

1. Justificando sua resposta com base no tipo de sistema a ser desenvolvido, sugira o modelo genérico de processo de software mais adequado para ser usado como base para a gerência do desenvolvimento dos sistemas a seguir:

a) Um sistema para controlar o antibloqueio de frenagem de um carro.

R: Modelo em Cascata, pois é um sistema do tipo A (Acurate), com escopo bem definido, ou seja, sem mudanças de requisitos; e necessidade de documentação suficientemente extensa, já que pode custar vidas humanas.

b) Um sistema de realidade virtual para dar apoio à manutenção de software.

R: Modelo Ágil, pois é um sistema disruptivo, ou seja, seu escopo está constantemente sujeito a mudanças de requisitos.

c) Um sistema de contabilidade para uma universidade, que substitua um sistema já existente.

R: Modelo em Cascata, pois o escopo do sistema já é bem definido pelo sistema já existente, o que descarta mudanças de requisitos durante o processo de desenvolvimento.

d) Um sistema interativo de planejamento de viagens que ajude os usuários a planejar viagens com menor impacto ambiental.

R: Modelo Ágil, pois o sistema é uma ideia nova e, conseqüentemente, precisa de um desenvolvimento incremental e iterativo para se adaptar e consolidar.

2. Sugira por que é importante, no processo de engenharia de requisitos, fazer uma distinção entre desenvolvimento dos requisitos do usuário e desenvolvimento de requisitos de sistema.

R: Essa distinção é importante pois os requisitos de usuário são essenciais para o usuário e o funcionamento do sistema, enquanto os requisitos de sistema são importantes para o bom funcionamento do sistema em termos de desempenho.

3. Explique por que, para as empresas, a entrega rápida e implantação de novos sistemas frequentemente é mais importante do que a funcionalidade detalhada desses sistemas.

R: Essa entrega rápida é chamada de MVP, é o Produto Mínimo Viável, que consiste em entregar, inicialmente, as funcionalidades essenciais para o mínimo

funcionamento do sistema. O MVP é importante para a inserção do sistema no mercado e suprir determinada demanda, o que, por sua vez, é importante para a medição do sucesso do produto e para o levantamento de melhorias e mudanças de requisitos.

4. Tem-se sugerido que um dos problemas de se ter um usuário participando de uma equipe de desenvolvimento de software é que eles 'se tornam nativos', ou seja, adotam a perspectiva da equipe de desenvolvimento e perdem de vista as necessidades de seus colegas usuários. Sugira três maneiras de evitar esse problema e discuta as vantagens e desvantagens de cada abordagem.

R: Levando em consideração que a nomenclatura desse usuário é Product Owner:

1 - Evitar o uso de termos técnicos nas reuniões com o PO.

Vantagens: O PO não vai se acostumar com o dialeto técnico, logo, não vai entendê-lo e, conseqüentemente, não vai adotar a perspectiva da equipe.

Desvantagens: Os demais membros da equipe de desenvolvimento perdem um pouco da liberdade de comunicação, o que pode, ainda, dificultar o repasse e o entendimento de informações, uma vez que os integrantes devem “adaptar” suas falas à linguagem não técnica do PO.

2 - Rotacionar os POs.

Vantagens: Um PO não ficará na equipe por tempo o suficiente para entender a linguagem técnica e adotar outra perspectiva.

Desvantagens: Dependendo da extensão do projeto e do número de profissionais disponíveis para assumir o papel de PO, em algum momento, os POs se repetiriam. Além disso, a troca constante de PO pode ocasionar dificuldades de comunicação.

3 - Limitar as interações entre o PO e a equipe de desenvolvimento.

Vantagens: O PO não terá tempo o suficiente de “exposição” ao dialeto técnico para compreendê-lo e mudar de postura.

Desvantagens: Possivelmente, o PO não será capaz de repassar todas as necessidades do usuário e a equipe de desenvolvimento não será capaz de compreendê-las, dado o tempo limitado.

5. Procure pensar em um sistema de uma área da qual tenha algum conhecimento. Pede-se:

a) Escreva então duas histórias para esse sistema (veja que histórias são especificações resumidas de funcionalidades, com 2-3 sentenças).

R: Considerando uma rede social qualquer:

1 - Eu, como usuário comum, desejo me cadastrar, para que possa utilizar a plataforma.

2 - Eu, como usuário comum, desejo seguir outro usuário, para que possa acompanhar seu perfil.

b) Em seguida, quebre a história que definiu em algumas tarefas.

R:

1 - Cadastro:

- a. Prototipar a tela de cadastro.
- b. Desenvolver o front-end da tela prototipada.
- c. Implementar o back-end e o banco de dados por trás do cadastro.
- d. Integrar o front-end ao back-end e banco de dados.

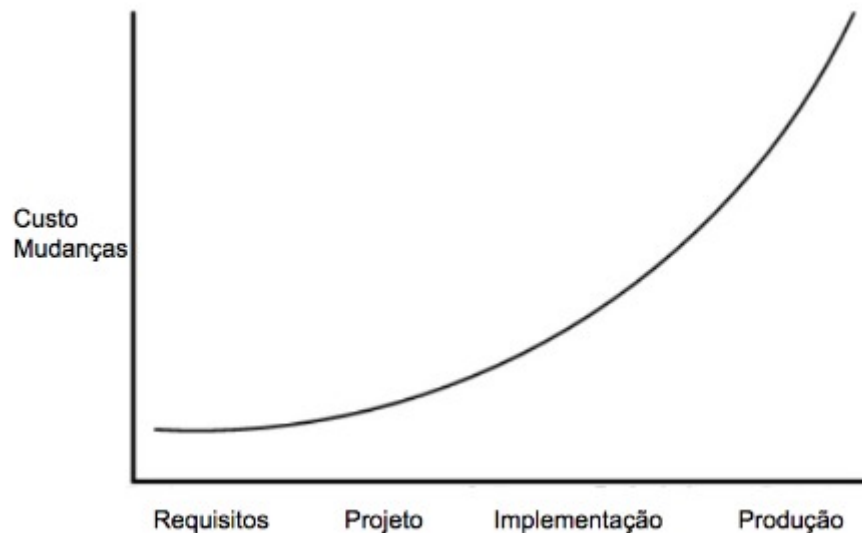
2 - Seguir, considerando que a tela de perfil de usuário é outra história:

- a. Implementar o back-end e o banco de dados por trás da ação de seguir.
- d. Integrar o back-end e o banco de dados ao front-end da tela de usuário.

c) Existem dependências entre essas tarefas? Ou elas podem ser implementadas em qualquer ordem?

R: Existem certas dependências nas duas histórias. Na história 1, a tarefa b deve ser implementada depois da tarefa a, enquanto a tarefa d deve ser implementada depois das tarefas b e c. Na história 2, a tarefa b deve ser implementada depois da tarefa a.

6. Seja o seguinte gráfico, que mostra – para um determinado sistema – como os custos de mudanças variam conforme a fase do desenvolvimento em que elas são realizadas.



a) Qual método de desenvolvimento você recomendaria para esse sistema? Justifique sua resposta.

R: Método em Cascata, uma vez que o custo de mudanças é pequeno na fase de elicitação de requisitos e cresce exponencialmente a cada fase. Esse método é o mais indicado, nesse caso, pois não permite mudanças de requisitos no decorrer do desenvolvimento do sistema, evitando grandes custos.

b) Sistemas de quais domínios podem ter uma curva de custos de mudanças semelhante a essa?

R: Sistemas críticos, geralmente do tipo A (Acurate), de domínios como o automobilismo, a exploração espacial e a cirurgia robótica.