Safety, Dependability, Fault Tolerance
And Verification

John Rushby

Computer Science Laboratory
SRI International
Menlo Park, California, USA

Safety, Dependability, Fault Tolerance And Formal Verification

- In the world of safety-critical systems
  - They don't care (much) about verified software
  - They care about certified systems
- Because it is systems that interact with the world and have the potential to do harm
  - The FAA, for example, certifies planes and engines (and propellers); not software
- Certification because that is the process that (attempts to) evaluate all the risks in deploying some system
  - The system doesn't have to be correct, it has to be safe

Certification

- Certification is not yet a properly engineered process
    - And its science base is poorly developed

  Similarly for its dual: accident investigation

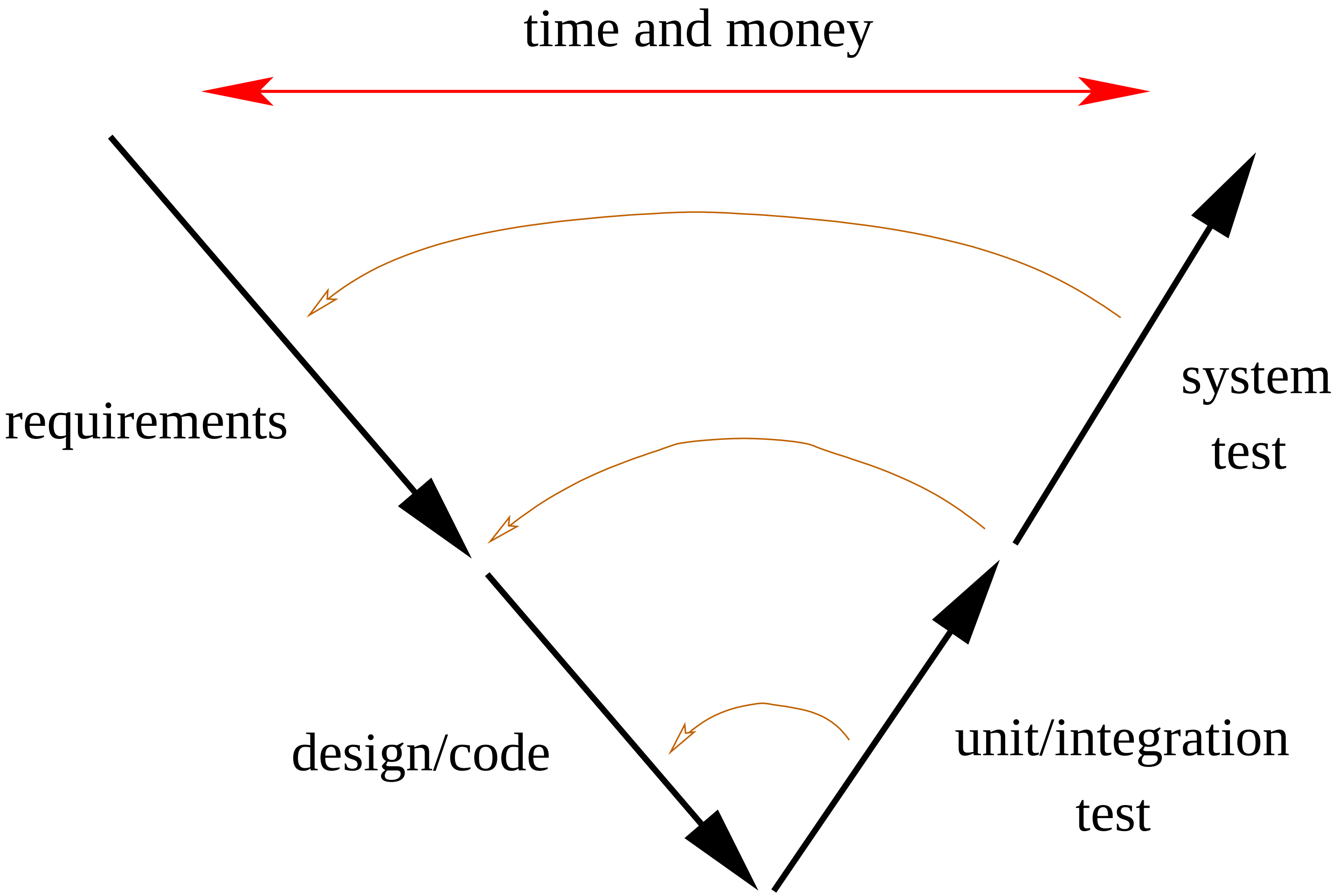
- Its most sophisticated expressions are built around the notion of a safety case
    - An argument that persuades an independent reviewer/agency that the risks are ALARP (“As Low As Reasonable Practicable”)

- Basically a systematic exploration of the space of “unbounded relevance”
    - Hazards (hazard analysis, HAZOP, fault tree analysis, failure modes and effects analysis)
    - And their mitigation (cf. Gerard Holzmann’s talk)

