

RESTRUCTURE, INTEGRATE... THEN ADD

The wrong way of adding a feature, performing a change or fixing a bug is by implementing it as a special case, ignoring the behaviour of the existing code.

The rule is quite simple: **restructure your code to accomodate new changes, integrate what is new in the form of reusable code, then add your change calling the reusable code.**

"Integrate" means blending the new code with the existing code. Don't add the new feature on the top of the application, but integrate it. A plumber that have to bring water to a new sink does extend the current hydraulic network, he doesn't build a new network from the bottom of the building to the top. That's exactly what happens if we don't reuse code and rewrite features from the scratch every time. Some developers might not even read the code and decide to do things in a different way for each new feature. For example, instead of reusing and extending a class that decodes an input string, he decides to parse all the new inputs in a special way. However, things are not always easy, and you might need some restructuring before. A mason can be asked to add a second bathroom to one of the flats. This was not considered at the beginning of the project and there is no space available to build that. The building should then be restructured to allow the mason to add bathroms, wherever requested. The same happens when a developer is asked to add some control information to a communication protocol. If we have a header, this is pretty simple. However, if we don't have it, we

have to restructure the code and implement the header, so that we can add control information. Don't just add a field somewhere and don't use some "smart" tricks. You might have to add other information in the future... create a header!

The formula is pretty clear now: **every time you have to add a new feature, restructure your code to make the change possible and logical; integrate your code to make the change smooth and imperceptible; finally, add you change with a final master-touch.**

If you carefully follow this formula, usually the third step is pretty simple. Adding the functionality can now be something like calling a new method and adding one line of code in the right place.

Not always you will need to restructure and integrate, but if you skip these steps, when required, you will add more and more complexity, you will make your code hard to maintain or understand, and above all... you will end up in a sea of troubles!