SAND2014-????
Unclassified Unlimited Release

SAND2014-19739C

Exceptional service in the national interest

Sandia National Laboratories

United States Department of Energy
Waste Isolation Pilot Plant

When the Safety and Security of the Free World are at Stake

and Other High Consequence Applications of Risk Assessment

Martin Pilch, PhD, PMP
Thermal Sciences and Engineering, Mgr 1514
Sandia National Laboratories

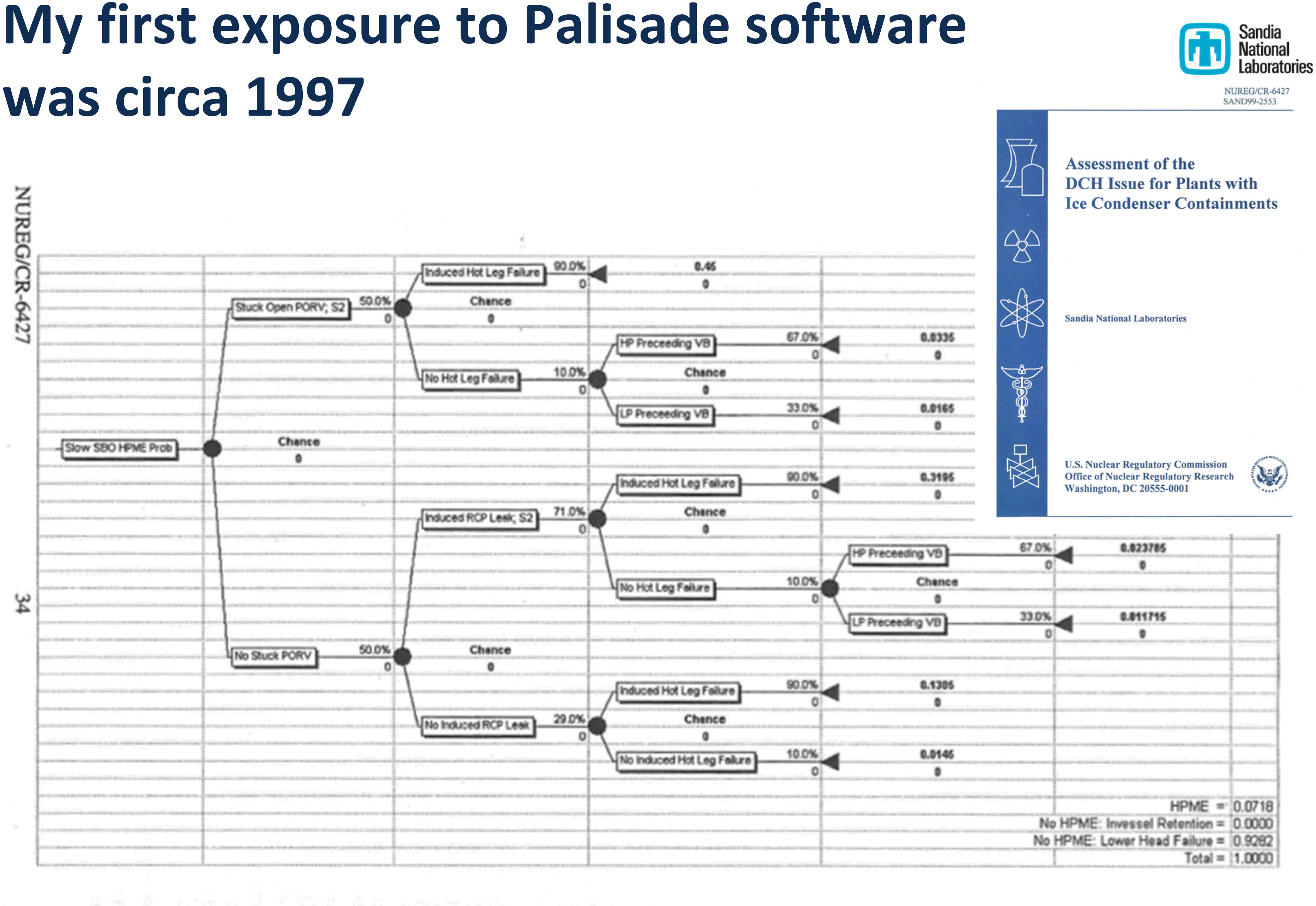
Palisade Risk Conference
New Orleans, LA
November 19-20 2014

U.S. DEPARTMENT OF ENERGY

NNSA National Nuclear Security Administration

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

My first exposure to Palisade software was circa 1997



Sandia National Laboratories

NUREG/CR-6427
SAND99-2553

Assessment of the
DCH Issue for Plants with
Ice Condenser Containments

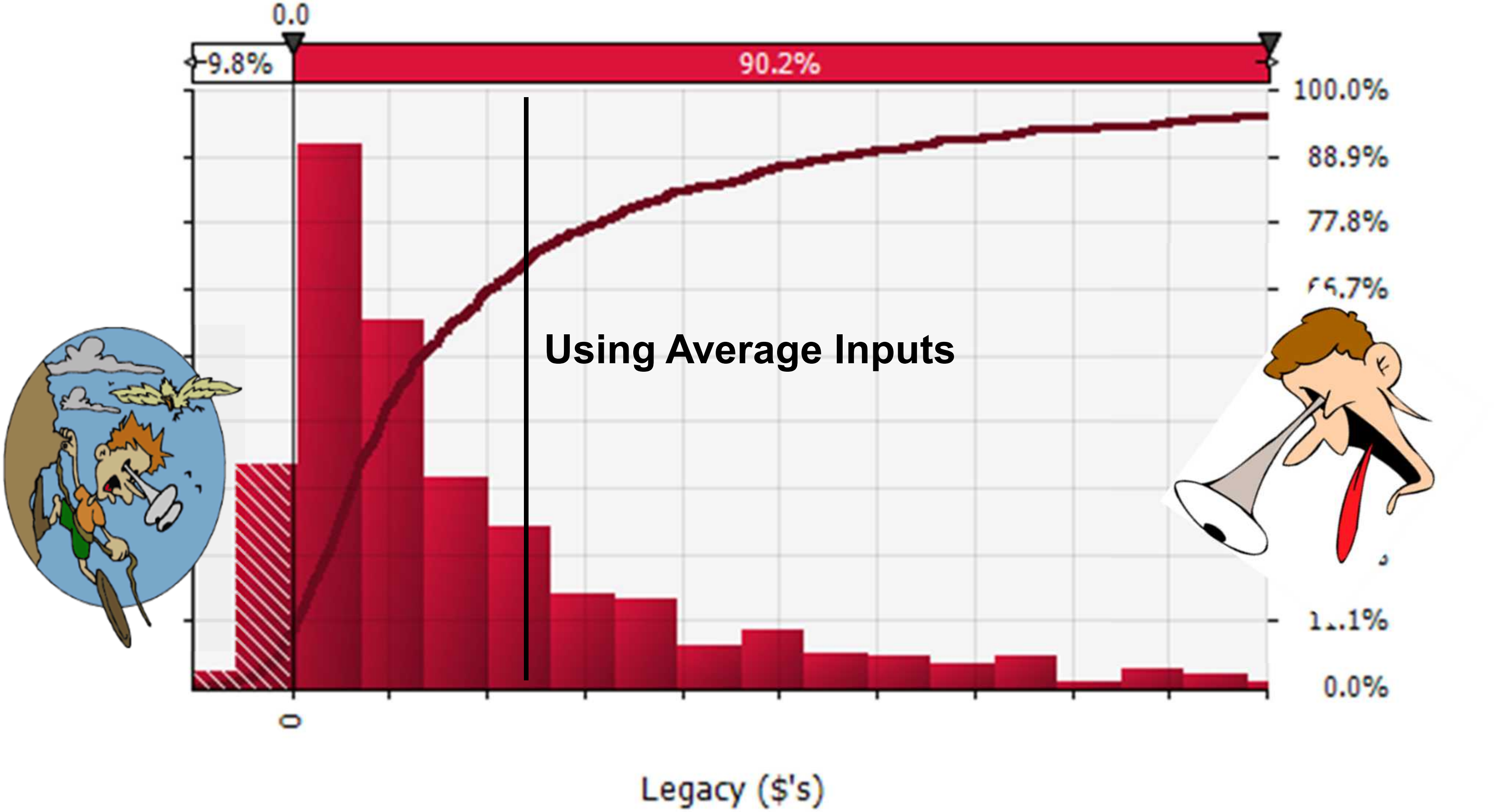
Sandia National Laboratories

U.S. Nuclear Regulatory Commission
Office of Nuclear Regulatory Research
Washington, DC 20555-0001

Figure 4.1 HPME probability for slow station blackouts in Sequoyah

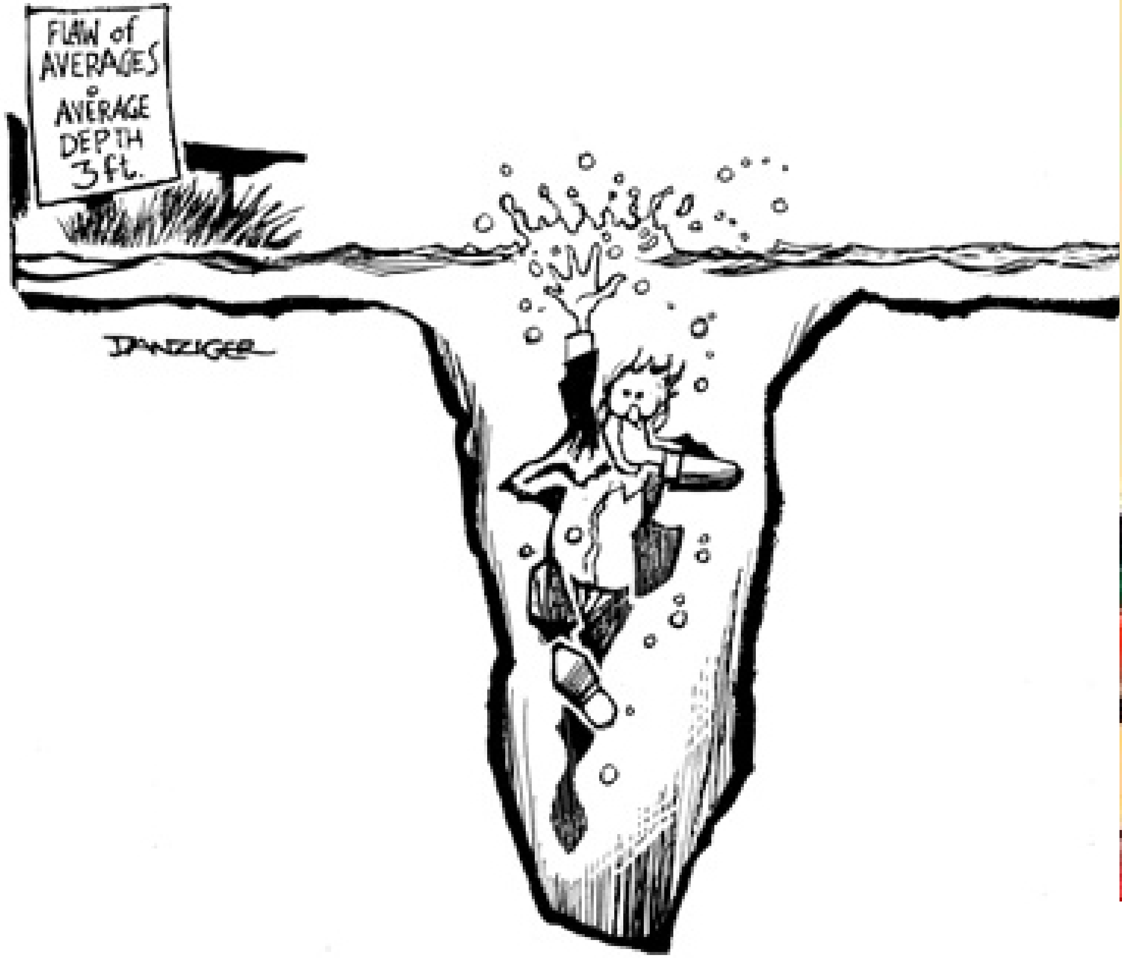
I currently use @Risk for my personal retirement planning

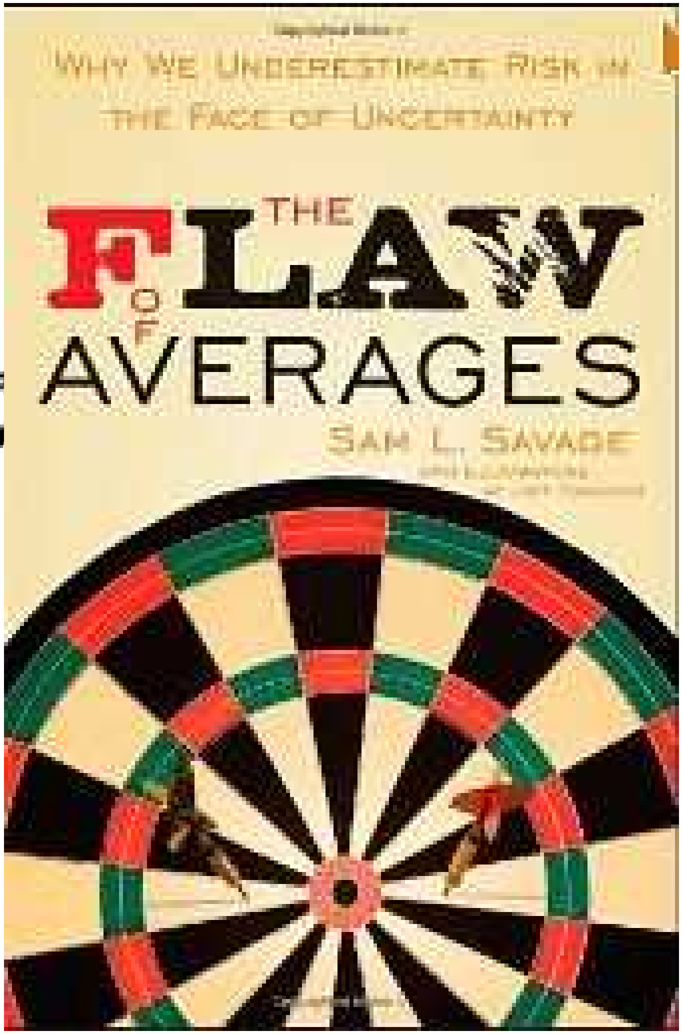
Assessments performed with my personal computer, on my personal time, with my own personally purchased software



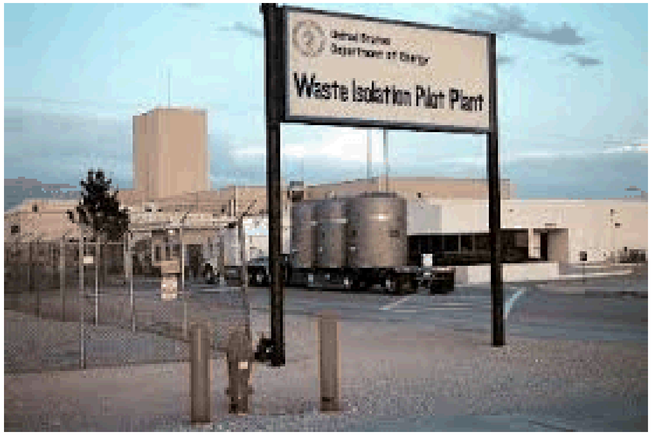
Using Average Inputs

"An estimate based on the assumption that average conditions will occur will almost always be wrong"



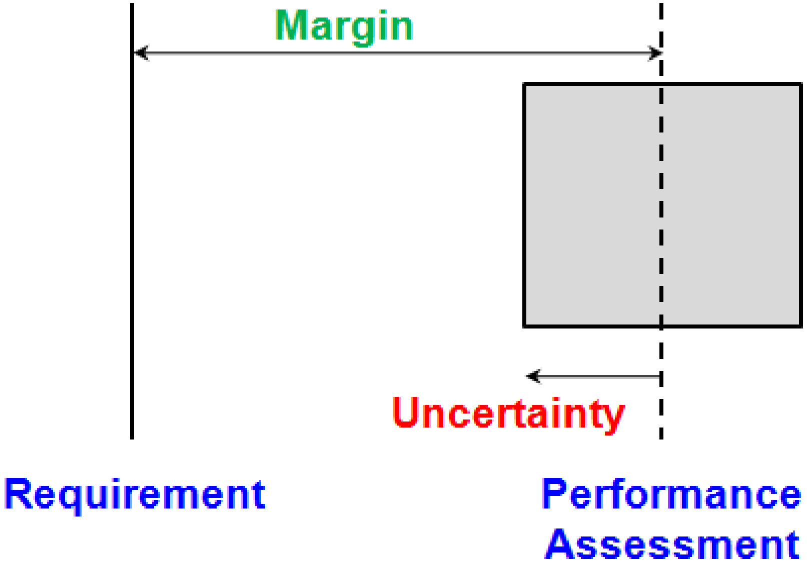


SNL has a rich history of risk assessment for high consequence applications



Best practice is to separate aleatory and epistemic uncertainties

The National Academy of Sciences has weighed in with an opinion



QMU = Margin / Uncertainty

EVALUATION OF QUANTIFICATION OF MARGINS AND UNCERTAINTIES METHODOLOGY FOR ASSESSING AND CERTIFYING THE RELIABILITY OF THE NUCLEAR STOCKPILE, National Research Council of the National Academy of Sciences, 2008

1. Take a system perspective
2. Separate aleatory and epistemic uncertainties

Uncertainty has a dual personality

Jon C. Helton, *Conceptual and Computational Basis for the Quantification of Margins and Uncertainty*, SAND2009-3055

- **Aleatory uncertainty: (perceived) randomness in the occurrence of future events (frequency interpretation)**
- **Epistemic uncertainty: Lack of knowledge wrt appropriate value to use for a quantity that has a fixed value in the context of a specific analysis (confidence or belief interpretation) – Associate "Probability" or "Subjective Probability" with Epistemic Uncertainty**

Aleatory

Epistemic

Does the distinction between aleatory and epistemic uncertainty matter?

Pure Aleatory
Frequency of failure is 2%

100% confident that 2% of us will not get home safely

Attribute of the system, therefore not reducible

Pure Epistemic
Probability of failure is 2%

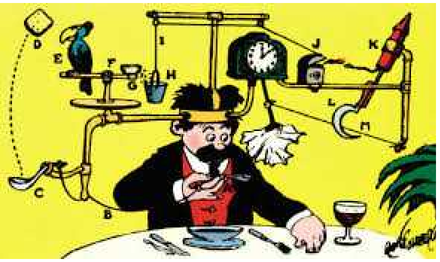
2% belief that none of us will get home safely

Attribute of the assessor, therefore reducible

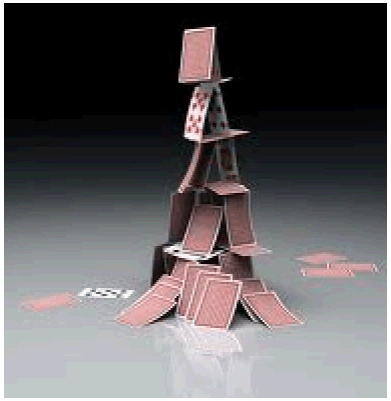
A contrived case study with features and methodologies representative of real applications

Subsystem A

Subsystem B



The “Package”



Delivery

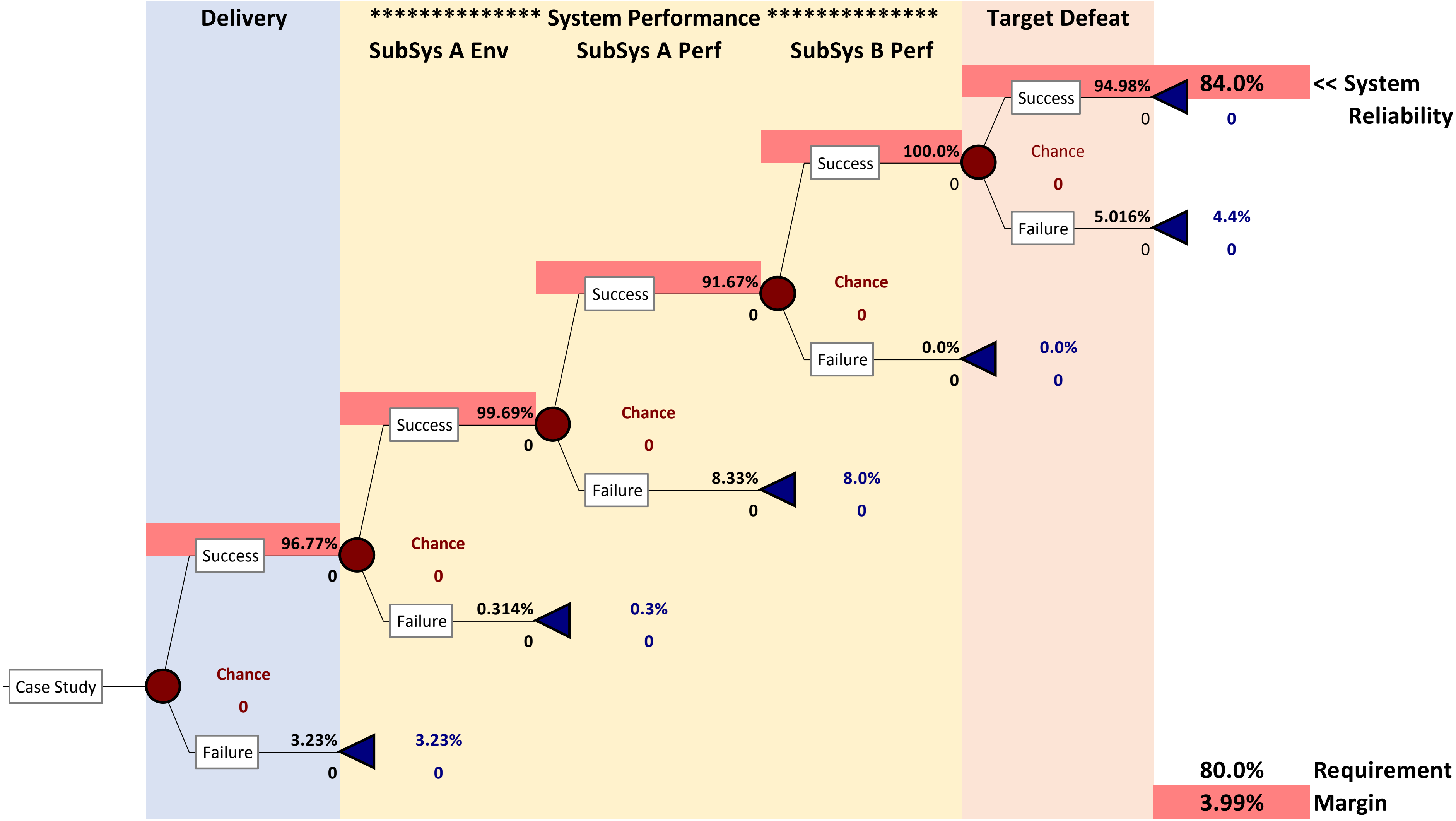
Survivability and Performance

Target Defeat

Requirement:

