abordagem no Capítulo 10, que aborda a dependabilidade do sistema).

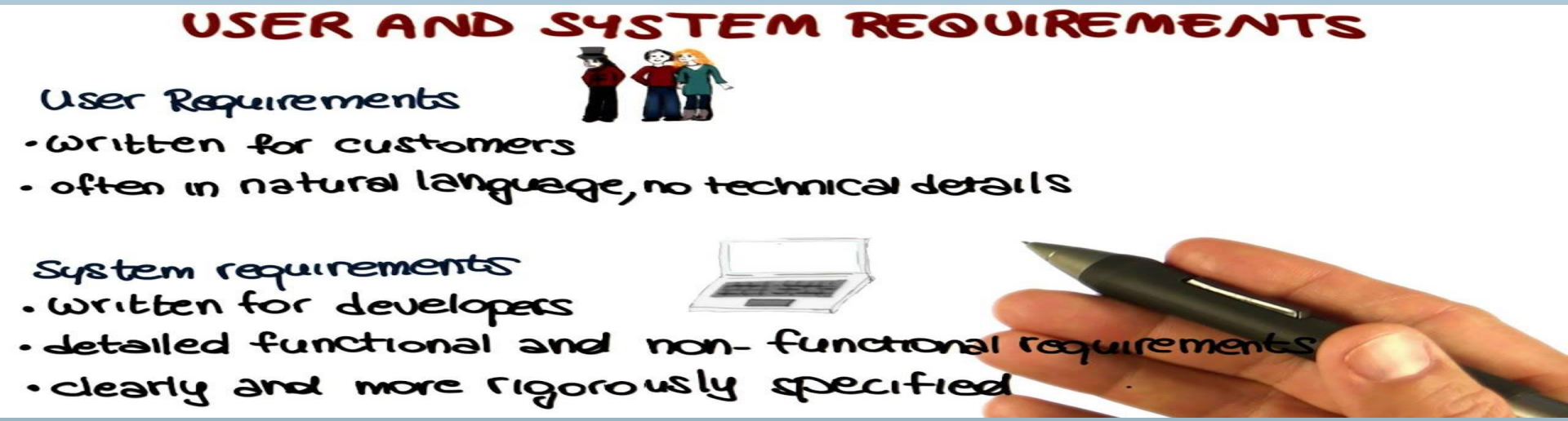
Formatos utilizados para especificação de requisitos.

Especificação de Requisitos (cont.)

- Os requisitos de usuário (*User Requirements*) de um sistema devem descrever os requisitos funcionais e não funcionais de modo que sejam compreensíveis para os usuários do sistema que não têm conhecimento técnico detalhado.
  - Em condições ideais, eles devem especificar apenas o comportamento externo do sistema.
  - O documento de requisitos não deve incluir detalhes da arquitetura ou do projeto (*design*) do sistema.
  - Consequentemente, ao escrever requisitos de usuário, não se deve usar jargões de software, notações estruturadas ou notações formais.
  - Os requisitos de usuário devem ser escritos em linguagem natural, com tabelas simples, formulários e diagramas intuitivos.

Especificação de Requisitos (cont.)

- Os requisitos de sistema (*System Requirements*) são versões ampliadas dos requisitos de usuário, que desenvolvedores usam como ponto de partida para o projeto do sistema, acrescentando detalhes e explicando como o sistema deverá atender os requisitos de usuário.
  - Eles podem ser utilizados como parte do contrato para a implementação do sistema, portanto devem ser uma especificação completa e detalhada do sistema inteiro.



Especificação de Requisitos (cont.)

User requirement

- UR1: A user of the work-based learning system shall be able to locate the person who is responsible for a document.
- UR2: A user of the work-based social networking system shall be able to control the amount of information about the user that the system presents to the other users like co-team member, group member and external for viewing.

System requirement

- SR1: The work-based learning system shall be able to retrieve the name, desk address, telephone number and email id fields of the record of a person responsible for the selected document.
- SR2: The work-based social networking system shall allow the user to present and hide fields from among the 48 data fields of the record of a person. The system presents these visible fields to the other users, based on three levels entitlements, namely co-team member, group member and external. The fields will be of view type only and not editable.

Especificação de Requisitos (cont.)

- Em condições ideais, os requisitos de sistema devem descrever apenas o comportamento externo do sistema e suas restrições operacionais.
  - Eles não devem se preocupar com o modo que o sistema deve ser projetado ou implementado.
  - No entanto, no nível de detalhe exigido para especificar completamente um sistema de software complexo, não é possível nem desejável excluir todas as informações de projeto (*design*).

