

Testes de Software
Prof.º Edson Saraiva de Almeida

05/09/2020

Bruno Bega Harnik – 1110481823052
Fernanda Pinheiro Reis – 1110481823022

1. Descreva as principais preocupações relacionadas ao gerenciamento da qualidade para sistemas de software.

Existem três principais preocupações:

- Estabelecimento de um conjunto de premissas que aumentem a qualidade do software - um framework, por exemplo. A equipe organizacional, de gerenciamento de qualidade deve definir os processos de desenvolvimento do software, bem como a documentação que inclui requisitos de sistema, projeto e código;
- Em nível de projeto, verificação de cumprimento dos processos planejados. Isso garante que as saídas do projeto estejam em conformidade com os padrões aplicáveis ao projeto;
- Também em nível de projeto, devemos nos preocupar em estabelecer um plano de qualidade, a fim de definir metas de qualidade e padrões aplicáveis ao projeto.

2. Descreva uma estrutura geral que pode ser utilizada na elaboração de um plano de qualidade.

Humpfrey, em seu livro sobre gerenciamento de software, sugere a seguinte estrutura:

- Introdução ao produto - descrição, ambiente de mercado e expectativas de qualidade;
- Planos de produto - datas críticas de release, responsabilidades de entrega, planos de distribuição e prestação de serviço do produto;

- Descrições de processo - padrões utilizados no gerenciamento e no desenvolvimento do produto;
- Metas de qualidade - identificação e justificativa para atributos críticos de qualidade do produto;
- Riscos e gerenciamento de riscos - riscos que podem afetar a qualidade do produto e ações que tomadas para lidar com eles.

3. Na análise da qualidade de software não se considera apenas se a funcionalidade foi corretamente implementada, a qualidade também depende dos atributos não funcionais de sistema. Boehm et al. (1978) sugerem um conjunto de 15 atributos importantes na avaliação da qualidade do software descreva quais são.

Os 15 atributos são:

- Segurança
- Proteção
- Confiabilidade
- Resiliência
- Robustez
- Compreensibilidade
- Testabilidade
- Adaptabilidade
- Modularidade
- Complexidade
- Portabilidade
- Usabilidade
- Reusabilidade
- Eficiência
- Capacidade de aprendizado

4. Descreva o fluxo da abordagem da qualidade baseada em processos.

A definição do processo é o primeiro passo em uma abordagem de qualidade baseada em processos. Em seguida, entramos em fase de desenvolvimento de produto. Em seguida, avaliamos sua qualidade. Caso esteja bom, segue ao processo de padronização. Caso não atinja a qualidade, segue para melhoria de processos, ou seja, retorna à fase de desenvolvimento do produto.

5. Descreva os dois tipos de padrões que podem ser selecionados e usados no gerenciamento da qualidade de software.

Os dois tipos de padrões utilizados são os padrões de produto e os padrões de processo. Os padrões de produto aplicam-se ao software e incluem documentação - requisitos e padrões de documentação - e padronização de código.

Já os padrões de processo definem os processos a serem seguidos durante o desenvolvimento de software. Devem incluir boas práticas de desenvolvimento e definições de especificação, projeto, processos de validação, ferramentas de suporte e descrição dos documentos que devem ser escritos durante o processo.

6. Faça uma pesquisa sobre padrões de código para linguagem Java, selecione uma das cláusulas do padrão e descreva seu objetivo. Como um padrão de codificação pode ajudar na gestão da qualidade do produto de software?

Padrão de nomenclatura

Aplicada a nomes de: Pacotes, classes, interfaces, métodos, variáveis e constantes.

O objetivo da convenção de nomenclaturas é fazer com que o código fonte se torne mais legível, facilitando a sua compreensão. O padrão de nomenclatura pode fornecer um melhor entendimento sobre qual a função daquele identificador, auxiliando assim na melhor compreensão do código, facilitando também futuras implementações e manutenções.

