

1. Explique o conceito de qualidade de software e o papel do processo de garantia de qualidade (QA) no desenvolvimento de software. Discuta as principais atividades e técnicas utilizadas no processo de QA, como revisões técnicas, testes de software e medição de qualidade. Discuta também a importância da melhoria contínua da qualidade de software e como ela pode ser alcançada.

R: O conceito de qualidade de software pode ser entendido como a colaboração da qualidade interna e da qualidade externa. A qualidade interna diz respeito às características da implementação do código, analisando fatores como a legibilidade, a testabilidade e a manutenibilidade. Por sua vez, a qualidade externa tem relação com fatores que não dependem da análise do código, levando em consideração fatores como a eficiência, a usabilidade e a portabilidade.

O processo de garantia de qualidade (QA) tem um papel fundamental no desenvolvimento de software ao analisar os fatores de qualidade interna e os de qualidade externa ao projeto, de forma a garantir que o sistema mantenha um padrão de qualidade adequado.

As revisões técnicas são realizadas antes de uma implementação entrar em produção, onde um desenvolvedor alheio ao desenvolvimento da implementação em questão revisa seu código, com o objetivo de detectar bugs antecipadamente. Já os testes de software submetem o programa a diversos casos de teste, com o objetivo de verificar se o mesmo apresenta o comportamento esperado. Por fim, a medição de qualidade aplica diversas métricas, como métricas de código fonte e métricas de processo.

A melhoria contínua da qualidade de software é importante para que o sistema se mantenha alinhado às atualizações das tecnologias por ele usadas e aos avanços tecnológicos no geral. Ela pode ser alcançada por meio da boa aplicação das atividades e técnicas utilizadas no processo de QA.

2. Explique o conceito de Engenharia de Requisitos e seu papel no processo de desenvolvimento de software. Descreva as principais atividades e técnicas envolvidas na engenharia de requisitos, como elicitação, análise, especificação e validação de requisitos. Discuta também os desafios comuns na gestão de requisitos e como eles podem ser superados.

R: A Engenharia de Requisitos é um conjunto de técnicas bem definidas aplicadas da descoberta à manutenção dos requisitos de um sistema. Essa engenharia é de suma importância para o desenvolvimento de software, uma vez que sua sistematização garante o uso de boas práticas e, consequentemente, o bom desenvolvimento e funcionamento do sistema.

A primeira fase da Engenharia de Requisitos se dá pela elicitação e pela análise de requisitos. Na elicitação é onde se descobre e entende os requisitos do sistema que se pretende desenvolver, seja por meio de reuniões com o cliente, seja por meio de outros estudos. A análise dos requisitos complementa a elicitação, de forma a promover a classificação, priorização, e modelagem dos requisitos que descrevem o sistema.

Em seguida, é realizada a especificação de requisitos do usuário e do sistema. Os requisitos de usuário são essenciais para o usuário e o funcionamento do sistema, enquanto os requisitos de sistema são importantes para o bom funcionamento do sistema em termos de desempenho. Nessa fase, se escreve a documentação dos requisitos, que, no caso do desenvolvimento ágil, se dá pelas histórias de usuário.

Por fim, é feita a verificação e a validação dos requisitos. Os objetivos dessa fase é garantir que os requisitos estejam corretos, precisos, completos, consistentes e verificáveis. Para a verificação, se pergunta “Estamos fazendo o sistema corretamente?”, ou seja, de acordo com os requisitos; enquanto, para a validação, se pergunta “Estamos fazendo o sistema correto?”, ou seja, o sistema que o cliente quer.

Os maiores desafios na gestão de requisitos envolvem problemas com a especificação, seja por incompletude, falta de documentação, inconsistência ou abstração; e problemas de falta de comunicação, seja por parte da equipe de desenvolvimento ou dos stakeholders. Todos esses problemas podem ser solucionados pela boa aplicação da Engenharia de Requisitos.

3. Discuta os princípios e práticas do desenvolvimento ágil de software. Explique os valores e os princípios do Manifesto Ágil e como eles influenciam o processo de desenvolvimento. Descreva os papéis e as atividades-chave em uma equipe ágil e discuta como o desenvolvimento ágil promove a colaboração, a adaptabilidade e a entrega de valor ao cliente.

R: O desenvolvimento ágil apresenta um novo modelo, incremental e iterativo, que preza pela rapidez, praticidade e maleabilidade proporcionadas por um desenvolvimento mais dinâmico, com menor ênfase em documentação e em planejamento detalhado (big upfront design).

O Manifesto Ágil apresenta uma crítica aos modelos sequenciais e pesados e propõe uma definição leve de processos. Esse manifesto valoriza as pessoas e suas interações, o funcionamento do software, a boa comunicação com os clientes e a receptividade de mudanças acima de qualquer “burocracia”, o que torna o processo de desenvolvimento mais fluido e leve.

Em uma equipe ágil, os papéis são definidos como: Product Owner (PO), o especialista no domínio do sistema; Scrum Master, o especialista em Scrum que

tem o objetivo de guiar a equipe no seguimento dos eventos; desenvolvedores, que são os demais membros.

O desenvolvimento ágil promove a colaboração, a adaptabilidade e a entrega de valor ao cliente por meio dos valores do Manifesto Ágil, que prezam pela boa convivência dos desenvolvedores, pela maleabilidade do desenvolvimento e pelo diálogo com o cliente.

4. Descreva o modelo de processo de desenvolvimento ágil, como o Scrum, e suas principais características. Explique os princípios fundamentais do desenvolvimento ágil e como eles se aplicam no contexto do Scrum. Discuta as vantagens do Scrum em relação a outros modelos de processo de desenvolvimento.

R: O modelo de processo de desenvolvimento ágil é incremental e iterativo, com menor ênfase em documentação e em (big upfront design) e maior foco em colaboração, adaptabilidade e entrega de valor. Esse modelo segue o Manifesto Ágil, que valoriza as pessoas e suas interações, o software em funcionamento e o bom relacionamento com o cliente.

O Scrum é um tipo de modelo ágil que define papéis, eventos e artefatos bem claros. Aqui, o bom relacionamento com o cliente é essencial para o Product Owner formular as histórias de usuário, as interações da equipe de desenvolvimento são essenciais nos processos envolvendo os requisitos e suas implementações, e o software em funcionamento é essencial para a incrementabilidade e iteratividade do desenvolvimento.

As vantagens do Scrum em relação a outros modelos ágeis, como o Extreme Programming, se dão pela boa definição dos papéis (Product Owner, Scrum Master e desenvolvedores), eventos (sprints) e artefatos (backlog do sistema e backlog do sprint). Já a vantagem do Scrum em relação aos modelos em cascata se configura pela praticidade e adaptabilidade promovida pelo desenvolvimento ágil.