Especificação de Requisitos (cont.)

- Exemplos de cenários com implementação:
  - Pode ser necessário fazer o projeto de uma arquitetura inicial do sistema para ajudar a estruturar a especificação dos requisitos.
    - Os requisitos de sistema são organizados de acordo com diferentes subsistemas que o compõem.
  - Na maioria dos casos, os sistemas devem interoperar com os sistemas existentes. o que restringe o projeto e impõe requisitos ao novo sistema.
  - Pode ser necessário o uso de uma arquitetura específica para satisfazer requisitos não funcionais, para por exemplo, para alcançar um dado grau de confiabilidade.
    - Um regulador externo que precise certificar-se de que o sistema é seguro (*safety*) pode especificar que deve ser utilizado um projeto de arquitetura já certificado.

Especificação em Linguagem Natural

- A linguagem natural tem sido utilizada para escrever requisitos de software desde os anos 1950.
  - É uma linguagem expressiva, intuitiva e universal, e dessa forma, requisitos podem ser compreendidos por usuários e clientes.
  - Outras documentações (modelos, código, etc.) podem não ser compreendidos.
  - Também é potencialmente vaga e ambígua, sendo que a sua interpretação depende da experiência do leitor.
  - Consequentemente, tem havido muitas propostas de maneiras alternativas para escrever os requisitos.
  - No entanto, nenhuma dessas propostas foi adotada amplamente, e a linguagem natural continuará sendo a maneira mais utilizada de especificar requisitos de sistema e de software.

Especificação em Linguagem Natural (cont.)

- Diretrizes para redução de ruído na linguagem natural:
  - Utilizar (ou inventar) um formato padrão e garantir que todas as definições de requisitos o sigam.
    - Padronizar o formato diminui a probabilidade de omissões e torna os requisitos mais fáceis de serem conferidos.
    - Sempre que for possível, sugere-se escrever o requisito em uma ou duas frases de linguagem natural.

Especificação em Linguagem Natural (cont.)

- Diretrizes para redução de ruído na linguagem natural:
  - Usar a linguagem coerentemente para distinguir entre requisitos obrigatórios e desejáveis.
    - Os requisitos obrigatórios são aqueles que o sistema deve apoiar - e normalmente são escritos usando a palavra “deve” (*shall*).
    - Os requisitos desejáveis (ou opcionais) não são essenciais - e são escritos usando a palavra “pode” (*must*).

Especificação em Linguagem Natural (cont.)

- Diretrizes para redução de ruído na linguagem natural:
  - Usar realce de texto (negrito, itálico ou cor) para destacar partes importantes do requisito.

Especificação em Linguagem Natural (cont.)

- Diretrizes para redução de ruído na linguagem natural:
  - Não supor que os leitores compreendem a linguagem técnica da engenharia de software.
    - É fácil que palavras como “arquitetura” e “módulo” sejam mal compreendidas.
    - Sempre que possível, evitar o uso de jargões, abreviações e acrônimos.

Especificação em Linguagem Natural (cont.)

- Diretrizes para redução de ruído na linguagem natural:
  - Sempre que possível, tentar associar um ponteiro (racional) de origem a cada requisito de usuário.
    - O ponteiro deve explicar por que o requisito foi incluído e quem o propôs (a origem do requisito), de modo que se saiba a quem recorrer se o requisito precisar ser alterado.
    - O ponteiro dos requisitos é particularmente útil quando isso acontece, já que essa mudança pode ajudar a decidir quais alterações seriam indesejáveis.

Especificação em Linguagem Natural (cont.)

- Exemplo de especificação de um requisito.

3.2 O sistema deve medir o nível de açúcar no sangue e fornecer insulina, se for necessário, a cada 10 minutos. (*As variações do açúcar no sangue são relativamente lentas, então é desnecessário medir com uma frequência maior; a medição menos frequente poderia levar a níveis de açúcar sanguíneo desnecessariamente elevados.*)

3.6 O sistema deve executar uma rotina de autoteste a cada minuto com as condições a serem testadas e as ações associadas, definidas na Tabela 1 do documento de requisitos. (*Uma rotina de autoteste pode descobrir problemas de hardware e software e alertar o usuário de que a operação normal pode ser impossível.*)

Especificação em Linguagem Natural (cont.)

