

AVALIAÇÃO I - ENGENHARIA DE SOFTWARE

Aluno: Daniel Sant' Anna Andrade
Matrícula: 20200036904
Aluno: Luan Gadioli Mendonça Berto
Matrícula: 20200034042

Capítulo 1

1.1 Explique por que o software profissional não é apenas os programas que são desenvolvidos para o cliente.

R= Não se trata apenas do programa em si, mas de toda a documentação associada e dados de configurações necessários para fazer esse programa operar corretamente. Um sistema de software desenvolvido profissionalmente é, com frequência, mais do que apenas um programa; ele normalmente consiste em uma série de programas separados e arquivos de configuração que são usados para configurar esses programas. Isso pode incluir a documentação do sistema, que descreve a sua estrutura; documentação do usuário, que explica como usar o sistema; e sites, para usuários baixarem a informação recente do produto.

1.2 Qual a diferença mais importante entre o desenvolvimento de um produto genérico de software e o desenvolvimento de software sob demanda? O que isso pode significar na prática para usuários de produtos de software genérico?

R= Os produtos genéricos são produzidos por uma organização de desenvolvimento e vendidos no mercado para qualquer cliente que esteja interessado em comprá-los. Já os sob demanda são os sistemas encomendados por um cliente em particular. Uma empresa de software desenvolve o software especialmente para esse cliente.

Uma diferença importante entre esses tipos de software é que, em softwares genéricos, a organização que o desenvolve controla sua especificação. Para produtos sob encomenda, a especificação é normalmente desenvolvida e controlada pela empresa que está adquirindo o software

1.3 Quais são os quatro atributos importantes que todo software profissional deve possuir? Sugira outros quatro atributos que, às vezes, podem ser significantes.

R= Os quatro atributos mais importantes que todo software profissional deve possuir é a Manutenibilidade, Confiança e Proteção, Eficiência, Aceitabilidade.

Atributos que às vezes podem ser significativos é a reutilização (podendo ser utilizados em outras aplicações), facilidade de integração a outros sistemas, portabilidade (pode operar em várias plataformas, por exemplo, laptop e plataformas móveis) e rastreabilidade.

1.4 Além dos desafios de heterogeneidade, mudanças sociais e corporativas, confiança e proteção, identifique outros problemas e desafios que a engenharia de software provavelmente enfrentará no século XXI (Dica: pense no meio ambiente).

R= Necessidade de produção Software Sustentável ou Green Software, precisa durante todo o seu ciclo de vida gerar o menor impacto possível do ponto de vista social, ecológico e econômico. Principalmente levando em consideração a redução do consumo de energia e emissão de carbono. Focando na qualidade das fase de planejamento e análise, que antecedem o processo e atividades de desenvolvimento.

1.5 Baseado em seu conhecimento de alguns tipos de aplicações discutidos na Seção 1.1.2, explique, com exemplos, por que tipos de aplicações diferentes requerem técnicas especializadas de engenharia de software para apoiar seu projeto e desenvolvimento

R= Utiliza-se diferentes técnicas de engenharia de software para organizar, planejar, gerir e fazer os processos, para cada tipo de sistema, porque cada software tem características bastante diversas. Se não é bem provável que o software possua muitos erros, que seriam complicados de serem resolvidos posteriormente. Para um sistema Web, uma abordagem baseada em desenvolvimento e entregas iterativas pode ser adequada, com o sistema sendo composto a partir de componentes reutilizáveis. No entanto, tal abordagem pode ser inviável para um sistema de sistemas, no qual as especificações detalhadas das interações do sistema precisam estar detalhadas antes para que cada sistema possa ser desenvolvido separadamente.

Capítulo 2

2.2 Explique por que o desenvolvimento incremental é o método mais eficaz para o desenvolvimento de sistemas de software de negócios. Por que esse modelo é menos adequado para a engenharia de sistemas de tempo real?

R= O desenvolvimento incremental, atualmente, é a abordagem mais comum para o desenvolvimento de sistemas aplicativos, como por exemplo um programa de computador utilizado para processamento de dados que é desenvolvido para cumprir funções específicas.

