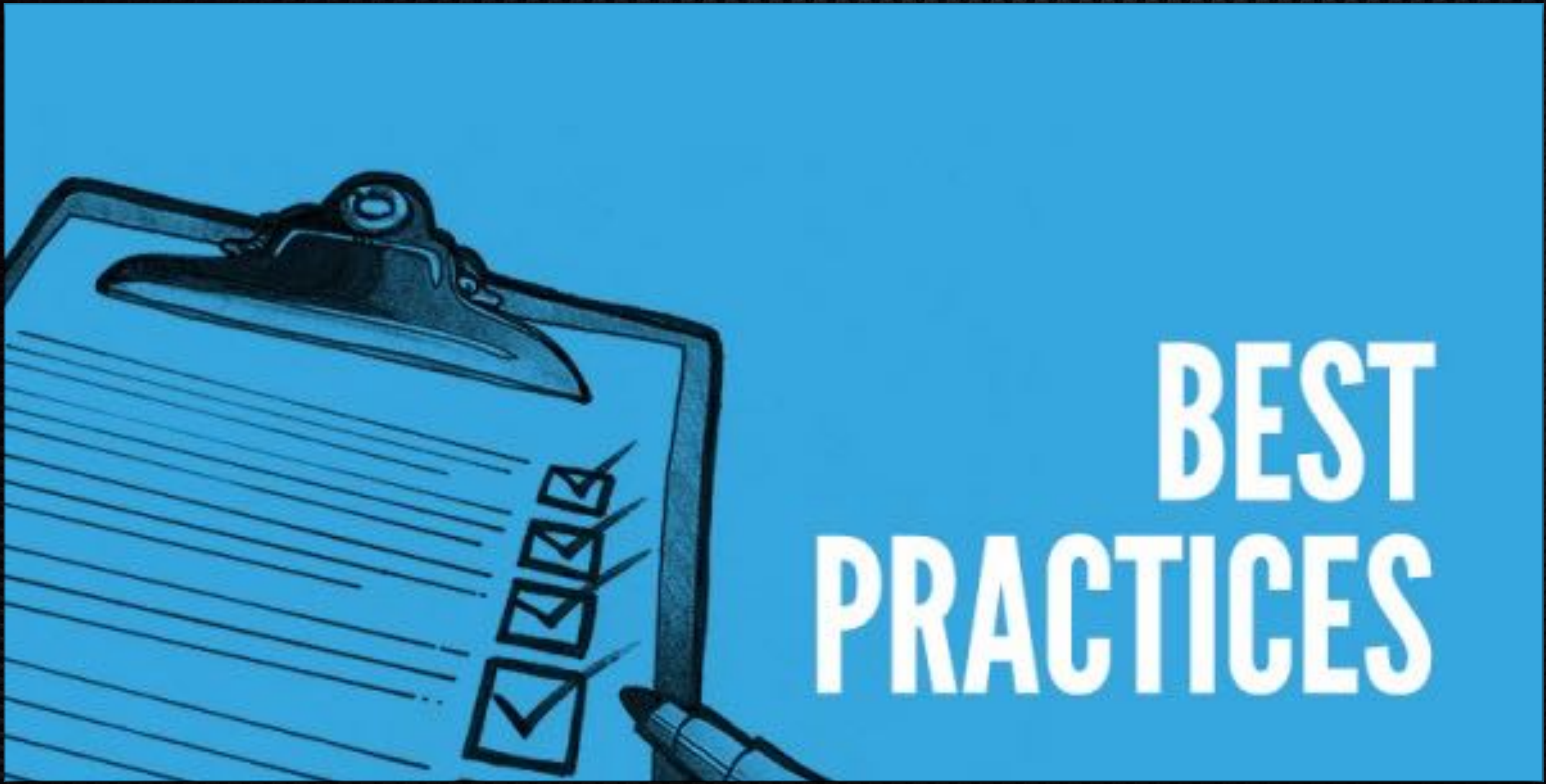


…Are the Worst

Why Would You Care What I Have to Say?