Certification and Software

- When the processes of design and certification work their way down into subsystems with large software content, the concern and analysis is almost exclusively focused on requirements
- Which mostly concern interactions with other entities
  - The environment
  - Controlled plant
  - Other systems
  - Humans
- Later stages of software development account for 5% of the costs and 2% of the problems in airborne software

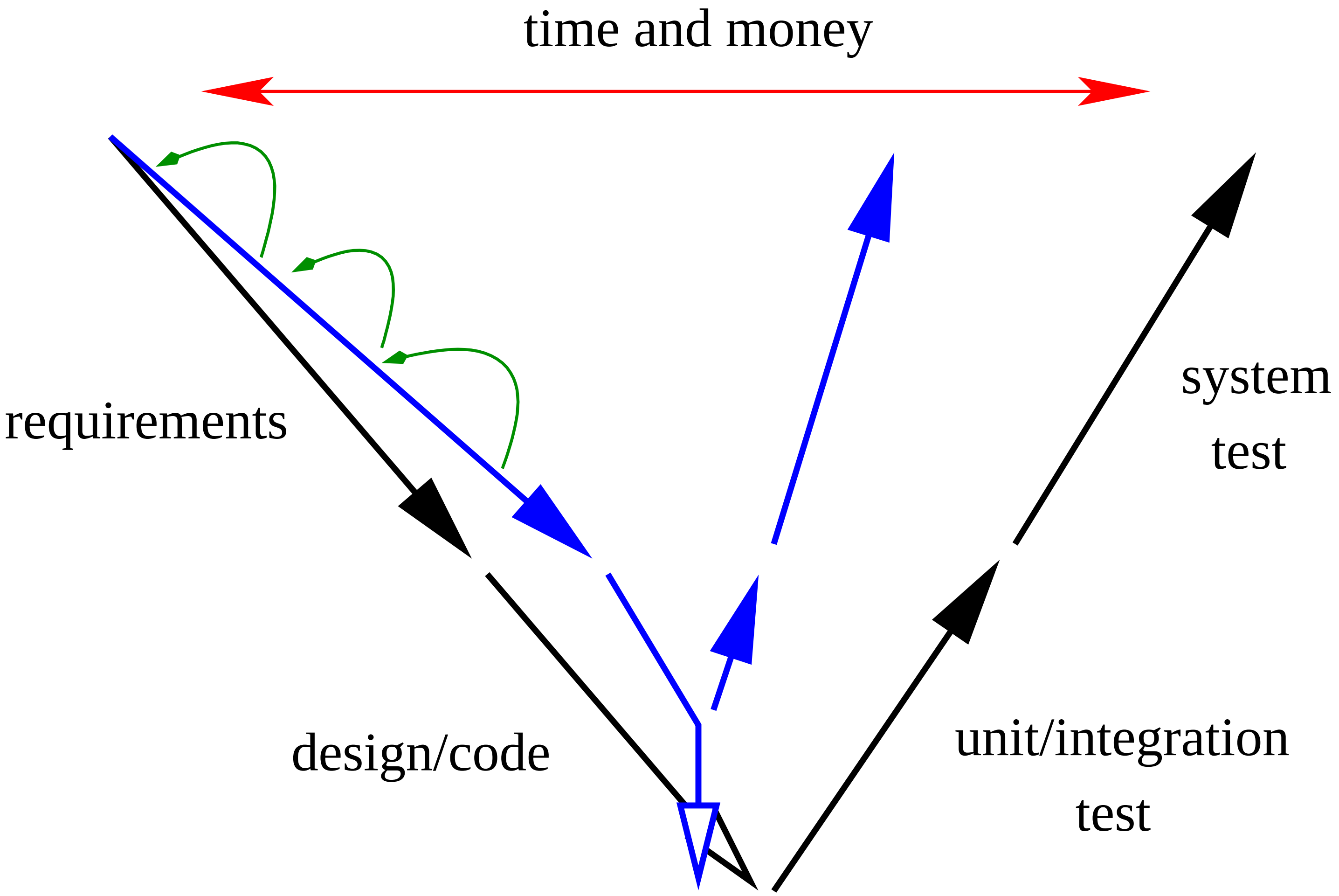
V&V: Simplified Vee Diagram



Certification and Formal Verification

- If it's construed narrowly (program verification), formal verification will make only a small impact on development and certification (tighten the bottom of the Vee)
- Construed broadly, it could provide a foundation for a science of certification
  - Model and explore the space of unbounded relevance
  - And its interaction with emerging requirements
    - ⋆ Will use many techniques from formal methods
    - ⋆ Hybrid systems models, probabilistic models, modeling the human, notions of evidence and of causation
    - ⋆ But probably not program verification
- Requires dialog with unfamiliar communities: systems engineers, certifiers, their committees (cf. SC200, SC205)
- The ideal is compositional certification

Tightened Vee Diagram


time and money
requirements
design/code
system
test
unit/integration
test