

Section 3

Reliability and Dependability

sinead.barton@mu.ie



Dependable = capable of being trusted



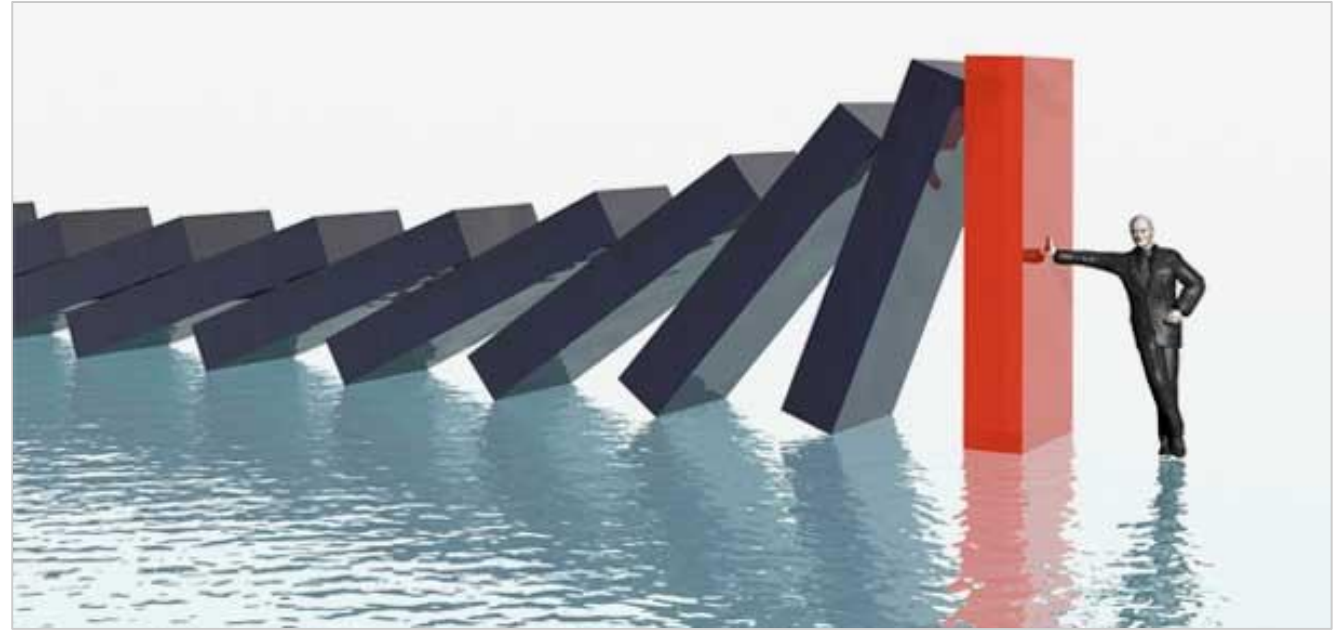
Not Dependable = not possible to trust

Trust



**To believe that something is safe
and reliable**

Reliability



Someone or something that is reliable can be **trusted to behave in the way you expect**

Wikipedia

In software engineering, dependability is the ability to provide services that can defensibly be trusted within a time-period

- In systems engineering, dependability is a measure of a
 - System's availability,
 - Reliability,
 - Its maintainability,
 - Maintenance support performance
- And, in some cases, other characteristics such as
 - Durability,
 - Safety
 - Security

Average User Perspective

Dependability in a piece of technology reflects on the extent of the user's confidence that it will operate as **users expect** and that it will not “**fail**” during normal use. *Ian Sommerville (St. Andrews)*

Another angle on this is that the technology consistently returns the correct answer or does the **correct** action. *Just boring old me (Maynooth)*



Dependability, trust, is subjective.

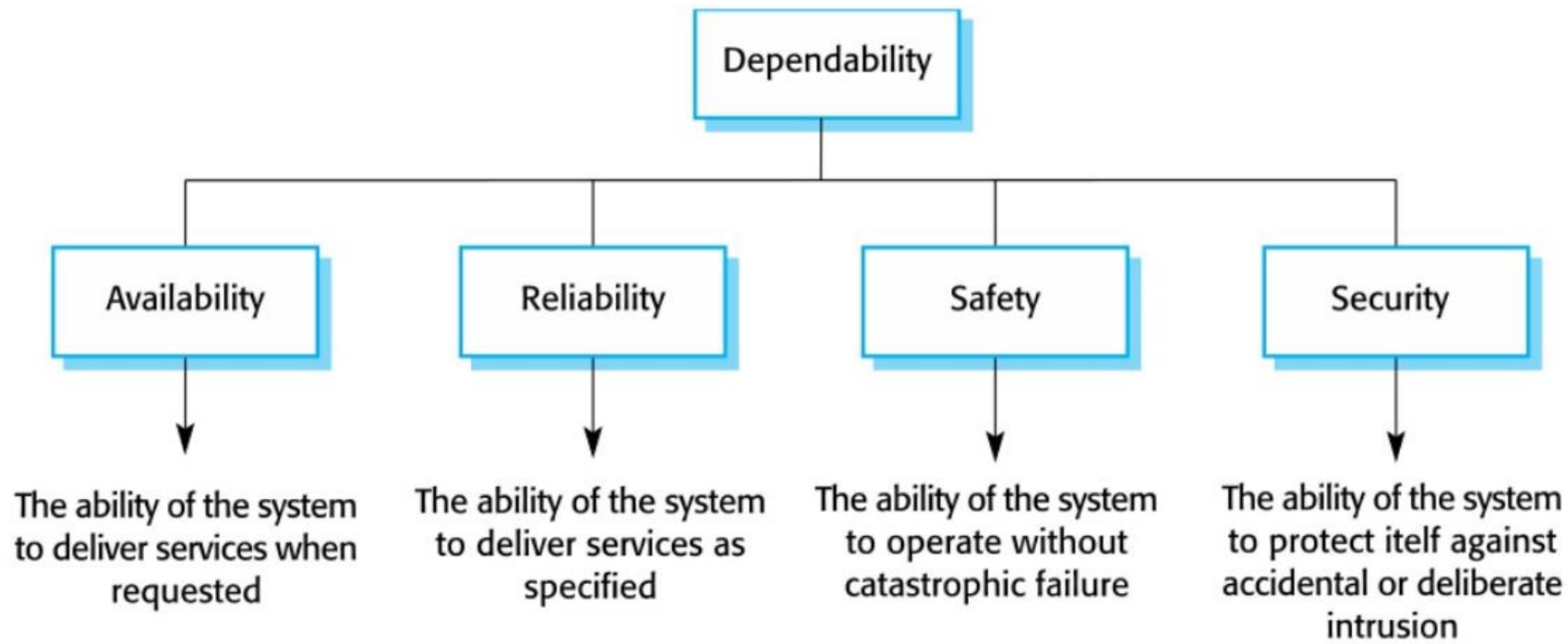
What the user expects may not be the same thing as what the designer built

