

QUALIDADE DE SOFTWARE COM SCRUM

Rafael de Figueiredo Grzebieluka *

RESUMO

Este artigo aborda o Scrum alinhado com a Qualidade de Software, são várias as melhorias no processo de qualidade de equipes de desenvolvimento de software que utilizam o modelo ágil Scrum. As características que definem a qualidade de um software se relacionam com as boas práticas de engenharia de software, o Scrum como framework para gerenciamento de projetos é capaz de oferecer melhora na qualidade de software, com a finalidade de entregar ao cliente o seu produto de acordo com as suas necessidades. A equipe atua com mais confiança e segurança quando for realizada uma grande mudança durante o ciclo do projeto, na ocorrência de problemas eles são contornados com mais facilidade utilizando o Scrum e juntamente com a qualidade de software devemos entender o padrão de qualidade que é garantir um produto que satisfaça às expectativas dentro do que foi acordado inicialmente com o cliente, onde deve ser medido de ponta-a-ponta trabalhando para que o produto fique cada vez melhor.

PALAVRAS-CHAVE:: Scrum. Qualidade de Software. Testes. Equipe.

* Pós-graduando no MBA em Gestão de Tecnologia da Informação - Faculdade Santana. Graduado em Sistemas para Internet – Faculdade União.
E-mail: rafaelfigueiredopg@gmail.com

1. INTRODUÇÃO

A cada dia o software torna-se mais presente na vida das pessoas, podemos encontrar em caminhões, barcos, agências bancárias, escolas, terminais rodoviários e em outros lugares. Todos estes bens e serviços utilizam de um software para auxiliar no controle de suas operações e informações.

A produção de softwares tem aumentado no decorrer dos anos devido uma maior demanda pelas empresas, a garantia de qualidade juntamente com um método de gerenciamento de projetos tem um papel fundamental durante o ciclo do projeto. Este artigo apresenta uma visão geral das atividades de testes e a sua relação com a qualidade do produto de software utilizando o método ágil de gerenciamento de projetos Scrum.

2. Conhecendo o Scrum

O Scrum como sabemos é um método ágil de gerenciamento e desenvolvimento de software focado na entrega de recursos que agregam valor nos negócios em ciclos pequenos de desenvolvimento de duas a quatro semanas no máximo. Os times têm duas características interessantes e bem definidas, são multifuncionais buscam conhecer a habilidade que for necessária para realizar o trabalho e também são auto organizáveis a própria equipe descobre a melhor forma de realizar o trabalho, tem como a transparência, integridade e a entrega de valores.

De acordo com Schwaber (2009, p.04) os times Scrum são projetados para otimizar flexibilidade e produtividade. Para esse fim, eles são auto organizáveis, interdisciplinares e trabalham em iterações. Para ele:

[...] cada Time Scrum possui três papéis: 1) o ScrumMaster, que é responsável por garantir que o processo seja entendido e seguido; 2) o Product Owner, que é responsável por maximizar o valor do trabalho que o Time Scrum faz; e 3) o Time, que executa o trabalho propriamente dito. O Time consiste em desenvolvedores com todas as habilidades necessárias para transformar os requisitos do Product Owner em um pedaço potencialmente entregável do produto ao final da Sprint.

Nos dias atuais muitas empresas utilizam o Scrum para melhor aproveitamento no processo de desenvolvimento de seus produtos, entre elas são Google, Globo, HP, Yahoo, Microsoft entre outras.

Segundo Freire (2013, p.03) se uma empresa não possui processos definidos, o Scrum é uma das melhores opções, pois ele é simples de entender e bem mais simples de implantar que outros processos.

[...] assim, as chances de sucesso são muito maiores. Ken Schwaber fala que Scrum não é um processo, mas sim um framework. O mais importante é que ele traz uma caixa de ferramentas de boas práticas de trabalho, permitindo obter bons resultados, mesmo que a equipe não tenha o domínio completo das técnicas que estão sendo utilizadas.

