

## **THE HIDDEN FACE OF FUNCTIONAL DOCUMENTATION**

The main point of functional documentation is better code. The code is more coherent and goes straight to the point, instead of following sinuous paths. It will be easy to change, without losing pieces.

The lack of functional documentation is like building a rocket, without anyone knowing what they are exactly doing or the clear purpose of the piece they are working on. Sometimes, functional documentation can just be verbal and this can work for a little bit of time. But one day a bug comes out and you have a dilemma: you know what the software does, but not what it is supposed to do. You will ask someone of the colleagues, that believe to know what it is supposed to do, but you realize he was wrong, after the customer complains and it's too late.

Good code is almost impossible without functional documentation, doesn't matter how good you are. It's like a good driver without a map: he will end up in the wrong place and will do the wrong things.

It's impossible to do complex implementation or radical changes without a map, and the code will be worthless. But with a map in your hands, the code will be like an open book, you can read and understand it.

How difficult is it to read a book from which someone has removed the cover and you don't know the title? How difficult is it to read a chapter from which someone has removed the title? How long does it take to figure out the main point of the text?

The same is true when you don't have any guide and explanation about a cryptic code. Remember, you will shortly forget the code you are writing, but your team will hardly forget a clear and well written functional description of it.