Uncertainty, Risk, and Information Value in Software Requirements and Architecture

Emmanuel Letier, David Stefan, Earl T. Barr
University College London, UK

Best of RESG Research 2014, 4 December 2014

Embrace Uncertainty!

Software Design Decisions

What software to build? What functions? What quality level?

What architectural style? What components and interfaces? How to deploy them?

Uncertainty is inevitable

We must decide without knowing everything

The Surfer’s Approach to Uncertainty

Mary Poppendieck, “Learning to Surf”, industry keynote @ ICSE2013

Instead of learning to surf, conventional organizations try to control the waves. This almost never works.

— Allen Ward

The Surfer’s Approach to Uncertainty

The Scientific Approach to Uncertainty

- Decision Analysis, a discipline for understanding, formalising, analysing, and communicating insights about situations in which important decisions must be made
- Founded on Bayesian Statistics

Ron Howard, Stanford

The *Pseudo*-Scientific Approach

Use formulae that resembles a scientific approach, except that

- the decision criteria are numbers without verifiable meaning
- the decision models are not falsifiable
- no retrospective evaluation of decisions and outcomes

Most widely used example, the Analytical Hierarchy Process (AHP)

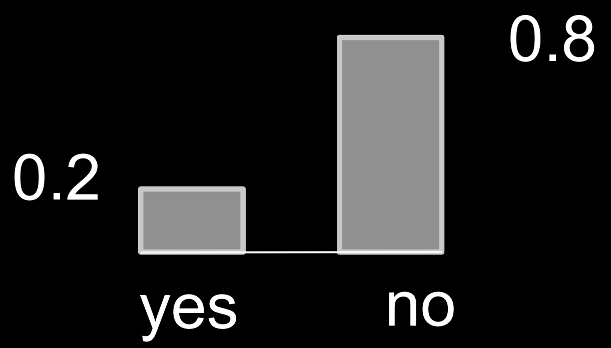
What do we mean by uncertainty ?

Uncertainty

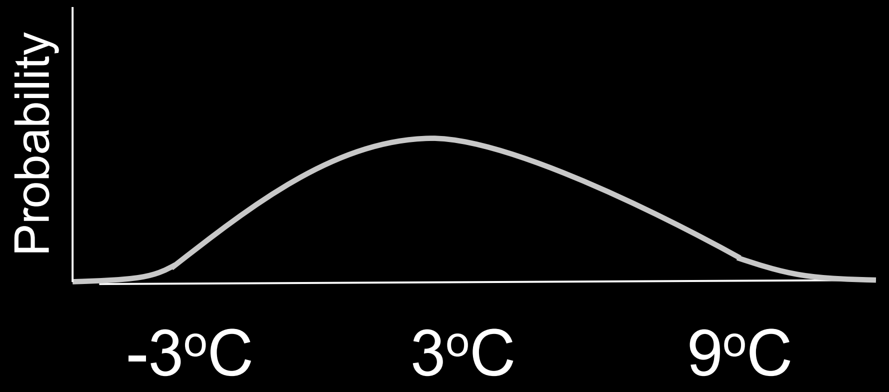
Uncertainty is the lack of complete knowledge about a state or quantity. There is more than one possible value and the “true” value is not known.

Measurement of uncertainty. A set of possible values with a probability assigned to each.

Will it snow at Christmas?



How cold will it be?



We always know *something* …

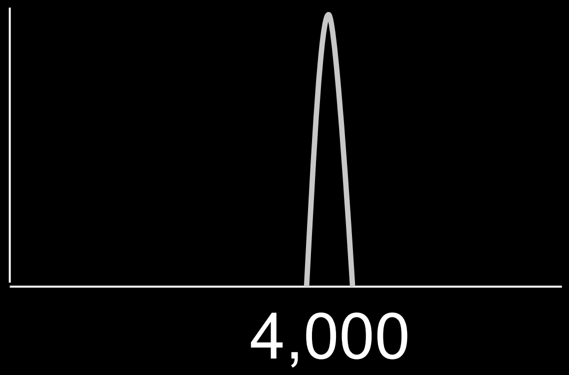
How many professional Business Analysts and Requirements Engineers in the UK?

