Better Software

Software Certification Consortium

The Road to a Cyber-Physical System with Designed-In Assurance

26 January, 2015

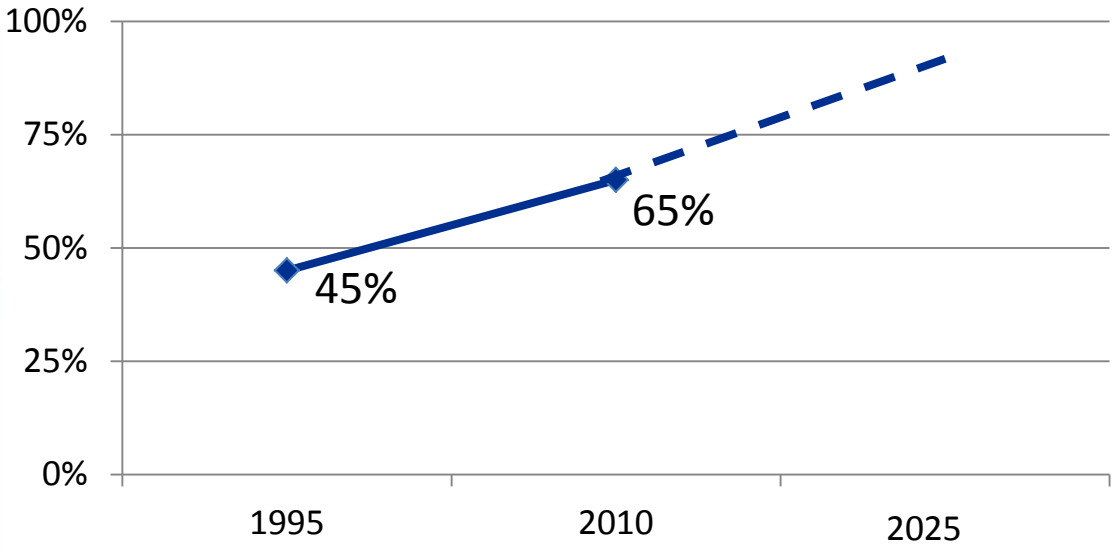
Michael May, Ph.D.
Associate Director for Software
Office of the Assistant Secretary of Defense
for Research & Engineering

What's Wrong With Software?

It's simple.

Really?

#1 Costs Too Much

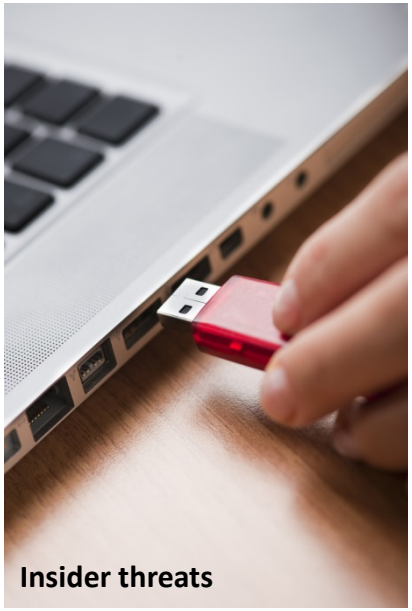


Based on industry figures from Ward, D., Helton, S., *Estimating Return on Investment for SAVI (a Model-Based Virtual Integration Process)*, Proceedings of SAE International Aerospace Technology Conference, 2011.

Software is Growing as a Percentage of Total System Cost

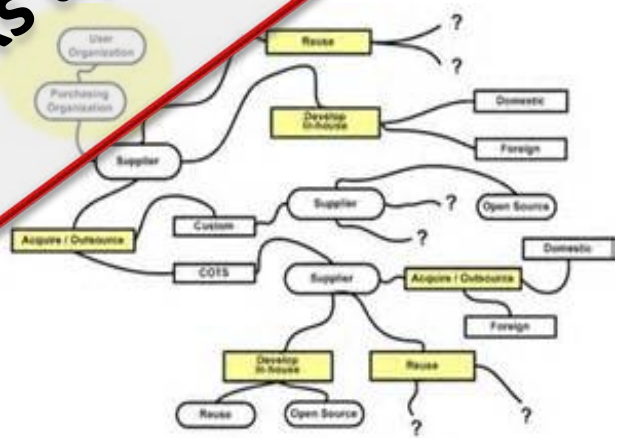
For complex DoD systems, we get as little as 6 lines of code per person per day when you count labor from requirements through verification testing. (Analysis by the Software Engineering Institute)

