BLACK SWAN IN THE SERVER ROOM

Avoiding Disaster in Disaster Planning

EXECUTIVE SUMMARY

Today's enterprise depends on technology, including smoothly operating critical services like email and file servers. Although many firms take measures to protect critical systems, such measures are often costly, unreliable, and fail to provide the expected protection in the face of catastrophe.

System Logic LLC, an independent research and consulting firm focused on issues of risk and complexity, has developed several frameworks to implement robust platforms without investing massive resources. In this document, we identify the systemic components of robustness—tolerance, loose coupling, and redundancy—and use them to enumerate principles of robust platform design. We also describe the organizational fundamentals of communication and skepticism that lead to strategies for robust organizational design.

Additionally included are two case studies. One describes a sophisticated financial trading firm's efforts to provide redundancy. The other outlines the organizational basis for effective testing and describes Google's sophisticated practices in this area.

THE IMPORTANCE OF RESILIENCE

The modern business enterprise depends on technology. Businesses rely on smoothly operating email, file servers, enterprise collaboration, and specialized software in order to function. As a result, firms take precautions to eliminate Information Technology (IT) continuity risks and service interruption. These precautions include the offsite backup of data, disaster recovery sites, and multiple network connections. However, a naive approach can often leave infrastructure in the "High-Cost, Low-Reliability" state (See Figure 1), where companies invest in solutions that appear to increase robustness, but in fact do not.

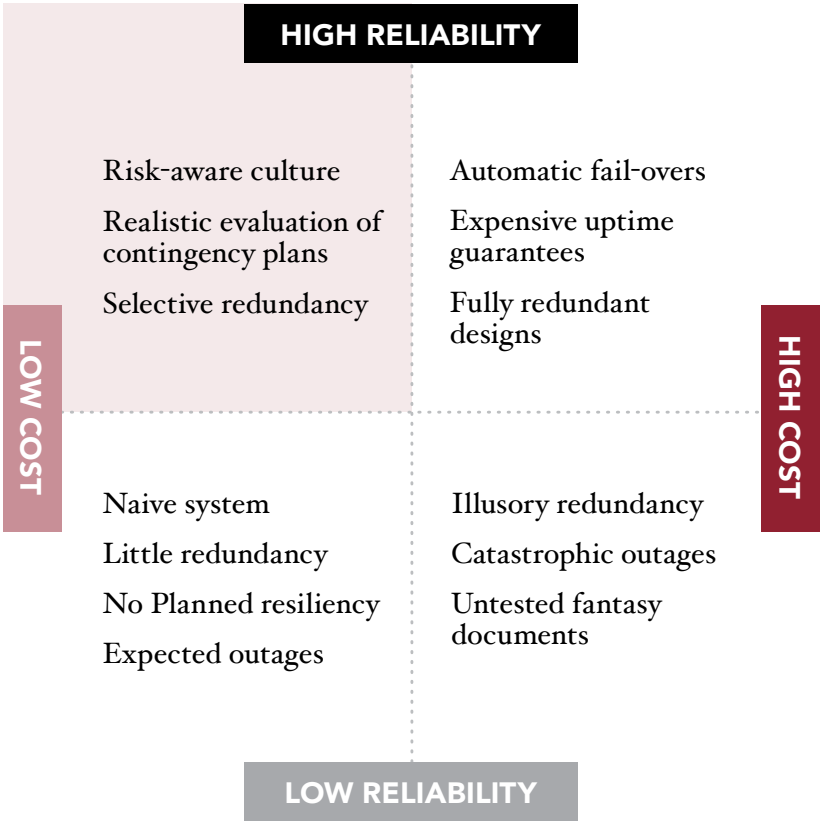

Figure 1

Unrobust platform design and continuity planning leave firms open to catastrophic risk. A disaster—natural or man made—could bring down a business for weeks or months. Bills and payroll needs continue even though insufficient access to applications, communication, and data can bring revenue-producing work to a halt.

Beyond preventing catastrophe and protecting the status quo, effective preparation confers a strategic advantage. Effectively prepared firms can continue to operate during disruptions, and can use their resources to adapt to the evolving environment and competitive landscape. Additionally, many firms' clients have increased needs during

Our framework eliminates the need for the costly duplication of every system.

and following widespread disruptions. Well-prepared firms can continue to deliver critical services to clients, establishing trust, a reputation for dependability, and cementing lasting relationships.