Accuracy and Precision

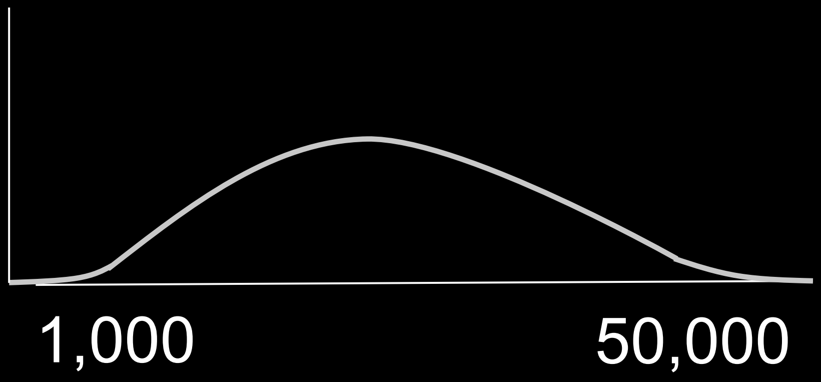
For a measurement or prediction

- Precision refers to how close the measured or predicted values are to each other
- Accuracy refers to how close the measured or predicted values are to the true value

How many BA and requirements engineers in UK?



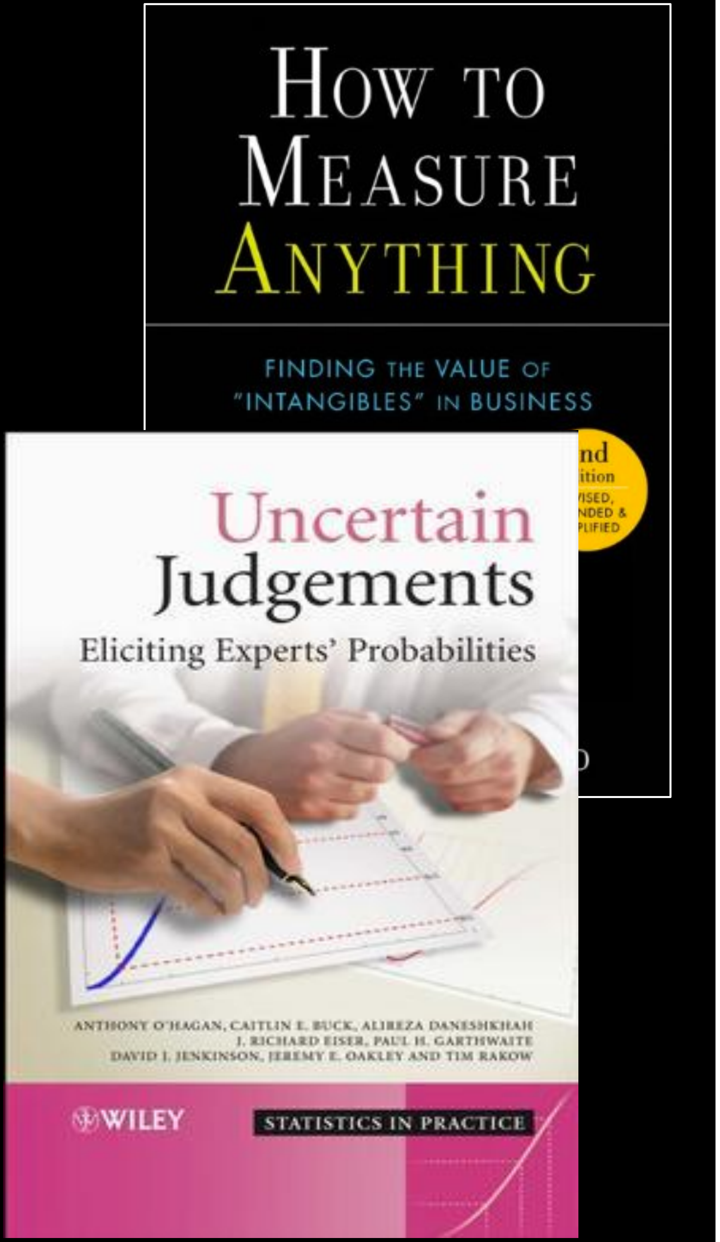
Precise: yes; Accurate: ?



