Data Risk Cygnology



Black Swans, Perfect Storms, Dragon Kings and Pudding Lanes – sound like plot elements of an elaborate fantasy novel! However, these are concepts that are relevant to system safety and describe different ways in which events can conspire and lead to catastrophic outcomes. Mike Parsons and Paul Hampton explain four of these concepts from a data perspective.

Black Swan Data

Data that is totally unexpected by those receiving it (i.e. 'out of the blue') and has huge (detrimental) impact, but in retrospect should have been anticipated



In 2013 a C-130 Hercules aircraft landed at Bar Yehuda airport near the Dead Sea [1], a saltwater lake sitting astride Israel and Jordan. The Dead Sea is the lowest place and the airfield lies -1,210 feet below sea level.

The aircraft's navigation system became unresponsive and the constellation of GPS satellites above, mysteriously winked out of existence. As it turned out, the plane's navigation electronics were not designed to operate at altitudes less than 400 feet below mean sea level. In a sense, the plane thought it was underwater!

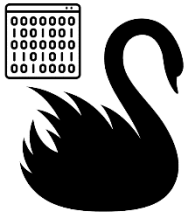


This is what we call a "Black Swan Data" event; the stream of altitude data going to the navigation systems surprisingly turned significantly negative and could have had a catastrophic result; a situation not considered possible. Yet, with hindsight, and knowledge of earth's geography, we can rationalise this and question why the assumption of positive altitude data was ever thought to be true in the first place.

Might your system be susceptible to Black Swan Data? Presumably not something by its very nature that has been considered; but what can be done to protect against it?

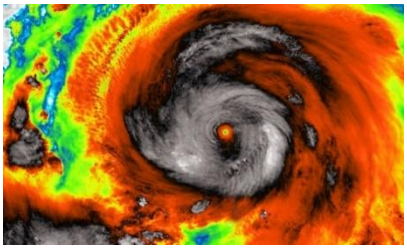
This is an area for further study, but we can offer the following suggestions:

- Be aware of changes of usage, environmental conditions or failures that may produce data that is not expected. The Ariane 501 launch disaster may be thought of in these terms, as the horizontal bias values that caused the exception leading to the complete loss of mission were not anticipated.
- Continually test the established thinking; challenge the oft quoted "that's just how we do things around here".
- What assumptions have we made about the data? Are these really valid and do they continue to be valid in a changing and dynamic world?
- What happens if different inputs (e.g. new sensors or information flows) are added to your system?
- Think outside the box; and we mean, really outside the box. It is the most wild and implausible (compared to current perceptions) ideas that will trip a Black Swan Data event – what could possibly be 'bad data' and how might we handle it?



Perfect Storm Data

Combinations, sets or occurrences of data (or absences of data) that would never have been thought possible to occur together, and when they do have a large and undesirable impact



A Perfect Storm in the risk world is when everything bad happens at the same time, and you did not anticipate that this could happen. These risks get their name from The Perfect Storm, both a 2000 American biographical disaster drama film [2] and a 1997 non-fiction book by Sebastian Junger [3].

In the data world a Perfect Storm is when the data that your system relies upon are all wrong (or missing) at the same time: it can be layers deep. A classic example might be to try and restore deleted data from a backup system, only to find that the backups had not been working properly due to a configuration problem.

So Perfect Storm Data is when multiple data faults or losses conspire to create a situation that has disproportionately catastrophic outcomes.

Another example might be from security: on 10th Dec 2021 a new critical zero-day vulnerability [4] was detected that affected Apache Log4j 2 Java library. It adversely impacted the digital domain and security systems worldwide. The vulnerability, when exploited, permitted remote code execution on the vulnerable server with system-level privileges. The exploit was a combination of both the Java code that contains different logging functions and the settings of a configuration file.

Another example might be the China Airlines A333 at Taipei on 14th June 2020 [5] when all primary computers, reversers and auto brakes failed on touchdown. During landing, flight controls reconfigured from 'normal law' to 'direct law' after all three flight control primary computers (FCPCs) became inoperative. While all aircraft primary control surfaces were still controllable, the deceleration devices including ground spoilers, thrust reversers, and autobrake were lost; the deceleration of the aircraft had to be performed by manual brake.

The following might be considered for management:

- Analyse combinations of data that are important to you: how might one data set being wrong influence another?
- Look at dependencies in your data: are seemingly different sets all derived from a common source?
- Examine as many levels of system and data as you can to establish commonalities and dependencies
- Even if no commonalities are found, consider the unlikely cases of independent data being wrong together, and follow through their consequences



Pudding Lane Data

Data which is unknowingly and unexpectedly critical to the whole operation and if missing or incorrect in some way has a dramatic and negative impact. After the event it may be obvious that it was critical.

