# The Value of Requirements Uncertainty

Emmanuel Letier
http://letier.cs.ucl.ac.uk

Joint work with David Stefan and Earl Barr

Louvain-la-Neuve, 4 October 2013

1

# Embrace Uncertainty!

2

## Software Design Decisions

What software to build? What quality level? What to build in next iteration?

What components and interfaces? How to deploy them? When to change the architecture?

Uncertainty is inevitable

We must decide without knowing everything

3

## The Surfer's Approach to Uncertainty

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

— Allen Ward

Mary Poppendieck "Learning to Surf" industry keynote @ ICSE2013

4

## The Surfer's Approach to Uncertainty

5

## 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

Ron Howard, Stanford

6

## The *Pseudo*-Scientific Approach

Resembles the 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)

7

## What do we mean by uncertainty ?

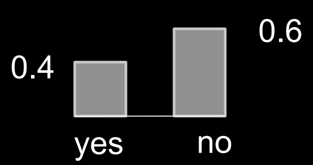
8

## Uncertainty

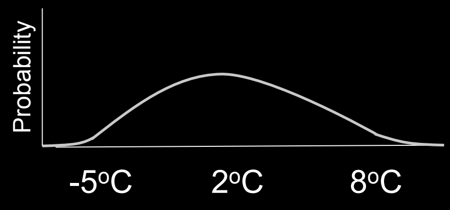
**Uncertainty** is the lack of complete knowledge about a state or quantity. There is more than one possible value and the "true" value is not known.

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

Will it snow at Christmas?




How cold will it be?



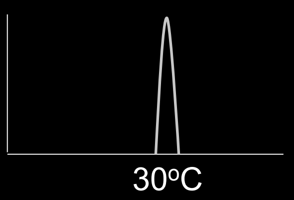

9

## Accuracy and Precision

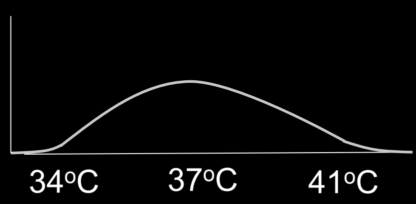
For a measurement or prediction

- 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 hot will it be in Hyderabad, India on 1st June 2014?




Precise: yes; Accurate: ?




Less precise, but more accurate

10

## Key Insights

The more precise, the higher risk of being wrong (inaccurate)

The less you know, the harder it is to be both precise and accurate; if you want to be accurate, you have to be less precise

Reducing uncertainty has economic value because it leads to better decisions that will, on average, increase profit

11

## Things Software Engineers Say ...

*Clients don't know what they want*

*Requirements documents are always too vague, incomplete, inconsistent, out-of-date, etc.*

*Requirements change is inevitable*

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

12

## Things Academics Say ...

*Requirements are inherently unknowable!*

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

13

## What they really mean...

Requirements are uncertain

14

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."

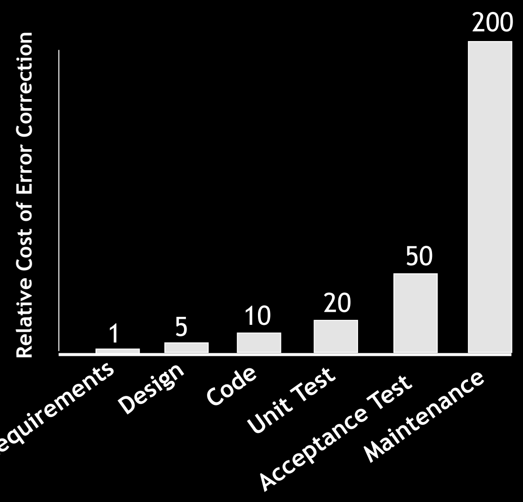
Pamela Zave, ACM Computing Surveys, 1997

15

Why do we want precision?

