Software Engineering

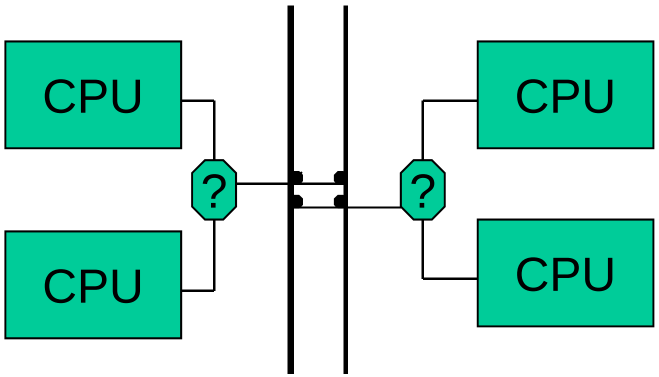
Computer Science Tripos 1B
Michaelmas 2011

Richard Clayton

Lecture Four

Redundancy

- Some vendors, like Stratus, developed redundant hardware for 'non-stop processing'



- Stratus users then found that the software is where things broke:
  - note that the 'backup' IN set in Arianne failed first
- Next idea: multi-version programming
  - BUT errors significantly correlated, and failure to understand requirements comes to dominate (Knight/Leveson 86/90)

737 Cockpit

Panama crash, June 6 1992

- Need to know which way is up!
- New Electronic Flight Information System (each side), old artificial horizon in middle
- Both EFIS fed off same gyros, thought to be OK because of the AH
- EFIS failed – loose wire
- Pilots watched EFIS, not AH (bigger and right in front of them)
- 47 fatalities
- And again: Korean Air cargo 747, Stansted Dec 22 1999

Kegworth crash, Jan 8 1989

- BMI London-Belfast, fan blade broke in port engine
- Crew shut down *starboard* engine and did emergency descent to East Midlands Airport
- Opened throttle on final approach: no power
- 47 dead, 74 serious injuries
- Initially blamed wiring technician! Later: cockpit design (pilots had misunderstood airflow bringing smoke into cockpit, and had not consulted relevant instruments)

Complex socio-technical systems

- Aviation is actually an easy case as it's a mature evolved system!
- Stable components: aircraft design, avionics design, pilot training, air traffic control ...
- Interfaces are stable too
- The capabilities of crew are known to engineers
- The capabilities of aircraft are known to crew, trainers, examiners
- The whole system has good incentives for learning and significant effort is made to learn every possible lesson from every incident

Cognitive factors I

- Trained-for problems are dealt with using rules we evolve, and are partly automatic
  - operators are taught (or just deduce) rules of what to do
  - operators may not have access to true state of system but infer it
  - when environment changes but rules don't, you get errors
- Over time, routine tasks are dealt with automatically
  - the rules have given way to skill
- Many errors derive from highly adaptive mental processes
  - we deal with novel problems using knowledge, in a conscious way
  - in unusual system states operators try to reason about what is going on; they may try experiments to test/refine their knowledge
  - if a test succeeds operator was clever, if it fails they are blamed
- Read up the psychology that underlies errors!

Cognitive factors II

- The ability to automatise routine actions leads to absent-minded slips, aka ‘capture errors’
  - driving ‘home’ to your old house
- Slips and lapses
  - forgetting plans, intentions; strong habit intrusion
  - misidentifying objects, signals (often Bayesian)
  - retrieval failures; tip-of-tongue, interference
  - premature exits from action sequences, e.g. leaving card in ATM
- Rule-based mistakes; applying wrong procedure
- Knowledge-based mistakes; heuristics and biases

Cognitive factors III

- Training and practice help – skill is more reliable than knowledge!
- Error rates (motor industry):
  - inexplicable errors, stress free, right cues – 10^{-5}
  - regularly performed simple tasks, low stress – 10^{-4}
  - complex tasks, little time, some cues needed – 10^{-3}
  - unfamiliar task dependent on situation, memory – 10^{-2}
  - highly complex task, much stress – 10^{-1}
  - creative thinking, unfamiliar complex operations, time short & stress high – $O(1)$

