# "Looking for a Good Argument": Assurance Case Frameworks

May 2003

Chuck Howell, howell@mitre.org
Scott Ankrum, ankrums@mitre.org



MITRE

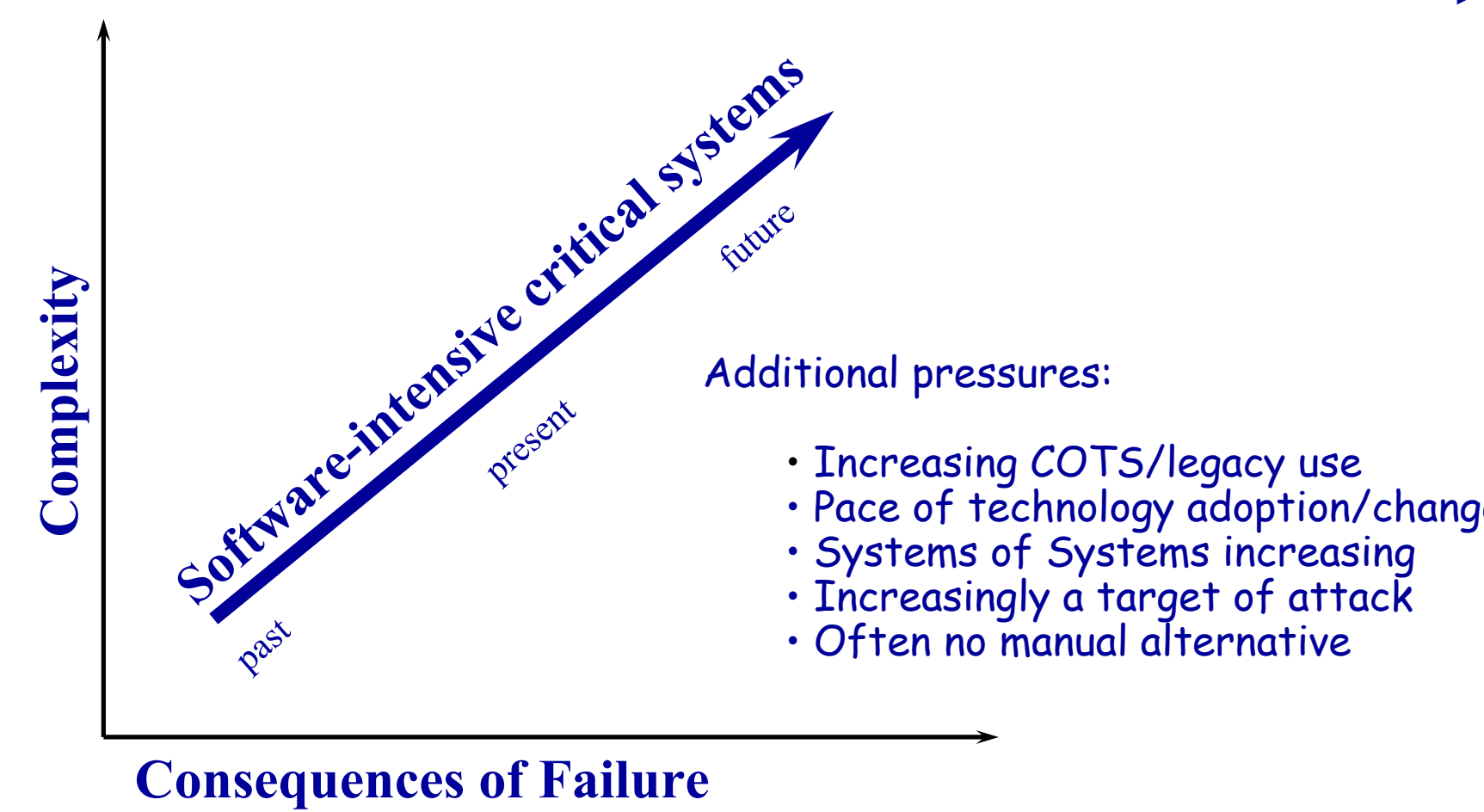
# Our Reach Exceeds Our Grasp in Software



4 June 1996: 1st flight of Ariane-5

- Cause: simple software error
  - Overflow converting 64 bit value to 16 bit
- Failure despite …
  - Hardware confidence measures: Redundant computers
  - Software confidence measures: Software used successfully on Ariane-4
- Why?
  - Both computers ran the same software
  - Different flight envelope, and evolutionary use changed software's environment in unanticipated ways

