# WHAT IS THE INTERSECTION BETWEEN AGILE & TRADITIONAL SYSTEMS ENGINEERING?

***That's a question people have been pondering for years*** (*with multiple levels of complexity ranging from the simple and practical to deeply philosophical similar to asking, "What's the meaning of life?"*) — ***Let's examine SOME possible responses*** ...

1. (*Simply*) **Definitely consider the SAFe methodology** (*for such purposes*):

- SAFe Overview (http://davidfrico.com/rico15m.pdf).
- SAFe Textbook (http://www.amazon.com/dp/0321635841).
- SAFe Website (http://www.scaledagileframework.com).

2. (*Also Simple*) **Consider some of the emerging paradigms** (*Continuous Integration, Continuous Delivery, & DevOps, etc.*):

- Continuous Integration, Continuous Delivery, and DevOps.
- Google & Amazon (http://davidfrico.com/google-agile-testing-roi.pdf & http://davidfrico.com/amazon-agile-delivery-speed.pdf).
- Continuous Delivery & DevOps Resources (http://davidfrico.com/devops-resources.txt & http://davidfrico.com/devops-roi-reports.txt).

3. (*Practically*) **Consider some of the Lean-Kanban approaches** (*that are also emerging specifically for systems engineering*):

- Most Popular Lean Book for SE (http://www.amazon.com/dp/1935401009).
- INCOSE Lean Enablers (http://www.lean-systems-engineering.org).
- Various Kanban Resources (http://davidfrico.com/kanban-resources.txt).

4. (*Importantly*) **Consider SHIFTING from hardware to software** (*to enable iterative, emergent, and evolutionary design*):

- Agile for Embedded Systems (http://davidfrico.com/agile-for-embedded-systems.pdf).
- Agile for Cloud Computing (http://davidfrico.com/agile-for-cloud-computing.pdf).
- Agile for Hardware Development (http://www.amazon.com/dp/1578517508).

5. (*More Complex*) **Consider models of customer interaction, innovation, and idea generation** (*which are very important*):

- Customer Active Paradigm (*by Eric Von Hippel*).
- Value Focused Thinking (*by Ralph Keeney*).
- Innovation (*by Keith Goffin & Rick Mitchell*).

6. (*Getting Deeper*) **Consider some of MY writings on this topic**:

- Agile Cost of Quality (http://davidfrico.com/agile-vs-trad-coq.pdf).
- Agile vs. Traditional Assumptions (http://davidfrico.com/agile-vs-trad-assumptions.pdf).
- Lean & Agile Systems Engineering (http://davidfrico.com/reagan-rico-atl.pdf).

(*Consider Lockheed's agile systems engineering legacy*, http://www.amazon.com/dp/0316743003 & http://www.f-117a.com/Rules.html ...)

7. (M*ost Complex*) **"What is [the definition of] systems engineering?"** (*philosophically AND in the context of WHAT you are doing*):

- Is systems engineering the art & science of building complex systems of systems (i.e., *the most complex human-made system is the Internet • Some view it as a medium for cybercrime and pornography • Others view it as fundamental to social and political revolution, liberation, and global integration • See "Architects of the Web" by Robert Reid & "Social Physics" by Alex Pentland*)?
- Is systems engineering the art & science of building mission & safety critical human-rated systems (i.e., *see "Into the Black" by Peter Westwick, "Day Phones Stopped" by Leonard Lee, "Developing Safety-Critical Software" by Leanna Rierson, "Safeware" by Leveson, etc. • Also Google "Agile for Safety Critical Systems" • Microsoft's security engineering model is also profound, see the green Bidstrup briefing in this zipfile*, http://download.microsoft.com/download/0/1/a/01a053e8-3e18-4f73-b8e7-68d53a8232da/Bidstrup-Kowalczyk_SSW-2005.ppt)?
- Is systems engineering the art & science of modeling & simulation (i.e., *If you're a day trader, What's the range of investment models you can apply? If you're an astrophysicist, What's the range of spectral analysis models you can apply? If you're a civil or mechanical engineer, What's is the range of CAD/CAM/CAE models you can apply? If you're a manufacturing engineer, What's the range of 3D printing you can apply? If you're an embedded systems manufacturer, How far can you shift algorithmic processing from hardware to software? If you're an environmentalist or meteorologist, What's the range of predictive models you can apply?, etc. — **See*** http://davidfrico.com/quant-oath.pdf)?

(*Many people are fascinated by 3D printing to RAPIDLY manufacture product designs • However, it's useless to rapidly manufacture the wrong product without customer feedback • Google Analytics and Social Physics is an innovative way to help predict market trends FAR IN ADVANCE of just rapid manufacturing • If you're an embedded systems manufacturer, then SHIFTING FROM HARDWARE to SOFTWARE is the way to go • Again, See Agile for Embedded Systems,* http://davidfrico.com/agile-for-embedded-systems.pdf *• BOTTOM LINE, the goal is NOT rapid rigor and discipline, but getting a product increment to a customer so they can define the real valid requirements • That is, let the customer's PULL what they want, don't PUSH unneeded, unwanted, or useless CRAP on your customers • If you try to PREDICT the market and then PUSH crap on the market, two things can happen • ONE, you can go bankrupt using a large and bureaucratic systems engineering process, or TWO, you won't get any revenue, sales, profit, or ROI, if the market doesn't need what you thought they wanted • You only WIN if the market wants, buys, or needs what you have to produce • This is what makes SOFTWARE applications so STRATEGIC and TACTICAL • I can form a focus group of lead customer end-users, ask them what they want, rapidly model their requirements in software, get their early feedback without breaking the bank, complete a final version in a reliable way, and then RINSE-and-REPEAT ...*)

P.S. Focus on SAFe (http://davidfrico.com/rico15m.pdf) • Shift from hardware to software (http://davidfrico.com/agile-for-embedded-systems.pdf) • Read this article (http://davidfrico.com/agile-vs-trad-coq.pdf) • Go deeper with Kanban and Don Reinertsen's book (http://www.amazon.com/dp/1935401009) • Shy AWAY from PMBoK, INCOSE's SE Handbook, CMMI, etc. • Look into the Continuous Delivery and DevOps phenomenon • Go to the Velocity Conference in Silicon Valley to see what the Internet giants are doing • Try to catch a DevOps webinar, Agile Meetup, IEEE meeting, etc. • Don't focus on quality from a traditional perspective • Don't fixate on speed alone • Figure out a way to get customer input and feedback on a series of rapid models, simulations, prototypes, beta versions, and incremental releases of simple SOFTWARE-INTENSIVE products and services • Feel free to reach out, because I've collected a ton of resources on this topic ...