What may be an acceptable error rate may not be acceptable to

Users

Idiot or fool proofing is easy, having your device survive a techie or semi-techie is much much harder (engineer-proof)

- Do NOT read the manual
- Definitely do NOT read the project specifications
- Will NOT do things the way you expected them too



These are related. It is a probability and can never be perfect. Often describe as MTBF mean-time-between-failures

In some scenarios small, harmless failures are acceptable

Or at least report when it has been hacked so you know it

IoT Security



Hacking and Security



What are IoT devices?


- Internet of Things
- Network of physical objects (things) that are embedded with sensors, software, and other technologies to allow them to connect with other devices over an internet connection.
- IoT devices are most commonly used for smart home appliances e.g. Nest



```
connected to address 112.33
username: *****
password: *****
Access granted...
```

Hacking and Security

- With distributed systems, it has become increasingly easy to “hack” the system to create outcomes that you don’t want.
- If the device is in the hands of the hacker, then you can be guaranteed that someone will be able to interfere with the device.
- Remember hacking goes two-ways.. The service to the user can be hacked, but the operator can be potentially hacked as well

- 
- Most people will try to be responsible
 - But some people might try to be destructive
 - There are shades in between
 - 99% are harmless

White Hat vs Black Hat

Hacking comes in many forms



Noise sensors for nightclubs – put a bag with foam over them



Traffic counters – hack the sensor to show more cars to make the city route around the “busy section”



Temperature sensor on a heating system – use a hairdryer to heat it up – or cool it down.



House window security sensor – use a magnet to hold the switch in place



Any sensor – cut the wires leading to it, or short-them

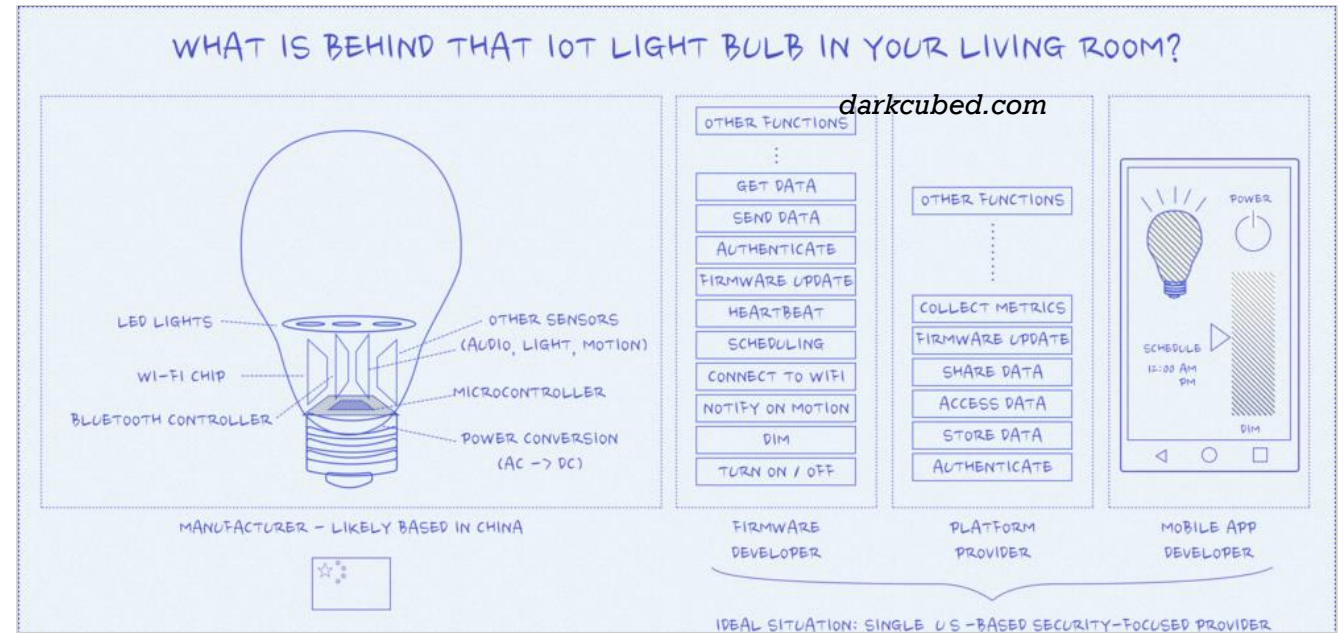


Wireless Car Ignition – clone the radio signal



Wireless Sensors – generate radio noise and block any transmissions

Complexity
opens many new
opportunities



- Was the device itself hacked? (hardware or firmware)
- Was the cloud/platform hacked (who owns it)
- Was the communication links hacked or interfered with
- Was the app on your phone interfered with?

Do you trust them??? (more on this later)



Alexa is always listening... what if a hacker got that data-stream... American police have already tried to get it from Amazon

Fitbit/Smartwatches know what you do, where you go, when you do it. Some companies were caught selling that data, are you sure you want to risk sharing it



Some electricity smart-meters can detect whether you are at home, what you are doing, what products you have in your house, whether they need replacing. Very useful to a burglar or a tele-marketer

A large red speech bubble graphic with a white outline, containing the text "Oooops...".

Oooops...

