Reflections on Chapter 2, “Meaningful names” in Clean Code by Robert Martin
Author: Jimmy Karlsson, 2023-09-22

In the words of Theodore Parker; “The books which help you most are those which make you think the most. The hardest way of learning is by easy reading; but a great book that comes from a great thinker – it is a ship of thought, deep freighted with truth and beauty.”

Chapter two of Clean Code titled “Meaningful names”, fails to inspire such contemplation.
Instead, it smothers any fleeting spark of thought with overwhelming, overwrought explanations, sending even the most valiant decorator pattern fleeing in abashed retreat .

Were the chapter about how to tie a shoelace, readers would be granted a convoluted journey through the history of shoelaces, culminating in a philosophical debate on the cultural complexity of “left over right” vs “right over left”.
With the possibility of a tangent on the moral implication of uneven loops.
Pardon my exaggeration for effect… but it carries the point across.
Namely, in way too many words I have said very little.

“Meaningful names” is not devoid of good ideas and valuable insight, these just get muffled by verbosity. As Edouard Manet once said, “The concise man makes one think: the verbose bores”.
I firmly believe that the fifteen rules that manage to cover thirteen pages together, could be distilled into a sentence or two each… rendering them, quite fittingly, more meaningful.

TLDR version of Meaningful names:

Names should be distinct, consistent, and to as precise a degree as possible describe the nature or purpose of what they represent.

Distinct: names should be unique for ease of searching and to eliminate confusion. As an example, avoid using ‘account’ and ‘accounts’ in the same project.

Consistent: names should mean the same and help developers understand the codes intent. If you use ‘fetch’ to retrieve external data, stick to that and avoid suddenly using for example ‘get’.

Descriptive:
- A method name should describe its action. Example: ‘writeTextToFile(text, filename)’ clearly indicates what action it takes: writing the text to a file.
- A variable name should describe its nature. Example: ‘xCoordinate’ indicates a position on the x-axis. ‘heightToWidthRatio’ is a number containing height compared to width.