**REVISÃO SOBRE OS TÓPICOS**

**Crise do Software**:

* **Problemas Frequentes e Sintomas da Crise**: Entender os problemas recorrentes no desenvolvimento de software, como estouro de orçamento e prazos, é essencial para reconhecer a importância de práticas de engenharia de software. Focar nisso ajuda a perceber por que o gerenciamento adequado de projetos é crucial.
* **Benefícios da Engenharia de Software**: Ressaltar como práticas bem estruturadas são mais econômicas e eficientes a longo prazo mostra aos alunos a necessidade de adotá-las desde o início de suas carreiras.

**Processo de Software**:

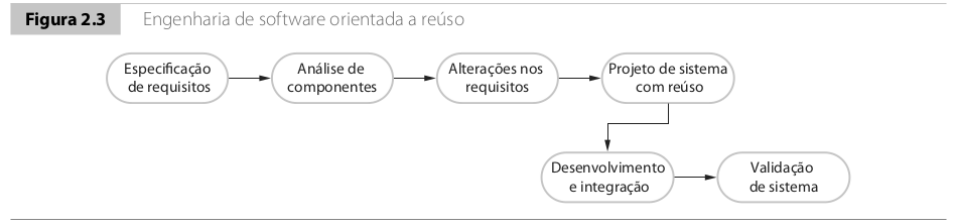
* **Um processo de software é uma sequência de atividades que leva à produção de um produto de software. Existem quatro atividades fundamentais comuns a todos os processos de software.**
* 1) Especificação de software, em que clientes e engenheiros definem o software a ser produzido e as restrições de sua operação.
* 2) Desenvolvimento de software, em que o software é projetado e programado.
* 3) Validação de software, em que o software é verificado para garantir que é o que o cliente quer.
* 4) Evolução de software, em que o software é modificado para refletir a mudança de requisitos do cliente e do mercado.

**Requisitos:** Os requisitos de um sistema são as descrições do que o sistema deve fazer, os serviços que oferece e as restrições a seu funcionamento.

**Requisitos de usuário:** são declarações, em uma linguagem natural com diagramas, de quais serviços o sistema deverá fornecer a seus usuários e as restrições com as quais este deve operar.

**Requisitos de sistema:** são descrições mais detalhadas das funções, serviços e restrições operacionais do sistema de software. O documento de requisitos do sistema (às vezes, chamado especificação funcional) deve definir exatamente o que deve ser implementado. Pode ser parte do contrato entre o comprador do sistema e os desenvolvedores de softwar

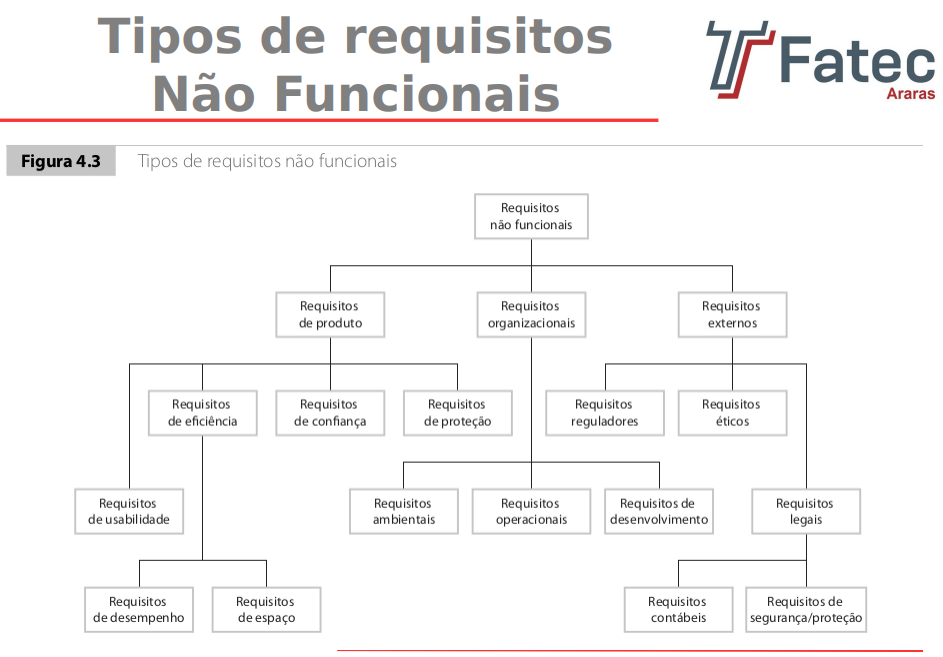
**Engenharia de software orientada a reuso:** a maioria dos projetos de software, ocorre algum nível de reúso de software, geralmente de maneira informal, quando as pessoas envolvidas conhecem códigos ou projetos semelhantes ao necessário. Este reúso informal acontece independentemente do processo de desenvolvimento utilizado.



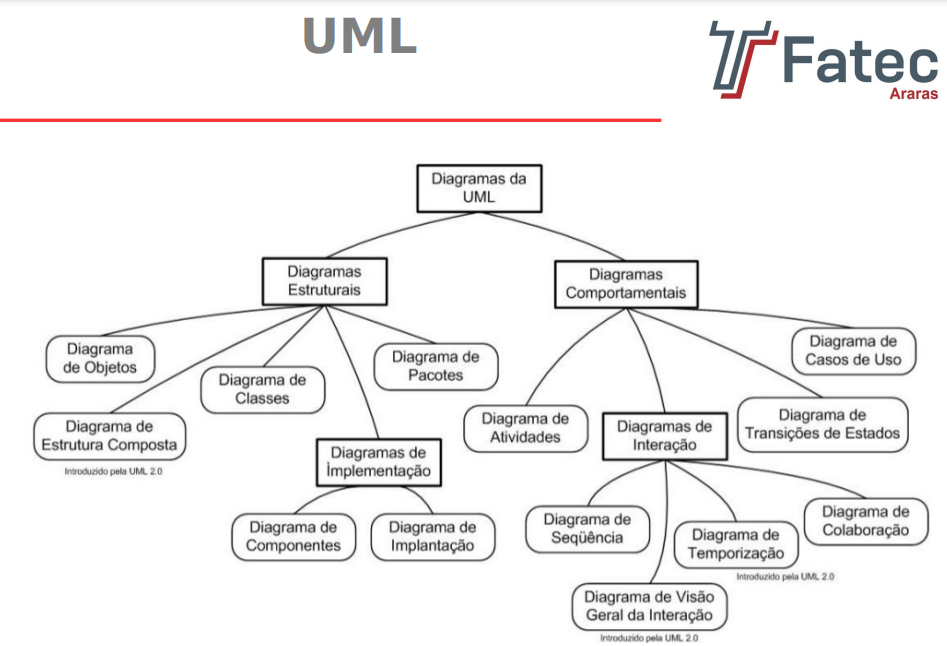
**Requisitos Funcionais e não funcionais**

**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



**UML:** UML (Unified Modeling Language) é uma linguagem visual para modelar sistemas orientados a objetos. Com uso destes elementos gráficos definidos nesta linguagem pode-se construir diagramas que representam diversas perspectivas de um sistema. Cada elemento gráfico possui uma sintaxe ( isto é, uma forma prédeterminada de desenhar o elemento) e uma semântica (o que significa o elemento e para que ele deve ser usado).



**DIAGRAMA DE CONTEXTO:** Em um estágio inicial da especificação de um sistema, você deve decidir os limites do sistema. Isso envolve trabalhar com os stakeholders do sistema para decidir qual funcionalidade deve ser incluída no sistema e o que é fornecido pelo ambiente do sistema.

