HOW COMPLEX SYSTEMS FAIL: A SYNOPSIS

As Tony Hoare, winner of the 1980 Turing Award, once wrote, “There are two ways of constructing a software design: One way is to make it so simple that there are obviously no deficiencies, and the other way is to make it so complicated that there are no obvious deficiencies.”

Complexity grows in conjunction with advancements in technology due to the nature of advancement itself. The software world is particularly cognizant of this fact due to the way programs are built upon each other to perform increasingly specialized roles. The layers of complexity created by even a simple program, such as a word processor, are astoundingly deep when you drill all the way down to the OS and machine layer.

The inevitability of complexity contributes to the importance of understanding how complex systems work, and how they fail. This latter tendency of complex systems is the focus of Richard Cook’s *How Complex Systems Fail* paper. Inside this fairly brief yet incredibly dense paper, Cook expresses and then expands upon 18 separate key observations on not only how complex systems fail but also why and what can be done to reduce the frequency and severity of failures.

The entire paper is worth taking the time to read and many systems design luminaries have suggested it should be mandatory reading for anyone in the industry. However, there are a few key points in the paper that are worth particular note and exploration:

#2 Complex systems are heavily and successfully defended against failure.

Cook starts off his "Short Treatise on the Nature of Failure" with the observation that systems become complex as methods are created to defend against the potential hazards of the system. In other words, layers of defense are established to fend off failures of the system and these layers of defense play a major role in increasing the complexity of the whole system.

The development of defensive layers involves multiple components such as backup systems, agent training, and policy creation. Therefore, the avoidance of failure inherently increases the complexity of any given system. The next logical inference given is that the more complex a system is, the more complex defensive measures must be.

#3 Catastrophe requires multiple failures - single point failures are not enough.

Continuing from point 2, Cook observes that layers of defense are effective at avoiding failure. As such, the only way that a system can fail catastrophically is through multiple, simultaneous points of failure. The defensive checks and balances in place ensure that an incident cannot occur as the result of a single point of failure.

The implication here is that complex system failures are complex and multi-faceted in their own right. Cook goes on to make observations relating to human psychology as a tangent of this core point where he discusses the propensity for finger-pointing. After all, it's completely natural to want to identify the reason why a failure occurred.

The issue with finger-pointing (which is prone to happen as a result of post-mortems) is that there is almost never a single "root cause" of any given incident. It's convenient to say that a plane crashed due to pilot error, but the pilot isn't in charge of inspecting the landing gear or repairing faulty instruments.

#8 Hindsight biases post-accident assessments of human performance.

It's easy to look back on the events of a failure and identify what went wrong, but that hindsight is what creates the sense that failure would clearly be the result of the given string of events leading up to it. The reality is that "6) Catastrophe is always just around the corner." As such, any action performed by an agent within the system could end up being the proverbial straw that broke the camel's back.

#10 All practitioner actions are gambles.

Because practitioners of a system are operating within a constantly changing and increasingly complex environment, every action they make has the potential for resulting in either success or failure. This is another carry-over from the layers of defense created to protect against failure.

Complex systems use layered checks and balances to avoid failure, but the system is in a constant state of flux and actors within it aren't always aware of potential points of failure in sectors of the

system that are outside of their field of view. An agent could perform an action 999 times with no issue only to have that same action result in failure on the 1000th time due to extraneous factors within the system that are outside of their own purview.

This reality is derived from the point that system failures are the result of multiple points of failure. Cook also states: "5) Complex systems run in a degraded mode." When you combine the idea that multiple points of failure are required to result in total system failure with the added complication brought on by layered defenses, you get a system that can be fully operational even under circumstances where individual points are experiencing localized failure.

#14 Change introduces new forms of failure.

A further complication thrown into the works of a complex system is the necessity for adaptation. This is especially true for modern technology due to its constant state of flux. In the pursuit of continual improvement, change is the only constant. Each change introduces new possibilities and avenues for failure.

Once again, this point loops back around into the reality of failures occurring at multiple points simultaneously. Changes may introduce localized failures that don't result in complete system failure until further down the road. This further increases the importance of having layered forms of defense against failures so that each issue can be identified and resolved before it results in catastrophic failure.

#16 Safety is a characteristic of systems and not of their components.

This point drives home the importance of adopting a DevOps mindset and culture for the entire organization, not just individual departments. Creating a safety team or buying some security software won't miraculously divert all possible failures. A system is a living organism that operates as a function of its individual parts working together.

Each agent, device, and department within the system is a part of a whole. A system's safety is in a constant state of flux as the system itself is ever-changing. This means that complex systems can only hope to avoid catastrophic failure by ensuring that each piece of the machine is working in tandem.