- Hacking and disruption of sensors is as likely to come from stupidity than maliciousness but it is still a problem
- This is where intentional design error can become a problem... more about that later...

Correctness

- **CORRECT:**
- Free from error or fault; true or accurate;
Conforming to standards; proper behaviour
(social)

Or

- You **agree** that the answer is correct



▼
Give the correct
answer to the
following...

- $5 * 4 = 20$
- The opposite of hot is cold
- When driving, it is better to crash into a child and kill them than to turn and crash and risk killing yourself?
- Landlords own property and they are able to set any prices they wish.
- Pesticides and fertilizers enhance food production and should be encouraged.



Failure



What is failure?

- What is failure?
 - When you don't get the answer/response that you were expecting
- Of course, failures come in a variety of forms and severity

Failures come in
different levels
of severity

- **Minor:** inconvenience, no harm
- **Serious:** loss of income, loss of function, harm, (probably repairable)
- **Catastrophic:** destruction of machine, serious injury, (probably not repairable)

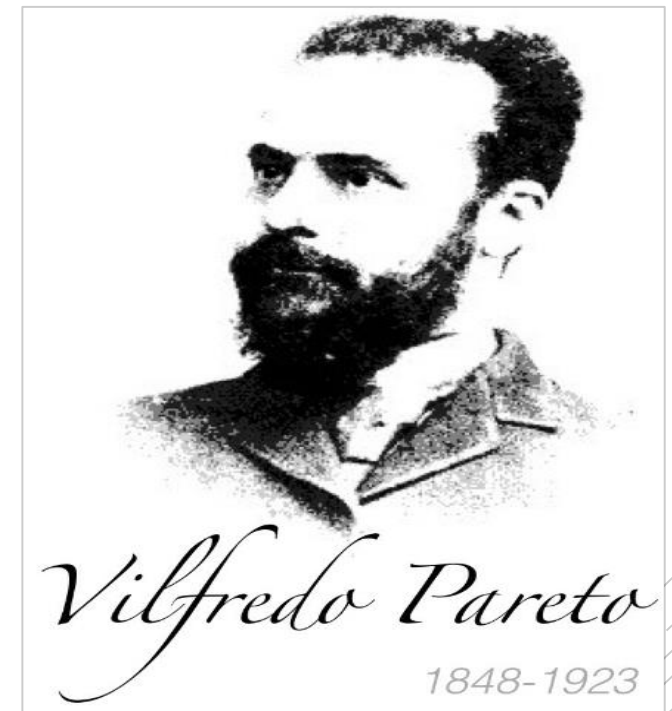
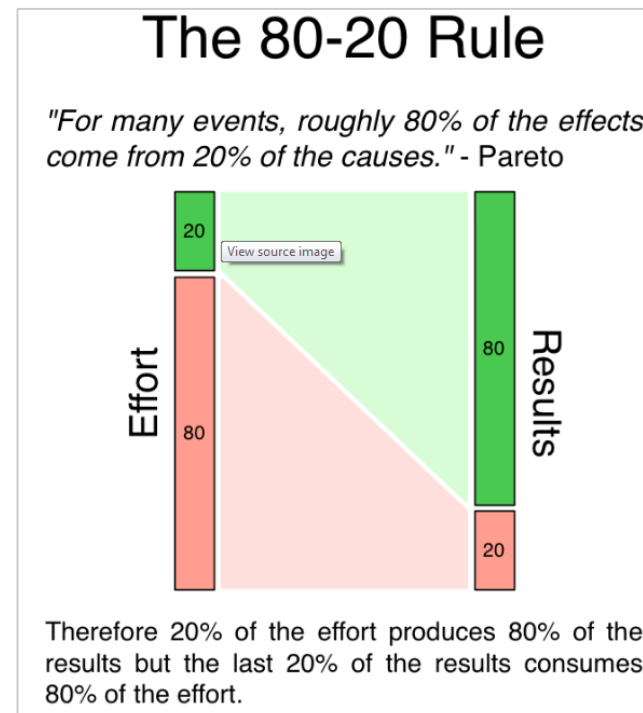


What are the forms of failure?

- Design
 - This occurs because the machine experiences a scenario that it cannot handle – nothing is broken, it is simply beyond its capability. This may lead to a variety of failures
- Manufacturing or Aging Failure
 - The failure occurred due to an error in the manufacture (crack in metal, a bug in software) or something that has developed in time (wear and tear) as the machine has been used.
- Operator
 - The operator did something that results in a failure.

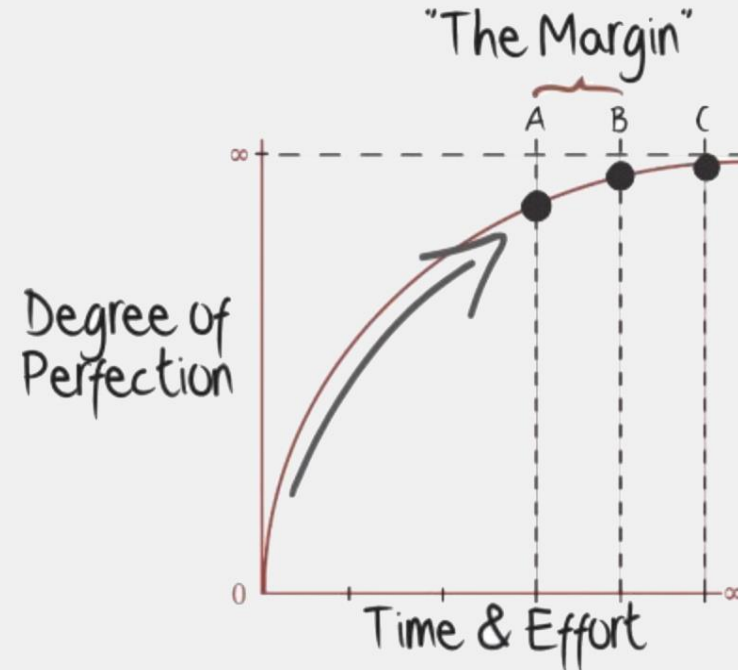
80:20 Pareto Rule

- *“For many activities, roughly 80% of the effects come from 20% of the causes” – Pareto*
- 20% of the effort produces 80% of the results, but the last 20% of the results consumes 80% of the effort
- Perfection is difficult, difficult means expensive



The Perfection Curve

THE EXPONENTIAL CURVE OF EXCELLENCE



- Getting a machine to work right 99% of the time might take 6 months
- Getting it to work 99.9% might take 6 years
- Getting it to work 99.99% might take 66 years

When is
something good
enough?

