Instituto de Educação, Ciência e Tecnologia – IFPI

Disciplina: Engenharia de Software

Professor: José Ritomar

Alunos: Hélio Rafael Miranda do Nascimento N°12

Philipe Vieira Oliveira N° 28

William Henrique de Moura e Silva N° 37

***Exercício***

1. Resposta: **D**
2. Resposta:

I – um processo de software é:

"é um conjunto de atividades, ligadas por padrões de relacionamento entre ela, pelas quais se as atividades operarem corretamente e de acordo com os padrões requeridos, o resultado desejado é produzido. O resultado desejado é um software de alta qualidade e baixo custo. Obviamente, um processo que não aumenta a produção (não suporta projetos de software grandes) ou não pode produzir software com boa qualidade não é um processo adequado."

Sucintamente podemos concluir que um processo de software como um conjunto de atividades uniformizadas a serem aplicadas sistematicamente que se encontram agrupadas em fases, cada uma das quais com os seus intervenientes com responsabilidades, que possui diversas entradas e produz diversas saídas. Isto é, define quem faz o quê, quando e como para atingir um certo objetivo.

II – Toda e qualquer definição de um software é dada em seu processo, sendo que cada processo é executado em uma fase especial. As principais fases de um processo de software são:

1. **Especificação de Requisitos:** tradução da necessidade ou requisito operacional para uma descrição da funcionalidade a ser executada.
2. **Projeto de Sistema**: tradução destes requisitos em uma descrição de todos os componentes necessários para codificar o sistema.
3. **Programação (Codificação)**: produção do código que controla o sistema e realiza a computação e lógica envolvida.
4. **Verificação e Integração (Verificação)**: verificação da satisfação dos requisitos iniciais pelo produto produzido.

Ao contrário do que possa parecer não existe uma sequência obrigatória de fases, sendo que diversos autores apontam a natureza não simultânea das fases como uma realidade na aplicação de processos de software, e também defendem que o processo de software é muito mais iterativo e cíclico do que a ideia de fases simples pode sugerir.

III. A validação do software também pode ser vista como uma fase do processo de software, que tem as suas atividades, assim como todos os outros. E em cada fase podemos identificar as seguintes atividades:

* 1. **Especificação**
     1. Engenharia de Sistema: estabelecimento de uma solução geral para o problema, envolvendo questões extra software.
     2. Análise de Requisitos: levantamento das necessidades do software a ser implementado. A Análise tem como objetivo produzir uma especificação de requisitos, que convencionalmente é um documento.
     3. Especificação de Sistema: descrição funcional do sistema. Pode incluir um plano de testes para verificar adequação.
  2. **Projeto**
     1. Projeto Arquitetural: onde é desenvolvido um modelo conceitual para o sistema, composto de módulos mais ou menos independentes.
     2. Projeto de Interface: onde cada módulo tem sua interface de comunicação estudada e definida.
     3. Projeto Detalhado: onde os módulos em si são definidos, e possivelmente traduzidos para pseudocódigo.
  3. **Implementação**
     1. Codificação: a implementação em si do sistema em uma linguagem de computador.
  4. **Validação**
     1. Teste de Unidade e Módulo: a realização de testes para verificar a presença de erros e comportamento adequado a nível das funções e módulos básicos do sistema.
     2. Integração: a reunião dos diferentes módulos em um produto de software homogêneo, e a verificação da interação entre estes quando operando em conjunto.
  5. **Manutenção e Evolução**
     1. Nesta fase, o software em geral entra em um ciclo iterativo que abrange todas as fases anteriores.

IV. Não existe um processo correto ou incorreto, como não existe um modelo de desenvolvimento que seja a panacéia universal para o problema do desenvolvimento de software.

Dependendo de sua aplicação, ambiente e objetivo, a utilização de um processo ou modelo específico pode ser vantajoso ou não. Cabe a cada organização avaliar o seu problema com cuidado e usar os modelos apresentados como um guia para o desenvolvimento do seu próprio processo de desenvolvimento.

1. Resposta:

O modelo também conhecido como abordagem*‘top-down’*, tem como principal característica a sequência de atividades onde cada fase transcorre completamente e seus produtos são vistos como entrada para uma nova fase. A ideia principal do modelo é que as diferentes etapas de desenvolvimento seguem uma sequência, ou seja, a saída da primeira etapa "fluí" para a segunda etapa e a saída da segunda etapa "fluí" para a terceira e assim por diante. As atividades a executar são agrupadas em tarefas, executadas sequencialmente, de forma que uma tarefa só poderá ter início quando a anterior tiver terminado. E assim o desenvolvimento segue até o fim do projeto.

1. Resposta:

O modelo pressupõe que o cliente participa ativamente no projeto e que sabe muito bem o que quer. Uma vez que se um processo não pode voltar atrás de modo a alterar os modelos e as conclusões das tarefas anteriores, é normal que as novas ideias sobre o sistema não sejam aproveitadas.

1. Resposta:

No processo de análise e negociação de requisitos um protótipo pode facilitar na recolhida de requisitos pois é usualmente mais fácil para os Stakeholders conseguirem identificarem exatamente o que querem de uma forma visual e aproximada do que poderá ser o produto final. Logo também será mais fácil na fase de validação de requisitos validar estes através do mesmo processo. Obviamente que através do protótipo visual é mais fácil detectar inconsistências e problemas nos requisitos. É também de referir a possibilidade de iniciar-se já nesta fase os manuais de utilização (pois normalmente são implementadas as interfaces). Deve-se notar duas pequenas desvantagens na utilização desta técnica. A implementação de um protótipo poderá levar a desilusões para os utilizadores finais, quando as interfaces da versão final não correspondem exatamente às do protótipo e poderá tentar os programadores a utilizar o protótipo como uma continuação do desenvolvimento do sistema. Todavia a prototipagem pode ser usada tanto em requisitos funcionais como em requisitos não funcionais.

1. Resposta:

Cada passagem pela parte de planejamento, resulta em ajustes no planejamento do projeto. O custo e o cronograma são sempre ajustados de acordo com o feedback obtido do cliente após uma entrega. Também teremos um ajuste no número de iterações planejadas para completar o software. Assim tomando mais tempo na documentação e também mais custos de aplicação.

1. Resposta:

O modelo de cascata. A afirmação corresponde à alguns dos defeitos do modelo, que apesar de ser teoricamente sequencial, apresenta também outras desvantagens como: É difícil para o cliente definir todos os requisitos no início do processo; Difícil adequação à mudanças de requisitos; Uma versão executável somente ficará pronta no final do processo.

1. Resposta:

Na verdade, ao invés do modelo cascata, a afirmativa sugere um modelo de prototipagem. Que procura suprir duas grandes limitações do modelo cascata. De acordo com Jalote a ideia básica deste modelo é que ao invés de manter inalterado os requisitos durante o projeto e codificação, um protótipo é desenvolvido para ajudar no entendimento dos requisitos. Este desenvolvimento passa por um projeto, codificação e teste, sendo que cada uma destas fases não é executada formalmente. Usando assim os protótipos o cliente pode entender melhor os requisitos do sistema.