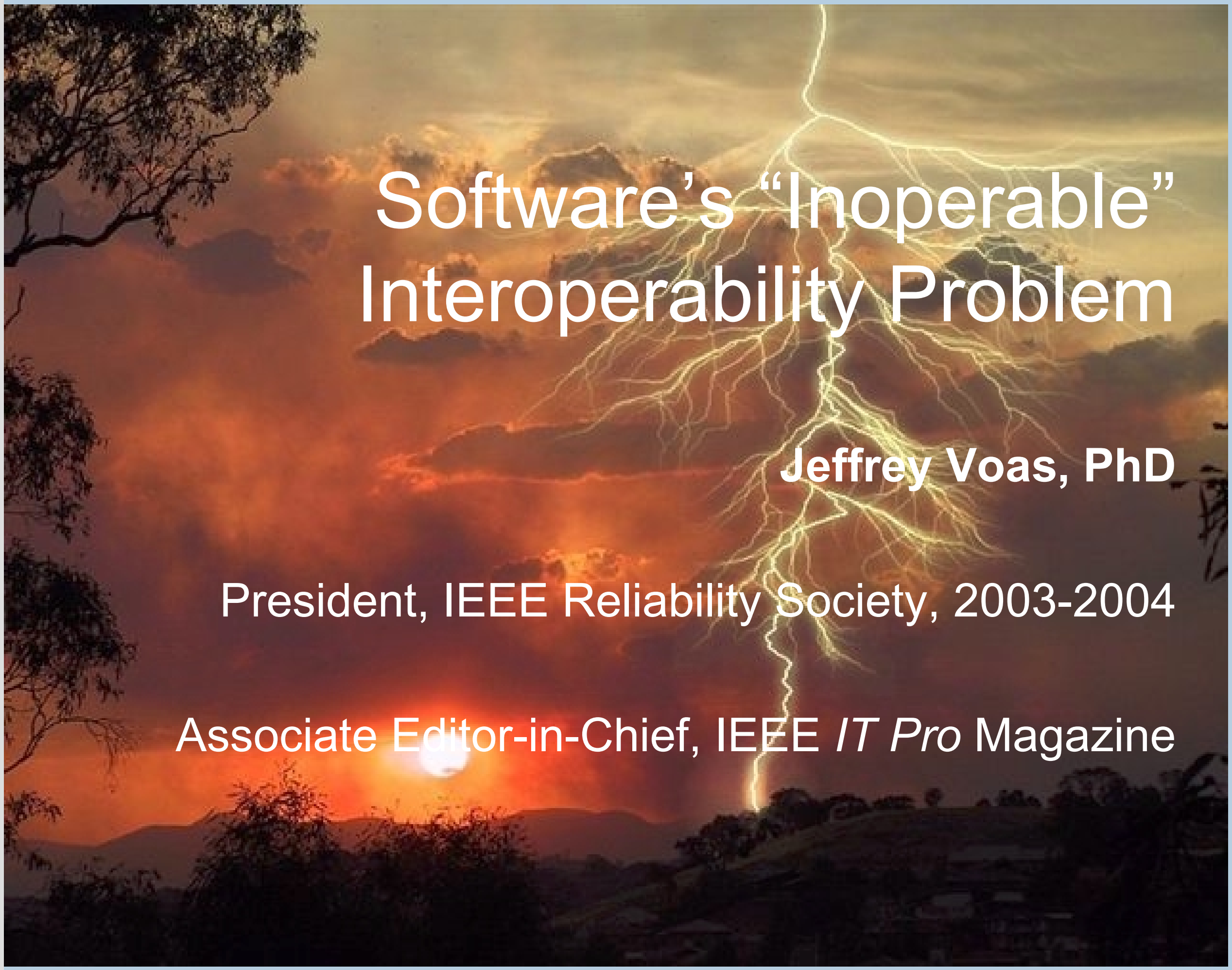
cigital



Software's "Inoperable" Interoperability Problem

Jeffrey Voas, PhD

President, IEEE Reliability Society, 2003-2004

Associate Editor-in-Chief, IEEE *IT Pro* Magazine

IEEE

ReliabilitySociety

IT Professional

What is a Standard?

Simply a line in the sand from which a certificate of compliance or non-compliance can occur.

Standards and Certification are inseparable.

Premise for Software Certification Standards

Third party software should be tagged with some guarantee (or at least a “warm fuzzy”) as to how “good” the software is, i.e., a certificate

Problem: Software Of Unknown Pedigree (SOUP)

Goal of Certification: Software of Known Pedigree

Problem: What is “good enough” software?

Three Key Messages That a Certificate Can Convey

- Compliance with standards *vs.*
- Fitness for purpose *vs.*
- Compliance with the requirements

Information to Support Certificates

Information to support the creation of certificates should be based on an *claims-evidence-arguments* framework, much as is done in courts of law.

Standards are Not Perfect

- Vague: Develop software that only does "good" things
  - Common sense "dos" and "don'ts" - Very watered done by voting time
- Disclaimers by publishing organizations
  - Profitable to organization that publishes them
- Used only if mandated
- Return-on-investment is un-quantified
- Thwart intellectual creativity
  - "Protectionist" legislation
- Paperwork
  - 2167A: ~400 English words per Ada code statement
- "Old news" before being ratified
- Relating one to another is very hard
  - Hundreds in existence

cigital

Standards are Not Perfect (cont)

- Different interpretations
- Process certifications are just documentation checks unless personnel remain on site during the project
- Re-certification
  - Client: over 300 mods to a safety-critical medical device that never requested re-certification for any of those mods.
- Cannot be easily tested for compliance
  - Mis-certifications are common
- Lack of fairness during certification judgment
  - FDA Center for Devices and Radiological Health (CDRH)
- So much legacy functionality exists that complies with no standards yet still gets integrated, making it’s impact to the system unknown.
  - WAAS

