

Podcast

Disciplina: Teste e inspeção de software: técnicas e automatização

Título do tema: Técnicas de Inspeção de Software

Autoria: Stella Marys Dornelas Lamounier

Leitura crítica: Marcilyanne Moreira Gois

Abertura:

Olá, ouvinte! No podcast de hoje vamos falar de Controle de Versões de Software

A revisão de software é sem dúvida uma ótima forma de melhorar a qualidade de software, devendo ser aplicada em várias fases de desenvolvimento de um sistema e deve apontar melhorias necessárias ao software, confirmar se tal melhoria é ou não necessária, é também responsável por realizar trabalho técnico e formal de forma que a revisão se torne mais fácil no decorrer do andamento do projeto.

Em um simples café no intervalo de uma tarefa ou outra podem ser discutidos problemas técnicos e isso pode sim ser tratado como uma revisão para melhoramento do software.

É importante lembrar que Inspeção e Teste se complementam, trabalhando juntos na remoção de defeitos. A Inspeção encontra defeitos não encontrados em teste e Teste encontra defeitos não encontrados na fase de inspeção ou que tenham sido inseridos depois da fase de Inspeção.

A técnica de inspeção verifica se o sistema atende aos requisitos, se estão de acordo como padrões pré-definidos, se o sistema está tomando mais uniforme, enfim, é capaz de eliminar cerca de 50 a 90% dos defeitos do processo de construção do software, antes de começar a execução dos testes. A Inspeção pode salvar tempo de desenvolvimento, pois remove defeitos em fases anteriores, reduzindo substancialmente os custos com correção em fases posteriores.

Contudo, a quantidade de empresas onde a inspeção de software faz parte da cultura e do processo organizacional ainda é pequena face os benefícios de sua utilização

Mas não estamos limitados a apenas técnicas manuais ou reuniões em grupos, existem ferramentas automatizadas que garantem com muito mais agilidade o processo de Inspeção, como softwares de controle de versões, que tem como finalidade controlar diferentes versões em documentos, artefatos ou código fonte, podemos citar Visual SourceSafe (VSS) produto da Microsoft para controle de versão, integrado a muitas IDEs da Microsoft sendo uma ferramenta proprietária e o Concurrent Version System (CVS) software livre onde seu controle de versões é feito de forma individual e enviados a um servidor e também o GIT, uma poderosa ferramenta também Open Source para controle de versões na nuvem, onde é possível criar históricos de

atualizações de projetos e se por ventura o desenvolvedor necessitar voltar ao projeto anterior ele estará lá armazenado.

É preciso ressaltar que cada um dos sistemas tem características próprias, mas a grande maioria destes sistemas possuem algumas similaridades: Basicamente são repletos de estações de trabalhos espalhadas ligadas a um servidor que é responsável pelo armazenamento de versões e históricos, onde o desenvolvedor pode enviar ou resgatar versões, a cada nova alteração são criadas versões que podem ser resgatadas por outros desenvolvedores. Uma característica interessante é o trabalho simultâneo de vários desenvolvedores, resolvendo conflitos de versões, como por exemplo: um desenvolvedor atualizou um projeto de software (uma função de cadastro), ao mesmo tempo em outro lugar do globo outro desenvolvedor fez alterações e atualizou a versão que está armazenada no servidor, quando o primeiro usuário enviar também as suas alterações você pode receber mensagens de alerta que sua versão encontra-se desatualizada, outro recurso é mostrar trechos de códigos que foram incluídos, testado ou removidos.

Portanto, a utilização de controles de versões seja ela feita de forma manual ou por diferentes tipos de sistemas é de extrema importância para a revisão/inspeção de um software, pois é a partir destas ferramentas que os profissionais poderão trabalhar em colaboração com várias áreas e membros de times em diferentes lugares do mundo, podem ainda ter uma maior organização daquilo que já foi alterado ou testado, resolvendo problemas ou conflitos que passam despercebidos durante o processo de desenvolvimento do sistema.

Fechamento:

Este foi nosso podcast de hoje! Até a próxima!