"Lord! what sad sight it was by moone-light to see, the whole City almost on fire, that you might see it plain at Woolwich, as if you were by it".



With these words from Samuel Pepys diary in 1666 [6], we see the account of the Fire of London that destroyed 13,200 houses, 87 parish churches, The Royal Exchange, Guildhall and St. Paul's Cathedral, all from a single fire thought to have started in a humble baker's shop in Pudding Lane.

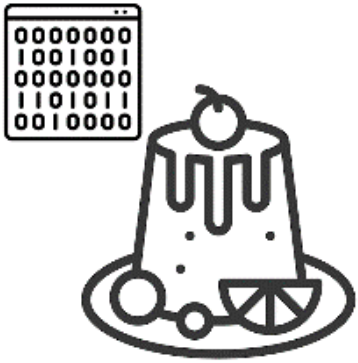
In the same way, unexpectedly critical data can also act as a catalyst to catastrophic outcomes if the dependencies are not identified and adequately mitigated. Often this data is part of an existing system and may be long forgotten about (e.g. a system configuration parameter). It may also be a hard-wired parameter in legacy source code which the maintainers of the code change because they do not understand its original purpose.



In 2015, an Airbus A400M Atlas cargo plane on a test flight crashed at La Rinconada, Spain, less than 5 kilometres from Seville Airport, killing 4 of the 6 crew [7]. Several reports suggested that as many as three of the aircraft's four engines failed during the A400M's departure from Seville.

The suspected cause of the failure was that the torque calibration parameter data had been accidentally wiped on three engines as the engine software was being installed, which would prevent the engine control software from operating properly. In this case, the seemingly innocuous calibration data had a critical role to play in the safety of the entire platform and we introduce the term "Pudding Lane Data" to encompass this form of data. We suggest the following considerations for this type of data:

- Understand and know all your data. The high profile data streams will of course be first and foremost, but do you have an inventory of all the data that the system depends upon and the implications if there is loss of their data properties?
- Configuration and tailoring data may seem static and uninteresting but what are the dependencies on this data? If loss of properties of this data can have catastrophic outcomes, then what processes and mitigations will you put in place to assure those properties are maintained?
- Consider the reporting and monitoring data: if this is incorrect will poor decisions be taken?



Dragon King Data

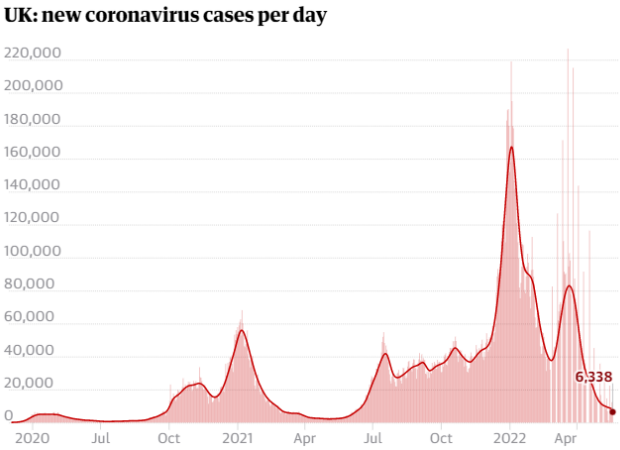


Data whose effect might have been foreseen, but leads to an unexpected and explosive escalation with major impact ('things get rapidly out of control').

In risk management a Dragon King is defined as an event that is both extremely large in size or impact (a "king") and born of unique origins (a "dragon") relative to other events in the system.

Dragon King events are generated by, or correspond to, mechanisms such as positive feedback, tipping points, bifurcations, and phase transitions that tend to occur in non-linear and complex systems, and serve to amplify events to extreme levels. In the data domain we introduce "Dragon King Data", to represent incorrect or missing data whose consequences unexpectedly escalate into an unpredicted and often catastrophic system event.

An example might be the loss of data relating to Covid-19 test results in the UK due to a row limit in Microsoft Excel. This led to Public Health England losing 15,841 positive test results which in turn meant that 50,000 potentially infectious people were missed by contact tracers and not told to self-isolate [8]. Because of the exponential way that viruses spread, this may have led to many more infections and deaths.



Another, older, example could be the infamous comment given by the weather forecaster Michael Fish about the Great Storm of 1987 [9]:

> *"Earlier on today, apparently, a woman rang the BBC and said she heard there was a hurricane on the way. Well, if you're watching, don't worry, there isn't!"*



These few words may well have led to precautions not being taken across the whole of Southern England, affecting millions of people. In fact the storm was the worst to hit South East England for three centuries, causing record damage and killing 19 people.

