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

Mudança padrão: Pré-autorizada pelo Gerenciamento de Mudança, script de procedimento para execução e fluxo de execução mais ágil.

Mudança normal: Por não ter um script de procedimento é preciso passar pelo fluxo mais extenso para ser autorizada e planejada antes de sua execução.

Mudança emergencial: Precisa ser implementada rapidamente para resolver falhas (incidentes). Este tipo de mudança é tratado pelo Comitê Consultivo de Mudanças Emergenciais.

2) Quem são os participantes de um comitê consultivo de mudanças?

R: Fornecedores, gerente de problemas gerente de nível de serviço, equipe de relações com o cliente e etc.

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

R: Não permitir fazer comits, alterações fazer apenas no branch e

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

R: CVS, Mercurial, Git e SVN.

5) O que é integração contínua?

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

6) Quais são os benefícios da integração contínua?

R: 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.

R: Jira e Trac.

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

R: Bamboo e Travis CI.

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

R: 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*.

R: 1- *Releases* devem ser identificados por um identificador (ID) de versão imutável.

2- *Releases* devem ser empacotados com todas as suas dependências.

3- O empacotamento de *releases* deve ser automatizado e desenhado para evitar erros humanos.

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

R: Uma versão é identificada por uma sequência de 3 números:

MAJOR.MINOR.PATCH

Exemplo: 1.2.14

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

R: O identificador MAJOR deve ser alterado somente quando as alterações tornam a API incompatível com versões anteriores.

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

R: Por ser um IDE de nuvem e servidor de espaço de trabalho do desenvolvedor permite que qualquer pessoa contribua para um projeto sem instalar o software.

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

R: O Heroku automatiza a criação de uma nova máquina virtual volátil