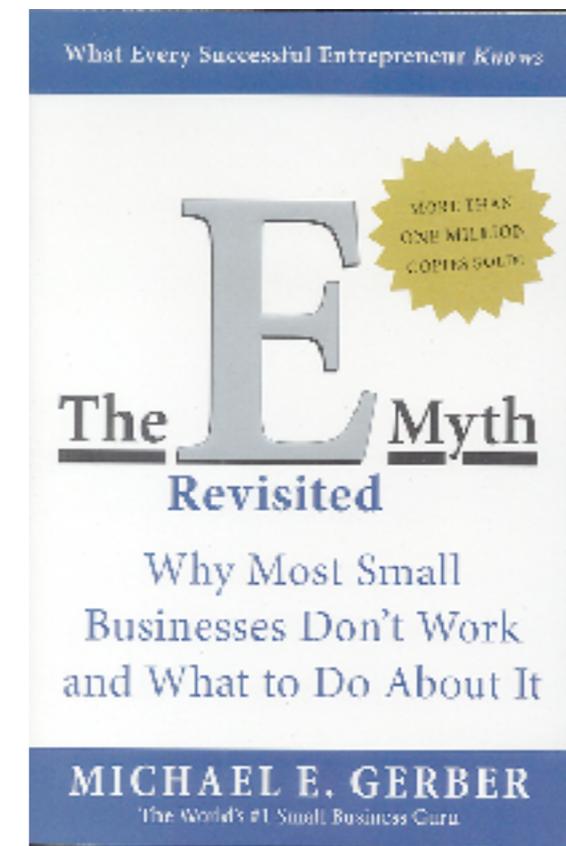
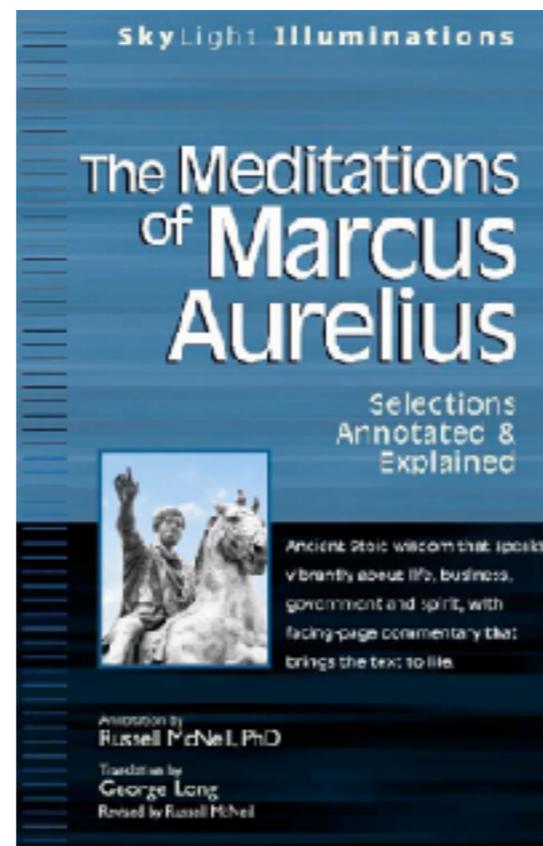
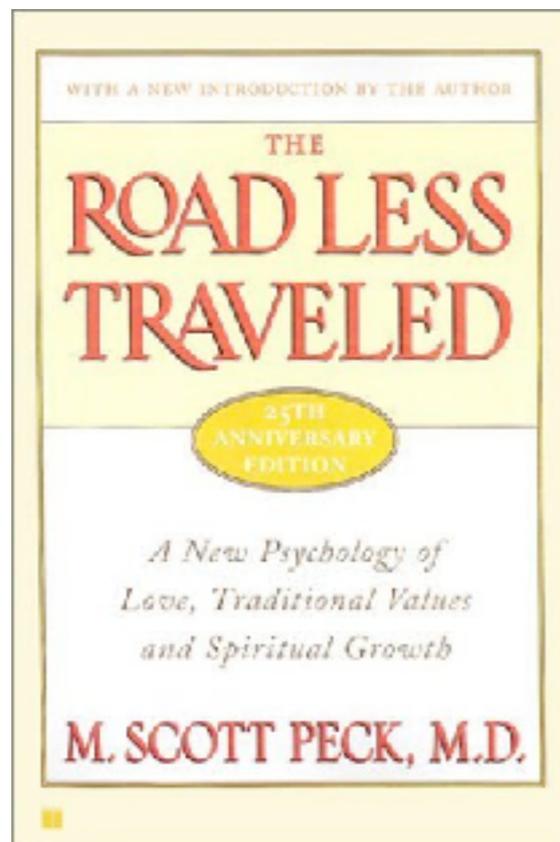


# Your Framework Sucks

Paul M. Jones

# Read These

- “The Road Less Traveled”, M. Scott Peck
- “Meditations”, Marcus Aurelius
- “The E-Myth Revisited”, Michael E.