- You build an implanted machine (like a pacemaker) that measures some physiological condition and gives you an automated injection of drugs. Typically you might need an injection once per week. The machine is 99.99% accurate.
- Done correctly it could save your life
- Done incorrectly, it could have traumatic, perhaps fatal results
- Is this machine safe enough?

When is
something good
enough?

- 99.99% accurate means 1 in 10,000 times is a mistake.
- 1 time per week = 50 times per year.
- If you have 200 users of the device, you will kill 1 per year.
- Is that ok?
- What about 99.999% of the time... you kill 1 in 2000 users per year.

When is
something good
enough?

- **People will usually accept other people making a mistake**
- **They have much less tolerance for a machine making a mistake**

- Errors in this mode are the responsibility of the designer
- **Un-intentional:** They did not foresee all “reasonable” circumstances
- **Intentional:** They excluded certain scenarios as being valid



Design Error

Intentional Design Errors

- It is acceptable to “intentionally” exclude certain scenarios – but in those cases you must make very clear what those scenarios are and that the purchaser/operator is aware of those constraints
- If you intentional ignore a “reasonable” scenario and do not suitably disclose it, then this fraudulent and potentially criminal. Honestly its just wrong and could be dangerous.
- Design errors can end up with “product recalls”

Example

- A company deliberately did not test if their domestic smoke alarms worked in zero-gravity.
 - They did not disclose this to their consumers.
 - Acceptable or unacceptable?
-
- A company deliberately did not test if the smoke alarms that they produced were waterproof.
 - There is a warning on their product that the smoke alarms cannot be installed in high humidity areas.
 - Acceptable or unacceptable?

- Errors in this mode are the responsibility of the manufacturer.
- **Un-intentional:** something goes wrong in the manufacture/coding and was not caught by “reasonable” testing.
- **Intentional:** short-cuts taken, cheaper materials used, reasonable care and testing not carried out.



Manufacturing
Error



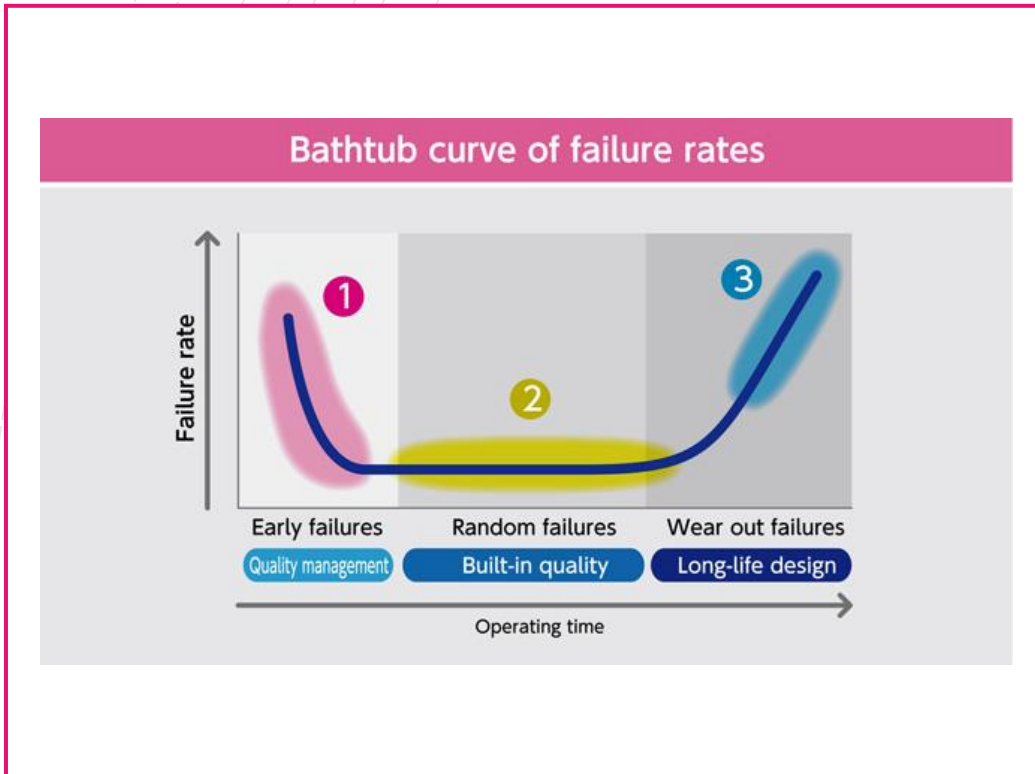
Intentional Manufacturing Errors

- Testing is essential to prove that you've not made an error. Provided that you have done a reasonable level of testing. "Reasonable" is a legal term and would be argued.
- Intentionally doing something that knowingly introduces a failure mode or weakness, and not disclosing it, makes you liable.
- Manufacturing errors can end up with "product recalls"

