

2003-

“Five variations on the theme:  
Software failure: avoiding the avoidable and living with the  
rest“

Variation 5: “Living with the rest”

Prof. Les Hatton

Computing Laboratory, University of Kent, Canterbury

Version 1.1: 18/Nov/2003

©Copyright, L.Hatton, 2003-

# *Overview*

- ❖ **Legal stuff**
- ❖ **Risk**
- ❖ **Some predictions**



# *How many ways can I sue thee ?*

## ❖ **Criminal Law**

- Best not think about this. It might spoil your day.

## ❖ **Civil Law**

- Law of Contract
- Law of Tort (negligence)
- Product Liability (Consumer Protection Act)



# *How many ways can I sue thee ?*

## ❖ **Law of Contract**

- Effectively the only avenue currently used by the courts
- Is software goods or a service ?
  - ◆ Different legal regimes (Sale of Goods Act) or (Goods and Services)
- Contracts should allow for failure and mitigate accordingly



# *How many ways can I sue thee ?*

## ❖ **Law of Tort (negligence)**

- Onus is on plaintiff to prove negligence
- Standard tests (proximity, special relationship)



# *How many ways can I sue thee ?*

## ❖ **Product liability (Consumer Protection Act)**

- Strict liability (unlike Tort)
- Relationship of software with the Act is not very clear



# *Overview*

- ❖ **Legal stuff**
- ❖ **Risk**
- ❖ **Some predictions**



# *Definitions*

- **Risk** is when you don't know what will happen but you do know the probabilities
- **Uncertainty** is when you don't even know the probabilities



# *The eternal conflict*

**The study of risk is an eternal struggle between:-**

- Those who wish to quantify it
- Those who feel it cannot be quantified



# *A mathematician's view of risk*

If R is the Risk, F the Frequency and C the Consequence:

$$\mathbf{R = F \times C}$$

So unlikely catastrophic events have a similar risk to very frequent but unimportant events.

Mathematician's always seek to quantify risk.



# *A risk practitioner's view of risk*

**It is fundamentally impossible to quantify risk  
because of:-**

- Problems of measurement
- Failure to take account of risk compensation,  
(people compensate for greater safety by taking  
more risks.)



# *Problems of measurement - A genius's view of risk*

“If a guy tells me that the probability of failure is 1 in  $10^5$ , I know he's full of crap.”

Richard P. Feynmann, Nobel Laureate commenting on the NASA Challenger disaster.



# *Risk compensation*

## **Problem:-**

- 500 motorcyclists a year are killed in accidents in the U.K.

## **Solution**

- Ban motorcycles

Discuss ...



# *The risk thermostat, (J. Adams)*

## **This view of risk argues:-**

- Everybody has a propensity to take risk
- This propensity varies between people
- Risk-taking is influenced by the rewards
- Perceptions of risk are influenced by experience of losses
  - one's own and others
- Risk-taking involved a balancing between the propensity to take risk and the perceived risk



# *The dance of the 'risk thermostats'*

## **Interaction in society involves:-**

- Continuous dance of every individual's risk thermostat and interaction with other risk thermostats
- Underlying chaos which further undermines quantification

If as it seems quantification seems impossible, are there any useful patterns ?

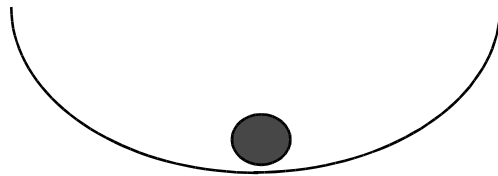


# *Patterns in uncertainty*

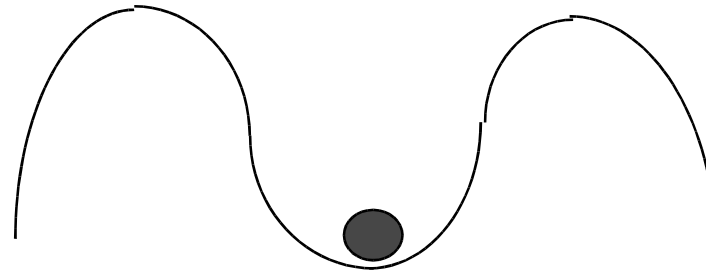
## The 4 managerial views of nature, (Holling)



