**Ciclo de vida software e engenharia de software**

**Engenharia de Software**

É uma área interdisciplinar de conhecimento, que agrega ciências exatas e humanas, e.g., engenharia e administração. A área rege o processo produtivo de sistemas baseados em rotinas computacionais, chamado [processo de *software*](https://pt.wikipedia.org/wiki/Engenharia_de_software#Processo_de_software), que por sua vez inclui especificação, [desenvolvimento](https://pt.wikipedia.org/wiki/Desenvolvimento_de_software), manutenção e criação de [*software*](https://pt.wikipedia.org/wiki/Software)

**valores de engenharia de software**

Rigor e Formalidade;

Rigor é um complemento necessário a criatividade que visa aumentar a confiança dos desenvolvimentos de software

Separação de Interesses:

Envolve dominar a complexidade, separando os problemas principais e concentrando-se em um de cada vez ( [dividir e conquistar](https://pt.wikipedia.org/wiki/Dividir_para_conquistar) ) suporte a paralelização de atividades e separação das responsabilidades.

Modularidade;

Alta Coesão;

Baixo Acoplamento.

Abstração;

Antecipação a Mudanças;

Generalidade;

Incrementação;

Requisitos de Software.

**Ciclo de vida do Software**

É uma estrutura que indica processos e atividades envolvidas no desenvolvimento, operação e manutenção de um software, abrangendo de fato toda a vida do sistema. Neste ciclo, existem modelos que definem como o software será desenvolvido, lançado, aprimorado e finalizado.

**Q que são metodologias ágeis e principais valores**:

É uma disciplina que estuda um conjunto de comportamentos, processos, práticas e ferramentas utilizados para a criação de produtos e sua subsequente disponibilização para os usuários finais.

**Principais valores**

Indivíduos e a interação entre eles, mais que processos e ferramentas;

Software em funcionamento, mais que documentação abrangente;

Colaboração com o cliente, mais que negociação de contratos;

Responder a mudanças, mais que seguir um plano.

**O que é scrum**

Scrum é uma estrutura ágil de colaboração em equipe comumente usada no desenvolvimento de software e em outros setores. O Scrum prescreve que as equipes dividam o trabalho em metas a serem concluídas dentro de iterações com limite de tempo, chamadas sprints.

**Engenharia está presente em nossas vidas**

Em nosso dia a dia, como as edificações, as obras de infraestrutura e todos os modais de transporte, tanto no meio urbano, quanto rural.

**Q que é sprint**

**É uma reunião de pessoas envolvidas num projeto para promover um desenvolvimento mais focalizado do projeto. O termo está fortemente relacionado ao framework de desenvolvimento ágil Scrum. Sprints normalmente têm duração de uma a quatro semanas.**

**Modelo cascata e incremental**

Enquanto que no modelo cascata a entrega da primeira versão funcional do software ocorre somente no final da sua construção,

O modelo incremental entrega a primeira versão funcional consideravelmente antes, pois o software é entregue por incrementos.

**Análise de requisitos**

É uma das etapas mais importantes do desenvolvimento de um projeto de software. Ela faz parte da fase de planejamento e tem como objetivo geral mapear o conjunto de ações e características que precisam compor o software

**Fale o que são e como surgiram as metodologias ágeis**

R: As metodologias ágeis surgiram como uma alternativa ao modelo tradicional de engenharia de software, herdado de outras engenharias.

**Fale sobre o scrum sem precisar se aprofundar na metodologia, mas o suficiente para passar uma base sobre a mesma**

R: É um framework de desenvolvimento ágil que promove a colaboração em equipes de desenvolvimento, que as equipes usam para se auto-organizar e trabalhar em direção a um objetivo em comum.

**Fale sobre como funciona a análise de requisitos em uma metodologia abordada no trabalho**

R: É um processo que integra o estudo das necessidades do usuário, para uma definição correta, completa gerando melhores softwares.

1) Os requisitos são fundamentais para definir as funcionalidades, comportamentos e restrições que um sistema deve atender. Eles servem como a base para testes e manutenção do software. Se um projeto não tivesse requisitos bem definidos ou se esses requisitos fossem negligenciados, diversos problemas poderiam surgir como no exemplo abaixo que mostra que os requisitos não foram levados a sério.

Ex: No Aeroporto de Denver, o requisito de limitar o tempo em solo das aeronaves a 30 minutos foi mal compreendido, levando um atraso de 16 meses e 1,1 milhões de dólares de prejuízo para cidade de Denver.