Problemas com o uso da linguagem natural na especificação dos requisitos

A flexibilidade da linguagem natural, tão útil para a especificação, costuma causar problemas. Existe espaço para escrever requisitos obscuros e os leitores (os projetistas) podem interpretar erroneamente os requisitos porque eles e os usuários têm experiências diferentes. É fácil fundir vários requisitos em uma única frase, o que pode dificultar a estruturação dos requisitos em linguagem natural.

Especificação Estruturada

- A linguagem natural estruturada é uma maneira de escrever os requisitos de sistema, de modo que estes sejam escritos em uma forma padrão em vez de em texto livre.
  - Essa abordagem mantém a maior parte da expressividade e da clareza da linguagem natural, mas garante que alguma uniformidade seja imposta à especificação.
  - As notações que adotam linguagem estruturada usam modelos (*templates*) para especificar requisitos de sistema.
  - Essa especificação pode usar construções de linguagem de programação para mostrar alternativas e iteração, podendo destacar elementos-chave por intermédio de sombreamento ou de fontes diferentes.

Especificação Estruturada (cont.)

- Para usar uma abordagem estruturada para especificar requisitos de sistema, é preciso definir um ou mais *templates* para os requisitos e representá-los como formulários estruturados.
  - A especificação pode ser estruturada em volta dos objetos manipulados pelo sistema, das funções realizadas por ele ou dos eventos processados.

Especificação Estruturada (cont.)

- Um exemplo de *template* é apresentado a seguir.
  1) Descrição da função ou entidade que está sendo especificada.
  2) Descrição das entradas e suas origens.
  3) Descrição das saídas e sua destinação.
  4) Informações sobre os dados necessários para computar ou outras entidades no sistema que sejam necessárias (a parte “requer”).
  5) Descrição da ação a ser tomada.
  6) Se for utilizada uma abordagem funcional, uma precondição estabelecendo o que deve ser verdadeiro antes da função ser invocada e uma pós-condição especificando o que é verdadeiro após a função ser invocada.
  7) Descrição dos efeitos colaterais (se houver) da operação.

Especificação Estruturada (cont.)

- Exemplo de especificação baseada em formulário, sendo a que define como calcular a dose de insulina a ser fornecida quando o açúcar no sangue estiver dentro da faixa segura.

Bomba de insulina/Software de controle/SRS/3.3.2	
Função	Computar a dose de insulina: nível de açúcar seguro.
Descrição	Computa a dose de insulina a ser fornecida quando o nível de açúcar atual estiver na zona segura entre 3 e 7 unidades.
Entradas	Leitura atual do açúcar (r2), as duas leituras prévias (r0 e r1).
Fonte	Leitura atual de açúcar do sensor. Outras leituras da memória.
Saídas	DoseComp – a dose de insulina a ser fornecida.
Destino	Laço de controle principal.
Ação	DoseComp é igual a zero se o nível de açúcar estiver estável ou caindo; ou se o nível estiver aumentando, mas a taxa de crescimento estiver diminuindo. Se o nível estiver aumentando e a taxa de crescimento também, então a DoseComp é obtida pela divisão por 4 da diferença entre o nível de açúcar atual e o nível anterior, arredondando o resultado. Se o resultado for arredondado para zero, então a DoseComp é definida como dose mínima que pode ser fornecida (ver Figura 4.14).
Requer	Duas leituras prévias para que a taxa de variação do nível de açúcar possa ser calculada.
Pré-condição	O reservatório de insulina contém pelo menos a dose máxima permitida.
Pós-condição	r0 é substituída por r1, então r1 é substituída por r2.
Efeitos colaterais	Nenhum.

Especificação Estruturada (cont.)

- O uso de especificações estruturadas remove alguns dos problemas da especificação em linguagem natural.
  - A variabilidade na especificação é reduzida, e os requisitos são organizados com mais eficácia.
- No entanto, às vezes é difícil escrever os requisitos de uma maneira clara e inequívoca, particularmente quando computações complexas (como calcular a dose de insulina) devem ser especificadas.

Especificação Estruturada (cont.)

- Para resolver esse problema, é possível acrescentar mais informações aos requisitos em linguagem natural, por exemplo, usando tabelas ou modelos gráficos do sistema.
  - Esses recursos podem mostrar como os cálculos são feitos, como o estado do sistema muda, como os usuários interagem com o sistema e como as sequências de ações são realizadas.