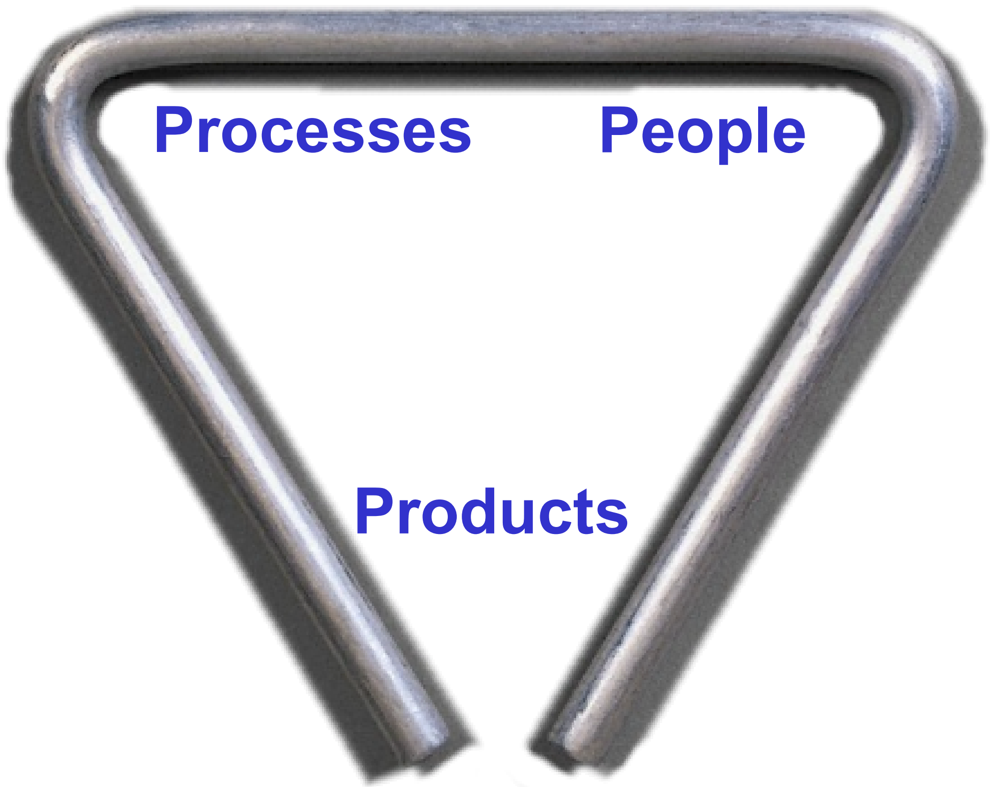
State-of-the-Practice/Art

"A consumer [patient] may not be able to assess accurately whether a particular drug is safe, but [they] can be reasonably confident that drugs obtained from approved sources have the endorsement of the U.S. Food and Drug Administration (FDA) which confers important safety information. Computer system trustworthiness has nothing comparable to the FDA. **The problem is both the absence of standard metrics and a generally accepted organization that could conduct such assessments. There is no Consumer Reports for Trustworthiness."**

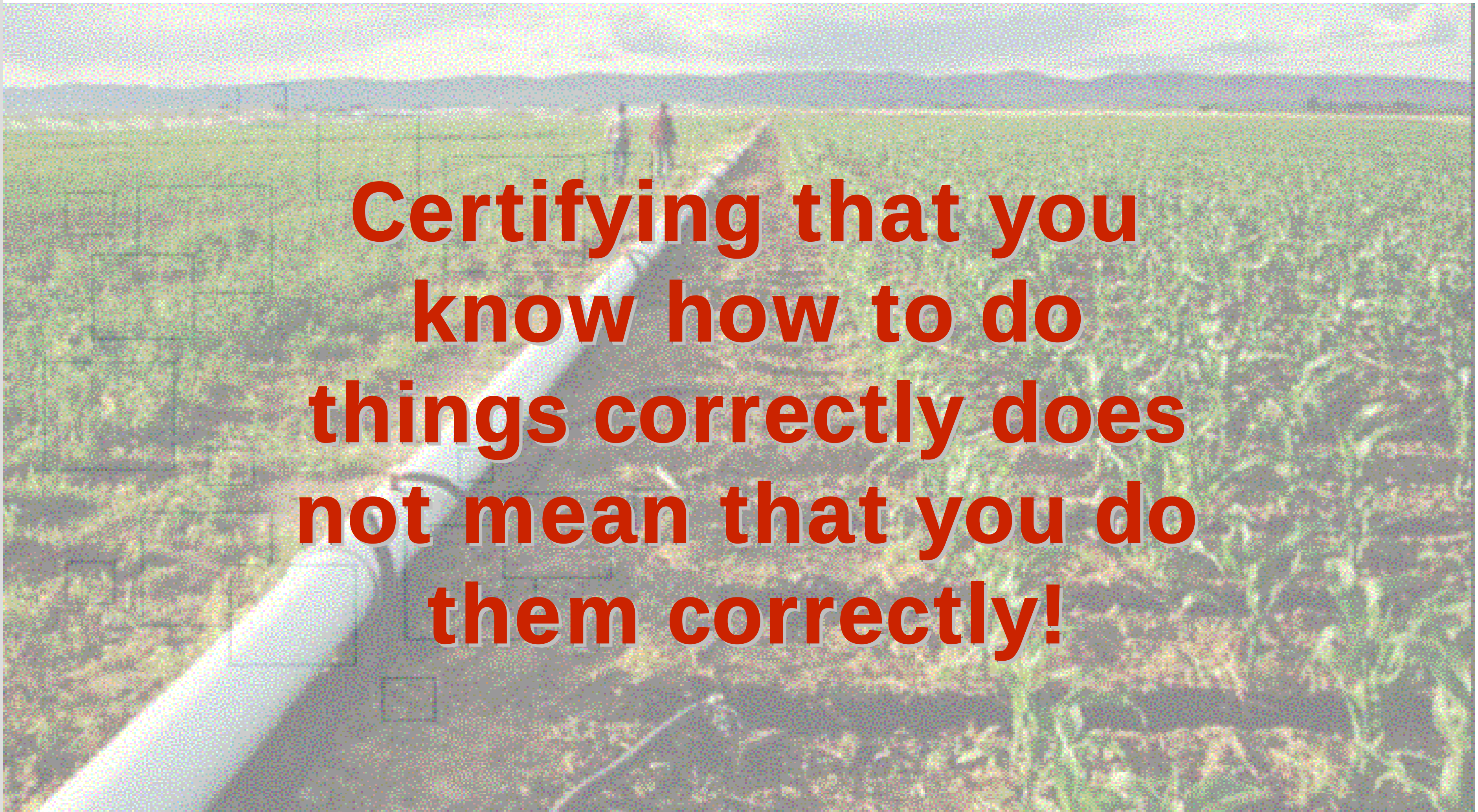
[Source: "Trust in Cyberspace," National Academy of Sciences report, National Academy Press, 1998.]

