

Anti-patterns



What is an Anti-Pattern?

A **pattern** is a named, proven approach to solving a technical problem in a context with generally positive consequences..

An **anti-pattern** is a named, common approach to solving a technical problem with decidedly negative consequences. Anti-patterns are like code smells on steroids:

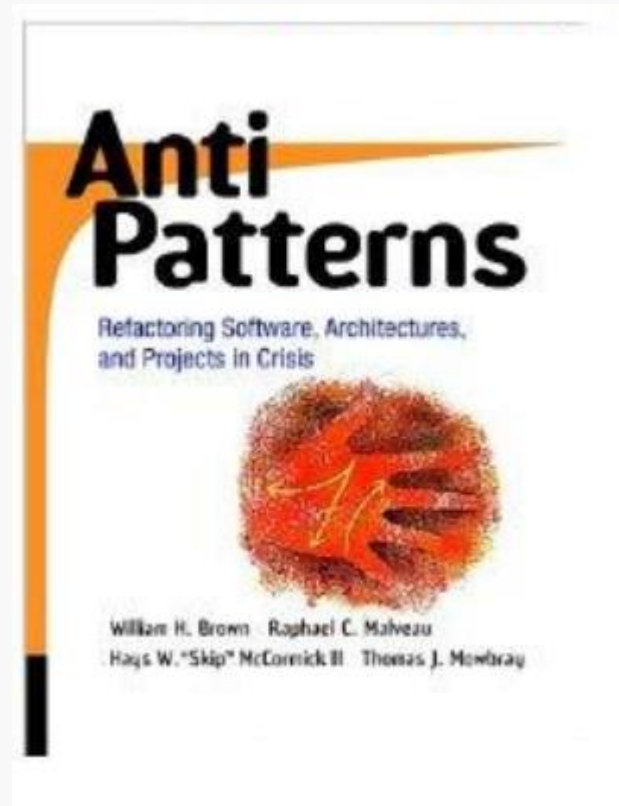
1. Generally at a higher level of abstraction than smells.
2. Generally of more strategic significance.
3. Generally have wider organizational impact.

Other examples: non-optimal solution; incorrect use of pattern or use of pattern in the **wrong context**

Key Anti-Pattern References

www.antipatterns.com

www.sourcemaking.com/antipatterns



Anti-Pattern Categories

Software Development Anti-Patterns

- Relate to maintaining solid code structure.
- **Solution:** targeted refactoring.

Software Architecture Anti-Patterns

- System and enterprise-level of application structure.
- **Solution:** Make architecture explicit (architecture driven development).

Project Management Anti-Patterns

- Communications and personal (team) issues.
- **Solution:** Varied.

Software Development 1 – The Blob

AKA: Winnebago; The God Class

Anecdotal Evidence:

“This is the class that is the heart of our architecture.”

Root Causes:

Sloth, Haste, Unfamiliarity With OO Technology

Solution: Refactor Responsibilities

- Extract Class
- Extract Method
- Move Method



Software Development 2 – Spaghetti Code

AKA: Winnebago; The God Class

Anecdotal Evidence:

“Ugh! What a mess!”

“It’s easier to rewrite this than to attempt to modify it.”

Root Causes:

Ignorance, Sloth, Inexperienced developers.

Solution: Refactor Responsibilities

- Refactoring.
- General cleanup.



Software Development 3 – Golden Hammer

AKA: Old Yeller, Head-in-the-sand

Anecdotal Evidence:

“I have a hammer; everything else is a nail.”

“Our database is our architecture.”

“Maybe Excel macros weren’t up to the job.”

