University of Essex

Online

Antipatterns

The term ‘antipattern’ was first discussed in 1995. Antipatterns tend to evolve when a pattern which is appropriate to the problem does not exist. The antipattern appears to be an appropriate solution to begin with, however, it then evolves into something which does not provide an effective solution. It is important to take note of these 'solutions', in inverted commas, which are not optimum, as they are tried and tested to not be effective. By making others aware that they are ineffective, it avoids the need of any developers having to take the time to try them, only to discover the same thing.

Examples of antipatterns include spaghetti code, big ball of mud and copy and paste programming. With spaghetti code, the code is very long. Like spaghetti, it can be difficult to find the other end of the code, and the code can be unorganised and messy. The big ball of mud antipattern refers to one which lacks any architecture. These types of systems are typically difficult to maintain. Copy and paste programming, refers to pasting the code in from another location to duplicate functionality. A more effective way to achieve the same objective is to create a method of the code instead, or to apply inheritance.