System Logic has developed a framework to empower information technology managers to create and implement robust platforms that provide resilience without investing massive resources. Our framework eliminates the need for the costly duplication of every system. We focus here neither on the day-to-day operation of a performant IT platform nor on the specifics of individual technologies, but rather we describe the **Systemic Components** and **Organizational Fundamentals** of robustness.

THE SYSTEMIC COMPONENTS OF ROBUSTNESS

How do we define robustness in the context of an IT platform? A robust platform is able to continue to operate in a wide array of unplanned or unexpected conditions, including power failure, the failure of network connectivity, security breaches, and natural disasters.

A robust IT platform should be as **loosely coupled** as possible. Loose coupling ensures that the failure of a single component will not affect the entire system. An organization that has a "hub-and-spoke" design with satellite offices and a large datacenter in its corporate headquarters has a tightly coupled platform—if the headquarters loses IT capabilities (no matter what the cause), then all of the satellite offices will be affected. By contrast, a firm that has distributed infrastructure across several sites will be able to continue operating even after the loss of a single site.

Additionally, loose coupling enables independence across components. For example, applications, servers, and workstations mounting shared network drives should not become unresponsive if a network drive becomes unavailable. While behavior like this can be dependent on low-level implementation details outside of an IT manager's control, the topology of the entire platform should be designed to reduce coupling to eliminate this type of occurrence.

A robust IT platform must be as **tolerant** as possible. Areas of intolerance must be identified and addressed by adding redundancy or through platform design. For example, Voice Over IP solutions are, by design, tolerant to a reduction of bandwidth. As available bandwidth decreases, protocols gracefully degrade the quality of the transmission, which persists until bandwidth is reduced so much that the call is terminated. This tolerance is inherent in the application. Tolerant components prevent failures, and loose coupling prevents failures from spreading.

For important intolerant systems, **redundancy** can provide robustness. Redundancy is a familiar concept within IT. Indeed, it sits at the center of technologies like Redundant Arrays of Inexpensive Disks (RAID) and load-balancing servers. Both technologies increase performance in normal circumstances and ensure that the loss of a subsystem (e.g., a hard disk or a web server) does not cripple an entire system.

Building a robust system is not free. Implementing the above design choices is analogous to insurance. In the short term, these elements may increase costs. This is comparable to a fixed premium on an insurance policy. However, in exchange, they reduce the likelihood of large-scale enterprise failures and protect against improbable but catastrophic events, just as insurance is designed to do.

Identifying the elements of robustness provides a framework to build the architecture of our system. We use these elements to implement a principled approach to robust platform design.

A Principled Approach to Robust Platform Design

Having identified the elements of robustness, we develop principles to anchor the design of a robust platform within real-world constraints. In a complex system, it is a fallacy to plan for the future by considering exclusively examples from the past; many catastrophic events are unprecedented or unexpected. Therefore, we advise using cohesive principles to guide the design of a robust system.

PRINCIPLES OF PLANNING AND DESIGN

Design Loose Coupling and Incorporate Tolerant Systems

Strive for loose coupling in your platform design. Limit the number of critical components whose failure will bring down your whole system, and ensure that sites and servers operate independently, so an isolated failure does not rapidly cascade through your system.

Choose tolerant subsystems that respond to failure by degrading their performance gracefully or by utilizing internal sources of redundancy, like RAID.

Plan Inherent Redundancy

Inherent redundancy is intrinsic to a system's design, not "bolted on" later.

When planning a disaster recovery site, consider ways to integrate

In a complex system, it is a fallacy to plan for the future by considering exclusively examples from the past.

its services into the day-to-day operation of the business. This serves two purposes. First, it will increase overall capacity by utilizing resources otherwise held in reserve for disaster recovery. Second, it will ensure that the recovery site is well functioning and continually tested.

For example, load-balancing web servers can be inherently redundant if they run independently and each can stand in for the other while serving pages. Should one server fail, the system merely stops sending traffic to the failed server. The remaining server continues to serve pages with the only change being a higher load.