# The Trend: Our Reach Keeps Increasing




Additional pressures:

- Increasing COTS/legacy use
- Pace of technology adoption/change
- Systems of Systems increasing
- Increasingly a target of attack
- Often no manual alternative

# Why is This Hard?



- Software is a discrete system; traditional engineering disciplines typically work with continuous systems
  - Sensitivity to small errors
  - Limited ability to generalize test results
  - Absence of safety margins
  - Enormous numbers of states to consider
- Good experiments and empirical studies are rare, and we have a weak scientific and engineering basis for decisions
- Many problems are latent in rarely used portions, waiting for "rare events" to trigger them
- "Best practices" often optimize specific activities without respect to the total process and problem

# What is an "Assurance Case"?

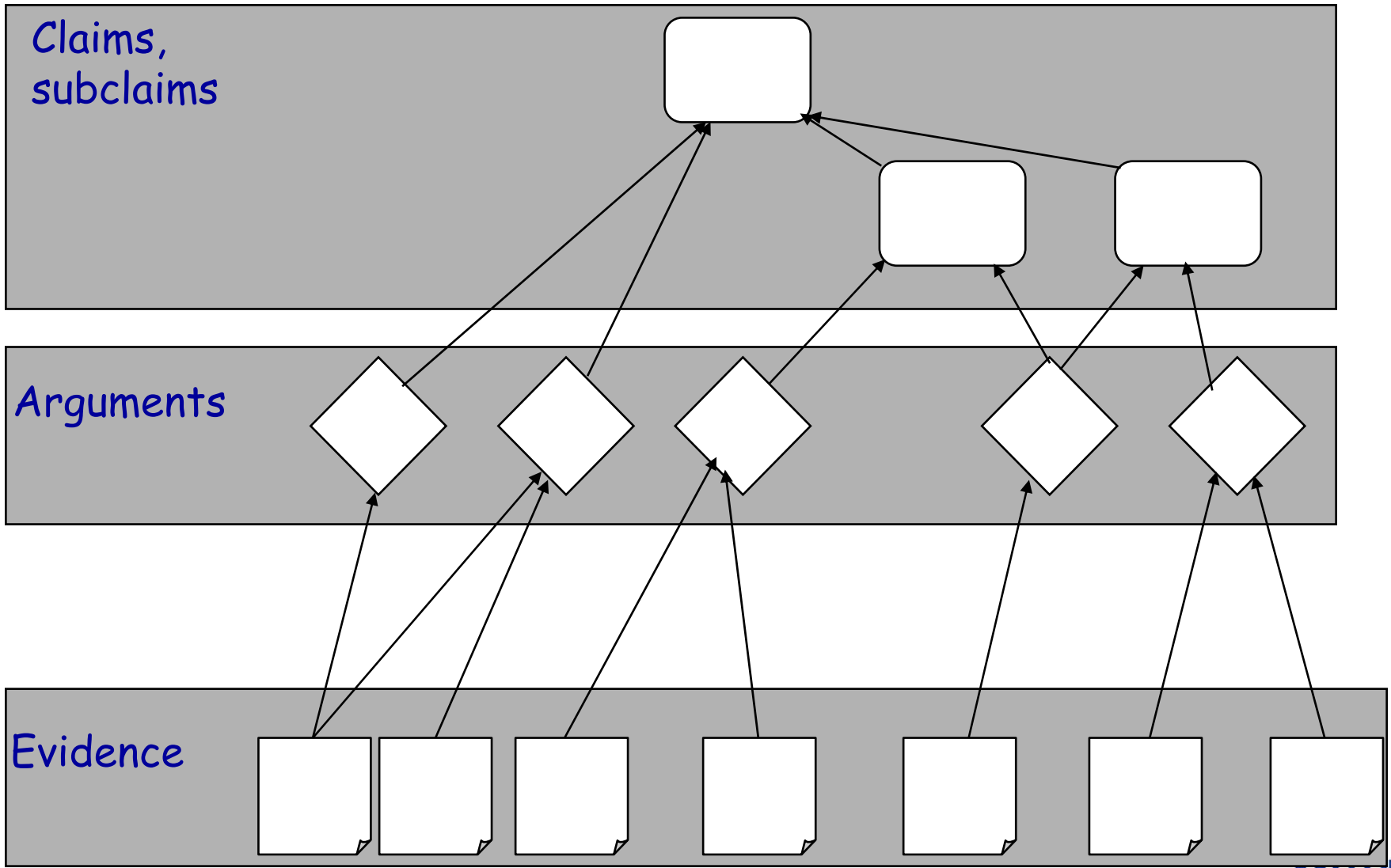


- Critical systems under regulation or acquisition constraints
  - Third party certification, approval, licensing, etc.
  - Require a documented body of evidence that provides a compelling case that the system satisfies certain critical properties for specific contexts (to "make the case")
  - Examples: DO-178B, Common Criteria, MIL-STD-882D
  - "safety case", "certification evidence", "security case"…
  - Collectively we'll refer to them as "*assurance cases*"

*A documented body of evidence that provides a convincing and valid argument that a specified set of critical claims about a system's properties are adequately justified for a given application in a given environment.*

# So What is an "Assurance Case"?

Assurance Case Research






MITRE

# Claims in Assurance Cases



- Assertion of compliance with key requirements and properties
- Must be in a specific context
  - Environment
  - Services or behavior
  - Threats

  - "Is this brick safe?" illustrates why…

- Subclaims may be analogous to "lemmas" in a proof
  - Supports separation of concerns, workflow, makes overall case more manageable

*Terminology used here and next few slides is from the SHIP project (Assessment of the Safety of Hazardous Industrial Processes in the Presence of Design Faults, an EU research program), but is pretty typical across domains*