Esse tipo de desenvolvimento é uma parte fundamental das abordagens ágeis. Ele reflete a maneira como resolvemos os problemas, raramente elaborando uma completa solução para o problema com antecedência e sim, gerando solução por solução, retornando ao cometer erros. Essa é uma forma de desenvolvimento que é mais barato e mais fácil de fazer mudanças no software durante seu desenvolvimento. Cada incremento incorpora alguma funcionalidade necessária para o cliente. Geralmente as primeiras features desenvolvidas são funcionalidades mais importantes ou mais urgentes. Assim, o cliente pode avaliar o sistema em um estágio relativamente inicial do desenvolvimento para ver se ele oferece o que foi requisitado. Em caso negativo, só o incremento que estiver em desenvolvimento no momento precisará ser alterado e, possivelmente, uma nova funcionalidade deverá ser definida para incrementos posteriores.

Apesar da facilidade desse sistemas, o desenvolvimento incremental possui dois problemas:

1. Para entregas regulares, não é viável produzir documentos que reflitam cada uma das versões do sistema.

2. O sistema tende a ficar confuso com a adição dos novos incrementos. A menos que de tempos em tempos sejam realizadas refatorações para melhoria do software, pois as constantes mudanças tendem a corromper sua estrutura.

2.5 Descreva as principais atividades do processo de projeto de software e as saídas dessas atividades. Usando um diagrama, mostre as possíveis relações entre as saídas dessas atividades.

R= As quatro principais atividades:

1. Especificação de software: É o processo de compreensão e definição dos serviços requisitados do sistema e identificação de restrições relativas à operação e ao desenvolvimento do sistema.

2. Projeto e implementação de software: É o processo de conversão de uma especificação do sistema em um sistema executável. Sempre envolve processos de projeto e programação de software, mas, se for usada uma abordagem incremental para o desenvolvimento, também pode envolver o refinamento da especificação do software.

3. Validação de software: É o processo que mostra que um software se adequa a suas especificações ao mesmo tempo que satisfaz as especificações do cliente do sistema. A validação pode envolver processos de verificação, como inspeções e revisões, em cada estágio do processo de software, desde a definição dos requisitos de usuários até o desenvolvimento do programa

4. Evolução de software: Apesar de ser muito caro fazer alterações no projeto, as mudanças no software podem ser feitas a qualquer momento durante ou após o desenvolvimento do sistema. Mesmo grandes mudanças são muito mais baratas do que as correspondentes alterações no hardware do sistema.

2.7 Explique por que os sistemas desenvolvidos como protótipos normalmente não devem ser usados como sistemas de produção:

R= Impossibilidade de ajustar o protótipo para atender a todos os requisitos básicos de forma eficiente, como requisitos de desempenho, proteção, robustez e confiabilidade, que foram ignorados durante o desenvolvimento do protótipo por necessitar de uma versão de uso rápido ou teste.

Mudanças rápidas durante o desenvolvimento pode significar que a mudança implementada não seja documentada. Sendo assim, a única especificação de projeto é o código do protótipo, sendo ruim para a manutenção a longo prazo.

As mudanças durante o desenvolvimento podem degradar a estrutura do sistema, aumentando o custo.

2.9 Quais são as vantagens de proporcionar visões estáticas e dinâmicas do processo de software, assim como no Rational Unified Process?

R= A vantagem de proporcionar visões estáticas e dinâmicas é que as fases do processo de desenvolvimento não estão associadas a workflows específicos. Ao menos em princípio, todos os workflows do RUP podem estar ativos em todas as fases do processo. Nas fases iniciais, provavelmente, maiores esforços serão empenhados em workflows, como modelagem de negócios e requisitos, e, nas fases posteriores, no teste e na implantação.

Capítulo 24

24.1 Explique por que um processo de software de alta qualidade deve levar a produtos de software de alta qualidade. Discuta possíveis problemas com esse sistema de gerenciamento de qualidade.

R= Padrões e processos são importantes, porém os gerentes de qualidade devem ter como objetivo desenvolver uma 'cultura de qualidade' na qual todos os responsáveis pelo desenvolvimento estejam comprometidos em alcançar um alto nível de qualidade de produto.