Cognitive factors IV

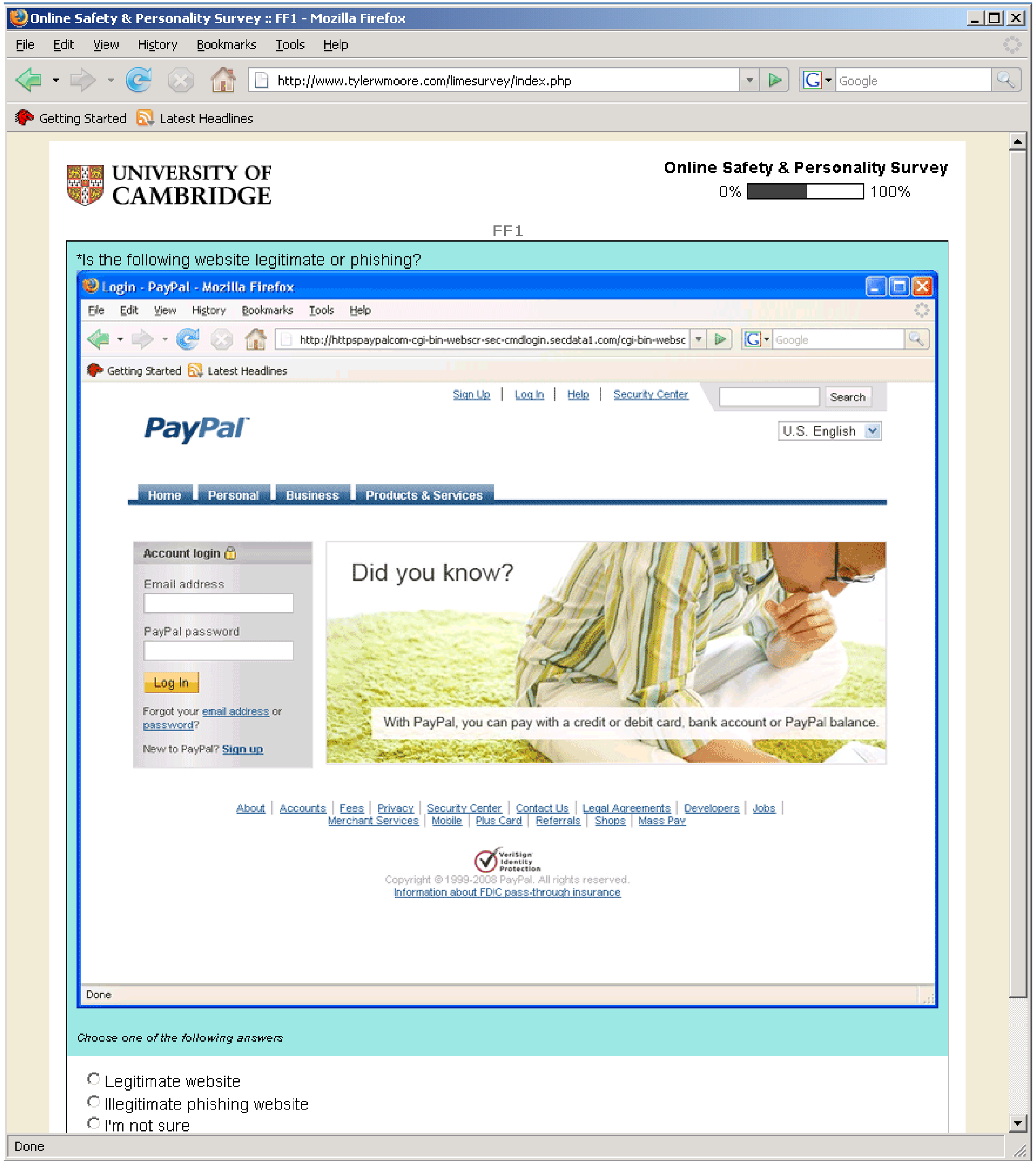
- Violations of rules matters
  - they're often an easier way of working, and sometimes necessary
  - you don't fix safety problems by telling people not to do something
  - the 'right' way of working should be easiest: look where people walk, and lay the path there
- 'Blame and train' as an approach to systematic violation is suboptimal
  - July 86 (LAX) pilot reaching for fuel switch accidentally turned off both engines, plane dropped from 1700 to 600 feet before restarted. Instead of just blaming pilot a safety guard was added.
- The fundamental attribution error
  - if he trips over a rock he's clumsy, if I do, the rock is in the way!
- Need right balance between 'person' and 'system' models of safety failure

