CSER 2024
conference on systems engineering research
March 25-27, 2024

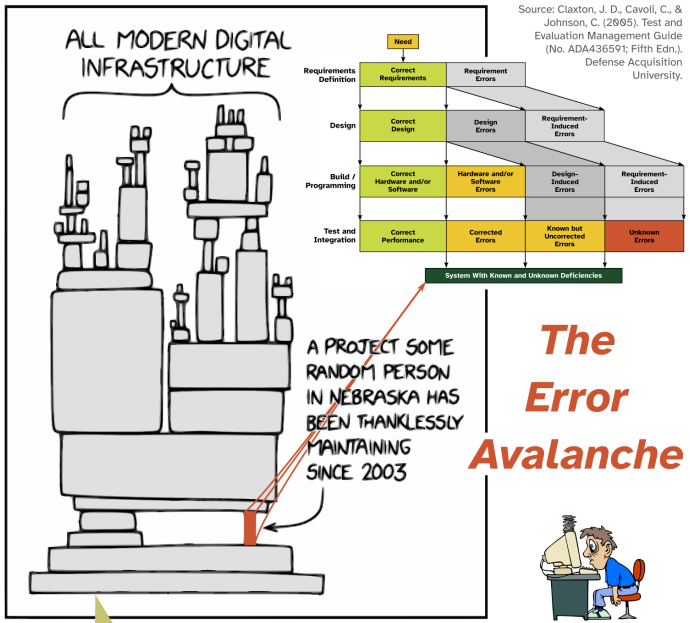
Bootstrapping a Trustworthy and Seamless Digital Engineering Appliance

James S. Wheaton Advisor: Dr. Daniel Herber

COLORADO STATE UNIVERSITY

Problem: Digital Engineering Ecosystems are hindered by a *reverse salient**

* Dedehayir, O., & Mäkinelf, S. J. (2008). "Dynamics of Reverse Salience As Technological Performance Gap: an Empirical Study of the Personal Computertechnology System". Journal of Technology Management & Innovation.

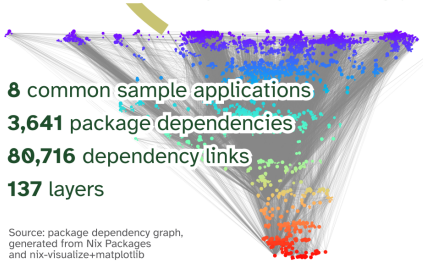


Source: Claxton, J. D., Cavoli, C., & Johnson, C. (2005). Test and Evaluation Management Guide (No. ADA436591; Fifth Edn.). Defense Acquisition University.

The Error Avalanche

XKCD #2347: Dependency

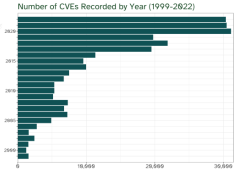
DEPENDENCY ICEBERG:



8 common sample applications
3,641 package dependencies
80,716 dependency links
137 layers

Source: package dependency graph, generated from Nix Packages and nix-visualize+matplotlib

SYSTEMIC VULNERABILITIES:

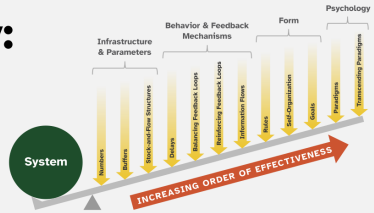


Source: generated from CVE Project open JSON dataset using R and ggplot2, : https://github.com/cveproject/cvelist

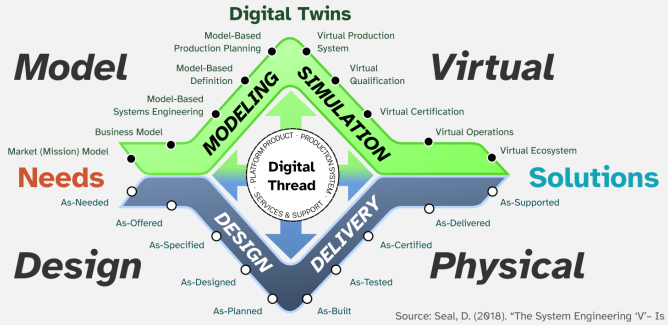
Research Question:

How can we achieve **safe** *and* **reliable** *Digital Engineering?*

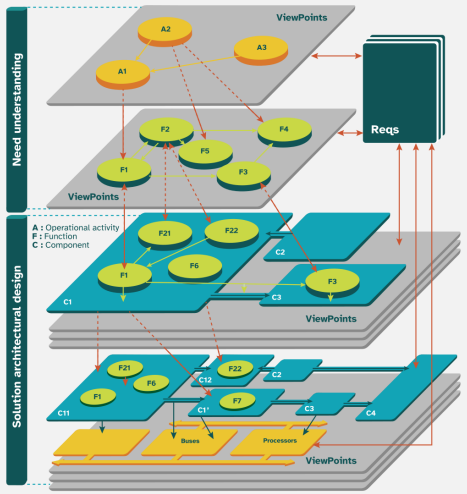
Methodology: *Apply highest leverage*



Source: Meadows, D. H. (2008) *Thinking in Systems: A Primer*, Chelsea Green Publishing.



