11711BCC008
Matheus José da Costa

"Se refactoring é tão importante no contexto de práticas ágeis, se práticas ágeis são tão adotadas hoje em dia, por que um estudo empírico mostrou que 80% dos bad smells permanacem no código ao longo do tempo e por que refactoring é responsável por eliminar apenas 9% dos smells?"

Em relação a essa indagação, primeiro necessitamos entender o processo de desenvolvimento de software. Nos dias atuais, temos que, nas empresas, os papéis, dentre os times, são divididos. Geralmente, é adotado algum tipo de metodologia ágil para aumentar a produtividade das entregas.

Sendo assim, é dividido em: Project Manager, Scrum Manager, Equipe de Desenvolvimento, Equipe de Teste e um cara de Infraestrutura. Geralmente, as sprints são negociadas com a Project Manager pelo time de desenvolvimento em conjunto com o Scrum, e ali é definido se dada tarefa entrará ou não na sprint.

Desde a concepção do software até seu estado de implantação em ambiente de Produção, muitas alterações são feitas. Sendo assim, Padrões de Projeto, muitas vezes, por conta de regras de negócio, podem ser feridos. E, mais ainda, além das regras de negócio, os próprios desenvolvedores podem fazer implantações que pra dada época fazia sentido, mas que pra outra época não faça.

Com isso, o código vai ganhando “mau cheiro”, e a equipe, por se preocupar com o futuro, faz a solicitação para o Scrum tentar encaixar uma tarefa de Refatoramento em certa sprint.

De fato o Refatoramento não tirará todos os Smells do código, e possivelmente colocará novos smells. Porém isso vai muito do contexto atual em relação a um contexto futuro.

Ainda assim, mesmo prevendo que um Refactoring não será 100% efetivo, é aconselhável fazê-lo para que gere uma melhor legibilidade do fluxo do projeto. Assim, novas pessoas que ingressarem no projeto, poderão trabalhar com maior fluidez.

A grande questão dos Refactorings é que nem sempre eles são priorizados pela Project Manager. O projeto tem que estar num estado de muita maturidade para esse tipo de “enhancement” seja acolhido. Justamente porque, geralmente, o cliente apenas quer colocar novas features. E isso acontecendo, o código cada vez mais ficará cheio e caso tenha a necessidade de um Refactoring a partir de novas funcionalidades, demorará mais. Então, o refactoring pode sim facilitar um desenvolvimento futuro, em que gastará menos horas para implementação ou correção de dado bug.