While redundancy that is not part of the original design can prevent catastrophic failures, it also has the capacity to cause such a failure by increasing coupling and the overall complexity of the platform.

Identify Potential Common-Mode Failures

Identify common-mode elements that your infrastructure cannot function without. Common-mode failures have a widespread effect beyond the failure of the element itself. For example, a loss of power, telecommunications circuits, or chilled water will affect every server and computer at a site. Identify these elements and consider how redundancy can ensure their functioning.

Avoid Illusory Redundancy

Having identified common-mode failures, many companies elect to use redundancy to protect these critical elements. Backup batteries, generators, and telecommunications circuits are standard for applications requiring high reliability.

Ensure that your redundant systems do not suffer from **illusory redundancy**, where a redundant component is unexpectedly vulnerable to the risk against which it is intended to protect. Illusory redundancy is often hard to detect given the complexity of modern IT solutions. For example, an "independent" backup telecommunications circuit should not be buried in the same location as the primary circuit and should run in a separate conduit on site.

Acknowledge Complexity and Correlated Risks

Imagine trying to protect your facility against a widespread loss of power. The simple solution involves batteries, a generator, and a fuel reserve to run the generator. This solves the primary effect of the loss of power.

To allow for secondary effects, consider that many root causes of power failure involve natural disasters like a hurricane, blizzard, or flooding. These disasters can affect large areas for extended periods

of time, and the effects of such a disruption can include difficulties in communication and transportation logistics. Employees' cell phones and internet may not function. Fuel shortages, the closure of public transportation, and traffic jams may limit their ability to move.

In this example, instead of assuming that having a generator means that you will be up and running, incorporate these constraints into a realistic view of the resilience of your system. If you require more resilience, consider incorporating other forms of protection, like broader geographic diversity and spare capacity across your sites, and develop a plan ahead of time to utilize that spare capacity to ensure continuity.

Plan Selective Redundancy

Planning selective redundancy will allow critical components of your system to function during critical times. Identify primary systems that should be ready to go immediately following a disaster and ensure those systems are properly replicated. This will increase resilience and reduce costs by limiting replication to a subset of components. It will also increase reliability by simplifying the system. Plan for systems useful for crisis response, like phone and email, to be available immediately.

Instead of replicating every system that could possibly be needed, have **spare capacity**—servers and bandwidth that can be flexibly allocated and are not needed for your primary systems—in your disaster recovery site for ancillary systems that can be brought up on an on-demand basis.

Finally, selective redundancy increases flexibility and responsiveness. When disaster recovery is being used, it means that there is a disruption in the operation of the business, and the ability to deploy systems as-needed is beneficial.

Consider the Cloud

By outsourcing the hosting of applications and data to the "cloud," companies can decorrelate operations from physical infrastructure. Because of this, the cloud has a different risk-profile than locally hosted servers.

For the cloud to be a viable solution, redundant connectivity is paramount. If critical services are hosted in the cloud, then the office always needs to be connected to the internet. For most businesses this is already the case.

A widely distributed cloud solution, where data is replicated across many data centers, offers robustness to local disturbances in connectivity and continuity. Even if your office has no power, critical cloud-hosted services will likely be unaffected, giving employees access to the extent that they can connect to the internet from off-site.

Even if your office depends on on-site applications and storage, access to cloud-hosted services can be an excellent backup solution. Google Drive, (née Google Docs), part of the standard business Google Apps rollout, provides excellent collaboration, good word processing, and basic spreadsheet capabilities. In addition, Google+ enables business users to hold multi-way video conferences, useful if transportation has been disrupted. Cloud-hosted email provides seamless communication even when physical infrastructure has been disrupted.

The downside of cloud-hosting is that events outside of your control may adversely affect continuity. In the absence of redundancy and loose coupling in the service provider's hosting solution, an interruption to service in one place could lead to a total loss of service.

UNEXPECTED CORRELATION AND ILLUSORY REDUNDANCY

In order to improve business continuity in a highly competitive, highly automated financial trading environment, a sophisticated electronic trading firm installed a generator in its primary facility within its corporate offices in New York City. The generator was designed to power the data center and critical parts of the office in the event of a blackout.

