UNIVERSIDADE FEDERAL DE GOIÁS
Instituto de Informática
Engenharia de Software

Fernando Henrique Callata
Gustavo Batista
Lucas Sampaio
Pedro Basílio
Saulo Calixto

IMPLEMENTAÇÃO

Goiânia, 30 de outubro de 2017.

Introdução

O processo de implementação se dará em interações de sprints. Em cada sprint será definido o backlog, ou escopo da sprint, que será o conjunto de funcionalidades que serão desenvolvidas. As sprints não deverão ter tempo maior do que uma semana e as atividades serão "*rankeadas*" por relevância.

Dentro de cada sprint além do desenvolvimento de funcionalidades também devem ser previstos nela testes de unidade e testes exploratórios para as mesmas funcionalidades desenvolvidas e também é imprescindível que haja revisão de código. Nesse caso o revisor deve ser um outro desenvolvedor que poderá ter uma visão imparcial do código implementado.

Também trabalhamos com o princípio da integração contínua utilizando a ferramenta *TeamCity,* com isso várias atividades manuais são automatizadas, como rodar os testes de unidade implementados, erros de estilo no código, etc. Dessa forma o desenvolvedor poderá focar apenas em implementar o software com qualidade.

Em todas as interações devem ser entregues artefatos funcionais do sistema desenvolvido, claro, em pequenas partes. Até que todo o backlog seja atendido.

Dessa forma, dentro de uma sprint há fases que precisam ser seguidas para que haja sucesso no que é proposto, que é a entrega de um artefato útil.

1ª Fase: Sprint Planning

Nessa fase é definido o escopo da sprint. Nesse caso temos um product backlog que são todos os requisitos que teremos que implementar. Esse artefato foi definido pelo nosso product owner.

Desse artefato o analista de negócio juntamente com o scrum master que nesse caso é o gerente de processo irá definir os requisitos mais pertinentes a serem desenvolvidos na sprint. Cada requisito irá ganhar um nível de importância.

A sprint deve ser planejada para não durar mais que uma semana, então backlog da sprint deve ser construído levando em conta a maturidade da equipe de desenvolvimento para que ela não fique ociosa nem sobrecarregada.

Essa fase não deve durar mais do que um dia, pois entre uma sprint e outra não se pode passar mais do que um dia.

2ª Fase: Reunião de alinhamento com scrum team.

Uma vez que a sprint backlog é definida é feita uma reunião com a equipe de desenvolvimento para alinhar as atividades que serão realizadas. Geralmente o analista de negócio participa da reunião afim de de esclarecer eventuais dúvidas sobre os requisitos. É nessa fase em que todas as dúvidas a respeito dos requisitos são sanadas.

Além do analista de negócio e Scrum Master também participam da reunião os analistas de desenvolvimento e analistas de qualidade que serão responsáveis pelo desenvolvimento e testes, de acordo com suas competências.

São então atribuídas as atividades para cada membro da equipe, além disso as atividades que possuem peso maior são priorizadas em detrimento às de nível mais baixo. É então previsto horas de desenvolvimento para cada uma das atividades, prevendo neste prazo o tempo para implementação e execução dos testes.

3ª Fase: Codificação

Nessa fase os analistas de desenvolvimento começam a implementar as atividades previstas no sprint backlog. O desenvolvimento é feito seguindo os princípios da *entrega contínua*, ou seja, toda a parte de build, execução de testes, verificação de estilo, boas práticas de codificação é feita pela ferramenta TeamCity.

Para que haja uma integração com toda a equipe é importante que o desenvolvedor se preocupe sempre em dar *commit* nas alterações que ele fez, para que os outros membros da equipe tenham acesso o mais rápido possível às alterações.

A cada *commit* feito o código será compilado pelo TeamCity, além disso este software verificará se há erros de estilo no código ou se o desenvolvedor saiu do padrão de codificação definido pela empresa. O software também analisará se há código duplicado e se algum teste de unidade foi quebrado com a nova atualização.

Se algum desses pontos não forem aprovados, o TeamCity lançará um erro, então o desenvolvedor terá que analisar o que deu errado, consertar o código e *commitar* novamente.

