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

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

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

Tells you why the solution is attractive.

The bad, yet attractive solution.

How to get to a good solution.

Example of where this anti-pattern has been observed.

---

### Anti-Pattern

**Name:** Golden Hammer

**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.

**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.

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

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

**Examples:**

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