Cognitive factors V

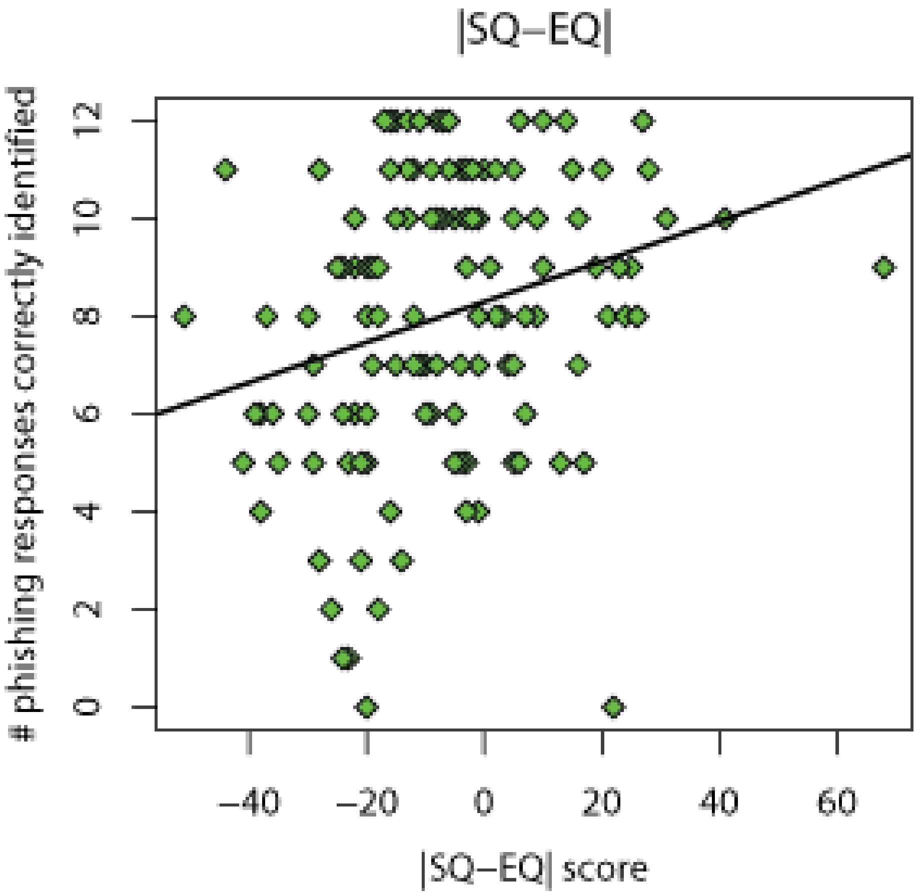
- Ability to perform certain tasks can very widely across subgroups of the population
- Age, sex, education, … can all be factors
- Risk thermostat – function of age, sex

For example:

- Banks tell people ‘parse URLs’
- Baron-Cohen: people can be sorted by SQ (systematizing) and EQ (empathising)
- Is this correlated with ability to detect phishing websites by understanding URLs?


