**1) Na gestão de mudanças, quais são as características dos tipos de mudança abaixo?**

Mudança padrão é uma mudança que é pré-autorizada pelo Gerenciamento de Mudanças e que se tornou rotineira, já tendo um script de procedimento para execução. Por este motivo, geralmente o fluxo para execução desta alteração é mais ágil.

Mudança normal é uma mudança para a qual não existe um script já pronto, e precisa passar pelo fluxo mais extenso para ser autorizada e planejada antes de sua execução.

Mudança emergencial é aquela que precisa ser implementada rapidamente para resolver falhas (incidentes). Neste caso, nem sempre será possível realizar todos os testes. Este tipo de mudança é tratado pelo Comitê Consultivo de Mudanças Emergenciais.

**2. Quem são os participantes de um comitê consultivo de mudanças?**

É formado de acordo com as mudanças a serem discutidas. Exemplos de participantes que podem estar envolvidos: Fornecedores. Gerente de Problemas. Gerente de Nível de Serviço. Equipe de relações com o cliente.

**3. Dê 3 exemplos de regras usuais para que os desenvolvedores realizem mudanças nas funcionalidades de um software.**

Custos Benefícios Prioridades Componentes impactados Deve-se levar em conta as respostas aos 7 R's.

**4. Cite 4 exemplos de ferramentas de controle de versão de código-fonte.**

* RCS, CVS, Git, SubVersion

**5. O que é integração contínua?**

É uma prática de desenvolvimento de software onde membros de um time integram seu trabalho frequentemente.

**6. Quais são os benefícios da integração contínua?**

Muitas equipes acreditam que esta abordagem reduz problemas de integração e permite que o software seja desenvolvido de forma rápida e coesa.

**7. Cite 2 exemplos de ferramentas de software que podem dar apoio à gestão de mudanças.**

Bugzilla, Microsoft Team Foundation

**8. Cite 2 exemplos de ferramentas de software para a realização de integração contínua**.

Go, Travis CI

**9. No que consiste fazer o gerenciamento de releases?**

Consiste no empacotamento de um sistema para promovê-lo de desenvolvimento para QA (Quality Assurance) e, posteriormente, para produção.

**10. Cite 3 princípios do gerenciamento de releases.**

* Releases devem ser identificados por um identificador (ID) de versão imutável.
* Releases devem ser empacotados com todas as suas dependências.
* O gerenciamento de releases deve ser rápido e confiável para facilitar o desenvolvimento iterativo.

**11. Explique como é feita a identificação de um release utilizando o versionamento semântico, e qual é o significado de cada parte desta identificação.**

Uma versão é identificada por uma sequência de 3 números: MAJOR.MINOR.PATCH O identificador MAJOR deve ser alterado somente quando as alterações tornam a API incompatível com versões anteriores. O identificador MINOR deve ser alterado quando as alterações são realizadas para adicionar funcionalidade, mantendo a compatibilidade com versões anteriores. O identificador PATCH deve ser alterado quando as alterações são realizadas para corrigir bugs, mantendo a compatibilidade com versões anteriores.

**12. Dê um exemplo de identificação de um release utilizando o versionamento semântico.**

Exemplo: 1.2.14

**13. Cite um benefício obtido com a utilização de uma ferramenta como o Codenvy.**

Permite que o desenvolvedor tenha seu workspace disponível em qualquer computador que o mesmo vá utilizar remotamente.

**14. Cite um benefício obtido com a utilização de uma ferramenta como o Heroku.**

Permite a hospedagem de uma aplicação em diferentes tecnologias.