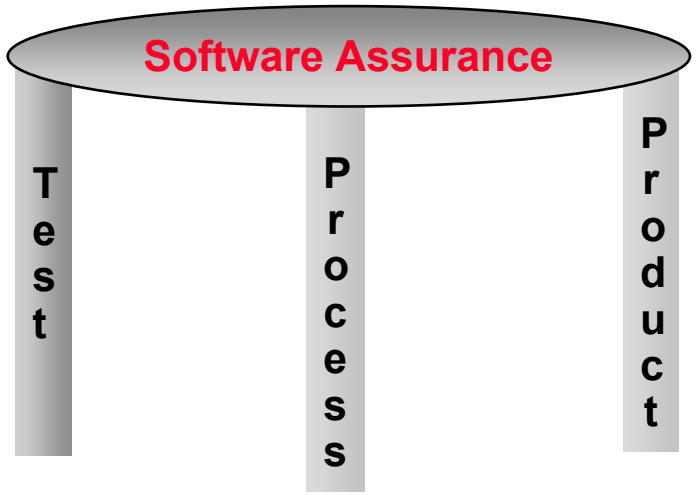
# Arguments in Assurance Cases



- Link evidence to claims via a series of inference rules
- Three types of argument
  - *Deterministic:* application of defined rules to produce a true/false assertion (e.g., formal proof, exhaustive test)
  - *Probabilistic:* quantitative statistical reasoning to establish a numerical threshold (e.g., MTTF)
  - *Qualitative:* Relying on adherence to rules that have an indirect link to desired properties (e.g., standards, process guides)
- No such thing as perfection: "It is quite possible to follow a faulty analytical process and write a clear and persuasive argument in support of an erroneous judgment."
  -- R. Heuer, The Psychology of Intelligence Analysis

# Evidence in Assurance Cases: The "Assurance Tripod"






- Combining evidence from multiple sources:
  - Process and people used to develop the system
  - Systematic testing
  - Product review and analyses
- Testing alone cannot provide adequate evidence, nor can process compliance alone, nor can product assessment alone

# "The Trouble with Assurance Cases Today"…



- Much room for improvement
  - Post-mortems of spectacular mishaps
  - Cost and burden of high assurance systems

- "It's not how hard you worked, it's what you accomplished"
- There are problems in every aspects of assurance cases
  - Building them
  - Reviewing them
  - Maintaining them
  - Reusing them
- The volume of material combined with little structuring support and ad hoc "rules of evidence" are the root of many problems

# Building the Assurance Case



- Most guidance is strong on excruciating detail for format, weak on gathering, merging, and reviewing technical evidence
- Guidance often uses the "cast a wide net" tactic
  - Assurance costs time and money
  - "Squandered diagnostic resources"
  - Some work on a "portfolio management" approach to assurance
- Unless you have tool support, free format text files makes coordination, tracking, and workflow management hard
  - Imagine building a 500 page project plan by hand, on paper
  - Some investigation of similar problems for intelligence analysts and structured argumentation

