

## Atividade 02, Part.2 – Conceitos de Ciclos de Vida

### 16. Cite e comente seis princípios de comunicação das práticas de Engenharia de Software

Os princípios da [engenharia de software](#) são sete de acordo com o cientista David Hooker. Eles são chamados gerais porque podem ser aplicados a uma única camada da engenharia de software, sua qualidade, processos, métodos ou ferramentas, bem como a ela como um todo.

- **Negociação não é um concurso ou um jogo. Funciona melhor quando ambas as partes ganham.**  
Muitas vezes, engenheiros de software e clientes precisam negociar e a negociação exige compromisso de todos.
- **Quando concordar com algo, prossiga; quando não concordar, prossiga; se algo não está claro e não pode ser esclarecido no momento, prossiga.**  
Prosseguir, às vezes, é a melhor forma de conseguir agilidade na comunicação.
- **Se algo não está claro, desenhe uma figura.**  
Quando as palavras não conseguem deixar algo bem esclarecido, normalmente, um desenho o faz.
- **Conserve-se enfocado, modularize sua discussão.**  
O facilitador deve manter a conversa modular, abandonando um tópico apenas depois que tiver sido resolvido.
- **Busque colaboração.**  
Procure fazer com que os membros da equipe façam colaborações, mesmo que pequenas, pois elas aumentam a confiança entre os membros.
- **Faça anotações e documente as decisões.**  
Alguém que participe da comunicação deve anotar todos os pontos e decisões importantes.

**17. Trace as semelhanças e diferenças entre o modelo Cascata e o modelo Incremental.**

O [modelo cascata](#) é um modelo que não permite a mudança do projeto após o seu início. Mudanças no decorrer do processo não são permitidas. Já o [modelo incremental](#), é um modelo onde você realiza as partes dos projetos separadamente, para depois integra-las em um sistema único, fornecendo mais flexibilidade, menos erros, e fácil correção, pois se uma etapa estiver errada, é possível corrigi-la facilmente, sem ter alterar o projeto inteiro.

**18. Qual a função do modelo RAD?**

O [Modelo RAD](#) é uma adaptação, de alta velocidade, do modelo em cascata, no qual a agilidade é conseguida com o uso de uma abordagem de construção baseada em componentes.

**19. Por que os modelos evolucionários recebem essa denominação?**

Modelos evolucionários são caracterizados por serem iterativos e apresentarem características que possibilitem desenvolvermos versões cada vez mais completas do software. Isso torna inadequado seguirmos um planejamento em linha reta de um produto. Os modelos de processo evolucionário tornaram-se realidade para que possamos desenvolver um produto que evolua ao longo do tempo.

**20. Cite as vantagens e desvantagens da Prototipagem.**

Desvantagens dos softwares de prototipação:

- **Falta de familiaridade:** como tudo, se você nunca usou, precisará aprender.
- **Fidelidade e funcionalidade limitadas:** com exceção de ferramentas como o Axure, a maioria das ferramentas de prototipagem é de baixa fidelidade ou de baixa funcionalidade.

Vantagens da Prototipagem:

- Criar uma representação semirealística de algo para que seja possível interagir e testar com usuários reais. A sua principal função é identificar problemas e oportunidades de melhorias muito antes de começar os refinamentos e a implementação.

**21. Em que momento a Prototipagem pode ser utilizada no modelo Espiral?**

A prototipagem pode ser utilizada no momento da Avaliação e Redução de riscos no modelo Espiral.

**22. Qual a diferença entre protótipo evolutivo e protótipo descartável?**

Na prototipagem evolutiva o sistema surge de evoluções refinadas dos protótipos enquanto um protótipo descartável é usado para descobrir problemas nos requisitos e depois é abandonado.

**23. Quais as vantagens em usar o modelo Espiral com relação ao modelo Cascata?**

O modelo espiral é mais versátil para lidar com as mudanças do que o modelo cascata. Com o modelo cascata a expectativa que o cliente possa gostar do software é bastante difícil, enquanto que no modelo espiral as estimativas são realistas.

**24. Caracterize o Processo Unificado.**

É um framework de processo da engenharia de software. Tem como fases Iniciação, Elaboração, Construção e Transição. Usa, assim como a maioria das metodologias, o modelo incremental e iterativo, o que possibilita aos desenvolvedores o aprendizado e localização de futuros problemas em fases iniciais.

**25. Qual a relação entre o Processo Unificado e a UML?**

O processo Unificado é derivado do Unified Software Process. Os diagramas UML são essenciais para processo de documentação e especificação dentro de um projeto de desenvolvimento de software.

**26. Cite e caracterize as fases do RUP**

- **Fase de Concepção/Iniciação:** esta fase do RUP abrange as tarefas de comunicação com o cliente e planejamento. É feito um plano de projeto avaliando os possíveis riscos, as estimativas de custo e prazos, estabelecendo as prioridades, levantamento dos requisitos do sistema e preliminarmente analisá-lo.
- **Fase de Elaboração:** abrange a Modelagem do modelo genérico do processo. O objetivo desta fase é analisar de forma mais detalhada a análise do domínio do problema, revisando os riscos que o projeto pode sofrer e a arquitetura do projeto começa a ter sua forma básica.
- **Fase de Construção:** desenvolve ou Adquire os componentes de Software. O principal objetivo desta fase é a construção do sistema de software, com foco no desenvolvimento de componentes e outros recursos do sistema.

- **Fase de Transição:** abrange a entrega do software ao usuário e a fase de testes. O objetivo desta fase é disponibilizar o sistema, tornando-o disponível e compreendido pelo usuário final.

**27. O que são Artefatos? Qual a relação com o RUP?**

Artefatos são Produtos de Trabalho bem definidos e tangíveis, consumidos, produzidos ou modificados pelas Tarefas. Podem ser compostos de outros Artefatos. Por exemplo, um modelo Artefato pode ser composto de elementos de modelos, os quais também são Artefatos. Eles podem servir como base para definição de RUP.

**28. Em que fase do RUP é feita a maior parte da codificação e os testes?**

É na fase de Construção que se desenvolver o software de maneira prática, havendo nesta a codificação e testes do projeto.

**29. Qual a importância da fase de Iniciação no RUP**

Essa fase estabelece o escopo e viabilidade econômica do projeto, sem a obtenção de uma base fornecida por essa fase não seria possível para a equipe a viabilização de um software de qualidade.

**30. O que você entende por abstração? Dê exemplos**

Tudo que não sabemos como funciona por baixo dos panos pode ser considerado uma abstração. Para exemplificar, irei tomar como exemplo uma concessionária que realiza manutenções no seu carro. Você o leva até lá com um problema e ele volta funcionando. Em questão, pouco importa os detalhes do que aconteceu durante a manutenção do seu carro, o que importa é que ele voltou funcionando.