Example

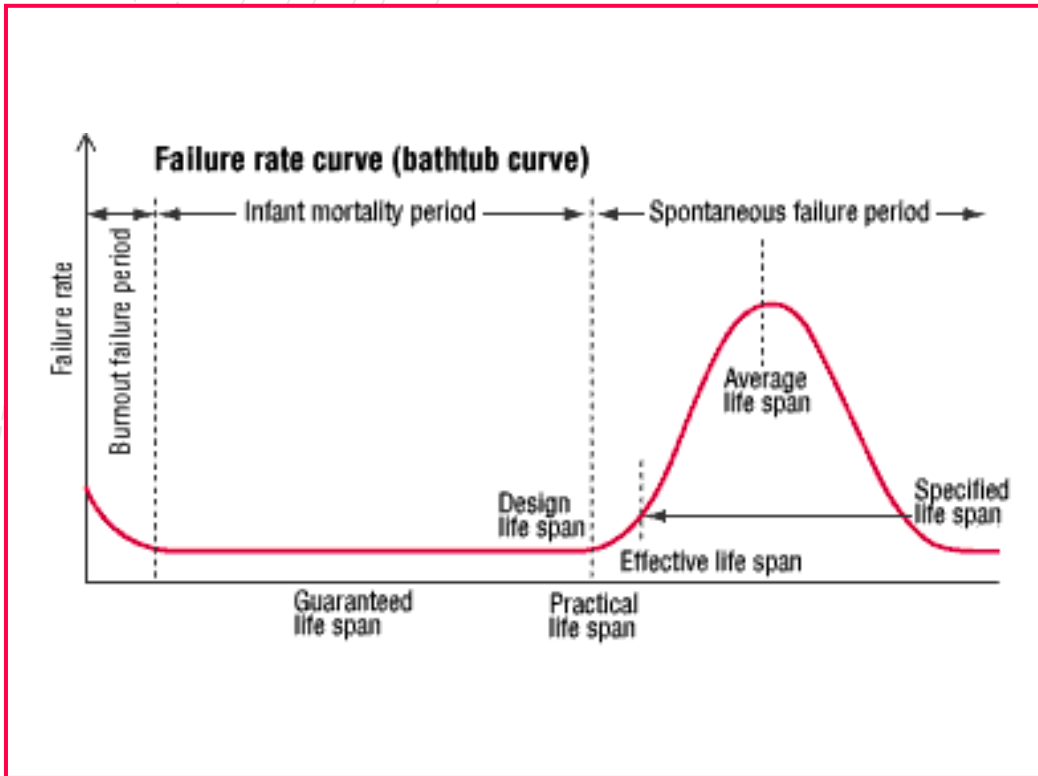
- A company decides to reduce costs by manufacturing a frame out of plastic instead of metal.
 - The plastic is as robust as metal and lighter as well. The company advertises the new material as eco-friendly because it is easier to recycle.
 - Acceptable or unacceptable?
-
- A company chooses to reduce the quality of the capacitors in a product that they produce. The lower quality capacitor will reduce the lifespan of the product.
 - They do not advertise the manufacturing decision that they made.
 - Acceptable or unacceptable?

Aging Error



- Some failures occur early on – and these can be detected with testing – “burn-in testing”
- Mid-life failure is down to your quality control. It shouldn’t regularly happen
- Wear-out failures occur because things get used, worn, and not repaired.

Aging Technology



- Capacitors age. Eventually they will fail
- Certain things make it worse...
- High Temperatures
- Frequent charge and discharge cycles.
- Excessive reverse voltage.
- Application of over-voltages.

Aging Technology

Most consumer electronics don't last more than 5 years, most less than this.



If you are building a 50,000€ robot, if you are implanting a medical device, then you want it to work for 15-20 years if not longer.



How are you going to handle possible faults?

Mechanical
wear and tear

Electronic
component
failure

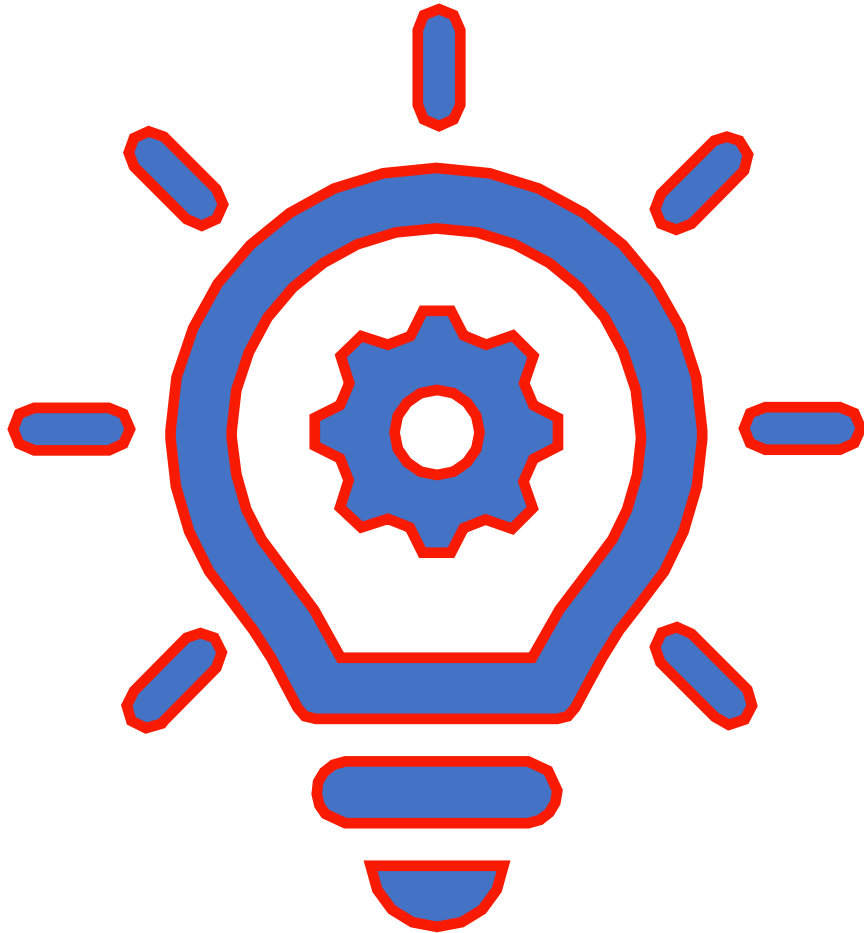
Bad software
updates (or no
updates)

How easy is it to
detect a failure
(diagnose)

How do you
repair your
device

When does a
machine
become unsafe
due to aging?

Possible faults



Human Error

- Often used to scapegoat people when actually the problem is often in the design of the system that makes it possible to have human errors
- Is it possible to have a fool proof system?