Overall end-to-end reliability > 80%

System reliability is a frequency statement

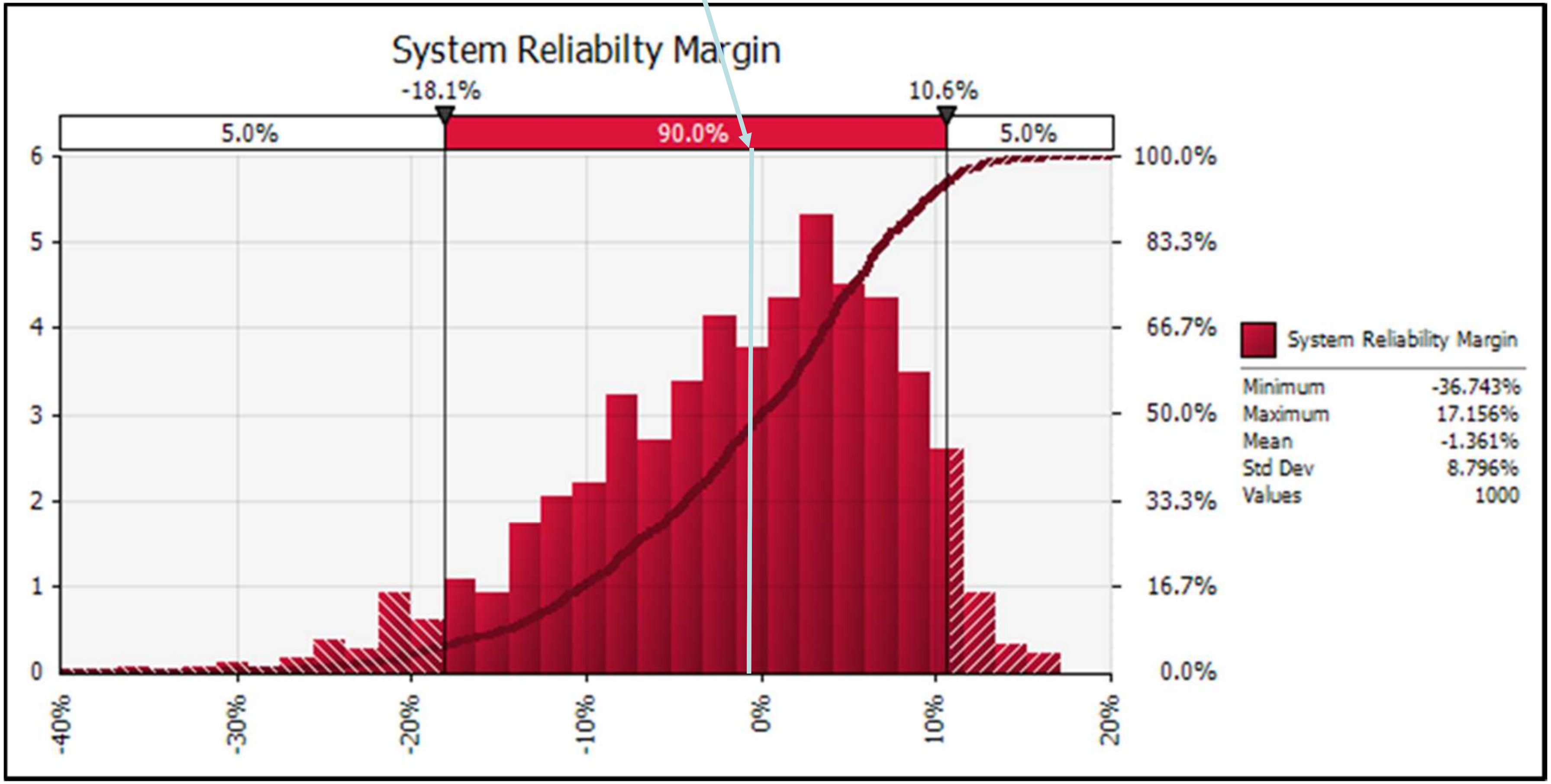


Presentation of the sausage

Yum!

Uncertainty in our assessment of reliability

Based on mean branch reliabilities: -1.4%

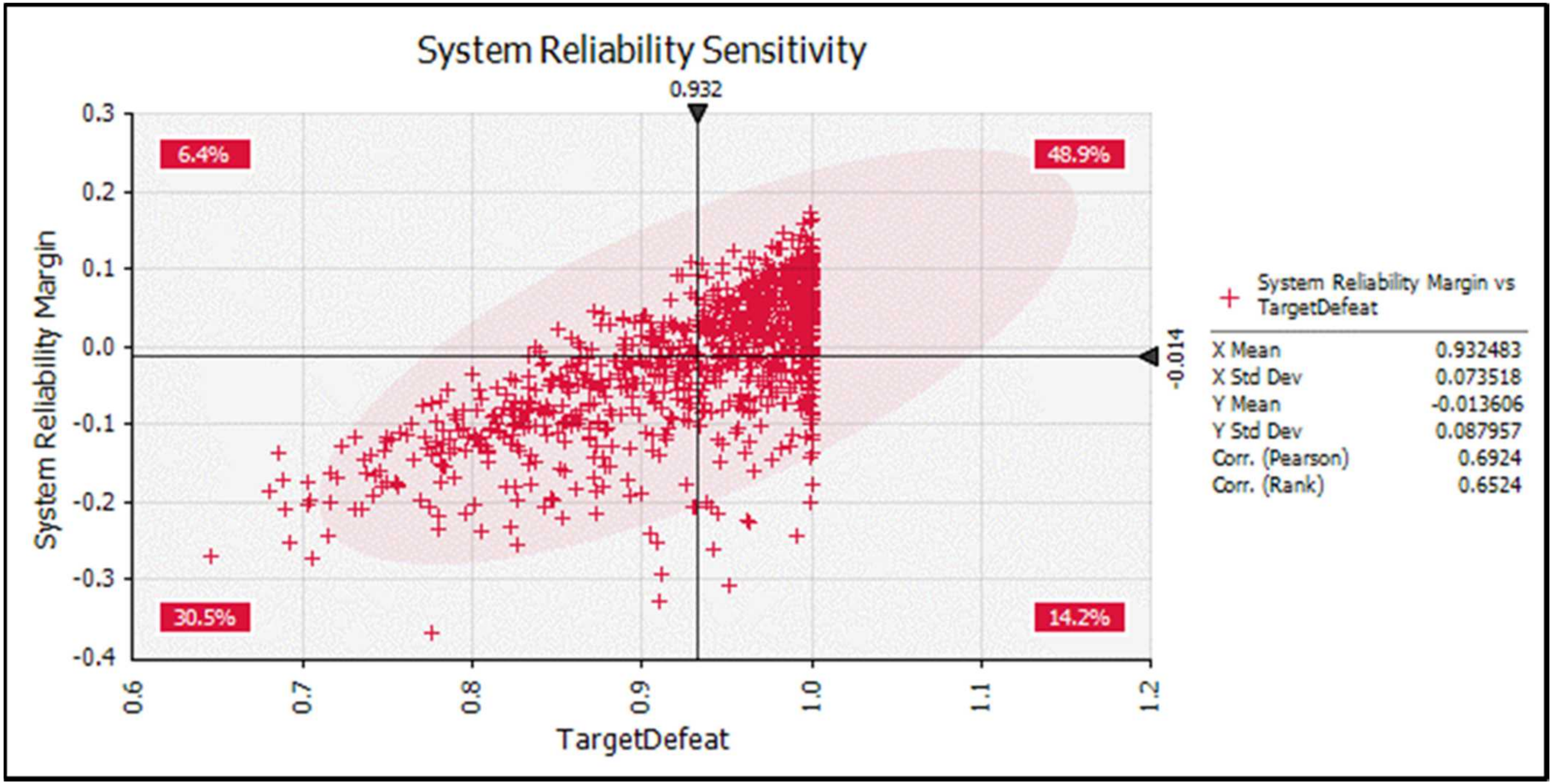


49.7% **Probability of positive margin**

Sensitivity analysis tells you what branch unc. contribute most to system unc.

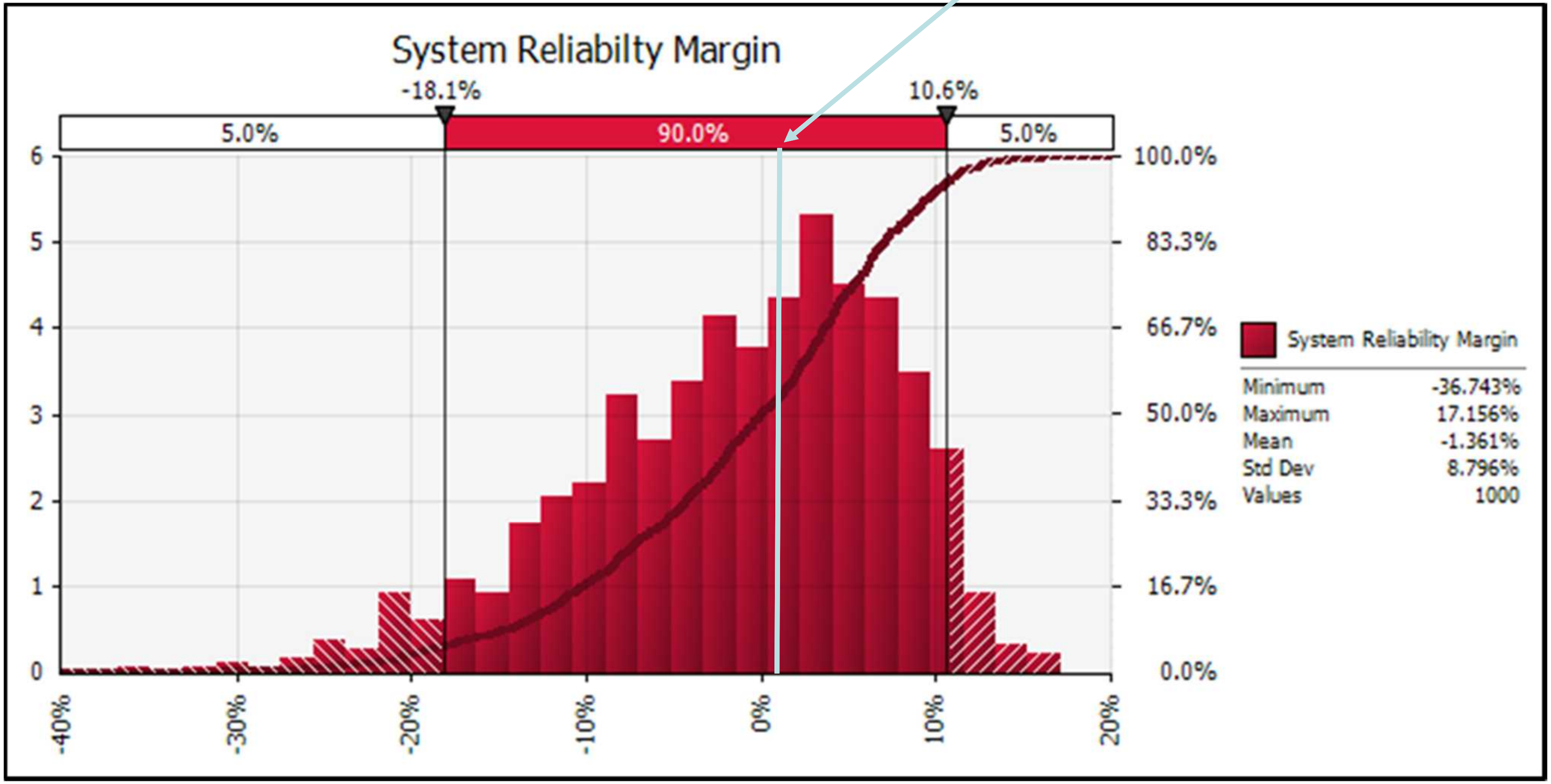
	r: Pearson Correlation	r^2
Delivery	0.285	0.081
SubSysA Env Survivability	0.551	0.304
SubSys A Perf	0.325	0.106
SubSysB Perf	0.000	0.000
Target Defeat	0.732	0.536
	sum r^2 =	1.026

Spend resources ($'s, time) to reduce uncertainty only if it can change decisions



We know the truth for this demonstration

Truth: 1.5%



Minimum	-36.743%
Maximum	17.156%
Mean	-1.361%
Std Dev	8.796%
Values	1000

49.7% **Probability of positive margin**

Spend some time on the reliability tab in the spreadsheet

- Turn your Mac into a PC
  - "Parallels" to run Windows OS and Excel
  - Launch blank Excel file, load Palisade products, open application file
- PrecisionTree settings to work with @Risk
  - Ribbon: Settings – Model Settings - @Risk – Select: Expected Value of Model
- @Risk settings
  - Ribbon: Settings – General – Select: Multiple CPU Support = Disabled
  - Ribbon: Settings – Sampling – Select: Sampling Type = Monte Carlo
  - Ribbon: Help – Select: Color @Risk Functions
- Reliability model
  - Success branches linked to calculations in other tabs
  - Failure = 1= Success
  - Demonstrate dynamic updating of model

Our customers care about presentation *AND* how the sausage is made

Technical peer review is an integral part of doing business

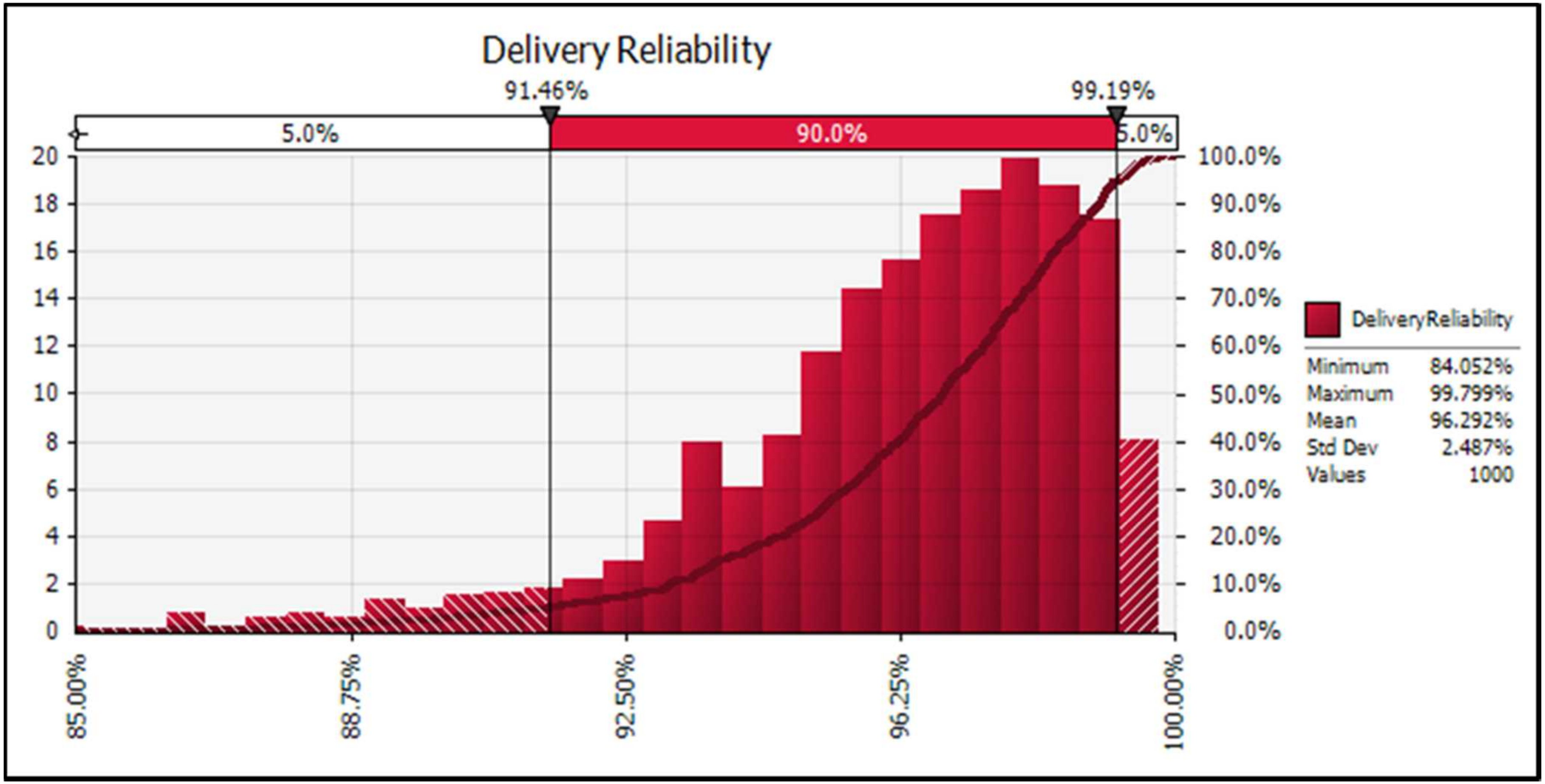
Delivery as an example of binary success/failure types of data

Success

Failure

Estimated launch reliability is uncertain

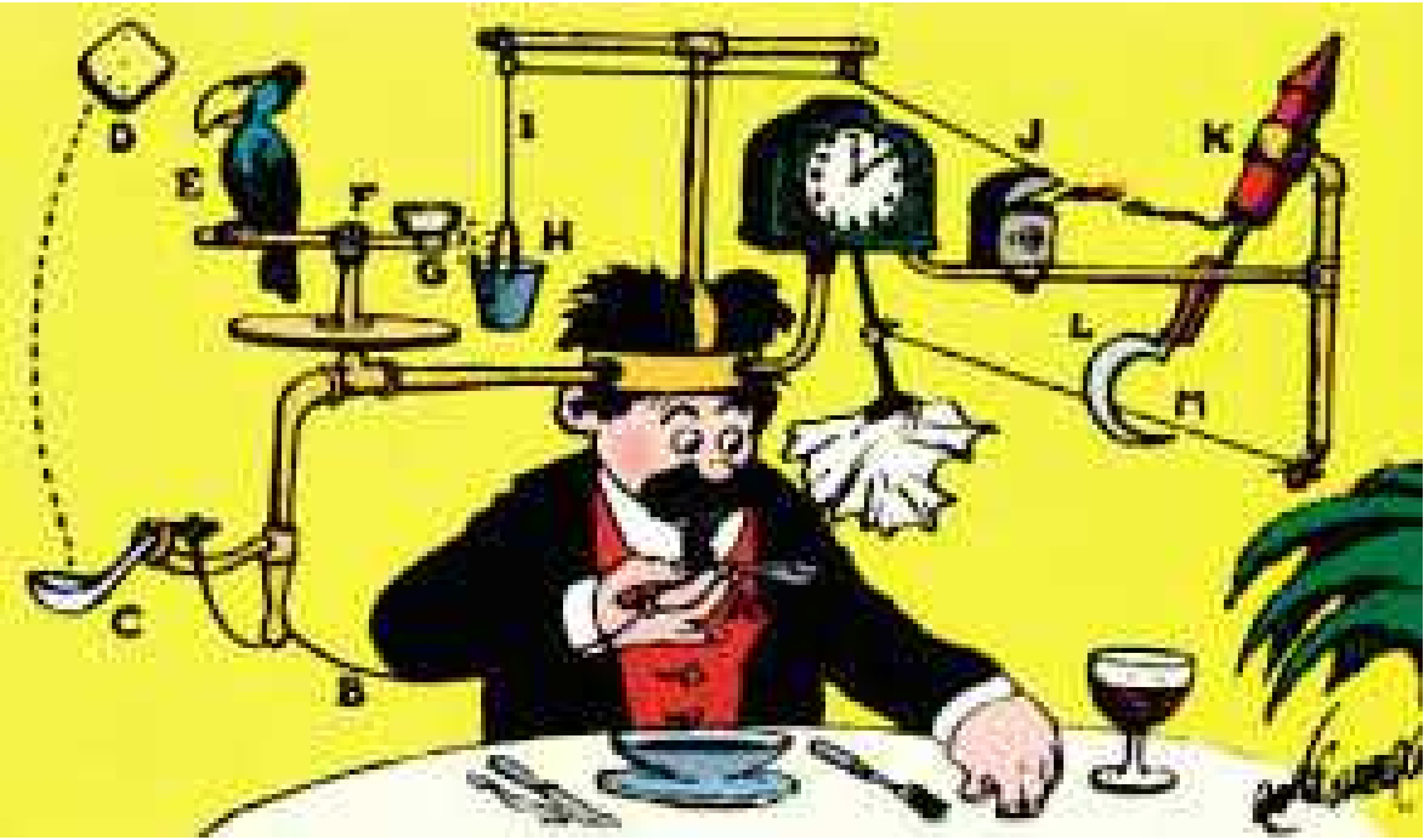
NLaunches	Launch Successes	Observed Reliability		
50	49	0.98	92.1%	<< = RiskBeta(Successes+1,NLaunches-Successes+1)



Spend some time on the delivery tab in the spreadsheet

- Introduce the truth model hidden to the far right
- Dynamic demonstration of different possible data sets and resulting success/failure histories
- Assessment based on one specific data set
- Pre-ran 1000 iterations
  - Ribbon: Simulation – Select: Iterations=1000
  - Ribbon: Results – Browse Results
- Updating of embedded graphs and simulation results
  - Ribbon: Insert Function – Miscellaneous – RiskResultsGraph
  - Ribbon: Insert Function – Simulation Results - RiskMean

Subsystem A makes the system go boom



Subsystem A has to survive extreme environments and perform!

