Grokking Libraries in Clojure-land

Feat. Component & Mount

Aditya Athalye
evalapply.org

Agenda

- Choosing libraries, in general
  - Component, Mount merely examples
- Focus on design tradeoffs
  - Hardest thing about choosing a lib
- Learn from each other, grow tribal knowledge pool
  - ~ 20 mins PPT, and then open floor to discuss

You Don’t Know You Have A Problem

Maybe There Is A Problem

Boring ol’ business app

Config / Config Store

DB

Cache

Object Store

Queue

Other Services

3rd party APIs

Logging & Monitoring … … …

Naah, it's fine

M-x cider-restart …

Actually, you have many problems

You don’t know them all.

Yet...

Start (transactional? degraded?)

Stop cleanly

Deterministic start/stop order

Restartless development

Partial start/stop/restart

Dev / Test context isolation

…

Internet people:

“There’s a library for that…”

Seven Stages of Library Selection

Denial

One

I don’t need another library, dang it. Atoms are *fine*.

Anger

Two

Atoms everywhere!

Wait which one to modify?

Fuuuuuu...

https://commons.wikimedia.org/w/index.php?curid=2583261

Bargaining

Three

Hmm, my problem isn't very complicated. I could just…

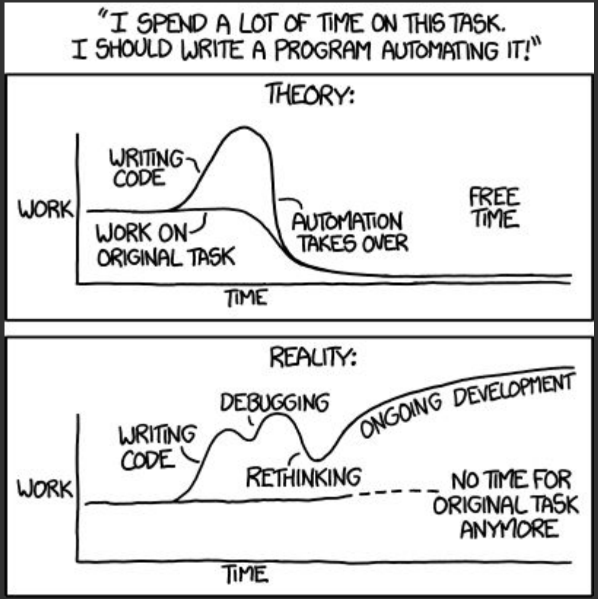
```
(ns core.utils.system …
  )
```

Depression

Four

Oh I didn't think about that...

Or about that… Or that…



https://xkcd.com/1319/

Acceptance

Five

Ok, I guess I do really need a library.

Angry Depressive Despair

Six

Which one????

AAAAAAAArrrrggghhh…

Hello, friend.

Seven

Ask around, and trust someone's opinion.

“But, why???”

Context is king. No one-size-fits-all solution.

Parameter	Library A	Library B	Library C
License			
Feature Set			
Maturity			
Documentation			
Support			
Size			
Performance			
Dependencies			
Design Tradeoffs	<<< HARD!		

Problem Definition

Articulate for self…

What’s the lib trying to solve?

History

What came before?

Alternatives?

How did community thinking evolve?

How does it work?

Documentation

“Meta-dot” and skim-read

REPL experiments

Using IDE features (summary, navigation)

…

Clone & pore over

My design goals?

Things I want to ease…

Things I want to mandate…

Things I want to prevent…

Things I want escape hatches for…

… by design.

Taking it Apart

What is implicit?

What should be explicit?

What’s the interface?

Where’s the complexity?

Taking it Apart

A “System” is essentially…

- Global composite object ("system") composed of units ("component")
- Set of protocols that suggest access/update semantics of states
- Implementation of protocols for the type of state (unit and composite)
- Pre-defined order to "start" and "stop" composites
- Maybe object registry of multiple systems inhabiting single runtime
- Query live state: active parts, order of start/stop for each, and current state of components within each system.
- Experiment with it...
    - https://gitlab.com/nilenso/cats/-/commit/cc52fad8a922729932ec73c7e7173a4aca747c06

Fin