Software Construction, Renovation,
& Maintenance

My Top 3 RTFMs

Jim Ladd

July 26, 2024

Recently, I was in an Agile sprint retrospective meeting with a client's development team, and the overall direction was toward an *even more* waterfall process than what was being followed. The only real similarities with Agile were the name and a time-based cadence. I was disappointed, however not surprised, how little the team understood the core concepts of the methodology.

I began to reflect how few followers of Agile, including certified scum masters, have read the cornerstone of the methodology. Even during my certification process, this document was never mentioned or referenced.

One of my mantras in both my professional career and personal journey is to seek out the headwaters for the topic of interest. I try very hard not to get trapped in a vacuum or, worse, an echo chamber. When I learn about an emerging technology or process, I strive to understand the origins, the advantages and the risks.

Over the years I have collected the source documents for a wide range of technologies, processes, and practices. By frequency of recommendations, here are my top three RTFMs:

Agile Development

The Agile methodology emerged during the early 2000's from the collaboration of a group of industry thought leaders. Today, practically everyone, at least everyone in my world, follows a version of Agile. However, few embrace the spirit of Agile which is best expressed in the *Manifesto*. I highly recommend *to learn it, love it, live it*.

Kent Beck, et al. (2001)

Manifesto for Agile Software Development

https://agilemanifesto.org/

REST API

The last of the large hold out vendors are finally converting their public APIs to the Representational State Transfer (REST) style. It's now a de facto standard in current systems. While REST is not new nor complex, I'm surprised how few developers using these APIs have even a mid-level understanding of the technology. The origins of REST can be traced to a PhD dissertation by Roy Fielding. Not only is the section on REST worthwhile, anyone designing software should read the entire document and then read it again.

Roy Thomas Fielding (2000)

Architectural Styles and the Design of Network-based Software Architectures

https://ics.uci.edu/~fielding/pubs/dissertation/fielding_dissertation.pdf

Bitcoin

In 2008, a paper describing a peer-to-peer network called Bitcoin was released by Satoshi Nakamoto. While Bitcoin is now a household name and the legend of the origins grows, I highly recommend the document to both software professionals and the general public. It's a very straightforward read and I tip my hat to the person(s) behind the pseudonym.

Satoshi Nakamoto (2008)

Bitcoin: A Peer-to-Peer Electronic Cash System

https://bitcoin.org/bitcoin.pdf