Condição	Ação
Nível de açúcar em queda (r2 < r1)	DoseComp = 0
Nível de açúcar estável (r2 = r1)	DoseComp = 0
Nível de açúcar em alta e taxa de crescimento em queda ((r2 – r1) < (r1 – r0))	DoseComp = 0
Nível de açúcar em alta e taxa de crescimento estável ou em alta r2 > r1 & ((r2 – r1) ≥ (r1 – r0))	DoseComp = arredondar ((r2 – r1) / 4) Se resultado arredondado = 0, então DoseComp = DoseMínima

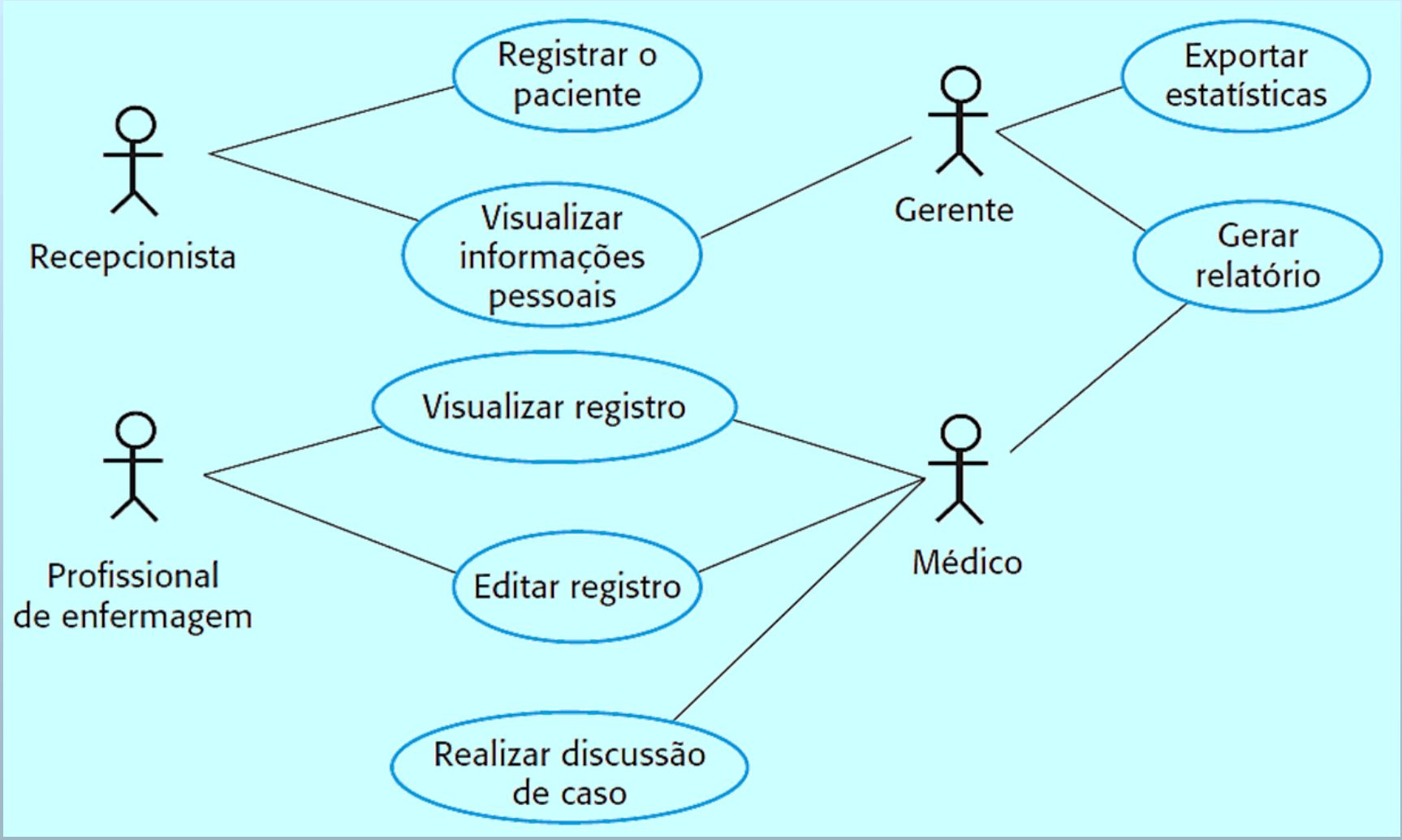
Caso de Uso

- Os Casos de Uso são uma maneira de descrever as interações entre usuários e um sistema usando um modelo gráfico e um texto estruturado.
  - Foram introduzidos pela primeira vez no método Objectory e hoje se tornaram uma característica fundamental da UML.
  - Em sua forma mais simples, um caso de uso identifica os atores envolvidos em uma interação e nomeia o tipo de interação.
  - Depois, são adicionadas informações descrevendo a interação com o sistema, que pode ser uma descrição textual ou um ou mais modelos gráficos

Caso de Uso (cont.)