Three Schools of Thought

All cert. standards incorporate one or more of these perspectives



1. Process: Clean Pipes, Dirty Water?



On a positive note, process improvement schemes at least, from an acquisition standpoint, alleviate some of the concerns associated with SOUP

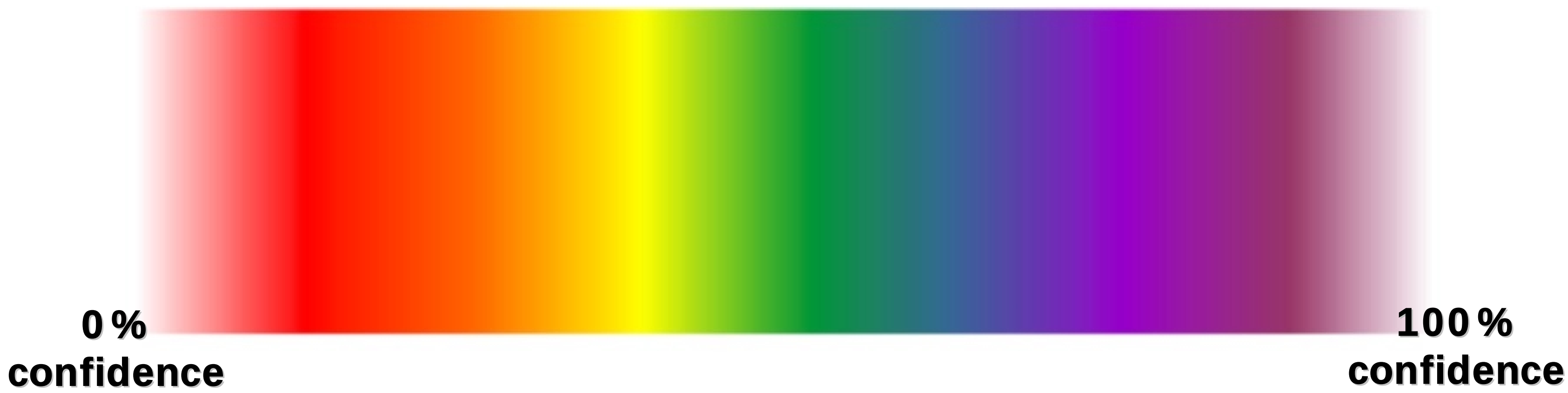
2. People

The IEEE Computer Society has developed a program to certify software engineering professionals. This program provides formal recognition of professionals who have successfully achieved a level of proficiency commonly accepted and valued by the industry.

Serious Question

What does process maturity and personnel accreditation say specifically about how the software will behave in the future?

3. Product: The Software Itself



Spectrum of possibilities as to what a certificate proclaiming that some “quantified” level of quality has been built in could state --- it could say anything in the range between “Nothing” (*e.g.*, “here is a piece of software”, etc.) to “This software will always work perfectly under all conditions” (i.e., a 100% guarantee of perfection).