16

## Boehm's Cost-to-Fix Curve (1981)





17

## An Hypothesis

The cost-to-fix curve crystallised software engineering thinking around questions of costs (time and money) and defects

⇩

Requirements engineering focuses on precision as a way to detect and fix defects as early as possible when it is cheaper to do so

⇩

We have lost sight of the end goal!

18

What is the end goal of Software Engineering?

19

The end goal of Software Engineering is ...

A. To deliver software on time
B. To deliver software on budget
C. To deliver software with low number of bugs
D. All of the above
E. None of the above

20

## The end goal of Software Engineering is ...

A. To deliver software on time

B. To deliver software on budget

C. To deliver software with low number of bugs

D. All of the above

E. None of the above

F. To deliver software that provides value for money

(or no software at all if there are better ways to provide value)

21

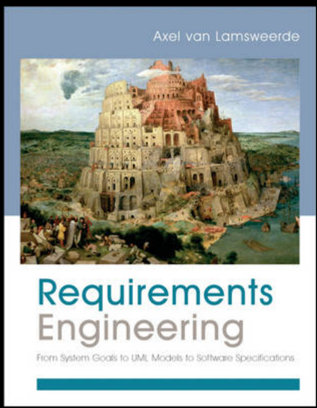
## Beware of Treating Subgoals as End Goals

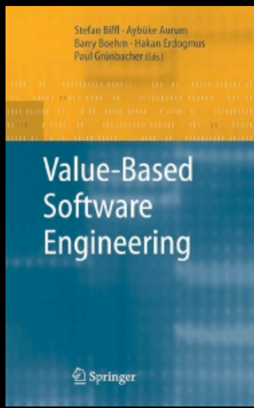
Delivering on time, on budget, with low defect rate doesn't necessarily provide value for money (e.g. £80 million mobile technology for UK police)

Minimising requirements defects (ambiguity, incompleteness, etc.) doesn't necessarily yield a valuable system