Some sources of Human errors



- **Skill-based errors (accidental)**
 - we know what to do but do the wrong thing
- **Rule-based errors (intentional)**
 - we fail to chose the right rule or violate rules
- **Knowledge-based errors**
 - we don't know what we're doing
- **Judgement based errors**
 - excessive trust in the technology in the presence of failure

Faults and Errors

- A **fault** is when something goes wrong in the system, perhaps a software bug, a blown sensor, a loose wire
- A **failure** is when a fault results in a bad or unsatisfactory outcome
- Faults do not have to cause failures.
- Systems should have **checks to detect and handle faults** so that safe or acceptable behaviour is guaranteed.

Reliability is about handling faults

Obviously none of these can be perfect but the aim is to reduce the chances of faults causing failures.

1

Fault Avoidance

- Techniques that minimise the possibility of mistakes or errors (e.g. redundancy, multiple sensors, etc)

2

Fault Detection

- Validation techniques that increase the probability of detecting errors so corrective action can be taken – for example by ensuring that all sensor readings are within an acceptable range

3

Fault Tolerance

- Ensuring that even if we have a fault and thus erroneous data, we do not have the ability to produce an unsafe or undesirable output.

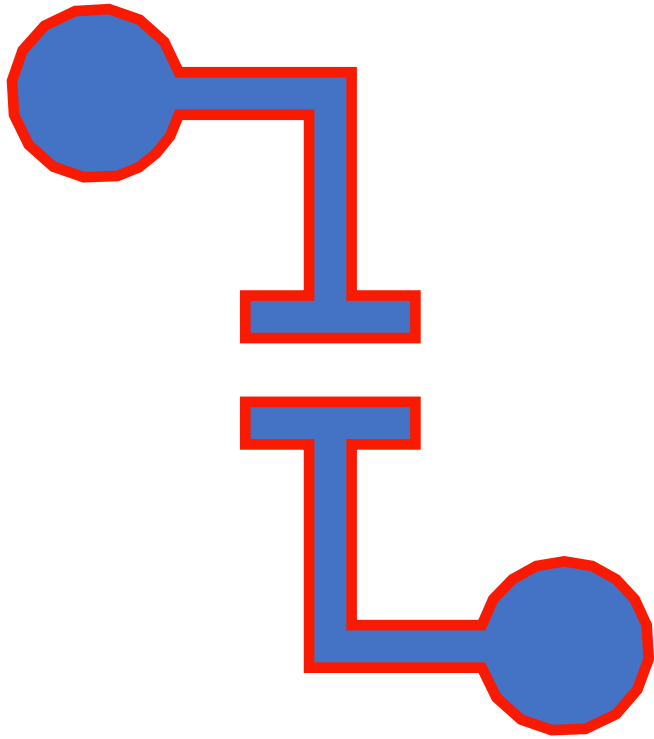
4

Fault Forecasting

- Can you predict when something is about to fail or have a fault?

Safe and Failsafe

- A **fail-safe** is a design feature that in the event of a specific type of failure, the system responds in a way that will cause no or minimal harm to other equipment, the environment or to people.
- So for example if power is lost, or if there is a key sensor failure (eg cameras in an self-driving car), then in that situation, the machine will shutdown in a guaranteed safe way.



Types of Fail-safe

- A hardware failsafe is historically more common than software fail-safes e.g. fuses
- Fail-safe software is possible, but harder to do because a crash due a bug is not a failsafe. That just leaves everything as is.

Safe and Failsafe – Hardware Example 1



- Many devices are protected from short circuit by fuses, circuit breakers, or current limiting circuits. The electrical interruption under overload conditions will prevent damage or destruction of wiring or circuit devices due to overheating.

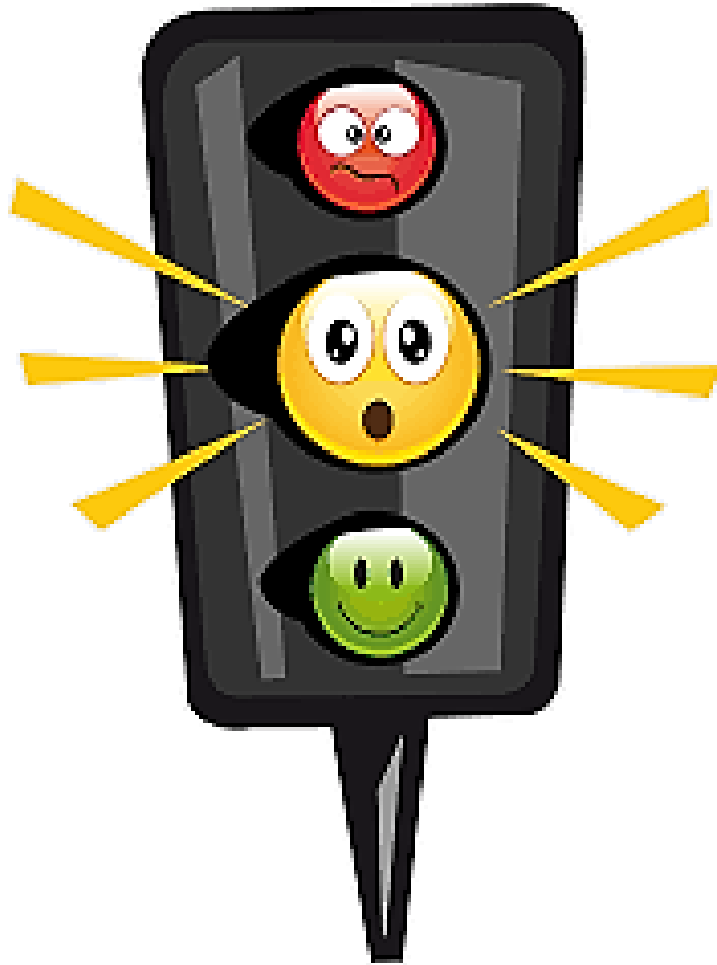
Safe and Failsafe – Hardware Example 2



Drive-by-wire and fly-by-wire controls are electrical or electro-mechanical systems for performing vehicle functions traditionally achieved by mechanical linkages such as an Accelerator Position Sensor.