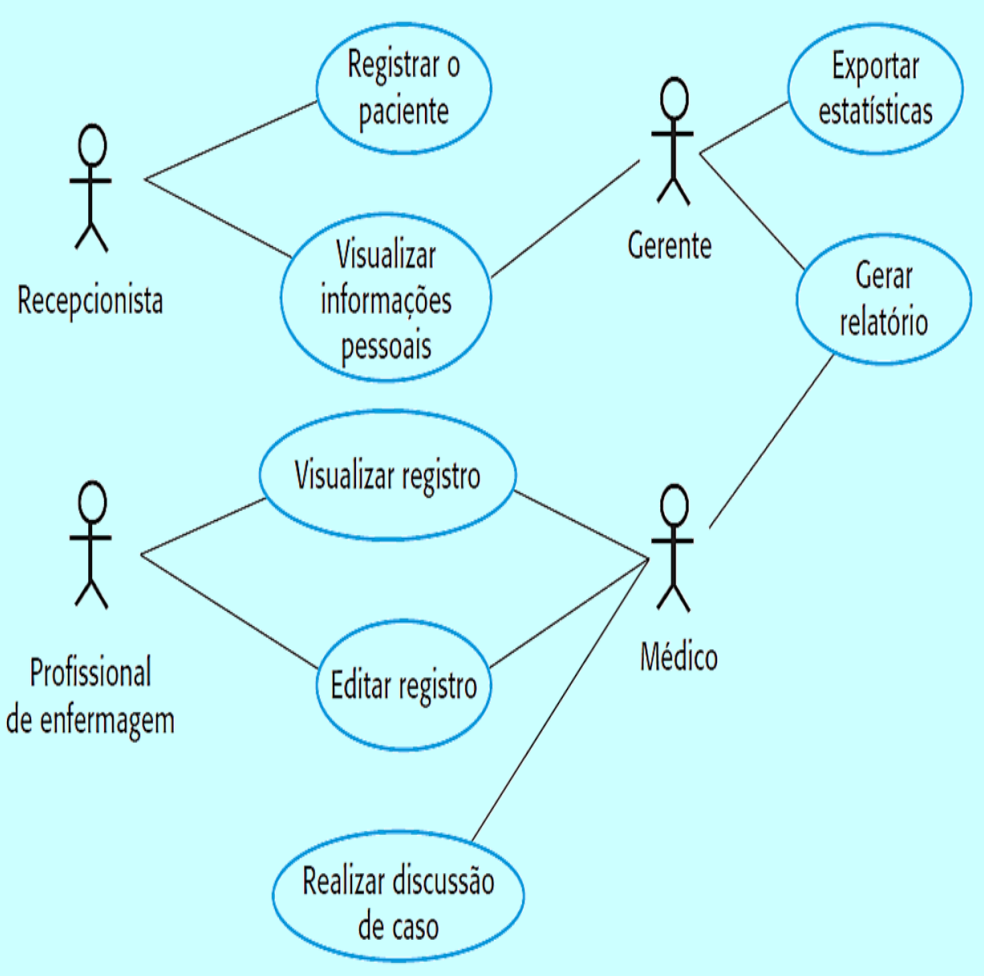
- O conjunto de casos de uso representa todas as interações possíveis que serão descritas nos requisitos de sistema.
  - Os atores no processo, que podem ser seres humanos ou outros sistemas, são representados como “bonecos palito”.
  - Cada classe de interação é representada como uma elipse nomeada.
  - Linhas fazem a ligação entre os atores e a interação.
  - Opcionalmente, pontas de seta podem ser acrescentadas às linhas para mostrar como a interação começa.

Caso de Uso (cont.)



Casos de uso do Sistema Mentcare.

Caso de Uso (cont.)



Casos de uso do Sistema Mentcare.

Descrição do Caso de Uso Realizar Discussão de Caso:

a. Realizar discussão de caso permite que dois ou mais médicos, trabalhando em consultórios diferentes, vejam o registro do mesmo paciente ao mesmo tempo.
b. Um médico inicia a discussão do caso de um paciente escolhendo as pessoas envolvidas em um menu suspenso de médicos que estão *on-line*.
c. O registro do paciente é exibido em suas telas, mas apenas o médico que iniciou a consulta pode editar o registro.
d. Além disso, cria-se um *chat* para ajudar a coordenar as ações.
e. Presume-se que uma chamada telefônica ou comunicação por voz possa ser providenciada separadamente.