22

## Approaches Focussing on Value for Money



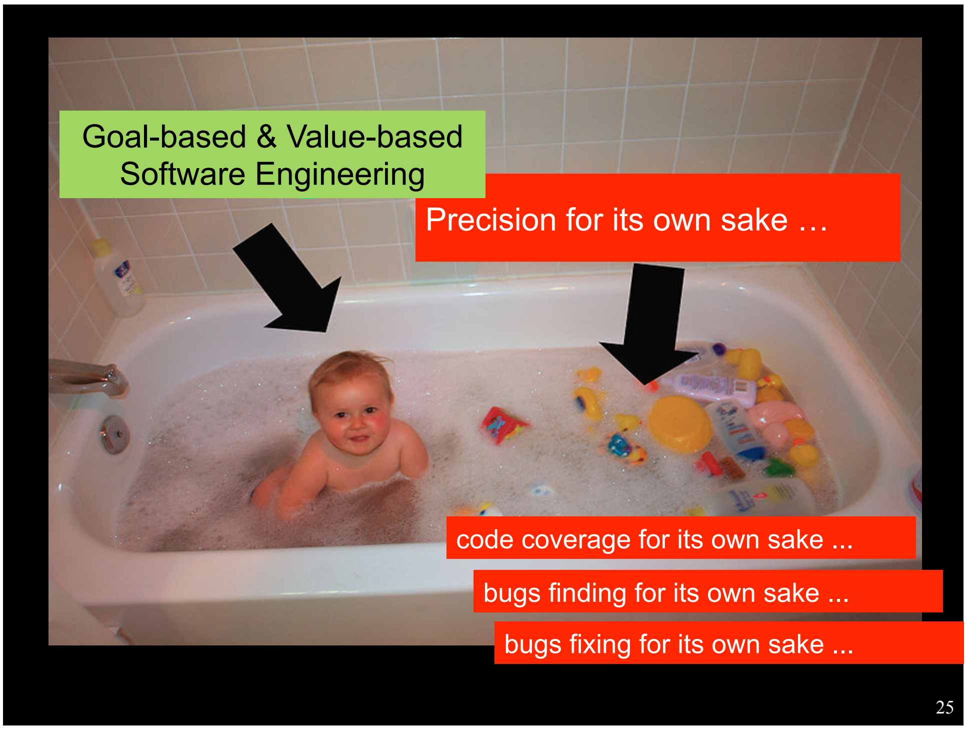





Goal Modelling

23


24




A new perspective on software engineering

Goal-Based Decisions Under Uncertainty

26

## Goal-Based Decisions Under Uncertainty

*The Simplest Possible Example*

27

## A Typical IT Project Business Case

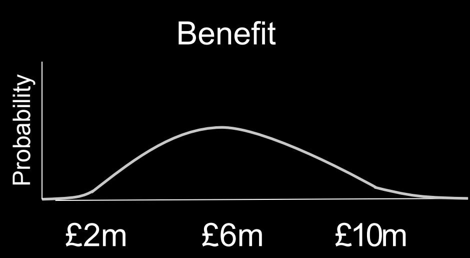
| | |
|---|---|
| Expected Cost | €2m |
| Expected Benefit | €10m |

| | |
|---|---|
| Expected Net Benefit | €8m |
| ROI | 400% |

28

## Cost-Benefit Analysis with Uncertainty

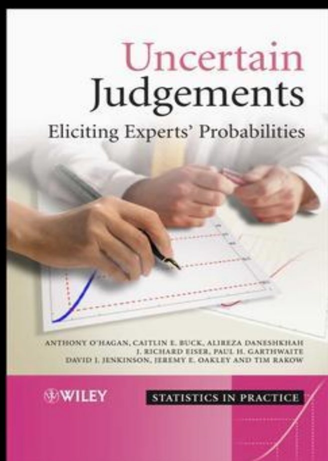
| | 90% Confidence Interval | Most Likely |
|---|---|---|
| Cost | [€2m , €5m] | €3.5m |
| Benefit | [€2m , €10m] | €6m |


29

## Where Do the Numbers Come From?

- Cost and benefit are functions of a set of uncertain variables (eg. development cost, operating cost, market size, ...)
- Uncertainty about each variable is elicited from experts and decision makers
  - using simple effective methods
  - having sound mathematical foundations and significant empirical validation




30

## Cost-Benefit Analysis with Uncertainty

| | 90% Confidence Interval | Most Likely |
|---|---|---|
| Cost | [€2m , €5m] | €3.5m |
| Benefit | [€2m , £10m] | €6m |

⇩

| | |
|---|---|
| Expected Net Benefit | €2.5m |
| Loss Probability | 16% |
| Average Loss Magnitude | €1.3m |

31

## The Expected Value of Perfect Information (EVPI)

(Ronald Howard, 1966)

EVPI(X) = the expected gain in net benefit from obtaining perfect information about X to inform decision

$$EVPI(X) = E[\max_{a \in A} E[NB(a)|X = x, BK]] - \max_{a \in A} E[NB(a)|BK]]$$

*Expected* gain (expectation over X)

Highest expected net benefit among all alternatives given current knowledge BK *and X = x*

Highest expected net benefit among all alternatives given current knowledge BK

32

## The Expected Value of Information

Reminder: Expected Net Benefit = €2.5m; Loss Probability = 16%

| | EVPI | Remaining Loss Probability |
|---|---|---|
| Total Perfect Information | €0.22m | 0% |
| Info about Benefit | €0.18m | 3% |
| Info about Cost | €0.001m | 16% |

- Information about benefit has high value and impact on risk
  - Current 90% confidence interval: €2m-€10m
- Information about cost has no value and impact on risk
  - Current 90% confidence interval: €2m-€5m

33

## The Measurement Inversion Paradox

(Douglas Hubbard, 1999)