These typically have two potentiometers which read in opposite directions, such that moving the control will result in one reading becoming higher, and the other generally equally lower. Mismatches between the two readings indicates a fault in the system, and the electronic control unit (ECU) can often deduce which of the two readings is faulty.



Safe and Failsafe – Hardware Example 3

- Traffic light controllers use a Conflict Monitor Unit to detect faults or conflicting signals and switch an intersection to an all flashing error signal, rather than displaying potentially dangerous conflicting signals, e.g. showing green in all directions.

Safe and Failsafe – Software Example

- The automatic protection of programs and/or processing systems when a computer hardware or software failure is detected in a computer system. A classic example is a watchdog timer.
- In industrial automation, alarm circuits are usually "normally closed". This ensures that in case of a wire break the alarm will be triggered.
- Power detect signals are “normally high” so if power is lost, the signal goes low and the system goes into a safe-shutdown process.
- Analog sensors can usually be installed such that a sensor failure results in an out-of-bound reading which can be detected. For example - a potentiometer indicating pedal position might only travel from 20% to 80% of its full sensor range, such that a cable break or short results in a 0% or 100% reading.

Reliability and Safety



A system can behave exactly as specified or designed, and still be unsafe.



Correctly behaving as designed is a measure of reliability.



If your design is insufficient or incomplete, then you may have a system that **works reliably in an unsafe way**.

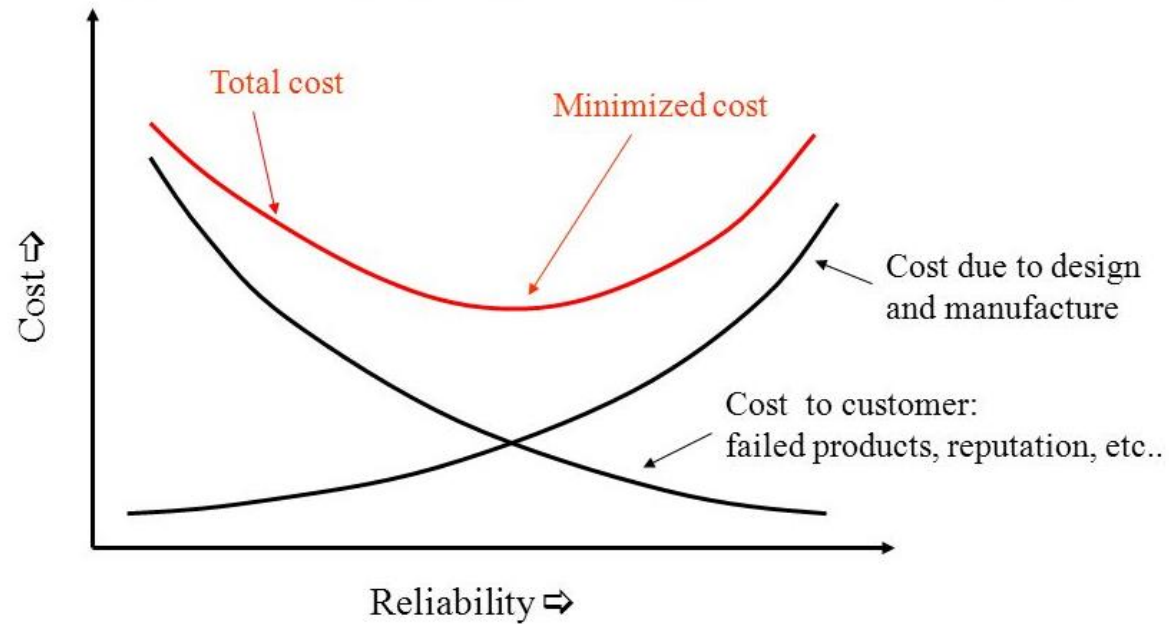


An unreliable system is a system that doesn't work as designed.



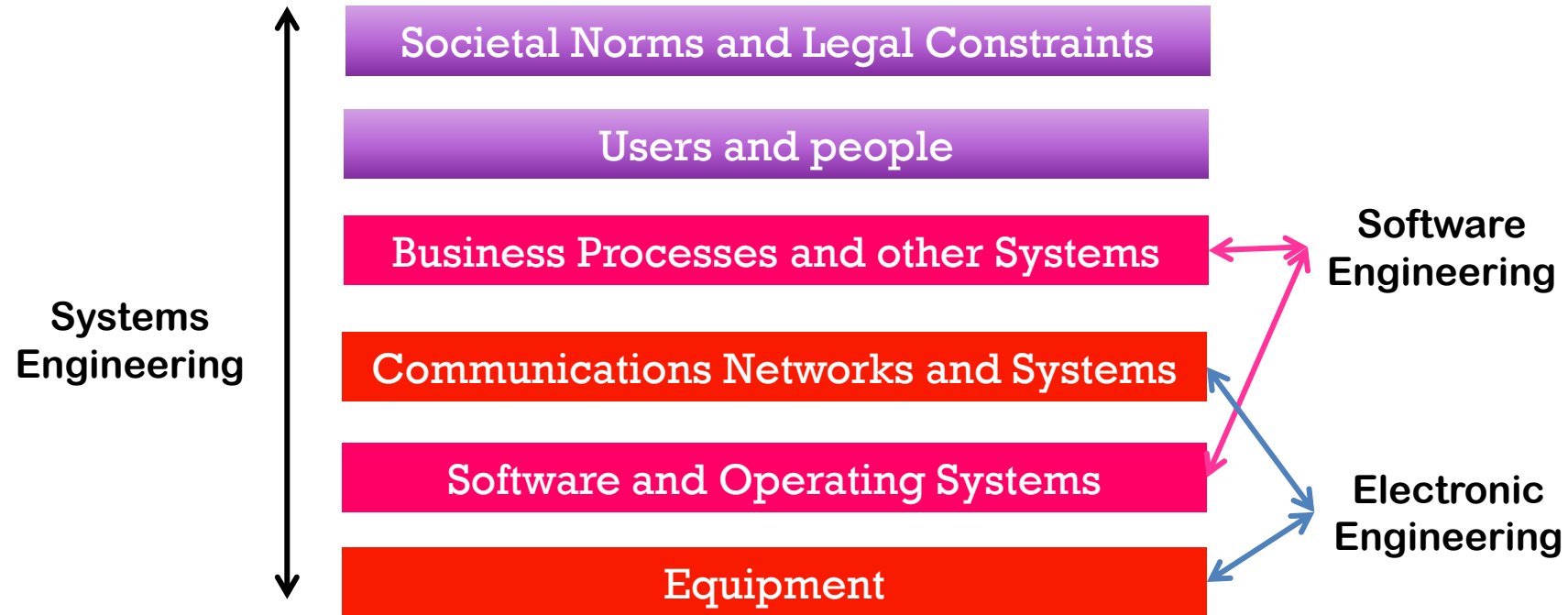
An unreliable system is inherently a less safe system.