Online Safety & Personality Survey :: FF1 - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://www.tylerwmoore.com/limesurvey/index.php
Google
Getting Started Latest Headlines
UNIVERSITY OF CAMBRIDGE
Online Safety & Personality Survey
0% 100%
FF1
*Is the following website legitimate or phishing?
Login - PayPal - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://httpspaypalcom-cgi-bin-webscr-sec-cmdlogin.secdata1.com/cgi-bin-websc
Google
Getting Started Latest Headlines
Sign Up | Log In | Help | Security Center
Search
PayPal
U.S. English
Home Personal Business Products & Services
Account login
Email address
PayPal password
Log In
Forgot your email address or password?
New to PayPal? Sign up
Did you know?
With PayPal, you can pay with a credit or debit card, bank account or PayPal balance.
About | Accounts | Fees | Privacy | Security Center | Contact Us | Legal Agreements | Developers | Jobs |
Merchant Services | Mobile | Plus Card | Referrals | Shops | Mass Pay
VeriSign Identity Protection
Copyright © 1999-2008 PayPal. All rights reserved.
Information about FDIC pass-through insurance
Done
Choose one of the following answers
Legitimate website
Illegitimate phishing website
I'm not sure
Done

Results



- Ability to detect phishing is correlated with SQ-EQ
- It is (independently) correlated with gender
- The ‘gender HCI’ issue applies to security too

Cognitive factors VI

- People’s behaviour is strongly influenced by their team
- Social psychology is a huge subject!
- Note selection effects – e.g. risk aversion
  - corporate security officers tend to be risk-averse
  - entrepreneurs tend to be more risk-loving
  - so large firms spend too much on security & small firms too little
- Some organisations focus on inappropriate targets
  - disabling safety interlocks to raise production by 5%
  - NASA were more concerned about schedules than safety and lost Challenger when the O-ring failed
- Add in risk dumping, blame games
- It can be hard to state the goal honestly!

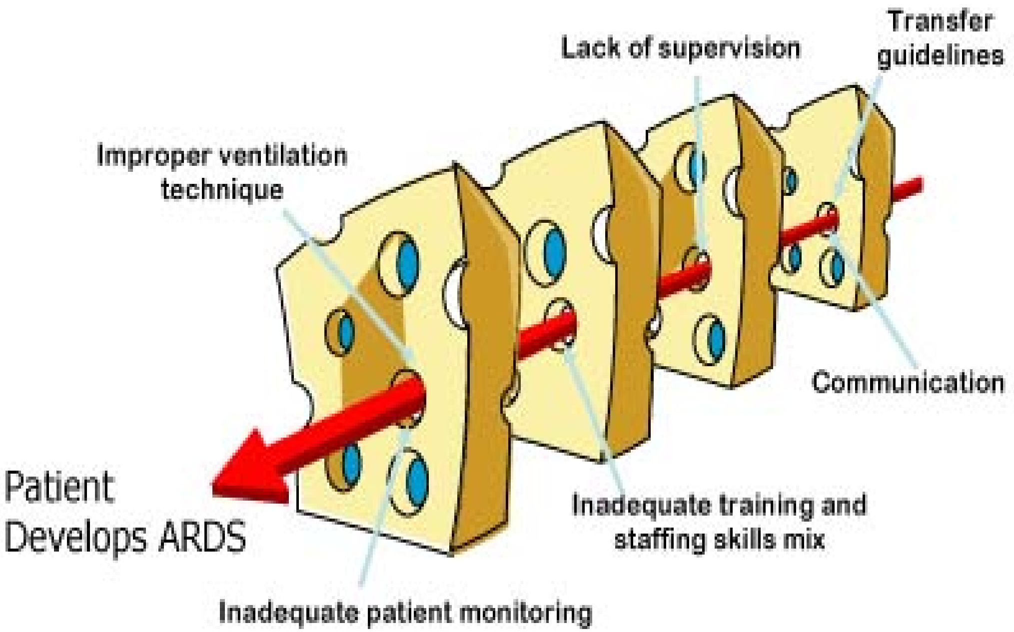
Software safety myths I

- ‘Computers are cheaper than analogue devices’
  - shuttle software costs $100m pa to maintain (1993)
- ‘Software is easy to change’
  - exactly! But it’s hard to change safely
- Computers are more reliable’
  - 16 potentially fatal bugs identified in shuttle software (to 1995); half of them had flown. 12 lower severity bugs triggered in flight
- ‘Increasing reliability increases safety’
  - they’re correlated but not completely
  - safety is a system property
- ‘Formal verification can remove all errors’
  - not even for 100-line programs. That said, is widely used on hardware & some subsets of real systems have been verified