Less precise, but more accurate

Key Observations

- We always know *something* even if our uncertainty is large
- The more precise, the higher risk of being wrong (inaccurate)
- People can be trained to become reliable estimators


How to Measure Anything
Finding the Value of "Intangibles" in Business
Uncertain Judgements
Eliciting Experts' Probabilities
Anthony O'Hagan, Caitlin E. Buck, Alireza Daneshkhah
J. Richard Eiser, Paul H. Garthwaite
David J. Jenkinson, Jeremy E. Oakley and Tim Rakow
Wiley
Statistics in Practice

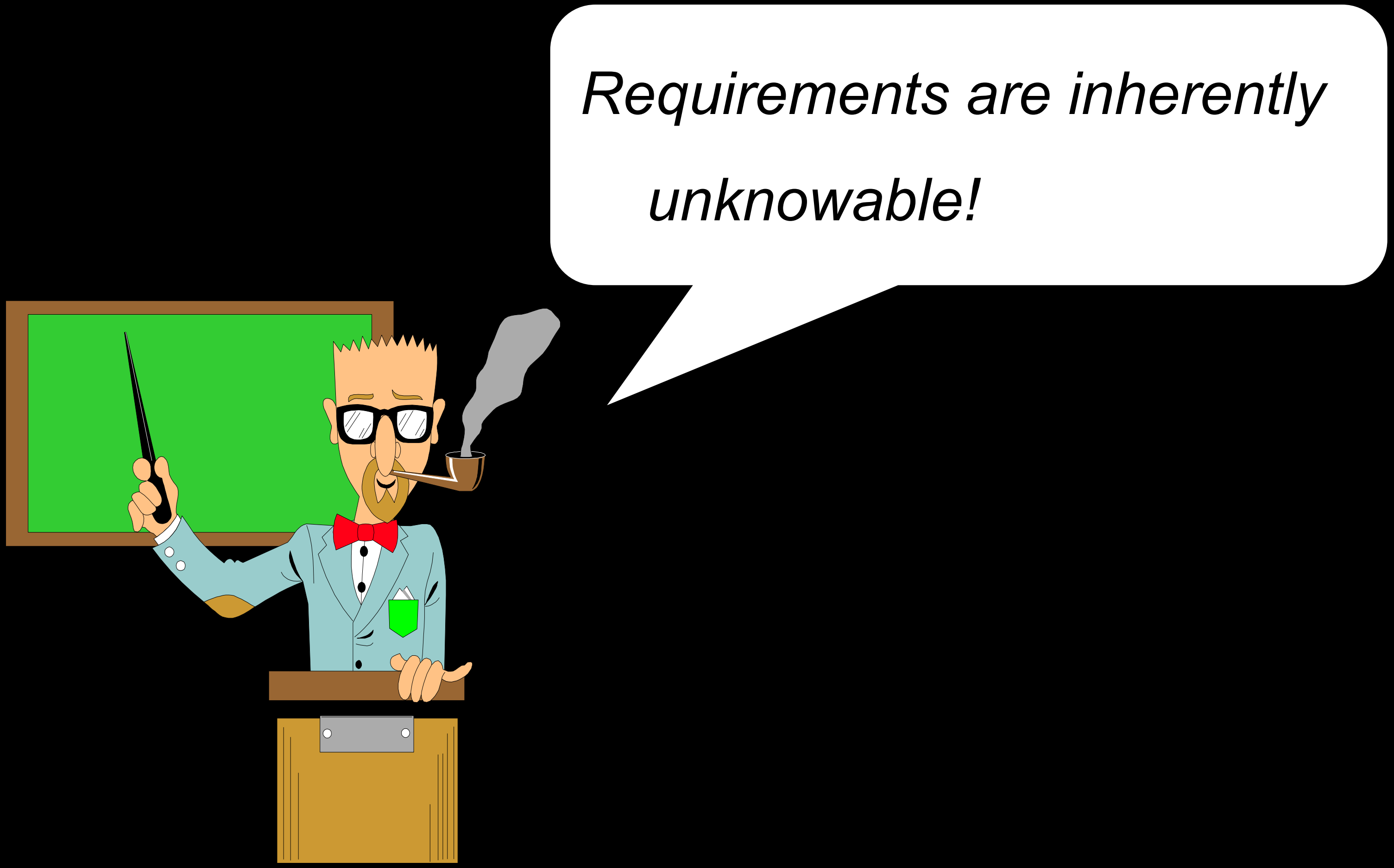
Sh*t Software Engineers Say ...