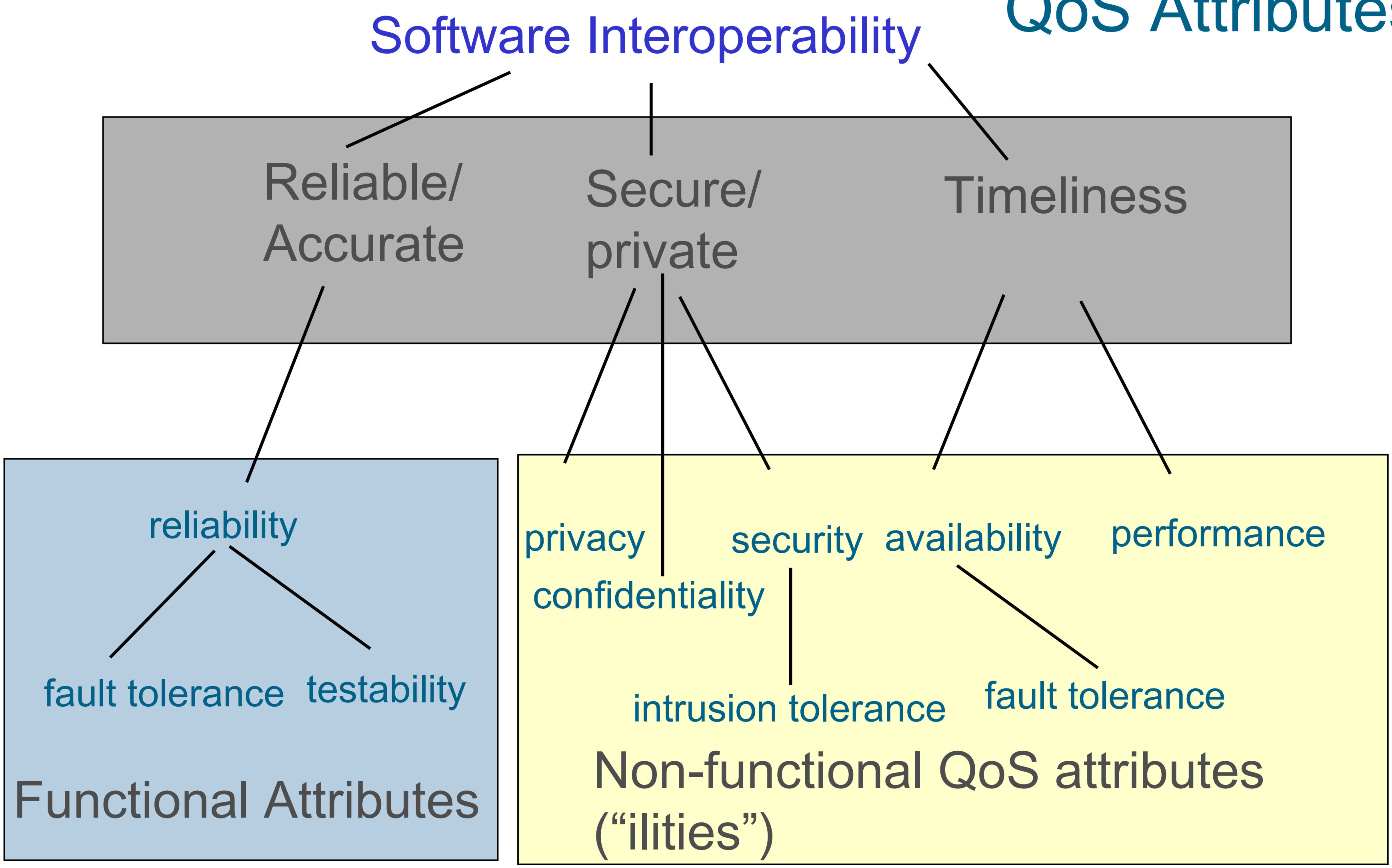
And So How Should a Certification Standard Be Created?

What Attribute is Being Certified?

- Reliability?
  - RTCA's DO178B (FAA)
- That the degree of testing done was appropriate?
  - RTCA's DO178B (FAA)
- Safety?
  - System (process) vs. component (product) safety
    - IEC 61508 vs. UL 1998
- Security?, Availability?, Fault Tolerance? Performance?, etc.
- That certain development procedures were followed?
  - SEI Capability Maturity Model
  - ISO 900x