Ex: Em 1996 que o foguete europeu Ariane 5 explodiu em pleno ar o requisito de reutilizar o software de seu antecessor, o Ariane 4, que tinha alguns campos internos com tamanhos diferentes do Ariane 5. O requisito de compatibilidade entre os modelos não foi corretamente analisado

2) **Ser Tecnicamente Inviável**

R.: Quando um requisito é considerado tecnicamente inviável, significa que sua implementação prática é impraticável devido a restrições tecnológicas, limitações de recursos ou contradições com a arquitetura existente.

**Ser Difícil de Testar e Validar**

R.: A dificuldade em testar e validar um requisito torna desafiador determinar se foi implementado corretamente e atende aos critérios de aceitação. Isso pode resultar em erros não detectados, atrasos na entrega e incerteza sobre a conformidade do sistema.

3)

### **Ciclo de Vida de Software**

Ciclo de vida de um software é uma estrutura que indica processos e atividades envolvidas no desenvolvimento, operação e manutenção de um software, abrangendo de fato toda a vida do sistema. Neste ciclo, existem modelos que definem como o software será desenvolvido, lançado, aprimorado e finalizado.

**Concepção**: Define o escopo do projeto, ou “business case”; onde é julgado se o projeto deve ir adiante ou ser cancelado.

**Desenvolvimento**: Esse processo é geralmente dividido em várias fases, cada uma com objetivos específicos e funcionalidades específicas.

**Implantação**: Envolve a distribuição e instalação do software em ambientes de produção, tornando-o disponível para uso pelos usuários finais

**Manutenção**: Ele é essencial para garantir que o software permaneça importante, seguro e eficaz ao longo do tempo. Ajuda a contribuir para a durabilidade e o sucesso contínuo de um sistema.

### **Engenharia de Requisitos**

é o passo primordial para o desenvolvimento de um programa. Ela é formada por um conjunto de métodos e processos que buscam montar uma base de informações confiáveis e validadas para que o software seja construído

**Identificação de Requisitos:** É uma etapa fundamental no processo de desenvolvimento de software. Trata-se de um conjunto de técnicas e atividades que têm como objetivo identificar, documentar e compreender as necessidades e expectativas dos usuários e stakeholders em relação a um sistema ou produto

**Análise de Requisitos:** É uma das etapas mais importantes do desenvolvimento de um projeto de software. Ela faz parte da fase de planejamento e tem como objetivo geral mapear o conjunto de ações e características que precisam compor o software

**Especificação de Requisitos:** É um documento que descreve as necessidades específicas de um projeto ou sistema. A especificação de requisitos é importante porque serve como base para todo o trabalho futuro no projeto.

**Validação de Requisitos:** É o processo de confirmação de que os requisitos descritos estão de acordo com as solicitações das partes interessadas. Em outras palavras, a verificação é verificar se os requisitos estão completos, corretos e consistentes.

**Gerenciamento de Requisitos:** É o processo sobre a forma que as empresas definem, administram, verificam e validam as ideias e cumprem com os requisitos em cada etapa do ciclo de vida do produto, tão logo a ideação do desenvolvimento e comercialização do produto seja feita.

4) **Manutenção**

Quando aplicar: Ao longo da vida útil do software.

Por que aplicar: Correção de bugs, atendimento a requisitos legais, melhorias e novas funcionalidades.

**Descontinuação**

Quando aplicar: Quando o software não atende mais aos propósitos.

Por que aplicar: Evitar manutenção desnecessária, migrar para soluções mais eficientes ou atender a mudanças estratégicas na empresa.

A manutenção é focada em garantir a relevância e eficiência contínuas do produto, enquanto a descontinuação é uma decisão estratégica para retirar o produto do mercado quando ele não atende mais aos objetivos da empresa.

5) As técnicas de coleta de dados são métodos e procedimentos utilizados para obter informações relevantes e necessárias para uma determinada finalidade.

É possível utilizar essas técnicas em várias áreas, como pesquisa científica, análise de mercado, planejamento estratégico, entre outros.

**6) Requisitos Funcionais:** Descrevem as funcionalidades específicas que o sistema deve oferecer e de como o sistema deve se comportar em determinadas situações.

**EX**: Botão de incluir, alterar e excluir um registro, Função geração de um relatório específico extraído do banco de dados.

**Requisitos Não Funcionais:** Relacionam-se com características gerais, como desempenho, segurança e usabilidade, que não são funcionalidades diretas do sistema.

**EX:** Consumo de memória, conexão, banco de dados, tipos de dispositivos em que o software pode ser usado.

bolo de 1KG está 300

3Kg 600

5Kg 1500