



práticas do eXtreme Programming: refactoring

**josé augusto fabri
fabri@futfpr.com.br**



Refactoring

- O que fazer com um código que está funcionando, mas é pouco legível?
- Como proceder diante de um código mal escrito, mas que funciona?
- O ditado popular já disse: em time que está ganhando não se mexe.
- Não mexer em um código que está funcionando é uma regra implícita na maioria dos processos de software.



Refactoring

- Não mexer em um código mal escrito pode acarretar problemas, por exemplo:
 - Qualquer parte do software pode precisar ser visitada no futuro para se fazer manutenções corretivas, preventivas ou adaptativas.
 - Java code convention
- Um código mal formulado cria sérias dificuldades para quem um dia tiver necessidade de modificá-lo ou compreendê-lo.
- Quando o código está legível é possível alterá-lo rapidamente.



Refactoring

- O XP recomenda que os desenvolvedores não fiquem parados diante desta situação.
- Se o desenvolvedor tem alguma dúvida em uma parte do código, ele deve procurar compreendê-lo e, em seguida, reescrevê-lo de modo a ficar mais legível.
- Ele não está alterando a funcionalidade, dado que o código continua fazendo a mesma coisa.
- Ele só está tornando mais legível.



Refactoring

- Toda a atividade citada, recebe o nome de refactoring.
- Na verdade, o XP vai além:

É OBRIGAÇÃO DO PROGRAMADOR FAZER O
REFACTORING DE QUALQUER PARTE DO CÓDIGO
QUE ESTEJA POUCO LEGÍVEL.

- A alteração não deve afetar o comportamento que originou do código, ela deve apenas torná-lo mais legível.



Refactoring

- O conceito de refactoring anda de mãos dadas com a prática de código coletivo.
- O refactoring torna simples a prática de código coletivo, pois os desenvolvedores conseguem ler qualquer parte do código facilmente.
- A prática do código coletivo ajuda a equipe a identificar a prática do refactoring.
- Quando várias pessoas lêem o mesmo código, fica fácil identificar práticas mal formuladas.



Refactoring

- Metáfora: O software é como nossa casa.
- A casa é um lugar que vive em constante evolução e mutação.
- Existe uma certa estabilidade, mas de vez em quando acontece uma coisa nova.
- Compramos um aparelho, colocamos um quadro na parede ou encontramos um belo tapete para o chão.



Refactoring

- Temos duas abordagens para nossa casa:
 - Organizados ou Desorganizados.
- As pessoas lavam os pratos, retiram o lixo, arrumam a cama e executam outras atividades.
- Estas tarefas têm um pequeno custo diário. Entretanto, elas geram flexibilidade, agilidade e conforto.
- Suponha que você queira convidar alguns amigos para jantar em sua casa. Se você for desorganizado, será preciso marcar o jantar com dois meses de antecedência.
 - Pois você necessita organizar a bagunça.



Refactoring

- Porém se a casa for organizada, você pode convidá-los para jantar no dia seguinte.
- A casa fornece agilidade, flexibilidade e conforto.
- O mesmo acontece com o software.
- Com algumas ações diárias a equipe pode mantê-lo organizado e legível.
- Isto permite ao cliente que incorpore qualquer necessidade com agilidade.



Refactoring

- XP e refactoring: uma relação importante, pois o refactoring fornece agilidade para a questão do feedback.
- Na comunidade XP existe um ditado:
 - Um dia sem refactoring é como um dia sem sol.



dúvidas

josé augusto fabri
fabri@utfpr.com.br