# Reviewing the Assurance Case



- Stacks of free-format text makes the review process tedious
  - Hard to see linkages or patterns
  - Hides key results in sheer volume
- Weak guidance on review of arguments and evidence often results in ad hoc criteria (be very nice to your reviewer!)
- Rarely is there explicit guidance for weighing conflicting or inconsistent evidence

> Often viewed as irrefutable, evidence is, in fact, an interpretive science, refracted through the varying perspectives of different disciplines. … [Judging evidence requires] reasoning based on evidence that is incomplete, inconclusive, and often imprecise.
>
> *The Evidential Foundations of Probabilistic Reasoning*, David Schum

# Maintaining the Assurance Case



- The one thing more brittle than software: the associated assurance case
- Volume and relatively rare use of tools make it difficult to understand impact of a change on assurance structure
  - Have the claims changed?
  - Are arguments invalidated or new ones needed?
  - Is evidence still relevant, new evidence needed?
  - "Weak link effect" of discrete systems compounds the problem
- Revalidation costs are a major burden for critical systems
- "Breakage" when bow-waving requirements another example

# Reusing the Assurance Case



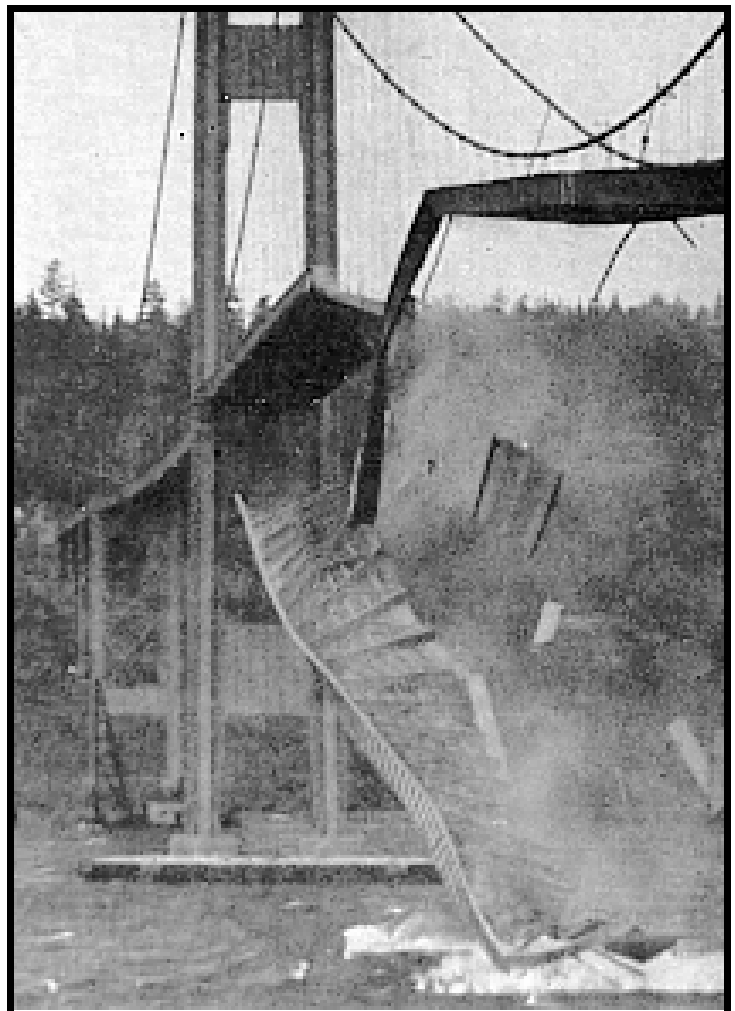
- Assurance case frameworks are rarely "1st class objects", the subject of study per se
- More attention for tool support, idioms and templates, extracting patterns for future use would be useful
- The relationship between claims, arguments, and evidence is not often explicit, making it hard to distinguish reusable from project specific portions of assurance case

- Compare this with building a deck using a project planning tool

# Bridge Building and *Learning* What Can go Wrong



- Engineers to Nature: "Fool me once, shame on you – fool me twice, shame on me"
- Software developers: "Fool me N times, hey, this is complex and anyway no one expects software to work the first time…"