Os *commits* são feitos por um sistema de controle de versão chamado *GIT*, os repositórios serão guardados em uma conta privada do *GITHUB*. Também é importante salientar que trabalhamos com *branches.* Durante o período da *branch* todos os commits são feitos nela, após três meses e ela estar estável é feito o *merge* com a *branch* ***master.***

Cabe a cada integrante da equipe cuidar dos seus *commits* e também de atualizar o repositório local sempre, afim de estar sempre com a versão mais atualizada em mãos para trabalhar. Com essa prática são evitados conflitos que geram *merge* (que é quando duas pessoas alteram o mesmo documento) e consequentemente desperdiça o tempo dos desenvolvedores.

É papel do desenvolvedor escrever seu código dentro dos parâmetros definidos pelo Arquiteto de Software. Nesse caso ele deve cuidar para nunca quebrar a arquitetura, tirando dúvidas quando necessário.

4ª Fase: Testes

A fase de testes não acontece necessariamente após a fase de codificação. Durante a codificação das novas funcionalidades é necessário que o desenvolvedor também desenvolva os testes de unidades relativos àquela funcionalidade. Caso ele esteja apenas atualizando alguma funcionalidade ele deve se preocupar com os testes existentes, para o caso de quebra. Contudo, vale lembrar que o TeamCity fará esse papel de executar os testes para evitar que haja essas quebras.

Além dos testes de unidades também são feitos testes exploratórios. Esses testes são feitos pelos analistas de qualidade. A função desses testes, diferente do teste de unidade que cumpre a função de testar uma parte do código, é de testar a aplicação em si. A função do analista de teste é simular o uso que o cliente fará ao utilizar o sistema, explorando todos os cenários possíveis a fim de encontrar erros.

Os erros encontrados durante os testes de exploração, se forem advindos de funcionalidades desenvolvidas na presente sprint, são repassadas para os

desenvolvedores para correção. Caso contrário, eles serão incluídos na próxima sprint para correção.

5ª Fase: Revisão de Código

A fase de revisão é feita pelos próprios desenvolvedores. Após uma funcionalidade ter sido implementada outro desenvolvedor é imcubido de revisar o código de seu colega.

Essa revisão visa manter a qualidade do código. Muitas vezes algum erro de lógica ocorre e nem o desenvolvedor e nem os testes identificam, contudo outro desenvolvedor com uma visão imparcial pode ser capaz de perceber.

Geralmente a revisão de código é feita por um desenvolvedor mais experiente, ou pelo menos de mesmo nível. Nessa fase todo código implementado é revisado, inclusive o código concernente aos testes de unidade.

O revisor que identifica algum erro no código de seu colega não tem a autonomia de corrigi-lo. Ele deve notificar o responsável pelo código para que ele mesmo faça a correção. Assim garantimos o crescimento desse membro da equipe.

6ª Fase: Entrega do artefato

Após planejar, codificar, revisar e testar, quando chegamos ao final da sprint todo o sprint backlog já deve ter sido atendido. Nessa fase é feita a entrega do artefato ao o cliente.

Esse artefato deve ser uma parte do sistema desenvolvido totalmente funcional, o cliente deve ser capaz de avaliar o que foi feito na sprint e assim poder estar mais seguro a respeito do que tem sido desenvolvido.

Não é necessariamente nessa fase que é feito o *merge* com o branch principal, pois isso deve ser feito ao final da entrega de determinada versão.

Nessa fase que acontece também a validação por parte do product owner do artefato entregue. Caso ele perceba que alguma coisa não está de acordo com o que foi solicitado as mudanças necessárias entraram em sprints futuras.

7ª Fase: Sprint Review

Após tudo estar concluído é feito a sprint review. É aqui que a equipe se reúne para discutir as atividades desenvolvidas na sprint. O que deu certo, o que não deu. Nessa reunião é possível que o processo de desenvolvimento sofra alguma modificação, caso seja verificado que algo não está ocorrendo tão bem como deveria.

Todas essas decisões são tomadas de acordo com o desempenho da equipe, o fim dessa reunião é sempre melhorar o processo de desenvolvimento para a próxima sprint.