

Notas sobre o capítulo 5 de Clean Code – by Lucas Trevizan

O Objetivo da formatação

Um dos principais objetivos da formatação é contribuir com um código fácil de ser manutenível, ele até cita que uma boa formatação é a primeira regra de negócio de um desenvolvedor profissional acima até da regra de “fazer funcionar”.

Eu concordo totalmente com esse ponto, depois de aplicar manutenção e modificações em diversos códigos de um sistema eu percebi como uma legibilidade impacta mais que a funcionalidade, já que um código dificilmente se mantém o mesmo.

Formatação vertical

O Exemplo mostrado sobre código vertical é muito interessante, ele é representado por um gráfico, mas a ideia central é mostrar projetos famosos no mundo Java e comparar a quantidade de linhas que os arquivos que compõem os projetos tem.

O ponto central é que é possível construir projetos relevantes utilizando códigos simples de mais ou menos 200 linhas, com um limite máximo de 500 e isso não é uma regra fixa, mas com certeza é uma base consistente para se adotar. Falando por experiência própria, ler um arquivo com 2 mil linhas é aterrorizante.

Metáfora do jornal

Embora o papel esteja sumindo da nossa vida (ou deveria), um exemplo fantástico é o de comparar o código fonte a um jornal. Um jornal tem vários artigos, a maioria deles são pequenos, alguns são grandes, MUITOS POUCOS tem textos que preenchem a página toda e isso torna o jornal aproveitável.

Trazendo os conceitos para um projeto, podemos deduzir que ***muitas classes devem ser pequenas***, algumas ***poucas podem ser maiores***, e ***menos ainda*** devem ser ***godClasses***.

Falando sobre a ***manchete***, a gente pode aplicar a ideia pro ***nome de uma classe***, ela tem que ***ser descritiva a ponto de sabermos se estamos no lugar certo***, abaixo da manchete vem um ***resumo***, que seriam ***os métodos principais ou algo que me dê um resumo do que é aquela classe*** e por fim vem os detalhes.

Espaçamento vertical entre conceitos

Famosa linha em branco entre conceitos ou pontos de separação na leitura do código, essa dica é simples, mas falo com tranquilidade que é PODEROSA!

Distância vertical

Esse trecho é longo e pouco amigável a um resumo. Mas aqui são relatadas diversas abordagens de distância entre variáveis, funções e etc.

Como por exemplo, algo que eu utilizo muito, que é se um método chama outro método, então eles devem ser escritos na ordem em que o primeiro que chama está DIRETAMENTE

acima do que é chamado. Essa ideia de escrita faz com que possamos ler a classe inteira como um livro, onde um método desencadeia o próximo e assim sucessivamente.

Formatação Horizontal

Ao falar sobre a largura das linhas, nitidamente a maioria dos programadores preferem linhas menores. Aqui o autor diz sobre 120 caracteres ser o limite dele, eu pessoalmente prefiro quebrar linhas por expressões, por exemplo, instância de um objeto é uma linha, uma stream aninhada eu quebro por método encadeado e etc.

Mas eu nunca deixo linhas longas ou confusas, então minha regra pessoal ao lidar com tamanho de linha é sempre perguntar ao ler: “Da pra diminuir e dar mais significado?”

Espaçamento e continuidade Horizontal

Aqui nós temos um detalhe que pode fazer muita diferença, o espaço em branco entre as linhas (***horizontalmente falando***), aqui mais comumente usados em operadores que separam sua variável a direita e a esquerda.

Esse pequeno espaço ou a falta dele que pode dizer que uma coisa está fortemente associada a outra como por exemplo um operador de incremento(***“contar++”***) é algo bem sutil e que as vezes passa batido, mas automaticamente fazemos e vale pensar nele tanto quanto pensamos no espaçamento vertical.

Endentação

A endentação é maravilhosa, é o que torna os programas inteligíveis, o “tab” do código que está em um escopo dentro de outro e de outro e assim consecutivamente que permite enxergar a hierarquia em métodos. Eu acho difícil um ser humano que é programador não usar a endentação.

Algo mencionado também é como fica a endentação em escopos minúsculos como um while que é resolvido em uma linha. Eu pessoalmente nunca vi um loop tão pequeno que fosse resolvido em uma linha e espero nem ver, e mesmo que a expressão seja pequena, ainda assim eu a endentaria.

Regra de equipes

Aqui ele salienta que um projeto é de um TIME e, portanto, deve seguir regras de um time. O mais perto que cheguei de ver esse padrão foi com um software chamado “melhores práticas” de uma empresa onde trabalhei, esse software escaneava o código fonte e apontava padrões que não estavam sendo seguidos e deviam ser corrigidos.

Esse “melhores práticas” nunca foi tão bom no que se propunha, mas ele não interferia nos nomes que eu escolhia, então no fim das contas eu não me importava.

Conclusão

Se um código por si só é a melhor “documentação” então a formatação tem um papel muito importante nisso.

O que mais me chamou a atenção e me fez refletir foram os espaços em branco que deixamos tanto **horizontalmente quanto verticalmente**, por instinto eu já praticava essas coisas e agora eu consegui entender do porquê parecer tão melhor usa-los.