Software safety myths II

- Testing can make software arbitrarily reliable
  - for MTBF of 109 hours you must test 109 hours
- Software re-use increases safety
  - not in Arianne, Patriot and Therac, it didn't
  - several aviation examples relating to Greenwich meridian, flying across the equator or over the Dead Sea ('below sea level')
- Automation can reduce risk
  - sure, if you do it right – which often takes an extended period of socio-technical evolution

Defence in depth



- Reason's 'swiss cheese' model
- Stuff fails when holes in defence layers line up
- Thus: ensure human factors, software, procedures complement each other

Pulling it all together I

- First, understand and prioritise hazards. e.g. the motor industry uses:

1. Uncontrollable: outcomes can be extremely severe and not influenced by human actions
2. Difficult to control: very severe outcomes, influenced only under favourable circumstances
3. Debilitating: usually controllable, outcome at worst severe
4. Distracting; normal response limits outcome to minor
5. Nuisance: affects customer satisfaction but not normally safety

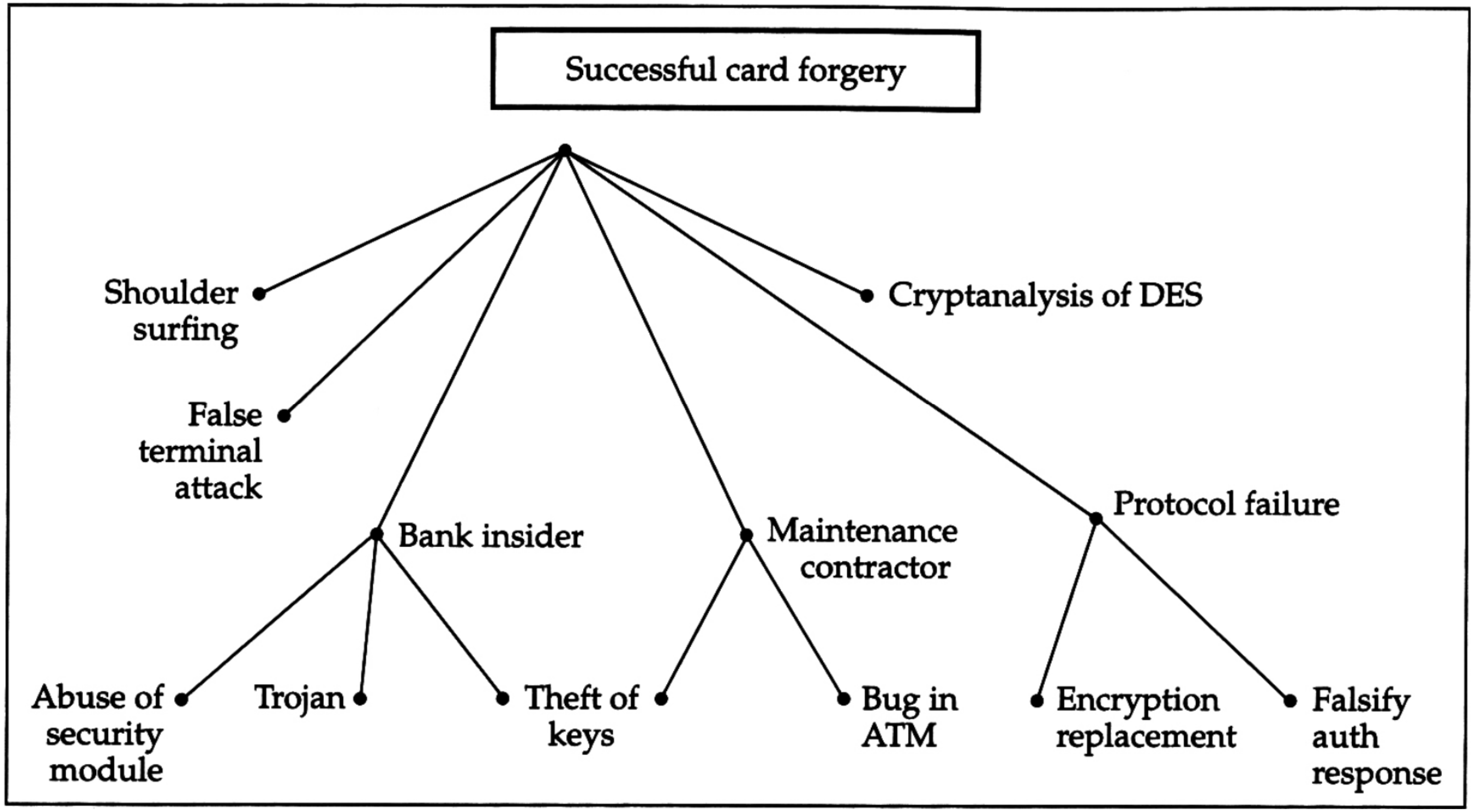
- Develop safety case: hazards, risks, and strategy per hazard (avoidance, constraint)