Interoperability

QoS Attributes

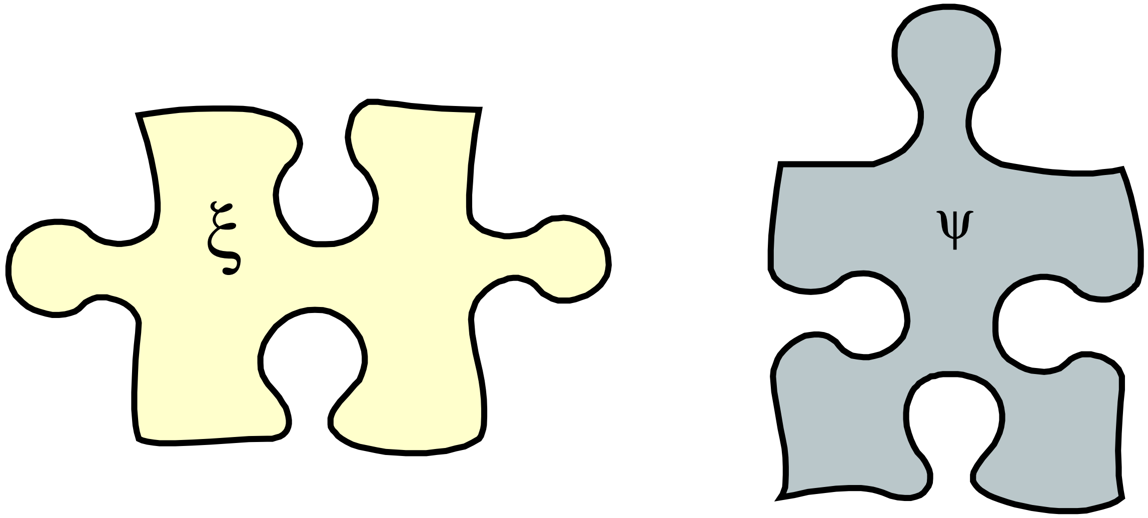


Position Statement

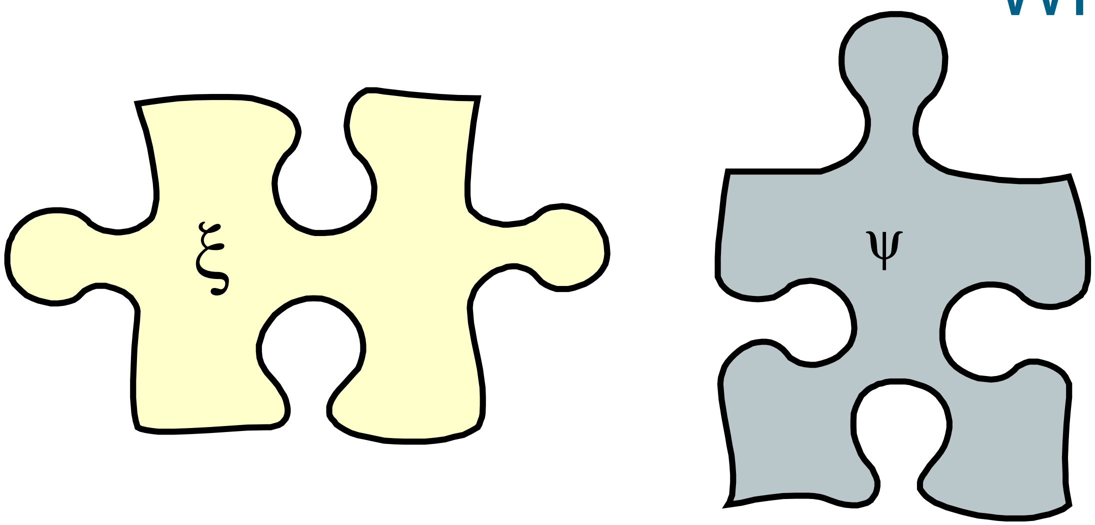
Software's Interoperability is some combination of:

(1) the degree to which the *functional* requirements are met, as well as, (2) the degree to which the *non-functional* requirements are met.

Two Components



With Attributes



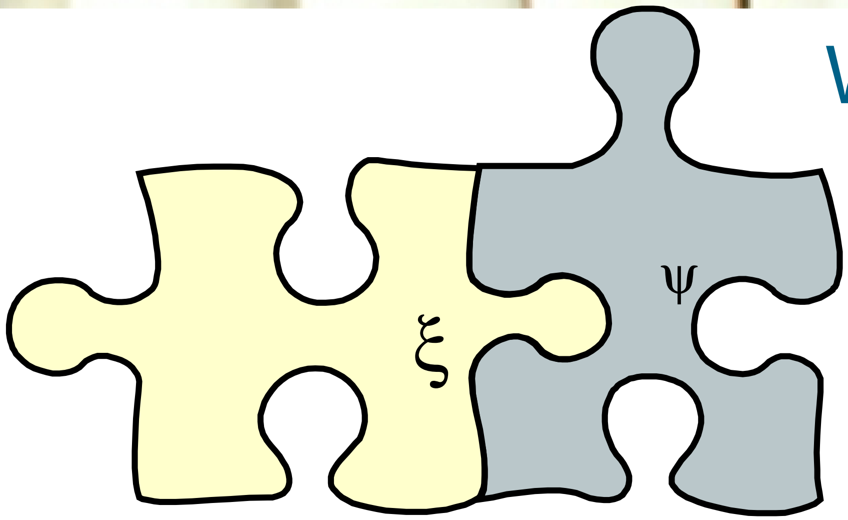
ξ has the following properties:

$(a\mathrm{R}, b\mathrm{P}, c\mathrm{F}, d\mathrm{Sa}, e\mathrm{Se}, f\mathrm{A}, g\mathrm{T}, h\mathrm{M})$

ψ has the following properties:

$(i\mathrm{R}, j\mathrm{P}, k\mathrm{F}, l\mathrm{Sa}, m\mathrm{Se}, n\mathrm{A}, o\mathrm{T}, p\mathrm{M})$

What Have You Got?



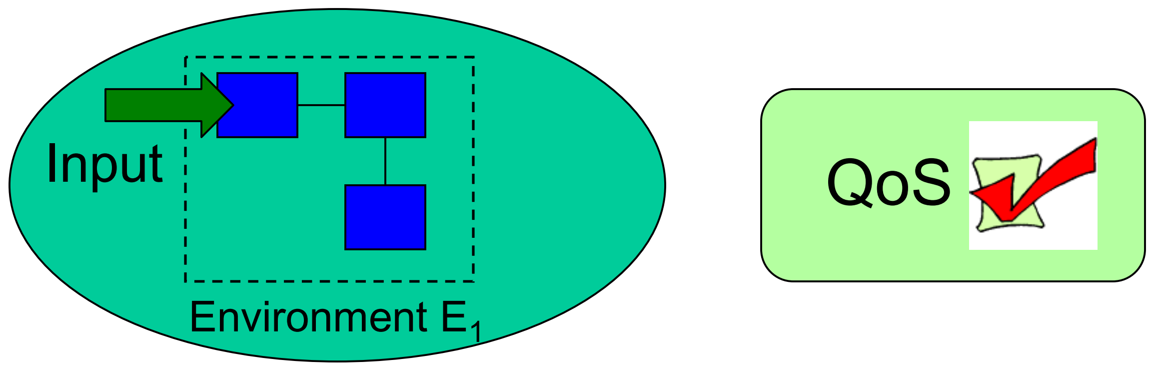
Then $F(\xi \circ \psi)$ will inherit some level of Quality of Service (QoS) from the individual components. Is that level of quality an integer? Probability? An n-tuple of values? Color coded (green red yellow)?