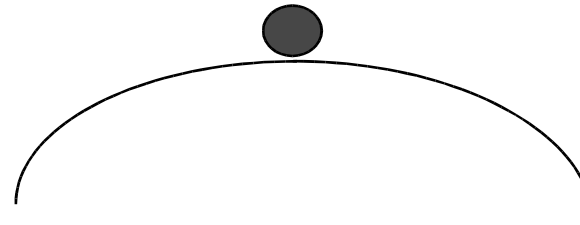
Nature capricious,  
*fatalist*



Nature benign,  
*laissez-faire*



Nature perverse / tolerant,  
*interventionist*



Nature ephemeral,  
*precautionary*



# *Different rationalities*

- Rational argument is based upon logic, mathematics and grammar
- In an uncertain world, rational arguments are constructed on premises beyond rationality
- People apply different views of nature in a rational argument



# *Different rationalities*

## **The four basic rationalities are:-**

- Individualist
  - ◆ relatively free from control by others and seek to control their environment. Example - hacker.
- Hierarchist
  - ◆ inhabit a world of strong group boundaries and hierarchical structures. Example - quality manager
- Egalitarian
  - ◆ Strong group loyalties but little respect for externally imposed rules. Example - users
- Fatalist
  - ◆ Resigned to their fate and make no effort to change it. Example - trombone player.



# *Different rationalities*

**If asked how we manage risk, the reactions of the four basic rationalities are:-**

- Individualist
  - ◆ asserts we are already over-regulated and we should leave it to market forces
- Hierarchist
  - ◆ says we need more research but things are basically OK
- Egalitarian
  - ◆ urge precaution and press for urgent action
- Fatalist
  - ◆ watch television and buy lottery tickets



# *Conclusions about risk*

- It is almost impossible to quantify risk or at least we have totally failed to achieve it so far
- Realising that each person approaches a risk with some dynamic mixture of the four basic rationalities is important to understanding the inherently associative nature of risk.



# *Conclusions about risk, (J. Adams)*

- Everyone else is seeking to manage risk too
- Everybody is guessing. If they knew, its not risk
- Guesses are extremely influenced by beliefs
- The behaviour of others and the behaviour of nature are your risk environment
- Unless people's propensity to take risk is reduced:-
  - ◆ Safety intervention simply leads to responses which re-establish the level of risk
  - ◆ Safety intervention redistributes risk but does not reduce it
- Science will continue to invent new risks
- In the dance of the risk thermostats, the music never stops



# *Relevance*

So what does all this have to do with risk and benefit in modern software engineering ?



# *What can we do ?*

## **As we saw earlier in the discussion of risk,**

- We must reduce the propensity of software managers and engineers to take risks
  - ◆ By making managers more aware of the cost of failure
  - ◆ By making engineers and managers more aware of the ability of testing technology to reduce the cost of failure
- Will it help to produce more reliable software ?
  - ◆ Probably not. Every observation of society suggests that risk compensation usually balances risk mitigation. In software engineering, an improvement in basic reliability will probably be offset by the addition of new features



# *A prediction*

*Improvements in software testing will not in general lead to improved reliability. They will simply lead to more features at least in the foreseeable future.*

If we judge such a system to be better then we are making progress, however if we are building critical systems, feature introduction must take second place to reliability improvement.

Both feature introduction *and* reliability improvement do not seem to be an option.



# *Overall Summary*

## **To conclude:**

- On the negative side
  - ◆ We don't learn from our mistakes
  - ◆ System diagnosis is at a very poor stage of evolution
- On the positive side
  - ◆ Some technologies are extraordinarily effective



# *Overview*

- ❖ **Legal stuff**
- ❖ **Risk**
- ❖ **Some predictions**



# *Some predictions*

Failure-aware programming