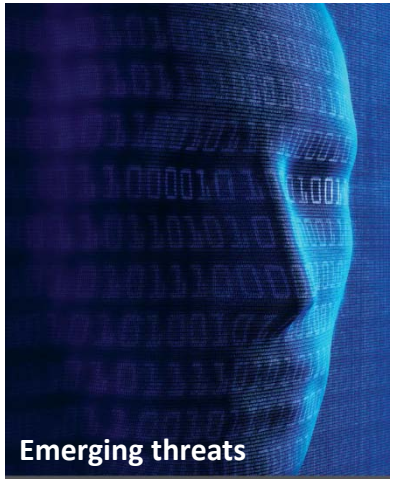
#2 Interconnectivity Increases Cyber Risk


Insider threats

Study: Cyberattacks up 48 percent in 2014
Thehill.com


Legacy system upgrades


Complex software supply chain



DoD Feels the Pain

- **DoD knows what's wrong with software**
  - Issues in software development cause schedule slippage and cost growth
  - Examples of programs with software issues: AARGM, B-2EFH SATCOM, JSF, FAB-T, GPS OCX, JLENS, JPALS, AMF JTRS, KC-46, BAMS UAS, SBIRS High, B-2 DMS, DDG 51 Destroyer, JTRS GMR, MEADS CAP (*GAO Assessments of Selected Weapon Programs, GAO-12-400SP*).
- **DoD Software Challenges**
  - Rapidly-evolving, complex operational environment
  - Requirements are increasingly met through software
  - DoD must make advances to respond to adversaries
  - No sign industry is addressing the challenges; looking to DoD to lead
- **Software Assurance**
  - "The software functions as intended, and only as intended."
  - RFI: 2013 on Assurance

DoD Software Assurance: Nov 2013, Request for Information

Can DoD help incentivize a public-private partnership to ensure software works as intended, and only as intended?

Responses from biggest players in defense and non-defense industry, small companies, consortia, other Government Agencies

- Pointers to existing assurance efforts: NIAP, NIST, DHS, SAFECode, etc.
- Offers of innovative, but proprietary, solutions
- Protection of proprietary source code dis-incentivizes third-party analyses
- Reminder: one size never fits all — don't over-prescribe the solution

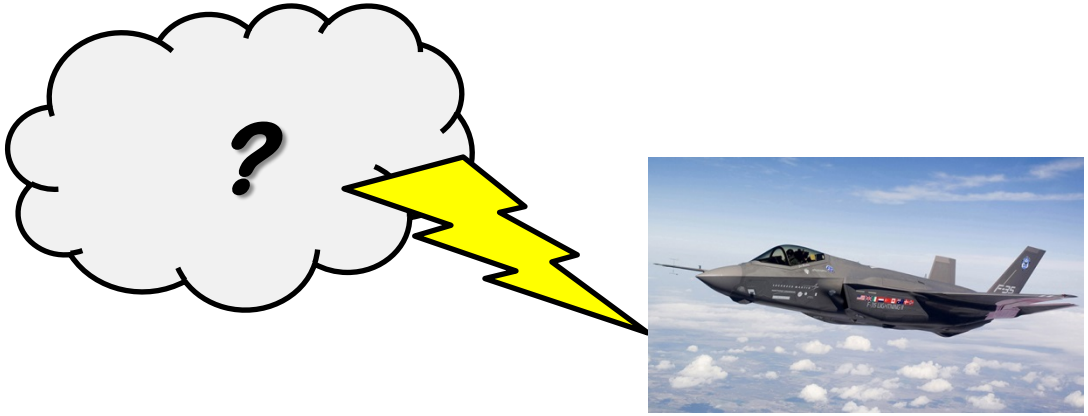
Back to First Principles

What are we trying to do? No kidding.