Research
The new whole body vibration
machines are a far cry from the
"jelly jigglers" of the early 1900

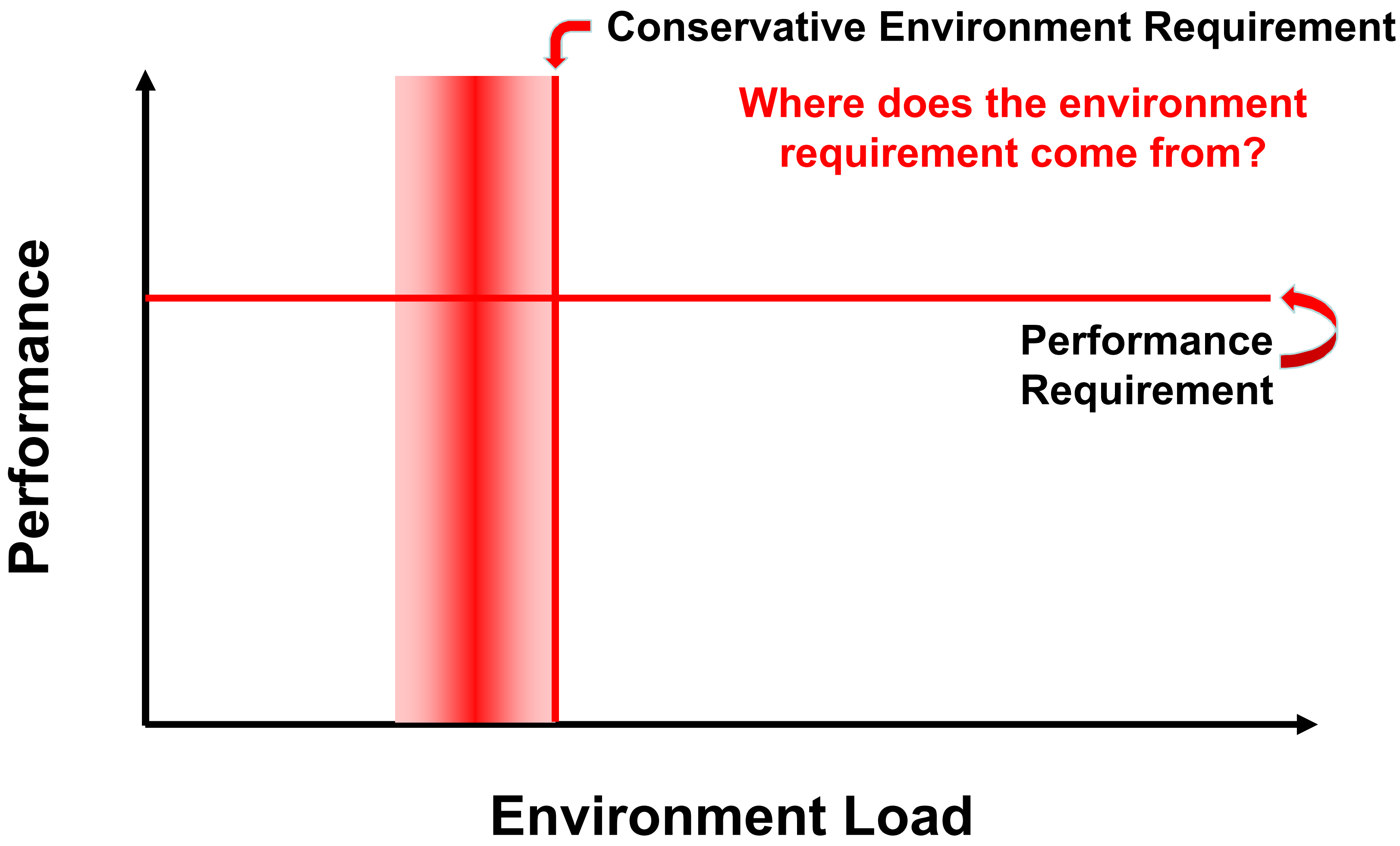
Vibration

Thermal

Radiation

Shock

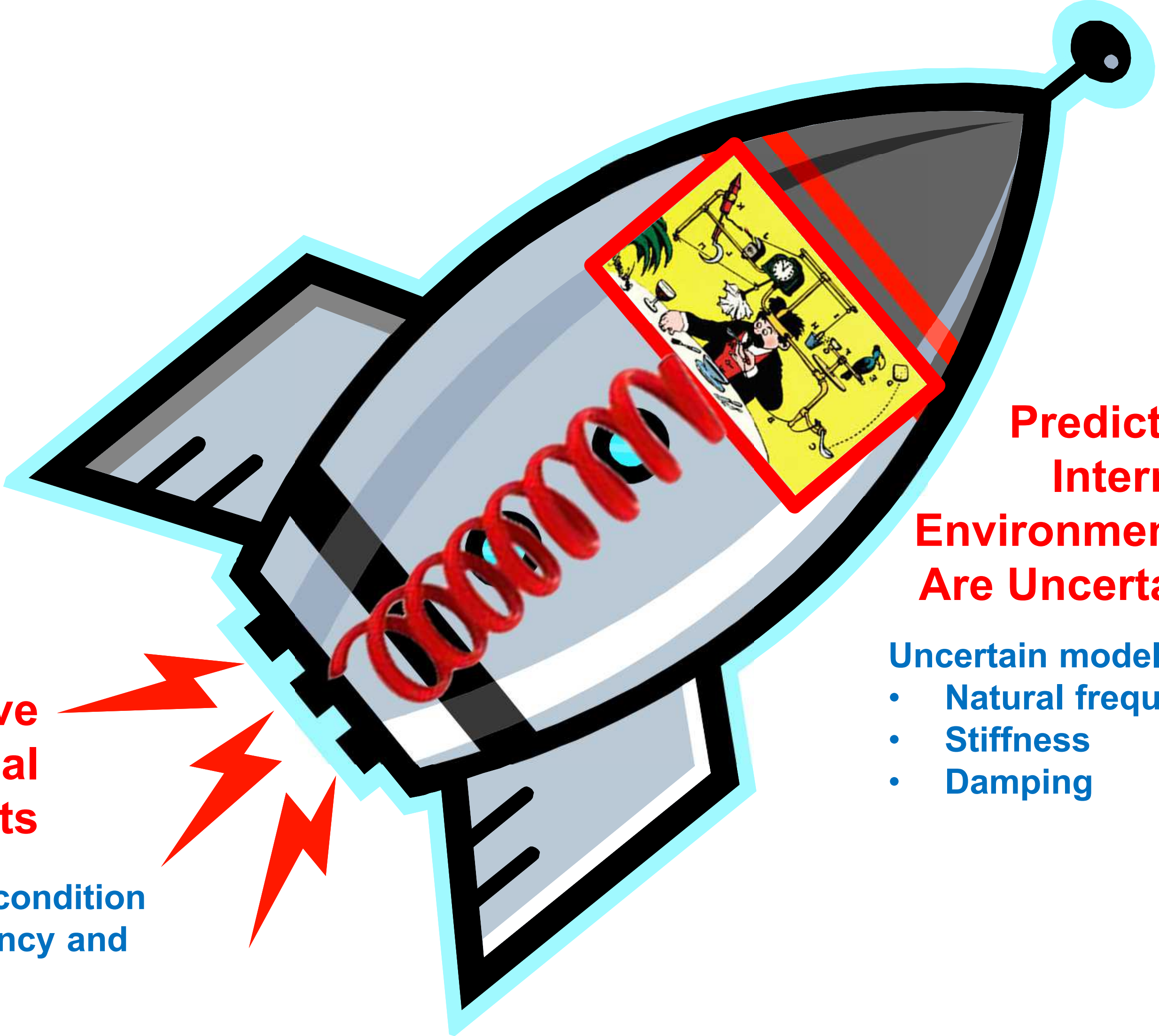
There are both performance requirements and survivability requirements for SubSys A


Conservative Environment Requirement
Where does the environment requirement come from?
Performance Requirement
Performance
Environment Load

Big computer codes are used to translate external environments to internal environments

Research
The new whole body vibration
machines are a far cry from the
"jelly jigglers" of the early 1900

Vibration Environments



Predicted Internal Environments Are Uncertain

Uncertain model parameters

- Natural frequency
- Stiffness
- Damping

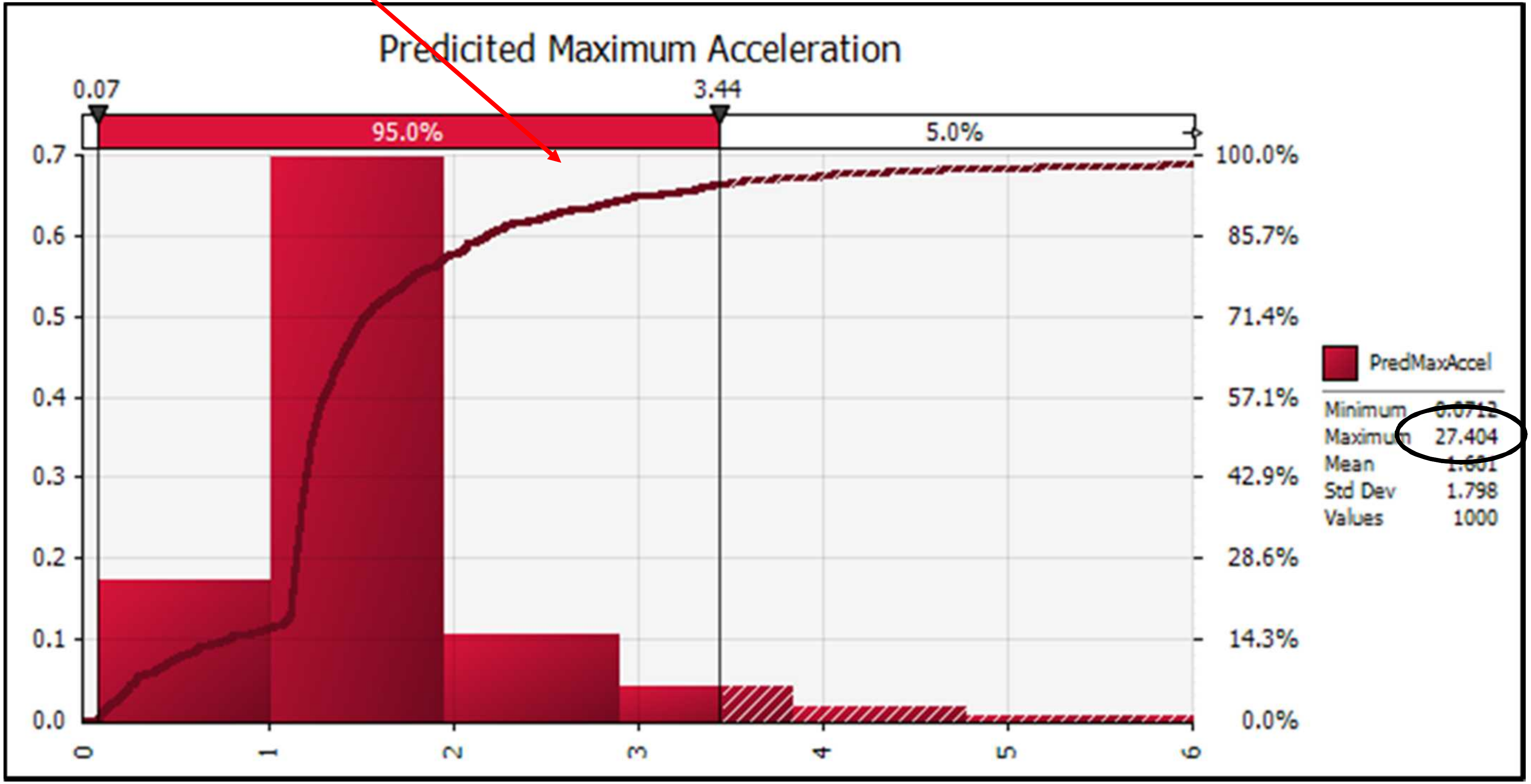
Conservative External Environments

Known boundary condition

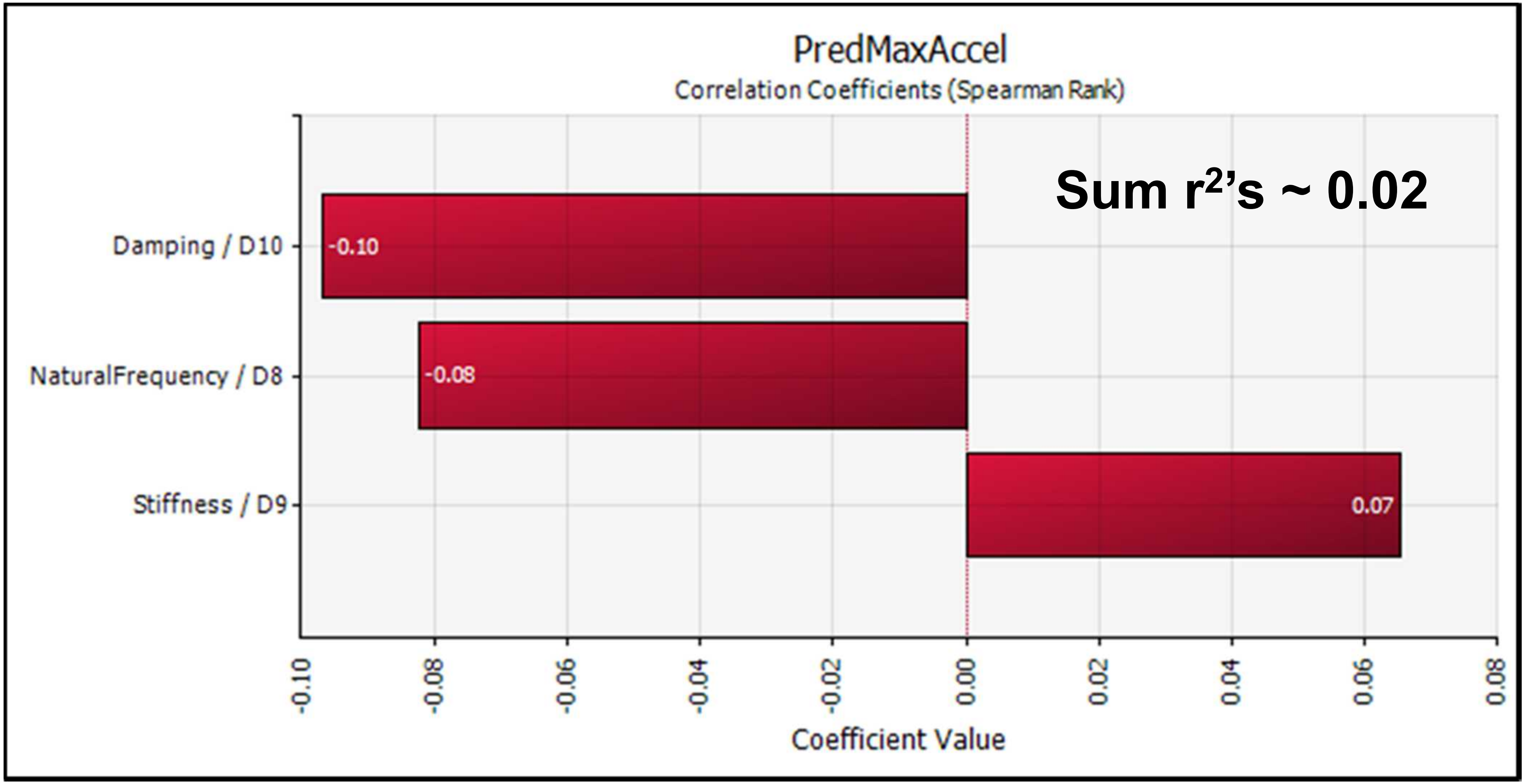
- Forcing frequency and magnitude

95% confidence limit used as SubSys A environment requirement

Distribution is bi-modal



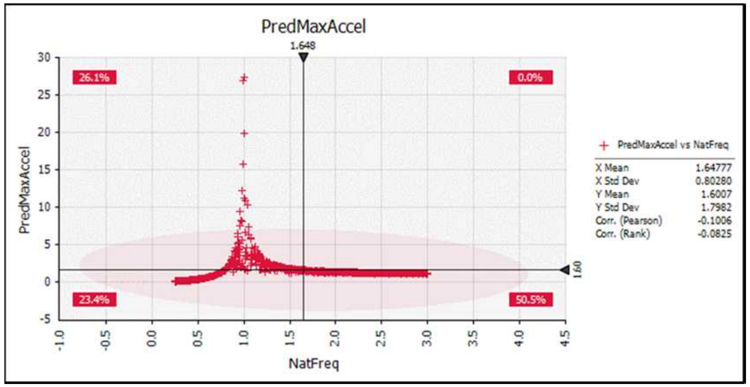
What can you say about contributors to uncertainty?



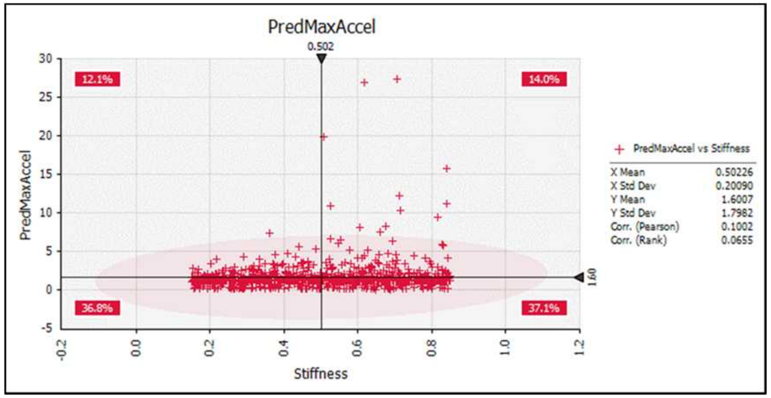
Sum r²’s ~ 0.02

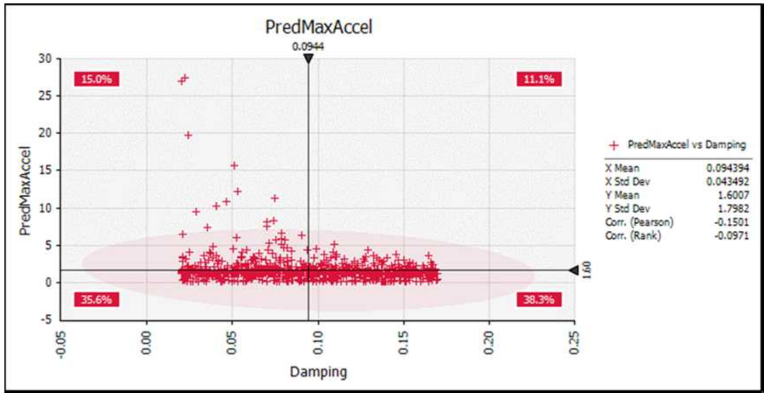
Answer: Nothing is important! How can that be?