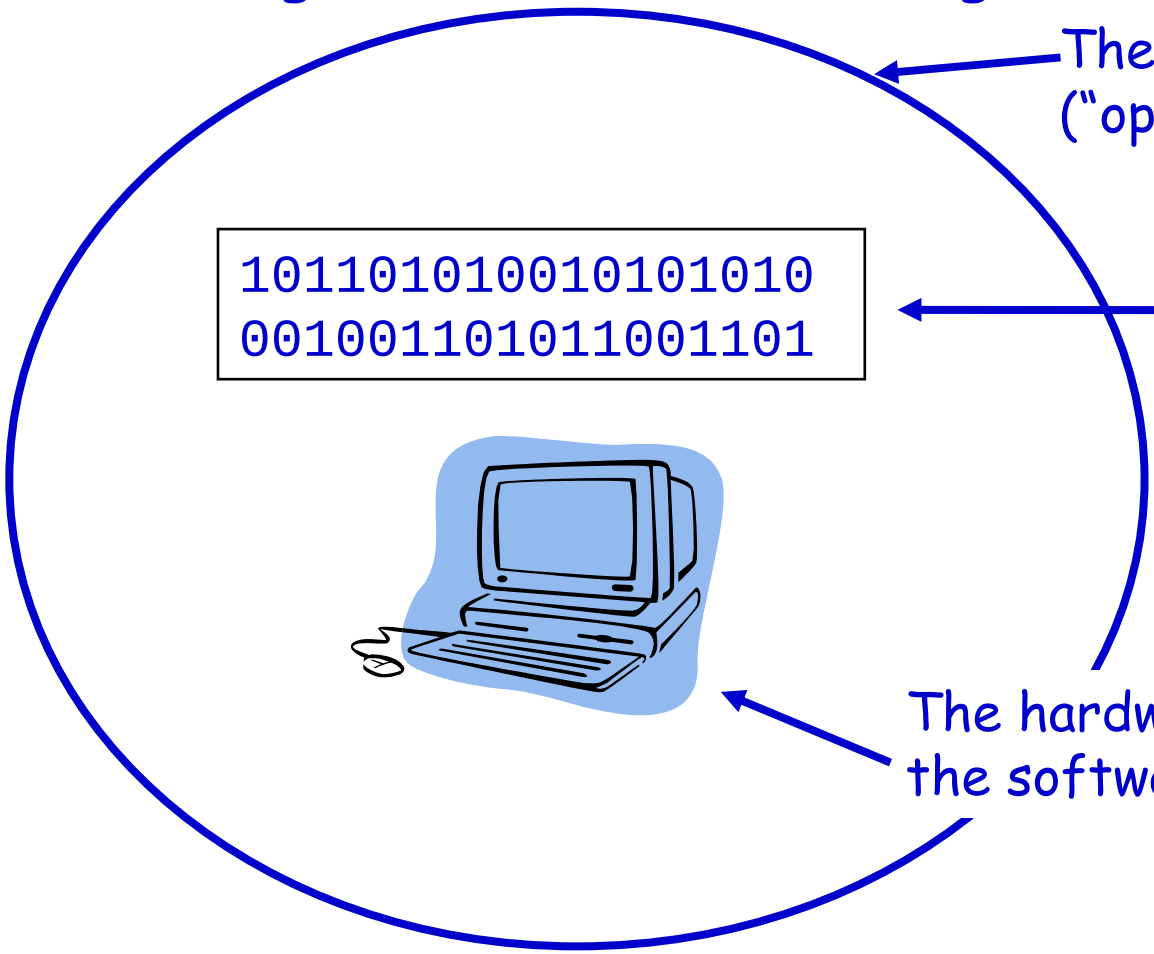


Galloping Gertie's final moments

# Software is Brittle: As forgiving of Mistakes as Dialing a Phone Number…



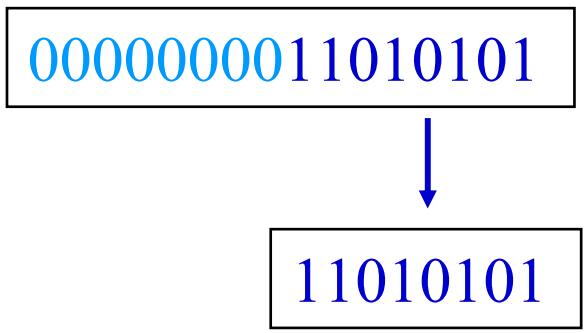
Breakage can come from changes to…



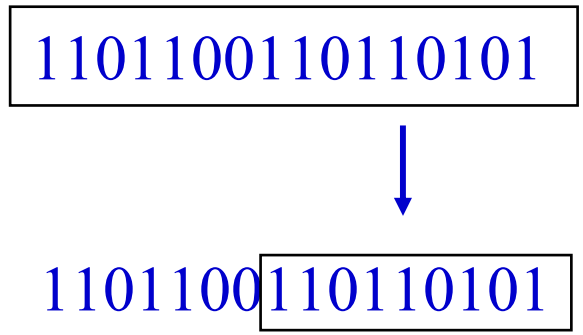
# Direct Cause of Ariane-5 Failure: Overflow Converting 64 bit value for 16 bit variable



"Horizontal bias" in Ariane-4




Same software, different Horizontal speed in Ariane-5




Breakage was caused by change to operational context

Software that caused failure actually had no role after launch, was left in as "harmless" to avoid changing software, *for fear of breaking it by changing it.*