O Scrum foi criado por Jeff Sutherland no ano de 1993 é uma analogia onde foi realizado um estudo em 1986 por Takeuchi e Nonaka, eles fizeram a comparação entre equipes de alto desempenho e multifuncionais com a formação Scrum existente nas equipes de Rugby. Mas somente em 1995 seu processo foi formalizado por Ken Schwaber no seu primeiro artigo que foi publicado em uma conferência no mesmo ano.

Sua utilização tem crescido no decorrer dos anos devido ao índice de sucesso ser satisfatório pelas empresas pois mostra que o seu uso diminui o atraso em projetos em relação a outros métodos, entre outros benefícios que o Scrum proporciona.

Quais são as vantagens de utilizar o Scrum alinhado com a qualidade de software? Terei bons resultados aplicando um método ágil no ciclo de desenvolvimento de software?

A Quality Assurance (QA) é a garantia de qualidade, ao realizar testes de software é realizado todo um acompanhamento para que no final do processo seja comprovado que o produto foi passado por uma intensa bateria de testes e com isso a qualidade do produto seja alcançada, abordaremos a forma de como o QA deve realizar os testes e qual a sua importância em um time Scrum.

3. Estimar as histórias

Na fase de estimar as histórias o analista de qualidade vai analisar com base nos requisitos do usuário, a partir disso ele irá identificar e documentar os cenários de casos de testes e montar todo o trajeto feliz da história do usuário.

A visão do controle de qualidade referente a criação dos cenários de testes vai muito além do que se pode imaginar “ estes planos de teste e atividades de planejamento constituem contribuição útil para desenvolvedores realçando potenciais debilidades (como descuidos ou contradições, e omissões ou ambiguidades na documentação) ” (IEEE, 2004, p.71).

Fazer estimativas não é uma tarefa fácil dependendo da complexidade ou tamanho do item a ser testado pode se levar horas para descrever a história, nesta fase do processo é importante ter a colaboração do time todo. Uma descrição bem relatada é aquela que o testador entende facilmente, pois tem uma linguagem clara, não tem muitos detalhes é direta afim de chegar a um propósito, ele aborda os pontos a qual o testador deve se atentar e com isso o resultado seja o esperado.

4. Elaborar os casos de testes

Nesses projetos geralmente antes de elaborar os casos de testes é realizada a análise de requisitos uma solução proposta que tem por objetivos detalhar os procedimentos que serão executados e sua aprovação tem fundamental importância no processo de desenvolvimento.

Em seguida é feita a descrição dos casos de testes com base na solução proposta para auxiliar o testador na execução dos testes e com isso ele verificar se as ações realizadas e o resultado obtido é conforme está determinado no caso de teste.

No time Scrum os envolvidos no projeto seja os analistas de negócios, desenvolvedores e analistas de QA trabalham juntos, não existe um time à parte são multifuncionais. Isso influencia positivamente na elaboração dos casos de testes pois cada pessoa sabe qual é o seu devido papel no projeto, com isso as

informações chegam mais rapidamente ao analista de QA para descrever os casos de testes.

É importante ressaltar que “teste de software agora é visto como uma atividade que deve abranger todo o processo de desenvolvimento e manutenção e ele é parte importante da construção atual do produto” (IEEE, 2004, p.71).

Os analistas de QA sempre estarão interagindo com o Product Owner (PO) dono do produto fazendo questionamentos e criticando as premissas do projeto e podem auxiliá-lo a escrever os casos de teste de aceite, a comunicação entre ambos é essencial para que não ocorra problemas durante o desenvolvimento e também após a entrega do produto.

Diante disso “teste de software consiste da verificação dinâmica do comportamento de um programa em um conjunto finito de casos de teste, adequadamente selecionados de um domínio de execução usualmente infinito, contra o comportamento esperado” (IEEE, 2004, p.70).