Existe também uma clara ligação entre a qualidade de processo e de produto na manufatura, pois o processo é fácil de ser padronizado e monitorado. Quando o sistemas de manufatura é calibrado, ele pode ser executado várias vezes para produzir produtos de alta qualidade. Porém, no desenvolvimento do software se utiliza processos criativo, em vez de mecânico. Sendo assim, a influência de competências e experiências individuais é significativa, pois, os fatores externos também afetam a qualidade do produto, independentemente do processo usado.

É por final, projetos que desenvolveram softwares de alta qualidade podem ser cancelados por mudanças nos negócios ou em seu ambiente operacional

24.2 Explique como os padrões podem ser usados para capturar a sabedoria organizacional a respeito de métodos eficazes de desenvolvimento de software. Sugira quatro tipos de conhecimentos que possam ser capturados em normas organizacionais.

R= Eles são baseados em conhecimentos sobre a prática do que é melhor ou mais adequado para a empresa. Muitas vezes, esse conhecimento é adquirido após bastante tentativa e erro. Inserir-lo em um padrão ajuda a empresa a reusar essa experiência e evitar erros anteriormente cometidos.

Conhecimentos:

1. Qualidade de software
2. Padrões de software
3. Revisões e Inspeções
4. Medições e métricas de software

24.3 Discuta a avaliação de qualidade de software de acordo com os atributos de qualidade mostrados na Tabela 24.1. Você deve considerar cada atributo e explicar como ele pode ser avaliado

R= Em relação a segurança: ser possível avaliar o quão protegido é um sistema da invasão de usuários não permitidos; ser capaz de não ter falhas na utilização que possam deixar o sistema exposto; ter capacidade para funcionar mesmo em condições anormais; e ser capaz de sofrer ataques simples por conta própria.

Em relação a compreensibilidade: testar com constância o funcionamento do produto; dividir o software em partes distintas gerando um aumento da produtividade; ter um software capaz de se adaptar a ambientes diferentes sem a aplicação de ações; possuir um produto que não seja complexo de mais a ponto de atrapalhar o desenvolvimento ou a utilização.

Em relação a portabilidade: capacidade do sistema em fazer com que o usuário tenha sucesso na execução de suas tarefas; utilização de soluções que melhorem a qualidade e a produtividade; produção capaz de ser implementada para diversos cenários de forma simples e não gerando problemas de implementação ou de gerenciamento.

24.6 Suponha que você trabalhe para uma organização que desenvolve produtos de banco de dados para indivíduos e empresas de pequeno porte. Essa organização está interessada na quantificação de seu desenvolvimento de software. Escreva um relatório sugerindo métricas adequadas e sugira como estas podem ser coletadas.

R= Para quantificar o desenvolvimento de software neste caso, é necessário a utilização de métricas de software e métricas de produto. As métricas de software avaliam a qualidade do software, sendo alguns exemplos como: número de colaboradores no desenvolvimento do software e a quantidade de tempo. Também será necessário ressaltar as métricas de produto, o tamanho do sistema, métodos, objetos necessários ao produto. Essas serão informações imprescindíveis no processo de elaboração, desenvolvimento e manutenção do software, visto que um banco de dados pode ter uma relação dinâmica ou estática, é importante ressaltar que ao depender da tecnologia ou do processo que será utilizado no produto, as métricas podem variar.

24.7 Explique por que inspeções de programa são uma técnica eficaz para descobrir erros em um programa. Que tipos de erros são improváveis de serem descobertos por meio de inspeções?

R= As inspeções de programa envolvem membros de equipe de diferentes origens fazendo uma revisão cuidadosa, linha por linha de código-fonte de programa. Eles procuram defeitos e problemas e os descrevem em uma reunião de inspeção. As inspeções podem descobrir problemas com testes e, assim, melhorar a eficácia desses testes em detectar bugs no programa.

24.8 Explique por que as métricas de projeto são, por si só, um método inadequado de previsão de qualidade de projeto.

R= A medição de software preocupa-se com a derivação de um valor numérico ou o perfil para um atributo de um componente de software, sistema ou processo. Comparando esses valores entre si e com os padrões que se aplicam a toda a organização, você pode ser capaz de tirar conclusões sobre a qualidade do software ou avaliar a eficácia dos métodos, das ferramentas e dos processos de software.