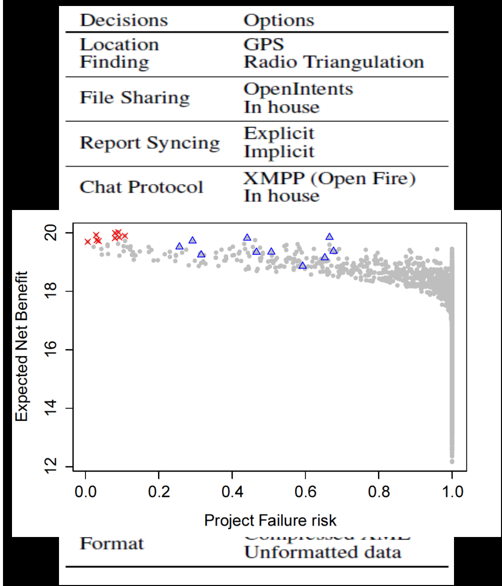
Lessons from applying decision analysis to 20 IT business cases, each having 40 to 80 variables

1. Most variables have zero information value
2. Variables with high information values were routinely those the client never measured
3. Clients spent most of their effort measuring quantities with low or even zero information value

34

## Application to Software Design Decisions

(with D. Stefan and E.T. Barr)

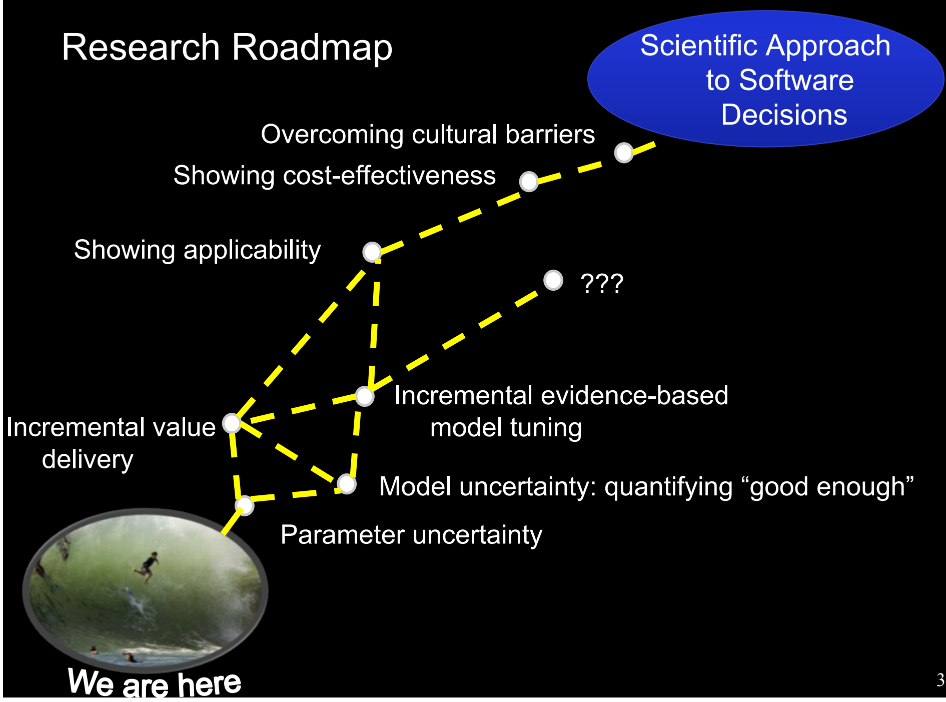



A mobile system for coordinating emergency rescue teams

- 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 objective satisfaction
- Lessons Learnt
  - Risks specific to requirements and architecture decisions
  - Need to reason about model uncertainty in addition to parameter uncertainty
  - Decision models must be falsifiable

35

## Research Roadmap




36

## A Call to Action

Uncertainty is at the heart of most major challenges for the 21st Century

Who do you want to inform our IT projects decisions?

The Surfers

The Pseudo-Scientists

The Scientists

37