Clients don’t know what they want

Requirements change is inevitable

It’s not possible to discover the true requirements before building the system

Sh*t Academics Say ...



Linda Northrop “Does Scale Really Matter? – Ultra-Large-Scale Systems Seven Years after the Study” plenary keynote @ ICSE2013

What they mean ...

Requirements are uncertain

Even if our uncertainty is large, we always know *something* about the requirements

Yet, we insist on requirements being precise

"Requirements engineering is the branch of software engineering concerned with the real-world goals for, functions of, and constraints on software systems. It is also concerned with the relationship of these factors to precise specifications of software behavior, and to their evolution over time and across software families."

Pamela Zave, ACM Computing Surveys, 1997

My working hypothesis

We need to rethink requirements engineering (and most of software engineering) from the ground up

- Stop focusing on precision for its own sake
- Maintain our focus on business goals and business value
- Introduce new focus on decisions under uncertainty

What is our ICSE'14 paper about?

A two minute summary

Software Design Decisions

What software to build? What functions? What quality level?

What architectural style? What components and interfaces? How to deploy them?

Uncertainty is inevitable

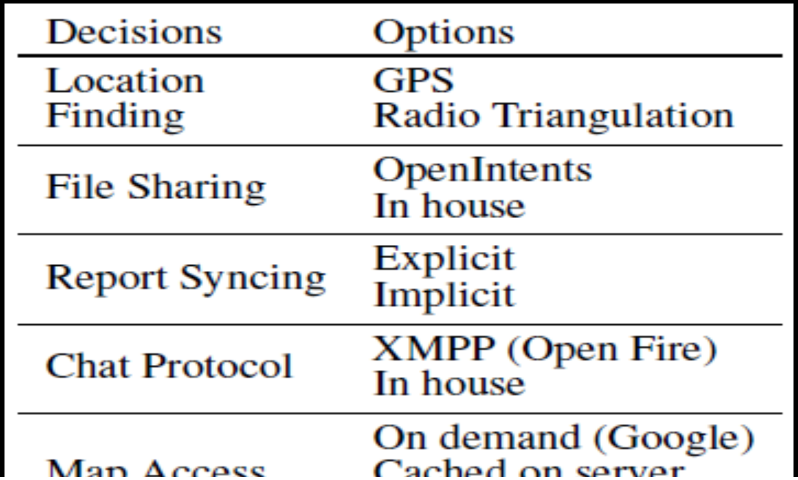
We must decide without knowing everything