Pulling it all together II

- Who will manage what?
  - trace hazards to hardware, software, procedures
  - trace constraints to code, and identify critical components / variables to developers
  - develop safety test plans, procedures, certification, training, etc
- Figure out how all this fits with your development methodology
  - waterfall, spiral, evolutionary ...
- Managing relationships between component failures and outcomes can be bottom-up or top-down
- Bottom-up: NASA's 'failure modes and effects analysis' (FMEA)
  - look at each component and list failure modes
  - then use secondary mechanisms to deal with interactions
  - software not within original NASA system – but other organisations apply FMEA to software

Pulling it all together III

- Top-down – fault tree (in security, a threat tree)
  - work back from identified hazards to identify critical components



Pulling it all together IV

- Although some failures happen during the ‘techie’ phases of design and implementation, most happen before or after
- The soft spots are requirements engineering, and later on operations / maintenance
    - these are the interdisciplinary phases, involving systems people, domain experts and users, cognitive factors, and institutional factors like politics, marketing and certification
- Managing a critical property – safety, security, real-time performance – is hard!

The “bug heard around the world” I

- April 10 1981 (with the world watching)
- Computer glitch delayed first shuttle orbital flight at T-20m
- Shuttle has 4-fold redundancy (Fail Operational / Fail Safe)
- The 4 control computers all ran the same code and voted
- The same code was a concern, so had added a fifth computer, with independently written software

The “bug heard around the world” II

- The fifth listened to bus traffic and compared decisions
  - if decisions incorrect, astronauts invited to switch system
  - bus traffic synchronised (so telemetry simpler to perform)
  - refused to listen to bus when it was supposed to be idle
- The 4 computers needed the same clock values
  - hardware access caused inconsistencies, so would examine top of ready-to-run process queue. This held a consistent value of “soon”
  - only at system start would hardware clock be consulted
- Pre-launch the 4 processors had a few processes out of synch
  - the 5th machine failed to see any data on the bus – hence the abort
  - in fact the majority of processes were one cycle late

The “bug heard around the world” III

- A software change 2 years earlier meant an invocation of a common routine to initialise the data bus. This had a delay in it which was achieved by putting oneself onto process queue.
  - then 1 year before launch the delay had been made slightly longer to prevent routine hogging CPU when it was used elsewhere during critical flight processing
- Hence the wrong time value seen by the first processor turned on – but only 1 chance in 67 that this affected the bus timing…
- So ‘switching it off and on again’ would have fixed problem
  - problem very hard to spot in testing – needs an almost complete set of components to manifest itself (or very accurate test harness to simulate them)
  - was in fact seen in the lab some 4 months before launch, but significance wasn’t realised and it never happened again …