Always examine the scatter plots



Uncertainties in natural frequency dominate

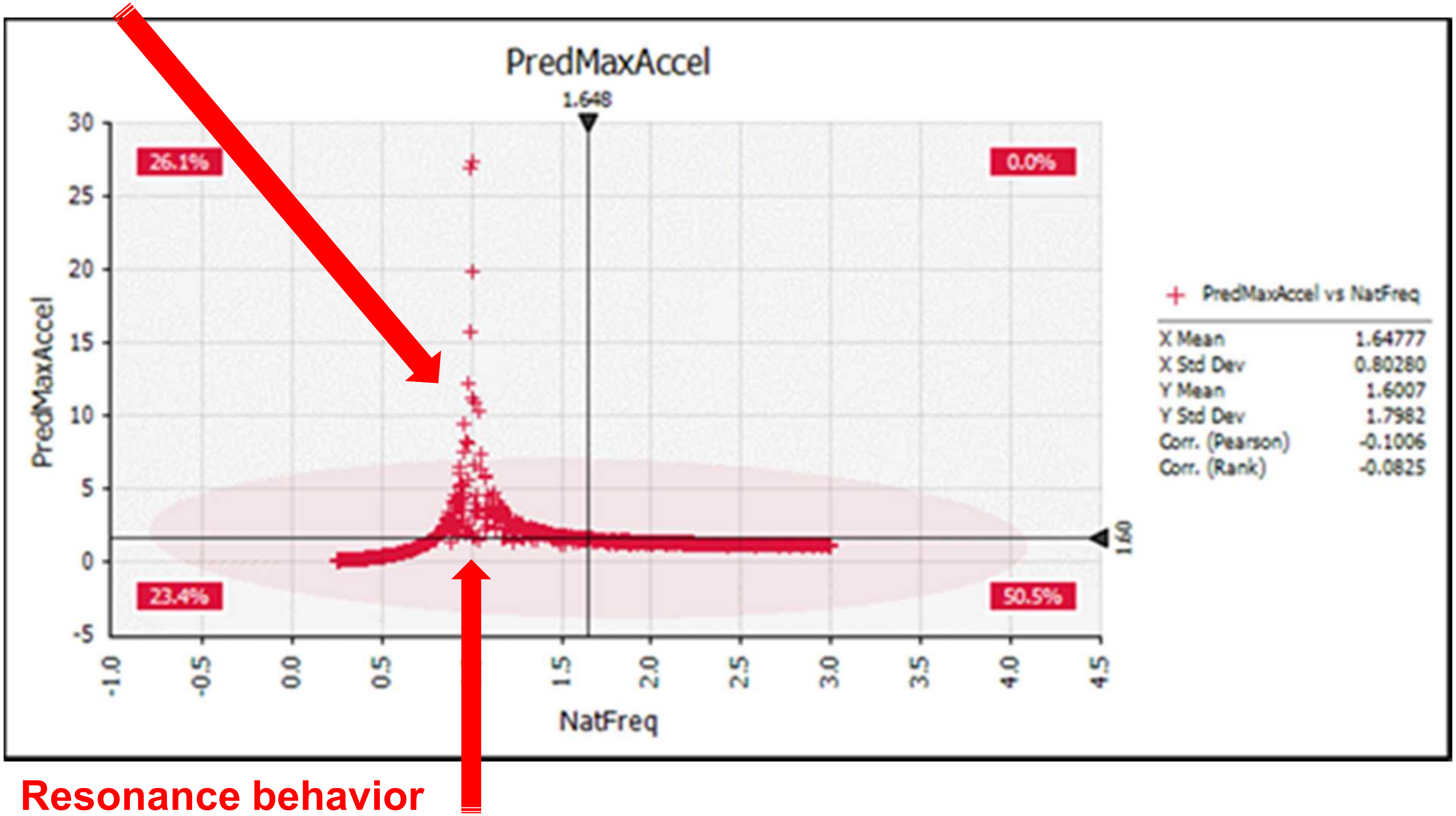




Peel the onion, understand *what the* results are really telling you

Highly non-linear non-monotonic dependence on natural frequency

- **Correlation coefficients are measures of linear association (*bad* assumption)**

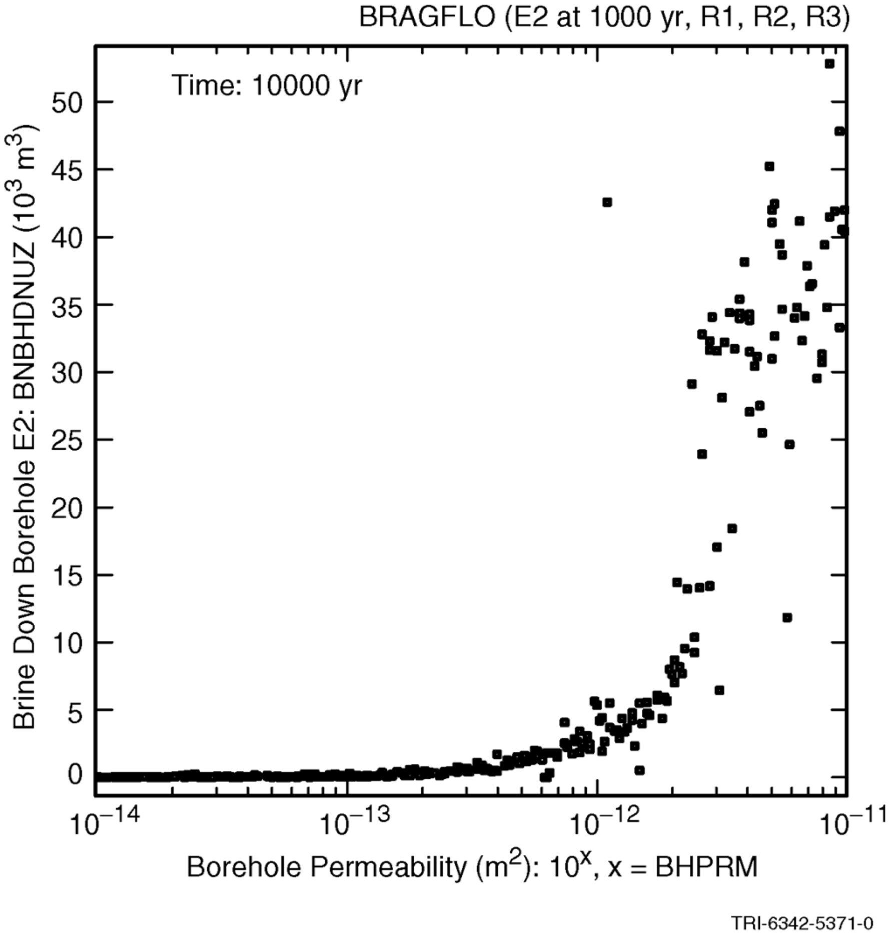


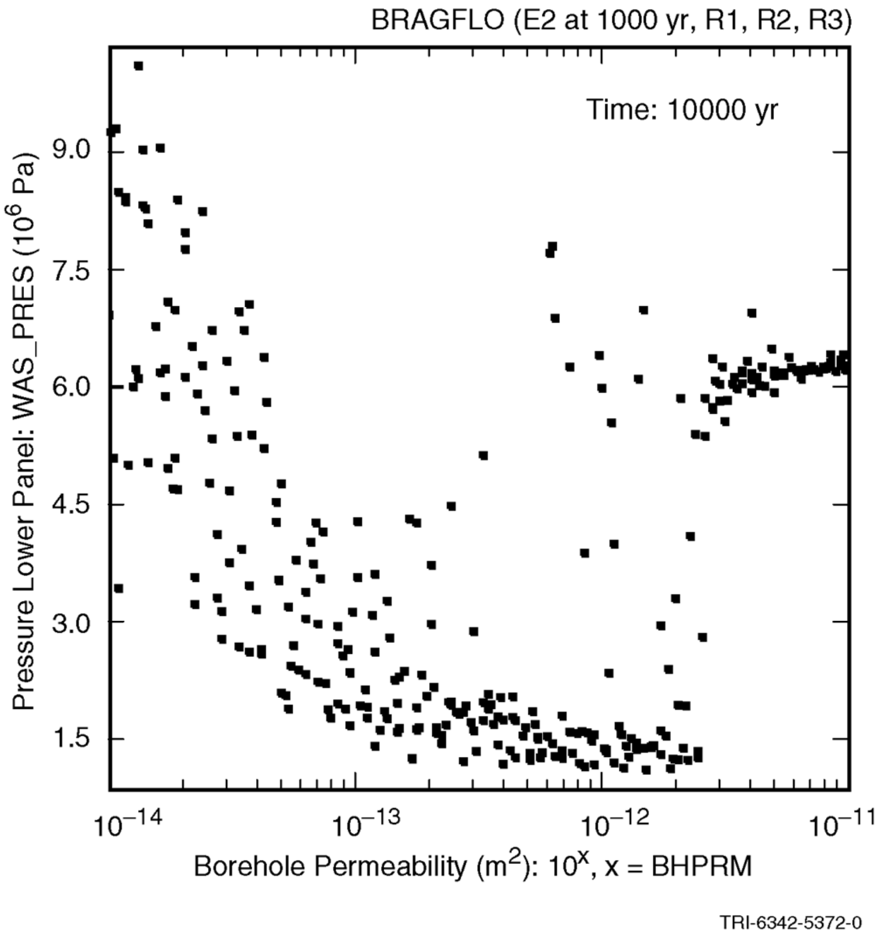
X Mean	1.64777
X Std Dev	0.80280
Y Mean	1.6007
Y Std Dev	1.7982
Corr. (Pearson)	-0.1006
Corr. (Rank)	-0.0825

Resonance behavior

- **Extreme response does not occur at the extremes of the input**

Assumptions of linearity could miss-characterize important features



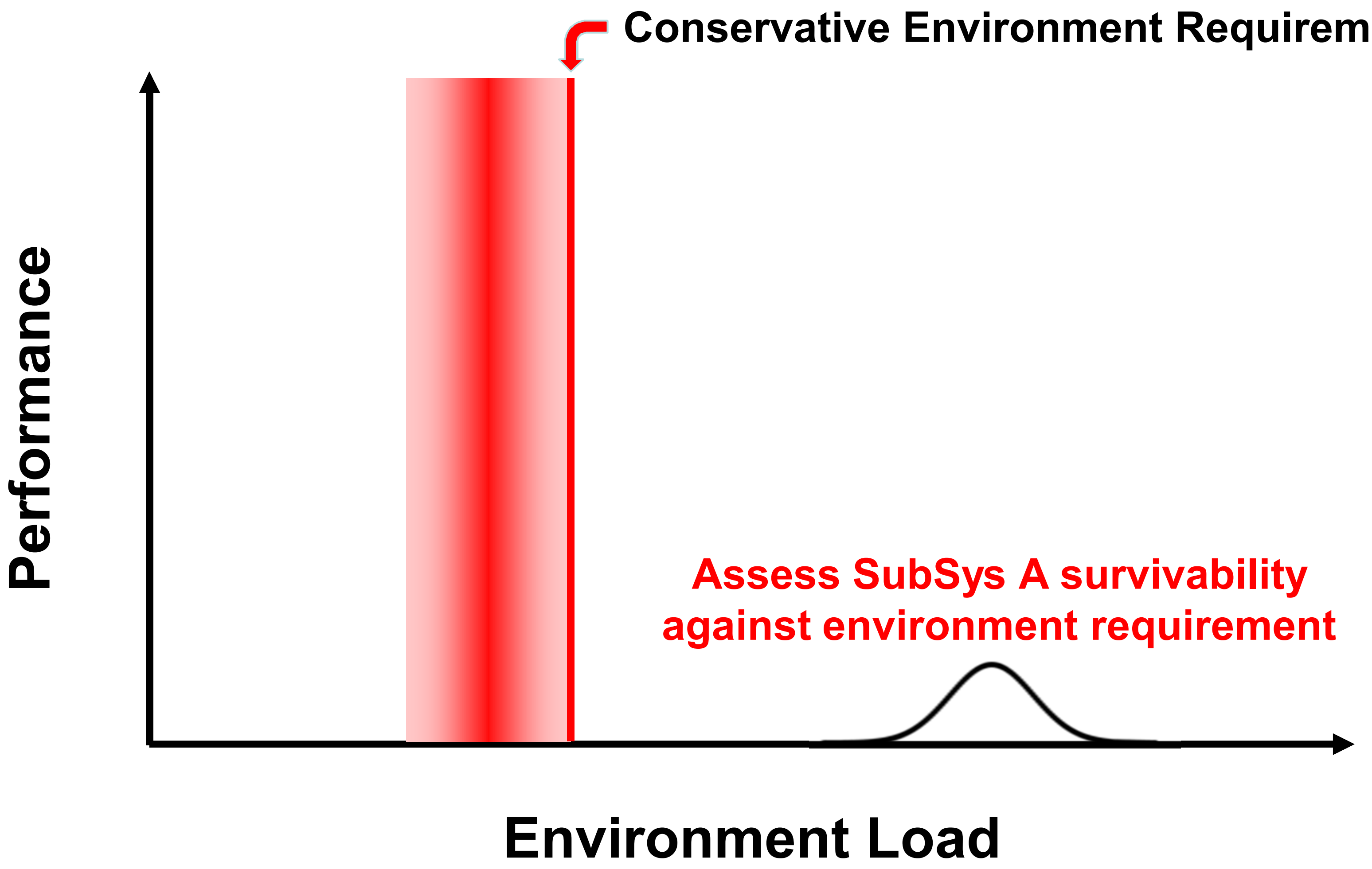


Sensitivity study form WIPP performance assessment

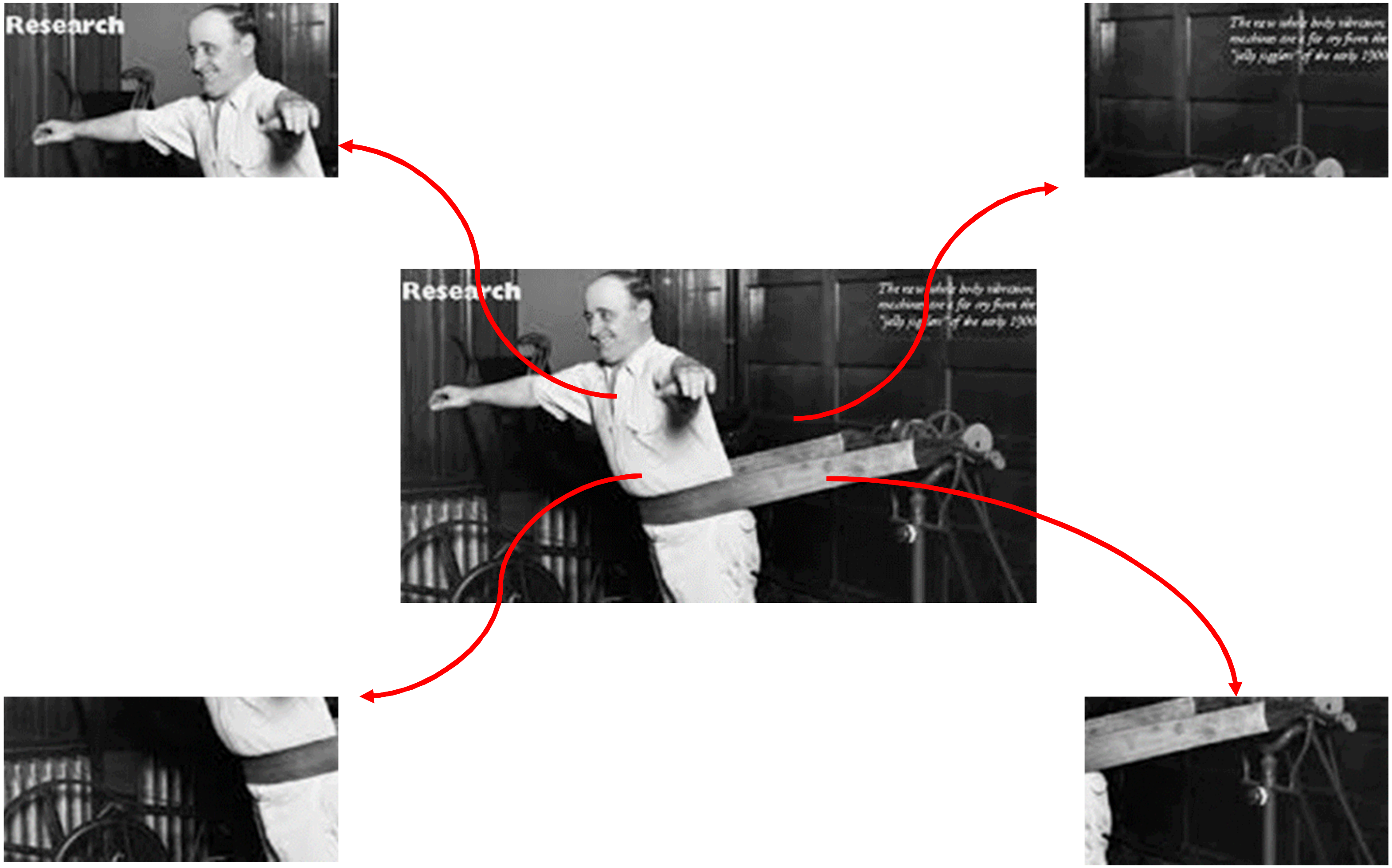
Spend some time on the environment requirement tab in the spreadsheet

- Demonstrate dynamic environment load calculation
  - VBA Function Procedure as replacement for really big computer code

There are survivability requirements on SubSys A

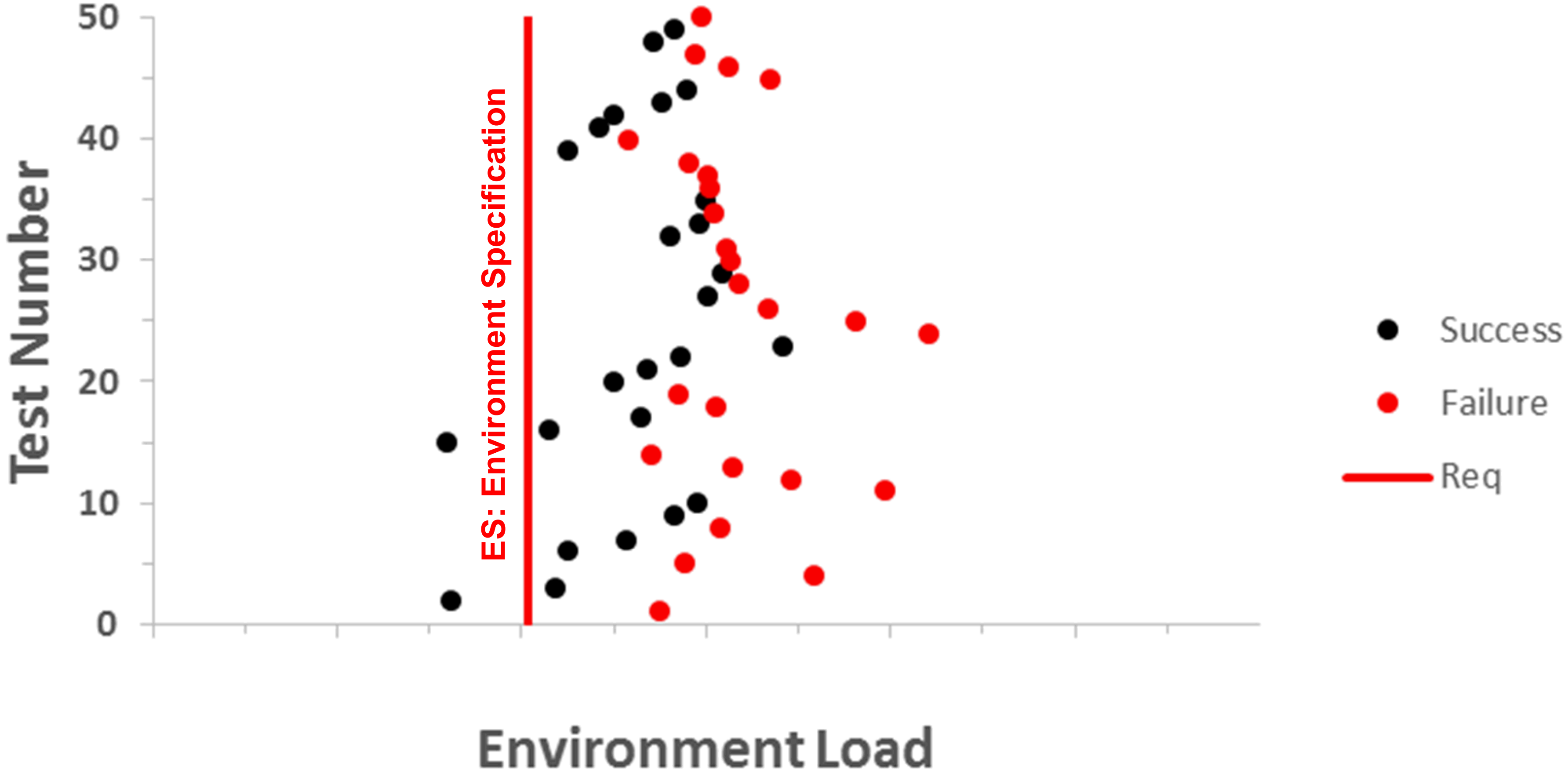


How hard do you have to shake it before it fails catastrophically?

