Notas sobre o capitulo 4 de Clean Code – by Lucas Trevizan

O livro começa abordando explicitamente a **escrita de comentários em código como um fracasso**, muito relacionado ao fato de que os comentários não se mantêm atualizados à medida que os códigos mudam.

Nesse ponto eu concordo, de fato muitas vezes vemos comentários não tão bem relacionados ao que o bloco de código faz exatamente, uma solução abordada é simplesmente nos forçar a escrever um código que não precise de comentário, olhando pelo ponto de que o comentário é no máximo um mal necessário.

Comentários compensam código ruim?

Uma das grandes motivações a escrever comentários é um código ruim, nós conseguimos saber quando escrevemos algo que está confuso e talvez usar um comentário seja uma dedução rápida de como explicar a bagunça, mas a verdade é que é muito melhor gastar seu tempo arrumando um código do que tentando explica-lo com comentários.

Comentários bons.

Nem todo comentário é ruim, existem situações onde eles são necessários ou muito uteis. Um bom exemplo são os comentários informativos (***@Javadoc por exemplo***) onde podemos transmitir que parâmetros um método espera e o tipo do seu retorno, além de uma explicação de 1 LINHA que resuma o seu objetivo, óbvio que se você colocar esse tipo de informação implicitamente no nome do método o comentário pode ser usado como um complemento.

Explicação da intenção e Esclarecimento.

Usar comentários para explicar a intenção por trás de uma decisão é uma maneira inteligente de se usar, assim você específica para o próximo que pegar aquele trecho de código o que motivou ir para um caminho específico.

Comentários também podem ser usados para esclarecer coisas obscuras, como por exemplo variáveis que tem a nomes monossilábicos, ou métodos com parâmetros “esquisitos”. PORÉM CUIDADO, um comentário de esclarecimento tem que ter certeza de que está informando a situação corretamente, esses são o tipo mais fácil de comentário a ficar datado.

Conclusão.

Esse capitulo fala sobre todo tipo de comentários, os engraçadinhos, os redundantes, os bons e os ruins, porém ele destaca fortemente que todo comentário é RUIM e se você precisou fazer um então pelo menos deve fazer direito.

E fazer direito significa: nada longo, ou repetitivo, ou inconclusivo e etc... Eu pessoalmente não vejo comentários com essa reputação, claro que um código bem feito deve falar por si só, mas não é a realidade que vemos no dia a dia.

Portanto eu prefiro manter o uso de comentários pautado bastante no ***@Javadoc***, onde eu falo com uma linha ou menos de maneira objetiva o que meu método/classe/atributo faz e complemento com as demais informações como ***@param, @return*** e ***@throws.***