- Senior Partner of Integrity Inspired Solutions
- Done actual agile software development, project management, and strategic planning in a lot of organizations
- This stuff comes from successes, failures, and repeatable victories once we got stuff figured out

How We Get Into Trouble

Dogma: not just a puppy’s mother

Janelle Klein

- Idea Flow
- Complete regression test coverage, all passing
- First release brings entire factory down
- So does the second
- The third one, months later, also needed fixing

Hold That Thought

Dogmatic Approach

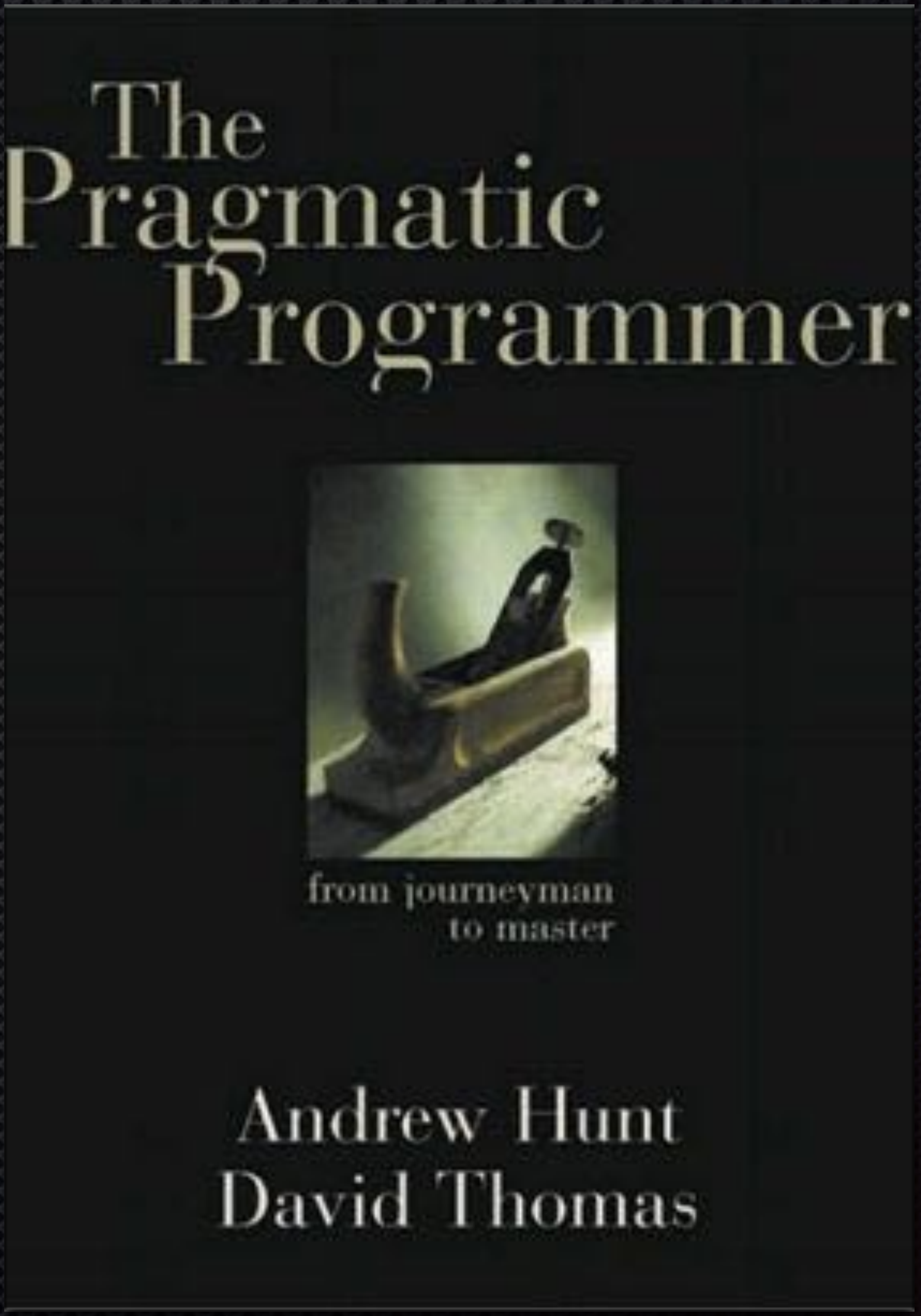
This way is right, and that way is wrong, regardless of circumstances.