Source: Seal, D. (2018). "The System Engineering 'V'– Is it Still Relevant in the Digital Age?" Boeing Company, Global Product Data Interoperability Summit



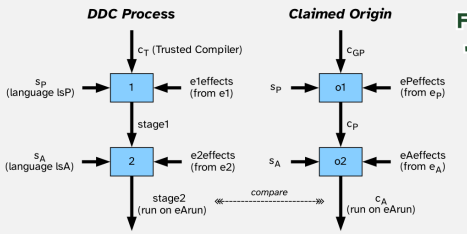
Operational Analysis
What the users of the system need to accomplish

Functional & Non-Functional Need
What the system has to accomplish for the users

Logical Architecture
How the system will work

Physical Architecture
How the system will be developed and built

Source: Voirin, J. (2017). Model-based System and Architecture Engineering with the Arcadia Method. Netherlands: Elsevier Science.



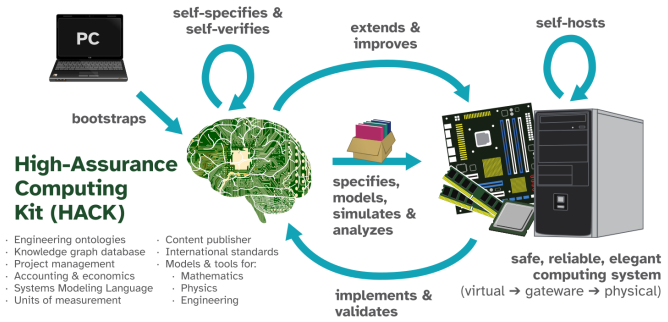
FULLY COUNTER THE "TRUSTING TRUST" ATTACK & MAINTAIN END-TO-END† FORMAL SYSTEM VERIFICATION

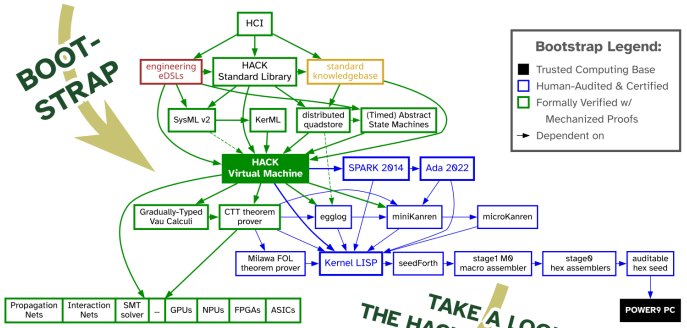
Source: Wheeler, D. A. (2009). "Fully countering trusting trust through diverse double-compiling". PhD thesis. George Mason University

† See: Moore, J. S. (2003). "A grand challenge proposal for formal methods: A verified stack". Lecture Notes in Computer Science. Springer.

Results:

Seamless Digital Engineering (SDE) is a digital engineering tooling paradigm that guarantees model coherence and integrity by affording an elegant human-computer interface for systems modeling that is end-to-end formally verified down thru the computer hardware.[1]





TAKE A LOOK AT THE HACK META-LANGUAGE!

```
(define:function hello-world   ;; specification of the pure function
  :doc       (document "A complete document object or AsciiDoc string")
  :type      [ String :-> String ]
  :params    [ name ]
  :requires  (> (length name) 0)  ;; Hoare triple pre-condition
  :ensures   (= (length out)      ;; Hoare triple post-condition
                (+ (length "Hello, ")
                   (length name)
                   (length "!")))
  :satisfies [ :FR/001 ]   ;; SysML Block "satisfies" Relationship
  :tests     [(test trivial-example
                    :doc "Test that the name is inserted into the greeting."
                    :verifies [ :FR/009 ]  ;; SysML Functional Requirement
                    (= "Hello, World!"
                       (hello-world "World")))
              (ref:test :T/HW-002)]  ;; Reference a test defined elsewhere
  :version  { :major 0 :minor 1 :patch 0 }) ;; granular semantic versioning
```

- Embedded SysML v2 & DocBook
- Dependent type system oriented toward program verification
- Knowledgebase integration
- Contract-based definitions
- Full traceability from requirements to system tests
- Embedded project model data

Conclusions:

- **Clean-slate design is necessary** to overcome the **DE reverse salient**, and a **high-assurance information appliance**[2] applies the greatest leverage against the current paradigm.
- We identified 4 primary system Quality Attributes of a **Seamless Digital Engineering appliance**: 1) **Seamless**, 2) **Trustworthy**, 3) **Elegant**[3], and 4) **Convivial**[4].
- **Full-source bootstrap** and **end-to-end mechanized formal verification** are required to satisfy **Seamless** and **Trustworthy** high-assurance Quality Attribute thresholds.
- **Activity-Based Computing**[5] & **language-oriented programming**[6] with **built-in MBSE affordances** help satisfy Seamless, Elegant & Convivial Quality Attributes.

[1] Wheaton, J. S., & Herber, D. R. (2024). Seamless digital engineering: a grand challenge driven by needs. In AIAA SCITECH 2024 Forum.
[2] Raskin, J. (2000). The humane interface: new directions for designing interactive systems. Addison-Wesley Professional.
[3] Watson, M. D. (2017). "Engineering elegant systems: design at the system level". Penn State University Graduate Seminar. M17-6309
[4] Voinea, C. (2018). "Designing for conviviality". Technology in Society 52. Technology and the Good Society.
[5] Bardram, J. E., Jeuris, S., & Houben, S. (2015). Activity-based computing: computational management of activities reflecting human intention. Ai Magazine, 36(2), 63-72.
[6] M. Felleisen et al. (2018). "A programmable programming language". Communications of the ACM 61.3.