# About Me

- Developer, Senior Developer, Team Lead, Architect, VP Engineering
- Military, education, non-profit, startups, products, consulting
- ZCE Education Advisory Board
- PHP since 1999
- Business, Religion, Military, Philosophy, Economics



# Why Bother With A Public Framework?

- They are “bloated” and “slow”, right?
  - Rasmus’ “no-framework” framework
  - Paul’s web framework benchmarks

# Speed Relative To Plain PHP

<i>Target</i>	<i>relative</i>	<i>average</i>
<i>baseline/html</i>	1.2669	2915.01
<i>baseline/php</i>	1	2300.95
<i>aura/dev-cached</i>	0.1327	305.37
<i>cake/1.3.10</i>	0.1141	262.58
<i>ci/2.0.2</i>	0.2637	606.78
<i>kohana/3.1.3.1</i>	0.2433	559.74
<i>lithium/0.9.9</i>	0.1077	247.75
<i>solar/1.1.1</i>	0.1682	387.13
<i>symfony/2.0.4-fp</i>	0.1601	368.44
<i>zend/1.11.9</i>	0.1024	235.61
<i>zend/2.0.0beta1</i>	0.0554	127.55

# Anecdotal Assertion

- **The apparent short-term strength of building your own likely to be a long-term weakness**
- Time spent learning could be time spent writing own... at least then you know it well.
- You may know your system inside and out, but nobody else does
- Might be better to adopt someone else's

# Business Franchise

# Business Format Franchise

- Buying a franchise == license to a ruleset
- Sophisticated manual about every aspect of business operations
- Owner not a cog in corporate machine ...
- ... independent operator of shared plan

# Franchise Benefits

- Standardized, consistent customer experience
- Predictable, reliable operation
- Familiar with one == familiar with all

# Manual Creation

- Original owner(s) build one or more businesses
- Extract rules for operation and record in detail
- Multiple iterations: keep the wins, discard the losses
- “Franchise Prototype” is the product -- jump-start to operations

# New Business?

- Advice to entrepreneurs:  
**Build “as if” you are going to franchise**
- “... define and write down every aspect of his operations, so that anyone else can do the same work the same way in a predictable and reliable manner”

# The Framework As Franchise

# Framework Definition (1/2)

- Loose terminology: “collection of tools, techniques, and procedures”
- Internal and private, or public and shared
- Informal collection from disparate sources, or well-integrated with formal structure

# Framework Definition

## (2/2)

- Almost any regular development practice used by an organization can be called a framework
- The framework is the set of implicit and explicit rules for how to build applications

# The Framework As Franchise (1/2)

- Sophisticated ruleset
- Built iteratively over many projects
- Common operations and environments
- Predictable development process
- Jump-start to development process

# The Framework As Franchise (2/2)

- Franchise: well-defined way of working to build an organization
- Framework: well-defined way of working to build an application
- Buying into a “way of working”

# Reducing Cognitive Friction (1/2)

- Coding style guides make developer code formatting look the same
- Style guide itself not as important as the fact that all code adheres to it
- Framework itself not as important as making sure everyone uses it for decision-making and organization ...  
“world view”

# Reducing Cognitive Friction (2/2)

- Using a framework is about consistent development process in/across teams
- Standard foundation for decisions about application architecture
- All developer decisions have common themes, because all using same rules

**All Frameworks Suck**

# ...Well, Not Exactly

- Laura Thomson: “All frameworks suck.”
- Paul M. Jones: “All frameworks *by other developers* suck.”
- By definition, your personal way of working sucks *to other people*.

# Picking A Framework

- On your own? Do what you like.  
 (“Personal framework.”)
- On a team? Must coordinate and cooperate with other people. “Personal” isn’t good enough ...

# Solution Attempts (1/3)

- Work with team to build framework together
- You know your own unique situation better than anyone else
- Problems? Well, they're "known."



©2005 JESSICA AND JOHN WILLIAMS

# UNIQUE

JUST BECAUSE YOU ARE UNIQUE DOES NOT MEAN YOU ARE USEFUL

# Solution Attempts (2/3)

- Your situation is likely *not* that unique
- [http://www.phpwact.org/php/mvc\\_frameworks](http://www.phpwact.org/php/mvc_frameworks)
- [http://en.wikipedia.org/wiki/Comparison\\_of\\_Web\\_application\\_frameworks#PHP](http://en.wikipedia.org/wiki/Comparison_of_Web_application_frameworks#PHP)
- Group into a few major “ways of working”
- Your framework will resemble one of them

# Solution Attempts (3/3)

- Design patterns? Implementation patterns.
- Pick a public framework that suits your way of working as a jump-start
- Tradeoffs: suckage not of your own design
- Coordination/cooperation issues already worked out
- What seemed unique has been regularized

# Resistance

- Research sucks: reading/learning much less fulfilling than writing/producing
- Narcissism of small differences
- Be honest with yourself

# Summary

- Frameworks are like franchises (rule sets for ways of working)
- Other people's rules suck
- Learn your personal way of working, without self-deception
- Give up your own small differences to gain greater effectiveness of shared world-views

# Thanks!

- [paul-m-jones.com](http://paul-m-jones.com)
- [twitter.com/pmjones](https://twitter.com/pmjones)