Dogmatic Approaches…

- Rely heavily on standards / best practices / authority
- Oppose changes to the recommended practice
- Do things “by the book”
- Assert this is what everyone should be doing
- Start with solutions and assumes problems will go away or be prevented if we do them

Pragmatic Approach

The right way is determined by the best way to get to the desired outcome given the present situation.



Pragmatic Approaches…

- Subject standards, best practices, and authority to what will work best in a given situation.
- Freely change practices if it makes them work better.
- Use “the book” for inspiration, not law.
- Assert this works best at this time for this situation.
- Start with the problems and uses them to determine which solutions to pursue.

“I had never seen a team come together quite like ours did after this tragedy. It didn’t matter anymore whether our process was the ‘right way’ to do things. We had to learn how best to fix things for our project and our team.”

–Janelle Klein, Idea Flow, p. 3

Find Your Actual Problems

Measurement, Collaboration, and Shining a Light into the Darkness

Fun Measurement Stories

- “We’re too slow because the users keep changing things.”
- “We need to get more features done. We need more developers.”

“Don’t assume you know something is a problem, and don’t assume you know you need a particular benefit. If you think something might be a problem, come up with an experiment to measure it.”

–Phil Ledgerwood, this presentation, slide 12

What We Want to Know

- The impact of a problem
- The impact of a benefit
- It isn’t good enough to simply assert a problem needs fixing or there’s a benefit to be had.
- We want to identify problems we actually have and benefits we actually need.

Measurement Examples

- How long does a user story spend in each stage of the development cycle? (Cycle time plots)
- How many user stories do we have in each stage? (Cumulative Flow Diagram)
- How much time is a developer spending in meetings? Being pulled off onto other things? Fixing broken tests? Fixing bugs? (Idea Flow Diagram)

More Measurement Examples

- Putting a user in front of a prototype and watching them go
- Which web pages get hit the most and what is the response time?
- How long are the Sprint Planning meetings compared to the value those meetings produce?

Finding Solutions

One Size Does Not Fit All

The warmer weather will allow me to get more exercise by walking to bars.

“Under what circumstances would you not recommend that?”

–Phil Ledgerwood, annoying his co-workers, regularly

Doing the Right Thing

- We have identified a problem we actually have or a benefit we actually need.
- We must choose a solution that will work for us and must do it in a way that works for us.
- The cost/effort of the solution must not exceed the value of solving the problem or having the benefit.

Retrospectives / Kaizen Events / Continuous Improvement

- My one must-have agile item: you must have a way for the team to regularly inspect and improve themselves.
- (unless this happens regularly and naturally in the workflow)
- What I like to do:
    - Go over our measurements
    - Brainstorm potential areas of improvement and pick one
    - Brainstorm potential solutions and pick one

"Our test suite didn't look like Mike Cohn's test pyramid at all, the bulk of our tests were component tests. Success wasn't a matter of creating the right number of tests, the right type of tests, or hitting the right code coverage targets. We took the time to understand our problems, then found solutions that worked."

–Janelle Klein, Idea Flow, p. 52

A Note of Caution

- Agile practices often challenge us in ways that we need to change. Don’t mistake “temporary pain because we suck at this” for “not working for us.”
- Don’t change an agile practice until you understand why it was designed that way and why your change will better deliver that same value.