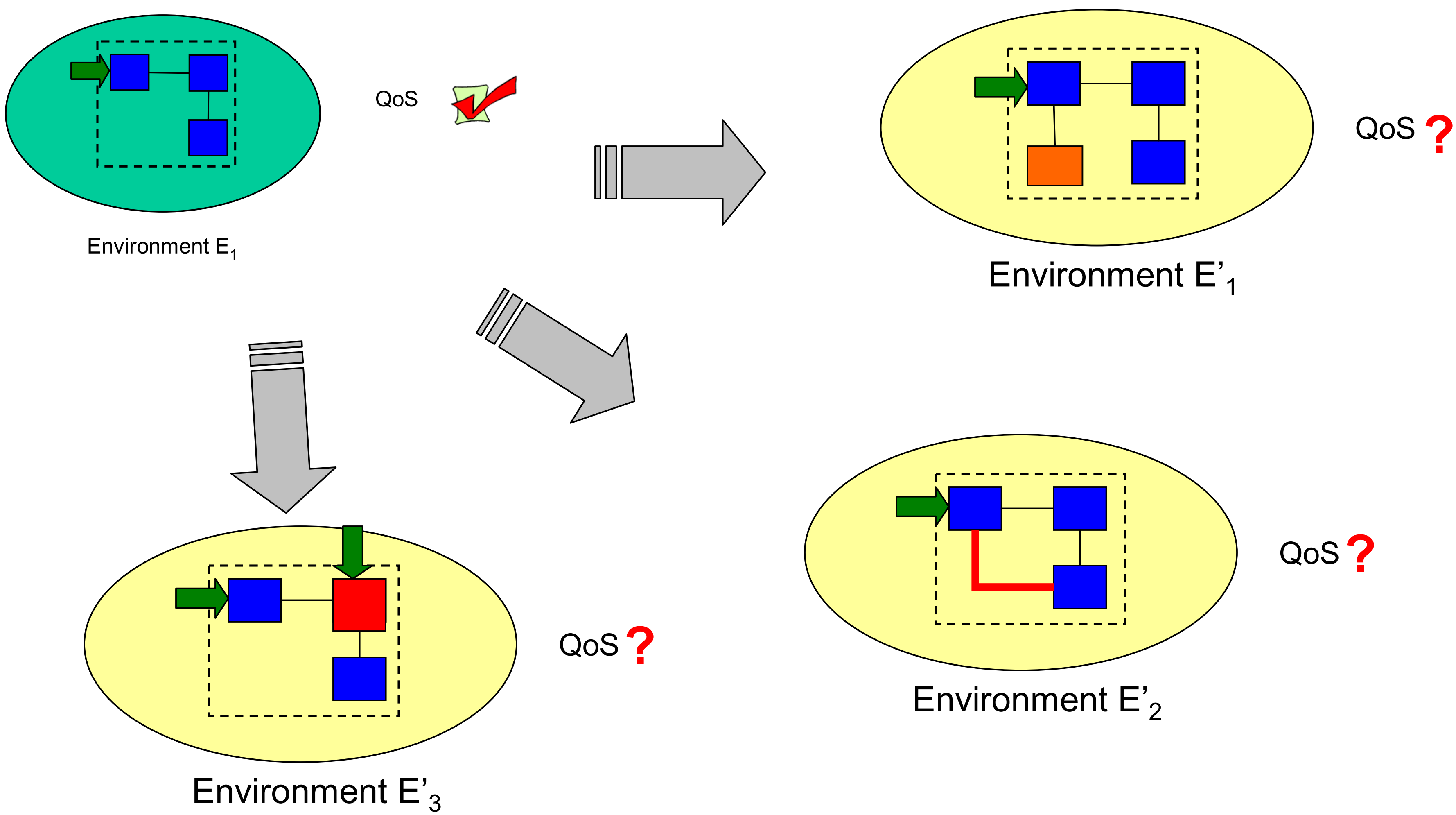
Key Point: The composite QoS must represent something from which predictions of future behavior can be made.

Difficult Because …

QoS attributes have little meaning in terms of their ability to be measured and traded off until they are defined in the context of the target system, i.e., their environment.

- Reliability
- Performance
- Fault Tolerance
- Safety
- Security
- Availability
- Testability
- Maintainability




QoS
Environment E_1
QoS ?
Environment E'_1
QoS ?
Environment E'_2
QoS ?
Environment E'_3