Proposed Solution

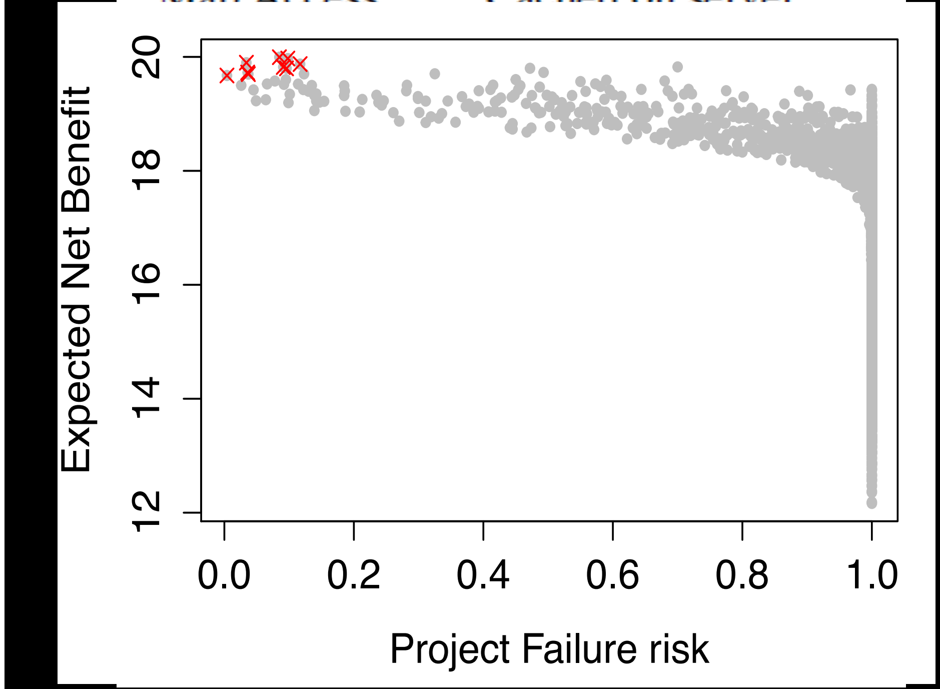
A systematic method for software design decisions under uncertainty

- Builds on SEI architecture decision models (ATAM and CBM) and KAOS quantitative goals models
- Use of Expected Value of Perfect Information (EVPI) to guide uncertainty reduction (through elicitation, analysis, prototyping, etc.)
- Under-the-hood: Monte-Carlo simulation, Pareto-based optimisation, Efficient EVPI computation (Code available in R package)

ICSE paper example: a mobile system for coordinating emergency rescue teams (Esfahani et al. ICSE'13)



