Clean Code Capítulo 4

Comentários

Resumo

Comentários bem colocados são úteis, por outro lado, comentários desnecessários só irão inflar o código e prejudicar a utilização do mesmo. Comentários são considerados um mal necessário, de forma que, se os códigos fossem totalmente explícitos, não seria necessários comentar para expressar a intenção que o código tem, assim sendo, o uso de comentários é sinônimo de fracasso, pois há sempre a necessidade de usar um comentário quando o código não consegue expressar si mesmo. Portanto, ao realizar um comentário em um código, pense se não há uma maneira de expressar a ideia pelo código, até mesmo pela evolução do código, de forma que conforme a atualização dos códigos, os comentários acabam ficando obsoletos.

Comentários compensam um código ruim

O que motiva inserir um comentário em um código, é um código ruim, ou seja, quando você observa que o código é desorganizado, você decide inserir comentários para explicar toda a bagunça criada, e para reparar isso, deve-se usar o tempo de criar um comentário para realizar a reparação do código.

**Comentários bons**

Como comentário bom, é aquele que você consegue transmitir através do código, inserindo de uma forma no código que você não precise comentar.

Comentários legais. Por questões de lei, algumas vezes códigos devem portar comentários que incitam direitos autorais ou informações corporativas.

Comentários informativos. Um comentário que informa o processo de uma função pode ser muito útil, no entanto, este comentário pode ser trocado ao alterar o nome da função para algo que transmita a informação.

Esclarecimento. O comentário de esclarecimento pode assumir seu risco de estar errado, entretanto, pode ser interessante encontrar uma forma de esclarecer um código.

Alerta sobre consequências. Um comentário lógico como alertar os riscos a outros programadores as consequências de um código, pode ser muito importante, fazendo com que, a pessoa que irá utilizar o código fique ciente das recomendações.

Comentários TODO. Os comentários TODO (To Do “Fazer”), são para alertar partes de um código que devem ser efetuadas, porém, por algum motivo, no momento não está em condições de uso.

**Comentários ruins**

Em maioria, todo comentário cai na categoria de ruim, pois, o uso destes, em suma, insinua a falta de congruência do código, servindo como justificativa ao fracasso do código.

Murmúrio. Em caso de murmúrio, destaca-se, a visão errônea ao comentar um código, de modo que, nesses casos, o autor, pensa que poderá inserir um comentário informativo, no entanto, o comentário feito é uma besteira enigmática, que só faz sentido para quem o criou.

Comentários redundantes. Este tipo de comentário não serve pra nada, de modo que, o comentário nada irá fazer a mais que explicar justamente o que código está fazendo.

Comentários enganadores. Em alguns casos, desenvolvedores acabam cometendo o erro de enganar através de um comentário, demonstrando algo que na verdade não irá funcionar da forma descrita, está desinformação pode levar a alguns problemas maiores.

Comentários imperativos. Estes tipos de comentários, são aqueles que estão unicamente para amontoar o código, podendo levar a desinformação do mesmo.

Comentários longos. Como seu nome já diz, o comentário longo é aquele que vai utilizar várias linhas de um código, apenas amontoando seu código, fazendo com que o processamento dele será reduzido.

Comentários ruidosos. Nada mais que comentários que explicam o óbvio, que tentam ensinar o que a função faz, quando o ideal, é escrever a função, de forma que ela mesmo se explique.

Marcadores de posição. Em casos, desenvolvedores gostam de marcar posições em código, quando na verdade, eles estão ali apenas para amontoar um código.

Créditos e autoria. Não há a necessidade de poluir um código, com comentários indicando a autoria de tal, pois esses comentários não possuem alguma relevância.

Explicação do código em comentários. Não utilize comentários para explicar um código, de modo que, o código deve se explicar por si mesmo, comentários como esse estão ali, apenas para acumular informações desnecessárias.

Comentários HTML. Esses comentários não são responsabilidade do desenvolvedor, pois em um código, esses comentários apenas dificultam a leitura e interpretação do código desenvolvido.

Informações não-locais. Se você deseja comentar um código, escreva-o perto da função, e não deixe em qualquer lugar, logo mais, estes comentários não tem a tendência de sofrerem atualizações.

Conexões nada óbvias. Não dificulte a leitura de um código, utilizando informações que além de desnecessárias, não são de fácil entendimento.

Cabeçalhos de funções. Funções curtas já apresentam por si, o que ela tem que fazer, ignorando a presença de um comentário.