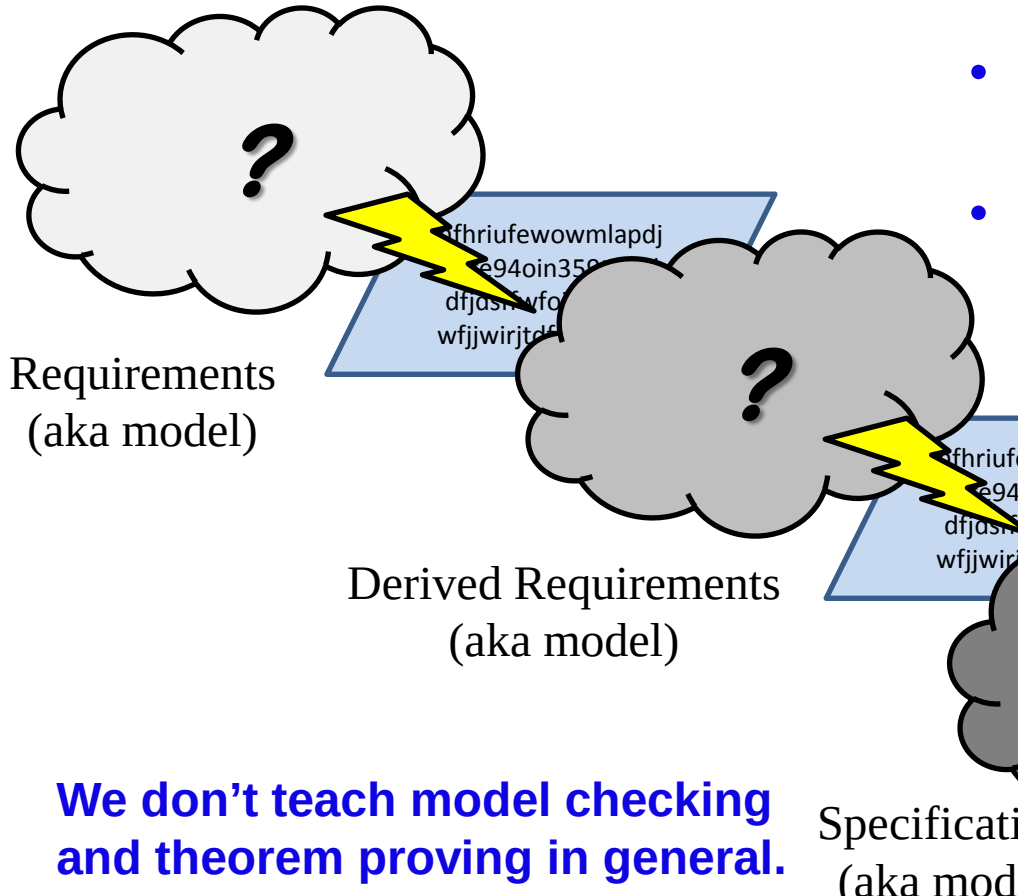
Here's a requirement…I want running code that meets the requirement.

What would that look like? Again, no kidding.

Here's a perfectly well-defined, verified-correct, complete requirement; and an error-free programmer and compiler with perfect knowledge of the requirement and computing platform.



What do we actually do?