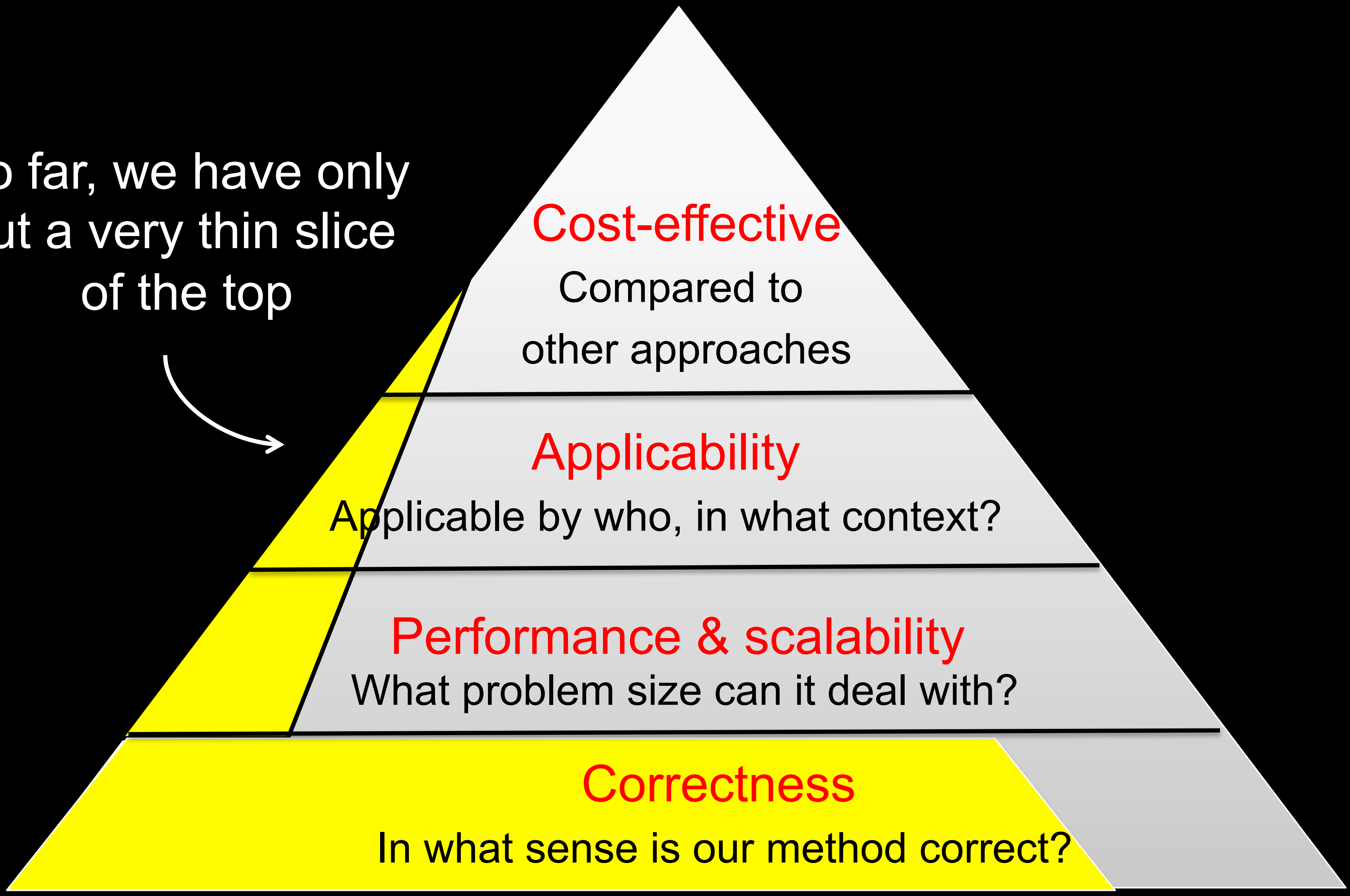
Decisions	Options
Location Finding	GPS Radio Triangulation
File Sharing	OpenIntents In house
Report Syncing	Explicit Implicit
Chat Protocol	XMPP (Open Fire) In house
Map Access	On demand (Google) Cached on server



- Design space: 10 design decisions; around 7,000 candidate architectures
- Objectives: Cost, Response Time, Reliability, Battery Life, ...
- Models given by design team: Utility score defined as weighted sum of objectives satisfaction (unfortunately not falsifiable)
- Method output: A shortlist of 10 architectures with highest expected net benefit and lowest risk

Evaluation Pyramid

So far, we have only cut a very thin slice of the top



Cost-effective

Compared to other approaches

Applicability

Applicable by who, in what context?

Performance & scalability

What problem size can it deal with?

Correctness

In what sense is our method correct?

What have we learned since writing the ICSE paper?

Good News

- First applications on two real case studies: Sustainability decisions for UCL Estates (David Stefan Thesis)

Bad News: progress will be much slower than expected

- Case study resources are not given, we must create them
- Modelling software design decisions is hard
  - High standard: models must be falsifiable
  - Weak foundations: e.g. what is an architectural decision?
  - Scalability: high number of inter-related decisions
- Transition to “Bayesian Thinking” takes time

A Call to Action

Uncertainty will be at the heart of many important decisions for the 21st Century

We have to rethink requirements in terms of decisions under uncertainty

How do you want people to make critical IT decisions in 10 years?

As Surfers

As Pseudo-Scientists

As Scientists