Os casos de testes não seguem um padrão a ser definido o próprio testador pode criar o seu caso de teste exploratório para auxiliar em seu teste, ou até mesmo automatizar utilizando-se de programas e outros meios que possam lhe ajudar a encontrar o máximo de situações possíveis que podem ocorrer no item a ser testado, com isso será mais fácil identificar um erro que talvez possa passar despercebido.

5. Manter visíveis as metas e objetivos

No Scrum a agilidade é essencial e como o período de testes e correções ocorre simultaneamente com o desenvolvimento, a comunicação entre os testadores e desenvolvedores deve ser rápida, para que o desenvolvimento realize a correção a tempo e não ocorra o atraso na entrega do produto.

Mantendo a funcionalidade do programa em alta, fazendo o possível para diminuir esse ciclo de testes e ajustes. Para manter a qualidade do produto em alta é importante prestar atenção nos requisitos do usuário, planejar os testes, executar os testes e definir o que é o “pronto”.

Segundo Schwaber (2009, p.21) no desenvolvimento de produtos, afirmar que a funcionalidade está pronta pode levar alguém a presumir que ela está pelo menos bem codificada, refatorada, que tenha passado por testes unitários, e que tenha passado por testes de aceitação.

[...] outros podem presumir que apenas o código tenha sido desenvolvido. Se ninguém sabe qual a definição de “pronto”, os outros dois pilares do controle de processos empíricos não funcionam. Quando alguém descreve algo como “pronto”, todos devem entender o que “pronto” significa.

O analista de QA tem o dever de manter o time todo focado nos testes, assim como o SM de fazer com que todos se sintam motivados durante o percurso do projeto, ambos devem fazer com que o objetivo e as metas estejam visíveis para todos.

Para ajudar o time a se interessar pelos testes o analista de QA pode criar algumas formas divertidas de testes, um ciclo onde todos trabalhem em equilíbrio e felizes para obter um teste bem feito e em seguida verificar se a solução atende aos objetivos do negócio e a seus requisitos, no que diz respeito à funcionalidade e usabilidade, com isso o resultado será satisfatório para todos.

6. Fornecer feedback rápido a equipe

Durante o projeto o analista de QA e o SM devem manter a equipe informada de quaisquer mudanças durante o processo, ter um feedback rápido para que não ocorra problemas no ciclo de desenvolvimento do produto, evitando que o time realize os testes e correções por diversas vezes para que não tenha desperdício de tempo e trabalho adicional.

A comunicação entre todos é essencial principalmente entre o analista de QA e os desenvolvedores onde podem sempre consultá-lo para saber qual é o resultado esperado, o critério de aceite e se o comportamento da funcionalidade está de acordo como planejado.

Conforme IEEE (2014, p.150) avalie o processo de mensuração/medição. Avalie o processo de medição contra os critérios de avaliação especificados e determine os pontos fortes e fracos do processo. Isto pode ser realizado por um processo interno ou por uma auditoria externa e deve incluir o feedback (retorno) dos usuários das medições.

[...] identifique melhorias potenciais. Tais melhorias podem ser mudanças nas formas dos indicadores, mudanças nas unidades de medida ou reclassificação das categorias. Determine os custos e benefícios das melhorias potenciais e selecione as ações de melhoria adequadas. Comunique as proposições de melhoria aos proprietários do processo de medição e aos stakeholders para revisão e aprovação. Também comunique a falta de melhorias potenciais caso a análise falhe ao identificar as melhorias.

O feedback tem um papel importante para o time pois ajuda a saber lidar com as mudanças de requisitos, fazer uma retrospectiva para verificar como foi o comportamento do projeto através de uma avaliação sobre o que aconteceu, é interessante pois o SM tem uma visão clara do que deve melhorar e quais foram as lições aprendidas, desta forma sabemos o que ganhamos e estamos trabalhando para conquistar.

7. Planejar estratégias de testes

Antes de começar os testes o analista de QA define algumas estratégias de testes, como realizar o teste de um item por mais de um testador, distribuir os itens mais complexos ao testador sênior, fazer com que ocorra várias baterias de testes, enfim existem várias estratégias que podemos implantar durante o ciclo de testes.