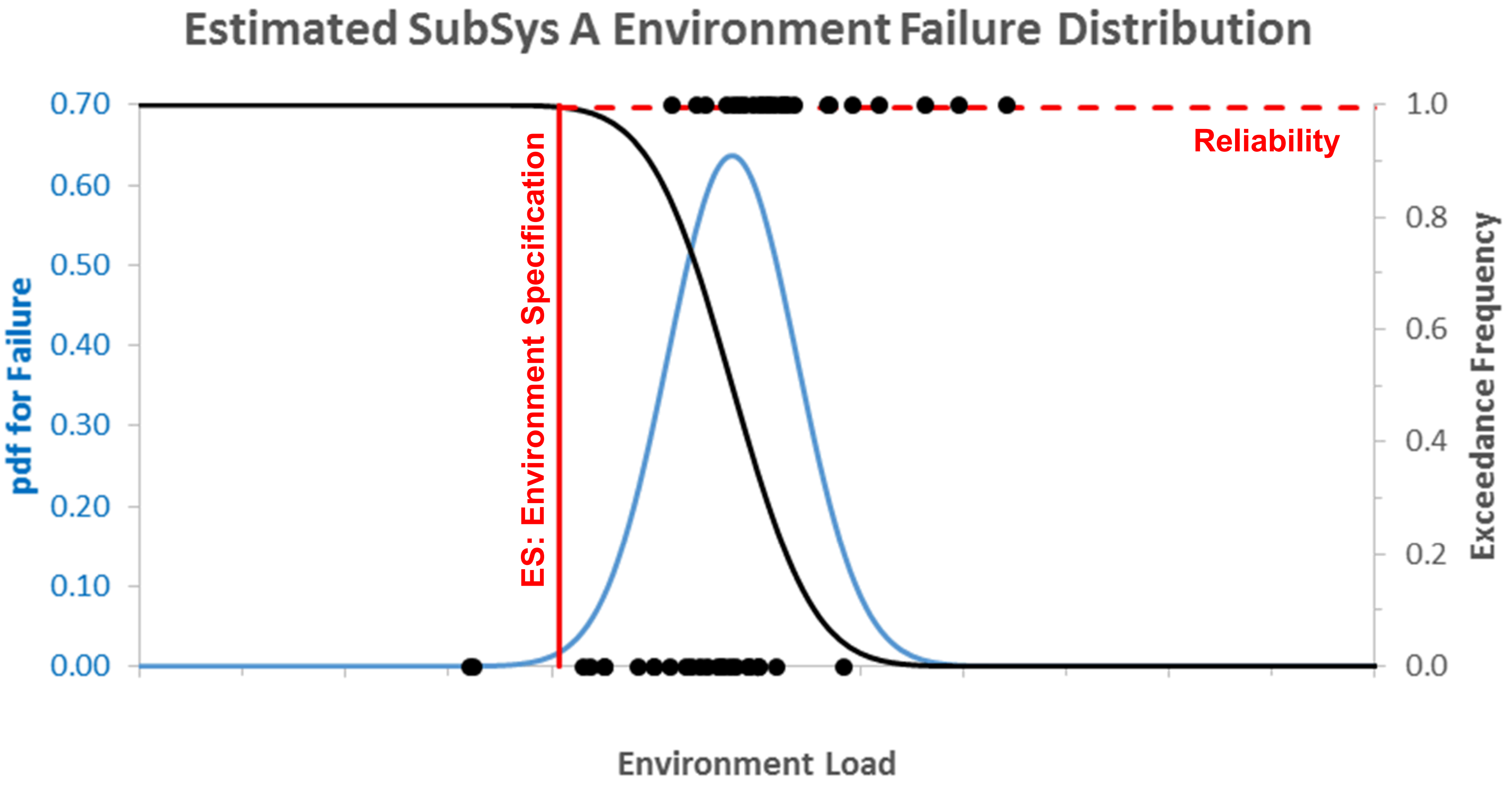


One shot tests

We look for a survivability transition by doing over-tests with different loads on different units

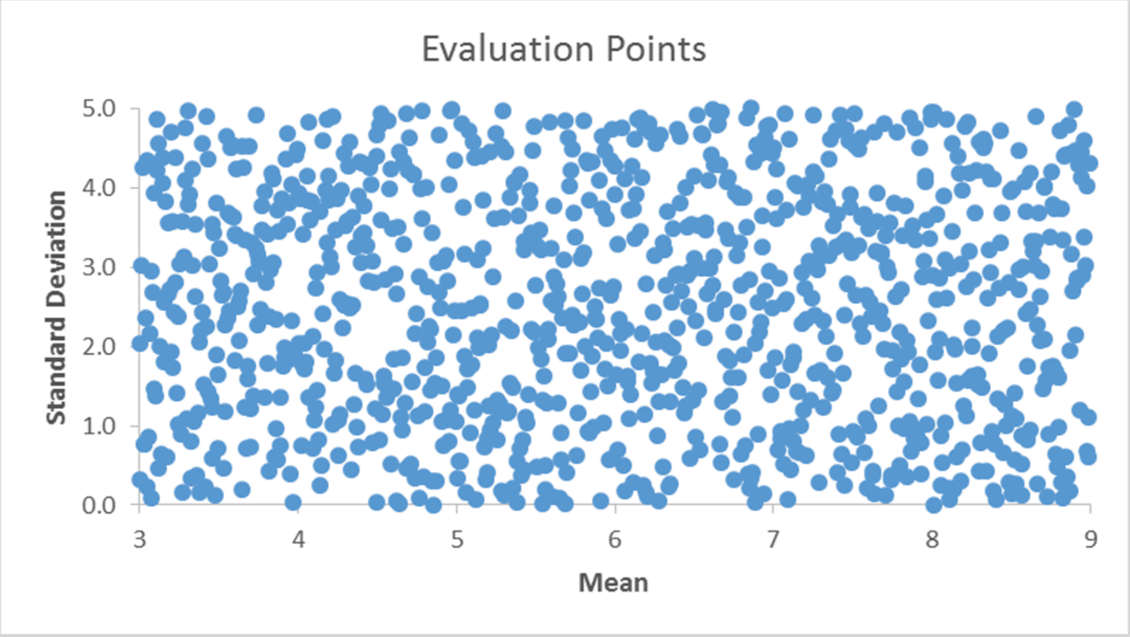


We want to transform success/failure data into a failure distribution as a function of load



Assume a normal (frequency) distribution with uncertain mean and standard deviation, Norm(μ,σ)

- Select evaluation points for μ,σ by casting a wide net



Mean = RiskUniform(3,9)
Std = RiskUniform(0,5)

- Use Bayes Theorem to calculate probability that any μ,σ pair could have produced the observed data

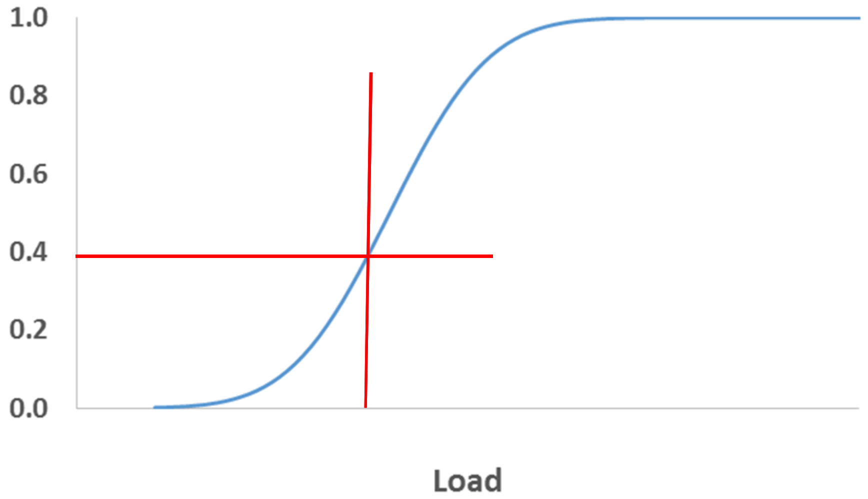
Distribution parameters are calibrated using Bayes Theorem

- Posterior (μ,σ) ~ Likelihood(Data: μ,σ) * Prior(μ) * Prior(σ)
- Un-informed priors for μ,σ
  - Prior(μ) ~ 1 any value is equally likely
  - Prior(σ) ~ 1/σ invariant to translation
- Likelihood(Data: μ,σ)

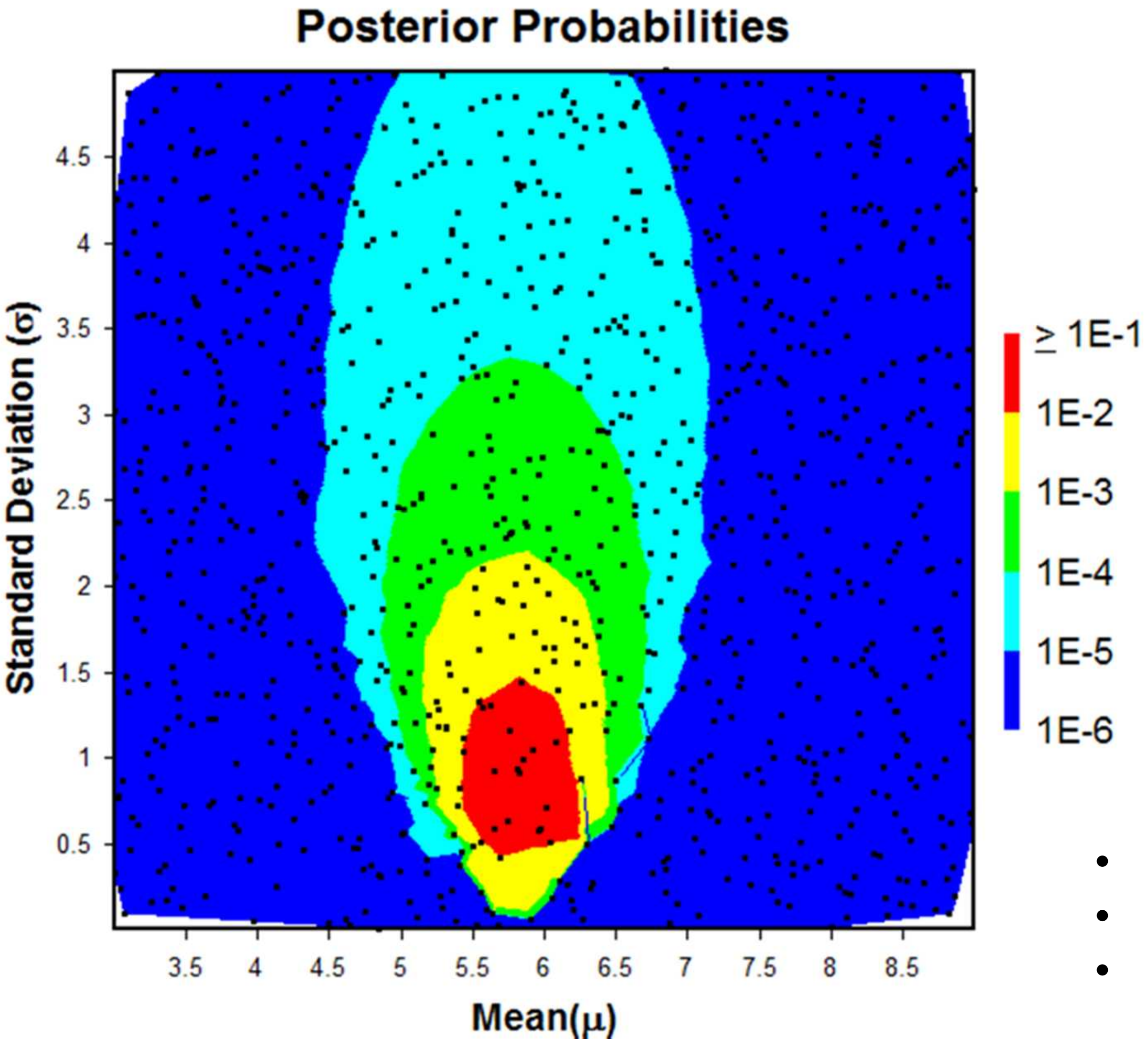
$$Likelihood(Data\colon\ \mu,\sigma) = \sum_{i=1}^{NData} \begin{Bmatrix} Index * CumNorm(\mu,\sigma) \\ +(1-Index) * (1-CumNorm(\mu,\sigma)) \end{Bmatrix}$$

Likelihood of Success:
Index = 0

Likelihood of Failure:
Index = 1

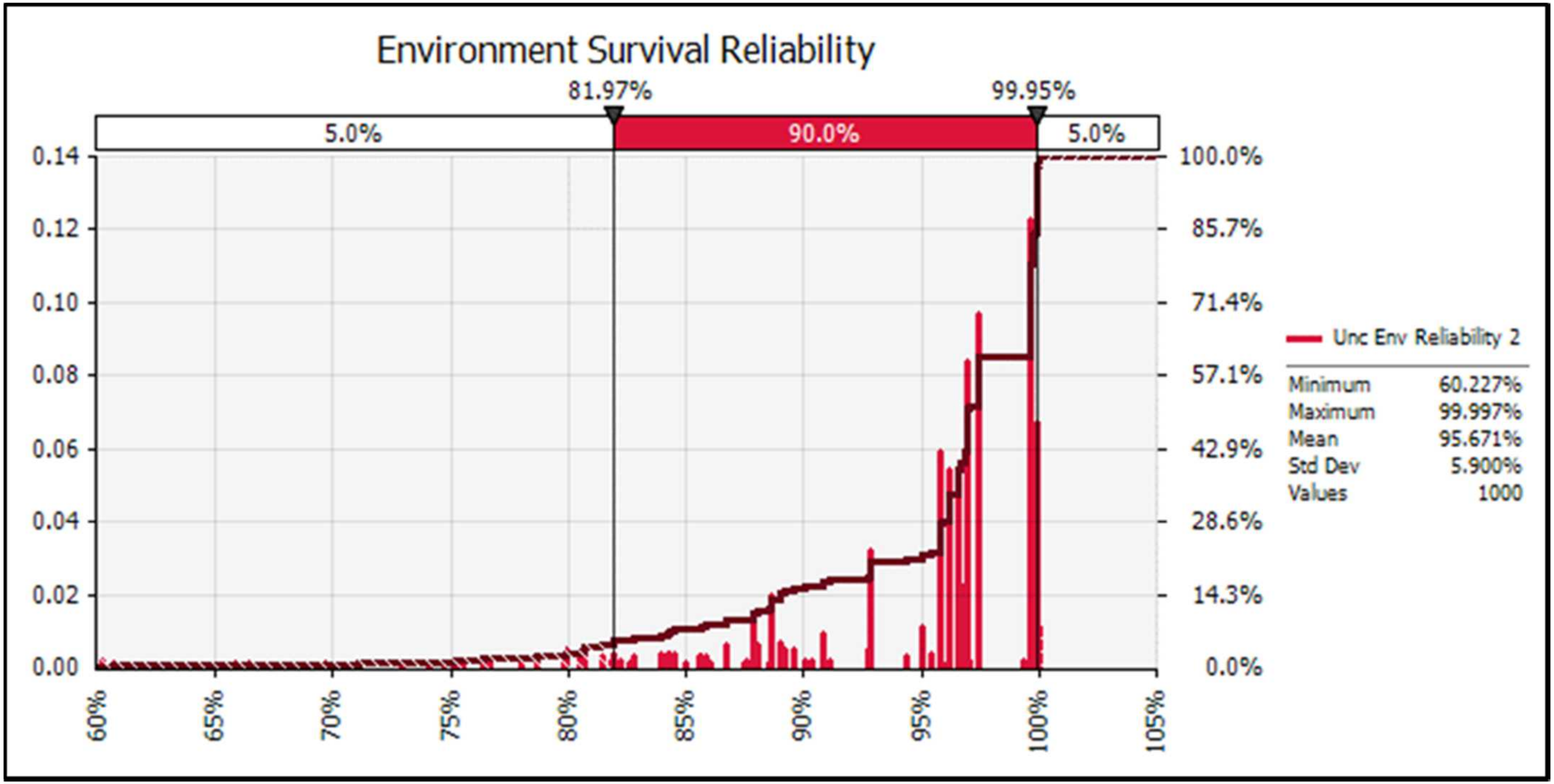


Sample from the posterior probabilities



- EvalPoint = RiskDiscrete
- μ = vlookup
- σ = vlookup

Estimated environment survivability is uncertain

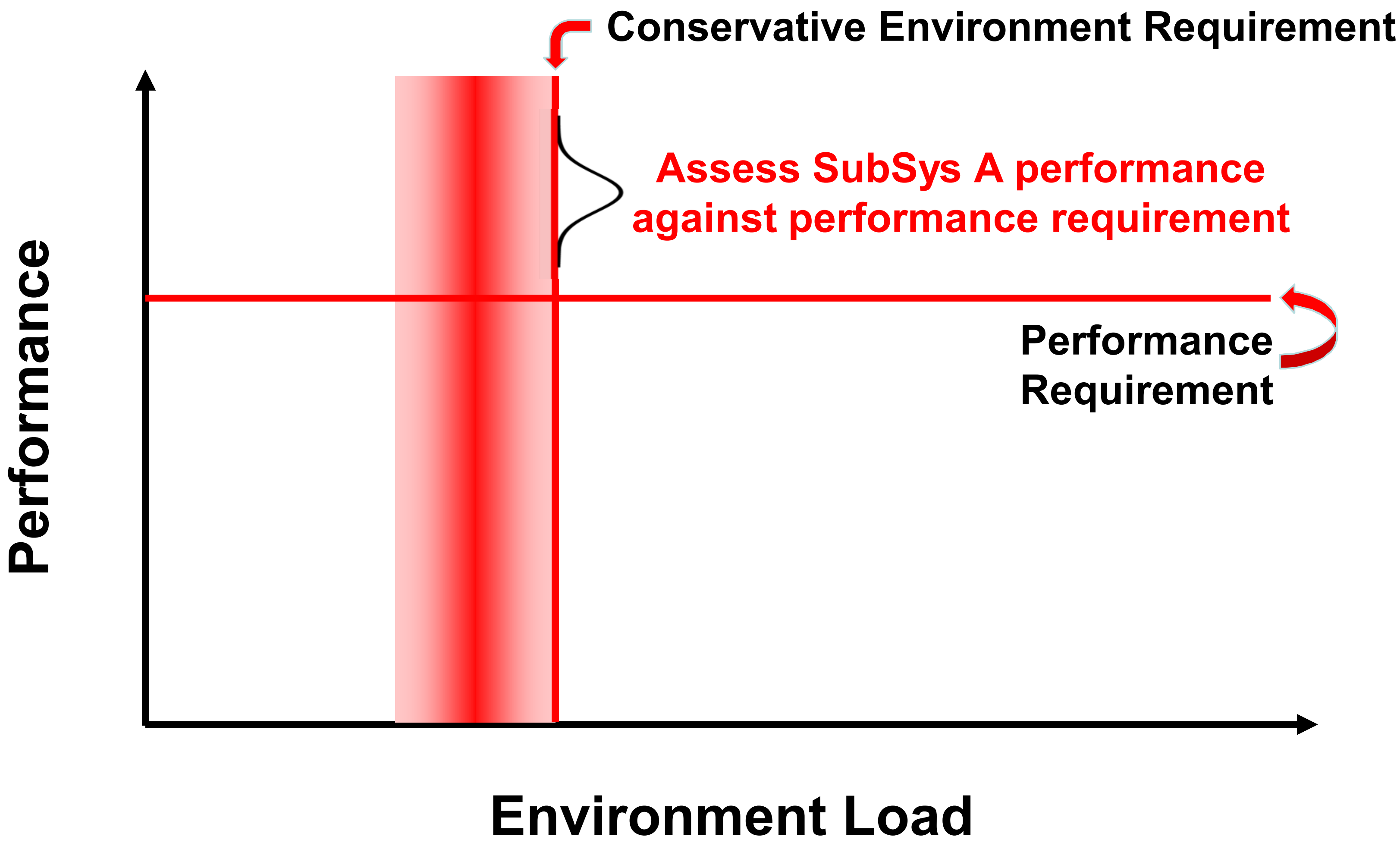


Minimum	60.227%
Maximum	99.997%
Mean	95.671%
Std Dev	5.900%
Values	1000

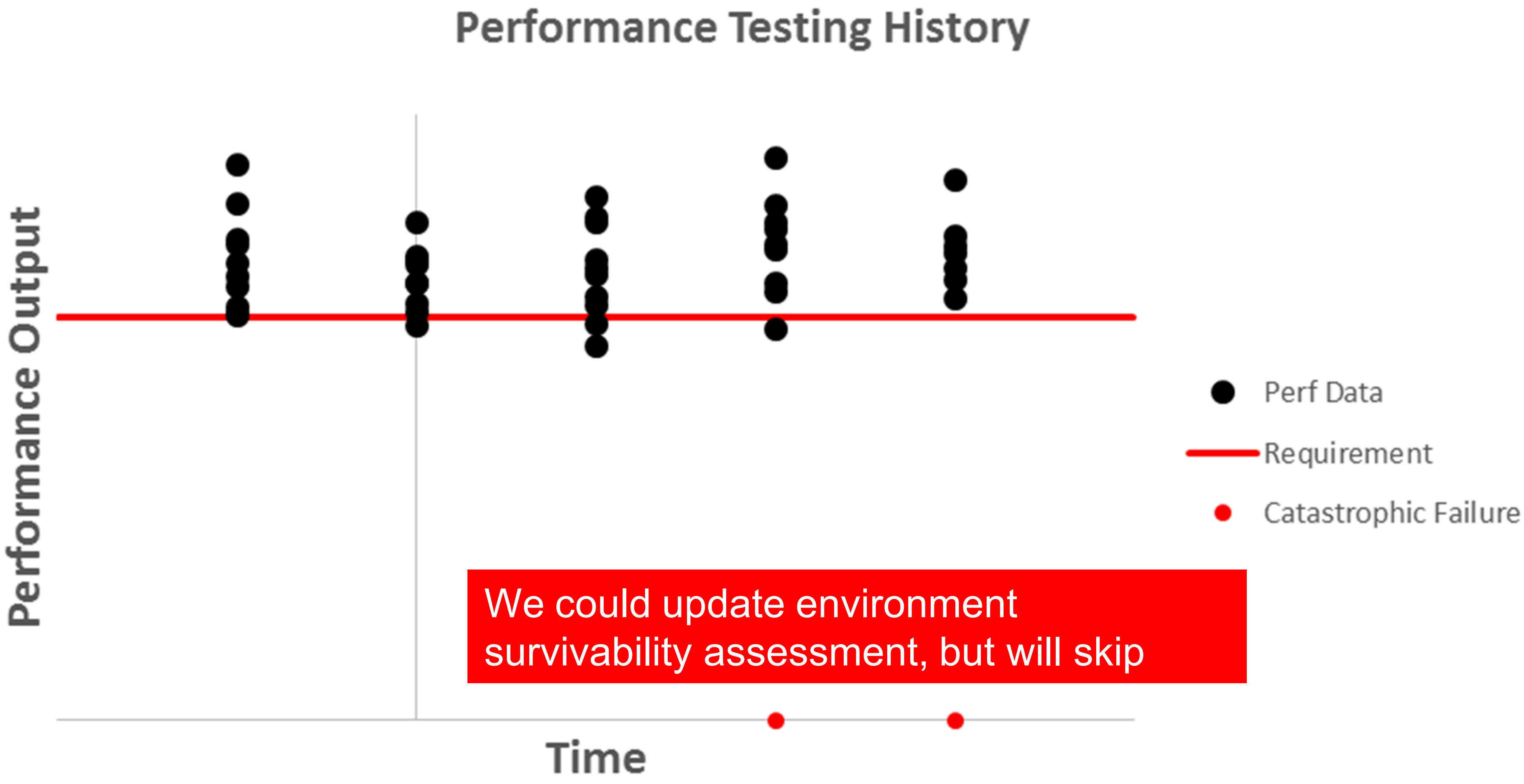
Spend some time on the SubSys A env. survivability tab in the spreadsheet

- Demonstrate dynamic data sets using Langley algorithm
- Walk through Bayesian updating
  - μ,σ evaluation points
  - μ,σ priors
  - Evaluation of likelihood function
  - Posterior and normalized posterior
- Demonstrate sampling of posterior
  - RiskDiscret and VLOOKUP
- Demonstrate dynamic failure load distribution