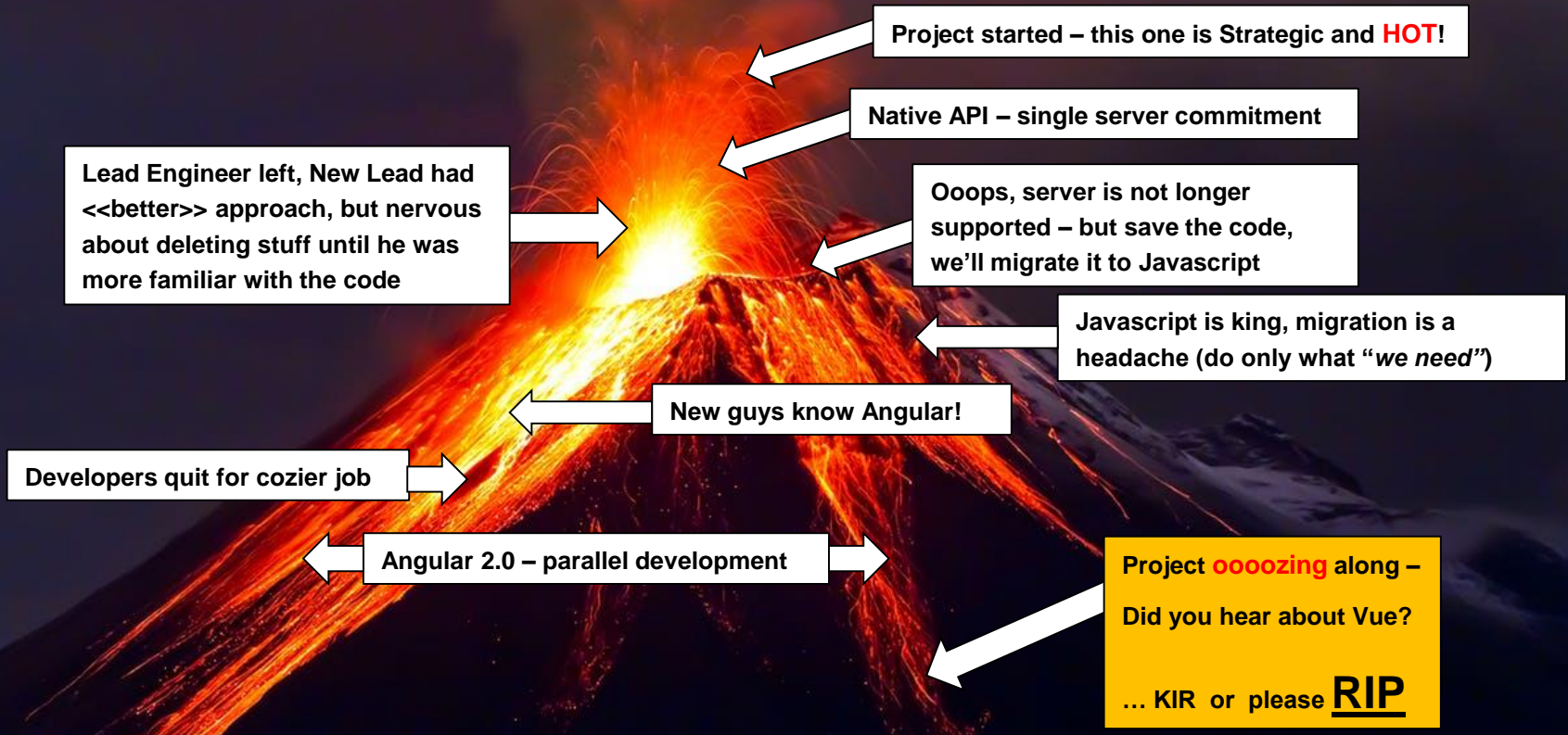
Root Causes:

Ignorance, Pride, Narrow-mindedness.

Solution: Refactor Responsibilities

- Commitment to exploring different technologies.
- Developer training.





Software Development 4 – Lava Flow

AKA: Dead Code, Ruins of Troy

Anecdotal Evidence:

*"Oh **that!** Well Ray and Emil (they're no longer with the company) wrote that routine back when Jim (who left last month) was trying a workaround for Irene's input processing code (she's in another department now, too). I don't think it's used anywhere now, but I'm not really sure. Irene didn't really document it very clearly, so we figured we would just leave well enough alone for now. After all, the bloomin' thing works doesn't it?!"*

Root Causes:

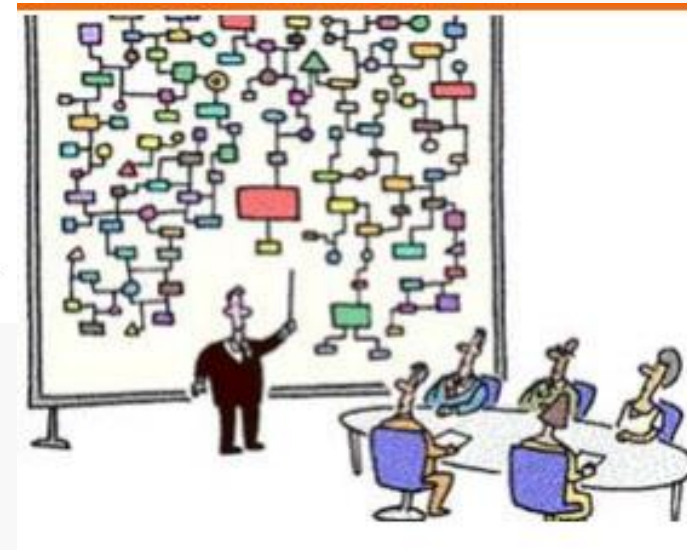
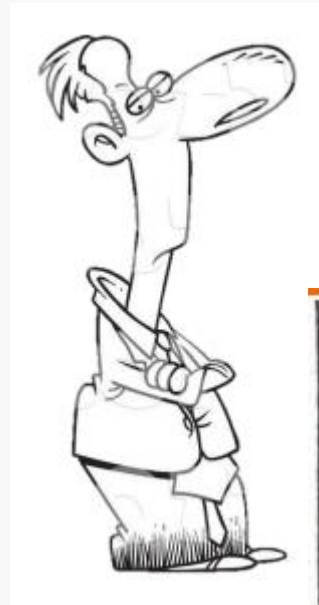
Sloth, FUD, failure to refactor, research to production.

Solution:

- Refactor
- Careful excision and testing.

Project management AntiPatterns

- **Blowhard Jamboree**
- Death By Planning
- **Intellectual Violence**
- Fire Drill
- **Analysis Paralysis**
- E-mail Is Dangerous



Remember much of it is up to you!