As estratégias de testes nos ajudam chegar a um objetivo que é eliminar o máximo de bugs, erros e outras inconsistências encontradas, para que seja entregue ao cliente o que ele pediu e funcionando.

Puleio (2005, p.19) relata a sua experiência de aplicação de práticas de teste em um projeto utilizado SCRUM. Descreve que, para cada iteração, os casos de teste deveriam ser escritos antes da implementação e conclui dizendo que testes automatizados são a chave do teste ágil, pois trazem grande melhoria em relação a testes de regressão, após a execução de novas funcionalidades.

Uma estratégia de teste de software fornece um roteiro que deve responder a questões do tipo:

Quais os passos a serem conduzidos como parte do teste?

Quando os passos são planejados e executados?

O programa deve ser testado como um todo ou em partes?

Testes que já foram conduzidos devem ser reexecutados quando o software é modificado?

Quando o cliente deve ser envolvido?

[...] uma estratégia de testes de software descreve a abordagem geral e os objetivos das atividades de teste. Ela deve contemplar os níveis ou fases de teste, os tipos de testes a serem realizados e as técnicas para sua execução. A estratégia de testes de softwares também deve descrever com clareza os critérios para a conclusão dos testes e os critérios de sucesso a serem usados.

O intuito é evitar desperdício de tempo, fazer com que a equipe não se esforce desnecessariamente e não ocorra a infiltração de erros que não foram descobertos pela equipe de testes.

8. Realizar testes exploratórios

Para realizar um teste exploratório o testador deve estudar o sistema, saber suas principais funcionalidades, quais são os módulos onde a atenção deve ser maior, entender sobre a regra de negócio da ferramenta e o seu comportamento, para que encontre o defeito que não foi pego pelo caso de teste.

O testador ao iniciar o teste exploratório faz o teste com base na sua experiência e conhecimento do sistema, deve explorar além do conhecido.

São vários os benefícios ao realizar um teste exploratório geralmente quando não existem os requisitos, pouco tempo disponível, pode ser descoberto um defeito aleatório, verificar que o comportamento não está correto e também você pode achar um defeito crítico.

Bach (2003, p.07) descreve em geral, testes exploratórios é chamado para em qualquer situação onde não é óbvio que o próximo teste deve ser, ou quando você quer ir além dos testes óbvios.

[...] mais especificamente, exploratório estilo livre de teste se encaixa em qualquer das situações seguintes:
Você precisa fornecer um feedback rápido sobre um novo produto ou recurso.
Você precisa aprender o produto rapidamente.
Você já testou o uso de scripts, e buscam diversificar os testes.

Você quer encontrar o único erro mais importante no menor tempo possível.
Você deseja verificar o trabalho de outro testador, fazendo uma breve independente investigação.
Você deseja investigar e isolar um defeito particular.
Você quer investigar o status de um risco particular, a fim de avaliar a necessidade de testes de scripts nessa área.

São várias as vantagens pois você executa ações no sistema que não estão previstas nos casos de testes, você faz o seu teste do seu jeito onde deve tentar caminhos impensáveis, pode aumentar o número de defeitos encontrados e agrega mais conhecimento para si.

9. Reforçar a definição de pronto

A definição de pronto tem como objetivo mostrar ao time e orientar sobre o que significa uma entrega pronta, porém nem todos sabem claramente o que é o “pronto”, é uma premissa onde o SM, Analista de QA e o time devem atender as necessidades do cliente entregando um software de qualidade.

Henrik Kniberg (2007, p.33) garante que é importante que o product owner e a equipe concordem com uma definição clara de “pronto”. Uma história está completa quando todo o código está no repositório? Ou está completa apenas quando foi feito deploy em um ambiente de teste e a história foi verificada por uma equipe de testes de integração?