There are performance requirements on SubSys A

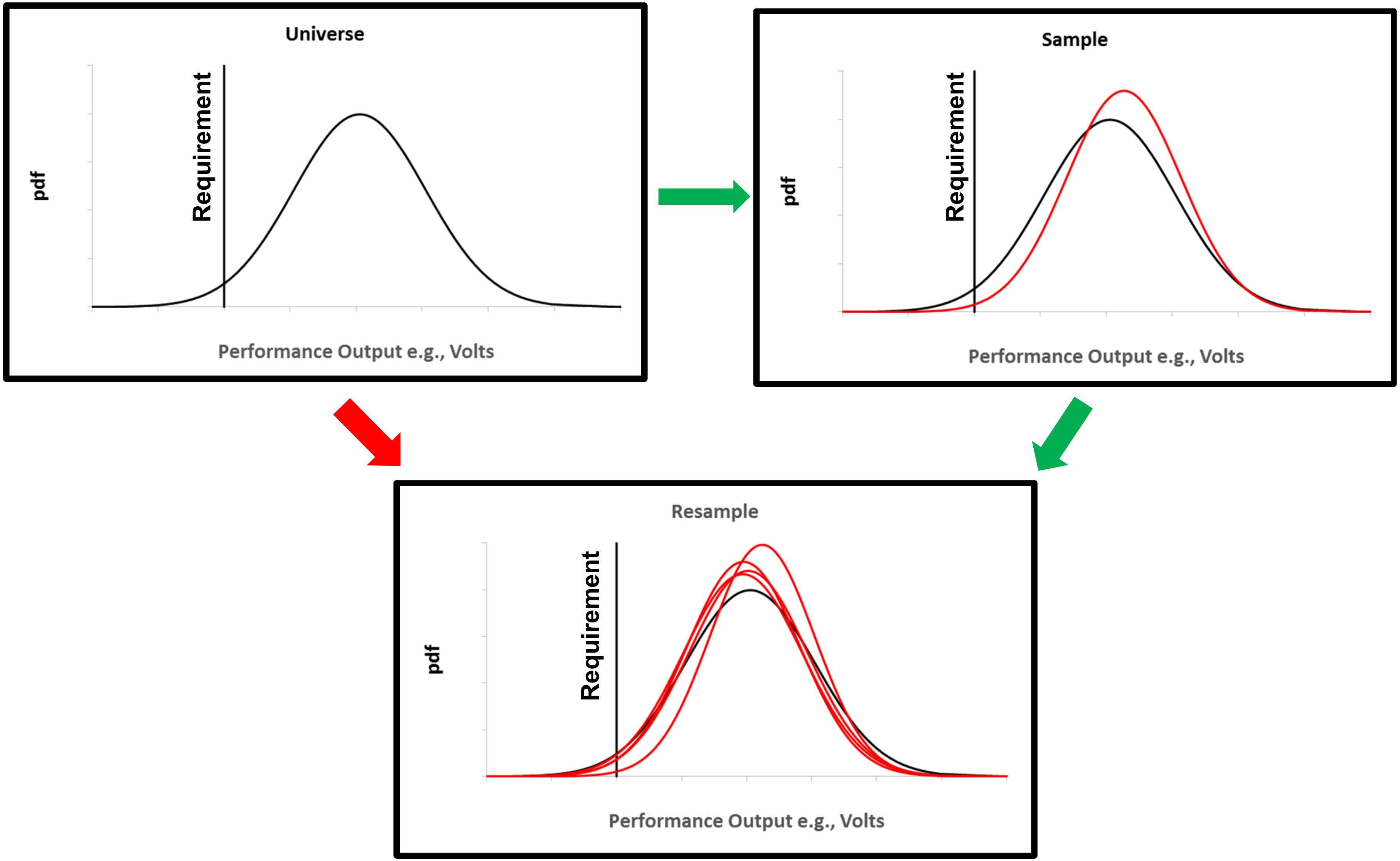


We have performance data taken on *SubSys A*

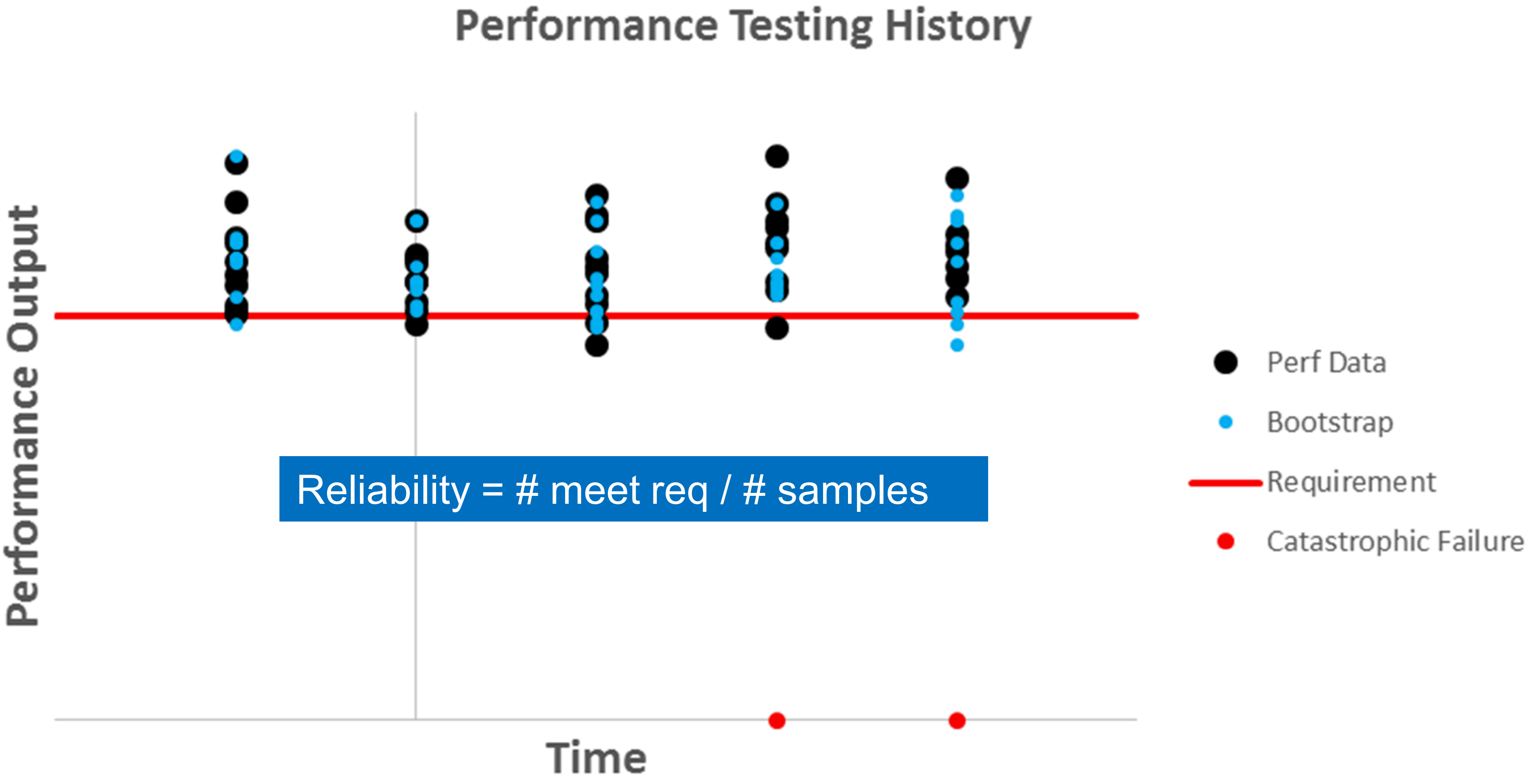


We could update environment survivability assessment, but will skip

How do you estimate the uncertainty in any statistic from any distribution? – Bootstrap!

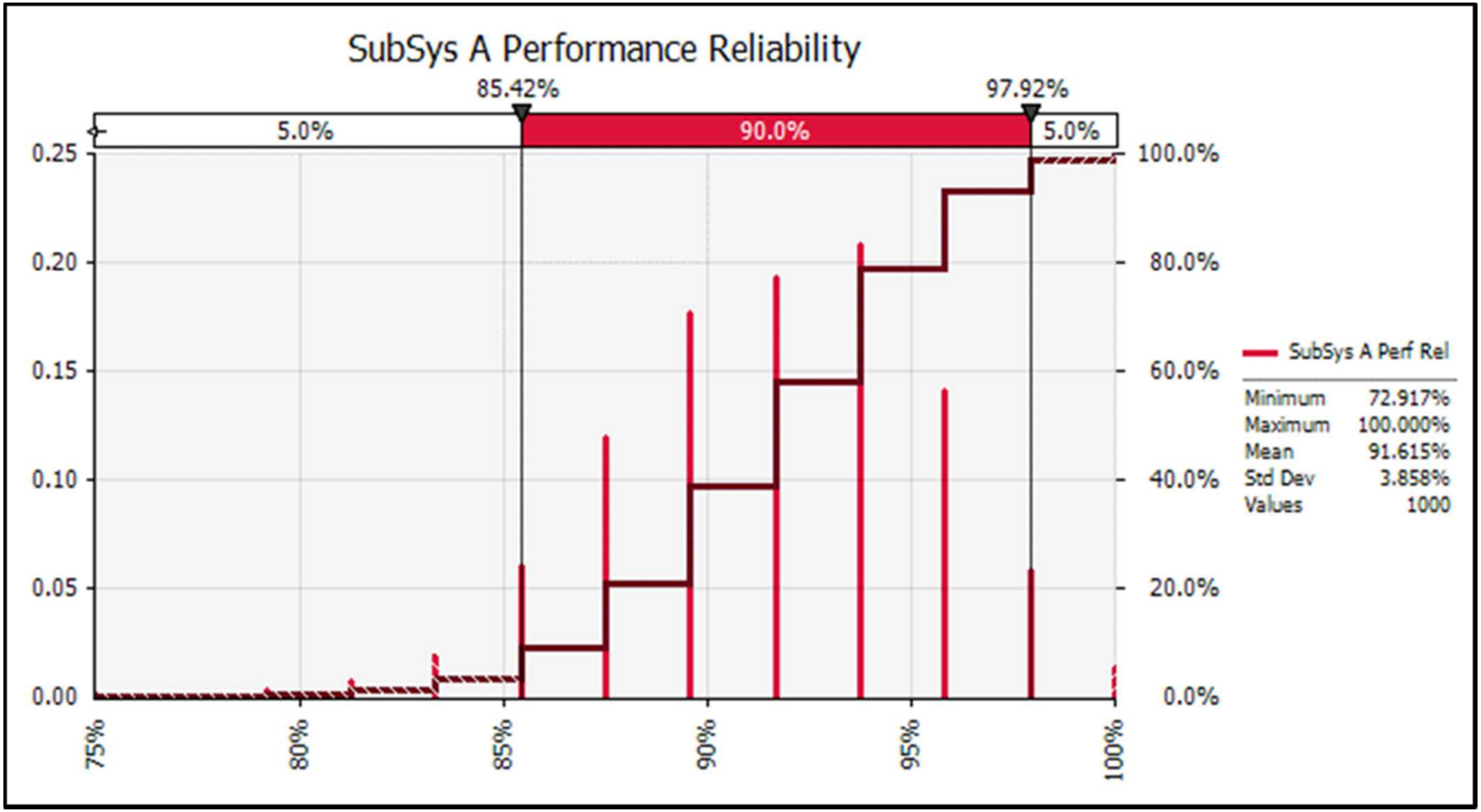

Universe
pdf
Requirement
Performance Output e.g., Volts
Sample
pdf
Requirement
Performance Output e.g., Volts
Resample
pdf
Requirement
Performance Output e.g., Volts

Bootstrap samples used to estimate SubSys A reliability



Reliability = # meet req / # samples

Estimated performance reliability for *SubSys A* is uncertain

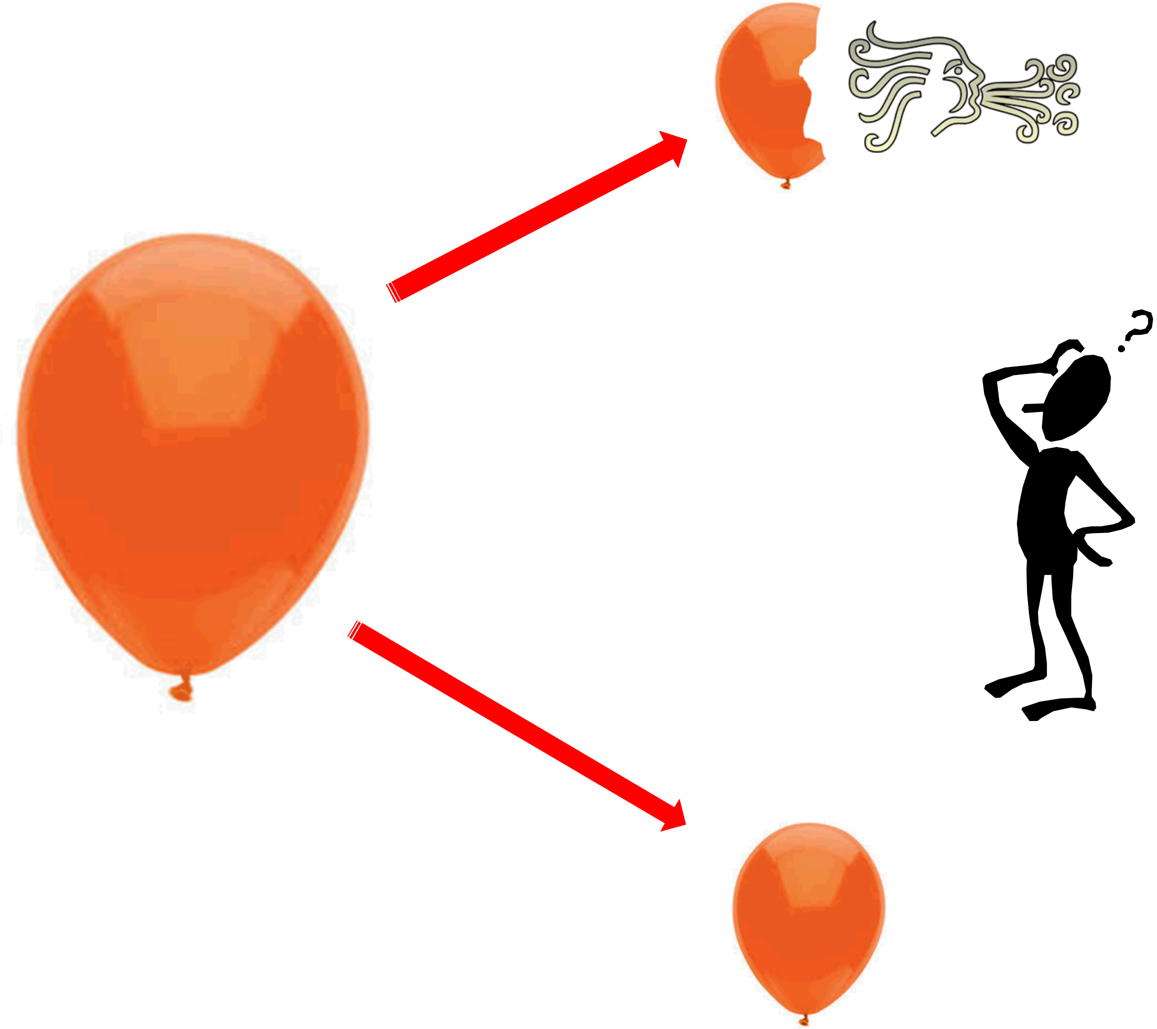


SubSys A Perf Rel	
Minimum	72.917%
Maximum	100.000%
Mean	91.615%
Std Dev	3.858%
Values	1000

Spend some time on the SubSys A performance tab in the spreadsheet

- Demonstrate dynamic bootstrapping
- Demonstrate dynamic reliability based on bootstrap samples
- We have skipped updating of environment survivability

SubSys B either performs all of the time or none of the time; but we are uncertain about which

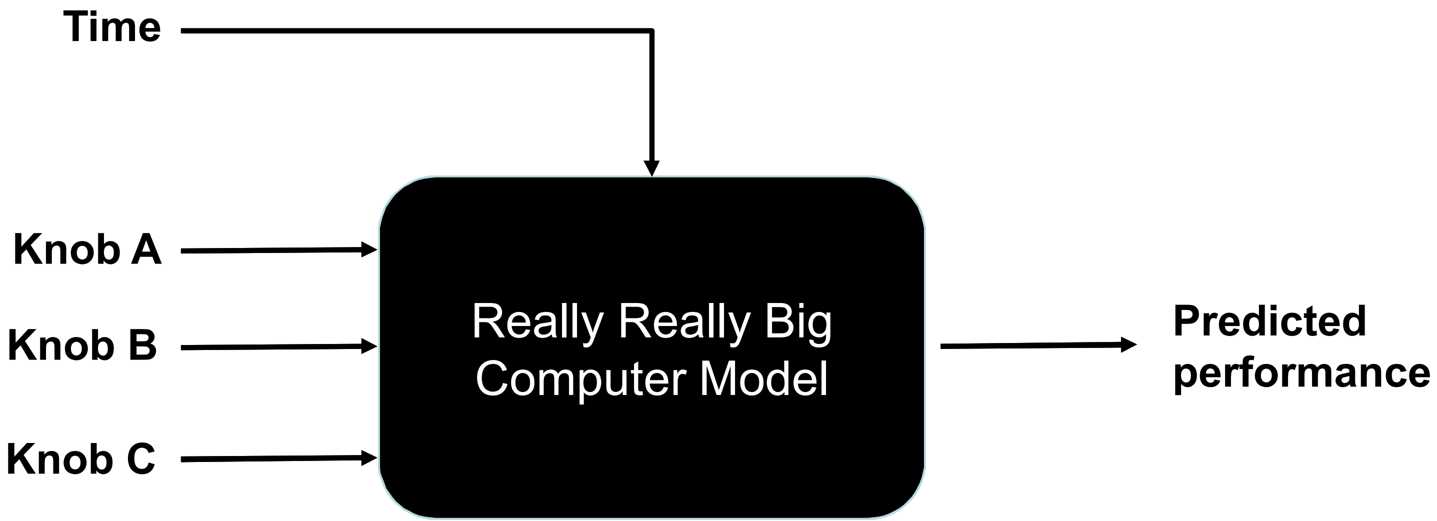


Either it Goes Boom!
Reliability = 1.0

Epistemic Uncertainty

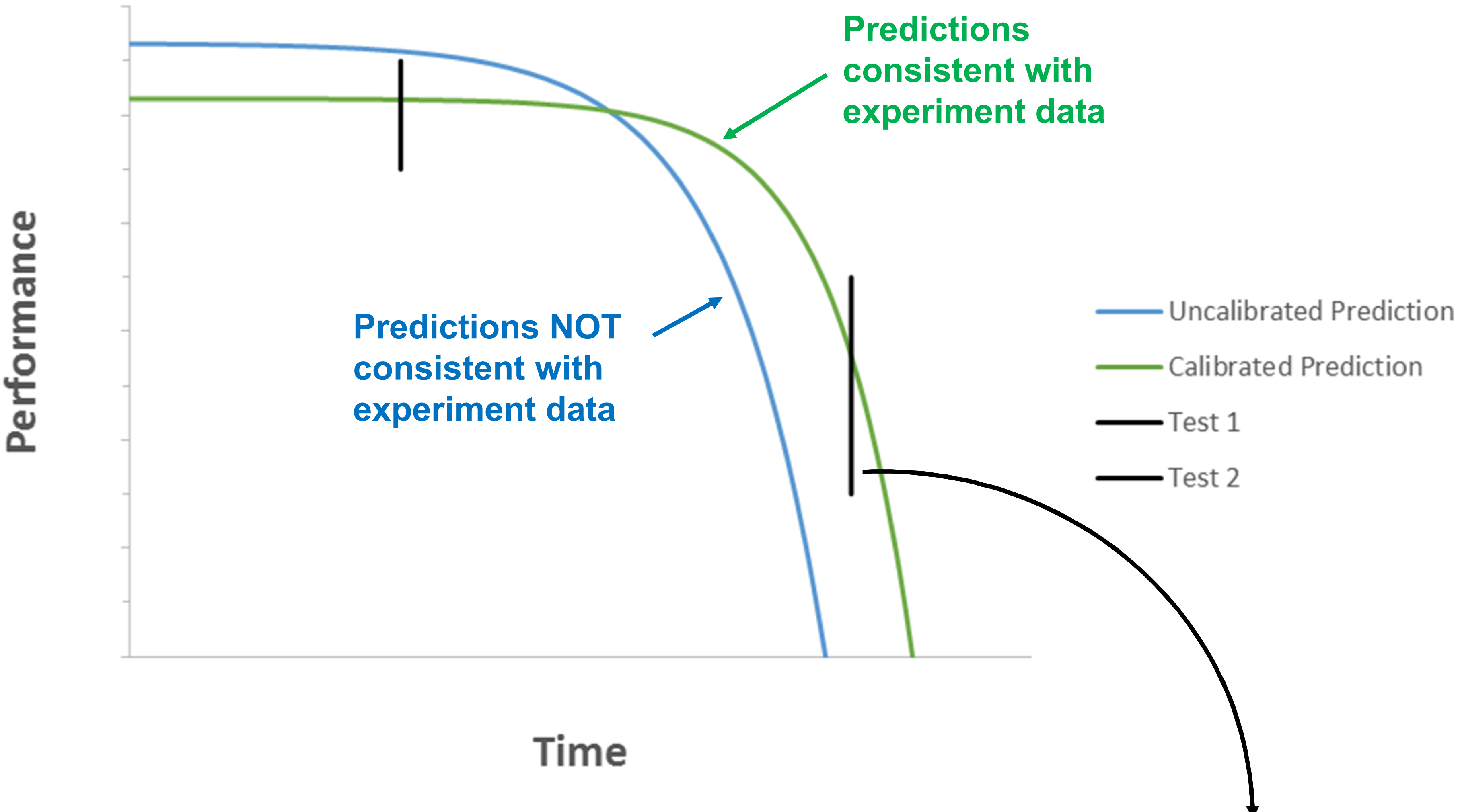
Or it's A Dud!
Reliability = 0.0

Really really big computer models are used to predict SubSys B performance



Knobs are surrogates for missing or unknown physics; consequently, they are treated as epistemic uncertainties