24.9 Explique por que é difícil validar os relacionamentos entre os atributos internos de produto, como complexidade ciclomática e atributos externos, como a manutenibilidade.

R= Eles são afetados por fatores subjetivos, como a experiência e a educação do usuário e, portanto, não podem ser medidos objetivamente. Necessitando de um julgamento sobre esses atributos, medir alguns atributos internos do software como tamanho e complexidade e assumir que estão relacionados com as características de qualidade com as quais você se preocupa.

Capítulo 26

26.1 Quais são as diferenças importantes entre a abordagem ágil e a abordagem de maturidade de processo para a melhoria de processos de software?

R= Os métodos ágeis foram desenvolvidos para serem usados por equipes de programação de pequeno e médio porte que poderiam trabalhar juntas na mesma sala e se comunicar de maneira informal. As principais características dos métodos ágeis são a entrega rápida de funcionalidade e a capacidade de resposta às mudanças de requisitos de cliente. Já na abordagem de maturidade de processos, os principais objetivos são produtos de melhor qualidade e previsibilidade de processo. Se centrando em melhorar o gerenciamento de processos e projetos e em introduzir boas práticas de engenharia de software em uma organização.

26.2 Em quais circunstâncias a qualidade de produto pode ser determinada pela qualidade da equipe de desenvolvimento? Dê exemplos de tipos de produtos de software que são particularmente dependentes de talentos e capacidades individuais

R= No caso de pequenos projetos com poucos membros, a qualidade da equipe de desenvolvimento será mais importante do que o processo utilizado. Se a equipe toda tiver um alto nível de capacidade e experiência, possivelmente a qualidade do produto será alta, não importando o processo usado.

Projetos na área de entretenimento rápido geralmente são os que mais precisam de qualidade de equipe. Muitas ideias simples e de fácil implementação não necessitam de grandes produções e processos de desenvolvimentos complicados, eles utilizam apenas o básico para reproduzir a ideia e entrar logo no mercado.

Aplicativos que fazem tarefas básicas também são muito mais necessários de equipe inovadora do que um processo para realizá-lo. Geralmente esse tipo de produto é simples e pode ser criado com poucas coisas.

26.5 Descreva três tipos de métricas de processo de software que podem ser coletadas como parte de um processo de melhoria de processos. Dê um exemplo de cada tipo de métrica.

R=

1° - o tamanho de um produto de software (exemplo: número de linhas de código - LOC)

2° - eficiência (exemplo: número de defeitos relatados em um produto de software entregue)

3º - Esforço (exemplo: número de pessoas/dia requerido para desenvolver um componente de sistema).

26.9 Quais são as vantagens e as desvantagens do uso de um modelo de maturidade de processo que se concentra nas metas a serem atingidas, em vez de nas boas práticas a serem introduzidas?

R= Vantagens: O desenvolvimento de software é com qualidade, garantindo o cumprimento dos prazos e atendendo às necessidades do cliente, deixando-o mais satisfeito com o produto entregue pela empresa, eliminando as inconsistências e reduzindo a duplicidade.

Desvantagens: A sua utilização não garante sucesso no desenvolvimento dos projetos. A melhoria deve ser contínua mesmo após atingir maiores níveis de maturidade dos processos. Sendo assim, empresas de pequeno e médio porte possuem empecilhos de tempo e de investimento para alcançar esses níveis mais altos.

26.10 Você acha que programas de melhoria de processos, que envolvem mensurar o trabalho de pessoas no processo e a introdução de mudanças nesse processo, podem ser inerentemente desumanizados? Qual resistência pode surgir a um programa de melhoria de processos e por quê?

R= Sim, porque você sempre deve usar a medição em conjunto com a avaliação qualitativa das mudanças. Isso envolve conversar com as pessoas envolvidas no processo a respeito das mudanças introduzidas e receber suas impressões sobre a eficácia dessas mudanças. Esse processo revela outros fatores que podem ter influenciado o processo, como também revela a extensão com que a equipe adotou as mudanças propostas e como elas têm afetado a prática real de desenvolvimento.