

# CONCURSO PÚBLICO

## TRIBUNAL REGIONAL ELEITORAL DO TOCANTINS

### CARGO 5: TÉCNICO JUDICIÁRIO – ÁREA: APOIO ESPECIALIZADO ESPECIALIDADE: PROGRAMAÇÃO DE SISTEMAS

#### PROVA DISCURSIVA

Aplicação: 3/12/2017

## PADRÃO DE RESPOSTA DEFINITIVO

### 1 *Product backlog* e sua relação com o *product owner*

O *backlog* do produto é **um artefato** do Scrum. É uma **lista ordenada de tudo** que deve ser necessário no produto, e é uma **origem única dos requisitos** para qualquer mudança a ser feita no produto. O *product owner* é **responsável pelo backlog do produto**, incluindo seu **conteúdo, disponibilidade e ordenação**. O *backlog* do produto lista todas as características, funções, requisitos, melhorias e correções que formam as mudanças que devem ser feitas no produto nas futuras versões.

Ken Schwaber e Jeff Sutherland. *Guia do Scrum* — Um guia definitivo para o Scrum: As regras do jogo. Jul./2016, p. 13 (com adaptações).

### 2 Reunião diária do Scrum e perguntas que devem ser respondidas nessa reunião

É **um evento do Scrum, time-boxed de quinze minutos**, para que o time de desenvolvimento possa sincronizar as atividades e criar um plano para as próximas vinte e quatro horas. Essa reunião é **feita para inspecionar o trabalho** desde a última reunião, e durante a reunião os membros do time de desenvolvimento esclarecem:

**O que eu fiz ontem** que ajudou o time de desenvolvimento a atender a meta da *Sprint*?

**O que eu farei hoje** para ajudar o time de desenvolvimento a atender a meta da *Sprint*?

Eu vejo **algum obstáculo que impeça** a mim ou ao time de desenvolvimento no atendimento reunião diária?

Ken Schwaber e Jeff Sutherland. *Guia do Scrum* — Um guia definitivo para o Scrum: As regras do jogo. Jul./2016, p. 11 (com adaptações).

### 3 Objetivos e características das *users stories*

As histórias de usuário na XP têm o **mesmo propósito que os casos de uso** e são usadas como **entrada para o jogo do planejamento**.

**São escritas pelos clientes** como coisas que o sistema precisa fazer por eles. Deve-se tentar manter as histórias focadas nas **necessidades e benefícios dos usuários** em oposição à especificação de *layouts* de GUI.

Ian Sommerville. *Engenharia de software*. 9.ª ed. Pearson, 2011, p. 44-6. Internet: <<http://www.extremeprogramming.org>> (com adaptações).

### 4 Objetivos da integração contínua

Prática da XP. Os desenvolvedores devem **integrar e efetuar commit do código sempre que possível** para manter o sistema integrado o tempo todo. A integração contínua **evita esforços de desenvolvimento divergentes** ou fragmentados, em que os desenvolvedores não se comunicam entre eles sobre o que pode ser reutilizado ou o que pode ser compartilhado.

Ian Sommerville. *Engenharia de software*. 9.ª ed. Pearson, 2011, p. 44-6. Internet: <<http://www.extremeprogramming.org>> (com adaptações).

### 5 TDD (*Test Driven Development*) e *test-first*

O TDD segue a **prática da XP do desenvolvimento test-first**: em vez de escrever algum código e, em seguida, escrever testes para esse código, **escrevem-se os testes antes do código**.

Ian Sommerville. *Engenharia de software*. 9.ª ed. Pearson, 2011, p. 44-6. Internet: <<http://www.extremeprogramming.org>> (com adaptações).

#### Ciclo do TDD

Escrever um teste onde a aplicação falhe.

Escrever a aplicação que passe nos testes.

Refatorar o código (melhorá-lo) e executar os testes novamente e garantir que eles continuem passando.

Repetir o processo até o término do projeto.

Kevin Harvey. *Test-Driven Development with Django* Packt Publishing Ltd. 30/7/2015, p. 1.

Bala Paranj. *Test Driven Development in Ruby: A Practical Introduction to TDD* — Using problem and solution domain analysis apress. Mar./2017, p. 13.