Knobs are calibrated so that predictions are consistent with limited experiment data



Predictions consistent with experiment data

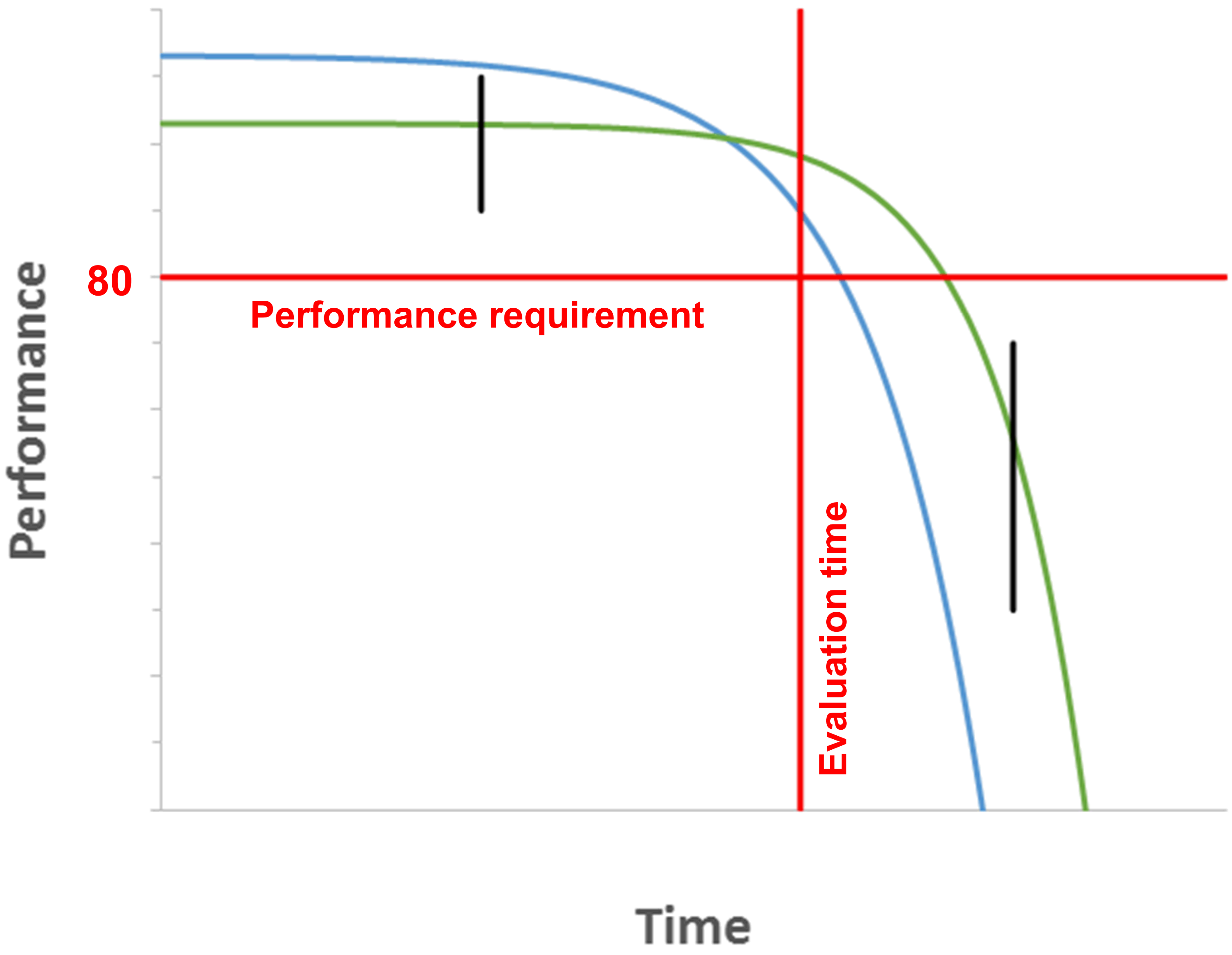
Predictions NOT consistent with experiment data

Knobs are treated as epistemically uncertain parameters

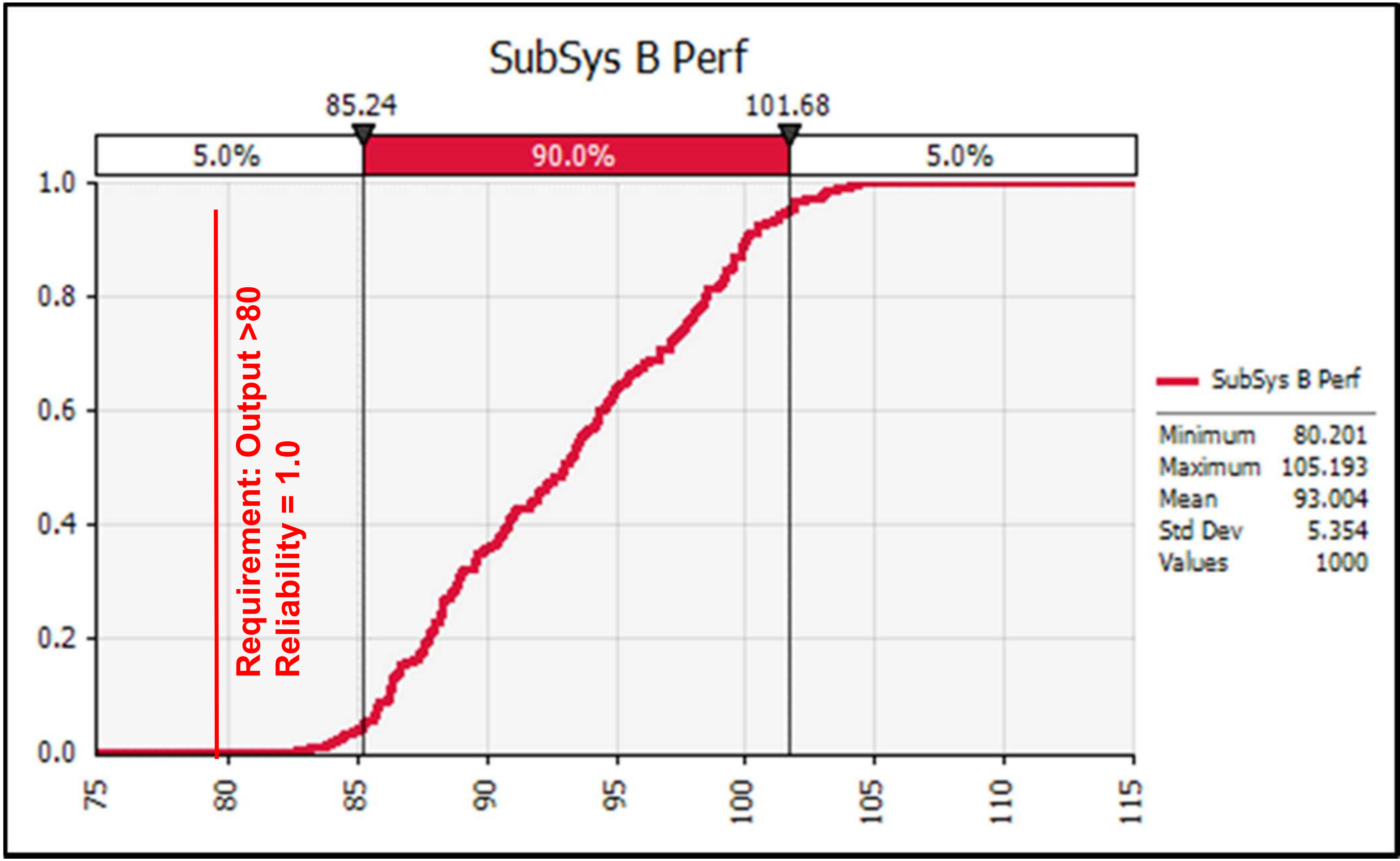
Computer model parameters (knobs) are calibrated using Bayes Theorem

- Posterior (Knobs:Data) ~ Likelihood(Data: Knobs) * Prior(Knobs)
- Un-informed priors for the Knobs
  - Prior(Knobs) ~ 1   any value is equally likely for wide range of possibilities
- Likelihood(Data: Knobs)
  - = 1 if prediction passes though error bar of both data points
  - = 0 otherwise

SubSys B has performance requirements



SubSys B has uncertain performance

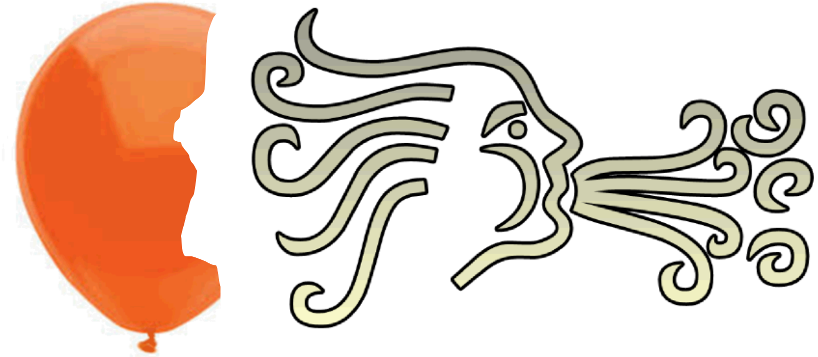


Requirement: Output >80
Reliability = 1.0

Spend some time on the SubSys B performance tab in the spreadsheet

- Demonstrate un-calibrated and calibrated performance curves

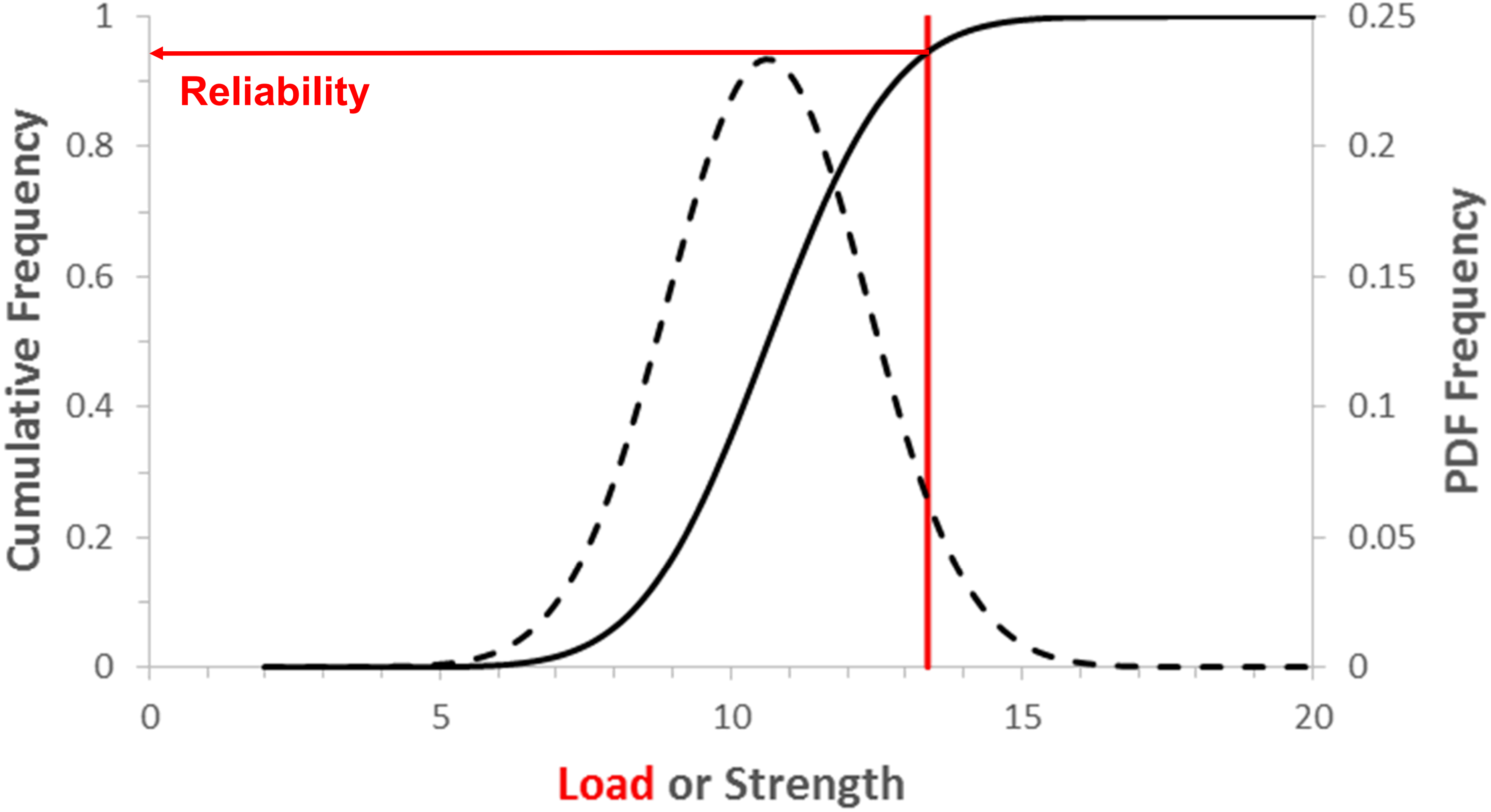
It’s all about target defeat



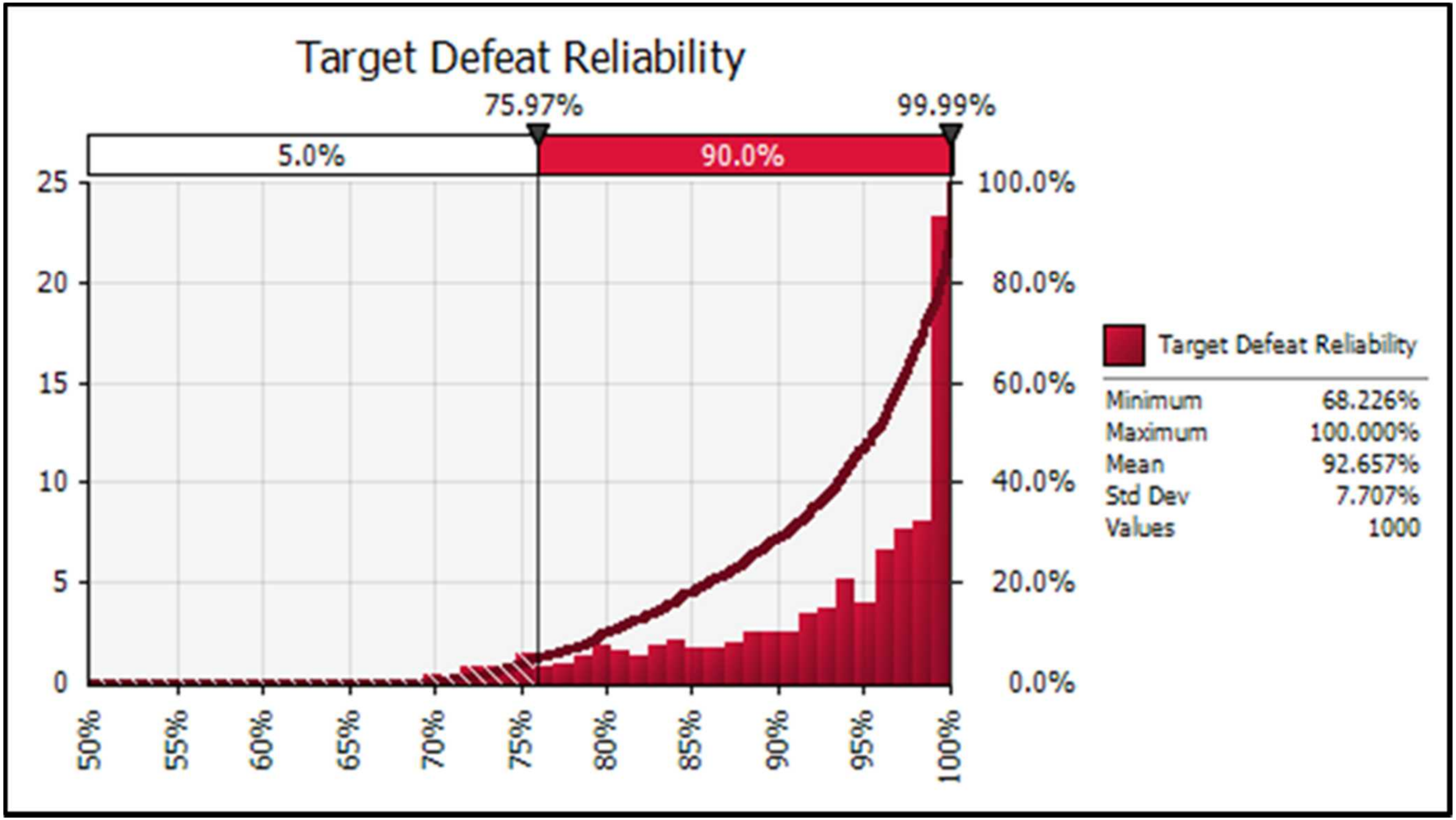
Load has attributes of epistemic uncertainty only

Strength has attributes of aleatory uncertainty and epistemic uncertainty

Different Target Defeat reliabilities for different load strength curves



Target Defeat reliability is uncertain

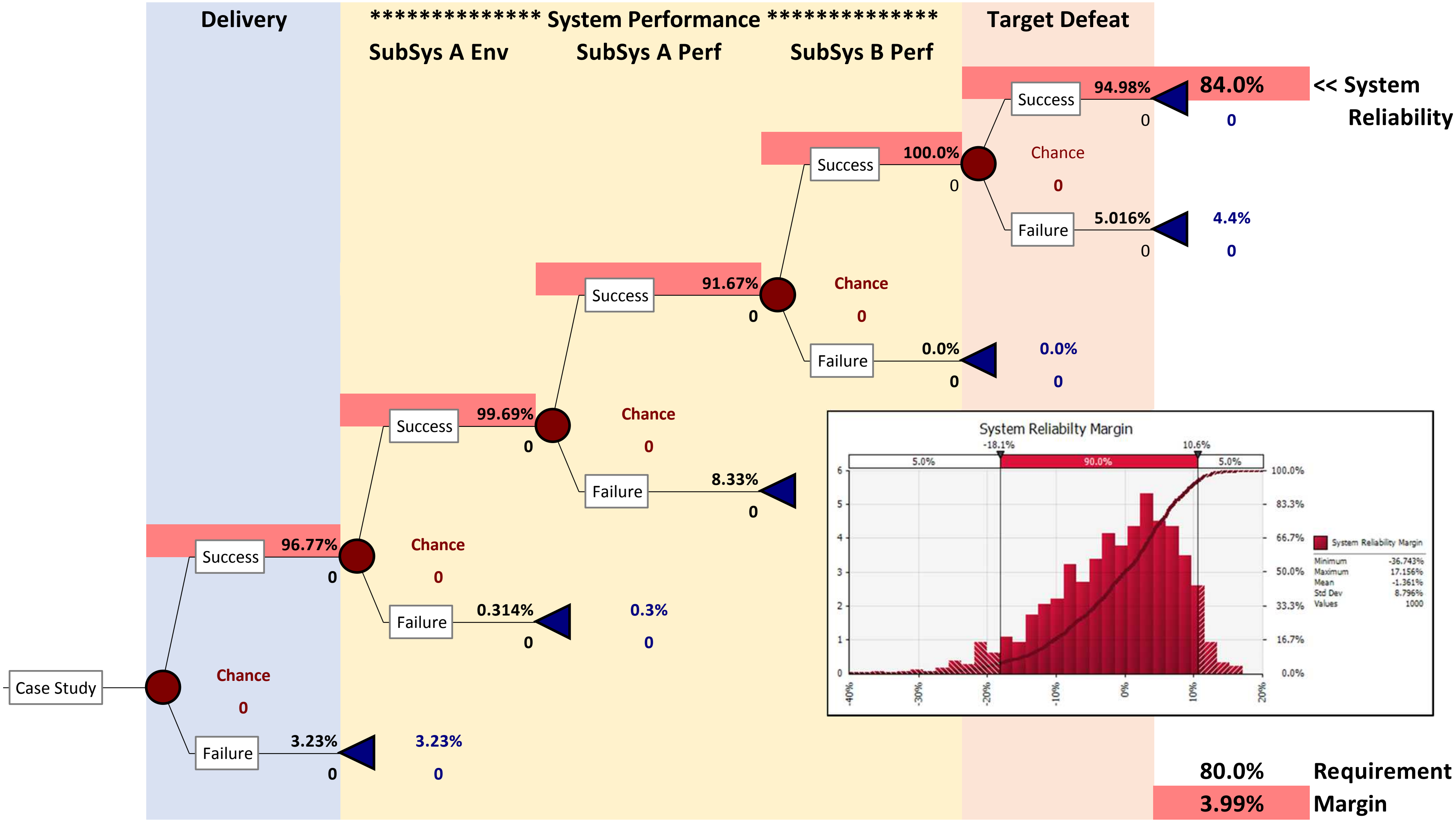


Minimum	68.226%
Maximum	100.000%
Mean	92.657%
Std Dev	7.707%
Values	1000

Spend some time on the Target Defeat tab in the spreadsheet

- Demonstrate dynamic load strength curves and reliability

We’ve used probabilistic methods to propagate both aleatory and epistemic uncertainty



