**Time: Cássia Ellen G. A. Oliveira, Josué da Silva Souza, Willian Junio da Silva.**

**Case Automacode**

Por se tratar de uma empresa de grandes proporções, na qual existem grandes projetos, o **primeiro** ponto a ser observado é a divisão dos times, de forma que cada time seja responsável por um projeto ou por um grupo de projetos nos quais os produtos estejam relacionados. Para isso, caso o número de Qa’s não seja o suficiente, será necessário a contratação de novos profissionais para compor os times.

Ao realizar a separação dos times, espera-se que haja um aperfeiçoamento do time naquele referido projeto, tecnologia ou linguagem, tornando as pessoas envolvidas mais especialistas em suas funções.

A separação pode ser feita entre tecnologias, por exemplo, Time A voltado para os produtos *mobile,* Time B responsável pelos produtos Web, Time C responsável por testar os Serviços, etc. A divisão pode ser feita também de acordo com os produtos, exemplo, Time D responsável por produtos relacionados a vendas e Time E responsável pelos produtos relacionados a Cadastros, e assim por diante. Ou a separação pode se dar até mesmo como um misto dos dois exemplos citados acima.

O **segundo** passo necessário para se obter uma maior qualidade tanto do processo quanto dos produtos entregues pelo time, é a adoção de uma metodologia ágil. Com isso podemos citar alguns pontos que serão melhorados ou que começarão a existir a partir da adoção pela empresa como:

* Entrega de valor contínuo.
* Envolvimento do cliente, contanto com feedbacks constantes.
* Facilidade de adaptação a mudanças.

Com a adoção dessa metodologia haverá redução do retrabalho existente devido a necessidade de se fazer grandes alterações no escopo do projeto quando esse já se encontra na fase de testes, pois, com a entrega contínua o intuito é realizar pequenas mudanças ao longo das entregas e não grandes alterações.

Outro benefício da adoção de uma metodologia ágil é que os testadores não necessitarão esperar que o produto já esteja “finalizado” para iniciar os testes, visto que podem iniciá-los antes mesmo do início do desenvolvimento, e sim nas fases iniciais como elaboração do projeto.

Temos ainda como ganho da adoção dessa metodologia, a integração entre todos os membros que compõem a equipe, onde podem ser ouvidos todos os participantes do projeto, possibilitando assim um compromisso maior na entrega do produto, um feedback dos participantes durante cada evento. Com isso o time fica mais próximo e assim pode ser observado ao longo do tempo o que pode ser melhorado ou alterado de forma que o time consiga ser o mais funcional possível.

Para isso foi escolhido o framework SCRUM, que possibilitará além da entrega contínua, que os times se auto gerenciem. Para tanto é necessário que cada time possui, além dos Tester’s/Qa’s e desenvolvedores, também um responsável por ajudar o time garantindo que esse seja funcional e produtivo e garantir que o processo esteja sendo seguido, o chamado ScrumMaster**.** E também o representante dos clientes, o PO.

Visto que o time de QA não possui conhecimentos adequados para realizar a automação de testes, o **terceiro** passo será a realização de treinamentos. Esses treinamentos podem ser internos ou externos, e será necessário um apoio da empresa para que os profissionais realizem cursos, participem de eventos e obtenham qualificações para elevar os conhecimentos. É interessante que cada QA procure se especializar em uma linguagem ou ferramenta que será utilizada dentro do seu time, por exemplo, QAs que trabalham com aplicações WEB estudarem sobre Cypress, os que trabalham com aplicações desktop aprenderem sobre o TestComplete, etc.

O **quarto** passo a ser implementado, é a definição dos pontos principais e críticos dos projetos. Será necessário que o time de QA se junte e identifique os pontos que mais agregam valor ou que possuam maior risco. Isso pode ser feito observando as principais funcionalidades, as mais utilizadas, e também as áreas nas quais mais são relatados bugs. Após esse levantamento o time terá uma base por onde começar a elaborar os testes, sejam eles automatizados ou não.

Após definidos os pontos críticos podem ser iniciados a criação dos testes, mas antes é necessário definir uma ferramenta onde seja possível gerenciar os testes que serão criados, além de possibilitar reportar os bugs, e realizar diversas outras tarefas de interesse do time. Para isso definimos o **quinto** passo, que é a adoção do JIRA para gerenciar nossos projetos. Dessa forma será possível também um melhor controle de gerenciamento de bugs, o que fará com o que o time tenha uma maior integração. É interessante que os QAs detalhem o máximo possível as evidências encontradas durante os testes, facilitando o entendimento dos Devs, e assim evitando possíveis conflitos.