Fonte:

<https://www.oracle.com/java/technologies/javase/codeconventions-namingconventions.html>

Um padrão de codificação auxilia na produção de código fonte mais limpo, ou seja, mais legível, melhor estruturado e auxilia as manutenções futuras, visto que se torna muito mais fácil corrigir algo que é melhor compreendido. Estes padrões definem uma estética comum ao código fonte, reduzindo também a chance de haver inconsistências ao se desenvolver uma nova funcionalidade ou aplicar uma atualização.

7. Descreva os procedimentos de revisão e inspeções utilizados no controle da qualidade do software.

Revisões e inspeções são atividades que realizam o controle de qualidade, verificando a qualidade dos componentes entregáveis de projeto, esta tarefa envolve examinar o software, a sua documentação e os registros do processo de desenvolvimento para descobrir erros e omissões e também verificar se os padrões de qualidade foram seguidos corretamente. Revisões e inspeções são usadas em conjunto com o teste de programa, como uma parte do processo geral de validação e verificação do software.

Durante a revisão, algumas pessoas verificam o software e a documentação a procura de possíveis problemas e inconsistências com os padrões. Então a equipe de revisão informada sobre o nível de qualidade de um sistema toma decisões, então os gerentes de projeto podem utilizar essa avaliação para tomar decisões do planejamento, alocando recursos para o processo de desenvolvimento.

As inspeções são revisões em que os membros da equipe se auxiliam na busca de bugs no programa que está sendo desenvolvido. As inspeções podem fazer parte dos processos de verificação e validação de software.

Envolvem membros de equipes diferentes, fazendo uma revisão minuciosa em linha a linha do programa, então os problemas encontrados são discutidos em uma reunião de inspeção, podendo ser erros lógicos, anomalias de código que possa indicar uma condição errada ou recursos que foram omitidos do código,

então a equipe de revisão examina em detalhes os modelos do projeto ou do código fonte e destaca as anomalias para que sejam corrigidas.

8. Como as medições e métricas de software podem ser utilizadas no processo de revisão do software.

A medição de software trata da derivação de um valor numérico ou perfil para um atributo de componente de software, sistema ou processo. Realizando a comparação destes valores com os valores que são aplicados na organização é possível avaliar a qualidade do software ou a eficácia de métodos, ferramentas e processos de software.

Uma métrica de software é uma característica de um sistema, documentação de sistema ou processo de desenvolvimento que pode ser medido a para atender a uma necessidade, como por exemplo o tamanho em linhas de código, o número de defeitos relatados no produto de software entregue, e o número de pessoas/dias necessários para desenvolver um componente de sistema. As métricas podem ser de controle ou de previsão. sendo as de controle as que suportam os processos de gerenciamento e as de previsão que ajudam a prever características do produto de software. Os gerentes fazem uso das métricas de processo para decidir se deverão ocorrer alterações no processo, já as métricas de previsão são utilizadas para auxiliar nos esforços necessários para que se realizem as devidas alterações no projeto de software.

9. No item “4 – Metas de Qualidade” deve-se selecionar o padrão de produto e ou de processo (questão 5) que serão estabelecidos para planejar a qualidade do projeto. Supondo que uma organização decidiu adotar uma estrutura para documentação dos requisitos como meta de qualidade nos seus projetos, faça uma pesquisa para identificar “padrões de escrita de requisitos” que poderiam ser utilizados por esta organização e descreva como seria possível controlar a qualidade para este projeto (significa verificar se o padrão está sendo seguido).

O levantamento de requisitos analisa as necessidades do cliente e combina elementos de resolução de problemas, elaboração, negociação e especificação. É

viável que as equipes trabalhem em conjunto para identificar o problema, propor elementos da solução, negociar diferentes abordagens e especificar um conjunto preliminar de requisitos da solução. Podem organizar uma reunião para discutir tais requisitos.

Os artefatos reunidos durante e depois uma reunião de levantamento de requisitos são os seguintes:

- Uma declaração da necessidade e viabilidade.
- Uma declaração restrita do escopo para o sistema ou produto.
- Uma lista de clientes, usuários e outros interessados que participaram do levantamento de requisitos.
- Uma descrição do ambiente técnico do sistema.
- Uma lista de requisitos (preferencialmente organizada por função) e as restrições de domínio que se aplicam a cada uma delas.
- Um conjunto de cenários de uso que esclarecem o uso do sistema ou produto sob diferentes condições operacionais.
- Quaisquer protótipos desenvolvidos para melhor definição dos requisitos.