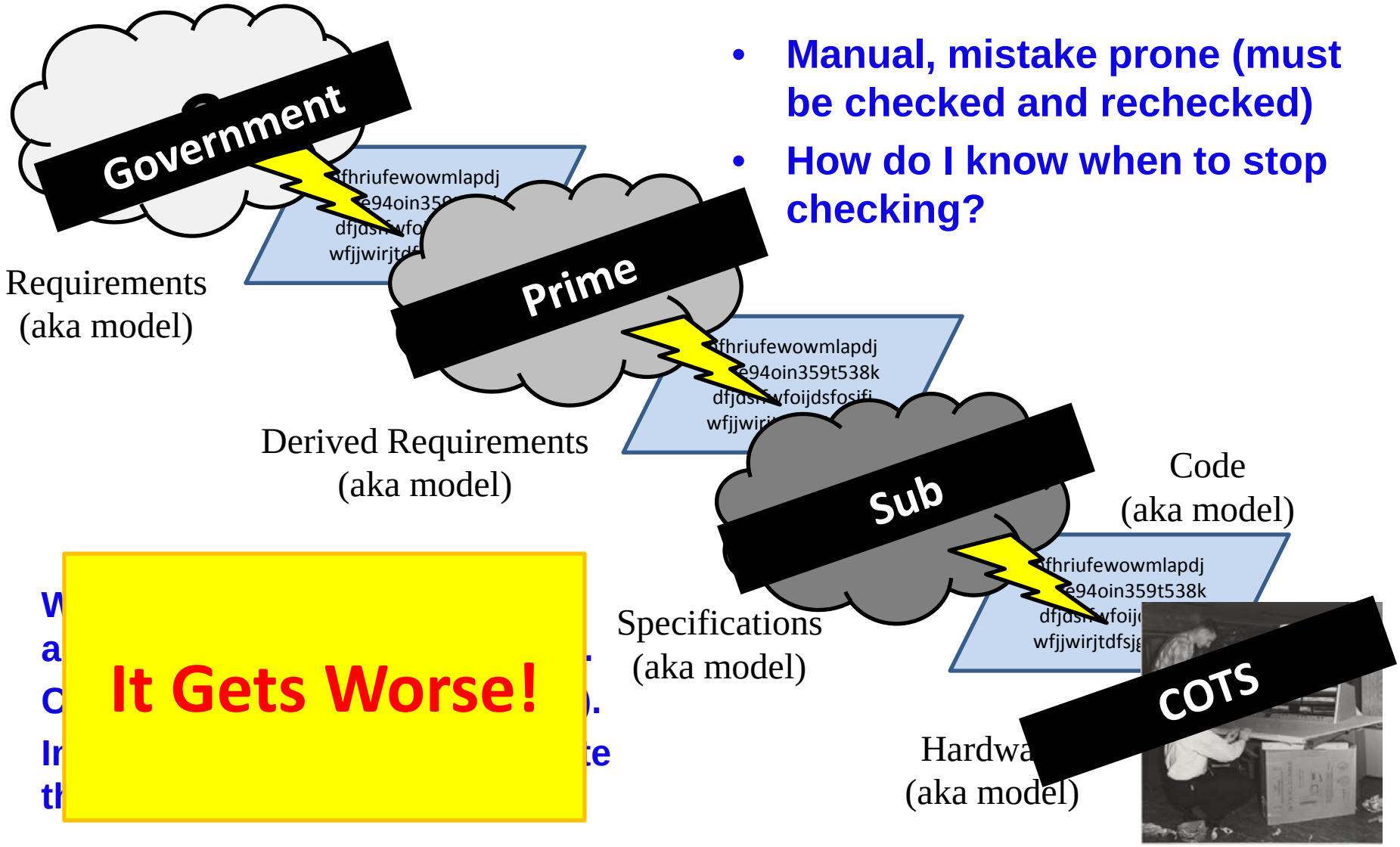


- **Manual, mistake prone (must be checked and rechecked)**
- **How do I know when to stop checking?**

- **We don't teach model checking and theorem proving in general.**
- **Considered a special skill (PhD).**
- **Imperative languages exacerbate the problem.**

What do we actually do?

