Here's an example of a software development anti-pattern.

Anti-Pattern

Just like a Design Pattern, an anti-pattern has a name so we can create a shared vocabulary.

Name: Golden Hammer

The problem and context, just like a Design Pattern description.

Problem: You need to choose technologies for your development and you believe that exactly one technology must dominate the architecture.

Context: You need to develop some new system or piece of software that doesn't fit well with the technology that the development team is familiar with.

Tells you why the solution is attractive.

Forces:

- The development team is committed to the technology they know.
- The development team is not familiar with other technologies.
- Unfamiliar technologies are seen as risky.
- It is easy to plan and estimate for development using the familiar technology.

The bad, yet attractive solution.

Supposed Solution: Use the familiar technology anyway. The technology is applied obsessively to many problems, including places where it is clearly inappropriate.

How to get to a good solution.

Refactored Solution: Expanding the knowledge of developers through education, training, and book study groups that expose developers to new solutions.

Examples:

Example of where this anti-pattern has been observed.

Web companies keep using and maintaining their internal homegrown caching systems when open source alternatives are in use.