Due to a lack of access rights, the firm could not install the generator on a sub-roof overhang. Instead, the company leased space in the basement parking garage and built a room for the generator, diesel fuel storage, and associated machinery. This necessitated additional ventilation and conduit infrastructure, increasing the overall cost of the project.

Flooding was a problem in the generator's basement location. These complications were being addressed when the first opportunity to utilize the backup power approached — Hurricane Irene. The hurricane was downgraded to a tropical storm and did not bring the feared storm surges to New York City, but the generator in the already flood-prone basement would likely not have provided the protection for which it was installed. [1]

Though it was prone to flooding, the generator was able to provide backup power in situations without the risk of rising water, like blackouts on hot summer days. However, an additional complication meant the backup power was not quite as useful as it first appeared. The company's data center, within its office, was cooled by chilled water.

Ongoing negotiations between the landlord and a key tenant meant that the building's chilled water distribution system was not covered by backup generators. Even as the firms' own generator provided backup power to its office, the pumps circulating the chilled water required to cool the data center would not function, and the data center would need to be shut down.

[1] Indeed, as an update, the basement completely flooded during Hurricane Sandy and the building was unable to be occupied for weeks. The firm ran its operations from a dedicated disaster-recovery site.

The firm invested significant resources to implement illusory redundancy. Redundancy was not included in the system's original design. The operating environment was complex. Flooding was correlated with power outages. The firm did not have end-to-end control of their infrastructure and the siting of the backup systems. Because of these complications and the lack of true redundancy they caused, the company is currently evaluating the removal of its mission-critical infrastructure from its headquarters space to multiple specialized data centers. Even very tech-savvy firms face significant challenges implementing a resilient system.

THE ORGANIZATIONAL FUNDAMENTALS OF ROBUSTNESS

Loose coupling, tolerance, and redundancy are the systemic components of robustness. To incorporate these components into a robust platform design and continuing operations, an organization needs to employ the tools of **communication** and **skepticism**.

Communication is the effective sharing of information across groups and throughout the hierarchy of an organization. Organizations that communicate well enable and encourage individuals and teams to share pertinent information with relevant decision makers quickly and clearly. Effective communication results in organizational improvement because mistakes and near-misses provide rich ground for learning.

Skepticism—recognizing the vulnerability of our assumptions—reduces reliance on untested assumptions and oversimplification in complex environments. Fostering skepticism in both operations and design helps eliminate overconfidence along with the assumption that designs and processes will "just work" when difficulty is encountered.

ROBUST ORGANIZATIONAL DESIGN

Make Genuine Decisions

To make decisions for robust platform design, one has to confront ambiguous, complex situations with different potential failures. Studies of decision making teach us that people substitute simple models for complex situations when faced with hard decisions with ambiguous factors. Consider, for example, a decision between an internally hosted or cloud–based website. A key question is: Which system will be more reliable? It's easy to subconsciously substitute this question for an easier one: Over which system do we have more control?

Answering the substitute question is easy. We think of the internally hosted solution, which we entirely control, as the more

reliable of the solutions. But this cognitive sleight of hand obscures many factors.

The cloud solution includes load balancing, seamless failover sites, redundant back–end databases, and a team of site–reliability engineers. By making an authentic comparison with a typical in-house solution of one web server inww the headquarters data center and an administrator that has many other duties, it is clear that the substitute question of control does not lead to reliability. Recognize and resist the tendency to substitute simpler questions.

Encourage Communication and Learn from Mistakes

Robustness is weakened by **fragmentation of knowledge.** The complexity of the modern IT platform means that no one person is an expert in every aspect of its design. For example, the decisions of a networking specialist, made when configuring a particular router, may influence the operation of a network–attached storage device or an application server. More generally, changes affect robustness. The second-order consequences of a particular implementation may not be entirely understood by any one group.

To combat fragmentation of knowledge, encourage open communication channels across groups and to management. Legitimize and model participation and discussion as part of the process of decision making, and encourage the flow of news, good and bad. Especially encourage the flow of bad news to managers, so they can employ resources to correct problems. Where hierarchy exists, it can distort communication so the experts making day-to-day decisions are reluctant to deliver or emphasize bad news, especially during times of stress.

