1. **O que é um requisito funcional?**

R: É o que o sistema deve fazer. Exemplo: Efetuar login, cadastrar usuário, deletar usuário, etc...

1. **O que é um requisito não funcional?**

R: Como o sistema deve ser. Exemplo: Meu software precisa abrir tal pagina em 2 segundos, vamos utilizar python para o back-end, etc...

1. **Quais são os principais elementos de um requisito bem definido?**

R: Descrição clara e objetiva, ou seja, não deve ter ambiguidade.

1. **Quais são as técnicas de elicitação de requisitos? Faça uma pesquisa detalhada sobre 5 técnicas.**

R: Eu cheguei a ler em um livro que existe uma técnica chamada de “Técnica de observação” que é feita quando o cliente não sabe o que ele quer. Exemplo: Ele precisa de um sistema de ponto eletrônico dinâmico por exemplo. O Engenheiro de software vai até a empresa e observa o comportamento e a movimentação dos funcionários nos horários de bater o ponto, como eles batem esse ponto, de que forma eles batem o ponto, etc... é bem legal.

1. **Requisitos têm papel central no desenvolvimento de software. De que maneira os requisitos influenciam outras atividades do processo de software?**

R: O que eu entendi foi, os requisitos funcionam como um mapa para todo o processo. Se os requisitos forem mal escritos e definidos, o sistema vai acabar saindo de uma forma diferente, ou seja, o design vai sair errado, o código vai sair errado, a manutenção vai ser difícil... nada daquilo que o cliente precisa.

1. **Diferencie requisitos de cliente de requisitos de sistema. Como tratar a documentação de requisitos nesses dois diferentes níveis? Qual o público alvo dessa documentação?**

R: Requisitos de cliente: Aquilo que o cliente quer. Requisitos do sistema: O que o sistema precisa para funcionar

Para tratar o requisitos de cliente podemos utilizar imagens ou com uma linguagem mais clara e fácil de entender

Para tratar os requisitos de sistema nós podemos desenhar fluxos simples tipo inicio > usuário faz o login > adiciona produto ao carrinho > paga... Ou um checklist.

1. **O levantamento de requisitos é uma atividade de descoberta de necessidades dos clientes e usuários e de informações relacionadas a elas. Neste contexto, o que um engenheiro de requisitos precisa conhecer e entender para poder realizar sua tarefa? Quais as potenciais fontes para essas informações? Que problemas os engenheiros de requisitos tipicamente enfrentam?**

R: Precisa entender o negócio do cliente, o que a empresa dele faz e o que é a empresa.

As potenciais fontes para as informações seriam os próprios clientes envolvidos no projeto e também os usuários que vão utilizar o sistema quando estiver pronto.

Os problemas que mais vamos enfrentar muitas das vezes é que o cliente não sabe o que ele quer e também dificuldade de comunicação

1. **Em relação à análise de requisitos, faz-se a seguinte afirmação: “A análise de requisitos é uma atividade de modelagem, na qual deve ser desenvolvido um modelo único, geral do sistema, que capture as necessidades de informação e serviços dos clientes, incorporando aspectos da solução tecnológica a ser adotada”. Avalie e critique essa afirmação, apontando possíveis problemas, corrigindo-os.**

R: Atividade de Modelagem, pelo que eu entendi está ok, modelo único... eu ia dizer que também estaria ok mas, acho meio difícil ser um modelo único já que vamos precisar de mais coisas e ir modelando isso. O texto fala sobre a modelagem do sistema, acho que não seria interessante focar em soluções tecnológicas já que o foco na modelagem é o cliente.

1. **Em relação ao levantamento e análise de requisitos, faz-se a seguinte afirmação: “Os requisitos de sistema devem ser capturados, documentados e acordados previamente, antes de se iniciar qualquer atividade de modelagem. Neste sentido, a atividade de análise de requisitos só deve ser iniciada após a conclusão da atividade de levantamento de requisitos”. Avalie e critique essa afirmação, apontando como Tipicamente devem se dar as atividades de levantamento e análise de requisitos. Em especial, aponte como essas atividades são realizadas quando modelos de ciclo de vida evolutivos (p.ex., modelo incremental, espiral) são adotados no desenvolvimento de software.**

R: Mas se após o engenheiro de software levantar os requisitos, em seguida analisar e ficar faltando algo? Quer dizer que não vai poder alterar? Acho que está afirmação não está me fazendo muito sentido... e se quando eu for analisar todos os requisitos que eu levantei e precisar alterar tudo? Acho que não seria legal, então eu acho que faz mais sentido é, o levantamento de requisitos e a analise caminharem juntos para ir melhorando.

Cheguei até aqui e entendi que para levamento de requisitos precisamos saber o que o cliente quer ou o que ele precisa. Depois precisamos analisar os requisitos que levantamos para saber se eles são claros e objetivos e ir aprimorando conforme o projeto avança.

Professor nessa parte de ciclo evolutivos eu precisei pesquisar sobre, não consegui entender muito bem. Com a pesquisa descobri que em modelos tradicionais como o de cascata (me lembrou do CSS) a ideia era levantar todos os requisitos primeiro, mas em modelos evolutivos, como o incremental, espiral e ágil, as atividades são iterativas.

**Modelo incremental:** O sistema é desenvolvido em partes

- Os requisitos são levantados e analisados para cada incremento separadamente.

**Modelo espiral:** Há ciclos de planejamento, desenvolvimento e refinamento.

- O levantamento e a análise acontecem de forma contínua, ajustando os requisitos conforme riscos e necessidades

**Métodos ágeis:** Priorização e entrega contínua de funcionalidades.

- Os requisitos são levantados e analisados em cada sprint(não sei o que é sprint ainda)

- Feedback constante dos usuários ajuda a refinar os requisitos ao longo do tempo.