# The Plot Thickens: The Developers Did Consider Overflow

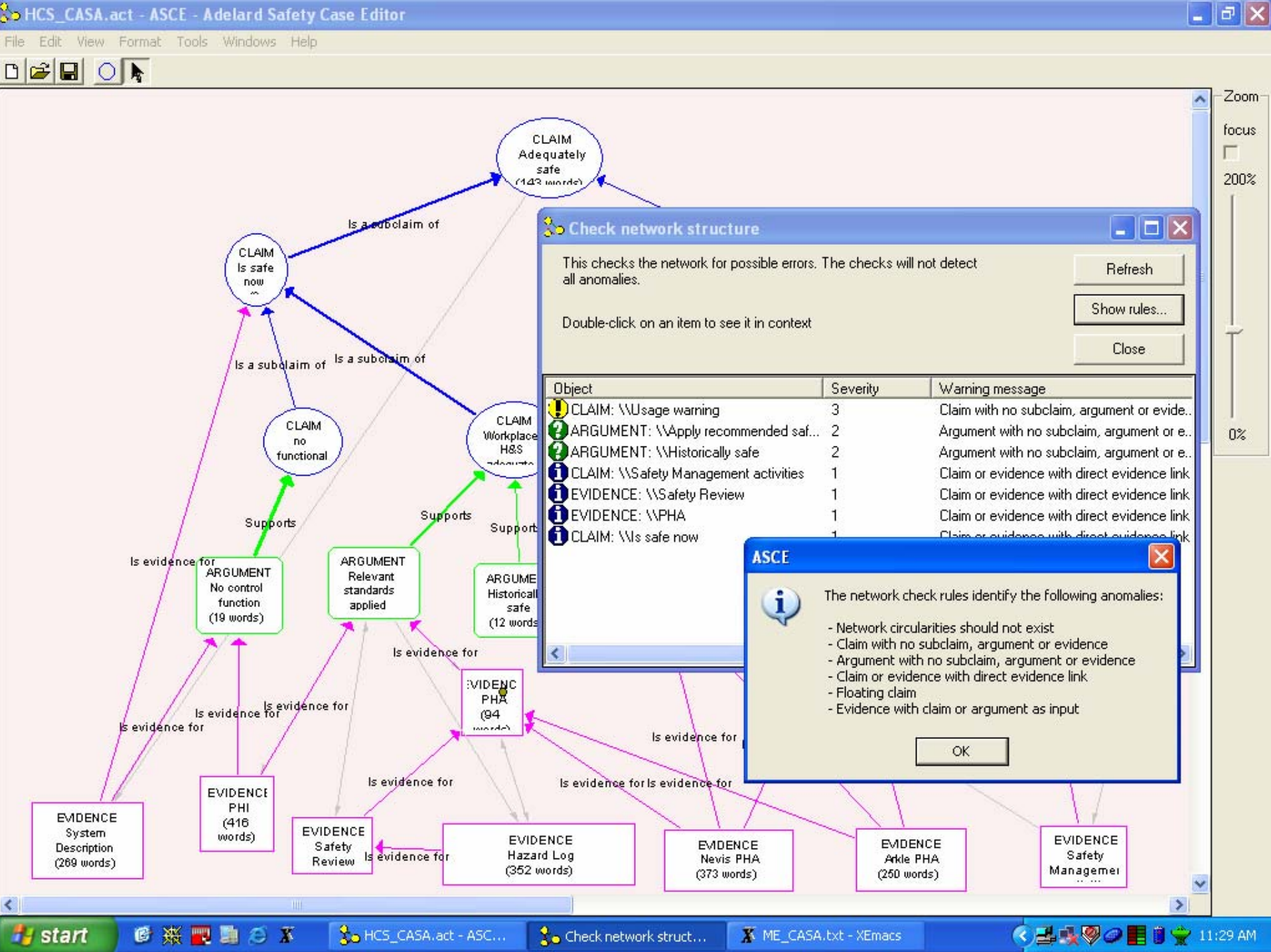


- Every conversion was analyzed for possible overflow risk
- If there could be overflow, a "handler" was put in place to deal with it – so the control program would NOT shut down
- For some conversions it was physically impossible to get a value too large, so these were not protected by a handler (to save CPU cycles)
- The offending conversion was fine in Ariane-4, but could indeed overflow in Ariane-5
- This assurance process was not revisited for Ariane-5, previously sound assumption was violated, boom. *Maintenance of the assurance case was the weak link.*

# What Could Tool Support Provide?



- Simple management of complexity and volume would help
  - E.g., Microsoft Project for planning and tracking complex project vs. all by hand…
  - Checking simple structural properties
  - Browsing and report generation
- Workflow and management of geographically dispersed efforts
- Replanning as things change… ("No plan survives contact with the enemy")
- Templates and tailoring to capture lessons learned, reduce wheel reinvention
- Support for using and exchanging consistent notation for issues of claims, evidence, and arguments


HCS_CASA.act - ASCE - Adelard Safety Case Editor
File Edit View Format Tools Windows Help
Zoom
focus
200%
0%
CLAIM Adequately safe
Is a subclaim of
CLAIM Is safe now
Is a subclaim of
Is a subclaim of
CLAIM no functional
CLAIM Workplace H&S
Supports
Supports
Supports
Is evidence for
ARGUMENT No control function (19 words)
ARGUMENT Relevant standards applied
ARGUME Historicall safe (12 words
Is evidence for
EVIDENC PHA (94
Is evidence for
Is evidence for
Is evidence for
Is evidence for
Is evidence for
Is evidence for
Is evidence for
EVIDENCE System Description (269 words)
EVIDENCE PHI (416 words)