Some ways this can be managed are:

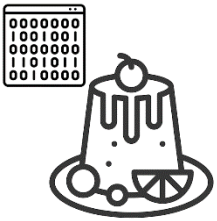
- Identify all the receiving systems or users of your data: what are the consequences of errors or losses in the data to those receiving it?
- Analyse the effects of data impact: are any effects subject to non-linear effects? Try to establish the consequence chains and knock-on effects
- Do any of your data potentially impact systems that have large fan-out or spread?
- Create models (or even Digital Twins) of the system in a way that different combinations of data can be tested and simulated to reveal extreme non-linear behaviours



Conclusions

Looking at Data Risks this way can help to tease out some unlikely but severe failures of systems involving data. It is possible that sometimes these risks are already considered in system Hazard and operability study (HAZOP) sessions but discounted as too unlikely to happen. We suggest this approach is a useful and visual 'Tool in the Box' when trying to identify extreme data-related risks of systems. Due to their serious impact we argue that they should be included, assessed and managed alongside more 'normal' risks.

Summary Table

Type	Brief Explanation	Icon
Black Swan Data	Data that is totally unexpected by those receiving it (i.e. 'out of the blue') and has huge (detrimental) impact, but in retrospect should have been anticipated.	
Perfect Storm Data	Combinations, sets or occurrences of data (or sets of absences of data) that would never have been thought possible to occur together, and when they do have a large and undesirable impact.	
Pudding Lane Data	Data that is unknowingly and unexpectedly critical to the whole operation and if missing or incorrect in some way, has a dramatic and negative impact. After the event it may be obvious that it was critical.	
Dragon King Data	Data that might have been foreseen, but leads to an unexpected and explosive escalation with major impact ('things get rapidly out of control').	

References

[1] C-130 Crew Gets a Rude Shock When They Fly Their Plane Below Sea Level, https://www.popularmechanics.com/military/aviation/a26598/c-130-sea-level-dead-sea, accessed August 2022

[2] The Perfect Storm, https://www.imdb.com/title/tt0177971, accessed August 2022

[3] The Perfect Storm: A True Story Of Man Against The Sea, Sebastian Junger, https://www.amazon.co.uk/Perfect-Storm-True-Story-Against/dp/0007230060, accessed August 2022

[4] NCSC Alert: Apache Log4j vulnerabilities, https://www.ncsc.gov.uk/news/apache-log4j-vulnerability, accessed August 2022

[5] A China Airlines Airbus A330 Lost Primary Computers After Landing, https://simpleflying.com/china-airlines-a330-lost-computers, accessed August 2022

[6] The Diary of Samuel Pepys, https://www.pepysdiary.com/diary/1666/09, accessed August 2022

[7] 2015 Seville Airbus A400M crash, https://en.wikipedia.org/wiki/2015_Seville_Airbus_A400M_crash, accessed August 2022

[8] PHE statement on delayed reporting of COVID-19 cases, https://www.gov.uk/government/news/phe-statement-on-delayed-reporting-of-covid-19-cases, accessed August 2022

[9] Quotes of famous people, https://quotepark.com/quotes/1829009-michael-fish-earlier-on-today-apparently-a-woman-rung-the-bbc, accessed August 2022

Image attributions
Top image © Jpldesigns | Dreamstime.com
C-130: © Mr Paul Hanley | Dreamstime.com
A400M © Abhishek Singh | Dreamstime.com
Storm © Zenobillis | Dreamstime.com
Fire © Eric Flamant | Dreamstime.com
Swan © Jerome Whittingham | Dreamstime.com
swan icon: © Skalapendra | Dreamstime.com
Michael Fish from Wikipedia published under the fair use license
Covid chart: data.gov.uk published under the Government OGL license v3.0
storm icon: © Андрей Морозюк | Dreamstime.com
pudding icon: © Amonrat Rungreangfangsai | Dreamstime.com
dragon icon: © Kuzzie | Dreamstime.com
binary code: © Webtechops Llc | Dreamstime.com
Log4J logo © Apache Software Foundation,licensed under: https://www.apache.org/licenses/LICENSE-2.0

Mike Parsons, SCSC DSIWG Chair

Mike is the SCSC Director and Events Coordinator. He also leads the SCSC Service Assurance, Data Safety Initiative and the new Systems Approach to Safety of the Environment Working Groups. He is currently a researcher at the University of York working on the Assuring Autonomy International Programme (AAIP). He has been in the business of safety since 1989.

Paul Hampton, SCSC Newsletter Editor

Paul is a Chartered Engineer with over 30 years' experience in IT. He has spent 15 of those designing and developing enterprise systems in many diverse sectors and the remainder involved in system safety in a variety of capacities including: safety engineering, safety management, independent auditing and corporate governance and assurance.