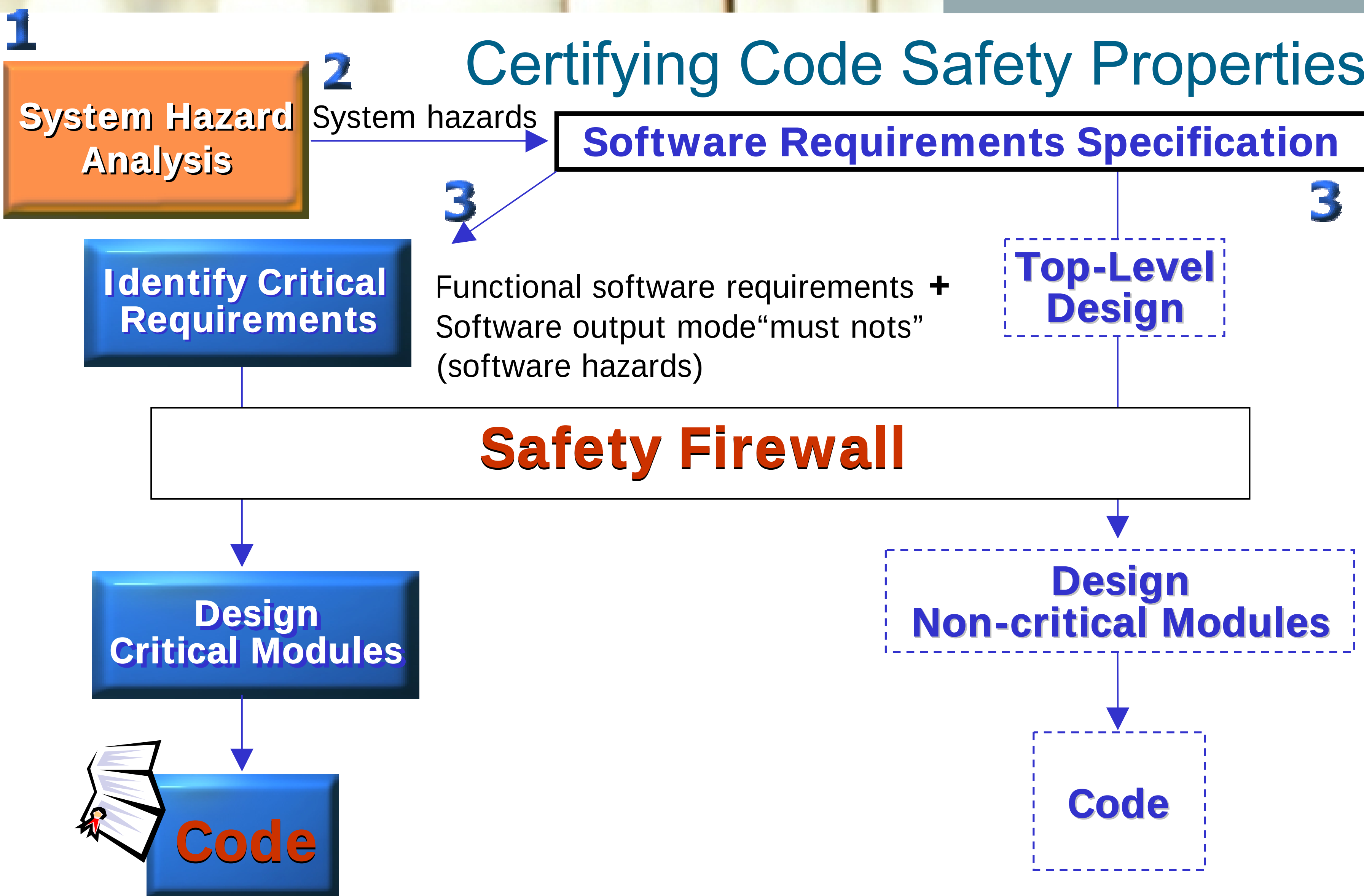
Bottom Line for Certification of Software Interoperability

The following 8 characteristics must be considered:

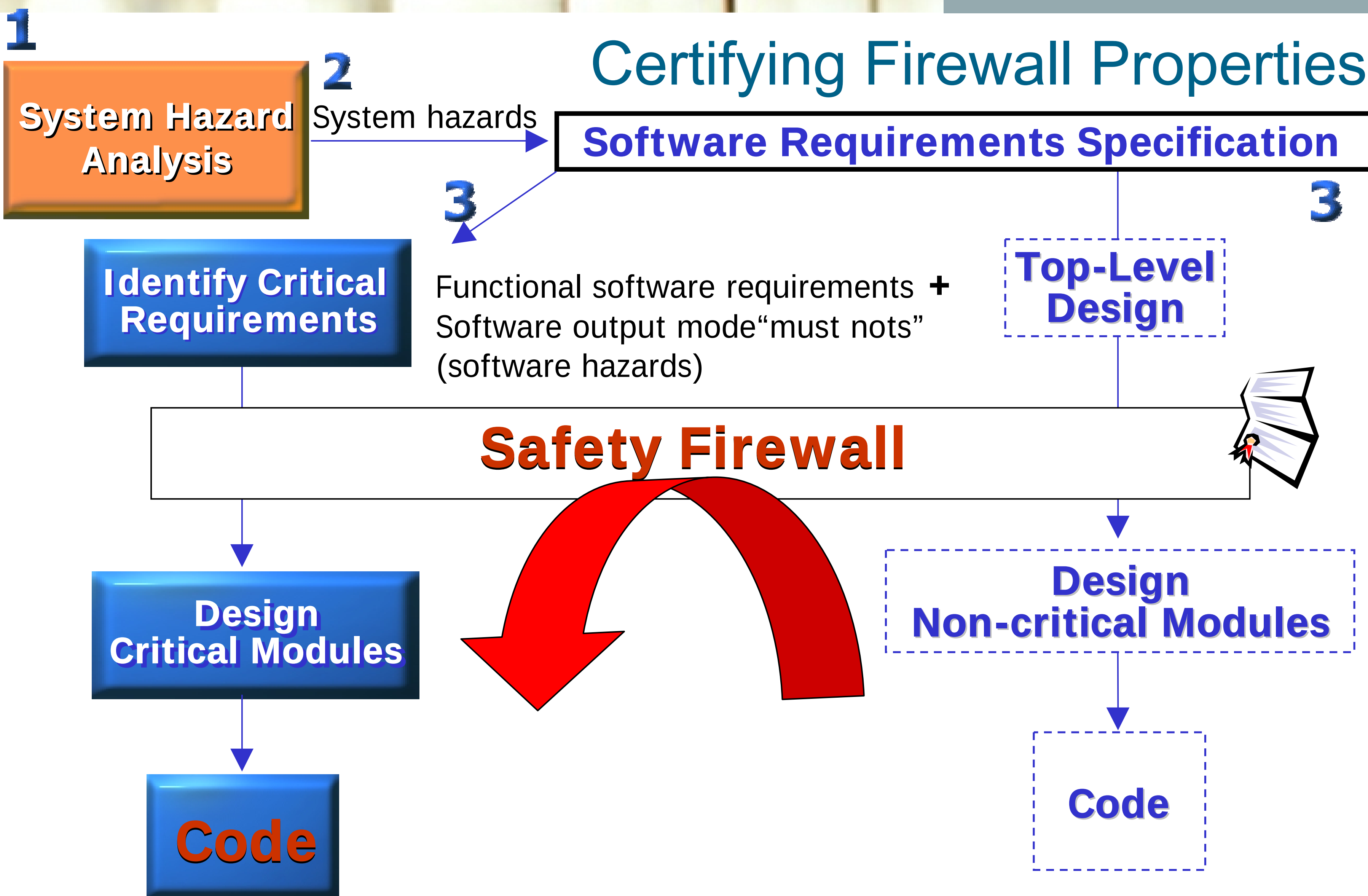
(1) compos-ability, (2) predictability, (3) attribute measurement, (4) QoS attribute trade-off analysis (technical and economic), (5) fault tolerance and non-interference analysis, (6) requirements trace-ability, (7) access to pre-qualified components, and (8) precise bounding of software's mission, environment, and threat space.