[...]sempre que possível nós usamos a definição “pronto para deploy em produção”, mas algumas vezes precisamos usar a definição “feito deploy em servidor de teste e pronto para os testes de aceitação”. No início costumávamos ter checklists detalhados para isso. Atualmente nós frequentemente dizemos “uma história está pronta quanto o tester da equipe Scrum diz que está”. É responsabilidade do tester se certificar de que a intenção do product owner foi compreendida pela equipe e que o item está suficientemente “pronto” para que passe pela definição de “pronto” acordada.

Pronto não é apenas definir as etapas que serão realizadas no decorrer do projeto, o importante é evoluir o conceito de pronto, fazer com que o time ao longo do projeto seja ágil, eficiente e que tenham uma visão clara de quais são os critérios de pronto para ser feita uma entrega de valor ao cliente.

De acordo com o Guia do Scrum (2013, p.16) quando o item do Backlog do Produto ou um incremento é descrito como “Pronto”, todos devem entender o que o “Pronto” significa.

[...] embora isso varie significativamente de um extremo ao outro para cada Time Scrum, os integrantes devem ter um entendimento compartilhado do que significa o trabalho estar completo, assegurando a transparência. Esta é a “Definição de Pronto” para o Time Scrum e é usado para assegurar quando o trabalho está completado no incremento do produto.

A definição de pronto é uma das definições mais importantes, deve ser entendida por todos do time antes mesmo de correr atrás de metas e objetivos, para manter a qualidade do trabalho e amadurecimento da equipe, se alguém falar que o projeto está pronto todos do time devem saber o que o pronto significa. Se falta coisas fáceis, grandes ou complexas não está pronto. Pronto é quando não se tem mais nada para fazer no projeto.

10. CONCLUSÃO

São vários os benefícios obtidos aplicando o Scrum na Qualidade de Software é possível destacar melhor integração da equipe, otimização do tempo de trabalho aumentando assim a produtividade, previsões mais acertadas na entrega de produtos, disseminação do conhecimento e a formação de equipes auto gerenciáveis. É uma forma de garantir que os produtos mantenham um bom nível de qualidade e ter um processo de desenvolvimento de software com qualidade.

Software quality with Scrum

ABSTRACT

This article discusses Scrum aligned with Software Quality, a number of improvements in the process of quality software development teams using Agile Scrum model. The characteristics that define the quality of a software relate to good software engineering practices, Scrum as a framework for project management is able to offer improved quality of software, in order to give the customer what your product according your needs. The team works with more confidence and security

when it held a major change during the project cycle, in the event of problems they are circumvented more easily using Scrum and together with the software quality we must understand the standard of quality that is to ensure a product that meets the expectations within what was initially agreed with the client, which should be measured end-to-end working to ensure that the product is better.

KEYWORDS :: Scrum. Software quality. Tests. Team.

11. REFERÊNCIAS

BACH, J. Exploratory Testing Explained. Disponível em:
< <http://www.satisfice.com/articles/et-article.pdf> >. Acesso em: 07 set.2015.

FREIRE, F. Desvendando o Scrum. Disponível em:
< <http://www.artecom.com.br/revistaidigital/downloads/Scrum.pdf> >. Acesso em: 05 jan.2014.

IEEE. Software Engineering Body of Knowledge - Swebok Traduzido. Disponível em:
< http://www.cin.ufpe.br/~bfs3/UFPE/Engenharia_Software/77017069-SWEBOK-traduzido.pdf >. Acesso em: 10 mai.2015.

KNIBERG, H. Scrum e XP direto das Trincheiras. Disponível em:
< <https://www.cti.ufu.br/sites/cti.ufu.br/files/scrum-e-xp-direto-das-trincheiras.pdf> >. Acesso em: 15 set.2015.

PULEIO, M. How Not to do Agile Testing. Disponível em:
< <http://www.agilexp.com/Agile200xPapers/Agile2006-ExperienceReports/XR21-puleio-HowNotToDoAgileTesting.pdf> >. Acesso em: 15 jun.2015.

SCHWABER, K. Guia do Scrum. Disponível em:
< http://www.training.com.br/download/GUIA_DO_SCRUM.pdf >. Acesso em: 20 mai.2015.