Respostas atividade de Engenharia de Software IV
Aluno(a): Julio César Gonçalves de Carvalho Filho
Professor(a): Rennê Stephany

1º
Pode haver desperdício de esforço na equipe, com a modificação de uma versão errada de um sistema, perder o controle de quais mudanças e versões de componentes foram incorporadas em cada versão de sistema, pode haver várias versões em desenvolvimento e em uso ao mesmo tempo, aumento da chance de entregar ao cliente uma versão errada do sistema e esquecer onde está armazenado o código-fonte do software para uma versão específica do sistema ou componente.

2º
Formulários eletrônicos de solicitação de mudança registram informações compartilhadas entre todos os grupos envolvidos no gerenciamento de mudanças. À medida que é processada a solicitação de mudança, a informação é adicionada ao formulário a fim de que se registrem as decisões tomadas em cada estágio do processo. Assim como registrar a mudança requerida, o formulário registra as recomendações relativas à mudança, a estimativa de custos da mudança e as datas de quando a mudança foi solicitada, aprovada, implementada e validada.

3º
Manter o controle das propostas de alteração, armazenar versões de componentes do sistema, construir sistemas a partir desses componentes, controlar o lançamento de versões para os clientes, acompanhamento de bug e para dar suporte a processos de CM.

4º
É necessário que cada versão do componentes que foram realizados mudanças sejam identificadas, é preciso que haja uma garantia de que as mudanças que serão feitas por diferentes desenvolvedores não irão interferir em outras versões. Identificações ambíguas podem causar problema e confusão as diferentes versões e releases de um sistema.

5º
As alterações que um desenvolvedor realizou em um componente não consigam ser vistas pelos outros desenvolvedores até que ele submeta suas alterações ao servidor onde os arquivos do projeto ficam. Ter que testar sempre tudo em todos os componentes quando os três componentes fossem devidamente modificados e colocados sobre controle de configuração. Se os desenvolvedores precisam modificar o mesmo componente, a ferramenta de controle de versão vai “bloquear” para os outros desenvolvedores o seu uso.

6º
1 - Identificação de versão e release: Versões gerenciadas recebem identificadores quando são submetidas ao sistema.
2 - Gerenciamento de armazenamento. Os sistemas de gerenciamento de versões em geral fornecem recursos de gerenciamento de armazenamento. Em vez de manter uma cópia completa de cada versão, o sistema armazena uma lista de diferenças entre uma versão e outra.

3 - Registro de histórico de mudanças. Todas as mudanças feitas no código de um sistema ou componente são registradas e listadas. Em alguns sistemas, essas mudanças podem ser usadas para selecionar uma versão específica do sistema.
4 - Desenvolvimento independente. Desenvolvedores diferentes podem trabalhar no mesmo componente ao mesmo tempo. O sistema de gerenciamento de versões acompanha os componentes que tenham sido verificados para edição e garante que as mudanças feitas em um componentes por diferentes desenvolvedores não interfiram.
5 - Suporte a projetos. Um sistema de gerenciamento de versões pode apoiar o desenvolvimento de vários projetos, que compartilham componentes. É possível fazer check-in e check-out de todos os arquivos associados a um projeto, em vez de precisar trabalhar com um arquivo ou diretório de cada vez.

7º
Se o sistema for muito grande, pode levar muito tempo para ser construído e testado. É impraticável construir esse sistema várias vezes ao dia. Se a plataforma de desenvolvimento é diferente da plataforma-alvo, pode não ser possível executar testes de sistema no espaço de trabalho privado do desenvolvedor. Pode haver diferenças de hardware, sistema operacional ou software instalado. Portanto, é necessário mais tempo para testar o sistema.

8º
Por conta que o software foi realizado no hardware obsoleto, o hardware deve ser mantido para futuros problemas relacionados ao software, onde os desenvolvedores podem voltar e analisar os problemas que podem ter ocorrido. Com isso o software pode ser modificado para o funcionamento em hardwares mais atualizados.

9º
1. Realizar check-out do sistema de mainline do sistema de gerenciamento de versões no espaço de trabalho privado do desenvolvedor.
2. Construir o sistema e executar testes automatizados para assegurar que o sistema construído passe em todos os testes. Caso contrário, a construção é interrompida e você deve informar quem realizou o último check-in de sistema de baseline. Eles são responsáveis por reparar o problema.
3. Fazer as mudanças para os componentes de sistema.
4. Construir o sistema no espaço de trabalho privado e executar novamente os testes de sistema. Se os testes falharem, continue a editar.
5. Uma vez que o sistema tenha passado nos testes, verificar isso no sistema construído, mas não aprovar como uma nova baseline de sistema.
6. Construir o sistema no servidor de construção e executar os testes. Você precisará fazer isso para o caso de outras pessoas terem modificado os componentes depois que você tenha feito o check-out do sistema. Se esse for o caso, conferir os componentes que falharam e editá-los, de maneira que os testes passem em seu espaço de trabalho privado.
7. Se o sistema passar em seus testes sobre o sistema de construção, então, aprovar as mudanças feitas como uma nova baseline no mainline de sistema.

10º
1 - Qualidade técnica do sistema: Caso sejam relatados defeitos graves de sistema, que afetem a maneira como muitos clientes o usam, pode ser necessário emitir um release de

reparação de defeitos. Pequenos defeitos de sistema podem ser reparados mediante a emissão de patches (normalmente distribuídos pela Internet) que podem ser aplicados no release atual do sistema;
2 - Mudanças de plataforma: Talvez você precise criar um novo release de uma aplicação de software quando uma nova versão da plataforma do sistema operacional for lançada;
3 - Quinta lei de Lehman: Essa ‘lei’ sugere que se você adicionar nova funcionalidade a um sistema, você também introduz bugs que limitam a quantidade de funcionalidade que pode ser incluída no próximo release. Portanto, um release de sistema com funcionalidade nova e significativa pode ser seguido por um release que se concentra em reparar os problemas e melhorar o desempenho;
4 - Concorrência: Para software de mercado de massa, um novo release de sistema pode ser necessário porque um produto concorrente introduziu novos recursos e a fatia de mercado pode ser perdida caso estes não sejam fornecidos aos clientes existentes;
5 - Requisitos de marketing: O departamento de marketing de uma organização pode ter feito um compromisso para releases estarem disponíveis em uma determinada data.