Cost



- Dependability and Reliability costs. There is a balance between affordability, performance and dependability.
- Remember dependability is user-subjective

Final Comments – A technology stack




Dependability is a design feature

It is difficult to retrofit significant improvements to dependability as the most challenging failures are due to interactions between the different layers, rather than the performance of an individual component

Case Studies





Case Studies – Maroochy Water Breach

- A disgruntled employee hacked the centralised control system and caused the water sensors to send back false data
- This resulted the system opening valves and releasing untreated sewerage into rivers and local parks.
- **System worked as designed**, it just wasn't expecting to have **malicious data**



Your PC ran into a problem and needs to restart. We're just collecting some error info, and then we'll restart for you. (0% complete)

If you'd like to know more, you can search online later for this error: HAL_INITIALIZATION_FAILED

Case Studies – Radiation Induced Bit-Flipping

- In the 1990's, IBM did a study that says cosmic rays would flip a bit in 256 MB of memory once a month. Chips are smaller and more vulnerable, so you could say a computer with 8 GB of RAM has a **bit changed each day**.
- One bit could crash your computer by changing a line of code.
- How do you handle that?

De Puy 2010 Product Recall



- In 2010 De Puy Orthopaedics recalled an ASR hip implant due to a large number of failures that seriously injured a significant number of people (1 in 8 patients).
- This cost De Puys parent company (Johnson and Johnson) more than \$3 billion in a lawsuits for damages caused by the implant.

How did this happen?

- US Food and Drug Administration
- FDA cleared vs FDA approved
- Both the ball and socket of the implant was made of metal
- No clinical testing was performed
- As the metal rubbed together it degraded and resulted in high failure rates for the implant.
- After 5 years it was necessary to remove the implant and many people required extensive medical care.

Case Studies - Galactic Virgin Crash



BBC

Sign in

News

Sport

Weather

Shop

Reel

Travel

More

NEWS

Home

Video

World

UK

Business

Tech

Science

Stories

Entertainment & Arts

US & Canada

Virgin Galactic spacecraft crash kills pilot

1 November 2014



Share

Case Studies - Galactic Virgin Crash

- The really clever part is how SpaceShipTwo re-enters the Earth's atmosphere. SpaceShipTwo doesn't actually go into orbit. It does, however, have to slow down its descent (by generating drag) and it needs to ensure it remains facing the 'right way up' throughout.
- One of the philosophies adopted in SpaceShipTwo's design is that automation is minimised, with control being left with the pilots
- Scaled Composites took the view that **minimising automation also minimised the number of systems that could go wrong**. Pilot intuition, reflexes and control would be the first and last line of defence.

Generating drag

- Generating drag, however, is a double-edged sword: while it's needed at re-entry, it needs to be minimised during the boost phase. SpaceShipTwo solves this conflict by using a feathered system – it changes its shape during different stages of the flight.
- During the boost stage the feather remains undeployed and drag is minimised, but during re-entry the pilot and co-pilot deploy the feather, which rotates through 60° and dramatically increases the drag on the vehicle (main image).

How the feather-drag works

- Once above 1.4 Mach the aerodynamic forces acting on the feather prevent its deployment, and since the actuators used to achieve deployment are not designed to prevent deployment, the feather can be safely unlocked at this speed without fear of the feather deploying unintentionally.
- However, below speeds of 1.4 Mach, during the transonic range (0.9–1.1 Mach), the aerodynamic forces acting on the feather are such that they act not to prevent, but to cause, deployment. Therefore, unlocking below 1.4 Mach can result in accidental deployment.

How the catastrophic failure occurred

- On the morning of October 31, 2014, SpaceShipTwo reached 0.8 Mach, and the forward-facing cockpit camera and flight data indicate that the co-pilot, Alsbury, called out the airspeed as '0.8 Mach'1 . He then moved the feather from the 'locked' to 'unlocked' position.
- Thus, unlocking occurred not at 1.4 Mach, but at about 0.82 Mach, in the transonic range, when the aerodynamic forces act to deploy the feather. These forces were sufficient to overcome the capacity of the deployment actuators, which occurred quickly after unlocking. The increased drag on the vehicle during this phase of flight resulted in it losing aerodynamic stability and breaking apart, **as it was essentially folded in half.**

Who was at fault?

- It transpires that Scaled Composites was very aware of the catastrophic consequences of early deployment during the boost phase, but the National Transportation Safety Board (NTSB) found that “there was insufficient evidence to determine whether the pilots fully understood the potential consequences of unlocking the feather early”.
- The NTSB investigation would conclude that **“the probable cause of this accident was Scaled Composites’ failure to consider and protect against the possibility that a single human error could result in a catastrophic hazard to the SpaceShipTwo vehicle.**

What lesson was learned?

- The philosophy of minimal automation in the design of the vehicle left a critical vulnerability: **no capability to prevent and manage a human error**. It was foremost a system failure – it ignored human fallibility, a constant threat regardless of the expertise and experience of the individuals involved.
- *“It is often the best people who make the worst mistakes”*, James Reason
- *“Human error may be the most important cause of failure, but human judgement may be our best safeguard against it. Between these two extremes is a line we must all tread”*, (slightly modified) comment by David Brosnan