EVIDENCE Safety Review
EVIDENCE Hazard Log (352 words)
EVIDENCE Nevis PHA (373 words)
EVIDENCE Arkle PHA (250 words)
EVIDENCE Safety Manageme
Check network structure
This checks the network for possible errors. The checks will not detect all anomalies.
Refresh
Show rules...
Double-click on an item to see it in context
Close
Object
Severity
Warning message
CLAIM: \\Usage warning 3 Claim with no subclaim, argument or evide..
ARGUMENT: \\Apply recommended saf... 2 Argument with no subclaim, argument or e..
ARGUMENT: \\Historically safe 2 Argument with no subclaim, argument or e..
CLAIM: \\Safety Management activities 1 Claim or evidence with direct evidence link
EVIDENCE: \\Safety Review 1 Claim or evidence with direct evidence link
EVIDENCE: \\PHA 1 Claim or evidence with direct evidence link
CLAIM: \\Is safe now
ASCE
The network check rules identify the following anomalies:
- Network circularities should not exist
- Claim with no subclaim, argument or evidence
- Argument with no subclaim, argument or evidence
- Claim or evidence with direct evidence link
- Floating claim
- Evidence with claim or argument as input
OK
start
HCS_CASA.act - ASC...
Check network struct...
ME_CASA.txt - XEmacs
11:29 AM

# Opportunities for Improvement: Focus of FY03 MITRE Research

*Assurance Case Research* →

- Ask me again this time next year… ☺

- Areas being explored (including connecting with existing work)
  - Notation (including graphical representations)
  - Tool support (using the Adelard Safety Case Editor, ASCE[1])
  - Empirical study
    - Comparing various existing assurance cases
    - Apply early results to guinea pig projects (in parallel)

[1] www.adelard.com

# Caveat: Impact of New Tools and Techniques

Assurance Case Research


"Knowing is not enough, we must apply. Willing is not enough, we must do."
Goethe