- **Manual, mistake prone (must be checked and rechecked)**
- **How do I know when to stop checking?**

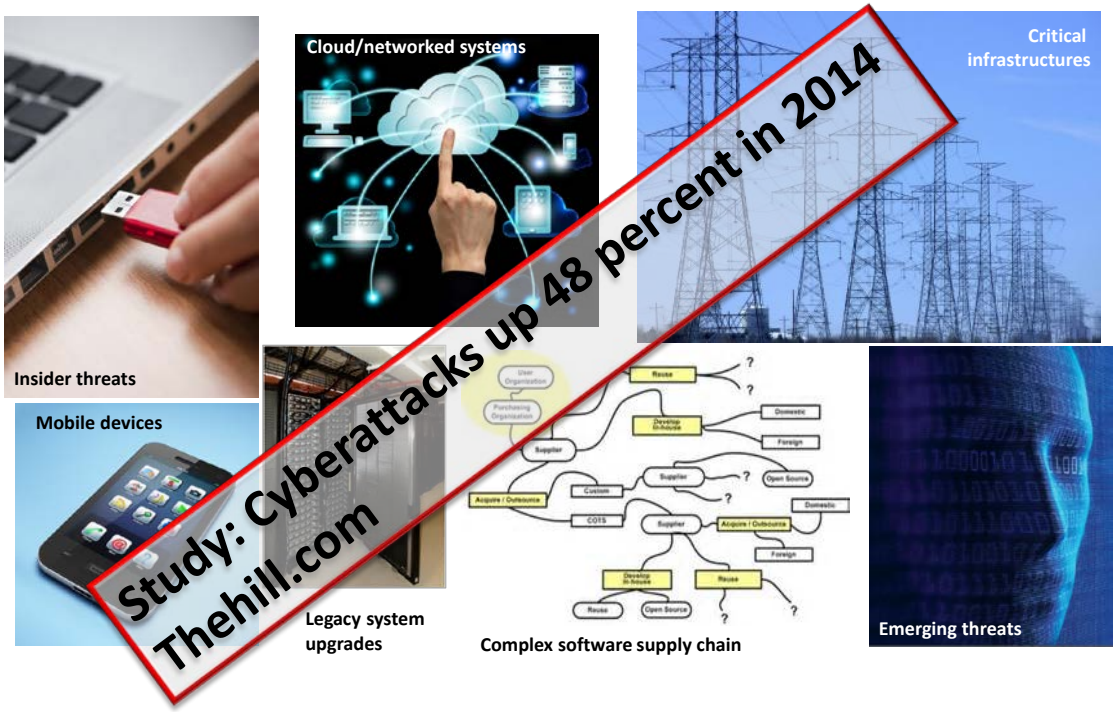


- W
  a
- C
- In
  th

It Gets Worse!

Why are we in this condition?

- History (and cost) drove us to abstractions.
- They were much more useful than an Electrical Engineer with punch cards, and infinitely more portable.
- This was fine, until…



How Do You Fix Software?

It’s simple.

Not Really.

What's the Future

- **Model-driven development**
  - Requirements-to-runtime, vertically integrated, mathematically proven.
  - Yes, this is hard. So, you can cheat and compose modules.
  - Business process for this is unknown. What's earned value management when you are correct by construction?
  - People need to do more of what people are good at.
- **That's it? Are we done? No.**
  - Correct by construction changes the conversation toward a solution people can comprehend/manage: "What do you mean by correct?"
  - Machines do the hard part.
  - What are the new tools? What does this automated "software assembly line" look like? What's the market?
  - How do we protect Intellectual Property?
  - How do we choose composable parts that scale?

How do we get to the future?

Put Humpty-Dumpty together?

- **Demonstrate that it can be done (research)**
  - Software Engineering Institute; AVSI-SAVI; DARPA, Service Labs, Industry, Academia
  - "Engineered Resilient Systems" DoD's physics-based model-driven development effort. (Here's the physical in cyber-physical)
- **Work with industry**
  - Challenge the status quo (where's the demand for formal logic?)
  - Tools that change the way we work
  - Division of labor not necessarily the way research did it (see above)
  - Intellectual Property opportunities will be different than today
- **Work with existing programs**
  - Pilot in component upgrades to legacy systems
  - Don't pontificate; socialize

All the king's horses and all the king's men Couldn't put Humpty together again.

Implies Humpties exist. Don't fight entropy. Lay an Egg.