Após implementados os passos acima podemos focar no **processo de testes** em si. O **primeiro passo** é a implementação de testes unitários como forma de melhorar a qualidade dos produtos desenvolvidos. Porém, essa prática deverá ser realizada pelos desenvolvedores, podendo os QAs auxiliarem os mesmos caso seja necessário.

Para que seja possível aumentar a cobertura dos testes feitos é interessante que sejam **definidas metas para criação de cenários de teste (descritivo e automatizado),** esses cenários serão criados de acordo com as demandas encontradas durante o levantamento dos pontos chave/críticos do projeto. Como forma de acompanhar essas metas pode ser adotado a **metodologia de OKR**, onde serão definidos os objetivos que podem ser progressivos aumentando a cobertura dos testes ao longo dos dias, semanas, meses ou sprints. A **criação de métricas** terá como objetivo nortear o time, aumentando a motivação e fazendo com que o time consiga acompanhar sua evolução ao longo do tempo.

Com a criação dos cenários será possível dar início a implementação de testes automatizados. Para isso é necessário que seja escolhida uma linguagem/ferramenta que atenda as necessidades de cada time. Para que essa ferramenta seja escolhida é interessante que seja realizado um estudo pelo time de QA sobre qual ferramenta utilizar, levantando os prós e contras de cada uma encontrada e ao final seja decidido a que melhor se encaixa nas necessidades dos times.

Para o desenvolvimento dos testes automatizados será necessário que exista uma padronização do ambiente dos QAs para que possam testar adequadamente os testes desenvolvidos, para tanto, pode-se tornar necessário o auxílio do time de desenvolvimento para que essa padronização ocorra.

Feito isso será possível dar **início ao desenvolvimentos dos cenários de teste automatizado** a partir dos casos/cenários de teste que já existem e que eram feitos de forma manual, e a medida em que os testes forem criados poderá ser criado as **suítes de teste e executados os testes de regressão** sempre que houver alguma alteração no projeto ou a medida que novos testes forem criados, dessa forma, haverá a redução de retrabalho e haverá menor probabilidade de quebra de funcionalidades.

Durante a criação dos testes automatizados, poderá haver um gasto significativo de mão de obra para que sejam criados dados para os testes, dessa forma, os QAs podem buscar ferramentas que auxiliem na criação de massas de dados e aplicar sua utilização na empresa.

É ideal que seja implementada a entrega contínua para os testes automatizados criados. Com isso, além de garantir menores riscos, mais qualidade e velocidade, também será possível que o time acompanhe a evolução dos testes. Dessa forma o time de QA poderá apresentar para o restante do time os resultados obtidos, o grau de cobertura de testes já feito, relatórios automáticos obtidos pela ferramenta utilizada. Um bom indicador é o número de chamados, e isso pode ser mostrado para o time em formato de gráficos, mostrando a redução em consequência da melhoria da qualidade dos produtos.

Uma forma de melhorar o desempenho do sistema é implementando testes de desempenho, stress e carga, como forma de validar não só a parte funcional mas também como o sistema se comporta em situações extremas, como elevado número de acessos simultâneos. Para esses testes podem ser utilizadas também ferramentas para auxiliar os QAs.

1. **Separar times responsáveis por cada projeto ou grupo de projetos.**
2. **Automatizar os testes funcionais onde existem cenários de teste já criados.**
   * **Implementar testes unitários.**
   * **Criar os cenários de teste das principais funcionalidades existentes em cada projeto**
   * **Criar os ambientes de teste (cobrar dos devs ou gerência)**
   * **Definir uma ferramenta de teste.**
   * **Automatizar os cenários existentes.**
   * **Implementar testes de regressão.**
3. **Fazer análise de risco/pontos críticos.**
4. **Utilizar ferramentas de geração de massa de dados.**
5. **Implementar uma metodologia ágil – SCRUM**
6. **Realizar treinamentos internos / Disponibilizar cursos, eventos, certificações para que os QAs participem e adquiram conhecimento.**
7. **Utilizar o JIRA para centralizar o desenvolvimento como um todo. Implementar entrega contínua.**
8. **Gerar relatórios das coberturas de testes. Burn down, gráficos mostrando a quantidade de chamados ao longo do tempo. Índice de satisfação dos clientes.**
9. **Auxílio aos desenvolvedores na criação dos testes unitários. Melhorar a apresentação de evidências e passos para reproduzir os bugs ou falhas encontradas.**
10. **Definir metas para aumentar a quantidade das coberturas de teste.**
11. **Implementar testes de desempenho, stress e carga.**