To work against the reluctance to deliver bad news, earn trust by openly discussing mistakes and **near misses**, especially one's own. This will help alleviate stigma and fragmentation of knowledge, and will promote organizational learning.

Acknowledge Imperfections

Even a well-designed platform will not be able to handle every contingency. Uncertainty and a lack of guarantees can be frustrating, but rather than craft **fantasy documents** that purport that the system will be operational through every potential catastrophe, acknowledge the imperfections inherent in your design.

Ignoring design or implementation flaws by making optimistic assumptions erodes skepticism and damages robustness. It shields key decision makers from committing resources to reduce the risks the organization faces. It is better to have a realistic portrait of the

potential continuity risks than falsely believe every contingency has been successfully addressed.

Employ Designated Skeptics

Humans do not operate as perfect rational decision makers. "Bounded rationality" is the term that behavioral economists like Nobel laureate Daniel Kahneman and his longtime collaborator Amos Tversky use to describe their understanding of the cognitive shortcuts that people habitually make when faced with non-trivial cognitive tasks. [2]

[2] For more of Kahneman's insights, see his recent book *Thinking, Fast and Slow* (2011).

One of these shortcuts is the tendency to look for facts that confirm an existing world view. This well documented **confirmation bias** means that humans tend to discard information that disagrees with their mental model.

In order to combat confirmation bias, groups should intentionally seek out disconfirming information that challenge their assumptions. Designate a group or an individual as a **blocker**, whose job it is to develop and defend worst-case scenarios that may need to be incorporated in your resilient platform. Blockers should evaluate claims of redundancy and evaluate the contingency plans and raise the inevitable imperfections of the contingency plans.

REALISTICALLY EVALUATING CONTINGENCY PLANS

Realistically evaluating contingency plans requires considering both systemic and organizational factors.

Any contingency plan to combat site loss should be evaluated regularly and in as close to realistic conditions as possible. Even with decoupled and redundant designs, later additions to a platform will add interactions where none were planned. Simulate the failure of redundant components (possibly during non-production times) and see if the system still functions as intended. This is the conceptual equivalent of white-hat, "penetration testing," but applied to the non-security aspects of the IT platform.

Have a blocker incorporate imperfections into the development and execution of contingency plans, and assume that more than one thing will go wrong. For example, on the day that the building loses power, assume that employees with infrastructure knowledge are out of town. Or perhaps power loss is due to a storm, and key employees have lost their home internet and with it the ability to remotely access the office to shut down or reconfigure systems. The specific imperfections are less important than overcoming the assumption that everything will function "as expected." The plan should be able to function when it's not the finest hour for the organization.

Google has honed this practice to a fine edge with the operations of its Site Reliability Engineers (SREs). SREs, often embedded within product groups, have many functions, including automating the proper responses to failures so that reliability is maintained despite the loss of a component.

In addition, SREs act as professional skeptics and run Disaster Recovery Testing (DiRT), an elaborate simulation that disrupts infrastructure and demands the attention of incident managers to respond to the disruption, which can affect running services. According to the Googler who heads up the annual DiRT exercises, his job is to develop tests that expose systemic weakness.

The practice of discovering and exposing systemic weakness should be explicitly encouraged and rewarded. While it does not make sense for every company to dedicate the same technical resources to reliability that Google does, testing, probing and skepticism can be carried out on a smaller scale. You do not have to have the same technology as Google to think like a Googler.

CONCLUSION

Nothing is more harmful to a business than a catastrophic disruption from an unexpected source. Even technologically sophisticated firms with formal contingency plans may have unknowingly implemented costly but unreliable systems that fail to protect them from external shocks. These measures may fail because of systemic oversights like illusory redundancy or organizational failures such as reliance on untested assumptions and fantasy documents.

System Logic has developed a framework that outlines how to create a resilient IT platform without an investment of massive resources. This resilience is based on the systemic components of robustness and the organizational fundamentals of communication and skepticism.

System Logic believes that any organization can design a robust platform and adopt the organizational fundamentals required for resilience. To further this goal, we are available to provide an outside perspective, train team members to think like risk experts, and coach managers on building communicative and skeptical teams. The resulting resilience protects an organization from unexpected disruptions and serves as a source of strategic advantage, both for themselves and their clients.