Além disso, temos que analisar quais são todos os cenários possíveis (casos de uso) para que todos os caminhos sejam cobertos. Até descobrir todos os atores, são realizadas algumas reuniões de equipe. As seguintes perguntas devem ser respondidas para elaborar os casos de uso:

- Quem é(são) o(s) ator(es) primário(s) e o(s) ator(es) secundário(s)?
- Quais são as metas do ator?
- Que condições devem existir antes de uma história começar?
- Que tarefas ou funções principais são realizadas pelo ator?
- Que exceções deveriam ser consideradas à medida que uma história é descrita?
- Quais são as variações possíveis na interação do ator?
- Que informações de sistema o ator adquire, produz ou modifica?
- O ator terá de informar o sistema sobre mudanças no ambiente externo?
- Que informações o ator deseja do sistema?

- O ator gostaria de ser informado sobre mudanças inesperadas?

Os padrões de escrita de requisitos devem ser definidos pelos engenheiros de software para que os requisitos sejam tecnicamente bem descritos, facilitando assim o trabalho de divisão das tarefas em partes menores. Algumas ferramentas como Jira , Helix e Orcano auxiliam na escrita dos requisitos pois ajudam as equipes a identificar e mapear os requisitos de negócios, colaborar com as partes interessadas, garantir que todas as tarefas se conectem diretamente a qualquer requisito de captura e fornecer às partes interessadas entregas de alta qualidade. Permitem um melhor monitoramento da progressão das atividades, tarefa a tarefa, e permite que a qualidade seja alcançada a partir do período dedicado a ela.

A disponibilização da função de qualidade (quality function deployment, QFD) é uma técnica de gestão da qualidade que traduz as necessidades do cliente para requisitos técnicos do software. O QFD "concentra-se em maximizar a satisfação do cliente por meio do processo de engenharia de software". O QFD é dividido em:

- Requisitos normais: objetivos e metas estabelecidos para um produto durante reuniões com o cliente;
- Requisitos esperados. Esses requisitos estão implícitos no produto (ex. segurança, usabilidade)
- Requisitos fascinantes. Esses recursos vão além da expectativa dos clientes

Controlar a qualidade:

- entrega do software
- satisfação do usuário = produto compatível + boa qualidade + entrega dentro do prazo previsto
- fatores de qualidade, segundo McCall: correção, confiabilidade, eficiência, integridade, usabilidade, facilidade de manutenção, flexibilidade, testabilidade, portabilidade, reusabilidade, interoperabilidade.

Os fatores de qualidade que devem ser cobertos pelo software que atendeu a todos os requisitos são:

- Funcionalidade. O grau com que o software satisfaz às necessidades declaradas conforme indicado pelos seguintes subatributos: adequabilidade, exatidão, interoperabilidade, conformidade e segurança.
- Confiabilidade. A quantidade de tempo que o software fica disponível para uso conforme indicado pelos seguintes subatributos: maturidade, tolerância a falhas, facilidade de recuperação.
- Usabilidade. O grau de facilidade de utilização do software conforme indicado pelos seguintes sub atributos: facilidade de compreensão, facilidade de aprendizagem, operabilidade.
- Eficiência. O grau de otimização do uso, pelo software, dos recursos do sistema conforme indicado pelos seguintes subatributos: comportamento em relação ao tempo, comporta mento em relação aos recursos.
- Facilidade de manutenção. A facilidade com a qual uma correção pode ser realizada no software conforme indicado pelos seguintes sub atributos: facilidade de análise, facilidade de realização de mudanças, estabilidade, testabilidade.
- Portabilidade. A facilidade com a qual um software pode ser transposto de um ambiente a outro conforme indicado pelos seguintes subatributos: adaptabilidade, facilidade de instalação, conformidade, facilidade de substituição.