Delivery

************** System Performance **************

Target Defeat

SubSys A Env

SubSys A Perf

SubSys B Perf

Success 94.98% 0 — 84.0% 0 << System Reliability

Success 100.0% 0 — Chance 0

Failure 5.016% 0 — 4.4% 0

Success 91.67% 0 — Chance 0

Failure 0.0% 0 — 0.0% 0

Success 99.69% 0 — Chance 0

Failure 8.33% 0

Success 96.77% 0 — Chance 0

Failure 0.314% 0 — 0.3% 0

Case Study — Chance 0

Failure 3.23% 0 — 3.23% 0

80.0% Requirement

3.99% Margin

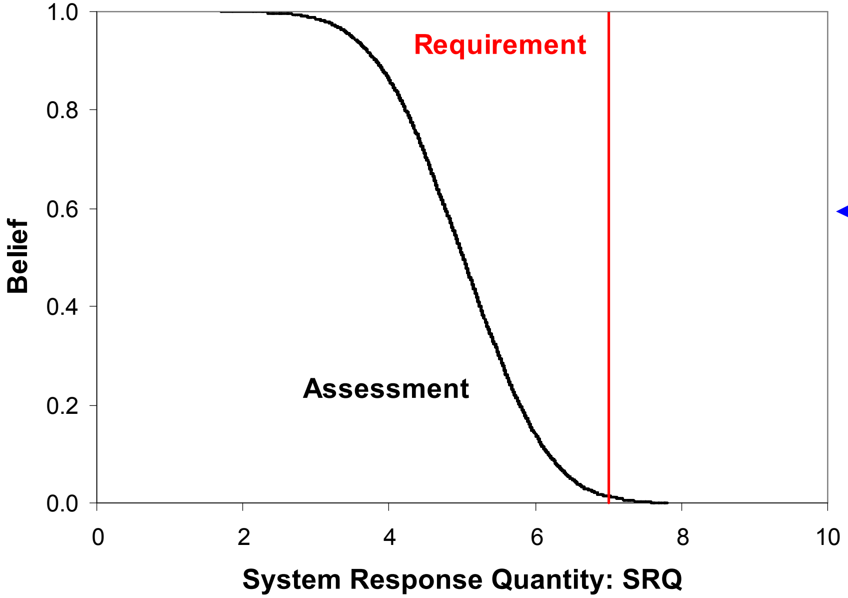
Your life, or that of a loved one, hangs in the balance of a risk assessment

$$SRQ = \sum_{i=1}^{10} X_i \qquad X_i = [0,1] \qquad \mathrm{Re}\, quirement : SRQ \leq 7$$

Dominated by epistemic uncertainties

You have to pick a framework for representing and propagating epistemic uncertainties

$$SRQ = \sum_{i=1}^{10} X_i \qquad X_i = [0,1] \qquad \mathrm{Re}\,quirement : SRQ \le 7$$



?

$$\textbf{Belief}(\textbf{SRQ} > 7) = \textbf{0.014}$$

$$\textbf{SRQ} \equiv [\textbf{0,10}]$$

Probabilistic framework

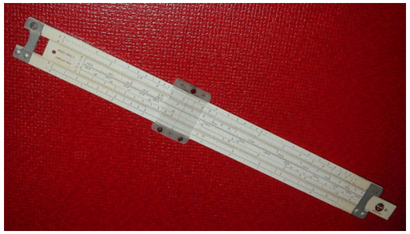
- Bayesian methods
- SOA for risk assessment community

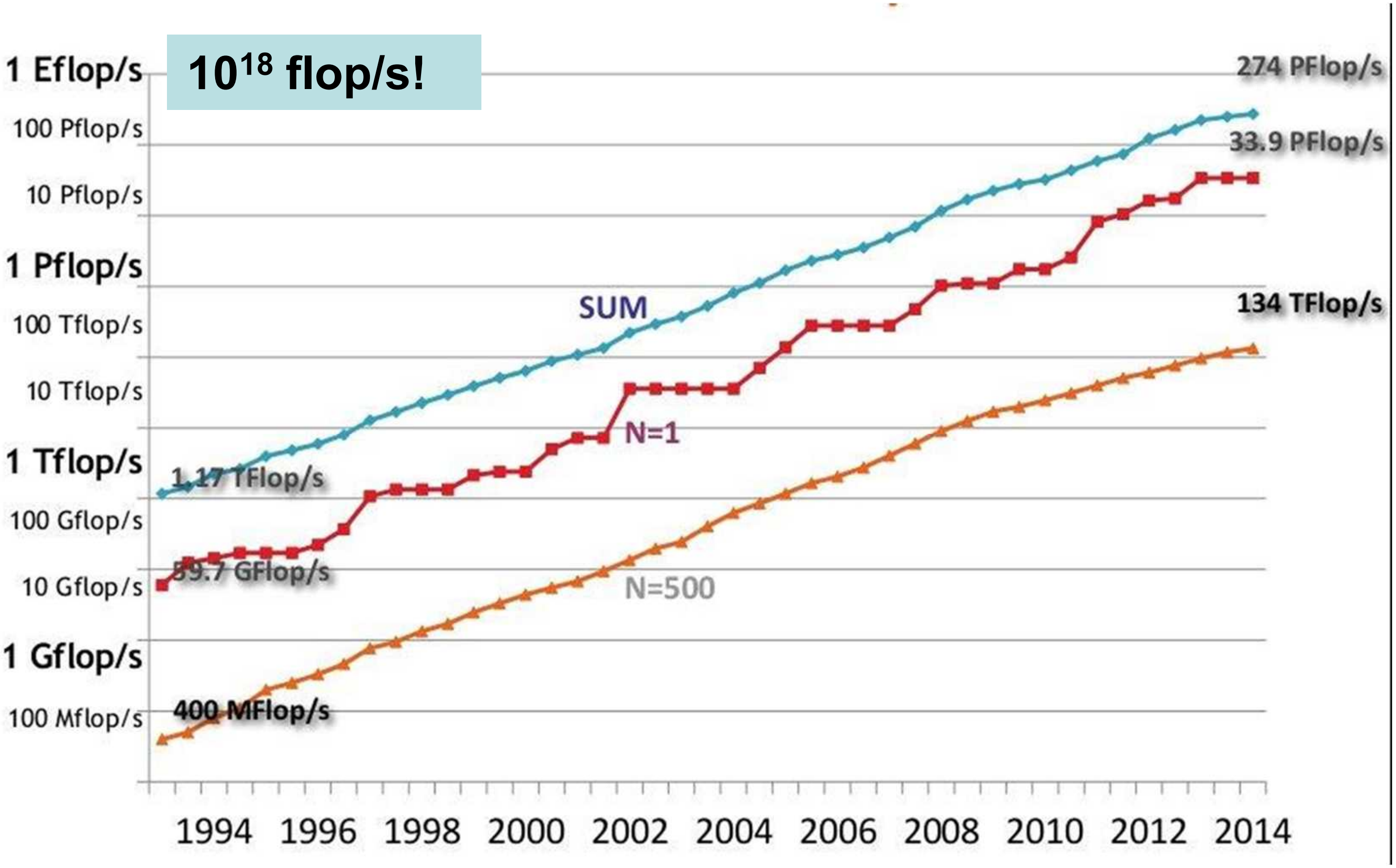
Non-probabilistic framework

- Interval analysis
- Dempster/Shafer (evidence) theory
- P-Boxs

The world of high performance computing has changed a lot in my lifetime

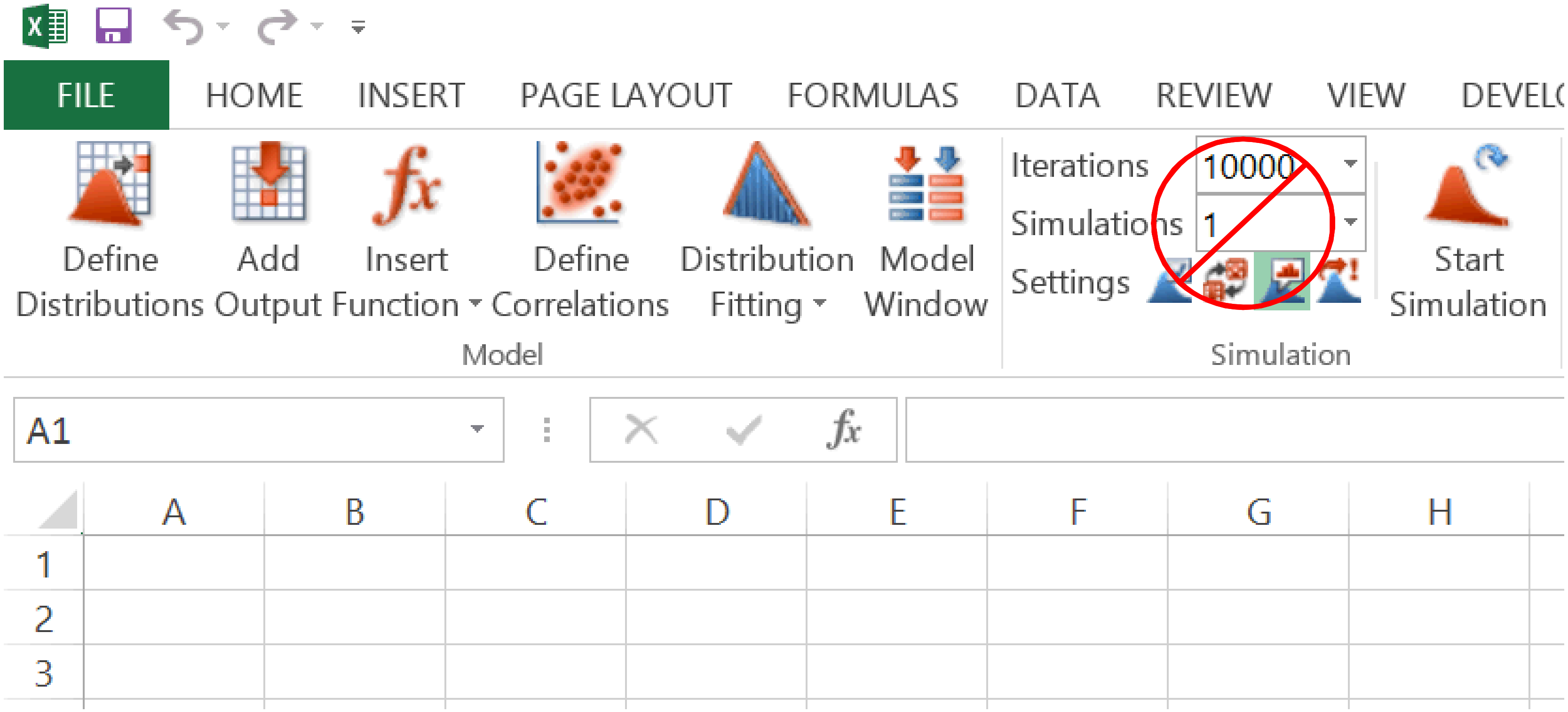
HPC (~ 1 flop/s)
1620 - 1950





Is todays engineering 18 orders of magnitude better?

Even with all this CPU, this is still not an option for our most challenging applications



Iterations 10000

Simulations 1

Use computer model to train surrogate models i.e.,NeuralTools

Credibility of comp. predictions is assessed in terms of six attributes and associated best practices

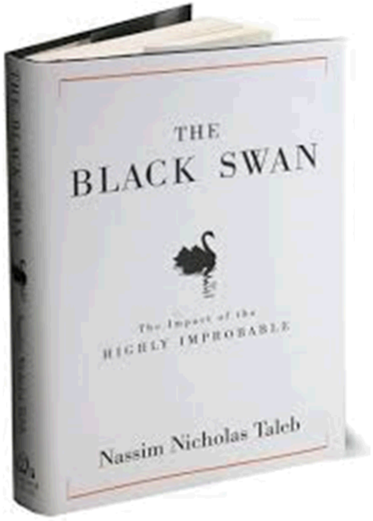
What *evidence* do we demand to assert credible computer predictions?

1. Representation or geometric fidelity
2. Physics and material model fidelity
3. Code verification
4. Solution verification
5. Validation
6. Uncertainty quantification

Best Practices:

1. Calibrate SMEs
2. Look broadly for uncertainties
3. Separate aleatory and epistemic uncertainties
4. Perform sensitivity analyses
5. Reduce uncertainties only if decisions could change
6. Avoid strong assumptions
7. Ensure that uncertainty propagation errors do not pollute results
8. Document evidence/rational for uncertain inputs
9. Perform technical peer review

A common criticism of risk assessment relates to the prevalence of *Black Swans**



Metaphor for events characterized by surprise and high consequence that seem obvious only in retrospect

*** Unknown/Unknowns**

Surprise???

High Consequence???

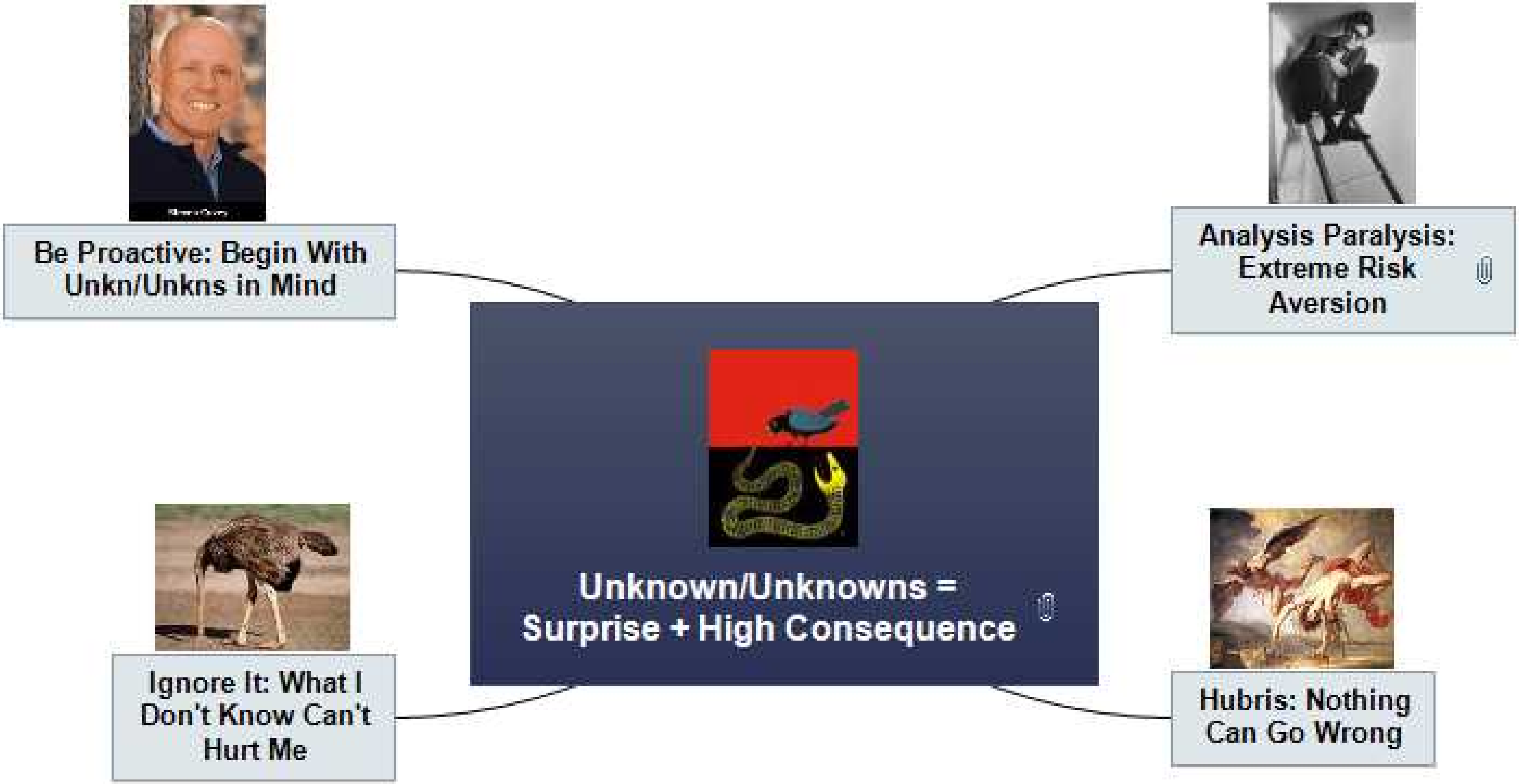
I'm Not Feeling It!

Now I feel it!

Surprise!

High Consequence!

How do *you* deal with “The Bird”?



Unknown/Unknowns =
Surprise + High Consequence

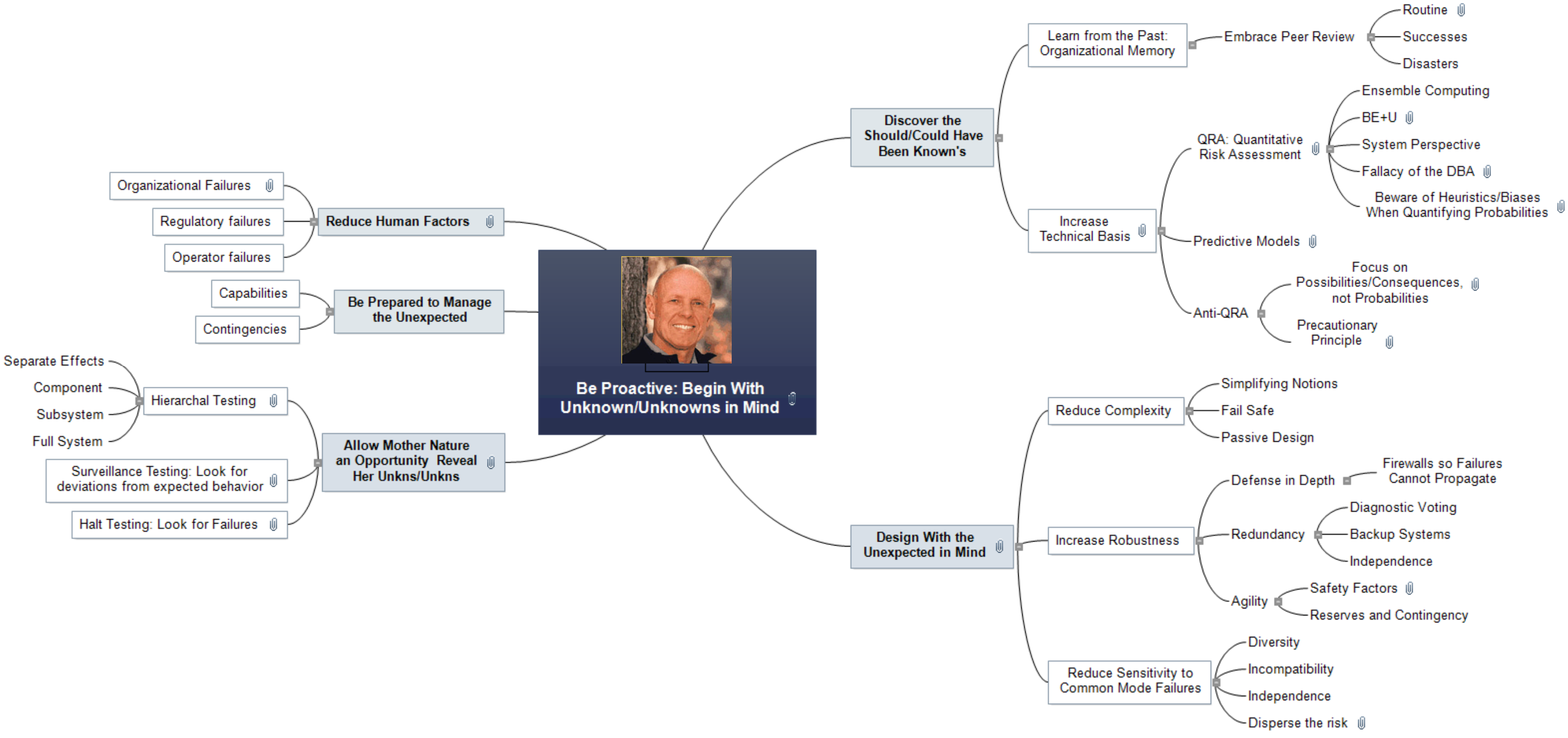
Be Proactive: Begin With
Unkn/Unkns in Mind

Analysis Paralysis:
Extreme Risk
Aversion

Ignore It: What I
Don't Know Can't
Hurt Me

Hubris: Nothing
Can Go Wrong

Be proactive, begin with “The Bird” in mind



- Be Proactive: Begin With Unknown/Unknowns in Mind
  - Reduce Human Factors
    - Organizational Failures
    - Regulatory failures
    - Operator failures
  - Be Prepared to Manage the Unexpected
    - Capabilities
    - Contingencies
  - Allow Mother Nature an Opportunity Reveal Her Unkns/Unkns
    - Hierarchal Testing
      - Separate Effects
      - Component
      - Subsystem
      - Full System
    - Surveillance Testing: Look for deviations from expected behavior
    - Halt Testing: Look for Failures
  - Discover the Should/Could Have Been Known's
    - Learn from the Past: Organizational Memory
      - Embrace Peer Review
        - Routine
        - Successes
        - Disasters
    - Increase Technical Basis
      - QRA: Quantitative Risk Assessment
        - Ensemble Computing
        - BE+U
        - System Perspective
        - Fallacy of the DBA
        - Beware of Heuristics/Biases When Quantifying Probabilities
      - Predictive Models
      - Anti-QRA
        - Focus on Possibilities/Consequences, not Probabilities
        - Precautionary Principle
  - Design With the Unexpected in Mind
    - Reduce Complexity
      - Simplifying Notions
      - Fail Safe
      - Passive Design
    - Increase Robustness
      - Defense in Depth
        - Firewalls so Failures Cannot Propagate
      - Redundancy
        - Diagnostic Voting
        - Backup Systems
        - Independence
      - Agility
        - Safety Factors
        - Reserves and Contingency
    - Reduce Sensitivity to Common Mode Failures
      - Diversity
      - Incompatibility
      - Independence
      - Disperse the risk

I’d like to hear your thoughts on this

You laughed when you first heard it because you didn't understand

Now you do!

Known/Knowns

- Fixed value
- Aleatory uncertainty if there is variability
- Not reducible

Known/Unknowns

- Epistemic uncertainty
- Reducible

Unknown/Unknowns

- Black swans
- Discovery