

Pragmatic Software Development: Curing the Architecture Astronaut by Cory House

Email: housecor@gmail.com | Blog: bitnative.com | Twitter: [@housecor](https://twitter.com/housecor)

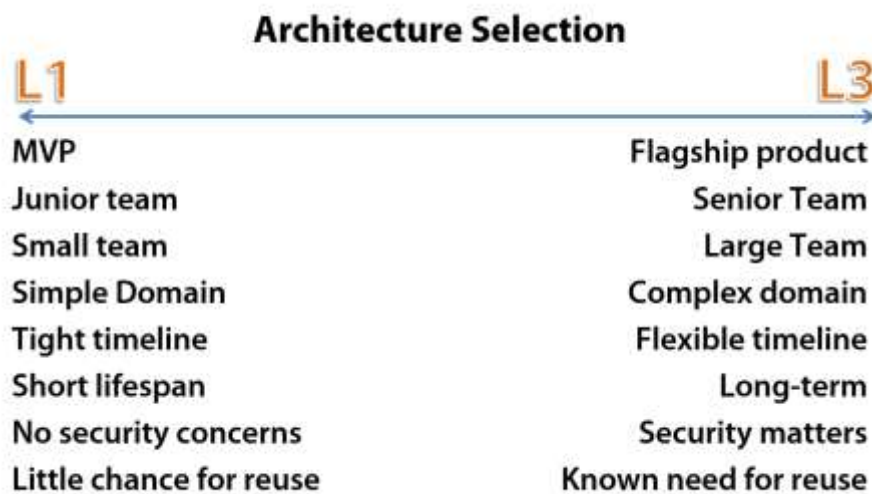
Pluralsight Course and Slides: pluralsight.com/author/cory-house

Please do me a favor and rate this talk: <http://speakerrate.com/talks/44931>

Core Principles:

1. Anyone can design a bridge that stands. It takes an engineer to design a bridge that barely stands.
2. Complexity must be justified.
3. In a small enough app, methods are all the layering you need.
4. Every “best practice” has a corresponding context. Consider metaphors:
Bike cops vs. cops in cars, Paint brushes vs Spray guns, Tape vs screws
5. Level 1 architecture is about doing the simplest thing that could possibly work. Consider Active Record.
6. Level 3 architecture is about quickly responding to change in large complex systems through:
 - a. Independence from DB, frameworks, and UI
 - b. The ability to test components in isolation
7. Migrate from L1 to L3 as complexity increases. Focus on the pain.

Considerations for Architecture Selection



Recommended Reading

- Patterns of Enterprise Application Architecture by Martin Fowler <http://bit.ly/enterprise-patterns>
- Architecting Applications in .NET by Dino Esposito <http://bit.ly/dotnetarch>

Is Your Team Writing Clean Code?

I offer on-site training sessions on Clean Code and Software Architecture. Session length varies from 1 hour to a full day.