

Introdução a Processo de Software

Extreme Programing

Atividades de Aprendizagem e Avaliação

Aluno: LUCAS GABRIEL SCHUMANN GARCIA RA: 2206293

Utilize esta formatação em seu texto

- 1) Considerando o texto no link “Texto - Processo de Software – cap 2”, complete
 - a) Todo e qualquer Software é produzido de acordo com um **processo**.
 - b) Um Processo de Desenvolvimento de Software define **um conjunto de passos, tarefas, eventos e práticas que devem ser seguidos por desenvolvedores de software, na produção de um sistema**.
 - c) O **Manifesto Ágil** foi lançado em **meados da década de 1970**.
 - d) Porque os documentos de requisitos ficam obsoletos?

R: **Porque foram substituídos por histórias de usuário. Que se aproximam mais daquilo que o cliente necessitará no software. São documentos simples, que focam nas funcionalidades do sistema, sempre na visão de seus usuários. Como exemplo, mostramos a seguir uma história de um sistema de perguntas e respostas — semelhante ao famoso Stack Overflow**
 - e) A principal característica de um processo ágil é **a adoção de ciclos curtos e iterativos de desenvolvimento, por meio dos quais um sistema é implementado de forma gradativa; começando por aquilo que é mais urgente para o cliente**
 - f) Em processos ágeis, as práticas de desenvolvimento incluem **programação em pares, testes automatizados e integração contínua**.
 - g) O Extreme Programing (XP) por Kent Beck surgiu em **1999**, o Scrum por Jeffrey Sutherkand e Ken Schwaber foi proposto em **1995**, Já o Kanban remonta às práticas nas fábricas da Toyota na década de **50**.
 - h) O XP é definido por meio de **um conjunto de valores, princípios e práticas de desenvolvimento**.
 - i) Segundo os princípios do XP, software não é uma obra de arte, mas **mas algo que tem que gerar resultados econômicos, como defendido por esse princípio de XP**.

- j) Ao escrever testes, um desenvolvedor se beneficia, pois [pois eles ajudam a detectar bugs no seu código](#), mas testes também ajudam outros desenvolvedores que futuramente [erão mais segurança de que o seu código não vai introduzir regressões](#).
- k) *Refactoring* é uma atividade que torna o código [mais limpo e fácil de entender](#), tanto [para quem o escreveu, como para quem futuramente terá que mantê-lo](#).
- l) A integração contínua prega que é melhor integrar diariamente, do que [passar pelo stress de fazer uma grande integração após semanas de trabalho](#)
- m) Histórias de usuário são documentos resumidos, com [com apenas duas ou três sentenças](#) onde o representante do cliente define [o que ele deseja que o sistema faça](#), usando sua própria [linguagem](#).
- n) Histórias são documentos simples, que focam [nas funcionalidades do sistema](#), sempre na [visão do usuário](#).
- o) *Planning Poker* é uma técnica para [estimar o tamanho de histórias](#).
- p) No XP, o horizonte do planejamento é uma [release](#), isto é, [alguns meses](#).
- q) *Planning Game* é a tarefa de [alocar histórias a iterações e releases](#).
- r) O planejamento da iteração tem por objetivo [decompor as histórias de uma iteração em tarefas](#), as quais devem corresponder a atividades de [programação que possam ser alocadas para um dos desenvolvedores](#) do time
- s) Histórias estão associadas a [requisitos funcionais](#), entretanto uma história pode disparar tarefas de [requisitos não-funcionais](#), tais como instalar um banco de dados.
- t) XP dá grande importância a tarefas de [desenvolvimento](#), e isso inclui a [programação em pares](#), testes [automatizados](#) e integração [contínua](#).
- u) A programação em pares preconiza que toda tarefa de [desenvolvimento](#) deve ser realizada [por dois desenvolvedores](#) trabalhando junto, compartilhando [o mesmo teclado e monitor](#).
- v) Programação em pares contribui para disseminar [conhecimento](#) e pode ser empregada para treinar [desenvolvedores iniciantes](#).
- w) Nos dias atuais, a programação em pares tem sido substituída pela prática de [revisão de código](#), onde todo código [produzido por um desenvolvedor](#) tem que ser revisado [por outro desenvolvedor](#), porém de forma [offline e assíncrona](#).
- x) Integração Contínua é a prática de [armazenar, em um sistema de controle de versão, o código produzido, por mínimo que seja](#).
- y) Nos contratos de escopo fechado, a contratante [define os requisitos do sistema](#) e a contratada define [um preço e um prazo de entrega](#).
- z) Nos contratos de escopo aberto, o pagamento é por [hora trabalhada](#).