Safety

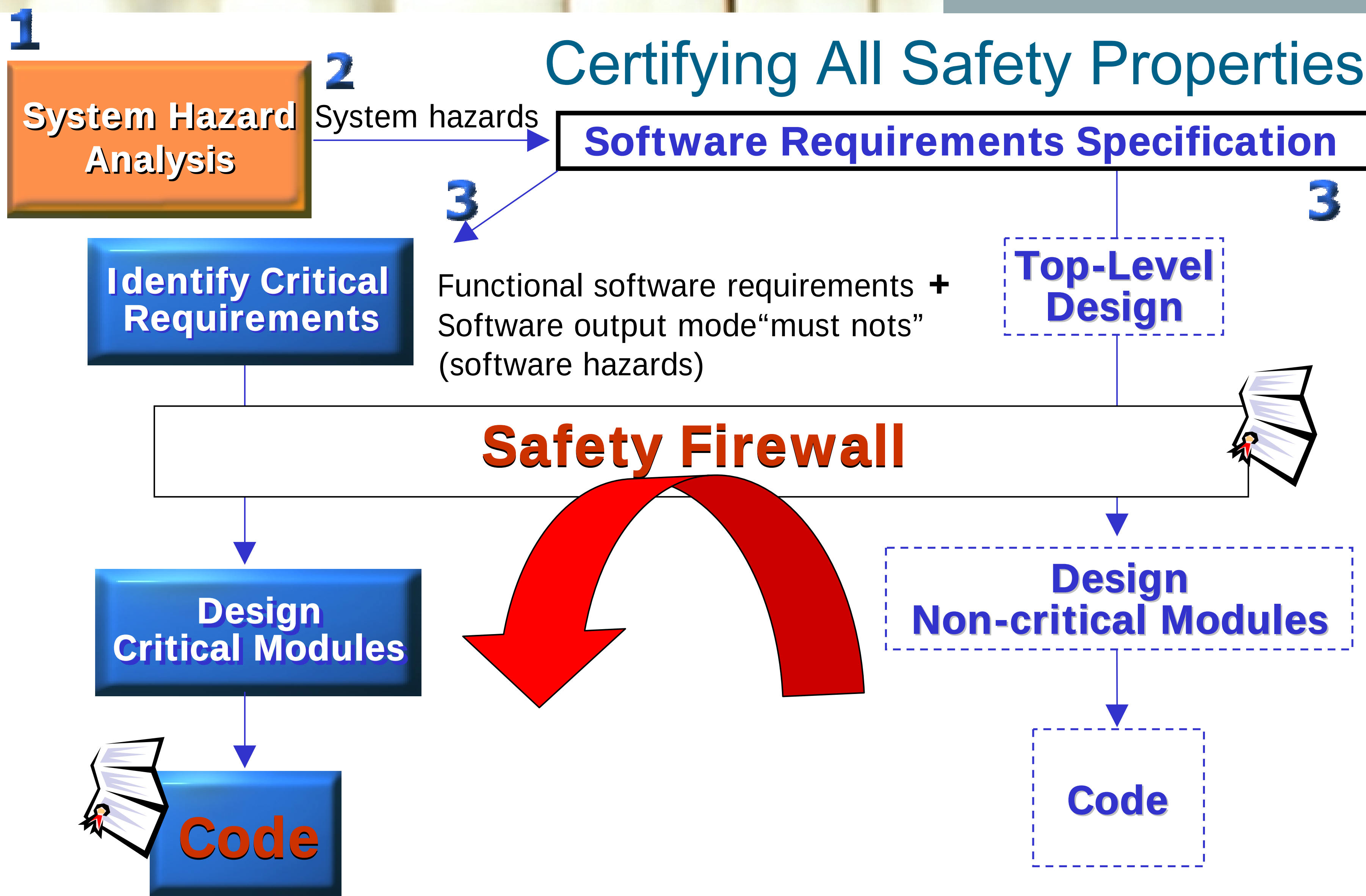
Certifying Code Safety Properties



Certifying Firewall Properties



Certifying All Safety Properties



Closing Thoughts

1. Standards and certification are inseparable in order to achieve the goal of interoperable and safe behavior

2. Product certification is distinct from process certification and personnel accreditation

3. The blending of existing standards, collecting quantifiable metrics, defining precisely what QoS attributes are warranted, and defining what a certificate implies or does not imply is pivotal to believable certificates.

4. “You cannot improve what you cannot measure” – Lord Kelvin

Recommended References

- Software Safety and Reliability, Debra S. Herrmann, IEEE Computer Society Press, 1999.
- Software Engineering Standards, James W. Moore, IEEE Computer Society Press, 1998.
- Guide to Software Engineering Standards and Specifications, Stan Magee and Leonard L. Tripp, Artech House, 1997.
- UL 1998
- RTCA DO-178B
- J. Voas, F. Charron, and L. Beltracchi, “Error Propagation analysis studies in a Nuclear Research Code”, *Proceedings of the 1998 IEEE Aerospace Conference*, Snowmass, CO.
- IEEE Educational Activities Department video: Software Reliability
- IEEE Educational Activities Department video: Software Safety
- IEEE Educational Activities Department video: Software Testing